



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη ηλεκτρονικού παιχνιδιού Πινγκ Πονγκ 2 διαστάσεων στην Unity3D Ping Pong 2D Game development in Unity3D
Όνοματεπώνυμο Φοιτητή	Ιωάννης Καλογερόπουλος
Πατρώνυμο	Δημήτριος Καλογερόπουλος
Αριθμός Μητρώου	ΜΠΠΛ18021
Επιβλέπων	Θεμιστοκλής Παναγιωτόπουλος, Καθηγητής

Δεκέμβριος 2022

Τριμελής Εξεταστική Επιτροπή

Όνομα Επώνυμο
Βαθμίδα
Θεμιστοκλής Παναγιωτόπουλος
Καθηγητής

Όνομα Επώνυμο
Βαθμίδα
Άγγελος Πικράκης
Επίκουρος Καθηγητής

Όνομα Επώνυμο
Βαθμίδα
Ιωάννης Τασούλας
Επίκουρος Καθηγητής

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στα πλαίσια της ολοκλήρωσης των σπουδών μου στο Μεταπτυχιακό Πρόγραμμα Σπουδών του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς.

Θέλω να ευχαριστήσω τον κ. Θέμη Παναγιωτόπουλο για το θέμα της παρούσας διπλωματικής που μου πρότεινε, την προθυμία του να τεθεί επιβλέπων και την άψογη συνεργασία μας, για την αμέριστη υποστήριξη που μου παρείχε σε όλη τη διάρκεια της εκπόνησης της παρούσας εργασίας καθώς και τους καθηγητές της τριμελούς επιτροπής που δέχτηκαν να παραβρεθούν στην εξέταση της διπλωματικής μου εργασίας. Επίσης, θα ήθελα να ευχαριστήσω των συμφοιτητή μου Νικόλα Δέγγλερη, με τον οποίο συνεργαστήκαμε σε ορισμένα κομμάτια στην διάρκεια υλοποίησης της μεταπτυχιακής διατριβής του καθενός ώστε να προκύψει ένα πιο άρτιο και χρήσιμο για το πανεπιστήμιο αποτέλεσμα της εργασίας.

Ακόμα οι ευχαριστίες απευθύνονται προς τους όλους συμφοιτητές μου και τους διδάσκοντες του μεταπτυχιακού προγράμματος για την πολύπλευρη συνεργασία που είχαμε κατά τη διάρκεια των σπουδών μου. Τέλος, θα ήθελα να ευχαριστήσω και να επισημάνω την σημαντική υποστήριξη που μου παρείχαν τα άτομα του οικογενειακού μου κύκλου από κάθε πλευρά.

Περίληψη

Η παρούσα μεταπτυχιακή εργασία έχει ως στόχο τον σχεδιασμό, την ανάπτυξη & τρόπο λειτουργία ενός «ρέτρο» ηλεκτρονικού παιχνιδιού αλλά σε μία σύγχρονη και εξελιγμένη μορφή σε σύγκριση με τότε που πρωτοδημιουργήθηκε. Η υλοποίηση της βασίστηκε στη λογική του αντικειμενοστραφούς προγραμματισμού καθώς και την διαδικασία της σύνθεσης με την βοήθεια της μηχανής Unity 3D Engine σε συνδυασμό με τη γλώσσα προγραμματισμού C#, όπου και η ανάλυση του θα είναι και το μεγαλύτερο μέρος της εργασίας. Επιπλέον θα υπάρξει ανάλυση των δυνατοτήτων που μας προσφέρουν τα εργαλεία που έχουμε σήμερα στην κατοχή μας καθώς και μία σύγκριση από τα πρώτα χρόνια δημιουργίας βιντεοπαιχνιδιών. Ακόμα θα υπάρξουν κάποιες σύντομες ιστορικές αναφορές σε βιντεοπαιχνίδια καθώς και στο πόσο μεγάλη επιρροή έχουν στην καθημερινή μας ζωή και γενικότερα.

Abstract

This master's thesis aims to design, develop & operate a "retro" electronic game but in a modern and sophisticated form compared to when it was first created. Its implementation was based on the logic of object-oriented programming as well as the composition process with the help of the Unity 3D Engine in combination with the C# programming language, where the analysis will also be the major part of the work. In addition there will be an analysis of the possibilities offered by the tools we have today as well as a comparison from the early years of video game creation. There will also be some brief historical references to video games as well as how much of an influence they have on our daily lives and in general.

Περιεχόμενα

1.Εισαγωγή.....	8
1.1 Γενική εισαγωγή.....	8
1.2 Ιστορική αναδρομή	8
2. Μηχανή Unity3D Engine και editors	13
2.1 Γενικά.....	13
2.2 Unity Studio Game engine	14
2.2.1 Η Unity ως Game engine.....	14
2.2.2 Σχετικά με τη C#	18
3.1 Εισαγωγή	20
3.2 Στόχος & Τρόπος παιχνιδιού.....	20
3.3 Κατασκευή περιβάλλοντος - Σταδίου	23
3.3.1 Τοποθετήσεις	24
3.3.2 Αλληλεπίδραση αντικειμένων	26
3.3.3 Score UI.....	28
3.3.4 Start Countdown	29
3.4 Scripting & Ενσωμάτωση σε objects	29
3.4.1 Γενικά.....	29
3.4.2 Scripting στη Unity.....	30
3.4.3 Βιβλιοθήκες.....	30
3.4.4 Ενσωμάτωση των scripts στα objects.....	30
3.4.5 Script των objects	30
4.Συμπεράσματα	40
5.Βιβλιογραφία.....	41

1.Εισαγωγή

1.1 Γενική εισαγωγή

Αν η γενιά μας μπορούσε να χαρακτηριστεί από μία φράση αυτή θα ήταν η γενιά των video games καθώς ήταν η γενιά που σε σχεδόν κάθε νοικοκυριό υπήρχε η δυνατότητα να υπάρχει μία κονσόλα βιντεοπαιχνιδιού ή ένας υπολογιστής. Σήμερα η ορολογία του video gaming έχει πάρει πολύ μεγάλες διαστάσεις από ότι θα φάνταζε πριν 30 χρόνια όπου κάποιος για να παίξει ένα απλό βιντεοπαιχνίδι θα έπρεπε να είχε πρόσβαση σε κάποιο κατάστημα με ηλεκτρονικά παιχνίδια για να απολαύσει μερικά λεπτά διασκέδασης. Με τη βοήθεια της τεχνολογίας ένας μέσος άνθρωπος μπορεί να απολαύσει ένα video game ακόμα και στο κινητό του και ακόμη πιο εύκολα να δημιουργήσει ο ίδιος το δικό του παιχνίδι με διάφορες μηχανές όπως unity3d engine, unreal engine αλλά και άλλες. Φανταστείτε λοιπόν τόσες γενιές που έχουν μεγαλώσει με τα Video Games να έχουν πλέον τη δυνατότητα να δημιουργήσει ο καθένας το δικό του βιντεοπαιχνίδι και δωρεάν. Στη δικιά μας περίπτωση θα χρησιμοποιήσουμε την unity3d engine και με τη βοήθεια της γλώσσας προγραμματισμού C# θα αναδημιουργήσουμε το πρώτο βιντεοπαιχνίδι που φτιάχτηκε με σύγχρονα δεδομένα.



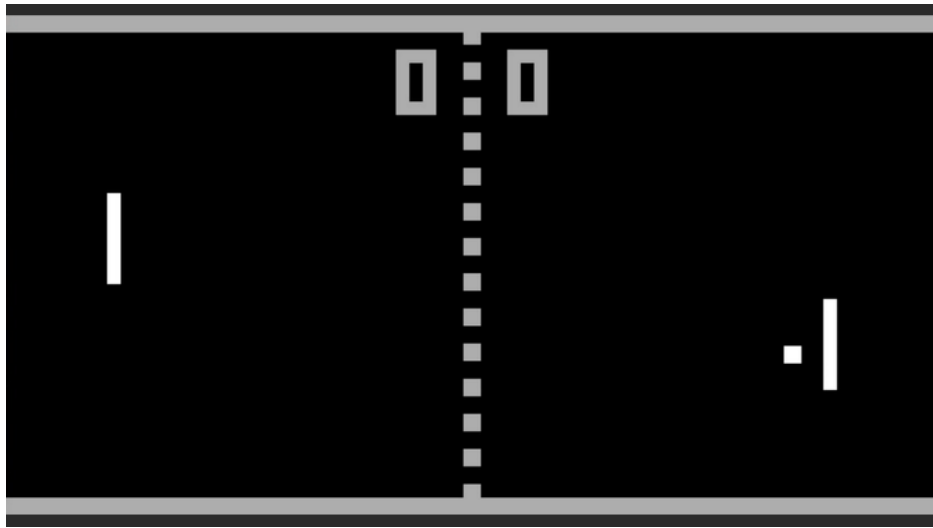
Space Invaders arcade game. Πηγή https://en.wikipedia.org/wiki/History_of_video_games

1.2 Ιστορική αναδρομή

Τον Οκτώβριο του 1958, ο φυσικός William Higinbotham δημιούργησε αυτό που θεωρείται το πρώτο βιντεοπαιχνίδι. Είναι ένα πολύ απλό παιχνίδι τένις, παρόμοιο με το κλασικό βιντεοπαιχνίδι Ping Pong της δεκαετίας του 1970, το οποίο ήταν δημοφιλές στα ανοιχτά γήπεδα στο Εθνικό Εργαστήριο Brookhaven. Ο Higinbotham γεννήθηκε στις 25 Οκτωβρίου 1910 στο Bridgerport του Κονέκτικατ και μεγάλωσε στην Καληδονία της Νέας Υόρκης. Αποφοίτησε από το Williams College το 1932 και μετά πήγε στο Πανεπιστήμιο Cornell για μεταπτυχιακές σπουδές στη φυσική. Ενώ ήταν μεταπτυχιακός φοιτητής στο Πανεπιστήμιο Cornell, εργάστηκε ως τεχνικός ηλεκτρονικών. Το 1941, εντάχθηκε στο Εργαστήριο Ακτινοβολίας του MIT, όπου εργάστηκε σε οθόνες καθοδικών λυχνιών για συστήματα ραντάρ. Το 1943, μετακόμισε στο Λος Άλαμος για να εργαστεί στα ηλεκτρονικά του συστήματος χρονισμού της ατομικής βόμβας. Το 1948, εντάχθηκε στο Instrumentation Group στο Brookhaven National Laboratory. Ήταν επικεφαλής της ομάδας από το 1951 έως το 1968. Κατά τη διάρκεια αυτής της περιόδου, το Brookhaven φιλοξενεί την ετήσια Ημέρα του Επισκέπτη τον Οκτώβριο, κατά την οποία χιλιάδες άνθρωποι έρχονται να δουν το εργαστήριο. Ο Higinbotham ανέλαβε να δημιουργήσει

μια έκθεση για να παρουσιάσει τη δουλειά του οργανικού τμήματος. Τα περισσότερα από τα υπάρχοντα εκθέματα είναι βαρετά. Ο Higinbotham σκέφτηκε ότι θα μπορούσε να προσελκύσει καλύτερα τους επισκέπτες δημιουργώντας διαδραστικές οθόνες. Αργότερα θυμήθηκε σε μια συνέντευξη σε περιοδικό ότι πίστευε ότι «θα μπορούσε να δώσει ενέργεια στον τόπο και να επιτρέψει στους ανθρώπους να παίζουν παιχνίδια όπως αυτό και να στείλουν το μήνυμα ότι οι επιστημονικές μας προσπάθειες είναι σημαντικές για την κοινωνία». Το ταμπλό οργάνων διαθέτει έναν μικρό αναλογικό υπολογιστή που μπορεί να εμφανίσει διάφορες καμπύλες στο πεδίο, συμπεριλαμβανομένης της διαδρομής μιας μπάλας που αναπηδά. Ο Higinbotham χρειάστηκε μόνο λίγες ώρες για να συλλάβει την ιδέα για το παιχνίδι τένις και μόλις λίγες μέρες για να συναρμολογήσει τα βασικά εξαρτήματα. Ο Higinbotham είχε εργαστεί σε οθόνες για συστήματα ραντάρ και πολλές άλλες ηλεκτρονικές συσκευές και δεν είχε κανένα πρόβλημα να σχεδιάσει απλές οθόνες gaming. Ο Higinbotham σχεδίασε μερικά σχέδια και σχεδίασε. Ο τεχνικός Robert Dvořák χρειάστηκε περίπου δύο εβδομάδες για να κατασκευάσει τη συσκευή. Μετά από κάποιες αλλαγές, το πρώτο βιντεοπαιχνίδι είναι έτοιμο. Αυτό το παιχνίδι το λένε τένις δύο παικτών. Οι παίκτες μπορούν να γυρίσουν ένα κουμπί για να προσαρμόσουν τη γωνία της μπάλας και, στη συνέχεια, να πατήσουν ένα κουμπί για να χτυπήσουν την μπάλα σε έναν άλλο παίκτη. Όσο πατιέται το κουμπί ενώ η μπάλα βρίσκεται στο δικό του γήπεδο, ο παίκτης δεν θα χάσει την μπάλα, αλλά αν τη χτυπήσει τη λάθος στιγμή ή από τη λάθος γωνία, η μπάλα δεν θα μπει στα δίχτυα. Οι μπάλες που χτυπούν στο έδαφος αναπηδούν σαν αληθινές μπάλες του τένις. Όταν η μπάλα φεύγει από το γήπεδο ή το φιλέ, ο παίκτης θα πατήσει το κουμπί επαναφοράς για να ξεκινήσει ο επόμενος γύρος. Το τένις δύο παικτών δεν έχει τα φανταχτερά γραφικά που χρησιμοποιούν τα βιντεοπαιχνίδια σήμερα. Μια οθόνη καθοδικού σωλήνα δείχνει απλώς μια πλάγια όψη ενός γηπέδου τένις, που αντιπροσωπεύεται από δύο μόνο γραμμές, μία για το έδαφος και μία για το δίχτυ. Η μπάλα είναι απλώς ένα σημείο που αναπηδά μπρος-πίσω. Οι παίκτες πρέπει επίσης να παρακολουθούν τους δικούς τους πόντους. Το κύκλωμα παιχνιδιού είναι πολύ απλό, χρησιμοποιώντας κυρίως αντιστάσεις, πυκνωτές και ρελέ, αν και χρησιμοποιεί τρανζίστορ για τη γρήγορη εναλλαγή που απαιτείται όταν η μπάλα βρίσκεται σε κίνηση. Οι τουρίστες το λατρεύουν. Γρήγορα έγινε το πιο δημοφιλές έκθεμα, με τους ανθρώπους να κάνουν ουρές για την ευκαιρία να παίξουν. Η πρώτη έκδοση, που χρησιμοποιήθηκε την εποχή των επισκεπτών του 1958, διέθετε μια μικρή οθόνη με διάμετρο μόνο 5 ίντσες. Την επόμενη χρονιά, η Higinbotham το βελτίωσε με μεγαλύτερη οθόνη. Προσθέτει επίσης ένα άλλο χαρακτηριστικό: το παιχνίδι μπορεί πλέον να προσομοιώνει ισχυρότερη ή ασθενέστερη βαρύτητα, ώστε οι επισκέπτες να μπορούν να παίζουν τένις στο φεγγάρι, τη Γη ή τον Δία. Το Tennis for Two αποσύρθηκε μετά από δύο χρόνια. Ο παλμογράφος και ο υπολογιστής επαναπροσδιορίστηκαν και ο Higinbotham σχεδίασε μια νέα οθόνη της Ημέρας του Επισκέπτη που δείχνει κοσμικές ακτίνες να περνούν μέσα από τον θάλαμο σπινθήρα. Ο Higinbotham, ο οποίος είχε ήδη κατοχυρώσει με δίπλωμα ευρεσιτεχνίας 20 εφευρέσεις, δεν είδε το παιχνίδι του τένις ως ιδιαίτερα καινοτόμο. Ήξερε ότι το παιχνίδι άρεσε στους επισκέπτες του Brookhaven, αλλά δεν είχε ιδέα πόσο δημοφιλή βιντεοπαιχνίδια θα γίνονταν στο μέλλον, ότι ήταν υπέροχο, αλλά επειδή όλα τα άλλα ήταν βαρετά», είπε κάποτε. Το τένις για δύο ξεχάστηκε για λίγο. Το 1964, η Sanders Associates έλαβε το πρώτο δίπλωμα ευρεσιτεχνίας για ένα βιντεοπαιχνίδι. Η Magnavox απέκτησε τα διπλώματα ευρεσιτεχνίας και κατασκευάζει συστήματα βιντεοπαιχνιδιών από τις αρχές της

δεκαετίας του 1970, αλλά η υπόθεση διευθετήθηκε εξωδικαστικά. Μόλις το 1982 ο Higginbotham έγινε γνωστός ως ο εφευρέτης των βιντεοπαιχνιδιών μετά τη δημοσίευση ενός άρθρου στο περιοδικό Creative Computing. Βοήθησε στην ίδρυση της Ομοσπονδίας Αμερικανών Επιστημόνων και υπηρέτησε ως πρώτος πρόεδρος και γενικός γραμματέας της. Ο Higginbotham, ο οποίος πέθανε τον Νοέμβριο του 1994, ήταν περισσότερο γνωστός για τα βιντεοπαιχνίδια του παρά για τη δουλειά του για τη μη διάδοση.



Το πρώτο arcade βιντεοπαιχνίδι που δημιουργήθηκε ποτέ. Πηγή https://en.wikipedia.org/wiki/History_of_video_games

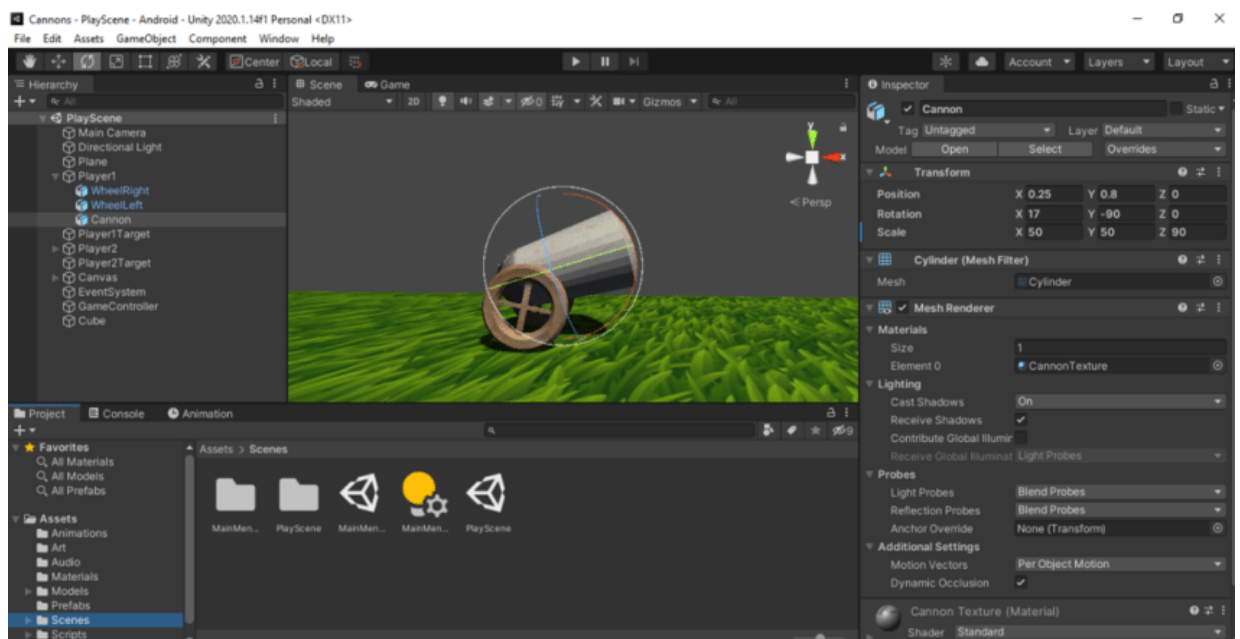


Πιθανότατα η πρώτη κονσόλα. Πηγή https://en.wikipedia.org/wiki/History_of_video_games.

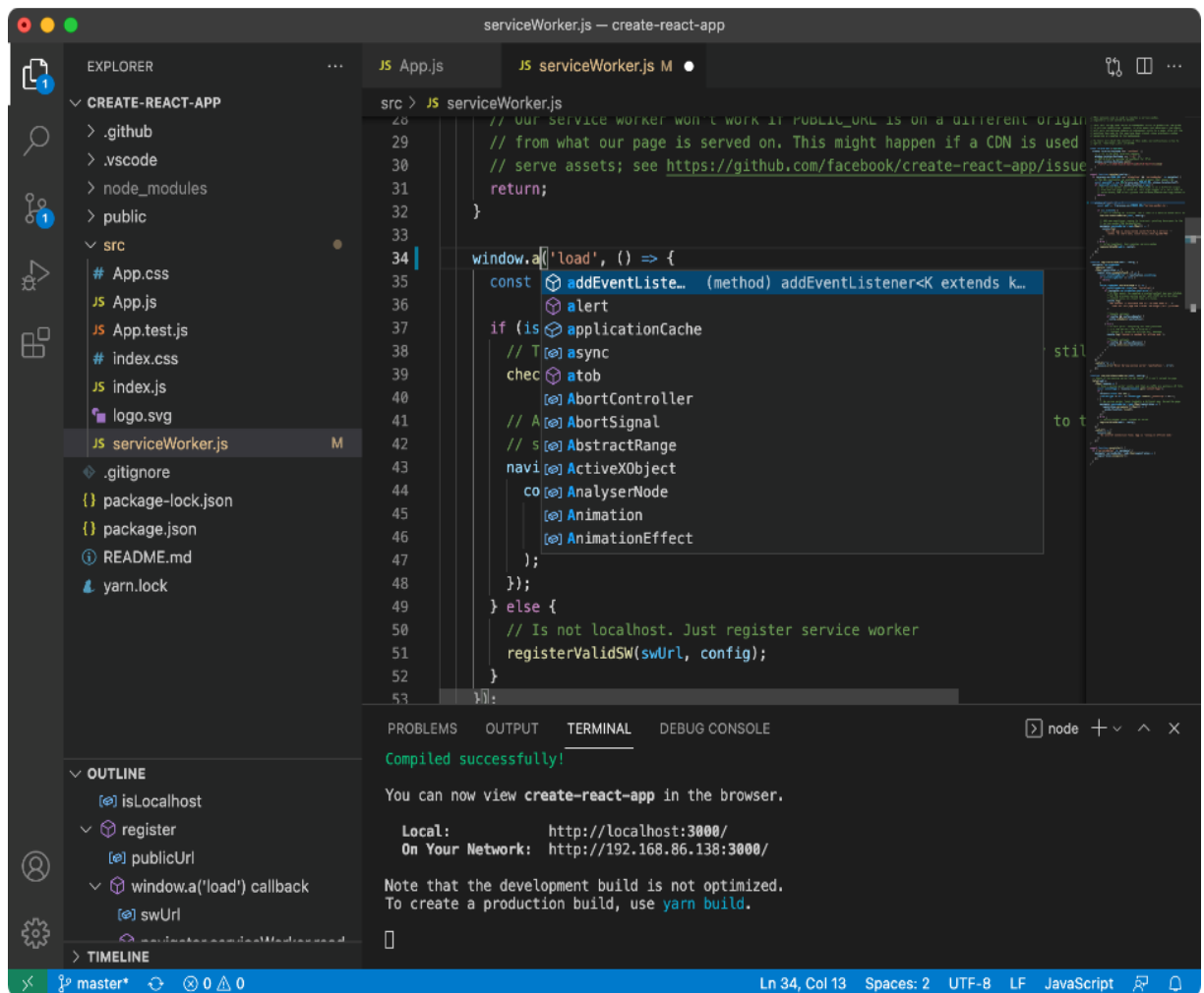
2. Μηχανή Unity3D Engine και editors

2.1 Γενικά

Σε αυτό το κεφάλαιο θα αναφερθούν τα απαραίτητα εργαλεία που χρησιμοποιήθηκαν για να αναπτυχθεί το παιχνίδι. Το πρώτο και κυριότερο πρόγραμμα που θα χρειαστεί κάποιος για τη δημιουργία ενός video game είναι μία μηχανή όπως unity 3D engine. Στη μηχανή αυτή μπορούμε να δημιουργήσουμε παιχνίδια του δύο διαστάσεων ή ακόμα και τριών διαστάσεων όπως είναι στη δικιά μας περίπτωση το παιχνίδι που έχουμε δημιουργήσει. Πέρα από τη μηχανή Unity θα χρησιμοποιήσουμε και κάποιους editors οι οποίοι θα μας βοηθήσουν στις εντολές προγραμματισμού και πιο συγκεκριμένα θα χρησιμοποιήσουμε το visual studio code. Μέσω αυτού του editor μας δίνεται η δυνατότητα να καταλάβουμε κάποια πράγματα για τις εντολές, όπως αν αφορά κάποιο class, method, function κ.α. Ακόμα η χρήση χρωμάτων βοηθά στο να γίνει πιο ευανάγνωστος ο κώδικας που έχουμε γράψει, όχι μόνο για εμάς αλλά και σε μία ομάδα που ενδεχομένως να συμμετέχουμε. Επίσης όταν αναπτύσσουμε ένα πρόγραμμα ορίζουμε κάποιους άτυπους κανόνες σχετικά με τον τρόπο γραφής εντολών προκειμένου ο κώδικας να είναι πιο κατανοητός όχι μόνο για εμάς αλλά και τα άλλα μέλη της ομάδας στην οποία ενδεχομένως να συμμετέχουμε.



Εικόνα 2.1.1. Περιβάλλον Unity. Πηγή https://www.researchgate.net/figure/Application-Development-on-Unity-3D-Engine_fig2_347711912



2.1.2. Editor Visual Studio Code. Πηγή www.google.com

Τα 3D μοντέλα που θα χρησιμοποιηθούν για την εργασία έχουν προμηθευθεί από εξωτερικό συνεργάτη ο οποίος εξειδικεύεται στο 3D design προκειμένου να πλησιάσουμε συνθήκες πραγματικής εργασιακής εμπειρίας και συνεργασίας με διάφορων ειδικοτήτων συνεργάτες.

2.2 Unity Studio Game engine

2.2.1 Η Unity ως Game engine

Πρόλογος

Δεδομένου ότι είναι μια πλήρης μηχανή ανάπτυξης παιχνιδιών, είναι η καλύτερη μηχανή παιχνιδιών για πολλούς ανθρώπους, έχει ένα υπέροχο φόρουμ και φαν κλαμπ υποστηρικτών που μπορούν να βοηθήσουν πολύ γρήγορα στα γραφικά και έχει το πλεονέκτημα ότι τα scripts τρέχουν πιο γρήγορα από άλλες μηχανές.

Τρέχοντα εργαλεία

Η Unity χρησιμοποιείται από 2 εκατομμύρια προγραμματιστές, εκ των οποίων 300.000 το χρησιμοποιούν μηνιαίως. Η Unity παρουσιάστηκε τον Μάρτιο του 2015 με σημαντικές αλλαγές στα γραφικά και τον καθαρό ήχο του παιχνιδιού.

Τεχνολογία γραφικών

Η Unity έχει βελτιώσει σημαντικά τα γραφικά βελτιώνοντας τα shaders και ενσωματώνοντας το Enlighten, μια τεχνολογία φωτισμού σε πραγματικό χρόνο. Η συντριπτική πλειονότητα των σύγχρονων παιχνιδιών χρησιμοποιεί τεχνικές απόδοσης βάσει φυσικής ιδιότητας (PBR) και η Unity μπήκε σε αυτήν την κατηγορία προσθέτοντάς το στον κλάδο γραφικών υψηλής επόμενης γενιάς.

Ο επεξεργαστής Unity

Ο επεξεργαστής Unity αποτελείται από πολλούς πίνακες που περιέχουν το έργο σας και μπορούν εύκολα να διαχωριστούν. Οι χρήστες μπορούν να τα μεταφέρουν και να τα δουν όπως θέλουν, δημιουργώντας έτσι τη δική τους μοναδική διεπαφή. Τα πιο σημαντικά από αυτά, τα οποία θα καλύψουμε στη συνέχεια, είναι:

- Προβολή σκηνής (Scene View)
- Προβολή παιχνιδιού (Game View)
- Ιεραρχία (Hierarchy)
- Έργο (Project)
- Επιθεωρητής (Inspector)

Προβολή σκηνής

Η «Σκηνή» επιτρέπει στο χρήστη να περιηγηθεί και να επεξεργαστεί αυτόν τον χώρο χρησιμοποιώντας τα πλήκτρα Q, W, E, R. μεγέθυνση.

Πλήκτρο Q (εργαλείο πλοήγησης): χρησιμοποιήστε τα πλήκτρα, τα οποία βρίσκονται στην οθόνη, για να μετακινηθείτε στον χώρο. Συγκεκριμένα, κρατώντας πατημένο το αριστερό κουμπί του ποντικιού μετακινείται η κάμερα πάνω, κάτω, αριστερά ή δεξιά. Κρατήστε ταυτόχρονα το πλήκτρο Alt για να περιστρέψετε γύρω από το εστιασμένο αντικείμενο. Τώρα πατήστε το δεξί κουμπί του ποντικιού για να μετακινήσετε την κάμερα στην επιλεγμένη κατεύθυνση (αριστερά, δεξιά, πάνω, κάτω) και ταυτόχρονα πατήστε Alt για μεγέθυνση.

W: χρησιμοποιείται όταν το τοποθετημένο αντικείμενο είναι εστιασμένο και σας επιτρέπει να κινηθείτε προς την κατεύθυνση που υποδεικνύεται από το βέλος του αντικειμένου, σύμφωνα με τους άξονες x, y, z.

E (Εργαλείο περιστροφής): Σας επιτρέπει να περιστρέψετε ένα αντικείμενο επιλέγοντας έναν άξονα περιστροφής κάθε φορά.

Πλήκτρο R (εργαλείο κλίμακας): Τέλος, το πλήκτρο R σας επιτρέπει να ορίσετε ανεξάρτητα το μέγεθος ενός αντικειμένου σε κάθε άξονα. Μπορείτε επίσης να κλιμακώσετε ένα αντικείμενο αναλογικά κάνοντας κλικ στο κέντρο του αντικειμένου.

Επιλογές απόδοσης :Ακριβώς κάτω από το παράθυρο της σκηνής θα δείτε πώς αποδίδονται τα αντικείμενα(υφές). Πρόσθετες επιλογές (Wireframe, Tex, RenderPass, Light map Resolution, Light Probs) μπορούν να επιλεγούν από το αναπτυσσόμενο μενού. τρόπος.

(Κίτρινο Πλαίσιο Εικόνας 2.2.1α)

Προβολή παιχνιδιού

Η Προβολή παιχνιδιού αντιπροσωπεύει το παιχνίδι στην τελική του μορφή. Παύση και «Επόμενο καρέ». Συγκεκριμένα :

Παίξτε :Αυτό το κουμπί σας επιτρέπει να δοκιμάσετε το παιχνίδι για να πάρετε μια ιδέα για την τελική του μορφή.

Παύση: Αυτό το κουμπί σας επιτρέπει να κάνετε παύση του παιχνιδιού σε ένα συγκεκριμένο πλαίσιο για να δείτε λεπτομέρειες ή να επεξεργαστείτε τιμές που επηρεάζουν την απόδοση (ταχύτητα χαρακτήρων, βαρύτητα, φωτισμός κ.λπ.).

Επόμενο πλαίσιο: Αυτό το κουμπί σας επιτρέπει: Να προβάλλετε και να επεξεργάζεστε με ακρίβεια κάθε καρέ του παιχνιδιού σας. Ένα αναπτυσσόμενο μενού θα εμφανιστεί στην αριστερή πλευρά της γραμμής εργαλείων Προβολή παιχνιδιού, επιτρέποντάς σας να κάνετε προεπισκόπηση του παιχνιδιού στην ανάλυση που έχετε ορίσει. Στη συνέχεια, θα δείτε το παράθυρο Μεγιστοποίηση κατά την αναπαραγωγή στα δεξιά. Εάν το επιλέξετε και πατήσετε play, το παιχνίδι θα ξεκινήσει σε λειτουργία πλήρους οθόνης. Υπάρχει επίσης μια επιλογή "Στατιστικά" που εμφανίζει γραφικά στατιστικά.

(Πορτοκαλί Πλαίσιο Εικόνας 2.2.1α)

Hierarchy and Project

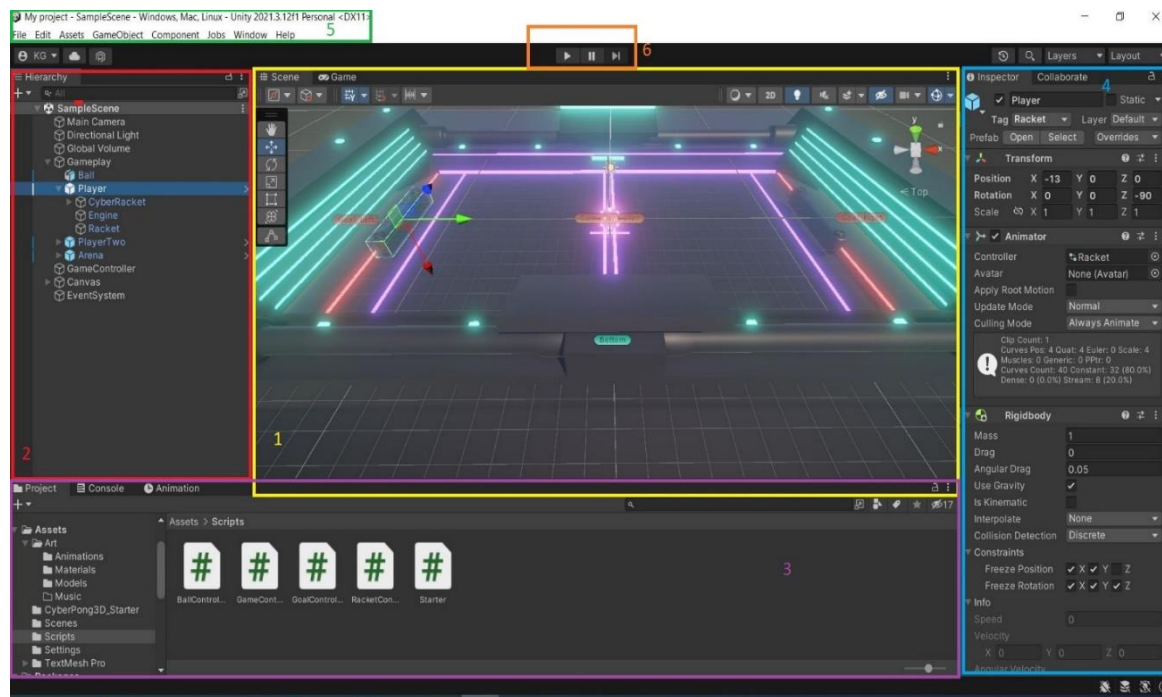
Hierarchy είναι ένα παράθυρο που περιέχει όλα τα αντικείμενα στη σκηνή και τα ονόματά τους, από το οποίο μπορείτε να αλλάξετε τις τελικές ρυθμίσεις. Εν ολίγοις, το παιχνίδι λαμβάνει τις πληροφορίες του από έναν συγκεκριμένο φάκελο, όχι από ένα έργο των Windows. Αναλύονται παρακάτω. Από την άλλη πλευρά, το παράθυρο του έργου περιέχει όλα τα στοιχεία του παιχνιδιού σας. Με απλά λόγια, όλα όσα χρειάζεστε για το παιχνίδι σας είναι εκεί (Υφές-Σενάρια-Ήχοι-Προκατασκευέςκ.λπ.)

(Κόκκινο και Μωβ Πλαίσιο Εικόνας 2.2.1α)

Ο Επιθεωρητής

Ο Επιθεωρητής παρουσιάζει όλα τα χαρακτηριστικά των αντικειμένων του παιχνιδιού με τη μορφή ιεραρχίας ή έργου. Επεξεργαστείτε τις ιδιότητές του ανάλογα με τον τύπο του Οι κύριες λειτουργίες είναι η θέση,η περιστροφή και η κλιμάκωση. Επιπλέον, τα αντικείμενα μπορούν να έχουν νέα χαρακτηριστικά που επηρεάζουν τον τρόπο αντίδρασης τους, όπως ήχος, φωτισμός, βαρύτητα , κινούμενα σχέδια και σενάρια.

(Μπλε Πλαίσιο Εικόνας 2.2.α)



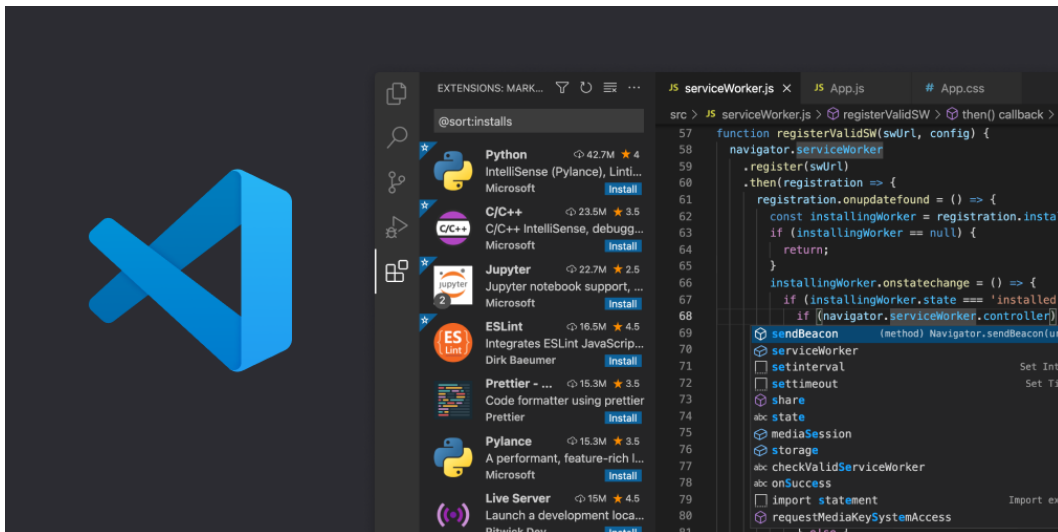
Εικόνα 2.2.2α. Interface Unity Editor με τα ειδικά παράθυρα.

Visual Studio Code

Το Visual Studio Code, κοινώς γνωστό και ως VS Code, είναι ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα που δημιουργήθηκε από τη Microsoft με το Electron Framework, για Windows, Linux και macOS. Είναι ένα ελαφρύ πρόγραμμα που μπορεί να γραφτεί σε οποιαδήποτε από τις τρεις αντικειμενοστρεφείς γλώσσες που υποστηρίζονται από τη Unity. Αυτά είναι:

- C#
- JavaScript
- Boo

Η πιο διάσημη και δημοφιλής γλώσσα είναι η C#, στην οποία και έχει υλοποιηθεί το πρόγραμμά μας.



Visual Studio Code Εικόνα 2.2.2β

2.2.2 Σχετικά με τη C#

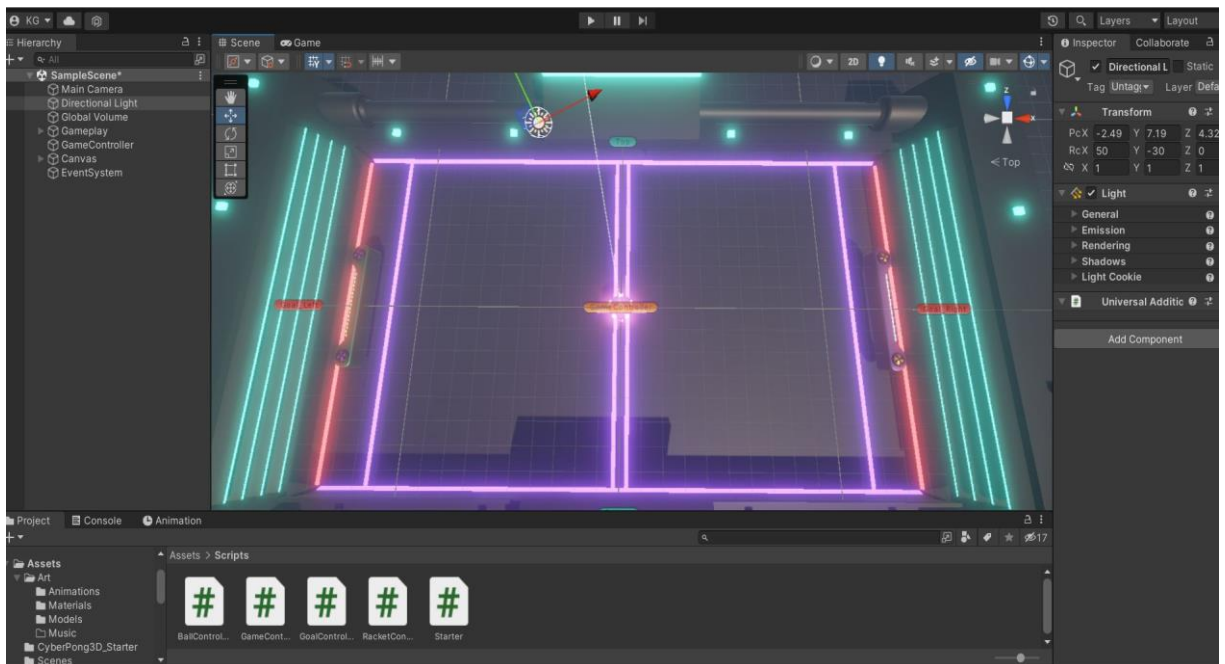
Η Unity προσφέρει ανάπτυξη scripted κώδικα χρησιμοποιώντας τη γλώσσα προγραμματισμού C#. Η C# είναι μια πολύ δημοφιλής αντικειμενοστραφής γλώσσα που σχεδιάστηκε και αναπτύχθηκε από τη Microsoft. Είναι πολύ παρόμοια με την Java και τη C++ στο ότι εστιάζει στη δημιουργία κλάσεων και μεθόδων, όπως συνήθως ορίζει ο αντικειμενοστραφής.

Η ευρεία επιλογή βιβλιοθηκών για κάθε πιθανή ανάγκη, σε συνδυασμό με μια απλοποιημένη δομή σε σύγκριση με τη C++, κάνει τη C++ πολύ ελκυστική. Για τους παραπάνω λόγους, η C# περιλαμβάνεται σε πολλά πανεπιστημιακά προγράμματα σπουδών και αντικειμενοστραφή μάθηση στοιχειώδους επιπέδου. Η Unity χρησιμοποιεί ειδικά το C# για να δημιουργήσει σενάρια που καθορίζουν τη λειτουργικότητα και τη συμπεριφορά των αντικειμένων παιχνιδιού ή εφαρμογής. Τα σενάρια επιτρέπουν στους συντάκτες εφαρμογών να δημιουργούν τα δικά τους στοιχεία από την αρχή που καθορίζουν τη συμπεριφορά ενός αντικειμένου με ιδιαίτερα περίπλοκους και εξατομικευμένους τρόπους. Εξαρτάται από τις φιλοδοξίες και τις δυνατότητες του προγραμματιστή. Η ίδια η Unity προσφέρει πολλές επιλογές για τη δόμηση περιεχομένου, αλλά η λογική του σεναρίου είναι διαφορετική και επιτρέπει στους προγραμματιστές να γράψουν τον δικό τους κώδικα C# και να είναι ελεύθεροι να κάνουν ό,τι θέλουν.

3. Ανάπτυξη και σχεδιασμός 3D Ping Pong

3.1 Εισαγωγή

Σύμφωνα με δημοσιεύματα το ring pong ήταν το πρώτο arcade video game που υπήρξε. Το πραγματικό Ring pong αποτελείται από ένα τραπέζι, μία μπάλα, ένα δίχτυ και δύο ρακέτες όπου κάθε παίκτης κρατάει από μία. Στην περίπτωση τη δική μας δεν θα χρειαστούμε το δίχτυ. Καθότι βλέπουμε το τραπέζι σε γωνία κάτοψης, το πλαίσιο της οθόνης αντικατοπτρίζει το περίγραμμα του τραπέζιου και τα πλαίσια στα οποία η μπάλα και οι ρακέτες μπορούν να κινούνται. Οι ρακέτες αναπαρίστανται σε μορφή κάθετων μπαρών όπου η κάθε μια αντιστοιχεί σε παίκτη.



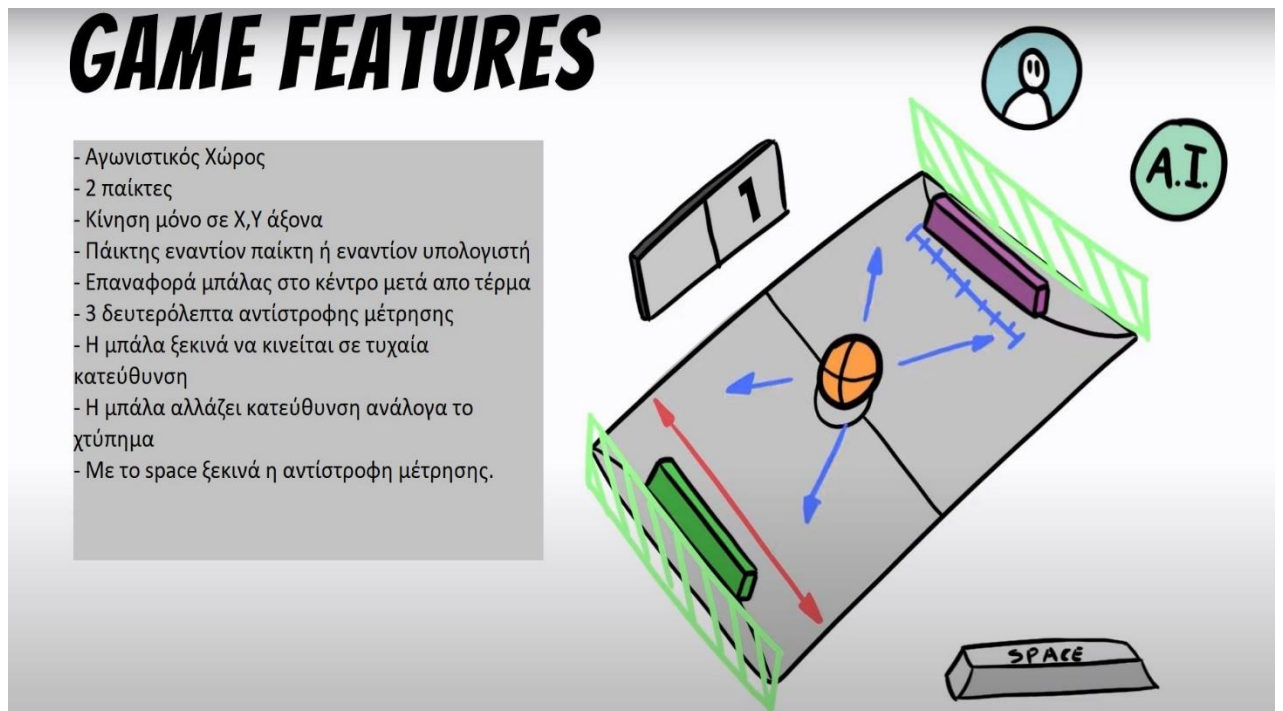
Εικόνα 3.1.1. Πλάνο παιχνιδιού

Το παιχνίδι έχει αναπτυχθεί σε windows pc με λογισμικό Windows 10 pro με specs 32gb ram, επεξεργαστή Ryzen 7 5800u (16cpus) και gpu AMD Radeon Graphics (™). Ένα ισχυρό μηχάνημα.

Η έκδοση της unity είναι 2021.3.12f1.

3.2 Στόχος & Τρόπος παιχνιδιού

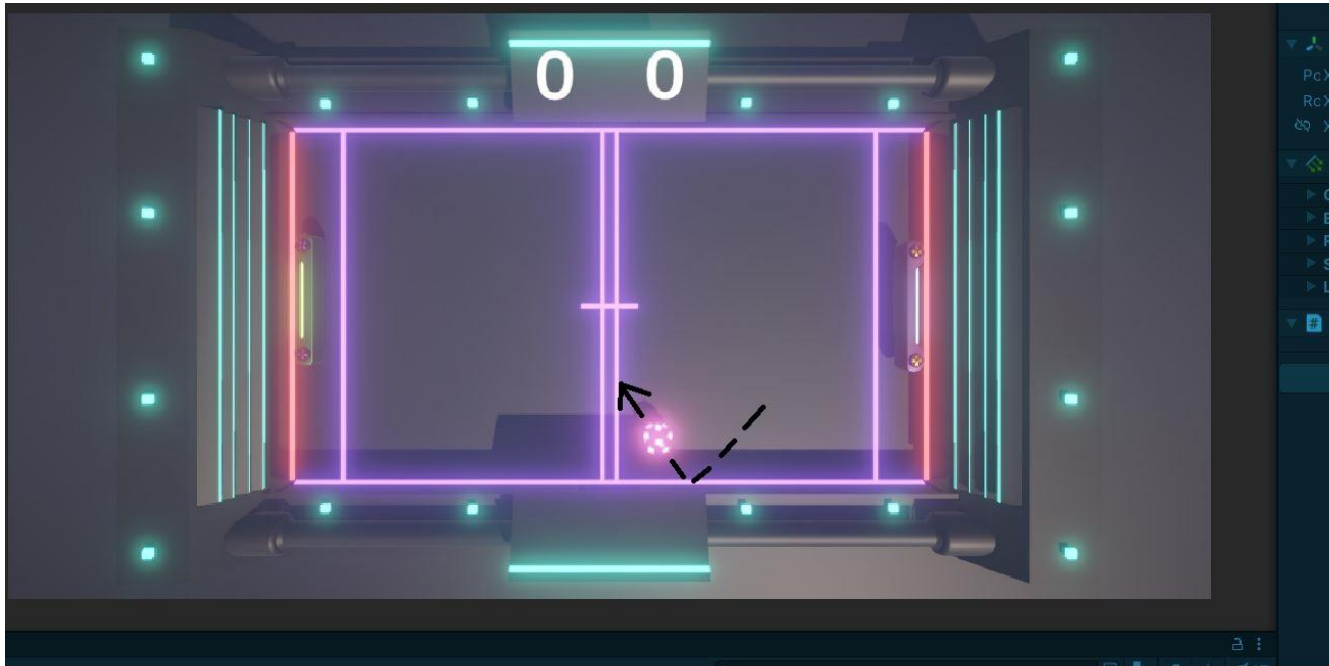
Στόχος του παιχνιδιού είναι οι παίκτες να χτυπήσουν την μπάλα με την ρακέτα τους. Σε περίπτωση που δεν καταφέρουν και περάσει η μπάλα θα δεχθούν γκολ και θα μετρήσει υπέρ του αντίπαλου παίκτη.



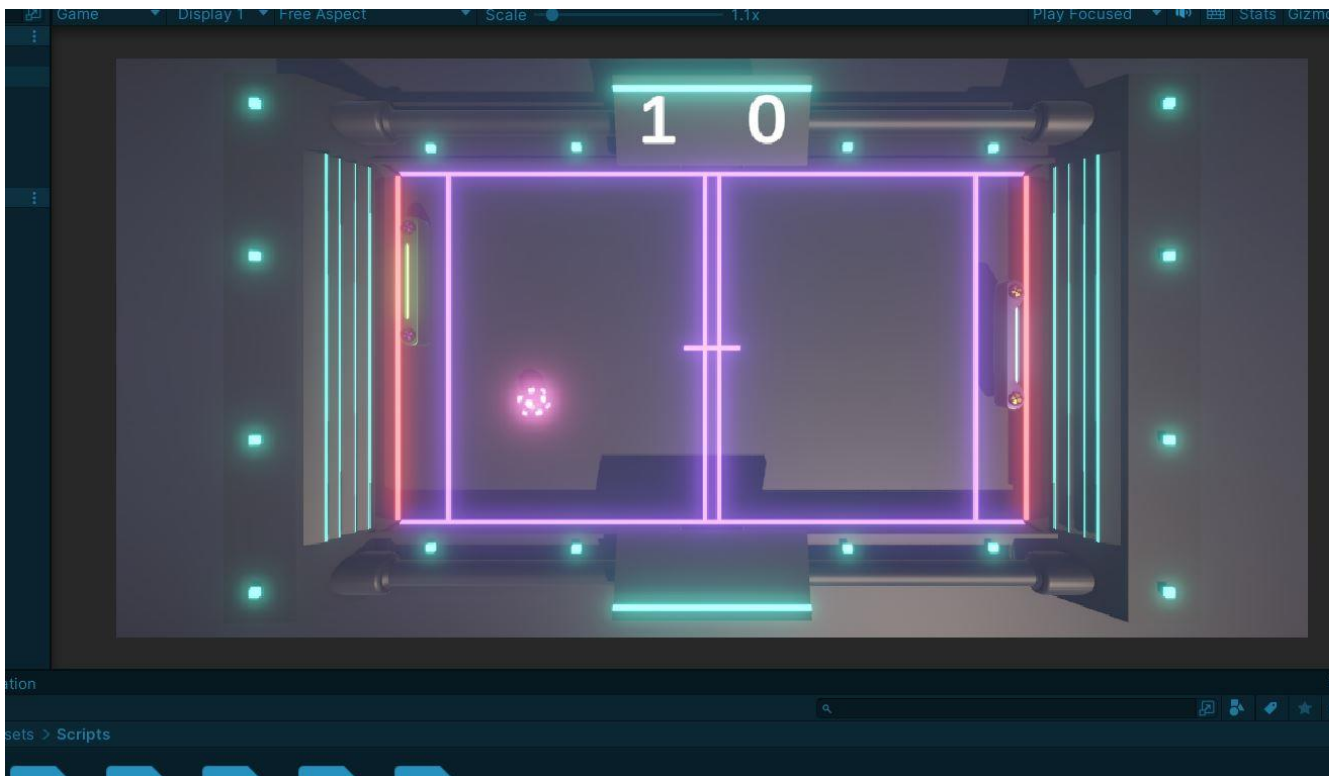
Εικόνα 3.2.1. Game features

Κατά την έναρξη του παιχνιδιού, οι παίκτες πρέπει να πατήσουν Space για να ξεκινήσει η αντίστροφη μέτρηση για την κίνηση της μπάλας σε τυχαία τροχιά. Μόλις χτυπήσει την μπάλα κάποιος παίκτης με τη ρακέτα του αυτή θα αρχίσει να κινείται ανάλογα με το σημείο της ρακέτας που χτύπησε ο παίκτης τη μπάλα. Έτσι όχι μόνο βγάζουν άμυνα αλλά μπορούν να δώσουν μια διαφορετική κατεύθυνση στη μπάλα η οποία θα δυσκολέψει τον αντίπαλο να αμυνθεί και να δεχθεί το γκολ. Μόλις ένας παίκτης δεχθεί ένα τέρμα η μπάλα επιστρέφει στο κέντρο του γηπέδου και ξεκινά να κινείται ξανά σε τυχαία τροχιά.

Η μπάλα μπορεί να κινείται σε κάθετο(Υ) και οριζόντιο(Χ) άξονα. Μόλις προσκρούσει σε κάποιο τοίχωμα του γηπέδου αλλάζει φορά ως προς τον Υ άξονα , μόλις χτυπήσει στη ρακέτα ενός παίκτη, η μπάλα θα αλλάξει τροχιά ως προς τον Χ άξονα και ίσως ως προς τον Υ ανάλογα το σημείο πρόσκρουσης της ρακέτας.



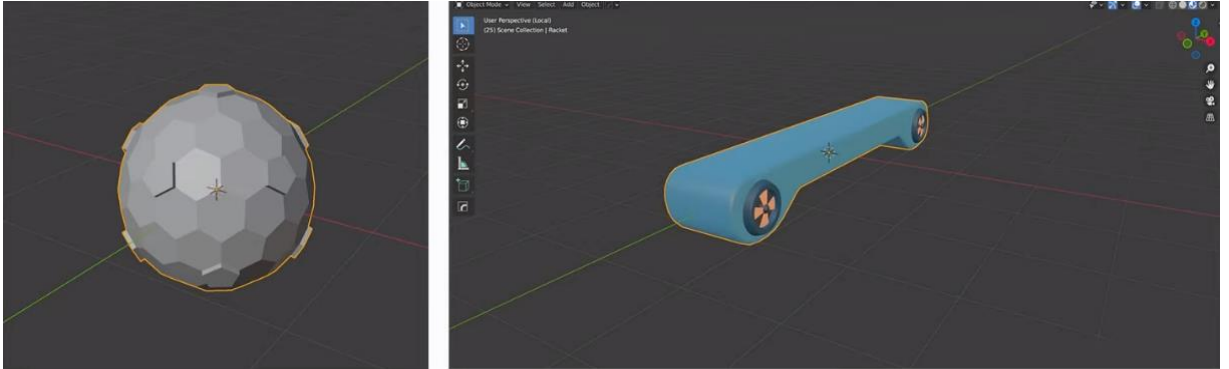
Εικόνα 3.2.2. Κίνηση μπάλας, αλλαγή πορείας.



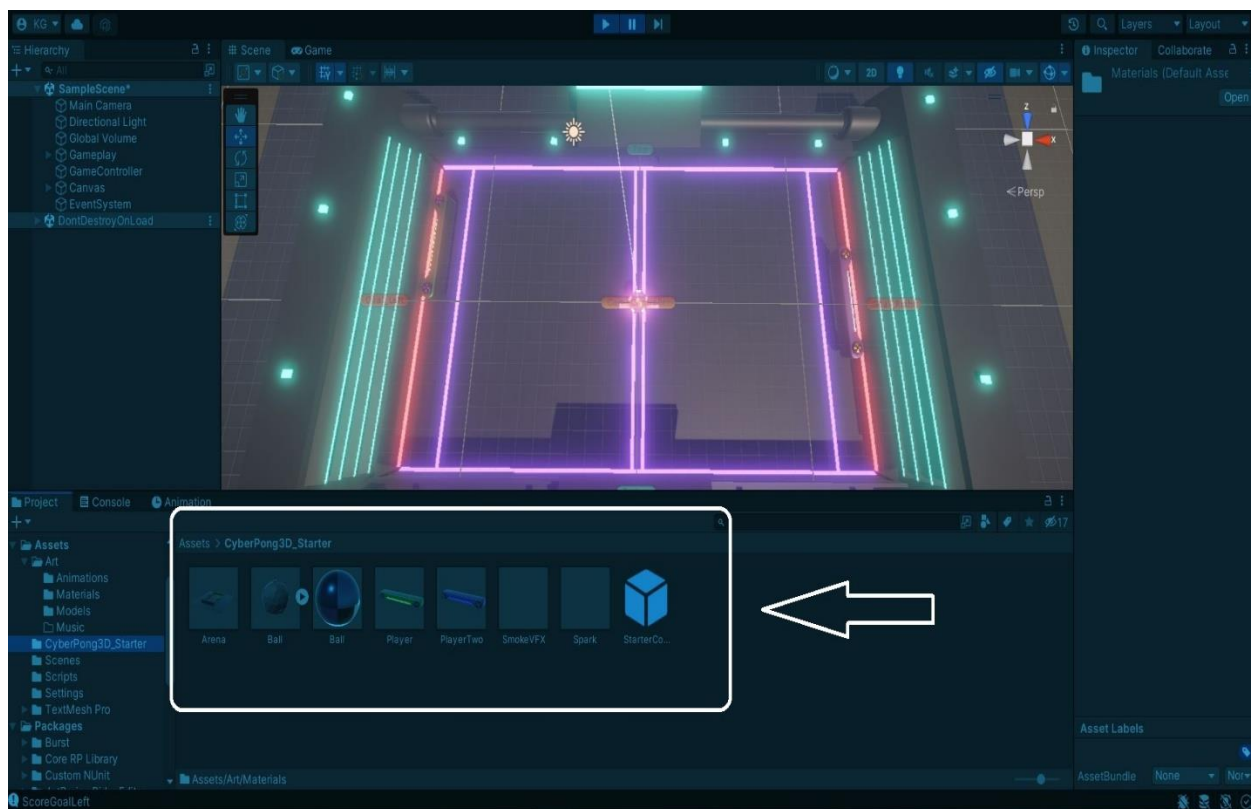
Εικόνα 3.2.3. Ο παίκτης με την πράσινη μπάλα προηγείται με 1-0.

3.3 Κατασκευή περιβάλλοντος - Σταδίου

Σε πρώτο στάδιο θα χρειαστεί να κατασκευάσουμε το παιχνίδι μας στο περιβάλλον της Unity. Τα αντικείμενα που θα χρησιμοποιήσουμε είναι το στάδιο μας που αντιπροσωπεύει το τραπέζι, οι μπάρες θα αντιπροσωπεύουν τις ρακέτες και η μπάλα. Σε συνεργασία με 3D Designer αποκτήσαμε αυτά τα Assets και τα κάναμε import στην unity.



Εικόνα 3.3.1. Μπάλα και Ρακέτα.



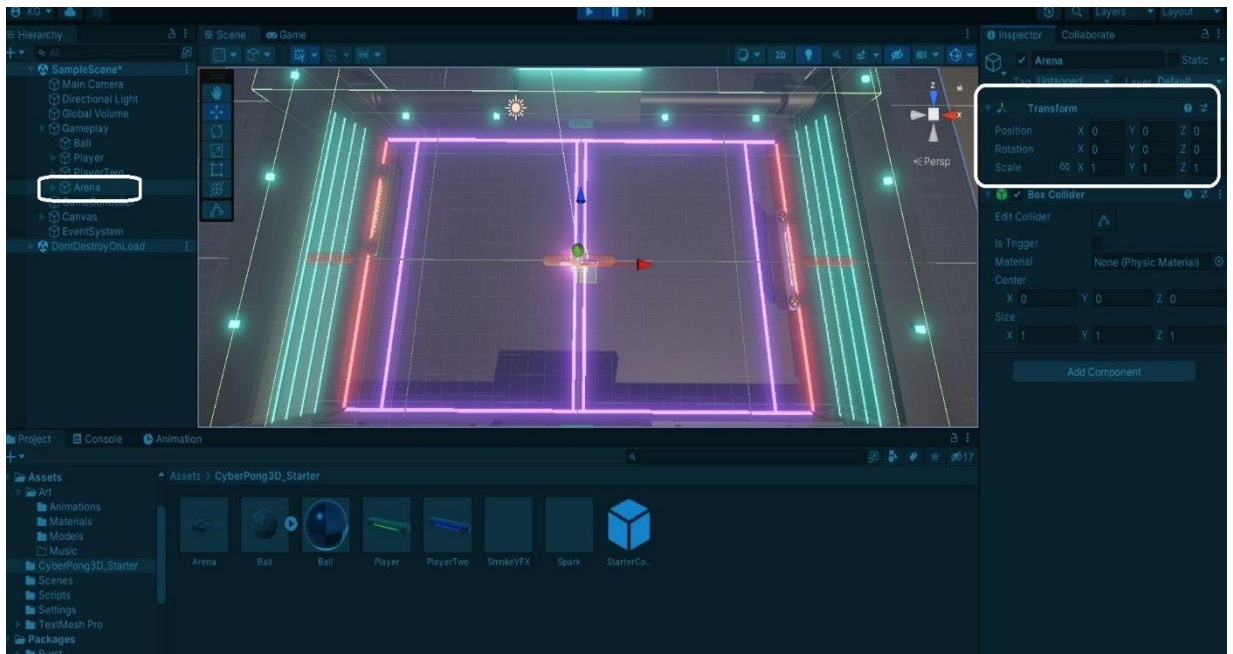
Εικόνα 3.3.2. Assets που χρησιμοποιήθηκαν.

Αφού κάναμε import τα assets τα οποία θα χρησιμοποιήσουμε το επόμενο βήμα που έπρεπε να κάνουμε είναι να τα τοποθετήσουμε στις θέσεις που χρειαζόμαστε με τη φορά που θέλουμε. Για να το πετύχουμε αυτό θα πρέπει να επιλέξουμε και να σύρουμε στη σκηνή το κάθε ένα.

3.3.1 Τοποθετήσεις

Τοποθέτηση αρένας

Εφόσον η αρένα αποτελεί το τραπέζι μας και θα πιάνει σχεδόν όλη την οθόνη μας θα την τοποθετήσουμε στο κέντρο. Για να την τοποθετήσουμε στο κέντρο θα την επιλέξουμε αριστερά από την ιεραρχία και στην συνέχεια από τον επιθεωρητή θα ορίσουμε μηδενικές τιμές στο position στα πεδία X,Y,Z οι οποίοι είναι οι άξονες μας καθώς δουλεύουμε σε 3D περιβάλλον. Ένας άλλος τρόπος είναι από το Transform να επιλέξουμε τις 3 τελείες και να κάνουμε reset. Με αυτό τον τρόπο θα επαναφέρει το αντικείμενο στις μηδενικές τιμές των αξόνων.



Εικόνα 3.3.1.1. Τοποθέτηση αρένας στο κέντρο. X,Y,Z μηδενικές τιμές.

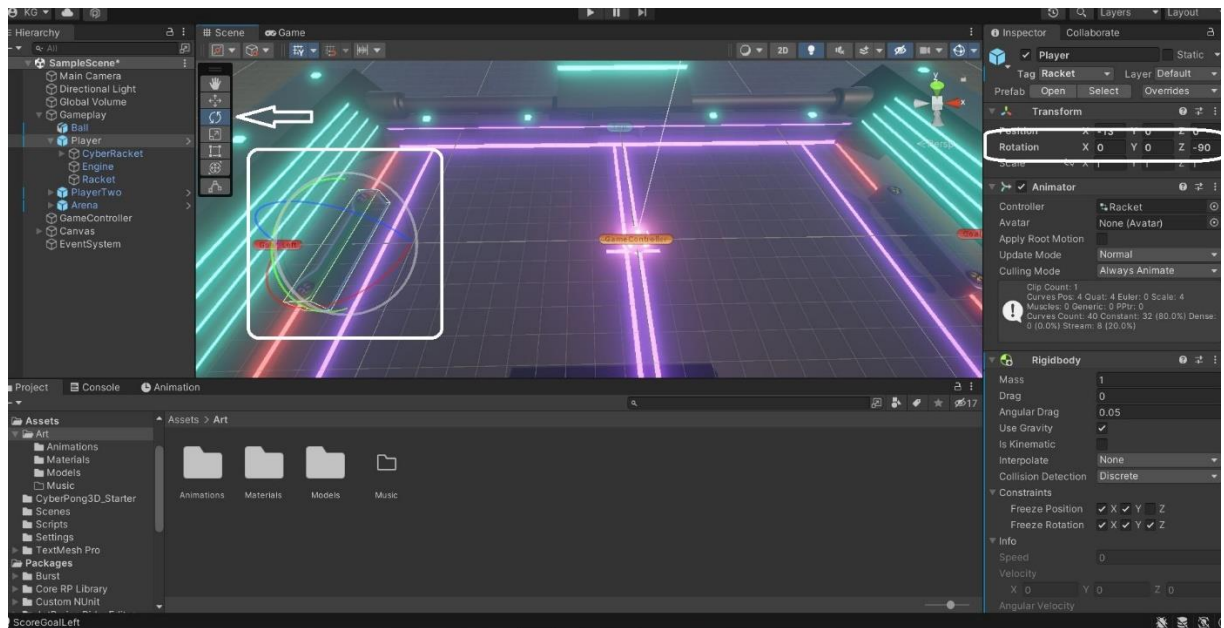
Ένας εναλλακτικός τρόπος για την μετατόπιση των αντικειμένων είναι χειροκίνητα από την μάσκα της σκηνής. Εφόσον επιλέξουμε από την ιεραρχία το αντικείμενο που θέλουμε, επιλέγουμε το σήμα με τα βέλακια από την μπάρα αριστερά στην σκηνή, μετά πάνω από το αντικείμενο εμφανίζονται 3 βελάκια που υποδεικνύουν τον άξονα στον οποίο θέλουμε να μετακινήσουμε το αντικείμενο. Αυτό βέβαια δεν αποτελεί και την πιο αποτελεσματική μέθοδο καθώς δεν μας δίνει και την δυνατότητα της ακρίβειας.



Εικόνα3.3.1.2. Τοποθέτηση asset με βελάκια

Τοποθέτηση Ρακετών

Για την τοποθέτηση των ρακετών θα κινηθούμε με τον ίδιο τρόπο. Τον αριστερό παίκτη θα τον τοποθετήσουμε στον X σε τιμή -13 και τον $+13$ ώστε να απέχουν το ίδιο από το κέντρο. Ακόμα θα πρέπει να περιστρέψουμε τους παίκτες ως προς τις μοίρες ώστε το πάνω μέρος των ρακετών να κοιτάει προς τον αντίπαλο καθώς θα είναι η επιφάνεια που θα προσκρούεται η μπάλα. Για την περιστροφή των αντικειμένων στα πεδία Rotation θα βάλουμε τιμές 90° και -90° ως προς τον άξονα Z. Εδώ πάλι μπορούμε να κάνουμε χειροκίνητα την περιστροφή αν επιλέξουμε το rotate tool από την σκηνή.



Εικόνα 3.3.1.2. Περιστροφή ρακετών.

Τοποθέτηση μπάλας και κάμερας

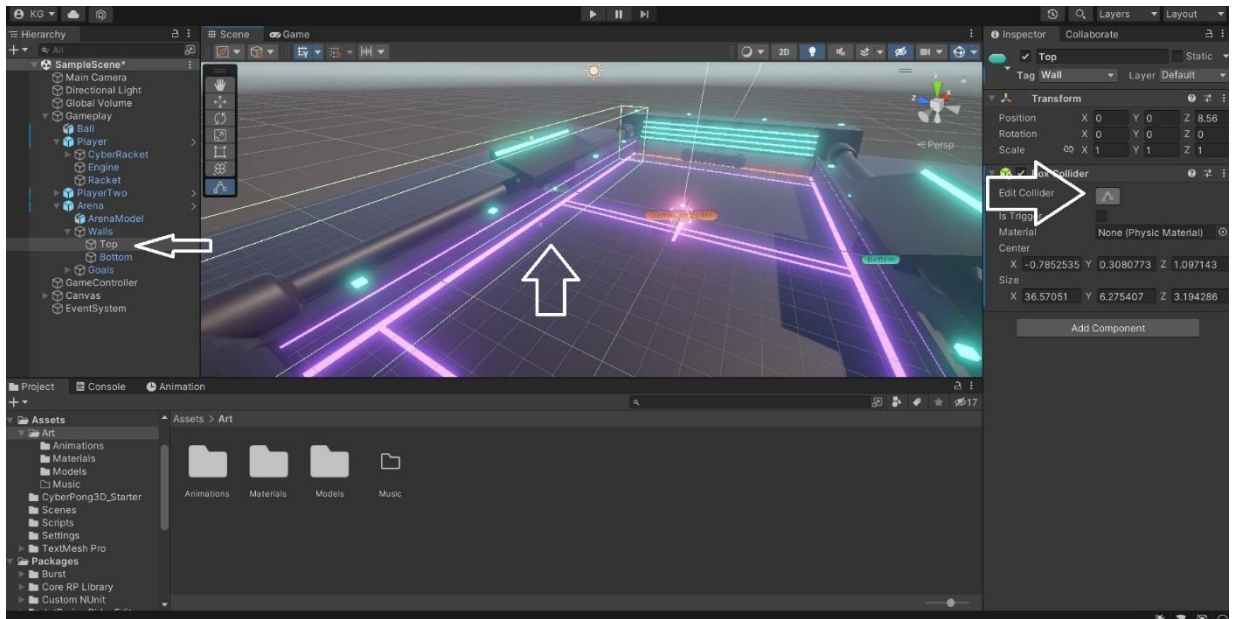
Τέλος τοποθετούμε την μπάλα στην θέση (0,0,0) και την κάμερα στην θέση (0,22,0) με περιστροφή (0,90,0) για να βλέπουμε την κάτοψη της αρένας.

3.3.2 Αλληλεπίδραση αντικειμένων

Εφόσον τοποθετήσουμε όλα μας τα αντικείμενα στην σκηνή το επόμενο βήμα είναι να κάνουμε τα αντικείμενα να αλληλοεπιδρούν μεταξύ τους. Σε διαφορετική περίπτωση τα αντικείμενα θα περνάνε το ένα μέσα από το άλλο. Για παράδειγμα η μπάλα θα περνάει μέσα από τις ρακέτες και τους τοίχους, οι ρακέτες θα περνάνε μέσα από τους τοίχους. Αυτό μπορεί να αντιμετωπιστεί τοποθετώντας box colliders στα objects, πιο συγκεκριμένα στα τοιχώματα της αρένας, στις ρακέτες αλλά και στην μπάλα.

Τοιχώματα Αρένας

Κάνοντας expand την arena, θα δούμε ότι υπάρχουν τοίχοι top & bottom. Θα πρέπει να προστεθούν και στα δύο box colliders. Για να προστεθεί ένα box collider θα πρέπει να επιλέξουμε Add Component και να βρούμε το “Box Collider” στο Inspector του Object. Αφού το προσθέσουμε στην συνέχεια επιλέγουμε το εικονίδιο που υπάρχει στο Edit Collider. Αυτό θα μας δώσει την δυνατότητα να ορίσουμε τα πλαίσια στα οποία θα αλληλοεπιδρά το αντικείμενο. Το ορίζουμε στα όρια του ως object. Στην ουσία ορίζουμε τα όρια.

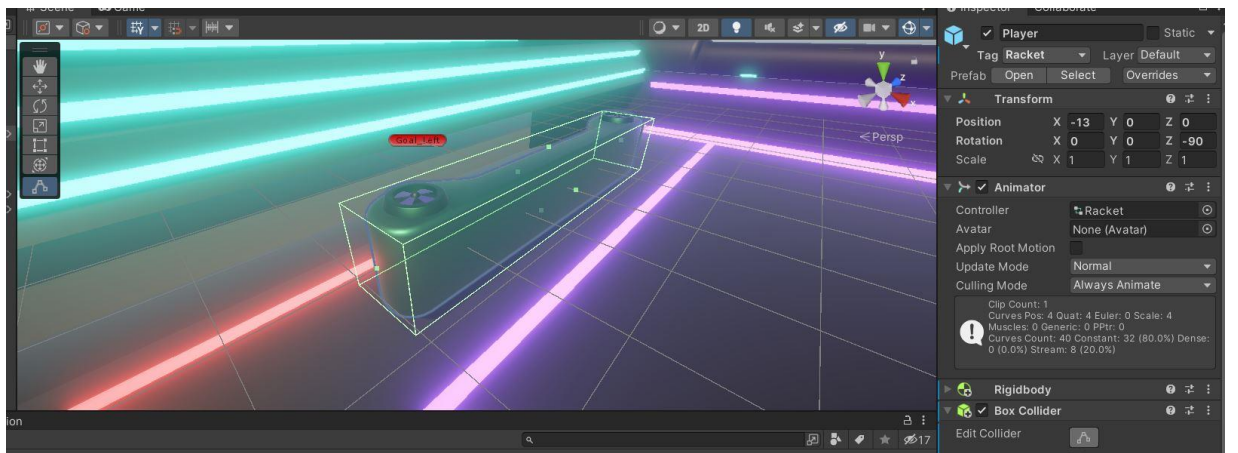


Εικόνα 3.3.2.1. Οριοθέτηση τοίχου – Box collider

Εδώ μπορούμε πάλι είτε από τις τελείες του box collider είτε από τον Inspector να ορίσουμε το μέγεθος του.

Τοιχώματα Ρακετών

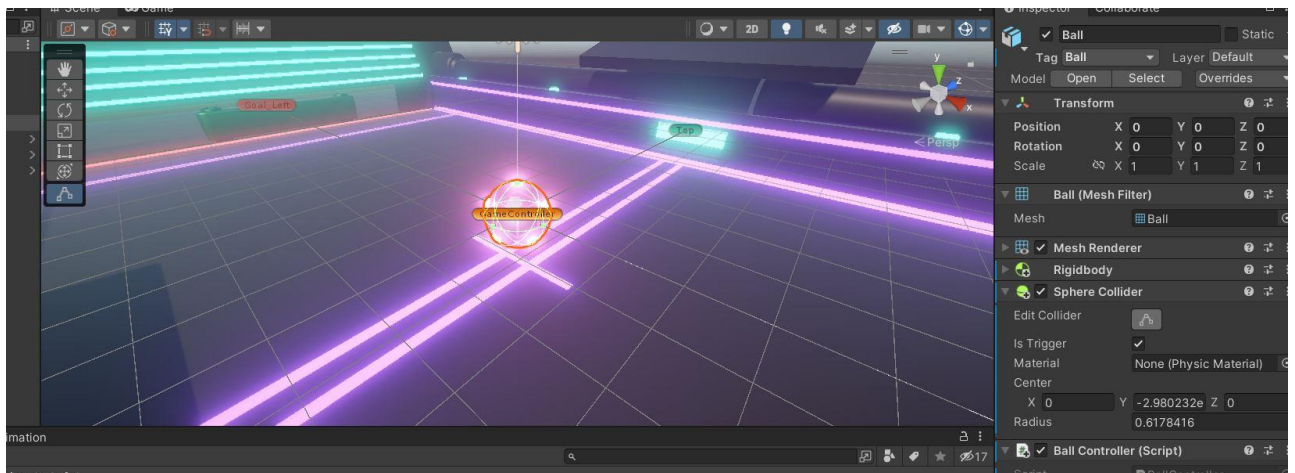
Με τον ίδιο ακριβώς τρόπο θα λειτουργήσουμε και για τις ρακέτες των παικτών.



Εικόνα 3.3.2.2. Οριοθέτηση ρακέτας – Box collider

Τοιχώματα Μπάλας

Στην περίπτωση της μπάλας αυτό που αλλάζει είναι ότι προσθέτουμε sphere collider αντί για box collider καθώς δεν πρόκειται για κουτί αλλά μία σφαίρα.

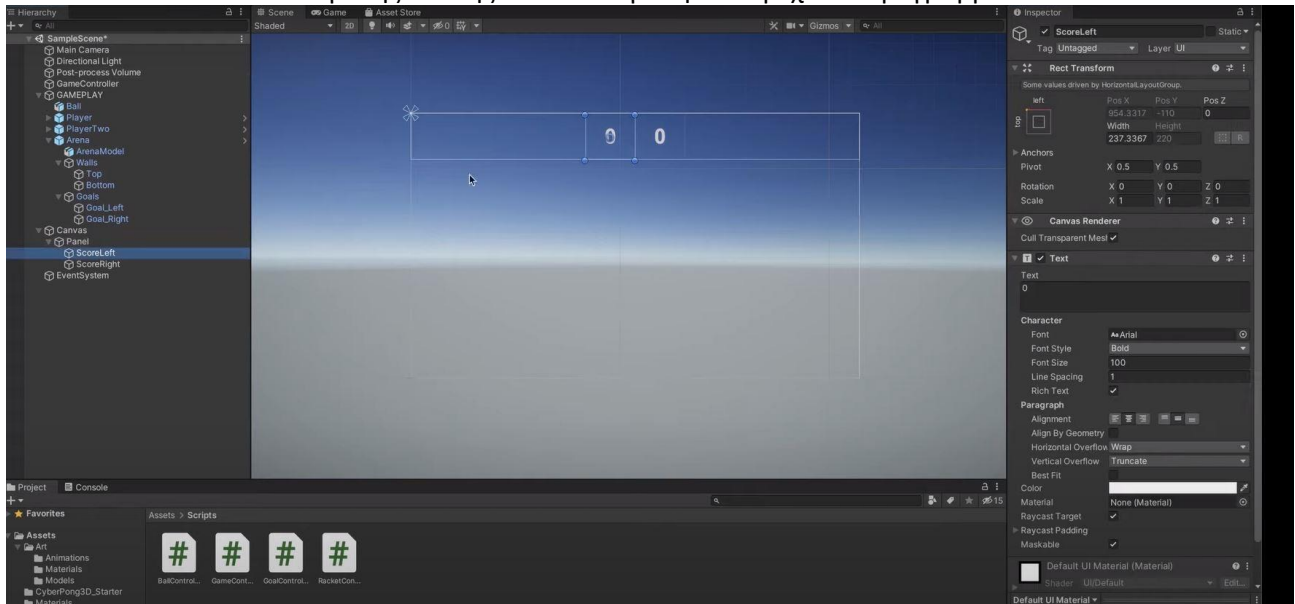


Εικόνα 3.3.2.3. Οριοθέτηση μπάλας – Sphere collider

3.3.3 Score UI

Φυσικά δεν θα μπορούσαμε να έχουμε ένα παιχνίδι με τέρματα χωρίς να έχουμε και πίνακα σκορ. Ο πίνακας που θα προσθέσουμε θα αποτελείται από δύο text πάνω σε ένα panel (canvas) και κάθε text θα αντιπροσωπεύει τα τέρματα που έχει σημειώσει κάθε παίκτης.

Αυτό θα το πετύχουμε προσθέτοντας ένα panel και διαχωρίζοντας το στη μέση με τα δυο text εκατέρωθεν. Ακόμα θα κάνουμε κάποιες ρυθμίσεις προκειμένου το scoreboard να είναι ανάλογο της οθόνης του υπολογιστή που τρέχει το πρόγραμμα.

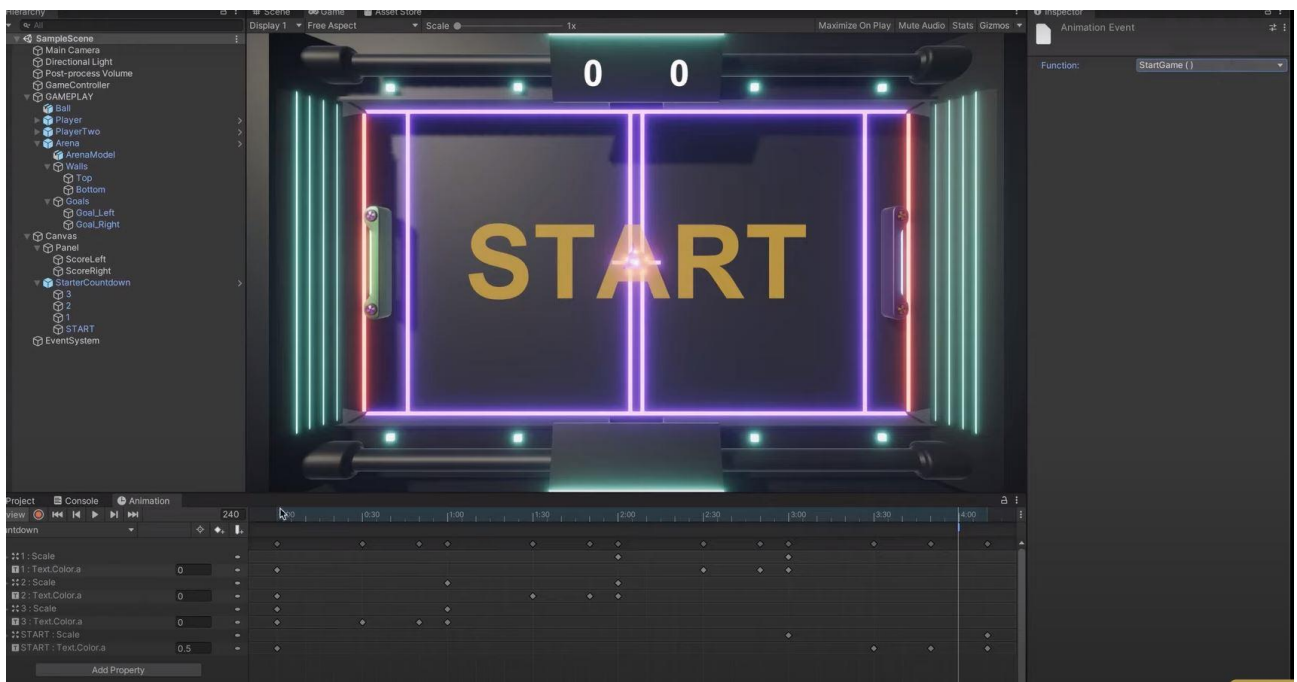


Εικόνα 3.3.3.1 Το πάνελ του scoreboard με τα δύο text.

3.3.4 Start Countdown

Κατά την έναρξη του γύρου οι παίκτες πρέπει να είναι προετοιμασμένοι και να γνωρίζουν πότε πρόκειται να κινηθεί η μπάλα, για αυτό το λόγο προστέθηκε ένα countdown μήνυμα όπου θα ενεργοποιείται κατά το πάτημα του Space. Η αντίστροφη μέτρηση θα 'ναι σύντομη και στη συνέχεια θα εμφανίζεται το μήνυμα Start όπου και θα σηματοδοτεί την έναρξη κινήσεως της μπάλας.

Θα κάνουμε import του asset καθώς το έχουμε ήδη έτοιμο. Το animation του countdown διαρκεί λίγο περισσότερο από 4 δευτέρα όπου σε αυτά γίνεται η αντίστροφη μέτρηση και στη συνέχεια εμφανίζεται το Start. Στο animation θα ορίσουμε ότι κοντά στα 3.90 δευτερόλεπτα θα δίνεται η εντολή για να ξεκινήσει η μπάλα να κινείται.



Εικόνα 3.3.4.1 Μήνυμα ειδοποίησης εκκίνησης παιχνιδιού.

3.4 Scripting & Ενσωμάτωση σε objects

3.4.1 Γενικά

Όλα αυτά που δημιουργήσαμε δεν θα είχαν κανένα απολύτως νόημα αν δεν συνοδεύονταν από έναν κώδικα. Στη δική μας περίπτωση, όπως έχουμε προαναφέρει παραπάνω, με την βοήθεια της C# μπορούμε και δίνουμε ζωή στα objects που έχουμε προσθέσει στη σκηνή μας και επιπλέον έχουμε σημαντικές ενέργειες οι οποίες είναι απαραίτητες για την λειτουργία του παιχνιδιού μας.

3.4.2 Scripting στη Unity

Αρχικά να αναφέρουμε λίγα λόγια για το πως λειτουργεί ένα script στη Unity. Σε ένα script μπορούμε να έχουμε Classes τα οποία μπορούν να εκτελούν κάποιες ενέργειες κατά την έναρξη του παιχνιδιού, ή να γίνεται frame by frame ή να ενεργοποιούνται μόλις προηγηθεί ένα άλλο γεγονός.

3.4.3 Βιβλιοθήκες

Στη C# (όπως και σε πολλές άλλες γλώσσες προγραμματισμού) χρησιμοποιούμε βιβλιοθήκες οι οποίες έχουν λειτουργίες που μας βοηθήσουν να πετύχουμε το αποτέλεσμα που επιθυμούμε. Επίσης με αυτές μπορούμε να γλιτώσουμε χρόνο και χώρο καθώς κώδικα που θα χρειαζόταν να γράψουμε για μία ενέργεια μέσα στο παιχνίδι μας, μπορούμε να την έχουμε με την επίκληση μιας μεθόδου ή συνάρτησης. Τις βιβλιοθήκες τις κάνουμε import στην κορυφή του script διαφορετικά ότι προηγείται από αυτές δεν θα είναι λειτουργικό αλλά και για λόγους καλύτερης οργάνωσης του κώδικα μας.

3.4.4 Ενσωμάτωση των scripts στα objects

Τα scripts θα πρέπει να ενσωματώνονται στα object προκειμένου να έχουν επιρροή πάνω του. Επιλέγοντας το κατάλληλο script, μπορούμε να το σύρουμε στον inspector του object που επιθυμούμε εφόσον έχουμε επιλέξει πρώτα το object από την ιεραρχία ή την σκηνή.

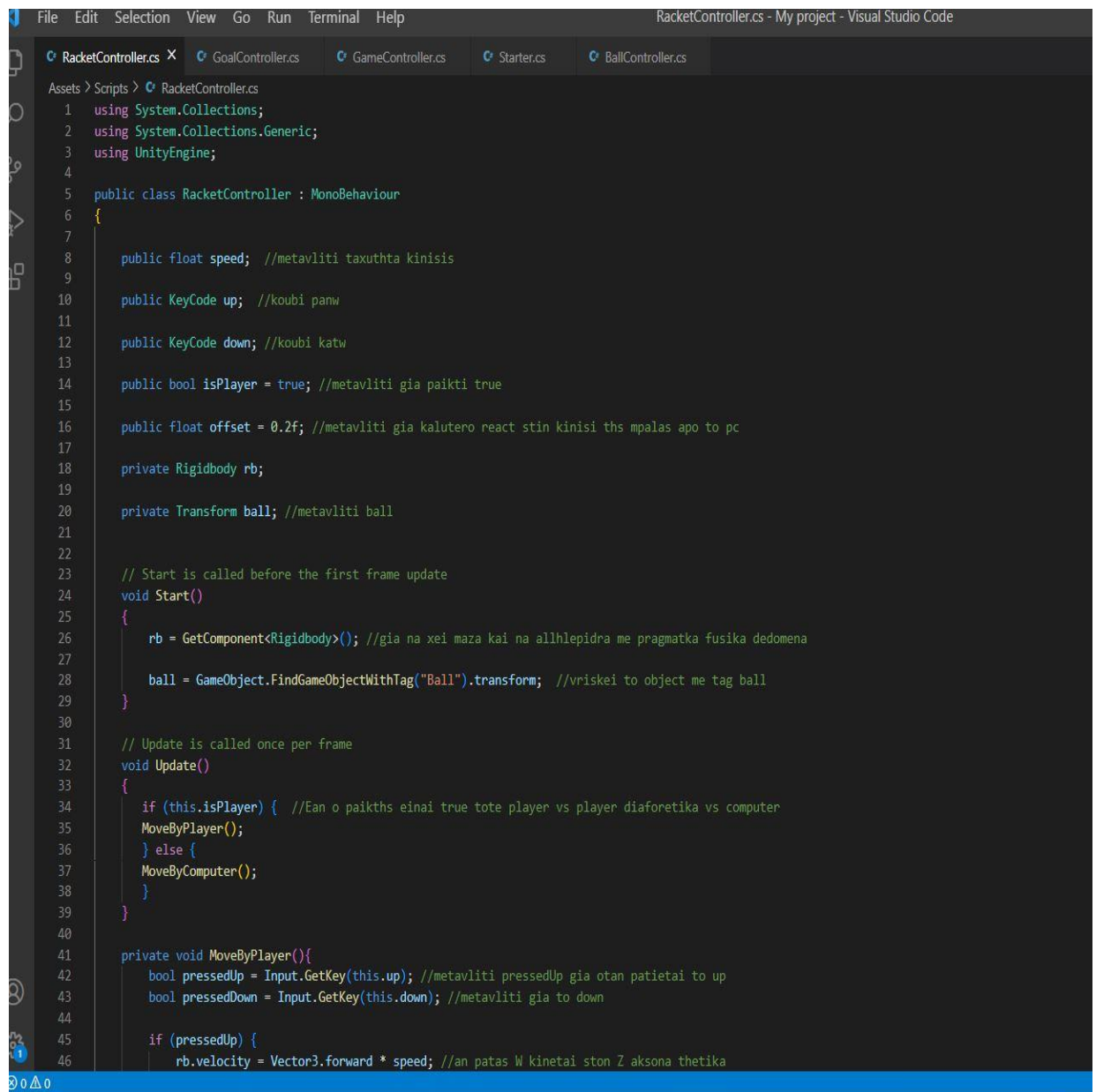
3.4.5 Script των objects

Σε αυτή την ενότητα θα αναλύσουμε τον κώδικα και τα scripts που δώσαμε στα objects Racket, ball, Game controller, Goal και Starter UI animation.

Racket

Το πρώτο πράγμα που έπρεπε να κάνουμε ήταν να δημιουργήσουμε ένα script το οποίο θα δέναμε στα rackets.

Μέσα στο script θα ορίσουμε την κίνηση της ρακέτας (πάνω, κάτω) για τους παίκτες, θα ορίσουμε την κίνηση της ρακέτας όταν την χειρίζεται ο υπολογιστής.



```
File Edit Selection View Go Run Terminal Help RacketController.cs - My project - Visual Studio Code
RacketController.cs X GoalController.cs GameController.cs Starter.cs BallController.cs
Assets > Scripts > RacketController.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class RacketController : MonoBehaviour
6 {
7
8     public float speed; //metavliti taxuthta kinisis
9
10    public KeyCode up; //koubi panw
11
12    public KeyCode down; //koubi katw
13
14    public bool isPlayer = true; //metavliti gia paikti true
15
16    public float offset = 0.2f; //metavliti gia kalutero react stin kinisi ths mpalas apo to pc
17
18    private Rigidbody rb;
19
20    private Transform ball; //metavliti ball
21
22
23    // Start is called before the first frame update
24    void Start()
25    {
26        rb = GetComponent<Rigidbody>(); //gia na xei maza kai na allhlepidra me pragmatka fusika dedomena
27
28        ball = GameObject.FindGameObjectWithTag("Ball").transform; //vriskei to object me tag ball
29    }
30
31    // Update is called once per frame
32    void Update()
33    {
34        if (this.isPlayer) { //Ean o paikths einai true tote player vs player diaforetika vs computer
35            MoveByPlayer();
36        } else {
37            MoveByComputer();
38        }
39    }
40
41    private void MoveByPlayer(){
42        bool pressedUp = Input.GetKey(this.up); //metavliti pressedUp gia otan patietai to up
43        bool pressedDown = Input.GetKey(this.down); //metavliti gia to down
44
45        if (pressedUp) {
46            rb.velocity = Vector3.forward * speed; //an patas W kinetai ston Z aksona thetika
```

```

if (this.isPlayer) { //Εαν ο παίκτης είναι true τότε player vs player διαφορετικά vs computer
    MoveByPlayer();
} else {
    MoveByComputer();
}
}

private void MoveByPlayer(){
    bool pressedUp = Input.GetKey(this.up); //metavliti pressedUp gia otan patietai to up
    bool pressedDown = Input.GetKey(this.down); //metavliti gia to down

    if (pressedUp) {
        rb.velocity = Vector3.forward * speed; //an patas W kinetai ston Z aksona thetika
    }

    if (pressedDown) {
        rb.velocity = Vector3.back * speed; //an patas S kinetai ston Z aksona arnhatika
    }

    if (!pressedUp && !pressedDown) { //an den patas kanena apo ta 2
        rb.velocity = Vector3.zero; } //diaforetika tpt
}

private void MoveByComputer(){ //O παίκτης computer κινείται vasei tn kinisi ths mpalas ston aksona Z
    if (ball.position.z > transform.position.z + offset) { //an h mpala einai se megalutero Z apo tn raketa tote proxwraei h raketa pros thn mpala
        rb.velocity = Vector3.forward * speed; //forward proxwra thetika ston Z aksona
    } else if (ball.position.z < transform.position.z - offset){ //an h mpala einai se mikrotero z tote kineitai antistoixa
        rb.velocity = Vector3.back * speed; //proxwra arnhatika
    } else {
        rb.velocity = Vector3.zero; //dn kineitai an einai sto idio shmeio ws pron Z
    }
}
}

```

Εικόνες 3.4.5.1 & 3.4.5.2 Το script των ρακετών με επεξηγήσεις σαν σχόλια

Ball

Για την μπάλα έπρεπε να δώσουμε την τυχαία κατεύθυνσης κίνηση της κατά την εκκίνηση, την συμπεριφορά της όταν χτυπά στα άλλα objects καθώς και τι συμβαίνει μόλις υπάρξει κάποιο τέρμα όπου επιστρέφει στην αρχική της θέση. Σημείωση : Στα τέρμα έχουν δημιουργήσει ένα collider που μόλις η μπάλα συναντηθεί με αυτό (δηλαδή γκολ) γίνεται το reset position και αλλάζει το σκορ.


```
File Edit Selection View GO Run Terminal Help
BallController.cs my project Visual Studio Code
RacketController.cs GoalController.cs GameController.cs Starter.cs BallController.cs x
Assets > Scripts > BallController.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class BallController : MonoBehaviour
6 {
7
8     public float speed; //metavliti taxuthtas
9
10    public float minDirection = 0.5f;
11
12    private Vector3 direction; //orizome neo position sthn mpala
13    private Rigidbody rb; //orizome rigidbody
14
15    private bool stopped = true; //metavliti stopped na einai true
16
17    // Start is called before the first frame update
18    void Start()
19    {
20        this.rb = GetComponent<Rigidbody>(); //denoume to rigidbody
21    }
22
23
24    // Update is called once per frame
25    void Update()
26    {
27        //Method 1
28        //transform.position += direction * speed * Time.deltaTime;
29    }
30
31    void FixedUpdate()
32    {
33        if (stopped)
34            return;
35        rb.MovePosition(rb.position + direction * speed * Time.fixedDeltaTime); //kinisi mpalas
36    }
37
38    private void OnTriggerEnter(Collider other) //trigger molis sugkroustei me allo object
39    {
40        if (other.CompareTag("Wall")){ //an vriskei se object me tag wall
41            direction.z = -direction.z; //allazei kateuthinsi pros ton z
42        }
43
44        if (other.CompareTag("Racket")){ //an vriskei se object me tag racket
45            Vector3 newDirection = (transform.position - other.transform.position).normalized; //new direction alla to megisto einai 1
46        }
47    }
48 }
```

```

31 void FixedUpdate()
32 {
33     if (stopped)
34         return;
35     rb.MovePosition(rb.position + direction * speed * Time.fixedDeltaTime); //kinisi mpalas
36 }
37
38 private void OnTriggerEnter(Collider other) //trigger molis sugkroustei me allo object
39 {
40     if (other.CompareTag("Wall")) { //an vriskei se object me tag wall
41         direction.z = -direction.z; //allazei kateuthinsi pros ton z
42     }
43
44     if (other.CompareTag("Racket")) { //an vriskei se object me tag racket
45         Vector3 newDirection = (transform.position - other.transform.position).normalized; //new direction alla to megisto einai 1
46
47         newDirection.x = Mathf.Sign(newDirection.x) * Mathf.Max(Mathf.Abs(newDirection.x), this.minDirection); //allazei h kateuthinsi analoga pou xtuapai h raketa
48         newDirection.z = Mathf.Sign(newDirection.z) * Mathf.Max(Mathf.Abs(newDirection.z), this.minDirection); //to idio
49
50         direction = newDirection; //pairnei nea kateuthinsi
51     }
52 }
53
54 private void ChooseDirection() { //na ksekinai me random direction
55     float signX = Mathf.Sign(Random.Range(-1f, 1f)); //fernei mia tuxaia float timi apo to -1 ews 1 gia ton aksona X
56     float signZ = Mathf.Sign(Random.Range(-1f, 1f)); //to idio alla gia ton aksona Z
57     this.direction = new Vector3(0.5f * signX, 0, 0.5f * signZ); //random kinisi vasei twm signx, signz
58 }
59
60 public void Stop() {
61     this.stopped = true; //h mpala einai stamathenh
62 }
63
64 public void Go() { //h mpala ksekinai
65     ChooseDirection(); //epilegei nea random kateuthinsi
66     this.stopped = false; //h metavliti stopped ginetai false
67 }
68 }
69
70

```

Εικόνες 3.4.5.3 & 3.4.5.4 Script μπάλας με επεξηγήσεις σαν σχόλια

Game Controller

Το κυριότερο script είναι ο game controller, το οποίο υπάρχει και σαν object αλλά δεν είναι φανερό κάπου. Αποτελεί το κύριο γρανάζι του παιχνιδιού μας καθώς συνδέει όλα τα scripts μεταξύ τους για τις επιμέρους λειτουργίες (αύξηση σκορ, reset μπάλας, countdown animation κλπ.)

```
Assets > Scripts > GameController.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using TMPro;
6
7
8
9  public class GameController : MonoBehaviour
10 {
11     public GameObject ball; //reference sti mpala
12
13     public TextMeshProUGUI scoreTextLeft; //metavlites twm score
14     public TextMeshProUGUI scoreTextRight;
15
16     public Starter starter; //reference sto starter script
17
18     private bool started = false; //metavliti gia tin arxi tou game
19
20     private int scoreLeft = 0; //metavliti gia to score toy paikth aristera
21     private int scoreRight = 0; //metavliti gia to score toy paikth deksia
22
23     private BallController ballController; //reference sto script BallController
24
25     private Vector3 startingPosition; //metavliti gia otan ginetai reset na fernei sto miden
26
27     // Start is called before the first frame update
28     void Start()
29     {
30         this.ballController = this.ball.GetComponent<BallController>();
31         this.startingPosition = this.ball.transform.position;
32     }
33
34     // Update is called once per frame
35     void Update()
36     {
37         if ( this.started ) //an h mpala ksekinise mhn kanei kati
38             return;
39
40         if (Input.GetKey(KeyCode.Space)) { //an pathsoume space
41             this.started = true; //ksekinaei to paixnidi
42             this.starter.StartCountdown(); //ksekinaei to countdown
43         }
44     }
45
46
```

```

Assets > Scripts > GameController.cs
41     this.started = true; //ksekinaei to paixnidi
42     this.starter.StartCountdown(); //ksekinaei to countdown
43 }
44 }
45
46
47 public void StartGame(){
48     this.ballController.Go(); //ksekinaei to script ths mpalas
49 }
50
51
52 public void ScoreGoalLeft() { //method molis scorarei o deksia paikths
53     Debug.Log("ScoreGoalLeft");
54     this.scoreRight +=1; //auksanei to score tou deksia
55     UpdateUI();
56     Resetball();
57 }
58
59 public void ScoreGoalRight() { //method molis scorarei o aristera paikths
60     Debug.Log("ScoreGoalRight"); //emfanizei sthn console mhnyma
61     this.scoreLeft +=1; //auksanei to score tou aristera kata 1
62     UpdateUI(); //kanei update to score
63     Resetball(); //kanei reset to position ths mpalas
64 }
65
66 private void UpdateUI() {
67     this.scoreTextLeft.text = this.scoreLeft.ToString(); //kanei update to score tou aristera
68     this.scoreTextRight.text = this.scoreRight.ToString(); //kanei update to score tou deksia
69 }
70
71 private void Resetball() {
72     this.ballController.Stop(); //stamataei th mpala
73     this.ball.transform.position = this.startingPosition; //epanaferei sthn arxikh thesi
74     this.starter.StartCountdown(); //ksekina thn antistrofh metrisi
75 }
76
77

```

Εικόνες 3.4.5.5 & 3.4.5.6 Script Game Controller με επεξηγήσεις σαν σχόλια.

Goal Controller

Το goal controller script κάνει update to score και Reset την μπάλα μόλις βρεθεί στο collider του τέρματος-τοίχου.

```
Assets > Scripts > GoalController.cs
1  using UnityEngine;
2  using UnityEngine.Events; //library gia events
3
4
5  public class GoalController : MonoBehaviour
6  {
7      public UnityEvent onTriggerEnter; //dimiourgoume event
8
9      private void OnTriggerEnter(Collider other){ //trigger kata to collider
10         if (other.CompareTag("Ball")) { //otan h mpala vrethei sto boxcollider
11             onTriggerEnter.Invoke(); //kanei trigger to score
12         }
13     }
14 }
15
```

Εικόνα 3.4.5.7 Goal Controller script με επεξηγήσεις ως σχόλια.

Starter

Το script starter είναι για την εκκίνηση του animation στο countdown και δίνει το σήμα για την εκκίνηση της μπάλας.

```
Assets > Scripts > Starter.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Starter : MonoBehaviour
6  {
7      private GameController gameController;
8      private Animator animator;
9
10     // Start is called before the first frame update
11     void Start()
12     {
13         gameController = FindObjectOfType<GameController>(); //dinoume to gamecontroller
14         animator = GetComponent<Animator>(); //dinoume sthn metavliti to animator
15     }
16
17     public void StartCountdown() {
18         animator.SetTrigger("StartCountdown"); //ksekinaei to countdown animation
19     }
20
21     public void StartGame() {
22         gameController.StartGame(); //ksekinaei to game molis teleiwsei to countdown
23     }
24 }
25
```

Εικόνα 3.4.5.8 Starter script με επεξηγήσεις ως σχόλια.

4. Συμπεράσματα

Η Unity είναι ένα πλήρες πακέτο που επιτρέπει στους προγραμματιστές να δημιουργούν τα δικά τους παιχνίδια 2D ή 3D με σχετικά απλό τρόπο. Η μηχανή παιχνιδιού Unity είναι στατιστικά η καλύτερη μηχανή παιχνιδιών σχετικά με τη δημιουργία για Android και iOS. Πιστεύεται ότι οι μεγάλες αλλαγές απόδοσης στη Unity θα κυριαρχήσουν επίσης στον υπολογιστή και κονσόλα παιχνιδιών γιατί είναι πολύ απλή και εύκολη στη χρήση. Ένα μεγάλο ατού της unity είναι το πλήθος πληροφοριών και βοηθειών που μπορεί να βρει κάποιος, είτε σε διαδικτυακά μαθήματα, forums, σεμινάρια, εκπαιδευτικά site κλπ. Τέλος, υποστηρίζει πολλές πλατφόρμες, όχι μόνο τη δωρεάν έκδοση.

Μέσα από αυτήν την εργασία προσωπικά είδα τι δυνατότητες υπάρχουν και πως μπορώ να φτιάξω κάτι δικό μου. Σίγουρα χρειάστηκε να αφιερώσω πολύ χρόνο σε videos, tutorials, forums & online courses αλλά οι βοήθειες υπήρχαν. Συνολικά αφιέρωσα 30+ ώρες για την εκπόνηση της εργασίας αλλά ήταν κάτι που με γέμισε και με ωθεί στο να ασχοληθώ σοβαρότερα από εδώ και εμπρός. Επίσης η ικανοποίηση που μπορεί να λάβει κάποιος βλέποντας να υλοποιείται το πλάνο του είναι από τις μεγαλύτερες ανταμοιβές που μπορεί να σου προσφέρει.

5.Βιβλιογραφία

- [1] <https://www.w3schools.com/cs/index.php> - C# Tutorial
- [2] <https://www.youtube.com/@Brackeys> – Youtuber Unity
- [3] <https://www.udemy.com/course/learn-program-csharp-make-videogame-unity-prefabs-animations-platform/> - Learn C# and make a videogame with Unity on Udemy
- [4] https://unity3d.com/legal/as_terms ‘Asset store terms of service and EULA’
- [5] https://en.wikipedia.org/wiki/Main_Page - Wikipedia, The Free encyclopedia
- [6] <https://docs.unity3d.com/Manual/index.html> – Online Unity Manual
- [7] <https://stackoverflow.com/> - Forum κώδικα
- [8] <https://opengameart.org/> - Assets για Unity

