



# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

## **ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

### **ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

#### **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**3D παιχνίδι ρόλων με στόχο την ψυχαγωγία και παροχή ανάλογων  
βοηθειών**

**ΧΡΗΣΤΟΣ ΣΑΜΠΑΚΙΔΗΣ  
Α.Μ. 19390207**

**Εισηγητής: Δρ. Χρήστος Τρούσσας, Επ. Καθηγητής**



**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**3D παιχνίδι ρόλων για ψυχαγωγία με παροχή βοήθειας**

**Χρήστος Σαμπακίδης  
Α.Μ. 19390207**

**Εισηγητής:**

**Δρ. Χρήστος Τρούσσας, Επ. Καθηγητής**

**Εξεταστική Επιτροπή:**

**Χρήστος Τρούσσας, Επ. Καθηγητής**

**Φοίβος Μυλωνάς, Αν. Καθηγητής**

**Ακριβή Κρούσκα, ΕΔΙΠ**

**Ημερομηνία εξέτασης: Σεπτέμβριος 2024**



## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η κάτωθι υπογεγραμμένος Σαμπακίδης Χρήστος του Νικολάου, με αριθμό μητρώου 19390207 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και 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».

«Βεβαιώνω ότι είμαι συγγραφέας της παρούσας διπλωματικής εργασίας και ότι έχω αναφέρει ή παραπέμπει σε αυτή, ρητά και συγκεκριμένα, όλες τις πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, προτάσεων ή λέξεων, είτε αυτές μεταφέρονται επακριβώς (στο πρωτότυπο ή μεταφρασμένες) είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για την συγκεκριμένη διπλωματική εργασία»

Ο Δηλών  
Σαμπακίδης Χρήστος





## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα διπλωματική εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, στο ενδιαφέρον γνωστικό αντικείμενο της ανάπτυξης παιχνιδιών. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου Χρήστος Τρούσσας, τον οποίο θα ήθελα να ευχαριστήσω.





## **ΠΕΡΙΛΗΨΗ**

Η παρούσα διπλωματική εργασία ασχολείται με την ανάπτυξη ενός τρισδιάστατου διαδικτυακού παιχνιδιού ρόλων το οποίο παρέχει και τις κατάλληλες συμβουλές ώστε ο παίκτης να αναπτύξει ομαλά τον χαρακτήρα του στο παιχνίδι. Το παιχνίδι το οποίο αναπτύχθηκε εστιάζει στην λογική του «ανοικτού κόσμου» και στηρίζεται βαριά στην ύπαρξη κοινότητας η οποία είναι απαραίτητη για να είναι το παιχνίδι ενδιαφέρον και οι στόχοι να έχουν νόημα.

## **ABSTRACT**

This thesis focuses on the development of a three-dimensional online role-playing game that also provides appropriate guidance to help the player smoothly develop their character within the game. The game that was developed emphasizes an "open world" concept and heavily relies on the existence of a community, which is essential to making the game engaging and giving the objectives meaning.

# ΠΕΡΙΕΧΟΜΕΝΑ

<b>1. ΕΙΣΑΓΩΓΗ .....</b>	<b>1</b>
1.1 Περιγραφή του αντικειμένου της διπλωματικής εργασίας.....	1
1.2 Ιστορική αναδρομή.....	1
1.3 Αναδρομή εργασίας .....	3
<b>2. ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ.....</b>	<b>5</b>
2.1 Βασικά στοιχεία και συστήματα του παιχνιδιού.....	5
2.2 Τεχνικές ανάπτυξης .....	12
2.3 Τρόποι προσκόμισης κέρδους.....	12
<b>3. ΥΛΟΠΟΙΗΣΗ ΠΑΙΧΝΙΔΙΟΥ .....</b>	<b>14</b>
3.1 Σκοπός του παιχνιδιού .....	14
3.2 Διαχωρισμός ανάπτυξης .....	15
3.3 Σχεδιασμός του γραφικού περιβάλλοντος .....	15
3.4 Ανάπτυξη του γραφικού περιβάλλοντος .....	17
3.5 Σχεδιασμός και ανάπτυξη της λειτουργικότητας .....	22
3.5.1 Σύστημα διαχείρισης δεδομένων .....	23
3.5.2 Σύστημα διαχείρισης αντικειμένων.....	25
3.5.3 Σύστημα ανταλλαγής αντικειμένων .....	28
3.5.4 Συστήματα δημιουργίας αντικειμένων .....	28
3.5.5 Σύστημα απόκτησης πόρων .....	30
3.5.6 Σύστημα υποβοήθησης παικτών .....	30
3.5.7 Σύστημα Μάχης .....	32
<b>4. ΤΡΟΠΟΙ ΜΕ ΤΟΥΣ ΟΠΟΙΟΥΣ ΠΑΙΖΕΤΑΙ ΤΟ ΠΑΙΧΝΙΔΙ .....</b>	<b>35</b>
4.1 Βασικός τρόπος παιχνιδιού.....	35
4.3 Ο ρόλος των Λόρδων.....	36
4.3 Έμποροι ή συλλέκτες.....	37
4.4 Λοιποί παίκτες.....	37
4.5 Διαφορές μεταξύ διαφορετικών ειδών παικτών .....	37
4.6 Κακόβουλοι παίκτες .....	38

<b>5. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ .....</b>	<b>40</b>
5.1 Σύστημα δημιουργίας κάστρων.....	41
5.2 Περισσότερες γλώσσες διεπαφής .....	41
<b>6. ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>42</b>

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

<b>Εικόνα 1.1:</b> Περιβάλλον Roblox Studio.....	<b>3</b>
<b>Εικόνα 2.1:</b> Μοντέλο προς υλοποίηση σε wireframe view στο πρόγραμμα Blender .....	<b>5</b>
<b>Εικόνα 2.2:</b> Μοντέλο υπό το στάδιο επεξεργασίας υφών και texturing στο Substance Painter.....	<b>6</b>
<b>Εικόνα 2.3:</b> Μοντέλο υπό το στάδιο επεξεργασίας υφών και texturing στο Substance Painter.....	<b>7</b>
<b>Εικόνα 2.4:</b> Μέρος του εικονικού κόσμου του παιχνιδιού .....	<b>8</b>
<b>Εικόνα 2.5:</b> Το σύστημα αποθήκευσης αντικειμένων του παιχνιδιού .....	<b>10</b>
<b>Εικόνα 3.1:</b> Μέρος του εικονικού κόσμου του παιχνιδιού .....	<b>16</b>
<b>Εικόνα 3.2:</b> Μέρος των ορυχείων του παιχνιδιού .....	<b>16</b>
<b>Εικόνα 3.3:</b> Εισαγωγή MeshPart στο Roblox Studio .....	<b>17</b>
<b>Εικόνα 3.4:</b> Παράθυρο explorer και properties του Roblox Studio.....	<b>18</b>
<b>Εικόνα 3.5:</b> Παράθυρο Lighting του Roblox Studio .....	<b>19</b>
<b>Εικόνα 3.6:</b> Αναπαράσταση SkyBox .....	<b>20</b>
<b>Εικόνα 3.7:</b> Παράθυρο ρυθμίσεων SkyBox στο Roblox Studio .....	<b>21</b>
<b>Εικόνα 3.8:</b> Όρια χάρτη .....	<b>22</b>
<b>Εικόνα 3.9:</b> Μερική διάταξη των scripts του server.....	<b>22</b>
<b>Εικόνα 3.10:</b> Παράδειγμα attributes ενός αντικειμένου .....	<b>26</b>
<b>Εικόνα 3.11:</b> Διεπαφή χρήστη των καταστημάτων .....	<b>29</b>
<b>Εικόνα 3.12:</b> Διεπαφή χρήστη του Crafting .....	<b>30</b>
<b>Εικόνα 3.13:</b> Μέρος του Tutorial.....	<b>31</b>
<b>Εικόνα 3.14:</b> Επίθεση 1 .....	<b>32</b>
<b>Εικόνα 3.15:</b> Επίθεση 2.....	<b>33</b>
<b>Εικόνα 3.16:</b> Επίθεση 3.....	<b>33</b>



## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

<b>Σχήμα 2.1:</b> Διάγραμμα επικοινωνίας Client-Server.....	<b>9</b>
<b>Σχήμα 2.1:</b> Στατιστικά συναλλαγών του παιχνιδιού .....	<b>13</b>

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

**API** Application Programming Interface

**GUID** Globally Unique Identifier

# ΚΕΦΑΛΑΙΟ 1

## ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο αναλύεται το αντικείμενο της διπλωματικής εργασίας και γίνεται μια ιστορική αναδρομή γύρω από τις μεθόδους που έχουν χρησιμοποιηθεί για την ανάπτυξη του παιχνιδιού και παρουσιάζονται στην παρούσα εργασία. Επιπλέον, γίνεται αναφορά στην πλατφόρμα και την γλώσσα προγραμματισμού στην οποία αναπτύχθηκε το παιχνίδι καθώς και στην εξέλιξή της με την πάροδο των χρόνων. Τέλος γίνεται αναφορά στον όρο παιχνίδια ρόλων.

### 1.1 Περιγραφή του αντικείμενου της διπλωματικής εργασίας

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η καταγραφή των βημάτων που έγιναν για την ανάπτυξη του παιχνιδιού καθώς και ανάλυση των διαφόρων προβλημάτων που υπάρχουν στην ανάπτυξη ενός διαδικτυακού παιχνιδιού και πως αυτά επιλύθηκαν. Επιπλέον περιγράφονται οι τρόποι με τους οποίους το παιχνίδι ενημερώνει τον χρήστη ανάλογα με τις ενέργειες τις οποίες παίρνει ώστε να βοηθήσει στην καλύτερη εξέλιξη του παίκτη. Η δημιουργία του παιχνιδιού έγινε με την χρήση της πλατφόρμας roblox και της γλώσσας luau ενώ τα μοντέλα αναπτύχθηκαν στο blender τα οποία ωστόσο δεν αποτελούν αντικείμενο μελέτης της παρούσας διπλωματικής και για αυτό δεν θα γίνει εκτενής ανάλυση σε αυτά..

### 1.2 Ιστορική αναδρομή

Το roblox είναι μία διαδικτυακή πλατφόρμα παιχνιδιών η οποία αναπτύχθηκε το 2004 από τους David Baszucki και Erik Cassel με τον Baszucki να είναι ο κύριος μέτοχος της εταιρείας ο οποίος το 2020 κατείχε το 13% το οποίο υπολογίζεται ότι αντιστοιχεί σε \$470 εκατομμύρια Αμερικανικά δολάρια . Αρχικά δόθηκε στο κοινό μία Beta έκδοση της πλατφόρμας η οποία διήρκεσε 2 χρόνια έως την 1<sup>η</sup> Σεπτεμβρίου του 2006 όταν και έγινε η επίσημη κυκλοφορία της πλατφόρμας. Από τότε η πλατφόρμα έχει εξελιχθεί ραγδαία και εκτιμάται ότι έχει καθημερινά 77.7 εκατομμύρια ενεργούς χρήστες. Να σημειωθεί ότι όταν γίνεται αναφορά σε στατιστικά στοιχεία τα οποία αφορούν την πλατφόρμα είναι συγκεντρωτικά στοιχεία από όλα τα παιχνίδια τα οποία έχουν αναπτυχθεί σε αυτήν. Η κεντρική ιδέα του roblox ήταν να δοθούν στους χρήστες όλα τα εργαλεία ώστε να μπορέσουν να αναπτύξουν τα παιχνίδια τα οποία αυτοί θέλουν στο οικοσύστημά τους ενώ παράλληλα η εταιρεία θα προσκομίζει ένα ποσοστό των κερδών αυτών των παιχνιδιών τα οποία προκύπτουν από διάφορες μικρο-συναλλαγές μέσα στα παιχνίδια, αγοροπωλησίες μοντέλων, αξεσουάρ για τους χαρακτήρες κ.α. Επιπροσθέτως, παρόλο που η πλατφόρμα μπορεί να παίρνει ένα ποσοστό των κερδών έχει και αρκετές παροχές όπως δωρεάν απεριόριστους Server, δωρεάν αποθήκευση στις βάσεις δεδομένων τους και δωρεάν πρόσβαση σε πολλαπλά εργαλεία τα οποία βοηθούν όχι μόνο στην ανάπτυξη αλλά και στην κατανόηση των εισροών ενός παιχνιδιού ώστε οι χρήστες οι οποίοι αναπτύσσουν παιχνίδια σε αυτήν να γνωρίζουν πού να εστιάσουν.

Το roblox αναπτύχθηκε στην γλώσσα προγραμματισμού lua σε συνδυασμό με την C++ ενώ οι χρήστες για τα παιχνίδια τους χρησιμοποιούν την luau η οποία είναι μία open source παραλλαγή της lua με ενσωματωμένο το API του roblox. Η luau είναι μία interpreted γλώσσα προγραμματισμού η οποία παρέχει όλες τις λειτουργίες μίας κανονικής γλώσσας προγραμματισμού με την εξαίρεση πρόσβασης στην μνήμη για λόγους που έχουν να κάνουν με την ασφάλεια. Παρόλο που η επιλογή μίας interpreted γλώσσας μπορεί να είναι περιέργη για μία πλατφόρμα ανάπτυξης παιχνιδιών καθώς μπορεί



να θεωρηθεί αρκετά αργή για εφαρμογές πραγματικού χρόνου, ήταν ωστόσο κομβικής σημασίας για την επιτυχία της πλατφόρμας. Μια interpreted γλώσσα παρέχει αμεσότερο debugging στους προγραμματιστές και είναι πολύ πιο κατανοητή για έναν αρχάριο προγραμματιστή το οποίο ήταν και το αρχικό κοινό στο οποίο απευθυνόταν το roblox. Με την πάροδο των χρόνων έχουν γίνει διάφορες αναβαθμίσεις στην γλώσσα και στο API οι οποίες αφορούν τόσο την απόδοση όσο και την προσθήκη νέων δυνατοτήτων. Μία ημερομηνία ορόσημο για αυτές τις αναβαθμίσεις είναι η 12<sup>η</sup> Δεκεμβρίου 2013 όταν και ενεργοποιήθηκε η δυνατότητα χρήσης http request το οποίο επέτρεπε στους προγραμματιστές να μπορούν πλέον να κάνουν κλήσεις σε εξωτερικά API και έτσι τους δόθηκε η δυνατότητα να συνδέσουν τα παιχνίδια τους με διάφορα εξωτερικά εργαλεία, βάσεις δεδομένων, bots και πολλά άλλα τα οποία βοήθησαν στην ραγδαία ανάπτυξη της πλατφόρμας. Ένα αξιοσημείωτο παράδειγμα είναι ότι πλέον οι προγραμματιστές δεν χρειάζονταν να εξαρτώνται από το roblox για την αποθήκευση των δεδομένων των παικτών καθώς πλέον μπορούσαν να χρησιμοποιούν εξωτερικές βάσεις όπως MongoDB<sup>1</sup>, MySQL<sup>2</sup>, να συνδέσουν το παιχνίδι με εξωτερικές υπηρεσίες για αμεσότερο Feedback όπως Google Forms καθώς και καταγραφή Logs<sup>3</sup> αλλά και αμεσότερου escalation σε περίπτωση που κάποια υπηρεσία του παιχνιδιού παρουσιάσει Downtime.

Όλα τα παραπάνω σε συνδυασμό με περαιτέρω αναβαθμίσεις και βελτίωση των υπηρεσιών τους βοήθησαν το roblox στο να μετατραπεί σε μία τέλεια πλατφόρμα ανάπτυξης indie<sup>4</sup> παιχνιδιών και να επιτρέψει σε μικρές ομάδες να αναπτύξουν παιχνίδια μεγάλης κλίμακας τα οποία σε άλλες πλατφόρμες θα απαιτούσαν σημαντικά μεγαλύτερο χρονικό διάστημα για την ανάπτυξή τους. Τέλος, παρόλο που η πλατφόρμα αρχικά απευθυνόταν σε indie προγραμματιστές και παιχνίδια μικρής κλίμακας, η εκρηκτική άνοδος των χρηστών γρήγορα οδήγησε στην δημιουργία game studios τα οποία δραστηριοποιούνταν αποκλειστικά στην πλατφόρμα αυτήν και ως αποτέλεσμα στην δημιουργία παιχνιδιών μεγάλης κλίμακας.

Ο όρος παιχνίδι ρόλων αφορά ένα παιχνίδι στο οποίο οι παίκτες μπορούν να αναλάβουν διάφορους ρόλους και να δημιουργήσουν τις δικές τους ιστορίες ενώ παράλληλα αλληλεπιδρούν με τον κόσμο του παιχνιδιού, άλλους παίκτες και NPC<sup>5</sup>. Ο στόχος ενός παιχνιδιού ρόλων συνήθως δεν είναι προκαθορισμένος καθώς οι κατευθυντήριες γραμμές δίνονται από την κοινότητα του παιχνιδιού χωρίς ωστόσο αυτό να σημαίνει ότι ο δημιουργός του παιχνιδιού δεν μπορεί να επηρεάσει τους στόχους. Ο δημιουργός μπορεί να θέσει έμμεσα διάφορους στόχους όπως είναι η απόκτηση μεγάλου πλούτου ή η απόκτηση μίας θέσης εξουσία στο παιχνίδι. Επίσης μπορεί να κατευθύνει, έμμεσα τους παίκτες παρέχοντας συμβουλές τις οποίες ο παίκτης μπορεί να διαλέξει αν θα ακολουθήσει. Ένα παιχνίδι ρόλων συνήθως έχει μεγάλο βάθος καθώς οι παίκτες ενθαρρύνονται να δημιουργήσουν τις δικές τους ιστορίες με βάση τον κόσμο του παιχνιδιού. Συνεπώς τα παιχνίδια αυτά ενθαρρύνουν την κοινωνικοποίηση καθώς στα περισσότερα παιχνίδια τέτοιου τύπου η δημιουργία σχέσεων με άλλους παίκτες είναι απαραίτητη για την πρόοδο στο παιχνίδι. Τα παιχνίδια ρόλων μπορούν να διαχωριστούν σε διάφορες υποκατηγορίες όπως φαντασία, μοντέρνα, μεσαιωνικά, steam-punk και ιστορικά. Το παιχνίδι το οποίο παρουσιάζεται στην παρούσα εργασία αφορά μεσαιωνικό παιχνίδι ρόλων.

---

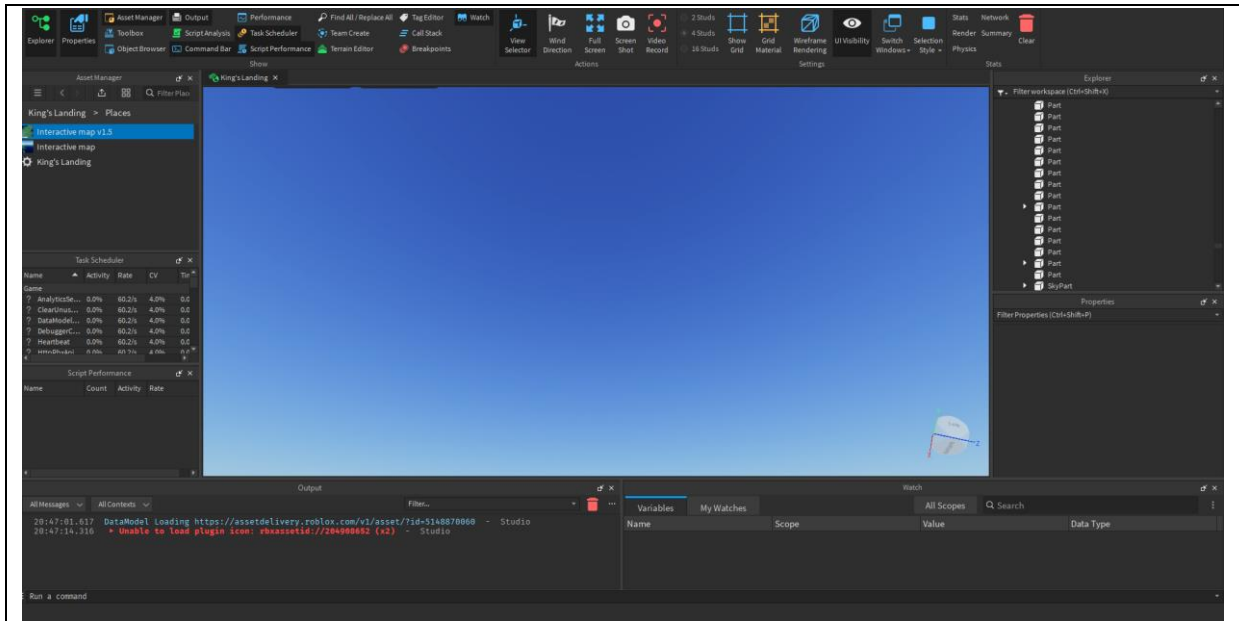
<sup>1</sup> Μη σχεσιακή βάση δεδομένων

<sup>2</sup> Σχεσιακή βάση δεδομένων

<sup>3</sup> Αρχεία ιστορικού σημαντικών γεγονότων που συμβαίνουν στο παιχνίδι

<sup>4</sup> Παιχνίδια μικρής κλίμακας ή παιχνίδια από μικρή ομάδα ανάπτυξης

<sup>5</sup> Non Playable Character ή αλλιώς τεχνητή νοῦμοσύνη με την οποία αλληλεπιδρούν οι παίκτες.



Εικόνα 1.1

### 1.3 Αναδρομή εργασίας

Η παρούσα διπλωματική εργασία επικεντρώνεται στην δημιουργία ενός 3D παιχνιδιού ρόλων το οποίο διαδραματίζεται στον μεσαίωνα και παρέχει υποδείξεις προς τον παίκτη οι οποίες εξαρτώνται από τις πράξεις του καθώς και τον χρόνο που του παίρνει ώστε να φτάσει σε ορισμένες τιμές στόχους μέσα σε ορισμένα χρονικά διαστήματα. Επιπλέον υπαρκτές είναι και μη δυναμικές υποδείξεις στόχος των οποίων είναι να βάλουν τον παίκτη στο κλίμα του παιχνιδιού και να δίνουν κατευθυντήριες γραμμές για τους στόχους. Ο κύριος στόχος του παιχνιδιού τον οποίο φαίνεται να προτιμούν οι περισσότεροι παίκτες είναι η απόκτηση της βασιλείας το οποίο γίνεται με συγκεκριμένα βήματα τα οποία θα αναλυθούν παρακάτω χωρίς αυτό όμως να περιορίζει την εξέλιξη των παικτών με οποιονδήποτε τρόπο. Το παιχνίδι αυτό δημιουργήθηκε στην πλατφόρμα roblox με την γλώσσα που παρέχει η πλατφόρμα την lua. Επίσης για την αποθήκευση μέρους των δεδομένων χρησιμοποιήθηκε και η πλατφόρμα Trello<sup>6</sup>. Τέλος χρησιμοποιήθηκαν διάφορες open source βιβλιοθήκες για την βελτίωση της αξιοπιστίας των βάσεων και την διευκόλυνση ανάπτυξης του Frontend.

Στην πρώτη ενότητα παρουσιάζεται το θεωρητικό υπόβαθρο της εργασίας στο οποίο αναλύονται τα διάφορα συστήματα του παιχνιδιού, οι μηχανισμοί προσκόμισης κέρδους που υπάρχουν στο παιχνίδι καθώς κάθε παιχνίδι έχει ως στόχο και την δημιουργία κέρδους, οι μηχανισμοί για την διατήρηση των παικτών καθώς και τα βασικά στοιχεία κάθε παιχνιδιού. Τέλος θα παρουσιαστούν μερικές έρευνες πάνω στις οποίες βασίστηκαν πολλές από τις αποφάσεις που πάρθηκαν κατά την ανάπτυξη.

Στην ενότητα της μεθοδολογίας περιγράφονται οι τρόποι ανάπτυξης του παιχνιδιού, οι τεχνικές ανάπτυξης του κώδικα, η αλληλεπίδραση Client-Server καθώς και θα παρουσιαστούν αποσπάσματα του κώδικα. Θα γίνει ειδική αναφορά και ανάλυση στο θέμα της αλληλεπίδρασης Client-Server καθώς και στα θέματα ασφαλείας τα οποία προκύπτουν με τους τρόπους αντιμετώπισής τους. Επιπροσθέτως, περιγράφεται λεπτομερώς η εξέλιξη της αρχιτεκτονικής και οι λόγοι για τους οποίους πάρθηκαν συγκεκριμένες αποφάσεις όσον αφορά την ανάπτυξη διαφόρων συστημάτων του παιχνιδιού. Παράλληλα γίνεται παρουσίαση αποσπάσματος του διαγράμματος κλάσεων σε υψηλό επίπεδο καθώς

<sup>6</sup> Πλατφόρμα διαχείρισης εργασιών

και των διαγραμμάτων της βάσης. Να σημειωθεί ότι επειδή η βάση χρησιμοποιεί το API του roblox το οποίο ακολουθεί μορφή Dictionary για την αποθήκευση των δεδομένων θα γίνει παρουσίαση τόσο του Dictionary όσο και προσπάθειας προσαρμογής σε Database Schema Diagram. Θα αναλυθούν οι βιβλιοθήκες οι οποίες χρησιμοποιήθηκαν, οι λόγοι για τους οποίους χρησιμοποιήθηκαν και τα οφέλη τους. Τέλος θα γίνει παρουσίαση του παιχνιδιού και διαφόρων δεδομένων που αποκτήθηκαν κατά την περίοδο δοκιμαστικής λειτουργίας του.

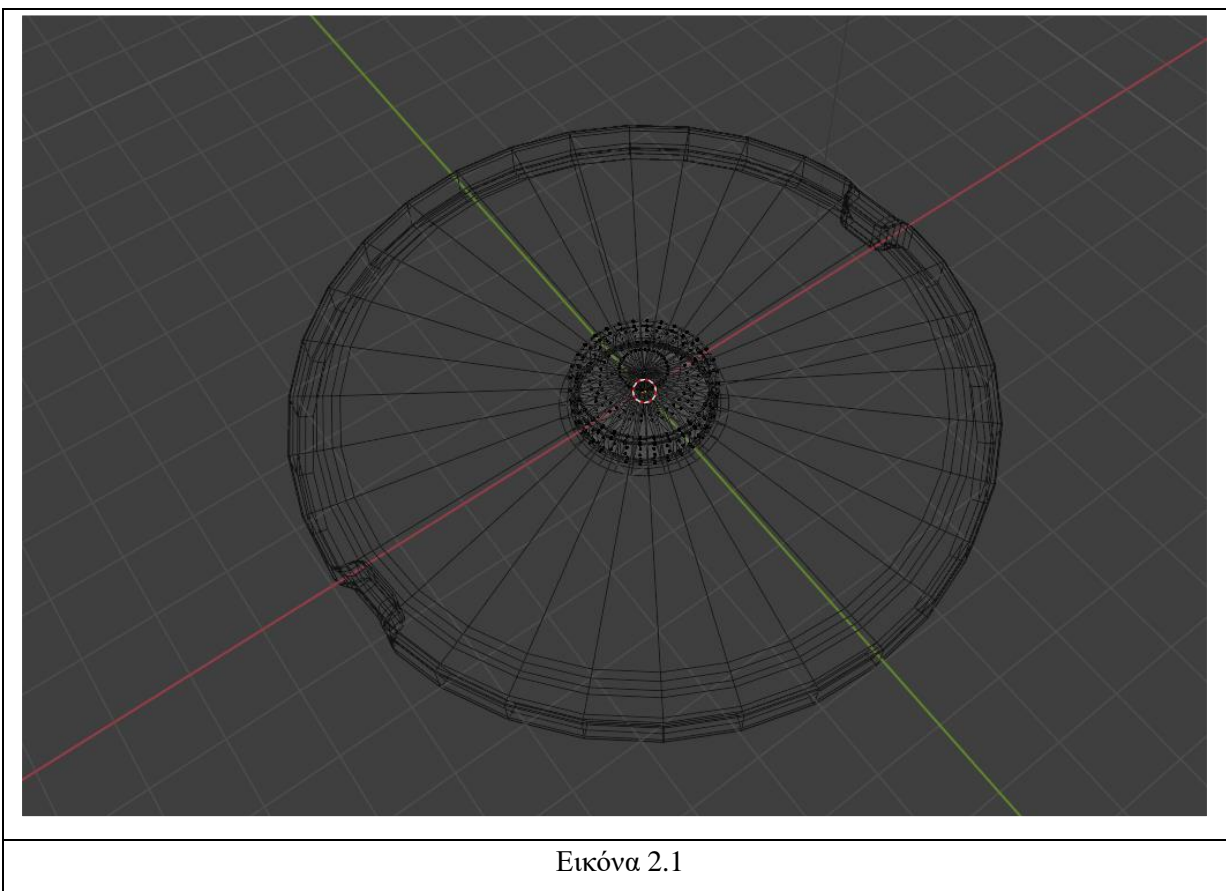
## ΚΕΦΑΛΑΙΟ 2

### ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

#### 2.1 Βασικά στοιχεία και συστήματα του παιχνιδιού

Έπειτα από έρευνα σε διάφορα άρθρα και βίντεο εκπαιδευτικού περιεχομένου που αφορούν την ανάπτυξη παιχνιδιών και εκτενής μελέτη του documentation που παρέχει η πλατφόρμα roblox, προέκυψε ότι κάθε παιχνίδι αποτελείται από 2 εξίσου σημαντικά μέρη. Τον κόσμο του παιχνιδιού δηλαδή το γραφικό περιβάλλον και την λειτουργικότητα.

Κάθε 3D παιχνίδι είναι αναγκαίο να έχει 3D μοντέλα από τα οποία αποτελείται ο κόσμος του παιχνιδιού ο οποίος συνήθως συνοδεύεται από ένα έδαφος “terrain” πάνω στο οποίο κινείται ο παίκτης. Τα μοντέλα αυτά σχεδιάζονται πρώτα σε εξωτερικά προγράμματα όπως για παράδειγμα είναι το Blender<sup>7</sup>. Παρόλο που το roblox δίνει την δυνατότητα δημιουργίας μοντέλων μέσα από τον editor του δεν παρέχει την απαιτούμενη ποιότητα και τα μοντέλα που παράγει αποτελούνται από πολλά Triangles<sup>8</sup> με αποτέλεσμα την δραστική μείωση της απόδοσης



<sup>7</sup> Πρόγραμμα επεξεργασίας εικονικού περιβάλλοντος

<sup>8</sup> Τρίγωνα από τα οποία η κάρτα επεξεργασίας γραφικών σχηματίζει την τελική εικόνα.

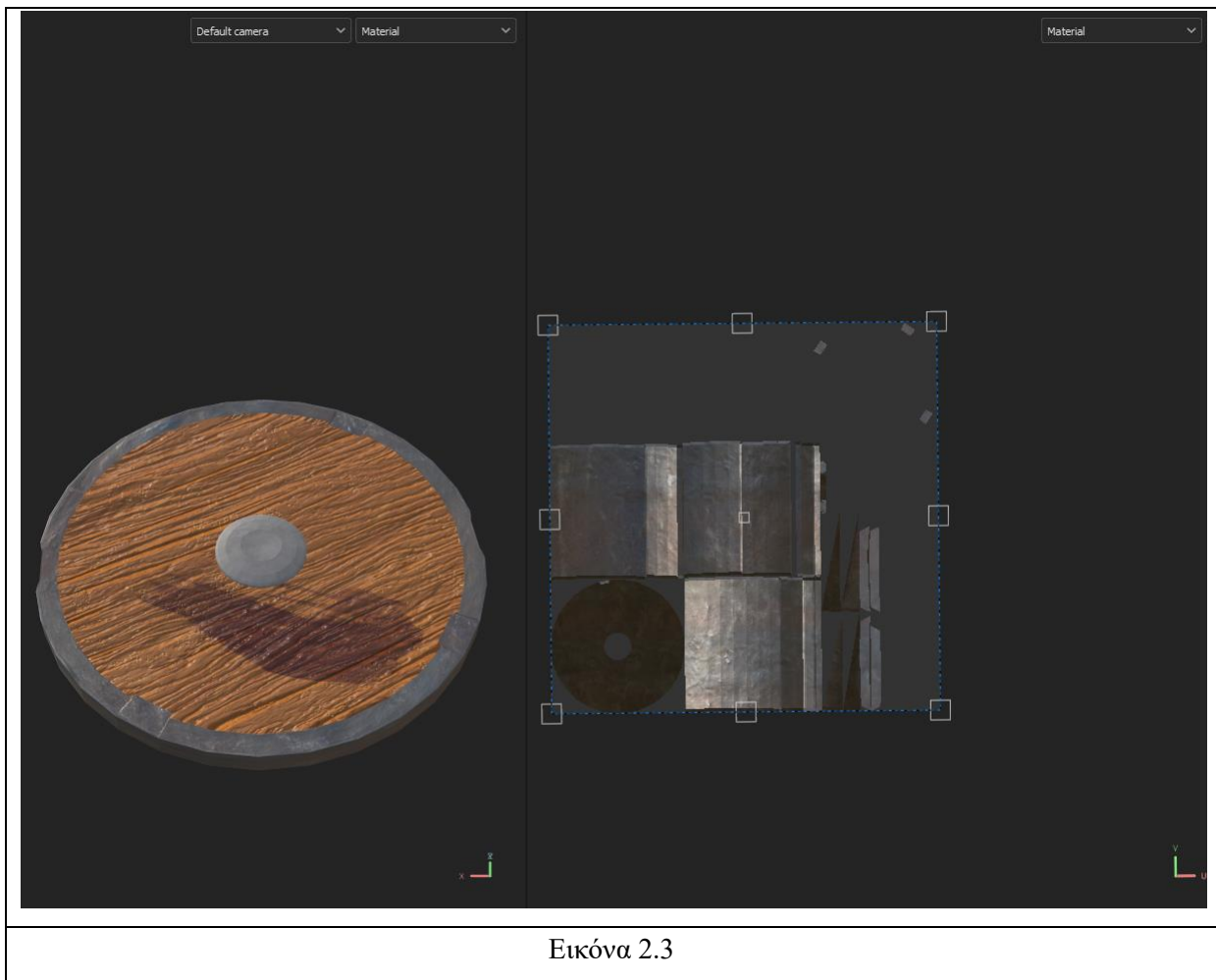
Τα μοντέλα αυτά αφού έχουν δημιουργηθεί στο blender είναι πλέον απαραίτητο να γίνουν textured δηλαδή να εφαρμοστούν πάνω τους διάφορες υφές ώστε να γίνουν αληθοφανή. Αυτό γίνεται μέσω άλλων προγραμμάτων όπως είναι το Substance Painter<sup>9</sup>. Η εργασία αυτή αποτελεί αντικείμενο ενός έμπειρου σχεδιαστή, ωστόσο για τις απαιτήσεις της εργασίας αυτής δημιούργησα ο ίδιος μερικές υφές.



Εικόνα 2.2

---

<sup>9</sup> Πρόγραμμα επεξεργασίας εικόνας/υφών

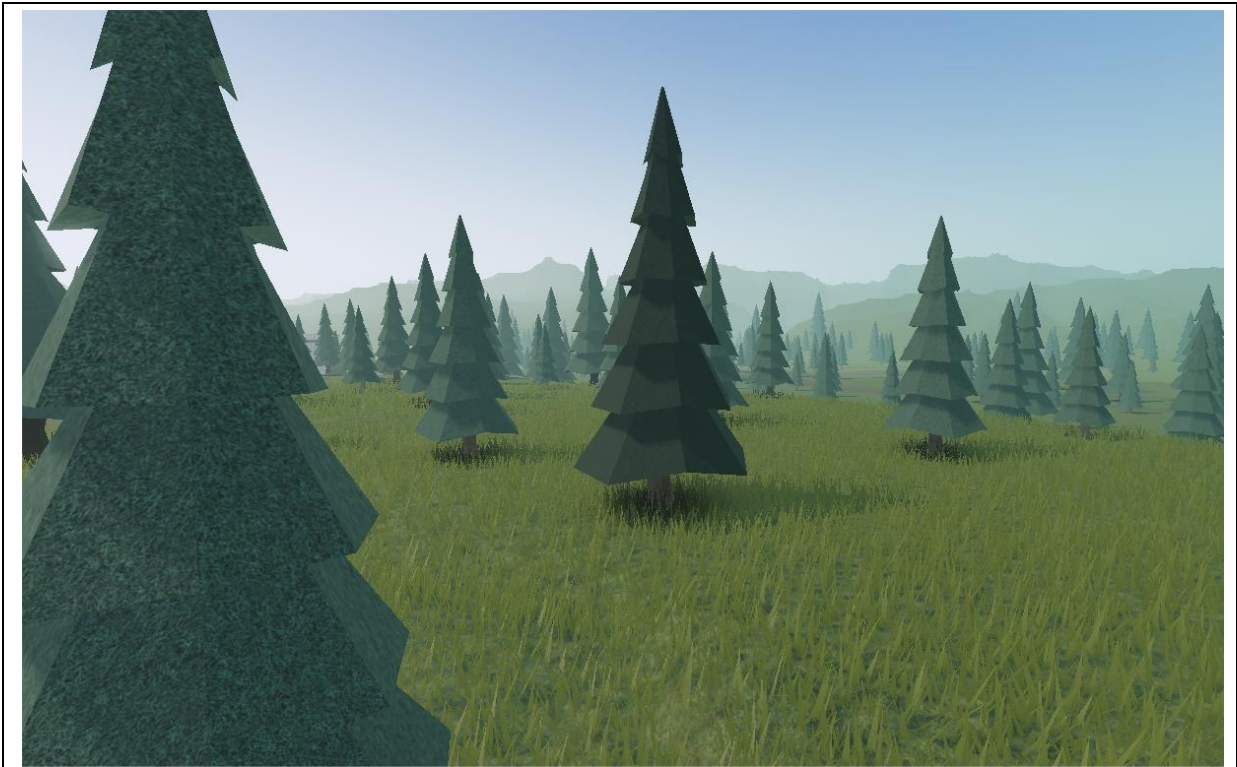


Εικόνα 2.3

Αφού έχει δημιουργηθεί το μοντέλο και έχει γίνει textured είναι πλέον έτοιμο για παραγωγή οπότε εισάγεται στον editor του roblox με την μορφή .fbx<sup>10</sup> αρχείου και έπειτα εφαρμόζονται πάνω του οι υφές που έχουν παραχθεί οι οποίες βρίσκονται σε μορφή .png. Κάθε μοντέλο πρέπει να είναι συναφής με το στυλ του παιχνιδιού αλλιώς υπάρχει κίνδυνος απώλειας την βύθισης του παίκτη στον κόσμο του παιχνιδιού.

---

<sup>10</sup> Τύπος αρχείου τρισδιάστατου μοντέλου



Εικόνα 2.4

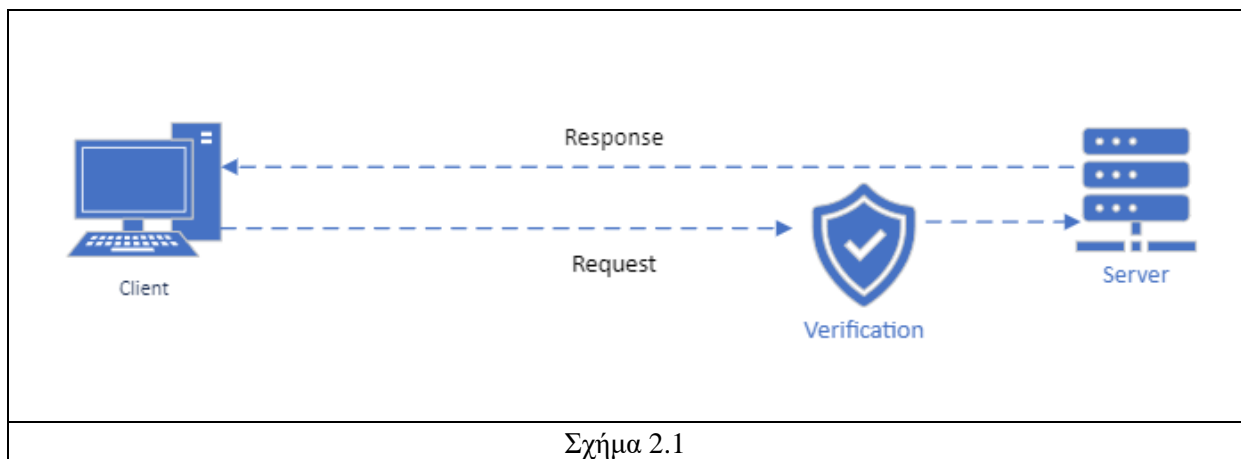
Ο όρος λειτουργικότητα σε ένα παιχνίδι είναι στενά συνδεδεμένος με τον προγραμματισμό του εκάστοτε παιχνιδιού καθώς ο προγραμματισμός είναι αυτός που θα επιτρέψει στον παίκτη την περιήγησή του στον κόσμο του παιχνιδιού και χωρίς αυτόν ο παίκτης δεν θα μπορούσε να τον εξερευνήσει. Η λειτουργικότητα του παιχνιδιού είναι ο τομέας πάνω στον οποίο θα εστιάσει η παρούσα διπλωματική εργασία.

Τα δύο βασικά μέρη της λειτουργικότητας είναι η δυνατότητα του παίκτη να κινηθεί μέσα στον κόσμο οπότε είναι αναγκαία η δημιουργία συστήματος το οποίο θα καταγράφει το input στο πληκτρολόγιο του παίκτη και θα το μεταφράζει σε κινήσεις. Το roblox παρέχει ένα σύστημα κίνησης το οποίο είναι αρκετά αξιόπιστο και έχει την δυνατότητα για επέκταση με την προσθήκη για παράδειγμα διαφόρων animation. Για αυτόν τον λόγο αποφασίστηκε να χρησιμοποιηθεί αυτό το σύστημα.

Επιπροσθέτως ιδιαίτερα σημαντική είναι η προσθήκη «κάμερας» η οποία αποτελεί τον τρόπο με τον οποίο ο χρήστης θα βλέπει τον κόσμο. Οι περισσότερες πλατφόρμες διαθέτουν ήδη αντικείμενο κάμερας ωστόσο το roblox εκτός από κάμερα έχει υλοποιημένες και διάφορες λειτουργίες πάνω στην κάμερα όπως είναι η ακολούθηση του χαρακτήρα. Ειδικότερα διαθέτει υλοποιημένο ένα σύστημα χαρακτήρα και κάμερας το οποίο δίνει την δυνατότητα στον προγραμματιστή να το επεκτείνει. Για τους παραπάνω λόγους αποφασίστηκε να γίνει χρήση και αυτού του συστήματος με ορισμένες προσαρμογές. Μερικές από τις προσαρμογές αυτές είναι τρανταγμός της κάμερας κατά την διάρκεια της μάχης και αλλαγή της γωνίας θέασης

Πριν αναλυθούν τα επιμέρους συστήματα του παιχνιδιού είναι ιδιαίτερα σημαντικό να γίνει αναφορά στην σχέση Client Server η οποία εφαρμόζεται σε κάθε παιχνίδι και η μη σωστή υλοποίηση αυτής της σχέσης μπορεί να έχει καταστροφικά αποτελέσματα. Η διασφάλιση αυτής της σχέσης είναι ιδιαίτερης σημασίας για το συγκεκριμένο παιχνίδι καθώς αφού είναι διαδικτυακό η πρόσβαση χρηστών σε δεδομένα ή σε functions στα οποία δεν θα έπρεπε να έχουν πρόσβαση θα επηρεάσει σημαντικά την

εμπειρία των άλλων παικτών αλλά και θα τους επιτρέψει να αποκτήσουν άδικο πλεονέκτημα. Το τελευταίο αποτελεί σημαντικό πρόβλημα σε όλα τα διαδικτυακά παιχνίδια καθώς η διασφάλιση της ακεραιότητας της κατάστασης του παιχνιδιού απαιτεί πολύ καλή τεχνογνωσία αλλά ακόμα και τότε δεν θα είναι εγγυημένη. Ειδικότερα η λογική πίσω από αυτήν την σχέση ακολουθεί την βασική αρχή σχεδόν κάθε πληροφοριακού συστήματος «Ποτέ μην εμπιστεύεσαι τον πελάτη». Ο λόγος που επικρατεί αυτή είναι λόγω της ύπαρξης κακόβουλων παικτών. Σύμφωνα με έρευνες ένας στους δέκα παίκτες στα διαδικτυακά παιχνίδια είναι κακόβουλοι (Irdeto, 2021). Η σχέση αυτή διατηρείται με την υλοποίηση δύο μερών σε ένα παιχνίδι. Το Client Side και το Server Side. Στο Client Side διατηρούνται όλα τα αρχεία στα οποία μπορεί να έχει πρόσβαση ο χρήστης και δεν περιέχουν κρίσιμα δεδομένα ενώ στο Server όλα τα αρχεία που είναι κρίσιμα για το παιχνίδι όπως όλα τα συστήματα τα οποία διαχειρίζονται δεδομένα και το Hit Detection<sup>11</sup>. Στο roblox η επικοινωνία Client Server γίνεται με remote-events τα οποία είναι σαν Endpoints<sup>12</sup>. Αφού ο χρήστης στείλει request σε ένα endpoint ο Server επιβεβαιώνει ότι το request<sup>13</sup> είναι ορθό και ύστερα επιτρέπει την πρόσβαση στα ανάλογα functions διαφορετικά υπάρχει σύστημα εντοπισμού ότι ο χρήστης επιχείρησε απαγορευμένη πρόσβαση και επιβάλλονται οι ανάλογες ποινές.



Σχήμα 2.1

Από τα παραπάνω γίνεται γρήγορα αντιληπτό ότι ενώ υπάρχουν πολλά Clients (ένα για κάθε παίκτη), υπάρχει ένας μοναδικός Server. Προκύπτουν έτσι δύο πολύ σημαντικά προβλήματα. Αρχικά το πρόβλημα του συγχρονισμού. Σε περίπτωση που δύο παίκτες προσπαθήσουν για παράδειγμα να ανακτήσουν την ίδια στιγμή το ίδιο αντικείμενο. Στην περίπτωση αυτή ο Server θα λάβει δύο ίδια request με μόνη διαφορά το id του client. Σε ποιόν όμως θα δώσει ο server το αντικείμενο; Ο server λοιπόν θα πρέπει να ελέγξει διάφορους παράγοντες όπως το Timestamp των δύο request, αν το αντικείμενο υπάρχει ακόμα και διάφορες άλλες μεταβλητές. Υπάρχει όμως και η περίπτωση όλα αυτά να είναι ακριβώς ίδια όταν οι παίκτες χρησιμοποιούν bots<sup>14</sup>. Για αυτόν τον λόγο λοιπόν απαραίτητη είναι η εφαρμογή Κρίσιμης περιοχής. Στο roblox αυτό υλοποιείται με το να διατηρεί ο server ένα table με όλα τα αντικείμενα τα οποία βρίσκονται υπό επεξεργασία κάθε χρονική στιγμή. Συνεπώς ο server θα εξυπηρετήσει μόνο το πρώτο request που έλαβε. Σε περίπτωση που αυτό γίνεται επανειλημμένα από τον ίδιο χρήστη τότε ο παίκτης κάνει απόπειρα dupring. Δηλαδή να δημιουργήσει δύο αντικείμενα από ένα και για αυτό ο Server θα του επιβάλει ποινή καθώς είναι ενάντια στα TOS<sup>15</sup> του παιχνιδιού. Το

<sup>11</sup> Εντοπισμός χτυπήματος σε συστήματα μάχης

<sup>12</sup> Ένα API endpoint είναι μια συγκεκριμένη διεύθυνση URL που επιτρέπει την επικοινωνία και την ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων λογισμικού.

<sup>13</sup> Αίτηση χρήστη στο σύστημα συνήθως για προσπέλαση δεδομένων.

<sup>14</sup> Αυτοματοποιημένα συστήματα

<sup>15</sup> Όροι χρήσης



δεύτερο πρόβλημα αφορά το latency<sup>16</sup>. Πιο συγκεκριμένα αυτό επηρεάζει το σύστημα μάχης. Ένα αξιοσημείωτο παράδειγμα αφορά την ύπαρξη δύο παικτών. Έστω δύο παίκτες ο Α και ο Β οι οποίοι έχουν χαμηλή ζωή. Ο Α πατάει πρώτος ο κουμπί επίθεσης και μετά από 50ms πατάει το ίδιο κουμπί ο Β αλλά επειδή ο Α έχει latency 200ms και ο παίκτης Β 30ms, ο παίκτης Α πεθαίνει ενώ ο Β θα ζήσει. Αυτό συμβαίνει γιατί παρόλο που το animation της επίθεσης γίνεται replicate μεταξύ όλων των clients και το οπτικό αποτέλεσμα μπορεί να δείξει ότι ο Α χτύπησε πρώτος στην οθόνη του, ο εντοπισμός του χτυπήματος γίνεται μόνο στον server για λόγους ασφαλείας και εφόσον ο Β έχει χαμηλότερο latency από τον Α το request του Β έφτασε και επεξεργάστηκε πρώτο στον server. Το πρόβλημα αυτό είναι αρκετά δύσκολο έως αδύνατο να λυθεί καθώς αφορά φυσικούς περιορισμούς που εξαρτώνται από την τοποθεσία του Server. Ωστόσο καθώς αποτελεί κρίσιμο ζήτημα ασφαλείας δεν γίνεται να εμπιστευτεί ο Server τον Client με τον εντοπισμό χτυπήματος και πάντα κάποιος παίκτης είναι αναγκασμένος να νιώθει αδικημένος.

Κανένα παιχνίδι τέτοιας κλίμακας δεν μπορεί να δημιουργηθεί χωρίς την ύπαρξη ενός συστήματος διαχείρισης δεδομένων. Το σύστημα αυτό αλληλοεπιδρά σχεδόν με όλα τα άλλα συστήματα του παιχνιδιού και χωρίς αυτό δεν είναι δυνατό να διατηρείται η πρόοδος του παίκτη. Επιπροσθέτως χωρίς την ύπαρξη ενός τέτοιου συστήματος θα χάνονταν κρίσιμα δεδομένα όπως η διατήρηση συναλλαγών και θα μπορούσε να αποτελέσει μεγάλο πρόβλημα κατά την λειτουργία του παιχνιδιού καθώς και να έχει νομικές επιπλοκές.

Ένα μεγάλο μέρος παιχνιδιών ρόλων διαθέτει σύστημα διαχείρισης αντικειμένων. Ο ορισμός ενός τέτοιου συστήματος είναι ένα σύστημα στο οποίο ο χρήστης μπορεί να αποθηκεύει αντικείμενα που αποκτά κατά την διάρκεια του παιχνιδιού και μπορεί να έχει διάφορες μορφές (Moore, 2016) Το σύστημα αυτό αλληλοεπιδρά με την βάση δεδομένων ώστε να καταγράφει συνεχώς τι αντικείμενα έχει ο χρήστης πάνω του και συνεπώς αποτελεί ένα πολύ σημαντικό στοιχείο του παιχνιδιού καθώς επιτρέπει στον χρήστη να αποθηκεύει σπαθιά, ασπίδες και άλλα αντικείμενα που μπορεί να αποκτήσει ο παίκτης. Επιπροσθέτως το σύστημα αυτό διαχειρίζεται την δημιουργία νέων αντικειμένων. Κατά την ανάπτυξη του παιχνιδιού έγιναν δύο διαφορετικές υλοποιήσεις στο παρόν σύστημα οι οποίες θα αναλυθούν στην ενότητα της μεθοδολογίας.



<sup>16</sup> Καθυστέρηση απόκρισης

Καθώς το παιχνίδι έχει βάση την αλληλεπίδραση των παικτών απαραίτητη είναι η ύπαρξη ενός συστήματος ανταλλαγής μεταξύ παικτών. Για τον λόγο αυτόν υλοποιήθηκε ένα σύστημα το οποίο επιτρέπει την ανταλλαγή αντικειμένων, πόρων και συναλλάγματος. Με αυτόν τον τρόπο οι παίκτες μπορούν να διαμορφώσουν μόνοι τους την οικονομία του παιχνιδιού χωρίς την ανάγκη εξωτερικής παρέμβασης παρά μόνο για λόγους ισορροπίας της οικονομίας.

Ωστόσο για να μπορούν οι χρήστες να ανταλλάξουν αντικείμενα πρέπει κάπως να μπορούν να τα δημιουργήσουν. Για τον λόγο αυτόν δίνεται η δυνατότητα αγοράς αντικειμένων από διάφορα NPC αλλά και η επιλογή για Crafting αντικειμένων. Δηλαδή οι χρήστες δίνουν υλικά και συνάλλαγμα λαμβάνοντας ως αντάλλαγμα διάφορα αντικείμενα. Πολλά αντικείμενα ωστόσο είναι κλειδωμένα πίσω από την απαίτηση ενός συγκεκριμένου level και άλλων περιορισμών ώστε να ενθαρρύνουν τον παίκτη να παίζει περισσότερο. Επιπλέον, ανάλογα τα υλικά που θα επιλέξουν οι χρήστες μπορούν να δημιουργήσουν καλύτερα και πιο σπάνια αντικείμενα. Για παράδειγμα σε περίπτωση που ο χρήστης επιλέξει έβενο <sup>17</sup> και διαμάντι μπορεί να δημιουργήσει ένα από τα σπανιότερα σπαθιά στο παιχνίδι με την μεγαλύτερη αξία. Βέβαια για να ενισχυθεί περαιτέρω η οικονομία κάθε αντικείμενο αντί να λαμβάνει συγκεκριμένες τιμές όπως για παράδειγμα 25 damage points έχει εύρος τιμών ώστε να εισαχθεί και ο παράγοντας της τύχης στο παιχνίδι.

Εξίσου σημαντική είναι και η ύπαρξη ενός συστήματος μάχης καθώς για να μπορούν οι παίκτες να δημιουργήσουν τις δικές τους ιστορίες οι οποίες είναι κοντά στην πραγματικότητα είναι αναγκαίο να μπορούν να μάχονται μεταξύ τους ιδιαίτερα σε ένα παιχνίδι που έχει ως θέμα τον μεσαίωνα και κύριο στόχο την απόκτηση της βασιλείας. Για τον λόγο αυτό δημιουργήθηκε ένα σύστημα μάχης με σπαθιά τα οποία διαθέτουν τρία διαφορετικά είδη επίθεσης, δυνατότητα απόκρουσης (Block) και σε περίπτωση που γίνει τέλεια απόκρουση ο αντίπαλος αδρανοποιείται για μερικά δευτερόλεπτα κάνοντάς τον ευάλωτο.

Βασικό αντικείμενο αυτής της εργασίας είναι και η παροχή υποδείξεων στους παίκτες. Για τις υποδείξεις αυτές γίνεται χρήση και του συστήματος ειδοποιήσεων που έχει δημιουργηθεί. Ειδικότερα στέλνονται δύο ειδών ειδοποιήσεις. Το πρώτο είδος αφορά μη δυναμικές ειδοποιήσεις και αφορά ειδοποιήσεις οι οποίες στέλνονται ανά τακτά διαστήματα και υπενθυμίζουν στον χρήστη τους διάφορους στόχους του παιχνιδιού παρουσιάζοντας τις διαφορές πτυχές του παιχνιδιού. Για παράδειγμα “Joining a house could be a good step in your path to knighthood”. Το δεύτερο είδος ειδοποιήσεων αφορά δυναμικές ειδοποιήσεις οι οποίες επικεντρώνονται κυρίως στην απόκτηση πόρων και συναλλάγματος σε αυστηρά καθορισμένα χρονικά διαστήματα. Το παιχνίδι παρακολουθεί δηλαδή τις ενέργειες του παίκτη και σε περίπτωση που σε 5 λεπτά δεν μαζεύει αρκετά νομίσματα, (η οποία είναι μία μεταβλητή τιμή που προκύπτει από το σύνολο των παικτών του Server) τότε ειδοποιεί τον χρήστη με το μήνυμα “You need to collect more coins! You are falling behind competition!”. Επιπροσθέτως αναγκαίο είναι να γίνουν γνωστές οι βασικές λειτουργίες του παιχνιδιού στους παίκτες. Για τον σκοπό αυτό έχει υλοποιηθεί ένα tutorial το οποίο αποτελείται από ένα σύστημα cutscene στις οποίες κάθε φορά εξηγείται μία διαφορετική λειτουργία.

Τέλος όλα τα παραπάνω συστήματα σε συνδιασμό με το voice chat και το chat system που παρέχει το roblox επιτυγχάνει την δημιουργία ενός περιβάλλοντος στο οποίο ο παίκτης έχει αμέτρητες επιλογές και μπορεί να εξερευνάει για μεγάλο χρονικό διάστημα καθώς ο τελικός προτεινόμενος στόχος είναι ιδιαίτερα δύσκολος και ακριβός σε θέμα συναλλάγματος παιχνιδιού να επιτευχθεί.

<sup>17</sup> Τύπος ξύλου

## 2.2 Τεχνικές ανάπτυξης

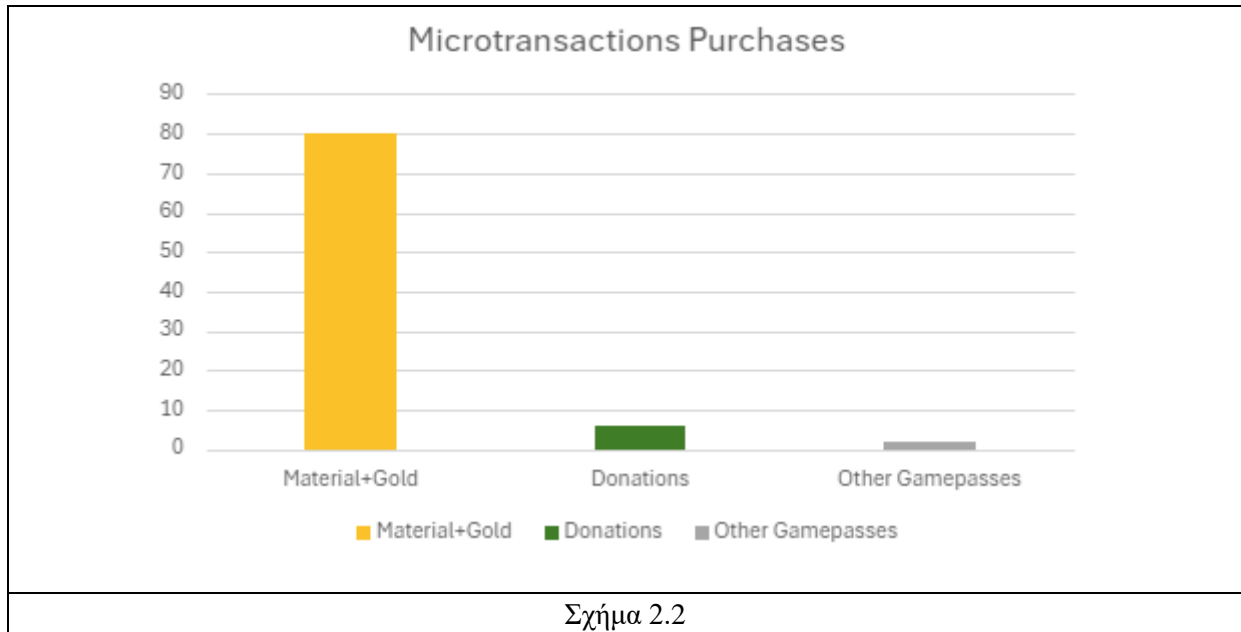
Το Frontend στα παιχνίδια μικρής κλίμακας είναι συνήθως αρκετά μη απαιτητικό σε θέμα χρόνου να υλοποιηθεί. Ωστόσο σε ένα παιχνίδι τέτοιας κλίμακας στο οποίο υπάρχουν πολλές διαφορετικές μεταβλητές απαραίτητη είναι η ύπαρξη ενός Framework με το οποίο θα είναι ευκολότερη η αλλαγή όλων των στοιχείων του περιβάλλοντος του χρήστη που χρησιμοποιούν μία συγκεκριμένη μεταβλητή αντί να αλλάζουν όλα ένα ένα καθώς έτσι ελαττώνεται τόσο ο φόρτος εργασίας του client όσο και ο φόρτος εργασίας του προγραμματιστή μειώνοντας έτσι σημαντικά τον χρόνο ανάπτυξης. Σε διαδικτυακές εφαρμογές αυτό ονομάζεται reactive programming και υπάρχουν πολλά Framework ή παραλλαγές που το εφαρμόζουν όπως η React, Next.js και Nuxt.js. Στο roblox μία αντίστοιχη open source βιβλιοθήκη είναι η Fusion η οποία εφαρμόζει τις τεχνικές τις οποίες χρησιμοποιεί η React και ιδιαίτερα τα states τα οποία διευκολύνουν την ανάπτυξη του frontend. Επομένως, στο παρόν παιχνίδι ο Client διατηρεί στο state όλες τις τιμές οι οποίες γίνονται render στο interface. Όταν ο server στείλει ένα event στον client για αλλαγή μίας τιμής ο Client αυτομάτως θα ανανεώσει όλα τα στοιχεία που την χρησιμοποιούν αντί να είναι αναγκαίο να γίνει hardcoded η αλλαγή.

Όσον αφορά την διαχείριση του server, των δεδομένων και των διαφόρων συστημάτων ακολουθήθηκε μία object-oriented τεχνική η οποία είναι σύμφωνη με τους κανόνες που αναγράφονται στο “Design Principles and Design Patterns” (Robert C. Martin, 2000). Κατασκευάζοντας το παιχνίδι βάση αυτών των κανόνων είναι ευκολότερο να αναπτυχθεί περαιτέρω το παιχνίδι στο μέλλον και να κατασκευαστούν νέα συστήματα σε πολύ μικρότερο χρονικό διάστημα με λιγότερο φόρτο εργασίας. Αρχικά, δημιουργήθηκαν διαφορετικές κλάσεις (οι οποίες στο roblox αναπαρίστανται ως modules) για κάθε σύστημα ακολουθώντας την αρχή μοναδικής ευθύνης. Με αυτόν τον τρόπο κάθε module έχει ξεκάθαρο σκοπό βελτιώνοντας έτσι την συνοχή και καθιστώντας το σύστημα απλούστερο στην κατανόηση και συντήρηση. Επιπροσθέτως κάθε κλάση είναι ανοικτή προς επέκταση αλλά κλειστή προς τροποποίηση μειώνοντας έτσι τον κίνδυνο σφαλμάτων και καθιστά το σύστημα πιο ανεκτικό σε αλλαγές. Πολύ σημαντικό είναι επίσης το πρόβλημα της αντιστροφής εξαρτήσεων. Ειδικότερα τα υψηλού επιπέδου modules δεν θα πρέπει να εξαρτούνται από χαμηλού επιπέδου αποσυνδέοντας έτσι την λογική υψηλού επιπέδου από τις λεπτομέρειες υλοποίησης καθιστώντας έτσι το σύστημα ακόμα πιο εύκολο στην τροποποίηση ή επέκταση.

## 2.3 Τρόποι προσκόμισης κέρδους

Σχεδόν κάθε παιχνίδι το οποίο έχει ως στόχο την ψυχαγωγία έχει και ως στόχο την απόκτηση κέρδους. Ωστόσο όταν ένα παιχνίδι είναι ελεύθερης πρόσβασης πρέπει να δημιουργηθούν άλλοι τρόποι πέρα από τον παραδοσιακό της αγοράς του παιχνιδιού στον οποίο η πρόσβαση απαιτεί την πληρωμή ενός ποσού. Ιδιαίτερα σημαντικό σε αυτές τις περιπτώσεις είναι να αποφευχθεί το να υπάρξουν pay to win τρόποι κέρδους. Τέτοιες πρακτικές μπορεί προσκομίσουν κέρδος σε ένα μικρό διάστημα αλλά φέρνουν δυσαρέσκεια στην κοινότητα του παιχνιδιού και μπορεί να έχουν αρνητικά αποτελέσματα οδηγώντας στην απώλεια παικτών. Pay to win στο συγκεκριμένο παιχνίδι μπορεί να οριστεί η παροχή ειδικών αντικειμένων τα οποία έχουν εξαιρετικά υψηλά στατιστικά έναντι ενός χρηματικού ποσού. Παρόλα αυτά σύμφωνα με έρευνες εκτιμάται ότι οι μικροσυναλλαγές το 2020 αποτελούν το 70% των κερδών στην βιομηχανία παιχνιδιών (Strickland, 2020). Οι μικροσυναλλαγές όμως δεν είναι άρρηκτα συνδεδεμένες με pay to win μεθόδους. Υπάρχει η δυνατότητα παροχής άλλων αντικειμένων ώστε ο παίκτης να νιώθει ότι υποστηρίζει το παιχνίδι ενώ παράλληλα λαμβάνει κάτι το οποίο θα τον βοηθήσει να προχωρήσει και να φτάσει τους στόχους του. Στο παρόν παιχνίδι αναπτύχθηκαν μικροσυναλλαγές με τις οποίες ο παίκτης μπορεί να αποκτήσει υλικά όπως ξύλα, ορείχαλκο, συνάλλαγμα κ.α με

αντάλλαγμα ένα μικρό χρηματικό ποσό γλυτώνοντας έτσι χρόνο τον οποίο θα έπρεπε να σπαταλήσει κόβοντας δέντρα. Σημαντικό είναι όμως το χρηματικό ποσό να αντικατροπτίζει την σπανιότητα του αντικειμένου που λαμβάνει ο χρήστης ενώ να υπάρχουν και μερικά αντικείμενα (τα πιο σπάνια) τα οποία να μην διατίθενται για αγορά ώστε να υπάρχει μία ισορροπία στην οικονομία του παιχνιδιού. Με τον τρόπο αυτό κατά την δοκιμαστική περίοδο παρατηρήθηκε ότι οι παίκτες προτιμούσαν ιδιαίτερα την αγορά πακέτων που περιλάμβαναν την πώληση συναλλάγματος έναντι άλλων προϊόντων. Επίσης παρατηρήθηκε ότι η κατηγορία Other Gamepasses η οποία περιλαμβάνει άλλου είδους υπηρεσίες όπως για παράδειγμα πρόσβαση σε συγκεκριμένη περιοχή είχε τις χαμηλότερες πωλήσεις, δηλαδή πουλούσε λιγότερο και από τις δωρεές.



## ΚΕΦΑΛΑΙΟ 3

### ΥΛΟΠΟΙΗΣΗ ΠΑΙΧΝΙΔΙΟΥ

#### 3.1 Σκοπός του παιχνιδιού

Το παιχνίδι αυτό αν και δεν έχει συγκεκριμένο σκοπό δημιουργεί έμμεσα διάφορους στόχους για την ψυχαγωγία του παίκτη. Ο μεγαλύτερος στόχος είναι για παράδειγμα η απόκτηση της βασιλείας. Ένας παίκτης για να το επιτύχει αυτό θα πρέπει να παίξει το παιχνίδι για ένα μεγάλο χρονικό διάστημα ώστε να δημιουργήσει σχέσεις με άλλους παίκτες και να τους εντάξει στον οίκο του και κατ' επέκταση στον στρατό του. Αφού ένας παίκτης μαζέψει αρκετούς στρατιώτες μπορεί να πραγματοποιήσει επιθέσεις σε άλλους οίκους οι οποίες λαμβάνουν χώρα σε ένα άλλο παιχνίδι το οποίο αν και άμεσα συνδεδεμένο με τον παρόν δεν θα αναλυθεί στην παρούσα διπλωματική εργασία. Κάνοντας αυτές τις μάχες ο παίκτης κερδίζει περιοχές στον χάρτη του παιχνιδιού μέχρι να φτάσει στην πρωτεύουσα στην οποία θα πρέπει να πολεμήσει τον κυρίαρχο οίκο και τους συμμάχους του. Μία τέτοια προσέγγιση ωστόσο ενέχει υψηλά ρίσκα καθώς αν το στέμμα κερδίσει την μάχη ο οίκος που πραγματοποιεί την επανάσταση είναι πλέον υποκείμενος σε βαριές κυρώσεις τις οποίες μπορεί να του επιβάλει το στέμμα. Οι κυρώσεις αυτές αφήνονται στην κρίση των παικτών και δεν έχουν περιορισμούς καθώς το παιχνίδι είναι ένα παιχνίδι ρόλων. Αντίθετα αν ο οίκος που επαναστατεί κερδίσει αποκτά πλέον το στέμμα και είναι πλέον ο κυριεύουν οίκος επιβάλλοντας τους νόμους του με την εξουσία που έχει αποκτήσει. Υπάρχει βέβαια και η περίπτωση ένας οίκος να χειροτερεύει την εμπειρία των άλλων παικτών με την επιβολή για παράδειγμα τυραννίας. Μία τυραννία μπορεί να πάρει διάφορες μορφές σε αυτό το παιχνίδι. Ένα είδος το οποίο παρατηρήθηκε κατά την διάρκεια των δοκιμών είναι η κατάληψη των ορυχείων από τον κυβερνών οίκο. Η αρχική προσέγγιση σε μία τέτοια περίπτωση είναι ότι δεν θα υπάρξει κάποια παρέμβαση καθώς πιθανότατα κάποιος οίκος θα επαναστατήσει ή θα γίνουν μάχες για την κατάληψη του ορυχείου. Ωστόσο αν δεν συμβεί αυτό θα πρέπει να υπάρξει επαφή με τον ιδιοκτήτη αυτού του οίκου ώστε να μην χάσει παίκτες το παιχνίδι λόγω της συμπεριφοράς του. Σημαντικό είναι όμως ο ιδιοκτήτης να μην νιώσει ότι του επιβάλλεται κάτι. Για τον λόγο αυτό σε περίπτωση ασυμφωνίας υπάρχει η δυνατότητα μείωσης των ορυκτών του ορυχείου ή η «κατάρρευση» του με την τοποθέτηση αντικειμένων στο περιβάλλον τα οποία αποτρέπουν την πρόσβαση σε όλους. Με αυτόν τον τρόπο παρόλο που είναι αρκετά παρεμβατικό δένει ωραία με την ιστορία του παιχνιδιού και έτσι οι γενικές αντιδράσεις ελαχιστοποιούνται.

Αν και ο παραπάνω στόχος παρατηρήθηκε ότι είναι ο πιο επιθυμητός οι παίκτες μπορούν να ακολουθήσουν και άλλους δρόμους. Για παράδειγμα μπορούν να ενταχθούν στις τάξεις της αστυνομίας και να επιβάλλουν τον νόμο ή να ενταχθούν σε έναν οίκο ως στρατιώτες με την φιλοδοξία να προαχθούν σε αξιωματικοί ή ιππότες. Να τονιστεί ότι λόγω του φόρτου ανάπτυξης χαρακτηριστικά όπως βαθμίδες σε οίκους ή στην αστυνομία διατηρούνται από το σύστημα ομάδων το οποίο παρέχει το roblox. Το σύστημα αυτό παρέχει την δυνατότητα δημιουργίας και διαχείρισης ομάδων με βαθμίδες. Επιπλέον οι οίκοι μπορούν να επικοινωνούν με τα μέλη τους και εκτός παιχνιδιού μέσω διαφόρων προγραμμάτων επικοινωνίας όπως είναι το Discord<sup>18</sup>.

---

<sup>18</sup> Πρόγραμμα επικοινωνίας

Συμπερασματικά το παιχνίδι παρέχει την βάση ώστε οι παίκτες να μπορούν να δημιουργήσουν οποιοδήποτε σενάριο επιθυμούν με μόνο όριο την φαντασία τους και φυσικά τους κανόνες του παιχνιδιού και της πλατφόρμας roblox.

### 3.2 Διαχωρισμός ανάπτυξης

Η ανάπτυξη ενός τόσο μεγάλου παιχνιδιού σε μία μόνο σκηνή θα ήταν ιδιαίτερα δύσκολο για έναν υπολογιστή να το διαχειριστεί και θα προκαλούσε μεγάλες καθυστερήσεις στην ανάπτυξη. Ειδικότερα ένα ολοκληρωμένο παιχνίδι αποτελείται από το γραφικό περιβάλλον και το λειτουργικό περιβάλλον όπως αναλύθηκε παραπάνω. Αν και τα δύο αυτά περιβάλλοντα βρισκόταν στην ίδια σκηνή (δηλαδή σε ολοκληρωμένη μορφή) κατά το στάδιο της ανάπτυξης, ο έλεγχος της λειτουργικότητας θα ήταν σχετικά αργός καθώς το game engine θα έπρεπε να φορτώσει τόσο το γραφικό περιβάλλον όσο και το λειτουργικό. Για τον λόγο αυτόν τα δύο αυτά περιβάλλοντα διαχωρίστηκαν σε δύο σκηνές κατά την ανάπτυξη. Πιο συγκεκριμένα, πρώτο αναπτύχθηκε το γραφικό περιβάλλον το οποίο αποτελούσε τον κόσμο με τον οποίο θα αλληλοεπιδρούσαν οι παίκτες. Έπειτα αφού αυτό ήταν ολοκληρωμένο ξεκίνησε η ανάπτυξη του λειτουργικού περιβάλλοντος. Δηλαδή των συστημάτων και της διεπαφής χρήστη. Τέλος, τα δύο αυτά περιβάλλοντα αφού είχαν ολοκληρωθεί και ελεγχθεί ενώθηκαν Ακολουθώντας αυτήν την τεχνική επιταχύνθηκε εξαιρετικά ο χρόνος ανάπτυξης καθώς η λειτουργικότητα μπορούσε να ελεγχθεί σε σημαντικά μικρότερο χρόνο.

### 3.3 Σχεδιασμός του γραφικού περιβάλλοντος

Για να γίνει σωστή ανάπτυξη του κόσμου με τον οποίο θα αλληλοεπιδρούσαν οι παίκτες έπρεπε να γίνει και ο κατάλληλος σχεδιασμός. Κατά τον σχεδιασμό αυτό λήφθηκαν υπόψιν όλα τα συστήματα τα οποία έπρεπε να προστεθούν. Ιδιαίτερη βαρύτητα δόθηκε στο σύστημα απόκτησης πόρων για το οποίο έπρεπε να αναπτυχθούν δάση ώστε να μπορούν οι παίκτες να κόβουν ξύλα αλλά και ορυχεία για την απόκτηση ορυκτών. Για την αποφυγή καταστροφής της οικονομίας τα ορυχεία και τα δάση αυτά έπρεπε να βρίσκονται σε κατάλληλα σημεία και να παρέχουν την κατάλληλη αναλογία πόρων ώστε να μην δημιουργηθεί υπερ. προσφορά υλικών. Μετά από κατάλληλη έρευνα και παρακολουθώντας τις τάσεις σε άλλα παιχνίδια αποφασίστηκε ότι για την κλίμακα του χάρτη ο κατάλληλος αριθμός ορυχείων ήταν δύο ενώ τα δάση ήταν μία δυσκολότερη περίπτωση.

Βασικός σκοπός κατά την ανάπτυξη γραφικού περιβάλλοντος είναι ο κόσμος να είναι αισθητικά ωραίος αλλά και γεμάτος. Συνήθως σε εξωτερικούς χώρους αυτό σημαίνει πολλά δέντρα. Ωστόσο αυτό πάει αντίθετα με την λογική της ισορροπίας των πόρων. Το πρόβλημα αυτό αντιμετωπίστηκε με την δημιουργία δύο ειδών δέντρων τα οποία θα διαχωρίζονταν με βάση το χρώμα των φύλλων τους. Ειδικότερα τα δέντρα με τα οποία μπορούν οι παίκτες να αλληλοεπιδράσουν έχουν σκοτεινότερο χρώμα δείχνοντας έτσι στον παίκτη ότι είναι πιο ώριμα και έτοιμα για κοπή. Αντίθετα τα υπόλοιπα δέντρα είχαν άλλο χρώμα και βασικός τους σκοπός ήταν να κάνουν το περιβάλλον να φαίνεται αισθητικά ωραίο.



Εικόνα 3.1

Αν και αποφασίστηκε σχετικά γρήγορα ο αριθμός των ορυχείων υπήρχε τώρα το πρόβλημα του «πόσα ορυκτά πρέπει να υπάρχουν κάθε χρονική στιγμή». Για λόγους ισορροπίας αλλά και για να δοθεί ένα κίνητρο στους παίκτες να αποκτήσουν τον έλεγχο των ορυχείων ορίστηκε ότι κάθε ορυχείο θα είχε έξι ορυκτά τα οποία αφού εξ ορυχθούν θα επανεμφανιστούν μετά από διάστημα δύο λεπτών. Κατά την διάρκεια των δοκιμών παρατηρήθηκε ότι οι παίκτες μάχονταν για τον έλεγχο των ορυχείων καθώς τα ορυκτά ήταν άμεση πηγή εισοδήματος.



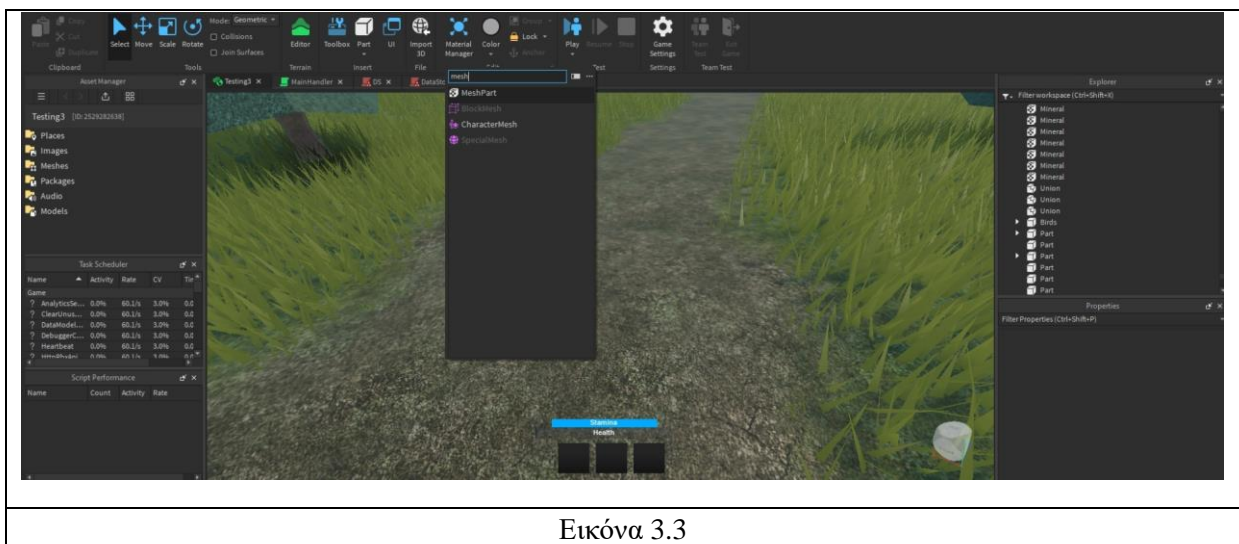
Εικόνα 3.2

Ανάλογη βαρύτητα δόθηκε και στην ανάγκη των παικτών για αλληλεπίδραση. Για τον σκοπό αυτό δημιουργήθηκαν διάφοροι χώροι στον χάρτη τα οποία είναι κέντρα κοινωνικοποίησης. Στους χώρους αυτούς παρέχονται όλες οι υπηρεσίες του παιχνιδιού. Πιο συγκεκριμένα οι χώροι αυτοί παρέχουν μαγαζιά, σταθμούς κατασκευής αντικειμένων και εμπόρους ορυκτών και ξύλων. Οι χώροι αυτοί διατηρούν την λογική ασφαλούς χώρου ή αλλιώς safezone<sup>19</sup>. Ωστόσο επειδή τον μεγαλύτερο κομμάτι της κοινότητας αλληλοεπιδρά σε αυτούς τους χώρους αποφασίστηκε να εφαρμοστεί με την εφαρμογή χαλαρών safezone. Ειδικότερα δεν εμπόδιζε κάποιο σύστημα στους παίκτες να επιτίθενται ο ένας στον άλλον αλλά υπήρχαν κανόνες με την μορφή νόμων τους οποίους εφάρμοζαν συγκεκριμένοι παίκτες οι οποίοι διατηρούσαν τον ρόλο της αστυνομίας.

### 3.4 Ανάπτυξη του γραφικού περιβάλλοντος

Για την ανάπτυξη του γραφικού περιβάλλοντος χρησιμοποιήθηκε το εργαλείο Blender και Substance Painter για την ανάπτυξη των μοντέλων και των textures. Ωστόσο για την εισαγωγή τους στο roblox έπρεπε να ακολουθηθούν κάποια συγκεκριμένα βήματα.

Αρχικά εισάγεται ένα αντικείμενο τύπου MeshPart το οποίο με τα κατάλληλα αρχεία θα πάρει την ανάλογη μορφή.

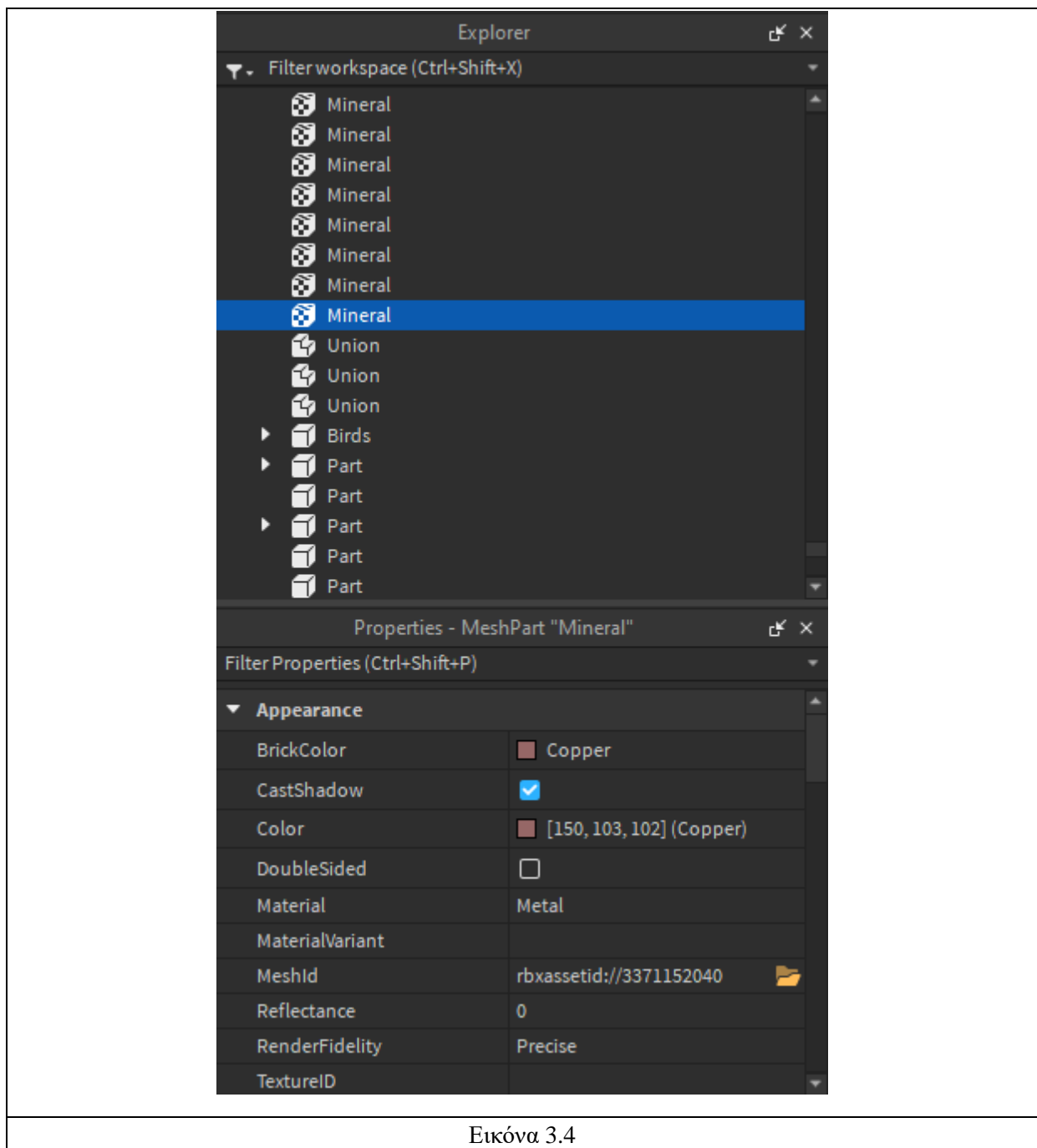


Εικόνα 3.3

Αφού εισαχθεί το αντικείμενο πρέπει να του αντιστοιχηθεί το ανάλογο .fbx αρχείο το οποίο θα παρέχει την απεικόνιση που θα έχει το μοντέλο. Το αρχείο αυτό το εισάγουμε στο property MeshId. Έπειτα θα εισαχθεί το κατάλληλο texture το οποίο έχει δημιουργηθεί στην εφαρμογή Substance Painter. Το texture αυτό εισάγεται στην θέση TextureId. Υπάρχει και η επιλογή για την δημιουργία πιο εξελιγμένων texture παρέχοντας τα ανάλογα maps.

<sup>19</sup> Χώροι στους οποίους δεν επιτρέπεται η επίθεση σε άλλους παίκτες





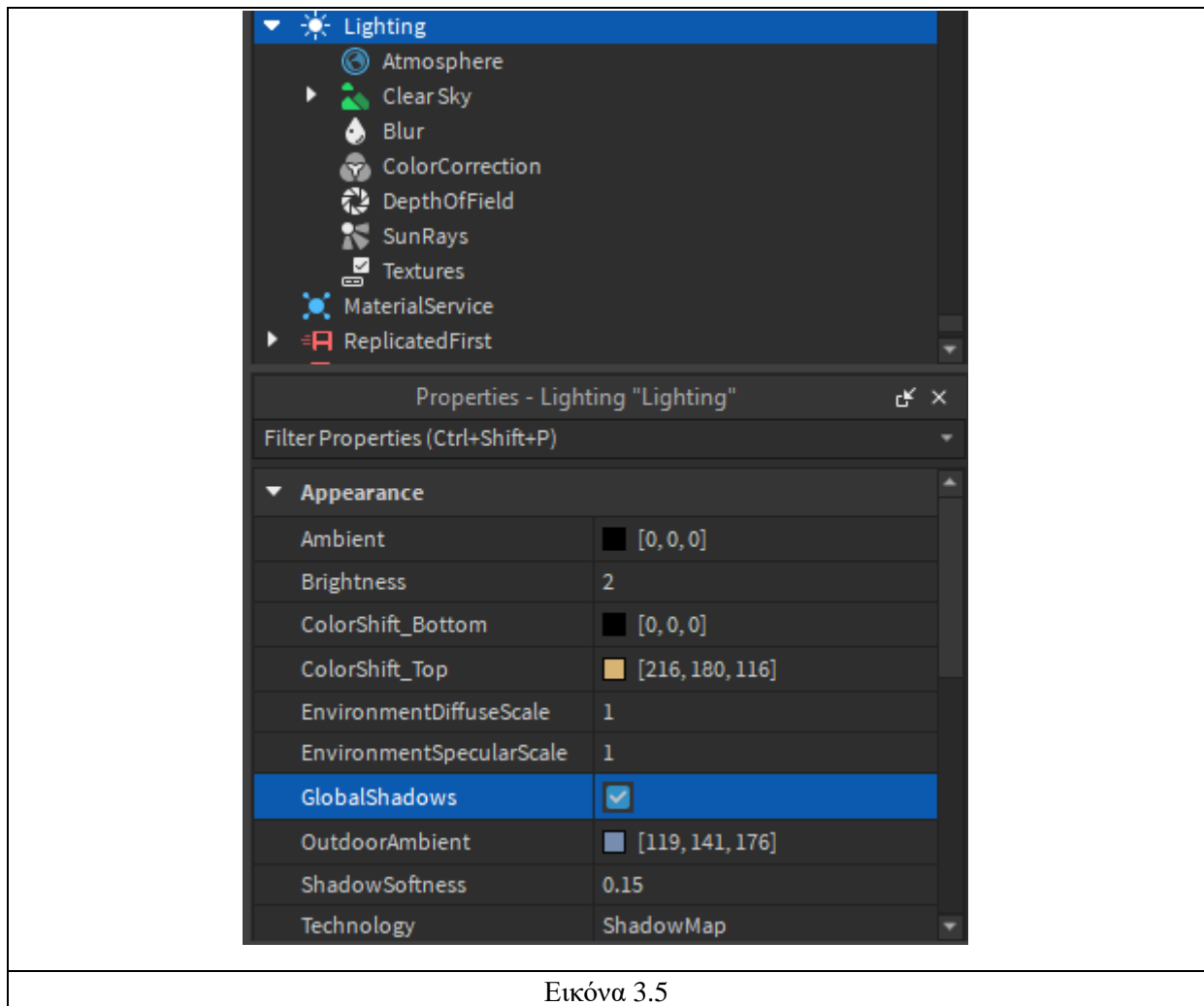
Εικόνα 3.4

Αφού εισαχθεί το αντικείμενο και έχει την μορφή την οποία θέλουμε σειρά έχει η απόφαση του αν το αντικείμενο θα είναι δυναμικό ή στατικό. Πιο συγκεκριμένα αν το αντικείμενο θα αλληλοεπιδράει με την φυσική του engine ή όχι. Αν δεν θέλουμε να αλληλοεπιδράει για παράδειγμα το τείχος στο κάστρο ή ένα τραπέζι τότε είναι αναγκαίο το property anchored να είναι true. Αντικείμενα τα οποία κινούνται μαζί με τον παίκτη όπως π.χ σπαθιά ή ασπίδες πρέπει να μην είναι anchored αλλιώς ο παίκτης δεν θα μπορεί να κουνηθεί όταν εξοπλίζεται με αυτά. Υπάρχουν διάφορες άλλες παράμετροι οι οποίες μπορούν να αλλαχθούν ώστε το αντικείμενο να πάρει την μορφή την οποία θέλουμε. Τέτοιες παράμετροι είναι για παράδειγμα το χρώμα το οποίο είναι σε μορφή rgb<sup>20</sup>. Το Transparency δηλαδή η διαφάνεια του αντικειμένου, η ανακλαστικότητα, το υλικό, το μέγεθος, η περιστροφή και πολλά άλλα. Ιδιαίτερα σημαντικό είναι να τονιστεί ότι όλα αυτά τα properties μπορούν να αλλαχθούν κατά την

<sup>20</sup> Red Green Blue ευρέως διαδεδομένη μορφή χρωματοδοσίας σε ηλεκτρονικά μέσα όπου κάθε στοιχείο παίρνει τιμές από 0 έως 255.

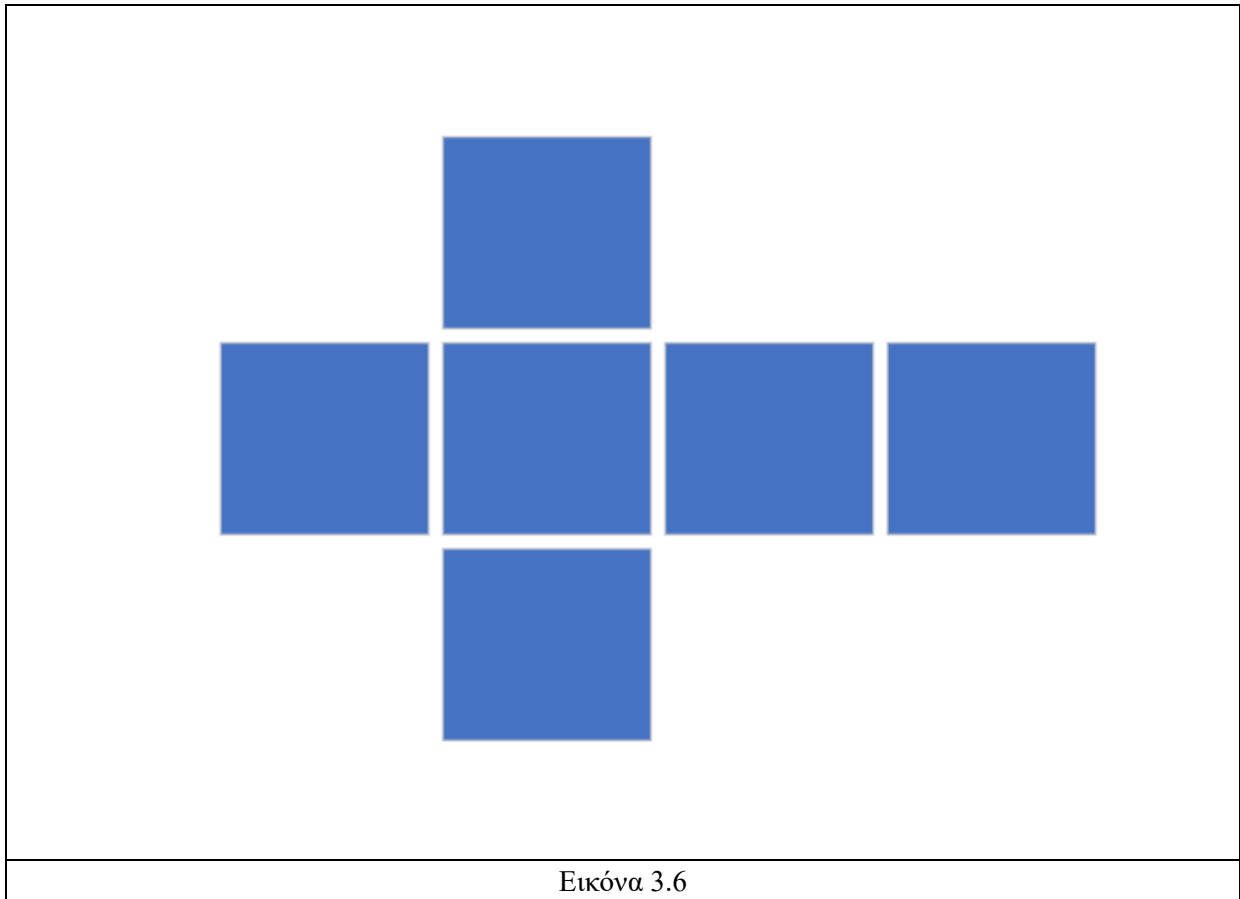
λειτουργία του παιχνιδιού μέσω των κατάλληλων εντολών επιτρέποντας έτσι την δημιουργία ενός δυναμικού περιβάλλοντος.

Για να φαίνεται ένας κόσμος αληθοφανής ένα από τα σημαντικότερα στοιχεία είναι ο φωτισμός. Το roblox παρέχει ένα ικανοποιητικό σύστημα το οποίο παρέχει την ρύθμιση διαφόρων παραμέτρων όπως η ομίχλη, η πυκνότητά της, το ύψος στο οποίο αρχίζει να εμφανίζεται, οι ακτίνες του ηλίου, το θάμπωμα της οθόνης και πολλά άλλα. Όλα αυτά γίνονται μέσω του lighting και άλλων υπό αντικειμένων που τοποθετούνται ως παιδιά του lighting και παρέχονται από το roblox.

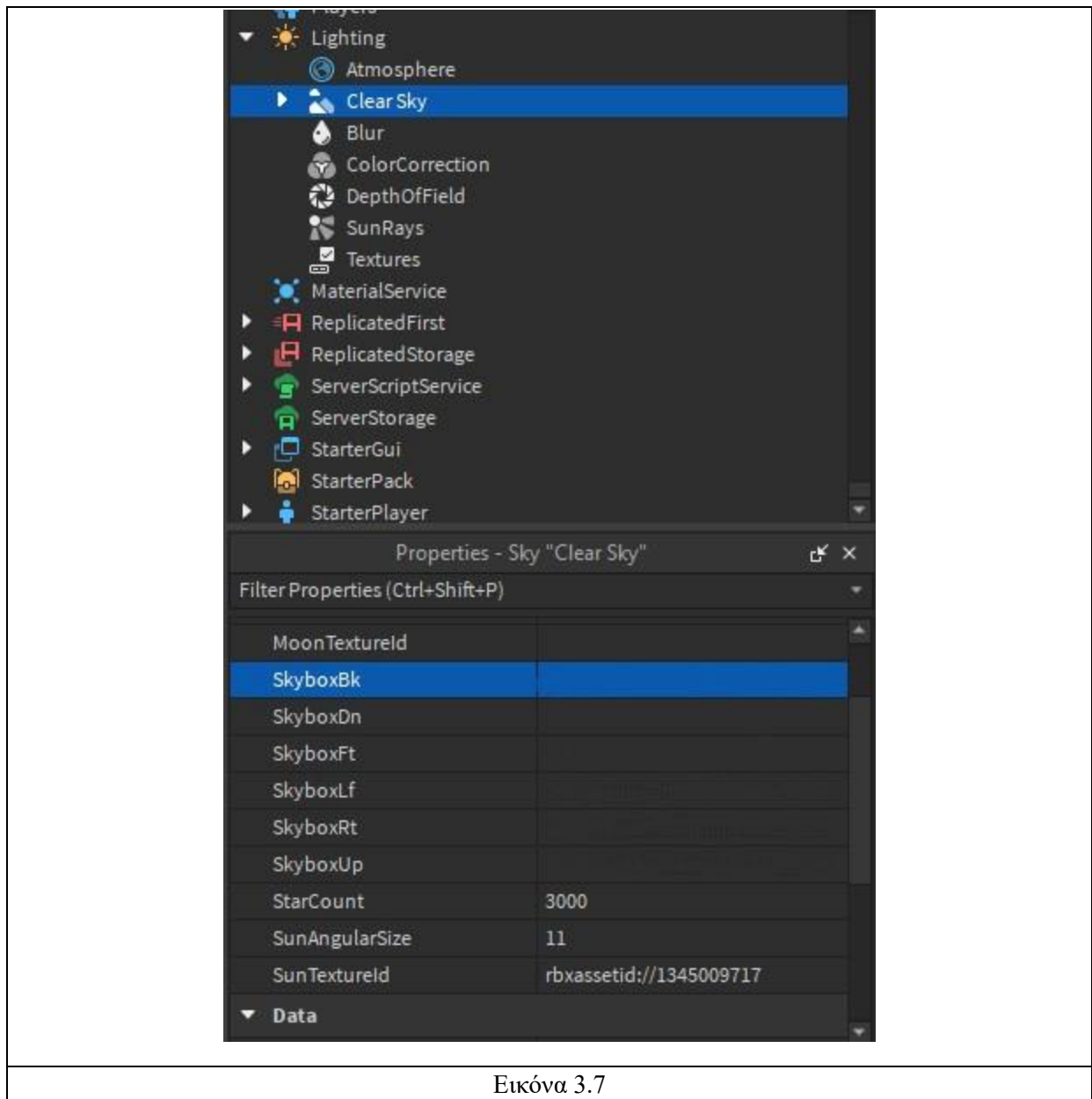


Εικόνα 3.5

Εξίσου σημαντική είναι και η παροχή ενός αληθοφανούς skybox δηλαδή του ουρανού του παιχνιδιού. Το skybox έχει την μορφή κύβου και είναι 6 φωτογραφίες οι οποίες αναπαριστούν όλες τις κατευθύνσεις. Με την δημιουργία σωστών φωτογραφιών ο ουρανός φαίνεται σαν να είναι ενοποιημένος.

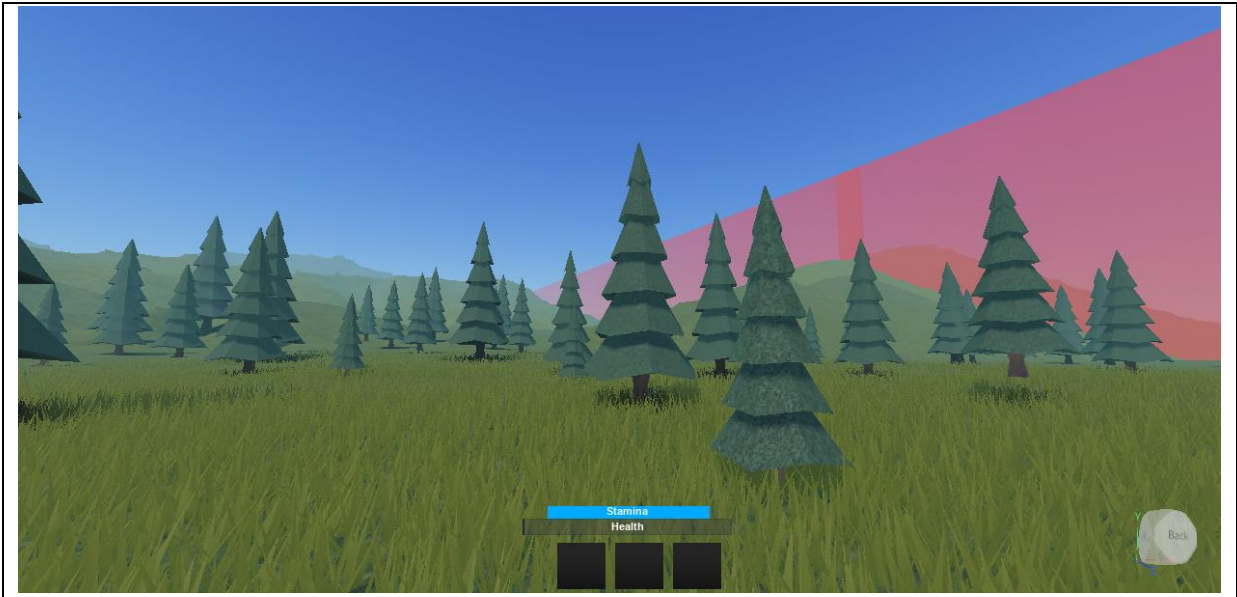


Στο roblox οι φωτογραφίες αυτές εισάγονται σε ένα υπό αντικείμενο τύπου sky το οποίο τοποθετείται μέσα στο Lighting. Στο αντικείμενο αυτό ωστόσο τοποθετούνται και δύο παραπάνω φωτογραφίες οι οποίες αναπαριστούν τον ήλιο και το φεγγάρι. Υπάρχουν και άλλες ρυθμίσεις οι οποίες μπορούν να γίνουν όπως για παράδειγμα ο αριθμός των αστεριών και το μέγεθος του ηλίου δίνοντας έτσι περισσότερες δυνατότητες.



Εικόνα 3.7

Ιδιαίτερα σημαντικό είναι και το ζήτημα του σε ποιες περιοχές θα έχει πρόσβαση ένας παίκτης αλλά και τα όρια του χάρτη. Καθώς το παιχνίδι αφορά μία μεσαιωνική εποχή ο κόσμος περιλαμβάνει κυρίως εξωτερικά περιβάλλοντα και πιο συγκεκριμένα βουνά και θάλασσα. Η δημιουργία ενός terrain το οποία θα επεκτείνεται στο άπειρο είναι αδύνατη λόγω τεχνικών περιορισμών. Για τον λόγο αυτόν επιβλήθηκαν όρια στον χάρτη με την μορφή αόρατων τοίχων και το terrain αναπτύχθηκε έτσι ώστε να φαίνεται άπειρο από την οπτική γωνία του παίκτη.

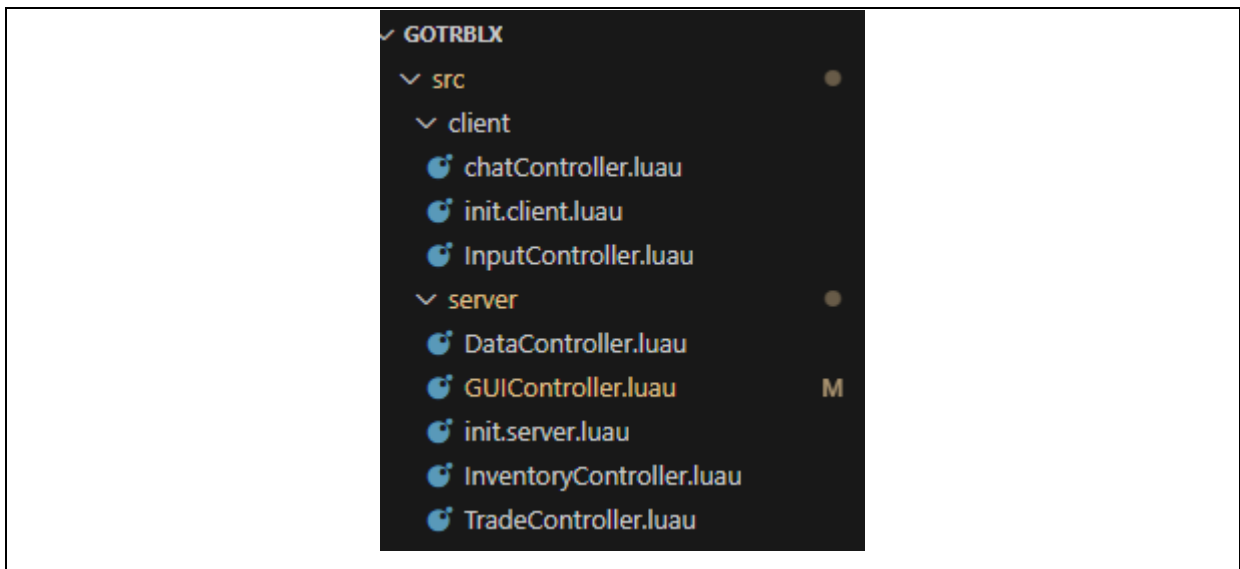


Εικόνα 3.8

### 3.5 Σχεδιασμός και ανάπτυξη της λειτουργικότητας

Όπως αναλύθηκε και σε προηγούμενο κεφάλαιο κάθε παιχνίδι αποτελείται από δύο μέρη. Τον Server και τον Client. Αρχικά θα αναλυθεί η υλοποίηση του Server.

Ο server αποτελείται από το initialization class και τα ανάλογα modules.



Εικόνα 3.9

Η κλάση του initialization έχει την παρακάτω μορφή

```

local profileService = require(game:GetService("ServerScriptService").ProfileService)
local InventoryController =
require(game:GetService("ServerScriptService").Server.InventoryController)
local DataController = require(game:GetService("ServerScriptService").Server.DataController)
local GUIController = require(game:GetService("ServerScriptService").Server.GUIController)
local TradeController = require(game:GetService("ServerScriptService").Server.TradeController)
...
GUIController.new(DataController,InventoryController,TradeController)
DataController.new(profileService,GUIController)
InventoryController.new(DataController,GUIController)
TradeController.new(InventoryController,GUIController)
...
game.Players.PlayerAdded:Connect(function(player)
    DataController.LoadProfile(player)
end)

```

Ειδικότερα για να εφαρμοστεί μία αρχιτεκτονική singleton για κάθε module καταγράφονται αρχικά όλα τα modules ή αλλιώς controllers. Αφού καταγραφθούν όλα καλείται η συνάρτηση new του κάθε module. Αρμοδιότητα της new είναι να κάνει initialize τις μεταβλητές των κλάσεων οι οποίες δείχνουν στο ανάλογο module το οποίο χρειάζεται για να λειτουργήσει σωστά το module (Αντιστοιχεί στην συνάρτηση constructor). Με αυτόν τον τρόπο διαβεβαιώνουμε ότι τρέχει μόνο ένας controller του κάθε συστήματος κάθε φορά αποφεύγοντας διάφορα προβλήματα τα οποία μπορούν να εμφανιστούν κατά την εκτέλεση. Στο τέλος του initialization script υπάρχει το event game.Players.PlayerAdded:Connect το οποίο καλείται κάθε φορά που εισέρχεται νέος παίκτης στον server και αναλαμβάνει να ανακτήσει τα δεδομένα του από την βάση και να εκτελέσει όλες τις ενέργειες απαραίτητες για την αρχικοποίησή του. Επιπλέον από αυτήν την κλάση αρχικοποιούνται διάφορα άλλα συστήματα που αφορούν τον κόσμο και πρέπει να εκτελεστούν κατά την εκκίνηση του server όπως το σύστημα μέρας / νύχτας.

### 3.5.1 Σύστημα διαχείρισης δεδομένων

Ως σύστημα θεωρείται οποιαδήποτε λειτουργία μεγάλης κλίμακας στο παιχνίδι. Ένα αρκετά μεγάλο σύστημα είναι αυτό της ανάκτησης δεδομένων το οποίο είναι κρίσιμο για την διατήρηση του progression και καταγραφή διαφόρων στατιστικών σχετικά με το παιχνίδι. Το σύστημα το οποίο αναπτύχθηκε στηρίζεται σε μία βιβλιοθήκη την Profile Service η οποία χρησιμοποιεί επίσης συναρτήσεις του roblox για πρόσβαση στις βάσεις δεδομένων του. Η βιβλιοθήκη αυτή παρέχει μονάχα την βάση του συστήματος και αποτελεί επέκταση του roblox API παρέχοντας προστασία ενάντια σε απώλεια δεδομένων λόγω απρόσκοπτης διακοπής του παιχνιδιού, παρέχει caching δεδομένων κ.α. Για την διευκόλυνση του προγραμματιστή το API αυτό χρησιμοποιεί μία μορφή τύπου dictionary για την καταγραφή των δεδομένων. Κάθε φορά που εισέρχεται ένας νέος παίκτης στο παιχνίδι το module προσπαθεί να ανακτήσει τα δεδομένα του παίκτη από το API. Σε περίπτωση που αποτύχει επειδή τα

δεδομένα δεν υπάρχουν τότε θα καταγράψει τα παρακάτω δεδομένα και θα τα αντιστοιχίσει με τον παίκτη χρησιμοποιώντας ως κλειδί το UID του χρήστη το οποίο παρέχεται από το roblox.

```
local ProfileTemplate = {
  Stats = {
    LogInTimes = 0,
    Cash = 0,
    BankCash = 0,
    BankMax = 1000,
    BankInterest = 0.1,
    BankLast = os.time(),
    Level = 1,
    XP = 0
  },
  Group = {
    GroupId = 0,
    GroupRank = 0,
    GroupName = "None"
  },
  Equipped={
    Helmet = {current = "",previous = ""},
    Chest = {current = "",previous = ""},
    Legs = {current = "",previous = ""},
    Shield = {current = "",previous = ""},
    Offensive = {current = "",previous = ""},
    Consumable = {current = "",previous = ""}
  },
  Inventory = {
    Offensive = {},
    Helmet = {},
    Chest = {},
    Legs = {},
    Shield = {},
    Consumable = {},
    Resource = {}
  },
  Settings = {
    Music = 0.5,
    SFX = 0.5,
    Notifications = true
  },
  Extra = {
    jailed = false,
    jailed_at = 0,
    jailed_until = 0
  },
  Max_Weight = "50"
}
```

Στο module του συστήματος της βάσης έχουν δημιουργηθεί διάφορες συναρτήσεις για την διαχείριση δεδομένων όπως οι παρακάτω

- `DataController.LoadProfile(player)` | Φόρτωση του προφίλ του χρήστη
- `DataController.Update(plr, Data)` | Αλλαγή κάποιου value
- `DataController.Get(plr)` | ανάκτηση πίνακα δεδομένων χρήστη
- `DataController.Increment(plr, Path, Value)` | increment ενός value
- `DataController.Delete(plr, Path)` | Διαγραφή στοιχείου
- `DataController.Transfer(from_plr, to_plr2, Path)` | Μεταφορά από έναν παίκτη σε άλλον

Ωστόσο παρόλο που η μορφή Dictionary μπορεί να διευκολύνει αρκετά την εργασία για έναν άπειρο προγραμματιστή δημιουργεί το πρόβλημα της αλλαγής στοιχείων. Ειδικότερα σε περίπτωση που θέλουμε να αλλάξουμε ένα στοιχείο το οποίο είναι εμφολευμένο θα πρέπει να φτιάξουμε ειδική συνάρτηση για το συγκεκριμένο βάθος. Όπως γίνεται γρήγορα αντιληπτό αυτό είναι ιδιαίτερα δύσκολο και επιβαρυντικό σε θέμα φόρτου εργασίας και απλότητας του κώδικα. Για τον λόγο αυτόν τα δεδομένα του παίκτη αντιγράφονται όταν καλείται η `LoadProfile` και διατηρείται ένα τοπικό αντίγραφο το οποίο χρησιμοποιεί ο παίκτης μέχρι να βγει από το παιχνίδι. Όταν ο παίκτης βγει από το παιχνίδι το αντίγραφο αντικαθιστά τα δεδομένα στην βάση. Δημιουργείται δηλαδή ένα σύστημα `Caching` από το οποίο θα ανακτάται κάθε φορά όλος ο πίνακας με την `get`, θα διαφοροποιούνται τα επιθυμητά στοιχεία στο ανάλογο module και έπειτα θα ενημερώνουμε το τοπικό αντίγραφο με την `Update`. Με την εφαρμογή αυτής της αρχιτεκτονικής διατηρούνται χαμηλές οι κλήσεις στο API της βάσης ώστε να παραμείνει το παιχνίδι κάτω από το όριο που έχει θέσει το `roblox` αλλά και δεν καθυστερούν τα δεδομένα καθώς βρίσκονται ήδη στην cache. Τέλος θα γίνει κλήση ξανά στο API για αποθήκευση στην βάση μόνο όταν ο παίκτης φύγει από το παιχνίδι ή πέσει ο server.

### 3.5.2 Σύστημα διαχείρισης αντικειμένων

Εξίσου σημαντικό είναι και το σύστημα διαχείρισης αντικειμένων καθώς είναι από τα πιο σημαντικά συστήματα σε ένα παιχνίδι ρόλων. Ένα σύστημα διαχείρισης αντικειμένων ή αλλιώς `inventory` μπορεί να πάρει διάφορες μορφές. Μερικά παραδείγματα τέτοιων συστημάτων είναι ότι σε ένα σύστημα A γίνεται αποθήκευση του αριθμού των ίδιων αντικειμένων που έχει ένας παίκτης πάνω του ενώ σε ένα σύστημα B γίνεται αποθήκευση του κάθε αντικειμένου ως μοναδικού. Τέλος υπάρχουν και τρίτα συστήματα τα οποία εφαρμόζουν συνδυασμό των δύο. Για το παρόν παιχνίδι αποφασίστηκε ένας συνδυασμός. Ειδικότερα τα αντικείμενα του που αποτελούν σπαθιά, ασπίδες, πανοπλίες κ.λπ διατηρούνται ως μοναδικά και έχουν ως κλειδί ένα GUID<sup>21</sup>. Τα αντικείμενα αυτά έχουν διάφορες παραμέτρους όπως `durability`, `max durability`, `damage`, `defense` και πολλά άλλα. Ο λόγος που

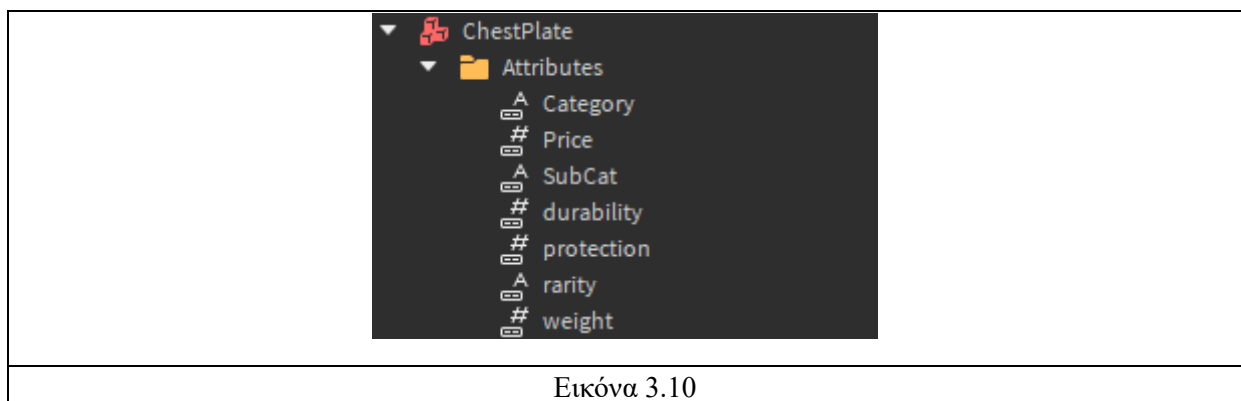
---

<sup>21</sup> globally unique identifier



επιλέχθηκε μία τέτοια προσέγγιση είναι έτσι ώστε να μπορούν να παραχθούν δύο αντικείμενα του ίδιου μοντέλου αλλά να έχουν διαφορετικά χαρακτηριστικά, εφαρμόζοντας δηλαδή και τον παράγοντα της τύχης. Ωστόσο ένα τέτοιο σύστημα θα ήταν ιδιαίτερα δύσχρηστο για την αποθήκευση πόρων καθώς οι πόροι θα βρίσκονται σε μεγάλες ποσότητες και δεν έχουν χαρακτηριστικά τα οποία τους κάνουν μοναδικούς. Ένα ξύλο για παράδειγμα μπορεί να είναι μόνο συγκεκριμένου είδους και δεν έχει χαρακτηριστικά όπως damage. Παρακάτω θα αναλυθούν τα δύο αυτά συστήματα αποθήκευσης και πως ενοποιούνται.

Αρχικά είναι αναγκαίο να διατηρείται κάπου ένα αντίγραφο κάθε αντικειμένου ώστε να μπορούν να δημιουργηθούν περαιτέρω αντίγραφα από αυτό για να δοθούν σε κάθε παίκτη. Τα αντικείμενα αυτά διατηρούνται σε υπό φακέλους μέσα στον ήδη υπάρχον φάκελο ServerStorage ο οποίος αποτελεί τον χώρο αποθήκευσης του server και οι clients δεν έχουν πρόσβαση σε αυτόν για λόγους που έχουν να κάνουν με την ασφάλεια. Η δημιουργία αντικειμένων στον φάκελο αυτόν γίνεται με τις τεχνικές περί δημιουργίας φυσικών αντικειμένων που περιεγράφηκαν παραπάνω. Για τα αντικείμενα τα οποία έχουν μοναδικά χαρακτηριστικά δημιουργείται ένας φάκελος μέσα σε κάθε αντικείμενο το οποίο περιέχει διάφορες τιμές. Όταν γίνει generate ένα νέο αντικείμενο αυτού του είδους αντιγράφονται όλα τα στοιχεία του φακέλου Attributes σε έναν πίνακα ο οποίος αργότερα θα αποθηκευτεί στο inventory του χρήστη και θα αποτελεί τα δεδομένα του συγκεκριμένου αντικειμένου. Σε μία SQL βάση δεδομένων ο πίνακας αυτός (item\_details) θα ήταν συνδεδεμένος χρησιμοποιώντας foreign key με το ID του του αντικειμένου το οποίο θα βρισκόταν στον πίνακα items. Πριν αποθηκευτεί όμως θα γίνει έλεγχος για το αν κάποιο στοιχείο χρειάζεται περαιτέρω επεξεργασία όπως για παράδειγμα η τυχαιοποίηση του damage ενός σπαθιού. Αφού γίνουν όλες οι επεξεργασίες και δημιουργηθεί ο τελικός πίνακας γίνεται append στον πίνακα δεδομένων που θα ληφθεί με την κλήση της συνάρτησης get που περιεγράφηκε παραπάνω και θα αποθηκευτεί με την update. Να τονιστεί ότι οι εργασίες της επεξεργασίας και της αποθήκευσης/λήψης του πίνακα γίνονται σε διαφορετικά modules για των διαχωρισμό των αρμοδιοτήτων.



Εικόνα 3.10

Αντίθετα η δημιουργία και η ανανέωση των πόρων είναι πολύ ευκολότερη. Όταν ένας παίκτης αποκτά έναν πόρο για πρώτη φορά δημιουργείται μία καταγραφή στο path Inventory.Resource η οποία θα περιέχει την ποσότητα του πόρου που αποκτήθηκε. Για παράδειγμα Resource = {"Oak" = 5, ...}. Όταν η ποσότητα αυτή αλλάξει επειδή ο χρήστης έκοψε ακόμα ένα δέντρο ή χρησιμοποίησε τον πόρο τότε αλλάζει απλά η τιμή της εγγραφής.

Μέρος του Inventory συστήματος είναι και η διαχείριση των αντικειμένων που έχει ο παίκτης εξοπλισμένα για τον λόγο αυτόν διατηρείται η εξής εγγραφή στην βάση:

```
Equipped={
  Helmet = {current = "",previous = ""},
  Chest = {current = "",previous = ""},
  Legs = {current = "",previous = ""},
  Shield = {current = "",previous = ""},
  Offensive = {current = "",previous = ""},
  Consumable = {current = "",previous = ""}
},
```

Στον πίνακα αυτόν διατηρείται το GUID του αντικειμένου που είναι εξοπλισμένο στην ανάλογη θέση καθώς και το προηγούμενο αντικείμενο που ήταν εξοπλισμένο. Όταν ο παίκτης εισέλθει στο παιχνίδι ελέγχεται αυτός ο πίνακας ώστε τα αντικείμενα τα οποία έχει εξοπλισμένα να αποκτήσουν υλική υπόσταση και να εμφανιστούν στον χαρακτήρα του. Το αντικείμενο που θα αποκτήσει θα αποτελεί αντίγραφο του ίδιου αντικειμένου που βρίσκεται στον ServerStorage. Παράλληλα θα κληθούν και οι initialization συνάρτησης των ανάλογων αντικειμένων ώστε ο χρήστης να μπορεί να πάρει στο σπαθί στα χέρια του από την θήκη και να εκτελέσει όλες τις ενέργειες του αντικειμένου όπως αυτήν της επίθεσης. Αυτό υλοποιείται κάνοντας bind διάφορα events κατά το initialization.

Ο παίκτης όμως δεν έχει άμεση πρόσβαση στον server οπότε είναι αδύνατο να καλέσει τις συναρτήσεις για να εξοπλίσει τα αντικείμενα. Για τον λόγο αυτό είναι αναγκαίο να δημιουργηθεί μία «γέφυρα» μεταξύ server και client χωρίς να εκθέσουμε τον server σε θέματα ασφαλείας. Πρέπει δηλαδή να διατηρήσουμε τα κρίσιμα δεδομένα στον server κρυφά από τον κάθε client και να επιτρέψουμε στον client να εκτελέσει συναρτήσεις μόνο αν τηρούνται συγκεκριμένοι όροι. Η γέφυρα αυτή δημιουργείται με την κατασκευή modules τα οποία αλληλοεπιδρούν με τα module του client με την χρήση remote event. Remote event είναι μέρος του API επικοινωνίας που παρέχει το roblox. Με την χρήση Remote events δίνεται η δυνατότητα αποστολής μηνυμάτων μεταξύ client και server. Τα μηνύματα αυτά περνάνε στον server με την μορφή κλήσης μίας συνάρτησης.

### 3.5.3 Σύστημα ανταλλαγής αντικειμένων

Για να διασφαλιστεί ο ρεαλισμός του παιχνιδιού και να παραμείνει η οικονομία δυναμική είναι απαραίτητο οι παίκτες να μπορούν να ανταλλάξουν αντικείμενα και πόρους μεταξύ τους. Αυτό διασφαλίζεται με την υλοποίηση ενός συστήματος ανταλλαγών. Ειδικότερα όταν ένας παίκτης θέλει να ανταλλάξει κάτι με κάποιον άλλον αρκεί να πληκτρολογήσει στο chat `!trade [player_name]` όπου `player_name` το όνομα του παίκτη με τον οποίο θέλει να ανταλλάξει. Τότε θα σταλθεί μία ειδοποίηση στον δεύτερο παίκτη ο οποίος θα αποδεχτεί ή θα απορρίψει την ανταλλαγή. Σε περίπτωση απόρριψης η συναλλαγή απορρίπτεται και στέλνεται ανάλογη ενημέρωση στον αποστολέα. Σε περίπτωση αποδοχής ανοίγει το παράθυρο ανταλλαγών στο οποίο ο κάθε παίκτης βλέπει τα αντικείμενά του και τα αντικείμενα τα οποία ο άλλος παίκτης έχει προσφέρει για ανταλλαγή. Αφού ο κάθε παίκτης αποφασίσει ποια αντικείμενα θέλει να ανταλλάξει πατάει το κουμπί `accept`. Αφού και οι δύο παίκτες πατήσουν `accept` ξεκινάει μία αντίστροφη μέτρηση 5 δευτερολέπτων κατά την οποία οι παίκτες μπορούν να ακυρώσουν την ανταλλαγή. Σε περίπτωση που κανείς δεν ακυρώσει τα αντικείμενα θα ανταλλαχθούν. Όταν αφορά φυσικά αντικείμενα απλά μεταφέρονται τα UID των αντικειμένων από τον έναν παίκτη στον άλλον. Αντίθετα όταν αφορά πόρους αρχικά ελέγχεται αν ο πόρος υπάρχει στην βάση δεδομένων του άλλου παίκτη. Αν υπάρχει τότε προσαρμόζεται η ποσότητα και στους δύο παίκτες, αν δεν υπάρχει δημιουργείται και ύστερα προσαρμόζεται η ποσότητα.

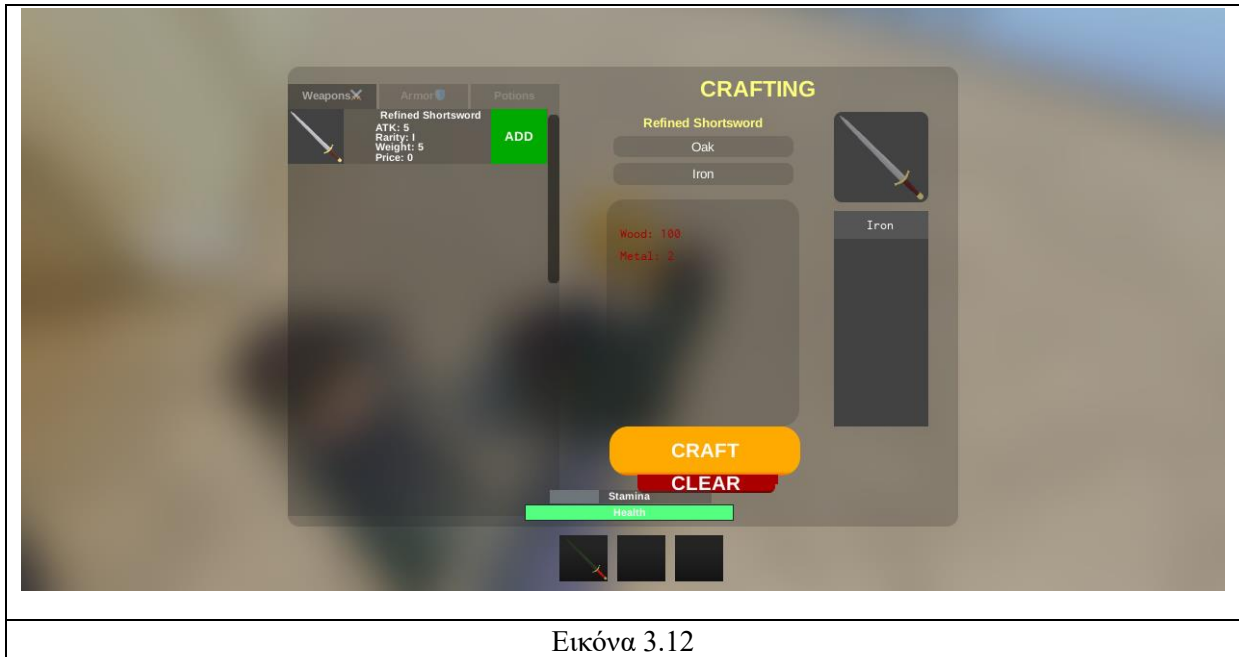
### 3.5.4 Συστήματα δημιουργίας αντικειμένων

Ιδιαίτερης σημασίας είναι και το σύστημα δημιουργίας αντικειμένων το οποίο περιλαμβάνει διάφορα NPC τα οποία εξυπηρετούν τον σκοπό των καταστηματαρχών και πωλούν βασικά αντικείμενα χαμηλής ποιότητάς όπως σιδερένια σπαθιά, πανοπλίες κλ.π. Το σύστημα αυτό είναι αρκετά απλό στην υλοποίηση. Ειδικότερα σε κάθε NPC δίνεται ένα tag χρησιμοποιώντας το σύστημα tag του roblox το οποίο προσδιορίζει το είδος του καταστήματος. Π.χ `blacksmith`, `tailor` και άλλα. Με αυτόν τον τρόπο όταν γίνεται `initialize` το σύστημα του καταστήματος παίρνει όλα τα στοιχεία τα οποία είναι αποθηκευμένα σε ένα `table` που περιέχει το όνομα του αντικειμένου και το κόστος του. Το σύστημα έπειτα δημιουργεί εικόνες του κάθε αντικειμένου στις οποίες αν ο παίκτης κάνει `hover` πάνω τους μπορεί να δει λεπτομέρειες του αντικειμένου καθώς και την τιμή του αντικειμένου. Πατώντας πάνω στην εικόνα αν ο παίκτης διαθέτει τον απαιτούμενο χρυσό αγοράζει το αντικείμενο.



Εικόνα 3.11

Παρόλο που ένα τέτοιο σύστημα μπορεί να είναι αρκετό για πολλά παιχνίδια στο συγκεκριμένο δεν θα καταφέρει μόνο του να στηρίξει την οικονομία. Ειδικότερα πρέπει να δοθεί ένας σκοπός στην απόκτηση πόρων πέρα από την πώλησή τους ώστε οι παίκτες να έχουν έναν λόγο να παίξουν παραπάνω το παιχνίδι αλλά και να ενθουσιάζονται όταν βρίσκουν έναν σπάνιο πόρο. Το πρόβλημα αυτό λύνει η ύπαρξη της δυνατότητας crafting αντικειμένων. Χάρη στο σύστημα αυτό οι παίκτες μπορούν να επιλέξουν τι αντικείμενο θέλουν να κατασκευάσουν και θα μπορέσουν να επιλέξουν τι πόρους θα χρησιμοποιήσουν. Η επιλογή πόρων είναι ιδιαίτερης σημασίας καθώς θα επηρεάσει τόσο την εμφάνιση του αντικείμενου όσο και τα στατιστικά του. Για παράδειγμα ένα σπαθί από διαμάντι θα έχει πολύ καλύτερα στατιστικά από ένα σπαθί φτιαγμένο από σίδηρο. Αφού ο παίκτης επιλέξει τους πόρους που θέλει να χρησιμοποιήσει πατάει το κουμπί craft. Βασική προϋπόθεση είναι να διαθέτει τους πόρους που έχει επιλέξει στις αναλογίες που ζητούνται. Αν η προϋπόθεση αυτή πληρείται οι πόροι θα αφαιρεθούν από τον παίκτη και θα γίνει generate ένα αντικείμενο με βάση το τι έχει επιλεγεί το οποίο θα προστεθεί στο inventory του παίκτη. Το αντικείμενο αυτό θα έχει τυχαία χαρακτηριστικά μέσα στο εύρος που έχουν οι πόροι που χρησιμοποιήθηκαν καθώς και όνομα το οποίο γίνεται generated από τους πόρους. Για παράδειγμα ένα “Standard sword” φτιαγμένο από Elm και Diamonds θα πάρει το όνομα “Magnificent Radiant Standard sword”. Τα υλικά από τα οποία είναι φτιαγμένο το σπαθί καταγράφονται στον πίνακα του σπαθιού με το κλειδί materials.



Εικόνα 3.12

### 3.5.5 Σύστημα απόκτησης πόρων

Για να αποκτήσει ο παίκτης πόρους είναι απαραίτητο να διαθέτει μία αξίνα για τα ορυκτά ή ένα τσεκούρι για τα ξύλα. Τα αντικείμενα αυτά μπορεί να τα αποκτήσει από τα διάφορα καταστήματα χρησιμοποιώντας το Gold που του παρέχεται όταν πρωτοπαίζει το παιχνίδι. Ο παίκτης πρέπει να πλησιάσει τον πόρο και όταν βρίσκεται πλέον σε κοντινή απόσταση θα του εμφανιστεί ένα Prompt να πατήσει το κουμπί E. Ο παίκτης αφού πατήσει και κρατήσει το κουμπί εμφανίζεται μία μπάρα η οποία δείχνει την πρόοδο του παίκτη. Όταν η μπάρα φτάσει το 100% υπολογίζεται το τι πόρος θα δοθεί στον παίκτη μέσω μιας συνάρτησης τυχαιοποίησης η οποία χρησιμοποιεί ως seed το τρέχον tick από την ημερομηνία EPOCH<sup>22</sup>. Ο υπολογισμός γίνεται υπολογίζοντας έναν τυχαίο float αριθμό από το 0 έως το 100. Στο διάστημα αυτό υπάρχουν υπό διαστήματα τα οποία αντιστοιχούν σε διαφορετικούς πόρους. Αφού επιλεγθεί ο αριθμός από το seed ελέγχεται σε ποιο διάστημα αντιστοιχεί ο αριθμός και δίνεται ο αντίστοιχος πόρος η ποσότητα του οποίου θα είναι και πάλι τυχαία από 1 έως 5.

### 3.5.6 Σύστημα υποβοήθησης παικτών

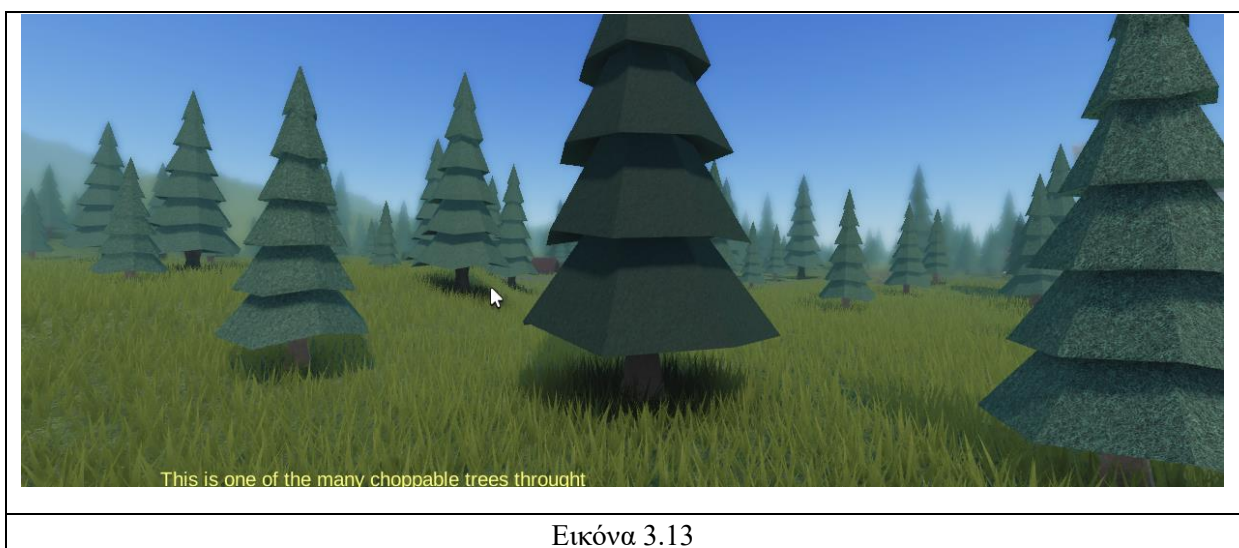
Βασικός σκοπός της παρούσας διπλωματικής εργασίας ωστόσο είναι η υποβοήθηση των παικτών. Ο σκοπός αυτός εξυπηρετείται μέσω της δημιουργίας ενός συστήματος το οποίο κάνει παρακολούθηση της προόδου των παικτών. Ειδικότερα έχουν μπει μερικά καρφωτά δεδομένα τα οποία αφορούν την απόκτηση πόρων και χρυσού. Το σύστημα περιοδικά ελέγχει πόσους πόρους, χρυσό κ.α έχει συλλέξει ο παίκτης στην συγκεκριμένη χρονική περίοδο. Αν οι ποσότητες αυτές είναι κάτω από

<sup>22</sup> Δευτερόλεπτα από 1<sup>η</sup> Ιανουαρίου, 1970

αυτές που έχουν δοθεί στο σύστημα ως «βάση» τότε στέλνεται ειδοποίηση στον παίκτη με τα ανάλογα μηνύματα ώστε να τον ωθήσουν να προχωρήσει στο παιχνίδι. Για παράδειγμα αν ανιχνευτεί από το σύστημα ότι ο παίκτης έχει μαζέψει λιγότερο από 500 χρυσό τα τελευταία 30 λεπτά τότε θα του σταλθεί ειδοποίηση “You are falling behind on gold! There are various ways you can earn gold through selling ores, wood or trading your items with other players!”. Η παρακολούθηση αυτή του παίκτη γίνεται με το initialization του monitoring module για να διατηρηθούν οι αρχές του SOLID. Εκεί εκτελούνται οι διάφορες ενέργειες παρακολούθησης ανά τακτά χρονικά διαστήματα και εκδίδονται οι ανάλογες ειδοποιήσεις με κλήσεις στο notifications modules. Για το monitoring το σύστημα ξεκινά ένα διαφορετικό Thread ώστε να υπάρχει ταχύτερη παράλληλη επεξεργασία και να μην καθυστερεί άλλες διαδικασίες.

Ωστόσο η παροχή βοήθειας δεν μπορεί να στηριχθεί μονάχα στην παροχή τέτοιων tips καθώς οι παίκτες πρέπει να μάθουν το παιχνίδι με κάποιον τρόπο. Για τον λόγο αυτό έχει δημιουργηθεί ένα tutorial το οποίο αποτελείται από πολλαπλά cutscenes που δείχνουν στον παίκτη τον κόσμο και τις βασικές λειτουργίες. Παρουσιάζουν στον παίκτη τα βασικά κουμπιά, τις τοποθεσίες απόκτησης πόρων, την κεντρική περιοχή του παιχνιδιού και προωθούν τον παίκτη στο να γίνει μέλος ενός οίκου ώστε να ενισχύσει την κοινότητα. Το κουμπί για το tutorial είναι τοποθετημένο στην επάνω μπάρα σε εμφανές σημείο ώστε να είναι προσβάσιμο ανά πάσα στιγμή σε περίπτωση που κάποιος παίκτης ξεχάσει κάτι.

Τέλος εκτός από την «συστημική» βοήθεια ιδιαίτερης σημασίας ήταν και η βοήθεια την οποία παρείχαν οι παίκτες της κοινότητας οι οποίοι θέλοντας να μεγαλώσουν περαιτέρω την κοινότητα έδειχναν σε άλλους παίκτες τι ενέργειες έπρεπε να κάνουν λαμβάνοντας τον ρόλο του δασκάλου. Η τελευταία ήταν η πιο αποτελεσματική καθώς λόγω της ανθρώπινης επαφής και νοημοσύνης υπερίσχυε κάθε άλλου είδους παροχής συμβουλών.



Εικόνα 3.13

### 3.5.7 Σύστημα Μάχης

Το σύστημα μάχης είναι ένα από τα μεγαλύτερα και πιο σημαντικά κομμάτια του παιχνιδιού. Παρατηρήθηκε ότι ένα ιδιαίτερα μεγάλο μέρος των παικτών αποφάσιζε αν θα παίζει ή όχι το παιχνίδι βασισμένο στην ποιότητα και την δυσκολία του συστήματος μάχης. Πολλά παρόμοια παιχνίδια έχουν ιδιαίτερα απλοποιημένα συστήματα μάχης ή συστήματα τα οποία βασίζονται σε μεγάλο παράγοντα στην τύχη. Ωστόσο αυτό δεν θα έδινε στους παίκτες την επιθυμία να βελτιώσουν τις ικανότητές τους και να θέσουν ως στόχο να γίνουν οι καλύτεροι μαχητές. Για τον λόγο αυτόν δημιουργήθηκε ένα σύστημα το οποίο περιλαμβάνει 3<sup>ων</sup> ειδών επιθέσεις και μία κίνηση απόκρουσης. Αρχικά θα αναλυθεί το πως ο παίκτης επιλέγει την επίθεση την οποία θα χρησιμοποιήσει και έπειτα γιατί αυτές οι επιθέσεις διαφέρουν μεταξύ τους. Το σύστημα διατηρεί ένα timestamp στο οποίο καταγράφεται κάθε φορά η χρονική στιγμή στην οποία ο χρήστης έκανε την τελευταία επίθεση. Για το timestamp αυτό χρησιμοποιείται η συνάρτηση `os.time()` η οποία επιστρέφει τα δευτερόλεπτα από το EPOCH. Επίσης να τονιστεί ότι η συνάρτηση αυτή καλείται από τον server γιατί σε διαφορετική περίπτωση θα διατηρούταν η διαφορά της τοπικής ώρας και θα επηρέαζε σημαντικά ακεραιότητα του συστήματος. Έπειτα όταν ο παίκτης αποφασίσει ότι θέλει να επιτεθεί ξανά και πατάει το κουμπί επίθεσης το οποίο είναι το αριστερό κλικ του ποντικιού, το σύστημα ελέγχει τότε έγινε η τελευταία επίθεση. Αν η διαφορά είναι λιγότερη από δύο δευτερόλεπτα τότε περνάει στην επόμενη επίθεση αλλιώς ξεκινάει πάλι από την αρχή. Για παράδειγμα έστω τα είδη επίθεσης A,B και Γ. Αν ο παίκτης εκτελέσει την A και εντός δύο δευτερολέπτων επιτεθεί ξανά θα εκτελεστεί η B. Σε διαφορετική περίπτωση, δηλαδή μετά το πέρας των δύο δευτερολέπτων θα εκτελεστεί ξανά η A. Παρακάτω αναλύονται περεταίρω τα είδη επίθεσης.

Η πρώτη επίθεση είναι ένα animation από την δεξιά προς την αριστερή μεριά και δεύτερη από την αριστερή προς την δεξιά.



Εικόνα 3.14



Εικόνα 3.15

Τα animation και η εμβέλεια αυτών των δύο επιθέσεων δεν διαφέρουν παρά μόνο στην πλευρά από την οποία εκτελούνται. Ωστόσο αυτό δεν σημαίνει ότι είναι ίδιες καθώς ο παίκτης γυρνώντας λίγο το σώμα του μπορεί να πετύχει τον αντίπαλο με μία από τις δύο αυτές επιθέσεις ενώ με την άλλη δεν θα μπορούσε τόσο εύκολα. Η Τρίτη επίθεση είναι η πιο σημαντική σε περίπτωση που ο χρήστης την χρησιμοποιήσει επιτυχώς καθώς έχει αυξημένη εμβέλεια και πολλαπλασιαστή damage 1.02 το οποίο σε σπαθιά με υψηλά στατιστικά κάνει την διαφορά.



Εικόνα 3.16

Τέλος μία εξίσου σημαντική κίνηση είναι αυτή της απόκρουσης. Όταν ένας παίκτης A επιτεθεί σε έναν παίκτη B ο παίκτης B έχει την δυνατότητα πατώντας το δεξί κλικ να απόκρουση την επίθεση αν το πατήσει την σωστή χρονική στιγμή. Αν η επίθεση αποκρουστεί επιτυχώς ο παίκτης A θα μπει σε μία κατάσταση «ζάλης» η οποία θα τον αποτρέψει από το να πραγματοποιήσει οποιαδήποτε κίνηση για ~1 δευτερόλεπτο κάνοντάς τον ευάλωτο σε επιθέσεις. Φυσικά υπάρχει και μία περίοδος αναμονής ανάμεσα στις επιθέσεις και τις απόκρουσης η οποία ισοδυναμεί με 1.2 s – την



διάρκεια του animation το οποίο συνήθως είναι από 0.2 έως 0.5 s. Η περίοδος αυτή υπολογίζεται στον συνολικό χρόνο των επιθέσεων για τα δύο δευτερόλεπτα.

## ΚΕΦΑΛΑΙΟ 4

### ΤΡΟΠΟΙ ΜΕ ΤΟΥΣ ΟΠΟΙΟΥΣ ΠΑΙΖΕΤΑΙ ΤΟ ΠΑΙΧΝΙΔΙ

#### 4.1 Βασικός τρόπος παιχνιδιού

Όπως αναφέρθηκε σε προηγούμενα κεφάλαια υπάρχουν πολλοί τρόποι για να παίζει κάποιος το παιχνίδι και το πιο πιθανό είναι ότι κανείς δεν θα μπορέσει να παίζει όλους τους πιθανούς ρόλους. Κατά την διάρκεια των δοκιμών παρατηρήθηκαν διάφοροι ρόλοι τους οποίους αποφάσισαν να πάρουν οι παίκτες. Οι βασικότεροι εξ' αυτών είναι:

- Έμπορος
- Λόρδος
- Ληστής
- Στρατιώτης
- Ιερέας
- Περιπλανώμενος

Παρόλο που πολλοί από τους παραπάνω ρόλους είναι εύκολα κατανοητοί, υπάρχει και ένας ο οποίος δημιουργεί ερωτήματα. Αυτός είναι ο ρόλος του περιπλανώμενου.

Περιπλανώμενος σε αυτό το παιχνίδι ορίζεται συνήθως ένας νέος παίκτης ο οποίος δεν έχει αποφασίσει τι θα κάνει. Ο περιπλανώμενος δεν ανήκει σε κάποιον οίκο οπότε δεν μπορεί να αποκτήσει κάποιον ρόλο ο οποίος προϋποθέτει να βρίσκεται σε οίκο. Για αυτόν τον λόγο παρατηρήθηκε ότι αυτοί οι παίκτες συνήθως ακολουθούσαν συνδυασμό άλλων ρόλων κατά την διάρκεια εκμάθησης του παιχνιδιού. Ένας συνδυασμός που εντοπίστηκε συχνά είναι αυτός του ληστή/εμπόρου. Ο συνδυασμός αυτός παρέχει πολλά από αυτά που μπορεί να προσφέρει το παιχνίδι καθώς δίνει στον παίκτη την δυνατότητα να συμμετέχει ενεργά στην οικονομία ενώ παράλληλα μπορεί να συμμετέχει σε καταλήψεις δρόμων, ορυχείων και με απλούστερα λόγια να βιώσει το σύστημα μάχης του παιχνιδιού.

Με την πρόοδο του παίκτη στο παιχνίδι όμως δημιουργούταν η επιθυμία να συμμετέχει στο πιο κοινωνικό μέρος του παιχνιδιού το οποίο είναι οι οίκοι. Συνήθως οι παίκτες ξεκινούσαν ως στρατιώτες σε κάποιον οίκο καθώς είναι ένας ρόλος αναλώσιμος και σε αφθονία. Επίσης για να γίνει κάποιος στρατιώτης δεν ήταν αναγκαίο να έχει γνωριμίες το οποίο είναι πολύ σπάνιο για έναν νέο παίκτη. Αφού ο παίκτης γινόταν μέλος ενός οίκου μπορούσε πλέον με την πάροδο του χρόνου και τις

κατάλληλες γνωριμίες να προωθηθεί σε μεγαλύτερες και πιο περιζήτητες θέσεις όπως αυτή του διοικητή στρατού ενός οίκου ή του ιπότη.

Εδώ βρίσκεται ένα κομβικό σημείο για την πρόοδο του παίκτη. Όταν δηλαδή ο παίκτης προαχθεί σε μία υψηλόβαθμη θέση και παραμείνει εκεί για κάποιο χρονικό διάστημα έχει πλέον να κάνει μία επιλογή. Να ιδρύσει έναν δικό του οίκο χρησιμοποιώντας τις γνωριμίες και τους πόρους που έχει συγκεντρώσει ή να προσπαθήσει να εξελιχθεί στον οίκο που βρίσκεται τώρα. Σε περίπτωση που επιλέξει το δεύτερο γρήγορα οδηγείται σε μία κατάσταση αδιεξόδου σε θέμα προόδου εκτός αν προβεί σε πράξεις προδοσίας. Ο λόγος που συμβαίνει αυτό είναι επειδή στον μεσαίωνα λόρδος ενός οίκου γινόταν ο γιός του λόρδου μετά τον θάνατό του (στο παιχνίδι ο θάνατος έρχεται όταν αποφασίσει ο παίκτης να σταματήσει να παίζει, αν εκτελεστεί από τον βασιλιά ή αν ο ίδιος αποφασίσει να τελειώσει την ζωή του χαρακτήρα του). Γι' αυτό συνήθως γίνονταν παίκτες οι οποίοι γνώριζαν τον λόρδο στον πραγματικό κόσμο. Έτσι τα υπόλοιπα υψηλόβαθμα μέλη του στρατού δεν μπορούσαν να αποκτήσουν την αρχοντιά εκτός και αν προδώσουν τον οίκο μέσω εξέγερσης ή έξυπνων πολιτικών κινήσεων. Αντίθετα αν επέλεγε να ιδρύσει τον δικό του οίκο θα μπορούσε πλέον να γίνει λόρδος, να επιβάλει τους δικούς του κανόνες, να δημιουργήσει συμμαχίες και να κυνηγήσει τον θρόνο. Οτιδήποτε όμως και αν επιλέξει ο παίκτης δεν οδηγεί στην λήξη του παιχνιδιού. Το μόνο που θα περιοριστεί είναι ο πρόοδος του αλλά αφού το παιχνίδι είναι ένα παιχνίδι ρόλων ο σκοπός δεν έχει να κάνει τόσο με την πρόοδο όσο με την ψυχαγωγία και την αλληλεπίδραση με άλλους παίκτες.

Ο παραπάνω αποτελεί τον κύριο τρόπο παιχνιδιού τον οποίο φάνηκε να επιλέγουν οι περισσότεροι παίκτες. Ωστόσο υπάρχουν και πιο ειδικές περιπτώσεις οι οποίες αξίζουν περαιτέρω ανάλυση.

## **4.2 Ο ρόλος των Λόρδων**

Ο Λόρδος είναι συνήθως ο πιο επιθυμητός ρόλος σε αυτό το παιχνίδι επειδή ξεκλειδώνει αμέτρητες πιθανότητες για την πρόοδο του παίκτη. Όταν κάποιος γίνει λόρδος μπορεί να διοικεί τον δικό του στρατό, να κλείνει συμφωνίες με άλλους οίκους και κυριότερο να κυνηγήσει την βασιλεία η οποία θα του δώσει ακόμα περισσότερες εξουσίες. Ωστόσο για να γίνει κάποιος λόρδος πρέπει να πληροί κάποιες προϋποθέσεις. Αυτές είναι να έχει αρκετούς πόρους για να συντηρήσει τον οίκο του σε θέματα εξοπλισμού, να έχει τον απαραίτητο αριθμό στρατιωτών στις βαθμίδες του ώστε να μπορέσει να διατηρήσει την μικρή εξουσία που του δίνεται και τέλος να έχει τις κατάλληλες γνωριμίες με άλλα μέλη της κοινότητας τα οποία θα τον προωθήσουν.

### **4.3 Έμποροι ή συλλέκτες**

Πολλοί παίκτες παρατηρήθηκε ότι αδιαφόρησαν εντελώς για την βασιλεία και την αλληλεπίδραση που παρέχει το σύστημα των οίκων. Αντίθετα εστίασαν στην αύξηση της περιουσίας τους τόσο σε θέμα χρυσού, πόρων αλλά και στην απόκτηση σπάνιων αντικειμένων. Οι παίκτες αυτοί ξόδευαν μεγάλο μέρος του χρόνου τους κάνοντας εξόρυξη ορυκτών ή κόβοντας δέντρα προσπαθώντας να βρουν τους πιο σπάνιους πόρους στις μεγαλύτερες ποσότητες. Έπειτα αντάλλαζαν τους πόρους αυτούς με άλλους παίκτες με χρηματικά ανταλλάγματα ή κατασκεύαζαν δικά τους αντικείμενα τα οποία διατηρούσαν για την προσωπική τους συλλογή ή πωλούσαν σε άλλους παίκτες σε περίπτωση που είχαν πολύ καλά στατιστικά. Οι παίκτες αυτοί πολλές φορές μπορεί να ειπωθεί ότι εξυπηρετούσαν και τον ρόλο της τράπεζας καθώς έδιναν δάνεια σε άλλους παίκτες ή οίκους και έτσι ήταν ιδιαίτερα δημοφιλείς στην κοινότητα. Χωρίς να συμμετέχουν στο μεγαλύτερο σύστημα των οίκων όμως προέκυψε το πρόβλημα της προσέλκυσης του ενδιαφέροντος αυτών των παικτών ώστε να συνεχίσουν να παίζουν το παιχνίδι ακόμα και όταν έχουν πολύ μεγάλη περιουσία. Αυτό επιλύεται με την περιοδική προσθήκη νέων όλο και πιο σπάνιων πόρων δημιουργώντας έτσι το αίσθημα της απόκτησης στους παίκτες αυτούς καθώς είναι ο μόνος στόχος τον οποίο ενεργά κυνηγάνε.

### **4.4 Λοιποί παίκτες**

Οι υπόλοιποι παίκτες ακολουθούσαν μικρότερους ρόλους καθώς φάνηκε ότι αδιαφορούσαν για το γενικό σύστημα των οίκων όσον αφορά το αίσθημα της προόδου αλλά και το σύστημα των πόρων. Αντίθετα αναλάμβαναν ρόλους όπως ιερείς, ναυτικοί κ.α. οι οποίοι δεν είχαν ιδιαίτερα χαρακτηριστικά. Οι παίκτες αυτοί φάνηκε να απολαμβάνουν το παιχνίδι απλά παίζοντας με τον ρόλο του χαρακτήρα τους και αλληλοεπιδρώντας με τους άλλους παίκτες. Επιπλέον παρατηρήθηκε ότι οι παίκτες αυτοί είχαν τις καλύτερες υποκριτικές ικανότητες και ήταν αυτοί που έκαναν τον κόσμο να μοιάζει πραγματικά ζωντανός.

### **4.5 Διαφορές μεταξύ διαφορετικών ειδών παικτών**

Η βασικότερες διαφορές στο παιχνίδι είναι ανάμεσα σε έναν παίκτη που παίζει μόνος του και σε έναν που παίζει με παρέα. Καθώς το παιχνίδι είναι MMORPG με αρκετά στοιχεία τα οποία το φέρνουν πιο κοντά στην αληθινή ζωή είναι μονάχα φυσικό το παιχνίδι να είναι πιο διασκεδαστικό όταν κάποιος παίζει με παρέα. Αυτό συμβαίνει γιατί ο άνθρωπος ως κοινωνικό ον αναζητά την κοινωνικοποίηση και την συνεργασία. Οι διαφορές στον τρόπο με τον οποίο κάποιος παίζει το παιχνίδι μόνος του ενάντια σε κάποιον που παίζει με παρέα είναι ξεκάθαρες. Όταν κάποιος παίζει το παιχνίδι μόνος του δεν έχει την ανάλογη βοήθεια και έτσι η πρόοδος του παίκτη μπορεί να μειωθεί σημαντικά. Όχι μόνο η πρόοδος μειώνεται σημαντικά αλλά μπορεί να αποτρέψει και τον παίκτη τελείως από το

παιχνίδι σε περίπτωση που δεν μπορέσει να καταλάβει το βαθύ σύστημα των οίκων. Αντίθετα όταν κάποιος παίζει με παρέα υπάρχει η πιθανότητα οι φίλοι του να γνωρίζουν ήδη το παιχνίδι και να του δείξουν πως παίζεται. Επιπλέον μπορούν να τον βοηθήσουν σημαντικά σε θέματα πόρων ή χρημάτων καθώς μπορούν να εργαστούν ομαδικά και να παράγουν περισσότερους πόρους και χρήμα τα οποία θα διαθέσουν για την πρόοδο της «ομάδας». Επιπροσθέτως όταν κάποιος είναι μόνος του συχνά μπορεί να γίνει στόχος ή να αποκλειστεί από περιοχές όπως τα ορυχεία τα οποία μπορεί να έχουν καταλείψει μεγαλύτερες ομάδες. Το ανάλογο δεν ισχύει απαραίτητα όταν κάποιος παίζει με παρέα καθώς έχοντας περισσότερα μέλη στην ομάδα μπορούν να πολεμήσουν για τον έλεγχο τέτοιων περιοχών.

Εκτός από τα ξεκάθαρα πλεονεκτήματα που έχουν οι παίκτες όταν παίζουν με παρέα παρατηρήθηκε ότι αυτοί είναι και οι παίκτες οι οποίοι παίζουν το παιχνίδι για περισσότερο καιρό και για περισσότερες ώρες. Αυτό οφείλεται στο γεγονός που αναφέρθηκε παραπάνω ότι οι άνθρωποι είναι κοινωνικά όντα και σε συνδιασμό με την προώθηση της κοινωνικότητας μέσω του περίπλοκου συστήματος αλληλεπιδράσεων του παιχνιδιού γρήγορα οδηγεί στο παραπάνω αποτέλεσμα. Με αυτόν τον τρόπο οι παίκτες που παίζουν μόνοι τους συνήθως οδηγούνται στο να πάρουν ρόλους όπως αυτόν του συλλέκτη καθώς απαιτούν ελάχιστες γνωριμίες και οι στόχοι των συλλεκτών είναι ατομικοί.

#### **4.6 Κακόβουλοι παίκτες**

Όπως και σε κάθε παιχνίδι υπήρχαν και εδώ περιπτώσεις κακόβουλων παικτών. Παικτών δηλαδή οι οποίοι δεν τηρούσαν τους κανόνες είτε της κοινότητας είτε του παιχνιδιού. Τα δύο βασικά είδη κακόβουλων παικτών είναι τα εξής:

- Exploiters
- Rule breakers

Ως exploiter χαρακτηρίζεται κάποιος ο οποίος είτε χρησιμοποιεί τρίτα προγράμματα είτε χρησιμοποιεί ατέλειες στα συστήματα / χάρτη ώστε να αποκτήσει ένα άδικο πλεονέκτημα. Μερικά Exploit από αυτά που παρατηρήθηκαν ήταν σε μορφή speed exploit δηλαδή δίνανε αφύσικη ταχύτητα στον χαρακτήρα τους, noclip δηλαδή να περνάνε μέσα από στερεά αντικείμενα, και εκμετάλλευση ατελειών του συστήματος μάχης ώστε να έχουν παραπάνω range στα χτυπήματά τους. Αυτή η κατηγορία παικτών ήταν εύκολο και γρήγορο να εντοπιστεί καθώς είναι επαναλαμβανόμενη συμπεριφορά και συνήθως αντιμετωπιζόταν με αποβολή από το παιχνίδι.

Αντίθετα το μεγάλο πρόβλημα δημιουργούσαν οι Rule breakers δηλαδή οι παίκτες οι οποίοι δεν ακολουθούσαν τους κανόνες. Τέτοια παραδείγματα είναι να σκοτώνουν άλλους παίκτες μέσα στο safezone ή σε περιοχές όπου ένας άνθρωπος στην πραγματική ζωή δεν θα έκανε επιθέσεις αν τον ένοιαζε η ζωή του. Αυτή η κατηγορία παικτών δεν αντιμετωπιζόταν με αποβολή καθώς αφού είναι κανόνες του παιχνιδιού υπεύθυνοι για την εφαρμογή τους είναι η αστυνομία και η βασιλεία.

Αφαιρώντας την δυνατότητα των παικτών να παραβούν αυτούς τους κανόνες αυτομάτως αφαιρούμε και τον σκοπό ύπαρξης της αστυνομίας καθώς θα μειωνόταν δραματικά η αλληλεπίδρασή της με άλλους παίκτες.

## ΚΕΦΑΛΑΙΟ 5

### ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Το παιχνίδι επέτυχε τον αρχικό του στόχο να προσφέρει ψυχαγωγία παρά την πολυπλοκότητά του, καθώς έχουν υλοποιηθεί τα κατάλληλα συστήματα για την εκμάθηση των παικτών. Με την προσφορά πολλαπλών πολύπλοκων συστημάτων, τα οποία αντιγράφουν δυνατότητες του πραγματικού κόσμου, δημιουργήθηκε ένα παιχνίδι πλούσιο σε περιεχόμενο και βάθος. Αυτά τα συστήματα περιλαμβάνουν μηχανισμούς που προσομοιώνουν πραγματικές καταστάσεις και αλληλεπιδράσεις, δίνοντας στους παίκτες τη δυνατότητα να εξερευνήσουν και να μάθουν μέσα από την εμπειρία τους. Επιπλέον, η ύπαρξη λεπτομερών οδηγιών και καθοδηγητικών εργαλείων επιτρέπει στους παίκτες να κατανοήσουν και να αφομοιώσουν την πολυπλοκότητα του παιχνιδιού, καθιστώντας το προσιτό και απολαυστικό για ένα ευρύ φάσμα παικτών. Η ενσωμάτωση αυτών των στοιχείων καθιστά το παιχνίδι μια ολοκληρωμένη και συναρπαστική εμπειρία.

Συγκεκριμένα, οι παίκτες φάνηκε να ξοδεύουν πολλές ώρες από τον χρόνο τους παίζοντας το παιχνίδι, αλλά και για να βελτιώσουν την κοινότητα. Η αφοσίωσή τους δεν περιορίστηκε μόνο στην προσωπική τους εξέλιξη μέσα στο παιχνίδι, αλλά επεκτάθηκε και στην ενεργό συμμετοχή τους στη βελτίωση και την ανάπτυξη της κοινότητας των παικτών. Μέσα από συνεργατικές δραστηριότητες, και ανταλλαγή γνώσεων, οι παίκτες συνέβαλαν στη δημιουργία ενός ζωντανού και υποστηρικτικού περιβάλλοντος. Αυτή η συλλογική προσπάθεια βελτίωσε την ποιότητα του παιχνιδιού δημιουργώντας μια ισχυρή κοινότητα.

Επιπλέον, εφαρμόζοντας τις διάφορες τεχνικές δημιουργίας εισοδήματος, επιτυγχάνεται η στήριξη του παιχνιδιού ώστε να είναι δυνατές οι μελλοντικές επεκτάσεις χωρίς να δυσχεραίνεται η εμπειρία των παικτών. Αυτές οι τεχνικές περιλαμβάνουν τη χρήση μικρο-πληρωμών και την πώληση ψηφιακών αγαθών, επιτρέποντας στο παιχνίδι να παραμείνει οικονομικά βιώσιμο και να συνεχίσει να εξελίσσεται. Η προσεκτική εφαρμογή αυτών των μοντέλων εισοδήματος εξασφαλίζει ότι οι παίκτες δεν αισθάνονται πιεσμένοι ή υποχρεωμένοι να ξοδέψουν χρήματα, διατηρώντας έτσι την ισορροπία ανάμεσα στην κερδοφορία και την ποιότητα της εμπειρίας τους. Με αυτόν τον τρόπο, το παιχνίδι μπορεί να επεκταθεί και να βελτιωθεί συνεχώς, προσφέροντας νέες δυνατότητες και περιεχόμενο, ενώ παράλληλα διατηρεί την αφοσίωση και την ικανοποίηση των παικτών.

Άλλα συστήματα και λειτουργικότητες οι οποίες έχουν σχεδιαστεί και θα εφαρμοστούν στο μέλλον είναι οι παρακάτω:

### **5.1 Σύστημα δημιουργίας κάστρων**

Ένα σύστημα δημιουργίας κάστρων ήταν μία από τις αρχικές ιδέες για αυτό το παιχνίδι ωστόσο διάφοροι περιορισμοί τόσο χρονικοί όσο και χρηματικοί το κατέστησαν απαγορευτικό για την αρχική έκδοση. Ειδικότερα θα δοθεί η δυνατότητα σε κάθε οίκο να διαθέτει το δικό του κάστρο στην γη την οποία έχει καταλάβει. Ο λόρδος του οίκου κάνοντας χρήση των πόρων τους οποίους έχει συγκεντρώσει θα μπορεί να επεκτείνει το κάστρο του τοποθετώντας τοίχους, φάρμες κ.α. αυξάνοντας το εισόδημά του και την προστασία του ενάντια σε επιθέσεις. Φυσικά η εφαρμογή τέτοιου συστήματος θα σήμαινε πως οι μάχες δεν θα λαμβάνουν πλέον χώρα σε τυχαία γεννημένους χάρτες όπως γίνεται τώρα αλλά στην περιοχή γύρω και μέσα στο κάστρο. Το σύστημα αυτό θα φέρει πολλούς νέους παίκτες και θα δώσει μία νέα διάσταση στο παιχνίδι. Απαραίτητη θα είναι και η δημιουργία ενός εκτεταμένου οδηγού χρήσης αυτού του συστήματος αλλά και κανόνων οι οποίοι θα απαγόρευαν τα απόρρητα κάστρα η κατασκευή των οποίων θα δυσχέραινε την εμπειρία όλων.

### **5.2 Περισσότερες γλώσσες διεπαφής**

Καθώς το παιχνίδι παίζουν άτομα από όλον τον κόσμο είναι επιθυμητό να μην γίνεται το παιχνίδι απαγορευτικό όταν κάποιος δεν γνωρίζει Αγγλικά. Το ζήτημα αυτό όμως είναι ιδιαίτερα περίπλοκο καθώς παρόλο που ο παίκτης θα μπορεί πλέον να καταλαβαίνει τι αναγράφεται στην διεπαφή και στις βοήθειες καλύτερα, θα παραμένει και πάλι στο περιθώριο όσον αφορά τις σχέσεις με τους άλλους παίκτες καθώς η πιθανότητα να μιλάνε την ίδια γλώσσα είναι μικρή. Η υλοποίηση του συστήματος αυτού θα γίνει με την προσθήκη αρχείων με μορφή JSON τα οποία θα περιέχουν τα λεκτικά στις διάφορες γλώσσες που θα υποστηρίζει το παιχνίδι. Τέλος θα προστεθεί ένα μενού επιλογής γλώσσας ενώ προεπιλεγμένη γλώσσα θα είναι η γλώσσα της χώρας στην οποία είναι εγγεγραμμένος ο λογαριασμός του παίκτη εκτός και αν δεν υπάρχει όπου προεπιλεγμένα θα γίνονται τα Αγγλικά.



## BIBΛΙΟΓΡΑΦΙΑ

- [1] Moore, M.E. (2016) Basics of Game Design. A K Peters/CRC Press
- [2] Irdeto global gaming survey report. (2021). Irdeto.
- [3] Menzel, J., Meier, M., & Maier, C. (2023). Online Gaming and Personality: Explaining Gamers' Cheating Intention.
- [4] <https://www.gabrielgambetta.com/client-server-game-architecture.html>
- [5] <https://www.gabrielgambetta.com/client-side-prediction-server-reconciliation.html>
- [6] Wikipedia, <https://www.wikipedia.org/>
- [7] Fusion Framework, <https://github.com/dphfox/Fusion>
- [8] Profile Service Framework, <https://github.com/MadStudioRoblox/ProfileService/>
- [9] Statista, <https://www.statista.com/statistics/1192573/daily-active-users-global-roblox/>
- [10] Roblox Documentation, <https://create.roblox.com/docs/reference/engine>
- [11] Roblox fandom, <https://roblox.fandom.com/>
- [12] Roblox Developer Forum, <https://devforum.roblox.com/>