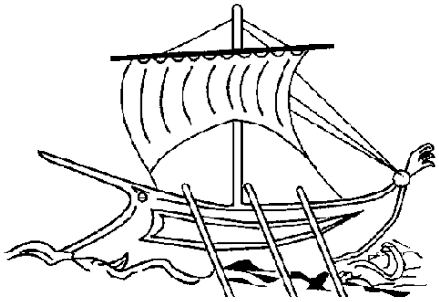


ΑΕΙ ΠΕΙΡΑΙΑ Τ.Τ.
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ Τ.Ε.

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη Παιχνιδιού με το Unity3D σε C#

Κωνσταντίνος Ανδρεαδάκης
Ιωάννης Εξιλζές

Εισηγητής: Δρ Πρεζεράκος Γεώργιος, Καθηγητής

ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2015

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδίαση ολοκληρωμένου συστήματος συγγραφής διπλωματικής εργασίας

Κωνσταντίνος Ανδρεαδάκης
ΑΜ : 42069
Ιωάννης Εξιλζές
ΑΜ : 42466

Εισηγητής:

Δρ Πρεζεράκος Γεώργιος, Καθηγητής

Εξεταστική Επιτροπή:

Ημερομηνία εξέτασης /7/2015

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε σε μεγάλο χρονικό διάστημα, σε ένα όχι και τόσο γνωστό αλλά ενδιαφέρον αντικείμενο, όπως αυτό της ανάπτυξης παιχνιδιών. Την προσπάθειά μας αυτή υποστήριξε ο επιβλέπων καθηγητής μας, τον οποίο θα θέλαμε να ευχαριστήσουμε. Επιπλέον, τους διαδικτυακούς προγραμματιστές μέσω YouTube και σε διάφορα forum που βοήθησαν στην εκμάθηση του προγράμματος που χρησιμοποιήσαμε, αλλά και στον προγραμματισμό της σωστής λειτουργίας του παιχνιδιού.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη παιχνιδιού με την βοήθεια της μηχανής Unity3D. Η βιομηχανία των παιχνιδιών με τα Game Engine έχει δώσει την δυνατότητα στους νέους προγραμματιστές να τις χρησιμοποιούν και να εργάζονται εύκολα ένα μεγάλο μέρος από αυτούς. Η παρακάτω πτυχιακή κάνει μια αναφορά στην Ιστορία των Game Engines αλλά και ειδικότερα στο Unity3D ,με σκοπό την δημιουργία ενός παιχνιδιού 3D από το μηδέν.

ABSTRACT

The present thesis concerns the game development procedure based on Unity 3D game engine. The game industry, nowadays, gives the opportunity to upcoming game developers to design and develop games with free tools that even high tech companies use. The following thesis refers to the history of games and in particular to Unity 3D, in order to develop our own video game from scratch.

ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: Αρχιτεκτονική Ηλεκτρονικών Υπολογιστών
ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Unity, game, engine, scripting, βιομηχανία παιχνίδιού

Περιεχόμενα

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Κεφάλαιο 1^{ο} : Εισαγωγή

1.1 Ιστορική αναδρομή

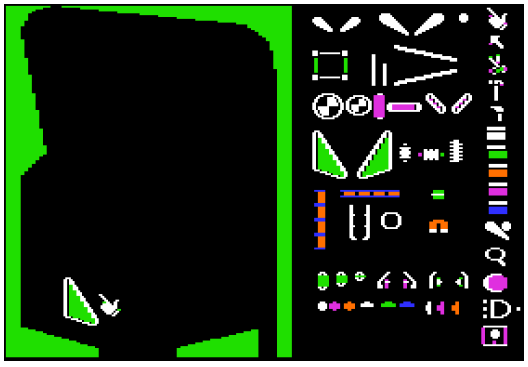
Η ιστορία των Game Engines ξεκινά με την δημιουργία των πρώτων παιχνιδιών ,όπου η κάθε εταιρεία είχε και την δικιά της μηχανή ανάπτυξης. Στις μέρες μας σίγουρα πολλές εταιρείες ακόμα ακολουθούν αυτό το πρότυπο αλλά πλέον, υπάρχουν διαθέσιμα δωρεάν προγράμματα ανάπτυξης παιχνιδιών(Unity,Unreal Engine κλπ) ,καθώς και ενημερωτικά βίντεο(tutorials) όπου ο καθένας μπορεί να έχει στην διάθεσή του.

Οι μηχανές παιχνιδιών κάνουν την εμφάνισή τους στις αρχές της δεκαετίας του '90 με το ξεκίνημα των 3d γραφικών, όπου απογείωσε την βιομηχανία των ηλεκτρονικών παιχνιδιών. Στα μέσα της σημαντικής αυτής δεκαετίας καθιερώθηκε ο όρος game engine ,με τα παιχνίδια "Doom" και "Quake" να κάνουν την επανάσταση στο game developing. Μετά την επιτυχία των δύο αυτών παιχνιδιών διάφοροι developers άρχισαν και αγόραζαν βασικά κομμάτια και άρχισαν να προσθέτουν δικά τους αντικείμενα στο παιχνίδι όπως όπλα και διάφορα γραφικά στοιχεία. Έτσι ο καθένας άρχιζε να αναπτύσσει το βασικό πρόγραμμα για όποια πλατφόρμα ήθελε και μπορούσε ακόμα να τα πουλήσει σαν ξεχωριστά video games.

Αυτό έγινε πραγματικότητα το 1998 με την Unreal Engine, η οποία προμήθευε την Epic Games ,όμως οι μηχανές αυτές ακόμα ήταν άγνωστες για τους developers καθώς οι μηχανές γραφικών χρησιμοποιούνταν ήδη από την δεκαετία του '80, αλλά μόνο για 2D. Οι μηχανές όμως αυτές δεν έδιναν ελευθερία στον προγραμματιστή ,αφού διατηρούσαν κλειδωμένο τον βασικό μηχανισμό.

Μερικές από τις πρώτες Μηχανές :

Pinball Construction Set (1983)
ASCII's War Game Construction Kit (1983)
Adventure Construction Set (1984)
Shooter-Up Construction Kit (1987)



Εικόνα 1. Pinball Construction Set (1983)

1.2 Μηχανές Ανάπτυξης Παιχνιδιών

Εικόνα 2 Πιο γνωστές μηχανές ανάπτυξης

Η τεχνολογία των βιντεοπαιχνιδιών έχει αναπτυχθεί πάρα πολύ με αποτέλεσμα οι μηχανές ανάπτυξης παιχνιδιών σήμερα αριθμούνται πάνω από 400!, αυτό δεν σημαίνει όμως ότι με όλες τις μηχανές σου δίνουν τα ίδια δικαιώματα και ότι έχεις τις ίδιες δυνατότητες. Κάθε μηχανή έχει τα πλεονεκτήματα και τα μειονεκτήματα της. Οι βασικές μηχανές σήμερα που χρησιμοποιούν οι περισσότεροι είναι :

- Unity3D
- UDK - Unreal Engine
- Cryengine

Εικόνα 3 Λογότυπο Unity

1.3 Γιατί Unity ;

Όπως είπαμε και στην αρχή κάθε μηχανή ανάπτυξης έχει τις ευκολίες της και τις δυσκολίες της, έχει το κοινό της και την υποστήριξη ως προς τον Developer.

Εμείς καταλήξαμε στο Unity3d επειδή έχει έκδοση Free, έχει ένα Forum-Community υποστήριξης forum.unity3d.com όπου μπορείς να βρεις λύση σε ότι δυσκολία συναντήσεις, και τέλος έχει ένα Asset- Store όπου σου παρέχει μερικά γραφικά μοντέλα - scripting code δωρεάν.

Κεφάλαιο 2^{ο} : Προγράμματα &Γραφικά στοιχεία

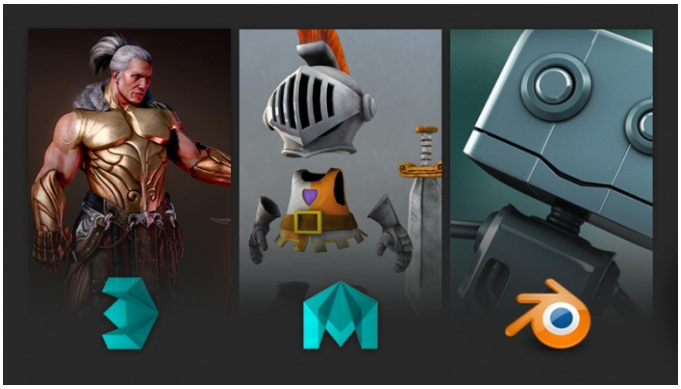
2.1 Πρόλογος

Μια εφαρμογή για να υλοποιηθεί χρειάζεται να χρησιμοποιήσουμε περισσότερα από ένα εργαλείο για να έχουμε σωστό και ικανοποιητικό αποτέλεσμα. Μερικά από αυτά είναι Photoshop - Google Sketchup - Blender - Audacity κ.α , γι' αυτό και τα μεγάλα παιχνίδια δημιουργούνται από εταιρείες όπου οι προγραμματιστές ποικίλουν σε αριθμό και γνώσεις ώστε ο καθένας να είναι υπεύθυνος για ένα συγκεκριμένο κομμάτι του παιχνιδιού όπως τμήμα γραφικών, animation, sound, programming, video editing κλπ.

2.2 Προγράμματα 3d Modeling

Ένα βασικό στοιχείο για την επιτυχία ενός παιχνιδιού είναι τα γραφικά και τα 3d μοντέλα που χρησιμοποιεί το παιχνίδι. Όπως και με τις μηχανές ανάπτυξης έτσι και εδώ συναντάμε πολλά προγράμματα για 3d modeling. Μερικά από τα πιο γνωστά είναι:

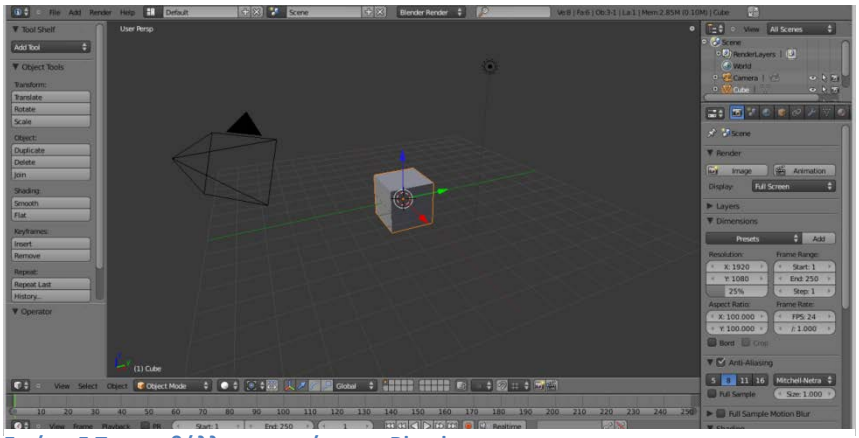
- 3ds max - Autodesk
- Maya
- Blender



Για την εφαρμογή μας θα χρησιμοποιήσουμε το Blender Engine 2.69

2.3 Blender 3d Engine

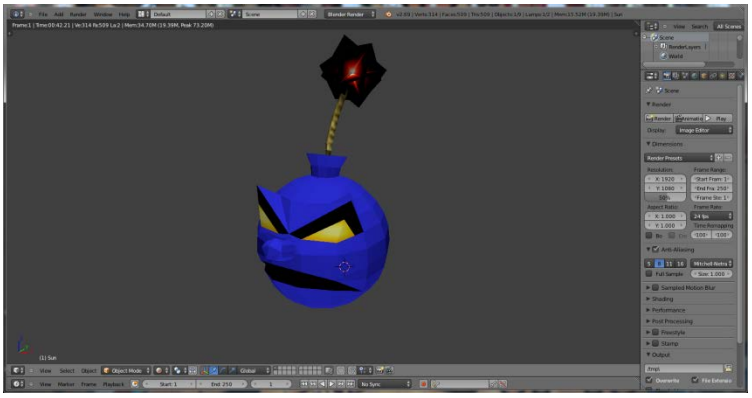
Το Blender 3d είναι ένα Open Source πρόγραμμα σχεδίασης 3d μοντέλων. Χρησιμοποιείται συνήθως για modeling , rigging , animation. Είναι μικρό σε μέγεθος, τρέχει σε όλες τις κύριες πλατφόρμες και υποστηρίζει γλώσσα προγραμματισμού python. Είναι λίγο δύσκολο στην πρώτη όψη αλλά κάθε εντολή είναι μια συντόμευση και αν το μάθεις μπορείς εύκολα και γρήγορα να φτιάξεις οτιδήποτε μπορείς να φανταστείς.



Εικόνα 5 Το περιβάλλον εργασίας του Blender

2.4 Διαμόρφωση για τα στοιχεία της Πτυχιακής

Για την πτυχιακή χρησιμοποιήσαμε το Blender για να φτιάξουμε αντικείμενα όπως τα βράχια του παιχνιδιού αλλά και το βασικό μας χαρακτήρα. Το χαρακτήρα τον πήραμε σαν 3d model έτοιμο από ένα παλιό παιχνίδι (Pacman στο GameCube - Pac-Man World Rally) και χρησιμοποιήσαμε τον χαρακτήρα Bomb του παιχνιδιού και τον διαμορφώσαμε μέσω του Blender.



Εικόνα 6 Χαρακτήρας - Player Blue

Κεφάλαιο 3ο : Unity 3d Game Engine

3.1 Πρόλογος

Είναι για πολλούς το καλύτερο Game Engine διότι είναι μια ολοκληρωμένη μηχανή ανάπτυξης παιχνιδιών, έχει ένα μεγάλο Forum και ένα Fan Club υποστηρικτών όπου μπορούν να σε βοηθήσουν πολύ γρήγορα. Ακόμα είναι η πιο διαδεδομένη μηχανή ανάπτυξης σε ότι αφορά δημιουργία παιχνιδιών σε Smart phones & Tablet, έχει μέτρια προς καλά γραφικά και το ατού είναι τα Script που τρέχουν πιο γρήγορα από άλλες μηχανές.

3.2 Ιστορική Αναδρομή

Η ιδέα του Unity ξεκίνησε το 2002 από ένα Post του Δανού Nicholas Francis όπου ρωτούσε για το ποιός ήθελε να φτιάξει μαζί του ένα Game Engine, λίγες ώρες αργότερα ο Joachim Ante απάντησε στο ερώτημα και έτσι έγινε η αρχή, τελικά οι προγραμματιστές έγιναν τρεις αφού μπήκε στο γκρούπ και ο Devid Helgason.

Η εταιρία τελικά ιδρύθηκε το 2004 στη Δανία από τους τρεις αυτούς προγραμματιστές. Η βασική επιτυχία της μηχανής αυτή στηρίζεται στο γεγονός ότι βοηθάει τους ανεξάρτητους Game Developers οι οποίοι δεν είναι σε θέση να δημιουργήσουν την δικιά τους μηχανή για να φτιάξουν το δικό τους παιχνίδι. Το μεγάλο "Μπαμ" έγινε όταν ήρθε στην επιφάνεια το Iphone και το appstore , καθώς η μηχανή ήταν ήδη έτοιμη για την συγκεκριμένη πλατφόρμα. Σύμφωνα με μια έρευνα το Unity χρησιμοποιείται πάνω από το 50% των προγραμματιστών που ασχολούνται με την δημιουργία παιχνιδιών σε iOS και Android.

Εικόνα 7 Mobile Game με το Unity

Τέλος το Unity έχει βγάλει μέχρι στιγμής 5 Version με τις τελευταίες να ανανεώνονται σχετικά συχνά λόγω της αυξανόμενης τάσης των Smartphone και των Tablets. Αξίζει να σημειωθεί ότι από το 2009 έχει μια δωρεάν έκδοση και μια pro όπου σου προσφέρει περισσότερα assets και δυνατότητες.

3.3 Το εργαλείο σήμερα

Το Unity σήμερα βρίσκεται σήμερα στην έκδοση 5 και οι Developers που χρησιμοποιούν το Unity φτάνει τα δύο εκατομμύρια από τους οποίους οι τριακόσιοι χιλιάδες το χρησιμοποιούν σε μηνιαία βάση.
Το Unity 5 παρουσιάστηκε τον Μάρτιο του 2015 με σημαντικές αλλαγές στα γραφικά αλλά και στον καθαρό ήχο του παιχνιδιού.

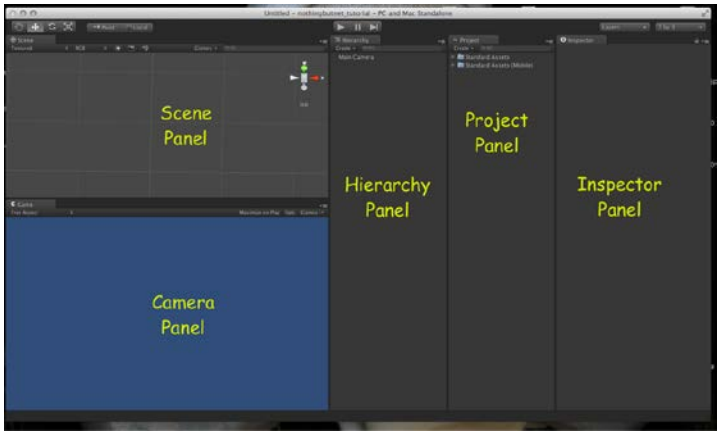
3.4 Τεχνολογία Γραφικών

Το Unity 5 έκανε μεγάλη αναβάθμιση στον τομέα των γραφικών αφού βελτίωσε τους Shaders και επέτρεψε την ενσωμάτωση του Enlighten ,μιας τεχνολογίας φωτισμού σε πραγματικό χρόνο.
Ένα αρκετά μεγάλο κομμάτι μοντέρνων παιχνιδιών χρησιμοποιούν την μέθοδο rendering Physically Based Rendering (PBR) όπου σε αυτή την κατηγορία μπήκε και το Unity 5 προσθέτοντάς το, στον κλάδο των γραφικών Next-Gen Texturing, όπως ονομάζεται.

3.5 Unity Editor

Ο Editor του Unity αποτελείτε από διάφορα Panel τα οποία κρατάνε το Project και εύκολα διαχωρίσιμο. Ο χρήστης έχει την δυνατότητα να δημιουργήσει το δικό του Interface καθώς έχει την δυνατότητα να μεταφέρει και να επιλέξει να φαίνεται ότι αυτός θέλει. Τα σημαντικότερα από αυτά και που θα αναλύσουμε στη συνέχεια είναι :

- Scene View
- Game View
- Hierarchy
- Project
- Inspector



Εικόνα 9 Interface Unity3D editor

3.5.1 Scene View

Η “σκηνή” είναι ο χώρος κατασκευής του παιχνιδιού όπου ο χρήστης μπορεί να πλοηγηθεί και να επεξεργαστεί τον χώρο αυτό μέσο των πλήκτρων Q, W, E, R.

Πλήκτρο Q(Navigation Tool): 

Μέσω του πλήκτρου αυτού που αντιπροσωπεύει το χεράκι στην οθόνη μας πλοηγούμαστε στον χώρο. Συγκεκριμένα, κρατώντας πατημένο το αριστερό κλικ του ποντικιού μετακινούμε την κάμερα αριστερά, δεξιά, πάνω και κάτω. Πιέζοντας ταυτόχρονα και το πλήκτρο alt μπορούμε να κάνουμε περιστροφή γύρω από το αντικείμενο που έχουμε εστιάσει. Πιέζοντας αυτή την φορά το δεξί κλικ του ποντικιού περιστρέφουμε την κάμερα προς την επιλεγμένη διεύθυνση που του δίνουμε(αριστερά, δεξιά, πάνω, κάτω)ενώ πατώντας ταυτόχρονα και το alt κάνουμε ζουμ.

Πλήκτρο W(Translate Tool): 

Χρησιμοποιείται όταν έχουμε εστιάσει σένα αντικείμενο που έχουμε τοποθετήσει και μας επιτρέπει να το μετακινήσουμε προς την διεύθυνση που δείχνουν τα βελάκια πάνω στο αντικείμενο σύμφωνα με τους άξονες x, y, z.

Πλήκτρο E(Rotation Tool): 

Μας επιτρέπει να περιστρέψουμε ένα αντικείμενο επιλέγοντας κάθε φορά έναν άξονα περιστροφής.

Πλήκτρο R(Scale Tool): 

Τέλος, με το πλήκτρο R μπορούμε να ρυθμίσουμε τις διαστάσεις για το αντικείμενό μας σε κάθε άξονα ξεχωριστά αλλά και να μεγεθύνουμε ή σμικρύνουμε αναλογικά το αντικείμενο πατώντας στο κέντρο του.

Rendering Options:

Ακριβώς κάτω από το παράθυρο Scene βλέπουμε την μέθοδο προβολής(Textured) των αντικειμένων όπου πρόκειται για ένα drop down menu που μας έχει επιπλέον επιλογές(Wireframe, Tex, Render Paths, Lightmap resolution και Light Probs) και επιλέγουμε την επιθυμητή λειτουργία.

Εικόνα 10 Παράδειγμα wireframe

3.5.2 Game View

Το Game view αναπαριστά το παιχνίδι στην τελική μορφή του, αυτό γίνεται με την χρήση του κουμπιού "Play" που βρίσκεται στην γραμμή εργαλείων μαζί με τα κουμπιά "Pause" και "Next Frame" του Unity. Συγκεκριμένα :

Play : 

Με το κουμπί αυτό μπορείς να δοκιμάσεις-τρέξεις το παιχνίδι και να πάρεις μια ιδέα για την τελική μορφή του.

Pause : 

Με το κουμπί αυτό παγώνουμε το παιχνίδι σε ένα συγκεκριμένο Frame ώστε να μπορέσουμε να διακρίνουμε μια λεπτομέρεια, καθώς και να επεξεργαστούμε τιμές που επηρεάζουν την λειτουργία του (ταχύτητα χαρακτήρα-βαρύτητα-φωτισμός κ.α)

Next Frame : 

Με το κουμπί αυτό μπορούμε να δούμε με ακρίβεια τα κάθε Frame του παιχνιδιού και να τα επεξεργαστούμε.

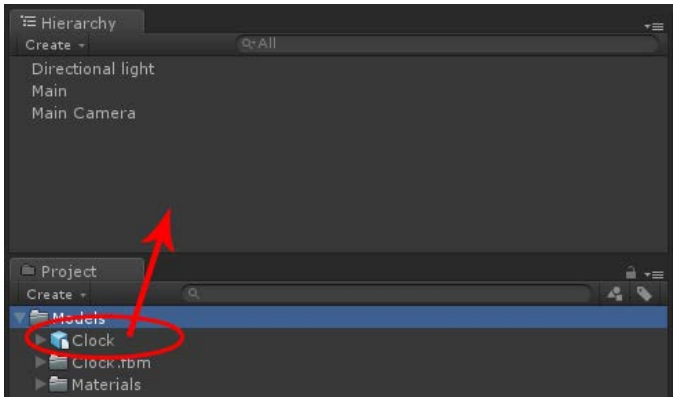
Στην γραμμή εργαλείων του game view βλέπουμε αρχικά από αριστερά ένα drop down menu το οποίο μας επιτρέπει να κάνουμε προεπισκόπηση του παιχνιδιού στην επιθυμητή ανάλυση που του ορίζουμε. Στη συνέχεια προς τα δεξιά βλέπουμε το παραθυράκι "maximize on play" με το οποίο όταν το έχουμε επιλεγμένο και πατήσουμε "Play" το παιχνίδι μας παίζει σε πλήρη οθόνη ακόμα υπάρχει και η επιλογή "Stats" όπου σου δείχνει στατιστικά γραφικών.

Εικόνα 11 Game View

3.5.3 Hierarchy and Project

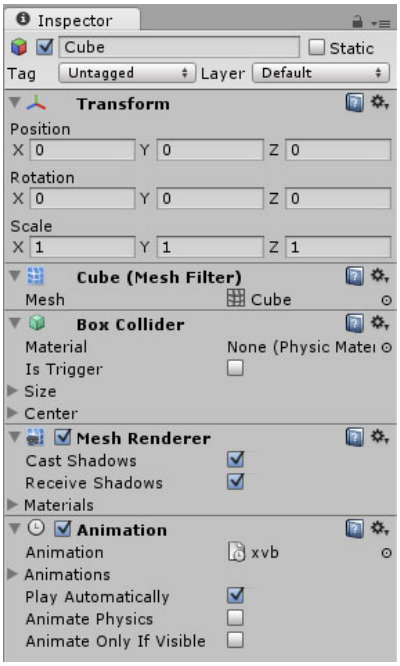
Το Hierarchy είναι ένα παράθυρο όπου περιέχει όλα τα αντικείμενα που βρίσκονται στην σκηνή με τα ονόματα τους και από εκεί μπορούμε να αλλάξουμε τις τελικές ρυθμίσεις του, με λίγα λόγια το παιχνίδι μας παίρνει πληροφορίες από τον συγκεκριμένο φάκελο και όχι από το παράθυρο project που θα αναλύσουμε παρακάτω.

Το παράθυρο Project από την άλλη περιέχει όλα τα Asset του παιχνιδιού με απλά λόγια οτιδήποτε θα χρειαστούμε στο παιχνίδι είναι εκεί (textures - scripts - ήχοι - prefabs κ.α)



Εικόνα 12 Εισαγωγή Prefab από το Project στο Hierarchy

3.5.4 Inspector



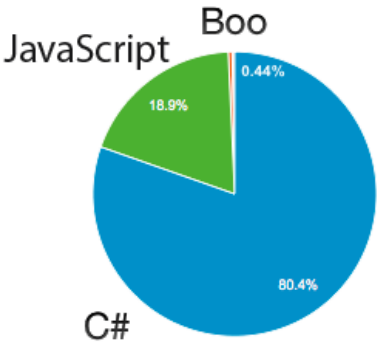
Στον Inspector εμφανίζονται όλα τα χαρακτηριστικά των αντικειμένων μας όλου του παιχνιδιού είτε αυτά βρίσκονται στο Hierarchy είτε στο Project .Από αυτό το παράθυρο μπορούμε να επεξεργαστούμε τα χαρακτηριστικά του ανάλογα με το είδος του. Βασικά χαρακτηριστικά είναι η θέση - position, η περιστροφή-rotation αλλά και το μέγεθος-scale.

Επιπλέον μπορούμε να του εισάγουμε νέα χαρακτηριστικά στο αντικείμενο μας όπως είναι ένα ήχος, φωτισμός, βαρύτητα,animations, scripts και άλλα που επηρεάζουν στην αντίδραση του Object μας.

3.6 Programming

Το Unity έχει ενσωματωμένο για δημιουργία και επεξεργασία Script το MonoDevelop. Είναι ένα πρόγραμμα ελαφρύ με το οποίο μπορούμε να γράψουμε στις τρεις αντικειμενοστραφείς γλώσσες που υποστηρίζει το Unity. Αυτές είναι :

- C#
- JavaScript
- Boo



Η πιο γνωστή και η πιο διαδεδομένη γλώσσα είναι η C# όπου και θα ασχοληθούμε. Το διπλανό διάγραμμα δείχνει ξεκάθαρα τα στατιστικά χρήσης των παραπάνω γλωσσών.

3.7 Networking

Το Unity υποστηρίζεις αρκετούς τρόπους δικτύωσης, τα δύο κυριότερα είδη διαδικτυακής επικοινωνίας είναι :

- State Synchronization

Το δίκτυο παρακολουθεί τα αντικείμενα και εντοπίζει τις αλλαγές που γίνονται μέσα στο χώρο. Αυτές οι αλλαγές έπειτα μοιράζονται στους χρήστες στο ίδιο δίκτυο και διασφαλίζεται οτι οι αλλαγές έχουν αντιληφθεί από όλους.

- Remote Procedure Controls (RPC)

Υπάρχουν μερικές περιπτώσεις όπου δεν είναι επιθυμητό να συγχρονίζετε κάποια κατάσταση μεταξύ των Clients για αντικείμενα τα οποία δεν αλλάζει κάποια θέση, όπως πχ. γραφικά στοιχεία.

Υπάρχουν διάφοροι τρόποι διασύνδεσης μεταξύ υπολογιστών, ο βασικότερος είναι με Master Server όπου είναι και ο φυσικός τρόπος σύνδεσης. Το Unity προσφέρει ένα Master server για testing αλλά επειδή πέφτει συχνά ένας άλλος τρόπος είναι το Lan-τοπικό ή μέσω του προγράμματος Hamachi.

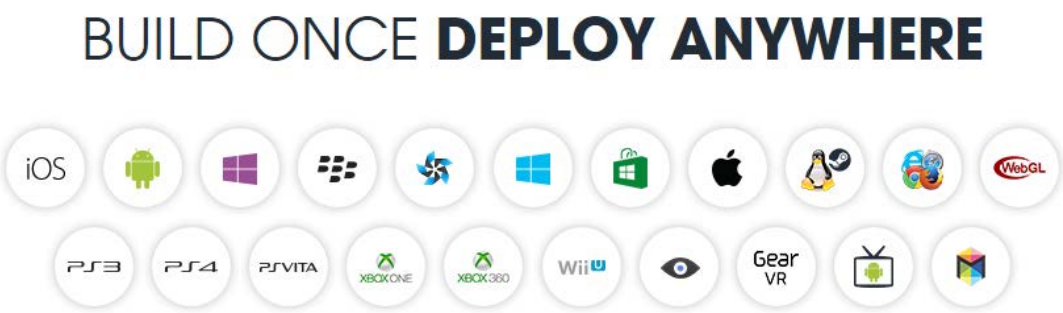
Στο Unity Asset-Store υπάρχουν έτοιμες πλατφόρμες διασύνδεσης που σου παρέχουν Server, μάλιστα μερικές από αυτές είναι και δωρεάν μέχρι κάποιο

σημείο. Ένα απλό παράδειγμα είναι το PUN όπου η έκδοση Desktop είναι free ενώ η mobile είναι επι-πληρωμής.

Τέλος αξίζει να σημειωθεί οτι το Unity με την έκδοση 5 δημιούργησε ένα πιο απλό σύστημα διασύνδεσης το Unet όπως το ονόμασε.

3.8 Publishing

Ένα από τα βασικά κριτήρια επιτυχίας του Unity είναι η δυνατότητα που δίνει στον Developer να κάνει build το παιχνίδι του σε οποιαδήποτε πλατφόρμα. Τελευταία μάλιστα το Unity 5 υποστηρίζει μέχρι και 21 διαφορετικές πλατφόρμες.



Εικόνα 13 Πλατφόρμες Build Unity5

Ακόμα από την επιλογή Project Settings μπορούμε να επηρεάσουμε τα γραφικά που θέλουμε αλλά και τις διαστάσεις που θα τρέχει αν μιλάμε για web player.Σημαντικό θετικό για τις εφαρμογές web είναι οτι το Unity έχει δικό του Web Player με πάνω από 60 εκατομμύρια εγκατεστημένους.

Τέλος να αναφέρουμε οτι η εφαρμογή μας θα υλοποιηθεί σε PC Standalone - Windows αλλά και σε Web player.

Εικόνα 14 Unity Web Player Install

Κεφάλαιο 4ο : Ανάπτυξη "Bombs land"

4.1 Πρόλογος - Περίληψη παιχνιδιού

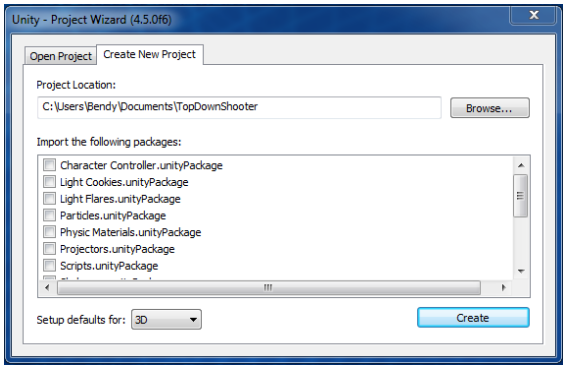
Αρχικά να αναφέρουμε οτι στο κεφάλαιο αυτό θα ασχοληθούμε κυρίως με το πρακτικό κομμάτι της εφαρμογής.

Γενικά να πούμε οτι στην αρχή δεν είχαμε κάτι στο μυαλό μας αλλά επειδή σχεδόν όλα τα παιχνίδια σήμερα είναι Multiplayer και για να δούμε σε πιο βάθος το Network του Unity, αποφασίσαμε να φτιάξουμε το "Bombs Land" - όπως το ονομάσαμε- όπου με λίγα λόγια είναι ένα First Person Shouter Game - multiplayer μέσω του προγράμματος hamachi.

Εικόνα 15 Bombs Land Logo

4.2 Πρώτη Επαφή με το Unity με την πτυχιακή

Όπως όλα τα προγράμματα έτσι και το Unity όταν ανοίγει επιλέγεις τι θα κάνεις δηλαδή New ή Open Project και επιπλέον μπορούμε να κάνουμε Import μερικά από τα packages που μας προσφέρει το Unity στην δική μας περίπτωση χρειαστήκαμε το Character.Controller όπου έχεις την δυνατότητα να ελέγχεις τον χαρακτήρα σου.



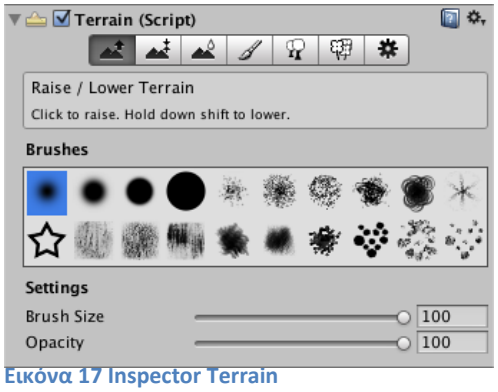
Εικόνα 16 Πρώτη επαφή με το Unity

Στο κάτω μέρος του μας έχει μια επιλογή για 2D ή 3D. Τέλος ανοίγει ο Editor του Unity με τα panel που αναφέραμε και τα διαμορφώνουμε όπως μας βολεύει.

Στην κατηγορία Project βλέπουμε το φάκελο Asset εκεί θα πρέπει να έχουμε όλα τα αρχεία μας που θα χρειαστούμε για την εφαρμογή μας κάνοντας τα import ή δημιουργώντας τα μέσα από το unity (πχ. ήχοι , textures, prefabs κ.α)

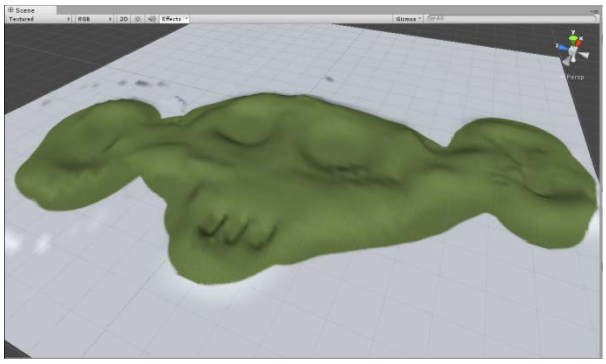
4.3 Διαμόρφωση Terrain

Το Unity μας δίνει την δυνατότητα με απλές κινήσεις να φτιάξουμε εύκολα και γρήγορα το Terrain όπως εμείς το θέλουμε και το φανταζόμαστε. Για να δημιουργήσουμε Terrain δεν κάνουμε τίποτα άλλο παρά να πάμε από την επιλογή Game Object - 3d Object -Terrain ,και αμέσως μας δημιουργεί ένα αντικείμενο Terrain εμφανίζοντας το μας και στο panel Hierarchy. Όταν το επιλέξουμε παρατηρούμε στον Inspector μερικές επιλογές όπως είχαμε πει όλα τα Object έχουν ένα βασικό Transform αλλά και επιπλέον το καθένα τα δικά ανάλογα με το τι κάνει και στο τι θέλουμε εμείς να κάνει. Το Object Terrain είναι από τα Default της Unity και έχει ενσωματωμένο ένα Script όπου από εκεί θα βρούμε τα εργαλεία και θα κάνουμε πιο εύκολο τον σχεδιασμό του Map.

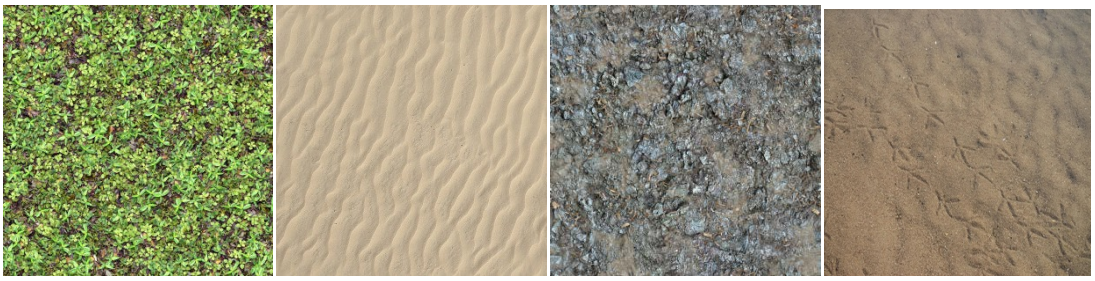


Εικόνα 17 Inspector Terrain

Με τις επιλογές που μας δίνει σχεδιάσαμε τον χάρτη όπου στην δική μας περίπτωση είναι ένα νησί σε σχήμα πειρατικής σημαίας. Η παρακάτω εικόνα αποτελεί ουσιαστικά είναι ένα Plane όπου το διαμορφώσαμε κατάλληλα για να πάρουμε το αποτέλεσμα που θέλαμε.


Εικόνα 18 1η διαμόρφωση Terrain

Στη συνέχεια για να έχουμε το τελικό αποτέλεσμα που είχαμε σχεδιάσει χρησιμοποιήσαμε μερικά textures.

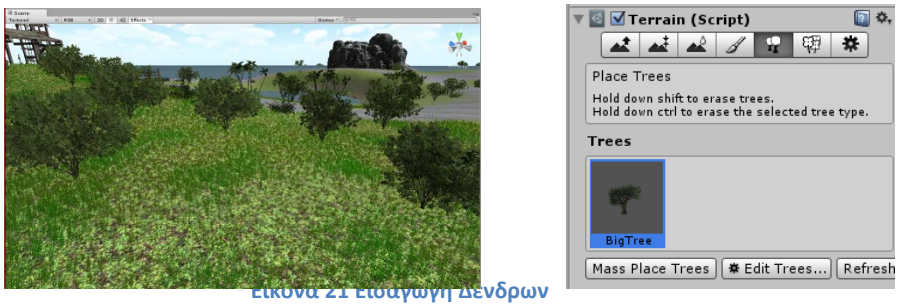


Εικόνα 19 Μερικά Textures από το παιχνίδι

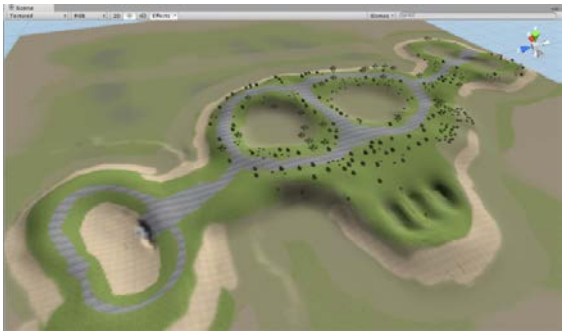
Για το γρασίδι χρησιμοποιήσαμε μια png εικόνα όπου μέσω μιας επιλογής του Unity μπορεί και το προσαρμόζει και φαίνεται κανονικά σαν 3d.

Εικόνα 20 Εισαγωγή Γρασιδιού

Και τέλος για τα δέντρα του παιχνιδιού χρησιμοποιήσαμε από τα έτοιμα που μας δίνει το unity.



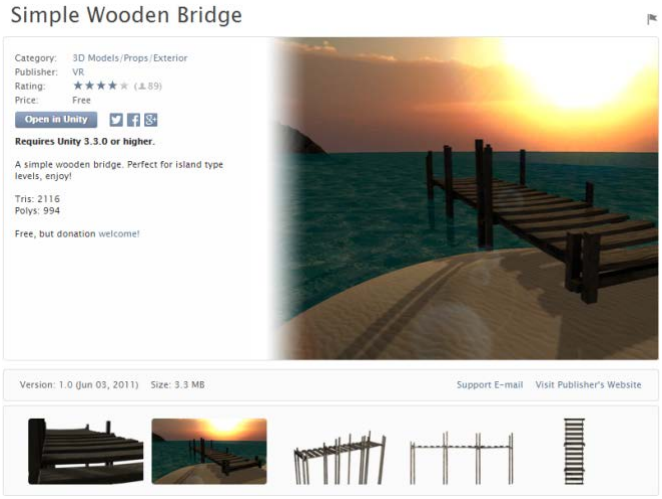
Εικόνα 21 Εισαγωγή Δένδρων



Εικόνα 22 Τελική μορφή Χάρτη

4.4 Εισαγωγή Prefabs

Όταν λέμε Prefabs εννοούμε όλα τα αντικείμενα που χρησιμοποιήσαμε και τοποθετήσαμε κατάλληλα στο παιχνίδι. Τα prefabs στην δική μας περίπτωση τα βρήκαμε από το Asset Store του Unity όπου εκεί βρίσκονται χιλιάδες assets όπου μπορείς να βρεις ότι χρειάζεσαι (www.assetstore.unity3d.com). Σίγουρα όμως μπορείς να δημιουργήσεις και το δικό σου prefab, που μπορεί να είναι το οτιδήποτε και να του δώσεις με κατάλληλα script διάφορες ενέργειες, ένα prefab μπορεί να είναι και η σφαίρα του παιχνιδιού που θα αναλύσουμε παρακάτω.



Εικόνα 23 ένα από τα Unity Assets που χρησιμοποιήσαμε

4.5 Programming - Scripting

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο με τον προγραμματισμό στο unity , μπορούμε να γράψουμε σε C#, JavaScript και boo. Η δική μας πτυχιακή είναι εξολοκλήρου C# εκτός από τον Character Controller που είναι σε JS όπου το χρησιμοποιήσαμε όπως μας το έδινε το Unity.

Πριν ξεκινήσουμε να δείχνουμε κομμάτια κώδικα από την εφαρμογή θα αναφέρουμε λίγο τα βασικά χαρακτηριστικά ενός script, και συγκεκριμένα στις Function του.

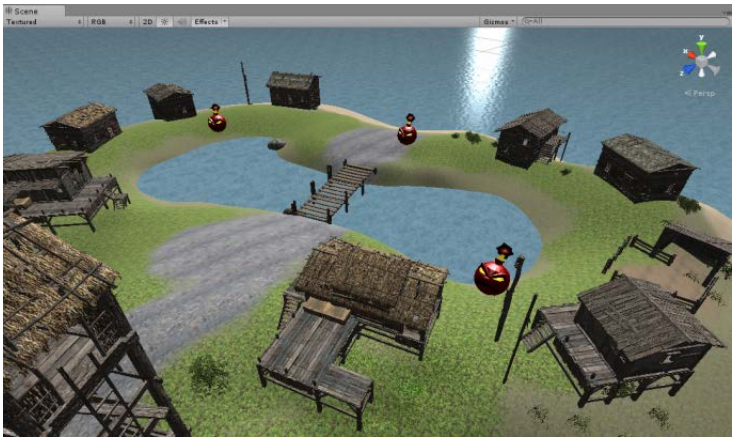
Κάθε script έχει διάφορες function όπου καλούνται διαδοχικά μερικές από αυτές είναι :

1. **Awake:** Η συνάρτηση αυτή ξεκινάει όταν φορτώσει η σκηνή
2. **Start:** Η συνάρτηση αυτή ξεκινάει πριν ξεκινήσει να τρέχει το 1ο Frame
3. **Update:** Η Βασικότερη συνάρτηση στο Unity αφού τρέχει κάθε Frame,εδώ ρυθμίζουμε τα animation του παιχνιδιού αλλά και ότι θέλουμε να αλληλεπιδρά, πχ μια κάμερα παρακολούθησης**.**
4. **LastUpdate:** Και αυτή η συνάρτηση τρέχει κάθε φορά ανά Frame με την διαφορά οτι τρέχει μετά την Update.

Βέβαια εκτός από τις βασικές αυτές function υπάρχουν και διάφορες άλλες όπου μας βοηθάνε να έχουμε τον πλήρη έλεγχο και να ρυθμίσουμε τα πάντα όπως τα έχουμε σχεδιάσει. Όπως οι συναρτήσεις Coroutines όπου με την yield WaitForSecond("sec") μπορούμε εσκεμμένα να παρουσιάσουμε μια καθυστέρηση.

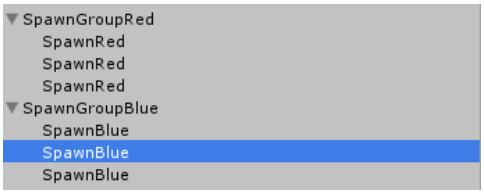
4.5 Spawn Χαρακτήρα

Με τον όρο Spawn εννοούμε το σημείο όπου ο χαρακτήρας θα εμφανίζεται ανάλογα με την ομάδα που έχει επιλέξει. Συγκεκριμένα το παιχνίδι μας αποτελείτε από δύο ομάδες - Μπλε και Κόκκινη - αριστερά και δεξιά αντίστοιχα του χάρτη.



Εικόνα 24 Spawn Red Team

Η κάθε ομάδα έχει από τρία διαφορετικά Spawn που τα έχουμε ορίσει εμείς σαν διάφανα GameObject (εικόνα 21 - 22) και οι παίχτες εμφανίζονται τυχαία σε ένα από τα τρία Spawns.



Εικόνα 25 GameObject Spawn

Αρχικά όταν ο παίχτης συνδεθεί στον Server έχει την επιλογή να διαλέξει ανάμεσα στις δύο ομάδες. Όταν επιλέγει την ομάδα τότε στον κώδικα του προγράμματος μας η μεταβλητή γυρνάει σε true ανάλογα με την ομάδα επιλογής του.

```
public bool amIOnTheRedTeam = false;.

public bool amIOnTheBlueTeam = false;
```

Τα GameObject τα επηρεάζουμε από τον κώδικα βάζοντας τους μέσω του Inspector ένα tag RedTeam και BlueTeam αντίστοιχα.

```
if(GUILayout.Button("Join Red Team", GUILayout.Height(buttonHeight)))
                        {
                                amIOnTheRedTeam = true;
                                SpawnRedTeamPlayer();
                        }

void SpawnRedTeamPlayer ()
        {
        redSpawnPoints=GameObject.FindGameObjectsWithTag("SpawnRedTeam");
        GameObject randomRedSpawn = redSpawnPoints[Random.Range (0,redSpawn
        Points.Length)];
        Network.Instantiate(redTeamPlayer, randomRedSpawn.transform.position,
                        randomRedSpawn.transform.rotation, redTeamGroup);
        }
```

Αντίστοιχα και για την Blue Team.

4.6 Master Server - Multiplayer

Ο Master server είναι ένα σύστημα network του Unity που σε συνδέει άν υπάρχει κάποιο ενεργό παιχνίδι με το ίδιο GameType.

Με πιο απλά λόγια όταν κάποιος προσπαθήσει να συνδεθεί στο Master Server τότε του επιστρέφει μια λίστα με τις εφαρμογές που τρέχουν με αυτό και ανάλογα το GameType μπαίνει στο αντίστοιχο παιχνίδι.

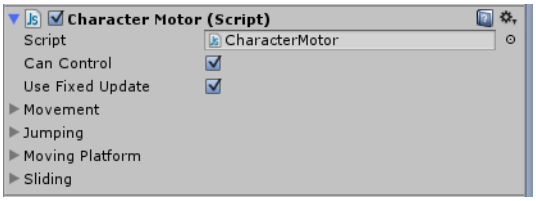
Επιπλέον ο Master Server έχει μια database όπου κρατάει πληροφορίες και ενημερώνει τα στοιχεία των παιχτών που συνδέθηκαν ή θα συνδεθουν (π.χ ip address , name , host).

Μέσα στο Multiplayer Script όπου θα δείξουμε κομμάτι κώδικα παρακάτω μπορούμε να κάνουμε όλες τις ρυθμίσεις του Server μας από την ip που θα έχει μέχρι το πόσοι παίχτες θα μπορεί να συνδέσει.

```
public class MultiplayerScript : MonoBehaviour {
        void OnGui() {
                if(GUILayout.Button("Setup a server"))
                {
                        Network.InitializeServer(numberOfPlayers, connectionPort,
                        useNAT);
                        PlayerPrefs.SetString("serverName", serverName);
                }
                if(GUILayout.Button("Go Back"))
                {
                        iWantToSetupAServer = false;
                }
        }
}
```

4.6 Κίνηση Χαρακτήρα - (Character Motor)

Ο βασικός κώδικας της κίνησης του χαρακτήρα μας όπως προαναφέραμε μπαίνει στην function update(); όπου καλείτε μια φορά κάθε frame. Η κίνηση του χαρακτήρα μας θα γίνει μέσω ενός Script κίνησης που μας το παρέχει το Unity το Character Motor.js (εικόνα 24). Όπως παρατηρείτε μας δίνει την δυνατότητα να ρυθμίσουμε Movement - Jumping(gravity) - Moving Platform - Sliding αλλά πάμε λίγο να το δούμε πως είναι το Script.



Εικόνα 27 Character Motor Script

Αρχικά δηλώνουμε τις μεταβλητές για το Movement

```
var maxForwardSpeed : float = 10.0;
var maxSidewaysSpeed : float = 10.0;
var maxBackwardsSpeed : float = 10.0;
```

Στην συνέχεια ορίζουμε ένα Vector για το transform.

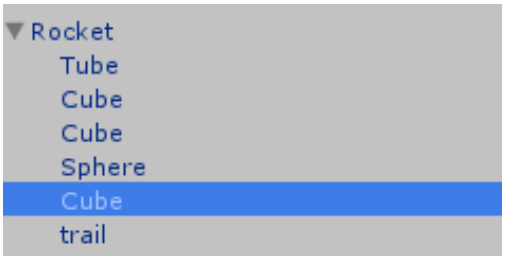
```
var inputMoveDirection : Vector3 = Vector3.zero;
```

Και τέλος γίνεται έλεγχος στο Gravity για το αν μπορεί να γίνει Jump

```
if (grounded && !IsGroundedTest()) {
                grounded = false;
else if (!grounded && IsGroundedTest()) {
                grounded = true;
                jumping.jumping = false;
```

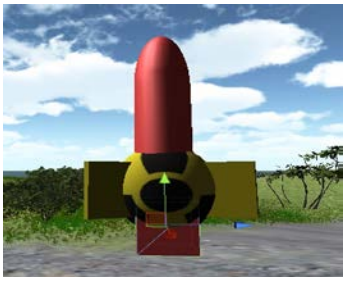
4.7 Δημιουργία Ρουκέτας

Στην πτυχιακή μας έχουμε συνολικά τρία όπλα με κάθε ένα να κάνει περισσότερο Damage εδώ θα αναλύσουμε την δημιουργία της ρουκέτας. Αρχικά η ρουκέτα σχεδιάστηκε στο Unity και αποτελείτε ένα prefab από πέντε GameObjects (εικόνα 25). Σε κάθε GameObject βάλαμε διαφορετικά materials με βάση το σχέδιο που είχαμε. Όλα τα G.O παρακάτω αποτελούν το βασικό μέρος της ρουκέτα εκτός το Trail όπου του εφαρμόσαμε ένα Trail Renderer για να δώσουμε την αίσθηση της φλόγας καθώς φεύγει η ρουκέτα από τον παίχτη.



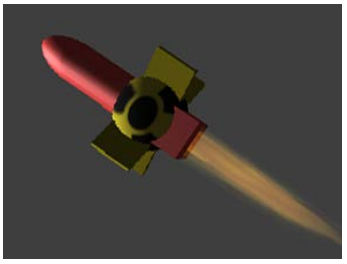
Εικόνα 28 GameObject Ρουκετας

Και τελικά καταλήξαμε το prefab μας να έχει την τελική μορφή όπως την εικόνα 26. Ακόμα στην ρουκέτα προσθέσαμε ένα explosion (εικόνα 27) , δηλαδή ένα effect όταν η ρουκέτα μας "σκάει". Τέλος επειδή θέλαμε τα όπλα να εμφανίζονται στο Gui του χαρακτήρα μας φτιάξαμε και μια τρίτη εικόνα για να καταλαβαίνει ο παίχτης πιο όπλο έχει ενεργό (Εικόνα 28).



Εικόνα 29 Rocket Prefab

Εικόνα 31 Rocket Explosion



Εικόνα 30 Rocket για το Gui

Επειδή όμως ένα prefab δεν κάνει απολύτως τίποτα χωρίς το κατάλληλο Script δημιουργήσαμε ένα RocketScript για να του ορίσουμε τι θα κάνει.
Αρχικά του ορίσαμε ένα χρόνο όπου μετά θα καταστρέφεται, αυτό το κάναμε γιατί αν δεν έβρισκε στόχο δεν θα καταστρεφόταν και θα μας έτρωγε μόνο χώρο στον server, ,ένα Speed, ένα Damage και τέλος το σημείο που θα κάνει Spawn κάθε φορά.

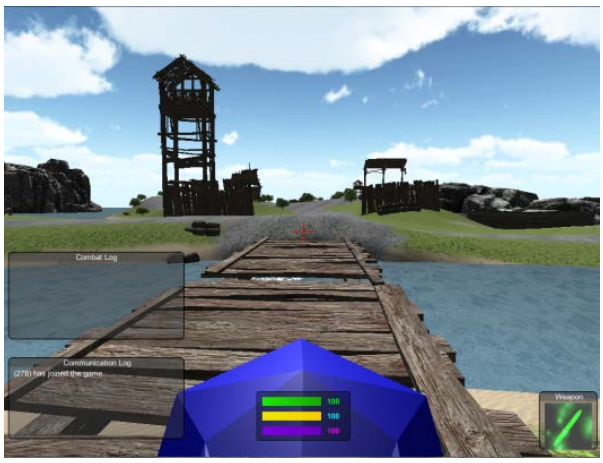
Στην δική μας εφαρμογή βάλαμε μια Camera στο κεφάλι του παίχτη μας αφού και το παιχνίδι είναι Fps (First Person Shouter) κατά 0.3 στο άξονα Z (x,y,z).

```
private float expireTime = 12f;
private float rocketSpeed = 120.0f;
private float blastRocketDamage = 80;

rocketFireFrom = cameraHeadTransform.TransformPoint(0, 0, 0.3f);
```

4.8 Κύρια οθόνη χαρακτήρα (GUI)

Η οθόνη του χαρακτήρα μας αποτελεί το τί θα βλέπει ο παίχτης καθώς κάνει Connect στον Server. Γενικά σε ένα FPS game τα βασικά που βλέπεις είναι το crosshair, τη ζωή του χαρακτήρα σου αλλά και οτιδήποτε άλλο είναι χρήσιμο για τον παίχτη.



Εικόνα 32Τελικό Gui Χαρακτήρα

Στην δικιά μας περίπτωση θα βάλουμε Crosshair - HealthBar - Weapon αλλά και τα διάφορα logs - όταν κάποιος παίχτης χάσει - και το Chat - για επικοινωνία των παιχτών μέσα από το παιχνίδι.

Εικόνα 33 Texture healthTex

Όλα αυτά είναι διάφορα Textures διαμορφωμένα μέσα σε Gui Box και ο κώδικας γράφεται μέσα στην συνάρτηση void OnGUI ().

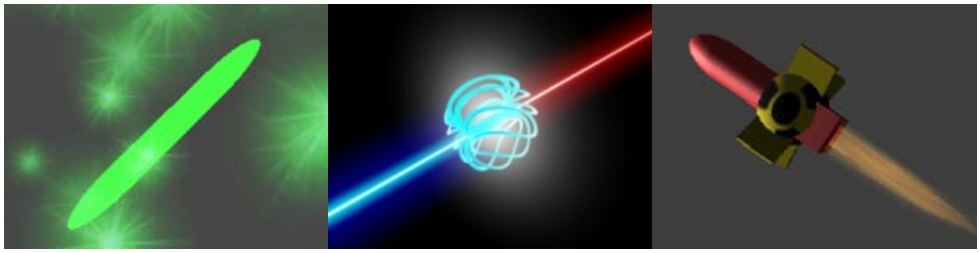
Ας το δούμε λίγο κομμάτι από τον κώδικα.

```
public Texture healthTex;
void OnGUI ()
{
GUI.DrawTexture(newRect(commonLeft+padding,commonTop+padding,he
althBarLength,healthBarHeight), healthTex);
}
```

Ακόμα για το Change Weapon επιλέξαμε ένα κουμπί το Ctrl όπου αν αυτό πατηθεί θα δείχνει το ανάλογο texture. Ο πίνακας WeaponList το ορίζουμε 0 αλλά να σημειώσουμε ξεκινάει από 1.

```
if(Input.GetButtonDown("Change Weapon"))
                {
                        selectedWeaponNumber ++;
                        if(selectedWeaponNumber == weaponList.Count)
                        {
                                selectedWeaponNumber = 0;
                        }

                        selectedWeapon                                          =
weaponList[selectedWeaponNumber];
                }
```



Εικόνα 34 Textures για ChangeWeaponList

Τέλος όπως έγινε το παραπάνω Gui κάνουμε και για το Change Weapon αλλά αλλάξουμε τις μεταβλητές ώστε να πάει right και bottom μείων το width και το height αντίστοιχα, ώστε να φαίνεται όλη η εικόνα.

```
void OnGUI()
	{
		weaponLeft = 100 ;
		weaponTop = 50;

		weaponRect = new Rect(Screen.width - weaponWidth, Screen.height - (weaponHeight+2), weaponWidth, weaponHeight);

		weaponRect = GUI.Window(9, weaponRect, selectedWeaponWindow, "Weapon");
	}
```

Κεφάλαιο 5ο : Επίλογος

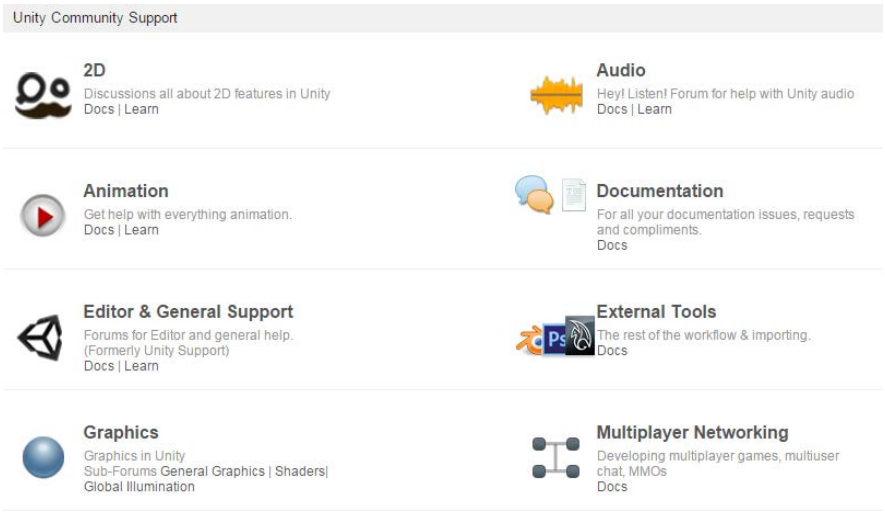
5.1 Σύνοψη

Στο κεφάλαιο αυτό θα αναφέρουμε τα προβλήματα που συναντήσαμε αλλά και τα συμπεράσματα μας πάνω στην εφαρμογή της ανάπτυξης του παιχνιδιού. Ακόμα θα δείξουμε πως μπορεί κάποιος να κατεβάσει και να συνδεθεί στην εφαρμογή μας αλλά και μια μικρή παρουσίασή του.

Εικόνα 35 Τελικό Scene - Game view

5.2 Προβλήματα Λύσεις

Λόγω οτι κανείς μας δεν είχε ξανασχοληθεί με το Unity3d Game Engine, τα προβλήματα που μας παρουσιάστηκαν ήταν αρκετά στην αρχή αλλά τα περισσότερα καθαρά εμπειρίας και τα λύσαμε γρήγορα ψάχνοντας σε κατάλληλα forum (Unity Community www.forum.unity3d.com - Εικόνα 36.)



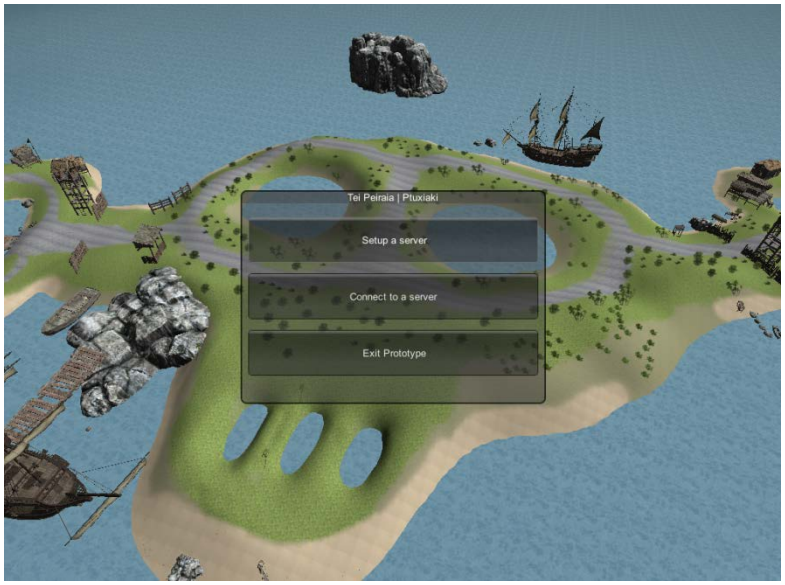
Εικόνα 36 Unity Community Support

Ένα όμως βασικό πρόβλημα που έπρεπε να επιλύσουμε ήταν τα animation του χαρακτήρα μας. Εκεί το πρόβλημα ήταν η ανταπόκριση του Server και όχι η υλοποίηση των animation. Αυτό γινόταν επειδή όλο το στήσιμο θα γινόταν σε Master Server όπου ο καθένας θα μπορεί να στήσει τον δικό του Server. Έτσι ψάξαμε να δημιουργήσουμε έναν χαρακτήρα όπου τα Animation δεν έπαιζαν σημαντικό ρόλο (βήματα - κούνημα χεριών), τελικά καταλήξαμε σε έναν χαρακτήρα τύπου Pac Man όπου με τις κατάλληλες μετατροπές πήρε την τελική μορφή που θέλαμε (εικόνα 37).

Εικόνα 37 Τελικοί Χαρακτήρες - Blue - Red

5.3 Παρουσίαση Τελικού αποτελέσματος

Στο κεφάλαιο αυτό θα παρουσιάσουμε το τελικό αποτέλεσμα της εφαρμογής μας από μεριά του Server και από την μεριά του Παίχτη. Ξεκινώντας το παιχνίδι φορτώνεται η πρώτη σκηνή του παιχνιδιού όπου έχεις την δυνατότητα να επιλέξεις αν θα φτιάξεις εσύ τον δικό σου Server ή θα συνδεθείς σαν απλός παίχτης.



Εικόνα 38 Αρχική Σκηνή Παιχνιδιού

Ως πρώτη επιλογή θα επιλέξουμε να φτιάξουμε εμείς τον Server για να δούμε τις επιλογές που μας δίνει.

Εικόνα 39 Server

Παρατηρούμε ότι έχουμε την δυνατότητα να βλέπουμε από την main camera απευθείας ότι γίνεται στο παιχνίδι, ακόμα το όνομα και τους συνδεδεμένους

παίχτες που βρίσκονται στον Server μας αλλά και να τερματίσουμε την λειτουργία του μέσω της επιλογής "Shutdown". Επιπλέον μπορούμε πολύ εύκολά να αλλάξουμε τους βαθμούς που πρέπει να μαζέψει η ομάδα για να κερδίσει. Τέλος, μέσω του Combat log και Communication log μπορούμε αντίστοιχα να δούμε τις βασικές πληροφορίες του server (join - kills) και να στείλουμε ένα μήνυμα προς όλους τους παίχτες.

Αφού δημιουργήσαμε τον Server και είδαμε όλες τις δυνατότητες του πάμε να τρέξουμε την εφαρμογή μας σας παίχτης. Από την επιλογή "Connect to a Server"

Εικόνα 40 Connect to a Server

Εδώ παρατηρούμε ότι έχουμε την επιλογή να αλλάξουμε το όνομα του παίχτη, την IP του Server όπου θα συνδεθεί (αφήνουμε την Default αν το τρέχουμε για τοπικό δίκτυο αλλιώς την IP του Hamachi Client που θέλουμε). και τέλος την Port που θα κάνουμε την σύνδεση (και εδώ αφήνουμε την Default εκτός αν την χρησιμοποιούμε για άλλη εφαρμογή). Τέλος πατάμε Connect για να συνδεθούμε.

Καθώς είμαστε συνδεδεμένοι με τον Server βλέπουμε - Εικόνα 41 - την επιλογή που μας δίνει να επιλέξουμε την ομάδα που θέλουμε (Red Team - Blue Team), Επίσης από την στιγμή που έχουμε κάνει από πριν Connect βλέπουμε την Main Camera αλλά και τα Combat Logs (όπως στην περίπτωση του Server).

Εικόνα 41 Team Selection

Όταν επιλέξουμε την ομάδα παρατηρούμε ότι μας βάζει στο αντίστοιχο Spawn της και είμαστε έτοιμοι να παίξουμε.

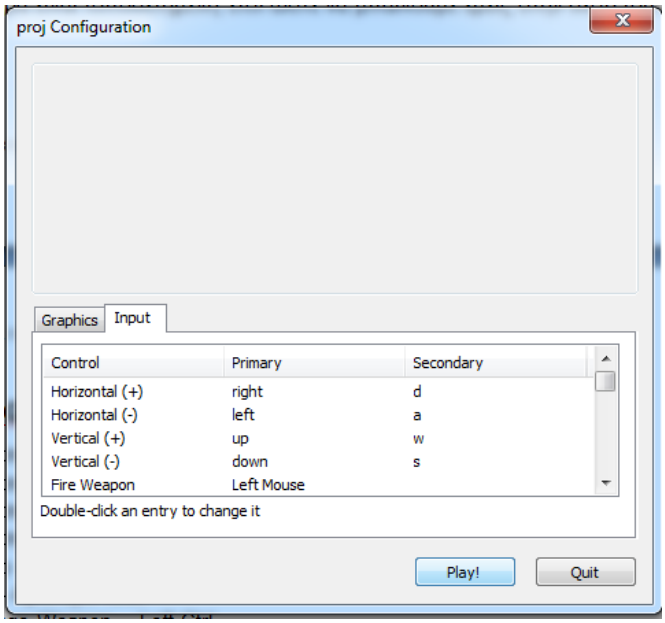
Εικόνα 42 Join Red Team

Τέλος όπως προαναφέραμε το παιχνίδι μας είναι στην κατηγορία των FPS (First person Shouter) γι' αυτό και δεν βλέπουμε τον χαρακτήρα μας αλλά η κάμερα είναι τοποθετημένη έτσι ώστε να μπαίνουμε εμείς στην θέση του.

5.4 GamePlay Inputs

Κίνηση Εμπρός - Πλήκτρο W - Up
Κίνηση Πίσω - Πλήκτρο S - Down
Κίνηση Αριστερά - Πλήκτρο A - Left
Κίνηση Δεξιά - Πλήκτρο D - Right
Jump - Space
Fire - Αριστερό Κλικ
Change Weapon - Left Ctrl
Communication - T
Send Message - Return - Enter
Show Scores - Tab

Αυτά είναι τα Inputs του παιχνιδιού μπορούμε βέβαια πριν ξεκινήσουμε από την επιλογή Input στο Unity (εικόνα 43) να τα δούμε αλλά και με διπλό κλικ πάνω τους να τα αλλάξουμε. Είναι πολύ σημαντικό γιατί μπορούμε εύκολα να συνδέσουμε το δικό μας Joystick στο παιχνίδι μας.



Εικόνα 43 Inputs

5.5 Μελλοντικές Επεκτάσεις

Ήδη το παιχνίδι έχει πάρει μια επέκταση και σε Web Player στημένο σε ένα Host όπου ο καθένας μπορεί να μπαίνει και να παίζει κατευθείαν. Τα θέματα που μελλοντικά σκεφτόμαστε να υλοποιήσουμε είναι το στήσιμο ενός Public Server όπου όλοι θα κάνουν κατευθείαν Connect , τα animation και η αλλαγή χαρακτήρα αν έχουμε και την υποστήριξη από μεριάς Server. Τέλος να βάλουμε ήχο στο παιχνίδι αφού αποτελεί ένα βασικό μέρος για το σωστό και καλό Gameplay.

5.6 Συμπεράσματα

Το Unity είναι ένα ολοκληρωμένο πακέτο όπου δίνει την δυνατότητα στον Developer να φτιάξει το δικό του παιχνίδι 2d ή 3d με σχετικά απλό τρόπο.
Το Unity Game Engine είναι στατιστικά το καλύτερο Game Engine όσον αφορά την δημιουργία παιχνιδιών σε Android και iOS. Με την έκδοση 5 και την μεγάλη αλλαγή στο rendering το Unity πιστεύουμε κυριαρχήσει και σε PC & Game Consoles γιατί πολύ απλά είναι πιο απλό και εύχρηστο για έναν αρχάριοι αλλά και έναν απαιτητικό Developer.
Το βασικό χαρακτηριστικό που το κάνει το πιο προσιτό Game Engine είναι οι βασικοί οδηγοί, τα πολλά tutorials που υπάρχουν στο διαδίκτυο, αλλά και το Community Forum όπου εύκολα και γρήγορα μπορούν να σε βοηθήσουν.
Τέλος, η δωρεάν έκδοση αλλά και οι πολλές πλατφόρμες που υποστηρίζει κάνει την μηχανή νούμερο ένα στις προτιμήσεις των μικρών εταιριών ανάπτυξης σε μια βιομηχανία παιχνιδιών που συνεχώς αυξάνεται.

Βιβλιογραφία

[1] Unity3d.com

[2] Forum Unity

[3] Unity Tutorials

[4] Wikipedia Unity Technologies

[5] A History of the Unity Game Engine

Προγράμματα που χρησιμοποιήθηκαν

- Adobe Photoshop cs6
- Blender 3d
- Gimp 2.0
- Unity 3d Game Engine

Εικόνα 44 Προγράμματα που Χρησιμοποιήθηκαν