

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Μηχανικών Πληροφορικής



Πτυχιακή Εργασία

Τίτλος Πτυχιακής: Παιχνίδι επαυξημένης  
πραγματικότητας τύπου PokemonGo

Τίτλος Παιχνιδιού: HeraklionGo

Αλέξανδρος Κολουδάς(ΑΜ:3685)

Επιβλέπων καθηγητής: Κ. Ιωάννης Παχουλάκης

ΗΡΑΚΛΕΙΟ  
2017

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον υπεύθυνο μου καθηγητή Κ. Ιωάννη Παχουλάκη που δέχτηκε την ιδέα μου για το συγκεκριμένο παιχνίδι και για την σωστή καθοδήγηση του, καθώς και όλα τα υπόλοιπα άτομα-παράγοντες που συνέβαλλαν στην ολοκλήρωση της πτυχιακής μου.

Αλέξανδρος Κολυδάς

## Σύνοψη

Η συγκεκριμένη πτυχιακή εργασία αφορά την ανάπτυξη ενός augmented reality παιχνιδιού το οποίο υλοποιήθηκε με την βοήθεια της μηχανής ανάπτυξης παιχνιδιών Unity και των γλωσσών προγραμματισμού C# και PHP.

Στο πλαίσιο αυτής της πτυχιακής θα γίνει μια σύντομη αναφορά στο περιβάλλον της Unity καθώς και μια αναφορά στο παιχνίδι έτσι ώστε να μπορέσει να καταλάβει όποιος διαβάσει αυτό το αρχείο τι ακριβώς παιχνίδι είναι και πώς μπορεί να το παίξει. Ο σκοπός του παιχνιδιού είναι ο κάθε χρήστης ψάχνοντας στην πόλη του Ηρακλείου να βρει σε συγκεκριμένες τοποθεσίες μέσω του GPS τα ανάλογα μαγαζιά όπου μπορεί να αγοράσει σπαθιά και ασπίδες για να γίνει πιο δυνατός, έτσι ώστε, όταν βρει σε συγκεκριμένες τοποθεσίες, πάλι, κάποια Κολοσσαία, να μπορέσει να δώσει μάχες με άλλους χρήστες για να ανέβει επίπεδο και να πάρει χρήματα.

Στην συνέχεια αναφέρονται μερικές λεπτομέρειες για το πώς δημιουργήθηκαν κάποια από τα μέρη του παιχνιδιού και παράλληλα εξηγούνται μερικά βασικά scripts για την λογική με την οποία λειτουργούν, το σκοπό που επιτελούν και τα αποτελέσματα που καταφέρνουν. Τέλος, παρουσιάζονται στιγμιότυπα από το gameplay του παιχνιδιού.

## Abstract

This specific thesis concerns the creation of an augmented reality game on android phones that was made with the assist of Unity game engine and some programming languages such as C# and PHP.

In this thesis there will be a short reference into Unity. Afterwards there will be a short reference to the game so anyone that reads this file can understand what kind of game this is and how can he play it. The goal of this game is that the player has to search Heraklion city with his Gps open in order to find some shops so he can buy weapons to increase his stats and also find some coliseums where he can fight with other players and if he wins he will gain more health and money so he can buy more weapons and get stronger.

Then, details about the creation of some parts of the game are being mentioned and also some of the basic scripts are also explained. Also some screenshots of the game play are being presented.

## Πίνακας Περιεχομένων

Ευχαριστίες .....	2
Σύνοψη.....	3
Abstract.....	4
Πίνακας Περιεχομένων.....	5
Πίνακας Εικόνων.....	7
1 Εισαγωγή.....	9
1.1 Περίληψη .....	9
1.2 Κίνητρο Για Την Διεξαγωγή Της Εργασίας .....	9
1.3 Σκοπός Και Στόχοι Εργασίας .....	10
2 Τεχνολογίες Και Έννοιες Που Συντέλεσαν Στην Πτυχιακή.....	10
2.1 Τι είναι η Unity.....	10
2.2 Τι Είναι Augmented Reality.....	11
2.3 Τι Είναι Η Vuuforia .....	12
2.4 Τι Είναι Το Multiplayer.....	13
2.5 Τι Είναι Οι Βάσεις Δεδομένων .....	15
2.6 Τι Είναι Ένα Framework .....	16
2.7 Τι Είναι Το Adobe Photoshop .....	16
3 Ανάπτυξη Παιχνιδιού Ανά Σκηνή .....	17
3.1 Σκηνή 1:Επιλογή Παίχτη Και Σύνδεση Στο Παιχνίδι.....	17
3.2 Σκηνή 2 : Χάρτης .....	18
3.3 Σκηνή 3: Augmented Reality Shop.....	21
3.3 Σκηνές 5 Και 6: Attack and Defense Weapons.....	23
3.4 Σκηνή 7: Augmented Reality Coliseum .....	25
3.5 Σκηνή 8:Μάχες Στο Κολοσσαίο .....	25
4 Κώδικας Παιχνιδιού Ανά Σκηνή .....	32
4.1 Σκηνή 1.....	32
4.2 Σκηνή 2.....	39
4.3 Σκηνές 3 και 7.....	41
4.4 Σκηνές 5 και 6.....	41

4.5 Σκηνή 8.....	44
5 Επίλογος.....	48
5.1 Συμπεράσματα.....	48
5.2 Δυσκολίες Στην Υλοποίηση.....	48
5.3 Δυνατότητες Βελτίωσης Του Παιχνιδιού.....	49
5.4 ΒΙΒΛΙΟΓΡΑΦΙΕΣ.....	50

## Πίνακας Εικόνων

Figure 1:Περιβάλλον Της Unity.....	11
Figure 2:Παράδειγμα Augmented Reality Εφαρμογής.....	12
Figure 3:Περιβάλλον Vuuforia .....	13
Figure 4:Multiplayer Split Screen.....	14
Figure 5:Multiplayer Online Gaming.....	15
Figure 6:Πίνακας Βάσεων Δεδομένων.....	16
Figure 7: Περιβάλλον Photoshop.....	17
Figure 8:Φόρμα Εγγραφής-Εισαγωγής Χρήστη Στο Παιχνίδι .....	18
Figure 9: Μήνυμα Κλειστού GPS Στην Σκηνή 2.....	19
Figure 10:Χάρτης Και Στατιστικά Του Χρήστη Στην Σκηνή 2.....	19
Figure 11: Μαγαζιά Πάνω Στον Χάρτη.....	20
Figure 12: Κολοσσαία Πάνω Στον Χάρτη .....	20
Figure 13: Τοποθέτηση Αντικειμένων Σε Ακριβείς Συντεταγμένες Του Χάρτη .....	21
Figure 14: Το 3D Μοντέλο Του Μαγαζιού Σε Augmented Reality Απεικόνιση .....	22
Figure 15: Επιλογή Επιθετικών Ή Αμυντικών Όπλων .....	23
Figure 16:Επιθετικά Όπλα Σε Augmented Reality Απεικόνιση.....	24
Figure 17: Αμυντικά Όπλα Σε Augmented Reality Απεικόνιση.....	24
Figure 18: Το 3D Μοντέλο Κολοσσαίο Σε Augmented Reality Απεικόνιση .....	25
Figure 19: Σκηνή Μέσα Στο Κολοσσαίο .....	26
Figure 20: Join Or Fight Buttons.....	27
Figure 21: Εισαγωγή Χρήστη Στην Λίστα Για Μάχες.....	28
Figure 22: Λίστα Με Χρήστες Που Δήλωσαν Ότι Θέλουν Να παλέψουν .....	29
Figure 23: Εικόνα Νίκης Μετά Την Μάχη .....	30
Figure 24: Εικόνα Ήττας Μετά Την Μάχη .....	31
Figure 25: Unity Libraries .....	32
Figure 26: Κώδικας Για Την Φόρμα Login .....	33
Figure 27: Η Εντολή SetActive.....	33
Figure 28: Εικόνα Λανθασμένων Στοιχείων Εγγραφής Ή Εισαγωγής Χρήστη Στο Παιχνίδι .....	34
Figure 29: Η Συνάρτηση DoLogin .....	34
Figure 30: Η Συνάρτηση Login Στον Controller Του Framework Της PHP .....	35
Figure 31: Η Συνάρτηση Login Στο Model Του Framework Της PHP .....	35
Figure 32: Έλεγχος Για Σωστά Δεδομένα Εισαγωγής Στο Παιχνίδι Του Χρήστη.....	36
Figure 33:Εικόνα Σωστών Στοιχείων Εισαγωγής Του Παιχνιδιού Από Τον Χρήστη .....	37
Figure 34: Εγγραφή Χρήστη Στον Controller Του Framework Της PHP .....	38
Figure 35: Εισαγωγή Δεδομένων Χρήστη Στην Βάση Δεδομένων .....	39
Figure 36: Εκτύπωση Όλων Των Στατιστικών Του Παίχτη Από Την Βάση Δεδομένων .....	39

Figure 37: Η Εντολή Split.....	40
Figure 38: Τα Text Objects Στα Οποία Φαίνονται Τα Στατιστικά Του Χρήστη .....	40
Figure 39: Η Εντολή DontDestroyOnLoad.....	40
Figure 40: BuyWeapons Script .....	42
Figure 41: Έλεγχος Αν Μπορεί Να Γίνει Αγορά Όπλου Και Αύξηση Στατιστικών .....	43
Figure 42: Η Συνάρτηση Η Οποία Επιστρέφει Όλα Τα Στοιχεία Ενός Χρήστη .....	43
Figure 43: Η Συνάρτηση Η Οποία Επιστρέφει Όλα Τα Στοιχεία Ενός Όπλου .....	44
Figure 44: Οι Συναρτήσεις Που Αλλάζουν Τα Στατιστικά Του Χρήστη .....	44
Figure 45: Η Συνάρτηση Που Εμφανίζει Όλους Τους Χρήστες Στο Κολοσσαίο .....	45
Figure 46: Σύνδεση Της Unity Με Την Function displayCol .....	45
Figure 47: Η Χρήση Της Εντολής Split Για Την Σωστή Εμφάνιση Όλων Των Χρηστών .....	46
Figure 48: Η Συνάρτηση Fight .....	46
Figure 49: Υπολογισμός Στατιστικών Ανάμεσα Σε Δύο Χρήστες Που Παλεύουν .....	47
Figure 50: Η Εντολή Split Για Την Ενημέρωση Των Στατιστικών Του Χρήστη .....	48



# 1 Εισαγωγή

## 1.1 Περίληψη

Η πτυχιακή εργασία αναφέρεται στην διαδικασία ανάπτυξης ενός ηλεκτρονικού παιχνιδιού, το οποίο για να δημιουργηθεί, βασικό ρόλο έπαιξε η μηχανή παιχνιδιών Unity καθώς και η PHP με το framework της το Code igniter για την σύνδεση του παιχνιδιού online και αντίστοιχα την μετατροπή του από single player σε multiplayer. Θα υπάρξει περιγραφή του παιχνιδιού τόσο μέσα στην Unity όσο και στην δημιουργία του online.

Για την δημιουργία του παιχνιδιού χρειάστηκαν επτά σκηνές. Η πρώτη σκηνή είναι η εγγραφή του χρήστη όπου εκεί μπορεί να διαλέξει ανάμεσα σε δύο παίκτες και η σύνδεση του στο παιχνίδι. Η δεύτερη σκηνή είναι ο χάρτης στον οποίο βρίσκεται ο παίκτης και μέσω του gps κινείται ανάλογα. Στον χάρτη βρίσκει τα shops για να αγοράσει αντικείμενα και τα coliseums για να παλέψει με άλλους χρήστες. Στην τρίτη σκηνή φαίνεται το 3D μοντέλο του shop σε augmented-reality αναπαράσταση και εκεί ο χρήστης μπορεί να επιλέξει αν θέλει όπλα για επίθεση ή για άμυνα. Στην τέταρτη σκηνή και στην πέμπτη ο χρήστης μπορεί να αγοράσει όπλα για να αυξήσει τα στατιστικά του και να γίνει πιο δυνατός. Στην έκτη σκηνή φαίνεται το 3D μοντέλο ενός coliseum σε augmented reality αναπαράσταση. Ακόμα, στην έβδομη σκηνή υπάρχει μια πινακίδα όπου ο χρήστης μπορεί να γραφτεί για να παλέψει και άλλη μία η οποία διαλέγει ποιον θέλει να παλέψει, χωρίς να χρειάζεται να έχει γραφτεί ήδη, από τους χρήστες οι οποίοι έχουν δηλώσει ότι θέλουν να παλέψουν.

Για να πετύχουν όλα αυτά χρησιμοποιήθηκαν πολλά εργαλεία της Unity αλλά επίσης χρησιμοποιήθηκαν διάφορα scripts από C# και PHP. Τέλος τα μοντέλα που χρησιμοποιήθηκαν ήταν έτοιμα είτε από το asset store της Unity είτε από διάφορες ιστοσελίδες.

## 1.2 Κίνητρο Για Την Διεξαγωγή Της Εργασίας

Το κίνητρο για την διεξαγωγή της εργασίας αυτής είναι μια συγκεκριμένη πρακτική άσκηση που έχω σκοπό να κάνω αίτηση στο εξωτερικό η οποία ζητούσε από τους αιτούντες να γνωρίζουν C# και PHP.

Θέλησα να κάνω το συγκεκριμένο παιχνίδι διότι πέρα από την προετοιμασία που μου προσφέρει για την θέση της πρακτικής, περιέχει και μια καινούργια μέθοδο στον τρόπο παιχνιδιού (την augmented reality) και είναι ένα καλό ξεκίνημα για κάποιον φοιτητή στον τομέα του gaming και του android development.

### **1.3 Σκοπός Και Στόχοι Εργασίας**

Σκοπός της συγκεκριμένης εργασίας είναι να δημιουργηθεί ένα ηλεκτρονικό παιχνίδι μέσω της μηχανής παιχνιδιών Unity. Στόχος είναι να αποτελέσει ένα λειτουργικό παιχνίδι και να κινήσει το ενδιαφέροντων χρηστών. Επίσης σαν στόχο έχει κάτι διαφορετικό εκτός από την ψυχαγωγία, έχει και την βόλτα. Ο χρήστης ταυτόχρονα με την βόλτα του μπορεί να παίξει το παιχνίδι διότι δεν έχει επιλογή να παίξει από το σπίτι του.

## **2 Τεχνολογίες Και Έννοιες Που Συντέλεσαν Στην Πτυχιακή**

Στο συγκεκριμένο κεφάλαιο θα γίνει μια γενική αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν ώστε να δημιουργηθεί η πτυχιακή.

### **2.1 Τι είναι η Unity**

Η Unity είναι μια πλατφόρμα δημιουργίας παιχνιδιών που χρησιμοποιείται κυρίως για την δημιουργία ηλεκτρονικών παιχνιδιών για υπολογιστές, κονσόλες καθώς και κινητών τηλεφώνων. Υποστηρίζει 2D και 3D παιχνίδια, drag-and-drop λειτουργικότητα και η κύρια γλώσσα προγραμματισμού που χρησιμοποιεί είναι η C#. Επίσης, άλλες γλώσσες που

χρησιμοποιεί η Unity είναι η JavaScript και η Boo.

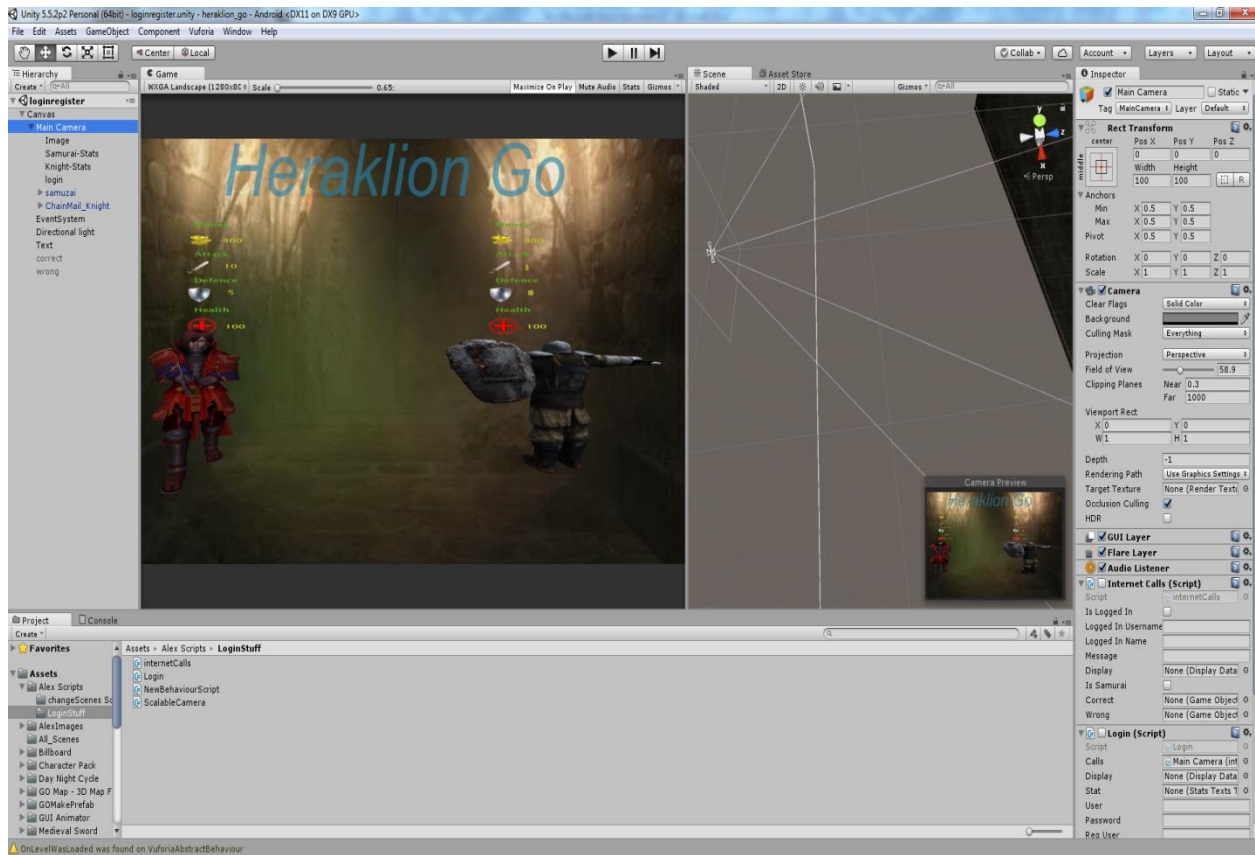


Figure 1: Περιβάλλον Της Unity

## 2.2 Τι Είναι Augmented Reality

Augmented reality είναι η ζωντανή απεικόνιση ενός ή πολλών στοιχείων που δεν είναι πραγματικά, σε πραγματικό περιβάλλον όμως, τα οποία τα επεξεργάζονται και τα χρησιμοποιούν κινητά τηλέφωνα ή ηλεκτρονικοί υπολογιστές και αυτά μπορεί να είναι στοιχεία όπως ο ήχος, βίντεο, κάποιο 3D μοντέλο ή δεδομένα τοποθεσίας (gps).



Figure 2: Παράδειγμα Augmented Reality Εφαρμογής

### 2.3 Τι Είναι Η Vuforia

Η vuforia είναι ένα augmented reality λογισμικό για κινητά τηλέφωνα που επιτρέπει την δημιουργία augmented reality εφαρμογών. Χρησιμοποιεί Computer Vision τεχνολογία για να αναγνωρίζει διάφορες εικόνες ή 3D αντικείμενα όπως για παράδειγμα κουτιά , στην πραγματική ζωή. Αυτή η δυνατότητα επιτρέπει στους προγραμματιστές που χρησιμοποιούν την τεχνολογία της vuforia να τοποθετούν virtual αντικείμενα όπως 3D μοντέλα σε συγκεκριμένες εικόνες που αναγνωρίζει η κάμερα του τηλεφώνου. Το virtual αντικείμενο εντοπίζει την ακριβής θέση της εικόνας σε real-time οπότε ο χρήστης βλέπει το 3D μοντέλο που απεικονίζεται στην ίδια οπτική γωνία όπως είναι και η εικόνα. Εφαρμογές της vuforia μπορούν να δημιουργηθούν από πολλές γλώσσες προγραμματισμού όπως η C++, Java, Objective-C, .Net και C#. Αυτό βοηθάει στην δημιουργία android και IOS εφαρμογών.



Figure 3:Περιβάλλον Vuforia

## 2.4 Τι Είναι Το Multiplayer

Το multiplayer σε ένα ηλεκτρονικό παιχνίδι είναι όταν περισσότεροι από έναν παίχτη μπορούν να παίξουν ταυτόχρονα στην ίδια πίστα ή κόσμο την ίδια στιγμή. Τα παιχνίδια που είναι για ένα άτομο λέγονται single player και αυτό που κάνουν είναι να βάζουν το άτομο να αντιμετωπίσει τον υπολογιστή. Τα multiplayer παιχνίδια επιτρέπουν στους παίχτες να αλληλεπιδρούν μεταξύ τους είτε στο να είναι σύμμαχοι είτε αντίπαλοι.

Multiplayer παιχνίδια μπορεί να υπάρξουν στην ίδια κονσόλα με διαχωρισμό της οθόνης





Figure 4:Multiplayer Split Screen

Ή μπορεί να υπάρξουν στον ίδιο χάρτη ή κόσμο άλλα ο κάθε παίχτης βρίσκεται σε διαφορετικό χώρο με διαφορετικές κονσόλες.



Figure 5: Multiplayer Online Gaming

## 2.5 Τι Είναι Οι Βάσεις Δεδομένων

Μια βάση δεδομένων είναι μια συλλογή από δεδομένα τα οποία έχουν την ιδιότητα να αποθηκεύουν και να ανακτούν άλλα δεδομένα μέσω αναζήτησης κατά απαίτηση. Μέσα στην βάση δεδομένων υπάρχουν οι πίνακες όπου ο κάθε πίνακας έχει ένα ορισμένο πλήθος κελιών και εκεί αποθηκεύονται τα δεδομένα.

Για παράδειγμα σε μια βάση δεδομένων ενός τηλεφωνικού καταλόγου θα υπάρξει ένας πίνακας ο οποίος θα έχει στα κελιά του αποθηκευμένα το όνομα, επώνυμο, διεύθυνση και το τηλέφωνο ενός χρήστη.

Server: 127.0.0.1 Database: test Table: katalogos\_tелефonikos

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total. Query took 0.0000 seconds.)

```
SELECT * FROM `katalogos_tелефonikos`
```

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table

+ Options

onoma	eponimo	dieuthinsi	telefono
Alexandros	Kolydas	Iakxou 3	698025351
Kostantinos	Kolydas	Kritonos 10	698025351

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label:   Let every user access this bookmark

Bookmark this SQL query

Figure 6: Πίνακας Βάσεων Δεδομένων

## 2.6 Τι Είναι Ένα Framework

Ένα framework είναι ο σκελετός για κάτι. Συγκεκριμένα στον τομέα της πληροφορικής ένα framework μιας γλώσσας προγραμματισμού είναι κάποιος κώδικας τον οποίο έχουν γράψει κάποιοι προγραμματιστές και τον έχουν ελεύθερο έτσι ώστε να τον πάρουν οι υπόλοιποι για να διευκολυνθούν. Αυτό είναι πολύ χρήσιμο σε μεγάλα project γιατί μπορούν να γίνουν πολύ πιο γρήγορα. Για αυτό τώρα πια οι περισσότερες μεγάλες εταιρίες τα χρησιμοποιούν συνέχεια.

## 2.7 Τι Είναι Το Adobe Photoshop



Το Adobe Photoshop είναι ένα πρόγραμμα επεξεργασίας εικόνων το οποίο θεωρείται ηγέτης σε αυτού του είδους προγραμμάτων και οτιδήποτε χρειάζεται γραφιστική παρέμβαση στις μεγάλες και στις μικρές εταιρίες, ή ακόμα και στα σπίτια, το Photoshop είναι μέσα.

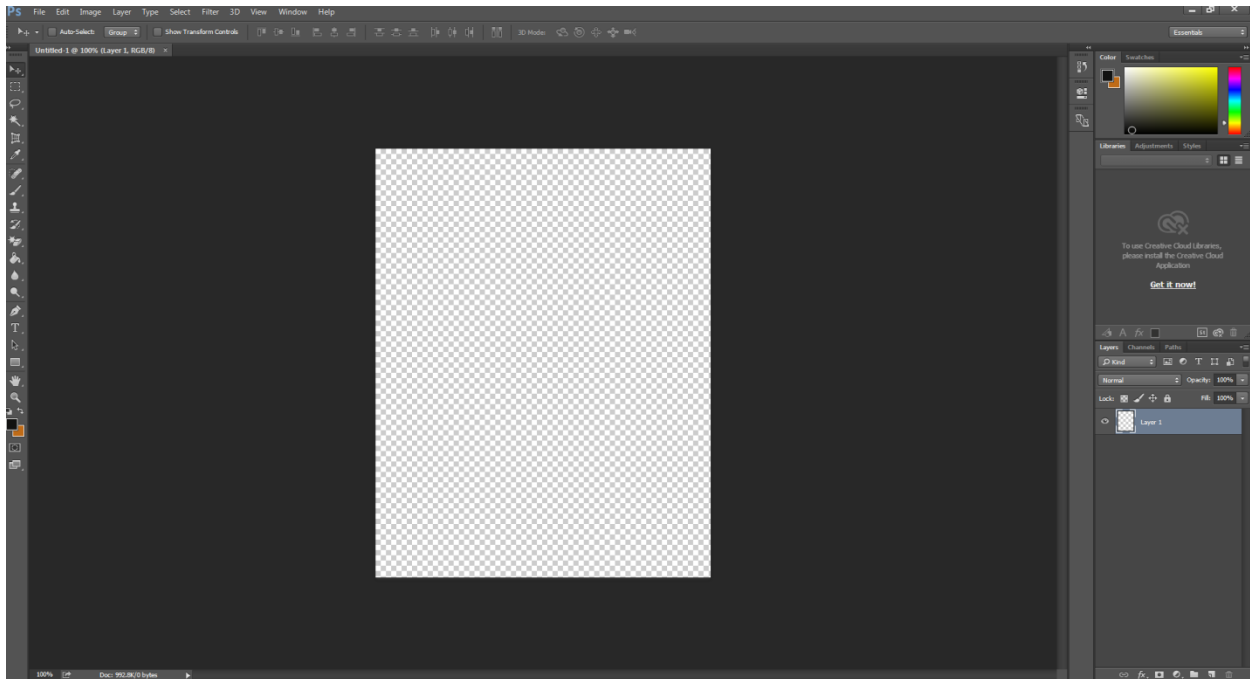


Figure 7: Περιβάλλον Photoshop

### 3 Ανάπτυξη Παιχνιδιού Ανά Σκηνή

Στο συγκεκριμένο κεφάλαιο θα παρουσιαστούν τα βασικότερα κομμάτια της διαδικασίας δημιουργίας του παιχνιδιού σε κάθε σκηνή.

#### 3.1 Σκηνή 1:Επιλογή Παίχτη Και Σύνδεση Στο Παιχνίδι

Στην πρώτη σκηνή ο χρήστης θα πρέπει να κάνει εγγραφή στο παιχνίδι και να επιλέξει ανάμεσα σε δύο παίκτες. Εάν θέλει να έχει περισσότερη επίθεση θα επιλέξει τον Samurai αλλιώς εάν θέλει περισσότερη άμυνα θα επιλέξει τον Knight. Μετά για να μπορέσει να συνεχίσει και να παίξει θα χρειαστεί να ξαναβάλει τα στοιχεία του και να πατήσει το κουμπί Login.



Figure 8:Φόρμα Εγγραφής-Εισαγωγής Χρήστη Στο Παιχνίδι

Για τον Samurai υπάρχει το 3D μοντέλο του στο Asset Store της Unity καθώς και για τον knight υπάρχει το 3D μοντέλο του στην ιστοσελίδα [www.turbosquid.com](http://www.turbosquid.com) και τα animations τους ήταν μέσα στο συνολικό πακέτο τους.

### 3.2 Σκηνή 2 : Χάρτης

Στην δεύτερη σκηνή ο χρήστης αφού έχει ήδη δώσει τα στοιχεία του θα πρέπει να ανοίξει το gps του κινητού του για να μπορέσει η εφαρμογή να τον πάει στο συγκεκριμένο σημείο του χάρτη που βρίσκεται. Ο παίκτης θα κινείται πάνω στον χάρτη όπως κινείται και ο χρήστης με το gps. Εάν δεν ανοίξει το gps θα βρεθεί σε συγκεκριμένη τοποθεσία πολύ μακριά από το Ηράκλειο με ανάλογο μήνυμα.

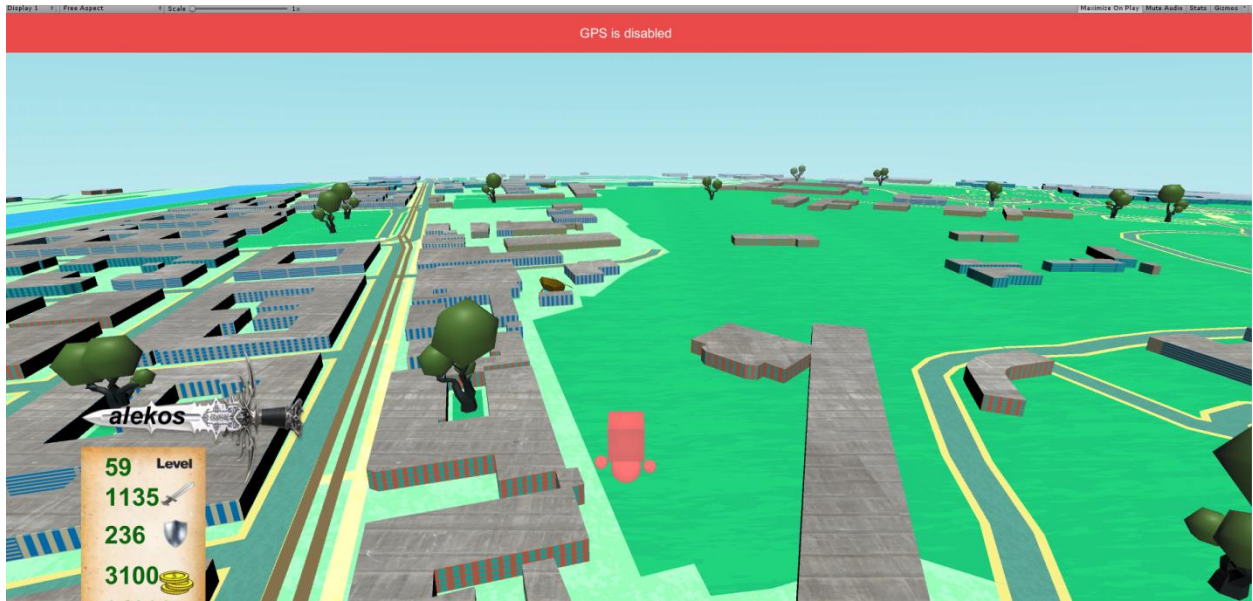


Figure 9: Μήνυμα Κλειστού GPS Στην Σκηνή 2

Στην συνέχεια, κάτω αριστερά υπάρχουν δύο εικόνες στις οποίες υπάρχουν τα στατιστικά του κάθε παίκτη και το όνομα που είχε δηλώσει στην εγγραφή του.

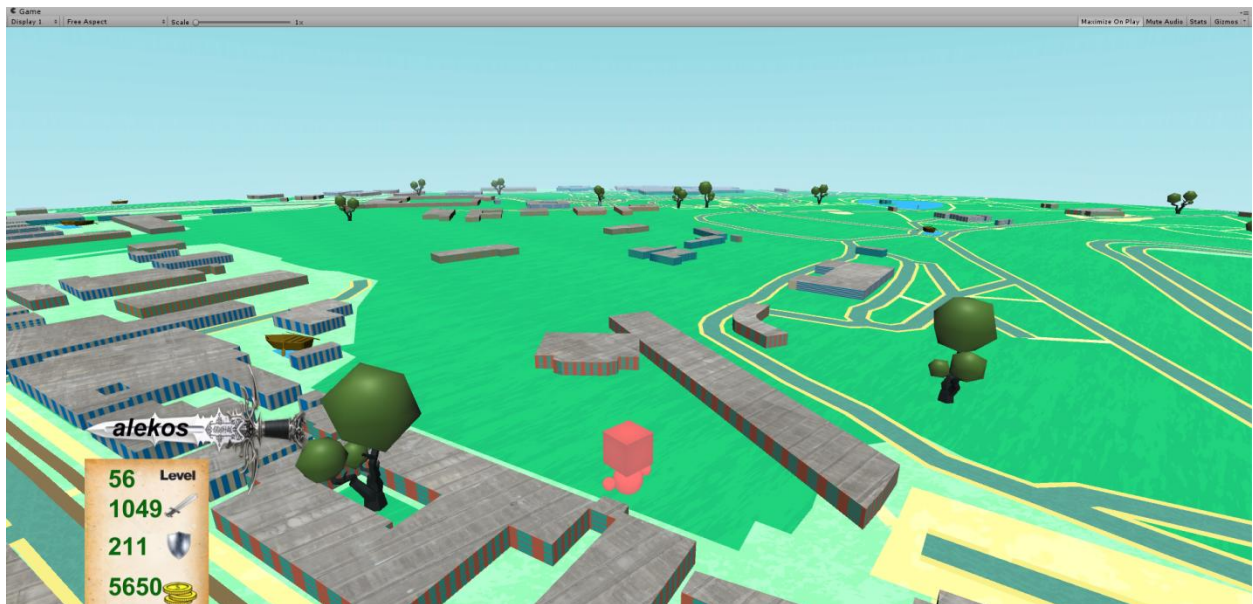


Figure 10:Χάρτης Και Στατιστικά Του Χρήστη Στην Σκηνή 2

Ακόμα, υπάρχουν σε συγκεκριμένες τοποθεσίες πάνω στον χάρτη μαγαζιά όπου εκεί θα μπορεί ο χρήστης να αγοράσει όπλα και να γίνει πιο δυνατός.



Figure 11: Μαγαζιά Πάνω Στον Χάρτη

Επίσης, υπάρχουν σε συγκεκριμένες τοποθεσίες πάλι ,πάνω στον χάρτη, Κολοσσαία όπου εκεί ο χρήστης έχει την δυνατότητα να παλέψει με άλλους χρήστες.

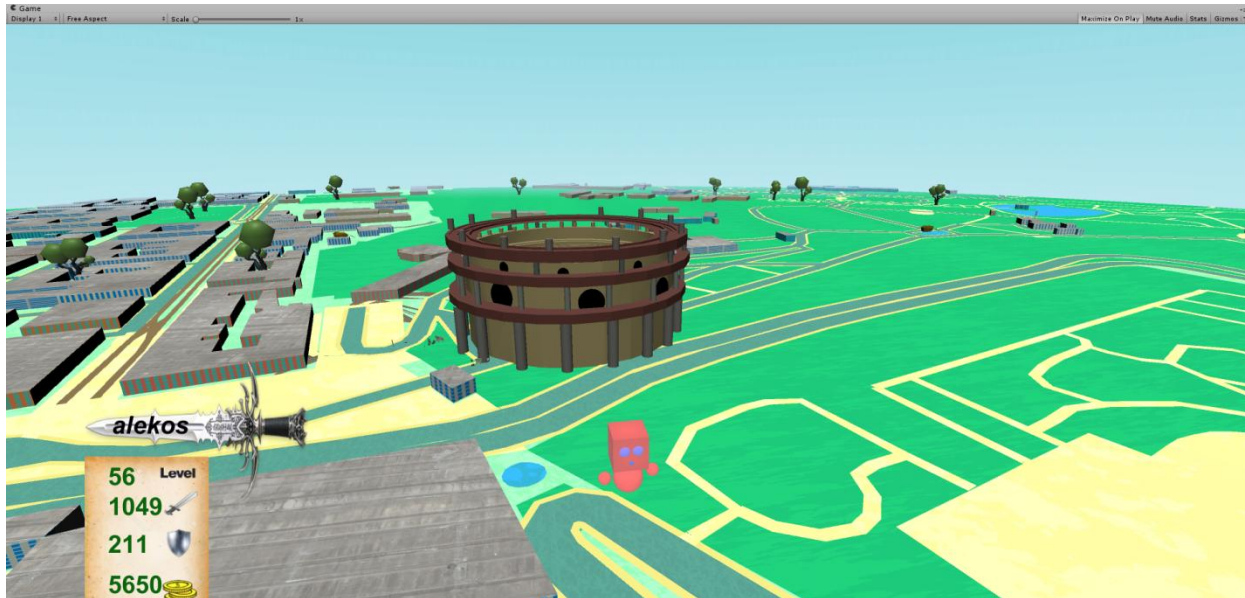


Figure 12: Κολοσσαία Πάνω Στον Χάρτη

Τα 3D μοντέλα των μαγαζιών τα βρήκα από το Asset Store της Unity και τα Κολοσσαία τα βρήκα από την ιστοσελίδα [free3d.com](http://free3d.com) .



Σχεδόν όλη η σκηνή είναι έτοιμη από ένα Asset στο Asset Store της Unity, εκτός από τα μοντέλα των shops και coliseums που είναι από άλλα Assets, αλλά ακόμα και αυτά για να οριστούν σε συγκεκριμένες τοποθεσίες στον χάρτη χρησιμοποιήθηκε το ανάλογο script από το Asset του χάρτη όπου δηλώνει ο προγραμματιστής τις ακριβείς συντεταγμένες.

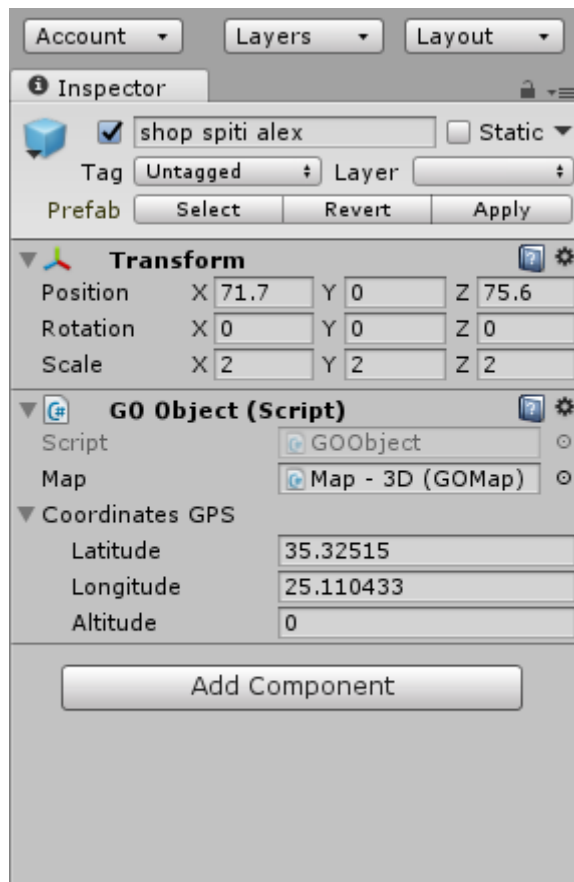


Figure 13: Τοποθέτηση Αντικειμένων Σε Ακριβείς Συντεταγμένες Του Χάρτη

### 3.3 Σκηνή 3: Augmented Reality Shop

Στην σκηνή αυτή, ο χρήστης βλέπει το 3D μοντέλο του shop σε augmented reality. Δηλαδή βλέπει ότι τραβάει η κάμερα του κινητού τηλεφώνου αλλά με το 3D μοντέλο επιπλέον μέσα στο πλάνο. Αυτό επιτυγχάνεται με την τεχνολογία της Vuforia, η οποία ειδικεύεται στο Augmented Reality.



Figure 14: Το 3D Μοντέλο Του Μαγαζιού Σε Augmented Reality Απεικόνιση

Έπειτα, όταν ο χρήστης πατήσει πάνω στο shop, του εμφανίζονται δύο εικόνες, ένα σπαθί και μία ασπίδα για να μπορέσει να επιλέξει αν θέλει να αγοράσει επιθετικά όπλα ή αμυντικά. Τέλος, υπάρχει πάνω δεξιά ένα κουμπί με την φωτογραφία ενός σπαθιού το οποίο πηγαίνει τον χρήστη στην προηγούμενη σκηνή, δηλαδή στον χάρτη.



Figure 15: Επιλογή Επιθετικών Ή Αμυντικών Όπλων

Το 3D μοντέλο του shop είναι το ίδιο με της σκηνής 2 που υπάρχει στο Asset Store της Unity και οι εικόνες των δύο όπλων καθώς και του κουμπιού back υπάρχουν στο Google images.

### 3.3 Σκηνές 5 Και 6: Attack and Defense Weapons

Στις δύο αυτές σκηνές, ο χρήστης έχει την δυνατότητα να αγοράσει σπαθιά για αύξηση της επίθεσης του και ασπίδες για αύξηση της άμυνας. Κάθε όπλο έχει από κάτω την τιμή του και πόσο βελτιώνει τα στατιστικά του χρήστη αγοράζοντας το. Επίσης υπάρχουν και στις δύο σκηνές περιορισμοί στο ποια όπλα μπορεί να αγοράσει ο χρήστης ανάλογα με το level του. Βλέποντας κάτω αριστερά τα στατιστικά του μπορεί να δει ποια όπλα μπορεί να αγοράσει και ποια όχι. Ακόμα και λάθος να κάνει και να προσπαθήσει να αγοράσει κάποιο όπλο που δεν θα μπορεί για οποιοδήποτε λόγο, τότε δεν θα συμβεί τίποτα. Αντιθέτως, εάν αγοράσει κάποιο όπλο θα δει τα στατιστικά του να αλλάζουν και θα ακούσει έναν ήχο ότι πληρώθηκε το όπλο. Φυσικά, υπάρχει το κουμπί back πάνω δεξιά που πηγαίνει τον χρήστη στην προηγούμενη σκηνή.

Όλη η σκηνή είναι augmented reality με την βοήθεια της Vuforia.



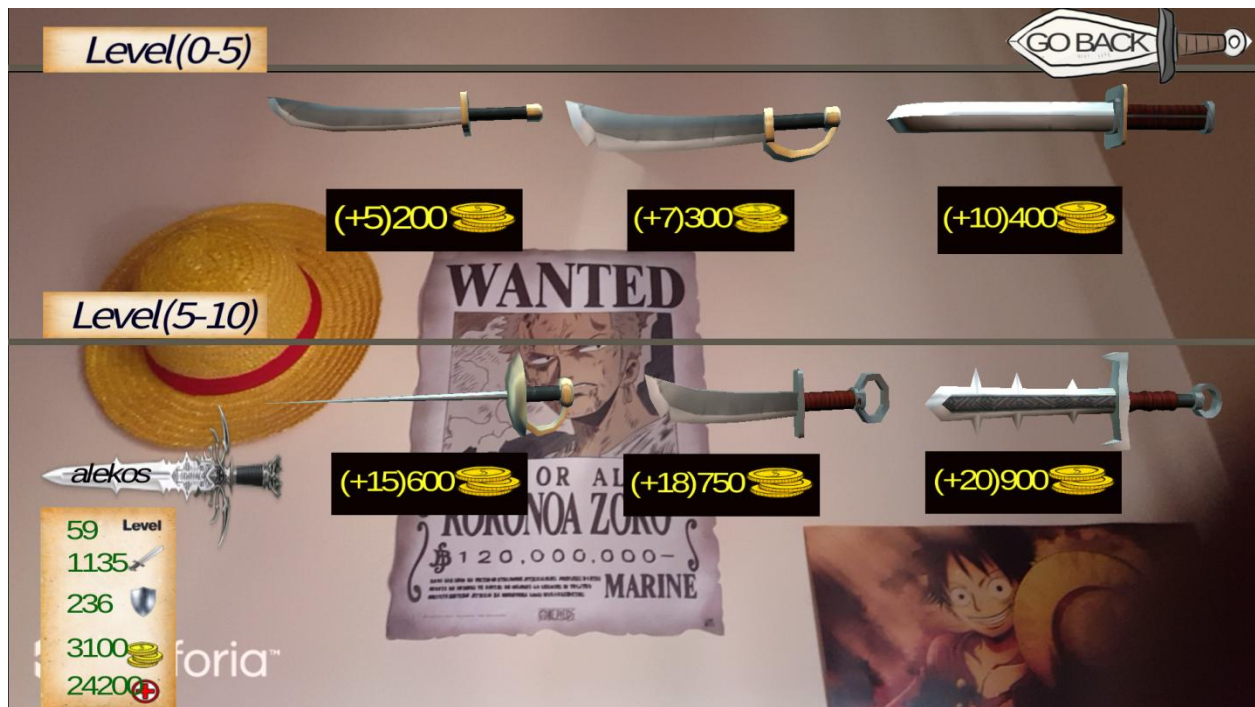


Figure 16: Επιθετικά Όπλα Σε Augmented Reality Απεικόνιση

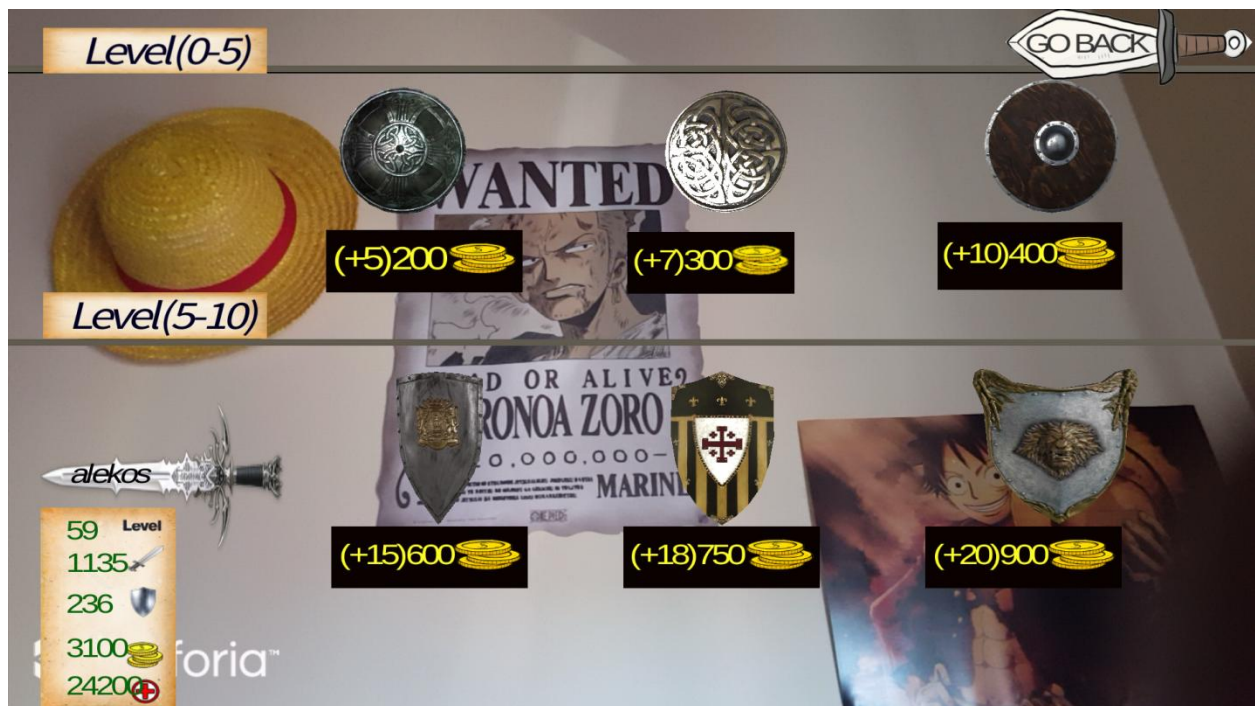


Figure 17: Αμυντικά Όπλα Σε Augmented Reality Απεικόνιση



### 3.4 Σκηνή 7: Augmented Reality Coliseum

Στην σκηνή αυτή ο χρήστης βλέπει το 3D μοντέλο του κολοσσαίου σε augmented reality. Δηλαδή βλέπει, όπως και στο shop, ότι τραβάει η κάμερα του κινητού τηλεφώνου αλλά με το 3D μοντέλο επιπλέον μέσα στο πλάνο. Αυτό επιτυγχάνεται με την τεχνολογία της Vuforia.

Υπάρχει το κουμπί back πάνω δεξιά που πηγαίνει τον χρήστη μια σκηνή πίσω. Πατώντας πάνω στο κολοσσαίο ο χρήστης πηγαίνει στην επόμενη σκηνή που είναι μέσα στο κολοσσαίο για να μπορεί να παλέψει με άλλους χρήστες.

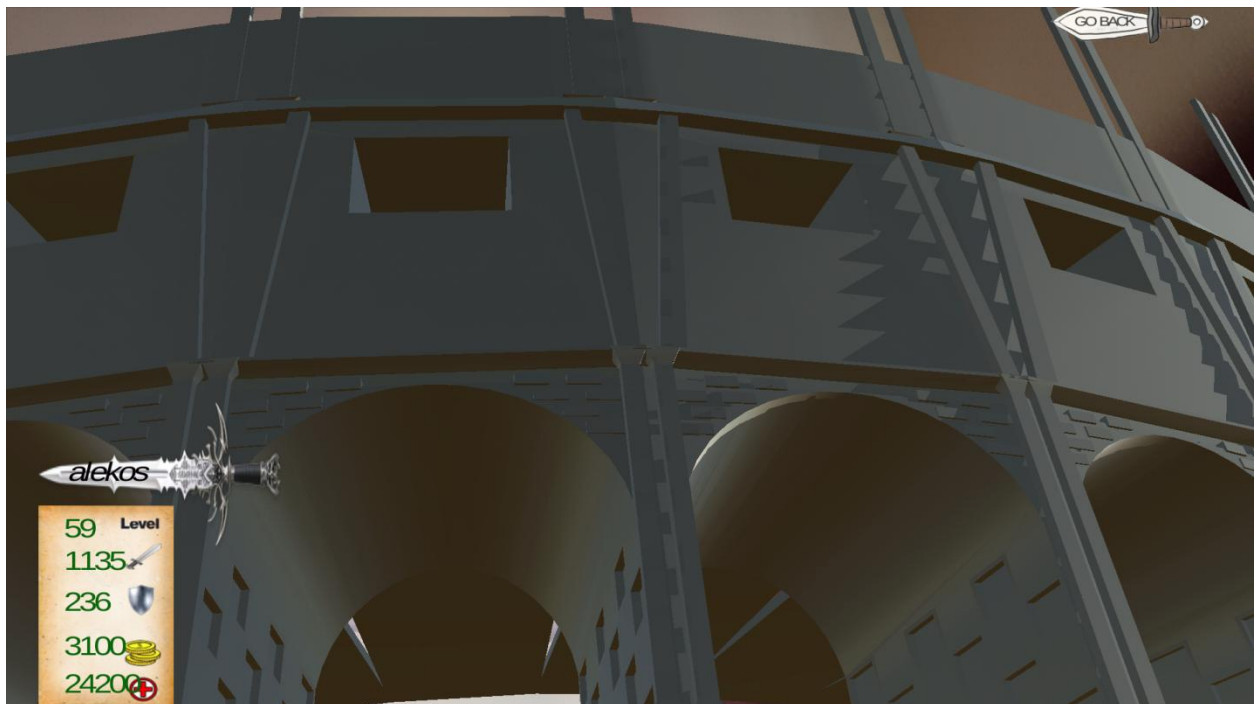


Figure 18: Το 3D Μοντέλο Κολοσσαίο Σε Augmented Reality Απεικόνιση

Το 3D μοντέλο του κολοσσαίου είναι το ίδιο με της σκηνής 2 και υπάρχει στην ιστοσελίδα [free3d.com](http://free3d.com).

### 3.5 Σκηνή 8:Μάχες Στο Κολοσσαίο

Στην σκηνή αυτή, ο χρήστης βρίσκεται μέσα στο κολοσσαίο το οποίο το βλέπει σε augmented reality. Υπάρχει το κουμπί πάνω δεξιά για να πάει τον χρήστη στην προηγούμενη σκηνή.



Figure 19: Σκηνή Μέσα Στο Κολοσσαίο

Υπάρχει και μία πινακίδα μέσα στην arena η οποία, όταν την πατήσει ο χρήστης εμφανίζονται δύο εικόνες σπαθιών και ο χρήστης μπορεί να επιλέξει είτε να κάνει απλά εγγραφή στην λίστα του κολοσσαίου έτσι ώστε να τον δουν άλλοι χρήστες και να κάνουν μάχη μαζί του, είτε να δει την λίστα με όσους είναι εγγεγραμμένοι για μάχη και να παλέψει ο ίδιος με αυτούς.

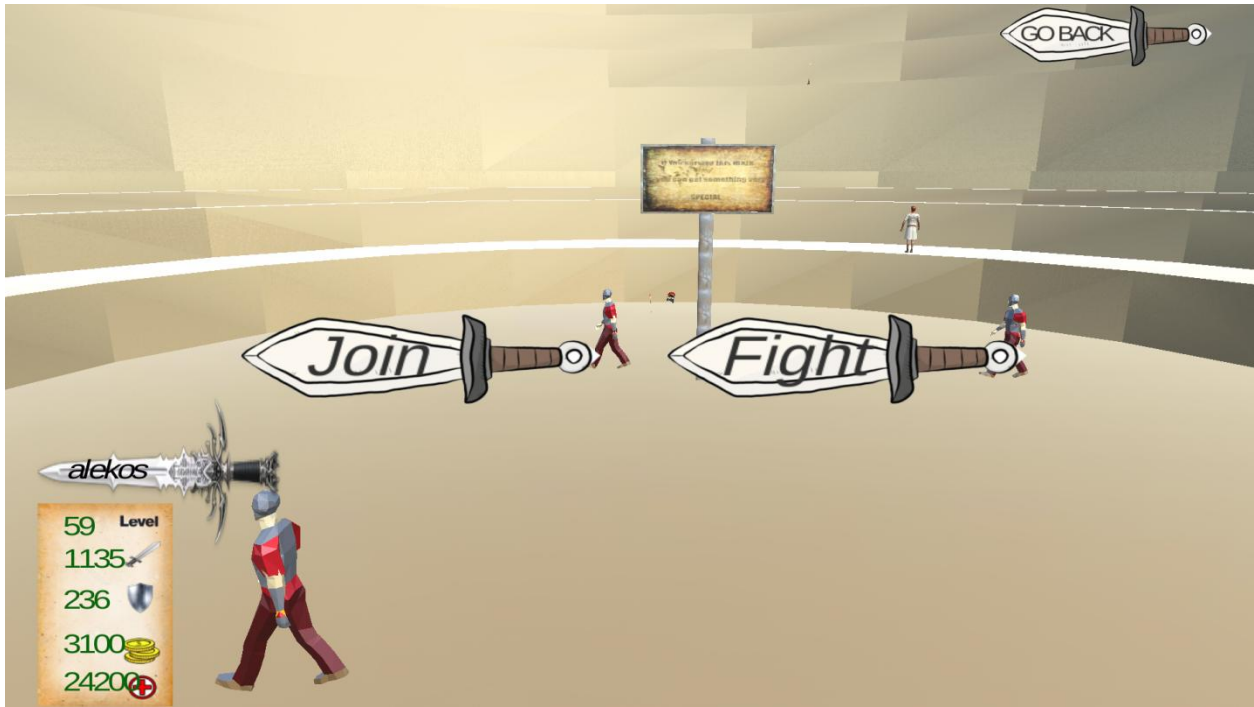


Figure 20: Join Or Fight Buttons

Πατώντας Join ο χρήστης λοιπόν δηλώνει ότι θέλει οι υπόλοιποι χρήστες να μπορούν να παλέψουν μαζί τους. Μόλις πατήσει λοιπόν το κουμπί τότε θα δει το ανάλογο μήνυμα



Figure 21: Εισαγωγή Χρήστη Στην Λίστα Για Μάχες

Αντιθέτως, εάν πατήσει το κουμπί Fight θα δει την λίστα με τους εγγεγραμμένους χρήστες.



Figure 22: Λίστα Με Χρήστες Που Δήλωσαν Ότι Θέλουν Να παλέψουν

Εδώ μπορεί να δει πέρα από το όνομα του κάθε χρήστη και τα στατιστικά του για να ξέρει αν μπορεί να τον νικήσει ή όχι. Πατώντας πάνω σε κάποιον αντίπαλο εάν τον νικήσει τότε θα δει

το ακόλουθο μήνυμα στην οθόνη του.



Figure 23: Εικόνα Νίκης Μετά Την Μάχη

Όταν λοιπόν νικήσει τότε αυτόματα θα κερδίσει χρήματα, ζωή και level και επίσης ο αντίπαλος που έχασε διαγράφεται αυτόματα από την λίστα.

Εάν όμως παλέψει με κάποιον και χάσει θα δει το ακόλουθο μήνυμα στην οθόνη του.



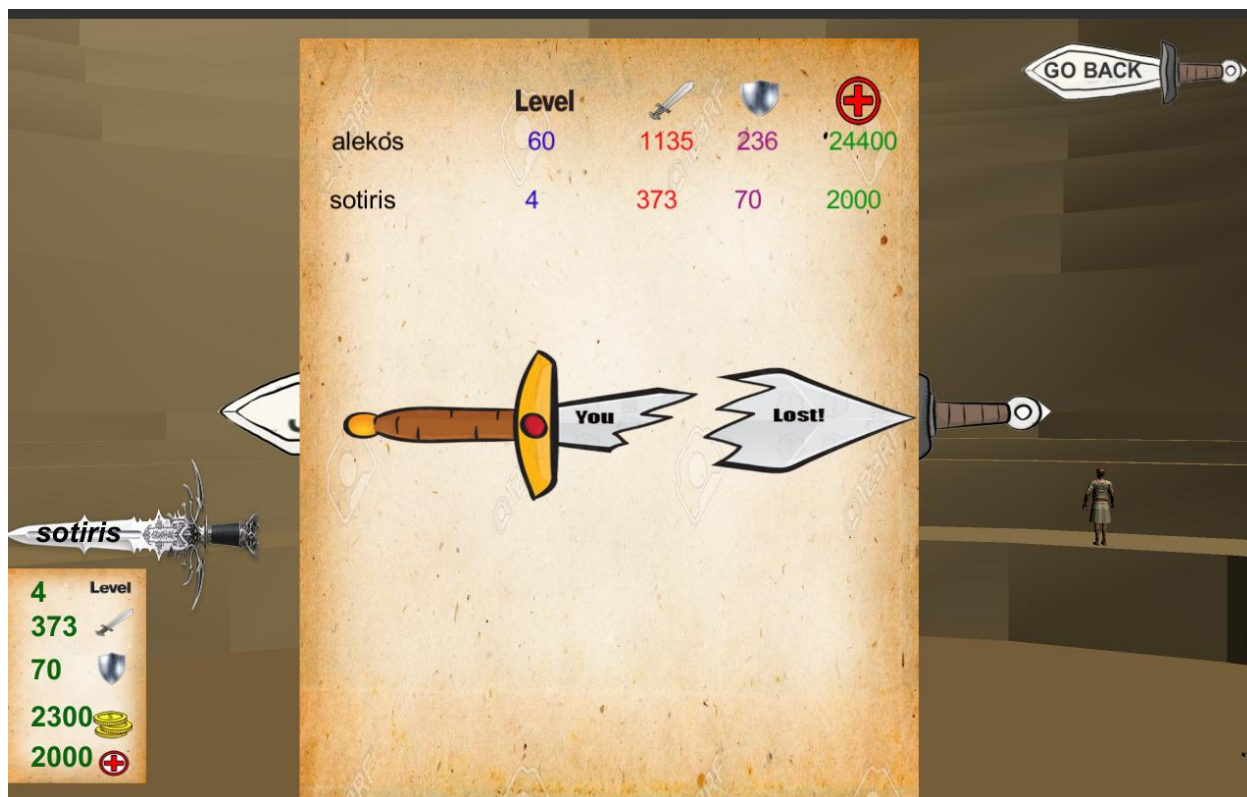


Figure 24: Εικόνα Ήττας Μετά Την Μάχη

Δεν θα γίνει καμία ενέργεια εις βάρος του χρήστη σε αυτήν την περίπτωση.

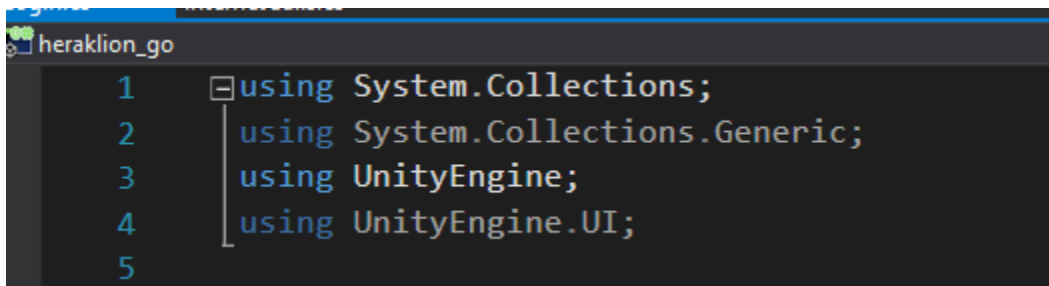
## 4 Κώδικας Παιχνιδιού Ανά Σκηνή

Στο συγκεκριμένο κεφάλαιο θα αναλυθούν βασικά κομμάτια κώδικα που ήταν πολύ σημαντικά για την δημιουργία του παιχνιδιού.

### 4.1 Σκηνή 1

Το κύριο κομμάτι στην πρώτη σκηνή είναι η εγγραφή και η εισαγωγή του χρήστη στο παιχνίδι. Χρειάστηκαν δύο scripts στην Unity για να το πετύχουν αυτό και φυσικά χρειάστηκε ένα php script για να αποθηκευτούν τα δεδομένα στην βάση.

Το πρώτο script είναι ο σχεδιασμός της φόρμας για να μπορεί να βάλει τα στοιχεία του ο χρήστης. Αρχικά σε όλα τα scripts πρέπει να εισαχθούν βιβλιοθήκες στην αρχή.



```
heraklion_go
1  using System.Collections;
2      using System.Collections.Generic;
3      using UnityEngine;
4      using UnityEngine.UI;
5
```

Figure 25: Unity Libraries

Έπειτα, το συγκεκριμένο script στηρίζεται στην function OnGUI η οποία έχει να κάνει με το γραφικό περιβάλλον στην Unity οπότε στην περίπτωση μας με την φόρμα.



```

void OnGUI()
{
    GUIStyle style = new GUIStyle();
    style.richText = true;
    if (message != "")
        GUILayout.Box(message);

    GUILayout.BeginArea(new Rect((Screen.width / 2) - 350 , (Screen.height / 2) - 200,500, 256), GUIContent.none, "box");

    GUILayout.BeginHorizontal();
    GUILayout.Label("<size=30><color=white>Username</color> </size>",style);
    user = GUI.TextArea(new Rect(200, 10, 300, 35), user, 10);

    //user = GUILayout.TextField(user); //ser = GUILayout.TextField(user,25);
    //user = GUILayout.TextField(new Rect(65, 10, 100, 25), user);
    GUILayout.EndHorizontal();

    GUILayout.BeginHorizontal();
    GUILayout.Label("<size=30><color=white>Password</color> </size>",style);
    password= GUI.PasswordField(new Rect(200, 45, 300, 35), password, "*" [0],20);
    //password = GUILayout.PasswordField(password, '*');
    GUILayout.EndHorizontal();

    if (GUILayout.Button("Login",GUILayout.Height(30)))
    {
        if (user == "" || password == "")
        {
            StartCoroutine(wait());
        }

        else
        {
            calls.DoLogin(user, password);
            Debug.Log("patises to login");
        }
    }
}

```

Figure 26: Κώδικας Για Την Φόρμα Login

Εδώ λοιπόν αυτό που βλέπουμε περιληπτικά είναι η δημιουργία της φόρμας μέσω της εντολής *GUILayout.Label* και *Gui.TextArea*. Μετά για να ελεγχθεί εάν έχουν εισαχθεί και τα δύο στοιχεία υπάρχει μια συνθήκη η οποία ελέγχει αν δεν έχουν μπει τα στοιχεία και πηγαίνει μέσω της εντολής *StartCoroutine* στην function *wait* η οποία εμφανίζει μια εικόνα που στην περίπτωση του λάθους είναι ένα κόκκινο X με την εντολή *onamaeikonas.SetActive(true)* και κάνει refresh την σκηνή μέσω της εντολής *UnityEngine.SceneManagement.SceneManager.LoadScene("Loginregister");*.

```

public IEnumerator wait() // vazó autin tin sinartisi gt mesa stin OnGui den mporo na kano return pragmata
{
    wrong.SetActive(true);
    yield return new WaitForSeconds(0.5f);
    UnityEngine.SceneManagement.SceneManager.LoadScene("loginregister");
}

```

Figure 27: Η Εντολή SetActive



Figure 28: Εικόνα Λανθασμένων Στοιχείων Εγγραφής Ή Εισαγωγής Χρήστη Στο Παιχνίδι

Εάν όμως τα στοιχεία είναι σωστά που δίνει ο χρήστης θα καλεστεί η συνάρτηση DoLogin η οποία θα πάρει τα στοιχεία user και password που θα δώσει ο χρήστης.

```
public void DoLogin(string user, string password)
{
    WWWForm www = new WWWForm();
    www.AddField("user", user);
    www.AddField("password", password);

    WWW w = new WWW("http://ptixiaki.sotirisoikonomou.com/CodeIgniterUnity/index.php/Users/login", www);
    StartCoroutine(Login(w, user));
    LoggedInUsername = user;
    LoggedInName = name;
    Debug.Log("Logged in: " + user);
}
```

Figure 29: Η Συνάρτηση DoLogin

Μέσω της εντολής WWW δίνει τα στοιχεία που υπάρχουν στα πεδία www.AddField στην συγκεκριμένη σελίδα η οποία είναι η login function της PHP.

```

function login(){
    $data['error'] = 0; // simenei oti den exei errors

    if($_POST){
        $user=$this->input->post('user',true);//pairnei to username p edose o xristis(einai idio me to $_POST)
        $password=$this->input->post('password',true);//pairnei to password p edose o xristis
        //$type=$this->input->post('charact',true);
        $user1=$this->User->login($user,$password);//,$stype);
        if(!$user1){
            $data['error']=1;
            echo"lathos";

        }else{

            $this->session->set_userdata('id',$user1['id']);
            $this->session->set_userdata('user',$user1['user']);
            $this->session->set_userdata('name',$user1['name']);
            $this->session->set_userdata('money',$user1['money']);
            $this->session->set_userdata('attack',$user1['attack']);
            $this->session->set_userdata('defence',$user1['defence']);
            $this->session->set_userdata('level',$user1['level']);
            $this->session->set_userdata('xp',$user1['xp']);
            //redirect(base_url().'index.php/Users');
            $data['id'] = $user1['id'];
            $this->load->view('Post_index',$data);

        }

    }
    else {
        $this->load->view('Login',$data);
    }
}

```

Figure 30: Η Συνάρτηση Login Στον Controller Του Framework Της PHP

Στην function login έρχονται τα στοιχεία από την Unity(username , password) και ελέγχονται αν είναι σωστά μέσω της γραμμής `$user1=$this->User->login($user,$password);//,$stype);` η οποία πηγαίνει στην function της επόμενης εικόνας όπου γίνεται ο έλεγχος.

```

function login($user,$password){
    $where=array(
        'user'=>$user,
        'password'=>$password
    );
    $this->db->select()->from('unityusers')->where($where);
    $query=$this->db->get();
    return $query->first_row('array');
}

```

Figure 31: Η Συνάρτηση Login Στο Model Του Framework Της PHP

Όταν τα στοιχεία επιστρέψουν στην Unity τότε καλείτε η συνάρτηση login μέσω της εντολής StartCoroutine.

```
public IEnumerator Login(WWW w, string user)
{
    yield return w;
    //Debug.Log(w.error);
    if (w.error == null && w.text != "lathos")
    {
        Debug.Log("you are logged in");
        correct.SetActive(true);
        IsLoggedIn = true;
        LoggedInUsername = user;
        UnityEngine.SceneManagement.SceneManager.LoadScene("New! 3D - Demo");
    }
    else
    {
        Debug.Log("you are NOT logged in"+w.error);

        wrong.SetActive(true);
        yield return new WaitForSeconds(1);
        wrong.SetActive(false);
    }
}
```

Figure 32: Έλεγχος Για Σωστά Δεδομένα Εισαγωγής Στο Παιχνίδι Του Χρήστη

Ανάλογα εάν επιστρέψει false η PHP τότε θα δείξει το κόκκινο X πάλι και θα κάνει refresh την σκηνή. Αλλιώς θα δείξει ένα πράσινο τικ και θα πάει στην επόμενη σκηνή.



Figure 33:Εικόνα Σωστών Στοιχείων Εισαγωγής Του Παιχνιδιού Από Τον Χρήστη

Υπάρχει ακόμα και η εγγραφή του χρήστη και η λογική του κώδικα είναι σχεδόν η ίδια με την μόνη διαφορά ότι δεν ελέγχεται εάν τα στοιχεία που έδωσε είναι λάθος ή σωστά αλλά να περάσει τα στοιχεία στην βάση δεδομένων μέσω της PHP.

```

function registerSam(){
    if($_POST){
        $data=array(
            'user'=>$_POST['user'],
            'name'=>$_POST['name'],
            'password'=>$_POST['password'],
            'charact'=>"Samurai",
            'money'=>400,
            'attack'=>10,
            'defence'=>5,
            'level'=>0,
            'health'=>400,
            'xp'=>0
        );
        $userid=$this->User->create_user($data);
    }
}

function registerKnight(){
    if($_POST){
        $data=array(
            'user'=>$_POST['user'],
            'name'=>$_POST['name'],
            'password'=>$_POST['password'],
            'charact'=>"Knight",
            'money'=>400,
            'attack'=>5,
            'defence'=>10,
            'level'=>0,
            'health'=>400,
            'xp'=>0
        );
        $userid=$this->User->create_user($data);
    }
}

```

Figure 34: Εγγραφή Χρήστη Στον Controller Του Framework Της PHP

```
function create_user($data){
    $this->db->insert('unityusers',$data);
}
```

Figure 35: Εισαγωγή Δεδομένων Χρήστη Στην Βάση Δεδομένων

## 4.2 Σκηνή 2

Η δεύτερη σκηνή είναι ο χάρτης. Σχεδόν όλη η σκηνή είναι έτοιμη από ένα Asset στην Unity. Τα κομμάτια που προστέθηκαν είναι τα στατιστικά του παίκτη τα οποία τα παίρνει η Unity από την βάση δεδομένων.

Χρησιμοποιώντας την WWWεντολή της Unity στέλνουμε ποιος παίκτης είναι αυτός που θέλουμε να πάρουμε τα στατιστικά του στην function guser της PHP

```
function guser(){
    if($_POST)
    {
        $user=$this->input->post('user',true);
        $data['user']=$this->User->get_user($user);
        echo $data['user']['name'].','.$data['user']['level'].','.$data['user']['attack'].','.$data['user']['defence'].','.$data['user']['money'].','.$data['user']['health'].','.$data['user']['user']; //painei apo tin unity to username apo tin global metavliti pou exo orisei(LoggedInUsername) apo to arxio (displaydatafromphp) me tin post request methodo kai elegxei an iparxei stin database kai meta ektupono ola to row apo to sugkekrimeno username
    }
}
```

Figure 36: Εκτύπωση Όλων Των Στατιστικών Του Παίκτη Από Την Βάση Δεδομένων

Έπειτα με την εντολή split χωρίζεται όλο το echo της php σε κομμάτια κάθε φορά που η εντολή βρίσκει το ",".Οπότε ξέροντας ότι για παράδειγμα το πρώτο στοιχείο πριν βρει το "," είναι το name του χρήστη το βάζει σε ένα text και το κάνει display στην οθόνη.

```

IEnumerator WaitForRequest(WWW www)
{
    yield return www;

    // check for errors
    if (www.error == null)
    {

        //Assign the data that was fetched to the variable answer
        answer = www.text.ToString();
        stats = answer.Split(","); // PAIRNEI APO TON CONTROLLER TIS PHP STIN FUNCTION guser TO $data kai opou vlepei ',' to vazei se ena keli tou pinaka
        warriorname = stats[0]; //to name
        dbUser = stats[6]; // to username
        dbMoney = stats[4]; // ta lefta
    }
    else {
        Debug.Log("WWW Error: " + www.error);
    }
}

```

Figure 37: Η Εντολή Split

```

void OnGUI()
{
    namee.text = stats[0]; //emfanizei apo tin vasi mou ta stoixeia
    lvl.text = stats[1];
    atc.text = stats[2];
    def.text = stats[3];
    cash.text = stats[4];
    hp.text = stats[5];
}

```

Figure 38: Τα Text Objects Στα Οποία Φαίνονται Τα Στατιστικά Του Χρήστη

Τα αντικείμενα σε κάθε σκηνή διαγράφονται μαζί με τα δεδομένα που κρατάνε όταν αλλαχθεί η σκηνή. Οπότε για να μείνουν τα στατιστικά του παίκτη καθ' όλη την διάρκεια του παιχνιδιού χρησιμοποιείται στην function Start η οποία ενεργοποιείται όταν ξεκινάει η σκηνή, η εντολή

```
DontDestroyOnLoad(this.gameObject);
```

Figure 39: Η Εντολή DontDestroyOnLoad



### 4.3 Σκηνές 3 και 7

Στις συγκεκριμένες σκηνές ο κώδικας ήταν ήδη τροποποιημένος από τις τεχνολογίες που χρησιμοποιούνται. Δηλαδή, το μόνο που έχουν αυτές οι σκηνές είναι τα 3D μοντέλα τα οποία είναι σε augmentedreality λόγω της Vuforia.

Η μόνη παρέμβαση που υπάρχει είναι τα κουμπιά στην κάθε σκηνή τα οποία πατώντας τα, μέσω της εντολής `UnityEngine.SceneManagement.SceneManager.LoadScene("");` ανακατευθύνουν τον παίκτη στις ανάλογες σκηνές.

### 4.4 Σκηνές 5 και 6

Στις δύο αυτές σκηνές σε κάθε όπλο που υπάρχει, είτε είναι σπαθί, είτε ασπίδα, υπάρχει ένα script που είναι πάνω σε κάθε `gameobject`.

```

public void sword1()
{
    WWWForm w_buywp = new WWWForm();
    w_buywp.AddField("user", internetCalls._Instance.LoggedInUsername);
    w_buywp.AddField("weapon", weaponName);
    WWW w_buy = new WWW(url, w_buywp);
    StartCoroutine(BuyWeapon(w_buy));
    buySound.SetActive(false);
}

public IEnumerator BuyWeapon(WWW w_buy)
{
    yield return w_buy;
    if (w_buy.error == null && w_buy.text != "You can't buy it!")
    {
        buySound.SetActive(true);
        wp = w_buy.text.ToString();
        weapons = wp.Split(",")[0];
        displayDataFromPhp._Instance.stats[4] = weapons[4]; //gia to cash
        displayDataFromPhp._Instance.stats[2] = weapons[2]; //gia to attack
        displayDataFromPhp._Instance.stats[3] = weapons[3]; //gia to defence
    }
    else
    {
        message += "Sorry you can't buy";
    }
}

```

Figure 40: BuyWeapons Script

Φυσικά χρησιμοποιείται η εντολή WWW αφού πρέπει να σταλούν το username του χρήστη και το όνομα του όπλου online έτσι ώστε να γίνει έλεγχος εάν μπορεί να αγοράσει ο χρήστης το όπλο ή όχι.

```

function buyWeapon(){
    $sep='';

    if($_POST)
    {
        $wpuser=$this->input->post('user',true);
        $wp=$this->input->post('weapon',true);
        $data['user']=$this->User->get_user($wpuser);
        $data['weapon']=$this->User->get_weapon($wp);
        if($data['user']['money']==$data['weapon']['cost'] || $data['user']['money'] > $data['weapon']['cost']
            && $data['user']['level']==$data['weapon']['level'] || $data['user']['level']>$data['weapon']['level'])
        {
            $userCash=$data['user']['money']-$data['weapon']['cost'];
            $userAttack=$data['user']['attack']+$data['weapon']['attack'];
            $userDefence=$data['user']['defence']+$data['weapon']['defence'];

            $updDat = array(
                'money' =>$userCash,
                'attack'=>$userAttack,
                'defence'=>$userDefence
            );
            $updStat=$this->User->updatePlayerData($wpuser,$updDat);

            echo $data['user']['name'].' '.$data['user']['level'].' '.$userAttack.' '.$userDefence.' '.$userCash.' '.$data['user']['health'].' '.$data['user']['
            user'].' '.$sep;

            echo $data['weapon']['name'].' '.$data['weapon']['level'].' '.$data['weapon']['attack'].' '.$data['weapon']['defence'].' '.$data['weapon']['cost'];
        }else{
            echo"You can't buy it!";
        }
    }
}

```

Figure 41: Έλεγχος Αν Μπορεί Να Γίνει Αγορά Όπλου Και Αύξηση Στατιστικών

Η functionBuyWeapon της PHP παίρνει λοιπόν τα δύο αυτά δεδομένα και τα στέλνει στο model στην αντίστοιχη function, δηλαδή το usernameστοget\_user

```

function get_user($user){
    $this->db->select()->from('unityusers')->where(array('user'=>$user));
    $query=$this->db->get();
    return $query->first_row('array');
}

```

Figure 42: Η Συνάρτηση Η Οποία Επιστρέφει Όλα Τα Στοιχεία Ενός Χρήστη

για να πάρει όλα τα στοιχεία του χρήστη με το συγκεκριμένο username και το όνομα του όπλου στο get\_weapon

```
function get_weapon($weapon){
    $this->db->select()->from('weapons')->where(array('name'=>$weapon));
    $query=$this->db->get();
    return $query->first_row('array');
}
```

Figure 43: Η Συνάρτηση Η Οποία Επιστρέφει Όλα Τα Στοιχεία Ενός Όπλου

για να πάρει όλα τα στοιχεία του όπλου.

Έπειτα, ελέγχει εάν ο χρήστης έχει αρκετά χρήματα ώστε να αγοράσει το όπλο και αν τηρεί επίσης την προϋπόθεση με το level. Εάν δεν την τηρεί τότε δεν γίνεται τίποτα, αλλιώς μέσω της functionUpdatePlayerData και functionUpdatePlayerLevel ,αλλάζει ανάλογα τα χρήματα, το level και την επίθεση ή την άμυνα.

```
function updatePlayerData($fuser,$updData){
    $this->db->where('user',$fuser);
    $this->db->update('unityusers',$updData);
}

function updatePlayerLevel($fuser,$updlvl){
    $this->db->where('user',$fuser);
    $this->db->update('unityusers',$updlvl);
}
```

Figure 44: Οι Συναρτήσεις Που Αλλάζουν Τα Στατιστικά Του Χρήστη

Τέλος, μέσω της εντολής Split αλλάζει τα στοιχεία στην Unity με αυτά που πήρε από την PHP.

## 4.5 Σκηνή 8

Στην σκηνή αυτή αξίζει να σημειωθεί το script που εμφανίζει όλους τους χρήστες οι οποίοι είναι για μάχη στο Κολοσσαίο.

Όπως είναι αναμενόμενο αφού χρειάζονται τα στοιχεία από την βάση θα χρησιμοποιηθεί η εντολή WWW για να δοθεί στο script της PHP ποιος χρήστης βλέπει εκείνη την στιγμή τους υπόλοιπους και θα γυρίσει πίσω όλους τους χρήστες που είναι εγγεγραμμένοι στο Κολοσσαίο.

```
function displayCol(){
    if($_POST){
        $duser=$this->input->post('user',true);
        $data['users']=$this->User->get_users();
        $this->load->view('Post_index',$data);
    }
}
```

Figure 45: Η Συνάρτηση Που Εμφανίζει Όλους Τους Χρήστες Στο Κολοσσαίο

Επειδή χρειάζεται να αρχίζει αυτό με το που ανοίγει η σκηνή, για αυτό είναι στην functionStart

```
void Start () {
    WWWForm www_dis = new WWWForm();
    www_dis.AddField("user", internetCalls.Instance.LoggedInUsername);
    WWW w_dis = new WWW("http://ptixiaki.sotirisoikonomou.com/CodeIgniterUnity/index.php/Users/displayCol", www_dis);
    StartCoroutine(DisplayCol(w_dis));
}
```

Figure 46: Σύνδεση Της Unity Με Την Function displayCol

Με την βοήθεια της εντολής Split εκτυπώνονται τα αντίστοιχα στοιχεία για κάθε χρήστη στην οθόνη (Name, Level, Attack, Defense, Health).

```

public IEnumerator DisplayCol(WWW w_dis)
{
    yield return w_dis;
    if (w_dis.error == null)
    {
        player = w_dis.text.ToString();
        cstats= player.Split(",")[0];
        display();
    }
    else
    {
        message += "No users";
    }
}

```

Figure 47: Η Χρήση Της Εντολής Split Για Την Σωστή Εμφάνιση Όλων Των Χρηστών

Πατώντας πάνω σε κάποιον παίκτη έτσι ώστε να μπορεί ο χρήστης να παλέψει μαζί του ενεργοποιείται η functionfight.

```

public void fight1()
{
    if (displayDataFromPhp._Instance.warriorname != cstats[0])
    {
        WWWForm www_fight = new WWWForm();
        www_fight.AddField("user", internetCalls._Instance.LoggedInUsername);
        www_fight.AddField("opponent", cstats[5]);
        WWW w_dis = new WWW("http://ptixiaki.sotirisoikonomou.com/CodeIgniterUnity/index.php/Users/fightCol", www_fight);
        StartCoroutine(FightCol(w_dis));
    }
    else
    {
        Debug.Log("ESU EISAI ");
    }
}

```

Figure 48: Η Συνάρτηση Fight

Η οποία στέλνει στην PHP το όνομα του χρήστη και το όνομα του παίκτη που θέλει ο χρήστης να κάνει μάχη μαζί του για να ελέγξει ποιος θα νικήσει.



```

function fightCol(){
    if($_POST){
        $fuser=$this->input->post('user',true);
        $opponentUser=$this->input->post('opponent',true);
        $data['user']=$this->User->get_user($fuser);
        $data['opUser']=$this->User->get_user($opponentUser);
        //echo $fuser.', '.$opponentUser;
        $userStat=$data['user']['attack']+$data['user']['defence']+$data['user']['health'];
        $opStat=$data['opUser']['attack']+$data['opUser']['defence']+$data['opUser']['health'];

        if($userStat==$opStat || $userStat>$opStat){
            $changeXp = $data['user']['xp']+1;
            $changeMoney = $data['user']['money']+200;
            $changeHealth = $data['user']['health']+100;
            $changeLevel=$data['user']['level'];
            $sep='';

            $updData= array(
                'xp' =>$changeXp,
                'money'=>$changeMoney,
                'health'=>$changeHealth
            );
            $updStats=$this->User->updatePlayerData($fuser,$updData);

            if($data['user']['xp']%3==0){
                $changeLevel=$data['user']['level']+1;
                $updlvl=array('level'=>$changeLevel);
                $updStats=$this->User->updatePlayerLevel($fuser,$updlvl);
            }
            $changeFight = $data['opUser']['fight']=0;
            $udpFight=array('fight'=>$changeFight);
            $updStats=$this->User->updatePlayerLevel($opponentUser,$udpFight);

            echo $changeLevel.', '.$changeMoney.', '.$changeHealth;

        }
        else{
            echo "won";
        }
    }
}

```

Figure 49: Υπολογισμός Στατιστικών Ανάμεσα Σε Δύο Χρήστες Που Παλεύουν

Εάν ο χρήστης χάσει θα δει το ανάλογο μήνυμα στην οθόνη του. Εάν νικήσει θα δει επίσης το ανάλογο μήνυμα στην οθόνη του αλλά με την βοήθεια της εντολής splitθα αυξηθούν τα ανάλογα στατιστικά του και ο αντίπαλος που έχασε θα διαγραφεί από την λίστα του Κολοσσαίου.

```

public IEnumerator FightCol(WWW w_fight)
{
    yield return w_fight;

    if (w_fight.error == null && w_fight.text!="won")
    {
        victory.SetActive(true);
        answer = w_fight.text.ToString();
        yield return new WaitForSeconds(3);
        users = answer.Split(","); // PAIRNEI APO TON CONTROLLER TIS PHP STIN FUNCTION guser TO $data kai opou vlepei ',' to vazei se ena keli tou pinaka
        displayDataFromPhp_Instance.stats[4] = users[1]; //gia na valei sto money tin kainourgia timi tis vasis
        displayDataFromPhp_Instance.stats[5] = users[2]; // gia na valei sto health tin kainourgia timi tis vasis
        displayDataFromPhp_Instance.stats[1] = users[0]; //gia na valei sto level tin kainourgia timi tis vasis
        UnityEngine.SceneManagement.SceneManager.LoadScene("New! 3D - Demo");
    }
    else
    {
        defeat.SetActive(true);
        yield return new WaitForSeconds(3);
        UnityEngine.SceneManagement.SceneManager.LoadScene("New! 3D - Demo");
    }
}

```

Figure 50: Η Εντολή Split Για Την Ενημέρωση Των Στατιστικών Του Χρήστη

## 5 Επίλογος

### 5.1 Συμπεράσματα

Τα ηλεκτρονικά παιχνίδια είναι από τους πιο αναπτυσσόμενους κλάδους της πληροφορικής. Σκοπό έχουν την ψυχαγωγία των χρηστών καθώς και να δημιουργήσουν σχέσεις μεταξύ τους. Όσο προχωράει η τεχνολογία τόσο και τα παιχνίδια γίνονται πιο ωραία από άποψη γραφικών και πιο σωστά φτιαγμένα. Τα τελευταία χρόνια κίολας έχουν αναπτυχθεί πολλές κοινότητες στο διαδίκτυο λόγω των online παιχνιδιών. Βέβαια, λόγω της συνεχής εξέλιξης των ηλεκτρονικών παιχνιδιών, οι απαιτήσεις των χρηστών είναι πιο πολλές όπως είναι αναμενόμενο και φυσικά για αυτό τώρα με τα game engines όλοι μπορούν να ασχοληθούν να φτιάξουν ένα παιχνίδι σε αντίθεση με το παρελθόν, όταν έφτιαχναν μόνο οι εταιρείες.

### 5.2 Δυσκολίες Στην Υλοποίηση

Όπως ήταν λογικό κατά την διάρκεια της δημιουργίας του παιχνιδιού υπήρχαν πολλές δυσκολίες. Λόγω της ελλιπής γνώσης μου σε θέματα προγραμματισμού χρειάστηκε να μάθω πολλά tutorials καθώς και να ρωτήσω πολλά πράγματα στο forum της unity και στο forum

stackoverflow. Λόγω και άλλων υποχρεώσεων η δημιουργία του παιχνιδιού κράτησε πολύ παραπάνω χρόνο και ήταν πιο δύσκολο το να γίνει το παιχνίδι καλύτερο.

### 5.3 Δυνατότητες Βελτίωσης Του Παιχνιδιού

Βελτιώσεις πολλές, λόγω το ότι δεν είμαι καλός στον σχεδιασμό, θα χρειαστεί το interface του παιχνιδιού. Για παράδειγμα το login-register form θα μπορούσε να ήταν κάτι καλύτερο από ένα απλό τετράγωνο. Άλλο παράδειγμα είναι το avatar στην σκηνή με τον χάρτη. Θα μπορούσε να ήταν στην θέση του ένας samurai ή ένας knight, ανάλογα τι έχει διαλέξει ο χρήστης, και με animation να περπατούσε στον χάρτη.

Επίσης βελτιώσεις θα μπορούσαν να γίνουν στις μάχες. Δηλαδή εάν ο αντίπαλος είναι πιο δυνατός να υπήρχε ένα ποσοστό νίκης και όχι να νικούσε πάντα. Έτσι θα είχαν ευκαιρία όλοι να ανεβάσουν τα στατιστικά τους.

## 5.4 ΒΙΒΛΙΟΓΡΑΦΙΕΣ

[en.wikipedia.org](http://en.wikipedia.org)