



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών  
«Πληροφορική»

**Μεταπτυχιακή Διατριβή**

Τίτλος Διατριβής	<b>Διερεύνηση δυνατοτήτων Vuforia και Unity για ανάπτυξη εφαρμογών επαυξημένης και εικονικής Πραγματικότητας</b>  <b>Investigation of Vuforia and Unity platforms for the development of augmented and virtual reality applications</b>
Όνοματεπώνυμο Φοιτητή	<b>Ηλίας Παπαδόδημας</b>
Πατρώνυμο	<b>Παναγιώτης</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ17037</b>
Επιβλέπων Καθηγητής	<b>Θεμιστοκλής Παναγιωτόπουλος</b>

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

ΚΑΘΗΓΗΤΗΣ  
Θ.  
ΠΑΝΑΓΙΩΤΟΠΟΥΛΟΣ

(υπογραφή)

ΚΑΘΗΓΗΤΗΣ  
Δ.  
ΑΠΟΣΤΟΛΟΥ

(υπογραφή)

ΕΠΙΚΟΥΡΟΣ  
ΚΑΘΗΓΗΤΗΣ  
Α. ΠΙΚΡΑΚΗΣ

Περιεχόμενα

Κατάλογος Εικόνων	4
Περίληψη – Abstract	5
<b>1.</b> Εισαγωγή	6
<b>1.1.</b> Μηχανή παιχνιδιών Unity	6
<b>1.2.</b> Vuuforia για την επίτευξη του Augmented Reality (AR)	7
<b>2.</b> Υλοποίηση Παιχνιδιού	8
<b>2.1.</b> Α' Φάση	8
<b>2.2.</b> Quality Check Augmented Reality	12
<b>2.3.</b> Β' Φάση	13
<b>3.</b> Game Manual	19
<b>4.</b> Scripting	21
<b>5.</b> Αποτελέσματα – Επεξήγηση Επιλογών	22
Συμπεράσματα και μελλοντικές επεκτάσεις	23
System Specifications	24
Βιβλιογραφία	25

Κατάλογος Εικόνων

Εικόνα 2.1.1 : License Manager για το Vuforia Engine	8
Εικόνα 2.1.2 : Δημιουργία schema στην βάση της Vuforia, για τα ολογράμματα	9
Εικόνα 2.1.3 : Εισαγωγή εικόνων, από τα οποία θα βγαίνουν τα ολογράμματα	9
Εικόνα 2.1.4 : Απαιτούμενο package αρχείο της Unity για την εισαγωγή των target images	10
Εικόνα 2.1.5 : Εισαγωγή AR Camera	10
Εικόνα 2.1.6 : Configuration Vuforia engine	11
Εικόνα 2.1.7 : License Vuforia engine	11
Εικόνα 2.1.8 : Εισαγωγή του 3D μοντέλου	11
Εικόνα 2.2.1 : Πρώτη επαφή με Augmented Reality	12
Εικόνα 2.3.1 : Παράδειγμα 3D μοντέλου στο Asset Store της Unity	13
Εικόνα 2.3.2 : Τελική εξαγωγή παιχνιδιού	13
Εικόνα 2.3.3 : Menu και πρώτη επαφή με το παιχνίδι	14
Εικόνα 2.3.4: Εναλλαγή μενού	14
Εικόνα 2.3.5 : Διαχωρισμός του project σε μενού και παιχνίδι	15
Εικόνα 2.3.6 : Κουμπί About	15
Εικόνα 2.3.7 : Κύρια οθόνη του παιχνιδιού	16
Εικόνα 2.3.8 : Onclick events κουμπιού	16
Εικόνα 2.3.9 : Animator που χρησιμοποιήθηκε	17
Εικόνα 2.3.10 : Παράδειγμα πολύπλοκου animator	17
Εικόνα 2.3.11 : Επιστροφή κουμπιών στην αρχική κατάσταση	18
Εικόνα 3.1 : Κατάσταση Player 2 μετά την επίθεση	20
Εικόνα 4.1 : Συλλογή Scripts στον inspector	21

## Περίληψη

Δημιουργήθηκε παιχνίδι όπου ο παίκτης καλείται να αντιμετωπίσει σε μάχη στρατηγικής ηλεκτρονικό αντίπαλο. Ονομάζεται μάχη στρατηγικής, διότι ο παίκτης καλείται να διαλέξει ανάμεσα σε επιλογές κινήσεων, με σκοπό το βέλτιστο αποτέλεσμα. Οι χαρακτήρες κινούνται πάνω σε συγκεκριμένα στοιχεία της πλατφόρμας, στην προκειμένη περίπτωση σε κάρτες. Η υλοποίηση του παιχνιδιού έγινε με γλώσσα C# σε περιβάλλον Visual Studio 2019. Η σύνδεση ανάμεσα στον κώδικα αυτόν, το παιχνίδι, τις κινήσεις καθώς επίσης και τα γραφικά έγιναν μέσω της Unity, μιας μηχανής παραγωγής παιχνιδιών για πολλές πλατφόρμες όπως υπολογιστές, κινητά κλπ. Χρησιμοποιήθηκε ένα είδος τακτικής σεναρίων, που ανάλογα τα κουμπιά που θα πατήσει ο παίκτης η μάχη μπορεί να εξελιχθεί με διαφορετικό τρόπο, αλλά και UI elements με σκοπό την διαδραστικότητα παίκτη-χαρακτήρα. Επίσης έχει προστεθεί στο παιχνίδι η δυνατότητα της εικονικής πραγματικότητας (AR Reality) που καθιστά εφικτή την δημιουργία εικονικού ολογράμματος. Για αυτήν την δυνατότητα χρησιμοποιήθηκε πάνω στην Unity, η πλατφόρμα Vuforia. Πρόκειται για μία προσπάθεια ακαδημαϊκού επιπέδου, με μεγάλο περιθώριο επέκτασης. Η εξαγωγή του παιχνιδιού έχει γίνει για υπολογιστές για λόγους ευκολίας για την παρουσίαση.

## Abstract

A game was created in which the player is called to face an electronic opponent in a strategy battle. It is called a strategy battle, because the player is asked to choose between movement options, in order to get the best result. The characters move on specific elements of the platform, in this case on cards. The game was implemented in C# language in Visual Studio 2019 environment. The connection between the code, the game, the animations as well as the graphics was made through Unity, a game production engine for many platforms such as computers, mobile phones, etc. Something like a tactic of scenarios was used, that depending on the buttons that the player will press, the battle can evolve in different ways. What is more, UI elements were also applied for the purpose of player-character interactivity. The possibility of virtual reality (AR Reality) has also been added to the game, which makes it possible to create a virtual hologram. For this feature, the Vuforia platform was used on Unity.

This is an effort at the academic level, with great scope for expansion. The game has been exported to computers for ease of presentation.

## 1. Εισαγωγή

Το παρόν project αφορά την προσπάθεια δημιουργίας παιχνιδιού-εφαρμογής που παρουσιάζει στοιχεία από παιχνίδια στρατηγικής και ρόλων σε περιβάλλον εικονικής πραγματικότητας. Στην ουσία είναι για να διερευνηθεί η αλληλεπίδραση εικονικής πραγματικότητας, σε πλατφόρμα Unity. Για την υλοποίηση του, προστέθηκε στην πλατφόρμα Unity, η επέκταση Vuforia Engine. Με αυτήν την επέκταση, είναι εφικτό στο τελικό παιχνίδι που θα παραχθεί, όταν «βλέπει» η κάμερα μια εικόνα την οποία έχω ορίσει, να εμφανίσει το μοντέλο που επίσης θα έχει οριστεί. Ένα γνωστό παράδειγμα παιχνιδιού που δουλεύει με αντίστοιχο τρόπο, με Augmented Reality, είναι το Pokemon GO.

Τα βιντεοπαιχνίδια στρατηγικής είναι ένα είδος βιντεοπαιχνιδιών που τονίζουν την επιδέξια σκέψης και σχεδιασμό για να επιτευχθεί η νίκη. Συγκεκριμένα, ένας παίκτης πρέπει να σχεδιάσει μια σειρά από ενέργειες ενάντια σε έναν ή περισσότερους αντιπάλους, και η μείωση των δυνάμεων του εχθρού είναι συνήθως ένας στόχος. Η νίκη επιτυγχάνεται με τον ανώτερο σχεδιασμό και το στοιχείο της τύχης παίζει μικρότερο ρόλο.

Βιντεοπαιχνίδι ρόλων (RPG – role playing game) είναι το είδος βιντεοπαιχνιδιού στο οποίο οι μάχες διαδραματίζονται με τη μορφή εντολών και η ιστορία εξελίσσεται με την εξερεύνηση του κόσμου του παιχνιδιού. Τα παιχνίδια ρόλων αν και ξεκίνησαν από αρκετά απλοϊκά με κείμενο σε οπτικά πλούσιες 3D εμπειρίες. Στα RPG ο παίκτης καλείται να χειριστεί τον κεντρικό χαρακτήρα με το σύστημα εντολών, ενώ η αποτελεσματικότητα του χαρακτήρα εξαρτάται από τη στατιστική ανάπτυξή του. Ο κάθε χαρακτήρας μπορεί να έχει στην κατοχή του διάφορα όπλα που βοηθούν στη δύναμη της επίθεσης ή της άμυνάς του.

### 1.1. Μηχανή παιχνιδιών Unity

Η Unity είναι μια μηχανή παιχνιδιών, δηλαδή ένα σύστημα λογισμικού σχεδιασμένο για τη δημιουργία και την ανάπτυξη βιντεοπαιχνιδιών<sup>12</sup>, που δημιουργήθηκε από την Unity Technologies το 2005. Κάνει εφικτή την υποστήριξη και τον συνδυασμό διαφορετικών πλατφορμών, όπως iOS, Android, PlayStation 4, Steam VR, Google's ARCore, Vuforia και άλλα. Η Unity μπορεί να χρησιμοποιηθεί για την δημιουργία 3d μοντέλων, VR αλλά και για παιχνίδια επαυξημένες πραγματικότητας.

Αν και αρχικά η Unity προοριζόταν μόνο για Mac OS X, στην συνέχεια άρχισε να υποστηρίζεται και από Microsoft Windows. Κάθε καινούργιο της update προσφέρει στους χρήστες της παραπάνω features και δυνατότητες, αλλά και καλύτερη υποστήριξη. Ειδικά οι τελευταίες εκδόσεις της Unity (π.χ. του 2018 και 2020) διαθέτουν υψηλής ανάλυσης rendering και καινούργια εργαλεία για την καλύτερη αποτύπωση της επαυξημένης πραγματικότητας οδηγώντας σε ένα πιο ολοκληρωμένο και άρτιο αποτέλεσμα.

<sup>1</sup> Ο όρος «μηχανή παιχνιδιού» πρωτοεμφανίστηκε στα μέσα της δεκαετίας του 1990, ειδικά σε σχέση με 3D παιχνίδια. Υπάρχουν, όμως, κάποια προγενέστερα συστήματα στη δεκαετία του 1980 τα οποία επίσης θεωρούνται μηχανές παιχνιδιών, όπως το σύστημα SCUMM της LucasArts.

<sup>2</sup> Η κεντρική λειτουργικότητα που μια μηχανή παιχνιδιού παρέχει περιλαμβάνει μια μηχανή ("renderer") για 2D και 3D γραφικά, ήχο, scripting, animation, νήματα (threading), όπως και άλλα στοιχεία.

Διερεύνηση δυνατοτήτων Vuforia και Unity για ανάπτυξη εφαρμογών επαυξημένης και εικονικής Πραγματικότητας

Η Unity ενδείκνυται για την δημιουργία παιχνιδιών 2D και 3D και προσφέρει scripting API σε C#. Η συγκεκριμένη μηχανή παιχνιδιών παρέχει όχι μόνο επί πληρωμή άδειες χρήσης, αλλά και δωρεάν. Η δωρεάν άδεια χρήσης χορηγείται σε είτε σε μικρές εταιρίες είτε σε περιπτώσεις προσωπικής χρήσης. Αυτό αποτελεί και ένα από τα σημαντικά πλεονεκτήματα της Unity, καθώς μειώνει τους περιορισμούς χρήσης της.

Ένα άλλο θετικό στοιχείο της Unity είναι το Asset Store της, στο οποίο μπορεί κανείς να βρει 3D και 2D assets και να χρησιμοποιήσει στα παιχνίδια που δημιουργεί. Τέτοια assets χρησιμοποιήθηκαν και στην περίπτωση αυτής της εργασίας, διότι πρόκειται για πακέτα με εργαλεία χρειαζόμενα, όπως για παράδειγμα τους χαρακτήρες, τα 3D μοντέλα αυτών καθώς επίσης και διάφορα animations μάχης που θα χρησιμοποιούσα.

---

## **1.2. Vuforia Engine για την επίτευξη του Augmented Reality (AR)**

Το Vuforia είναι ένα kit ανάπτυξης λογισμικού επαυξημένης πραγματικότητας (SDK) για φορητές συσκευές που επιτρέπει τη δημιουργία εφαρμογών επαυξημένης πραγματικότητας. Χρησιμοποιεί τεχνολογία υπολογιστικής όρασης για να αναγνωρίζει και να παρακολουθεί επίπεδες εικόνες και τρισδιάστατα αντικείμενα σε πραγματικό χρόνο. Αυτή η δυνατότητα εγγραφής εικόνας δίνει την δυνατότητα στους προγραμματιστές να τοποθετούν και να προσανατολίζουν εικονικά αντικείμενα, όπως μοντέλα 3D και άλλα μέσα, σε σχέση με αντικείμενα πραγματικού κόσμου όταν προβάλλονται μέσω της κάμερας μιας κινητής συσκευής. Στη συνέχεια, το εικονικό αντικείμενο παρακολουθεί τη θέση και τον προσανατολισμό της εικόνας σε πραγματικό χρόνο, έτσι ώστε η προοπτική του θεατή για το αντικείμενο να αντιστοιχεί στην προοπτική του στόχου. Φαίνεται, λοιπόν, ότι το εικονικό αντικείμενο είναι μέρος της πραγματικής σκηνής.

Το Vuforia SDK υποστηρίζει μια ποικιλία τύπων στόχων 2D και 3D, συμπεριλαμβανομένων: «Στόχοι εικόνας χωρίς σήμανση», Στόχος μοντέλου 3D και μια μορφή διευθυνσιοδοτούμενου Fiducial Marker, γνωστού ως VuMark. Οι πρόσθετες δυνατότητες του SDK περιλαμβάνουν τον εντοπισμό συσκευών ελευθερίας 6 μοιρών στο διάστημα, την εντοπισμένη ανίχνευση απόφραξης με τη χρήση «εικονικών κουμπιών», την επιλογή στόχου εικόνας χρόνου εκτέλεσης και τη δυνατότητα δημιουργίας και αναδιάρθρωσης συνόλων στόχων μέσω προγραμματισμού κατά την εκτέλεση.

Η Vuforia παρέχει διεπαφές προγραμματισμού εφαρμογών (API) σε γλώσσες C++, Java, Objective-C++ και .NET μέσω μιας επέκτασης στη μηχανή παιχνιδιών Unity. Με αυτόν τον τρόπο, το SDK υποστηρίζει τόσο την εγγενή ανάπτυξη για iOS, Android και UWP, ενώ επιτρέπει επίσης την ανάπτυξη εφαρμογών AR στο Unity που είναι εύκολα φορητές και στις δύο πλατφόρμες.

## 2. Υλοποίηση Παιχνιδιού

### 2.1. Α' Φάση

Τα πρωταρχικά βήματα που ακολουθήθηκαν για να το επιτευχθεί το project είναι τα εξής:

- 1) Αίτηση αδειας (License Key) από την Vuforia, προκειμένου να δηλωθεί πως πρόκειται για εκπαιδευτικό σκοπό ώστε να το έχω δωρεάν.

The screenshot shows the Vuforia Engine Developer Portal interface. At the top, there is a navigation bar with links for Home, Pricing, Downloads, Library, Develop, and Support. The user is logged in as 'Hello papiliias'. Below the navigation bar, there are two tabs: 'License Manager' (selected) and 'Target Manager'. The main content area shows the 'License Manager' page for a license named 'ptuxiaki'. There are two sub-tabs: 'License Key' (selected) and 'Usage'. The 'License Key' tab displays a long alphanumeric license key. Below the key, there are details about the license: Plan Type: Develop, Status: Active, Created: Jun 04, 2020 12:28, License UUID: b1b9b72f4f164738a04b8a537da0f94f, Permissions: Advanced Camera, External Camera, Model Targets, Watermark, and History: License Created - Jun 04, 2020 12:28.

#### Εικόνα 2.1.1 : License Manager για το Vuforia Engine

- 2) Δημιουργία ένα schema μέσα στην βάση της ίδιας της Vuforia. Αυτό το βήμα είναι απαραίτητο ώστε να μπορώ μετέπειτα μέσα στην Unity να «καλέσω» την βάση και να προστεθούν οι φωτογραφίες. Συγκεκριμένα, στην εργασία αυτή έχουν χρησιμοποιηθεί τα *scifiwarrior* και *devil*.



The screenshot shows the 'License Manager' section of the Vuforia Engine Developer Portal. At the top, there is a navigation bar with links for Home, Pricing, Downloads, Library, Develop, and Support. The user is logged in as 'Hello papillas'. Below the navigation bar, there are two tabs: 'License Manager' (selected) and 'Target Manager'. The main heading is 'License Manager', with two buttons: 'Get Development Key' and 'Buy Deployment Key'. Below this, there is a text prompt: 'Create a license key for your application.' and a search input field. A table lists the license keys:

Name	Primary UUID	Type	Status	Date Modified
ptuxiaki	N/A	Develop	Active	Jun 04, 2020

At the bottom, there is a pagination control showing '25 per page', 'Showing 1-1 of 1', and 'Last updated: Today 2:15 PM' with a 'Refresh' button.

**Εικόνα 2.1.2 : Δημιουργία schema στην βάση της Vuforia, για τα ολογράμματα**

- 3) Αργότερα, στην βάση αυτή της Vuforia, είναι αναγκαίο να οριστούν σαν targets τα μοντέλα των παραπάνω φωτογραφιών, δηλαδή οι εικόνες που θα αναγνωρίζει η Vuforia Engine, μέσα από τις οποίες θα βγαίνουν και τα αντίστοιχα ολογράμματα των χαρακτήρων.

The screenshot shows the 'Target Manager' interface for the 'ptuxiaki' target. The target is identified as a 'Device'. Below this, there is a search bar for 'Targets (7)' and an 'Add Target' button. A 'Download Database (All)' button is also present. A table lists the targets:

Target Name	Type	Rating	Status	Date Modified
scifi	Single Image	★★★★★	Active	Jun 04, 2020 16:05
ratassasin	Single Image	★★★★★	Active	Jun 04, 2020 16:05
herowarrior	Single Image	★★★★★	Active	Jun 04, 2020 16:05
devil	Single Image	★★★★★	Active	Jun 04, 2020 16:05
chestmonster	Single Image	★★★★★	Active	Jun 04, 2020 16:04
biomecha	Single Image	★★★★★	Active	Jun 04, 2020 16:04
versus-vs-fight-battle-screen-bac...	Single Image	★★★★★	Active	Jun 04, 2020 13:15








**Εικόνα 2.1.3 : Εισαγωγή εικόνων, από τα οποία θα βγαίνουν τα ολογράμματα**

- 4) Μαρκάρω τις φωτογραφίες που θέλω (στην προκειμένη περίπτωση αυτές των scifi και devil) και ύστερα προχωρώ με Download Database. Στη συνέχεια, επιλέγω να το κατεβάσω σαν Unity Editor, ώστε να το έχω ως Unity package. Από τον File Explorer της Unity και κάνοντας δεξί κλικ στο Import Asset επιλέγεται το αρχείο package που κατέβασα νωρίτερα.

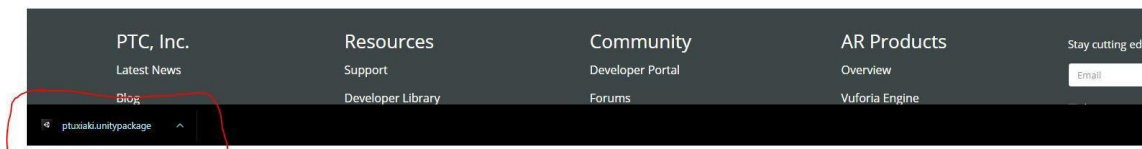
ptuxiaki Edit Name  
Type: Device

Targets (7)

Add Target Download Database (All)

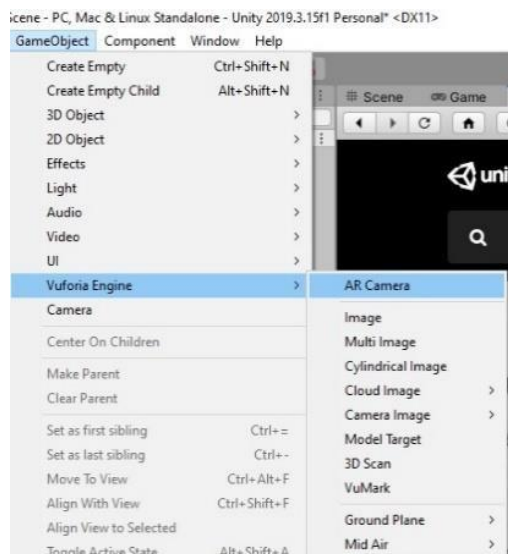
Target Name	Type	Rating	Status	Date Modified
<input type="checkbox"/>  scifi	Single Image	★★★★★	Active	Jun 04, 2020 16:05
<input type="checkbox"/>  ratassasin	Single Image	★★★★★	Active	Jun 04, 2020 16:05
<input type="checkbox"/>  herowarrior	Single Image	★★★★★	Active	Jun 04, 2020 16:05
<input type="checkbox"/>  devil	Single Image	★★★★★	Active	Jun 04, 2020 16:05
<input type="checkbox"/>  chestmonster	Single Image	★★★★☆	Active	Jun 04, 2020 16:04
<input type="checkbox"/>  biomecha	Single Image	★★★★★	Active	Jun 04, 2020 16:04
<input type="checkbox"/>  versus-vs-fight-battle-screen-bac...	Single Image	★★★★☆	Active	Jun 04, 2020 13:15

Last updated: Today 04:59 PM Refresh



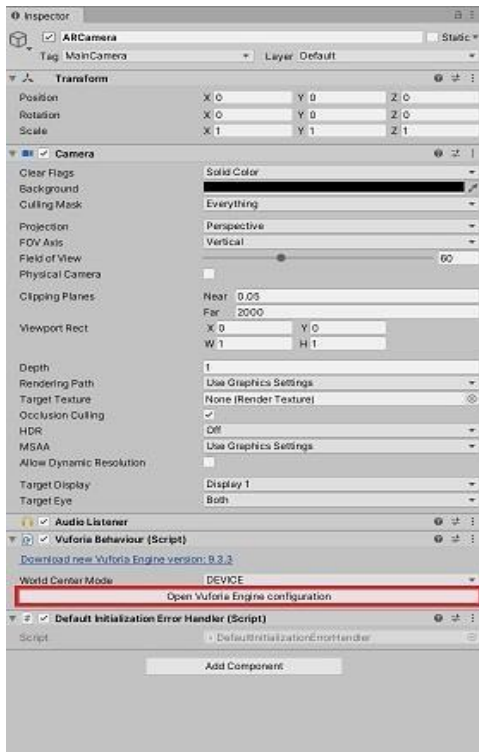
Εικόνα 2.1.4 : Απαιτούμενο package αρχείο της Unity για την εισαγωγή των target images

- 5) Εισαγωγή του απαιτούμενου GameObject, δηλαδή της AR Camera, ώστε να λειτουργήσει το augmented reality. Πλέον η χρήση της απλής Camera ως game object δεν είναι πλέον απαραίτητη.

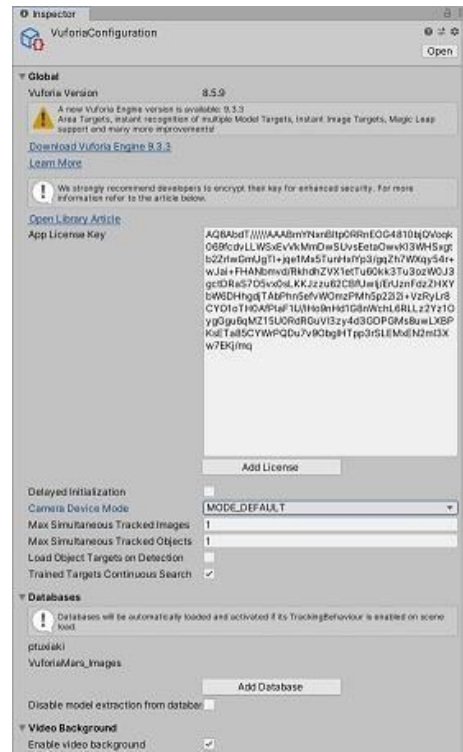


Εικόνα 2.1.5 : Εισαγωγή AR Camera

6) Στη συνέχεια, ακολουθεί και το απαιτούμενο configuration του Vuforia Engine μέσα στην Unity.

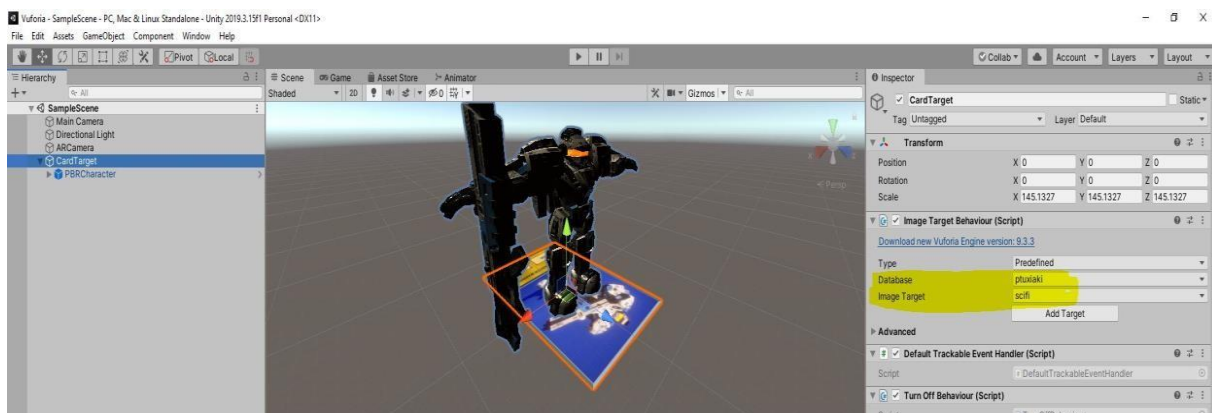


Εικόνα 2.1.6 : Configuration Vuforia engine



Εικόνα 2.1.7 : License Vuforia engine

7) Στην συνέχεια, ορίζω την βάση που θα «βλέπει» η Vuforia, εκείνη δηλαδή που είχα δημιουργήσει στο site της, και πάνω σε αυτήν πατάω για να το ορίσω τι ολόγραμμα θα εμφανίζεται στην οθόνη. Από το Asset Store της Unity έχω «κατεβάσει» αρκετά 3D models. Στο συγκεκριμένο παράδειγμα χρησιμοποιώ το μοντέλο του scifi warrior. Στο πεδίο Animator που έχει το μοντέλο, του έχω προσθέσει την κίνηση που θα έχει. Στο σημείο αυτό θα μπορούσαν να προστεθούν και οποιοσδήποτε άλλες κινήσεις.



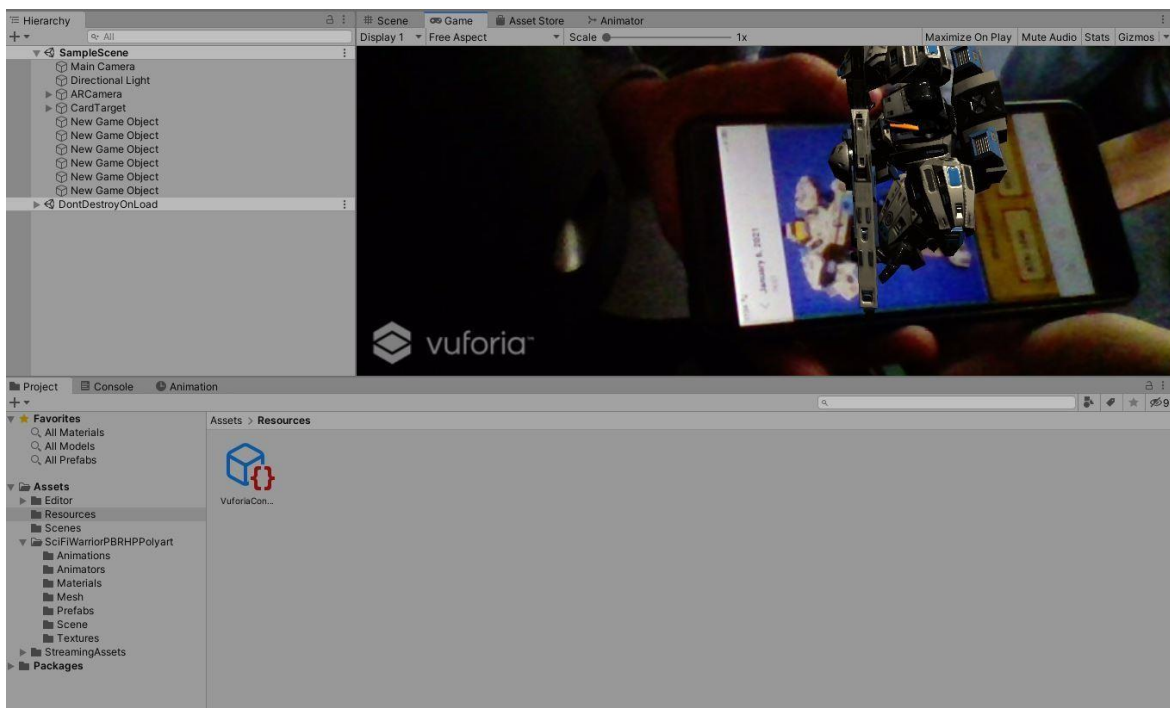
Εικόνα 2.1.8 : Εισαγωγή του 3D μοντέλου

## 2.2. Quality check Augmented Reality

Μέσα από την Unity, πατήθηκε το κουμπί Play ώστε να «τρέξει» το project χωρίς να χρειαστεί να γίνει η εξαγωγή του σε αρχείο .exe. Ανοίγοντας την επισκόπηση του παιχνιδιού, δίνεται η δυνατότητα εναλλαγής των tabs. Στο project αφού έχει μπει το στοιχείο AR Camera, ανοίγει την κάμερα της συσκευής στην οποία τρέχει. Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκε η κάμερα του laptop.

Επιλέγοντας το tab Game βλέπουμε τι φαίνεται από την κάμερα. Αν στην κάμερα φέρουμε με κάποιον τρόπο την κάρτα του χαρακτήρα, είτε ως εκτυπωμένη κάρτα είτε -όπως στην δική μου προσπάθεια- την φωτογραφία της κάρτας από το κινητό, βλέπουμε ότι σε αυτή εμφανίζεται το μοντέλο. Κατά τη δημιουργία του μοντέλου όρισα σε ποιο σημείο της κάρτας αυτό θα εμφανίζεται. Αυτό, πρακτικά, σημαίνει ότι όσο και αν μετακινήσω την φωτογραφία στην κάμερα, το ίδιο θα μετακινείται και το μοντέλο, αλλά ταυτόχρονα χωρίς να αλλάζει την θέση του πάνω στην ίδια την κάρτα. Επιπλέον, για το λόγο του ότι πρόκειται για 3D μοντέλο που αντικατοπτρίζεται σε εικονική πραγματικότητα, σε οποιαδήποτε πλευρά και να περιστρέψω την κάρτα, αλλάζει αντίστοιχα και κατεύθυνση το μοντέλο. Έτσι, κανένας μπορεί να δει όλες τις πλευρές των χαρακτήρων του παιχνιδιού. Αυτό το feature καθιστά το παιχνίδι ιδιαίτερα ενδιαφέρον καθώς παρέχει στον χρήστη τη δυνατότητα να περιεργαστεί τους χαρακτήρες και να έχει μια ξεχωριστή εμπειρία.

Τέλος, αξιοσημείωτο είναι ότι στο μοντέλο έχει προστεθεί με συγκεκριμένη διαδικασία που θα αναλυθεί αργότερα (animator), ώστε να του δώσει κίνηση για να είναι πιο πειστικό. Αυτό το κομμάτι ανοίγει πολλές πόρτες για ιδέες παιχνιδιών στο μέλλον, αν υπάρξει και ο κατάλληλος εξοπλισμός.

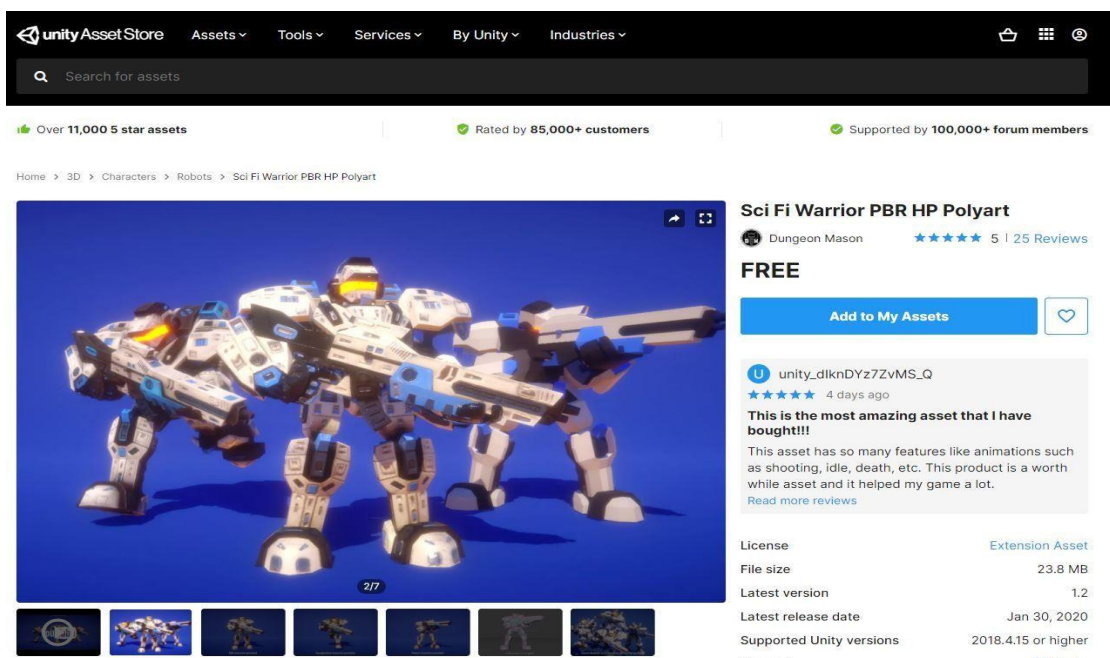


Εικόνα 2.2.1 : Πρώτη επαφή με Augmented Reality

Στο σημείο αυτό, το Unity έκανε update. Συγκεκριμένα από 2019.3.15f1 μετατράπηκε σε 2020.1.4f1. Το γεγονός αυτό είχε ως αποτέλεσμα πολλά στοιχεία (όπως το UI) να αλλάξουν. Το κυριότερο πρόβλημα, όμως, που προκάλεσε αυτό το update ήταν ότι έπαψε να είναι πλέον συμβατό με τη Vuforia. Επί της ουσίας, τα περισσότερα από τα scripts που αφορούσαν στη Vuforia διαγράφηκαν καθιστώντας την πλέον μη λειτουργική. Πλέον ήταν αδύνατο να συνεχιστεί η αρχική ιδέα του παιχνιδιού, επομένως ξεκίνησε ένα νέο project, παραλλαγή του αρχικού και προσαρμοσμένου στους περιορισμούς που προέκυψαν από το update της Unity.

### 2.3. Β' Φάση

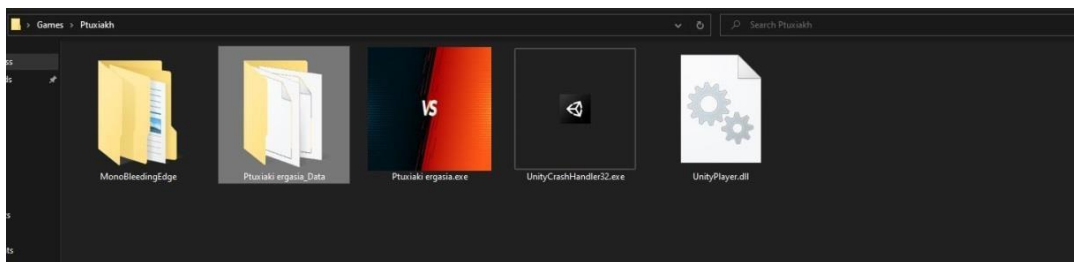
Περνώντας πλέον στο κύριο τμήμα της υλοποίησης του παιχνιδιού, στο κομμάτι δηλαδή που το παιχνίδι δημιουργείται από το μηδέν και αποκτά τη μορφή που ο προγραμματιστής επιθυμεί, ακολουθήθηκαν οι ενέργειες που θα αναλυθούν παρακάτω. Για την υλοποίηση του παιχνιδιού χρειάστηκε πρώτα να αποφασιστούν τα βασικά –και απαραίτητα- στοιχεία. Τέτοια είναι ο Canvas ώστε να οριστεί το πραγματικό background του παιχνιδιού, τα ImageTarget πάνω στα οποία θα εμφανίζονται οι χαρακτήρες, καθώς επίσης και οι ίδιοι οι χαρακτήρες. Προς το παρόν - και για να γίνει πιο εύκολη η παρουσίαση- έχουν χρησιμοποιηθεί μόνο δύο χαρακτήρες, το scifiwarrior και το devil.



Εικόνα 2.3.1 : Παράδειγμα 3D μοντέλου στο Asset Store της Unity

Ήδη από την αρχή έχει δοθεί έμφαση στις λεπτομέρειες που κάνουν την εμφάνιση του παιχνιδιού πιο περιποιημένη και προσεγμένη. Συγκεκριμένα, στο build που έγινε για την εξαγωγή του project, ορίστηκε ώστε το εικονίδιο της εφαρμογής να είναι μια φωτογραφία “versus” (versus-vs-fight-battle-screen-bac)

και όχι το κλασσικό λογότυπο που βγάζει η Unity.



Εικόνα 2.3.2 : Τελική εξαγωγή παιχνιδιού

Το παιχνίδι αποτελείται από το αρχικό μενού, το οποίο ανοίγει πρώτο, μόλις τρέξει η εφαρμογή (πρώτη σκηνή). Σαν δεύτερη σκηνή ακολουθεί το καθαυτό παιχνίδι. Στο μενού βρίσκονται κουμπιά με επιλογές. Υπάρχει το κουμπί **Play**, το **About** το οποίο πατώντας το ο χρήστης μπορεί να δει πιο συγκεκριμένες λεπτομέρειες που αφορούν την προέλευση και τον λόγο δημιουργίας του project, καθώς επίσης και το κουμπί **Quit**.



Εικόνα 2.3.3 : Menu και πρώτη επαφή με το παιχνίδι

Αυτή η εναλλαγή μεταξύ μενού – παιχνιδιού, επιτυγχάνεται αν ορίσουμε στο build της εφαρμογής 2 σκηνές. Χρειάζεται, δηλαδή, να διαχωρίσουμε την σκηνή του μενού από εκείνη του παιχνιδιού. Ορίζεται σαν onclick event του κουμπιού **Play** χρησιμοποιώντας των παρακάτω κώδικα:

```

public class MainMenu : MonoBehaviour {

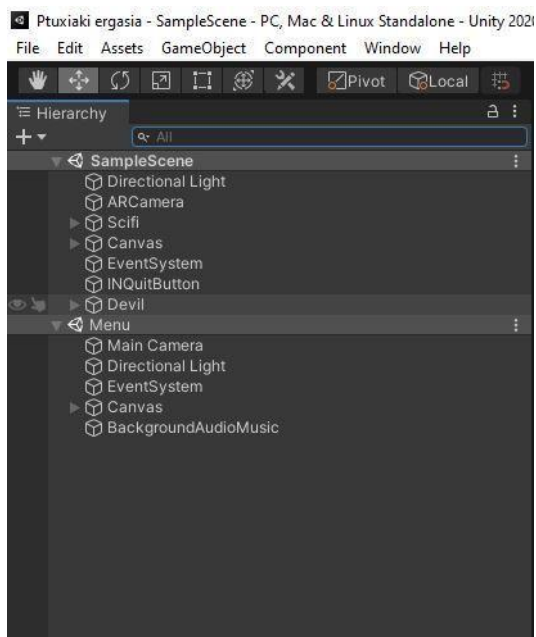
    public void PlayGame ()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    public void QuitGame ()
    {
        Application.Quit();
    }
}

```

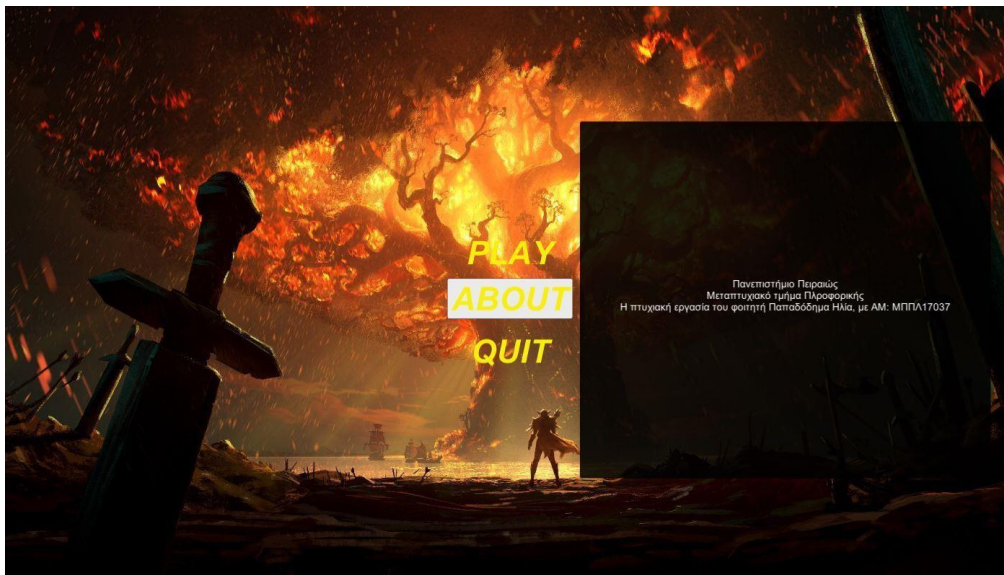
Εικόνα 2.3.4: Εναλλαγή Menu

Είναι, επίσης, πολύ σημαντικό να τονιστεί ότι χρειάζεται να οριστεί ποια σκηνή θα ξεκινήσει πρώτη και ποια θα ακολουθήσει στη συνέχεια. Στο παιχνίδι μου φαίνονται με αυτόν τον τρόπο οι 2 σκηνές, ορισμένες βέβαια αντίθετα κατά το build:



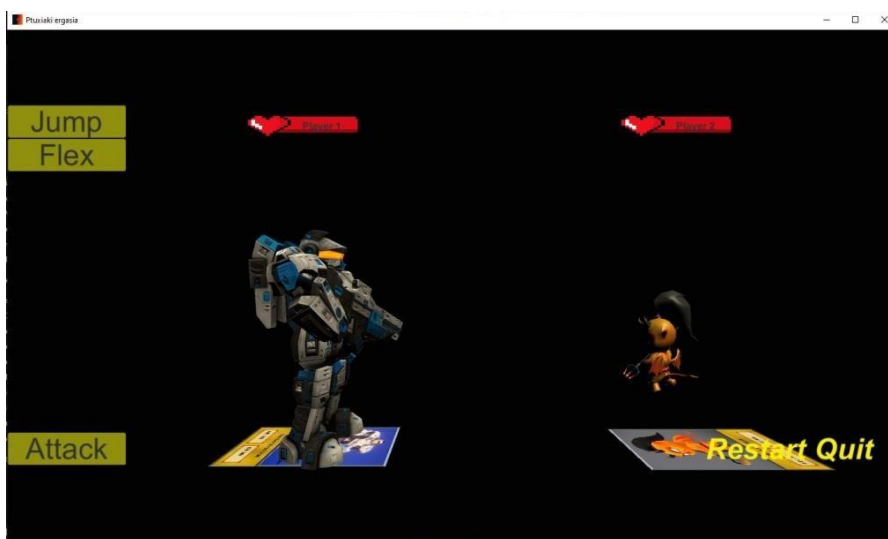
Εικόνα 2.3.5 : Διαχωρισμός του project σε μενού και παιχνίδι

Ξεκινώντας το παιχνίδι ανοίγει το μενού που έχει φωτογραφία background σχετική με το θέμα του παιχνιδιού, όπως επίσης έχει και ένα ηχητικό εφέ που παίζει για την διασκέδαση του χρήστη, καθώς διαβάζει τις λεπτομέρειες από την επιλογή **About**.



Εικόνα 2.3.6 : Κουμπί About

Περνώντας στη δεύτερη σκηνή, δηλαδή πατώντας το κουμπί **Play**, ο αριστερός χαρακτήρας, ο Player 1, όπως γράφει και στην μπάρα ζωής (health bar), είναι ο χαρακτήρας που κινεί ο παίκτης. Στην οθόνη φαίνεται ότι πατάει πάνω σε μία κάρτα, που έχει ζωγραφισμένο το είδωλο του.



Εικόνα 2.3.7 : Κύρια οθόνη του παιχνιδιού

Πάνω στον Canvas της δεύτερης σκηνής, προσθέτω και τα κουμπιά με τα οποία ο παίκτης μπορεί να χειριστεί τον χαρακτήρα του. Ωστόσο, αυτά τα κουμπιά είναι αρκετά απλοϊκά, με εξαίρεση το κουμπί της επίθεσης. Βέβαια ακόμα και έτσι, η δημιουργία των κουμπιών αυτών είναι ιδιαίτερα απαιτητική, επειδή πρέπει να συνδεθεί και κώδικας με scripts, όπως επίσης και των controllers που θα αναφερθούν παρακάτω. Πρακτικά, δηλαδή, το πάτημα ενός κουμπιού κάνει πολλά περισσότερα από ένα πράγμα. Χαρακτηριστικό παράδειγμα αποτελεί το κουμπί **Attack** (Επίθεση του αριστερού χαρακτήρα, δηλαδή του παίχτη, προς τον αντίπαλο). Το κουμπί αυτό προκαλεί την αλλαγή

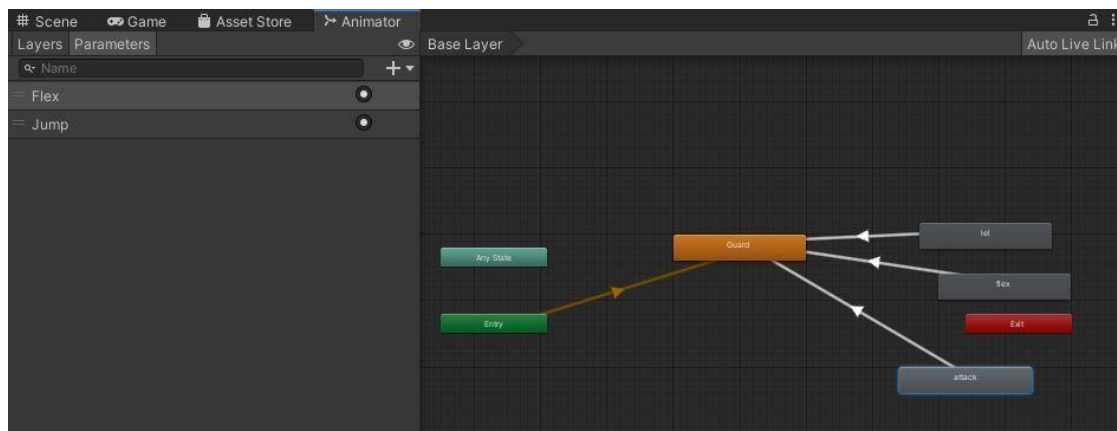


κατάστασης του αριστερού χαρακτήρα ώστε να κάνει την κίνηση της επίθεσης και, επίσης, περιλαμβάνει την επιστροφή του χαρακτήρα στην αρχική του κατάσταση. Ως δεύτερο onclick event έχει την μείωση του slider της μπάρας ζωής του αντίπαλου χαρακτήρα. Παρακάτω ακολουθεί παράδειγμα με φωτογραφία, που δείχνει ότι ένα και μόνο κουμπί από τα πολλά του παιχνιδιού έχουν περισσότερα από ένα onclick events.



**Εικόνα 2.3.8 : Onclick events κουμπιού**

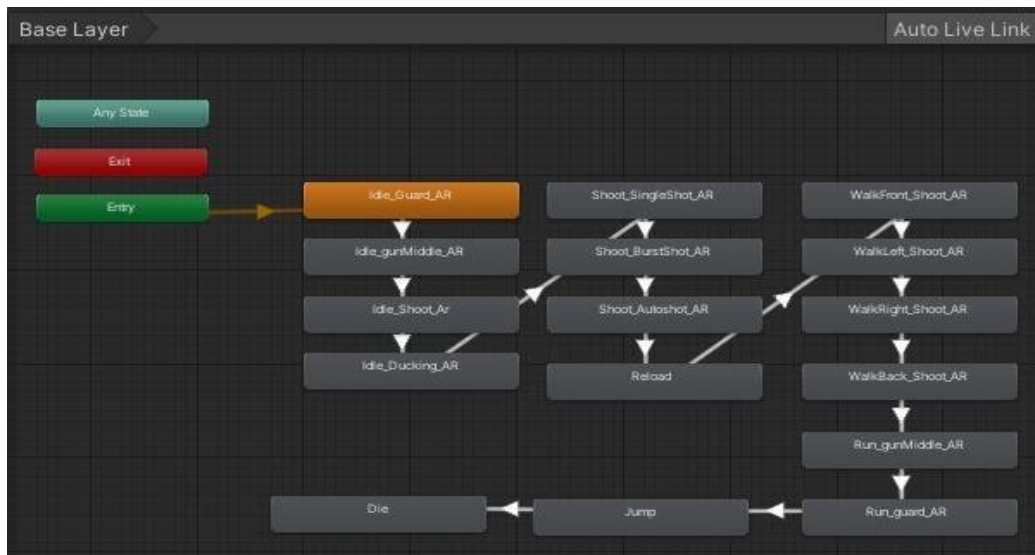
Εδώ λοιπόν πρέπει να αναφερθούν και να εξηγηθούν οι έννοιες Controller και Animation. Ο controller είναι υπεύθυνος τόσο για τα Physics όσο και για τις εναλλαγές μεταξύ των διάφορων κινήσεων που έχουν είτε προστεθεί είτε δημιουργηθεί στο παράθυρο του Animator. Για την λειτουργία του Controller αυτού, υπάρχει Script σε ξεχωριστή κατηγορία που θα αναφερθεί αργότερα. Όσον αφορά στο παράθυρο του Animator, παρόλο που το Asset που χρησιμοποιήθηκε είχε μέσα αρκετό υλικό, δημιουργήθηκε ξεχωριστά για το παιχνίδι αυτό ένα αρκετά απλοϊκό Animation. Έτσι, ο χαρακτήρας ξεκινάει με αρχική κατάσταση την Idle Guard, και βάσει των κινήσεων που θα κάνει ο χρήστης μπορεί να αλλάζει καταστάσεις ή να κάνει ενέργειες, και τελικά αν δεν πεθαίνει, να ξαναεπιστρέφει στην αρχική του κατάσταση, δηλαδή στο αρχικό του Animation.



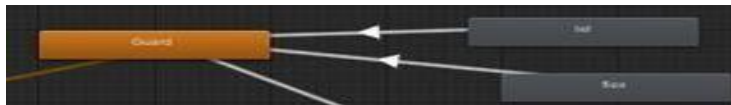
**Εικόνα 2.3.9 : Animator που χρησιμοποιήθηκε**

Η παρακάτω εικόνα που απεικονίζει τον Animator που ήταν embedded στο asset που κατέβασα, με την πλήρη κινησιολογία του χαρακτήρα sci-fi warrior και είναι χαρακτηριστική ώστε να γίνει πιο κατανοητή η πολυπλοκότητα που μπορεί να υπάρχει σε έναν Animator.

**Εικόνα 2.3.10 : Παράδειγμα πολύπλοκου animator**



Μέσα στο animation έχει οριστεί ένας trigger, ώστε να γίνεται αλλαγή κατάστασης του χαρακτήρα. Για την σωστή λειτουργία του παιχνιδιού και για την αποφυγή bugs έχει γίνει ένα ακόμα αυτόματο animation. Αυτό το animation προκαλεί και δεύτερη αλλαγή στην κίνηση του χαρακτήρα, την επιστροφή δηλαδή στην αρχική κατάσταση του κατάστασης. Στον animator έχει γίνει αμφίδρομη ένωση.



**Εικόνα 2.3.11 : Επιστροφή κουμπιών στην αρχική κατάσταση**

Έχει επίσης φτιαχτεί και εξαχθεί μαζί με την εφαρμογή αυτή και φάκελος «Data», σε περίπτωση που μελλοντικά θα μπορούσε να προστεθεί, με την βοήθεια της SQL, η δημιουργία λογαριασμού χρήστη.

### 3. Game manual

Η πλοήγηση του παίκτη στις διάφορες παρεχόμενες λειτουργίες του παιχνιδιού γίνονται μέσω ποντικιού. Ο παίκτης εισερχόμενος στο παιχνίδι βλέπει την πρώτη σκηνή, όπου υπάρχουν τρία κουμπιά που εξυπηρετούν διαφορετικές λειτουργίες το καθένα:

- **Play** : πατώντας το κουμπί αυτό ΔΥΟ ΦΟΡΕΣ ,ο παίκτης προχωράει στο παιχνίδι καθαυτό (στο πάτημα του κουμπιού μία φορά, απλά φορτώνει την εικόνα).
- **About** : πατώντας το κουμπί αυτό εμφανίζεται ένα παράθυρο που περιέχει συγκεκριμένες λεπτομέρειες για την προέλευση του παιχνιδιού
- **Quit** : πατώντας το κουμπί αυτό το παιχνίδι τερματίζεται

Αφού πατηθεί το κουμπί Play, τότε ο παίκτης μεταφέρεται στη δεύτερη σκηνή, δηλαδή στο ίδιο το παιχνίδι.

Δεξιά και αριστερά της οθόνης εμφανίζονται οι χαρακτήρες του παιχνιδιού σαν να αιωρούνται πάνω από τις αντίστοιχες κάρτες του. Στα αριστερά βρίσκεται ο Player 1, ο χαρακτήρας δηλαδή του οποίου οι κινήσεις ελέγχονται από τον παίκτη, και στα δεξιά ο Player 2.

Εδώ τα κουμπιά είναι περισσότερα.

Κάτω δεξιά ο παίκτης βλέπει τα παρακάτω κουμπιά:

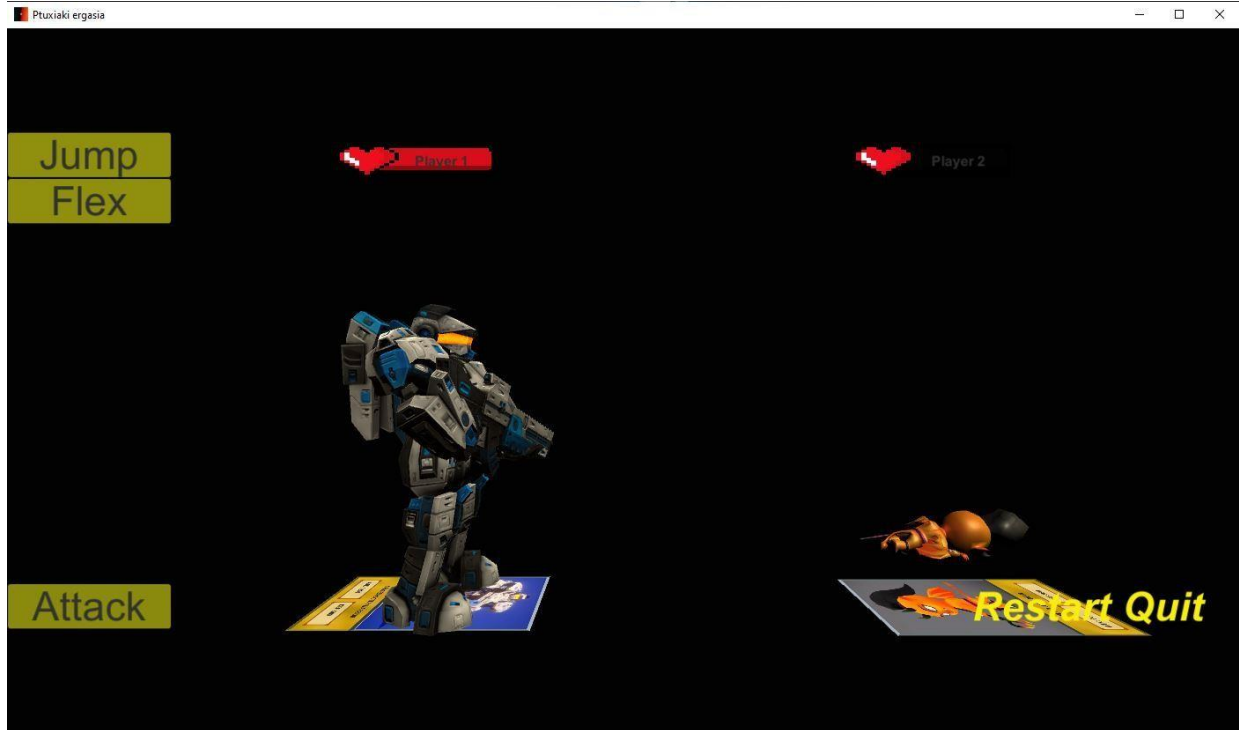
- **Quit**: πατώντας το ο παίκτης βγαίνει από το παιχνίδι
- **Restart**: πατώντας το ο παίκτης επιστρέφει στο αρχικό μενού. Αυτό επιτυγχάνεται με τον κώδικα onclick event να πάει στην προηγούμενη σκηνή

Στα αριστερά ο παίκτης βλέπει τα παρακάτω κουμπιά:

- **Jump**: πατώντας το ο χαρακτήρας του παιχνιδιού αλλάζει το motion του, την κίνηση του, και κάνει επιτόπιο άλμα.
- **Flex**: πατώντας το ο χαρακτήρας σκύβει προς την αντίθετη κατεύθυνση, αλλάζει δηλαδή την κατάστασή του, και στη συνέχεια επανέρχεται στο αρχικό του animation.
- **Attack**: πατώντας το επιτυγχάνεται η αλλαγή κατάστασης του αριστερού χαρακτήρα κάνοντας την κίνηση της επίθεσης και την άμεση επαναφορά του στην αρχική του κατάσταση/θέση. Τέλος, το κουμπί attack, έχει την μείωση του slider της μπάρας ζωής του αντίπαλου χαρακτήρα.

Πέραν των χαρακτήρων και των κουμπιών που βλέπει ο παίκτης στην σκηνή του παιχνιδιού, στο πάνω μέρος της οθόνης εντοπίζονται τα health bars των παικτών που υποδηλώνουν την κατάσταση ζωής του κάθε χαρακτήρα. Αριθμός αφετηρίας έχει οριστεί το 20. Με το κουμπί Attack που αναφέρθηκε νωρίτερα, μειώνεται ο αντίστοιχος slider. Αν η μπάρα του Player 2 μηδενιστεί, τότε αλλάζει η κατάσταση του στο animation του θανάτου. Σε αυτήν την περίπτωση, είναι αδύνατο

να επιστρέψει στην αρχική κατάσταση, διότι ο χαρακτήρας έχει πεθάνει.



Εικόνα 3.1 : Κατάσταση Player 2 μετά την επίθεση

#### 4. Scripting

Για τις λειτουργίες εναλλαγής σκηνών, τις λειτουργίες των κουμπιών και τα λοιπά, χρησιμοποιήθηκαν σύντομα και εύχρηστα σενάρια (scripts).

Τα σενάρια είναι αρχεία κατάληξης .cs, τα οποία ο προγραμματιστής επεξεργάζεται μέσω Visual Studio ή editor της επιλογής του, διαθέτουν εγγενή υποστήριξη λειτουργιών στο περιβάλλον της Unity και είναι βασικό συστατικό για τη δημιουργία μιας ολοκληρωμένης λύσης παιχνιδιού. Το scripting μπορεί να χαρακτηριστεί ως βασικό συστατικό σε όλες τις εφαρμογές που αναπτύσσονται σε Unity περιβάλλοντα. Οι περισσότερες, αν όχι όλες οι εφαρμογές εκεί, χρειάζονται σενάρια σε τομείς όπως :

- ανταπόκριση στο input του παίκτη
- καθορισμός συμβάντων στο gameplay να συμβαίνουν όταν πρέπει.
- δημιουργία γραφικών αποτελεσμάτων
- έλεγχος της φυσικής συμπεριφοράς αντικειμένων
- εφαρμογή ενός προσαρμοσμένου συστήματος AI για χαρακτήρες στο παιχνίδι.



Εικόνα 4.1 : Συλλογή Scripts στον inspector

## **5. Αποτελέσματα – Επεξήγηση επιλογών**

Ξεκινώντας το project και δουλεύοντας στην τότε τελευταία version της Unity, είχα καταφέρει να προσθέσω όλα τα μοντέλα (χαρακτήρες παιχνιδιού) -που υπάρχουν στον φάκελο «Images»- να εμφανίζονται στο παιχνίδι. Κατά την διάρκεια της υλοποίησης, αναγκάστηκα να κάνω update την Unity στη νέα version. Η Vuforia, όμως, πλέον δεν υποστηριζόταν με τον ίδιο τρόπο, και αφαίρεσε αρκετά από τα components, συγκεκριμένα αυτά των scripts, πράγμα που εμπόδιζε την εμφάνιση των μοντέλων στο παιχνίδι.

Επιθυμώντας να δημιουργήσω ένα project που να έχει άμεση ανταπόκριση αποφάσισα πως θα ήταν πιο συνετό να προβώ σε κινήσεις που θα έκαναν το παιχνίδι μου lag/bug free. Χαρακτηριστικά παραδείγματα ήταν η αποφυγή της χρήσης ήχου και εικόνας σαν background στην δεύτερη σκηνή, αυτή του παιχνιδιού. Άλλωστε, το απλό μαύρο φόντο στη συγκεκριμένη σκηνή εξυπηρετεί έναν ακόμα σκοπό, την ανάδειξη των γραφικών των χαρακτήρων με μεγαλύτερη ευκρίνεια.

## **Συμπεράσματα και μελλοντικές επεκτάσεις**

Η δημιουργία ενός παιχνιδιού σε περιβάλλον Unity είναι μια ενδιαφέρουσα διαδικασία που απαιτεί γνώσεις πάνω στον προγραμματισμό. Επομένως, για την υλοποίηση ενός project σύνθετου και stable δεν αρκεί απλώς να γνωρίζει κάποιος να χρησιμοποιεί τη συγκεκριμένη πλατφόρμα. Έτσι λοιπόν, η εισαγωγή αντίστοιχων animations ώστε να κινούνται οι χαρακτήρες του παιχνιδιού, καθώς επίσης και τα ποικίλα scripts που έχουμε εισάγει στα διάφορα κουμπιά έχουν διαδραματίσει σημαντικό ρόλο προκειμένου να επιτευχθεί η δημιουργία ενός εκτελέσιμου παιχνιδιού, που ήταν και το ζητούμενο.

Σίγουρα το συγκεκριμένο project είναι σε πρωταρχικό στάδιο και αναμφίβολα επιδέχεται βελτιώσεις αλλά και επεκτάσεις. Μια από τις μελλοντικές επεκτάσεις του παιχνιδιού θα μπορούσε να είναι η εναλλαγή των απλών κουμπιών του παιχνιδιού σε διαφορετικές επιθέσεις, ή ακόμα και ο εμπλουτισμός των ήδη υπαρχόντων κουμπιών.

Όπως προαναφέρθηκε, αρχικός στόχος ήταν να ανοίγει η κάμερα του υπολογιστή/κινητού και να εμφανίζονταν στην αντίστοιχη οθόνη μόνο τα κουμπιά με κίτρινο χρώμα ή πλαίσιο. Μελλοντικά, θα μπορούσε να γίνεται και η εισαγωγή και δεύτερου παίκτη μέσω της επίδειξης αντίστοιχης κάρτας. Στην αριστερή κάρτα θα εμφανιζόταν από πάνω η μπάρα ζωής με την ονομασία «Player 1», και στην περίπτωση που η κάμερα θα εντόπιζε και δεύτερη ταυτόχρονη κάρτα, θα εμφανιζόταν και ο δεύτερος χαρακτήρας με την διαφορά ότι η μπάρα ζωής του δεύτερου (δεξιά) θα είχε την ονομασία «Player 2».

Επίσης, μέσω ενός UI element και συγκεκριμένα ενός switch, ο χρήστης θα μπορεί να είναι σε θέση να επιλέγει με ποιον χαρακτήρα θα θέλει να επιλέξει για να παίξει, ποιος χαρακτήρας δηλαδή θα βρίσκεται στο αριστερό μέρος της οθόνης της συσκευής, και σε αυτήν την περίπτωση θα διαφοροποιούνται οι επιθέσεις/τα ξόρκια που ο κάθε χαρακτήρας μπορεί να κάνει. Ιδανικά, αν η Vuforia εκδώσει update που να υποστηρίζει πλήρως την νέα version της Unity, δεν θα χρειάζεται να επιλέγει ο παίκτης χαρακτήρα μέσω του παιχνιδιού –θα αρκεί απλώς να επιδείξει στην κάμερα την κάρτα του χαρακτήρα που εκείνος επιθυμεί.

## System Full Specifications

**Case:** 15.6" TUF Gaming FX504GD Laptop from ASUS

**CPU:** 2.3 GHz Intel Core i7-8750 @2.2 (9M Cache, up to 4.1 Ghz) Hexa-Core

**MOBO:** Asus TUF FX504GD-E4021T, Intel HM370

**Display:** 15.60 inch 16:9, 1920 x 1080 pixel 141 PPI, AU Optonics B156HTN03.8, TN LED, ID: AUO38ED, glossy: no

**GPU:** NVIDIA GeForce GTX 1050 Mobile - 4096 MB, Core: 1354 MHz, Memory: 7008 MHz, GDDR5, 388.73, Optimus

**RAM:** Vengeance 16GB DDR4 2666MHz CL 8

**SSD:** Samsung 970 Evo Plus 124 GB NVMe SSD

**Storage:** Seagate Mobile HDD 1TB ST1000LX015-1U7172, 1027 GB , 5400 rpm

**Weight:** 2.3 kg \ Power Supply: 505 g

**Screen:** AOC C24G1 Curved Gaming Monitor 24" FHD 144Hz

**Mouse:** SteelSeries Rival 310 RGB

**Keyboard:** Motospeed Inflictor CK104 RGB Silver (Outemu Blue) GR



## **Βιβλιογραφία**

<https://unity.com/solutions/game>

<https://developer.vuforia.com/>

<https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy/devil-animated-character-60777>

<https://assetstore.unity.com/packages/3d/characters/robots/sci-fi-warrior-pbr-hp-polyart-106154>

<https://assetstore.unity.com/packages/templates/packs/vuforia-engine-163598>

[https://www.bhphotovideo.com/c/product/1398355-REG/asus\\_fx504gd\\_rs51\\_15\\_6\\_tuf\\_gaming\\_fx504gd.html](https://www.bhphotovideo.com/c/product/1398355-REG/asus_fx504gd_rs51_15_6_tuf_gaming_fx504gd.html)

