ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ & ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

Ανάπτυξη λογισμικού για μοντέλο επιτραπέζιου παιχνιδιού

Φοιτητής: Σπηλιώτης Εμμανουήλ
ΑΜ: 46088

Επιβλέπων Καθηγητής

ΔΗΜΗΤΡΙΟΣ ΜΕΤΑΦΑΣ
Επίκουρος Καθηγητής

ΑΘΗΝΑ-ΑΙΓΑΛΕΩ, ΦΕΒΡΟΥΑΡΙΟΣ 2023

UNIVERSITY OF WEST ATTICA
FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Diploma Thesis

Developing Software for board game

Student: Spiliotis Emmanouil
Registration Number: 46088

Supervisor

DIMITRIOS METAFAS
Associate Professor

ATHENS-EGALEO, FEBRUARY 2023

Η Διπλωματική Εργασία έγινε αποδεκτή και βαθμολογήθηκε από την εξής τριμελή επιτροπή:

(Ονοματεπώνυμο), (βαθμίδα)	(Ονοματεπώνυμο), (βαθμίδα)	(Ονοματεπώνυμο), (βαθμίδα)
Μετάφας Δημήτριος Επίκουρος Καθηγητής (Υπογραφή)	Ραγκούση Μαρία Καθηγήτρια (Υπογραφή)	Ζαχαριάδου Αικατερίνη - Στυλιανή Καθηγήτρια (Υπογραφή)
Dimitrios Metafas Digitally signed by Dimitrios Metafas Date: 2023.03.26 20:40:21 +03'00'	Maria Ragkousi Digitally signed by Maria Ragkousi Date: 2023.03.22 09:00:20 +02'00'	Aikaterini-Styliani Zachariadou Digitally signed by Aikaterini-Styliani Zachariadou Date: 2023.03.21 22:19:08 +02'00'

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος/η …Σπηλιώτης Εμμανουήλ…. του…Γεωργίου.., με αριθμό μητρώου …46088…… φοιτητής/τρια του Πανεπιστημίου Δυτικής Αττικής της Σχολής ΜΗΧΑΝΙΚΩΝ του Τμήματος ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ,

δηλώνω υπεύθυνα ότι:

«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.
Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου.

Ο Δηλών
Εμμανουήλ Σπηλιώτης

Υπογραφή

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω θερμά το κ. Δημήτρη Μετάφα για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα αλλά και την αμέριστη καθοδήγηση του καθ' όλη τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.

Περίληψη

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια του προγράμματος σπουδών της Σχολής Ηλεκτρολόγων & Ηλεκτρονικών Μηχανικών του Πανεπιστημίου Δυτικής Αττικής. Αντικείμενο μελέτης της είναι η ανάπτυξη κώδικα λογισμικού σε γλώσσα αντικειμενοστραφούς προγραμματισμού Java που αφορά το επιτραπέζιο παιχνίδι διαχείρισης πόρων το οποίο εμπορικά είναι γνωστό ως Le Havre(the Harbor) του Uwe Rosenberg και περιλαμβάνει επίσης αλγόριθμους τεχνιτής νοημοσύνης. Το παιχνίδι αυτό παίζεται από έναν έως πέντε παίκτες, ενώ στη συγκεκριμένη εργασία η ροή του παιχνιδιού περιλαμβάνει δύο παίκτες ανάλογης τεχνητής νοημοσύνης η οποία θα τους δίδεται με βάση τα χαρακτηριστικά τους, τους πόρους που συλλέγουν κατά τη διάρκεια του παιχνιδιού ή με το πως εξελίσσεται το παιχνίδι. Για τη σύνταξη του κώδικα συντάχθηκε με χρήση του Apache Netbeans IDE.

Λέξεις – κλειδιά

Αντικειμενοστραφής προγραμματισμός, Τεχνητή Νοημοσύνη, Java, Le Havre

Abstract

This dissertation was prepared as part of the study program of the School of Electrical and Electronic Engineers at the University of West Attica. The subject of the study is the development of software code in the Java programming language that concerns the resource management board game known commercially as Le Havre, designed by Uwe Rosenberg, and also includes artificial intelligence algorithms. This game is played by one to five players, while in this particular work, the flow of the game involves two players of similar artificial intelligence, which will be given to them based on their characteristics, the resources they collect during the game, or how the game progresses. Apache Netbeans IDE was used to write the code.

Keywords

Artificial Intelligence , IDE , Java, Le Havre

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Πινάκων

ΕΙΣΑΓΩΓΗ

Σκοπός και στόχοι

Σκοπός αυτής της διπλωματικής εργασίας είναι οι ακόλουθοι. Η δημιουργία ενός βασικού σκελετού για τη μοντελοποίηση αρκετών στοιχείων του παιχνιδιού όσο αυτό κατέστη δυνατό. Η προσθήκη ευφυίας στους παίκτες που δημιουργούνται και ενεργούν στο βασικό σκελετό του παιχνιδιού που αναπτύχθηκε. Η Εξαγωγή δεδομένων, παρατηρήσεων και τυχόν συμπερασμάτων για την πορεία των παικτών Μεθοδολογία.

Αντικείμενο διπλωματικής εργασίας

Το αντικείμενο της συγκεκριμένης εργασίας αφορά την ανάπτυξη ενός λογισμικού που προσομοιώνει τους κανόνες παιχνιδιού, καλύπτει ένα μέρος στρατηγικών καθώς και τις αποφάσεις που παίρνονται από τους παίκτες όχι απλώς με τυχαίο τρόπο αλλά και με βάση τη ροή εξέλιξης του παιχνιδιού. Δηλαδή, το κύριο μέλημα δεν είναι να ανταποκρίνονται οι παίκτες μόνο στους κανόνες του παιχνιδιού αλλά και στις αποφάσεις που πρέπει να λαμβάνουν έτσι όπως εξελίσσεται το παιχνίδι. Δεν υπάρχει κάποια συνταγή που μας οδηγεί στο βέλτιστο τρόπο παιξίματος του παιχνιδιού από τον υπολογιστή αλλά μόνο προσεγγιστικά γιατί ο τρόπος και οι τεχνικές που ο κώδικας τρέχει το παιχνίδι βασίζονται στον τρόπο που σκέφτεται ο άνθρωπος που παίζει το παιχνίδι. Όπως και στα βιντεοπαιχνίδια, της αγοράς στα οποία πάντα υπάρχει περιθώριο βελτίωσης και προσθήκη νέων στοιχείων

Δομή

Η εργασία αυτή αποτελείται από έξι κεφάλαια. Στο πρώτο κεφάλαιο περιγράφεται το περιβάλλον και τα αντικείμενα του παιχνιδιού καθώς και οι κανόνες του. Το δεύτερο κεφάλαιο ασχολείται με την περιγραφή σε κώδικα βασικών αντικειμένων του παιχνιδιού και στο επόμενο κεφάλαιο παρατίθενται παραδείγματα τεχνητής ευφυίας. Οι κλάσεις που αναπτύχθηκαν αποτελούν το τέταρτο κεφάλαιο της διπλωματικής εργασίας, ενώ στο πέμπτο κεφάλαιο κεφάλαιο έχουν συγκεντρωθεί σενάρια λειτουργίας του παιχνιδιού. Η διπλωματική ολοκληρώνεται με το έκτο κεφάλαιο το οποίο περιέχει τα σχετικά συμπεράσματα και τις μελλοντικές προοπτικές.

1 ΚΕΦΑΛΑΙΟ 1ο : Τεχνητή νοημοσύνη

1.1 Τεχνητή νοημοσύνη

Η ιστορία της Τεχνητής Νοημοσύνης (ΤΝ) ξεκινάει από τα πρώτα βήματα της επιστήμης των υπολογιστών και της λογικής, κατά τη διάρκεια της πρώτης μισής του 20ού αιώνα[1]. Η ιδέα της ΤΝ έχει ως πυρήνα τη δυνατότητα δημιουργίας μηχανών που μπορούν να προσομοιώνουν την ανθρώπινη σκέψη. Οι πρώτες προσπάθειες στη δημιουργία της ΤΝ περιλάμβαναν την ανάπτυξη τεχνικών για τη μηχανική μετάφραση γλωσσών, την κατασκευή λογιστικών και υπολογιστικών μηχανών και τη δημιουργία απλών μορφών τεχνητής νοημοσύνης, όπως τα αυτόματα αποφάσεων. Έτσι η παρουσία της ΤΝ υπάρχει όταν μία μηχανή εκδηλώνει συμπεριφορές που ομοιάζουν με την ανθρώπινη σκέψη, είτε αυτό σχετίζεται με λήψη λογικών αποφάσεων ή μαθηματικών υπολογισμών.

Σκοπός των συστημάτων ΤΝ είναι πάντα η αξιοποίηση δυνατοτήτων επεξεργασίας δεδομένων που προσφέρει ο υπολογιστής στο ρυθμό της ανθρώπινης λογικής. Δεν είναι αρκετό να μπορεί ο υπολογιστής απλά να επεξεργάζεται έτοιμα δεδομένα αλλά να προσαρμόζεται σε περιβάλλοντα που απαιτούν επεξεργασία δεδομένων σε συνθήκες πραγματικού χρόνου. Έτσι χρειάζεται ένα πρότυπο στο οποίο η μηχανή μπορεί να στηριχθεί για να οδηγηθεί στην επίλυση προβλημάτων που περιλαμβάνουν πολλές παραμέτρους και δυναμικές αλλαγές μιμούμενη τις ανθρώπινες σκέψεις.[2]

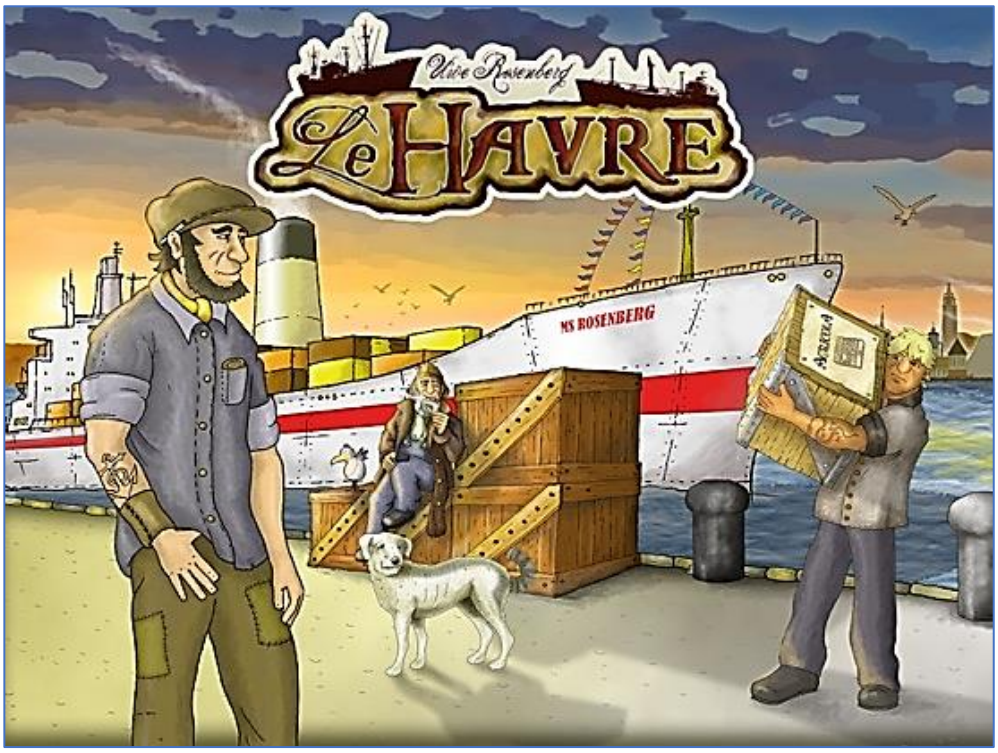
1.2 Εφαρμογές ΤΝ στη βιομηχανία παιχνιδιών

Η τεχνητή νοημοσύνη έχει μία αξιοσημείωτη παρουσία στη βιομηχανία των παιχνιδιών[3]. Τα παιχνίδια μπορούν να χρησιμοποιήσουν την ΤΝ για να δημιουργήσουν ρεαλιστικότερες εμπειρίες τόσο στο περιβάλλον του παιχνιδιού όσο και στην αλληλεπίδραση παίκτη–υπολογιστή με στόχο να βελτιώσουν το παίξιμο αυτού.

Μια από τις βασικές εφαρμογές της ΤΝ στα παιχνίδια είναι η δημιουργία επιθετικής ΤΝ που μπορεί να ανταγωνιστεί τον ανθρώπινο παίκτη. Οι εταιρείες παιχνιδιών χρησιμοποιούν διάφορες τεχνικές μάθησης μηχανής, όπως μάθηση ενισχυτική, μάθηση βαθιάς επαγωγής και μάθηση επαναλαμβανόμενης, για να δημιουργήσουν ΤΝ που μπορεί να ανταποκριθεί σε διαφορετικές καταστάσεις στο παιχνίδι. Στήριξη στην λήψη αποφάσεων: Η ΤΝ μπορεί να βοηθήσει στη λήψη αποφάσεων στα παιχνίδια. Για παράδειγμα, σε ένα παιχνίδι στρατηγικής, η ΤΝ μπορεί να βοηθήσει τους παίκτες στην ανάλυση των επιλογών τους και στην εύρεση της καλύτερης λύσης. Αυτό μπορεί να επιτευχθεί μέσω αλγορίθμων μάθησης με ενισχυτή, που μπορούν να εκπαιδευτούν να λαμβάνουν αποφάσεις σε διάφορες καταστάσεις και να βελτιώνουν τις επιδόσεις τους μέσω του συνεχούς πειραματισμού. Η ΤΝ στον τομέα της προσαρμοστικότητας μπορεί να βοηθήσει τα παιχνίδια να προσαρμόζονται στα προσωπικά χαρακτηριστικά και τις προτιμήσεις του κάθε παίκτη. Μέσω της συλλογής δεδομένων και της ανάλυσης της συμπεριφοράς του παίκτη, η ΤΝ μπορεί να παραγάγει προσαρμοστικά παιχνίδια που θα είναι πιο ενδιαφέροντα και διασκεδαστικά για τον καθένα. Συνολικά, η ΤΝ μπορεί να βοηθήσει τα παιχνίδια να παρέχουν μια καλύτερη εμπειρία παιχνιδιού για τους παίκτες, καθώς και να επεκτείνει τα όρια των δυνατοτήτων των παιχνιδιών. Αυτό έχει οδηγήσει σε έναν νέο τρόπο σκέψης στον σχεδιασμό και την ανάπτυξη παιχνιδιών, με την ενσωμάτωση ΤΝ για να βελτιωθεί η εμπειρία του παίκτη.

ΚΕΦΑΛΑΙΟ 2ο : Περιγραφή Παιχνιδιού

Στο παρόν αυτό κεφάλαιο περιγράφουμε το μοντέλο του παιχνιδιού στο οποίο αναπτύξαμε το λογισμικό μας. Το παιχνίδι με εμπορικό όνομα Le Havre έχει ως επίκεντρο το λιμάνι της Χάβρης και βασίζεται στην απόκτηση αγαθών, κτιρίων, πλοίων κλπ. με τη μορφή καρτών μέσω των οποίων ο παίκτης θα καταφέρει να διαχειριστεί με τον κατάλληλο τρόπο όσους πόρους διαθέτει η συλλέγει στη συνέχεια προκειμένου να τους μετατρέψει σε χρήματα τα οποία ονομάζονται francs. Στο τέλος του παιχνιδιού όλα τα αγαθά συμπεριλαμβανομένων και των χρημάτων μετατρέπονται σε francs ώστε να αναδειχτεί ο τελικός νικητής του παιχνιδιού.



Εικόνα 1 Le Havre board game (https://appsliced.co/app?n=le-havre-the-harbor)

1.3 Κανόνες παιχνιδιού

Στην έκδοση του παιχνιδιού με δύο παίκτες συμμετέχουν συνολικά ογδόντα έξι κάρτες[4]. Στον πίνακα που ακολουθεί περιγράφονται οι κατηγορίες των καρτών

Πίνακας 1 Κατηγορίες καρτών παιχνιδιού

Όνομα κατηγορίας	περιγραφή
starting builidngs	κτίρια που αρχικά ανήκουν στην πόλη
standrard buildings	κτίρια που αρχικά δεν ανήκουν στην πόλη
special buildings	κτίρια που αρχικά δεν ανήκουν στην πόλη
round cards	μπόνους, απολογισμός κλπ.
ship cards	αφορούν την πίσω όψη της κάρτας συγκομιδής
loan cards	κάρτα δανείου

εκτός από τις κάρτες υπάρχουν και τα αγαθά με τη μορφή πλακιδίων(tokens) που βρίσκονται στις αποθήκες(supply) και αποβάθρες(docks) του λιμανιού.

Τα αντικείμενα που συμμετέχουν στο παιχνίδι έχουν κάποια αξία, είτε σε χρήματα, είτε σε πρώτες ύλες, γεύματα είτε σε άλλες παρόμοιες ιδιότητες που χρησιμεύουν και αναγράφονται στις κάρτες ή στα πλακίδια.

Η εξέλιξη του παιχνιδιού περιλαμβάνει δεκατέσσερεις γύρους(rounds) των επτά κινήσεων(turns) ο καθένας. Σε κάθε μία από τις επτά κινήσεις οι παίκτες παίζουν εναλλάξ μέχρι να ολοκληρωθεί ο τρέχον γύρος. Οι ενέργειες που μπορούν να γίνουν από κάθε παίκτη όταν είναι η σειρά του μπορούν να επιλεγούν ανάμεσα από αυτές που συγκεντρώνονται στον παρακάτω πίνακα

Πίνακας 2 Πίνακας ενεργειών παίκτη

Περιγραφή ενέργειας	Λεπτομέριες
Ενέργεια παίκτη	επίσκεψη κτιρίου η συλλογή όλων των υλικών μίας αποβάθρας
Ενέργεια μπόνους	αγορά, πώληση, χτίσιμο κτιρίου
Απολογισμός	πληρωμή γευμάτων

Σε περιπτώσεις που ο παίκτης δεν έχει να πληρώσει το φαγητό που απαιτεί η κάρτα στο τέλος του κάθε γύρου(round card), είναι υποχρεωμένος να πληρώσει με χρήματα(francs) το ισοδύναμο ποσό. Αν δεν κατέχει ούτε χρήματα τότε οφείλει να πάρει δάνειο το οποίο θα αποπληρώνει τα φαγητά της κάρτας. Για κάθε δάνειο δίνει τέσσερα francs, και αποπληρώνεται σε πέντε francs. Στο τέλος του παιχνιδιού κάθε δάνειο που δεν έχει αποπληρωθεί από τον κάτοχο του, χρεώνει τον παίκτη επτά φράγκα από το τελικό του σκορ.

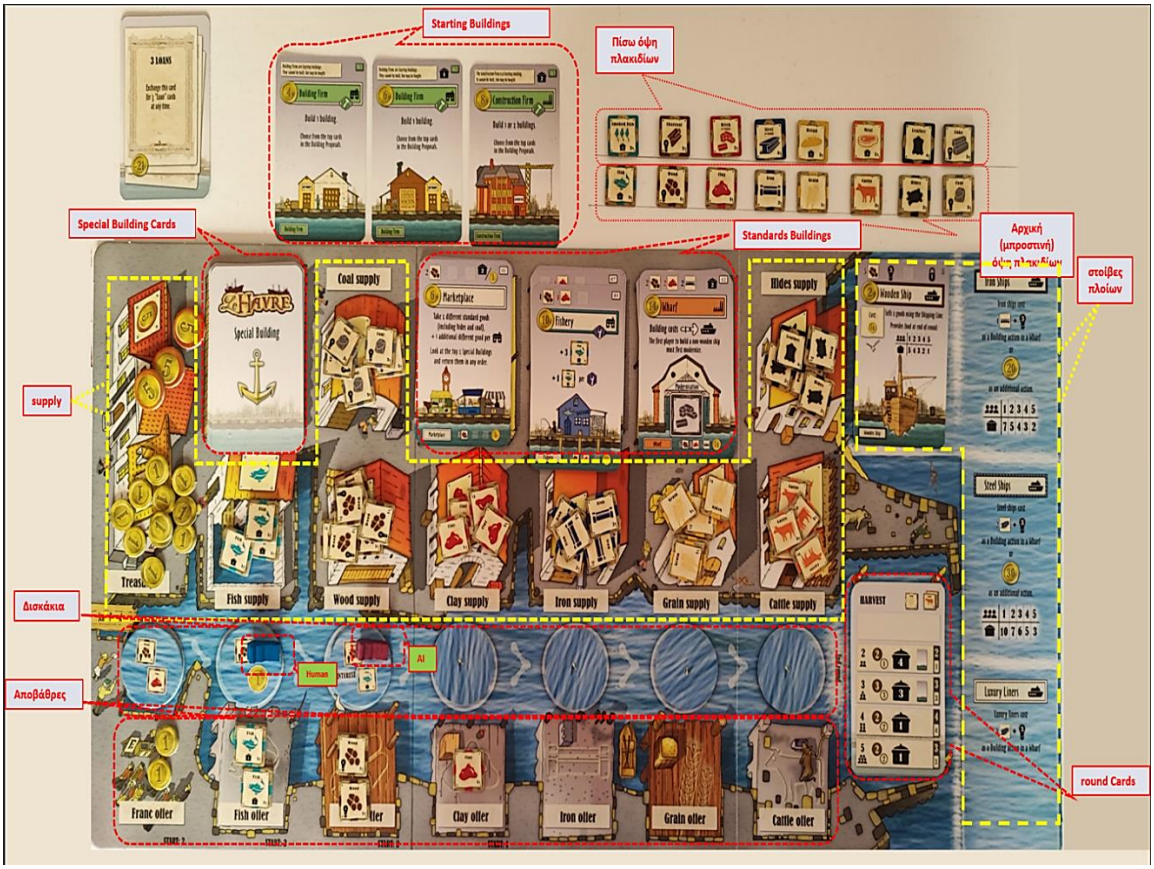
Στο τέλος του γύρου εφόσον η round card έχει HARVEST (δεκατρείς από τις δεκατέσσερεις κάρτες έχουν HARVEST για την έκδοση του παιχνιδιού που παίζεται από δύο παίκτες) ο παίκτης που έχει τουλάχιστον ένα Grain token λαμβάνει άλλο ένα Grain, και επίσης εφόσον έχει τουλάχιστον δύο Cattle λαμβάνει άλλο ένα Cattle. Ενώ στο τέλος του γύρου χτίζεται, δηλαδή απελευθερώνεται μία Special η Standard κάρτα από τη στοίβα της και διατίθεται στην πόλη, πάλι με βάση το τι δείχνει η round card που σφραγίζει το τέλος του τρέχοντος γύρου.

Οι κάρτες που ανήκουν στην πόλη είναι ήδη κτισμένες από αυτήν και άρα δε μπορούν να κτιστούν από τον παίκτη ακόμη και αν αυτός έχει τα υλικά που αυτές απαιτούν για το κτίσιμο. Οι δε κάρτες που ακόμη βρίσκονται σε στοίβες(δηλαδή οι Special και οι Standard κάρτες κτιρίων), μόνο αυτές που βρίσκονται στην κορυφή της στοίβας είναι διαθέσιμες για αγορά ή κτίσιμο από τον παίκτη(εκτός και αν υπάρχουν ορισμένες εξαιρέσεις καρτών που αναγράφουν ότι δε μπορούν να κτιστούν ούτως η άλλως). Τέλος, όταν η round card ολοκληρώνει τη διαδικασία πληρωμής, συγκομιδής(HARVEST) ή κτισίματος και σφραγίζει το κλείσιμο του γύρου τότε η ίδια γυρίζει στην πίσω όψη της και μετατρέπεται σε πλοίο το οποίο διατίθεται σε ξεχωριστή στοίβα.

Εικόνα 2 Game Round Card (εικόνα από το επιτραπέζιο παιχνίδι)

1.4 Ταμπλό παιχνιδιού

Οι δύο παίκτες από την αφετηρία προχωρούν στους δίσκους μεταθέτοντας από τις αποθήκες(supply) προς τις αποβάθρες(docks) τα υλικά που αναγράφουν οι δίσκοι κάθε φορά.



Εικόνα 3 Game Tableau (Le Havre board game επιτραπέζιο)

Ο κάθε δίσκος αναγράφει δύο αγαθά επάνω του εκτός από έναν ο οποίος περιέχει και την πληροφορία INTEREST που δηλώνει ότι άπαξ και πατήσει παίκτης πάνω του, όλοι οι παίκτες που έχουν δάνειο πληρώνουν ένα franc.

Εικόνα 4 Game Disk Token (https://www.ultraboardgames.com/le-havre/rule-clarifications.php)

Τα πλακίδια υλικών του παιχνιδιού είναι οκτώ συνολικά και χωρίζονται σε κατεργασμένα και ακατέργαστα υλικά. Η μπροστινή όψη πλακιδίου αφορά την ακατέργαστη μορφή του υλικού.

Εικόνα 5 game goods tokens (https://www.ultraboardgames.com/le-havre/rule-clarifications.php)

2 ΚΕΦΑΛΑΙΟ 2ο : Υλοποίηση

Στο συγκεκριμένο κεφάλαιο περιγράφονται αναλυτικά οι κλάσεις που αναπτύχθηκαν για το λογισμικό στο περιβάλλον Apache NetBeans[5]. Για κάθε μία κλάση αναφέρεται η λειτουργικότητα της ως στοιχείο του συνόλου της εφαρμογής.

2.1 Δημιουργία αντικειμένου δίσκου

Η κλάση Disk χρησιμοποιείται για τη δημιουργία του αντικειμένου disk του παιχνιδιού. Αναθέτονται τα χαρακτηριστικά των δίσκων μέσω του κατασκευαστή Disk. Υπάρχει μέθοδος κατάλληλη για το γύρισμα του δίσκου καθώς και μέθοδος για την εκτύπωση των πληροφοριών που αναγράφονται στο δίσκο

```
public class Disk extends Const{
    int item1,item2;
    boolean hasInterest;
    boolean Turned;
    public Disk(int item1,int item2,boolean hasInterest)
    {
        this.item1=item1;
        this.item2=item2;
        this.hasInterest=hasInterest;
        Turned=false;
    }
    public void turn() {Turned=true;}
    public String toString()
    {
        if(!Turned) return "*** ***";
        String h=new String();
        if(hasInterest) h="INTEREST";
        return name1[item1]+" "+name1[item2]+" "+h;
    }
}
```

2.2 Ανακάτεμα δίσκων

Αρχικά, οι δίσκοι τοποθετούνται με τυχαία σειρά γυρισμένοι και αποκρύβονται από τους παίκτες. Ανοίγουν μόνο όταν ο παίκτης πατήσει πρώτη φορά επάνω. Δε μπορεί ο παίκτης δηλαδή εξαρχής να προβλέψει τις δυνατές στρατηγικές που μπορούν να εφαρμοστούν μέχρι να φτάσει στο τέλος του πρώτου γύρου. Είναι σημαντικό, για λόγους στρατηγικής να παρατηρήσει κανείς με ποιο ρυθμό διατίθενται τα υλικά στις αποβάθρες κάθε φορά πριν προχωρήσει σε οποιαδήποτε ενέργεια.

Γι' αυτό το λόγο κρίθηκε απαραίτητη η δημιουργία μίας μεθόδου που ανακατεύει τους 7 δίσκους και τους τοποθετεί με τυχαία σειρά κάθε φορά που παίζεται το παιχνίδι από την αρχή.

```
public void sh(Object ob[])
    {
        int N=ob.length;
        int arr[]=new int[N];



        Object temp[]=new Object[N];
        for(int i=0;i<N;i++) temp[i]=ob[i];

        for(int i=0;i<N;i++)
        {
           int r=getRnd(N);
           boolean exists=false;
           for(int j=0;j<i;j++)
               if(arr[j]==r)
                   exists=true;
           if(exists)
               i--;
           else
               arr[i]=r;
        }

        for(int i=0;i<N;i++)
            ob[i]=temp[arr[i]];

    }
```

2.2.1 Δημιουργία των επτά δίσκων του παιχνιδιού

Οι επτά δίσκοι δημιουργούνται κατά την εκκίνηση του παιχνιδιού ενώ με την κλήση της μεθόδου sh ο κατασκευαστής Disks εξασφαλίζει το ανακάτεμα τους με τον πλέον αμερόληπτο τρόπο

```
public class Disks {
    Disk disk[]=new Disk[7];
    Shuffle Sh=new Shuffle();
    public Disks()
    {
        disk[0]=new Disk(1,2,true);
        disk[1]=new Disk(4,0,false);
        disk[2]=new Disk(2,0,false);
        disk[3]=new Disk(1,3,false);
        disk[4]=new Disk(2,6,false);
        disk[5]=new Disk(2,3,false);
        disk[6]=new Disk(1,5,false);

        Sh.sh(disk);

    }
    public String toString()
    {
        String s=new String();
        s="Disks : ";
        for(int i=0;i<7;i++)
            s+="("+ disk[i]+")\t";
        return s;
    }

}
```

Εικόνα 6 shuffled disks (https://benjaminsboardgameblog.wordpress.com/tag/strategy/)

Έτσι κάθε φορά που αρχίζει το παιχνίδι οι δίσκοι βρίσκονται αναποδογυρισμένοι με τυχαίο τρόπο

2.3 Δημιουργία πλακιδίων(tokens)

Για τα αντικείμενα του παιχνιδιού δημιουργήθηκε η κλάση Const η οποία αποθηκεύει όλα τα δεδομένα των καρτών, υλικών ή άλλες χρήσιμες πληροφορίες για τον παίκτη ή το παιχνίδι, τα οποία αργότερα θα τροφοδοτήσουν τις υπόλοιπες κλάσεις εκεί που κρίνεται απαραίτητο ώστε να δημιουργηθούν τα αντικείμενα του παιχνιδιού

Για παράδειγμα, στα υλικά αγαθά(goods) κάθε πλακίδιο έχει δύο όψεις (την ακατέργαστη και την κατεργασμένη μορφή του υλικού) όπως φαίνεται στην Εικόνα 5

```
String name1[]={"Franc", //1η όψη (ακατέργαστα πλακίδια)
                "Fish ",
                "Wood ",
                "Clay ",
                "Iron ",
                "Grain ",
                "Cattle",
                "Coal",
                "Hides"};

    String name2[]={"Franc     ", //2η όψη (κατεργασμένα πλακίδια)
                    "Smokedfish",
                    "Charcoal  ",
                    "Brick     ",
                    "Steel     ",
                    "Bread     ",
                    "Meat      ",
                    "Coke      ",
                    "Leather   "};
    // initial values of items
    int value1[]={1,0,0,0,0,0,0,0,0};
```

```
int  meal1[]={0,1,0,0,0,0,0,0,0};
int energy1[]={0,0,1,0,0,0,0,1,0};

//values of turned items
int value2[]={1,0,0,0,0,0,0,0,0};
int  meal2[]={0,2,0,0,0,2,3,0,0};
int energy2[]={0,0,3,0,0,0,0,10,0};
```

3 ΚΕΦΑΛΑΙΟ 3ο : Παραδείγματα τεχνητής ευφυίας

Εδώ παρουσιάζονται ορισμένα παραδείγματα που αφορούν τις ενέργειες και τον τρόπο δράσης των παικτών

3.1 Δημιουργία παικτών

Ιδιαίτερα σημαντική κλάση για την εξέλιξη του παιχνιδιού είναι η κλάση Player. Στην κλάση αυτή δημιουργούνται τα χαρακτηριστικά των 2 παικτών που θα παίξουν το παιχνίδι. Συγκεκριμένα, ο ένας παίκτης θα έχει το όνομα Emmanouil ενώ ο δεύτερος θα έχει το όνομα Computer. Αυτό γίνεται για να καταστεί σαφές ποιος παίκτης θα έχει περισσότερη ευφυία όσο παίζεται το παιχνίδι. Ο παίκτης Computer προφανώς είναι αυτός ο οποίος θα έχει και τις μεθόδους που θα του επιτρέπουν μεγαλύτερη ευελιξία στις αποφάσεις που θα παίρνει.

```
public Player(String name,boolean isHuman)
    {
        this.name=name;
        this.isHuman=isHuman;
        position=-1;
        loan=0;
        items.add(new Item(franc), 5);
        items.add(new Item(coal));
    }
```

Στην αρχή οι παίκτες βάσει κανόνων ξεκινούν πάντα έχοντας στη διάθεση τους πέντε francs και 1 πλακίδιο coal. Αυτά είναι τα υλικά με τα οποία πρέπει να πορευτούν και γι' αυτό το λόγο υπάρχει η μέθοδος add της κλάσης Items που φροντίζει ώστε οι παίκτες να έχουν αυτά τα υλικά στη λίστα τους αλλά και όσα υλικά καταφέρουν να συλλέξουν περαιτέρω στη συνέχεια

```
public class Items extends Const{
    ArrayList<Item> items=new ArrayList<>();
    public Items() {}
    public void add(Item item) {items.add(item);}
    public void add(Item item,int N)
    {
        for(int i=0;i<N;i++) items.add(item);
    }
```

Στην κλάση playGame οι παίκτες του παιχνιδιού δημιουργούνται και μάλιστα με τυχαία σειρά ώστε πάλι να υπάρχει αμεροληψία στο ποιος ξεκινά πρώτος και όταν τελειώνει τη σειρά του ο τρέχων παίκτης να εναλλάσσεται επιτρέποντας στον άλλον παίκτη να ακολουθήσει αμέσως μετά.

```
    public void createPlayers()
    {
        currentPlayer=randomStartPlayer();
        player[0]=new Player("Emmanouil",true);
        player[1]=new Player("Computer",false);
    }
```

```
    public int alterPlayer(int currentPlayer) {return (currentPlayer==1)?0:1;}
```

3.2 Επιλογή αντικειμένων από τον παίκτη

Οι παίκτες επιλέγουν τυχαία αντικείμενα από την αποβάθρα με χρήση μαθηματικής μεθόδου στατιστικών βαρών με τη βοήθεια της μεθόδου getRndWeight(int w[ ]) της κλάσης Shuffle:

```
public void getItemsFromDock(Dock dock)
    {
       while(true)
       {

            int dockNumber=sh.getRndWeight(weight);
            if (dock.hasItems(dockNumber))
            {
                items.add(new Item(dockNumber), dock.getItems(dockNumber));
                System.out.println("I'm "+name+", my arraylist
has:");items.groupedItems();

System.out.println("Franc:"+items.getTotalMoney()+",Energy:"+items.getTotalEnerg
y()+",Meal:"+items.getTotalMeals()+" Loans:"+loan);
                break;
            }
       }
    }
```

Η αποβάθρα επιλέγεται ανάμεσα σε αυτές που περιέχουν αντικείμενα
και η τυχαία επιλογή γίνεται με χρήση πίνακα βαρών(weight)

```
public class Const
{

    //πιθανότητες επιλογής πλακιδίου
    int weight[]={1,1,1,1,1,1,1};
```

Ανάλογα με τα περιεχόμενα του πίνακα των βαρών τέτοια είναι και η πιθανότητα επιλογής μιας αποβάθρας πλακιδίων από τον παίκτη. Αν αλλαχθούν τα περιεχόμενα του πίνακα αλλάζει και ο τυχαίος τρόπος επιλογής και οι αποβάθρες επιλέγονται με πιο επιλεκτικό τρόπο πλέον.

Η τιμή που επιστρέφει η μέθοδος getRndWeight(int w[ ]) ακολουθεί την κατανομή βαρών όπως αυτή περιγράφεται στον πίνακα w[ ]

```
public int getRndWeight(int w[])

    {
        int len=w.length;
        int limit[]=new int[len];
        limit[0]=w[0];
        int sum=0;
        for(int i=0;i<len;i++) sum+=w[i];
        for(int i=1;i<len;i++) limit[i]=limit[i-1]+w[i];

        int rnd=(int) (Math.random()*sum);

        for(int i=0;i<len;i++) if(rnd<limit[i]) return i;

        return len-1;

    }

}
```

3.3 Αποπληρωμή των γευμάτων

Ένα άλλο παράδειγμα ευφυίας που εισήχθη είναι ο τρόπος με τον οποίο οι παίκτες καταφέρνουν να αποπληρώνουν τα γεύματα τους:

```
public boolean canPayMeals(int meals)
    {

        return (items.getTotalMeals()+items.getTotalMoney())>=meals;
    }
```

```
    public void payMeals(int meals)
    {
        System.out.println("\t\t\t\t and paying "+meals+" meals");
        if(!canPayMeals(meals))
        {

            getLoans((meals-items.getTotalMeals()-
items.getTotalMoney())/4+((meals-items.getTotalMeals()-
items.getTotalMoney())%4==0?0:1));
```

```
        }

        if (items.getTotalMeals()>meals)
        {
            items.payMeals(meals);
        }
        else
        {
            items.payFrancs(meals-items.getTotalMeals());
            items.payMeals(items.getTotalMeals());
        }

    }
```

Ο παίκτης δίνει προτεραιότητα αποπληρωμής με γεύματα αντί να ξοδέψει χρήματα. Στο παιχνίδι όταν ζητείται από τον παίκτη να πληρώσει με γεύματα, επιτρέπεται αυτός να δώσει ακόμα και χρήμα ώστε να πραγματοποιηθεί η αποπληρωμή, όμως ο παίκτης επιλέγει να μη δώσει χρήμα αλλά γεύματα γιατί τα χρήματα είναι σημαντικά και μετρούν στο τελικό σκορ του παιχνιδιού, έτσι ο παίκτης δε ρισκάρει να ξοδέψει τα χρήματα του αλλά τα γεύματα που έχει στη διάθεση του. Μόνο όταν του τελειώσουν τα φράγκα ο παίκτης επιλέγει να πληρώσει χρήματα.

Όταν ο παίκτης δεν έχει ούτε χρήματα να πληρώσει τότε ο ίδιος παίρνει δάνειο:

```
public void getLoans(int N)
    {
        System.out.println("----- Player:"+name+" has just got "+N+"
loan"+((N==1)?"":"s")+"!");
        loan+=N;
        items.add(new Item(franc),4*N);
        counter++;
    }
```

3.4 Αποπληρωμή βάσει παικτών βάση δανείων

Άλλο ένα παράδειγμα είναι το πως ο παίκτης διακρίνει πότε πρέπει να πληρώσει με χρήμα όταν τυχαίνει σε δισκάκι που έχει το χαρακτηριστικό INTEREST

```
  public Disk(int item1,int item2,boolean hasInterest)
    {
        this.item1=item1;
        this.item2=item2;
        this.hasInterest=hasInterest;
        Turned=false;
    }
    public void turn() {Turned=true;}
    public String toString()
    {
        if(!Turned) return "*** ***";
        String h=new String();
        if(hasInterest) h="INTEREST";
        return name1[item1]+" "+name1[item2]+" "+h;
    }
}
```

Πραγματοποιείται έλεγχος για τα δάνεια του παίκτη

```
    public void interest()
    {
        System.out.println("player "+name+" is in interest check!");
        if(loan>0)
        {
            System.out.println("\t paying 1F for interest");
            if(items.getTotalMoney()==0)
                getLoans(1);
            items.payFrancs(1);
        }
        else System.out.println("\t Nothing to pay!");
    }
```

```
public void interest(boolean hasInterest)
    {
        if(hasInterest)
        {
            player[Human].interest();player[Computer].interest();
```

```
        }
    }
```

Όπως δείχνουν και τα αποτελέσματα του αρχείου καταγραφής:

```
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:5
            *Clay :1
            *Coal:1
Franc:5,Energy:1,Meal:0 Loans:0

I'm Computer, my arraylist has:
            **** Items ****
            *Franc:8
            *Coal:1
Franc:8,Energy:1,Meal:0 Loans:0



player Emmanouil is in interest check!
     Nothing to pay!
player Computer is in interest check!
     Nothing to pay!
```

3.5 Αποπληρωμή δανείου

Επίσης η αποπληρωμή δανείων από τους παίκτες δε γίνεται αψήφιστα αλλά οι ίδιοι γνωρίζουν κάθε πότε επιλέγουν να αποπληρώνουν

```
  public void PlayerPaysLoan(int turn)
      {
        int count=0;
        int weightOfComputerpaying=80;
        int weightOfHumanpaying=50;
        Shuffle s= new Shuffle();
      if(player[1].loan>=player[0].loan && s.getRnd(100)<weightOfComputerpaying)
      {
        if(player[1].payLoan()) {count++;}
        if(count>0)
        System.out.println("Player "+player[1].name+" paid "+count+"
Loan"+((count==1)?"":"s")+"!!!");
      }

      if(player[0].loan>=player[1].loan && s.getRnd(100)<weightOfHumanpaying)
      {
        if(player[1].payLoan()) {count++;}
        if(count>0)
        System.out.println("Player "+player[0].name+" paid "+count+"
Loan"+((count==1)?"":"s")+"!!!");
      }

     }
```

Ένας παίκτης υπάρχει πιθανότητα να επιλέξει να κάνει αποπληρωμή δανείων όταν δει ότι ο αντίπαλος έχει λιγότερα ή ίσα δάνεια σε αριθμό από αυτόν. Αυτό δε συμβαίνει πάντα, αλλά τις περισσότερες φορές

Συγκεκριμένα, ο παίκτης Computer επιλέγει με μεγαλύτερη πιθανότητα να αποπληρώσει δάνειο σε σχέση με τον παίκτη Human όταν δει ότι ο Human έχει λιγότερα ή ίδια δάνεια

Όπως φαίνεται και στο αρχείο καταγραφής:

```
$$$$$$$$$$$$$$$$$$$$ END OF ROUND 7 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:23
TOTAL LOANS OF Emmanouil:11
TOTAL LOANS OF Computer:12


Player Computer paid 1 Loan!!!
Player Emmanouil paid 1 Loan!!!
```

Ο παίκτης Emmanouil στο τέλος του συγκεκριμένου γύρου κατέχει λιγότερα δάνεια από τον Computer.

Ο παίκτης Computer ξαφνικά βλέπει ότι έχει περισσότερα δάνεια από τον Emmanouil και αποπληρώνει 1 δάνειο, ενώ την ίδια στιγμή μόλις ο Emmanouil αντιλαμβάνεται ότι ο Computer κατάφερε να ισορροπήσει τη διαφορά των δανείων τότε την ίδια στιγμή αποφασίζει ότι πρέπει κι αυτός να πληρώσει 1 δάνειο ώστε να διατηρήσει τη διαφορά.

Σε επόμενο σημείο βλέπουμε από τα αποτελέσματα του αρχείου καταγραφής ότι οι παίκτες έχουν ίδιο αριθμό δανείων:

```
$$$$$$$$$$$$$$$$$$$$ END OF ROUND 6 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:18
TOTAL LOANS OF Emmanouil:9
TOTAL LOANS OF Computer:9


Player Computer paid 1 Loan!!!
```

```
public boolean payLoan() //αποπληρωμή δανείου
    {
        if(loan>0)
        {
            if(items.removeFrancs(5))
            {
                loan--;

                return true;

            }
        }
        return false;
    }
```

Όμως μόνο ο παίκτης Computer αποφασίζει να πληρώσει δάνειο

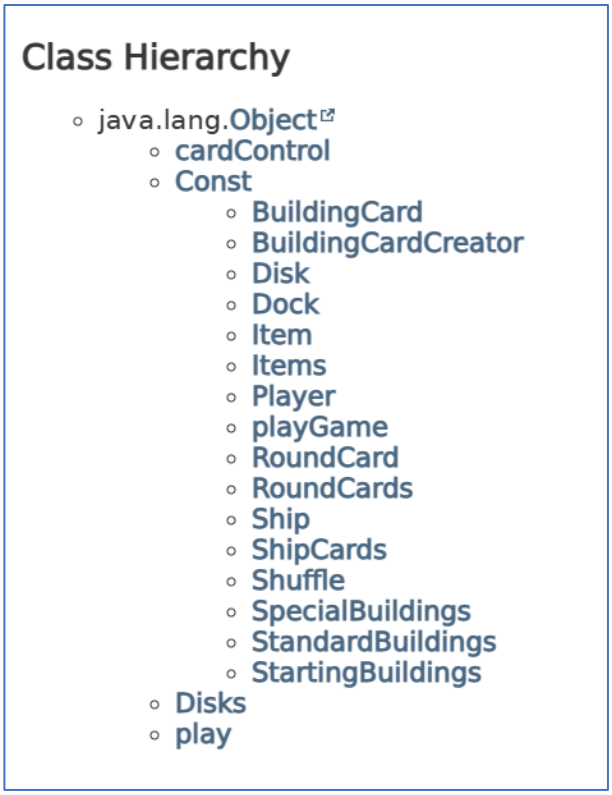
Αυτό συμβαίνει επειδή ο Computer αντιλαμβάνεται καλύτερα το περιβάλλον σε σχέση με τον Human οπότε δε διστάζει να αποπληρώσει δάνειο παρότι ο παίκτης Human έχει χρήματα που του φτάνουν να αποπληρώσει το δάνειο εξίσου

```
I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:5
            *Wood :10
            *Clay :1
            *Iron :2
            *Grain :4
            *Cattle:1
            *Coal:1
```

4 ΚΕΦΑΛΑΙΟ 4ο : Κλάσεις & κώδικας του project

Το project περιλαμβάνει συνολικά 20 κλάσεις

Σχεδόν όλες οι κλάσεις που δημιουργήθηκαν στο παιχνίδι κληρονομούν την κλάση Const



Εικόνα 7 Ιεραρχική δομή εφαρμογής(JavaDoc)

Η νοοτροπία ανάπτυξης του μοντέλου που τρέχει το παιχνίδι σε Java ήταν ότι έπρεπε να δημιουργηθεί η κλάση Const που περιέχει όλο τον όγκο και τη χρήσιμη πληροφορία για όλες τις σταθερές του παιχνιδιού που θα τροφοδοτήσουν τα στοιχεία και τις ιδιότητες των υπόλοιπων κλάσεων ώστε να γίνει πιο ξεκάθαρη και αποκεντρωμένη κάθε κλάση που συμμετέχει στη λειτουργία του παιχνιδιού. Κάθε κλάση αφορά τη δημιουργία και το ρόλο ενός αντικειμένου στο παιχνίδι εκτός από τις κλάσεις Shuffle και cardControl οι οποίες παρέχουν μεθόδους που χειρίζονται τα ήδη υπάρχοντα αντικείμενα των υπόλοιπων κλάσεων. Η κλάση όμως στην οποία καλούνται όλες οι μέθοδοι,τα αντικείμενα και οι constructor του παιχνιδιού είναι η κλάση playGame. Εκεί πέρα όλα έρχονται και δένουν μεταξύ τους σε μία λογική σειρά βάσει των κανόνων και της λειτουργίας του παιχνιδιού. Στόχος της συγκεκριμένης κλάσης είναι ότι παρά την πολυπλοκότητα και τη ροή εξέλιξης του παιχνιδιού πρέπει να είναι όσο το δυνατόν πιο άμεση και συμπτυγμένη σε γραμμές κώδικα. Αυτό συμβάλλει στην καλύτερη κατανόηση του προγράμματος και συνάμα της λειτουργίας του παιχνιδιού σε οπτικό επίπεδο πρωτίστως αλλά κάνει και

ευκολότερη τη βελτίωση του κώδικα συνάμα όπου χρειαστεί επέμβαση. Η κλάση playGame στην ουσία είναι η πλατφόρμα που εκτελεί όλες τις λειτουργίες του παιχνιδιού.

Η κλάση play απλώς παρέχει το κύριο σώμα κώδικα που είναι απαραίτητο για να τρέξουν οι εντολές της playGame constructor. Παρέχει επίσης και κάποιες εναλλακτικές εντολές για να οδηγηθούν τα αποτελέσματα του κώδικα σε αρχείο καταγραφής LogFile:

```
File file = new File("c:\\LeHavreLog\\LogFile.txt");
PrintStream stream = new PrintStream(file);
System.setOut(stream);
```

4.1 Κλάση Const

Σε αυτή την κλάση περιλαμβάνονται όλες οι τιμές των σταθερών που αφορούν όλα τα αντικείμενα και τα χαρακτηριστικά τους.

```
/**
 *  κλάση που περιλαμβάνει όλις τις τιμές των σταθερών του παιχνιδιού
**/
public class Const
{
    //παράμετροι προτεραιοτήτων αγορας καρτών standard building
    String prioBuy[]={"Marketplace","Colliery","Cokery"};

    //παράμετροι προτεραιοτήτων επίσκεψης καρτών standard building
    String prioVisit[]={"Wharf","Marketplace","Charcoal Kiln","Clay
Mound","Brickworks","Colliery","Cokery","Bakehouse","Abatoir","Tannery","Smokeho
use","Church","Steel Mill","Ironworks"};

    //παράμετροι προτεραιοτήτων για χτίσιμο καρτών standard building
    String prioBuild[]={"Charcoal Kiln","Marketplace","Colliery","Cokery"};

    //παράμετροι πιθανότητας αποπληρωμής δανείου
    int weightOfPayLoan[]={20,80}; //human,computer
    //πιθανότητες επιλογής πλακιδίου
    int weight[]={20,20,5,5,5,5,40};
    int empty=-1;
```

```
    int None=0;
    int Human=0,Computer=1,Town=2,noOwner=3,notVisited=4;
    int Wooden=0,Iron=1,Steel=2,Luxury=3;
    String typeName[]={"Wooden","Iron","Steel","Luxury"};
    int starting=0 ,standard=1,special=2;
    int Craftsman=0,Economic=1,Industrial=2,Public=3,NonBuildings=4,Yacht=5;
    int padLock=-1;
    int franc=0,fish=1,wood=2,clay=3,iron=4,grain=5,cattle=6,coal=7,hides=8;
    int
smokedfish=11,charcoal=12,brick=13,steel=14,bread=15,meat=16,coke=17,leather=18;
    int noBuilding=0,standardBuilding=1,specialBuilding=2;
    int hammer=1,fisherman=2,hammerANDfisherman=3,hammerAND2fisherman=4;
    int nothing=-1,start=0,stand=1;



    /*μέσω αυτού του πίνακα οι μέθοδοι θα ενημερώνονται ποιος παίκτης
    μπορεί να χτίσει ένα stadard building, στην 1η γραμμή βρίσκονται
    οι δείκτες των καρτών που μπορεί να χτίσει ο human και στη
    2η γραμμή οι δείκτες των καρτών που μπορεί να χτίσει ο computer
    εν δυνάμει μπορεί ο κάθε παίκτης να χτίσει το πολύ 3 κτίρια όσα και
    τα building proposals
    */
   public int
indexesOfStandardBuildingCardsThatPlayersCanBuild[][]={{None,None,None},{None,No
ne,None}};
   public void S(Object o)
   {
       System.out.println(o);
   }
   public void Sn(Object o)
   {
       System.out.print(o);
   }
//ονόματα πλακιδίων(Items) και δίσκων (Disks)

    String name1[]={"Franc",
                    "Fish ",
                    "Wood ",
                    "Clay ",
                    "Iron ",
                    "Grain ",
                    "Cattle",
                    "Coal",
                    "Hides"};

    String name2[]={"Franc     ",
                     "Smokedfish",
                     "Charcoal  ",
                     "Brick     ",
                     "Steel     ",
                     "Bread     ",
                     "Meat      ",
                     "Coke      ",
                     "Leather   "};
    // initial values of items
    int value1[]={1,0,0,0,0,0,0,0,0};
    int  meal1[]={0,1,0,0,0,0,0,0,0};
    int energy1[]={0,0,1,0,0,0,0,1,0};

    //values of turned items
```

```
    int value2[]={1,0,0,0,0,0,0,0,0};
    int  meal2[]={0,2,0,0,0,2,3,0,0};
    int energy2[]={0,0,3,0,0,0,0,10,0};

    //for creating round cards
    int
typeOfBuildingInRoundCard[]={noBuilding,standardBuilding,specialBuilding,noBuild
ing,standardBuilding,specialBuilding,noBuilding,standardBuilding,specialBuilding
,noBuilding,standardBuilding,specialBuilding,noBuilding,noBuilding};
    int Food[]={3,4,5,7,9,11,13,15,16,17,18,19,20,20};
    int Grain[]={1,1,1,1,1,1,1,1,1,1,1,1,1,0};
    int Cattle[]={2,2,2,2,2,2,2,2,2,2,2,2,2,0};


    // for creating Ship Cards
    int
type[]={Wooden,Wooden,Wooden,Wooden,Wooden,Iron,Iron,Iron,Iron,Steel,Steel,Steel
,Luxury,Luxury}; //περιέχει τις ακέραιες μεταβλητές της Const
    int value[]={2,2,4,4,6,4,6,8,10,16,20,24,34,30};
    int buyPrice[]={14,14,14,14,14,20,20,20,20,30,30,30,0,0};
    int sellPrice[]={2,2,4,4,6,4,6,8,10,16,20,24,34,30};
    int numItem1[]={5,5,5,5,5,3,3,3,3,3,3,3,3,3};
//αριθμός υλικών που απαιτούνται για να κτιστεί κάθε πλοίο
    int numItem2[]={3,3,3,3,3,4,4,4,4,2,2,2,3,3};
//αριθμός υλικών που προσφέρουν ενέργεια(λάμπα) για να κτιστεί κάθε πλοίο
    String
Item1[]={"Wood","Wood","Wood","Wood","Wood","energy","energy","energy","energy",
"energy","energy","energy","energy","energy","energy"};
//Υλικά που απαιτούνται για κτίσιμο πλοίου
    String
Item2[]={"energy","energy","energy","energy","energy","Iron","Iron","Iron","Iron
","Steel","Steel","Steel","Steel","Steel"};
//Υλικά που προσφέρουν ενέργεια για κτίσιμο πλοίου
    int numOfSellingGoods[]={2,2,2,2,2,3,3,3,3,4,4,4,0,0};
    int numOfProvidingFoods[]={4,4,4,4,4,5,5,5,5,7,7,7,0,0};

    //for BuildingCards
    String nameOfBuildingCard[]={"Building Firm I",  "Building Firm
II","Construction Firm",  //starting building

        "Marketplace","Bakehouse","Abattoir","Clay Mound","Shipping
Line","Church","Fishery",
        "Charcoal
Kiln","Smokehouse","Brickworks","Wharf","Colliery","Ironworks",
        "Cokery","Town Hall","Bank","Tannery", "Steel Mill",
//standrard building (για την έκδοση με 2 παίκτες)

         "Baguette Shop", "Bakery", "Brick Manufacturer", "Business Park",
"Clothing Industry", "Coal Trader",
         "Diner", "Farm", "Feedlot", "Fish Market", "Fish Restaurant", "Fishpond
and Wood",
         "Football Stadium", "Forest Hut", "Furniture Factory", "Furriery",
"Guildhouse", "Harbor Watch",
         "Haulage Firm", "Hunting Lodge", "Iron Mine and Coal Seam", "Kiln",
"Labor Exchange", "Leather Industry",
         "Luxury Yacht", "Masons' Guild", "Patisserie", "Plant Nursery",
"Schnaps Distillery", "Smelter",
         "Steakhouse","Steelworks", "Tavern", "Town Square", "Wind Farm",
"Zoo"}; //special buildings (για την έκδοση με 2 παίκτες)
    int typeofBuildingCard[]={starting,starting,starting,
```

```
standard,standard,standard,standard,standard,standard,standard,standard,standard
,
standard,standard,standard,standard,standard,standard,standard,standard,standard
,
        special,special,special,special,special,special,
        special,special,special,special,special,special,
        special,special,special,special,special,special,
        special,special,special,special,special,special,
        special,special,special,special,special,special,
        special,special,special,special,special,special};
    int owner[]={Town,Town,Town,

        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,

        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner

    };
    int valueOfBuildingCard[]={4,6,8, //starting building

            6,8,8,2,10,26,
            10,8,6,14,14,10,
            12,18,6,16,12,22, //standard building

            4,6,8,10,8,4,
            6,8,6,4,6,4,
            24,4,8,6,4,6,
            6,6,6,6,6,8,
            20,8,6,6,6,10,
            6,8,4,6,8,8};//special building

     int Cost[]={4,6,8, //starting building

            6,8,8,2,10,0,
            10,8,6,14,14,10,
            12,18,30,40,12,22, //standard building

            4,6,8,12,8,4,
            6,8,8,4,6,4,
            24,4,8,6,8,6,
            6,6,6,6,6,8,
            20,10,6,6,6,10,
            6,8,4,6,12,8};//special building

    int IDnum[]={2,4,6,

        1,5,9,10,18,30,
        3,7,8,14,12,16,
        22,25,28,29,20,23,

        11,13,15,17,19,21,
        24,26,27,31,32,33,
        34,35,36,37,38,39,
```

```
        40,41,42,43,44,45,
        46,47,48,49,50,51,
        52,53,54,55,56,57
       };
    int symbol[]={hammer,hammer,hammer,

        None,None,None,None,fisherman,None,
        fisherman,None,fisherman,None,None,None,
        hammer,None,None,None,None,None,


        None,None,hammer,hammer,None,None,
        fisherman,fisherman,None,fisherman,fisherman,fisherman,
        None,fisherman,None,None,hammerANDfisherman,None,
        None,hammerAND2fisherman,None,None,hammerAND2fisherman,None,
        fisherman,hammer,None,hammer,None,None,
        None,hammer,fisherman,None,None,fisherman

    };
    int typeOfBuilding[]={Craftsman,Craftsman,Industrial,
//starting building

        NonBuildings,Craftsman,Craftsman,NonBuildings,Economic,Public,
        Craftsman,Craftsman,Craftsman,Industrial,Industrial,Industrial,
        Industrial,Industrial,Public,Economic,Craftsman,Industrial,
//standrard building

        Economic,Economic,Industrial,NonBuildings,Industrial,Economic,
        Economic,Economic,Economic,Economic,Economic,NonBuildings,
        Public,Economic,Industrial,Craftsman,Economic,Public,
        Economic,Craftsman,NonBuildings,Craftsman,Public,Industrial,
        Yacht,Craftsman,Economic,Craftsman,Craftsman,Industrial,
        Economic,Industrial,Economic,NonBuildings,NonBuildings,Public,
    };//special buildings


    int entryFee[][]=
    {{0,0},{1,0},{2,0}, ///starting building

        {2,1},{1,0},{0,2},{1,0},{2,0},{0,0},
        {0,0},{0,0},{2,1},{1,0},{2,0},{2,0},
        {3,1},{0,1},{padLock,padLock},{padLock,padLock},{0,0},{0,2},
//standard building

        {1,0},{1,0},{2,0},{padLock,padLock},{2,1},{1,0},
        {1,0},{0,1},{padLock,padLock},{1,0},{1,0},{1,0},
        {padLock,padLock},{1,0},{2,0},{1,0},{padLock,padLock},{1,0},
        {1,0},{1,0},{1,0},{1,0},{0,0},{2,0},
        {padLock,padLock},{padLock,padLock},{1,0},{1,0},{1,0},{0,2},
        {1,0},{2,1},{0,0},{0,1},{padLock,padLock},{0,1}};//special building

    boolean plusSign[]={false,false,false,

    false,false,false,false,false,false,
    false,false,false,false,false,false,
    false,false,true,true,false,false,

    false,false,false,true,false,false,
    false,false,false,false,false,false,
    false,false,false,false,true,false,
    false,false,false,false,false,false,
```

```
    false,false,false,false,false,false,
    false,false,false,false,false,false


    };

    int buildingCost[][]={
        {0,0,0,0,0},
        {0,0,0,0,0},
        {0,0,0,0,0},//starting building

        {2,0,0,0,0},
        {0,2,0,0,0},
        {1,1,0,1,0},
        {0,0,0,0,0},
        {2,0,3,0,0},
        {5,0,3,1,0},

        {1,1,0,0,0},
        {0,1,0,0,0},
        {2,1,0,0,0},
        {2,1,0,1,0},
        {2,2,0,2,0},
        {1,3,0,0,0},

        {3,0,2,0,0},
        {0,0,2,1,0},
        {4,0,3,0,0},
        {0,0,4,1,0},
        {1,0,1,0,0},
        {0,0,4,2,0},

        //standard building


        {0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},
        {0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},
        {1,2,0,2,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},
        {0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},
        {0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},

{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}};



//special building

    String actions[][];

    }//end of class
```

4.2 Κλάση Disk

Κλάση που δημιουργεί το αντικείμενο του δίσκου

```
public class Disk extends Const{
    int item1,item2;
    boolean hasInterest;
    boolean Turned;
    public Disk(int item1,int item2,boolean hasInterest)
    {
        this.item1=item1;
        this.item2=item2;
        this.hasInterest=hasInterest;
        Turned=false;
    }
    public void turn() {Turned=true;}
    public String toString()
    {
        if(!Turned) return "*** ***";
        String h=new String();
        if(hasInterest) h="INTEREST";
        return name1[item1]+" "+name1[item2]+" "+h;
    }
}
```

4.3 Κλάση Disks

Δημιουργία των επτά δίσκων του παιχνιδιού

```
    public class Disks {
    Disk disk[]=new Disk[7];
    Shuffle Sh=new Shuffle();
    public Disks()
    {
        disk[0]=new Disk(1,2,true);
        disk[1]=new Disk(4,0,false);
        disk[2]=new Disk(2,0,false);
        disk[3]=new Disk(1,3,false);
        disk[4]=new Disk(2,6,false);
        disk[5]=new Disk(2,3,false);
        disk[6]=new Disk(1,5,false);

        Sh.sh(disk);

    }
    public String toString()
    {
        String s=new String();
        s="Disks : ";
        for(int i=0;i<7;i++)
            s+="("+ disk[i]+")\t";
        return s;
    }
```

```
}
```

4.4 Κλάση Dock

Δημιουργία των επτά αποβαθρών του παιχνιδιού

```
public class Dock extends Const{
    int count[]=new int[]{2,2,2,1,0,0,0};

    public Dock()
    {
        toString();

    }
    public void addItem(int i)
    {
        count[i]++;
        System.out.print("Added 1 "+name1[i]+" to dock \""+name1[i]+"
offer\".\t");
    }
    public String toString()
    {
        String s="Docks:\t";
        for(int i=0;i<7;i++)
        {
           s+=name1[i]+" offer->"+"("+count[i]+")\t";
        }
        s+="\n";
        return s;
    }
    public int getItems(int i)
    {
        int temp=count[i];
        System.out.println("\nTaking "+count[i]+" items of "+name1[i]+" from
dock \""+name1[i]+" offer\"");
        count[i]=0;
        return temp;
    }
    public boolean hasItems(int i)
    {
        if(count[i]>0) return true;
        return false;
    }
}
```

4.5 Κλάση Item

Δημιουργία αντικειμένου πλακιδίου(token)

```
public class Item extends Const{
    boolean isTurned;
    String name;
    int value,meal,energy,nameOfItem;

    public Item(int nameOfItem)
    {
        if(nameOfItem<10) createFront(nameOfItem);
        else createBack(nameOfItem-10);


    }
    public void createFront(int nameOfItem)
    {
        this.nameOfItem=nameOfItem;
        this.name=name1[nameOfItem];
        this.value=value1[nameOfItem];
        this.meal=meal1[nameOfItem];
        this.energy=energy1[nameOfItem];
        isTurned=false;
    }

    public void createBack(int nameOfItem)
    {
        this.nameOfItem=nameOfItem;
        this.name=name2[nameOfItem];
        this.value=value2[nameOfItem];
        this.meal=meal2[nameOfItem];
        this.energy=energy2[nameOfItem];
        isTurned=true;
    }

    public void turnItem()
    {
        if(isTurned) return;
        S("Turned!");
        this.name=name2[this.nameOfItem];
        this.value=value2[this.nameOfItem];
        this.meal=meal2[this.nameOfItem];
        this.energy=energy2[this.nameOfItem];
```

```
            this.isTurned=true;

        }
        public String toString()
        {
            if(nameOfItem==0) return "[1F]";
            String s="["+name+" ";
            if (value>0) s+=value+"F ";
            if (meal>0) s+=meal+"M ";
            if (energy>0) s+=energy+"E ";
            return s+"]";

        }

}
```

4.6 Κλάση Items

Δημιουργία λίστας πλακιδίων κάθε παίκτη

```
import java.util.ArrayList;
import java.lang.Math;
public class Items extends Const{
    ArrayList<Item> items=new ArrayList<>();
    public Items() {}
    public void add(Item item) {items.add(item);}
    public void add(Item item,int N)
    {
        for(int i=0;i<N;i++) items.add(item);
    }
    static int count_brick=0;
    public Item get(int i) {return items.get(i);}
    public void remove(int i) {items.remove(i);}
    public int size(){return items.size();}

    public void actionForCharcoalKiln()
    {  S("action for Charcoal Kiln\n");
       S(" action: \"Upgrade any number of woods to franc\" \n");
        int N=getTotalWoods();
        int M=N/2;
        removeWoods(M);
        add(new Item(charcoal),N-M);
        S("Number of woods upgraded:"+M);
    }


    public void actionForCokery()
    {
        S("<M01> action for Cokery\n");
        S(" action: \"Upgrade any number of coal to francs receiving 1 franc for
each\"\n");
        int countFrancs=0;
        for(Item it:items)
        {
            if(it.nameOfItem==coal) it.turnItem();
            countFrancs++;
        }
```

```
        add(new Item(franc),countFrancs);
        S("Number of coal upgraded:"+countFrancs);
    }


    public void actionForColliery(boolean hammer)
    {
        S("<M02> action for Colliery\n");
        S(" action: \"Take 3 coals + 1 adittional coal for each hammer symbol
building you own\" \n");
        add(new Item(coal),3+(hammer?1:0));

        if(hammer)
        {   S("hammer combo activates...");
            S("Number of coal gained: 4");
        }
        else       S("Number of coal gained: 3");
    }

    public void actionForMarketPlace(int numOfCraftsman)
    {
        S("<M03> action for MarketPlace\n");
        S("action: \"Take 2 different standard goods(including hides and coal)
+1 additional different good per Craftsman type building you own\" \n");
        add(new Item(hides));
        add(new Item(coal));
        add(new Item(franc),numOfCraftsman);

        if(numOfCraftsman>0)
        {
            S("Craftsman combo activates...");
            S("Number of francs gained: "+numOfCraftsman);
        }
        else S("Player takes 1 hide + 1 coal");
    }



     public void actionForFishery(int numOfisherman)
    {
         S("<M02> action for Fishery\n");
         S("action: \"Take 3 fish + 1 fish per fisherman symbol buildings you
own\"\n");
         add(new Item(fish),3);
         add(new Item(fish),1*numOfisherman);

          if(numOfisherman>0) { S("Number of fish gained:"+3+numOfisherman);}
          else { S("Number of fish gained: 3");}
    }


     public void actionForClayMound(int numOfhammer)
    {
         S("<M02> action for Clay Mound\n");
         S("action: \"Take 3 clay + 1 clay per hammer symbol buildings you
own\"\n");
         add(new Item(clay),3);
         add(new Item(clay),1*numOfhammer);

          if(numOfhammer>0) { S("Number of clay gained:"+3+numOfhammer);}
          else { S("Number of clay gained: 3");}
```

```
    }


 public void actionForBakehouse()
    {
    S("action for Bakehouse\n");
        S("action: \"Upgrade any number of grain to bread by paying 1/2
energy(rounded up) per grain and receive 1/2 franc(rounded down) per
bread\"\n");
    int N=getTotalGrain();
        int M=N-1;
        removeGrain(M);
        removeEnergy((int)Math.ceil((1/2)*M));
        add(new Item(bread),M);
        add(new Item(franc),(int) Math.floor(M/2));
        S("Total Number of bread gained: "+ M);
        S("Total Number of franc gained: "+Math.floor(M/2));

    }

 public void actionForBrickworks()
    {
    S("action for Brickworks\n");
        S("action: \"Upgrade any number of clay to brick by paying 1/2
energy(rounded up) per clay and receive 1/2 franc(rounded down) per brick\"\n");

    int N=getTotalClay();
        int M=N-1;
        removeClay(M);
        removeEnergy((int)Math.ceil((1/2)*M));
        add(new Item(brick),M);
        add(new Item(franc),(int) Math.floor(M/2));
        S("Total Number of brick gained: "+ M);
        S("Total Number of franc gained: "+Math.floor(M/2));
    }



  public void actionForSmokehouse()
    {
    S("action for Smokehouse\n");
        S("action: \"At the cost of 1 energy(overall) upgrade up to 6 fish into
smoked fish 'receiving 1/2 franc(rounded down) for each'\n");
    int N=getTotalFish();
        int countlimit6=0;

        removeEnergy(1);
        for(int i=0; i<6; i++)
        {     if(countlimit6<6 && N!=0)
              {
               add(new Item(smokedfish));
               countlimit6++;
              }
        }
        removeFish(countlimit6);
        add(new Item(franc),(int) Math.floor(countlimit6/2));
    S("Total Number of smokedfish by paing 1 energy gained: "+ N);
        S("Total Number of franc gained: "+Math.floor(countlimit6/2));
    }
```

```
   public void actionForAbatoir()
 {    S("action for Abatoir\n");
      S("action: \"Upgrade any number of cattle to meat receiving 1/2
hide(rounded down) per meat\n");
    int N=getTotalCattle();
    int M=N-2;
        removeCattle(M);
    add(new Item(meat),M);
    add(new Item(hides),(int) Math.floor(M/2));
    S("Total Number of meat gained: "+ M);
        S("Total Number of hides gained: "+ (int) Math.floor(M/2));

 }



   public void actionForIronworks()
 {    S("action for Ironworks\n");
      S("action: \"Take 3 iron + 1 additional iron by paying 6 energy\"\n");
    add(new Item(iron), 3);
        S("Total Number of iron gained: 3");
    if(getTotalIron()<4 && getTotalEnergy()>6)
    {   removeEnergy(6);
           add(new Item(iron));
               S("Total Number of iron gained: 4");
    }

 }


 public void actionForSteelMill()
 {
     S("action for SteelMill\n");
      S("action: \"Upgrade any number of iron to steel by paying 5 energy for
each\"\n");
    if(getTotalIron()>=2)
    {
        removeIron(2);
    removeEnergy(5*2);
        add(new Item(steel), 2);
        S("Total Number of steel gained: 2");
    }

    else if(getTotalIron()>=1)
    {
        removeIron(1);
    removeEnergy(5);
        add(new Item(steel), 1);
        S("Total Number of steel gained: 1");
    }

      else  if(getTotalIron()>=3 && getTotalEnergy()>=15)
    {
        removeIron(3);
        removeEnergy(5*3);
                add(new Item(steel), 3);
                S("Total Number of steel gained: 3");
    }
 }
```

```
   public void actionForTannery()
 {    S("action for SteelMill\n");
      S("action: \"Upgrade up 4 hides to leather receiving 1 franc for
each\"\n");
     removeHides(4);
     add(new Item(leather), 4);
     add(new Item(franc), 4);
         S("Total Number of leather gained: 4");
         S("Total Number of franc gained: 4");

 }


 public void actionForChurch()
 {
      S("action for Church\n");
      S("action: \"Receive 5 bread and 3 fish\"\n");
     add(new Item(bread), 5);
     add(new Item(fish), 3);

 }


  public boolean actionForWharf()
{  S("State of Wharf:\n");
  boolean modernized=false;


    if (count_brick==1)
   {
     S("Wharf is already modernized: any type of Ship can be built...");
        modernized=true;
   }
    else if(count_brick==0)
    {
        S("Wharf must first be modernized in order to build a non Wooden ship");

    }
  if (getTotalBricks()>0 && count_brick==0 && modernized==false)
 {
     S("The Wharf must first got modernized");
        count_brick=1;
        modernized=true;
 }
  else{S("Player cannot Modernize the Wharf");}
 return modernized;

 }


 public void TownHall(int numOfPublic, int numOfCraftsman)
{

        S("Town Hall Combo activates...\n");
        add(new Item(franc),4*numOfPublic);
        add(new Item(franc),2*numOfCraftsman);
        S("Number of francs gained:"+4*numOfPublic);
        S("Town Hall Combo activated:"+2*numOfCraftsman);
 }
```

```
 public void Bank(int numOfIndustrial, int numOfEconomic)
{
        S("Bank Combo activates...\n");
        add(new Item(franc),3*numOfIndustrial);
        add(new Item(franc),2*numOfEconomic);
        S("Number of francs¨gained:"+4*numOfIndustrial);
        S("Town Hall Combo activated:"+2*numOfEconomic);
}
```

```
    public int getTotalNumberOfItem(int item)
    {
        int count=0;
        for(Item it:items)
            count+=(it.nameOfItem==item)?1:0;
        return count;
    }
    public void payMeals(int N)
    {
        int count=0;
        while(count<N)
        {
            for(int i=0;i<items.size();i++)
                 if(items.get(i).meal>0)
                 {
                     count+=items.get(i).meal;
                     items.remove(i);
                     break;
                 }
         }

    }

    public void payFrancs(int N) //for roundcard
    {
        removeFrancs(N);
        /*
        for(int i=0;i<items.size();i++)
        {
            if(items.get(i).value>0) {items.remove(i);N--;}
            if(N==0) break;
        }
        */
    }

    public boolean removeFrancs(int N)
    {
        int count=0;
        if (getTotalMoney()>=N)
        {

            while(count<N)
            {
                for(int i=0;i<items.size();i++)
                    if(items.get(i).nameOfItem==franc)
                    {
                        items.remove(i);count++;
                        break;
                    }
```

```
            }
            return true;
        }
        return false;
    }

    public int getTotalWoods()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Wood ")) count++;
        return count;
    }

        public int getTotalCharcoal()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Charcoal ")) count++;
        return count;
    }

     public int getTotalGrain()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Grain ")) count++;
        return count;
    }

     public int getTotalClay()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Clay ")) count++;
        return count;
    }

     public int getTotalFish()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Fish ")) count++;
        return count;
    }

     public int getTotalSmokedFish()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Smokedfish ")) count++;
        return count;
    }

      public int getTotalBread()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Bread ")) count++;
        return count;
    }



     public int getTotalCattle()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Cattle ")) count++;
```

```
        return count;
    }

     public int getTotalMeat()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Meat ")) count++;
        return count;
    }

        public int getTotalCoal()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Coal ")) count++;
        return count;
    }

        public int getTotalCoke()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Coke ")) count++;
        return count;
    }


      public int getTotalIron()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Iron ")) count++;
        return count;
    }

      public int getTotalSteel()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Steel ")) count++;
        return count;
    }

     public int getTotalHides()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Hides ")) count++;
        return count;
    }

      public int getTotalLeather()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Leather ")) count++;
        return count;
    }


    public int getTotalBricks()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Brick ")) count++;
        return count;
    }
```

```
public int getTotalMeals()
{
    int count=0;
    for(Item it:items) count+=it.meal;
    return count;
}

public int getTotalEnergy()
{
    int count=0;
    for(Item it:items) count+=it.energy;
    return count;
}
public int getTotalMoney()
{
    int count=0;
    for(Item it:items) count+=it.value;
    return count;

}
public boolean tryToBuildWoodenShip()
{
    int countWood=0,countEnergy=0;
    boolean canBuild=false;
    for(Item it:items)
    {
        if(it.nameOfItem==wood)
            countWood++;
        else
            countEnergy+=it.energy;
        if(countWood>=5 && countEnergy>=3) canBuild=true;
    }
    Sn("Total woods="+countWood+" plus total energy="+countEnergy);
    if(canBuild)
    {
        removeWoods(5);
        removeEnergy(3);
        Sn(" that's enough to build the wooden ship!");S("");
        return true;
    }
    else
        Sn(" not enough to build wooden ship!");S("");
    return false;

}


 public boolean tryToBuildIronShip()
{
    int countIron=0,countEnergy=0;
    boolean canBuild=false;
    for(Item it:items)
    {
        if(it.nameOfItem==iron && countIron<4)
            countIron++;
        else
            countEnergy+=it.energy;
        if(countIron>=4 && countEnergy>=3) canBuild=true;
    }
    S(countIron+" "+countEnergy);
```

```
        if(canBuild)
        {
            S("");
            removeIron(4);
            removeEnergy(3);
             Sn(" that's enough to build the iron ship!");S("");
        }
        else
            Sn(" not enough to build iron ship!");S("");
        return false;

    }


     public boolean tryToBuildSteelShip()
    {
        int countSteel=0,countEnergy=0;
        boolean canBuild=false;
        for(Item it:items)
        {
            if(it.nameOfItem==wood && countSteel<2)
                countSteel++;
            else
                countEnergy+=it.energy;
            if(countSteel>=2 && countEnergy>=3) canBuild=true;
        }
        S(countSteel+" "+countEnergy);
        if(canBuild)
        {
            S("<M10");
            removeSteel(2);
            removeEnergy(3);
            Sn(" that's enough to build the steel ship!");S("");
        }
        else
            Sn(" not enough to build steel ship!");S("");
        return false;

    }


     public boolean tryToBuildLuxuryShip()
    {
        int countSteel=0,countEnergy=0;
        boolean canBuild=false;
        for(Item it:items)
        {
            if(it.nameOfItem==steel && countSteel<3)
                countSteel++;
            else
                countEnergy+=it.energy;
            if(countSteel>=3 && countEnergy>=3) canBuild=true;
        }
        S(countSteel+" "+countEnergy);
        if(canBuild)
        {
            S("");
            removeSteel(3);
            removeEnergy(3);
     Sn(" that's enough to build the luxury ship!");S("");
        }
```

```
        else
            Sn(" not enough to build luxury ship!");S("");
        return false;

    }


public void removeWoods(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==wood)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}

public void removeCharcoal(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==charcoal)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeGrain(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==grain)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


 public void removeBread(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==bread)
               {
                   items.remove(i);count++;
```

```
                    break;
                }
    }
}


public void removeClay(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==clay)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}


public void removeCattle(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==cattle)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}


public void removeMeat(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==meat)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}

public void removeCoal(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==coal)
                {
                    items.remove(i);count++;
                    break;
                }
```

```
    }
}


public void removeCoke(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==coke)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}

public void removeIron(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==iron)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}


public void removeSteel(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==steel)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}


public void removeHides(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==hides)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}
```

```
public void removeLeather(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==leather)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}

public void removeBrick(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==brick)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeFish(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==fish)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}

public void removeSmokedFish(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==smokedfish)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeEnergy(int N)
```

```
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).energy>0)
               {
                   count+=items.get(i).energy; items.remove(i);
                   break;
               }
   }
}


    public void groupedItems()
    {
        int count1[]={0,0,0,0,0,0,0,0,0};
        int count2[]={0,0,0,0,0,0,0,0,0};

        for(Item it:items)
        {

            if(!it.isTurned)
                count1[it.nameOfItem]++;
            else
                count2[it.nameOfItem]++;
        }

        S("\t\t\t**** Items ****");
        for(int i=0;i<9;i++)
        {
            if(count1[i]!=0) S("\t\t\t*"+name1[i]+":"+count1[i]);
            if(count2[i]!=0) S("\t\t\t*"+name2[i]+":"+count2[i]);
        }
    }


    public boolean hasAtLeastOneGrain()
    {
        for(Item it:items) if (it.nameOfItem==grain) return true;
        return false;

    }
    public boolean hasAtLeastTwoCattles()
    {
        int cnt=0;
        for(Item it:items) if (it.nameOfItem==cattle) cnt++;
        return (cnt>1);

    }


}
```

4.7 Κλάση Player

```
// κλάση για τη δημιουργία του παίκτη


 import java.util.*;

import java.util.Random;
import java.util.ArrayList;
public class Player extends Const{
    String name;
    int position;
    boolean isHuman;
    Items items=new Items();
    Stack<Ship> ship=new Stack<>();
    int loan;
    Shuffle sh=new Shuffle();
    static int counter;
    int canBuy=nothing;
    boolean hasMaterialsForBuild=false;
    String stdbldNames[]=new String[3];
    ShipCards s=new ShipCards();
    ArrayList<String> visitedBuildingsHistory=new ArrayList<>();

    public Player(String name,boolean isHuman)
    {
        this.name=name;
        this.isHuman=isHuman;
        position=-1;
        loan=0;
        items.add(new Item(franc), 5);
        items.add(new Item(coal));
    }
    public int timesHasVisitedBeforeBuilding(String name)
    {
        int times=0;
        for(String str:visitedBuildingsHistory)
        {
            if(str.equals(name)) times++;
        }
        return times;
    }
    public void upDateBuildingHistory(String name)
    {
        visitedBuildingsHistory.add(name);
    }
    public String shipsOfPlayer()
    {
        S("<M5> "+" "+name+" "+ship.capacity());
        String s="";
        if(ship.capacity()>0)
            for(int i=0;i<ship.capacity();i++);
               // s+=ship.get(i)+"\n";
        return s;
    }
    public int mealsEarnedFromShips()
    {
        int sum=0;
        if(ship.capacity()>0)
            for(Ship sh:ship)
                sum+=sh.numOfProvidingFoods;
```

```
        return sum;
    }

    public void resetArray()
    {
        for(int i=0;i<3;i++)
        {
            stdbldNames[i]="";
        }

    }


    public void updateMaterialArrayOfPlayerToBuildStandardBuilding(BuildingCard
sb[])
    {   resetArray();
        int it[]={wood,clay,brick,iron,steel};
        int getFromProposal[]={1,2,3};
        String str="";
        for(int i=0;i<3;i++)
            for(int j=0;j<5;j++)
                if (sb[i].BuildingCost[j]>items.getTotalNumberOfItem(it[j]))
                {
                    getFromProposal[i]=0; break;
                }
       boolean canBuildAtLeastOne=false;
       for(int i=0;i<3;i++)
       {
            if(getFromProposal[i]>0)
            {
                str+=sb[i].Name+" | ";
                stdbldNames[i]=sb[i].Name;
                canBuildAtLeastOne=true;
            }
       }
       if(!canBuildAtLeastOne)
       {
           S("Player "+name+" can't build any standard
building!");hasMaterialsForBuild=false;
       }
       else
       {
           S("Player "+name+" can build the following standard
buildings:"+str);hasMaterialsForBuild=true;
       }
    }


public void listOfBuildingsThatPlayerCanBuy(BuildingCard starting[] ,
BuildingCard standard[])
{
    String str="";
    boolean canBuyAtLeastOneStarting=false;
    boolean canBuyAtLeastOneStandard=false;
    boolean prio=false;

    for(int i=0;i<3;i++)
if(standard[i].nameOfBuildingCard.equals("Marketplace")) prio=true;

    if(!prio)
    {
```

```
        for(int i=0; i<3; i++)
            if(starting[i].Owner==Town)
                if(starting[i].Value<=items.getTotalMoney())
                {
                    str+=starting[i].Name+" | ";
                    canBuyAtLeastOneStarting=true;
                }
        if(canBuyAtLeastOneStarting) {canBuy=start; S("Player " +name+ " can buy
the following Starting Buildings:" + str);return ;}
    }
    for(int i=0; i<3; i++)
            if(standard[i].Owner==noOwner || standard[i].Owner==Town)
                    if(standard[i].Value<=items.getTotalMoney())
                    {
                            str+=standard[i].Name+" | ";
                            canBuyAtLeastOneStandard=true;
                    }
    if(canBuyAtLeastOneStandard) {canBuy=stand;S("Player " +name+ " can buy the
following Standard Buildings:" + str);return;}

    canBuy=nothing;
}




    public boolean canPayMeals(int meals)
    {

        return
(items.getTotalMeals()+items.getTotalMoney()+mealsEarnedFromShips())>=meals;
    }

    public void payMeals(int meals)
    {
        if(!canPayMeals(meals))
        {

            getLoans((meals-items.getTotalMeals()-items.getTotalMoney()-
mealsEarnedFromShips())/4+((meals-items.getTotalMeals()-items.getTotalMoney()-
mealsEarnedFromShips())%4==0?0:1));
        }

        if (items.getTotalMeals()+mealsEarnedFromShips()>meals)
        {
            items.payMeals(meals-mealsEarnedFromShips());
        }
        else
        {

            items.payFrancs(meals-mealsEarnedFromShips()-items.getTotalMeals());
            items.payMeals(items.getTotalMeals());
        }

        S((isHuman)?"\t\t\t\t Emmanouil paying "+meals+" meals":"\t\t\t\t
Computer paying "+meals+" meals");
    }


    public void getLoans(int N)
```

```
    {
        S("----- Player:"+name+" has just got "+N+" loan"+((N==1)?"":"s")+"!");
        loan+=N;
        items.add(new Item(franc),4*N);
        counter++;
    }



public void CheckTotalLoans()
{
   S("Total Loans:" + counter);

}


    public boolean payLoan()
    {
        if(loan>0)
            if(items.removeFrancs(5))
            {
                loan--;
                return true;

            }
        return false;
    }



    public void getShipCard(Ship ship)
    {
        S("<M5> "+ship);
        this.ship.push(ship);
        items.payFrancs(ship.buyPrice);


    }

    public void getItemsFromDock(Dock dock)
    {
       while(true)
       {

            int dockNumber=sh.getRndWeight(weight);
            if (dock.hasItems(dockNumber))
            {
                items.add(new Item(dockNumber), dock.getItems(dockNumber));
                S("I'm "+name+", my arraylist has:");items.groupedItems();

S("Franc:"+items.getTotalMoney()+",Energy:"+items.getTotalEnergy()+",Meal:"+item
s.getTotalMeals()+" Loans:"+loan);
                break;
            }
       }
    }

    public void Harvest()
    {
        if(items.hasAtLeastOneGrain()) items.add(new Item(grain));
```

```
        if(items.hasAtLeastTwoCattles()) items.add(new Item(cattle));

        S("\nI'm "+name+", my arraylist after Harvest
is:");items.groupedItems();
        S("\tTotal
[Franc:"+items.getTotalMoney()+",Energy:"+items.getTotalEnergy()+",Meal:"+items.
getTotalMeals()+"]"+" Loans:"+loan);

    }

    public void interest()
    {
        S("player "+name+" is in interest check!");
        if(loan>0)
        {
            S("\t paying 1F for interest");
            if(items.getTotalMoney()==0)
                getLoans(1);
            items.payFrancs(1);
        }
        else S("\t Nothing to pay!");
    }

    public String toString()
    {
        return "Player: "+name+"
(Fr:"+items.getTotalMoney()+",En:"+items.getTotalEnergy()+",Ml:"+items.getTotalM
eals()+",Position at Disk No:"+position;
    }
```

4.8 Κλάση RoundCard

Δημιουργία αντικειμένου κάρτας του γύρου

```
public class RoundCard extends Const{
    boolean HARVEST;
    int buildingInRoundCard;
    int Food;
    int Grain;
    int Cattle;
    public RoundCard(boolean HARVEST,int buildingInRoundCard,int Food,int
Grain,int Cattle)


    {
        this.HARVEST=HARVEST;
        this.buildingInRoundCard=buildingInRoundCard;
        this.Food=Food;
        this.Grain=Grain;
        this.Cattle=Cattle;
    }
```

```
    public String toString()
    {
        String s=new String();
        s="\n+--ROUND CARD--+\n";
        if(HARVEST)
            s+="HARVEST  Grain:"+Grain+" Cattle:"+Cattle+"\n";
        else  //αλλιώς
            s+="NO HARVEST\n";
        s+="Food:"+Food+"\n";
        if (buildingInRoundCard==standardBuilding) s+="Standard Building\n";
        else if (buildingInRoundCard==specialBuilding) s+="Special Building\n";
        s+="+-------------+\n";
        return s;

    }
}
```

4.9 Κλάση RoundCards

Δημιουργία δεκατεσσάρων καρτών του γύρου

```
public class RoundCards extends Const{
    RoundCard roundcards[]=new RoundCard[14];
    int topLevelRoundCardIndex=0;

    public RoundCards()
    {
        createRoundCards();
    }

    public void popRoundCard()
    {
        if(topLevelRoundCardIndex<13) topLevelRoundCardIndex++;
    }

    public void createRoundCards()
    {
        for(int i=0;i<14;i++)
            roundcards[i]=new
RoundCard(i!=13,typeOfBuildingInRoundCard[i],Food[i],Grain[i],Cattle[i]);
    }

    public int getBuildingInRoundCard()
    {
        return roundcards[topLevelRoundCardIndex].buildingInRoundCard;
    }


    public String toString()
    {
        String s=roundcards[topLevelRoundCardIndex].toString()+"\n\n";
        return s;
    }
```

```
}
```

4.10 Κλάση Ship

Δημιουργία αντικειμένου κάρτας πλοίου

```
public class Ship extends Const{
    public int type;
    public int value;
    public int buyPrice;
    public int sellPrice;

    public int numItem1;
    public String Item1;

    public int numItem2;
    public String Item2;

    public int numOfSellingGoods;
    public int numOfProvidingFoods;

    public int Owner;



    public Ship(int type,int value,int buyPrice,int sellPrice,int
numItem1,String Item1,int numItem2,String Item2,int numOfSellingGoods,int
numOfProvidingFoods)
    {
        this.type=type;
        this.value=value;
        this.buyPrice=buyPrice;
        this.sellPrice=sellPrice;

        this.numItem1=numItem1;
        this.Item1=Item1;

        this.numItem2=numItem2;
        this.Item2=Item2;

        this.numOfSellingGoods=numOfSellingGoods;
        this.numOfProvidingFoods=numOfProvidingFoods;
        this.Owner=noOwner;

    }

    public String toString()
    {
        String s=new String();
        String str[]={"Human","Computer","","no Owner"};
                s="** "+ typeName[type]+" SHIP CARD **   ( owner:"+str[Owner]+"
)\n";
        if(value>0) s+="value:"+value+"\n";
                if(buyPrice>0) s+="buy Price:"+buyPrice+"\n";
        else s+="Cannot be bought\n";
        s+=Item1+":"+numItem1+"\n";
        s+=Item2+":"+numItem2+"\n";
```

```
        s+="Selling goods:"+numOfSellingGoods +"\n";
        s+="Providing foods:"+numOfProvidingFoods +"\n";
        return s;

    }


}
```

4.11 Κλάση ShipCards

Δημιουργία των δεκατεσσάρων καρτών πλοίου του παιχνιδιού

```
import java.util.*;


public class ShipCards extends Const {


    public Stack<Ship> allShipCards=new Stack<>();
    public Stack<Ship> WoodenShips=new Stack<>();
    public Stack<Ship> IronShips=new Stack<>();
    public Stack<Ship> SteelShips=new Stack<>();
    public Stack<Ship> LuxuryShips=new Stack<>();

    public ShipCards()    {createShipCards();}


     public void createShipCards()
    {
        for(int i=13;i>=0;i--)
            allShipCards.push(new
Ship(type[i],value[i],buyPrice[i],sellPrice[i],numItem1[i],Item1[i],numItem2[i],
Item2[i],numOfSellingGoods[i],numOfProvidingFoods[i]));

    }



     public void popShipCard()
    {

        Ship ship=allShipCards.pop();
        if(ship.type==Wooden) WoodenShips.push(ship);
        else if(ship.type==Iron) IronShips.push(ship);
        else if(ship.type==Steel) SteelShips.push(ship);
        else LuxuryShips.push(ship);
    }


    public Ship cardToPlayer(int typeOfPile)
    {
        if(typeOfPile==Wooden && !WoodenShips.empty()) return WoodenShips.pop();
        else if(typeOfPile==Iron && !IronShips.empty()) return IronShips.pop();
        else if(typeOfPile==Steel && !SteelShips.empty()) return
SteelShips.pop();
```

```
        else if(typeOfPile==Luxury && !LuxuryShips.empty()) return
LuxuryShips.pop();
        return null;
    }

    public String toString()
    {
        String s="\n\t[--------(Piles)---------]\n";
        s+=((WoodenShips.empty())?"":WoodenShips.peek())+"\n";
        s+=((IronShips.empty())?"":IronShips.peek())+"\n";
        s+=((SteelShips.empty())?"":SteelShips.peek()+"\n");
        s+=((LuxuryShips.empty())?"":LuxuryShips.peek()+"\n");
        return s;
    }
}
```

4.12 Κλάση Shuffle

Κλάση για ανακάτεμα αντικειμένων του παιχνιδιού

```
public class Shuffle extends Const
{
    public void sh(Object ob[])
    {
        int N=ob.length;
        int arr[]=new int[N];

        Object temp[]=new Object[N];
        for(int i=0;i<N;i++) temp[i]=ob[i];

        for(int i=0;i<N;i++)
        {
           int r=getRnd(N);
           boolean exists=false;
           for(int j=0;j<i;j++)
               if(arr[j]==r)
                   exists=true;
           if(exists)
               i--;
           else
               arr[i]=r;
        }

        for(int i=0;i<N;i++)
            ob[i]=temp[arr[i]];

    }


    public int getRnd(int N)
    {
        return (int) (Math.random() * N);

    }
    public int getRndWeight(int w[])
```

```
    {
        int len=w.length;
        int limit[]=new int[len];
        limit[0]=w[0];
        int sum=0;
        for(int i=0;i<len;i++) sum+=w[i];
        for(int i=1;i<len;i++) limit[i]=limit[i-1]+w[i];

        int rnd=(int) (Math.random()*sum);

        for(int i=0;i<len;i++) if(rnd<limit[i]) return i;

        return len-1;

    }

}
```

4.13 Κλάση BuildingCard

Κλάση που περιέχει όλα τα χαρακτηριστικά των καρτών

```
public class BuildingCard extends Const{
    int TypeOfBuildingCard;
    String Name;
    int Value;
    int Cost;
    String Action;
    int TypeOfBuilding;
    int EntryFee[]={0,0};
    boolean PlusSign;
    int BuildingCost[]={0,0,0,0,0};
    int Owner;
    int Visited;
    int IDnum;
    int symbol;
    int index;
    public BuildingCard(int typeOfBuildingCard,String name,int value,int Cost,
            String action,int typeOfBuilding,int entryFee[],
            boolean plusSign,int buildingCost[],int IDnum,int symbol,int
owner,int index)
    {
        this.TypeOfBuildingCard=typeOfBuildingCard;
        this.Name=name;
        this.Value=value;
        this.Cost=Cost;
        this.Action=action;
        this.TypeOfBuilding=typeOfBuilding;
        this.EntryFee=entryFee;
```

```
        this.PlusSign=plusSign;
        this.BuildingCost=buildingCost;
        this.Owner=owner;
        this.Visited=notVisited;
        this.IDnum=IDnum;
        this.symbol=symbol;
        this.index=index;
    }
    public BuildingCard(){}
    public void setOwner(int owner)
    {
        this.Owner=owner;
    }
    public void buyBuilding(int owner)
    {
        this.Owner=owner;
    }

    public boolean hasHammer()
    {
        return symbol==hammer || symbol==hammerANDfisherman ||
symbol==hammerAND2fisherman;
    }
     public boolean hasFisherman()
    {
        return symbol==fisherman;
    }
    public String toString()
    {

        String t="\t\t\t\t\t\t\t\t\t";
        String s[]={"STARTING","STANDARD","SPECIAL"};
    String
s2[]={"Craftsman","Economic","Industrial","Public","NonBuildings","Yacht"};
    String s3[]={"Human","Computer","Town","no Owner"};
    String sym[]={"-","H","F","H-F","H-F-F"};
        String bc="";
        bc+=(BuildingCost[0]==0)?"":"Wood("+BuildingCost[0]+")  ";
        bc+=(BuildingCost[1]==0)?"":"Clay("+BuildingCost[1]+")  ";
        bc+=(BuildingCost[2]==0)?"":"Brick("+BuildingCost[2]+")  ";
        bc+=(BuildingCost[3]==0)?"":"Iron("+BuildingCost[3]+")  ";
        bc+=(BuildingCost[4]==0)?"":"Steel("+BuildingCost[4]+")  ";
        String ef="";
        ef+=(EntryFee[0]==0)?"":"Meals("+EntryFee[0]+") ";
        ef+=(EntryFee[1]==0)?"":"Franc("+EntryFee[1]+") ";
        ef=(EntryFee[0]==-1)?"padlock":ef;
        String S=t+"+--["+s[TypeOfBuildingCard]+"],  OWNER->"+ s3[Owner]+ " ----
----index="+index+"---+\n";
        S+=t+"|      Name:["+ Name+"]\n";
        S+=t+"|  ID:"+IDnum+", Symbol:"+sym[symbol]+" , (+):"+
(PlusSign?"Yes":"No")+" , visited:"+ (Visited==notVisited?"No":"Yes")+"\n";
        S+=t+"|             Value:"+ Value+",  Cost:"+Cost+"\n";
        S+=t+"|             Action:"+ Action+"\n";
        S+=t+"|     Type of building:"+ s2[TypeOfBuilding]+"\n";
        S+=t+"|            entry Fee:"+ef+"\n";
        S+=t+"|       Building Cost:"+bc+"\n";
        return S;
    }


}
```

4.14 Κλάση StartingBuildings

Δημιουργία των τριών Starting καρτών του παιχνιδιού

```
public class StartingBuildings extends Const
{
    BuildingCard buildingCard[]=new BuildingCard[3];


    public  StartingBuildings()
    {
        for(int i=0;i<3;i++)
            buildingCard[i]=new BuildingCard(starting,nameOfBuildingCard[i],

valueOfBuildingCard[i],Cost[i],"action",typeOfBuilding[i],entryFee[i],
            plusSign[i],buildingCost[i],
            IDnum[i],symbol[i],owner[i],i);

    }
    public String toString()
    {
        String s="======= Starting Buildings =======\n";
        for(int i=0;i<3;i++)
            s+=buildingCard[i]+"\n";
              s+="===== End of Starting Buildings =====\n";
        return s;
    }
}
```

4.15 Κλάση StandardBuildings

Δημιουργία δεκαοκτώ Standard καρτών του παιχνιδιού

```
import java.util.Stack;

public class StandardBuildings extends Const{
```

```
    BuildingCard BuildingProposalsI[]=new BuildingCard[6];
    BuildingCard BuildingProposalsII[]=new BuildingCard[6];
    BuildingCard BuildingProposalsIII[]=new BuildingCard[6];
    int topLevelIndexI=0;
    int topLevelIndexII=0;
    int topLevelIndexIII=0;
    BuildingCard buildingCard[]=new BuildingCard[18];


    public  StandardBuildings()
    {
        for(int i=3;i<21;i++)
            buildingCard[i-3]=new BuildingCard(standard,nameOfBuildingCard[i],

valueOfBuildingCard[i],Cost[i],"action",typeOfBuilding[i],entryFee[i],
            plusSign[i],buildingCost[i],
            IDnum[i],symbol[i],owner[i],i);
        shuffleCards();
    }

    public void shuffleCards()
    {
        Shuffle shuffle=new Shuffle();
        shuffle.sh(buildingCard);
        sort();
        for(int i=0;i<6;i++)
        {

            BuildingProposalsI[i]=buildingCard[i];
            BuildingProposalsII[i]=buildingCard[i+6];
            BuildingProposalsIII[i]=buildingCard[i+12];

        }
    }
    public void sort()
    {
        BuildingCard temp=new BuildingCard();
        for(int k=0;k<3;k++)
            for(int i=1+6*k;i<6+6*k;i++)
                for(int j=6+6*k-1;j>=i;j--)
                   if(buildingCard[j].IDnum<buildingCard[j-1].IDnum)
                    {
                        temp=buildingCard[j];
                        buildingCard[j]=buildingCard[j-1];
                        buildingCard[j-1]=temp;
                    }
    }
    public boolean pop()
    {
        int I,II,III,Min;
        I=(topLevelIndexI<6)?BuildingProposalsI[topLevelIndexI].IDnum:empty;
        II=(topLevelIndexII<6)?BuildingProposalsII[topLevelIndexII].IDnum:empty;

III=(topLevelIndexIII<6)?BuildingProposalsIII[topLevelIndexIII].IDnum:empty;
        Min=min(min(I,II),III);
        if(Min!=empty)
        {
            if(Min==I) pop(1);
            else if(Min==II) pop(2);
            else pop(3);
```

```
            return true;
        }
        return false;
    }
    public boolean pop(int numberOfPile)
    {
        if(numberOfPile==1)
            if(topLevelIndexI<6)
            {
                BuildingProposalsI[topLevelIndexI].setOwner(Town);
                topLevelIndexI++;return true;
            }
        if(numberOfPile==2)
            if(topLevelIndexII<6)
            {
                BuildingProposalsII[topLevelIndexII].setOwner(Town);
                topLevelIndexII++;return true;
            }
        if(numberOfPile==3)
            if(topLevelIndexIII<6)
            {
                BuildingProposalsIII[topLevelIndexIII].setOwner(Town);
                topLevelIndexIII++;return true;
            }
        return false;
    }
    public int min(int a,int b) {return (a<b)?a:b;}


  public int getTotalCraftsmanOfPlayer(int player)
  {
    int count=0;
    for(int i=0;i<6;i++)
    {
         if(i<topLevelIndexI )
             count+=(BuildingProposalsI[i].TypeOfBuilding==Craftsman &&
BuildingProposalsI[i].Owner==player)?1:0;
         if(i<topLevelIndexII)
             count+=(BuildingProposalsII[i].TypeOfBuilding==Craftsman &&
BuildingProposalsII[i].Owner==player)?1:0;
         if(i<topLevelIndexIII)
             count+=(BuildingProposalsIII[i].TypeOfBuilding==Craftsman &&
BuildingProposalsIII[i].Owner==player)?1:0;
    }
    return count;


  }

  public boolean hasHammer(int player)
   {
       int count=0;
       for(int i=0;i<6;i++)
       {
            if(i<topLevelIndexI )
                if (BuildingProposalsI[i].hasHammer() &&
BuildingProposalsI[i].Owner==player) return true;
            if(i<topLevelIndexII)
                if (BuildingProposalsII[i].hasHammer() &&
BuildingProposalsII[i].Owner==player) return true;
            if(i<topLevelIndexIII)
```

```
                if (BuildingProposalsIII[i].hasHammer() &&
BuildingProposalsIII[i].Owner==player) return true;
        }
        return false;
    }


  public String toString()
    {

        String s="======= Building Proposals =======\n";
        s+="Pile I\n";
        s+=BuildingProposalsI[topLevelIndexI];
        s+="\nPile II\n";
        s+=BuildingProposalsII[topLevelIndexII];
        s+="\nPile III\n";
        s+=BuildingProposalsIII[topLevelIndexIII]+"\n";
        s+="======= End of Building Proposals =======\n";

        for(int i=0;i<6;i++)
        {

            if(i<topLevelIndexI )
                s+=BuildingProposalsI[i]+"\n";
            if(i<topLevelIndexII)
                s+=BuildingProposalsII[i]+"\n";
            if(i<topLevelIndexIII)
                s+=BuildingProposalsIII[i]+"\n";
        }
        return s;

    }
}
```

4.16 Κλάση SpecialBuildings

Δημιουργία, ανακάτεμμα των τριανταέξι Special καρτών του παιχνιδιού και επιλογή μόνο έξι καρτών που συμμετάσχουν στο παιχνίδι

```
public class SpecialBuildings extends Const{
    public BuildingCard buildingCard[]=new BuildingCard[6];
    public int topLevelIndex=0;


    public SpecialBuildings()
    {
        BuildingCard temp[]=new BuildingCard[36];

        for(int i=21;i<57;i++)
        {
            temp[i-21]=new BuildingCard(special,nameOfBuildingCard[i],
            valueOfBuildingCard[i],Cost[i],
            "action",typeOfBuilding[i],entryFee[i],
            plusSign[i],buildingCost[i],
```

```
                IDnum[i],symbol[i],owner[i],i);
            }
            Shuffle shuffle=new Shuffle();

            shuffle.sh(temp);
            for(int i=0;i<6;i++)
                buildingCard[i]=temp[i];
        }

        public boolean pop()
        {
            if (topLevelIndex<6)
            {
                buildingCard[topLevelIndex].Owner=Town;
                topLevelIndex++;
                return true;
            }
            return false;
        }



        public String toString()
        {
            String s="";
            for(int i=0;i<topLevelIndex;i++)
                s+=buildingCard[i]+"\n";
            return s;
        }
}
```

4.17 Κλάση BuildingCardCreator

Δημιουργία Starting, Standard και Special καρτών

```
import java.util.ArrayList;
import java.util.Collections;
public class BuildingCardCreator extends Const{

    BuildingCard buildingCard[][]=new BuildingCard[5][36];
```

```
    SpecialBuildings specialBuildings=new SpecialBuildings();
    StartingBuildings startingBuildings=new StartingBuildings();
    StandardBuildings standardBuildings=new StandardBuildings();


    public BuildingCardCreator()
    {
        copyToOne();
    }


    public void pop(int type)
    {
        if(type==standard) standardBuildings.pop();
        else specialBuildings.pop();
        copyToOne();

    }



    public void visit(int i,int j,int player)
    {
        buildingCard[i][j].Visited=player;
        copyToMany();
    }
    public void visit(int IDnum,int player)
    {
        for(int i=1;i<4;i++)
            for(int j=0;j<6;j++)
                if (buildingCard[i][j].IDnum==IDnum)
                {
                    visit(i,j,player);return;
                }

    }
    public void visit(String name,int player)
    {
        for(int i=1;i<4;i++)
            for(int j=0;j<6;j++)
                if (buildingCard[i][j].Name.equals(name))
                {
                    visit(i,j,player);return;
                }


    }


   public BuildingCard[] getStartingBuilding()
  {
   BuildingCard bc[]=new BuildingCard[3];
   bc[0]= buildingCard[0][0];
   bc[1]= buildingCard[0][1];
   bc[2]= buildingCard[0][2];
   return bc;
  }
```

```
    public BuildingCard[] getStandardBuildingFromTopOfPiles()
    {
        BuildingCard bc[]=new BuildingCard[3];
        bc[0]=buildingCard[1][standardBuildings.topLevelIndexI];
        bc[1]=buildingCard[2][standardBuildings.topLevelIndexII];
        bc[2]=buildingCard[3][standardBuildings.topLevelIndexIII];
        return bc;
    }

    public ArrayList<BuildingCard> getStandardBuildsOfTown()
    {
        ArrayList<BuildingCard> bc=new ArrayList<>();
        boolean CharcoalExists=false;int position=0;
        boolean MarketplaceExists=false;
        for(int i=1;i<4;i++)
            for(int j=0;j<6;j++)
                if((buildingCard[i][j].Owner==Town  ||
buildingCard[i][j].Owner==Human || buildingCard[i][j].Owner==Computer)&&
buildingCard[i][j].Visited==notVisited)
                {
                    bc.add(buildingCard[i][j]);
                    if(buildingCard[i][j].nameOfBuildingCard.equals("Charcoal
Kiln"))
                    {
                        CharcoalExists =true;
                        position=j;
                    }

                }
        if(CharcoalExists && bc.size()>1)
            Collections.swap(bc, 0, position);
        return bc;
    }



      public int getNumOfCraftsmanOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].TypeOfBuilding==Craftsman)
                    {
                            count++;
                            var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].TypeOfBuilding==Craftsman)
                    {
                              count2++;
                              var=false;
                    }
            }
```

```
        }
         if(var) return count;
         else       return count2;

    }



     public int getNumOfHammerOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].symbol==hammer)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].symbol==hammer)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
         if(var) return count;
         else       return count2;

    }

      public boolean getIfHasHammerOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=false;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].symbol==hammer)
                    {
                           count++;
                           var=true;

                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].symbol==hammer)
                    {
                              count2++;
                              var=true;

                    }
```

```
            }
        }
        return var;
    }

     public int getNumOfFishermanOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].symbol==fisherman)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].symbol==fisherman)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
         if(var) return count;
         else       return count2;

    }


    public int getNumOfPublicOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].TypeOfBuilding==Public)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].TypeOfBuilding==Public)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
```

```
         if(var) return count;
         else       return count2;

    }


    public int getNumOfEconomicOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].TypeOfBuilding==Economic)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].TypeOfBuilding==Economic)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
         if(var) return count;
         else       return count2;

    }


     public int getNumOfIndustrialOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].TypeOfBuilding==Industrial)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].TypeOfBuilding==Industrial)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
         if(var) return count;
```

```
          else       return count2;

    }



    public void releaseBuilding(int currentPlayer)
    {
        releaseStartingBuilding(currentPlayer);
        releaseStandardBuilding(currentPlayer);


    }
    public void releaseStartingBuilding(int currentPlayer)
    {
        for(int i=0;i<3;i++)
        {
            if(buildingCard[0][i].Visited==currentPlayer)
            {
                S("Building Released:"+buildingCard[0][i].Name+" player was
"+currentPlayer);
                buildingCard[0][i].Visited=notVisited;
            }
            }
        copyToMany();
    }

   public void releaseStandardBuilding(int currentPlayer)
    {
        for(int i=1;i<4;i++)
        {
            for(int j=0;j<6;j++)
            if(buildingCard[i][j].Visited==currentPlayer)
            {
                S("Building Released:"+buildingCard[i][j].Name+" player was
"+currentPlayer);
                buildingCard[i][j].Visited=notVisited;
            }
        }
        copyToMany();
    }
    public void setOwnerToCardByName(String name,int owner)
    {
        for(int i=1;i<4;i++)
            for(int j=0;j<36;j++)
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Name.equals(name))
                    {
                        buildingCard[i][j].Owner=owner;
                        if(i==1) standardBuildings.topLevelIndexI++;
                        else if(i==2) standardBuildings.topLevelIndexII++;
                        else if(i==3) standardBuildings.topLevelIndexIII++;
                        copyToMany();
                    }
    }


    public void setOwnerToCardByIndex(int index,int owner)
    {
        for(int i=1;i<4;i++)
            for(int j=0;j<36;j++)
```

```
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].index==index)
                    {
                        buildingCard[i][j].Owner=owner;
                        if(i==1) standardBuildings.topLevelIndexI++;
                        else if(i==2) standardBuildings.topLevelIndexII++;
                        else if(i==3) standardBuildings.topLevelIndexIII++;
                        copyToMany();
                    }
    }
    public BuildingCard getBuildingCardByIDnum(int IDnum)
    {
        for(int i=0;i<5;i++)
            for(int j=0;j<36;j++)
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].IDnum==IDnum)
                        return buildingCard[i][j];
        return null;
    }
    public void copyToOne()
    {
        System.arraycopy(startingBuildings.buildingCard, 0, buildingCard[0], 0,
3);
        System.arraycopy(standardBuildings.BuildingProposalsI, 0,
buildingCard[1], 0, 6);
        System.arraycopy(standardBuildings.BuildingProposalsII, 0,
buildingCard[2], 0, 6);
        System.arraycopy(standardBuildings.BuildingProposalsIII, 0,
buildingCard[3], 0, 6);
        System.arraycopy(specialBuildings.buildingCard, 0, buildingCard[4], 0,
6);
    }
    public void copyToMany()
    {

System.arraycopy(buildingCard[0],0,startingBuildings.buildingCard,0,3);

System.arraycopy(buildingCard[1],0,standardBuildings.BuildingProposalsI,0,6);

System.arraycopy(buildingCard[2],0,standardBuildings.BuildingProposalsII,0,6);

System.arraycopy(buildingCard[3],0,standardBuildings.BuildingProposalsIII,0,6);
        System.arraycopy(buildingCard[4],0,specialBuildings.buildingCard,0,6);
    }



    public String printCard(String name)
    {
        for(int i=1;i<4;i++)
            for(int j=0;j<6;j++)
                if(buildingCard[i][j].Name.equals(name))
                    return buildingCard[i][j].toString();
        return null;
    }
    public String toString()
    {
        String s="\t\t\t\t\t\t((((((( Cards on board ))))))))\n";
        s+=startingBuildings+"\n";
        s+=standardBuildings+"\n";
        s+=specialBuildings+"\n";
```

```
        return s;

    }

}
```

4.18 Κλάση cardControl

Κλάση ελέγχου των καρτών

```
public class cardControl
{
    public ShipCards shipCards;
    public RoundCards roundCards;


    public cardControl()
    {
        shipCards=new ShipCards();
        roundCards=new RoundCards();
    }


    public void turnCard()
    {
        roundCards.popRoundCard();
        shipCards.popShipCard();
    }

    public String toString()
    {
        String s="[---- Round & Ship Cards ----]\n";
        s+=roundCards.toString();
        s+=shipCards.toString();
        return s;
    }
}
```

4.19 Κλάση playGame

Κλάση που περιέχει όλο το παιχνίδι

```
import java.util.Random;
import java.util.ArrayList;

public class playGame extends Const
```

```
{
    Dock dock=new Dock();
    Disks disks=new Disks();
    Player []player=new Player[2];
    cardControl cc=new cardControl();
    Items it= new Items();
    BuildingCardCreator buildingCards=new BuildingCardCreator();
    int currentPlayer;
   Shuffle s=new Shuffle();

public playGame()
{
    createPlayers();
    for(int round=0;round<14;round++)
    {
        S("\n\n======== ROUND No:"+(round+1)+"===========\n");
        printRoundShipAndBuildingCards();
        for(int turn=0;turn<7;turn++)
        {
            buildingCards.releaseBuilding(currentPlayer);
            printTurnAndCurrentPlayerDetails(turn,round);
            setPlayerToDiskAndTurnDisk(turn);
            printDisks();
            addItemsToDock(turn);
            checkIfcurrentPlayerHasToPayLoan();
            interest(disks.disk[turn].hasInterest);
            printIfPlayersHasMaterialsToBuildStandardBuildings();
            printIfPlayerHasFrancsToBuyBuilding();
            playerDecidesAction();
            currentPlayer=alterPlayer(currentPlayer);

        }
        endOfRoundActions(round);
        printTotalLoans();
    }

}


public void playerDecidesAction()
{
    boolean actionDone=false;
    int p;
    while(!actionDone)
    {
        if(preCheckOfWharf()) break;
        int w[]={60,5,35}; //πίνακας πιθανοτήτων
        p=new Shuffle().getRnd(100);
        if(p<w[0])
        {

                S("Player "+cp().name+" is trying to visit building");
                actionDone=tryToVisit();

        }
        else if(p<w[0]+w[1])
        {
                S("Player "+cp().name+" is trying to build building");
                actionDone=tryToBuild();
        }
        else
```

```
        {

                S("Player "+cp().name+" is getting items from dock");
                  cp().getItemsFromDock(dock);
                  actionDone=true;

        }
    }


}
public boolean preCheckOfWharf()
{
    if(new Shuffle().getRnd(100)<20)
    {

cp().updateMaterialArrayOfPlayerToBuildStandardBuilding(buildingCards.getStandar
dBuildingFromTopOfPiles());
    for(int i=0;i<3;i++)
        if(cp().stdbldNames[i].equals("Wharf"))
        {
            buildingCards.setOwnerToCardByName("Wharf",currentPlayer);
            S("<M300> Player "+cp().name+ " has build the Wharf!");
            return true;
        }
    }
    return canVisitWharf();
}
public boolean tryToVisit()
{
    return canVisitStandardBuilding(buildingCards.getStandardBuildsOfTown());
}

public boolean tryToBuild()
{

cp().updateMaterialArrayOfPlayerToBuildStandardBuilding(buildingCards.getStandar
dBuildingFromTopOfPiles());
    return checkForPriorityAndBuildStandardBuilding();
}

public boolean canVisitCharcoalKiln()
{

    if(cp().items.getTotalWoods()<2) return false;
    S("M100 Player "+cp().name+" visits Charcoal Kiln");
    buildingCards.visit(7, currentPlayer);
    return true;

}


  public boolean canVisitMarketPlace()
{
    if(cp().items.getTotalMeals()<2 && cp().items.getTotalMoney()<1) return
false;
       S("M<101> player "+cp().name+" visits MarketPlace");
       buildingCards.visit(1, currentPlayer);
       return true;
}
```

```
  public boolean canVisitClayMound()
{
    if(cp().items.getTotalMeals()<1) return false;
       S("M<101> player "+cp().name+" visits Clay Mound");
       buildingCards.visit(10, currentPlayer);
       return true;
}

  public boolean canVisitBakeHouse()
{
   if(cp().items.getTotalMeals()<1 && cp().items.getTotalGrain()<2 &&
cp().items.getTotalEnergy()<1)return false;

       S("M<101> player "+cp().name+" visits Bakehouse");
       buildingCards.visit(5, currentPlayer);
       return true;
}


  public boolean canVisitSmokeHouse()
{
   if(cp().items.getTotalMeals()<2 && cp().items.getTotalMoney()<1 &&
cp().items.getTotalFish()<1 && cp().items.getTotalEnergy()<1) return false;

       S("M<101> player "+cp().name+" visits Smokehouse");
       buildingCards.visit(8, currentPlayer);
       return true;
}

  public boolean canVisitAbatoir()
{
    if(cp().items.getTotalMoney()<2 && cp().items.getTotalCattle()<1) return
false;

       S("M<101> player "+cp().name+" visits Abatoir");
       buildingCards.visit(9, currentPlayer);
       return true;
}

  public boolean canVisitIronworks()
{
   if(cp().items.getTotalMeals()<3 && cp().items.getTotalMoney()<1 ) return
false;

       S("M<101> player "+cp().name+" visits Ironworks");
       buildingCards.visit(22, currentPlayer);
       return true;
}


  public boolean canVisitSteelMill()
{
    if(cp().items.getTotalMoney()<2 && cp().items.getTotalIron()<1 &&
cp().items.getTotalEnergy()<5) return false;

       S("M<101> player "+cp().name+" visits SteelMill");
       buildingCards.visit(23, currentPlayer);
       return true;
}
```

```
 public boolean canVisitTannery()
{
    if(cp().items.getTotalHides()<1) return false;

       S("M<101> player "+cp().name+" visits Tannery");
       buildingCards.visit(20, currentPlayer);
       return true;
}



 public boolean canVisitChurch()
{
   if(cp().items.getTotalBread()<5 && cp().items.getTotalFish()<2 ) return
false;

       S("M<101> player "+cp().name+" visits Church");
       buildingCards.visit(30, currentPlayer);
       return true;
}

 public boolean canVisitWharf()
{
    S("check for visit Wharf!");
    if(cp().items.getTotalMeals()<2) {S("...can't visit Wharf");return false;}
            if(cp().items.getTotalWoods()<5 || cp().items.getTotalEnergy()<3)
return false;
            S("M<101> player "+cp().name+" visits Wharf and paying 2 meals");
            cp().items.payMeals(2);
            buildingCards.visit(12, currentPlayer);
            cp().items.actionForWharf();
            buildShip();
            return true;
}


 public boolean canVisitTownHall()
{

    if(buildingCards.getBuildingCardByIDnum(28).Owner==Town &&
buildingCards.getBuildingCardByIDnum(28).Visited==notVisited)
    {
       S("M<101> player "+cp().name+" visits Town Hall");
       buildingCards.visit(28, currentPlayer);
       S("before");cp().items.groupedItems();
       cp().items.TownHall(buildingCards.getNumOfPublicOfPlayer(),
buildingCards.getNumOfCraftsmanOfPlayer());
       S("after");cp().items.groupedItems();
       return true;
    }

    return false;
}

 public boolean canVisitBank()
{

    if(buildingCards.getBuildingCardByIDnum(29).Owner==Town &&
buildingCards.getBuildingCardByIDnum(29).Visited==notVisited)
    {
```

```
        S("M<101> player "+cp().name+" visits Bank");
        buildingCards.visit(29, currentPlayer);
        S("before");cp().items.groupedItems();
        cp().items.Bank(buildingCards.getNumOfIndustrialOfPlayer(),
buildingCards.getNumOfEconomicOfPlayer());
        S("after");cp().items.groupedItems();
        return true;
    }

    return false;
}


 public boolean canVisitBrickWorks()
{
   if(cp().items.getTotalMeals()<1 && cp().items.getTotalClay()<1 &&
cp().items.getTotalEnergy()<1) return false;

        S("M<101> player "+cp().name+" visits Brickworks");
        buildingCards.visit(14, currentPlayer);
        return true;
}

 public boolean canVisitFishery()
{

        S("M<101> player "+cp().name+" visits Fishery");
        buildingCards.visit(3, currentPlayer);
        return true;
}

 public boolean canVisitCokery()
{
   if(cp().items.getTotalMoney()<1 && cp().items.getTotalCoal()<1) return false;

        S("M<101> player "+cp().name+" visits Cokery");
        buildingCards.visit(25, currentPlayer);
        return true;
}

 public boolean canVisitColliery()
{
   if(cp().items.getTotalMoney()<2) return false;

        S("M<101> player "+cp().name+" visits Colliery");
        buildingCards.visit(16, currentPlayer);
        return true;
}

public Player cp() {return player[currentPlayer];}


public void finallyDecidesToBuildStandardBuilding()
{

    int ind=selectAmongStartingBuildings();
    S("Player:"+cp().name+" decides to visit the starting building
"+buildingCards.buildingCard[0][ind].Name);
    buildingCards.visit(0, ind, currentPlayer);

    cp().payMeals(buildingCards.buildingCard[0][ind].EntryFee[0]);
```

```
    checkForPriorityAndBuildStandardBuilding();
}



public boolean checkForPriorityAndBuildStandardBuilding()
{
    for(String str:prioBuild)
        for(int i=0;i<3;i++)
            if(str.equals(cp().stdbldNames[i]))
            {
                 buildingCards.setOwnerToCardByName(str,currentPlayer);
                 S("<M200> Player "+cp().name+ " has build the "+str);
                 return true;
            }
    return false;
}



public int selectAmongStartingBuildings()
{
    if(buildingCards.buildingCard[0][0].Visited==notVisited)
        if(buildingCards.buildingCard[0][0].Owner==Town)
            return 0;
    if(buildingCards.buildingCard[0][1].Visited==notVisited)
        if(buildingCards.buildingCard[0][1].Owner==Town)
            return 1;
    return 2;
}


public void checkIfcurrentPlayerHasToPayLoan()
{
    S(cp().name+ " has loans:"+ cp().loan+", "+player[otherPlayer()].name + "
has:"+player[otherPlayer()].loan+" loans");
    Shuffle s= new Shuffle();
    if(cp().loan>player[otherPlayer()].loan &&
s.getRnd(100)<weightOfPayLoan[currentPlayer])
        if(cp().payLoan())
          S("Player "+cp().name+" paid 1 loan"+"\nList of player after paying
is:"+cp());

}

public void printTurnAndCurrentPlayerDetails(int turn,int round)
{
    S("\n\t ---- Turn No:"+(turn+1)+" of Round:"+round+" ----(playing
"+cp().name+")-----------");
}


 public void setPlayerToDiskAndTurnDisk(int turn)
{
    cp().position=turn;
    disks.disk[turn].turn();
}
public void printRoundShipAndBuildingCards()
{
    S(cc);
```

```
    S(buildingCards);

}

public void printIfPlayersHasMaterialsToBuildStandardBuildings()
{
    S("~~~~~~~~~~~~~~~~~~~");

cp().updateMaterialArrayOfPlayerToBuildStandardBuilding(buildingCards.getStandar
dBuildingFromTopOfPiles());
    S("~~~~~~~~~~~~~~~~~~~");
}


public void printIfPlayerHasFrancsToBuyBuilding()
{
    S("~~~~~~~~~~~~~~~~~~~");

cp().listOfBuildingsThatPlayerCanBuy(buildingCards.getStartingBuilding(),buildin
gCards.getStandardBuildingFromTopOfPiles());
    S("~~~~~~~~~~~~~~~~~~~");
}


public void endOfRoundActions(int round)
{
    harvest(round);
    payMeals(round);
    S(cc.roundCards);
    buildingCards.pop(cc.roundCards.getBuildingInRoundCard());
    cc.turnCard();
    S("$$$$$$$$$$$$$$$$$$$$ END OF ROUND "+(round+1)+"
$$$$$$$$$$$$$$$$$$$$$$$$$$");


}


public int randomStartPlayer()
{
    return new Random(System.currentTimeMillis()).nextInt(2);
}


public void buyStartingBuilding(BuildingCard starting[])
    {
    for(BuildingCard st:starting)
        if(st.Owner==Town && st.Visited==notVisited)
            if(cp().items.getTotalMoney()>=st.Value)
            {
                st.setOwner(currentPlayer);
                S("buying of starting building: "+cp().name+" has bought the "
+st.Name+"!" );
                S("Items of "+cp().name+" before buy action:"+cp());
                cp().items.payFrancs(st.Cost);

                return;
            }
    }
```

```
public void buyStandardBuilding(BuildingCard standard[])
{
    for(BuildingCard st:standard)
        if(st.Owner == noOwner && st.Visited==notVisited)
            if(st.Value<cp().items.getTotalMoney())
            {
                String str="Items of "+cp().name+" before action:"+cp();
                if(hasPriorityTheStandardBuildingCardToBeVisited(st))
                    if(cp().items.removeFrancs(st.Cost))                    {
                            st.setOwner(currentPlayer);
                            S("<M1> Player "+cp().name+" has bought the
"+st.Name+"!");
                            S(str);
                            S("Items of "+cp().name+" after action:"+cp());
                            return;
                    }
            }
}


public void callActionForStandardBuilding(BuildingCard bc)
{
    S("before");cp().items.groupedItems();
    if(bc.Name.equals("Marketplace"))
cp().items.actionForMarketPlace(buildingCards.standardBuildings.getTotalCraftsma
nOfPlayer(currentPlayer));
    else if(bc.Name.equals("Charcoal Kiln")) cp().items.actionForCharcoalKiln();
    else if(bc.Name.equals("Wharf")) cp().items.actionForWharf();
    else if(bc.Name.equals("Clay Mound"))
cp().items.actionForClayMound(buildingCards.getNumOfHammerOfPlayer());
    else if(bc.Name.equals("Brickworks")) cp().items.actionForBrickworks();
    else if(bc.Name.equals("Colliery"))
cp().items.actionForColliery(buildingCards.getIfHasHammerOfPlayer());
    else if(bc.Name.equals("Cokery")) cp().items.actionForCokery();
    else if(bc.Name.equals("Bakehouse")) cp().items.actionForBakehouse();
    else if(bc.Name.equals("Abatoir")) cp().items.actionForAbatoir();
    else if(bc.Name.equals("Tannery")) cp().items.actionForTannery();
    else if(bc.Name.equals("Smokehouse")) cp().items.actionForSmokehouse();
    else if(bc.Name.equals("Fishery"))
cp().items.actionForFishery(buildingCards.getNumOfFishermanOfPlayer());
    else if(bc.Name.equals("Church")) cp().items.actionForChurch();
    else if(bc.Name.equals("SteelMill")) cp().items.actionForSteelMill();
    else if(bc.Name.equals("Ironworks")) cp().items.actionForIronworks();

    S("after");cp().items.groupedItems();

}

public boolean canVisitStandardBuilding(ArrayList<BuildingCard> standard)
{
    for(BuildingCard st:standard)
    {
        if(hasPriorityTheStandardBuildingCardToBeVisited(st))
        {
            S("EntryFee Of Building: "+st.EntryFee[0]+" "+st.EntryFee[1]+"
"+st.Name);
            if(st.EntryFee[0]==0 && st.EntryFee[1]==0 &&
timesOfPlayerIsLessThan(st.Name,1))
            {
                visitsPlayerTheBuilding(st.Name);
                callActionForStandardBuilding(st);return true;
```

```
            }

            if(st.EntryFee[0]>0 && st.EntryFee[0]<=cp().items.getTotalMeals() &&
timesOfPlayerIsLessThan(st.Name,1))
            {
                visitsPlayerTheBuilding(st.Name);
                cp().items.payMeals(st.EntryFee[0]);
                callActionForStandardBuilding(st);return true;

            }
            if(st.EntryFee[1]>0 && st.EntryFee[1]<=cp().items.getTotalMoney() &&
timesOfPlayerIsLessThan(st.Name,1))
            {
                visitsPlayerTheBuilding(st.Name);
                cp().items.payFrancs(st.EntryFee[1]);
                callActionForStandardBuilding(st);return true;
            }
        }


    }
    return false;
}



public boolean hasVisitedStandardBuilding2(ArrayList<BuildingCard> standard)
{
    for(BuildingCard st:standard)
    {
        if(hasPriorityTheStandardBuildingCardToBeVisited(st))
        {
            if(st.Name.equals("Charcoal Kiln"))
                if(timesOfPlayerIsLessThan(st.Name,1) && canVisitCharcoalKiln())
                {
                    visitsPlayerTheBuilding(st.Name);
                    return true;
                }
            else
            {
                if(st.EntryFee[0]>0 &&
st.EntryFee[0]<=cp().items.getTotalMeals() &&
timesOfPlayerIsLessThan(st.Name,3))
                {
                    S("Player: "+cp().name+" "+" visits the "+st.Name);
                    visitsPlayerTheBuilding(st.Name);
                    cp().items.payMeals(st.EntryFee[0]);
                    callActionForStandardBuilding(st);return true;

                }

                if(st.EntryFee[1]>0 &&
st.EntryFee[1]<=cp().items.getTotalMoney() &&
timesOfPlayerIsLessThan(st.Name,3))
                {
                    visitsPlayerTheBuilding(st.Name);
                    cp().items.payFrancs(st.EntryFee[1]);
                    callActionForStandardBuilding(st);return true;
                }
            }
```

```
        }
    }
    return false;
}

public boolean timesOfPlayerIsLessThan(String name,int times,int possibility)
{

    return ((cp().timesHasVisitedBeforeBuilding(name)<times) && new
Shuffle().getRnd(100)<possibility);

}

public boolean timesOfPlayerIsLessThan(String name,int times)
{
    return cp().timesHasVisitedBeforeBuilding(name)<times;

}
public void visitsPlayerTheBuilding(String name)
{
    S("Player "+cp().name+" visits the standard building "+name);
    cp().upDateBuildingHistory(name);
    buildingCards.visit(name, currentPlayer);

}

public void buildShip()
{
    S("trying to build Ship...");
   if(cp().items.actionForWharf())
   {
        if(cp().items.tryToBuildLuxuryShip())
        {
        S(cp().name+" managed to build Luxury ship!");
        cp().s.cardToPlayer(Luxury);
        S("List Of Player "+cp().name+" Ships"+cp().ship);
        }
        if(cp().items.tryToBuildSteelShip())
        {
        S(cp().name+" managed to build Steel ship!");
        cp().s.cardToPlayer(Steel);
        S("List Of Player "+cp().name+" Ships"+cp().ship);
        }
        if(cp().items.tryToBuildIronShip())
        {
            S(cp().name+" managed to build Iron ship!");
            cp().s.cardToPlayer(Iron);
            S("List Of Player "+cp().name+" Ships"+cp().ship);
        }
        if(cp().items.tryToBuildWoodenShip())
        {
            S(cp().name+" managed to build Wooden ship!");
            cp().s.cardToPlayer(Wooden);
            S("List Of Player "+cp().name+" Ships"+cp().ship);
        }
   }
   else
    if(cp().items.tryToBuildWoodenShip())
    {
        S(cp().name+" managed to build Wooden ship!");
         cp().s.cardToPlayer(Wooden);
```

```
            S("List Of Player "+cp().name+" Ships"+cp().ship);
        }

}


public void buyShip()
{

    S("<M7");
    if(!cc.shipCards.WoodenShips.empty())
        if(cc.shipCards.WoodenShips.peek().buyPrice<=cp().items.getTotalMoney()
&& cc.shipCards.WoodenShips.peek().buyPrice>0)
        {cp().getShipCard(cc.shipCards.cardToPlayer(Wooden));return;}

    if(!cc.shipCards.IronShips.empty())
     if(cc.shipCards.IronShips.peek().buyPrice<=cp().items.getTotalMoney() &&
cc.shipCards.IronShips.peek().buyPrice>0)
        {cp().getShipCard(cc.shipCards.cardToPlayer(Iron));return;}

    if(!cc.shipCards.SteelShips.empty())
      if(cc.shipCards.SteelShips.peek().buyPrice<=cp().items.getTotalMoney() &&
cc.shipCards.SteelShips.peek().buyPrice>0)
      {cp().getShipCard(cc.shipCards.cardToPlayer(Steel));return;}

    if(!cc.shipCards.LuxuryShips.empty())
     if(cc.shipCards.LuxuryShips.peek().buyPrice<=cp().items.getTotalMoney() &&
cc.shipCards.LuxuryShips.peek().buyPrice>0)
        cp().getShipCard(cc.shipCards.cardToPlayer(Luxury));



}



public void sellBuilding(BuildingCard bc)
{
    bc.setOwner(Town);
    cp().items.add(new Item(franc),bc.Value/2);
    S("Player "+cp().name+" has just sold "+bc.Name+" to Town for "+bc.Value/2+"
francs");

}


public boolean hasPriorityTheStandardBuildingCardToBeVisited(BuildingCard bc)
{
    for(String st:prioVisit)
        if(st.equals(bc.Name) || bc.symbol!=None) return true;
    return false;
}


public boolean hasPriorityTheStandardBuildingCardToBeBuilt(BuildingCard bc)
{
    for(String st:prioBuild)
        if(st.equals(bc.Name) || bc.symbol!=None) return true;
    return false;
}
```

```
public boolean hasPriorityTheStandardBuildingCardToBeBought(BuildingCard bc)
{
    for(String st:prioBuy)
        if(st.equals(bc.Name) || bc.symbol!=None) return true;
    return false;
}

public void bonusAction()
{
    //buildShip();
    //buyShip();
    if ( cp().items.getTotalMoney()>=8)
    {
    buyBuilding();
    }

    //sellBuilding();

}



public void buyBuilding()
{

cp().listOfBuildingsThatPlayerCanBuy(buildingCards.getStartingBuilding(),buildin
gCards.getStandardBuildingFromTopOfPiles());
    String str[]={"nothing","starting","standard"};
    S("action for buy building"+cp().name+", type is:"+str[cp().canBuy+1]);

    if(cp().canBuy==start)
    {

        buyStartingBuilding(buildingCards.getStartingBuilding());
        S("Items of "+cp().name+" after action:"+cp());
    }
    else
    {
    if(cp().canBuy==stand)
        S("Items of "+cp().name+" before action:"+cp());
        buyStandardBuilding(buildingCards.getStandardBuildingFromTopOfPiles());
        S("Items of "+cp().name+" after action:"+cp());
    }

}


 public void printTotalLoans()
{   int sum=0;
    sum+=player[0].loan+player[1].loan;
    S("Total Loans:"+sum);//συνολικά δάνεια
    S("TOTAL LOANS OF "+player[0].name+":"+player[0].loan);
    S("TOTAL LOANS OF "+player[1].name+":"+player[1].loan);
}



public void payMeals(int round)
{
```

```
    int meals=cc.roundCards.roundcards[round].Food;
    S("END OF ROUND, PAYING MEALS:"+meals);
    player[Human].payMeals(meals);
    player[Computer].payMeals(meals);

}
public void interest(boolean hasInterest)
{
    if(hasInterest)
    {
        player[Human].interest();player[Computer].interest();
    }
}

public void harvest(int round)
{
    if(round<13)
    {
        player[Human].Harvest();
        player[Computer].Harvest();
    }
}

public void getCardFromPile(int typeOfPile,int Player)
{
    player[Player].getShipCard(cc.shipCards.cardToPlayer(typeOfPile));
}

public void addItemsToDock(int i)
{
    dock.addItem(disks.disk[i].item1);
    dock.addItem(disks.disk[i].item2);
    S("\n"+dock);
}
public void createPlayers()
{
    currentPlayer=randomStartPlayer();
    player[0]=new Player("Emmanouil",true);
    player[1]=new Player("Computer",false);
}
public int otherPlayer()
{
    return currentPlayer==1?0:1;
}
public int alterPlayer(int currentPlayer) {return (currentPlayer==1)?0:1;} ]

public void printDisks()
{
    System.out.print("\nDISKS->[");
    for(int i=0;i<7;i++)
    {
        char c=' ';
        if(player[0].position==i)
            c=player[0].name.charAt(0);
        else if(player[1].position==i)
            c=player[1].name.charAt(0);
        System.out.print("=("+c+"  |"+disks.disk[i]+"| )=");
    }
    S("]");
}
```

```
}
```

4.20 Κλάση play

Εκτέλεση παιχνιδιού και καταγραφή του output στο LogFile

```
import java.io.File;
import java.io.IOException;
import java.io.PrintStream;
import java.util.Scanner;

public class play  {

    public static void main(String args[]) throws IOException
    {
        File file = new File("c:\\LeHavreLog\\LogFile.txt");
        PrintStream stream = new PrintStream(file);
        System.setOut(stream);

        new playGame();

        }
    }
```

5 ΚΕΦΑΛΑΙΟ 5ο : ΣΕΝΑΡΙΑ ΕΚΤΕΛΕΣΗΣ

Στο κεφάλαιο αυτό παρουσιάζουμε τα αποτελέσματα του log file που περιλαμβάνει τη ροή του παιχνιδιού κατά την εκτέλεση της εφαρμογής.

```
======== ROUND No:1============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:3
+-------------+


    [--------(Piles)---------]


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                +--[STARTING],  OWNER->Town --------index=0---+
                                |      Name:[Building Firm I
                                |  ID:2, Symbol:H , (+):No , visited:No
                                |             Value:4,  Cost:4
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |      Name:[Building Firm II]
                                |  ID:4, Symbol:H , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=2---+
                                |      Name:[Construction Firm]
                                |  ID:6, Symbol:H , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
```

```
                                +--[STANDARD],  OWNER->no Owner --------index=3-
--+
                                |      Name:[Marketplace]
                                |  ID:1, Symbol:- , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(2) Franc(1)
                                |        Building Cost:Wood(2)

Pile II
                                +--[STANDARD],  OWNER->no Owner --------index=9-
--+
                                |      Name:[Fishery]
                                |  ID:3, Symbol:F , (+):No , visited:No
                                |             Value:10,  Cost:10
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:Wood(1)  Clay(1)

Pile III
                                +--[STANDARD],  OWNER->no Owner --------index=4-
--+
                                |      Name:[Bakehouse]
                                |  ID:5, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:Clay(2)

======= End of Building Proposals =======



     ---- Turn No:1 of Round:0 ----(playing Computer)-----------

DISKS->[=(C  |Wood  Clay  | )==(   |*** ***| )==(   |*** ***| )==(   |*** ***|
)==(   |*** ***| )==(   |*** ***| )==(   |*** ***| )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(2)    Fish  offer->(2)    Wood  offer->(3)    Clay  offer-
>(2)    Iron  offer->(0)    Grain  offer->(0)   Cattle offer->(0)

Computer has loans:0, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:5
            *Wood :3
```

```
            *Coal:1
Franc:5,Energy:4,Meal:0 Loans:0

     ---- Turn No:2 of Round:0 ----(playing Emmanouil)-----------

DISKS->[=(C  |Wood  Clay  | )==(E  |Fish  Grain  | )==(   |*** ***| )==(   |***
***| )==(   |*** ***| )==(   |*** ***| )==(   |*** ***| )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(2)    Fish  offer->(3)    Wood  offer->(0)    Clay  offer-
>(2)    Iron  offer->(0)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 3 items of Fish  from dock "Fish  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:5
            *Fish :3
            *Coal:1
Franc:5,Energy:1,Meal:3 Loans:0

     ---- Turn No:3 of Round:0 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(E  |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(   |*** ***| )==(   |*** ***| )==(   |*** ***| )==(   |*** ***| )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(0)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:0, Emmanouil has:0 loans
player Emmanouil is in interest check!
     Nothing to pay!
player Computer is in interest check!
     Nothing to pay!
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Marketplace |
~~~~~~~~~~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:5
            *Wood :4
            *Coal:1
Franc:5,Energy:5,Meal:0 Loans:0

     ---- Turn No:4 of Round:0 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(   |*** ***| )==(   |*** ***| )==(   |*** ***| )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(3)    Fish  offer->(1)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(0)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Emmanouil is trying to visit building
check for visit Wharf!
Player Emmanouil is trying to visit building
check for visit Wharf!
Player Emmanouil is trying to visit building
Player Emmanouil can't build any standard building!
check for visit Wharf!
Player Emmanouil is getting items from dock

Taking 2 items of Clay  from dock "Clay  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:5
            *Fish :3
            *Clay :2
            *Coal:1
Franc:5,Energy:1,Meal:3 Loans:0

     ---- Turn No:5 of Round:0 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(C  |Fish  Clay  | )==(   |*** ***| )==(   |*** ***|
)=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(3)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(1)    Iron  offer->(0)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:0, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Marketplace |
```

```
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:5
            *Wood :5
            *Coal:1
Franc:5,Energy:6,Meal:0 Loans:0

     ---- Turn No:6 of Round:0 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(C  |Fish  Clay  | )==(E  |Iron  Franc | )==(   |***
***| )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(4)    Fish  offer->(2)    Wood  offer->(0)    Clay  offer-
>(1)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Emmanouil is getting items from dock

Taking 4 items of Franc from dock "Franc offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:9
            *Fish :3
            *Clay :2
            *Coal:1
Franc:9,Energy:1,Meal:3 Loans:0

     ---- Turn No:7 of Round:0 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(E  |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(0)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(1)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(1)

Computer has loans:0, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Marketplace |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
```

```
...can't visit Wharf
Player Computer is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Fish  from dock "Fish  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:5
            *Fish :2
            *Wood :5
            *Coal:1
Franc:5,Energy:6,Meal:2 Loans:0

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:9
            *Fish :3
            *Clay :2
            *Coal:1
    Total [Franc:9,Energy:1,Meal:3] Loans:0

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:5
            *Fish :2
            *Wood :5
            *Coal:1
    Total [Franc:5,Energy:6,Meal:2] Loans:0
END OF ROUND, PAYING MEALS:3
                Emmanouil paying 3 meals
                Computer paying 3 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:3
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 1 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:0
TOTAL LOANS OF Emmanouil:0
TOTAL LOANS OF Computer:0


======== ROUND No:2============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:4
Standard Building
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
```

```
value:2
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                  +--[STARTING],  OWNER->Town --------index=0---+
                                  |      Name:[Building Firm I
                                  |  ID:2, Symbol:H , (+):No , visited:No
                                  |             Value:4,  Cost:4
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:
                                  |        Building Cost:

                                  +--[STARTING],  OWNER->Town --------index=1---+
                                  |      Name:[Building Firm II]
                                  |  ID:4, Symbol:H , (+):No , visited:No
                                  |             Value:6,  Cost:6
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:Meals(1)
                                  |        Building Cost:

                                  +--[STARTING],  OWNER->Town --------index=2---+
                                  |      Name:[Construction Firm]
                                  |  ID:6, Symbol:H , (+):No , visited:No
                                  |             Value:8,  Cost:8
                                  |             Action:action
                                  |     Type of building:Industrial
                                  |            entry Fee:Meals(2)
                                  |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                                  +--[STANDARD],  OWNER->no Owner --------index=3-
--+
                                  |      Name:[Marketplace]
                                  |  ID:1, Symbol:- , (+):No , visited:No
                                  |             Value:6,  Cost:6
                                  |             Action:action
                                  |     Type of building:NonBuildings
                                  |            entry Fee:Meals(2) Franc(1)
                                  |        Building Cost:Wood(2)

Pile II
                                  +--[STANDARD],  OWNER->no Owner --------index=9-
--+
                                  |      Name:[Fishery]
                                  |  ID:3, Symbol:F , (+):No , visited:No
                                  |             Value:10,  Cost:10
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:
```

```
                                |        Building Cost:Wood(1)  Clay(1)

Pile III
                               +--[STANDARD],  OWNER->no Owner --------index=4-
--+
                                |      Name:[Bakehouse]
                                |  ID:5, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:Clay(2)

======= End of Building Proposals =======

                               +--[SPECIAL],  OWNER->Town --------index=46---+
                               |      Name:[Masons Guild]
                               |  ID:47, Symbol:H , (+):No , visited:No
                               |             Value:8,  Cost:10
                               |             Action:action
                               |     Type of building:Craftsman
                               |            entry Fee:padlock
                               |        Building Cost:



     ---- Turn No:1 of Round:1 ----(playing Emmanouil)-----------

DISKS->[=(E  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(0)    Fish  offer->(0)    Wood  offer->(2)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(1)

Emmanouil has loans:0, Computer has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II | Construction Firm |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:9
```

```
            *Clay :2
            *Cattle:1
            *Coal:1
Franc:9,Energy:1,Meal:0 Loans:0

     ---- Turn No:2 of Round:1 ----(playing Computer)-----------

DISKS->[=(E  |Wood  Clay  | )==(C  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(0)    Fish  offer->(1)    Wood  offer->(2)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:0, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Marketplace |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
Player Computer can build the following standard buildings:Marketplace |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
Player Computer can build the following standard buildings:Marketplace |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:4
            *Wood :7
            *Coal:1
Franc:4,Energy:8,Meal:0 Loans:0

     ---- Turn No:3 of Round:1 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(C  |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(0)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(2)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:0 loans
player Emmanouil is in interest check!
     Nothing to pay!
player Computer is in interest check!
     Nothing to pay!
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II | Construction Firm |
```

```
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:9
            *Wood :1
            *Clay :2
            *Cattle:1
            *Coal:1
Franc:9,Energy:2,Meal:0 Loans:0

     ---- Turn No:4 of Round:1 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:0, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Marketplace |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Marketplace |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Fish  from dock "Fish  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:4
            *Fish :2
            *Wood :7
            *Coal:1
Franc:4,Energy:8,Meal:2 Loans:0

     ---- Turn No:5 of Round:1 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(E  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(1)    Wood  offer->(1)    Clay  offer-
>(3)    Iron  offer->(1)    Grain  offer->(2)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Fishery | Bakehouse
|
```

```
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II | Construction Firm |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:9
            *Wood :2
            *Clay :2
            *Cattle:1
            *Coal:1
Franc:9,Energy:3,Meal:0 Loans:0

     ---- Turn No:6 of Round:1 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(E  |Fish  Clay  | )==(C  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(0)    Clay  offer-
>(3)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:0, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Marketplace |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
M<101> player Computer visits Wharf and paying 2 meals
State of Wharf:

Wharf must first be modernized in order to build a non Wooden ship
Player cannot Modernize the Wharf
trying to build Ship...
State of Wharf:

Wharf must first be modernized in order to build a non Wooden ship
Player cannot Modernize the Wharf
Total woods=7 plus total energy=3 that's enough to build the wooden ship!
Computer managed to build Wooden ship!


     ---- Turn No:7 of Round:1 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(C  |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(1)    Clay  offer-
>(3)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(1)

Emmanouil has loans:0, Computer has:0 loans
~~~~~~~~~~~~~~~~~~~
```

```
Player Emmanouil can build the following standard buildings:Marketplace |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II | Construction Firm |
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Marketplace |
Fishery | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 2 items of Grain  from dock "Grain  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:9
            *Wood :2
            *Clay :2
            *Grain :2
            *Cattle:1
            *Coal:1
Franc:9,Energy:3,Meal:0 Loans:0

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:9
            *Wood :2
            *Clay :2
            *Grain :3
            *Cattle:1
            *Coal:1
    Total [Franc:9,Energy:3,Meal:0] Loans:0

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:4
    Total [Franc:4,Energy:0,Meal:0] Loans:0
END OF ROUND, PAYING MEALS:4
                Emmanouil paying 4 meals
                Computer paying 4 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:4
Standard Building
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 2 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:0
```

```
TOTAL LOANS OF Emmanouil:0
TOTAL LOANS OF Computer:0


======== ROUND No:3============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:5
Special Building
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:2
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4



                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                +--[STARTING],  OWNER->Town --------index=0---+
                                |      Name:[Building Firm I
                                |  ID:2, Symbol:H , (+):No , visited:No
                                |             Value:4,  Cost:4
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |      Name:[Building Firm II]
                                |  ID:4, Symbol:H , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=2---+
                                |      Name:[Construction Firm]
                                |  ID:6, Symbol:H , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
```

```
                                  +--[STANDARD],  OWNER->no Owner --------
index=10---+
                                  |      Name:[Charcoal Kiln]
                                  |  ID:7, Symbol:- , (+):No , visited:No
                                  |             Value:8,  Cost:8
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:
                                  |        Building Cost:Clay(1)

Pile II
                                  +--[STANDARD],  OWNER->no Owner --------index=9-
--+
                                  |      Name:[Fishery]
                                  |  ID:3, Symbol:F , (+):No , visited:No
                                  |             Value:10,  Cost:10
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:
                                  |        Building Cost:Wood(1)  Clay(1)

Pile III
                                  +--[STANDARD],  OWNER->no Owner --------index=4-
--+
                                  |      Name:[Bakehouse]
                                  |  ID:5, Symbol:- , (+):No , visited:No
                                  |             Value:8,  Cost:8
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:Meals(1)
                                  |        Building Cost:Clay(2)

======= End of Building Proposals =======
                                  +--[STANDARD],  OWNER->Town --------index=3---+
                                  |      Name:[Marketplace]
                                  |  ID:1, Symbol:- , (+):No , visited:No
                                  |             Value:6,  Cost:6
                                  |             Action:action
                                  |     Type of building:NonBuildings
                                  |            entry Fee:Meals(2) Franc(1)
                                  |        Building Cost:Wood(2)


                                  +--[SPECIAL],  OWNER->Town --------index=46---+
                                  |      Name:[Masons Guild]
                                  |  ID:47, Symbol:H , (+):No , visited:No
                                  |             Value:8,  Cost:10
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:padlock
                                  |        Building Cost:



Building Released:Wharf player was 1

     ---- Turn No:1 of Round:2 ----(playing Computer)-----------

DISKS->[=(C  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(E  |Wood
Cattle | )=]
```

```
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(2)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(0)   Cattle offer->(1)

Computer has loans:0, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :2
Franc:0,Energy:2,Meal:0 Loans:0

     ---- Turn No:2 of Round:2 ----(playing Emmanouil)-----------

DISKS->[=(C  |Wood  Clay  | )==(E  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(2)    Fish  offer->(2)    Wood  offer->(0)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(1)

Emmanouil has loans:0, Computer has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
Player Emmanouil visits the standard building Marketplace
before
            **** Items ****
            *Franc:4
            *Wood :2
            *Clay :2
            *Grain :3
            *Cattle:1
            *Coal:1
<M03> action for MarketPlace

action: "Take 2 different standard goods(including hides and coal) +1 additional
different good per Craftsman type building you own"

Player takes 1 hide + 1 coal
after
```

```
            **** Items ****
            *Franc:4
            *Wood :2
            *Clay :2
            *Grain :3
            *Cattle:1
            *Coal:2
            *Hides:1

     ---- Turn No:3 of Round:2 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(E  |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(2)    Fish  offer->(3)    Wood  offer->(1)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(1)

Computer has loans:0, Emmanouil has:0 loans
player Emmanouil is in interest check!
     Nothing to pay!
player Computer is in interest check!
     Nothing to pay!
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
Player Computer can't build any standard building!
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to build building
Player Computer can't build any standard building!
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
Player Computer can't build any standard building!
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :2
            *Cattle:1
Franc:0,Energy:2,Meal:0 Loans:0
Building Released:Marketplace player was 0
```

```
     ---- Turn No:4 of Round:2 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(3)    Fish  offer->(3)    Wood  offer->(2)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 4 items of Clay  from dock "Clay  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:4
            *Wood :2
            *Clay :6
            *Grain :3
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:4,Energy:4,Meal:0 Loans:0

     ---- Turn No:5 of Round:2 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(C  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(3)    Fish  offer->(4)    Wood  offer->(2)    Clay  offer-
>(1)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:0, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
```

```
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
Player Computer can't build any standard building!
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :4
            *Cattle:1
Franc:0,Energy:4,Meal:0 Loans:0

     ---- Turn No:6 of Round:2 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(C  |Fish  Clay  | )==(E  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(4)    Fish  offer->(4)    Wood  offer->(0)    Clay  offer-
>(1)    Iron  offer->(3)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 4 items of Franc from dock "Franc offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:8
            *Wood :2
            *Clay :6
            *Grain :3
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:8,Energy:4,Meal:0 Loans:0

     ---- Turn No:7 of Round:2 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(E  |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
```

```
Docks:  Franc offer->(0)    Fish  offer->(4)    Wood  offer->(1)    Clay  offer-
>(1)    Iron  offer->(3)    Grain  offer->(1)   Cattle offer->(1)

Computer has loans:0, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to build building
Player Computer can't build any standard building!
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :4
            *Cattle:2
Franc:0,Energy:4,Meal:0 Loans:0

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:8
            *Wood :2
            *Clay :6
            *Grain :4
            *Cattle:1
            *Coal:2
            *Hides:1
    Total [Franc:8,Energy:4,Meal:0] Loans:0

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Wood :4
            *Cattle:3
    Total [Franc:0,Energy:4,Meal:0] Loans:0
END OF ROUND, PAYING MEALS:5
                Emmanouil paying 5 meals
----- Player:Computer has just got 2 loans!
                Computer paying 5 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:5
Special Building
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 3 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:2
TOTAL LOANS OF Emmanouil:0
TOTAL LOANS OF Computer:2
```

```
======== ROUND No:4============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:7
+-------------+


    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:4
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                +--[STARTING],  OWNER->Town --------index=0---+
                                |      Name:[Building Firm I
                                |  ID:2, Symbol:H , (+):No , visited:No
                                |             Value:4,  Cost:4
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |      Name:[Building Firm II]
                                |  ID:4, Symbol:H , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=2---+
                                |      Name:[Construction Firm]
                                |  ID:6, Symbol:H , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                                +--[STANDARD],  OWNER->no Owner --------
index=10---+
                                |      Name:[Charcoal Kiln]
                                |  ID:7, Symbol:- , (+):No , visited:No
```

```
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:Clay(1)

Pile II
                               +--[STANDARD],  OWNER->no Owner --------index=9-
--+
                                |      Name:[Fishery]
                                |  ID:3, Symbol:F , (+):No , visited:No
                                |             Value:10,  Cost:10
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:Wood(1)  Clay(1)

Pile III
                               +--[STANDARD],  OWNER->no Owner --------index=4-
--+
                                |      Name:[Bakehouse]
                                |  ID:5, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:Clay(2)

======= End of Building Proposals =======
                               +--[STANDARD],  OWNER->Town --------index=3---+
                               |      Name:[Marketplace]
                               |  ID:1, Symbol:- , (+):No , visited:No
                               |             Value:6,  Cost:6
                               |             Action:action
                               |     Type of building:NonBuildings
                               |            entry Fee:Meals(2) Franc(1)
                               |        Building Cost:Wood(2)


                               +--[SPECIAL],  OWNER->Town --------index=46---+
                               |      Name:[Masons Guild]
                               |  ID:47, Symbol:H , (+):No , visited:No
                               |             Value:8,  Cost:10
                               |             Action:action
                               |     Type of building:Craftsman
                               |            entry Fee:padlock
                               |        Building Cost:

                               +--[SPECIAL],  OWNER->Town --------index=37---+
                               |      Name:[Guildhouse]
                               |  ID:38, Symbol:H-F , (+):Yes , visited:No
                               |             Value:4,  Cost:8
                               |             Action:action
                               |     Type of building:Economic
                               |            entry Fee:padlock
                               |        Building Cost:



     ---- Turn No:1 of Round:3 ----(playing Emmanouil)-----------
```

```
DISKS->[=(E  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(0)    Fish  offer->(4)    Wood  offer->(2)    Clay  offer-
>(2)    Iron  offer->(3)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:2 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 2 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :4
            *Clay :6
            *Grain :4
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:3,Energy:6,Meal:0 Loans:0

     ---- Turn No:2 of Round:3 ----(playing Computer)-----------

DISKS->[=(E  |Wood  Clay  | )==(C  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(0)    Fish  offer->(5)    Wood  offer->(0)    Clay  offer-
>(2)    Iron  offer->(3)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:2, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
Player Computer visits the standard building Marketplace
before
            **** Items ****
            *Franc:2
            *Wood :4
            *Cattle:3
<M03> action for MarketPlace
```

```
action: "Take 2 different standard goods(including hides and coal) +1 additional
different good per Craftsman type building you own"

Player takes 1 hide + 1 coal
after
            **** Items ****
            *Franc:2
            *Wood :4
            *Cattle:3
            *Coal:1
            *Hides:1

     ---- Turn No:3 of Round:3 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(C  |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(0)    Fish  offer->(6)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(3)    Grain  offer->(2)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:2 loans
player Emmanouil is in interest check!
     Nothing to pay!
player Computer is in interest check!
     paying 1F for interest
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 2 items of Clay  from dock "Clay  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :4
            *Clay :8
            *Grain :4
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:3,Energy:6,Meal:0 Loans:0
Building Released:Marketplace player was 1
```

```
     ---- Turn No:4 of Round:3 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(6)    Wood  offer->(2)    Clay  offer-
>(0)    Iron  offer->(3)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:2, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Iron  from dock "Iron  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :4
            *Iron :3
            *Cattle:3
            *Coal:1
            *Hides:1
Franc:1,Energy:5,Meal:0 Loans:2

     ---- Turn No:5 of Round:3 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(E  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(7)    Wood  offer->(2)    Clay  offer-
>(1)    Iron  offer->(0)    Grain  offer->(2)   Cattle offer->(0)

Emmanouil has loans:0, Computer has:2 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 2 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :6
            *Clay :8
```

```
            *Grain :4
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:3,Energy:8,Meal:0 Loans:0

     ---- Turn No:6 of Round:3 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(E  |Fish  Clay  | )==(C  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(2)    Fish  offer->(7)    Wood  offer->(0)    Clay  offer-
>(1)    Iron  offer->(1)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:2, Emmanouil has:0 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to build building
Player Computer can't build any standard building!
Player Computer can't build any standard building!
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
Player Computer can't build any standard building!
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Grain  from dock "Grain  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :4
            *Iron :3
            *Grain :2
            *Cattle:3
            *Coal:1
            *Hides:1
Franc:1,Energy:5,Meal:0 Loans:2

     ---- Turn No:7 of Round:3 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(C  |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(2)    Fish  offer->(7)    Wood  offer->(1)    Clay  offer-
>(1)    Iron  offer->(1)    Grain  offer->(0)   Cattle offer->(1)

Emmanouil has loans:0, Computer has:2 loans
```

```
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :7
            *Clay :8
            *Grain :4
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:3,Energy:9,Meal:0 Loans:0

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:3
            *Wood :7
            *Clay :8
            *Grain :5
            *Cattle:1
            *Coal:2
            *Hides:1
    Total [Franc:3,Energy:9,Meal:0] Loans:0

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:1
            *Wood :4
            *Iron :3
            *Grain :3
            *Cattle:4
            *Coal:1
            *Hides:1
    Total [Franc:1,Energy:5,Meal:0] Loans:2
END OF ROUND, PAYING MEALS:7
----- Player:Emmanouil has just got 1 loan!
                Emmanouil paying 7 meals
----- Player:Computer has just got 2 loans!
                Computer paying 7 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:7
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 4 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:5
TOTAL LOANS OF Emmanouil:1
TOTAL LOANS OF Computer:4
```

```
======== ROUND No:5============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:9
Standard Building
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:4
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4



                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                  +--[STARTING],  OWNER->Town --------index=0---+
                                  |      Name:[Building Firm I
                                  |  ID:2, Symbol:H , (+):No , visited:No
                                  |             Value:4,  Cost:4
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:
                                  |        Building Cost:

                                  +--[STARTING],  OWNER->Town --------index=1---+
                                  |      Name:[Building Firm II]
                                  |  ID:4, Symbol:H , (+):No , visited:No
                                  |             Value:6,  Cost:6
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:Meals(1)
                                  |        Building Cost:

                                  +--[STARTING],  OWNER->Town --------index=2---+
                                  |      Name:[Construction Firm]
                                  |  ID:6, Symbol:H , (+):No , visited:No
                                  |             Value:8,  Cost:8
                                  |             Action:action
                                  |     Type of building:Industrial
                                  |            entry Fee:Meals(2)
                                  |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                                  +--[STANDARD],  OWNER->no Owner --------
index=10---+
                                  |      Name:[Charcoal Kiln]
                                  |  ID:7, Symbol:- , (+):No , visited:No
```

```
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:Clay(1)

Pile II
                               +--[STANDARD],  OWNER->no Owner --------index=9-
--+
                                |      Name:[Fishery]
                                |  ID:3, Symbol:F , (+):No , visited:No
                                |             Value:10,  Cost:10
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:Wood(1)  Clay(1)

Pile III
                               +--[STANDARD],  OWNER->no Owner --------index=4-
--+
                                |      Name:[Bakehouse]
                                |  ID:5, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:Clay(2)

======= End of Building Proposals =======
                               +--[STANDARD],  OWNER->Town --------index=3---+
                               |      Name:[Marketplace]
                               |  ID:1, Symbol:- , (+):No , visited:No
                               |             Value:6,  Cost:6
                               |             Action:action
                               |     Type of building:NonBuildings
                               |            entry Fee:Meals(2) Franc(1)
                               |        Building Cost:Wood(2)


                               +--[SPECIAL],  OWNER->Town --------index=46---+
                               |      Name:[Masons Guild]
                               |  ID:47, Symbol:H , (+):No , visited:No
                               |             Value:8,  Cost:10
                               |             Action:action
                               |     Type of building:Craftsman
                               |            entry Fee:padlock
                               |        Building Cost:

                               +--[SPECIAL],  OWNER->Town --------index=37---+
                               |      Name:[Guildhouse]
                               |  ID:38, Symbol:H-F , (+):Yes , visited:No
                               |             Value:4,  Cost:8
                               |             Action:action
                               |     Type of building:Economic
                               |            entry Fee:padlock
                               |        Building Cost:

                               +--[SPECIAL],  OWNER->Town --------index=32---+
                               |      Name:[Fishpond and Wood]
                               |  ID:33, Symbol:F , (+):No , visited:No
                               |             Value:4,  Cost:4
```

```
|              Action:action
|     Type of building:NonBuildings
|            entry Fee:Meals(1)
|        Building Cost:
```

```
     ---- Turn No:1 of Round:4 ----(playing Computer)-----------

DISKS->[=(C  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(2)    Fish  offer->(7)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(0)   Cattle offer->(1)

Computer has loans:4, Emmanouil has:1 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:2
            *Wood :4
            *Iron :3
            *Grain :3
            *Cattle:5
            *Coal:1
            *Hides:1
Franc:2,Energy:5,Meal:0 Loans:4

     ---- Turn No:2 of Round:4 ----(playing Emmanouil)-----------

DISKS->[=(C  |Wood  Clay  | )==(E  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(2)    Fish  offer->(8)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:1, Computer has:4 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
```

```
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Wood :8
            *Clay :8
            *Grain :5
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:0,Energy:10,Meal:0 Loans:1

     ---- Turn No:3 of Round:4 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(E  |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(2)    Fish  offer->(9)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:4, Emmanouil has:1 loans
player Emmanouil is in interest check!
     paying 1F for interest
----- Player:Emmanouil has just got 1 loan!
player Computer is in interest check!
```

```
     paying 1F for interest
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :5
            *Iron :3
            *Grain :3
            *Cattle:5
            *Coal:1
            *Hides:1
Franc:1,Energy:6,Meal:0 Loans:4

     ---- Turn No:4 of Round:4 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(3)    Fish  offer->(9)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:2, Computer has:4 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 3 items of Franc from dock "Franc offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:6
            *Wood :8
            *Clay :8
            *Grain :5
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:6,Energy:10,Meal:0 Loans:2

     ---- Turn No:5 of Round:4 ----(playing Computer)-----------
```

```
DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(C  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(0)    Fish  offer->(10)   Wood  offer->(1)    Clay  offer-
>(3)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:4, Emmanouil has:2 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
Player Computer can't build any standard building!
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :6
            *Iron :3
            *Grain :3
            *Cattle:5
            *Coal:1
            *Hides:1
Franc:1,Energy:7,Meal:0 Loans:4

     ---- Turn No:6 of Round:4 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(C  |Fish  Clay  | )==(E  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(10)   Wood  offer->(0)    Clay  offer-
>(3)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:2, Computer has:4 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln |
Fishery | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
```

```
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 10 items of Fish  from dock "Fish  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:6
            *Fish :10
            *Wood :8
            *Clay :8
            *Grain :5
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:6,Energy:10,Meal:10 Loans:2

     ---- Turn No:7 of Round:4 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(E  |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(1)    Fish  offer->(0)    Wood  offer->(1)    Clay  offer-
>(3)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(1)

Computer has loans:4, Emmanouil has:2 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can't build any standard building!
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :7
            *Iron :3
            *Grain :3
            *Cattle:5
            *Coal:1
            *Hides:1
Franc:1,Energy:8,Meal:0 Loans:4

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:6
            *Fish :10
            *Wood :8
            *Clay :8
            *Grain :6
            *Cattle:1
            *Coal:2
            *Hides:1
```

```
    Total [Franc:6,Energy:10,Meal:10] Loans:2

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:1
            *Wood :7
            *Iron :3
            *Grain :4
            *Cattle:6
            *Coal:1
            *Hides:1
    Total [Franc:1,Energy:8,Meal:0] Loans:4
END OF ROUND, PAYING MEALS:9
                Emmanouil paying 9 meals
----- Player:Computer has just got 2 loans!
                Computer paying 9 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:9
Standard Building
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 5 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:8
TOTAL LOANS OF Emmanouil:2
TOTAL LOANS OF Computer:6


======== ROUND No:6============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:11
Special Building
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:6
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4



                  ((((((( Cards on board ))))))))
======= Starting Buildings =======
                               +--[STARTING],  OWNER->Town --------index=0---+
                               |      Name:[Building Firm I
                               |  ID:2, Symbol:H , (+):No , visited:No
                               |             Value:4,  Cost:4
                               |             Action:action
```

```
                                |      Type of building:Craftsman
                                |             entry Fee:
                                |         Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |       Name:[Building Firm II]
                                |   ID:4, Symbol:H , (+):No , visited:No
                                |              Value:6,  Cost:6
                                |              Action:action
                                |      Type of building:Craftsman
                                |             entry Fee:Meals(1)
                                |         Building Cost:

                                +--[STARTING],  OWNER->Town --------index=2---+
                                |       Name:[Construction Firm]
                                |   ID:6, Symbol:H , (+):No , visited:No
                                |              Value:8,  Cost:8
                                |              Action:action
                                |      Type of building:Industrial
                                |             entry Fee:Meals(2)
                                |         Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                                +--[STANDARD],  OWNER->no Owner --------
index=10---+
                                |       Name:[Charcoal Kiln]
                                |   ID:7, Symbol:- , (+):No , visited:No
                                |              Value:8,  Cost:8
                                |              Action:action
                                |      Type of building:Craftsman
                                |             entry Fee:
                                |         Building Cost:Clay(1)

Pile II
                                +--[STANDARD],  OWNER->no Owner --------index=6-
--+
                                |       Name:[Clay Mound]
                                |   ID:10, Symbol:- , (+):No , visited:No
                                |              Value:2,  Cost:2
                                |              Action:action
                                |      Type of building:NonBuildings
                                |             entry Fee:Meals(1)
                                |         Building Cost:

Pile III
                                +--[STANDARD],  OWNER->no Owner --------index=4-
--+
                                |       Name:[Bakehouse]
                                |   ID:5, Symbol:- , (+):No , visited:No
                                |              Value:8,  Cost:8
                                |              Action:action
                                |      Type of building:Craftsman
                                |             entry Fee:Meals(1)
                                |         Building Cost:Clay(2)

======= End of Building Proposals =======
                                +--[STANDARD],  OWNER->Town --------index=3---+
                                |       Name:[Marketplace]
```

```
|  ID:1, Symbol:- , (+):No , visited:No
|             Value:6,  Cost:6
|             Action:action
|     Type of building:NonBuildings
|            entry Fee:Meals(2) Franc(1)
|        Building Cost:Wood(2)

+--[STANDARD],  OWNER->Town --------index=9---+
|      Name:[Fishery]
|  ID:3, Symbol:F , (+):No , visited:No
|             Value:10,  Cost:10
|             Action:action
|     Type of building:Craftsman
|            entry Fee:
|        Building Cost:Wood(1)  Clay(1)


+--[SPECIAL],  OWNER->Town --------index=46---+
|      Name:[Masons Guild]
|  ID:47, Symbol:H , (+):No , visited:No
|             Value:8,  Cost:10
|             Action:action
|     Type of building:Craftsman
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=37---+
|      Name:[Guildhouse]
|  ID:38, Symbol:H-F , (+):Yes , visited:No
|             Value:4,  Cost:8
|             Action:action
|     Type of building:Economic
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=32---+
|      Name:[Fishpond and Wood]
|  ID:33, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:NonBuildings
|            entry Fee:Meals(1)
|        Building Cost:



     ---- Turn No:1 of Round:5 ----(playing Emmanouil)-----------

DISKS->[=(E  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(0)    Wood  offer->(1)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(1)

Emmanouil has loans:2, Computer has:6 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:6
            *Fish :1
            *Wood :9
            *Clay :8
            *Grain :6
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:6,Energy:11,Meal:1 Loans:2

     ---- Turn No:2 of Round:5 ----(playing Computer)-----------

DISKS->[=(E  |Wood  Clay  | )==(C  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(1)    Fish  offer->(1)    Wood  offer->(0)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(1)

Computer has loans:6, Emmanouil has:2 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
Player Computer visits the standard building Fishery
before
            **** Items ****
            *Wood :7
            *Iron :3
            *Grain :4
            *Cattle:6
            *Coal:1
            *Hides:1
<M02> action for Fishery

action: "Take 3 fish + 1 fish per fisherman symbol buildings you own"

Number of fish gained: 3
after
            **** Items ****
            *Fish :3
            *Wood :7
            *Iron :3
            *Grain :4
```

```
            *Cattle:6
            *Coal:1
            *Hides:1

     ---- Turn No:3 of Round:5 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(C  |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(1)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(1)

Emmanouil has loans:2, Computer has:6 loans
player Emmanouil is in interest check!
     paying 1F for interest
player Computer is in interest check!
     paying 1F for interest
----- Player:Computer has just got 1 loan!
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:5
            *Fish :1
            *Wood :10
            *Clay :8
            *Grain :6
            *Cattle:1
            *Coal:2
            *Hides:1
Franc:5,Energy:12,Meal:1 Loans:2
Building Released:Fishery player was 1

     ---- Turn No:4 of Round:5 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(2)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(1)

Computer has loans:7, Emmanouil has:2 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
M<101> player Computer visits Wharf and paying 2 meals
State of Wharf:

Wharf must first be modernized in order to build a non Wooden ship
Player cannot Modernize the Wharf
trying to build Ship...
State of Wharf:

Wharf must first be modernized in order to build a non Wooden ship
Player cannot Modernize the Wharf
Total woods=7 plus total energy=3 that's enough to build the wooden ship!
Computer managed to build Wooden ship!


     ---- Turn No:5 of Round:5 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(E  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(2)    Fish  offer->(3)    Wood  offer->(1)    Clay  offer-
>(5)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(1)

Emmanouil has loans:2, Computer has:7 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
Player Emmanouil visits the standard building Fishery
before
            **** Items ****
            *Franc:5
            *Fish :1
            *Wood :10
            *Clay :8
            *Grain :6
            *Cattle:1
            *Coal:2
            *Hides:1
<M02> action for Fishery

action: "Take 3 fish + 1 fish per fisherman symbol buildings you own"

Number of fish gained: 3
after
            **** Items ****
            *Franc:5
            *Fish :4
            *Wood :10
```

```
            *Clay :8
            *Grain :6
            *Cattle:1
            *Coal:2
            *Hides:1
Building Released:Wharf player was 1

     ---- Turn No:6 of Round:5 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(E  |Fish  Clay  | )==(C  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(3)    Fish  offer->(3)    Wood  offer->(1)    Clay  offer-
>(5)    Iron  offer->(3)    Grain  offer->(2)   Cattle offer->(1)

Computer has loans:7, Emmanouil has:2 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Iron  from dock "Iron  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:3
            *Fish :1
            *Iron :6
            *Grain :4
            *Cattle:6
```

```
            *Hides:1
Franc:3,Energy:0,Meal:1 Loans:7
Building Released:Fishery player was 0

     ---- Turn No:7 of Round:5 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(C  |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(3)    Fish  offer->(3)    Wood  offer->(2)    Clay  offer-
>(5)    Iron  offer->(0)    Grain  offer->(2)   Cattle offer->(2)

Emmanouil has loans:2, Computer has:7 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
M<101> player Emmanouil visits Wharf and paying 2 meals
State of Wharf:

Wharf must first be modernized in order to build a non Wooden ship
Player cannot Modernize the Wharf
trying to build Ship...
State of Wharf:

Wharf must first be modernized in order to build a non Wooden ship
Player cannot Modernize the Wharf
Total woods=10 plus total energy=3 that's enough to build the wooden ship!
Emmanouil managed to build Wooden ship!


I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:5
            *Fish :2
            *Wood :4
            *Clay :8
            *Grain :7
            *Cattle:1
            *Hides:1
    Total [Franc:5,Energy:4,Meal:2] Loans:2

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:3
            *Fish :1
            *Iron :6
            *Grain :5
            *Cattle:7
            *Hides:1
    Total [Franc:3,Energy:0,Meal:1] Loans:7
END OF ROUND, PAYING MEALS:11
----- Player:Emmanouil has just got 1 loan!
                Emmanouil paying 11 meals
----- Player:Computer has just got 2 loans!
                Computer paying 11 meals
```

```
+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:11
Special Building
+-------------+


$$$$$$$$$$$$$$$$$$$$ END OF ROUND 6 $$$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:12
TOTAL LOANS OF Emmanouil:3
TOTAL LOANS OF Computer:9


======== ROUND No:7============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:13
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:6
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4

** Iron SHIP CARD **    ( owner:no Owner )
value:4
buy Price:20
energy:3
Iron:4
Selling goods:3
Providing foods:5


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                +--[STARTING],  OWNER->Town --------index=0---+
                                |      Name:[Building Firm I
                                |  ID:2, Symbol:H , (+):No , visited:No
                                |             Value:4,  Cost:4
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |      Name:[Building Firm II]
                                |  ID:4, Symbol:H , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
```

```
                                |            entry Fee:Meals(1)
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=2---+
                                |      Name:[Construction Firm]
                                |  ID:6, Symbol:H , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                                +--[STANDARD],  OWNER->no Owner --------
index=10---+
                                |      Name:[Charcoal Kiln]
                                |  ID:7, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:Clay(1)

Pile II
                                +--[STANDARD],  OWNER->no Owner --------index=6-
--+
                                |      Name:[Clay Mound]
                                |  ID:10, Symbol:- , (+):No , visited:No
                                |             Value:2,  Cost:2
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(1)
                                |        Building Cost:

Pile III
                                +--[STANDARD],  OWNER->no Owner --------index=4-
--+
                                |      Name:[Bakehouse]
                                |  ID:5, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:Clay(2)

======= End of Building Proposals =======
                                +--[STANDARD],  OWNER->Town --------index=3---+
                                |      Name:[Marketplace]
                                |  ID:1, Symbol:- , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(2) Franc(1)
                                |        Building Cost:Wood(2)

                                +--[STANDARD],  OWNER->Town --------index=9---+
                                |      Name:[Fishery]
                                |  ID:3, Symbol:F , (+):No , visited:No
```

```
|             Value:10,  Cost:10
|             Action:action
|     Type of building:Craftsman
|            entry Fee:
|        Building Cost:Wood(1)  Clay(1)


+--[SPECIAL],  OWNER->Town --------index=46---+
|      Name:[Masons Guild]
|  ID:47, Symbol:H , (+):No , visited:No
|             Value:8,  Cost:10
|             Action:action
|     Type of building:Craftsman
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=37---+
|      Name:[Guildhouse]
|  ID:38, Symbol:H-F , (+):Yes , visited:No
|             Value:4,  Cost:8
|             Action:action
|     Type of building:Economic
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=32---+
|      Name:[Fishpond and Wood]
|  ID:33, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:NonBuildings
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=21---+
|      Name:[Baguette Shop]
|  ID:11, Symbol:- , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:Economic
|            entry Fee:Meals(1)
|        Building Cost:



     ---- Turn No:1 of Round:6 ----(playing Computer)-----------

DISKS->[=(C  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(3)    Fish  offer->(3)    Wood  offer->(3)    Clay  offer-
>(6)    Iron  offer->(0)    Grain  offer->(2)   Cattle offer->(2)

Computer has loans:9, Emmanouil has:3 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
```

```
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :3
            *Iron :6
            *Grain :5
            *Cattle:7
            *Hides:1
Franc:1,Energy:3,Meal:0 Loans:9
Building Released:Wharf player was 0

     ---- Turn No:2 of Round:6 ----(playing Emmanouil)-----------

DISKS->[=(C  |Wood  Clay  | )==(E  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(3)    Fish  offer->(4)    Wood  offer->(0)    Clay  offer-
>(6)    Iron  offer->(0)    Grain  offer->(3)   Cattle offer->(2)

Emmanouil has loans:3, Computer has:9 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 3 items of Franc from dock "Franc offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :4
            *Clay :8
            *Grain :7
            *Cattle:1
            *Hides:1
Franc:3,Energy:4,Meal:0 Loans:3

     ---- Turn No:3 of Round:6 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(E  |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(0)    Fish  offer->(5)    Wood  offer->(1)    Clay  offer-
>(6)    Iron  offer->(0)    Grain  offer->(3)   Cattle offer->(2)

Computer has loans:9, Emmanouil has:3 loans
player Emmanouil is in interest check!
     paying 1F for interest
player Computer is in interest check!
     paying 1F for interest
```

```
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Grain  from dock "Grain  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :3
            *Iron :6
            *Grain :8
            *Cattle:7
            *Hides:1
Franc:0,Energy:3,Meal:0 Loans:9

     ---- Turn No:4 of Round:6 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
```

```
Docks:  Franc offer->(1)    Fish  offer->(5)    Wood  offer->(2)    Clay  offer-
>(6)    Iron  offer->(0)    Grain  offer->(0)   Cattle offer->(2)

Emmanouil has loans:3, Computer has:9 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 2 items of Cattle from dock "Cattle offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:2
            *Wood :4
            *Clay :8
            *Grain :7
            *Cattle:3
            *Hides:1
Franc:2,Energy:4,Meal:0 Loans:3

     ---- Turn No:5 of Round:6 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(C  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(6)    Wood  offer->(2)    Clay  offer-
>(7)    Iron  offer->(0)    Grain  offer->(0)   Cattle offer->(0)

Computer has loans:9, Emmanouil has:3 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Franc from dock "Franc offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :3
            *Iron :6
            *Grain :8
            *Cattle:7
            *Hides:1
Franc:1,Energy:3,Meal:0 Loans:9

     ---- Turn No:6 of Round:6 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(C  |Fish  Clay  | )==(E  |Iron  Franc | )==(   |Wood
Cattle | )=]
```

```
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(6)    Wood  offer->(2)    Clay  offer-
>(7)    Iron  offer->(1)    Grain  offer->(0)   Cattle offer->(0)

Emmanouil has loans:3, Computer has:9 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 6 items of Fish  from dock "Fish  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:2
            *Fish :6
            *Wood :4
            *Clay :8
            *Grain :7
            *Cattle:3
            *Hides:1
Franc:2,Energy:4,Meal:6 Loans:3

     ---- Turn No:7 of Round:6 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(E  |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(1)    Fish  offer->(0)    Wood  offer->(3)    Clay  offer-
>(7)    Iron  offer->(1)    Grain  offer->(0)   Cattle offer->(1)

Computer has loans:9, Emmanouil has:3 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
```

```
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :3
            *Iron :6
            *Grain :8
            *Cattle:8
            *Hides:1
Franc:1,Energy:3,Meal:0 Loans:9

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:2
            *Fish :6
            *Wood :4
            *Clay :8
            *Grain :8
            *Cattle:4
            *Hides:1
    Total [Franc:2,Energy:4,Meal:6] Loans:3

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:1
            *Wood :3
            *Iron :6
            *Grain :9
            *Cattle:9
            *Hides:1
    Total [Franc:1,Energy:3,Meal:0] Loans:9
END OF ROUND, PAYING MEALS:13
----- Player:Emmanouil has just got 2 loans!
                Emmanouil paying 13 meals
----- Player:Computer has just got 3 loans!
                Computer paying 13 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:13
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 7 $$$$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:17
TOTAL LOANS OF Emmanouil:5
TOTAL LOANS OF Computer:12


======== ROUND No:8============

[---- Round & Ship Cards ----]
```

```
+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:15
Standard Building
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:6
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4

** Iron SHIP CARD **    ( owner:no Owner )
value:6
buy Price:20
energy:3
Iron:4
Selling goods:3
Providing foods:5


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                +--[STARTING],  OWNER->Town --------index=0---+
                                |      Name:[Building Firm I
                                |  ID:2, Symbol:H , (+):No , visited:No
                                |             Value:4,  Cost:4
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |      Name:[Building Firm II]
                                |  ID:4, Symbol:H , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=2---+
                                |      Name:[Construction Firm]
                                |  ID:6, Symbol:H , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                                +--[STANDARD],  OWNER->no Owner --------
index=10---+
```

```
                                |      Name:[Charcoal Kiln]
                                |  ID:7, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |    Type of building:Craftsman
                                |            entry Fee:
                                |       Building Cost:Clay(1)

Pile II
                               +--[STANDARD],  OWNER->no Owner --------index=6-
--+
                                |      Name:[Clay Mound]
                                |  ID:10, Symbol:- , (+):No , visited:No
                                |             Value:2,  Cost:2
                                |             Action:action
                                |    Type of building:NonBuildings
                                |            entry Fee:Meals(1)
                                |       Building Cost:

Pile III
                               +--[STANDARD],  OWNER->no Owner --------index=4-
--+
                                |      Name:[Bakehouse]
                                |  ID:5, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |    Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |       Building Cost:Clay(2)

======= End of Building Proposals =======
                               +--[STANDARD],  OWNER->Town --------index=3---+
                               |      Name:[Marketplace]
                               |  ID:1, Symbol:- , (+):No , visited:No
                               |             Value:6,  Cost:6
                               |             Action:action
                               |    Type of building:NonBuildings
                               |            entry Fee:Meals(2) Franc(1)
                               |       Building Cost:Wood(2)

                               +--[STANDARD],  OWNER->Town --------index=9---+
                               |      Name:[Fishery]
                               |  ID:3, Symbol:F , (+):No , visited:No
                               |             Value:10,  Cost:10
                               |             Action:action
                               |    Type of building:Craftsman
                               |            entry Fee:
                               |       Building Cost:Wood(1)  Clay(1)


                               +--[SPECIAL],  OWNER->Town --------index=46---+
                               |      Name:[Masons Guild]
                               |  ID:47, Symbol:H , (+):No , visited:No
                               |             Value:8,  Cost:10
                               |             Action:action
                               |    Type of building:Craftsman
                               |            entry Fee:padlock
                               |       Building Cost:

                               +--[SPECIAL],  OWNER->Town --------index=37---+
                               |      Name:[Guildhouse]
```

```
|  ID:38, Symbol:H-F , (+):Yes , visited:No
|             Value:4,  Cost:8
|             Action:action
|     Type of building:Economic
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=32---+
|      Name:[Fishpond and Wood]
|  ID:33, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:NonBuildings
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=21---+
|      Name:[Baguette Shop]
|  ID:11, Symbol:- , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:Economic
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=38---+
|      Name:[Harbor Watch]
|  ID:39, Symbol:- , (+):No , visited:No
|             Value:6,  Cost:6
|             Action:action
|     Type of building:Public
|            entry Fee:Meals(1)
|        Building Cost:
```

```
     ---- Turn No:1 of Round:7 ----(playing Emmanouil)-----------

DISKS->[=(E  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(0)    Wood  offer->(4)    Clay  offer-
>(8)    Iron  offer->(1)    Grain  offer->(0)   Cattle offer->(0)

Emmanouil has loans:5, Computer has:12 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
```

```
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 4 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :8
            *Clay :8
            *Grain :8
            *Cattle:4
            *Hides:1
Franc:3,Energy:8,Meal:0 Loans:5

     ---- Turn No:2 of Round:7 ----(playing Computer)-----------

DISKS->[=(E  |Wood  Clay  | )==(C  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(1)    Fish  offer->(1)    Wood  offer->(0)    Clay  offer-
>(8)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:12, Emmanouil has:5 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Franc from dock "Franc offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :3
            *Iron :6
            *Grain :9
            *Cattle:9
            *Hides:1
Franc:1,Energy:3,Meal:0 Loans:12

     ---- Turn No:3 of Round:7 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(C  |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(0)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(8)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:5, Computer has:12 loans
player Emmanouil is in interest check!
     paying 1F for interest
player Computer is in interest check!
     paying 1F for interest
~~~~~~~~~~~~~~~~~~~
```

```
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 8 items of Clay  from dock "Clay  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:2
            *Wood :8
            *Clay :16
            *Grain :8
            *Cattle:4
            *Hides:1
Franc:2,Energy:8,Meal:0 Loans:5

     ---- Turn No:4 of Round:7 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(2)    Wood  offer->(2)    Clay  offer-
>(0)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:12, Emmanouil has:5 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Fish  from dock "Fish  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Fish :2
            *Wood :3
            *Iron :6
            *Grain :9
            *Cattle:9
            *Hides:1
Franc:0,Energy:3,Meal:2 Loans:12

     ---- Turn No:5 of Round:7 ----(playing Emmanouil)-----------
```

```
DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(E  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(1)    Wood  offer->(2)    Clay  offer-
>(1)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:5, Computer has:12 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Fish  from dock "Fish  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:2
            *Fish :1
            *Wood :8
            *Clay :16
            *Grain :8
            *Cattle:4
            *Hides:1
Franc:2,Energy:8,Meal:1 Loans:5

     ---- Turn No:6 of Round:7 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(E  |Fish  Clay  | )==(C  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(2)    Fish  offer->(0)    Wood  offer->(2)    Clay  offer-
>(1)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:12, Emmanouil has:5 loans
~~~~~~~~~~~~~~~~~~~
```

```
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
Player Computer is getting items from dock

Taking 2 items of Franc from dock "Franc offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:2
            *Fish :2
            *Wood :3
            *Iron :6
            *Grain :9
            *Cattle:9
            *Hides:1
Franc:2,Energy:3,Meal:2 Loans:12

     ---- Turn No:7 of Round:7 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(C  |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(0)    Fish  offer->(0)    Wood  offer->(3)    Clay  offer-
>(1)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(1)

Emmanouil has loans:5, Computer has:12 loans
~~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Bakehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock
```

```
Taking 1 items of Clay  from dock "Clay  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:2
            *Fish :1
            *Wood :8
            *Clay :17
            *Grain :8
            *Cattle:4
            *Hides:1
Franc:2,Energy:8,Meal:1 Loans:5

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:2
            *Fish :1
            *Wood :8
            *Clay :17
            *Grain :9
            *Cattle:5
            *Hides:1
    Total [Franc:2,Energy:8,Meal:1] Loans:5

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:2
            *Fish :2
            *Wood :3
            *Iron :6
            *Grain :10
            *Cattle:10
            *Hides:1
    Total [Franc:2,Energy:3,Meal:2] Loans:12
END OF ROUND, PAYING MEALS:15
----- Player:Emmanouil has just got 3 loans!
                Emmanouil paying 15 meals
----- Player:Computer has just got 3 loans!
                Computer paying 15 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:15
Standard Building
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 8 $$$$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:23
TOTAL LOANS OF Emmanouil:8
TOTAL LOANS OF Computer:15


======== ROUND No:9============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:16
```

```
Special Building
+-------------+


    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:6
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4

** Iron SHIP CARD **    ( owner:no Owner )
value:8
buy Price:20
energy:3
Iron:4
Selling goods:3
Providing foods:5


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                  +--[STARTING],  OWNER->Town --------index=0---+
                                  |      Name:[Building Firm I
                                  |  ID:2, Symbol:H , (+):No , visited:No
                                  |             Value:4,  Cost:4
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:
                                  |        Building Cost:

                                  +--[STARTING],  OWNER->Town --------index=1---+
                                  |      Name:[Building Firm II]
                                  |  ID:4, Symbol:H , (+):No , visited:No
                                  |             Value:6,  Cost:6
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:Meals(1)
                                  |        Building Cost:

                                  +--[STARTING],  OWNER->Town --------index=2---+
                                  |      Name:[Construction Firm]
                                  |  ID:6, Symbol:H , (+):No , visited:No
                                  |             Value:8,  Cost:8
                                  |             Action:action
                                  |     Type of building:Industrial
                                  |            entry Fee:Meals(2)
                                  |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                                  +--[STANDARD],  OWNER->no Owner --------
index=10---+
                                  |      Name:[Charcoal Kiln]
                                  |  ID:7, Symbol:- , (+):No , visited:No
                                  |             Value:8,  Cost:8
```

```
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:Clay(1)

Pile II
                               +--[STANDARD],  OWNER->no Owner --------index=6-
--+
                                |      Name:[Clay Mound]
                                |  ID:10, Symbol:- , (+):No , visited:No
                                |             Value:2,  Cost:2
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(1)
                                |        Building Cost:

Pile III
                               +--[STANDARD],  OWNER->no Owner --------

index=11---+
                                |      Name:[Smokehouse]
                                |  ID:8, Symbol:F , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(2) Franc(1)
                                |        Building Cost:Wood(2)  Clay(1)

======= End of Building Proposals =======
                               +--[STANDARD],  OWNER->Town --------index=3---+
                               |      Name:[Marketplace]
                               |  ID:1, Symbol:- , (+):No , visited:No
                               |             Value:6,  Cost:6
                               |             Action:action
                               |     Type of building:NonBuildings
                               |            entry Fee:Meals(2) Franc(1)
                               |        Building Cost:Wood(2)

                               +--[STANDARD],  OWNER->Town --------index=9---+
                               |      Name:[Fishery]
                               |  ID:3, Symbol:F , (+):No , visited:No
                               |             Value:10,  Cost:10
                               |             Action:action
                               |     Type of building:Craftsman
                               |            entry Fee:
                               |        Building Cost:Wood(1)  Clay(1)

                               +--[STANDARD],  OWNER->Town --------index=4---+
                               |      Name:[Bakehouse]
                               |  ID:5, Symbol:- , (+):No , visited:No
                               |             Value:8,  Cost:8
                               |             Action:action
                               |     Type of building:Craftsman
                               |            entry Fee:Meals(1)
                               |        Building Cost:Clay(2)


                               +--[SPECIAL],  OWNER->Town --------index=46---+
                               |      Name:[Masons Guild]
                               |  ID:47, Symbol:H , (+):No , visited:No
                               |             Value:8,  Cost:10
                               |             Action:action
```

```
|     Type of building:Craftsman
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=37---+
|      Name:[Guildhouse]
|  ID:38, Symbol:H-F , (+):Yes , visited:No
|             Value:4,  Cost:8
|             Action:action
|     Type of building:Economic
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=32---+
|      Name:[Fishpond and Wood]
|  ID:33, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:NonBuildings
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=21---+
|      Name:[Baguette Shop]
|  ID:11, Symbol:- , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:Economic
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=38---+
|      Name:[Harbor Watch]
|  ID:39, Symbol:- , (+):No , visited:No
|             Value:6,  Cost:6
|             Action:action
|     Type of building:Public
|            entry Fee:Meals(1)
|        Building Cost:
```

```
     ---- Turn No:1 of Round:8 ----(playing Computer)-----------

DISKS->[=(C  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(0)    Fish  offer->(0)    Wood  offer->(4)    Clay  offer-
>(1)    Iron  offer->(2)    Grain  offer->(1)   Cattle offer->(1)

Computer has loans:15, Emmanouil has:8 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
```

```
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :3
            *Iron :6
            *Grain :10
            *Cattle:11
            *Hides:1
Franc:1,Energy:3,Meal:0 Loans:15

     ---- Turn No:2 of Round:8 ----(playing Emmanouil)-----------

DISKS->[=(C  |Wood  Clay  | )==(E  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(0)    Fish  offer->(1)    Wood  offer->(4)    Clay  offer-
>(1)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(0)

Emmanouil has loans:8, Computer has:15 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Smokehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to build building
Player Emmanouil can build the following standard buildings:Charcoal Kiln | Clay
Mound | Smokehouse |
<M200> Player Emmanouil has build the Charcoal Kiln

     ---- Turn No:3 of Round:8 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(E  |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(0)    Fish  offer->(2)    Wood  offer->(5)    Clay  offer-
>(1)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:15, Emmanouil has:8 loans
player Emmanouil is in interest check!
```

```
     paying 1F for interest
----- Player:Emmanouil has just got 1 loan!
player Computer is in interest check!
     paying 1F for interest
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Fish  from dock "Fish  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Fish :2
            *Wood :3
            *Iron :6
            *Grain :10
            *Cattle:11
            *Hides:1
Franc:0,Energy:3,Meal:2 Loans:15

     ---- Turn No:4 of Round:8 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(0)    Wood  offer->(6)    Clay  offer-
>(1)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(0)

Emmanouil has loans:9, Computer has:15 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 6 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :14
            *Clay :17
            *Grain :9
            *Cattle:5
            *Hides:1
Franc:3,Energy:14,Meal:0 Loans:9

     ---- Turn No:5 of Round:8 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(C  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
```

```
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(1)    Wood  offer->(0)    Clay  offer-
>(2)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:15, Emmanouil has:9 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
Player Computer visits the standard building Charcoal Kiln
before
            **** Items ****
            *Fish :2
            *Wood :3
            *Iron :6
            *Grain :10
            *Cattle:11
            *Hides:1
action for Charcoal Kiln

 action: "Upgrade any number of woods to franc"

Number of woods upgraded:1
after
            **** Items ****
            *Fish :2
            *Wood :2
            *Charcoal  :2
            *Iron :6
            *Grain :10
            *Cattle:11
            *Hides:1

     ---- Turn No:6 of Round:8 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(C  |Fish  Clay  | )==(E  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(0)    Clay  offer-
>(2)    Iron  offer->(3)    Grain  offer->(2)   Cattle offer->(0)

Emmanouil has loans:9, Computer has:15 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
```

```
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to build building
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to build building
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 2 items of Clay  from dock "Clay  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :14
            *Clay :19
            *Grain :9
            *Cattle:5
            *Hides:1
Franc:3,Energy:14,Meal:0 Loans:9
Building Released:Charcoal Kiln player was 1

     ---- Turn No:7 of Round:8 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(E  |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(1)    Clay  offer-
>(0)    Iron  offer->(3)    Grain  offer->(2)   Cattle offer->(1)
```

```
Computer has loans:15, Emmanouil has:9 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Computer is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Fish :2
            *Wood :3
            *Charcoal  :2
            *Iron :6
            *Grain :10
            *Cattle:11
            *Hides:1
Franc:0,Energy:9,Meal:2 Loans:15

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:3
            *Wood :14
            *Clay :19
            *Grain :10
            *Cattle:6
            *Hides:1
    Total [Franc:3,Energy:14,Meal:0] Loans:9

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Fish :2
            *Wood :3
            *Charcoal  :2
            *Iron :6
            *Grain :11
            *Cattle:12
            *Hides:1
    Total [Franc:0,Energy:9,Meal:2] Loans:15
END OF ROUND, PAYING MEALS:16
----- Player:Emmanouil has just got 4 loans!
                Emmanouil paying 16 meals
----- Player:Computer has just got 4 loans!
                Computer paying 16 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:16
Special Building
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 9 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:32
TOTAL LOANS OF Emmanouil:13
TOTAL LOANS OF Computer:19
```

```
======== ROUND No:10============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:17
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:6
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4

** Iron SHIP CARD **    ( owner:no Owner )
value:10
buy Price:20
energy:3
Iron:4
Selling goods:3
Providing foods:5


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                +--[STARTING],  OWNER->Town --------index=0---+
                                |      Name:[Building Firm I
                                |  ID:2, Symbol:H , (+):No , visited:No
                                |             Value:4,  Cost:4
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |      Name:[Building Firm II]
                                |  ID:4, Symbol:H , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=2---+
                                |      Name:[Construction Firm]
                                |  ID:6, Symbol:H , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
```

```
Pile I
                                +--[STANDARD],  OWNER->no Owner --------index=5-
--+
                                |      Name:[Abattoir]
                                |  ID:9, Symbol:- , (+):No , visited:No
                                |              Value:8,  Cost:8
                                |              Action:action
                                |     Type of building:Craftsman
                                |             entry Fee:Franc(2)
                                |        Building Cost:Wood(1)  Clay(1)  Iron(1)

Pile II
                                +--[STANDARD],  OWNER->no Owner --------index=6-
--+
                                |      Name:[Clay Mound]
                                |  ID:10, Symbol:- , (+):No , visited:No
                                |              Value:2,  Cost:2
                                |              Action:action
                                |     Type of building:NonBuildings
                                |             entry Fee:Meals(1)
                                |        Building Cost:

Pile III
                                +--[STANDARD],  OWNER->no Owner --------
index=11---+
                                |      Name:[Smokehouse]
                                |  ID:8, Symbol:F , (+):No , visited:No
                                |              Value:6,  Cost:6
                                |              Action:action
                                |     Type of building:Craftsman
                                |             entry Fee:Meals(2) Franc(1)
                                |        Building Cost:Wood(2)  Clay(1)

======= End of Building Proposals =======
                                +--[STANDARD],  OWNER->Town --------index=3---+
                                |      Name:[Marketplace]
                                |  ID:1, Symbol:- , (+):No , visited:No
                                |              Value:6,  Cost:6
                                |              Action:action
                                |     Type of building:NonBuildings
                                |             entry Fee:Meals(2) Franc(1)
                                |        Building Cost:Wood(2)

                                +--[STANDARD],  OWNER->Town --------index=9---+
                                |      Name:[Fishery]
                                |  ID:3, Symbol:F , (+):No , visited:No
                                |              Value:10,  Cost:10
                                |              Action:action
                                |     Type of building:Craftsman
                                |             entry Fee:
                                |        Building Cost:Wood(1)  Clay(1)

                                +--[STANDARD],  OWNER->Town --------index=4---+
                                |      Name:[Bakehouse]
                                |  ID:5, Symbol:- , (+):No , visited:No
                                |              Value:8,  Cost:8
                                |              Action:action
                                |     Type of building:Craftsman
                                |             entry Fee:Meals(1)
                                |        Building Cost:Clay(2)
```

```
                          +--[STANDARD],  OWNER->Human --------index=10---
+
                          |      Name:[Charcoal Kiln]
                          |  ID:7, Symbol:- , (+):No , visited:No
                          |             Value:8,  Cost:8
                          |             Action:action
                          |     Type of building:Craftsman
                          |            entry Fee:
                          |        Building Cost:Clay(1)


                          +--[SPECIAL],  OWNER->Town --------index=46---+
                          |      Name:[Masons Guild]
                          |  ID:47, Symbol:H , (+):No , visited:No
                          |             Value:8,  Cost:10
                          |             Action:action
                          |     Type of building:Craftsman
                          |            entry Fee:padlock
                          |        Building Cost:

                          +--[SPECIAL],  OWNER->Town --------index=37---+
                          |      Name:[Guildhouse]
                          |  ID:38, Symbol:H-F , (+):Yes , visited:No
                          |             Value:4,  Cost:8
                          |             Action:action
                          |     Type of building:Economic
                          |            entry Fee:padlock
                          |        Building Cost:

                          +--[SPECIAL],  OWNER->Town --------index=32---+
                          |      Name:[Fishpond and Wood]
                          |  ID:33, Symbol:F , (+):No , visited:No
                          |             Value:4,  Cost:4
                          |             Action:action
                          |     Type of building:NonBuildings
                          |            entry Fee:Meals(1)
                          |        Building Cost:

                          +--[SPECIAL],  OWNER->Town --------index=21---+
                          |      Name:[Baguette Shop]
                          |  ID:11, Symbol:- , (+):No , visited:No
                          |             Value:4,  Cost:4
                          |             Action:action
                          |     Type of building:Economic
                          |            entry Fee:Meals(1)
                          |        Building Cost:

                          +--[SPECIAL],  OWNER->Town --------index=38---+
                          |      Name:[Harbor Watch]
                          |  ID:39, Symbol:- , (+):No , visited:No
                          |             Value:6,  Cost:6
                          |             Action:action
                          |     Type of building:Public
                          |            entry Fee:Meals(1)
                          |        Building Cost:

                          +--[SPECIAL],  OWNER->Town --------index=53---+
                          |      Name:[Tavern]
                          |  ID:54, Symbol:F , (+):No , visited:No
                          |             Value:4,  Cost:4
                          |             Action:action
```

```
                                |     Type of building:Economic
                                |           entry Fee:
                                |       Building Cost:



     ---- Turn No:1 of Round:9 ----(playing Emmanouil)-----------

DISKS->[=(E  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(1)    Clay  offer-
>(1)    Iron  offer->(3)    Grain  offer->(2)   Cattle offer->(1)

Emmanouil has loans:13, Computer has:19 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
Player Emmanouil visits the standard building Charcoal Kiln
before
            **** Items ****
            *Franc:3
            *Wood :14
            *Clay :19
            *Grain :10
            *Cattle:6
            *Hides:1
action for Charcoal Kiln

 action: "Upgrade any number of woods to franc"

Number of woods upgraded:7
after
            **** Items ****
            *Franc:3
            *Wood :7
            *Charcoal  :7
            *Clay :19
            *Grain :10
            *Cattle:6
            *Hides:1

     ---- Turn No:2 of Round:9 ----(playing Computer)-----------

DISKS->[=(E  |Wood  Clay  | )==(C  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(2)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(1)    Iron  offer->(3)    Grain  offer->(3)   Cattle offer->(1)
```

```
Computer has loans:19, Emmanouil has:13 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Grain  from dock "Grain  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:2
            *Wood :3
            *Charcoal  :2
            *Iron :6
            *Grain :14
            *Cattle:12
            *Hides:1
Franc:2,Energy:9,Meal:0 Loans:19
Building Released:Charcoal Kiln player was 0

     ---- Turn No:3 of Round:9 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(C  |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(2)    Fish  offer->(3)    Wood  offer->(2)    Clay  offer-
>(1)    Iron  offer->(3)    Grain  offer->(0)   Cattle offer->(1)

Emmanouil has loans:13, Computer has:19 loans
player Emmanouil is in interest check!
     paying 1F for interest
player Computer is in interest check!
     paying 1F for interest
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
```

```
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:2
            *Wood :7
            *Charcoal  :7
            *Clay :19
            *Grain :10
            *Cattle:7
            *Hides:1
Franc:2,Energy:28,Meal:0 Loans:13

     ---- Turn No:4 of Round:9 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(3)    Fish  offer->(3)    Wood  offer->(3)    Clay  offer-
>(1)    Iron  offer->(3)    Grain  offer->(0)   Cattle offer->(0)

Computer has loans:19, Emmanouil has:13 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Franc from dock "Franc offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:4
            *Wood :3
            *Charcoal  :2
            *Iron :6
            *Grain :14
            *Cattle:12
            *Hides:1
Franc:4,Energy:9,Meal:0 Loans:19

     ---- Turn No:5 of Round:9 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(E  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(0)    Fish  offer->(4)    Wood  offer->(3)    Clay  offer-
>(2)    Iron  offer->(3)    Grain  offer->(0)   Cattle offer->(0)

Emmanouil has loans:13, Computer has:19 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
```

```
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 4 items of Fish  from dock "Fish  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:2
            *Fish :4
            *Wood :7
            *Charcoal  :7
            *Clay :19
            *Grain :10
            *Cattle:7
            *Hides:1
Franc:2,Energy:28,Meal:4 Loans:13

     ---- Turn No:6 of Round:9 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(E  |Fish  Clay  | )==(C  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(0)    Wood  offer->(3)    Clay  offer-
>(2)    Iron  offer->(4)    Grain  offer->(0)   Cattle offer->(0)

Computer has loans:19, Emmanouil has:13 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:4
            *Wood :6
            *Charcoal  :2
            *Iron :6
            *Grain :14
            *Cattle:12
            *Hides:1
Franc:4,Energy:12,Meal:0 Loans:19

     ---- Turn No:7 of Round:9 ----(playing Emmanouil)-----------
```

```
DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(C  |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(1)    Fish  offer->(0)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(4)    Grain  offer->(0)   Cattle offer->(1)

Emmanouil has loans:13, Computer has:19 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
M<101> player Emmanouil visits Wharf and paying 2 meals
State of Wharf:

Wharf must first be modernized in order to build a non Wooden ship
Player cannot Modernize the Wharf
trying to build Ship...
State of Wharf:

Wharf must first be modernized in order to build a non Wooden ship
Player cannot Modernize the Wharf
Total woods=14 plus total energy=14 that's enough to build the wooden ship!
Emmanouil managed to build Wooden ship!


I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:2
            *Fish :2
            *Charcoal  :6
            *Clay :19
            *Grain :11
            *Cattle:8
            *Hides:1
    Total [Franc:2,Energy:18,Meal:2] Loans:13

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:4
            *Wood :6
            *Charcoal  :2
            *Iron :6
            *Grain :15
            *Cattle:13
            *Hides:1
    Total [Franc:4,Energy:12,Meal:0] Loans:19
END OF ROUND, PAYING MEALS:17
----- Player:Emmanouil has just got 4 loans!
                Emmanouil paying 17 meals
----- Player:Computer has just got 4 loans!
                Computer paying 17 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:17
```

```
+-------------+


$$$$$$$$$$$$$$$$$$$$ END OF ROUND 10 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:40
TOTAL LOANS OF Emmanouil:17
TOTAL LOANS OF Computer:23


======== ROUND No:11============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:18
Standard Building
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:6
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4

** Iron SHIP CARD **    ( owner:no Owner )
value:10
buy Price:20
energy:3
Iron:4
Selling goods:3
Providing foods:5

** Steel SHIP CARD **    ( owner:no Owner )
value:16
buy Price:30
energy:3
Steel:2
Selling goods:4
Providing foods:7


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                +--[STARTING],  OWNER->Town --------index=0---+
                                |      Name:[Building Firm I
                                |  ID:2, Symbol:H , (+):No , visited:No
                                |             Value:4,  Cost:4
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |      Name:[Building Firm II]
```

```
                                |  ID:4, Symbol:H , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:

                               +--[STARTING],  OWNER->Town --------index=2---+
                                |      Name:[Construction Firm]
                                |  ID:6, Symbol:H , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                               +--[STANDARD],  OWNER->no Owner --------index=5-
--+
                                |      Name:[Abattoir]
                                |  ID:9, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Franc(2)
                                |        Building Cost:Wood(1)  Clay(1)  Iron(1)

Pile II
                               +--[STANDARD],  OWNER->no Owner --------index=6-
--+
                                |      Name:[Clay Mound]
                                |  ID:10, Symbol:- , (+):No , visited:No
                                |             Value:2,  Cost:2
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(1)
                                |        Building Cost:

Pile III
                               +--[STANDARD],  OWNER->no Owner --------
index=11---+
                                |      Name:[Smokehouse]
                                |  ID:8, Symbol:F , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(2) Franc(1)
                                |        Building Cost:Wood(2)  Clay(1)

======= End of Building Proposals =======
                               +--[STANDARD],  OWNER->Town --------index=3---+
                                |      Name:[Marketplace]
                                |  ID:1, Symbol:- , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(2) Franc(1)
                                |        Building Cost:Wood(2)
```

```
                          +--[STANDARD],  OWNER->Town --------index=9---+
                          |      Name:[Fishery]
                          |  ID:3, Symbol:F , (+):No , visited:No
                          |             Value:10,  Cost:10
                          |             Action:action
                          |     Type of building:Craftsman
                          |            entry Fee:
                          |        Building Cost:Wood(1)  Clay(1)

                          +--[STANDARD],  OWNER->Town --------index=4---+
                          |      Name:[Bakehouse]
                          |  ID:5, Symbol:- , (+):No , visited:No
                          |             Value:8,  Cost:8
                          |             Action:action
                          |     Type of building:Craftsman
                          |            entry Fee:Meals(1)
                          |        Building Cost:Clay(2)

                          +--[STANDARD],  OWNER->Human --------index=10---
+
                          |      Name:[Charcoal Kiln]
                          |  ID:7, Symbol:- , (+):No , visited:No
                          |             Value:8,  Cost:8
                          |             Action:action
                          |     Type of building:Craftsman
                          |            entry Fee:
                          |        Building Cost:Clay(1)


                          +--[SPECIAL],  OWNER->Town --------index=46---+
                          |      Name:[Masons Guild]
                          |  ID:47, Symbol:H , (+):No , visited:No
                          |             Value:8,  Cost:10
                          |             Action:action
                          |     Type of building:Craftsman
                          |            entry Fee:padlock
                          |        Building Cost:

                          +--[SPECIAL],  OWNER->Town --------index=37---+
                          |      Name:[Guildhouse]
                          |  ID:38, Symbol:H-F , (+):Yes , visited:No
                          |             Value:4,  Cost:8
                          |             Action:action
                          |     Type of building:Economic
                          |            entry Fee:padlock
                          |        Building Cost:

                          +--[SPECIAL],  OWNER->Town --------index=32---+
                          |      Name:[Fishpond and Wood]
                          |  ID:33, Symbol:F , (+):No , visited:No
                          |             Value:4,  Cost:4
                          |             Action:action
                          |     Type of building:NonBuildings
                          |            entry Fee:Meals(1)
                          |        Building Cost:

                          +--[SPECIAL],  OWNER->Town --------index=21---+
                          |      Name:[Baguette Shop]
                          |  ID:11, Symbol:- , (+):No , visited:No
                          |             Value:4,  Cost:4
```

```
|             Action:action
|     Type of building:Economic
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=38---+
|      Name:[Harbor Watch]
|  ID:39, Symbol:- , (+):No , visited:No
|             Value:6,  Cost:6
|             Action:action
|     Type of building:Public
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=53---+
|      Name:[Tavern]
|  ID:54, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:Economic
|            entry Fee:
|        Building Cost:
```

```
     ---- Turn No:1 of Round:10 ----(playing Computer)-----------

DISKS->[=(C  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(0)    Wood  offer->(2)    Clay  offer-
>(3)    Iron  offer->(4)    Grain  offer->(0)   Cattle offer->(1)

Computer has loans:23, Emmanouil has:17 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
```

```
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :6
            *Charcoal  :2
            *Iron :6
            *Grain :15
            *Cattle:14
            *Hides:1
Franc:3,Energy:12,Meal:0 Loans:23
Building Released:Wharf player was 0

     ---- Turn No:2 of Round:10 ----(playing Emmanouil)-----------

DISKS->[=(C  |Wood  Clay  | )==(E  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(1)    Fish  offer->(1)    Wood  offer->(2)    Clay  offer-
>(3)    Iron  offer->(4)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:17, Computer has:23 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Clay Mound |
Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 4 items of Iron  from dock "Iron  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Charcoal  :6
            *Clay :19
            *Iron :4
            *Grain :11
            *Cattle:8
            *Hides:1
Franc:3,Energy:18,Meal:0 Loans:17

     ---- Turn No:3 of Round:10 ----(playing Computer)-----------
```

```
DISKS->[=(   |Wood  Clay  | )==(E  |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(1)    Fish  offer->(2)    Wood  offer->(3)    Clay  offer-
>(3)    Iron  offer->(0)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:23, Emmanouil has:17 loans
player Emmanouil is in interest check!
     paying 1F for interest
player Computer is in interest check!
     paying 1F for interest
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
```

```
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:2
            *Wood :9
            *Charcoal  :2
            *Iron :6
            *Grain :15
            *Cattle:14
            *Hides:1
Franc:2,Energy:15,Meal:0 Loans:23

     ---- Turn No:4 of Round:10 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(2)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(3)    Iron  offer->(0)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:17, Computer has:23 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:2
            *Wood :1
            *Charcoal  :6
            *Clay :19
            *Iron :4
```

```
            *Grain :11
            *Cattle:8
            *Hides:1
Franc:2,Energy:19,Meal:0 Loans:17

     ---- Turn No:5 of Round:10 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(C  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(2)    Fish  offer->(3)    Wood  offer->(0)    Clay  offer-
>(4)    Iron  offer->(0)    Grain  offer->(1)   Cattle offer->(0)

Computer has loans:23, Emmanouil has:17 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Franc from dock "Franc offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:4
            *Wood :9
            *Charcoal  :2
            *Iron :6
            *Grain :15
            *Cattle:14
            *Hides:1
Franc:4,Energy:15,Meal:0 Loans:23

     ---- Turn No:6 of Round:10 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(C  |Fish  Clay  | )==(E  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(3)    Wood  offer->(0)    Clay  offer-
>(4)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(0)

Emmanouil has loans:17, Computer has:23 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Smokehouse |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
```

```
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 1 items of Franc from dock "Franc offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Wood :1
            *Charcoal  :6
            *Clay :19
            *Iron :4
            *Grain :11
            *Cattle:8
            *Hides:1
Franc:3,Energy:19,Meal:0 Loans:17

     ---- Turn No:7 of Round:10 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(E  |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(0)    Fish  offer->(3)    Wood  offer->(1)    Clay  offer-
>(4)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(1)

Computer has loans:23, Emmanouil has:17 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 3 items of Fish  from dock "Fish  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:4
            *Fish :3
            *Wood :9
            *Charcoal  :2
            *Iron :6
            *Grain :15
            *Cattle:14
```

```
            *Hides:1
Franc:4,Energy:15,Meal:3 Loans:23

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:3
            *Wood :1
            *Charcoal  :6
            *Clay :19
            *Iron :4
            *Grain :12
            *Cattle:9
            *Hides:1
    Total [Franc:3,Energy:19,Meal:0] Loans:17

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:4
            *Fish :3
            *Wood :9
            *Charcoal  :2
            *Iron :6
            *Grain :16
            *Cattle:15
            *Hides:1
    Total [Franc:4,Energy:15,Meal:3] Loans:23
END OF ROUND, PAYING MEALS:18
----- Player:Emmanouil has just got 4 loans!
                Emmanouil paying 18 meals
----- Player:Computer has just got 3 loans!
                Computer paying 18 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:18
Standard Building
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 11 $$$$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:47
TOTAL LOANS OF Emmanouil:21
TOTAL LOANS OF Computer:26


======== ROUND No:12============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:19
Special Building
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:6
```

```
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4

** Iron SHIP CARD **    ( owner:no Owner )
value:10
buy Price:20
energy:3
Iron:4
Selling goods:3
Providing foods:5

** Steel SHIP CARD **    ( owner:no Owner )
value:20
buy Price:30
energy:3
Steel:2
Selling goods:4
Providing foods:7


                    ((((((( Cards on board ))))))))
======== Starting Buildings ========
                                  +--[STARTING],  OWNER->Town --------index=0---+
                                  |      Name:[Building Firm I
                                  |  ID:2, Symbol:H , (+):No , visited:No
                                  |             Value:4,  Cost:4
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:
                                  |        Building Cost:

                                  +--[STARTING],  OWNER->Town --------index=1---+
                                  |      Name:[Building Firm II]
                                  |  ID:4, Symbol:H , (+):No , visited:No
                                  |             Value:6,  Cost:6
                                  |             Action:action
                                  |     Type of building:Craftsman
                                  |            entry Fee:Meals(1)
                                  |        Building Cost:

                                  +--[STARTING],  OWNER->Town --------index=2---+
                                  |      Name:[Construction Firm]
                                  |  ID:6, Symbol:H , (+):No , visited:No
                                  |             Value:8,  Cost:8
                                  |             Action:action
                                  |     Type of building:Industrial
                                  |            entry Fee:Meals(2)
                                  |        Building Cost:

===== End of Starting Buildings =====

======== Building Proposals ========
Pile I
                                  +--[STANDARD],  OWNER->no Owner --------index=5-
--+
                                  |      Name:[Abattoir]
                                  |  ID:9, Symbol:- , (+):No , visited:No
                                  |             Value:8,  Cost:8
```

```
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Franc(2)
                                |        Building Cost:Wood(1)  Clay(1)  Iron(1)

Pile II
                               +--[STANDARD],  OWNER->no Owner --------index=6-
--+
                                |      Name:[Clay Mound]
                                |  ID:10, Symbol:- , (+):No , visited:No
                                |             Value:2,  Cost:2
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(1)
                                |        Building Cost:

Pile III
                               +--[STANDARD],  OWNER->no Owner --------
index=13---+
                                |      Name:[Wharf]
                                |  ID:12, Symbol:- , (+):No , visited:No
                                |             Value:14,  Cost:14
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:Wood(2)  Clay(2)  Iron(2)

======= End of Building Proposals =======
                               +--[STANDARD],  OWNER->Town --------index=3---+
                                |      Name:[Marketplace]
                                |  ID:1, Symbol:- , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(2) Franc(1)
                                |        Building Cost:Wood(2)

                               +--[STANDARD],  OWNER->Town --------index=9---+
                                |      Name:[Fishery]
                                |  ID:3, Symbol:F , (+):No , visited:No
                                |             Value:10,  Cost:10
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:Wood(1)  Clay(1)

                               +--[STANDARD],  OWNER->Town --------index=4---+
                                |      Name:[Bakehouse]
                                |  ID:5, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:Clay(2)

                               +--[STANDARD],  OWNER->Human --------index=10---
+
                                |      Name:[Charcoal Kiln]
                                |  ID:7, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
```

```
|     Type of building:Craftsman
|            entry Fee:
|       Building Cost:Clay(1)

+--[STANDARD],  OWNER->Town --------index=11---+
|      Name:[Smokehouse]
|  ID:8, Symbol:F , (+):No , visited:No
|             Value:6,  Cost:6
|             Action:action
|     Type of building:Craftsman
|            entry Fee:Meals(2) Franc(1)
|       Building Cost:Wood(2)  Clay(1)


+--[SPECIAL],  OWNER->Town --------index=46---+
|      Name:[Masons Guild]
|  ID:47, Symbol:H , (+):No , visited:No
|             Value:8,  Cost:10
|             Action:action
|     Type of building:Craftsman
|            entry Fee:padlock
|       Building Cost:

+--[SPECIAL],  OWNER->Town --------index=37---+
|      Name:[Guildhouse]
|  ID:38, Symbol:H-F , (+):Yes , visited:No
|             Value:4,  Cost:8
|             Action:action
|     Type of building:Economic
|            entry Fee:padlock
|       Building Cost:

+--[SPECIAL],  OWNER->Town --------index=32---+
|      Name:[Fishpond and Wood]
|  ID:33, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:NonBuildings
|            entry Fee:Meals(1)
|       Building Cost:

+--[SPECIAL],  OWNER->Town --------index=21---+
|      Name:[Baguette Shop]
|  ID:11, Symbol:- , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:Economic
|            entry Fee:Meals(1)
|       Building Cost:

+--[SPECIAL],  OWNER->Town --------index=38---+
|      Name:[Harbor Watch]
|  ID:39, Symbol:- , (+):No , visited:No
|             Value:6,  Cost:6
|             Action:action
|     Type of building:Public
|            entry Fee:Meals(1)
|       Building Cost:

+--[SPECIAL],  OWNER->Town --------index=53---+
|      Name:[Tavern]
```

```
|  ID:54, Symbol:F , (+):No , visited:No
|            Value:4,  Cost:4
|            Action:action
|     Type of building:Economic
|           entry Fee:
|        Building Cost:



     ---- Turn No:1 of Round:11 ----(playing Emmanouil)-----------

DISKS->[=(E  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(0)    Fish  offer->(0)    Wood  offer->(2)    Clay  offer-
>(5)    Iron  offer->(1)    Grain  offer->(1)   Cattle offer->(1)

Emmanouil has loans:21, Computer has:26 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Wharf |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
Player Emmanouil visits the standard building Smokehouse
before
            **** Items ****
            *Wood :1
            *Charcoal  :6
            *Clay :19
            *Iron :4
            *Grain :12
            *Cattle:9
            *Hides:1
action for Smokehouse

action: "At the cost of 1 energy(overall) upgrade up to 6 fish into smoked fish
'receiving 1/2 franc(rounded down) for each'

Total Number of smokedfish by paing 1 energy gained: 0
Total Number of franc gained: 0.0
after
            **** Items ****
            *Wood :1
            *Charcoal  :5
            *Clay :19
            *Iron :4
            *Grain :12
            *Cattle:9
            *Hides:1
```

```
     ---- Turn No:2 of Round:11 ----(playing Computer)-----------

DISKS->[=(E  |Wood  Clay  | )==(C  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(0)    Fish  offer->(1)    Wood  offer->(2)    Clay  offer-
>(5)    Iron  offer->(1)    Grain  offer->(2)   Cattle offer->(1)

Computer has loans:26, Emmanouil has:21 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :9
            *Charcoal  :2
            *Iron :6
            *Grain :16
            *Cattle:16
            *Hides:1
Franc:1,Energy:15,Meal:0 Loans:26
Building Released:Smokehouse player was 0

     ---- Turn No:3 of Round:11 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(C  |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(0)    Fish  offer->(2)    Wood  offer->(3)    Clay  offer-
>(5)    Iron  offer->(1)    Grain  offer->(2)   Cattle offer->(0)

Emmanouil has loans:21, Computer has:26 loans
player Emmanouil is in interest check!
     paying 1F for interest
----- Player:Emmanouil has just got 1 loan!
player Computer is in interest check!
     paying 1F for interest
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Wharf |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
```

```
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Wharf |
<M300> Player Emmanouil has build the Wharf!

     ---- Turn No:4 of Round:11 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(2)    Wood  offer->(4)    Clay  offer-
>(5)    Iron  offer->(1)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:26, Emmanouil has:22 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 4 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :13
            *Charcoal  :2
            *Iron :6
            *Grain :16
            *Cattle:16
            *Hides:1
Franc:0,Energy:19,Meal:0 Loans:26

     ---- Turn No:5 of Round:11 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(E  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(3)    Wood  offer->(0)    Clay  offer-
>(6)    Iron  offer->(1)    Grain  offer->(2)   Cattle offer->(0)

Emmanouil has loans:22, Computer has:26 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
```

```
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 3 items of Fish  from dock "Fish  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Fish :3
            *Wood :1
            *Charcoal  :5
            *Clay :19
            *Iron :4
            *Grain :12
            *Cattle:9
            *Hides:1
Franc:3,Energy:16,Meal:3 Loans:22

     ---- Turn No:6 of Round:11 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(E  |Fish  Clay  | )==(C  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(2)    Fish  offer->(0)    Wood  offer->(0)    Clay  offer-
>(6)    Iron  offer->(2)    Grain  offer->(2)   Cattle offer->(0)

Computer has loans:26, Emmanouil has:22 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
```

```
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to build building
Player Computer can build the following standard buildings:Clay Mound |
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Iron  from dock "Iron  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :13
            *Charcoal  :2
            *Iron :8
            *Grain :16
            *Cattle:16
            *Hides:1
Franc:0,Energy:19,Meal:0 Loans:26

     ---- Turn No:7 of Round:11 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(C  |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(2)    Fish  offer->(0)    Wood  offer->(1)    Clay  offer-
>(6)    Iron  offer->(0)    Grain  offer->(2)   Cattle offer->(1)

Emmanouil has loans:22, Computer has:26 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Emmanouil is getting items from dock

Taking 6 items of Clay  from dock "Clay  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:3
            *Fish :3
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :12
            *Cattle:9
```

```
            *Hides:1
Franc:3,Energy:16,Meal:3 Loans:22

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:3
            *Fish :3
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :13
            *Cattle:10
            *Hides:1
    Total [Franc:3,Energy:16,Meal:3] Loans:22

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Wood :13
            *Charcoal  :2
            *Iron :8
            *Grain :17
            *Cattle:17
            *Hides:1
    Total [Franc:0,Energy:19,Meal:0] Loans:26
END OF ROUND, PAYING MEALS:19
----- Player:Emmanouil has just got 4 loans!
                Emmanouil paying 19 meals
----- Player:Computer has just got 5 loans!
                Computer paying 19 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:19
Special Building
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 12 $$$$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:57
TOTAL LOANS OF Emmanouil:26
TOTAL LOANS OF Computer:31


======== ROUND No:13============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:20
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:6
buy Price:14
Wood:5
```

```
energy:3
Selling goods:2
Providing foods:4

** Iron SHIP CARD **    ( owner:no Owner )
value:10
buy Price:20
energy:3
Iron:4
Selling goods:3
Providing foods:5

** Steel SHIP CARD **    ( owner:no Owner )
value:24
buy Price:30
energy:3
Steel:2
Selling goods:4
Providing foods:7


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                +--[STARTING],  OWNER->Town --------index=0---+
                                |      Name:[Building Firm I
                                |  ID:2, Symbol:H , (+):No , visited:No
                                |             Value:4,  Cost:4
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |      Name:[Building Firm II]
                                |  ID:4, Symbol:H , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=2---+
                                |      Name:[Construction Firm]
                                |  ID:6, Symbol:H , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                                +--[STANDARD],  OWNER->no Owner --------index=5-
--+
                                |      Name:[Abattoir]
                                |  ID:9, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
```

```
                                |            entry Fee:Franc(2)
                                |        Building Cost:Wood(1)  Clay(1)  Iron(1)

Pile II
                               +--[STANDARD],  OWNER->no Owner --------index=6-
--+
                                |      Name:[Clay Mound]
                                |  ID:10, Symbol:- , (+):No , visited:No
                                |             Value:2,  Cost:2
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(1)
                                |        Building Cost:

Pile III
                               +--[STANDARD],  OWNER->no Owner --------
index=12---+
                                |      Name:[Brickworks]
                                |  ID:14, Symbol:- , (+):No , visited:No
                                |             Value:14,  Cost:14
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(1)
                                |        Building Cost:Wood(2)  Clay(1)  Iron(1)

======= End of Building Proposals =======
                               +--[STANDARD],  OWNER->Town --------index=3---+
                                |      Name:[Marketplace]
                                |  ID:1, Symbol:- , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(2) Franc(1)
                                |        Building Cost:Wood(2)

                               +--[STANDARD],  OWNER->Town --------index=9---+
                                |      Name:[Fishery]
                                |  ID:3, Symbol:F , (+):No , visited:No
                                |             Value:10,  Cost:10
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:Wood(1)  Clay(1)

                               +--[STANDARD],  OWNER->Town --------index=4---+
                                |      Name:[Bakehouse]
                                |  ID:5, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:Clay(2)

                               +--[STANDARD],  OWNER->Human --------index=10---
+
                                |      Name:[Charcoal Kiln]
                                |  ID:7, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
```

```
|        Building Cost:Clay(1)

+--[STANDARD],  OWNER->Town --------index=11---+
|      Name:[Smokehouse]
|  ID:8, Symbol:F , (+):No , visited:No
|             Value:6,  Cost:6
|             Action:action
|     Type of building:Craftsman
|            entry Fee:Meals(2) Franc(1)
|        Building Cost:Wood(2)  Clay(1)

+--[STANDARD],  OWNER->Human --------index=13---
+
|      Name:[Wharf]
|  ID:12, Symbol:- , (+):No , visited:No
|             Value:14,  Cost:14
|             Action:action
|     Type of building:Industrial
|            entry Fee:Meals(2)
|        Building Cost:Wood(2)  Clay(2)  Iron(2)


+--[SPECIAL],  OWNER->Town --------index=46---+
|      Name:[Masons Guild]
|  ID:47, Symbol:H , (+):No , visited:No
|             Value:8,  Cost:10
|             Action:action
|     Type of building:Craftsman
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=37---+
|      Name:[Guildhouse]
|  ID:38, Symbol:H-F , (+):Yes , visited:No
|             Value:4,  Cost:8
|             Action:action
|     Type of building:Economic
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=32---+
|      Name:[Fishpond and Wood]
|  ID:33, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:NonBuildings
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=21---+
|      Name:[Baguette Shop]
|  ID:11, Symbol:- , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:Economic
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=38---+
|      Name:[Harbor Watch]
|  ID:39, Symbol:- , (+):No , visited:No
```

```
|             Value:6,  Cost:6
|             Action:action
|     Type of building:Public
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=53---+
|      Name:[Tavern]
|  ID:54, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:Economic
|            entry Fee:
|        Building Cost:



     ---- Turn No:1 of Round:12 ----(playing Computer)-----------

DISKS->[=(C  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(2)    Fish  offer->(0)    Wood  offer->(2)    Clay  offer-
>(1)    Iron  offer->(0)    Grain  offer->(2)   Cattle offer->(1)

Computer has loans:31, Emmanouil has:26 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:1
            *Wood :15
            *Charcoal  :2
            *Iron :8
            *Grain :17
            *Cattle:17
            *Hides:1
Franc:1,Energy:21,Meal:0 Loans:31

     ---- Turn No:2 of Round:12 ----(playing Emmanouil)-----------

DISKS->[=(C  |Wood  Clay  | )==(E  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(0)    Clay  offer-
>(1)    Iron  offer->(0)    Grain  offer->(3)   Cattle offer->(1)

Emmanouil has loans:26, Computer has:31 loans
~~~~~~~~~~~~~~~~~~~
```

```
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 2 items of Franc from dock "Franc offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:5
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :13
            *Cattle:10
            *Hides:1
Franc:5,Energy:16,Meal:0 Loans:26

     ---- Turn No:3 of Round:12 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(E  |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(0)    Fish  offer->(2)    Wood  offer->(1)    Clay  offer-
>(1)    Iron  offer->(0)    Grain  offer->(3)   Cattle offer->(1)

Computer has loans:31, Emmanouil has:26 loans
player Emmanouil is in interest check!
     paying 1F for interest
player Computer is in interest check!
     paying 1F for interest
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 1 items of Cattle from dock "Cattle offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :15
            *Charcoal  :2
            *Iron :8
            *Grain :17
            *Cattle:18
            *Hides:1
Franc:0,Energy:21,Meal:0 Loans:31

     ---- Turn No:4 of Round:12 ----(playing Emmanouil)-----------
```

```
DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(C  |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(2)    Wood  offer->(2)    Clay  offer-
>(1)    Iron  offer->(0)    Grain  offer->(3)   Cattle offer->(0)

Emmanouil has loans:26, Computer has:31 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
...can't visit Wharf
Player Emmanouil is getting items from dock

Taking 2 items of Fish  from dock "Fish  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:4
            *Fish :2
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :13
            *Cattle:10
            *Hides:1
Franc:4,Energy:16,Meal:2 Loans:26

     ---- Turn No:5 of Round:12 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(E  |Wood  Franc | )==(C  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(1)    Wood  offer->(2)    Clay  offer-
>(2)    Iron  offer->(0)    Grain  offer->(3)   Cattle offer->(0)

Computer has loans:31, Emmanouil has:26 loans
```

```
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :17
            *Charcoal  :2
            *Iron :8
            *Grain :17
            *Cattle:18
            *Hides:1
Franc:0,Energy:23,Meal:0 Loans:31

     ---- Turn No:6 of Round:12 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(C  |Fish  Clay  | )==(E  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(0)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(3)   Cattle offer->(0)

Emmanouil has loans:26, Computer has:31 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
~~~~~~~~~~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
Player Emmanouil visits the standard building Bakehouse
before
            **** Items ****
            *Franc:4
            *Fish :1
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :13
            *Cattle:10
            *Hides:1
action for Bakehouse

action: "Upgrade any number of grain to bread by paying 1/2 energy(rounded up)
per grain and receive 1/2 franc(rounded down) per bread"

Total Number of bread gained: 12
Total Energy spent: 0
Total Number of franc gained: 6.0
after
            **** Items ****
            *Franc:10
            *Fish :1
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :1
            *Bread     :12
            *Cattle:10
            *Hides:1

     ---- Turn No:7 of Round:12 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(E  |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(2)    Fish  offer->(1)    Wood  offer->(1)    Clay  offer-
>(2)    Iron  offer->(1)    Grain  offer->(3)   Cattle offer->(1)

Computer has loans:31, Emmanouil has:26 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
```

```
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Franc from dock "Franc offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:2
            *Wood :17
            *Charcoal  :2
            *Iron :8
            *Grain :17
            *Cattle:18
            *Hides:1
Franc:2,Energy:23,Meal:0 Loans:31

I'm Emmanouil, my arraylist after Harvest is:
            **** Items ****
            *Franc:10
            *Fish :1
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :2
            *Bread     :12
            *Cattle:11
            *Hides:1
    Total [Franc:10,Energy:16,Meal:25] Loans:26

I'm Computer, my arraylist after Harvest is:
            **** Items ****
            *Franc:2
            *Wood :17
            *Charcoal  :2
            *Iron :8
            *Grain :18
            *Cattle:19
            *Hides:1
```

```
    Total [Franc:2,Energy:23,Meal:0] Loans:31
END OF ROUND, PAYING MEALS:20
                Emmanouil paying 20 meals
----- Player:Computer has just got 5 loans!
                Computer paying 20 meals

+--ROUND CARD--+
HARVEST  Grain:1 Cattle:2
Food:20
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 13 $$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:62
TOTAL LOANS OF Emmanouil:26
TOTAL LOANS OF Computer:36


======== ROUND No:14============

[---- Round & Ship Cards ----]

+--ROUND CARD--+
NO HARVEST
Food:20
+-------------+



    [--------(Piles)---------]
** Wooden SHIP CARD **    ( owner:no Owner )
value:6
buy Price:14
Wood:5
energy:3
Selling goods:2
Providing foods:4

** Iron SHIP CARD **    ( owner:no Owner )
value:10
buy Price:20
energy:3
Iron:4
Selling goods:3
Providing foods:5

** Steel SHIP CARD **    ( owner:no Owner )
value:24
buy Price:30
energy:3
Steel:2
Selling goods:4
Providing foods:7

** Luxury SHIP CARD **    ( owner:no Owner )
value:34
Cannot be bought
energy:3
Steel:3
Selling goods:0
```

```
Providing foods:0


                    ((((((( Cards on board ))))))))
======= Starting Buildings =======
                                +--[STARTING],  OWNER->Town --------index=0---+
                                |      Name:[Building Firm I
                                |  ID:2, Symbol:H , (+):No , visited:No
                                |             Value:4,  Cost:4
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=1---+
                                |      Name:[Building Firm II]
                                |  ID:4, Symbol:H , (+):No , visited:No
                                |             Value:6,  Cost:6
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Meals(1)
                                |        Building Cost:

                                +--[STARTING],  OWNER->Town --------index=2---+
                                |      Name:[Construction Firm]
                                |  ID:6, Symbol:H , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Industrial
                                |            entry Fee:Meals(2)
                                |        Building Cost:

===== End of Starting Buildings =====

======= Building Proposals =======
Pile I
                                +--[STANDARD],  OWNER->no Owner --------index=5-
--+
                                |      Name:[Abattoir]
                                |  ID:9, Symbol:- , (+):No , visited:No
                                |             Value:8,  Cost:8
                                |             Action:action
                                |     Type of building:Craftsman
                                |            entry Fee:Franc(2)
                                |        Building Cost:Wood(1)  Clay(1)  Iron(1)

Pile II
                                +--[STANDARD],  OWNER->no Owner --------index=6-
--+
                                |      Name:[Clay Mound]
                                |  ID:10, Symbol:- , (+):No , visited:No
                                |             Value:2,  Cost:2
                                |             Action:action
                                |     Type of building:NonBuildings
                                |            entry Fee:Meals(1)
                                |        Building Cost:

Pile III
                                +--[STANDARD],  OWNER->no Owner --------
index=12---+
                                |      Name:[Brickworks]
```

```
                                |  ID:14, Symbol:- , (+):No , visited:No
                                |             Value:14,  Cost:14
                                |             Action:action
                                |    Type of building:Industrial
                                |            entry Fee:Meals(1)
                                |       Building Cost:Wood(2)  Clay(1)  Iron(1)

======= End of Building Proposals =======
                               +--[STANDARD],  OWNER->Town --------index=3---+
                               |     Name:[Marketplace]
                               |  ID:1, Symbol:- , (+):No , visited:No
                               |             Value:6,  Cost:6
                               |             Action:action
                               |    Type of building:NonBuildings
                               |            entry Fee:Meals(2) Franc(1)
                               |       Building Cost:Wood(2)

                               +--[STANDARD],  OWNER->Town --------index=9---+
                               |     Name:[Fishery]
                               |  ID:3, Symbol:F , (+):No , visited:No
                               |             Value:10,  Cost:10
                               |             Action:action
                               |    Type of building:Craftsman
                               |            entry Fee:
                               |       Building Cost:Wood(1)  Clay(1)

                               +--[STANDARD],  OWNER->Town --------index=4---+
                               |     Name:[Bakehouse]
                               |  ID:5, Symbol:- , (+):No , visited:Yes
                               |             Value:8,  Cost:8
                               |             Action:action
                               |    Type of building:Craftsman
                               |            entry Fee:Meals(1)
                               |       Building Cost:Clay(2)

                               +--[STANDARD],  OWNER->Human --------index=10---
+
                               |     Name:[Charcoal Kiln]
                               |  ID:7, Symbol:- , (+):No , visited:No
                               |             Value:8,  Cost:8
                               |             Action:action
                               |    Type of building:Craftsman
                               |            entry Fee:
                               |       Building Cost:Clay(1)

                               +--[STANDARD],  OWNER->Town --------index=11---+
                               |     Name:[Smokehouse]
                               |  ID:8, Symbol:F , (+):No , visited:No
                               |             Value:6,  Cost:6
                               |             Action:action
                               |    Type of building:Craftsman
                               |            entry Fee:Meals(2) Franc(1)
                               |       Building Cost:Wood(2)  Clay(1)

                               +--[STANDARD],  OWNER->Human --------index=13---
+
                               |     Name:[Wharf]
                               |  ID:12, Symbol:- , (+):No , visited:No
                               |             Value:14,  Cost:14
                               |             Action:action
                               |    Type of building:Industrial
```

```
|            entry Fee:Meals(2)
|        Building Cost:Wood(2)  Clay(2)  Iron(2)


+--[SPECIAL],  OWNER->Town --------index=46---+
|      Name:[Masons Guild]
|  ID:47, Symbol:H , (+):No , visited:No
|             Value:8,  Cost:10
|             Action:action
|     Type of building:Craftsman
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=37---+
|      Name:[Guildhouse]
|  ID:38, Symbol:H-F , (+):Yes , visited:No
|             Value:4,  Cost:8
|             Action:action
|     Type of building:Economic
|            entry Fee:padlock
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=32---+
|      Name:[Fishpond and Wood]
|  ID:33, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:NonBuildings
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=21---+
|      Name:[Baguette Shop]
|  ID:11, Symbol:- , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:Economic
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=38---+
|      Name:[Harbor Watch]
|  ID:39, Symbol:- , (+):No , visited:No
|             Value:6,  Cost:6
|             Action:action
|     Type of building:Public
|            entry Fee:Meals(1)
|        Building Cost:

+--[SPECIAL],  OWNER->Town --------index=53---+
|      Name:[Tavern]
|  ID:54, Symbol:F , (+):No , visited:No
|             Value:4,  Cost:4
|             Action:action
|     Type of building:Economic
|            entry Fee:
|        Building Cost:


Building Released:Bakehouse player was 0
```

```
     ---- Turn No:1 of Round:13 ----(playing Emmanouil)-----------

DISKS->[=(E  |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(C  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(0)    Fish  offer->(1)    Wood  offer->(2)    Clay  offer-
>(3)    Iron  offer->(1)    Grain  offer->(3)   Cattle offer->(1)

Emmanouil has loans:26, Computer has:36 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II | Construction Firm |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
Player Emmanouil visits the standard building Wharf
before
            **** Items ****
            *Franc:10
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :2
            *Bread     :1
            *Cattle:11
            *Hides:1
State of Wharf:

Wharf must first be modernized in order to build a non Wooden ship
Player cannot Modernize the Wharf
after
            **** Items ****
            *Franc:10
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :2
            *Bread     :1
            *Cattle:11
            *Hides:1

     ---- Turn No:2 of Round:13 ----(playing Computer)-----------

DISKS->[=(E  |Wood  Clay  | )==(C  |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Grain  to dock "Grain  offer".
```

```
Docks:  Franc offer->(0)    Fish  offer->(2)    Wood  offer->(2)    Clay  offer-
>(3)    Iron  offer->(1)    Grain  offer->(4)   Cattle offer->(1)

Computer has loans:36, Emmanouil has:26 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can buy the following Standard Buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
Player Computer visits the standard building Smokehouse
before
            **** Items ****
            *Franc:1
            *Wood :17
            *Charcoal  :2
            *Iron :8
            *Grain :18
            *Cattle:19
            *Hides:1
action for Smokehouse

action: "At the cost of 1 energy(overall) upgrade up to 6 fish into smoked fish
'receiving 1/2 franc(rounded down) for each'

Total Number of smokedfish by paing 1 energy gained: 0
Total Number of franc gained: 0.0
after
            **** Items ****
            *Franc:1
            *Wood :16
            *Charcoal  :2
            *Iron :8
            *Grain :18
            *Cattle:19
            *Hides:1
Building Released:Wharf player was 0

     ---- Turn No:3 of Round:13 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(C  |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Wood  to dock "Wood  offer".
Docks:  Franc offer->(0)    Fish  offer->(3)    Wood  offer->(3)    Clay  offer-
>(3)    Iron  offer->(1)    Grain  offer->(4)   Cattle offer->(1)

Emmanouil has loans:26, Computer has:36 loans
player Emmanouil is in interest check!
     paying 1F for interest
player Computer is in interest check!
     paying 1F for interest
```

```
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II | Construction Firm |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Emmanouil is getting items from dock

Taking 4 items of Grain  from dock "Grain  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:9
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :6
            *Bread     :1
            *Cattle:11
            *Hides:1
Franc:9,Energy:16,Meal:2 Loans:26
Building Released:Smokehouse player was 1

     ---- Turn No:4 of Round:13 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(E  |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(   |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(1)    Fish  offer->(3)    Wood  offer->(4)    Clay  offer-
>(3)    Iron  offer->(1)    Grain  offer->(0)   Cattle offer->(1)

Computer has loans:36, Emmanouil has:26 loans
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
...can't visit Wharf
Player Computer is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 4 items of Wood  from dock "Wood  offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Wood :20
            *Charcoal  :2
            *Iron :8
```

```
            *Grain :18
            *Cattle:19
            *Hides:1
Franc:0,Energy:26,Meal:0 Loans:36

     ---- Turn No:5 of Round:13 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(C  |Wood  Franc | )==(E  |Fish  Clay  | )==(   |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Fish  to dock "Fish  offer".    Added 1 Clay  to dock "Clay  offer".
Docks:  Franc offer->(1)    Fish  offer->(4)    Wood  offer->(0)    Clay  offer-
>(4)    Iron  offer->(1)    Grain  offer->(0)   Cattle offer->(1)

Emmanouil has loans:26, Computer has:36 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II | Construction Firm |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
check for visit Wharf!
Player Emmanouil is getting items from dock

Taking 4 items of Fish  from dock "Fish  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:9
            *Fish :4
            *Wood :1
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :6
            *Bread     :1
            *Cattle:11
            *Hides:1
Franc:9,Energy:16,Meal:6 Loans:26

     ---- Turn No:6 of Round:13 ----(playing Computer)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(E  |Fish  Clay  | )==(C  |Iron  Franc | )==(   |Wood
Cattle | )=]
Added 1 Iron  to dock "Iron  offer".    Added 1 Franc to dock "Franc offer".
Docks:  Franc offer->(2)    Fish  offer->(0)    Wood  offer->(0)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(0)   Cattle offer->(1)

Computer has loans:36, Emmanouil has:26 loans
```

```
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Computer can build the following standard buildings:Clay Mound |
check for visit Wharf!
...can't visit Wharf
Player Computer is getting items from dock

Taking 2 items of Franc from dock "Franc offer"
I'm Computer, my arraylist has:
            **** Items ****
            *Franc:2
            *Wood :20
            *Charcoal  :2
            *Iron :8
            *Grain :18
            *Cattle:19
            *Hides:1
Franc:2,Energy:26,Meal:0 Loans:36

     ---- Turn No:7 of Round:13 ----(playing Emmanouil)-----------

DISKS->[=(   |Wood  Clay  | )==(   |Fish  Grain  | )==(   |Fish  Wood  INTEREST|
)==(   |Wood  Franc | )==(   |Fish  Clay  | )==(C  |Iron  Franc | )==(E  |Wood
Cattle | )=]
Added 1 Wood  to dock "Wood  offer".    Added 1 Cattle to dock "Cattle offer".
Docks:  Franc offer->(0)    Fish  offer->(0)    Wood  offer->(1)    Clay  offer-
>(4)    Iron  offer->(2)    Grain  offer->(0)   Cattle offer->(2)

Emmanouil has loans:26, Computer has:36 loans
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can build the following standard buildings:Abattoir | Clay
Mound | Brickworks |
~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~
Player Emmanouil can buy the following Starting Buildings:Building Firm I> |
Building Firm II | Construction Firm |
~~~~~~~~~~~~~~~~~~~
check for visit Wharf!
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
Player Emmanouil is trying to visit building
EntryFee Of Building: 2 1 Marketplace
EntryFee Of Building: 0 0 Charcoal Kiln
EntryFee Of Building: 0 0 Fishery
```

```
EntryFee Of Building: 1 0 Bakehouse
EntryFee Of Building: 2 1 Smokehouse
EntryFee Of Building: 2 0 Wharf
check for visit Wharf!
Player Emmanouil is getting items from dock

Taking 1 items of Wood  from dock "Wood  offer"
I'm Emmanouil, my arraylist has:
            **** Items ****
            *Franc:9
            *Fish :4
            *Wood :2
            *Charcoal  :5
            *Clay :25
            *Iron :4
            *Grain :6
            *Bread     :1
            *Cattle:11
            *Hides:1
Franc:9,Energy:17,Meal:6 Loans:26
END OF ROUND, PAYING MEALS:20
----- Player:Emmanouil has just got 2 loans!
                Emmanouil paying 20 meals
----- Player:Computer has just got 5 loans!
                Computer paying 20 meals

+--ROUND CARD--+
NO HARVEST
Food:20
+-------------+



$$$$$$$$$$$$$$$$$$$$ END OF ROUND 14 $$$$$$$$$$$$$$$$$$$$$$$$$$$$
Total Loans:69
TOTAL LOANS OF Emmanouil:28
TOTAL LOANS OF Computer:41
```

6 ΚΕΦΑΛΑΙΟ 6ο : ΣΥΜΠΕΡΑΣΜΑΤΑ & ΜΕΛΛΟΝΤΙΚΕΣ ΠΡΟΟΠΤΙΚΕΣ

Στην εργασία αυτή μελετήθηκε και αναπτύχθηκε λογισμικό για μοντέλο επιτραπέζιου παιχνιδιού. Η ανάπτυξη στηρίχτηκε στη γλώσσα προγραμματισμού Java με τη χρήση του αναπτυξιακού περιβάλλοντος Apache NetBeans. Κατά την ανάπτυξη της εφαρμογής μας συναντήσαμε αρκετές δυσκολίες οι οποίες αφορούσαν το τρόπο παρουσίασης της ροής του παιχνιδιού με λεκτικό τρόπο και όχι γραφικό. Επίσης χρησιμοποιήσαμε μοντέλα τεχνητής νοημοσύνης που καθοδηγούσαν τη ροή του παιχνιδιού προς όφελος του εκάστοτε παίκτη.

Ως μελλοντική προοπτική αυτής της εργασίας θα μπορούσε να είναι η ανάπτυξη γραφικού περιβάλλοντος για τη ροή του παιχνιδιού το οποίο θα βοηθούσε και τον παίκτη να παρακολουθεί την εξέλιξη του αλλά και η προσθήκη επιπλέον τεχνητής ευφυίας αυξάνοντας το συντελεστή δυσκολίας για τον παίκτη σε σχέση με τον παίκτη που καθοδηγείται από τον υπολογιστή. Τα συγκεκριμένα δύο στοιχεία δεν μελετήθηκαν καθώς ήταν εκτός των πλαισίων της παρούσας διπλωματικής.

Βιβλιογραφία – Αναφορές - Διαδικτυακές Πηγές

[1] - Buchanan, Bruce G. (Winter 2005), "A (Very) Brief History of Artificial Intelligence" (PDF), AI Magazine, pp. 53–60, archived from the original (PDF) on 26 September 2007, retrieved 30 August 2007.
[2] - Σ. Τζαφέστας, Εισαγωγή στην Τεχνητή Νοημοσύνη και τα Έμπειρα Συστήματα, τόμος Α', 3η έκδοση, 2002.
[3] - https://blog.scienceandmediamuseum.org.uk/how-artificial-intelligence-is-used-in-video-games/
[4] - LeHavre PDF rules, https://www.fgbradleys.com/rules/rules2/lehavre-rules.pdf
[5] - Apache NetBeans 16, NetBeans, https://netbeans.apache.org/

Παράρτημα Α

Στο παράρτημα Α παρατίθενται οι κώδικες των κλάσεων της εφαρμογής

```
  public class Const
{

    String prioBuy[]={"Marketplace","Colliery","Cokery"};

    String prioVisit[]={"Wharf","Marketplace","Charcoal Kiln","Clay
Mound","Brickworks","Colliery","Cokery","Bakehouse","Abatoir","Tannery","Smokeho
use","Church","Steel Mill","Ironworks"};


    String prioBuild[]={"Charcoal Kiln","Marketplace","Colliery","Cokery"};


    int weightOfPayLoan[]={20,80};

    int weight[]={20,20,5,5,5,5,40};
    int empty=-1;
    int None=0;
    int Human=0,Computer=1,Town=2,noOwner=3,notVisited=4;
    int Wooden=0,Iron=1,Steel=2,Luxury=3;
    String typeName[]={"Wooden","Iron","Steel","Luxury"};
    int starting=0 ,standard=1,special=2;
    int Craftsman=0,Economic=1,Industrial=2,Public=3,NonBuildings=4,Yacht=5;
    int padLock=-1;
    int franc=0,fish=1,wood=2,clay=3,iron=4,grain=5,cattle=6,coal=7,hides=8;
    int
smokedfish=11,charcoal=12,brick=13,steel=14,bread=15,meat=16,coke=17,leather=18;
    int noBuilding=0,standardBuilding=1,specialBuilding=2;
    int hammer=1,fisherman=2,hammerANDfisherman=3,hammerAND2fisherman=4;
    int nothing=-1,start=0,stand=1;



   public int
indexesOfStandardBuildingCardsThatPlayersCanBuild[][]={{None,None,None},{None,No
ne,None}};
   public void S(Object o)
   {
       System.out.println(o);
   }
   public void Sn(Object o)
   {
       System.out.print(o);
   }

    String name1[]={"Franc",
                    "Fish ",
                    "Wood ",
                    "Clay ",
                    "Iron ",
                    "Grain ",
                    "Cattle",
                    "Coal",
                    "Hides"};
```

```
    String name2[]={"Franc     ",
                    "Smokedfish",
                    "Charcoal  ",
                    "Brick     ",
                    "Steel     ",
                    "Bread     ",
                    "Meat      ",
                    "Coke      ",
                    "Leather   "};

    int value1[]={1,0,0,0,0,0,0,0,0};
    int  meal1[]={0,1,0,0,0,0,0,0,0};
    int energy1[]={0,0,1,0,0,0,0,1,0};


    int value2[]={1,0,0,0,0,0,0,0,0};
    int  meal2[]={0,2,0,0,0,2,3,0,0};
    int energy2[]={0,0,3,0,0,0,0,10,0};


    int
typeOfBuildingInRoundCard[]={noBuilding,standardBuilding,specialBuilding,noBuild
ing,standardBuilding,specialBuilding,noBuilding,standardBuilding,specialBuilding
,noBuilding,standardBuilding,specialBuilding,noBuilding,noBuilding};
    int Food[]={3,4,5,7,9,11,13,15,16,17,18,19,20,20};
    int Grain[]={1,1,1,1,1,1,1,1,1,1,1,1,1,0};
    int Cattle[]={2,2,2,2,2,2,2,2,2,2,2,2,2,0};



    int
type[]={Wooden,Wooden,Wooden,Wooden,Wooden,Iron,Iron,Iron,Iron,Steel,Steel,Steel
,Luxury,Luxury};
    int value[]={2,2,4,4,6,4,6,8,10,16,20,24,34,30};
    int buyPrice[]={14,14,14,14,14,20,20,20,20,30,30,30,0,0};
    int sellPrice[]={2,2,4,4,6,4,6,8,10,16,20,24,34,30};
    int numItem1[]={5,5,5,5,5,3,3,3,3,3,3,3,3};

    int numItem2[]={3,3,3,3,3,4,4,4,4,2,2,2,3,3};

    String
Item1[]={"Wood","Wood","Wood","Wood","Wood","energy","energy","energy","energy",
"energy","energy","energy","energy","energy"};

    String
Item2[]={"energy","energy","energy","energy","energy","Iron","Iron","Iron","Iron
","Steel","Steel","Steel","Steel","Steel"};
    int numOfSellingGoods[]={2,2,2,2,2,3,3,3,3,4,4,4,0,0};
    int numOfProvidingFoods[]={4,4,4,4,4,5,5,5,5,7,7,7,0,0};


    String nameOfBuildingCard[]={"Building Firm I",  "Building Firm
II","Construction Firm",

        "Marketplace","Bakehouse","Abattoir","Clay Mound","Shipping
Line","Church","Fishery",
        "Charcoal
Kiln","Smokehouse","Brickworks","Wharf","Colliery","Ironworks",
        "Cokery","Town Hall","Bank","Tannery", "Steel Mill",
```

```
        "Baguette Shop", "Bakery", "Brick Manufacturer", "Business Park",
"Clothing Industry", "Coal Trader",
        "Diner", "Farm", "Feedlot", "Fish Market", "Fish Restaurant", "Fishpond
and Wood",
        "Football Stadium", "Forest Hut", "Furniture Factory", "Furriery",
"Guildhouse", "Harbor Watch",
        "Haulage Firm", "Hunting Lodge", "Iron Mine and Coal Seam", "Kiln",
"Labor Exchange", "Leather Industry",
        "Luxury Yacht", "Masons' Guild", "Patisserie", "Plant Nursery",
"Schnaps Distillery", "Smelter",
        "Steakhouse","Steelworks", "Tavern", "Town Square", "Wind Farm",
"Zoo"};
    int typeofBuildingCard[]={starting,starting,starting,

standard,standard,standard,standard,standard,standard,standard,standard,standard
,

standard,standard,standard,standard,standard,standard,standard,standard,standard
,
        special,special,special,special,special,special,
        special,special,special,special,special,special,
        special,special,special,special,special,special,
        special,special,special,special,special,special,
        special,special,special,special,special,special,
        special,special,special,special,special,special};
    int owner[]={Town,Town,Town,

        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,

        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner,
        noOwner,noOwner,noOwner,noOwner,noOwner,noOwner

    };
    int valueOfBuildingCard[]={4,6,8,

            6,8,8,2,10,26,
            10,8,6,14,14,10,
            12,18,6,16,12,22,

            4,6,8,10,8,4,
            6,8,6,4,6,4,
            24,4,8,6,4,6,
            6,6,6,6,6,8,
            20,8,6,6,6,10,
            6,8,4,6,8,8};

     int Cost[]={4,6,8,

            6,8,8,2,10,0,
            10,8,6,14,14,10,
            12,18,30,40,12,22,

            4,6,8,12,8,4,
            6,8,8,4,6,4,
```

```
            24,4,8,6,8,6,
            6,6,6,6,6,8,
            20,10,6,6,6,10,
            6,8,4,6,12,8};

    int IDnum[]={2,4,6,

        1,5,9,10,18,30,
        3,7,8,14,12,16,
        22,25,28,29,20,23,

        11,13,15,17,19,21,
        24,26,27,31,32,33,
        34,35,36,37,38,39,
        40,41,42,43,44,45,
        46,47,48,49,50,51,
        52,53,54,55,56,57
       };
    int symbol[]={hammer,hammer,hammer,

        None,None,None,None,fisherman,None,
        fisherman,None,fisherman,None,None,None,
        hammer,None,None,None,None,None,


        None,None,hammer,hammer,None,None,
        fisherman,fisherman,None,fisherman,fisherman,fisherman,
        None,fisherman,None,None,hammerANDfisherman,None,
        None,hammerAND2fisherman,None,None,hammerAND2fisherman,None,
        fisherman,hammer,None,hammer,None,None,
        None,hammer,fisherman,None,None,fisherman

    };
    int typeOfBuilding[]={Craftsman,Craftsman,Industrial,


        NonBuildings,Craftsman,Craftsman,NonBuildings,Economic,Public,
        Craftsman,Craftsman,Craftsman,Industrial,Industrial,Industrial,
        Industrial,Industrial,Public,Economic,Craftsman,Industrial,


        Economic,Economic,Industrial,NonBuildings,Industrial,Economic,
        Economic,Economic,Economic,Economic,Economic,NonBuildings,
        Public,Economic,Industrial,Craftsman,Economic,Public,
        Economic,Craftsman,NonBuildings,Craftsman,Public,Industrial,
        Yacht,Craftsman,Economic,Craftsman,Craftsman,Industrial,
        Economic,Industrial,Economic,NonBuildings,NonBuildings,Public,
    };


    int entryFee[][]=
    {{0,0},{1,0},{2,0},

        {2,1},{1,0},{0,2},{1,0},{2,0},{0,0},
        {0,0},{0,0},{2,1},{1,0},{2,0},{2,0},
        {3,1},{0,1},{padLock,padLock},{padLock,padLock},{0,0},{0,2},


        {1,0},{1,0},{2,0},{padLock,padLock},{2,1},{1,0},
        {1,0},{0,1},{padLock,padLock},{1,0},{1,0},{1,0},
        {padLock,padLock},{1,0},{2,0},{1,0},{padLock,padLock},{1,0},
```

```
        {1,0},{1,0},{1,0},{1,0},{0,0},{2,0},
        {padLock,padLock},{padLock,padLock},{1,0},{1,0},{1,0},{0,2},
        {1,0},{2,1},{0,0},{0,1},{padLock,padLock},{0,1}};

    boolean plusSign[]={false,false,false,

    false,false,false,false,false,false,
    false,false,false,false,false,false,
    false,false,true,true,false,false,

    false,false,false,true,false,false,
    false,false,false,false,false,false,
    false,false,false,false,true,false,
    false,false,false,false,false,false,
    false,false,false,false,false,false,
    false,false,false,false,false,false


    };

    int buildingCost[][]={
        {0,0,0,0,0},
        {0,0,0,0,0},
        {0,0,0,0,0},

        {2,0,0,0,0},
        {0,2,0,0,0},
        {1,1,0,1,0},
        {0,0,0,0,0},
        {2,0,3,0,0},
        {5,0,3,1,0},

        {1,1,0,0,0},
        {0,1,0,0,0},
        {2,1,0,0,0},
        {2,1,0,1,0},
        {2,2,0,2,0},
        {1,3,0,0,0},

        {3,0,2,0,0},
        {0,0,2,1,0},
        {4,0,3,0,0},
        {0,0,4,1,0},
        {1,0,1,0,0},
        {0,0,4,2,0},



        {0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},
        {0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},
        {1,2,0,2,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},
        {0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},
        {0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},

{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0},{0,0,0,0,0}};
```

```
public class Disk extends Const{
    int item1,item2;
    boolean hasInterest;
    boolean Turned;
    public Disk(int item1,int item2,boolean hasInterest)
    {
        this.item1=item1;
        this.item2=item2;
        this.hasInterest=hasInterest;
        Turned=false;
    }
    public void turn() {Turned=true;}
    public String toString()
    {
        if(!Turned) return "*** ***";
        String h=new String();
        if(hasInterest) h="INTEREST";
        return name1[item1]+" "+name1[item2]+" "+h;
    }
}
```

```
    public class Disks {
    Disk disk[]=new Disk[7];
    Shuffle Sh=new Shuffle();
    public Disks()
    {
        disk[0]=new Disk(1,2,true);
        disk[1]=new Disk(4,0,false);
        disk[2]=new Disk(2,0,false);
        disk[3]=new Disk(1,3,false);
        disk[4]=new Disk(2,6,false);
        disk[5]=new Disk(2,3,false);
        disk[6]=new Disk(1,5,false);

        Sh.sh(disk);

    }
    public String toString()
    {
        String s=new String();
        s="Disks : ";
        for(int i=0;i<7;i++)
            s+="("+ disk[i]+")\t";
        return s;
    }

}
```

```
public class Dock extends Const{
    int count[]=new int[]{2,2,2,1,0,0,0};

    public Dock()
    {
        toString();

    }
    public void addItem(int i)
    {
        count[i]++;
        System.out.print("Added 1 "+name1[i]+" to dock \""+name1[i]+"
offer\".\t");
    }
    public String toString()
    {
        String s="Docks:\t";
        for(int i=0;i<7;i++)
        {
           s+=name1[i]+" offer->"+"("+count[i]+")\t";
        }
        s+="\n";
        return s;
    }
    public int getItems(int i)
    {
        int temp=count[i];
        System.out.println("\nTaking "+count[i]+" items of "+name1[i]+" from
dock \""+name1[i]+" offer\"");
        count[i]=0;
        return temp;
    }
    public boolean hasItems(int i)
    {
        if(count[i]>0) return true;
        return false;
    }
}
```

```
public class Item extends Const{
    boolean isTurned;
    String name;
    int value,meal,energy,nameOfItem;

    public Item(int nameOfItem)
    {
        if(nameOfItem<10) createFront(nameOfItem);
        else createBack(nameOfItem-10);


    }
    public void createFront(int nameOfItem)
    {
        this.nameOfItem=nameOfItem;
        this.name=name1[nameOfItem];
```

```
        this.value=value1[nameOfItem];
        this.meal=meal1[nameOfItem];
        this.energy=energy1[nameOfItem];
        isTurned=false;
    }

    public void createBack(int nameOfItem)
    {
        this.nameOfItem=nameOfItem;
        this.name=name2[nameOfItem];
        this.value=value2[nameOfItem];
        this.meal=meal2[nameOfItem];
        this.energy=energy2[nameOfItem];
        isTurned=true;
    }

    public void turnItem()
    {
        if(isTurned) return;
        S("Turned!");
        this.name=name2[this.nameOfItem];
        this.value=value2[this.nameOfItem];
        this.meal=meal2[this.nameOfItem];
        this.energy=energy2[this.nameOfItem];
        this.isTurned=true;

    }
    public String toString()
    {
        if(nameOfItem==0) return "[1F]";
        String s="["+name+" ";
        if (value>0) s+=value+"F ";
        if (meal>0) s+=meal+"M ";
        if (energy>0) s+=energy+"E ";
        return s+"]";

    }

}



import java.util.ArrayList;
import java.lang.Math;
public class Items extends Const{
    ArrayList<Item> items=new ArrayList<>();
    public Items() {}
    public void add(Item item) {items.add(item);}
    public void add(Item item,int N)
    {
        for(int i=0;i<N;i++) items.add(item);
    }
    static int count_brick=0;
    public Item get(int i) {return items.get(i);}
    public void remove(int i) {items.remove(i);}
    public int size(){return items.size();}

    public void actionForCharcoalKiln()
    {  S("action for Charcoal Kiln\n");
       S(" action: \"Upgrade any number of woods to franc\" \n");
        int N=getTotalWoods();
```

```
        int M=N/2;
        removeWoods(M);
        add(new Item(charcoal),N-M);
        S("Number of woods upgraded:"+M);
    }


    public void actionForCokery()
    {
        S("<M01> action for Cokery\n");
        S(" action: \"Upgrade any number of coal to francs receiving 1 franc for
each\"\n");
        int countFrancs=0;
        for(Item it:items)
        {
            if(it.nameOfItem==coal) it.turnItem();
            countFrancs++;
        }
        add(new Item(franc),countFrancs);
        S("Number of coal upgraded:"+countFrancs);
    }


    public void actionForColliery(boolean hammer)
    {
        S("<M02> action for Colliery\n");
        S(" action: \"Take 3 coals + 1 adittional coal for each hammer symbol
building you own\" \n");
        add(new Item(coal),3+(hammer?1:0));

        if(hammer)
        {   S("hammer combo activates...");
            S("Number of coal gained: 4");
        }
        else       S("Number of coal gained: 3");
    }

    public void actionForMarketPlace(int numOfCraftsman)
    {
        S("<M03> action for MarketPlace\n");
        S("action: \"Take 2 different standard goods(including hides and coal)
+1 additional different good per Craftsman type building you own\" \n");
        add(new Item(hides));
        add(new Item(coal));
        add(new Item(franc),numOfCraftsman);

        if(numOfCraftsman>0)
        {
            S("Craftsman combo activates...");
            S("Number of francs gained: "+numOfCraftsman);
        }
        else S("Player takes 1 hide + 1 coal");
    }



     public void actionForFishery(int numOfisherman)
    {
         S("<M02> action for Fishery\n");
         S("action: \"Take 3 fish + 1 fish per fisherman symbol buildings you
own\"\n");
```

```
          add(new Item(fish),3);
          add(new Item(fish),1*numOfisherman);

           if(numOfisherman>0) { S("Number of fish gained:"+3+numOfisherman);}
           else { S("Number of fish gained: 3");}
    }


     public void actionForClayMound(int numOfhammer)
    {
          S("<M02> action for Clay Mound\n");
          S("action: \"Take 3 clay + 1 clay per hammer symbol buildings you
own\"\n");
          add(new Item(clay),3);
          add(new Item(clay),1*numOfhammer);

           if(numOfhammer>0) { S("Number of clay gained:"+3+numOfhammer);}
           else { S("Number of clay gained: 3");}
    }


 public void actionForBakehouse()
    {
    S("action for Bakehouse\n");
        S("action: \"Upgrade any number of grain to bread by paying 1/2
energy(rounded up) per grain and receive 1/2 franc(rounded down) per
bread\"\n");
    int N=getTotalGrain();
        int M=N-1;
        removeGrain(M);
        removeEnergy((int)Math.ceil((1/2)*M));
        add(new Item(bread),M);
        add(new Item(franc),(int) Math.floor(M/2));
        S("Total Number of bread gained: "+ M);
        S("Total Number of franc gained: "+Math.floor(M/2));

    }

 public void actionForBrickworks()
    {
    S("action for Brickworks\n");
        S("action: \"Upgrade any number of clay to brick by paying 1/2
energy(rounded up) per clay and receive 1/2 franc(rounded down) per brick\"\n");

    int N=getTotalClay();
        int M=N-1;
        removeClay(M);
        removeEnergy((int)Math.ceil((1/2)*M));
        add(new Item(brick),M);
        add(new Item(franc),(int) Math.floor(M/2));
        S("Total Number of brick gained: "+ M);
        S("Total Number of franc gained: "+Math.floor(M/2));
    }


  public void actionForSmokehouse()
    {
    S("action for Smokehouse\n");
        S("action: \"At the cost of 1 energy(overall) upgrade up to 6 fish into
smoked fish 'receiving 1/2 franc(rounded down) for each'\n");
```

```
    int N=getTotalFish();
        int countlimit6=0;

        removeEnergy(1);
        for(int i=0; i<6; i++)
        {     if(countlimit6<6 && N!=0)
              {
               add(new Item(smokedfish));
               countlimit6++;
              }
        }
        removeFish(countlimit6);
        add(new Item(franc),(int) Math.floor(countlimit6/2));
    S("Total Number of smokedfish by paing 1 energy gained: "+ N);
        S("Total Number of franc gained: "+Math.floor(countlimit6/2));
    }


   public void actionForAbatoir()
 {   S("action for Abatoir\n");
     S("action: \"Upgrade any number of cattle to meat receiving 1/2
hide(rounded down) per meat\n");
    int N=getTotalCattle();
    int M=N-2;
        removeCattle(M);
    add(new Item(meat),M);
    add(new Item(hides),(int) Math.floor(M/2));
    S("Total Number of meat gained: "+ M);
        S("Total Number of hides gained: "+ (int) Math.floor(M/2));

 }



   public void actionForIronworks()
 {    S("action for Ironworks\n");
      S("action: \"Take 3 iron + 1 additional iron by paying 6 energy\"\n");
    add(new Item(iron), 3);
        S("Total Number of iron gained: 3");
    if(getTotalIron()<4 && getTotalEnergy()>6)
    {   removeEnergy(6);
           add(new Item(iron));
               S("Total Number of iron gained: 4");
    }

 }


 public void actionForSteelMill()
 {
     S("action for SteelMill\n");
     S("action: \"Upgrade any number of iron to steel by paying 5 energy for
each\"\n");
    if(getTotalIron()>=2)
    {
        removeIron(2);
    removeEnergy(5*2);
        add(new Item(steel), 2);
        S("Total Number of steel gained: 2");
    }
```

```
    else if(getTotalIron()>=1)
    {
        removeIron(1);
    removeEnergy(5);
        add(new Item(steel), 1);
        S("Total Number of steel gained: 1");
    }

      else  if(getTotalIron()>=3 && getTotalEnergy()>=15)
    {
        removeIron(3);
        removeEnergy(5*3);
                add(new Item(steel), 3);
                S("Total Number of steel gained: 3");
    }
 }


   public void actionForTannery()
 {   S("action for SteelMill\n");
     S("action: \"Upgrade up 4 hides to leather receiving 1 franc for
each\"\n");
    removeHides(4);
    add(new Item(leather), 4);
    add(new Item(franc), 4);
        S("Total Number of leather gained: 4");
        S("Total Number of franc gained: 4");

 }


 public void actionForChurch()
 {
     S("action for Church\n");
     S("action: \"Receive 5 bread and 3 fish\"\n");
    add(new Item(bread), 5);
    add(new Item(fish), 3);

 }


  public boolean actionForWharf()
{  S("State of Wharf:\n");
  boolean modernized=false;


    if (count_brick==1)
   {
     S("Wharf is already modernized: any type of Ship can be built...");
        modernized=true;
   }
    else if(count_brick==0)
    {
        S("Wharf must first be modernized in order to build a non Wooden ship");

    }
  if (getTotalBricks()>0 && count_brick==0 && modernized==false)
 {
     S("The Wharf must first got modernized");
        count_brick=1;
        modernized=true;
```

```
 }
  else{S("Player cannot Modernize the Wharf");}
 return modernized;

 }


 public void TownHall(int numOfPublic, int numOfCraftsman)
{

        S("Town Hall Combo activates...\n");
        add(new Item(franc),4*numOfPublic);
        add(new Item(franc),2*numOfCraftsman);
        S("Number of francs¨gained:"+4*numOfPublic);
        S("Town Hall Combo activated:"+2*numOfCraftsman);
 }


 public void Bank(int numOfIndustrial, int numOfEconomic)
{
        S("Bank Combo activates...\n");
        add(new Item(franc),3*numOfIndustrial);
        add(new Item(franc),2*numOfEconomic);
        S("Number of francs¨gained:"+4*numOfIndustrial);
        S("Town Hall Combo activated:"+2*numOfEconomic);
}



    public int getTotalNumberOfItem(int item)
    {
        int count=0;
        for(Item it:items)
            count+=(it.nameOfItem==item)?1:0;
        return count;
    }
    public void payMeals(int N)
    {
        int count=0;
        while(count<N)
        {
            for(int i=0;i<items.size();i++)
                 if(items.get(i).meal>0)
                 {
                     count+=items.get(i).meal;
                     items.remove(i);
                     break;
                 }
         }

    }

    public void payFrancs(int N)
    {
        removeFrancs(N);
    }

    public boolean removeFrancs(int N)
    {
        int count=0;
```

```
        if (getTotalMoney()>=N)
        {

            while(count<N)
            {
                for(int i=0;i<items.size();i++)
                    if(items.get(i).nameOfItem==franc)
                    {
                        items.remove(i);count++;
                        break;
                    }
            }
            return true;
        }
        return false;
    }

    public int getTotalWoods()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Wood ")) count++;
        return count;
    }

        public int getTotalCharcoal()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Charcoal ")) count++;
        return count;
    }

     public int getTotalGrain()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Grain ")) count++;
        return count;
    }

     public int getTotalClay()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Clay ")) count++;
        return count;
    }

     public int getTotalFish()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Fish ")) count++;
        return count;
    }

     public int getTotalSmokedFish()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Smokedfish ")) count++;
        return count;
    }

      public int getTotalBread()
    {
```

```
        int count=0;
        for(Item it:items) if(it.name.equals("Bread ")) count++;
        return count;
    }


     public int getTotalCattle()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Cattle ")) count++;
        return count;
    }

         public int getTotalMeat()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Meat ")) count++;
        return count;
    }

        public int getTotalCoal()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Coal ")) count++;
        return count;
    }

        public int getTotalCoke()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Coke ")) count++;
        return count;
    }


      public int getTotalIron()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Iron ")) count++;
        return count;
    }

      public int getTotalSteel()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Steel ")) count++;
        return count;
    }

     public int getTotalHides()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Hides ")) count++;
        return count;
    }

      public int getTotalLeather()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Leather ")) count++;
```

```
        return count;
    }


    public int getTotalBricks()
    {
        int count=0;
        for(Item it:items) if(it.name.equals("Brick ")) count++;
        return count;
    }


    public int getTotalMeals()
    {
        int count=0;
        for(Item it:items) count+=it.meal;
        return count;
    }

    public int getTotalEnergy()
    {
        int count=0;
        for(Item it:items) count+=it.energy;
        return count;
    }
    public int getTotalMoney()
    {
        int count=0;
        for(Item it:items) count+=it.value;
        return count;

    }
    public boolean tryToBuildWoodenShip()
    {
        int countWood=0,countEnergy=0;
        boolean canBuild=false;
        for(Item it:items)
        {
            if(it.nameOfItem==wood)
                countWood++;
            else
                countEnergy+=it.energy;
            if(countWood>=5 && countEnergy>=3) canBuild=true;
        }
        Sn("Total woods="+countWood+" plus total energy="+countEnergy);
        if(canBuild)
        {
            removeWoods(5);
            removeEnergy(3);
            Sn(" that's enough to build the wooden ship!");S("");
            return true;
        }
        else
            Sn(" not enough to build wooden ship!");S("");
        return false;

    }


     public boolean tryToBuildIronShip()
    {
```

```
        int countIron=0,countEnergy=0;
        boolean canBuild=false;
        for(Item it:items)
        {
            if(it.nameOfItem==iron && countIron<4)
                countIron++;
            else
                countEnergy+=it.energy;
            if(countIron>=4 && countEnergy>=3) canBuild=true;
        }
        S(countIron+" "+countEnergy);
        if(canBuild)
        {
            S("");
            removeIron(4);
            removeEnergy(3);
             Sn(" that's enough to build the iron ship!");S("");
        }
        else
            Sn(" not enough to build iron ship!");S("");
        return false;

    }


     public boolean tryToBuildSteelShip()
    {
        int countSteel=0,countEnergy=0;
        boolean canBuild=false;
        for(Item it:items)
        {
            if(it.nameOfItem==wood && countSteel<2)
                countSteel++;
            else
                countEnergy+=it.energy;
            if(countSteel>=2 && countEnergy>=3) canBuild=true;
        }
        S(countSteel+" "+countEnergy);
        if(canBuild)
        {
            S("<M10");
            removeSteel(2);
            removeEnergy(3);
            Sn(" that's enough to build the steel ship!");S("");
        }
        else
            Sn(" not enough to build steel ship!");S("");
        return false;

    }


     public boolean tryToBuildLuxuryShip()
    {
        int countSteel=0,countEnergy=0;
        boolean canBuild=false;
        for(Item it:items)
        {
            if(it.nameOfItem==steel && countSteel<3)
                countSteel++;
            else
```

```
                countEnergy+=it.energy;
            if(countSteel>=3 && countEnergy>=3) canBuild=true;
        }
        S(countSteel+" "+countEnergy);
        if(canBuild)
        {
            S("");
            removeSteel(3);
            removeEnergy(3);
     Sn(" that's enough to build the luxury ship!");S("");
        }
        else
            Sn(" not enough to build luxury ship!");S("");
        return false;

    }


public void removeWoods(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==wood)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}

public void removeCharcoal(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==charcoal)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeGrain(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==grain)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}
```

```
 public void removeBread(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==bread)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeClay(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==clay)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeCattle(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==cattle)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeMeat(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==meat)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}
```

```
public void removeCoal(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==coal)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeCoke(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==coke)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}

public void removeIron(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==iron)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeSteel(int N)
{
   int count=0;
   while(count<N)
   {
           for(int i=0;i<items.size();i++)
               if(items.get(i).nameOfItem==steel)
               {
                   items.remove(i);count++;
                   break;
               }
   }
}


public void removeHides(int N)
{
```

```
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==hides)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}


public void removeLeather(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==leather)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}

public void removeBrick(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==brick)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}


public void removeFish(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==fish)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}

public void removeSmokedFish(int N)
{
    int count=0;
    while(count<N)
    {
```

```
            for(int i=0;i<items.size();i++)
                if(items.get(i).nameOfItem==smokedfish)
                {
                    items.remove(i);count++;
                    break;
                }
    }
}


public void removeEnergy(int N)
{
    int count=0;
    while(count<N)
    {
            for(int i=0;i<items.size();i++)
                if(items.get(i).energy>0)
                {
                    count+=items.get(i).energy; items.remove(i);
                    break;
                }
    }
}


     public void groupedItems()
     {
         int count1[]={0,0,0,0,0,0,0,0,0};
         int count2[]={0,0,0,0,0,0,0,0,0};

         for(Item it:items)
         {

             if(!it.isTurned)
                 count1[it.nameOfItem]++;
             else
                 count2[it.nameOfItem]++;
         }

         S("\t\t\t**** Items ****");
         for(int i=0;i<9;i++)
         {
             if(count1[i]!=0) S("\t\t\t*"+name1[i]+":"+count1[i]);
             if(count2[i]!=0) S("\t\t\t*"+name2[i]+":"+count2[i]);
         }
     }


     public boolean hasAtLeastOneGrain()
     {
         for(Item it:items) if (it.nameOfItem==grain) return true;
         return false;

     }
     public boolean hasAtLeastTwoCattles()
     {
         int cnt=0;
         for(Item it:items) if (it.nameOfItem==cattle) cnt++;
         return (cnt>1);

     }
```

```
}
```

```
 import java.util.*;

import java.util.Random;
import java.util.ArrayList;
public class Player extends Const{
    String name;
    int position;
    boolean isHuman;
    Items items=new Items();
    Stack<Ship> ship=new Stack<>();
    int loan;
    Shuffle sh=new Shuffle();
    static int counter;
    int canBuy=nothing;
    boolean hasMaterialsForBuild=false;
    String stdbldNames[]=new String[3];
    ShipCards s=new ShipCards();
    ArrayList<String> visitedBuildingsHistory=new ArrayList<>();

    public Player(String name,boolean isHuman)
    {
        this.name=name;
        this.isHuman=isHuman;
        position=-1;
        loan=0;
        items.add(new Item(franc), 5);
        items.add(new Item(coal));
    }
    public int timesHasVisitedBeforeBuilding(String name)
    {
        int times=0;
        for(String str:visitedBuildingsHistory)
        {
            if(str.equals(name)) times++;
        }
        return times;
    }
    public void upDateBuildingHistory(String name)
    {
        visitedBuildingsHistory.add(name);
    }
    public String shipsOfPlayer()
    {
        S("<M5> "+" "+name+" "+ship.capacity());
        String s="";
        if(ship.capacity()>0)
            for(int i=0;i<ship.capacity();i++);

        return s;
    }
    public int mealsEarnedFromShips()
    {
        int sum=0;
        if(ship.capacity()>0)
```

```
            for(Ship sh:ship)
                sum+=sh.numOfProvidingFoods;
        return sum;
    }

    public void resetArray()
    {
        for(int i=0;i<3;i++)
        {
            stdbldNames[i]="";
        }

    }


    public void updateMaterialArrayOfPlayerToBuildStandardBuilding(BuildingCard
sb[])
    {   resetArray();
        int it[]={wood,clay,brick,iron,steel};
        int getFromProposal[]={1,2,3};
        String str="";
        for(int i=0;i<3;i++)
            for(int j=0;j<5;j++)
                if (sb[i].BuildingCost[j]>items.getTotalNumberOfItem(it[j]))
                {
                    getFromProposal[i]=0; break;
                }
       boolean canBuildAtLeastOne=false;
       for(int i=0;i<3;i++)
       {
            if(getFromProposal[i]>0)
            {
                str+=sb[i].Name+" | ";
                stdbldNames[i]=sb[i].Name;
                canBuildAtLeastOne=true;
            }
       }
       if(!canBuildAtLeastOne)
       {
           S("Player "+name+" can't build any standard
building!");hasMaterialsForBuild=false;
       }
       else
       {
           S("Player "+name+" can build the following standard
buildings:"+str);hasMaterialsForBuild=true;
       }
    }


public void listOfBuildingsThatPlayerCanBuy(BuildingCard starting[] ,
BuildingCard standard[])
{
    String str="";
    boolean canBuyAtLeastOneStarting=false;
    boolean canBuyAtLeastOneStandard=false;
    boolean prio=false;

    for(int i=0;i<3;i++)
if(standard[i].nameOfBuildingCard.equals("Marketplace")) prio=true;
```

```
    if(!prio)
    {
        for(int i=0; i<3; i++)
            if(starting[i].Owner==Town)
                if(starting[i].Value<=items.getTotalMoney())
                {
                    str+=starting[i].Name+" | ";
                    canBuyAtLeastOneStarting=true;
                }
        if(canBuyAtLeastOneStarting) {canBuy=start; S("Player " +name+ " can buy
the following Starting Buildings:" + str);return ;}
    }
    for(int i=0; i<3; i++)
            if(standard[i].Owner==noOwner || standard[i].Owner==Town)
                    if(standard[i].Value<=items.getTotalMoney())
                    {
                            str+=standard[i].Name+" | ";
                            canBuyAtLeastOneStandard=true;
                    }
    if(canBuyAtLeastOneStandard) {canBuy=stand;S("Player " +name+ " can buy the
following Standard Buildings:" + str);return;}

    canBuy=nothing;
}




    public boolean canPayMeals(int meals)
    {

        return
(items.getTotalMeals()+items.getTotalMoney()+mealsEarnedFromShips())>=meals;
    }

    public void payMeals(int meals)
    {
        if(!canPayMeals(meals))
        {

            getLoans((meals-items.getTotalMeals()-items.getTotalMoney()-
mealsEarnedFromShips())/4+((meals-items.getTotalMeals()-items.getTotalMoney()-
mealsEarnedFromShips())%4==0?0:1));
        }

        if (items.getTotalMeals()+mealsEarnedFromShips()>meals)
        {
            items.payMeals(meals-mealsEarnedFromShips());
        }
        else
        {

            items.payFrancs(meals-mealsEarnedFromShips()-items.getTotalMeals());
            items.payMeals(items.getTotalMeals());
        }

        S((isHuman)?"\t\t\t\t Emmanouil paying "+meals+" meals":"\t\t\t\t
Computer paying "+meals+" meals");
    }
```

```
    public void getLoans(int N)
    {
        S("----- Player:"+name+" has just got "+N+" loan"+((N==1)?"":"s")+"!");
        loan+=N;
        items.add(new Item(franc),4*N);
        counter++;
    }


public void CheckTotalLoans()
{
   S("Total Loans:" + counter);

}


    public boolean payLoan()
    {
        if(loan>0)
            if(items.removeFrancs(5))
            {
                loan--;
                return true;

            }
        return false;
    }



    public void getShipCard(Ship ship)
    {
        S("<M5> "+ship);
        this.ship.push(ship);
        items.payFrancs(ship.buyPrice);


    }

    public void getItemsFromDock(Dock dock)
    {
       while(true)
       {

            int dockNumber=sh.getRndWeight(weight);
            if (dock.hasItems(dockNumber))
            {
                items.add(new Item(dockNumber), dock.getItems(dockNumber));
                S("I'm "+name+", my arraylist has:");items.groupedItems();

S("Franc:"+items.getTotalMoney()+",Energy:"+items.getTotalEnergy()+",Meal:"+item
s.getTotalMeals()+" Loans:"+loan);
                break;
            }
       }
    }

    public void Harvest()
```

```
    {
        if(items.hasAtLeastOneGrain()) items.add(new Item(grain));
        if(items.hasAtLeastTwoCattles()) items.add(new Item(cattle));

        S("\nI'm "+name+", my arraylist after Harvest
is:");items.groupedItems();
        S("\tTotal
[Franc:"+items.getTotalMoney()+",Energy:"+items.getTotalEnergy()+",Meal:"+items.
getTotalMeals()+"]"+" Loans:"+loan);

    }

    public void interest()
    {
        S("player "+name+" is in interest check!");
        if(loan>0)
        {
            S("\t paying 1F for interest");
            if(items.getTotalMoney()==0)
                getLoans(1);
            items.payFrancs(1);
        }
        else S("\t Nothing to pay!");
    }

    public String toString()
    {
        return "Player: "+name+"
(Fr:"+items.getTotalMoney()+",En:"+items.getTotalEnergy()+",Ml:"+items.getTotalM
eals()+",Position at Disk No:"+position;
    }
```

```
public class RoundCard extends Const{
    boolean HARVEST;
    int buildingInRoundCard;
    int Food;
    int Grain;
    int Cattle;
    public RoundCard(boolean HARVEST,int buildingInRoundCard,int Food,int
Grain,int Cattle)


    {
        this.HARVEST=HARVEST;
        this.buildingInRoundCard=buildingInRoundCard;
        this.Food=Food;
        this.Grain=Grain;
        this.Cattle=Cattle;
    }



    public String toString()
    {
        String s=new String();
        s="\n+--ROUND CARD--+\n";
```

```
        if(HARVEST)
            s+="HARVEST  Grain:"+Grain+" Cattle:"+Cattle+"\n";
        else  //αλλιώς
            s+="NO HARVEST\n";
        s+="Food:"+Food+"\n";
        if (buildingInRoundCard==standardBuilding) s+="Standard Building\n";
        else if (buildingInRoundCard==specialBuilding) s+="Special Building\n";
        s+="+-------------+\n";
        return s;

    }
}


public class RoundCards extends Const{
    RoundCard roundcards[]=new RoundCard[14];
    int topLevelRoundCardIndex=0;

    public RoundCards()
    {
        createRoundCards();
    }

    public void popRoundCard()
    {
        if(topLevelRoundCardIndex<13) topLevelRoundCardIndex++;
    }

    public void createRoundCards()
    {
        for(int i=0;i<14;i++)
            roundcards[i]=new
RoundCard(i!=13,typeOfBuildingInRoundCard[i],Food[i],Grain[i],Cattle[i]);
    }

    public int getBuildingInRoundCard()
    {
        return roundcards[topLevelRoundCardIndex].buildingInRoundCard;
    }


    public String toString()
    {
        String s=roundcards[topLevelRoundCardIndex].toString()+"\n\n";
        return s;
    }



}


public class Ship extends Const{
    public int type;
    public int value;
    public int buyPrice;
    public int sellPrice;

    public int numItem1;
    public String Item1;
```

```
    public int numItem2;
    public String Item2;

    public int numOfSellingGoods;
    public int numOfProvidingFoods;

    public int Owner;



    public Ship(int type,int value,int buyPrice,int sellPrice,int
numItem1,String Item1,int numItem2,String Item2,int numOfSellingGoods,int
numOfProvidingFoods)
    {
        this.type=type;
        this.value=value;
        this.buyPrice=buyPrice;
        this.sellPrice=sellPrice;

        this.numItem1=numItem1;
        this.Item1=Item1;

        this.numItem2=numItem2;
        this.Item2=Item2;

        this.numOfSellingGoods=numOfSellingGoods;
        this.numOfProvidingFoods=numOfProvidingFoods;
        this.Owner=noOwner;

    }

    public String toString()
    {
        String s=new String();
        String str[]={"Human","Computer","","no Owner"};
                s="** "+ typeName[type]+" SHIP CARD **    ( owner:"+str[Owner]+"
)\n";
        if(value>0) s+="value:"+value+"\n";
                if(buyPrice>0) s+="buy Price:"+buyPrice+"\n";
        else s+="Cannot be bought\n";
        s+=Item1+":"+numItem1+"\n";
        s+=Item2+":"+numItem2+"\n";
        s+="Selling goods:"+numOfSellingGoods +"\n";
        s+="Providing foods:"+numOfProvidingFoods +"\n";
        return s;

    }


}



import java.util.*;


public class ShipCards extends Const {


    public Stack<Ship> allShipCards=new Stack<>();
    public Stack<Ship> WoodenShips=new Stack<>();
```

```
    public Stack<Ship> IronShips=new Stack<>();
    public Stack<Ship> SteelShips=new Stack<>();
    public Stack<Ship> LuxuryShips=new Stack<>();

    public ShipCards()    {createShipCards();}


     public void createShipCards()
    {
        for(int i=13;i>=0;i--)
            allShipCards.push(new
Ship(type[i],value[i],buyPrice[i],sellPrice[i],numItem1[i],Item1[i],numItem2[i],
Item2[i],numOfSellingGoods[i],numOfProvidingFoods[i]));

    }



     public void popShipCard()
    {

        Ship ship=allShipCards.pop();
        if(ship.type==Wooden) WoodenShips.push(ship);
        else if(ship.type==Iron) IronShips.push(ship);
        else if(ship.type==Steel) SteelShips.push(ship);
        else LuxuryShips.push(ship);
    }


    public Ship cardToPlayer(int typeOfPile)
    {
        if(typeOfPile==Wooden && !WoodenShips.empty()) return WoodenShips.pop();
        else if(typeOfPile==Iron && !IronShips.empty()) return IronShips.pop();
        else if(typeOfPile==Steel && !SteelShips.empty()) return
SteelShips.pop();
        else if(typeOfPile==Luxury && !LuxuryShips.empty()) return
LuxuryShips.pop();
        return null;
    }

    public String toString()
    {
        String s="\n\t[--------(Piles)---------]\n";
        s+=((WoodenShips.empty())?"":WoodenShips.peek())+"\n";
        s+=((IronShips.empty())?"":IronShips.peek())+"\n";
        s+=((SteelShips.empty())?"":SteelShips.peek()+"\n");
        s+=((LuxuryShips.empty())?"":LuxuryShips.peek()+"\n");
        return s;
    }
}


public class Shuffle extends Const
{
    public void sh(Object ob[])
    {
        int N=ob.length;
        int arr[]=new int[N];

        Object temp[]=new Object[N];
        for(int i=0;i<N;i++) temp[i]=ob[i];
```

```
        for(int i=0;i<N;i++)
        {
           int r=getRnd(N);
           boolean exists=false;
           for(int j=0;j<i;j++)
               if(arr[j]==r)
                   exists=true;
           if(exists)
               i--;
           else
               arr[i]=r;
        }

        for(int i=0;i<N;i++)
            ob[i]=temp[arr[i]];

    }


    public int getRnd(int N)
    {
        return (int) (Math.random() * N);

    }
    public int getRndWeight(int w[])

    {
        int len=w.length;
        int limit[]=new int[len];
        limit[0]=w[0];
        int sum=0;
        for(int i=0;i<len;i++) sum+=w[i];
        for(int i=1;i<len;i++) limit[i]=limit[i-1]+w[i];

        int rnd=(int) (Math.random()*sum);

        for(int i=0;i<len;i++) if(rnd<limit[i]) return i;

        return len-1;

    }

}



public class BuildingCard extends Const{
    int TypeOfBuildingCard;
    String Name;
    int Value;
    int Cost;
    String Action;
    int TypeOfBuilding;
    int EntryFee[]={0,0};
    boolean PlusSign;
    int BuildingCost[]={0,0,0,0,0};
    int Owner;
    int Visited;
    int IDnum;
    int symbol;
```

```
    int index;
    public BuildingCard(int typeOfBuildingCard,String name,int value,int Cost,
            String action,int typeOfBuilding,int entryFee[],
            boolean plusSign,int buildingCost[],int IDnum,int symbol,int
owner,int index)
    {
        this.TypeOfBuildingCard=typeOfBuildingCard;
        this.Name=name;
        this.Value=value;
        this.Cost=Cost;
        this.Action=action;
        this.TypeOfBuilding=typeOfBuilding;
        this.EntryFee=entryFee;
        this.PlusSign=plusSign;
        this.BuildingCost=buildingCost;
        this.Owner=owner;
        this.Visited=notVisited;
        this.IDnum=IDnum;
        this.symbol=symbol;
        this.index=index;
    }
    public BuildingCard(){}
    public void setOwner(int owner)
    {
        this.Owner=owner;
    }
    public void buyBuilding(int owner)
    {
        this.Owner=owner;
    }

    public boolean hasHammer()
    {
        return symbol==hammer || symbol==hammerANDfisherman ||
symbol==hammerAND2fisherman;
    }
     public boolean hasFisherman()
    {
        return symbol==fisherman;
    }
    public String toString()
    {

        String t="\t\t\t\t\t\t\t\t\t";
        String s[]={"STARTING","STANDARD","SPECIAL"};
    String
s2[]={"Craftsman","Economic","Industrial","Public","NonBuildings","Yacht"};
    String s3[]={"Human","Computer","Town","no Owner"};
    String sym[]={"-","H","F","H-F","H-F-F"};
        String bc="";
        bc+=(BuildingCost[0]==0)?"":"Wood("+BuildingCost[0]+")  ";
        bc+=(BuildingCost[1]==0)?"":"Clay("+BuildingCost[1]+")  ";
        bc+=(BuildingCost[2]==0)?"":"Brick("+BuildingCost[2]+")  ";
        bc+=(BuildingCost[3]==0)?"":"Iron("+BuildingCost[3]+")  ";
        bc+=(BuildingCost[4]==0)?"":"Steel("+BuildingCost[4]+")  ";
        String ef="";
        ef+=(EntryFee[0]==0)?"":"Meals("+EntryFee[0]+") ";
        ef+=(EntryFee[1]==0)?"":"Franc("+EntryFee[1]+") ";
        ef=(EntryFee[0]==-1)?"padlock":ef;
        String S=t+"+--["+s[TypeOfBuildingCard]+"],  OWNER->"+ s3[Owner]+ " ----
----index="+index+"---+\n";
```

```
        S+=t+"|       Name:["+ Name+"]\n";
        S+=t+"|  ID:"+IDnum+", Symbol:"+sym[symbol]+" , (+):"+
(PlusSign?"Yes":"No")+" , visited:"+ (Visited==notVisited?"No":"Yes")+"\n";
        S+=t+"|             Value:"+ Value+",  Cost:"+Cost+"\n";
        S+=t+"|             Action:"+ Action+"\n";
        S+=t+"|     Type of building:"+ s2[TypeOfBuilding]+"\n";
        S+=t+"|            entry Fee:"+ef+"\n";
        S+=t+"|        Building Cost:"+bc+"\n";
        return S;
    }


}
```

```
public class StartingBuildings extends Const
{
    BuildingCard buildingCard[]=new BuildingCard[3];


    public  StartingBuildings()
    {
        for(int i=0;i<3;i++)
            buildingCard[i]=new BuildingCard(starting,nameOfBuildingCard[i],

valueOfBuildingCard[i],Cost[i],"action",typeOfBuilding[i],entryFee[i],
            plusSign[i],buildingCost[i],
            IDnum[i],symbol[i],owner[i],i);

    }
    public String toString()
    {
        String s="======= Starting Buildings =======\n";
        for(int i=0;i<3;i++)
            s+=buildingCard[i]+"\n";
              s+="===== End of Starting Buildings =====\n";
        return s;
    }
}
```

```
import java.util.Stack;

public class StandardBuildings extends Const{


    BuildingCard BuildingProposalsI[]=new BuildingCard[6];
    BuildingCard BuildingProposalsII[]=new BuildingCard[6];
    BuildingCard BuildingProposalsIII[]=new BuildingCard[6];
    int topLevelIndexI=0;
    int topLevelIndexII=0;
    int topLevelIndexIII=0;
    BuildingCard buildingCard[]=new BuildingCard[18];


    public  StandardBuildings()
    {
        for(int i=3;i<21;i++)
```

```
            buildingCard[i-3]=new BuildingCard(standard,nameOfBuildingCard[i],

valueOfBuildingCard[i],Cost[i],"action",typeOfBuilding[i],entryFee[i],
            plusSign[i],buildingCost[i],
            IDnum[i],symbol[i],owner[i],i);
        shuffleCards();
    }

    public void shuffleCards()
    {
        Shuffle shuffle=new Shuffle();
        shuffle.sh(buildingCard);
        sort();
        for(int i=0;i<6;i++)
        {

            BuildingProposalsI[i]=buildingCard[i];
            BuildingProposalsII[i]=buildingCard[i+6];
            BuildingProposalsIII[i]=buildingCard[i+12];

        }
    }
    public void sort()
    {
        BuildingCard temp=new BuildingCard();
        for(int k=0;k<3;k++)
            for(int i=1+6*k;i<6+6*k;i++)
                for(int j=6+6*k-1;j>=i;j--)
                   if(buildingCard[j].IDnum<buildingCard[j-1].IDnum)
                    {
                        temp=buildingCard[j];
                        buildingCard[j]=buildingCard[j-1];
                        buildingCard[j-1]=temp;
                    }
    }
    public boolean pop()
    {
        int I,II,III,Min;
        I=(topLevelIndexI<6)?BuildingProposalsI[topLevelIndexI].IDnum:empty;
        II=(topLevelIndexII<6)?BuildingProposalsII[topLevelIndexII].IDnum:empty;

III=(topLevelIndexIII<6)?BuildingProposalsIII[topLevelIndexIII].IDnum:empty;
        Min=min(min(I,II),III);
        if(Min!=empty)
        {
            if(Min==I) pop(1);
            else if(Min==II) pop(2);
            else pop(3);
            return true;
        }
        return false;
    }
    public boolean pop(int numberOfPile)
    {
        if(numberOfPile==1)
            if(topLevelIndexI<6)
            {
                BuildingProposalsI[topLevelIndexI].setOwner(Town);
                topLevelIndexI++;return true;
            }
        if(numberOfPile==2)
```

```
            if(topLevelIndexII<6)
            {
                BuildingProposalsII[topLevelIndexII].setOwner(Town);
                topLevelIndexII++;return true;
            }
        if(numberOfPile==3)
            if(topLevelIndexIII<6)
            {
                BuildingProposalsIII[topLevelIndexIII].setOwner(Town);
                topLevelIndexIII++;return true;
            }
        return false;
    }
    public int min(int a,int b) {return (a<b)?a:b;}


  public int getTotalCraftsmanOfPlayer(int player)
  {
    int count=0;
    for(int i=0;i<6;i++)
    {
         if(i<topLevelIndexI )
             count+=(BuildingProposalsI[i].TypeOfBuilding==Craftsman &&
BuildingProposalsI[i].Owner==player)?1:0;
         if(i<topLevelIndexII)
             count+=(BuildingProposalsII[i].TypeOfBuilding==Craftsman &&
BuildingProposalsII[i].Owner==player)?1:0;
         if(i<topLevelIndexIII)
             count+=(BuildingProposalsIII[i].TypeOfBuilding==Craftsman &&
BuildingProposalsIII[i].Owner==player)?1:0;
    }
    return count;


  }

  public boolean hasHammer(int player)
   {
       int count=0;
       for(int i=0;i<6;i++)
       {
            if(i<topLevelIndexI )
                if (BuildingProposalsI[i].hasHammer() &&
BuildingProposalsI[i].Owner==player) return true;
            if(i<topLevelIndexII)
                if (BuildingProposalsII[i].hasHammer() &&
BuildingProposalsII[i].Owner==player) return true;
            if(i<topLevelIndexIII)
                if (BuildingProposalsIII[i].hasHammer() &&
BuildingProposalsIII[i].Owner==player) return true;
       }
       return false;
   }


  public String toString()
    {

        String s="======= Building Proposals =======\n";
        s+="Pile I\n";
        s+=BuildingProposalsI[topLevelIndexI];
```

```
        s+="\nPile II\n";
        s+=BuildingProposalsII[topLevelIndexII];
        s+="\nPile III\n";
        s+=BuildingProposalsIII[topLevelIndexIII]+"\n";
        s+="======= End of Building Proposals =======\n";

        for(int i=0;i<6;i++)
        {

            if(i<topLevelIndexI )
                s+=BuildingProposalsI[i]+"\n";
            if(i<topLevelIndexII)
                s+=BuildingProposalsII[i]+"\n";
            if(i<topLevelIndexIII)
                s+=BuildingProposalsIII[i]+"\n";
        }
        return s;

    }
}



public class SpecialBuildings extends Const{
    public BuildingCard buildingCard[]=new BuildingCard[6];
    public int topLevelIndex=0;


    public SpecialBuildings()
    {
        BuildingCard temp[]=new BuildingCard[36];

        for(int i=21;i<57;i++)
        {
            temp[i-21]=new BuildingCard(special,nameOfBuildingCard[i],
            valueOfBuildingCard[i],Cost[i],
            "action",typeOfBuilding[i],entryFee[i],
            plusSign[i],buildingCost[i],
            IDnum[i],symbol[i],owner[i],i);
        }
        Shuffle shuffle=new Shuffle();

        shuffle.sh(temp);
        for(int i=0;i<6;i++)
            buildingCard[i]=temp[i];
    }

    public boolean pop()
    {
        if (topLevelIndex<6)
        {
            buildingCard[topLevelIndex].Owner=Town;
            topLevelIndex++;
            return true;
        }
        return false;
    }



    public String toString()
```

```
    {
        String s="";
        for(int i=0;i<topLevelIndex;i++)
            s+=buildingCard[i]+"\n";
        return s;
    }
}
```

```
import java.util.ArrayList;
import java.util.Collections;
public class BuildingCardCreator extends Const{

    BuildingCard buildingCard[][]=new BuildingCard[5][36];


    SpecialBuildings specialBuildings=new SpecialBuildings();
    StartingBuildings startingBuildings=new StartingBuildings();
    StandardBuildings standardBuildings=new StandardBuildings();


    public BuildingCardCreator()
    {
        copyToOne();
    }


    public void pop(int type)
    {
        if(type==standard) standardBuildings.pop();
        else specialBuildings.pop();
        copyToOne();

    }



    public void visit(int i,int j,int player)
    {
        buildingCard[i][j].Visited=player;
        copyToMany();
    }
    public void visit(int IDnum,int player)
    {
        for(int i=1;i<4;i++)
            for(int j=0;j<6;j++)
                if (buildingCard[i][j].IDnum==IDnum)
                {
                    visit(i,j,player);return;
                }

    }
    public void visit(String name,int player)
    {
        for(int i=1;i<4;i++)
            for(int j=0;j<6;j++)
```

```
                if (buildingCard[i][j].Name.equals(name))
                {
                    visit(i,j,player);return;
                }


    }


   public BuildingCard[] getStartingBuilding()
  {
   BuildingCard bc[]=new BuildingCard[3];
   bc[0]= buildingCard[0][0];
   bc[1]= buildingCard[0][1];
   bc[2]= buildingCard[0][2];
   return bc;
  }



    public BuildingCard[] getStandardBuildingFromTopOfPiles()
    {
        BuildingCard bc[]=new BuildingCard[3];
        bc[0]=buildingCard[1][standardBuildings.topLevelIndexI];
        bc[1]=buildingCard[2][standardBuildings.topLevelIndexII];
        bc[2]=buildingCard[3][standardBuildings.topLevelIndexIII];
        return bc;
    }

    public ArrayList<BuildingCard> getStandardBuildsOfTown()
    {
        ArrayList<BuildingCard> bc=new ArrayList<>();
        boolean CharcoalExists=false;int position=0;
        boolean MarketplaceExists=false;
        for(int i=1;i<4;i++)
            for(int j=0;j<6;j++)
                if((buildingCard[i][j].Owner==Town  ||
buildingCard[i][j].Owner==Human || buildingCard[i][j].Owner==Computer)&&
buildingCard[i][j].Visited==notVisited)
                {
                    bc.add(buildingCard[i][j]);
                    if(buildingCard[i][j].nameOfBuildingCard.equals("Charcoal
Kiln"))
                    {
                        CharcoalExists =true;
                        position=j;
                    }

                }
        if(CharcoalExists && bc.size()>1)
            Collections.swap(bc, 0, position);
        return bc;
    }



      public int getNumOfCraftsmanOfPlayer()
    {
        int count=0;
        int count2=0;
```

```
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].TypeOfBuilding==Craftsman)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].TypeOfBuilding==Craftsman)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
         if(var) return count;
         else       return count2;

    }



     public int getNumOfHammerOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].symbol==hammer)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].symbol==hammer)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
         if(var) return count;
         else       return count2;

    }

      public boolean getIfHasHammerOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=false;
```

```
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].symbol==hammer)
                    {
                            count++;
                            var=true;

                    }
                            else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].symbol==hammer)
                    {
                                count2++;
                                var=true;

                    }


            }
        }
        return var;
    }

     public int getNumOfFishermanOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].symbol==fisherman)
                    {
                            count++;
                            var=true;
                    }
                            else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].symbol==fisherman)
                    {
                                count2++;
                                var=false;
                    }
            }
        }
         if(var) return count;
         else       return count2;

    }


    public int getNumOfPublicOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
```

```
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].TypeOfBuilding==Public)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].TypeOfBuilding==Public)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
         if(var) return count;
         else       return count2;

    }


    public int getNumOfEconomicOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].TypeOfBuilding==Economic)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].TypeOfBuilding==Economic)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
         if(var) return count;
         else       return count2;

    }


     public int getNumOfIndustrialOfPlayer()
    {
        int count=0;
        int count2=0;
        boolean var=true;
        for(int i=0;i<5;i++)
```

```
        {
            for(int j=0;j<36;j++)
            {
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Owner==Human &&
buildingCard[i][j].TypeOfBuilding==Industrial)
                    {
                           count++;
                           var=true;
                    }
                          else if(buildingCard[i][j].Owner==Computer &&
buildingCard[i][j].TypeOfBuilding==Industrial)
                    {
                              count2++;
                              var=false;
                    }
            }
        }
         if(var) return count;
         else       return count2;

    }



    public void releaseBuilding(int currentPlayer)
    {
        releaseStartingBuilding(currentPlayer);
        releaseStandardBuilding(currentPlayer);


    }
    public void releaseStartingBuilding(int currentPlayer)
    {
        for(int i=0;i<3;i++)
        {
            if(buildingCard[0][i].Visited==currentPlayer)
            {
                S("Building Released:"+buildingCard[0][i].Name+" player was
"+currentPlayer);
                buildingCard[0][i].Visited=notVisited;
            }
            }
        copyToMany();
    }

   public void releaseStandardBuilding(int currentPlayer)
    {
        for(int i=1;i<4;i++)
        {
            for(int j=0;j<6;j++)
            if(buildingCard[i][j].Visited==currentPlayer)
            {
                S("Building Released:"+buildingCard[i][j].Name+" player was
"+currentPlayer);
                buildingCard[i][j].Visited=notVisited;
            }
        }
        copyToMany();
    }
    public void setOwnerToCardByName(String name,int owner)
```

```
    {
        for(int i=1;i<4;i++)
            for(int j=0;j<36;j++)
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].Name.equals(name))
                    {
                        buildingCard[i][j].Owner=owner;
                        if(i==1) standardBuildings.topLevelIndexI++;
                        else if(i==2) standardBuildings.topLevelIndexII++;
                        else if(i==3) standardBuildings.topLevelIndexIII++;
                        copyToMany();
                    }
    }


    public void setOwnerToCardByIndex(int index,int owner)
    {
        for(int i=1;i<4;i++)
            for(int j=0;j<36;j++)
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].index==index)
                    {
                        buildingCard[i][j].Owner=owner;
                        if(i==1) standardBuildings.topLevelIndexI++;
                        else if(i==2) standardBuildings.topLevelIndexII++;
                        else if(i==3) standardBuildings.topLevelIndexIII++;
                        copyToMany();
                    }
    }
    public BuildingCard getBuildingCardByIDnum(int IDnum)
    {
        for(int i=0;i<5;i++)
            for(int j=0;j<36;j++)
                if( buildingCard[i][j]!=null)
                    if(buildingCard[i][j].IDnum==IDnum)
                        return buildingCard[i][j];
        return null;
    }
    public void copyToOne()
    {
        System.arraycopy(startingBuildings.buildingCard, 0, buildingCard[0], 0,
3);
        System.arraycopy(standardBuildings.BuildingProposalsI, 0,
buildingCard[1], 0, 6);
        System.arraycopy(standardBuildings.BuildingProposalsII, 0,
buildingCard[2], 0, 6);
        System.arraycopy(standardBuildings.BuildingProposalsIII, 0,
buildingCard[3], 0, 6);
        System.arraycopy(specialBuildings.buildingCard, 0, buildingCard[4], 0,
6);
    }
    public void copyToMany()
    {

System.arraycopy(buildingCard[0],0,startingBuildings.buildingCard,0,3);

System.arraycopy(buildingCard[1],0,standardBuildings.BuildingProposalsI,0,6);

System.arraycopy(buildingCard[2],0,standardBuildings.BuildingProposalsII,0,6);

System.arraycopy(buildingCard[3],0,standardBuildings.BuildingProposalsIII,0,6);
```

```
            System.arraycopy(buildingCard[4],0,specialBuildings.buildingCard,0,6);
        }



        public String printCard(String name)
        {
            for(int i=1;i<4;i++)
                for(int j=0;j<6;j++)
                    if(buildingCard[i][j].Name.equals(name))
                        return buildingCard[i][j].toString();
            return null;
        }
        public String toString()
        {
            String s="\t\t\t\t\t(((((((( Cards on board )))))))\n";
            s+=startingBuildings+"\n";
            s+=standardBuildings+"\n";
            s+=specialBuildings+"\n";
            return s;

        }

}



public class cardControl
{
    public ShipCards shipCards;
    public RoundCards roundCards;


    public cardControl()
    {
        shipCards=new ShipCards();
        roundCards=new RoundCards();
    }


    public void turnCard()
    {
        roundCards.popRoundCard();
        shipCards.popShipCard();
    }

    public String toString()
    {
        String s="[---- Round & Ship Cards ----]\n";
        s+=roundCards.toString();
        s+=shipCards.toString();
        return s;
    }
}




import java.util.Random;
import java.util.ArrayList;
```

```
public class playGame extends Const
{
    Dock dock=new Dock();
    Disks disks=new Disks();
    Player []player=new Player[2];
    cardControl cc=new cardControl();
    Items it= new Items();
    BuildingCardCreator buildingCards=new BuildingCardCreator();
    int currentPlayer;
   Shuffle s=new Shuffle();

public playGame()
{
    createPlayers();
    for(int round=0;round<14;round++)
    {
        S("\n\n======== ROUND No:"+(round+1)+"============\n");
        printRoundShipAndBuildingCards();
        for(int turn=0;turn<7;turn++)
        {
            buildingCards.releaseBuilding(currentPlayer);
            printTurnAndCurrentPlayerDetails(turn,round);
            setPlayerToDiskAndTurnDisk(turn);
            printDisks();
            addItemsToDock(turn);
            checkIfcurrentPlayerHasToPayLoan();
            interest(disks.disk[turn].hasInterest);
            printIfPlayersHasMaterialsToBuildStandardBuildings();
            printIfPlayerHasFrancsToBuyBuilding();
            playerDecidesAction();
            currentPlayer=alterPlayer(currentPlayer);

        }
        endOfRoundActions(round);
        printTotalLoans();
    }

}


public void playerDecidesAction()
{
    boolean actionDone=false;
    int p;
    while(!actionDone)
    {
        if(preCheckOfWharf()) break;
        int w[]={60,5,35};
        p=new Shuffle().getRnd(100);
        if(p<w[0])
        {

                S("Player "+cp().name+" is trying to visit building");
                actionDone=tryToVisit();

        }
        else if(p<w[0]+w[1])
        {
                S("Player "+cp().name+" is trying to build building");
                actionDone=tryToBuild();
```

```
        }
        else
        {

                S("Player "+cp().name+" is getting items from dock");
                  cp().getItemsFromDock(dock);
                  actionDone=true;

        }
    }


}
public boolean preCheckOfWharf()
{
    if(new Shuffle().getRnd(100)<20)
    {

cp().updateMaterialArrayOfPlayerToBuildStandardBuilding(buildingCards.getStandar
dBuildingFromTopOfPiles());
    for(int i=0;i<3;i++)
        if(cp().stdbldNames[i].equals("Wharf"))
        {
            buildingCards.setOwnerToCardByName("Wharf",currentPlayer);
            S("<M300> Player "+cp().name+ " has build the Wharf!");
            return true;
        }
    }
    return canVisitWharf();
}
public boolean tryToVisit()
{
    return canVisitStandardBuilding(buildingCards.getStandardBuildsOfTown());
}

public boolean tryToBuild()
{

cp().updateMaterialArrayOfPlayerToBuildStandardBuilding(buildingCards.getStandar
dBuildingFromTopOfPiles());
    return checkForPriorityAndBuildStandardBuilding();
}

public boolean canVisitCharcoalKiln()
{

    if(cp().items.getTotalWoods()<2) return false;
    S("M100 Player "+cp().name+" visits Charcoal Kiln");
    buildingCards.visit(7, currentPlayer);
    return true;

}


  public boolean canVisitMarketPlace()
{
    if(cp().items.getTotalMeals()<2 && cp().items.getTotalMoney()<1) return
false;
       S("M<101> player "+cp().name+" visits MarketPlace");
       buildingCards.visit(1, currentPlayer);
       return true;
```

```
}


  public boolean canVisitClayMound()
{
    if(cp().items.getTotalMeals()<1) return false;
       S("M<101> player "+cp().name+" visits Clay Mound");
       buildingCards.visit(10, currentPlayer);
       return true;
}

  public boolean canVisitBakeHouse()
{
   if(cp().items.getTotalMeals()<1 && cp().items.getTotalGrain()<2 &&
cp().items.getTotalEnergy()<1)return false;

       S("M<101> player "+cp().name+" visits Bakehouse");
       buildingCards.visit(5, currentPlayer);
       return true;
}


  public boolean canVisitSmokeHouse()
{
   if(cp().items.getTotalMeals()<2 && cp().items.getTotalMoney()<1 &&
cp().items.getTotalFish()<1 && cp().items.getTotalEnergy()<1) return false;

       S("M<101> player "+cp().name+" visits Smokehouse");
       buildingCards.visit(8, currentPlayer);
       return true;
}

  public boolean canVisitAbatoir()
{
    if(cp().items.getTotalMoney()<2 && cp().items.getTotalCattle()<1) return
false;

       S("M<101> player "+cp().name+" visits Abatoir");
       buildingCards.visit(9, currentPlayer);
       return true;
}

  public boolean canVisitIronworks()
{
   if(cp().items.getTotalMeals()<3 && cp().items.getTotalMoney()<1 ) return
false;

       S("M<101> player "+cp().name+" visits Ironworks");
       buildingCards.visit(22, currentPlayer);
       return true;
}


  public boolean canVisitSteelMill()
{
    if(cp().items.getTotalMoney()<2 && cp().items.getTotalIron()<1 &&
cp().items.getTotalEnergy()<5) return false;

       S("M<101> player "+cp().name+" visits SteelMill");
       buildingCards.visit(23, currentPlayer);
       return true;
```

```
}


 public boolean canVisitTannery()
{
    if(cp().items.getTotalHides()<1) return false;

       S("M<101> player "+cp().name+" visits Tannery");
       buildingCards.visit(20, currentPlayer);
       return true;
}



 public boolean canVisitChurch()
{
   if(cp().items.getTotalBread()<5 && cp().items.getTotalFish()<2 ) return
false;

       S("M<101> player "+cp().name+" visits Church");
       buildingCards.visit(30, currentPlayer);
       return true;
}

 public boolean canVisitWharf()
{
    S("check for visit Wharf!");
    if(cp().items.getTotalMeals()<2) {S("...can't visit Wharf");return false;}
            if(cp().items.getTotalWoods()<5 || cp().items.getTotalEnergy()<3)
return false;
            S("M<101> player "+cp().name+" visits Wharf and paying 2 meals");
            cp().items.payMeals(2);
            buildingCards.visit(12, currentPlayer);
            cp().items.actionForWharf();
            buildShip();
            return true;
}


 public boolean canVisitTownHall()
{

    if(buildingCards.getBuildingCardByIDnum(28).Owner==Town &&
buildingCards.getBuildingCardByIDnum(28).Visited==notVisited)
    {
       S("M<101> player "+cp().name+" visits Town Hall");
       buildingCards.visit(28, currentPlayer);
       S("before");cp().items.groupedItems();
       cp().items.TownHall(buildingCards.getNumOfPublicOfPlayer(),
buildingCards.getNumOfCraftsmanOfPlayer());
       S("after");cp().items.groupedItems();
       return true;
    }

    return false;
}

 public boolean canVisitBank()
{
```

```
    if(buildingCards.getBuildingCardByIDnum(29).Owner==Town &&
buildingCards.getBuildingCardByIDnum(29).Visited==notVisited)
    {
       S("M<101> player "+cp().name+" visits Bank");
       buildingCards.visit(29, currentPlayer);
       S("before");cp().items.groupedItems();
       cp().items.Bank(buildingCards.getNumOfIndustrialOfPlayer(),
buildingCards.getNumOfEconomicOfPlayer());
       S("after");cp().items.groupedItems();
       return true;
    }

    return false;
}


 public boolean canVisitBrickWorks()
{
   if(cp().items.getTotalMeals()<1 && cp().items.getTotalClay()<1 &&
cp().items.getTotalEnergy()<1) return false;

       S("M<101> player "+cp().name+" visits Brickworks");
       buildingCards.visit(14, currentPlayer);
       return true;
}

 public boolean canVisitFishery()
{

       S("M<101> player "+cp().name+" visits Fishery");
       buildingCards.visit(3, currentPlayer);
       return true;
}

 public boolean canVisitCokery()
{
   if(cp().items.getTotalMoney()<1 && cp().items.getTotalCoal()<1) return false;

       S("M<101> player "+cp().name+" visits Cokery");
       buildingCards.visit(25, currentPlayer);
       return true;
}

 public boolean canVisitColliery()
{
   if(cp().items.getTotalMoney()<2) return false;

       S("M<101> player "+cp().name+" visits Colliery");
       buildingCards.visit(16, currentPlayer);
       return true;
}

public Player cp() {return player[currentPlayer];}


public void finallyDecidesToBuildStandardBuilding()
{

    int ind=selectAmongStartingBuildings();
    S("Player:"+cp().name+" decides to visit the starting building
"+buildingCards.buildingCard[0][ind].Name);
```

```
    buildingCards.visit(0, ind, currentPlayer);

    cp().payMeals(buildingCards.buildingCard[0][ind].EntryFee[0]);

    checkForPriorityAndBuildStandardBuilding();
}



public boolean checkForPriorityAndBuildStandardBuilding()
{
    for(String str:prioBuild)
        for(int i=0;i<3;i++)
            if(str.equals(cp().stdbldNames[i]))
            {
                 buildingCards.setOwnerToCardByName(str,currentPlayer);
                 S("<M200> Player "+cp().name+ " has build the "+str);
                 return true;
            }
    return false;
}



public int selectAmongStartingBuildings()
{
    if(buildingCards.buildingCard[0][0].Visited==notVisited)
        if(buildingCards.buildingCard[0][0].Owner==Town)
            return 0;
    if(buildingCards.buildingCard[0][1].Visited==notVisited)
        if(buildingCards.buildingCard[0][1].Owner==Town)
            return 1;
    return 2;
}


public void checkIfcurrentPlayerHasToPayLoan()
{
    S(cp().name+ " has loans:"+ cp().loan+", "+player[otherPlayer()].name + "
has:"+player[otherPlayer()].loan+" loans");
    Shuffle s= new Shuffle();
    if(cp().loan>player[otherPlayer()].loan &&
s.getRnd(100)<weightOfPayLoan[currentPlayer])
        if(cp().payLoan())
          S("Player "+cp().name+" paid 1 loan"+"\nList of player after paying
is:"+cp());

}

public void printTurnAndCurrentPlayerDetails(int turn,int round)
{
    S("\n\t ---- Turn No:"+(turn+1)+" of Round:"+round+" ----(playing
"+cp().name+")-----------");
}


 public void setPlayerToDiskAndTurnDisk(int turn)
{
    cp().position=turn;
    disks.disk[turn].turn();
}
```

```
public void printRoundShipAndBuildingCards()
{
    S(cc);
    S(buildingCards);

}

public void printIfPlayersHasMaterialsToBuildStandardBuildings()
{
    S("~~~~~~~~~~~~~~~~~~~");

cp().updateMaterialArrayOfPlayerToBuildStandardBuilding(buildingCards.getStandar
dBuildingFromTopOfPiles());
    S("~~~~~~~~~~~~~~~~~~~");
}


public void printIfPlayerHasFrancsToBuyBuilding()
{
    S("~~~~~~~~~~~~~~~~~~~");

cp().listOfBuildingsThatPlayerCanBuy(buildingCards.getStartingBuilding(),buildin
gCards.getStandardBuildingFromTopOfPiles());
    S("~~~~~~~~~~~~~~~~~~~");
}


public void endOfRoundActions(int round)
{
    harvest(round);
    payMeals(round);
    S(cc.roundCards);
    buildingCards.pop(cc.roundCards.getBuildingInRoundCard());
    cc.turnCard();
    S("$$$$$$$$$$$$$$$$$$$$ END OF ROUND "+(round+1)+"
$$$$$$$$$$$$$$$$$$$$$$$$$$");


}


public int randomStartPlayer()
{
    return new Random(System.currentTimeMillis()).nextInt(2);
}


public void buyStartingBuilding(BuildingCard starting[])
    {
    for(BuildingCard st:starting)
        if(st.Owner==Town && st.Visited==notVisited)
            if(cp().items.getTotalMoney()>=st.Value)
            {
                st.setOwner(currentPlayer);
                S("buying of starting building: "+cp().name+" has bought the "
+st.Name+"!" );
                S("Items of "+cp().name+" before buy action:"+cp());
                cp().items.payFrancs(st.Cost);

                return;
            }
```

```
    }


public void buyStandardBuilding(BuildingCard standard[])
{
    for(BuildingCard st:standard)
        if(st.Owner == noOwner && st.Visited==notVisited)
            if(st.Value<cp().items.getTotalMoney())
            {
                String str="Items of "+cp().name+" before action:"+cp();
                if(hasPriorityTheStandardBuildingCardToBeVisited(st))
                    if(cp().items.removeFrancs(st.Cost))                    {
                            st.setOwner(currentPlayer);
                            S("<M1> Player "+cp().name+" has bought the
"+st.Name+"!");
                            S(str);
                            S("Items of "+cp().name+" after action:"+cp());
                            return;
                    }
            }
}


public void callActionForStandardBuilding(BuildingCard bc)
{
    S("before");cp().items.groupedItems();
    if(bc.Name.equals("Marketplace"))
cp().items.actionForMarketPlace(buildingCards.standardBuildings.getTotalCraftsma
nOfPlayer(currentPlayer));
    else if(bc.Name.equals("Charcoal Kiln")) cp().items.actionForCharcoalKiln();
    else if(bc.Name.equals("Wharf")) cp().items.actionForWharf();
    else if(bc.Name.equals("Clay Mound"))
cp().items.actionForClayMound(buildingCards.getNumOfHammerOfPlayer());
    else if(bc.Name.equals("Brickworks")) cp().items.actionForBrickworks();
    else if(bc.Name.equals("Colliery"))
cp().items.actionForColliery(buildingCards.getIfHasHammerOfPlayer());
    else if(bc.Name.equals("Cokery")) cp().items.actionForCokery();
    else if(bc.Name.equals("Bakehouse")) cp().items.actionForBakehouse();
    else if(bc.Name.equals("Abatoir")) cp().items.actionForAbatoir();
    else if(bc.Name.equals("Tannery")) cp().items.actionForTannery();
    else if(bc.Name.equals("Smokehouse")) cp().items.actionForSmokehouse();
    else if(bc.Name.equals("Fishery"))
cp().items.actionForFishery(buildingCards.getNumOfFishermanOfPlayer());
    else if(bc.Name.equals("Church")) cp().items.actionForChurch();
    else if(bc.Name.equals("SteelMill")) cp().items.actionForSteelMill();
    else if(bc.Name.equals("Ironworks")) cp().items.actionForIronworks();

    S("after");cp().items.groupedItems();

}

public boolean canVisitStandardBuilding(ArrayList<BuildingCard> standard)
{
    for(BuildingCard st:standard)
    {
        if(hasPriorityTheStandardBuildingCardToBeVisited(st))
        {
            S("EntryFee Of Building: "+st.EntryFee[0]+" "+st.EntryFee[1]+"
"+st.Name);
            if(st.EntryFee[0]==0 && st.EntryFee[1]==0 &&
timesOfPlayerIsLessThan(st.Name,1))
```

```
            {
                visitsPlayerTheBuilding(st.Name);
                callActionForStandardBuilding(st);return true;
            }

            if(st.EntryFee[0]>0 && st.EntryFee[0]<=cp().items.getTotalMeals() &&
timesOfPlayerIsLessThan(st.Name,1))
            {
                visitsPlayerTheBuilding(st.Name);
                cp().items.payMeals(st.EntryFee[0]);
                callActionForStandardBuilding(st);return true;

            }
            if(st.EntryFee[1]>0 && st.EntryFee[1]<=cp().items.getTotalMoney() &&
timesOfPlayerIsLessThan(st.Name,1))
            {
                visitsPlayerTheBuilding(st.Name);
                cp().items.payFrancs(st.EntryFee[1]);
                callActionForStandardBuilding(st);return true;
            }
        }


    }
    return false;
}



public boolean hasVisitedStandardBuilding2(ArrayList<BuildingCard> standard)
{
    for(BuildingCard st:standard)
    {
        if(hasPriorityTheStandardBuildingCardToBeVisited(st))
        {
            if(st.Name.equals("Charcoal Kiln"))
                if(timesOfPlayerIsLessThan(st.Name,1) && canVisitCharcoalKiln())
                {
                    visitsPlayerTheBuilding(st.Name);
                    return true;
                }
            else
            {
                if(st.EntryFee[0]>0 &&
st.EntryFee[0]<=cp().items.getTotalMeals() &&
timesOfPlayerIsLessThan(st.Name,3))
                {
                    S("Player: "+cp().name+" "+" visits the "+st.Name);
                    visitsPlayerTheBuilding(st.Name);
                    cp().items.payMeals(st.EntryFee[0]);
                    callActionForStandardBuilding(st);return true;

                }

                if(st.EntryFee[1]>0 &&
st.EntryFee[1]<=cp().items.getTotalMoney() &&
timesOfPlayerIsLessThan(st.Name,3))
                {
                    visitsPlayerTheBuilding(st.Name);
                    cp().items.payFrancs(st.EntryFee[1]);
                    callActionForStandardBuilding(st);return true;
```

```
                }
            }

        }
    }
    return false;
}

public boolean timesOfPlayerIsLessThan(String name,int times,int possibility)
{

    return ((cp().timesHasVisitedBeforeBuilding(name)<times) && new
Shuffle().getRnd(100)<possibility);

}

public boolean timesOfPlayerIsLessThan(String name,int times)
{
    return cp().timesHasVisitedBeforeBuilding(name)<times;

}
public void visitsPlayerTheBuilding(String name)
{
    S("Player "+cp().name+" visits the standard building "+name);
    cp().upDateBuildingHistory(name);
    buildingCards.visit(name, currentPlayer);

}

public void buildShip()
{
    S("trying to build Ship...");
   if(cp().items.actionForWharf())
   {
        if(cp().items.tryToBuildLuxuryShip())
        {
        S(cp().name+" managed to build Luxury ship!");
        cp().s.cardToPlayer(Luxury);
        S("List Of Player "+cp().name+" Ships"+cp().ship);
        }
        if(cp().items.tryToBuildSteelShip())
        {
        S(cp().name+" managed to build Steel ship!");
        cp().s.cardToPlayer(Steel);
        S("List Of Player "+cp().name+" Ships"+cp().ship);
        }
        if(cp().items.tryToBuildIronShip())
        {
            S(cp().name+" managed to build Iron ship!");
            cp().s.cardToPlayer(Iron);
            S("List Of Player "+cp().name+" Ships"+cp().ship);
        }
        if(cp().items.tryToBuildWoodenShip())
        {
            S(cp().name+" managed to build Wooden ship!");
            cp().s.cardToPlayer(Wooden);
            S("List Of Player "+cp().name+" Ships"+cp().ship);
        }
   }
   else
    if(cp().items.tryToBuildWoodenShip())
```

```
    {
        S(cp().name+" managed to build Wooden ship!");
         cp().s.cardToPlayer(Wooden);
         S("List Of Player "+cp().name+" Ships"+cp().ship);
    }

}


public void buyShip()
{

    S("<M7");
    if(!cc.shipCards.WoodenShips.empty())
        if(cc.shipCards.WoodenShips.peek().buyPrice<=cp().items.getTotalMoney()
&& cc.shipCards.WoodenShips.peek().buyPrice>0)
        {cp().getShipCard(cc.shipCards.cardToPlayer(Wooden));return;}

    if(!cc.shipCards.IronShips.empty())
     if(cc.shipCards.IronShips.peek().buyPrice<=cp().items.getTotalMoney() &&
cc.shipCards.IronShips.peek().buyPrice>0)
        {cp().getShipCard(cc.shipCards.cardToPlayer(Iron));return;}

    if(!cc.shipCards.SteelShips.empty())
      if(cc.shipCards.SteelShips.peek().buyPrice<=cp().items.getTotalMoney() &&
cc.shipCards.SteelShips.peek().buyPrice>0)
      {cp().getShipCard(cc.shipCards.cardToPlayer(Steel));return;}

    if(!cc.shipCards.LuxuryShips.empty())
     if(cc.shipCards.LuxuryShips.peek().buyPrice<=cp().items.getTotalMoney() &&
cc.shipCards.LuxuryShips.peek().buyPrice>0)
        cp().getShipCard(cc.shipCards.cardToPlayer(Luxury));



}



public void sellBuilding(BuildingCard bc)
{
    bc.setOwner(Town);
    cp().items.add(new Item(franc),bc.Value/2);
    S("Player "+cp().name+" has just sold "+bc.Name+" to Town for "+bc.Value/2+"
francs");

}


public boolean hasPriorityTheStandardBuildingCardToBeVisited(BuildingCard bc)
{
    for(String st:prioVisit)
        if(st.equals(bc.Name) || bc.symbol!=None) return true;
    return false;
}


public boolean hasPriorityTheStandardBuildingCardToBeBuilt(BuildingCard bc)
{
    for(String st:prioBuild)
        if(st.equals(bc.Name) || bc.symbol!=None) return true;
```

```
    return false;
}


public boolean hasPriorityTheStandardBuildingCardToBeBought(BuildingCard bc)
{
    for(String st:prioBuy)
        if(st.equals(bc.Name) || bc.symbol!=None) return true;
    return false;
}

public void bonusAction()
{
    //buildShip();
    //buyShip();
    if ( cp().items.getTotalMoney()>=8)
    {
    buyBuilding();
    }

    //sellBuilding();

}




public void buyBuilding()
{

cp().listOfBuildingsThatPlayerCanBuy(buildingCards.getStartingBuilding(),buildin
gCards.getStandardBuildingFromTopOfPiles());
    String str[]={"nothing","starting","standard"};
    S("action for buy building"+cp().name+", type is:"+str[cp().canBuy+1]);

    if(cp().canBuy==start)
    {

        buyStartingBuilding(buildingCards.getStartingBuilding());
        S("Items of "+cp().name+" after action:"+cp());
    }
    else
    {
    if(cp().canBuy==stand)
        S("Items of "+cp().name+" before action:"+cp());
        buyStandardBuilding(buildingCards.getStandardBuildingFromTopOfPiles());
        S("Items of "+cp().name+" after action:"+cp());
    }

}


 public void printTotalLoans()
{   int sum=0;
    sum+=player[0].loan+player[1].loan;
    S("Total Loans:"+sum);
    S("TOTAL LOANS OF "+player[0].name+":"+player[0].loan);
    S("TOTAL LOANS OF "+player[1].name+":"+player[1].loan);
}
```

```
public void payMeals(int round)
{
    int meals=cc.roundCards.roundcards[round].Food;
    S("END OF ROUND, PAYING MEALS:"+meals);
    player[Human].payMeals(meals);
    player[Computer].payMeals(meals);

}
public void interest(boolean hasInterest)
{
    if(hasInterest)
    {
        player[Human].interest();player[Computer].interest();
    }
}

public void harvest(int round)
{
    if(round<13)
    {
        player[Human].Harvest();
        player[Computer].Harvest();
    }
}

public void getCardFromPile(int typeOfPile,int Player)
{
    player[Player].getShipCard(cc.shipCards.cardToPlayer(typeOfPile));
}

public void addItemsToDock(int i)
{
    dock.addItem(disks.disk[i].item1);
    dock.addItem(disks.disk[i].item2);
    S("\n"+dock);
}
public void createPlayers()
{
    currentPlayer=randomStartPlayer();
    player[0]=new Player("Emmanouil",true);
    player[1]=new Player("Computer",false);
}
public int otherPlayer()
{
    return currentPlayer==1?0:1;
}
public int alterPlayer(int currentPlayer) {return (currentPlayer==1)?0:1;} ]

public void printDisks()
{
    System.out.print("\nDISKS->[");
    for(int i=0;i<7;i++)
    {
        char c=' ';
        if(player[0].position==i)
            c=player[0].name.charAt(0);
        else if(player[1].position==i)
            c=player[1].name.charAt(0);
        System.out.print("=("+c+"  |"+disks.disk[i]+"| )=");
    }
```

```
    S("]");
}

}
```

```
import java.io.File;
import java.io.IOException;
import java.io.PrintStream;
import java.util.Scanner;

public class play  {

    public static void main(String args[]) throws IOException
    {
        File file = new File("c:\\LeHavreLog\\LogFile.txt");
        PrintStream stream = new PrintStream(file);
        System.setOut(stream);

        new playGame();

        }
    }
```