ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΗΣ ΣΧΕΔΙΑΣΗΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ

Πτυχιακή Εργασία

Ανάπτυξη εκπαιδευτικών παιχνιδιών- Serious game development

Κωνσταντίνος Χαλκιάς (ΑΜ: 71444738)

Επιβλέπουσα: ΕΛΕΝΗ ΑΙΚΑΤΕΡΙΝΗ ΛΕΛΙΓΚΟΥ

Αθήνα, Ιούλιος 2023

Μέλη Εξεταστικής Επιτροπής

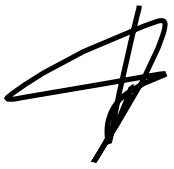
Η πτυχιακή/διπλωματική εργασία εξετάστηκε επιτυχώς από την κάτωθι Εξεταστική Επιτροπή :

ΛΕΛΙΓΚΟΥ ΕΛΕΝΗ ΑΙΚΑΤΕΙΝΗ	ΑΝ. ΚΑΘΗΓΗΤΗΣ	Eleni Aikaterini Leligkou Digitally signed by Eleni Aikaterini Leligkou Date: 2023.08.03 10:51:12 +03'00'
ΚΑΝΤΖΟΣ ΔΗΜΗΤΡΙΟΣ	ΚΑΘΗΓΗΤΗΣ	Dimitrios Kantzos Digitally signed by Dimitrios Kantzos Date: 2023.08.03 12:03:42 +03'00'
ΔΡΟΣΟΣ ΧΡΗΣΤΟΣ	ΕΔΙΠ Α	Christos Drosos Digitally signed by Christos Drosos Date: 2023.08.03 09:14:19 +03'00'

Δήλωση συγγραφέα πτυχιακής/διπλωματικής εργασίας

Ο κάτωθι υπογεγραμμένος Χαλκιάς Κωνσταντίνος του Νικολάου, με αριθμό μητρώου 71444738 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Βιομηχανικής Σχεδίασης και Παραγωγής , δηλώνω υπεύθυνα ότι: «Είμαι συγγραφέας αυτής της πτυχιακής/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 πτυχίου μου».

Ο Δηλών



Κωνσταντίνος Χαλκιάς

Περίληψη

Στην πτυχιακή εργασία κατασκευάστηκε ένα παιχνίδι σοβαρού σκοπού (serious game), δηλαδή ένα παιχνίδι που ο σχεδιασμός του έχει σαν σκοπό να διαχειριστεί ένα πρόβλημα και να ευαισθητοποιήσει τον κόσμο πάνω σε αυτό μέσω ενός διασκεδαστικού και ψυχαγωγικού περιβάλλοντος. Επιπλέον η παρούσα πτυχιακή εργασία αναλύει την ιστορία και τις επιπτώσεις των σοβαρών ηλεκτρονικών παιχνιδιών και των ηλεκτρονικών παιχνιδιών , εστιάζοντας στους διάφορους τύπους που υπάρχουν και στις θετικές και αρνητικές τους επιδράσεις. Ακόμα, εξετάζονται τα serious games και η χρήση τους στην εκπαίδευση, καθώς και ο ρόλος των παιχνιδιών στην ανακύκλωση. Επίσης αναφέρω τα οφέλη που έχουμε από την ανακύκλωση και αναλύω κάποια από τα σημαντικότερα ανακυκλώσιμα υλικά, όπως είναι το χαρτί, το γυαλί, το πλαστικό και το αλουμίνιο και τα υποκατάστατα τους. Το παιχνίδι σοβαρού σκοπού που δημιούργησα αφορά το περιβάλλον και συγκεκριμένα την ανακύκλωση, χρησιμοποιώντας τη δημοφιλή μηχανή ανάπτυξης παιχνιδιών Unity και τη γλώσσα προγραμματισμού C#, με περιβάλλον ανάπτυξης Visual Studio code. Το παιχνίδι σχεδιάστηκε για να προωθεί την ευαισθητοποίηση και να εκπαίδευση του κόσμου πάνω στην ανακύκλωση, προσφέροντας μια δια δραστική και εκπαιδευτική εμπειρία στους χρήστες. Η πτυχιακή εργασία περιλαμβάνει επίσης ανάλυση των μηχανών γραφικών Unreal, CryEngine κ.α. καθώς και ανάλυση του παιχνιδιού που έχει αναπτυχθεί. Επιπλέον αναφέρω μερικά παραδείγματα από τα ποιο γνωστά serious games που υπάρχουν και αναλύω ορισμένα δημοφιλή προγράμματα επεξεργασίας πηγαίου κώδικα. Με τη συγκέντρωση και ανάλυση αυτών των πληροφοριών, παρουσιάζονται συμπεράσματα και προτάσεις για περαιτέρω βελτιώσεις και ανάπτυξη παιχνιδιών σοβαρού σκοπού.

Abstract

The bachelor's thesis focuses on the development of a serious game, which is a game designed with the purpose of addressing and raising awareness about a specific issue within an entertaining and engaging environment. The thesis also analyzes the history and impact of serious electronic games and examines various types of games, as well as their positive and negative effects. Furthermore, it explores the use of serious games in education and their role in recycling. The benefits of recycling are discussed, along with an analysis of some of the most important recyclable materials such as paper, glass, plastic, and aluminum, including their substitutes. The serious game created for this thesis specifically targets the environment and recycling. It was developed using the popular Unity game engine and programmed in the C# programming language, utilizing the Visual Studio Code development environment. The game aims to promote awareness and educate players about recycling, offering an interactive and educational experience. The thesis also includes an analysis of graphic engines such as Unreal and CryEngine, as well as an in-depth analysis of the developed game. Additionally, it provides examples of well-known serious games and discusses popular source code editing programs. By gathering and analyzing this information, the thesis presents conclusions and recommendations for further improvements and the development of serious games.

Περιεχόμενα

Κεφάλαιο 1

1.1 Τι ειναι ανακυκλωση

Εικόνα 1 Το διεθνές σύμβολο της ανακύκλωσης

Η ανακύκλωση είναι η διαδικασία συλλογής και επεξεργασίας «άχρηστων» υλικών που τα μετατρέπουν σε νέα προϊόντα και διαφορετικά θα πετιόνταν ως σκουπίδια σε χωματερές, μέσω της ανακύκλωσης μπορεί να βοηθηθεί η κοινότητά και το περιβάλλον. Υπάρχουν πολλά οφέλη από την ανακύκλωση, και πλέον με τις νέες τεχνολογίες που διαθέτουμε έχουμε την δυνατότητα να ανακυκλώσουμε περισσότερα υλικά. Τα οφέλη είναι μεγάλα και όλοι κερδίζουν όταν οι άνθρωποι υιοθετούν την ανακύκλωση ως καθημερινή συνήθεια, είτε όταν πρόκειται για προσπάθεια μίας μικρής κοινότητας ή και όταν πρόκειται για μια μεγάλη επιχείρηση που μέσω της ανακύκλωσης μπορεί να εξοικονομήσει αρκετά λεφτά με την διαχείριση απορριμμάτων. Αν θέλουμε να διατηρήσουμε τη Γη για τις επόμενες γενιές, η ανακύκλωση είναι πραγματικά σημαντική επίσης εκτός από περιβαλλοντικά οφέλη έχει και οικονομικά. Μερικά ανακυκλώσιμα προϊόντα είναι:

1) Χαρτί
2) Γυαλί (μπουκάλια και γυαλιά)
3) Μέταλλα (σιδηρούχα και μη σιδηρούχα μέταλλα όπως αλουμίνιο, χαλκός, κράματα μετάλλων κ.ά.)
4) Πλαστικά (μπουκάλια, συσκευασίες, πλαστικές σακούλες, πλαστικά αντικείμενα κ.ά.)
5) Αλουμίνιο (κουτιά, συσκευασίες, φύλλα αλουμινίου)
6) Μπαταρίες και ακόμα και ηλεκτρονικά απόβλητα (ηλεκτρονικές συσκευές, κινητά τηλέφωνα, υπολογιστές κ.ά.)
7) Υφάσματα και ρούχα
8) Ξύλο
9) Οργανικά απόβλητα (τρόφιμα)
10) Μπουκάλια και συσκευασίες από πλαστικό PET, HDPE, LDPE κ.ά.
11) Πλαστικές σακούλες και φιλμ από πολυαιθυλένιο
12) Ελαστικά αυτοκινήτων
13) Προϊόντα εικόνας και ήχου
14) Σκυρόδεμα
15) Κ.α

(Fabio M. Lamberti, 2020)

1.1.2 Ορισμός ανακυκλωμένου και ανακυκλώσιμου υλικού

Εικόνα 2 Σύμβολο ανακυκλώσιμου υλικού

Το ανακυκλωμένο υλικό είναι υλικό που προέρχεται από προηγούμενα χρησιμοποιημένα προϊόντα ή υλικά που έχουν υποστεί διαδικασία ανακύκλωσης για να δημιουργηθεί νέο προϊόν. Το σύμβολο των τριών τόξων αντιπροσωπεύει τα τρία στάδια της διαδικασίας παραγωγής από ανακυκλωμένα υλικά. Το πρώτο τόξο συμβολίζει τη συλλογή των υλικών, δηλαδή τη συγκέντρωση των χρησιμοποιημένων προϊόντων που θα υποστούν ανακύκλωση. Αυτά τα υλικά συλλέγονται και μεταφέρονται στη μονάδα παραγωγής. Το δεύτερο τόξο αντιπροσωπεύει τη διαδικασία παραγωγής, όπου τα υλικά ανακυκλώνονται και χρησιμοποιούνται για να δημιουργηθεί ένα νέο προϊόν. Αυτή η διαδικασία είναι ουσιαστικά η δευτερογενής αξιοποίηση των υλικών. Το τρίτο τόξο αντιπροσωπεύει την αγορά του νέου προϊόντος, ολοκληρώνοντας τον κύκλο. Αν ένα προϊόν δεν έχει άλλη ένδειξη(π.χ. 40% προερχόμενο από ανακυκλωμένα υλικά), τότε υπονοείται ότι είναι 100% ανακυκλώσιμο. Ανακυκλώσιμο υλικό σημαίνει ότι ένα υλικό ή ένα προϊόν μπορεί να υποστεί ανακύκλωση. Αυτό σημαίνει ότι το υλικό μπορεί να επαναχρησιμοποιηθεί για να δημιουργηθεί ένα νέο προϊόν, το προϊόν αυτό μπορεί να περιέχει ανακυκλωμένα υλικά ή και όχι. Οι συμβολισμοί του ανακυκλωμένου και του ανακυκλώσιμου υλικού μοιάζουν μεταξύ τους, αλλά έχουν διαφορετική σημασία. Το σύμβολο του ανακυκλωμένου υλικού υποδεικνύει ότι το υλικό προέρχεται από προηγούμενα ανακυκλωμένα υλικά. Αντίθετα, το σύμβολο του ανακυκλώσιμου υλικού απλά υποδεικνύει ότι το προϊόν μπορεί να υποστεί ανακύκλωση, αλλά δεν απαιτείται απαραίτητα η χρήση ανακυκλωμένων υλικών για την παραγωγή του. Συνεπώς, η μελλοντική ανακύκλωση εξαρτάται από την επιλογή και την καλή βούληση του καταναλωτή.

Εικόνα 3 Σύμβολο ανακυκλωμένου υλικού

(A. Rehan, 2018)

1.2 Ιστορική αναδρομή για την ανακύκλωση

Από την αρχαιότητα γνώριζαν ήδη για την ανακύκλωση και την επαναχρησιμοποίηση υλικών. Στην αρχαία Ρώμη, για παράδειγμα, πραγματοποιούνταν η επεξεργασία και επαναχρησιμοποίηση μετάλλων και το σίδερο χρησιμοποιούνταν ξανά και ξανά για να δημιουργηθούν νέα αντικείμενα. Κατά τη διάρκεια του Μεσαίωνα, η ανακύκλωση ήταν ανεπτυγμένη σε διάφορες περιοχές. Οι βιομηχανίες κεριού και χαρτιού, για παράδειγμα,

επαναχρησιμοποιούσαν υλικά και απορρίμματα για την παραγωγή νέων προϊόντων. Με την έναρξη της βιομηχανικής επανάστασης, η ανακύκλωση απέκτησε νέα σημασία. Οι βιομηχανίες εκμεταλλεύτηκαν την επαναχρησιμοποίηση υλικών και την ανακύκλωση για να εξοικονομήσουν πόρους και να μειώσουν το κόστος παραγωγής. Στον τομέα του χαρτιού, για παράδειγμα, η ανακύκλωση έπαιξε σημαντικό ρόλο στην παραγωγή νέου χαρτιού. Τέλος στην σύγχρονη εποχή έχουμε επίγνωση ότι για την περιβαλλοντική βιωσιμότητα, η ανακύκλωση έχει γίνει κρίσιμης σημασίας. Προγράμματα ανακύκλωσης έχουν επιβληθεί σε πολλές χώρες και προωθούνται οι πρακτικές της ανακύκλωσης σε ολόκληρο τον κόσμο. (Miller, 2012)

1.2.1 Το σύμβολο της ανακύκλωσης

Το σύμβολο της ανακύκλωσης έχει μια ενδιαφέρουσα ιστορία που ξεκινά από τη δεκαετία του 1970. Το πρώτο σχέδιο του συμβόλου ανακύκλωσης δημιουργήθηκε από τον γραφίστα Gary Anderson το 1970 στο πλαίσιο μιας διαγωνιστικής εκδήλωσης του Λογοτύπου της Επέτειου της Γης. Αυτό το αρχικό σχέδιο περιλάμβανε τρία βέλη που αντιπροσώπευαν τα τρία βασικά υλικά που μπορούν να ανακυκλωθούν: το γυαλί, το χαρτί και το μέταλλο. Το σχέδιο του Anderson δεν έλαβε αρχικά μεγάλη αναγνώριση, αλλά τη δεκαετία του 1980 αρχίσαμε να βλέπουμε την εμφάνισή του σε συσκευασίες και πινακίδες ανακύκλωσης. Το 1988, οι Ηνωμένες Πολιτείες εισήγαγαν μια πρωτοβουλία για την καθιέρωση ενός κοινού συμβόλου ανακύκλωσης που θα χρησιμοποιούν όλες οι πολιτείες. Ο σχεδιασμός του Anderson υιοθετήθηκε ως το επίσημο σύμβολο της ανακύκλωσης. (Lardner, 2003)

1.3 ΙΕΡΑΡΧΗΣΗ ΤΩΝ ΕΡΓΑΣΙΩΝ ΔΙΑΧΕΙΡΙΣΗΣ ΤΩΝ ΑΠΟΒΛΗΤΩΝ

1) Πρόληψη: Δηλαδή την μείωση της ποσότητας των αποβλήτων, και παράλληλα μείωση των οποιοδήποτε συνεπειών για το περιβάλλον και την ανθρώπινη υγεία από επικίνδυνες ουσίες.
2) Επαναχρησιμοποίηση: Τα προϊόντα που αποτελούν πλέον απόβλητα προετοιμάζονται να επαναχρησιμοποιηθούν.
3) Ανακύκλωση: Είναι η κάθε ενέργεια, διαδικασία ή εργασία με την οποία τα απόβλητα μετατρέπονται εκ νέου σε προϊόντα ή υλικά που προορίζονται να εξυπηρετήσουν και πάλι τον αρχικό τους σκοπό ή και άλλους σκοπούς.
4) Ανάκτηση: Δηλαδή είναι η κάθε ενέργεια που επιδιώκει την αξιοποίηση του αποβλήτου με ακόμα να πετύχει και βελτίωση από κάποια χαρακτηριστικά τους. Επίσης έχει να κάνει και με την ανάκτηση ενέργειας από τα ίδια τα απόβλητα.
5) Διάθεση: Αφορά την διάθεση των αποβλήτων. (Pandey, 2019)

1.4 ΟΦΕΛΗ ΑΝΑΚΥΚΛΩΣΗΣ

1) Με την ανακύκλωση οι άνθρωποι μπορούν να αποτρέψουν την είσοδο εκατομμυρίων τόνων υλικού στο Χ.Υ.Τ.Α (Χώρος Υγειονομικής Ταφής Απορριμμάτων). Με αυτόν τον τρόπο εξοικονομούν χώρο για απόβλητα που δεν μπορούν να ανακυκλωθούν αλλά ακόμα εξοικονομούνται σημαντικοί οικονομικοί πόροι που οι Δήμοι οφείλουν να πληρώσουν ως ‘’τέλος ταφής’’ για τα απόβλητα που οδηγούνται σε ταφή ενώ θα μπορούσαν να τα ανακυκλώσουν . Επίσης με την συνεχείς σπατάλη των χώρων Χ.Υ.Τ.Α όχι μόνο μολύνουμε το περιβάλλον αλλά εμποδίζουμε και την ομορφιά της πόλης.

2) Μέσω της ανακύκλωσης δημιουργήθηκε η κυκλική οικονομία που είναι ένα οικονομικό μοντέλο που εστιάζει στη ανάκτηση , την επαναχρησιμοποίηση προϊόντων και στην μείωση

της σπατάλης των πόρων που χρησιμοποιούνται στην παραγωγική διαδικασία. Σύμφωνα με την έκθεση της ευρωπαϊκής επιτροπής προς το ευρωπαϊκό κοινοβούλιο, το 2015 η επιτροπή εφάρμοσε σχέδιο δράσης που αφορά την κυκλική οικονομία. Σε ένα χρόνο οι τομείς που σχετίζονται με την κυκλική οικονομία υπήρχε αύξηση σε νέες θέσεις εργασίας της τάξης του 6% σε σχέση με το 2012.

3) Όταν ανακυκλώνουμε, τα χρησιμοποιημένα υλικά μετατρέπονται σε νέα προϊόντα, μειώνοντας την ανάγκη κατανάλωσης φυσικών πόρων. Εάν τα χρησιμοποιημένα υλικά δεν ανακυκλώνονται, δημιουργούνται νέα προϊόντα με την εξαγωγή φρέσκων πρώτων υλών από τη Γη, μέσω της εξόρυξης και της δασοκομίας. Η ανακύκλωση βοηθά στη διατήρηση των φυσικών πόρων της Γης, όπως πρώτες ύλες, ορυκτά, δέντρα κ.λπ. Προστατεύει τους φυσικούς οικότοπους και διατηρεί τους φυσικούς πόρους για τις μελλοντικές γενιές.

4) Σύμφωνα με τον Rinkesh Kukreja το να χρησιμοποιούμε ανακυκλώσιμα υλικά στην διαδικασία παραγωγής καταναλώνεται πολύ λιγότερη ενέργεια από το να παράγουμε νέα προϊόντα από πρώτες ύλες. Επίσης η ανακύκλωση μειώνει την ανάγκη για εξόρυξη (λατομεία και υλοτόμηση), διύλιση και επεξεργασία πρώτων υλών, τα οποία προκαλούν σημαντική ρύπανση του αέρα και των υδάτων. Οι ρύποι που απελευθερώνονται στον αέρα και το νερό μπορούν να μειωθούν σημαντικά με την αύξηση της ανακύκλωσης

5)) Η υπερθέρμανση του πλανήτη είναι πλέον ένα φλέγον ζήτημα. Οι συνεχείς εκπομπές αερίων του θερμοκηπίου στον αέρα συμβάλλουν στην υπερθέρμανση του πλανήτη και, ως εκ τούτου, στην κλιματική αλλαγή. Η ανακύκλωση παίζει μεγάλο ρόλο στην αντιμετώπιση του προβλήματος

6) Η ανακύκλωση μπορεί να τονώσει τον τουρισμό μιας περιοχής, ένα καθαρό περιβάλλον είναι φιλόξενο και θα προσελκύει περισσότερο κόσμο.

7) Με βάση το νέο σύστημα ‘’Πληρώνω όσο πετάω’’ τα νοικοκυριά που ανακυκλώνουν θα πληρώνουν λιγότερα δημοτικά τέλη, αφού πλέον θα χρεώνονται με βάση την ποσότητα των παραγόμενων από αυτά αποβλήτων. (Fabio M. Lamberti, 2020), (A. Rehan, 2018)

1.4.1 ΠΡΩΤΟΓΕΝΗΣ ΑΝΑΚΥΚΛΩΣΗ

Η πρωτογενής ανακύκλωση αναφέρεται στη διαδικασία κατασκευής νέων προϊόντων από πρώτες ύλες, που προέρχονται από φυσικούς πόρους. Αυτή η μορφή ανακύκλωσης συμβάλλει στη μείωση της εξόρυξης πρώτων υλών και στην εξοικονόμηση ενέργειας. Η διαδικασία της πρωτογενούς ανακύκλωσης περιλαμβάνει τα εξής βήματα:

1) Εξόρυξη πρώτων υλών: Η πρώτη φάση είναι η εξόρυξη φυσικών πόρων, όπως ορυκτά, ξύλο ή πετρέλαιο, ανάλογα με τον τύπο του προϊόντος που θα κατασκευαστεί.

2) Μεταποίηση: Οι πρώτες ύλες υπόκεινται σε μια σειρά επεξεργασιών για να γίνουν κατάλληλες για την παραγωγή νέων προϊόντων. Αυτές οι επεξεργασίες μπορεί να περιλαμβάνουν κοπή, άλεση, χημική επεξεργασία και άλλες τεχνικές.

3) Κατασκευή νέων προϊόντων: Οι επεξεργασμένες πρώτες ύλες χρησιμοποιούνται για την κατασκευή νέων προϊόντων, όπως συσκευασίες, είδη ένδυσης, έπιπλα, ηλεκτρονικά και άλλα.

Η πρωτογενής ανακύκλωση έχει σημαντική συνεισφορά στην προστασία του περιβάλλοντος και την βιώσιμη ανάπτυξη. Τα κύρια πλεονεκτήματά της περιλαμβάνουν:

1) Μείωση της εξόρυξης πρώτων υλών: Η πρωτογενής ανακύκλωση μειώνει την ανάγκη για εξόρυξη φυσικών πόρων, προστατεύοντας οικοσυστήματα και μειώνοντας την περιβαλλοντική καταπόνηση.

2) Εξοικονόμηση ενέργειας: Η παραγωγή νέων προϊόντων από ανακυκλωμένα υλικά απαιτεί συνήθως λιγότερη ενέργεια από ό,τι η παραγωγή από πρώτες ύλες. Αυτό συμβάλλει στη μείωση της ενεργειακής κατανάλωσης και των εκπομπών αερίων του θερμοκηπίου.

3) Μείωση των αποβλήτων: Η πρωτογενής ανακύκλωση συμβάλλει στη μείωση των αποβλήτων, καθώς τα υλικά ανακυκλώνονται και δεν καταλήγουν σε χώρους υγειονομικής ταφής.

4) Προώθηση της κυκλικής οικονομίας: Η πρωτογενής ανακύκλωση συμβάλλει στην προώθηση της κυκλικής οικονομίας, μειώνοντας την εξάρτηση από την ανεξέλεγκτη εξόρυξη και παραγωγή νέων υλικών.

(A. Rehan, 2018)

1.4.2 ΔΕΥΤΕΡΟΓΕΝΗΣ ΑΝΑΚΥΚΛΩΣΗ

Η δευτερογενής ανακύκλωση αποτελεί μια σημαντική διαδικασία στον τομέα της ανακύκλωσης, καθώς επιτρέπει την επαναχρησιμοποίηση των υλικών από τα απόβλητα μετά την πρώτη χρήση τους. Αυτή η μορφή ανακύκλωσης συμβάλλει στη μείωση της ρύπανσης, την εξοικονόμηση πόρων και την προστασία του περιβάλλοντος. Η δευτερογενής ανακύκλωση αναφέρεται στη διαδικασία μετατροπής και επεξεργασίας των αποβλήτων με σκοπό την επαναχρησιμοποίηση των υλικών. Η σημασία της δευτερογενούς ανακύκλωσης έγκειται στην προστασία του περιβάλλοντος, καθώς μειώνει την ανάγκη για εξόρυξη νέων πρώτων υλών, την εξοικονόμηση ενέργειας και τη μείωση των αποβλήτων που καταλήγουν σε χώρους υγειονομικής ταφής. Οι τεχνολογίες δευτερογενούς ανακύκλωσης περιλαμβάνουν:

1) Μηχανική ανακύκλωση: Περιλαμβάνει τη μηχανική διαλογή και επεξεργασία των αποβλήτων για την εξαγωγή και επαναχρησιμοποίηση συγκεκριμένων υλικών. Για παράδειγμα περιλαμβάνει την αποδοτική διαλογή των πλαστικών, του χαρτιού και του γυαλιού για την παραγωγή ανακυκλωμένων προϊόντων.

2) Χημική ανακύκλωση: Περιλαμβάνει τη χρήση χημικών διεργασιών για την αποσύνθεση και ανακατασκευή των υλικών από τα απόβλητα. Για παράδειγμα περιλαμβάνει την πυρόλυση πλαστικών για την παραγωγή πετρελαϊκών προϊόντων ή την ανάκτηση μετάλλων από ηλεκτρονικά απόβλητα.

3) Βιολογική ανακύκλωση: Περιλαμβάνει την ανάπτυξη βιομηχανικών διεργασιών για τη μετατροπή των αποβλήτων σε βιολογικά προϊόντα, όπως λίπασμα ή βιοαέριο. Αυτή η τεχνολογία επιτρέπει την αποτελεσματική εκμετάλλευση των οργανικών αποβλήτων και τη μείωση της ρύπανσης του περιβάλλοντος. (A. Rehan, 2018)

1.4.3 ΤΡΙΤΟΓΕΝΗΣ ΑΝΑΚΥΚΛΩΣΗ

Η τριτογενής ανακύκλωση αναφέρεται στην ανακύκλωση υλικών που έχουν ήδη υποστεί ανακύκλωση μία ή δύο φορές. Σε αντίθεση με την πρωτογενή ανακύκλωση, όπου τα υλικά επεξεργάζονται για πρώτη φορά για να παραχθούν νέα προϊόντα, η τριτογενής ανακύκλωση επικεντρώνεται στην επεξεργασία υλικών που έχουν ήδη ανακυκλωθεί για να

παραχθούν νέα υψηλής ποιότητας προϊόντα ή υλικά. Επίσης η τριτογενής ανακύκλωση περιλαμβάνει διάφορες διαδικασίες, όπως η απομάκρυνση ρύπων και επικίνδυνων ουσιών από τα υλικά, η απομόνωση και η επεξεργασία διαφορετικών συστατικών, και η ανακατασκευή ή η μετατροπή τους σε νέα προϊόντα. Μερικές από τις κύριες τεχνολογίες που χρησιμοποιούνται στην τριτογενή ανακύκλωση περιλαμβάνουν τη χημική ανακύκλωση, τη μηχανική ανακύκλωση, την ανάκτηση ενέργειας και την εννοιολογική ανακύκλωση (A. Rehan, 2018)

1.5 ΕΙΔΗ ΑΝΑΚΥΚΛΩΣΗΣ ΑΝΑ ΥΛΙΚΟ

1)Ανακύκλωση χαρτιού:

Αρχικά με την ανακύκλωση του χαρτιού ελαττώνεται ο αριθμός των δέντρων που κόβονται για την παραγωγή του, σύμφωνα με το ΔΟΑΝΥΣ (Διεθνής Οργανισμός Ανακύκλωσης Υλικών Συσκευασίας) με την βιομηχανία ανακύκλωσης έχουν δημιουργηθεί νέες θέσεις εργασίας, δεν σπαταλάτε άσκοπα χώρος στα Χ.Υ.Τ.Α άρα μεγαλώνει η διάρκεια ζωής τους, έχουμε μείωση της κατανάλωσής ενέργειας και κατανάλωσής νερού συγκεκριμένα με την ανακύκλωση χαρτιού διασώζονται 15-20 δέντρα, 30 κυβικά νερό, εκατοντάδες κιλοβατώρες ηλεκτρικής ενέργειες και περίπου 230 κιλά ισοδύναμου πετρελαίου. Επίσης με την ανακύκλωση χαρτιού έχουμε μείωση στις εκπομπές αερίων και μείωση των ρυπογόνων εκπομπών του νερού, που συμβάλουν στο φαινόμενο του θερμοκηπίου. Η ζήτηση για το χαρτί έχει ανέβει κατακόρυφα, το 1915 η κατανάλωση ήταν 5.000.000 τόνους για όλο τον πλανήτη ενώ σήμερα η ετήσια κατανάλωση ξεπερνά τους 200.000.000 τόνους, άρα καταλαβαίνουμε ότι για να παραχθεί τόση μεγάλη ποσότητα χαρτιού οδηγούμαστε σε μεγάλη κλίμακα καταστροφής του οικοσυστήματος. (Tyler Miller, 2004)

2)Ανακύκλωση Πλαστικών:

Με μελέτη που πραγματοποίησε η WWF προκύπτει ότι τα σκουπίδια που εντοπίζονται στην Μεσόγειο αποτελούν το 95%, ακόμα η παραγωγή πλαστικού αυξάνεται κατά 4% ετησίως. Σύμφωνα με την ιστοσελίδα free-recycle στην βιομηχανία πλαστικών κατασκευάζονται πολλά είδη αλλά κυρίως χρησιμοποιούνται τα εξής παράγωγα :

PET (τερεφθαλικός εστέρας του πολυαιθυλένιο)

HDPE (υψηλής πυκνότητας πολυαιθυλένιο)

LDPE (χαμηλής πυκνότητας πολυαιθυλένιο)

PP (πολυπροπυλένιο)

Σχεδόν όλα τα πλαστικά βασίζονται στο πετρέλαιο λόγω της επεξεργασίας που πρέπει να υποστεί το πετρέλαιο για να προκύψει το πλαστικό απαιτούνται μεγάλα ποσοστά ενέργειας. Με την ανακύκλωση μπορούμε να εξοικονομήσουμε κατά μέσω όρο 70% ενέργεια, για παράδειγμα η ανακύκλωση ενός πλαστικού μπουκαλιού εξοικονομεί ενέργεια για να ανάψει μια λάμπα ισχύος 60W για 6 ώρες. Ακόμα γίνεται αποσυμφόρηση στις χωματερές και περιορίζονται οι εκπομπές αερίων στην ατμόσφαιρα, γιατί πολλά πλαστικά όταν καίγονται παράγουν πολύ τοξικές ενώσεις και για να υποδομηθούν πλήρως υπάρχουν εκτιμήσεις ότι χρειάζονται πάνω από 500 χρόνια. (Tyler Miller, Βιώνοντας στο Περιβάλλον II:, 1999)

3)Ανακύκλωση μετάλλων:

Τα μέταλλα είναι πολύτιμα υλικά που μπορούν να ανακυκλωθούν ξανά και ξανά χωρίς να υποβαθμίζονται οι ιδιότητές τους, με την ανακύκλωση των μετάλλων διατηρούμε τους φυσικούς πόρους του πλανήτη. Επιπλέον μέσω της ανακύκλωσης εκπέμπεται λιγότερο διοξείδιο του άνθρακα καθώς και άλλες επιβλαβείς για το περιβάλλον ουσίες, ακόμα μειώνεται ο βαθμός των αποβλήτων που αποστέλλονται σε χώρους υγειονομικής ταφής. Σύμφωνα με στοιχεία από την ιστοσελίδα free-recycle από 1 τόνο ανακυκλωμένου αλουμινίου εξοικονομούνται 4 τόνοι βιωξίτης, 500 κιλά σόδα, 700 κιλά πετρέλαιο καύσης, 35 κιλά φθοριούχο αλουμίνιο και απαιτεί μόνο 2000KWh για την παραγωγή του, ακόμα εντείνεται και το φαινόμενο του θερμοκηπίου από τις εκπομπές διοξειδίου του άνθρακα για την παρασκευή νέων προϊόντων. Εκτός από το αλουμίνιο, πολύ σημαντική είναι και η ανακύκλωση των μεταλλικών συσκευασιών από σιδηρούχα μέταλλα. Οι περισσότερες σιδηρούχες μεταλλικές συσκευασίες είναι κατασκευασμένες από λευκοσίδηρο με την ανακύκλωση αυτού του υλικού βοηθάει στο να έχουμε μείωση σε ετήσια βάση ρύπανση ατμόσφαιρας κατά 85%, μόλυνση υδάτων κατά 76% και χρήση υδάτων κατά 40% (Tyler Miller, Περιβαλλοντικές επιστήμες, 2004)

	Αλουμίνιο	Χάλυβας	Χαρτί	Γυαλί
Μείωση χρήση ενέργειας	90-97%	47-45%	23-77%	32%
Μείωση ρύπανσης αέρα	95%	85%	75%	20%
Μείωση ρύπανσης νερού	97%	76%	35%	-
Μείωση χρήσης νερού	-	40%	58%	50%

4)Ανακύκλωση γυαλιού

Το γυαλί ήταν ένα ακριβό υλικό μέχρι το 1800 τότε που τα τρία κύρια συστατικά του άμμος σόδα και ασβεστόλιθος ήταν δυσεύρετα, σήμερα όμως οι πρώτες ύλες είναι πιο εύκολα διαθέσιμες επομένως είναι αρκετά φθηνή η κατασκευή τους. Το γυαλί όταν ανακυκλώνεται δεν χάνει την αξία και την ποιότητα του αυτό σημαίνει ότι έχουμε την δυνατότητα να το ανακυκλώσουμε πολλές φορές. Υπολογίζεται ότι ο χρόνος που χρειάζεται για να αποσυντεθεί πλήρως το γυαλί είναι στα ένα εκατομμύριο χρόνια, που αυτό δημιουργεί μεγάλο πρόβλημα στους χώρους υγειονομικής ταφής. Ακόμα όπως γίνεται και στα άλλα ανακυκλώσιμα υλικά εξοικονομούνται φυσικοί πόροι και μειώνονται οι εκπομπές αερίων του θερμοκηπίου στον αέρα που συμβάλλουν στην υπερθέρμανση του πλανήτη. Το ανακυκλωμένο γυαλί εξοικονομεί 50 τοις εκατό ενέργεια από το να κατασκευάσουμε νέο και η ανακύκλωση μόνο ενός γυάλινου δοχείου εξοικονομεί αρκετή ενέργεια για να ανάψει μια λάμπα 100 watt για τέσσερις ώρες. Το ανακυκλωμένο γυαλί δημιουργεί 20% λιγότερη ατμοσφαιρική ρύπανση και 50% λιγότερη ρύπανση του νερού, και ένα τόνος γυαλιού που κατασκευάζεται από 50% ανακυκλωμένα υλικά εξοικονομεί 250 κιλά απόβλητα εξόρυξης. (Wright, 2013)

5)Ανακύκλωση Ηλεκτρονικών Συσκευών

Ανακύκλωση ηλεκτρονικών συσκευών αφορά την ανακύκλωση ηλεκτρικού και ηλεκτρονικού εξοπλισμού η οποία τροφοδοτείται από μια μπαταρία ή βύσμα όπως είναι οι υπολογιστές, κινητά τηλέφωνα, τηλεοράσεις , tablet , συσκευές κουζίνας κ.λπ. Η συνεχής

αύξηση ηλεκτρονικών αποβλήτων ο οποίος μεγάλος όγκος τους οφείλεται από τις αναπτυγμένες χώρες, προκαλεί μεγάλο πρόβλημα για το περιβάλλον και την ανθρώπινη υγεία. Με την αύξηση ηλεκτρονικών αποβλήτων αυξάνεται και η ανάγκη για ανακύκλωση, σύμφωνα με έρευνες το 75% των παλιών ηλεκτρικών συσκευών εξακολουθούν να αποθηκεύονται σε νοικοκυριά λόγω της μη διαθεσιμότητας σημείων ανακύκλωσης. Η ανακύκλωση πρώτων υλών από ηλεκτρονικά στο τέλος του κύκλου ζωής τους είναι η αποτελεσματικότερη λύση στο πρόβλημα των ηλεκτρονικών αποβλήτων. Οι περισσότερες ηλεκτρονικές συσκευές περιέχουν υλικά που μπορούν να επαναχρησιμοποιηθούν για μελλοντικές χρήσεις, με αυτόν τον τρόπο μέσω της ανακύκλωσης μπορούμε να κατασκευάσουμε νέα προϊόντα εξοικονομώντας ενέργεια , να διατηρούνται οι πόροι της Γης και να μειώνονται τα αποτυπώματα άνθρακα. . Επιπλέον, η ανακύκλωση μειώνει την ποσότητα των εκπομπών αερίων του θερμοκηπίου που προκαλούνται από την παραγωγή νέων προϊόντων. Ακόμα αποτρέπει τη ρύπανση του οικοσυστήματος, τα ηλεκτρονικά εξαρτήματα περιέχουν μια σειρά από τοξικές ουσίες οι οποίες θέτουν σε κίνδυνο την υγεία και το περιβάλλον. (Wright, 2013)

6)Ένδυση και Κλωστοϋφαντουργία

Τα τελευταία χρόνια, αγοράζουμε περισσότερα ρούχα από ποτέ με αποτέλεσμα όλο και περισσότερα κλωστοϋφαντουργικά προϊόντα που πετιούνται στις χωματερές αλλά μέσω της ανακύκλωσης μπορούμε να εξοικονομήσουμε χώρο. Οι φυσικές ίνες χρειάζονται χρόνια για να αποσυντεθούν, ενώ οι τεχνητές ίνες δεν αποσυντίθενται. Τα μάλλινα ρούχα όταν αποσυντίθενται απελευθερώνουν μεθάνιο και διοξείδιο του άνθρακα στην ατμόσφαιρα, που αυτό συμβάλλει στην υπερθέρμανση του πλανήτη. Τα συνθετικά υφάσματα στη χωματερή απελευθερώνουν υποξείδιο του αζώτου που είναι ένα ισχυρό αέριο του θερμοκηπίου. Επιπλέον, τοξικές ουσίες μολύνουν τα υπόγεια ύδατα και το περιβάλλον έδαφος. Προκειμένου να περιοριστεί η ανάπτυξη των κλωστοϋφαντουργικών απορριμμάτων, η ανακύκλωση υφασμάτων είναι η μόνη επιλογή. Επίσης οι πόροι της Γης είναι πεπερασμένοι, τα δύο βασικά στοιχεία στην κλωστοϋφαντουργία είναι ο πολυεστέρας και το βαμβάκι, ο πολυεστέρας έχει ως βάση το πετρέλαιο του οποίου η εξαγωγή είναι επιβλαβής για το περιβάλλον. Το 2014 η παγκόσμια παραγωγή νημάτων πολυεστέρα και ινών βαμβακιού ήταν περίπου 65 εκατομμύρια τόνοι, ο αριθμός αυτός εκτιμάται ότι θα αυξηθεί τα επόμενα χρόνια. Τα κλωστοϋφαντουργικά προϊόντα έχουν σημαντική επίδραση στο περιβάλλον κατά τη διάρκεια του κύκλου ζωής τους για να κατασκευαστούν απαιτείται μεγάλη ποσότητα ηλεκτρικής ενέργειας, νερού και χημικών. Προκειμένου να μειωθούν οι εκπομπές αερίων του θερμοκηπίου πρέπει να γίνουν προσπάθειές αύξηση της ανακύκλωσης. Επίσης ένας ακόμα λόγος και πιθανών ο κύριος λόγος που οι άνθρωποι επιλέγουν να ανακυκλώνουν ρούχα είναι ότι δωρίζονται στους ανθρώπους που έχουν ανάγκη. (Wright, 2013)

7) Ανακύκλωση ΑΕΚΚ (απόβλητα εκσκαφών, κατασκευών και κατεδαφίσεων)

Ως δομικά υλικά ορίζονται τα υλικά εκείνα που χρησιμοποιούνται στην κατασκευή και στην συντήρηση των τεχνικών-δομικών έργων τα οποία χωρίζονται σε δύο κατηγορίες :

1)Σε φυσικά υλικά που είναι το ξύλο, ο πηλός και η πέτρα τα οποία δεν έχει προηγηθεί κάποια ανθρώπινη επεξεργασία

2)Σε σύνθετα υλικά στα οποία έχει προηγηθεί ανθρώπινη επεξεργασία όπως είναι το σκυρόδεμα, ο γύψος, η άσφαλτος, ο χάλυβας και το αλουμίνιο.

Η ανακύκλωση και επαναχρησιμοποίηση των αδρανών υλικών είναι σημαντική για την διαφύλαξη του περιβάλλοντος και την κάλυψη αναγκών της κοινωνίας. Η κατασκευή των σύνθετων υλικών προκαλούν μεγάλες περιβαλλοντικές επιπτώσεις όπως για παράδειγμα με την παραγωγή του σκυροδέματος απελευθερώνονται στην ατμόσφαιρα διοξείδιο του άνθρακα , υδρογονάνθρακες, μονοξείδιο του άνθρακα και διοξείδιο του θείου, καθώς και πολλά βαρέα μέταλλα. Επίσης για την κατασκευή των σύνθετων υλικών απαιτούνται φυσικοί πόροι οι οποίοι είναι πεπερασμένοι, η ανακύκλωση ορισμένων υλικών, βοηθά στον έλεγχο και στη καθυστέρηση της εξάντλησης των φυσικών αποθεμάτων. Μέσω της ανακύκλωσης η παραγωγή δευτερογενών υλικών απαιτεί λιγότερη ενέργεια, άρα προκαλεί μικρότερες εκπομπές αερίων του θερμοκηπίου, ακόμα πολλά δομικά υλικά δεν καταλήγουν στο περιβάλλον για να το ρυπαίνουν και να υποβαθμίζεται αισθητά (Tyler Miller, Περιβαλλοντικές επιστήμες, 2004)

1.5.1 Εναλλακτικές λύσεις για τη χρήση πλαστικού

Η χρήση πλαστικού έχει αρνητικές επιπτώσεις στο περιβάλλον και την υγεία μας, και γι' αυτόν τον λόγο αναζητούνται εναλλακτικές λύσεις που μπορούν να μειώσουν την εξάρτησή μας από το πλαστικό. Ορισμένες από αυτές τις εναλλακτικές λύσεις περιλαμβάνουν:

1) Βιοδιασπώμενα υλικά: Τα βιοδιασπώμενα υλικά είναι φυσικής προέλευσης υλικά που μπορούν να διασπαστούν από φυσικές διεργασίες, όπως η βιοδιάσπαση από μικροοργανισμούς. Αυτά τα υλικά μπορούν να χρησιμοποιηθούν για πολλά προϊόντα, όπως συσκευασίες τροφίμων, πιάτα και μαχαιροπήρουνα.
2) Ανακυκλωμένα υλικά: Η χρήση ανακυκλωμένων υλικών μπορεί να μειώσει την ανάγκη για νέα πλαστικά. Τα ανακυκλωμένα υλικά μπορούν να χρησιμοποιηθούν για την παραγωγή νέων προϊόντων, όπως μπουκάλια νερού, σακούλες αγορών και έπιπλα.
3) Φιλικές προς το περιβάλλον εναλλακτικές: Υπάρχουν πολλές φιλικές προς το περιβάλλον εναλλακτικές για τη χρήση πλαστικού. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε αντικαταστάσεις για πλαστικές σακούλες, όπως υφασμάτινες ή χαρτοπετσέτες, και να επιλέξουμε προϊόντα με λιγότερη συσκευασία.

1.5.2 Εναλλακτικές λύσεις για τη χρήση γυαλιού

Οι εναλλακτικές λύσεις για τη χρήση γυαλιού περιλαμβάνουν την αντικατάσταση των πλαστικών υλικών με γυάλινα προϊόντα. Το γυαλί είναι ένα υλικό που μπορεί να προσφέρει πολλά πλεονεκτήματα σε σχέση με το πλαστικό, όπως η ανακυκλωσιμότητα και η μεγαλύτερη αντοχή. Ορισμένες εναλλακτικές λύσεις για τη χρήση γυαλιού περιλαμβάνουν:

1) Ανακυκλωμένο γυαλί: Η χρήση ανακυκλωμένου γυαλιού σε προϊόντα συσκευασίας, όπως μπουκάλια και γυάλινα βάζα, μπορεί να μειώσει την κατανάλωση νέων πόρων και να μειώσει τις εκπομπές αερίων του θερμοκηπίου.
2) Αντικατάσταση πλαστικών συσκευασιών: Το γυαλί μπορεί να αντικαταστήσει πλαστικές συσκευασίες, όπως μπουκάλια νερού, αναψυκτικών και καλλυντικών. Αυτό μπορεί να μειώσει τη χρήση πλαστικού και να συμβάλει στην ανακύκλωση των γυάλινων συσκευασιών.
3) Γυάλινα αντικείμενα για την καθημερινή χρήση: Μπορούμε να προτιμήσουμε γυάλινα αντικείμενα, όπως ποτήρια, πιάτα και μπολ, αντί για πλαστικά αντίστοιχα. Αυτό μπορεί να μειώσει την κατανάλωση πλαστικού και να προάγει την ανακύκλωση των γυάλινων αντικειμένων.

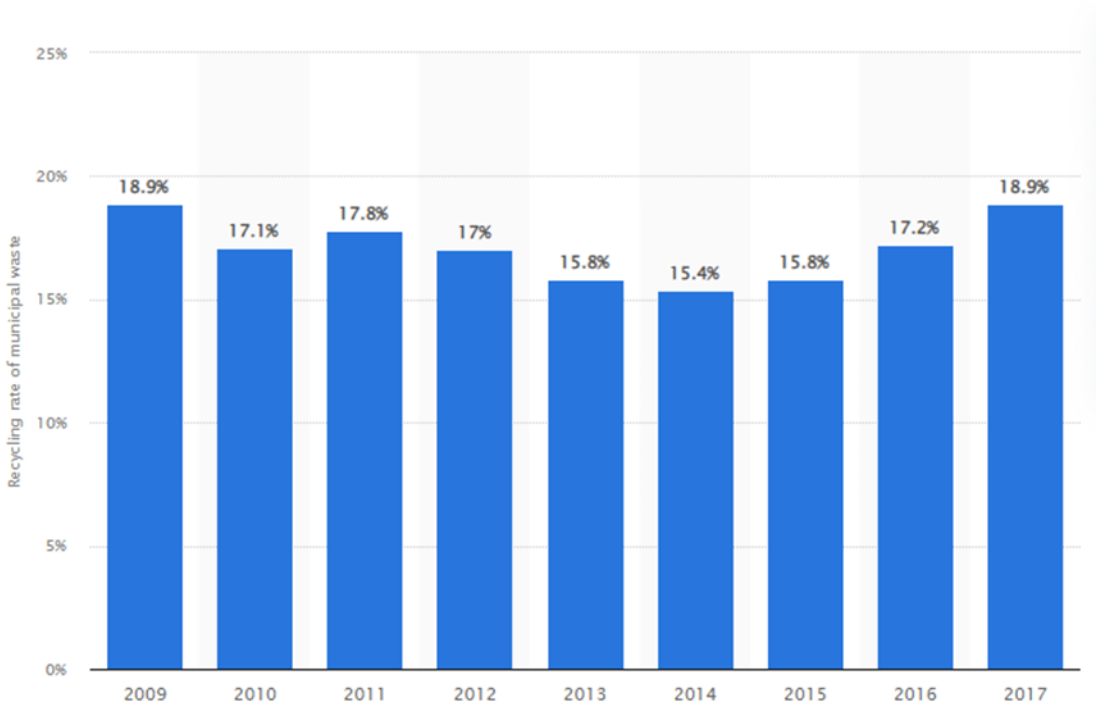
1.5.3 Εναλλακτικές λύσεις για τη χρήση αλουμινίου

Οι εναλλακτικές λύσεις για τη χρήση αλουμινίου περιλαμβάνουν την αντικατάσταση των προϊόντων από αλουμίνιο με άλλα υλικά που είναι πιο φιλικά προς το περιβάλλον. Οι κύριες εναλλακτικές λύσεις περιλαμβάνουν:

1) Ανακυκλωμένα υλικά: Η χρήση ανακυκλωμένων υλικών αντί για το αλουμίνιο μπορεί να μειώσει την ανάγκη για εξόρυξη και παραγωγή νέων αλουμινένιων προϊόντων. Οι ανακυκλωμένοι πόροι αλουμινίου μπορούν να χρησιμοποιηθούν για την κατασκευή νέων προϊόντων, όπως συσκευασίες και μεταλλικά αντικείμενα.
2) Βιώσιμα υλικά: Η χρήση βιώσιμων υλικών, όπως τα οργανικά υλικά, το ξύλο και τα φυτικά ινοπλάστικα, μπορεί να αντικαταστήσει τη χρήση αλουμινίου σε ορισμένες εφαρμογές. Αυτά τα υλικά είναι φιλικά προς το περιβάλλον, ανανεώσιμα και μπορούν να αποτελέσουν βιώσιμες εναλλακτικές λύσεις.
3) Σχεδιασμός προϊόντων με μεγαλύτερη διάρκεια ζωής: Ο σχεδιασμός προϊόντων που είναι ανθεκτικά, επισκευάσιμα και μπορούν να χρησιμοποιηθούν για μεγαλύτερο χρονικό διάστημα μπορεί να μειώσει την ανάγκη για νέα αλουμινένια προϊόντα.

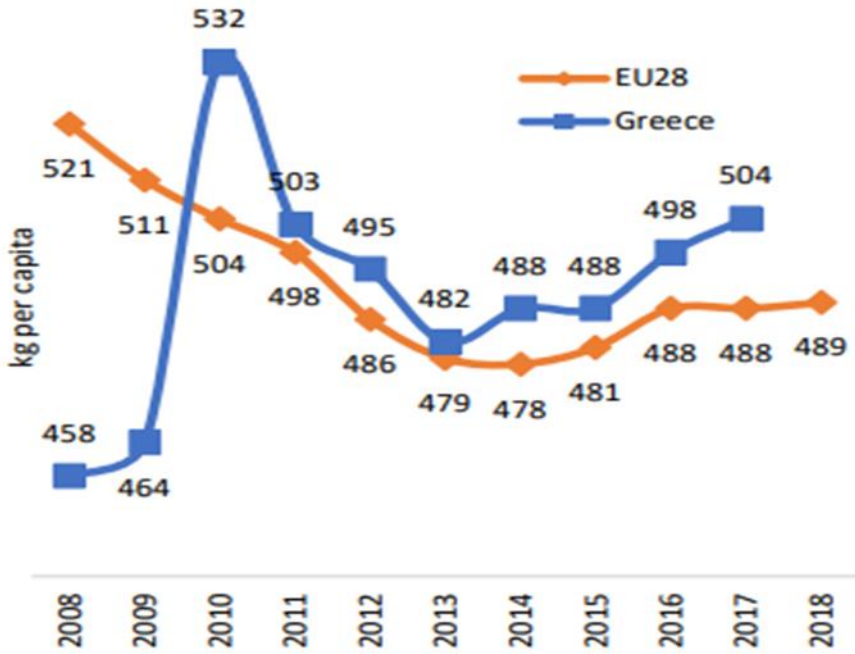
1.6 ΔΙΑΧΕΊΡΗΣΗ ΣΤΕΡΕΩΝ ΑΠΟΒΛΗΤΩΝ ΚΑΙ ΑΝΑΚΥΚΛΩΣΗ ΣΤΗΝ ΕΛΛΑΔΑ

Η ανακύκλωση παίζει πλέον μεγαλύτερο ρόλο στη διαχείριση των ελληνικών απορριμμάτων τα τελευταία χρόνια. Το ποσοστό ανακύκλωσης υπερδιπλασιάστηκε την περίοδο 2001–2007, από 8,8 % σε 20 %. Ωστόσο, αυτή η αυξητική τάση δεν συνεχίστηκε και το συνολικό ποσοστό ανακύκλωσης παρέμεινε στάσιμο στο 17-19%, όπως αυτό διαγράφεται και στις ακόλουθες εικόνες (Εικόνα 4). Από το 2007, η Ελλάδα δεν έχει εντείνει τις προσπάθειές της για την ανακύκλωση, επομένως θα χρειαστεί να καταβάλει εξαιρετική προσπάθεια ως προς την επίτευξη των στόχων που έχει θέσει η Ευρωπαϊκή Ένωση



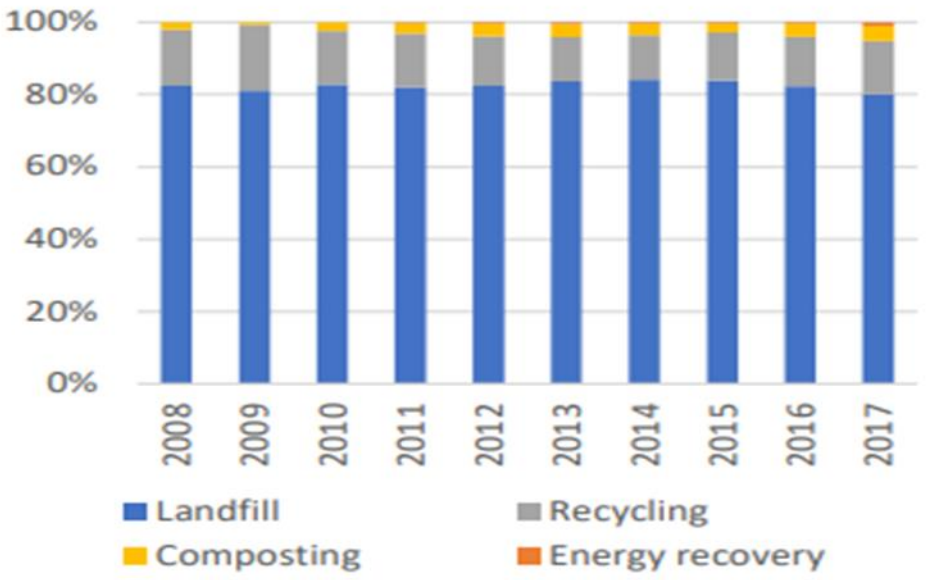
Εικόνα 4 Ρυθμός ανακύκλωσης στην Ελλάδα την περίοδο 2009-2017

Τα τελευταία 30 χρόνια τα αστικά κέντρα στην Ελλάδα έχουν επεκταθεί και σε συνδυασμό με την έλλειψη κατάλληλου χωροταξικού σχεδιασμού, την αλλαγή των καταναλωτικών προτύπων και συνηθειών, την βελτίωση του βιοτικού επιπέδου οδηγηθήκαμε σε αύξηση των αστικών στερεών αποβλήτων. Αυτό συνεπάγεται καταστροφικές περιβαλλοντικές συνέπειες όπως η αύξηση της ατμοσφαιρικής ρύπανσης (Μάτση, 2021). Στην Ελλάδα ο μέσος όρος απορριμμάτων ανά άτομο κατά τη διάρκεια ενός έτους ήταν 441kg το 2004 με αυξανόμενο ρυθμό σύμφωνά με τον ΕΣΥΕ (2014). Αυτή η αυξητική διάθεση συνεχίστηκε όπως επιβεβαιώνεται και απο το παρακάτω διάγραμμα (Εικόνα 2). Η μείωση που παρατηρείται μετά το 2010 και πιο συγκεκριμένα την περίοδο 2011-2013 οφείλεται στην οικονομική επιβράδυνση που υπέστει η χώρα εξαιτίας της οικονομικής κρίσης που αντιμετώπιζε. Επίσης παρατηρείται το γεγονός ότι στην Ελλάδα ένα μέσο άτομο παράγει περισσότερα αστικά στερεά απόβλητα από τον μέσο όρο της ΕΕ (Fabio M. Lamberti, 2020).

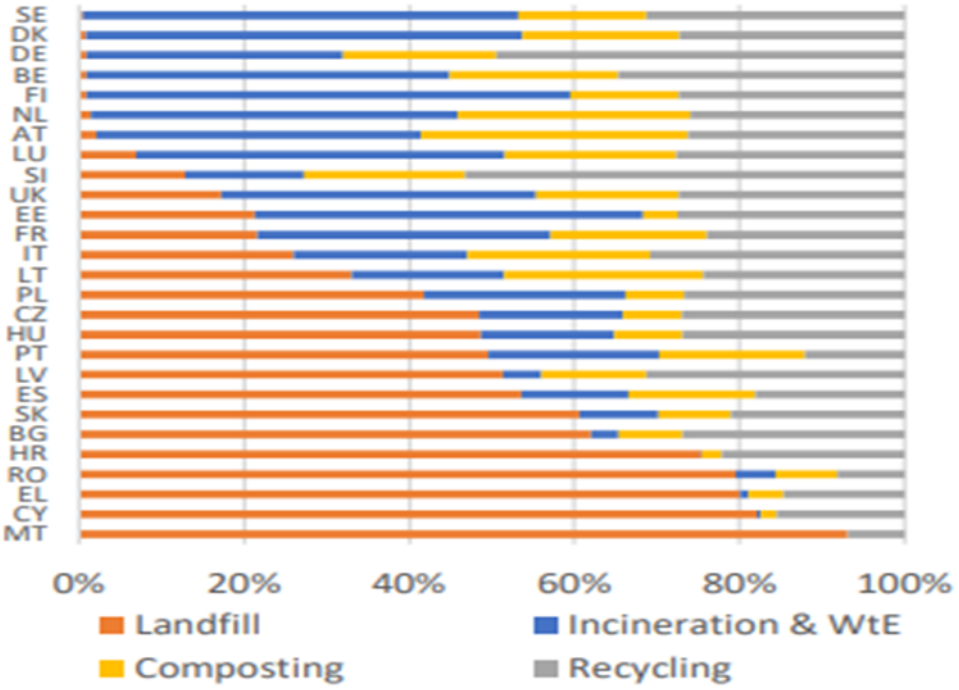


Εικόνα 5 Κατά κεφαλήν παραγωγή ΑΣΑ, Ελλάδα και 28 μέλη ΕΕ

Στην Ελλάδα το 80% των αστικών στερεών αποβλήτων (ΑΣΑ) διατίθεται σε χώρους υγειονομικής ταφής (ΧΥΤΑ) όπως μπορεί να διακριθεί από το παρακάτω διάγραμμα (Εικόνα 3). Στα νησιά αυτή η τάση παρατηρείται να συμβαίνει με πιο ανεξέλεγκτους ρυθμούς (Drimili et al., 2020). Η Ελλάδα βρίσκεται ακόμη αρκετά πίσω όσον αφορά την ολοκληρωμένη διαχείριση απορριμμάτων σε σύγκριση με τα υπόλοιπα 27 κράτη μέλη της ΕΕ. Περίπου το 15% των συλλεγόμενων ΑΣΑ ανακυκλώνεται, ενώ το 4,1% κομποστοποιείται. Γεγονός που κατατάσσει την Ελλάδα πολύ χαμηλά στους δείκτες της κυκλικής οικονομίας ανάμεσα στις 27 χώρες (ΙΟΒΕ, 2020). Παρόλο που τα ποσοστά ανακύκλωσης αυξάνονται τα τελευταία χρόνια, η Ελλάδα έχει το 4ο χαμηλότερο ποσοστό ανακύκλωσης μεταξύ των χωρών της ΕΕ, μπροστά από την Πορτογαλία, τη Ρουμανία και τη Μάλτα όπως φαίνεται και παρακάτω στην Εικόνα 3. Επιπροσθέτως, λιγότερο από το 1,1% των συλλεγόμενων ΑΣΑ χρησιμοποιείται για την ανάκτηση ενέργειας (Research, 2020)



Εικόνα 6 . Διαχείρηση Στερεών Αστικών Αποβλήτων στην Ελλάδα για την περίοδο 2008 – 2017 (ΙΟΒΕ, 2020).



Εικόνα 7 Σύγκριση Ελλάδας με τις υπόλοιπες χώρες της ΕΕ για το 2017.

Τέλος, σύμφωνα με την ευρωπαϊκή πολιτική, η διάθεση στη γη (π.χ. ΧΥΤΑ) είναι η λιγότερο προτιμώμενη διαδικασία διαχείρισης απορριμμάτων και δεν εναρμονίζεται με το μοντέλο της κυκλικής οικονομίας. Υλικά που θα μπορούσαν να επανεισαχθούν στην αλυσίδα αξίας και παραγωγής διατίθενται στο φυσικό περιβάλλον, προκαλώντας περιβαλλοντική υποβάθμιση και αυξάνοντας την εξάρτηση της οικονομίας από τους φυσικούς πόρους (ΙΟΒΕ, 2020). Η Ευρωπαϊκή Ένωση έχει τοποθετήσει κάποιους στόχους ανακύκλωσης ώστε να περιοριστεί η εξάρτηση χρήσης των χώρων υγειονομικής ταφής για την διάθεση των απορριμμάτων. Αυτοί οι στόχοι τέθηκαν το 2018 (με ορίζοντα 10-15 έτη) και θα μπορούσαν να χαρακτηριστούν αρκετά φιλόδοξοι αλλά και αυστηροί. Αυτοί πρέπει να εφαρμοστούν

από όλα τα Κράτη-Μέλη της, συμπεριλαμβάνοντας μέσα σε αυτές και την Ελλάδα. Η Ελλάδα, είναι αντιμέτωπη με μία σημαντική πρόκληση, λαμβάνοντας υπόψιν τα στοιχεία των προηγούμενων εικόνων. Η Ελλάδα θα πρέπει να επιτύχει ρυθμούς ανακύκλωσης 50% μέχρι το 2025 (μετά και την πενταετή παράταση που εξασφάλισε η χώρα) και 65% μέχρι το 2035 (ΥΠΕΝ, 2020). Ωστόσο, η Ελλάδα βρίσκεται μακριά απο την επίτευξη αυτών των στόχων (Καθημερινή, 2019). Σύμφωνα με την Εικόνα 1, ο ρυθμός ανακύκλωσης στην Ελλάδα το 2016 ήταν 17 % (με το 14 % να ανακυκλώνεται και το 3 % να κομποστοποιείται), σε σύγκριση με τον μέσο όρο της ΕΕ που είναι 45 %. Ορισμένες χώρες, όπως η Γερμανία, η Σλοβενία και η Αυστρία, είχαν ποσοστό ανακύκλωσης άνω του 50%. Στον Ελλαδικό χώρο και πιο συγκεκριμένα στην περιοχή της Αττικής, μεγάλη εντύπωση προκαλεί το γεγονός πως εώς και το 95% των απορριμμάτων της καταλήγει στη Φυλή, το 90,5% των οποίων θάβεται και μόνο το 7,5% ανακτάται. Ωστόσο, μόνο το 5 τοις εκατό όλων των απορριμμάτων θα πρέπει να θάβονται (Καθημερινή, 2019). Επίσης το γεγονός οτι το 38% των συνολικών αστικών απορριμμάτων της χώρας παράγεται στην Αττική αναδεικνύει στην σημαντικότητα του προβλήματος (Drimili, Herrero-Martin, Suardiaz-Muro, & Zervas, 2020)

1.7 Κυκλική οικονομία

Η κυκλική οικονομία είναι ένας οικονομικός και βιομηχανικός προσανατολισμός που στοχεύει στη μείωση της χρήσης πρώτων υλών, τη μείωση των αποβλήτων και την αξιοποίηση των πόρων μέσω της αναπαραγωγής και της ανακύκλωσης των υλικών. Στόχος είναι η δημιουργία ενός κλειστού κύκλου όπου οι πόροι παραμένουν σε χρήση για όσο το δυνατόν περισσότερο χρόνο. Στην κυκλική οικονομία, η παραγωγή, η κατανάλωση και η απόρριψη των προϊόντων γίνονται με τρόπο που προάγει την ανακύκλωση, την επαναχρησιμοποίηση και την ανανέωση των πόρων. Η κυκλική οικονομία βασίζεται σε τέσσερις βασικές αρχές:

1)Ανακύκλωση και αναπαραγωγή: Η χρήση των υλικών προωθείται με τέτοιο τρόπο ώστε να είναι δυνατή η ανακύκλωση και η επαναχρησιμοποίησή τους στο τέλος του κύκλου ζωής τους.

2) Σχεδιασμός για ανακύκλωση: Τα προϊόντα σχεδιάζονται με τέτοιο τρόπο ώστε να είναι εύκολη η ανακύκλωσή τους, με αποσπώμενα μέρη και υλικά που μπορούν να χωριστούν και να τα επεξεργαστούν ξεχωριστά.

3) Αναδιάρθρωση των επιχειρήσεων: Οι επιχειρήσεις αναδιοργανώνονται ώστε να προάγουν την ανακύκλωση και την επαναχρησιμοποίηση των προϊόντων και των υλικών τους.

4) Προώθηση της βιώσιμης κατανάλωσης: Οι καταναλωτές ενθαρρύνονται να επιλέγουν προϊόντα με βάση την περιβαλλοντική τους επίπτωση και να προτιμούν την επαναχρησιμοποίηση και την ανακύκλωση.

(Foundation, 2012)

1.8 ΤΟ ΝΕΟ ΕΘΝΙΚΟ ΣΧΕΔΙΟ ΔΙΑΧΕΙΡΙΣΕΙΣ ΑΠΟΒΛΗΤΩΝ (2020-2030)

1) Θέτει σε κύριο στόχο την μείωση της υγειονομικής ταφής των Αστικών Στερεών Αποβλήτων (ΑΣΑ), που είναι η κατώτερη βαθμίδα διαχείρισης στην πυραμίδα ιεράρχησης των αποβλήτων σε ποσοστό μικρότερο του 10% το έτος 2030, φέρνοντας το συγκεκριμένο στόχο πέντε χρόνια νωρίτερα από τις αντίστοιχες ευρωπαϊκές διατάξεις, οι οποίες προβλέπουν μέγιστο ποσοστό ταφής 10% το έτος 2030.

2) Έχει σαν στόχο να εφαρμόσει την ‘περιβαλλοντικής εισφορά’’ (τέλος ταφής) για να αποτρέψει την ταφή για τα απόβλητα που οδηγούνται στα ΧΥΤΑ (Χώρους Υγειονομικής Ταφής Απορριμμάτων), αλλά επίσης και στα απόβλητα που οδηγούνται στα ΧΥΤΥ (Χώρους Υγειονομικής Ταφής Υπολειμμάτων).
3) Προτείνει μια προσέγγιση για να ευαισθητοποιήσει και να ενημερώσει τους πολίτες σχετικά με την περιβαλλοντική και οικονομική σημασία της ανακύκλωσής και της εκτροπής των αποβλήτων από την ταφή, με απώτερο σκοπό την υιοθέτηση των αρχών της κυκλικής οικονομίας. Ιδιαίτερη έμφαση θα δοθεί σε ειδικές καμπάνιες απευθυνόμενες στις μικρές ηλικίες, όπως στα σχολεία.
4) Προωθεί τον καφέ κάδο που αφορά τα οργανικά απόβλητα και τα βιοαπόβλητα και τους μπλε κάδους που αφορά την ανακύκλωση. Σκοπός είναι να επιτευχθεί μια σημαντική αύξηση πάνω στα ποσοστά της ανακύκλωσης, η συλλογή των ανακυκλώσιμων υλικών να είναι υψηλής ποιότητας και καθαρότητας και μαζί με αυτό να επιτευχθεί μια μεγάλη σε βαθμό μείωση των αποβλήτων που οδηγούνται προς επεξεργασία σε μονάδες υπολειμμάτων σύμμεικτων Αστικών Στερεών Αποβλήτων και κατ’ επέκταση σε ταφή.
5) Ενισχύει τη Διαλογή στην Πηγή (ΔοΠ), προβλέποντας ταυτόχρονα σχέδιο για την ανάπτυξη νέων και την ενίσχυση των υφιστάμενων δικτύων συλλογής των ανακυκλώσιμων υλικών και βιοαποβλήτων.

(Έγκριση Εθνικού Σχεδίου Διαχείρισης Επικίνδυνων Αποβλήτων (ΕΣΔΕΑ), 2016)

ΚΕΦΑΛΑΙΟ 2

2.1 Ιστορία και εξέλιξη ηλεκτρονικών παιχνιδιών

Η ιστορία των ηλεκτρονικών παιχνιδιών ξεκινά στα μέσα του 20ο αιώνα, όταν οι πρώτοι υπολογιστές άρχισαν να εμφανίζονται. Το 1950, ο Christopher Strachey δημιούργησε ένα από τα πρώτα προγράμματα παιχνιδιού για τον υπολογιστή του Manchester Mark 1, το οποίο ονομάστηκε "Tic-Tac-Toe" (Γκρινιάρης). Από εκεί και πέρα, η εξέλιξη των ηλεκτρονικών παιχνιδιών ήταν συνεχής και ταχύτατη. Τη δεκαετία του 1970, η Atari κατέκτησε την αγορά με επιτυχημένα παιχνίδια όπως το "Pong", που ήταν ένα από τα πρώτα εμπορικά επιτυχημένα ηλεκτρονικά παιχνίδια. Αργότερα, το 1977, η Atari κυκλοφόρησε το Atari 2600, μία από τις πρώτες οικιακές κονσόλες παιχνιδιών που επέτρεπε στους χρήστες να απολαμβάνουν παιχνίδια στην άνεση του σπιτιού τους. Η δεκαετία του 1980 είχε την εμφάνιση πολλών σημαντικών πλατφόρμων παιχνιδιών, όπως οι προσωπικοί υπολογιστές Commodore 64 και Apple II. Επίσης, η Nintendo κατέκτησε την αγορά με το Nintendo Entertainment System (NES), το οποίο γνώρισε μεγάλη επιτυχία με παιχνίδια όπως το "Super Mario Bros." και το "The Legend of Zelda".Η επόμενη μεγάλη επανάσταση στην ιστορία των ηλεκτρονικών παιχνιδιών ήρθε με την εισαγωγή των 3D γραφικών. Το 1992, η id Software κυκλοφόρησε το "Wolfenstein 3D", ένα από τα πρώτα παιχνίδια πρώτου προσώπου που χρησιμοποιούσε την τεχνολογία 3D γραφικών. Ακολούθησε το "Doom" το 1993, το οποίο έγινε παγκόσμιο φαινόμενο. Στη συνέχεια, η εξέλιξη των ηλεκτρονικών παιχνιδιών περιλάμβανε πολλές καινοτομίες, όπως η κυκλοφορία των CD-ROMs, των φορητών κονσολών, όπως η Game Boy της Nintendo, και την εμφάνιση των online παιχνιδιών.

(Newman, 2012)

2.2 Χαρακτηριστικά των ηλεκτρονικών παιχνιδιών

Πλέον στη καθημερινότητα του κόσμου και ειδικά στις νέες ηλικίες η ενασχόληση με τα ηλεκτρονικά παιχνίδια όλο και αυξάνεται, η ραγδαία εξέλιξη της τεχνολογίας τα τελευταία χρόνια έχει ως αποτέλεσμα και την ανάπτυξη στα ηλεκτρονικά παιχνίδια. Πλέον ένα τεχνικό περιβάλλον παρέχει στον χρήστη ένα πλήθος από επιλογές και όχι μόνο σε θέμα ψυχαγωγίας αλλά και σε θέμα εκπαίδευσης, με κύριο χαρακτηριστικό τη συμμετοχή του παίκτη. Συγκεκριμένα το κάθε παιχνίδι απαρτίζεται από κάποια βασικά χαρακτηριστικά που αφορούν την δομή τους, τα οποία είναι τα εξής:

1. Διαδραστικότητα: Τα ηλεκτρονικά παιχνίδια προσφέρουν τη δυνατότητα στους παίκτες να αλληλοεπιδρούν με τον εικονικό κόσμο του παιχνιδιού και να πραγματοποιούν ενέργειες που επηρεάζουν την πρόοδο και την ανάπτυξη της ιστορίας.
2. Αισθητική: Τα ηλεκτρονικά παιχνίδια χαρακτηρίζονται από προηγμένα γραφικά, ήχο και εφέ που δημιουργούν εκθαμβωτικές ατμόσφαιρες και εμβαθύνουν την εμπειρία του παίκτη.

3. Προκλήσεις και επίπεδα δυσκολίας: Τα παιχνίδια προσφέρουν διάφορα επίπεδα δυσκολίας και προκλήσεις που απαιτούν από τους παίκτες να χρησιμοποιήσουν τις δεξιότητές τους και να αναπτύξουν στρατηγικές για να προχωρήσουν στο παιχνίδι.

4. Κοινωνική διάσταση: Πολλά ηλεκτρονικά παιχνίδια προσφέρουν λειτουργίες πολλαπλών παικτών, που επιτρέπουν στους παίκτες να αλληλοεπιδρούν και να συνεργάζονται με άλλους παίκτες από διάφορες τοποθεσίες.

5. Πρόοδος και επιβράβευση: Τα ηλεκτρονικά παιχνίδια προσφέρουν συστήματα προόδου και επιβράβευσης, με στόχους, επιτεύγματα, αναβαθμίσεις και αναγνώριση που ενθαρρύνουν την συνέχιση και την ανάπτυξη των παικτών.

6. Κανόνες: Σημαντική προϋπόθεση για την σωστή και ομαλή λειτουργία των παιχνιδιών είναι η θέσπιση των κανόνων , οι οποίοι χαρακτηρίζονται ως η ψυχή και η καρδιά του παιχνιδιού, δηλαδή δεν γίνεται να υφίσταται ένα παιχνίδι χωρίς κανόνες.

(Newman, 2012)

2.3 Κατηγορίες ηλεκτρονικών παιχνιδιών

Τα ηλεκτρονικά παιχνίδια μπορούν γενικότερα να κατηγοριοποιούνται σε 7 είδη τα οποία πολλές φορές συνδυάζονται μεταξύ τους και είναι δύσκολο να γίνει ο διαχωρισμός τους . Οι κατηγορίες των ηλεκτρονικών παιχνιδιών είναι:

1. Δράση (Action): Παιχνίδια που επικεντρώνονται στην ενέργεια, τη μάχη και την περιπέτεια. Παραδείγματα περιλαμβάνουν τα παιχνίδια της σειράς "Call of Duty" και "Assassin's Creed".

2. Περιπέτεια (Adventure): Παιχνίδια που έχουν έμφαση στην εξερεύνηση, την ανακάλυψη και την αφήγηση ιστορίας. Παραδείγματα περιλαμβάνουν τα παιχνίδια της σειράς "The Legend of Zelda" και "Uncharted".

3. Στρατηγική (Strategy): Παιχνίδια που απαιτούν σκέψη, σχεδιασμό και λήψη αποφάσεων για την επίτευξη στόχων. Παραδείγματα περιλαμβάνουν τα παιχνίδια της σειράς "Civilization" και "StarCraft".

4. Παζλ (Puzzle): Παιχνίδια που απαιτούν λογική σκέψη, ανακρίβεια και επίλυση γρίφων. Παραδείγματα περιλαμβάνουν τα παιχνίδια "Tetris" και "Portal".

5. Αθλητικά (Sports): Παιχνίδια που αναπαριστούν διάφορα αθλήματα και αθλητικές δραστηριότητες. Παραδείγματα περιλαμβάνουν τα παιχνίδια της σειράς "FIFA" και "NBA 2K".

6. Ρόλων (Role-Playing): Παιχνίδια όπου οι παίκτες υποδύονται έναν χαρακτήρα και εξερευνούν φανταστικούς κόσμους, εκτελώντας καθήκοντα και αναπτύσσοντας τις ικανότητές τους. Παραδείγματα περιλαμβάνουν τα παιχνίδια της σειράς "The Elder Scrolls" και "Final Fantasy".

7. Παιχνίδια Επιτραπέζιας Ρόλων (Tabletop Role-Playing Games): Παιχνίδια όπου οι παίκτες υποδύονται χαρακτήρες και αλληλεπιδρούν με έναν φανταστικό κόσμο μέσω συνομιλιών και ρίψεων ζαριών. Παράδειγμα είναι το παιχνίδι "Dungeons & Dragons"

(Salen, 2004)

2.3.1 Πλεονεκτήματα και μειονεκτήματα των ηλεκτρονικών παιχνιδιών

Τα ηλεκτρονικά παιχνίδια έχουν πολλά πλεονεκτήματα που τα καθιστούν αξιόλογη μορφή ψυχαγωγίας και μάθησης. Ένα από τα πλεονεκτήματα των ηλεκτρονικών παιχνιδιών είναι η διασκέδαση - ψυχαγωγία. Τα ηλεκτρονικά παιχνίδια προσφέρουν μια μοναδική εμπειρία διασκέδασης, που επιτρέπει στους παίκτες να απολαύσουν τη στιγμή και να χαλαρώσουν. Από τα απλά παιχνίδια δράσης μέχρι τα πολυπλοκότερα παιχνίδια, υπάρχει κάτι για κάθε γούστο. Ένα ακόμα σημαντικό πλεονέκτημα των παιχνιδιών είναι τα εκπαιδευτικού τύπου, πολλά ηλεκτρονικά παιχνίδια έχουν ενσωματωμένες εκπαιδευτικές συνιστώσες, που βοηθούν στην ανάπτυξη δεξιοτήτων και γνώσεων. Αυτά τα παιχνίδια μπορούν να ενθαρρύνουν την εκμάθηση νέων γλωσσών, μαθηματικών, επιστημών και άλλων ακαδημαϊκών θεμάτων. Επίσης μπορούννα προσφέρουν ψυχαγωγία και ανακούφιση από το άγχος, τα ηλεκτρονικά παιχνίδια έχουν αυτή την δυνατότητα να μπορούν να λειτουργήσουν ως μέσο χαλάρωσης και διασκέδασης. Ακόμα να βοηθήσει τον χρήστη να ανάπτυξη τις κοινωνικές του δεξιότητες, ορισμένα παιχνίδια παρέχουν τη δυνατότητα συνεργασίας και αλληλεπίδρασης με άλλους παίκτες. Αυτό ενθαρρύνει την ανάπτυξη κοινωνικών δεξιοτήτων, όπως η επικοινωνία, η ανταλλαγή απόψεων και η συνεργασία. Αυτή η αλληλεπίδραση με άλλους παίχτες από όλο τον κόσμο προωθεί τη δημιουργία φιλίας, τη συνεργασία και την ανταλλαγή απόψεων. Τέλος τα ηλεκτρονικά παιχνίδια μπορούν να φανούν χρήσιμα και στην ανάπτυξη διάφορων δεξιοτήτων, μερικά παιχνίδια απαιτούν στρατηγική σκέψη, λήψη αποφάσεων, γρήγορα αντανακλαστικά και συγκέντρωση (Gee, 2003) . Όμως υπάρχουν και κάποια μειονεκτήματα που πρέπει να ληφθούν υπόψη. Όπως είναι ο κίνδυνος εθισμού, ορισμένα ηλεκτρονικά παιχνίδια μπορεί να είναι τόσο εθιστικά που οι παίκτες να δυσκολεύονται να τα εγκαταλείψουν. Ο εθισμός σε παιχνίδια μπορεί να οδηγήσει σε απομόνωση, προβλήματα υγείας και επιδείνωση των κοινωνικών σχέσεων. Περιορισμένη φυσική δραστηριότητα, οι παίκτες θα περάσουν πολύ χρόνο μπροστά σε οθόνες, περιορίζοντας τη φυσική τους δραστηριότητα. Αυτό μπορεί να οδηγήσει σε προβλήματα υγείας, όπως αύξηση του βάρους, κακή στάση σώματος και αδυναμία αντιμετώπισης του στρες. Ακόμα υπερβολική χρήση χρόνου, η εθιστική φύση των ηλεκτρονικών παιχνιδιών μπορεί να οδηγήσει σε υπερβολική χρήση χρόνου. Οι παίκτες μπορεί να παραμελήσουν τις καθημερινές τους υποχρεώσεις και κοινωνικές σχέσεις, καθώς δαπανούν πολλές ώρες παίζοντας παιχνίδια. (Anderson, 2000)

2.3.2 Ταξινόμηση των απολαύσεων των παιχνιδιών

Τα παιχνίδια προσφέρουν απολαύσεις σε διάφορες κατηγορίες:

1) Αίσθηση: Προκαλούν αισθήσεις όπως η οπτική ευχαρίστηση από όμορφα γραφικά, η ακουστική απόλαυση από μουσική, το αίσθημα απαλότητας μέσω της αφής ή ακόμα και η ευωδιά από νόστιμο φαγητό. Η αισθητική του παιχνιδιού είναι υπεύθυνη για αυτές τις απολαύσεις.
2) Φαντασία: Προσφέρουν απόλαυση μέσω της εξερεύνησης φανταστικών κόσμων και φαντασίωση του παίκτη για κάτι που δεν είναι στην πραγματική ζωή.
3) Αφήγηση: Προσφέρουν απόλαυση μέσω της αποκάλυψης μιας ενδιαφέρουσας ιστορίας ή πλοκής, στην οποία ο παίκτης μπορεί να είναι εμπλεκόμενος ή απλά να την παρακολουθεί.
4) Πρόκληση: Προσφέρουν απόλαυση μέσω της πρόκλησης που παρουσιάζουν. Κάθε παιχνίδι παρουσιάζει στον παίκτη ένα πρόβλημα ή μια πρόκληση που πρέπει να επιλύσει.

5) Συντροφικότητα: Προσφέρουν απολαύσεις που πηγάζουν από τη φιλία, τη συνεργασία και την κοινωνική αλληλεπίδραση με άλλους παίκτες.
6) Ανακάλυψη: Προσφέρουν απόλαυση μέσω της ανακάλυψης και της εξερεύνησης νέων πτυχών του παιχνιδιού.
7) Έκφραση: Αφορά την απόλαυση που πηγάζει από τη διαδικασία της προσωπικής έκφρασης. Τα σύγχρονα παιχνίδια συνήθως επιτρέπουν ένα βαθμό εξατομίκευσης, όπως τον σχεδιασμό του χαρακτήρα του παίκτη. (Anderson, 2000)

2.4 Τι είναι τα Serious games

Το 2003 μια ομάδα από ειδικούς στο χώρο του video gaming δημιούργησαν το "Serious Games Foundation" με στόχο την δημιουργία Σοβαρών Παιχνιδιών το ίδιο έτος η Woodrow Wilson Center for International Scholar στην Washington, D.C. ίδρυσε τη Serious Games Initiative και καθιερώθηκε ο όρος «σοβαρά παιχνίδια » (Susi κ.ά., 2007). Ένα serious game (‘σοβαρό’ παιχνίδι) είναι ένα παιχνίδι σχεδιασμένο για να μάθει και να εκπαιδευτεί ο χρήστης σε μια συγκεκριμένη έννοια ή δεξιότητα δηλαδή πέρα από την ψυχαγωγία έχουν και άλλο σκοπό. Τα Serious games σύμφωνα με τους Michael και Chen (2005) ταξινομούνται σε εκπαιδευτικά, στην υγειονομική περίθαλψη, στο μάρκετινγκ, σε στρατιωτικά, κυβερνητικά, πολιτικά, θρησκευτικά, καλλιτεχνικά, σε επιχειρήσεις και βιομηχανίες. Το να συνδυάσεις λέξεις όπως serious-games δηλαδή σοβαρό με παιχνίδι, μπορεί να φαίνεται αδύνατον αλλά δεν είναι. Τα τελευταία χρόνια είναι αποδεδειγμένο ότι μέσω των Serious games μπορείς να μαθαίνεις ενώ παίζεις. Αυτή η μέθοδος διδασκαλίας είναι γνωστή ως μάθηση με βάση το παιχνίδι, μια τάση που τα τελευταία χρόνια τα σχολεία και οι εταιρείες έχουν δείξει μεγάλη προθυμία να χρησιμοποιούν. Και τα σημερινά παιχνίδια έχουν γραφικά και τεχνολογικές εξελίξεις εφάμιλλα με τα πιο δημοφιλή ψυχαγωγικά βιντεοπαιχνίδια. Αν και τα serious games μπορεί να είναι διασκεδαστικά, ο βασικός τους στόχος είναι να εκπαιδεύσουν τους χρήστες, για αυτό τον λόγο θα θυσιάσουν τη διασκέδαση προς όφελος μιας πιο επιτυχημένης προόδου του χρήστη. Μερικές βασικές διαφορές των Serious games με τα καθαρά ψυχαγωγικού τύπου παιχνίδια είναι (Mitchell, 2004)

Δραστηριότητα vs πλούσια εμπειρία	Στο επίκεντρο η επίλυση προβλημάτων	Προτιμάται η πλούσια εμπειρία
Εστίαση σε…	Σημαντικά στοιχεία μάθησης	Διασκέδαση
Προσομοιώσεις	Παραδοχές απαραίτητες για λειτουργικές προσομοιώσεις	Απλοποιημένες διεργασίες προσομοίωσης

Τα serious-games χωρίζονται σε 2 κατηγορίες σε δύο διαστάσεων (2D) και σε τρισδιάστατη (3D) (Informatics Department, Universitas Muhammadiyah Surakarta, 2018) :

2D πλατφόρμα παιχνιδιών: Η δημιουργία παιχνιδιών σε 2D είναι πολύ δημοφιλής στους προγραμματιστές για τον λόγω της ευκολίας του προγραμματισμού τους. Επίσης τα δισδιάστατα παιχνίδια απαιτούν μικρότερη μνήμη από αυτές του τρισδιάστατου παιχνιδιού

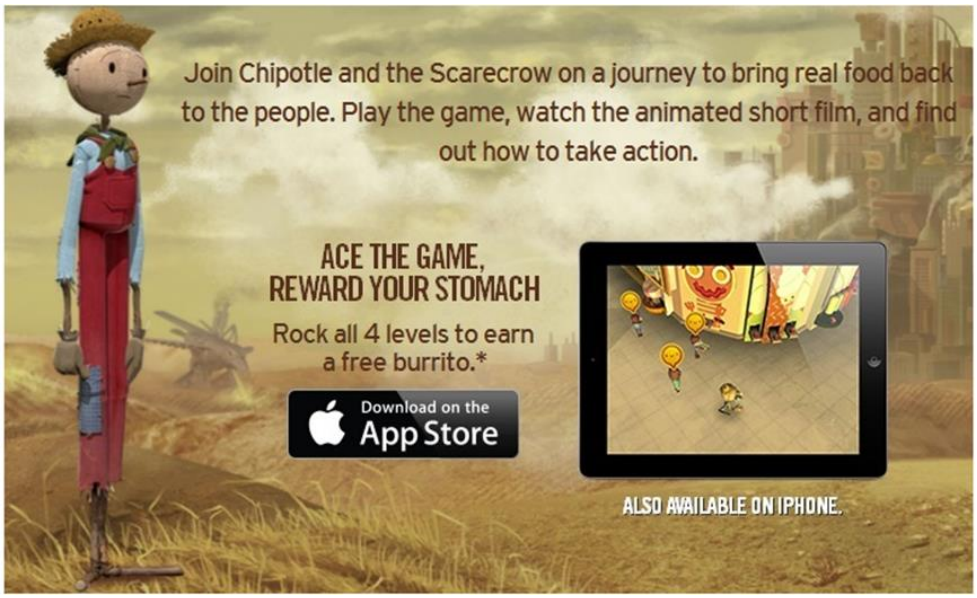
3D πλατφόρμα παιχνιδιών: Οπτικά τα τρισδιάστατα παιχνίδια είναι πιο δελεαστικά από αυτά των δισδιάστατων . Ο παίκτης έχει την αίσθηση ότι βρίσκεται σε πραγματικό περιβάλλον, αυτό όμως απαιτεί μεγαλύτερη χωρητικότητα μνήμης και δυσκολίας προγραμματισμού.

2.4.1 Θετικά και αρνητικά των Serious games, Entertainment Games

Τα σοβαρά παιχνίδια έχουν πολλά οφέλη που έχουν την τάση να αυξάνονται λόγω της μεγάλης ζήτησης και της εξέλιξης της τεχνολογίας . Τα Serious-games είναι διαδραστικά έτσι ώστε οι χρήστες να πρέπει να παίρνουν αποφάσεις από την αρχή, η δυναμική του παιχνιδιού ενθαρρύνει την προσπάθεια και δίνει κίνητρο. Τα πιο εξελιγμένα και καλοφτιαγμένα βιντεοπαιχνίδια μπορούν ακόμη και να μετατρέψουν τη μάθηση σε εθισμό (Mitgutsch 2009). Επίσης μέσω των serious-games υπάρχουν προσομοιωτές που αναπαράγουν την πραγματικότητα όσο το δυνατόν πιο πιστά , αυτό επιτρέπει στον χρήστη να εξασκηθεί με ασφάλεια χωρίς να διατρέχει κανένα κίνδυνο. Ακόμα προσανατολίζουν τους μαθητές στον επαγγελματικό τομέα μέσω της ενημέρωσης που προσφέρουν από μία πληθώρα επαγγελμάτων, διαφορετικούς εργασιακούς τύπους και διεξόδους επαγγελματικής σταδιοδρομίας , ανάλογα με τις δυνατότητες του κάθε μαθητή , τα ενδιαφέροντά του, τις δεξιότητές του , το μορφωτικό του επίπεδο και την οικονομική κατάσταση του καθενός. Επιπλέον τα Serious-games λειτουργούν ως εργαλεία παρακίνησης και διαρκής αλληλεπίδρασης με τον χρήστη (Corti, 2006). Τα Serious-games ενθαρρύνουν και δίνουν κίνητρο στους χρήστες μέσω του σχεδιασμού του παιχνιδιού που εξασφαλίζει ότι μαθαίνουν καθώς παίζουν. Τα παιχνίδια μπορούν να υποστηρίξουν την ανάπτυξη πολλών διαφορετικών δεξιοτήτων, σύμφωνα με τους Mitchell και Savill-Smith (2004), αναλυτικές και χωρικές δεξιότητες, στρατηγικές δεξιότητες και διορατικότητα, ικανότητες μάθησης και ανάμνησης, ψυχοκινητικές δεξιότητες, οπτική επιλεκτική προσοχή κλπ. Επιπλέον, οι DeLisi και Wolford (2002) αναφέρουν για το πώς μπορούν να βελτιωθούν οι ικανότητες του παίχτη παίζοντας παιχνίδια όπως το Tetris. Επιπλέον και άλλα οφέλη που μπορούμε να έχουμε μέσω των παιχνιδιών είναι το να αναγνωρίζουμε ένα πρόβλημα και να προσπαθούμε να το λύσουμε, λήψη αποφάσεων, καλύτερη βραχυπρόθεσμη και μακροπρόθεσμη μνήμη και αυξημένες κοινωνικές δεξιότητες όπως η συνεργασία, η διαπραγμάτευση και η κοινή λήψη αποφάσεων (ELSPA, 2006; Mitchell & Savill-Smith, 2004). Για όλους αυτούς τους λόγους καταλαβαίνουμε ότι τα Serious-games μπορούν να φανούν χρήσιμα εργαλεία για την μάθηση και την εκπαίδευση. Παρά τα πολλά οφέλη και τις δυνατότητες που προσφέρουν τα serious games, υπάρχουν και μειονεκτήματα που πρέπει να ληφθούν υπόψη. Κάποια από αυτά τα μειονεκτήματα είναι τα εξής: Έλλειψη αποδείξεων αποτελεσματικότητας, παρόλο που υπάρχουν αρκετές μελέτες που υποστηρίζουν την αποτελεσματικότητα των serious games, εξακολουθούν να υπάρχουν κενά στην επιστημονική βάση που υποστηρίζει τις ισχυριζόμενες θετικές επιπτώσεις τους. Κίνδυνος εξάρτησης και υπερβολικής χρήσης, όπως συμβαίνει με οποιοδήποτε ηλεκτρονικό παιχνίδι, τα serious games μπορούν να δημιουργήσουν κίνδυνο εξάρτησης ή να χρησιμοποιούνται υπερβολικά, ανεξέλεγκτα και με αρνητικές επιπτώσεις στην καθημερινή ζωή και τις κοινωνικές σχέσεις του ανθρώπου. Επίσης υπάρχει έλλειψη χρηματοδότησης για τα παιχνίδια που αφορούν τον εκπαιδευτικό τομέα και όχι τον ψυχαγωγικό, ακόμα μερικά από τα serious games απαιτούν κάποιες γνώσεις οι οποίες είναι εξειδικευμένες και αυτό τα καθιστά απρόσιτα στο ευρύ κοινό. Τέλος η ανάπτυξη ποιοτικών σοβαρών παιχνιδιών κοστίζει πολύ περισσότερο συγκριτικά με τα πιο απλά ηλεκτρονικά παιχνίδια ψυχαγωγίας. (Connolly T. M., 2012)

2.4.2 Παραδείγματα και Κατηγορίες Serious-games

Advergames: έχουν ως στόχο την διαφήμιση και προώθηση ενός προϊόντος ένα τέτοιο παιχνίδι είναι το Chipotle Scarecrow. Το 2013 η Chipotle και η Moonbot Studios παρήγαγαν ένα παιχνίδι περιπέτειας για κινητά σε στυλ arcade για την προώθηση της καμπάνιας Food with Integrity. Το μήνυμα που ήθελαν να περάσουν ήταν σχετικά με τα επεξεργασμένα τρόφιμα σε όλο τον κόσμο. Το παιχνίδι τοποθετήθηκε στις 15 κορυφαίες δωρεάν εφαρμογές iOS στο US iOS app store, έχοντας προκαλέσει το ενδιαφέρον 250.000 χρηστών μέσα σε τέσσερις ημέρες από την κυκλοφορία του.



Εικόνα 8 Chipotle Scarecrow

Educational entertainment –εκπαιδευτική ψυχαγωγία: έχουν ως στόχο την εκπαίδευση του χρήστη, τέτοιου είδους παιχνίδια απευθύνονται πολλές φορές σε παιδιά με στόχο να τους εκπαιδεύσουν και να τους μεταδώσουν γνώσεις μέσω μιας ευχάριστης διαδικασίας. Επίσης μπορούν να χρησιμοποιηθούν για τη διδασκαλία σχεδόν κάθε μαθήματος, συμπεριλαμβανομένων των θετικών επιστημών, της ιστορίας, των μαθηματικών, της γλώσσας και άλλων εκπαιδευτικών μαθημάτων. Το MentalUP με περισσότερες από 150+ ασκήσεις βοηθά στην ανάπτυξη της προσοχής, της μνήμης, της λογικής, των οπτικών και λεκτικών δεξιοτήτων προσφέρει επίσης ειδικές ασκήσεις σε παιδιά με μαθηματική και μουσική νοημοσύνη

Εικόνα 9 MentalUP

Το MentalUP δεν βοηθάει μόνο το μυαλό των παιδιών αλλά και το σώμα τους όπου διαθέτει περισσότερες από 240+ ασκήσεις φυσικής κατάστασης για κάθε τύπου σώματος.

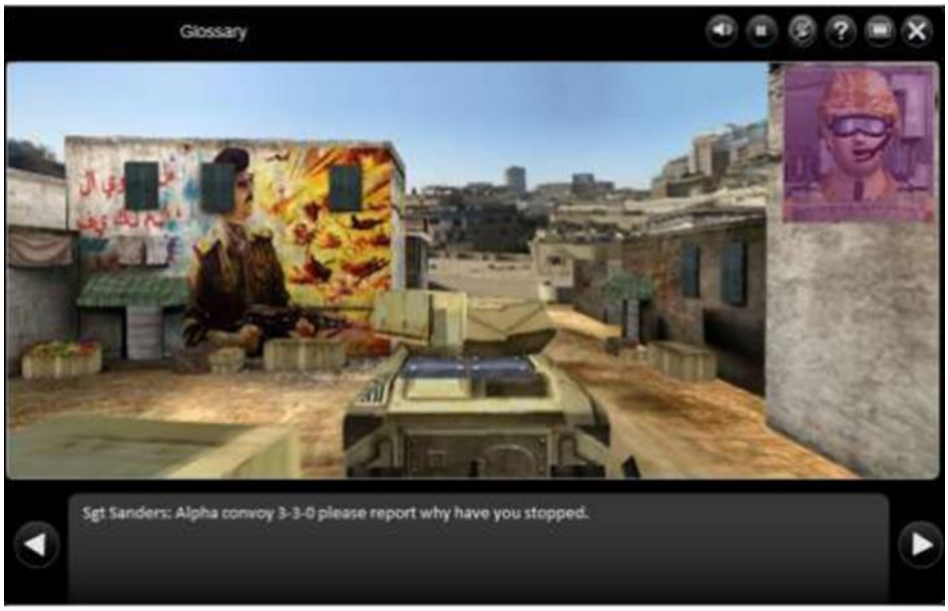
Training serious games: προσφέρουν ένα προσομοιωμένο σύστημα όπου οι χρήστες μπορούν να εξασκήσουν τις ικανότητές τους σε ένα ασφαλές περιβάλλον. Για παράδειγμα στο Pulse! το οποίο αναπαράγει τις συνθήκες μιας πτέρυγας έκτακτης ανάγκης σε ένα νοσοκομείο. Χάρη σε αυτό το βιντεοπαιχνίδι, οι μελλοντικοί νοσοκόμοι μπορούν να εξασκούν όλα όσα έχουν μάθει στα θεωρητικά τους μαθήματα και αποκτούν εμπειρία χειρισμού πραγματικών καταστάσεων

Εικόνα 10 Pulse!

Επίσης ένα άλλο παράδειγμα προσομοίωσης είναι το Train Simulator όπου οι χρήστες εξασκούνται στο να οδηγούν και να διαχειρίζονται τρένα

Εικόνα 11 Train Simulator

military games: Η στρατιωτική εκπαίδευση σε υπαρκτά πεδία μάχης είναι πολύ επικίνδυνες και δαπανηρές. Μέσω της συγκεκριμένης κατηγορίας serious-games ο στρατιώτης εκπαιδεύεται σε ένα ασφαλές περιβάλλον που πρέπει να διαχειριστεί δύσκολες και μεταβαλλόμενες καταστάσεις. Το Battlefield Awareness ο παίκτης είναι αρχιλόχειας σε ένα κομβόι και εκτιμώντας διάφορες καταστάσεις δίνει εντολές στους στρατιώτες. Οι πόντοι που συλλέγονται αντικατοπτρίζουν την διαχείριση της αποστολής.



Εικόνα 12 Battlefield Awareness

Corporate Serious Games: Η χρήση τους στον επιχειρησιακό τομέα αυξάνεται με σταθερό ρυθμό. Μεγάλες εταιρείες όπως Deloitte, IBM και Cisco χρησιμοποιούν τα serious games για εκπαίδευση, μάρκετινγκ και προσλήψεις και διακρίνουν θετικές επιδράσεις και στους τρεις επιχειρησιακούς τομείς. Το παιχνίδι Deloitte Business Simulation έχει σχεδιαστεί για να εκπαιδεύει τους υπαλλήλους στην εταιρική ευθύνη και βιωσιμότητα της εταιρίας. Το παιχνίδι δίνει τη δυνατότητα στους παίκτες να πειραματιστούν με ένα ρεαλιστικό μοντέλο της εταιρείας και με πιθανά μελλοντικά σενάρια. Κατά τη διάρκεια του παιχνιδιού οι παίκτες περνούν από διάφορες καταστάσεις και είναι αντιμέτωποι με τις συνέπειες των αποφάσεών τους όπως και γίνεται και στον πραγματικό κόσμο.

Εικόνα 13 Deloitte Business Simulation

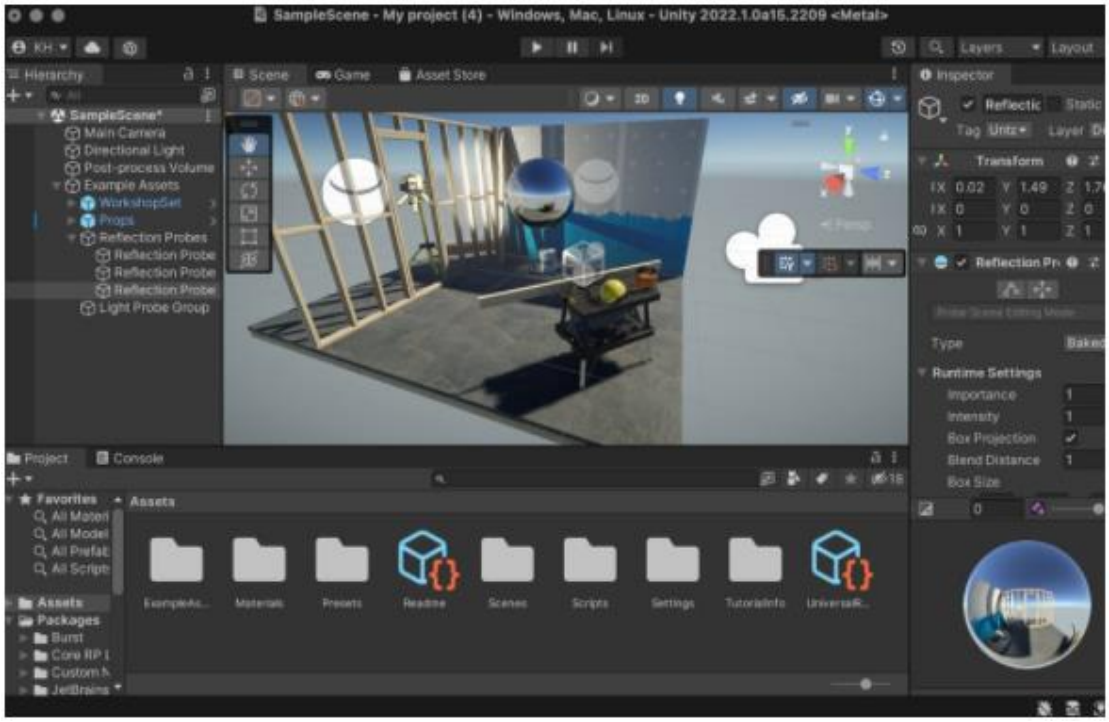
2.5 Παρουσίαση μηχανή παιχνιδιού – Game engines

Μια μηχανή παιχνιδιού είναι ένα framework πάνω στο οποίο δημιουργείται ένα παιχνίδι, Χρησιμοποιείται για να δημιουργηθούν παιχνίδια σε κονσόλες, φορητές συσκευές και υπολογιστές. Οι βασικές λειτουργίες που μπορεί να παρέχει μια μηχανή παιχνιδιού είναι: η απεικόνιση γραφικών (renderer), παραγωγή ήχου, μηχανή φυσικής/φυσικών νόμων (physics), animation, τεχνητή νοημοσύνη, ανίχνευσης σύγκρουσης και απόκρισης σύγκρουσης σενάρια(scripting), διαχείριση πόρων του υπολογιστή του χρήστη από το πρόγραμμα και προσαρμογή του λογισμικού του παιχνιδιού για διαφορετικά λειτουργικά συστήματα. Σήμερα οι εταιρείες χρησιμοποιούν τις μηχανές παιχνιδιών για να υπάρχει ευκολία στην μεταφορά παιχνιδιών σε πολλές πλατφόρμες, επίσης δεν απαιτούνται βαθιές γνώσεις προγραμματισμού και κάποιος ιδιαίτερος εξοπλισμός. Αυτό για τις εταιρίες είναι μείωση κόστους, ανθρώπινου δυναμικού και εξοικονόμηση χρόνου (Smith, 2014)

1)Unity

Εικόνα 14 Unity

Η μηχανή παιχνιδιού Unity αναπτύχθηκε από την Unity Technologies στη Δανία ανακοινώθηκε και κυκλοφόρησε για πρώτη φορά τον Ιούνιο του 2005 στο Παγκόσμιο Συνέδριο Προγραμματιστών της Apple, ως αποκλειστική μηχανή παιχνιδιών για Mac OS X, Η unity είναι η εταιρία με τους περισσότερους χρήστες στον τομέα των Game engines, μπορεί να χρησιμοποιηθεί για τη δημιουργία διαδικτυακών παιχνιδιών και παιχνιδιών εκτός σύνδεσης σε δισδιάστατη (2D) ,τρισδιάστατη(3D) VR μορφή. Υπάρχει μια ενεργή on-line κοινότητα προγραμματιστών που μπορεί που βοηθούν τους νέους χρήστες, ενώ λόγω της μεγάλης κοινότητας που έχει αναπτυχθεί μπορεί να βρεθεί η λύση σχεδόν σε κάθε πρόβλημα που θα συναντήσει κάποιος. Η ανάπτυξη μιας εφαρμογής στην Unity μπορεί να γίνει με γρήγορο ρυθμό ειδικά για mobile πλατφόρμες, ακόμα τα projects και η διαδικασία κατασκευής του παιχνιδού είναι απλή, για παράδειγμα το να εισάγει ο χρήστης ένα assert στο project του χρειάζεται απλά να το κάνει drag and drop και η Unity φροντίζει για τα υπόλοιπα. Επίσης η Unity έχει την δυνατότητα τα project της να τρέχουν σε όλες τις πλατφόρμες: κινητά, υπολογιστές, web και κονσόλες , με τον ίδιο κώδικα και και τα ίδια assert. Για τα κινητά η Unity τρέχει σε iOS, Android, BlackBerry και Windows Phone 8, ενώ για τους υπολογιστές σε Windows, Widows Store, Mac OS X και Linux.
Οι γλώσσες προγραμματισμού που μπορούν να χρησιμοποιηθούν είναι η Unity JavaScript, Boo και C#, ο προγραμματισμός σε C# είναι γρήγορος και αποτελεσματικός. Τέλος υπάρχει το κατάστημα asset store όπου είναι σχετικά φθηνό και περιέχει πολλά χρήσιμα asset. Το μόνο που υστερεί η Unity σε σύγκριση με τη Unreal και το Cry Engine είναι στα γραφικά (Haas, 2014)



Εικόνα 15 Unity interface

Το Interface του Unity αποτελεί το γραφικό περιβάλλον που χρησιμοποιείται για την ανάπτυξη παιχνιδιών. Παρέχει μια μεγάλη γκάμα απο εργαλεία και παράθυρα που επιτρέπει στους προγραμματιστές να δημιουργούν, να επεξεργάζονται και να διαχειρίζονται τα παιχνίδια τους. Ας αναλύσουμε μερικά από τα κύρια στοιχεία του Interface του Unity:

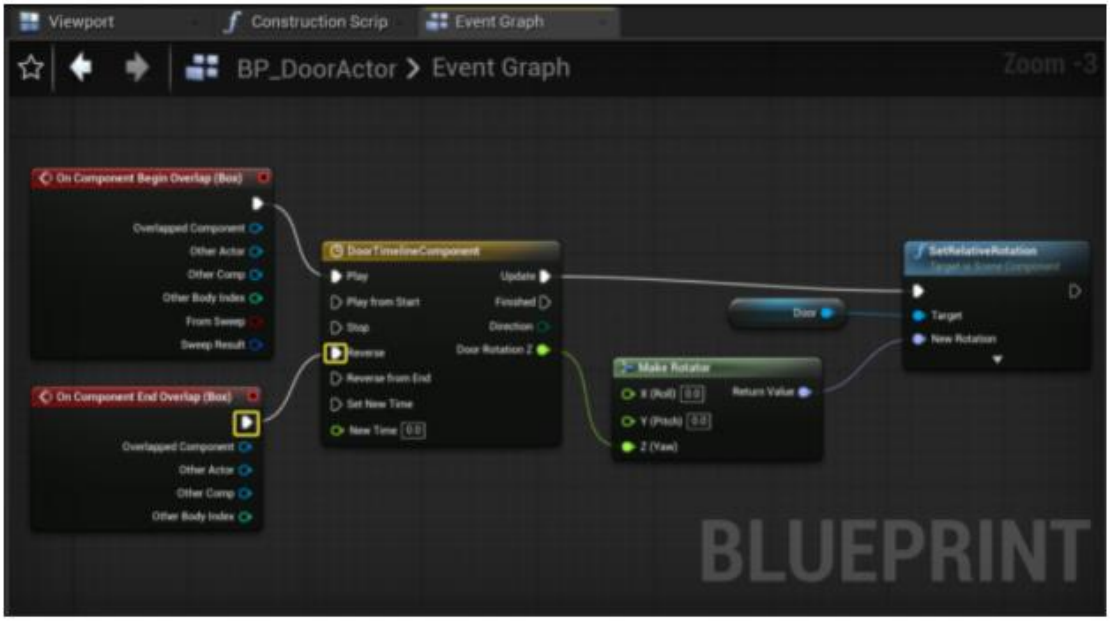
1) Hierarchy (Ιεραρχία): Αυτό το παράθυρο εμφανίζει μια ιεραρχική δομή των αντικειμένων στη σκηνή. Μπορείτε να δείτε και να οργανώσετε τα διάφορα αντικείμενα, όπως μοντέλα, φώτα, κάμερες και άλλα στοιχεία του παιχνιδιού σας.
2) Scene View (Προβολή Σκηνής): Αυτή η προβολή εμφανίζει την τρέχουσα σκηνή του παιχνιδιού σας. Μπορείτε να δείτε και να επεξεργαστείτε τα αντικείμενα της σκηνής, να τα τοποθετήσετε, να τα περιστρέψετε και να τα κλικάρετε για να τα επεξεργαστείτε.
3) Inspector (Επιθεωρητής): Αυτό το παράθυρο εμφανίζει τις λεπτομέρειες και τις ιδιότητες των επιλεγμένων αντικειμένων. Μπορείτε να αλλάξετε τις ρυθμίσεις τους, να προσθέσετε συμπεριφορές, να εφαρμόσετε φυσική σε αυτά και πολλά άλλα.
4) Project (Έργο): Αυτό το παράθυρο εμφανίζει το δέντρο των αρχείων και των φακέλων του έργου σας. Μπορείτε να οργανώσετε τους φακέλους, να φορτώσετε πόρους όπως εικόνες, ήχους, μοντέλα και να τα εισάγετε στη σκηνή σας.
5) Console (Κονσόλα): Αυτή η προβολή εμφανίζει τα μηνύματα και τις πληροφορίες από την εκτέλεση του παιχνιδιού σας. Μπορείτε να δείτε προειδοποιήσεις, σφάλματα και άλλες ενημερώσεις κατά τη διάρκεια της ανάπτυξης και της εκτέλεσης του παιχνιδιού.

Το Interface του Unity είναι σχεδιασμένο με σκοπό να παρέχει μια ολοκληρωμένη και ευέλικτη εμπειρία ανάπτυξης παιχνιδιών. Παρέχει έναν οργανωμένο και ευκολονόητο τρόπο για τη διαχείριση των αντικειμένων, την επεξεργασία τους και την ανάπτυξη της λειτουργικότητας του παιχνιδιού σας. (Haas, 2014)

2) Unreal Engine

Εικόνα 16 Unreal Engine

Η Unreal Engine (Epic Games, 1998) είναι μια από τις πιο γνωστές και αναγνωρισμένες μηχανές παιχνιδιών τα τελευταία χρόνια. Είναι πλήρες open-source και μπορεί εύκολα να τροποποιηθεί για έρευνα, έχει την ιδιότητα να παράγει ρεαλιστικά γραφικά, επιπλέον περιέχει τα κατάλληλα εργαλεία, που επικοινωνούν και με άλλα λογισμικά και τις κατάλληλες πληροφορίες για να μπορέσει ο χρήστης να δημιουργήσει έναν εικονικό κόσμο. Η Unreal Engine 4 είναι πολύ δημοφιλής για παιχνίδια VR αλλά και επειδή τα 3D γραφικά της είναι σε υψηλή ποιότητα μπορεί να χρησιμοποιηθεί και για αρχιτεκτονική απεικόνιση. Υποστιρίζει 2D και 3D γραφικά καθώς και τελευταίες τεχνολογίες, όπως PBR (Physically – ευκολότερη, καθώς απαιτεί λιγότερη παραμετροποίηση ώστε να δουλεύουν σωστά. Ο φωτισμός και οι σκιές της Unreal δείχνουν πολύ αληθοφανές με ελάχιστα στον αριθμό δημιουργήσει υλικά με ευκολία ακόμα και να τοποθετήσει διάφορα trigger στο χώρο τα οποία μπορούν να χρησιμοποιηθούν και ως οπτικά εφέ(W. Qiu and A. Yuille,Johns Hopkins κόμβων χωρίς να γίνει χρήση κώδικά) (Caron, 2008)



Εικόνα 17 'Unreal' Blueprints

3) CryEngine

Εικόνα 18 CryEngine

Η CryEngine είναι μια μηχανή παιχνιδιών που δημιουργήθηκε το 2002 και σχεδιάστηκε από τη γερμανική εταιρεία ανάπτυξης παιχνιδιών Crytek, το πρώτο παιχνίδι που δημιουργήθηκε ως ντέμο για να δείξει τις δυνατότητες που έχει η Nvidia ήταν το Far cry(Wikipedia, ακρίβεια το φωτισμό σε εσωτερικούς και εξωτερικούς χώρους και να είναι απίστευτα ρεαλιστικός , ενώ υποστηρίζονται και οι λεπτομέρειες της σκηνής. Η CryEngine υποστηρίζει μια σειρά από λειτουργίες που είναι χρήσιμες για να δημιουργηθούν ρεαλιστικά παιχνίδια, τα εργαλεία ανάπτυξης είναι ενσωματωμένα στο ίδιο το πρόγραμμα συμπεριλαμβανομένου του CryEngine Sandbox world editing system και το Mod SDK που είναι διαθέσιμα δωρεάν. Η CryEngine διαθέτει πολλά assets και game source code για να βοηθήσουν τους χρήστες να δημιουργήσουν τα δικά τους παιχνίδια. Η μόνη επιλογή γλώσσας που μπορούν να χρησιμοποιήσουν οι προγραμματιστές είναι η C++, καθώς δεν διαθέτει κάποιο σύστημα visual scripting και δεν υποστηρίζει άλλες γλώσσες. (MITTRING, 2007)

Εικόνα 19 Godot Engine

Η Godot Engine είναι μια ανοιχτού κώδικα μηχανή παιχνιδιού που παρέχει ένα πλήρες σετ εργαλείων για τη δημιουργία παιχνιδιών. Αναπτύχθηκε από τον Juan Linietsky και την ομάδα της Godot Engine, και κυκλοφόρησε για πρώτη φορά το 2014. Από τότε, έχει κερδίσει δημοτικότητα και χρησιμοποιείται ευρέως από ανεξάρτητους προγραμματιστές, εταιρείες παιχνιδιών και εκπαιδευτικά ιδρύματα. Η Godot Engine παρέχει ένα ευέλικτο γραφικό περιβάλλον ανάπτυξης που επιτρέπει τη δημιουργία 2D και 3D παιχνιδιών. Έχει ενσωματωμένα εργαλεία για τη δημιουργία γραφικών, διαχείρισης ήχου, φυσικής μηχανής, σεναρίων, εφέ και άλλων. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν τη γλώσσα προγραμματισμού GDScript, που έχει σχεδιαστεί ειδικά για την Godot Engine και παρέχει μια ευκολονόητη σύνταξη. Επιπλέον, η μηχανή υποστηρίζει και άλλες γλώσσες προγραμματισμού όπως η C# και η VisualScript. Μία από τις μεγαλύτερες δυνατότητες της

Godot Engine είναι η πολυπλατφορμικότητα. Μπορεί να αναπτύξει παιχνίδια για διάφορες πλατφόρμες όπως Windows, macOS, Linux, Android, iOS και άλλες. Επιπλέον, υποστηρίζει πολλά διαφορετικά είδη παιχνιδιών, από μικρά παιχνίδια μέχρι πιο πολύπλοκες παραγωγές. Η Godot Engine έχει αποκτήσει φήμη για την ενεργή και φιλική κοινότητά της. Οι χρήστες μπορούν να βρουν πολλούς πόρους, συμπεριλαμβανομένων εγχειριδίων, μαθημάτων και παραδειγμάτων κώδικα, που τους βοηθούν να μάθουν και να αναπτύξουν τα παιχνίδια τους.

(https://godotengine.org/, χ.χ.)

5) RPG Maker

Εικόνα 20 RPG Maker

Η RPG Maker είναι μια γνωστή game engine που επιτρέπει στους χρήστες να δημιουργήσουν τα δικά RPGs (παιχνίδια ρόλων). Αρχικά αναπτύχθηκε από την ASCII Corporation το 1988 στην Ιαπωνία και έκτοτε έχει κυκλοφορήσει αρκετές εκδόσεις και αναβαθμίσεις. Η RPG Maker έχει αποκτήσει μεγάλη δημοτικότητα λόγω της ευκολίας χρήσης και της δυνατότητας που προσφέρει στους μη προγραμματιστές να δημιουργήσουν τα δικά τους παιχνίδια χωρίς να χρειάζεται να γνωρίζουν προγραμματισμό. Η RPG Maker παρέχει ένα γραφικό περιβάλλον ανάπτυξης που επιτρέπει τη δημιουργία των χαρακτήρων, των χαρτών, των γραφικών και της ιστορίας του παιχνιδιού. Οι χρήστες μπορούν να επιλέξουν από διάφορα προκαθορισμένα assets, όπως χαρακτήρες, περιβάλλοντα, μουσική και ήχους, και να τα προσαρμόσουν στις ανάγκες τους. Επιπλέον, η RPG Maker παρέχει ένα ευέλικτο σύστημα γεγονότων και ενεργειών, που επιτρέπει στους χρήστες να ορίσουν τη λογική και τη συμπεριφορά των χαρακτήρων και του παιχνιδιού. Η RPG Maker έχει επίσης μια ενεργή κοινότητα χρηστών, όπου οι προγραμματιστές και οι δημιουργοί παιχνιδιών μπορούν να μοιραστούν ιδέες, παραδείγματα και πόρους, καθώς και να συζητούν και να ανταλλάσσουν γνώσεις. Αυτό δίνει τη δυνατότητα στους χρήστες να αντλήσουν έμπνευση και να βελτιώσουν τις δεξιότητές τους στη δημιουργία παιχνιδιών.

(https://www.rpgmakerweb.com/, n.d.)

2.6 Γλώσσες προγραμματισμού

Οι γλώσσες προγραμματισμού είναι συμβολικές γλώσσες που χρησιμοποιούνται για την ανάπτυξη λογισμικού και την επικοινωνία μεταξύ του ανθρώπου και του υπολογιστή. Κάθε γλώσσα προγραμματισμού έχει τη δομή και τους κανόνες της που καθορίζουν τον τρόπο με τον οποίο γίνεται η εκτέλεση του προγράμματος. Οι γλώσσες προγραμματισμού μπορούν να κατηγοριοποιηθούν σε διάφορες οικογένειες, όπως οι γλώσσες υψηλού επιπέδου (High-Level Languages) και οι γλώσσες χαμηλού επιπέδου (Low-Level Languages). Οι γλώσσες υψηλού επιπέδου προσφέρουν πιο αφαιρετική και ανθρώπινη σύνταξη, ενώ οι γλώσσες

χαμηλού επιπέδου είναι πιο κοντά στη μηχανική γλώσσα του υπολογιστή. Κάποιες από τις πιο δημοφιλείς γλώσσες προγραμματισμού περιλαμβάνουν την C, την C++, την Java, την Python, την JavaScript και την C#. Κάθε γλώσσα έχει τη δυνατότητα να επιτελέσει συγκεκριμένες λειτουργίες και να είναι προσανατολισμένη προς συγκεκριμένες εφαρμογές. Οι γλώσσες προγραμματισμού έχουν μια πολύ ενδιαφέρουσα ιστορία που ξεκινά από τη δεκαετία του 1940 με την εποχή των πρώτων υπολογιστών. Κατά τη διάρκεια αυτής της περιόδου, οι γλώσσες προγραμματισμού ήταν πολύ χαμηλού επιπέδου και απαιτούσαν από τους προγραμματιστές να γράφουν κώδικα σε μηχανική γλώσσα, όπως η μηχανική γλώσσα αρχής (machine language). Ωστόσο, το 1950, οι προγραμματιστές αναζητούσαν τρόπους για να γράψουν πιο αφηρημένο και ευανάγνωστο κώδικα. Αυτό οδήγησε στη δημιουργία των πρώτων γλωσσών υψηλού επιπέδου (high-level languages), όπως η Fortran το 1957 και η COBOL το 1959. Αυτές οι γλώσσες επέτρεπαν στους προγραμματιστές να γράφουν κώδικα με πιο ανθρώπινη σύνταξη, παρέχοντας πιο αφηρημένες εντολές. Στη δεκαετία του 1960, εμφανίστηκαν περισσότερες γλώσσες προγραμματισμού, όπως η ALGOL, η BASIC και η PL/I. Κατά τη διάρκεια των επόμενων δεκαετιών, η ανάπτυξη γλωσσών προγραμματισμού συνεχίστηκε με την εμφάνιση γλωσσών όπως η C, η C++, η Java, η Python και πολλές άλλες. Καθώς η τεχνολογία προχωρά και οι ανάγκες των προγραμματιστών αλλάζουν, εμφανίζονται και νέες γλώσσες προγραμματισμού που προσπαθούν να καλύψουν αυτές τις ανάγκες. Παρακάτω αναλύω μερικές από τις γλώσσες προγραμματισμού: (Lukosek, 2016)

1) C# Στην Unity

Εικόνα 21 C#

Η C# είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού η οποία δημιουργήθηκε από την Microsoft. Δανείζεται πολλά στοιχεία, και έχει παρόμοια σύνταξη, με την C++ και την Java. Η C# είναι μια πολύ γνωστή και ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού που αναπτύχθηκε από την Microsoft για τη δημιουργία εφαρμογών, για αυτό το λόγο υπάρχει μεγάλος όγκος πληροφοριών στο ίντερνετ για κάθε πρόβλημα που θα αντιμετωπίσει ο χρήστης. Με την C# η Unity εντοπίζει κατευθείαν κάποιο σφάλμα στον προγραμματισμό και σε ποιο σημείο βρίσκεται (Lukosek, 2016)

2) C

Εικόνα 22 C

Η γλώσσα προγραμματισμού C είναι μία από τις πιο επιτυχημένες και ευρέως χρησιμοποιούμενες γλώσσες στον κόσμο του προγραμματισμού. Δημιουργήθηκε από τον Dennis Ritchie τη δεκαετία του 1970 στα εργαστήρια της Bell Telephone Laboratories. Ο στόχος της C ήταν να προσφέρει έναν γενικό και ευέλικτο τρόπο προγραμματισμού, καθιστώντας την κατάλληλη για διάφορες εφαρμογές. Η C παρέχει και πρόσβαση σε χαμηλού επιπέδου λειτουργίες, όπως η διεπαφή με το σύστημα και ο χειρισμός μνήμης. Αυτή η ιδιότητα της την καθιστά ιδανική για την ανάπτυξη λογισμικού που απαιτεί γρήγορη εκτέλεση και αποδοτική χρήση πόρων. Η C έχει επηρεάσει σημαντικά τον προγραμματισμό και έχει γίνει βάση για πολλές άλλες γλώσσες, όπως η C++, η Java και η C#. Επίσης, η C χρησιμοποιείται ευρέως σε περιβάλλοντα εμπορικού λογισμικού, ενσωματωμένων συστημάτων, και για την ανάπτυξη λειτουργικών συστημάτων. (Britannica)

3) Java

Εικόνα 23 Java

Η Java είναι μια δημοφιλής γλώσσα προγραμματισμού που αναπτύχθηκε από την εταιρεία Sun Microsystems (αργότερα αποκτήθηκε από την Oracle Corporation). Η γλώσσα Java εμφανίστηκε για πρώτη φορά το 1995 και σχεδιάστηκε για να είναι ανεξάρτητη από την πλατφόρμα, δηλαδή να μπορεί να τρέχει σε οποιοδήποτε σύστημα υπολογιστή χωρίς αλλαγές. Η Java χρησιμοποιείται ευρέως για την ανάπτυξη διάφορων εφαρμογών, όπως desktop εφαρμογές, web εφαρμογές, εφαρμογές για κινητές συσκευές και ενσωματωμένα συστήματα. Έχει γίνει επίσης πολύ δημοφιλής στον τομέα της server-side ανάπτυξης, καθώς προσφέρει ασφάλεια, αξιοπιστία και ευελιξία. Ένα από τα κύρια χαρακτηριστικά της Java είναι η εικονική μηχανή Java (JVM), η οποία εκτελεί τον κώδικα Java. Αυτό επιτρέπει την ανεξαρτησία της γλώσσας από την πλατφόρμα, καθώς ο κώδικας Java μεταγλωττίζεται σε μια ενδιάμεση μορφή που μπορεί να εκτελεστεί σε οποιοδήποτε σύστημα που έχει μια JVM

εγκατεστημένη. Η Java είναι μια γλώσσα αντικειμενοστραφούς προγραμματισμού και υψηλού επιπέδου με απλή σύνταξη και εύκολη κατανόηση, τέλος προσφέρει πολλές ενσωματωμένες βιβλιοθήκες που διευκολύνουν την ανάπτυξη των προγραμμάτων.

(Daniel A. Chappell, 2002)

4) C++

Εικόνα 24 C++

Η C++ είναι μια πολυδιάστατη γλώσσα προγραμματισμού που αναπτύχθηκε από τον Bjarne Stroustrup στις αρχές της δεκαετίας του 1980 και αποτελεί ένα από τα πιο δημοφιλή εργαλεία για την ανάπτυξη λογισμικού. Η C++ ενσωματώνει τον αντικειμενοστραφή προγραμματισμό, επιτρέποντας τη δημιουργία κλάσεων και αντικειμένων, προσφέρει ισχυρές δυνατότητες και είναι κατάλληλη για μια ευρεία γκάμα εφαρμογών. Χρησιμοποιείται ευρέως στην ανάπτυξη λογισμικού, όπως εφαρμογές γραφικών χρήστη, παιχνίδια, συστήματα λειτουργικού, συστήματα ενσωματωμένων συστημάτων και πολλά άλλα. Επίσης προσφέρει πολλά χαρακτηριστικά, όπως αντικείμενα, κλάσεις, κληρονομικότητα, πολυμορφισμό, templates και πρόσβαση σε χαμηλού επιπέδου λειτουργίες της γλώσσας C. Αυτό την καθιστά μια ισχυρή γλώσσα με ευελιξία και απόδοση. (Williams, 2007)

5) Python

Εικόνα 25 Python

Η Python είναι μια δημοφιλής γλώσσα προγραμματισμού υψηλού επιπέδου που έχει σχεδιαστεί για να είναι ευανάγνωστη και ευέλικτη. Αναπτύχθηκε από τον Guido van Rossum και πρωτοκυκλοφόρησε το 1991. Η Python έχει αποκτήσει μεγάλη δημοτικότητα λόγω της απλότητας της και της ευκολίας χρήσης της. Οι βασικές αρχές της Python είναι η αναγνωσιμότητα του κώδικα και η εστίαση στην καθαρότητα και τη συνοχή. Ο κώδικας Python είναι γραμμένος με έναν συντακτικό που προωθεί την κατανόηση του κώδικα, με χρήση κενών αντικατάστασης, απλών δομών και λίγων συμβόλων. Αυτό το καθιστά ιδιαίτερα φιλικό για αρχάριους προγραμματιστές και επιτρέπει την ταχεία ανάπτυξη εφαρμογών. Η Python υποστηρίζει πολλές σύγχρονες τεχνολογίες και πλατφόρμες, καθιστώντας την ευέλικτη για πολλούς τύπους ανάπτυξης. Χρησιμοποιείται ευρέως σε

πεδία όπως οι επιστημονικοί υπολογισμοί, οι εφαρμογές web, οι βιομηχανικές εφαρμογές, η τεχνητή νοημοσύνη, οι διαδικτυακές εφαρμογές και πολλά άλλα. Μια από τις κύριες δυνατότητες της Python είναι η ύπαρξη μιας πλούσιας βιβλιοθήκης, γνωστής ως "Python Standard Library", που περιλαμβάνει πολλές έτοιμες λειτουργίες και εργαλεία για διάφορες εργασίες. Αυτό συνεπάγεται ότι ο προγραμματιστής μπορεί να επωφεληθεί από έτοιμες λύσεις και να εξοικονομήσει χρόνο και κόπο. Η Python υποστηρίζεται από μια ενεργή κοινότητα προγραμματιστών που συνεχίζει να αναπτύσσει και να βελτιώνει τη γλώσσα. Υπάρχουν πολλές τρίτες βιβλιοθήκες και εργαλεία που έχουν δημιουργηθεί από την κοινότητα και προσφέρουν επιπλέον λειτουργικότητα και δυνατότητες.

(https://www.python.org, n.d.)

2.7 Προγράμματα επεξεργασίας πηγαίου κώδικα

Τα προγράμματα επεξεργασίας πηγαίου κώδικα είναι εργαλεία λογισμικού που χρησιμοποιούνται για την ανάπτυξη και την τροποποίηση προγραμμάτων. Αυτά τα προγράμματα παρέχουν ένα ολοκληρωμένο περιβάλλον εργασίας για τους προγραμματιστές, τους βοηθά να γράφουν, να επεξεργάζονται και να οργανώνουν τον πηγαίο κώδικα. Επίσης προσφέρουν διάφορα χαρακτηριστικά και λειτουργίες που βοηθούν στην αποτελεσματική προγραμματιστική διαδικασία. Ορισμένα από τα συνηθισμένα χαρακτηριστικά των προγραμμάτων επεξεργασίας πηγαίου κώδικα περιλαμβάνουν:

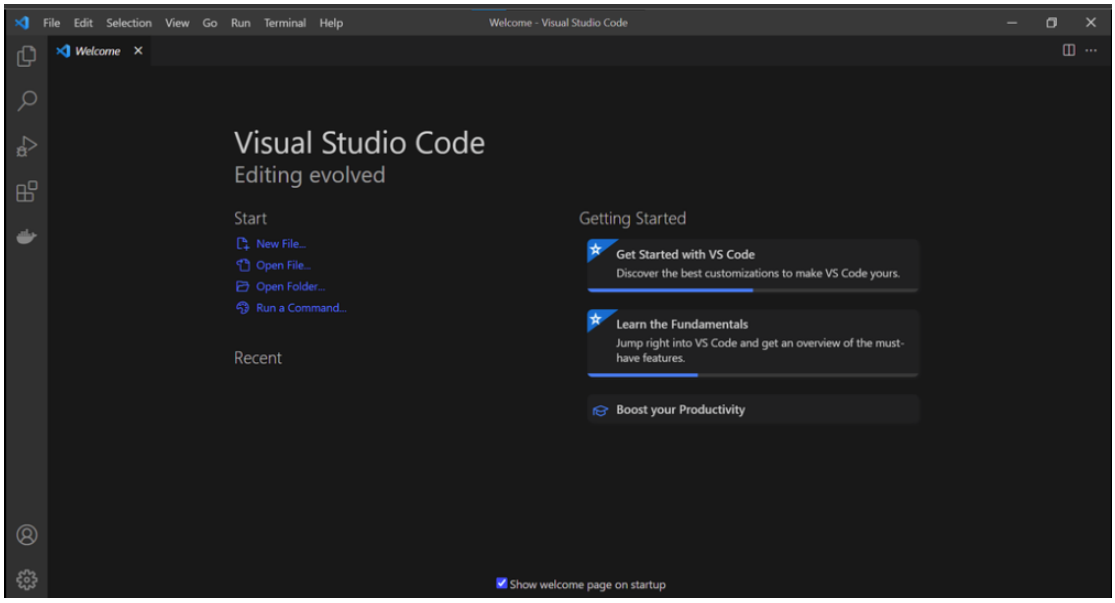
1) Σύνταξη και χρωματισμός κώδικα: Τα προγράμματα επεξεργασίας πηγαίου κώδικα προσφέρουν υποστήριξη για σύνταξη κώδικα, με χρωματισμό και επισήμανση σύνταξης για διάφορες γλώσσες προγραμματισμού. Αυτό βοηθάει στην ευκολότερη ανάγνωση και κατανόηση του κώδικα.
2) Αυτόματη συμπλήρωση κώδικα: Οι περισσότεροι επεξεργαστές προσφέρουν λειτουργία αυτόματης συμπλήρωσης κώδικα, όπου προβάλλονται προτάσεις κώδικα καθώς ο προγραμματιστής πληκτρολογεί, με βάση το περιβάλλον ανάπτυξης και τις δομές της γλώσσας προγραμματισμού.
3) Αναζήτηση και αντικατάσταση: Οι προγραμματιστές μπορούν να εκτελούν αναζητήσεις και αντικαταστάσεις κειμένου στον κώδικα, επιτρέποντας τους να βρίσκουν γρήγορα και να αλλάζουν τμήματα κώδικα.
4) Πλοήγηση στον κώδικα: Τα προγράμματα επεξεργασίας πηγαίου κώδικα παρέχουν εργαλεία για την πλοήγηση στον κώδικα, όπως προβολή της δομής του προγράμματος, πηγαίοι αναγνώστες και εργαλεία για την προβολή των συναρτήσεων και των κλάσεων.
5) αποσφαλμάτωση: Τα προγράμματα επεξεργασίας πηγαίου κώδικα παρέχουν εργαλεία που βοηθούν τους προγραμματιστές να εντοπίζουν και να διορθώνουν σφάλματα στον κώδικα
   (Liang, 2009)

2.7.1 Ορισμένα δημοφιλή προγράμματα επεξεργασίας πηγαίου κώδικα είναι:

1) Το Visual Studio Code:

Εικόνα 26 Visual Studio Code

Το Visual Studio Code , κοινώς γνωστό και ως VS Code , είναι ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα που δημιουργήθηκε από τη Microsoft για Windows , Linux και macOS . Τα χαρακτηριστικά του Visual Studio Code περιλαμβάνουν υποστήριξη για εντοπισμό σφαλμάτων , επισήμανση σύνταξης , έξυπνη συμπλήρωση κώδικα , αποσπάσματα , ανακατασκευή κώδικα και ενσωματωμένο Git . Επίσης, οι χρήστες μπορούν να αλλάξουν το θέμα του συντάκτη , τις συντομεύσεις πληκτρολογίου , τις προτιμήσεις και να εγκαταστήσουν επεκτάσεις που προσθέτουν επιπλέον λειτουργίες. Είναι δωρεάν και ανοικτού κώδικα, αν και η επίσημή λήψη είναι υπό ιδιωτική άδεια**.** Το Visual Studio Code ανακοινώθηκε για πρώτη φορά στις 29 Απριλίου 2015, από τη Microsoft στο συνέδριο Build 2015. Το Visual Studio Code είναι ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα που μπορεί να χρησιμοποιηθεί με διάφορες γλώσσες προγραμματισμού, όπως Java , JavaScript , Go , Node.js , Python , C++ , C , C# , Rust και Fortran . Είναι βασισμένο στο Electron, ένα πλαίσιο που χρησιμοποιείται για την ανάπτυξη εφαρμογών Node.js για την επιφάνεια εργασίας που εκτελείται στη μηχανή εμφάνισης Blink. Παρόλο που χρησιμοποιεί το ηλεκτρονικό πλαίσιο, το λογισμικό δεν χρησιμοποιεί το Atom και χρησιμοποιεί την ίδια συνιστώσα επεξεργαστή (με την κωδική ονομασία "Monaco") που χρησιμοποιείται στα Visual Studio Team Services (πρώην Visual Studio Online). Το Visual Studio για τις περισσότερες γλώσσες προγραμματισμού είναι βοηθητικό προς τον χρήστη όπως: επισήμανση σύνταξης , αντιστοίχιση αγκυλών και δίπλωμα κώδικα. Ο κώδικας Visual Studio συνδυάζει την απλότητα ενός επεξεργαστή πηγαίου κώδικα με ισχυρά εργαλεία ανάπτυξης, όπως η ολοκλήρωση και η αποσφαλμάτωση κώδικα IntelliSense. Στην Έρευνα Προγραμματιστών Stack Overflow 2018, ο κώδικας Visual Studio κατατάχθηκε ως το πιο δημοφιλές εργαλείο περιβάλλοντος για προγραμματιστές, με το 34,9% των 75.398 ερωτηθέντων να ισχυρίζονται ότι το χρησιμοποιούν. (https://code.visualstudio.com/.)



Εικόνα 27 (Visual Studio Code που εκτελείται σε Windows 10)

2) Visual Studio:

Εικόνα 28 Visual Studio

Το Visual Studio είναι ένα ενσωματωμένο περιβάλλον ανάπτυξης (IDE) που αναπτύχθηκε από τη Microsoft και η πρώτη του έκδοση κυκλοφόρησε τον Απρίλιο του 1997. Παρέχει ένα πλήρες σετ εργαλείων για τη δημιουργία και την επεξεργασία πηγαίου κώδικα σε διάφορες γλώσσες προγραμματισμού, όπως C++, C#, Visual Basic, Java κ.α. Από τότε, έχουν κυκλοφορήσει πολλές εκδόσεις του Visual Studio, με συνεχείς αναβαθμίσεις και προσθήκες νέων λειτουργιών. Η πιο πρόσφατη έκδοση είναι το Visual Studio 2022, η οποία κυκλοφόρησε τον Νοέμβριο του 2021. Τα χαρακτηριστικά και πλεονεκτήματα του Visual Studio περιλαμβάνουν ένα ολοκληρωμένο περιβάλλον ανάπτυξης που συνδυάζει τη συγγραφή κώδικα, την εύρεση σφάλματος, τη διαχείριση της εργασίας και άλλες λειτουργίες σε ένα μόνο παράθυρο. Ακόμα το Visual Studio υποστηρίζει πολλές γλώσσες προγραμματισμού, όπως C++, C#, Visual Basic, Java, Python, JavaScript και πολλές άλλες. Αυτό το καθιστά ιδανικό για προγραμματιστές που εργάζονται σε διάφορες πλατφόρμες και τεχνολογίες. Επίσης είναι πλούσιο σε εργαλεία ανάπτυξης όπως γραφικοί σχεδιαστές διεπαφής χρήστη, εργαλεία διαχείρισης βάσεων δεδομένων, ενσωματωμένα συστήματα αποσφαλμάτωσης και προφίλερ για βελτιστοποίηση της απόδοσης. Το Visual Studio παρέχει εργαλεία για τη διαχείριση των έργων ανάπτυξης, συμπεριλαμβανομένης της δυνατότητας δημιουργίας, σύγκρισης και ενημέρωσης αρχείων, καθώς και διαχείρισης των εξαρτημάτων και βιβλιοθηκών του έργου. Τέλος είναι σχεδιασμένο για να υποστηρίζει πλήρως τις πλατφόρμες της Microsoft, όπως το Windows, το Azure, το .NET Framework και το Xamarin για ανάπτυξη εφαρμογών για κινητές συσκευές.

(Gosselin, 2017)

3) IntelliJ IDEA:

Εικόνα 29 IntelliJ IDEA

Το IntelliJ IDEA είναι ένα περιβάλλον ανάπτυξης (IDE) για την προγραμματιστική γλώσσα Java, αλλά υποστηρίζει επίσης πολλές άλλες γλώσσες προγραμματισμού, όπως η Kotlin, η Scala, η Groovy και άλλες. Αναπτύχθηκε από την εταιρεία JetBrains και προσφέρει ένα πλούσιο σε δυνατότητες περιβάλλον για την ανάπτυξη λογισμικού. Το 2009, η JetBrains κυκλοφόρησε τον πηγαίο κώδικα για το IntelliJ IDEA υπό την open-source κώδικα Apache 2.0. Η JetBrains άρχισε να διανέμει μια περιορισμένη έκδοση του IntelliJ IDEA που αποτελείται από χαρακτηριστικά open-source με το ψευδώνυμο Community Edition. Τον Δεκέμβριο του 2014, η Google ανακοίνωσε την version 1.0 of Android Studio. Ένα από τα βασικά χαρακτηριστικά του IntelliJ IDEA είναι το ισχυρό σύστημα αυτόματης συμπλήρωσης κώδικα που προβλέπει την επόμενη λέξη ή φράση που πρόκειται να πληκτρολογήσει ο χρήστης βάσει του περιβάλλοντος ανάπτυξης και των προηγούμενων προτιμήσεων του. Με αυτό τον τρόπο ο χρήστης αυξάνει την παραγωγικότητά του και εξοικονομεί χρόνο. Ακόμα το IntelliJ IDEA παρέχει προηγμένες δυνατότητες επεξεργασίας κώδικα όπως αναδίπλωση κώδικα, αυτόματη διόρθωση, αναδιατύπωση κώδικα, αναγνώριση τύπων, αποσφαλμάτωση και άλλα. Αυτό επιτρέπει στους προγραμματιστές να γράφουν κώδικα με μεγαλύτερη ευκολία και ακρίβεια. Επίσης περιέχει πολλά ενσωματωμένα εργαλεία που βοηθούν στη διαχείριση του κύκλου ζωής του έργου, όπως το σύστημα ελέγχου ταυτότητας (Version Control), το σύστημα διαχείρισης εξαρτήσεων (Dependency Management) και το ενσωματωμένο σύστημα οργάνωσης και διαχείρισης έργων. Ένα σημαντικό ακόμα χαρακτηριστικό του είναι ότι προσφέρει ένα ισχυρό σύστημα αυτόματης ολοκλήρωσης κώδικα, που παρέχει προτάσεις κώδικα, μεθόδους, κλάσεις και άλλες δομές δεδομένων, βάσει του περιβάλλον ανάπτυξης και του συγκεκριμένου κώδικα που γράφετε. Τέλος το IntelliJ IDEA διαθέτει εξελιγμένα εργαλεία ανάλυσης κώδικα που επιτρέπουν στους προγραμματιστές να εντοπίζουν και να διορθώνουν σφάλματα, να βελτιστοποιούν την απόδοση του κώδικα και να βελτιώνουν την ποιότητά του.
(https://www.jetbrains.com/idea/., n.d.)

4) Eclipse

Εικόνα 30 Eclipse

Το Eclipse είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που χρησιμοποιείται κυρίως για την ανάπτυξη λογισμικού σε Java, αλλά υποστηρίζει επίσης και άλλες γλώσσες προγραμματισμού όπως C/C++, Python, PHP και άλλες. Αναπτύχθηκε αρχικά από την εταιρεία IBM και κυκλοφόρησε το 2001 ως ανοιχτού κώδικα λογισμικό. Το Eclipse προσφέρει πλούσια λειτουργικότητα για την ανάπτυξη λογισμικού, συμπεριλαμβανομένων των εργαλείων για την αποσφαλμάτωση, τη διαχείριση των εκδόσεων του κώδικα, την αυτόματη συμπλήρωση κώδικα, τη διαχείριση των εξαρτήσεων του έργου και πολλά άλλα. Είναι επίσης επεκτάσιμο μέσω πρόσθετων (plugins) που επιτρέπουν την προσαρμογή του περιβάλλοντος ανάπτυξης σύμφωνα με τις ανάγκες του προγραμματιστή. Ένα από τα κύρια πλεονεκτήματα του Eclipse είναι η δυνατότητα προβολής του πηγαίου κώδικα και των συναρτήσεων σε πραγματικό χρόνο, παρέχοντας ένα ισχυρό εργαλείο για την κατανόηση και την αποσφαλμάτωση του κώδικα. Επιπλέον, παρέχει ευέλικτες επιλογές για τη διαχείριση έργων, την οργάνωση αρχείων και φακέλων, καθώς και για την ομαδική εργασία μέσω ενσωματωμένων συστημάτων ελέγχου εκδόσεων. (https://www.eclipse.org, n.d.)

5) Xcode

Εικόνα 31 Xcode

Το Xcode είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που αναπτύχθηκε από την Apple για τη δημιουργία εφαρμογών για τα λειτουργικά συστήματα iOS, macOS, watchOS και tvOS. Είναι το πρωτεύον εργαλείο που χρησιμοποιείται από τους προγραμματιστές iOS για την ανάπτυξη iPhone και iPad εφαρμογών. Το Xcode παρέχει ένα πλούσιο σύνολο εργαλείων για την ανάπτυξη εφαρμογών, συμπεριλαμβανομένων ενός επεξεργαστή κειμένου με σύνταξη επισημάνσεων, αυτόματη συμπλήρωση κώδικα, αποσφαλμάτωση, εργαλεία δημιουργίας διεπαφών χρήστη, διαχείριση εξαρτήσεων και έκδοσης, και πολλά άλλα. Επιπλέον, το Xcode παρέχει ενσωματωμένες προσομοιώσεις συσκευών για να δοκιμάζετε τις εφαρμογές σας κατά τη διάρκεια της ανάπτυξης. Η Xcode αναπτύχθηκε ειδικά για την πλατφόρμα της Apple παρέχει εργαλεία για την ανάπτυξη γλωσσών προγραμματισμού όπως Swift, Objective-C και C++. (https://developer.apple.com/xcode/), n.d.)

2.8 Αρχιτεκτονική για serious games

Αν ο σχεδιασμός του serious game δεν είναι καλά καθορισμένος δεν θα λύσει όλες τις προκλήσεις , οι επαγγελματίες που εργάζονται στον σχεδιασμό των serious games συμφωνούν ότι αποτελούνται από 3 κύρια στοιχεία: περιεχόμενο, παιχνίδι και μάθηση. Η αρχιτεκτονική για serious games αποτελεί ένα κρίσιμο στοιχείο για την ανάπτυξη επιτυχημένων παιχνιδιών με σοβαρό περιεχόμενο. Αυτή η αρχιτεκτονική περιλαμβάνει τη σχεδίαση και τη δομή του παιχνιδιού, την ανάπτυξη των λειτουργιών, την επιλογή των τεχνολογιών και τη διαχείριση των δεδομένων. Στην αρχιτεκτονική αυτή, λαμβάνονται υπόψη οι στόχοι του παιχνιδιού, οι απαιτήσεις του κοινού και οι εκπαιδευτικοί στόχοι, προκειμένου να δημιουργηθεί μια αποτελεσματική και ενδιαφέρουσα εμπειρία για τον χρήστη. Η αρχιτεκτονική ενός serious game επίσης εξετάζει ζητήματα όπως η διαχείριση της αλληλεπίδρασης του παίκτη με το παιχνίδι, η αξιολόγηση της προόδου και η παρακολούθηση της απόδοσης του παίκτη. (Alvarez, Origins of Serious Games, 2011)

Κεφάλαιο 3

3.1.1 Το παιχνίδι– Εισαγωγή

Για την κατασκευή του serious game χρησιμοποίησα το Unity, επειδή είχα πρόσβαση σε ένα εκτεταμένο σύνολο εργαλείων που με βοήθησαν να κατασκευάσω την οπτική αισθητική, τις αλληλεπιδράσεις και τη λογική του παιχνιδιού μου. Η γλώσσα προγραμματισμού C# μου προσέφερε τη δυνατότητα να δημιουργήσω και να διαχειριστώ τους μηχανισμούς του παιχνιδιού, όπως την κίνηση των χαρακτήρων, την απόκριση σε ενέργειες του χρήστη και την επικοινωνία με τα αντικείμενα του παιχνιδιού. Η χρήση του Visual Studio Code ως περιβάλλον ανάπτυξης μου επέτρεψε να γράψω τον κώδικα με άνεση και αποτελεσματικότητα. Το περιβάλλον προσφέρει πολλές χρήσιμες λειτουργίες, όπως αυτόματη συμπλήρωση κώδικα, αναζήτηση σφαλμάτων και ολοκλήρωση λειτουργιών. Με τη συνδυασμένη χρήση του Unity, της γλώσσας C# και του Visual Studio Code, μπόρεσα να αναπτύξω ένα παιχνίδι που ανταποκρίνεται στις προσδοκίες μου και παρέχει μια ευχάριστη εμπειρία στους παίκτες.

Εικόνα 32 Το παιχνίδι όταν είναι σε λειτουργία

Όταν ο παίκτης θα πιάνει με τον κάδο ανακύκλωσης τα σωστά αντικείμενα τότε θα προθέτονται +1 πόντος. Πάνω δεξιά θα υπάρχει ένα πλαίσιο για το σκορ του παιχνιδιού όπου εκεί θα φαίνονται οι πόντοι που θα έχει ο παίκτης, επίσης ακριβώς από κάτω από τους πόντους αναγράφεται η υψηλότερη βαθμολογία(highscore) , ενώ πάνω αριστερά θα υπάρχει ένα πλαίσιο που θα δείχνει πόσες ζωές θα έχει ακόμα ο παίκτης. Όταν ο χρήστης θα πιάνει λάθος αντικείμενο με τον κάδο ανακύκλωσης θα χάνει μια ζωή. Γενικότερα η ιδέα του παιχνιδιού είναι πολύ απλή, απλά ο σκοπός του παίκτη είναι να πιάνει με τον κάδο ανακύκλωσης τα σωστά αντικείμενα . Κάθε φορά που θα το πετυχαίνει αυτό ο κάθε παίκτης θα παίρνει 1 πόντο ενώ κάθε φορά που θα τοποθετείται το λάθος αντικείμενο στο κάδο ανακύκλωσης θα χάνει μια ζωή. Το παιχνίδι ουσιαστικά θα περιέχει κινούμενες εικόνες και ένα σύστημα καταγραφής σκορ. Το επόμενο βήμα ήταν να σκεφτώ τα game mechanics(

μηχανισμούς του παιχνιδιού) που έπρεπε να φτιάξω για το παιχνίδι , δηλαδή τι προγραμματισμό να χρησιμοποιήσω για να έχω ένα λειτουργικό παιχνίδι :

- Έναν χαρακτήρα ικανό να κινείται δεξιά αριστερά σε ένα δισδιάστατο περιβάλλον.
- Ο χειριστής του παιχνιδιού να είναι σε θέση να αυξάνει την ταχύτητα του χαρακτήρα με το πάτημα ενός κουμπιού.
- Επίσης να υπάρχει ο κατάλληλος προγραμματισμός όταν υπάρχει η επαφή λάθους αντικειμένου με τον χαρακτήρα να χάνει ζωή (π.χ αν ο κάδος ανακυκλώσεις και ο χειριστής ακουμπήσει μη ανακυκλώσιμο αντικείμενο).
- Τέλος, όταν υπάρχει η επαφή σωστού αντικειμένου με τον χαρακτήρα να κερδίζει πόντους (π.χ. ο χειριστής με τον κάδο ανακυκλώσεις ακουμπήσει τα ανακυκλώσιμα αντικείμενα).

3.1.2 Αρχική οθόνη- Το Menu

Εικόνα 33 Το Menu

Το μενού αποτελεί ένα από τα κύρια στοιχεία ενός παιχνιδιού καθώς αποτελεί το πρώτο σημείο αλληλεπίδρασης με τον χρήστη. Παρόλο που δεν το αντιλαμβανόμαστε ως μία πίστα στην κυριολεξία αλλά στη λογική την οποία ακολουθήσαμε, το μενού είναι ακόμα μία πίστα. Εάν ακολουθήσουμε αυτήν την προσέγγιση, θα είναι πιο εύκολο για εμάς να συνδέσουμε προγραμματιστικά το μενού με τις υπόλοιπες πίστες του παιχνιδιού, καθώς και τις πίστες που μας επιτρέπουν να επιστρέψουμε πίσω στο μενού. Αυτή η προσέγγιση μας βοηθά να οργανώσουμε τον κώδικα και να διαχειριστούμε τη μετάβαση ανάμεσα στις διάφορες "πίστες" του παιχνιδιού, συμπεριλαμβανομένου του μενού. Στο συγκεκριμένο μενού που έφτιαξα, για το serious game με θέμα την ανακύκλωση, έχω τοποθετήσει ένα button που αυτό όταν το πατήσει ο χρήστης θα ξεκινήσει το παιχνίδι. Επιπλέον έχω τοποθετήσει κάποιες εικόνες (assets), που χρησιμοποιώ στο παιχνίδι, και δίνω πληροφορίες για ποια αντικείμενα μπορούν να ανακυκλωθούν και σε πιο κάδο ανακύκλωσης αλλά και ποια από αυτά δεν είναι ανακυκλώσιμα. Τέλος στο κέντρο της οθόνης φαίνεται η γη με το σύμβολο της ανακύκλωσης και ένα τίτλο και που περιγράφει τον σκοπό του παιχνιδιού.

3.1.3 Οθόνη τέλους- Game Over

Εικόνα 34 Game Over

Το στοιχείο "Game Over" αναφέρεται στην κατάσταση όπου ο παίκτης έχει χάσει στο παιχνίδι και έχει φτάσει σε ένα τερματικό σημείο. Όπως και με το μενού και αυτό αποτελεί ένα σημαντικό στοιχείο για το παιχνίδι και ακολούθησα πάλι την λογική ότι αποτελεί ακόμα μία πίστα, για να μπορέσω προγραμματιστικά να συνδέσω την οθόνη τέλους με τις άλλες πίστες του παιχνιδιού. Στην πίστα "Game Over" που δημιούργησα έχω τοποθετήσει δυο button που το ένα παρέχει την δυνατότητα στον χρήστη να ξεκινήσει ξανά το παιχνίδι, και το δεύτερο την δυνατότητα να επιστρέψει στο μενού. Επίσης πάνω αριστερά φαίνεται ότι στο χρήστη του απέμειναν 0 ζωές και πάνω δεξιά τους πόντους που μάζεψε στο παιχνίδι. Τέλος στο κέντρο αναγράφεται "Game Over" δηλαδή την κατάσταση που βρίσκεται εκείνη την στιγμή ο χρήστης.

1.1.4 Τα Assets και τα αντικείμενα που χρησιμοποίησα στο Project

Τα Assets είναι οτιδήποτε θέλουμε να χρησιμοποιήσουμε για το παιχνίδι – γραφικά, εικόνες, ήχος, animations, scripts κ.α. Κάθε αντικείμενο στην Unity μπορεί να περιέχει πολλαπλά στοιχεία που καθορίζουν τη λειτουργικότητά του. Ένα από τα βασικά στοιχεία ενός αντικειμένου είναι το "Transform" ή το "Rect Transform" για γραφικά στοιχεία διεπαφής χρήστη. Το "Transform" περιλαμβάνει πληροφορίες σχετικά με τη θέση, το μέγεθος, τις συντεταγμένες και την περιστροφή του αντικειμένου στη σκηνή. Αυτό το στοιχείο μας επιτρέπει να μετακινούμε, να περιστρέφουμε και να αλλάζουμε το μέγεθος του αντικειμένου. Για γραφικά στοιχεία διεπαφής χρήστη, χρησιμοποιείται το "Rect Transform" που περιλαμβάνει πληροφορίες για τη θέση και το μέγεθος του στοιχείου στον χώρο της διεπαφής. Αυτό μας επιτρέπει να τοποθετούμε στοιχεία όπως επιλογές παιχνιδιού, μενού, στατιστικά και βαθμολογίες. Ουσιαστικά, αυτά τα στοιχεία παρέχουν πληροφορίες σχετικά με τη θέση, το μέγεθος και τη μορφή του αντικειμένου, επιτρέποντάς μας να το αλλάξουμε και να το τοποθετήσουμε σωστά στην σκηνή του παιχνιδιού. Το "Physics" αναφέρεται σε ό,τι σχετίζεται με τη φυσική και τους φυσικούς νόμους στο παιχνίδι. Στην Unity, υπάρχουν δύο βασικά στοιχεία της φυσικής που χρησιμοποιούνται, τα Colliders2D είναι συνιστώσες που προστίθενται στα αντικείμενα για να ανιχνεύουν την αλληλεπίδρασή τους με άλλα αντικείμενα στη σκηνή. Όταν δύο Colliders2D αλληλεπιδρούν, μπορούν να προκαλέσουν αντιδράσεις όπως ανίχνευση σύγκρουσης, αλλαγή κατεύθυνσης, αναπήδηση κ.λπ. Αυτά τα στοιχεία χρησιμοποιούνται για τη δημιουργία της φυσικής

αλληλεπίδρασης μεταξύ των αντικειμένων στον χώρο του παιχνιδιού. Τα Rigidbody2D είναι συνιστώσες που προστίθενται στα αντικείμενα για να επηρεάζουν τη φυσική τους συμπεριφορά. Αποδίδουν ρεαλιστική κίνηση και αλληλεπίδραση με τις δυνάμεις της βαρύτητας και άλλες φυσικές δυνάμεις. Τα αντικείμενα με Rigidbody2D μπορούν να μετακινηθούν, να αντιδρούν σε συγκρούσεις και να αλληλεπιδρούν με το περιβάλλον τους με βάση τους φυσικούς νόμους. Ουσιαστικά, τα Colliders2D και τα Rigidbody2D είναι στοιχεία που προσθέτονται στα αντικείμενα για να επιτρέψουν την ανίχνευση σύγκρουσης και την εφαρμογή φυσικής συμπεριφοράς στο παιχνίδι. Τα Colliders2D μπορούν επίσης να οριστούν ως "triggers" στον Inspector της Unity. Αυτό σημαίνει ότι η λειτουργία της φυσικής προσκόλλησης απενεργοποιείται και το Collider2D χρησιμοποιείται αποκλειστικά για την ανίχνευση του πότε δύο αντικείμενα έρχονται σε επαφή. Μέσω κώδικα, μπορούμε να χρησιμοποιήσουμε αυτήν την ανίχνευση για να δημιουργήσουμε "γεγονότα" που συμβαίνουν όταν αυτή η επαφή πραγματοποιείται. Για παράδειγμα, ας υποθέσουμε ότι έχουμε τα Colliders2D του παίκτη και των εχθρών στο παιχνίδι. Μέσω των scripts, μπορούμε να αφαιρέσουμε "health" από τον παίκτη όταν ο εχθρός έρχεται σε επαφή μαζί του. Όταν το "health" του παίκτη φτάσει στο μηδέν, τότε ο χρήστης χάνει το παιχνίδι. Τέλος τα layers που στην Unityείναι ένα εργαλείο που χρησιμοποιείται για να ορίσει τα επίπεδα της διαδραστικότητας των αντικειμένων στη σκηνή. Κάθε αντικείμενο μπορεί να ανήκει σε ένα συγκεκριμένο layer. Η χρήση των layers επιτρέπει τον έλεγχο της αλληλεπίδρασης μεταξύ των αντικειμένων. Για παράδειγμα, όταν δύο αντικείμενα ανήκουν σε διαφορετικά layers, παρόλο που και τα δύο έχουν colliders, δεν θα συγκρούονται μεταξύ τους. Αυτό σημαίνει ότι το ένα αντικείμενο μπορεί να περάσει μέσα από το άλλο χωρίς να υπάρξει αλληλεπίδραση. Ο ορισμός των layers δεν είναι απαραίτητος, αλλά είναι ένα χρήσιμο εργαλείο για τον έλεγχο της συμπεριφοράς των αντικειμένων στη σκηνή. Μπορεί να χρησιμοποιηθεί για να διαχωρίσεις αντικείμενα που δεν πρέπει να αλληλεπιδρούν μεταξύ τους ή για να δημιουργήσεις συγκεκριμένη συμπεριφορά βάσει του επιπέδου τους.

Τα Assets που χρησιμοποίησα στο παιχνίδι:

To background

Εικόνα 35 background

Για τον player

Εικόνα 36 Player

Για τα Recycle Items

3.2 ΚΩΔΙΚΑΣ- Scrip

Ένα script είναι ένα αρχείο κειμένου σε μια γλώσσα προγραμματισμού που περιέχει κώδικα για να ορίσει τη συμπεριφορά ενός αντικειμένου στο παιχνίδι. Μπορεί να περιέχει μεθόδους, μεταβλητές και λογική για να επιτελέσει συγκεκριμένες λειτουργίες ή να αλληλεπιδράσει με άλλα αντικείμενα στο παιχνίδι. Στο πλαίσιο της Unity, ένα ολοκληρωμένο έργο συνήθως περιλαμβάνει πολλά scripts. Ο χρήστης προσθέτει κάθε script σε ένα αντικείμενο του παιχνιδιού για να καθορίσει τη λειτουργία του. Ένα αντικείμενο μπορεί να έχει πολλαπλά scripts, ανάλογα με τις επιθυμητές λειτουργίες. Για παράδειγμα, σε ένα παιχνίδι, το αντικείμενο που αναπαριστά τον παίκτη μπορεί να έχει ξεχωριστά scripts για την κίνηση, την επίθεση, την υγεία και την αλληλεπίδραση με άλλα αντικείμενα. Κάθε script έχει τον δικό του ρόλο και συνεργάζεται με άλλα scripts για να δημιουργήσει την επιθυμητή συμπεριφορά του παιχνιδιού. Για παράδειγμα έχουμε ένα παίκτη με το όνομα "kostas" στη σκηνή, ο οποίος του έχει προστεθεί ένας collider και ένας rigidbody για τη μάζα του, καθώς και ένα script με όνομα "Health" για την υγεία του, που περιέχει μια μεταβλητή τύπου int με όνομα "currentHealth" και αρχική τιμή 100. Τώρα, αν θέλουμε να δημιουργήσουμε μια παγίδα που θα προκαλεί ζημιά στον παίκτη όταν αυτός την ακουμπάει, μπορούμε να ακολουθήσουμε τα παρακάτω βήματα: Σερνουμε το γραφικό της παγίδας στη σκηνή. Προσθέτουμε έναν collider στο αντικείμενο της παγίδας και επιλέγουμε να το μετατρέψουμε σε trigger collider από τις επιλογές του. Δημιουργούμε ένα script με όνομα "Trap" και το προσθέτουμε στο αντικείμενο της παγίδας στην ιεραρχία, μέσω του Inspector. Για να επηρεάσουμε την υγεία του παίκτη με την παγίδα, ανοίγουμε το script "Trap" και δηλώνουμε μια μεταβλητή που θα αναφέρεται στο script "Health". Έτσι, μπορούμε να έχουμε πρόσβαση στην υγεία του παίκτη και να προγραμματίσουμε τη ζημιά που προκαλείται. Συνοψίζοντας, οι βασικές ενέργειες που πρέπει να κάνουμε είναι:

Προσθέτουμε έναν trigger collider στο αντικείμενο της παγίδας.

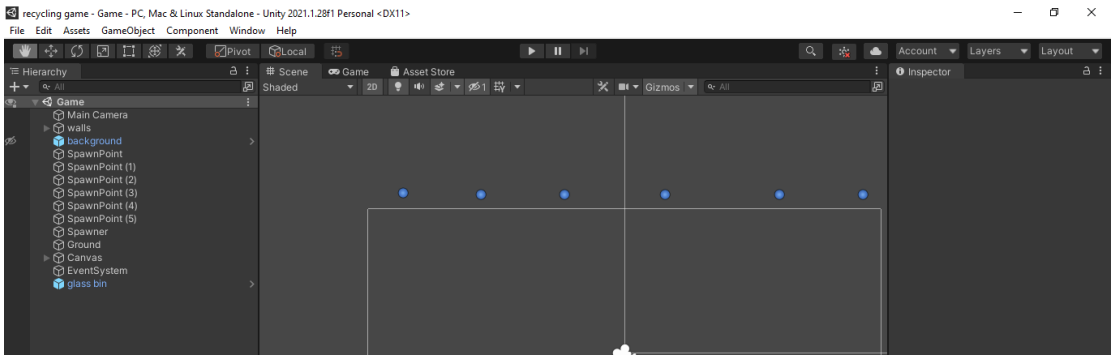
Δημιουργούμε ένα script με όνομα "Trap" και το προσθέτουμε στο αντικείμενο της παγίδας.

Στο σενάριο "Trap", δηλώνουμε μια μεταβλητή για να αναφερόμαστε στο σενάριο "Health" του παίκτη.

Από εκεί και πέρα, μπορούμε να προγραμματίσουμε την απαραίτητη λογική για την αλληλεπίδραση μεταξύ της παγίδας και της υγείας του παίκτη στο σενάριο "Trap".

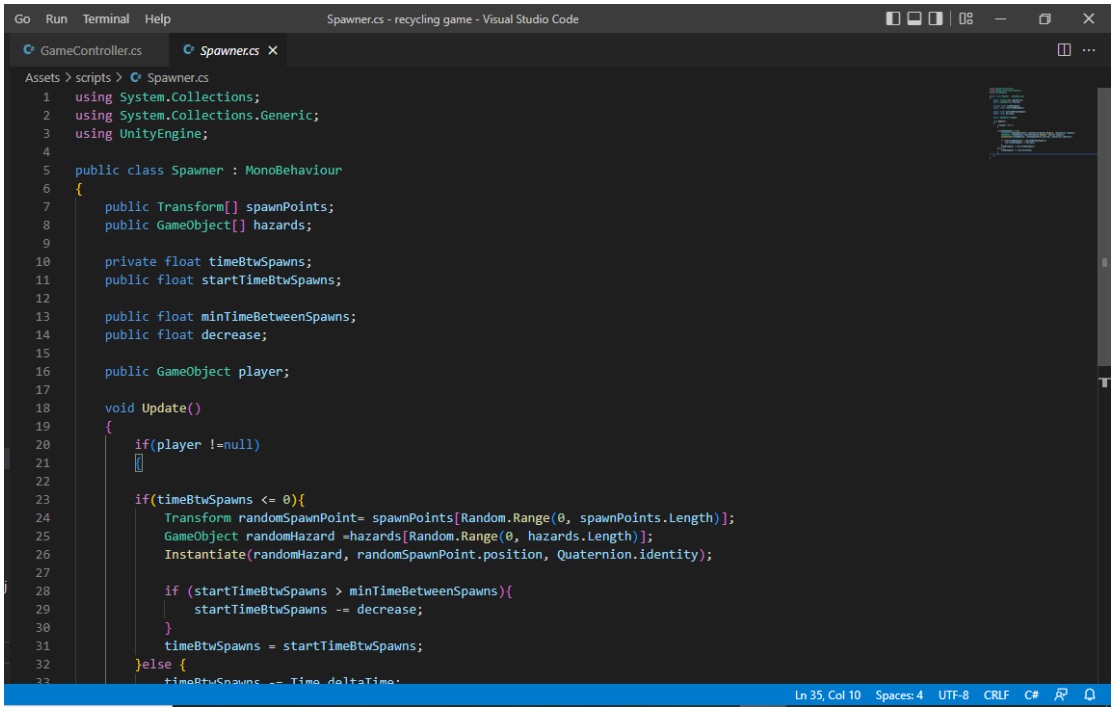
3.2.1 Script SpawnPoint

Το πρώτο που έκανα πριν φτιάξω το spawners scrip ήταν να βρώ τις θέσεις όπου τα αντικείμενα θα έκαναν την αρχική τους εμφάνιση, έτσι δημιούργησα empty game object για να ξέρω που θα βρίσκεται η αρχική τους θέση.

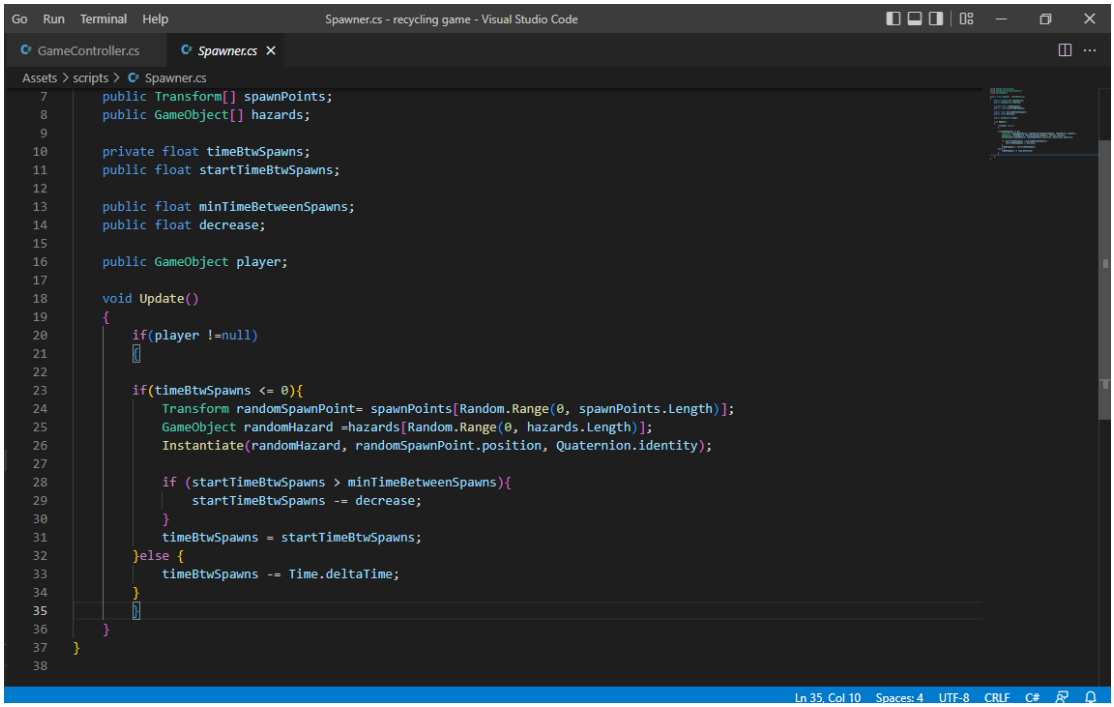


Εικόνα 37 SpawnPoint

Τα SpawnPoint τα τοποθέτησα εκτός οθόνης, δηλαδή εκτός οπτικού πεδίου του χρήστη, έτσι ώστε όταν πέφτουν να εμφανίζονται στη οθόνη. Στην συνέχεια δημιούργησα ένα ακόμα empty game object και το ονόμασα Spawner στο οποίο πέρασα το spawners scrip. Παρακάτω αναλύω το πρόγραμμα γραμμή προς γραμμή:



Εικόνα 38 Script για τα SpawnPoint



Εικόνα 39 Script για τα SpawnPoint

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

Αρχικά αυτή η ενότητα κώδικα περιλαμβάνει τις απαραίτητες δηλώσεις για τη χρήση βιβλιοθηκών και των κλάσεων που θα χρησιμοποιηθούν.

public class Spawner : MonoBehaviour

{

Δημιουργούμε την κλάση Spawner, η οποία είναι μια υποκλάση της κλάσης MonoBehaviour και υλοποιεί τη λειτουργία του αναπαραγωγέα.

public Transform[] spawnPoints;

public GameObject[] hazards;

Δημιουργούμε δύο δημόσιες μεταβλητές. Ο πίνακας spawnPoints περιέχει τις θέσεις αναπαραγωγής των εχθρών, ενώ ο πίνακας hazards περιέχει τα διάφορα εχθρικά αντικείμενα που μπορούν να αναπαραχθούν.

private float timeBtwSpawns;

public float startTimeBtwSpawns;

Δημιουργούμε δύο ιδιωτικές μεταβλητές. Η timeBtwSpawns αντιπροσωπεύει τον χρόνο που πρέπει να περάσει μέχρι την επόμενη αναπαραγωγή, ενώ η startTimeBtwSpawns είναι η αρχική τιμή αυτού του χρόνου.

public float minTimeBetweenSpawns;

```
public float decrease;
```

Οι μεταβλητές minTimeBetweenSpawns και decrease καθορίζουν τον ελάχιστο χρόνο μεταξύ των αναπαραγωγών και το ποσοστό μείωσης αυτού του χρόνου ανά κύκλο.

```
public GameObject player;
```

Η μεταβλητή player αναφέρεται στο παικτή του παιχνιδιού.

```
void Update()
{
   if(player != null)
   {
```

Η μέθοδος Update καλείται κατά τη διάρκεια κάθε καρέ του παιχνιδιού. Ελέγχουμε αν ο παίκτης υπάρχει (δεν έχει εξαφανιστεί).

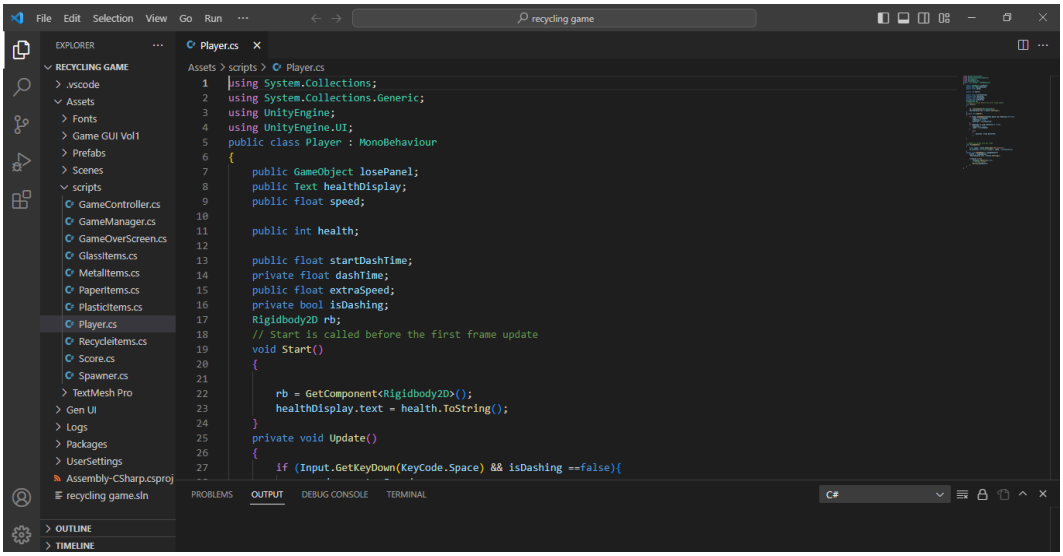
```
if(timeBtwSpawns <= 0)
{
   Transform randomSpawnPoint = spawnPoints[Random.Range(0, spawnPoints.Length)];
   GameObject randomHazard = hazards[Random.Range(0, hazards.Length)];
   Instantiate(randomHazard, randomSpawnPoint.position, Quaternion.identity);

   if(startTimeBtwSpawns > minTimeBetweenSpawns)
   {
      startTimeBtwSpawns -= decrease;
   }
   timeBtwSpawns = startTimeBtwSpawns;
}
else
{
   timeBtwSpawns -= Time.deltaTime;
}
```

Αν ο χρόνος μεταξύ των αναπαραγωγών (timeBtwSpawns) είναι μικρότερος ή ίσος με 0, εκτελούμε τις ενέργειες αναπαραγωγής. Επιλέγουμε τυχαία ένα σημείο αναπαραγωγής (randomSpawnPoint) από τον πίνακα spawnPoints και ένα τυχαίο εχθρικό αντικείμενο (randomHazard) από τον πίνακα hazards. Στη συνέχεια, χρησιμοποιούμε τη μέθοδο Instantiate για να δημιουργήσουμε ένα νέο εχθρικό αντικείμενο στη θέση του randomSpawnPoint. Αν ο χρόνος εκκίνησης μεταξύ των αναπαραγωγών (startTimeBtwSpawns) είναι μεγαλύτερος από τον ελάχιστο χρόνο μεταξύ των

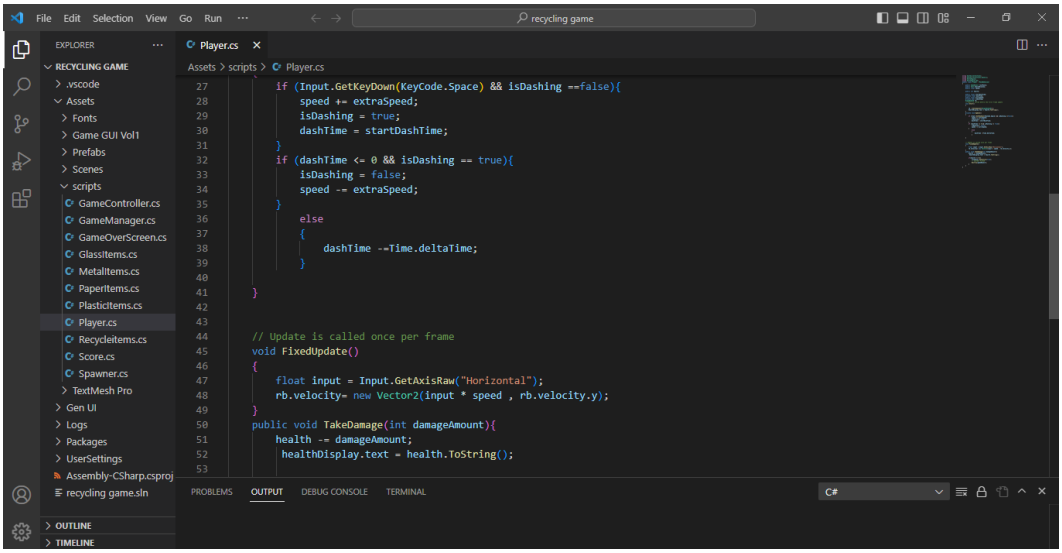
αναπαραγωγών (minTimeBetweenSpawns), μειώνουμε τον χρόνο εκκίνησης μεταξύ των αναπαραγωγών κατά το ποσοστό decrease. Τέλος, θέτουμε τον timeBtwSpawns ίσο με τον χρόνο εκκίνησης μεταξύ των αναπαραγωγών. Αν ο χρόνος μεταξύ των αναπαραγωγών δεν έχει μηδενιστεί, μειώνουμε τον timeBtwSpawns κατά τον χρόνο που έχει περάσει από τον προηγούμενο καρέ (Time.deltaTime). Με αυτόν τον τρόπο, ο κώδικας ελέγχει ανά τακτά χρονικά διαστήματα αν πρέπει να αναπαράγει ένα αντικείμενο, εναλλάσσοντας τα σημεία αναπαραγωγής και τα εχθρικά αντικείμενα που εμφανίζονται στο παιχνίδι. Επίσης, ο χρόνος μεταξύ των αναπαραγωγών μειώνεται με κάθε αναπαραγωγή, δημιουργώντας μια αυξανόμενη δυσκολία. Όλα αυτά εκτελούνται μόνο αν ο παίκτης είναι ενεργός (δεν έχει εξαφανιστεί από το παιχνίδι).

3.2.2 Script Player

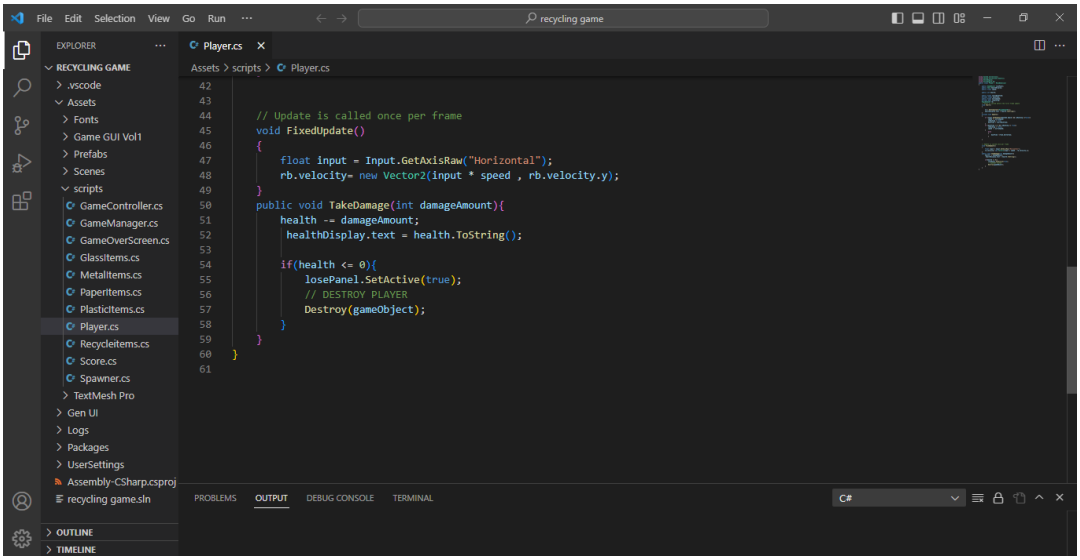
Το script κίνησης παίκτη που είναι προσκολλημένο στο αντικείμενο του παίκτη και μας επιτρέπει να κινούμε τον παίκτη με το input του χρήστη. Το script περιλαμβάνει και άλλες λειτουργίες οι οποίες προστέθηκαν αργότερα και θα εξηγηθούν παρακάτω.



Εικόνα 40 Script για το Player



Εικόνα 41 Script για το Player



Εικόνα 42 Script για το Player

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

Αυτή η γραμμή κώδικα δηλώνει τις απαραίτητες εισαγωγές για τη χρήση των κλάσεων από το περιβάλλον του Unity και το σύστημα UI.

public class Player : MonoBehaviour

{

public GameObject losePanel;

public Text healthDisplay;

```
public float speed;

public int health;

public float startDashTime;

private float dashTime;

public float extraSpeed;

private bool isDashing;

Rigidbody2D rb;


void Start()

{

    rb = GetComponent<Rigidbody2D>();

    healthDisplay.text = health.ToString();

}
```

Η γραμμή public class Player : MonoBehaviour δηλώνει την κλάση Player η οποία κληρονομεί από την κλάση MonoBehaviour του Unity. Αυτό επιτρέπει στον Player να χρησιμοποιείται ως σενάριο ελέγχου για τον χαρακτήρα του παίκτη στο παιχνίδι. Οι μεταβλητές public GameObject losePanel;, public Text healthDisplay;, public float speed;, public int health;, public float startDashTime;, public float extraSpeed;, δηλώνουν δημόσιες μεταβλητές που εμφανίζονται και μπορούν να ρυθμιστούν μέσω του περιβάλλοντος του Unity. Για παράδειγμα, η μεταβλητή losePanel αναφέρεται σε ένα παιχνίδι αντικείμενο (GameObject) που περιέχει το πάνελ "Game Over".

```
void Start()

  {

    rb = GetComponent<Rigidbody2D>();

    healthDisplay.text = health.ToString();

  }
```

Η μέθοδος Start() καλείται μία φορά στην αρχή του παιχνιδιού. Σε αυτή την περίπτωση, χρησιμοποιείται για να αρχικοποιήσει τον χαρακτήρα του παίκτη. Η μεταβλητή rb παίρνει το στοιχείο Rigidbody2D του παικτη με τη χρήση της μεθόδου GetComponent(). Η γραμμή healthDisplay.text = health.ToString(); αναθέτει την τρέχουσα τιμή της μεταβλητής health στον κείμενο (Text) του αντικειμένου (GameObject) healthDisplay, μετατρέποντας την σε συμβολοσειρά.

```
private void Update()

  {

    if (Input.GetKeyDown(KeyCode.Space) && isDashing == false)
```

```
        {
            speed += extraSpeed;
            isDashing = true;
            dashTime = startDashTime;
        }

        if (dashTime <= 0 && isDashing == true)
        {
            isDashing = false;
            speed -= extraSpeed;
        }
        else
        {
            dashTime -= Time.deltaTime;
        }
    }
```

Η μέθοδος Update() καλείται κάθε καρέ του παιχνιδιού. Εδώ ελέγχουμε αν ο παίκτης πατήσει το πλήκτρο Space και αν δεν είναι ήδη σε κατάσταση dash. Σε αυτή την περίπτωση, αυξάνουμε την ταχύτητα κατά extraSpeed, ενεργοποιούμε την κατάσταση dash (isDashing = true) και ορίζουμε τον χρόνο dash (dashTime) στην αρχική του τιμή (startDashTime). Έπειτα, ελέγχουμε αν ο χρόνος dash (dashTime) έχει περάσει κάτω από το 0 και αν η κατάσταση dash είναι true. Σε αυτή την περίπτωση, απενεργοποιούμε την κατάσταση dash (isDashing = false) και μειώνουμε την ταχύτητα κατά extraSpeed. Σε αντίθετη περίπτωση, μειώνουμε τον χρόνο dash (dashTime) κατά τη διάρκεια του παιχνιδιού.

```
void FixedUpdate()
{
    float input = Input.GetAxisRaw("Horizontal");
    rb.velocity = new Vector2(input * speed, rb.velocity.y);
}
```

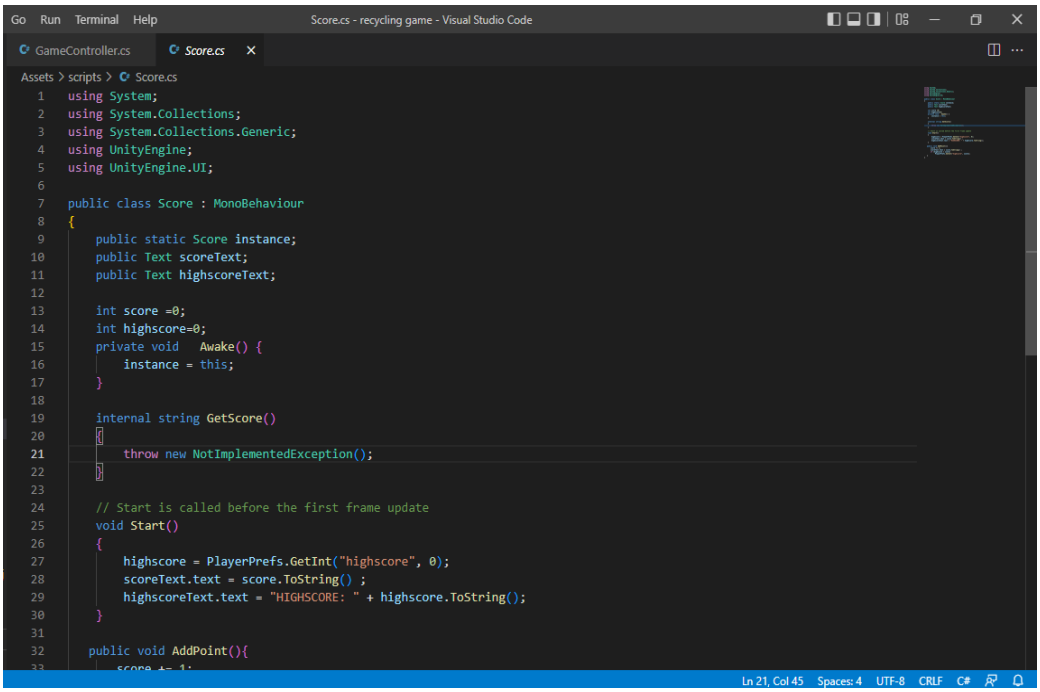
Η μέθοδος FixedUpdate() καλείται σε συγκεκριμένα χρονικά διαστήματα και χρησιμοποιείται για την ενημέρωση της φυσικής του παιχνιδιού. Εδώ, αποθηκεύουμε την είσοδο του χρήστη για τον οριζόντιο άξονα στη μεταβλητή input και αλλάζουμε την ταχύτητα του παίκτη (rb.velocity) ανάλογα με την είσοδο και την τρέχουσα ταχύτητα (speed).

```
public void TakeDamage(int damageAmount)
```

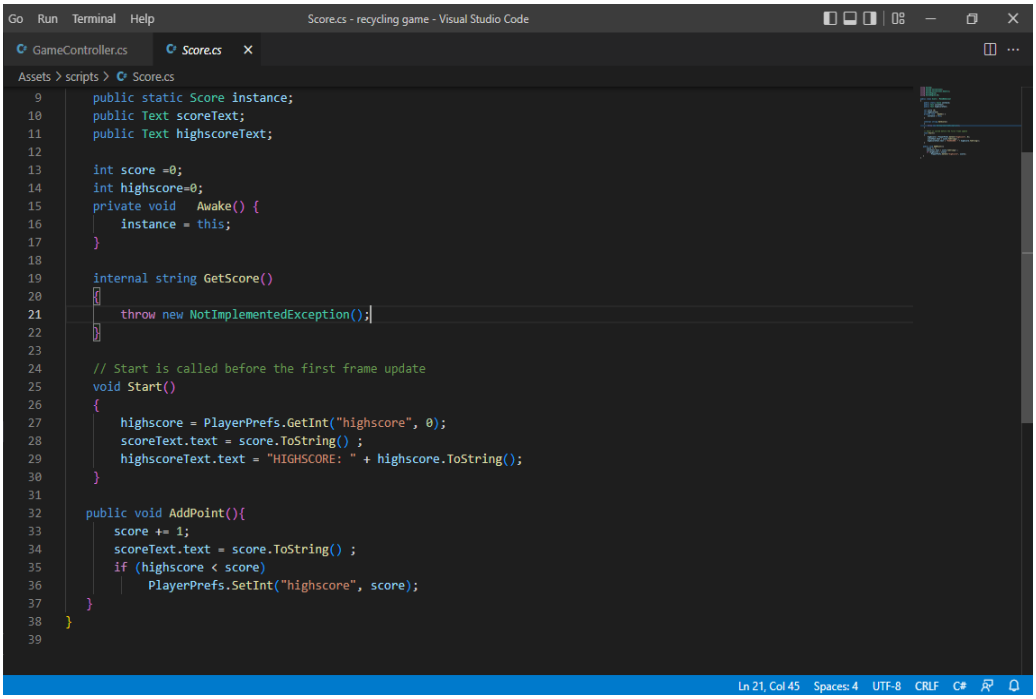
```
{
    health -= damageAmount;
    healthDisplay.text = health.ToString();

    if (health <= 0)
    {
        losePanel.SetActive(true);
        Destroy(gameObject);
    }
}
```

Η μέθοδος **TakeDamage()** χρησιμοποιείται για να μειώσει την υγεία του παίκτη κατά ένα ποσό (**damageAmount**). Μειώνουμε την υγεία του παίκτη (**health**) κατά το **damageAmount** και ενημερώνουμε το κείμενο της υγείας (**healthDisplay.text**) για να εμφανίζει τη νέα τιμή της υγείας. Αν η υγεία (health) είναι μικρότερη ή ίση του 0, ενεργοποιούμε το πάνελ ήττας (losePanel) για να εμφανιστεί στην οθόνη και κατόπιν καταστρέφουμε το αντικείμενο του παίκτη (gameObject) από το παιχνίδι.

3.2.3 Script του Score



Εικόνα 43 Script για το Score



Εικόνα 44 Script για το Score

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Score : MonoBehaviour
{
    public static Score instance;
    public Text scoreText;
    public Text highscoreText;

    int score = 0;
    int highscore = 0;

    private void Awake()
    {
        instance = this;
```

```
    }


    internal string GetScore()
    {
        throw new NotImplementedException();
    }
```

Σε αυτό το τμήμα κώδικα, ορίζουμε την κλάση Score. Έχουμε τις δημόσιες μεταβλητές scoreText και highscoreText, που αναφέρονται στα αντικείμενα κειμένου για την εμφάνιση των βαθμολογιών στην οθόνη. Έχουμε επίσης δύο ιδιωτικές μεταβλητές score και highscore για την αποθήκευση των τρεχουσών βαθμολογιών.

```
private void Awake()
{
    instance = this;
}
```

Η μέθοδος **Awake()** καλείται κατά την εκκίνηση του αντικειμένου και ορίζει την μεταβλητή **instance** στο τρέχον αντικείμενο της κλάσης. Αυτό μας επιτρέπει να έχουμε πρόσβαση στην κλάση **Score** από άλλες κλάσεις μέσω της **Score.instance**.

```
internal string GetScore()
{
    throw new NotImplementedException();
}
```

Υπάρχει μια μέθοδος **GetScore()** που επιστρέφει το σκορ ως string

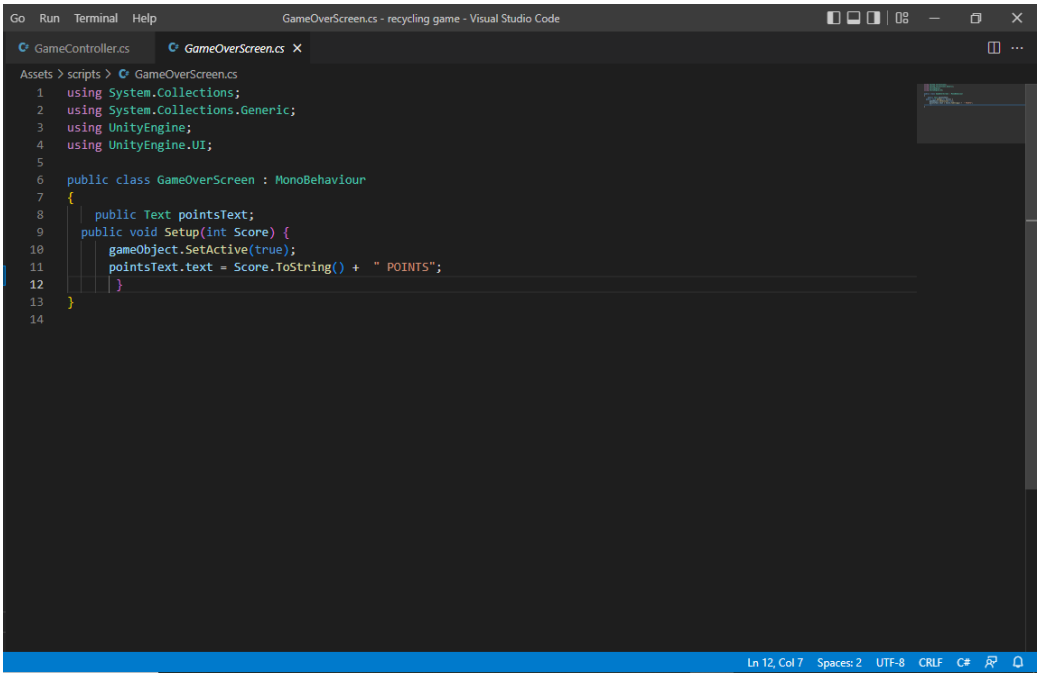
```
void Start()
{
    highscore = PlayerPrefs.GetInt("highscore", 0);
    scoreText.text = score.ToString();
    highscoreText.text = "HIGHSCORE: " + highscore.ToString();
}
```

Η μέθοδος **Start()** καλείται κατά την εκκίνηση του παιχνιδιού και ρυθμίζει τις αρχικές τιμές για τις βαθμολογίες. Χρησιμοποιούμε την **PlayerPrefs.GetInt()** για να ανακτήσουμε την αποθηκευμένη υψηλότερη βαθμολογία από τη μνήμη (με προεπιλεγμένη τιμή 0 αν δεν υπάρχει αποθηκευμένη βαθμολογία). Στη συνέχεια, ενημερώνουμε το κείμενο της τρέχουσας βαθμολογίας (**scoreText.text**) και το κείμενο της υψηλότερης βαθμολογίας (**highscoreText.text**) για να εμφανίζουν τις αντίστοιχες τιμές.

```
public void AddPoint()
{
   score += 1;
   scoreText.text = score.ToString();
   if (highscore < score)
      PlayerPrefs.SetInt("highscore", score);
}
```

Η μέθοδος **AddPoint()** καλείται για να προσθέσει έναν πόντο στην τρέχουσα βαθμολογία. Αυξάνουμε την μεταβλητή **score** κατά 1 και ενημερώνουμε το κείμενο της τρέχουσας βαθμολογίας (**scoreText.text**) για να εμφανίζει τη νέα τιμή. Επιπλέον, αν η τρέχουσα βαθμολογία είναι μεγαλύτερη από την υψηλότερη βαθμολογία (**highscore**), αποθηκεύουμε την τρέχουσα βαθμολογία στην μνήμη με την **PlayerPrefs.SetInt()**.

3.2.4 Script του GameOverScreen



Εικόνα 45 Scrip για το GameoverScreen

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class GameOverScreen : MonoBehaviour
{
```

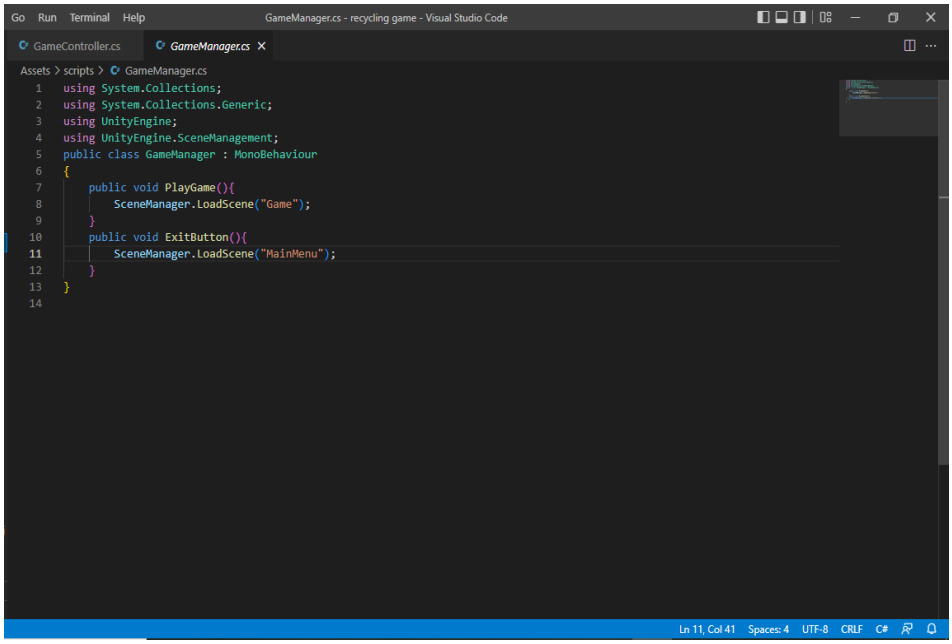
```
    public Text pointsText;


    public void Setup(int Score)
    {
        gameObject.SetActive(true);
        pointsText.text = Score.ToString() + " POINTS";
    }
}
```

Σε αυτό το τμήμα κώδικα, ορίζουμε την κλάση **GameOverScreen**. Έχουμε τη δημόσια μεταβλητή **pointsText**, η οποία αναφέρεται στο αντικείμενο κειμένου για την εμφάνιση των πόντων στην οθόνη.

```
public void Setup(int Score)
{
    gameObject.SetActive(true);
    pointsText.text = Score.ToString() + " POINTS";
}
```

Η μέθοδος **Setup()** καλείται για να ρυθμίσει την οθόνη Game Over. Όταν καλείται, η μέθοδος ενεργοποιεί το παιχνίδι αντικείμενο (**gameObject.SetActive(true)**) για να εμφανίσει την οθόνη Game Over. Στη συνέχεια, η μέθοδος ενημερώνει το κείμενο των πόντων (**pointsText.text**) με την τιμή του **Score** που έχει δοθεί, προσθέτοντας την λέξη "POINTS" μετά τον αριθμό των πόντων.

3.2.4 Script του GameManager



Εικόνα 46 Scrip για το GameManager

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;


public class GameManager : MonoBehaviour

{

   public void PlayGame()

   {

      SceneManager.LoadScene("Game");

   }


   public void ExitButton()

   {

      SceneManager.LoadScene("MainMenu");

   }

}
```

Σε αυτό το τμήμα κώδικα, έχουμε την κλάση GameManager. Αυτή η κλάση υπεύθυνη για τον έλεγχο του παιχνιδιού.

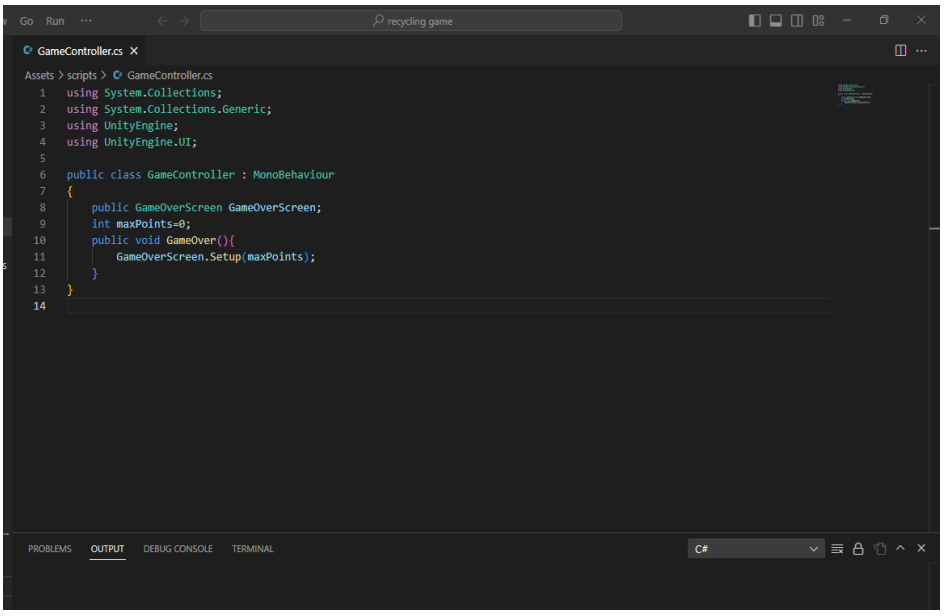
```
public void PlayGame()

{

   SceneManager.LoadScene("Game");

}
```

Η μέθοδος **PlayGame()** καλείται όταν ο παίκτης πατά το κουμπί για να ξεκινήσει το παιχνίδι. Αυτή η μέθοδος φορτώνει τη σκηνή με όνομα "Game", η οποία περιέχει το κύριο παιχνίδι

```
public void ExitButton()

{

   SceneManager.LoadScene("MainMenu");

}
```

Η μέθοδος ExitButton() καλείται όταν ο παίκτης πατά το κουμπί για έξοδο από το παιχνίδι. Αυτή η μέθοδος φορτώνει τη σκηνή με όνομα "MainMenu", η οποία περιέχει το κύριο μενού του παιχνιδιού.

3.2.5 Script του GameController



Εικόνα 47 Scrip για το GameController

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;


public class GameController : MonoBehaviour

{
```

```
    public GameOverScreen GameOverScreen;
    int maxPoints = 0;


    public void GameOver()
    {
        GameOverScreen.Setup(maxPoints);
    }
}
```

Σε αυτό το τμήμα κώδικα, ορίζουμε την κλάση GameController. Έχουμε τη δημόσια μεταβλητή GameOverScreen, η οποία αναφέρεται στο αντικείμενο που είναι υπεύθυνο για την οθόνη Game Over.

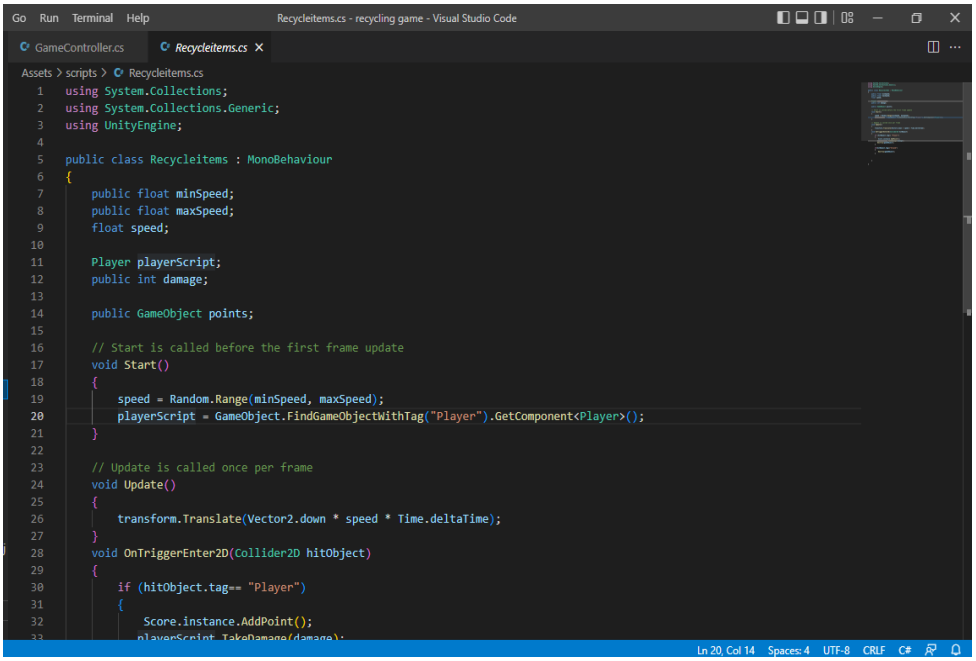
```
int maxPoints = 0;
```

Έχουμε μια ιδιωτική μεταβλητή maxPoints που αρχικοποιείται με την τιμή 0.

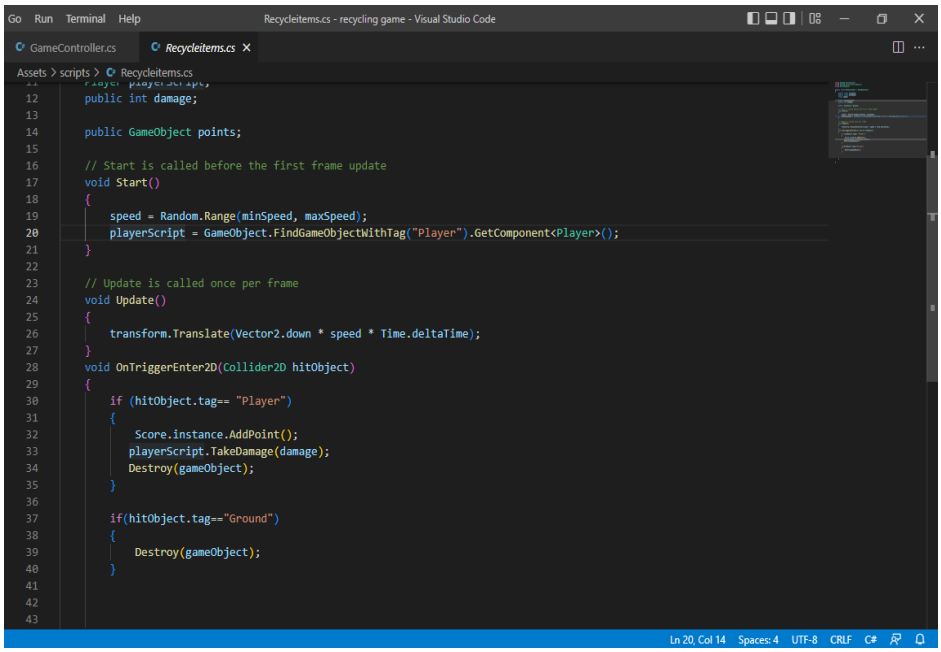
```
public void GameOver()
{
    GameOverScreen.Setup(maxPoints);
}
```

Η μέθοδος GameOver() καλείται όταν το παιχνίδι ολοκληρώνεται. Καλεί τη μέθοδο Setup() του αντικειμένου GameOverScreen, περνώντας την τιμή της μεταβλητής maxPoints. Αυτό σημαίνει ότι η οθόνη Game Over θα εμφανίσει το μέγιστο πόντο που έχει επιτευχθεί στο παιχνίδι.

3.2.6 Script των RecycleItems



Εικόνα 48 Scrip για τα RecycleItems



Εικόνα 49 Scrip για τα RecycleItems

Σε αυτόν τον κώδικα, η κλάση RecycleItems είναι υπεύθυνη για την κίνηση, τον έλεγχο των και τον αντίκτυπο που έχουν τα αντικείμενα που ανακυκλώνονται στο παιχνίδι.

public float minSpeed;

public float maxSpeed;

float speed;

Οι μεταβλητές minSpeed και maxSpeed καθορίζουν την ελάχιστη και μέγιστη ταχύτητα των αντικειμένων.

```
Player playerScript;

public int damage;

public GameObject points;
```

Οι μεταβλητές playerScript, damage και points αναφέρονται στον παίκτη και στο σκορ του παιχνιδιού.

```
void Start()

{

   speed = Random.Range(minSpeed, maxSpeed);

   playerScript = GameObject.FindGameObjectWithTag("Player").GetComponent<Player>();

}
```

Η μέθοδος Start() ορίζει την αρχική ταχύτητα του αντικειμένου και αναθέτει τον παίκτη στη μεταβλητή playerScript.

```
void Update()

{

   transform.Translate(Vector2.down * speed * Time.deltaTime);

}
```

Η μέθοδος Update() εκτελείται σε κάθε καρέ και είναι υπεύθυνη για την κίνηση του αντικειμένου προς τα κάτω.

```
void OnTriggerEnter2D(Collider2D hitObject)

{

   if (hitObject.tag == "Player")

   {

      Score.instance.AddPoint();

      playerScript.TakeDamage(damage);

      Destroy(gameObject);

   }


   if (hitObject.tag == "Ground")

   {

      Destroy(gameObject);

   }

}
```

Η μέθοδος OnTriggerEnter2D(Collider2D hitObject) εκτελείται όταν το αντικείμενο συγκρούεται με άλλο αντικείμενο. Αναλόγως το αντικείμενο που συγκρούστηκε με τον παίκτη, επηρεάζεται το σκορ η ζωή του πρωταγωνιστή και παράλληλα καταστρέφεται το αντικείμενο. Τέλος αν το αντικείμενο έρθει σε επαφή με το έδαφος πάλι καταστρέφεται.

3.3 Βασικός τρόπος δήλωσης και αρχικοποίησης των μεταβλητών

Αφού δημιουργήσει ο χρήστης ένα script στην Unity, το επόμενο βήμα είναι η δήλωση των μεταβλητών που θα χρησιμοποιηθούν στο script. Η δήλωση των μεταβλητών γίνεται συνήθως στην αρχή του script, αλλά μπορεί να προστεθούν και αργότερα, αν χρειάζεται να προστεθούν επιπλέον μεταβλητές. Εκτός από τους κλασικούς τύπους μεταβλητών όπως ακέραιοι (int), δεκαδικοί (float), λογική (bool) κ.λπ., στη βιβλιοθήκη της Unity υπάρχουν και οι τύποι μεταβλητών που αναπαριστούν αντικείμενα της Unity. Για παράδειγμα, αν θέλουμε να αποθηκεύσουμε ένα αντικείμενο που εκπροσωπεί τον παίκτη μας, μπορούμε να δηλώσουμε μια μεταβλητή με τον τύπο δεδομένων που η Unity χρησιμοποιεί για να αναφέρεται στα αντικείμενα, δηλαδή το GameObject. Έτσι, η δήλωση της μεταβλητής που αποθηκεύει το αντικείμενο του παίκτη μας θα μοιάζει κάπως έτσι:

```
public GameObject playerObject;
```

Με αυτόν τον τρόπο, έχουμε δηλώσει μια μεταβλητή με το όνομα "playerObject" η οποία μπορεί να αποθηκεύσει ένα GameObject, το οποίο αντιπροσωπεύει τον παίκτη.

3.4 Βιβλιοθήκες της Unity:

Όταν χρησιμοποιούμε την Unity για να δημιουργήσουμε παιχνίδια ή εφαρμογές, μπορούμε να επωφεληθούμε από τις έτοιμες συναρτήσεις και λειτουργίες που παρέχονται από τις βιβλιοθήκες της Unity. Αυτές οι βιβλιοθήκες παρέχουν ένα ευρύ φάσμα λειτουργιών για τη διαχείριση γραφικών, φυσικής μηχανής, ήχου, εισόδου, αντικειμένων και πολλά άλλα. Όταν έχουμε ένα συγκεκριμένο στόχο ή λειτουργία που θέλουμε να επιτύχουμε μέσω του κώδικα, μπορούμε να ανατρέξουμε στο εγχειρίδιο της Unity (Unity Manual) που παρέχεται στη διεύθυνσηhttps://docs.unity3d.com/Manual/index.html. Σε αυτό το εγχειρίδιο, μπορούμε να βρούμε πληροφορίες για όλες τις διαθέσιμες συναρτήσεις και λειτουργίες της Unity, καθώς και παραδείγματα κώδικα που δείχνουν πώς να τις χρησιμοποιήσουμε. Το εγχειρίδιο της Unity είναι ένα πολύτιμο εργαλείο για να αντλήσουμε πληροφορίες και να μάθουμε πώς να επιτύχουμε συγκεκριμένους στόχους μέσω του προγραμματισμού στην Unity. Μελετώντας το εγχειρίδιο και εξερευνώντας τις διάφορες συναρτήσεις και λειτουργίες που παρέχονται, μπορούμε να αναπτύξουμε πιο προηγμένες εφαρμογές και παιχνίδια με την χρήση της Unity. Στην παρούσα πτυχιακή διατέθηκαν κάποια παραδείγματα συναρτήσεων και λειτουργιών που χρησιμοποιήθηκαν συχνά.

GameObject.FindWithTag("Player"): Αυτή η συνάρτηση εντοπίζει το πρώτο αντικείμενο με την συγκεκριμένη ετικέτα "Player" και το αντιστοιχεί στη μεταβλητή που του έδωσα. Έτσι, μπορούμε να αναφερθούμε στο αντικείμενο που έχει αυτήν την ετικέτα και να εκτελέσουμε κώδικα ή να αλληλεπιδράσουμε με αυτό.

GetComponent<Collider2D>(): Αυτή η συνάρτηση εντοπίζει το στοιχείο του αντικειμένου (στην συγκεκριμένη περίπτωση το collider του), στο οποίο βρίσκεται το script που την περιέχει και το αποθηκεύει σε μια μεταβλητή.

void OnTriggerEnter2D(Collider2D other): Αυτή η συνάρτηση εκτελείται μόνο όταν ο collider ενός άλλου αντικειμένου στη σκηνή έρχεται σε επαφή με τον collider του αντικειμένου που περιέχει αυτό το σενάριο. Μπορούμε να χρησιμοποιήσουμε αυτήν τη συνάρτηση για να

εκτελέσουμε κώδικα όταν υπάρχει σύγκρουση μεταξύ δύο αντικειμένων και να προγραμματίσουμε αντίστοιχες ενέργειες.

void OnTriggerStay2D(Collider2D other): Αυτή η συνάρτηση είναι παρόμοια με την προηγούμενη, αλλά εκτελείται συνεχώς όσο οι colliders των αντικειμένων παραμένουν σε επαφή.

Επιπλέον, αν πρέπει να δημιουργήσουμε στοιχεία διεπαφής χρήστη (UI elements), πρέπει να προσθέσουμε την κατάλληλη βιβλιοθήκη στην αρχή του σεναρίου για να αναγνωρίζονται οι σχετικές συναρτήσεις και λειτουργίες.

Κεφάλαιο 4

4.1 Συμπεράσματα

Αρχικά το Unity είναι ένα ολοκληρωμένο πακέτο που παρέχει στους προγραμματιστές τη δυνατότητα να δημιουργήσουν τα δικά τους παιχνίδια 2D ή 3D με σχετικά απλό τρόπο. Αυτό το Game Engine έχει αποκτήσει δημοτικότητα κυρίως στη δημιουργία παιχνιδιών για Android και iOS. Για να μπορέσει όμως κάποιος καινούργιος χρήστης να το χρησιμοποιήσει θα αναγκαστεί να ακολουθήσει αρκετά tutorials. Ωστόσο, πολλά από αυτά τα tutorials αναφέρονταν σε λειτουργίες της μηχανής που είτε δεν υπάρχουν πλέον είτε λειτουργούν διαφορετικά στις νεότερες εκδόσεις. Επίσης παρά το γεγονός ότι το παιχνίδι φαίνεται μικρό απαιτήθηκε πολύς κόπος και χρόνος για την ολοκλήρωση όλων των μηχανισμών. Από την αρχή της ανάπτυξης, ήταν προφανές ότι απαιτείται εξοικείωση με τις θεμελιώδεις αρχές του αντικειμενοστραφούς προγραμματισμού, οι γλώσσες προγραμματισμού που υποστηρίζει το Unity και ο τρόπος λειτουργίας του το καθιστούν αυτό σαφές. Επιπλέον, γίνεται εμφανής η ανάγκη για γνώσεις σχετικά με αλγορίθμους και τον τρόπο εφαρμογής τους. Με αυτές τις γνώσεις, κάποιος μπορεί να αναλάβει τη δημιουργία ενός απλού παιχνιδιού, αλλά αυτό δεν είναι αρκετό για πιο περίπλοκα έργα. Το Unity είναι ένα εξαιρετικό εργαλείο για την ανάπτυξη παιχνιδιών, αλλά δεν είναι κατάλληλο για οποιονδήποτε και απαιτεί σημαντική δουλειά από τον προγραμματιστή.

Βιβλιογραφικές Αναφορές

1. A. Rehan, A. Mishra, A. Ahmed, M. Masood, M.A. Gondal, M. Sarfraz, T.M. Ansari(2018), "Primary Recycling: A Comprehensive Review" Waste Management, Volume 79, Pages 661-676,
2. Fabio M. Lamberti, Luis A. Román-Ramírez & Joseph Wood (2020), Recycling of Bioplastics: Routes and Benefits, Journal of Polymers and the Environment (2020) 28:2551–2571
3. Wright, R., Boorse, D., (2013). Περιβαλλοντική Επιστήμη: Προς ένα βιώσιμο μέλλον. Στο Θ. Πετανίδου, & Σ. Ριζοπούλου, (Επιμ.), Κοινοτικά Στερεά Απόβλητα: Διάθεση και Ανάκτηση (627-650).
4. Tyler Miller, JG. (2004). Περιβαλλοντικές επιστήμες. Στο Κ. Παυλόπουλος, (Επιμ.), Στερεά και Επικίνδυνα Απόβλητα (σελ. 364-390).
5. Ellen MacArthur Foundation. (2012). Towards the Circular Economy: Economic and business rationale for an accelerated transition.
6. Tyler Miller, JG. (1999). Βιώνοντας στο Περιβάλλον ΙΙ: Προβλήματα Περιβαλλοντικών Συστημάτων (Μ. Ταλαντοπούλου, Μτφρ.)
7. Drimili, Efi; Herrero-Martin, Ruth; Suardiaz-Muro, Juan; Zervas, Efthimios (2020). Public views and attitudes about municipal waste management: Empirical evidence from Athens, Greece. Waste Management & Research
8. "A historical review of recycling in ancient civilizations" - Miller, G. T., & Spoolman, S. E. (2012).
9. "The history and significance of recycling symbols" - Lardner, R. (2003).
10. "Waste Management Hierarchy: A Literature Review" - Pandey, A., et al. (2019).
11. Newman, J. (2012). Playing with video games: Genre, storytelling, and interaction. Routledge.
12. Salen, K., & Zimmerman, E. (2004). Rules of Play: Game Design Fundamentals. The MIT Press.
13. Gee, J.P. (2003). What Video Games Have to Teach Us About Learning and Literacy. Palgrave Macmillan.
14. Anderson, C.A., & Dill, K.E. (2000). Video games and aggressive thoughts, feelings, and behavior in the laboratory and in life. Journal of Personality and Social Psychology, 78(4), 772-790.
15. Liang, Y., & Leach, P. J. (2009). Visual Basic 2008: Programming at its best! McGraw-Hill.
16. Gosselin, D. (2017). Mastering Visual Studio 2017. Packt Publishing Ltd.
17. JetBrains. (2023). IntelliJ IDEA
18. Schell, J. (2008), The Art of Game Design, Elsevier Inc
19. Mitchell, A. & Savill-Smith, C. (2004) The use of computer and video games for learning: A review of the literature. Learning and Skills Development Agency
20. Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. Computers & Education, 59(2), 661-686. (DOI: 10.1016/j.compedu.2012.03.004)
21. H. Smith, 11 Dec 2014. [Online]. Available: https://prezi.com/w56f8xawwcyg/the-history-of-gameengines/
22. A History of the Unity Game Engine, John K. Haas: March 2014

23. Lukosek,G.2016. Learning C# by Developing Games with Unity 5.x. Packt Publishing
24. Caron, F. (2008, March 31). Unreal Engine 4 to “exclusively target” next-gen consoles. Retrieved from arstechnica: https://arstechnica.com/gaming/2008/03/unreal-engine-4-toexclusively-target-next-gen-consoles/
25. MITTRING, M. 2007. Finding Next Gen – CryEngine 2
26. Daniel A. Chappell, Tyler Jewell, Java Web Services, O’Reilly Media, 2002.
27. D. Williams, (2007): C++ Portability Guide -