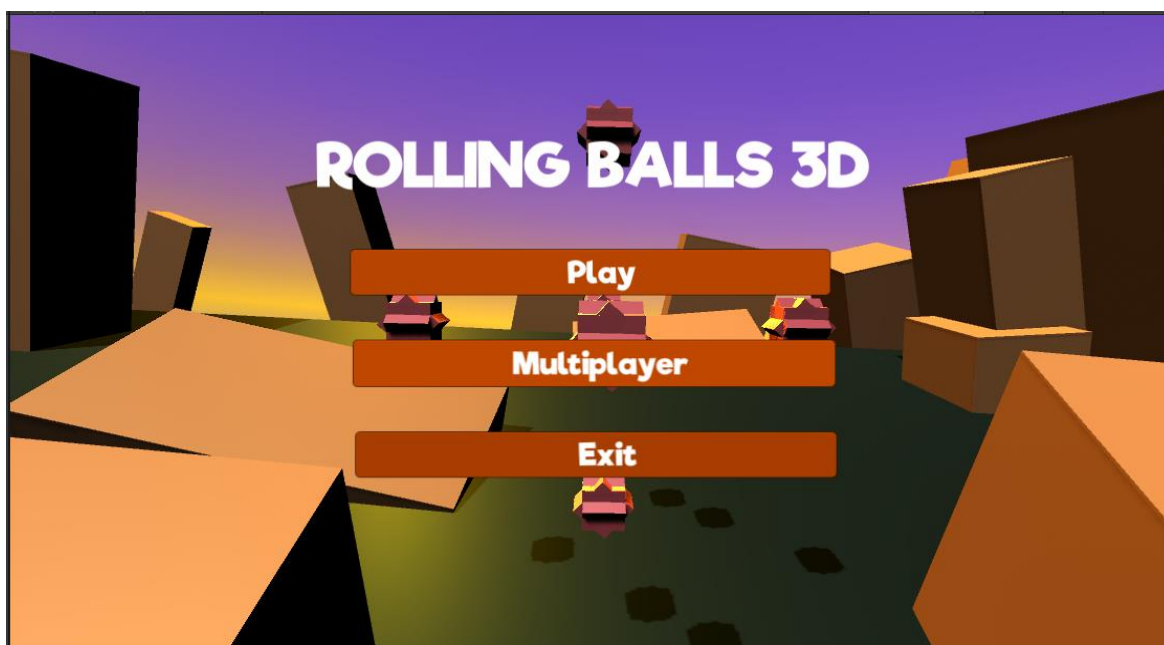




Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ – ΠΑΙΧΝΙΔΙΟΥ ΜΕΣΩ ΤΟΥ
ΠΡΟΓΡΑΜΜΑΤΟΣ UNITY 3D**



ΠΕΤΣΙΝΗΣ ΑΝΤΩΝΙΟΣ

ΑΜ : 1539

ΕΙΣΗΓΗΤΗΣ ΚΑΘΗΓΗΤΗΣ : ΑΣΗΜΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ

ΠΑΤΡΑ 2020

ΠΕΡΙΛΗΨΗ

Στην πτυχιακή αυτή θα ασχοληθούμε με την δημιουργία ενός παιχνιδιού 3D με την βοήθεια της πλατφόρμας του Unity 3D. Στα δύο πρώτα κεφάλαια θα κάνουμε μια ιστορική αναδρομή στην ιστορία των Video Games και το πως ξεκίνησαν να μπαίνουν στη ζωή μας και θα αναλύσουμε τα δύο βασικά προγράμματα για την δημιουργία Video Games. Στη συνέχεια από την θεωρία περνάμε στην πράξη και στην εγκατάσταση του Unity 3D και του Blender όπου είναι τα δυο βασικά προγράμματα που θα χρησιμοποιήσουμε. Έπειτα περνάμε στην αναλυτική δημιουργία του παιχνιδιού μας όπου περιγράφονται όλες οι δυσκολίες και λεπτομέρειες για την σωστή δημιουργία ενός τέτοιου project. Στο τελευταίο κεφάλαιο έχουμε την παρουσίαση του Rolling Balls 3D σαν εφαρμογή πλέον που είναι έτοιμη να χρησιμοποιηθεί από τον κάθε χρήστη που θα το εγκαταστήσει στον ηλεκτρονικό υπολογιστή του.

ABSTRACT

In this dissertation we will show the creation of a 3D game with the help of the Unity 3D platform. In the first two chapters we will make a historical review of the history of Video Games and how they began to enter our lives and we will analyze the two main programs for creating Video Games. Then from theory to practice and the installation of Unity 3D and Blender where are the two basic programs that we will use. Next we move on to the detailed creation of our game which describes all the difficulties and details for the correct creation of such a project. In the last chapter we have the presentation of Rolling Balls 3D as an application that is ready to be used by every user who will install it on their computer.

ΠΕΡΙΛΗΨΗ	2
ΚΕΦΑΛΑΙΟ 1: ΗΛΕΚΤΡΟΝΙΚΑ ΠΑΙΧΝΙΔΙΑ	5
1.1 ΟΡΙΣΜΟΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ	5
1.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ	7
ΚΕΦΑΛΑΙΟ 2 : ΠΡΟΓΡΑΜΜΑΤΑ ΔΗΜΙΟΥΡΓΙΑΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΑΙΧΝΙΔΙΩΝ	15
2.1 ΠΡΟΛΟΓΟΣ	15
2.2 UNITY	15
2.3 GAMEMAKER	16
2.3 UNREAL ENGINE	16
ΚΕΦΑΛΑΙΟ 3 : ΚΑΤΕΒΑΣΜΑ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΔΗΜΙΟΥΡΓΙΑΣ ΠΑΙΧΝΙΔΙΟΥ ΚΑΙ ΓΡΑΦΙΚΩΝ	18
3.1 ΠΡΟΛΟΓΟΣ	18
3.2 ΕΓΚΑΤΑΣΤΑΣΗ UNITY 3D	18
3.3 ΕΓΚΑΤΑΣΤΑΣΗ BLENDER	25
ΚΕΦΑΛΑΙΟ 4 : ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΑΙΧΝΙΔΙΟΥ	33
4.1 ΓΝΩΡΙΜΙΑ ΜΕ ΤΟ UNITY	33
4.1.1 Scene View	34
4.1.2 Game View	35
4.1.3 Hierarchy Panel	36
4.1.4 Project Panel	36
4.1.5 Inspector Panel	37
4.2 ΠΡΟΛΟΓΟΣ ΠΑΙΧΝΙΔΙΟΥ	39
4.3 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΕΛΑΦΟΥΣ	39
4.4 ΔΗΜΙΟΥΡΓΙΑ ΧΑΡΑΚΤΗΡΑ – PLAYER	44
4.5 ΔΗΜΙΟΥΡΓΙΑ ΚΑΜΕΡΑΣ	49
4.6 ΔΗΜΙΟΥΡΓΙΑ ΣΚΗΝΗΣ – LEVEL	52
4.7 ΔΗΜΙΟΥΡΓΙΑ ΕΠΟΜΕΝΗΣ ΣΚΗΝΗΣ (LEVEL)	53
4.8 ΔΗΜΙΟΥΡΓΙΑ PAUSE MENU ΚΑΙ MAIN MENU	55
ΚΕΦΑΛΑΙΟ 5 : ROLLINGBALLS 3D	61
ΣΥΜΠΕΡΑΣΜΑΤΑ	65
ΠΑΡΑΡΤΗΜΑ 1	66
ΒΙΒΛΙΟΓΡΑΦΙΑ	74
ΧΡΗΣΙΜΟΙ ΣΥΝΔΕΣΜΟΙ	75

ΚΕΦΑΛΑΙΟ 1: ΗΛΕΚΤΡΟΝΙΚΑ ΠΑΙΧΝΙΔΙΑ

1.1 ΟΡΙΣΜΟΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ

Οι αντιδράσεις των ανθρώπων διαφέρουν ακούγοντας τον όρο ηλεκτρονικό παιχνίδι ανάλογα με το αν έχουν ασχοληθεί ποτέ ή όχι με ένα τέτοιο παιχνίδι. Από τις αρχές της δεκαετίας του '80 και μετά, όλη η νέα γενιά έχει εκτεθεί σε ηλεκτρονικά παιχνίδια καθ' όλη την διάρκεια της ζωής της. Ένα παιχνίδι είναι ένας σωματικός ή πνευματικός διαγωνισμός που διέπεται από κανόνες και έχει ως στόχο στη διασκέδαση ή την ανταμοιβή των συμμετεχόντων (Zyda, 2005). Σύμφωνα με τον Zyda (2005), ο ορισμός αυτός δεν διαφέρει από αυτόν για το ηλεκτρονικό παιχνίδι, το οποίο ορίζει ως έναν πνευματικό διαγωνισμό, με τη βοήθεια ενός υπολογιστή, με συγκεκριμένους κανόνες και σκοπό τη διασκέδαση, την ψυχαγωγία ή κάποιο έπαθλο. Τα ηλεκτρονικά παιχνίδια για υπολογιστή ή κονσόλες είναι ένας όρος που χρησιμοποιείται πιο συχνά. Στην ουσία υπάρχει μια οθόνη, από την οποία ικανοποιούνται οι αισθήσεις του παίχτη και αναλόγως αντιμετωπίζει το παιχνίδι. Αναλόγως το παιχνίδι υπάρχουν και διαφορετικές συσκευές εισόδου όπως για παράδειγμα ένα πληκτρολόγιο, ένα joystick ή μια οθόνη αφής κλπ. Η συσκευή εισόδου που απαιτεί το ηλεκτρονικό παιχνίδι μπορεί να μην είναι ίδια για κάθε παιχνίδι, υπάρχει περίπτωση να είναι και διαφορετική ακόμα και για το ίδιο παιχνίδι αν αυτό παράγεται για διαφορετικές κονσόλες. Ένα παράδειγμα είναι και τα παιχνίδια που κυκλοφορούν για υπολογιστές και κονσόλες (Forza Horizon 3, NBA, FIFA) ή τα παιχνίδια που είναι φτιαγμένα για να διατίθενται σε smartphones ή ιστότους κοινωνικής δικτύωσης (Cookiecrush, EverWing, Total Soccer). Οπότε, αυτά τα παιχνίδια μπορούν να παίζονται στο μόνιτορ ενός υπολογιστή, στην οθόνη ενός κινητού τηλεφώνου (smartphone), στην οθόνη μιας τηλεόρασης, σε μια κονσόλα ακόμα και σε πιο εξελιγμένης μορφής συστήματα VR (virtual reality), όπως είναι το Oculus Rift που παρέχει στον χρήστη μια 3D εμπειρία. Ως κάποια μορφή προσομοίωσης μπορούν να θεωρηθούν τα περισσότερα παιχνίδια που κυκλοφορούν στην αγορά. Θα μπορούσαν ευκολά να χωριστούν σε δυο μεγάλες ομάδες με βάση το ποσό ρεαλιστικά είναι. Η πρώτη ομάδα είναι τα παιχνίδια προσομοίωσης με τον μεγαλύτερο ρεαλισμό. Σε αυτή την ομάδα θα μπορούσε κάποιος να πει πως εντάσσονται κυρίως τα σύγχρονα παιχνίδια μαχών, αγώνων, αθλητισμού, πολιτισμού και προσομοίωσης επιχειρήσεων. Τέτοια παιχνίδια είναι για παράδειγμα το FIFA, το Forza Horizon 3, το UFC, το Stronghold και διάφορα άλλα.



Εικόνα 1: Παιχνίδι μεγαλύτερου ρεαλισμού FIFA 2018 (ElectronicArts).

Στην δεύτερη ομάδα βρίσκονται παιχνίδια που παρουσιάζουν πιο λίγο ρεαλισμό και είναι τα παιχνίδια φαντασίας, περιπέτειας αλλά και διαστημικών μαχών όπως για παράδειγμα το Halo ή τα Star Wars.



Εικόνα 2: Παιχνίδι μικρότερου ρεαλισμού HALO 5 (Microsoft Studios).

Παράλληλα βέβαια υπάρχουν παιχνίδια προσομοίωσης που περιλαμβάνουν παιχνίδια γρίφων και διάφορα άλλα πιο παραδοσιακά παιχνίδια όπως για παράδειγμα ο Ναρκαλιευτής ή διάφορα παιχνίδια με τράπουλα.



Εικόνα 3: Παιχνίδι προσομοίωσης (Πασιέντζα Αράχνη).

1.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Το ηλεκτρονικό παιχνίδι είναι η πρώτη ποιοτικά διαφορετική μορφή παιχνιδιού μετά από εκατοντάδες χρόνια, ενώ αρχίζουν να εκλείπουν παραδοσιακά παιχνίδια που είχαν μια συνέχεια από την αρχαιότητα έως σήμερα (όπως το τσέρκι, γνωστό από την αρχαιότητα ως κρικηλασία). Τα πρώτα ηλεκτρονικά παιχνίδια γεννήθηκαν την δεκαετία του '60 ενώ η διάδοση τους στις βιομηχανικές χώρες ξεκίνησε στα τέλη της δεκαετίας του '70, για αρχή σε ειδικά καταστήματα ηλεκτρονικών παιχνιδιών και αργότερα στα σπίτια. Στην γρήγορη εξάπλωση τους βοήθησε αρχικά η ραγδαία εξέλιξη της τεχνολογίας αλλά και η κυριαρχία της εικόνας στα Μ.Μ.Ε , όπως επίσης μεγάλο ρολό έπαιξε ο αυξανόμενος φόβος των γονέων για την ασφάλεια των παιδιών τους, στους δρόμους των μεγαλουπόλεων και η σταδιακή μετάβαση του μοναχικού παιχνιδιού. Στις αρχές του 1970 κάνει την εμφάνιση του το παιχνίδι Pong, το οποίο είναι σαν να μας προαναγγέλλει την έλευση των ηλεκτρονικών παιχνιδιών. Εγκαινιάζει την Computer Control Games (παιχνίδια ελεγχόμενα από τον υπολογιστή) εποχή, μιας και ελέγχεται εξολοκλήρου από τον υπολογιστή. Δεν έχει κινούμενα μέρη και τα πάντα είναι ηλεκτρονικά, εκτός από τα joysticks οπου είναι μεγάλα και κινούνται με δυσκολία. Γνωρίζει μεγάλη κοινωνική αποδοχή και αυτό έχει ως αποτέλεσμα να το συναντά κάνεις σε

μέρη κάθε είδους. Στα μέσα της δεκαετίας του '70 μάλιστα κάποιες εκδοχές του έγιναν και ατομικά παιχνίδια σπιτιού. Μεγάλο ρόλο σε αυτό έπαιξε το μικρό μέγεθος του και η χαμηλή του τιμή. Η μεγάλη κοινωνική αποδοχή του Pong οφείλεται στο γεγονός ότι η θεματολογία του δεν είχε σχέση με τη βία και την κακογουστιά των προηγούμενων arcade παιχνιδιών. Σε αντίθεση, αναπαριστούσε ευγενή αθλήματα όπως το squash και το τένις.



Εικόνα 4: Παιχνίδι Pong (Atari 1972).

Η κατάσταση αυτή μοιάζει εξωπραγματική για τις παλαιότερες γενιές που μεγάλωσαν παίζοντας κνηνητό η κρυφτό στις γειτονίες, αλλά τα δεδομένα πλέον έχουν αλλάξει πολύ από τότε. Τα παιδιά των πόλεων δεν μπορούν πλέον να παίζουν ελεύθερα, οπότε έχουν μεταφέρει την έννοια του παιχνιδιού από τους δρόμους στην κονσόλα η τον υπολογιστή τους. Από την άλλη πλευρά η πρόοδος της τεχνολογίας ήταν τεράστια τα τελευταία χρόνια. Τα ηλεκτρονικά παιχνίδια της δεκαετίας του '80 «έσβησαν» σταδιακά πίσω από την μεγάλη επιτυχία των καινούργιων ηλεκτρονικών παιχνιδιών που προσφέρουν στον χρήστη περισσότερη ποικιλία, καλύτερα γραφικά, συναρπαστικούς ήρωες και πιο ενδιαφέρουσα πλοκή από αυτά που υπήρχαν έως τώρα. Το 1981 παρήχθησαν 500 νέοι τίτλοι παιχνιδιών, ενώ το πρώτο μόλις εξάμηνο του 1982, 375 νέα παιχνίδια έκαναν την δική τους είσοδό στην αγορά. Η Sega το έτος 1989 κυκλοφόρησε το Sega Genesis όπου είναι μια παιχνιδιομηχανή με

μεγάλη χωρητικότητα μνήμης, ενώ η Nintendo κυκλοφόρησε το GameBoy όπου είχε ένα μεγάλο προβάδισμα απέναντι στους ανταγωνιστές του λόγω του μικρού μεγέθους που είχε. Έτσι ο χρήστης μπορεί να το πάρει μαζί του παντού και να το χρησιμοποιεί χωρίς να χρειάζεται κάποια επιπλέον οθόνη, πληκτρολόγιο ή κάποιο καλώδιο για ρεύμα.



Εικόνα 5: Παιχνιδομηχανές Sega Genesis & Nintendo Game Boy (1989).

Το έτος 1991 η Sega κυκλοφόρησε την δική της φορητή παιχνιδομηχανή με το όνομα Sega Game Gear, ενώ καινοτόμησε προσθέτοντας έγχρωμα γραφικά σε αντίθεση με τις τότε παιχνιδομηχανές άλλων εταιριών, εμπορικά ήταν μια αποτυχία για την εταιρία λόγω χαμηλών πωλήσεων. Την ίδια χρόνια η Nintendo κυκλοφόρησε το SuperNES. Οι χαρακτήρες των παιχνιδιών πλέον όπως επίσης και τα σενάρια είναι πιο αληθοφανή από ότι ήταν στο παρελθόν. Επίσης παρέχονται στους παίκτες μεγαλύτερες δυνατότητες για τον έλεγχο των χαρακτήρων τους μέσα στο παιχνίδι. Αυτό έχει σαν αποτέλεσμα στα παιχνίδια πολεμικών τεχνών για παράδειγμα οι παίκτες να μπορούν να πραγματοποιούν περισσότερες κινήσεις με τον χαρακτήρα τους λόγω ότι οι κονσόλες με την πάροδο του καιρού γίνονται και πιο ισχυρές.



Εικόνα 6: Παιχνιδομηχανές Nintendo SNES & Sega Game Gear (1991).

Η Sega το 1995 κυκλοφορεί το Sega Saturn όπου για πρώτη φορά στην ιστορία των παιχνιδομηχανών θα υπάρξουν τα τρισδιάστατα γραφικά, ενώ πλέον αρχίζει και η αντικατάσταση των cartridges (κασέτες) με τα CD. Στην αρχή τα τρισδιάστατα γραφικά ήταν πολυγωνικά, παρουσιάζοντας όμως μια πολύ σημαντική βελτίωση όσο αναφορά την απεικόνιση των χαρακτήρων του παιχνιδιού, παράλληλα όμως μετατρέπουν τις κινήσεις των χαρακτήρων πολύ πιο ρεαλιστικές βάσει των άλλων παιχνιδιών που υπήρχαν στο παρελθόν. Οι χαρακτήρες πλέον μπορούν να κινηθούν προς όλες τις κατευθύνσεις.



Εικόνα 7 : Παιχνιδομηχανή Sega Saturn (1995).

Το 1996 κυκλοφορεί από την Nintendo το Nintendo 64 όπου εισήγαγε στην αγορά έναν νέο επεξεργαστή 64 bit (από τον αριθμό των bits πηρέ και το όνομα η κονσόλα), αλλά

την συγκεκριμένη χρονική περίοδο δεν υπήρχαν τα κατάλληλα software όπου απαιτούσε ο συγκεκριμένος επεξεργαστής. Ήταν και η τελευταία κονσόλα της συγκεκριμένης εταιρίας όπου λειτουργούσε με cartridges (κασέτες).



Εικόνα 8: Παιχνιδομηχανή Nintendo 64 (1996).

Το έτος 1999 η Sony κυκλοφορεί το Sony PlayStation που είναι πλέον η πιο εξελιγμένη κονσόλα για εκείνη την εποχή, με μεγαλύτερο ρεαλισμό και φυσική κίνηση που μοιάζει πάρα πολύ με την ανθρώπινη αλλά και με τρισδιάστατους χαρακτήρες. Την ίδια χρόνια η Sega Entertainment παράγει την κονσόλα Dreamcast.



Εικόνα 9: Παιχνιδομηχανές Sega Dreamcast & Sony PlayStation (1999).

Το έτος 2007 θα βγει σε κυκλοφορία ο επεξεργαστής Φυσικής (PhysX). Ο επεξεργαστής αυτός βοήθησε τους κατασκευαστές παιχνιδιών να μπορούν να προσομοιώνουν τα επίπεδα ρεαλισμού με ευκολία και να προσομοιώνουν τις φυσικές ιδιότητες αντικειμένων με πολύ

μεγάλη ακρίβεια. Γενικότερα η αγορά των ηλεκτρονικών παιχνιδιών προσφέρει τέσσερις τύπους λειτουργικών συστημάτων:

- τα παιχνίδια χειρός (GameBoy, PSP)
- τις οικιακές βιντεοκονσόλες (PlayStation, Xbox)
- τις παιχνιδομηχανές Arcade (Mortal Kombat, Pong)
- τον προσωπικό ηλεκτρονικό υπολογιστή (Laptop, Desktop)

Τα τελευταία έτη είναι άξιο αναφοράς ότι γνωρίζουν πολύ μεγάλη απήχηση τα online παιχνίδια τα οποία χωρίζονται στις παρακάτω κατηγορίες.

- Παιχνίδια μέσω κάποιου browser : είναι τα παιχνίδια που έχουν ένα απλό χαρακτήρα και παρέχονται συνήθως χωρίς αντίτιμο από ιστοσελίδες που παρέχουν και διαφημίσεις. Για παράδειγμα τέτοια παιχνίδια είναι το Farmville, JellyCrush, dodGames.



Εικόνα 10: Αρχική οθόνη στο dod Games.

- Παιχνίδια διαφημιστικού τύπου : είναι παιχνίδια που σχεδιάστηκαν για την προώθηση μιας υπηρεσίας , ενός προϊόντος η μιας ιδεολογίας. Τέτοια παιχνίδια είναι το CoolSpot (7Up), το Zool (Chupa Chups), και το Americas Army (USArmy).



Εικόνα 11: Παιχνίδι Διαφημιστικού τύπου CoolSpot (7Up).

- Παιχνίδια δικτύου: είναι τα παιχνίδια που τα εγκαθιστά ο χρήστης σε ένα ηλεκτρονικό υπολογιστή αλλά για να τα παίξει θα χρειαστεί να υπάρχει κάποια σύνδεση στο διαδίκτυο. Παραδείγματα τέτοιων παιχνιδιών είναι το Fortnite (EpicGames), η το League Of Legends (RiotGames).



Εικόνα 11: Παιχνίδι δικτύου Fortnite (EpicGames).

- Παιχνίδια τύπου MMORPG: είναι τα παιχνίδια τα οποία να μεν είναι παιχνίδια δικτύου, αλλά σε κάθε ένα παιχνίδι από αυτά είναι εγγεγραμμένος ένας τεράστιος αριθμός παικτών από όλο τον κόσμο και το παιχνίδι συνεχίζει να προοδεύει και ως είναι εκτός σύνδεσης ο παίκτης. Τέτοια παιχνίδια είναι το World of Warcraft (Blizzard Entertainment), το Guardians of Divinity (37Games) και το Guild Wars 2 (Arena Net).



Εικόνα12: ΠαιχνίδιMMORPG Guild Wars 2 (Arena Net).

ΚΕΦΑΛΑΙΟ 2 : ΠΡΟΓΡΑΜΜΑΤΑ ΔΗΜΙΟΥΡΓΙΑΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΑΙΧΝΙΔΙΩΝ

2.1 ΠΡΟΛΟΓΟΣ

Για να καταφέρει ένα παιχνίδι η εφαρμογή να υλοποιηθεί θα χρειαστεί ο δημιουργός του να χρησιμοποιήσει περισσότερα από ένα εργαλεία με σκοπό να πέτυχει ένα ικανοποιητικό αποτέλεσμα. Προγράμματα όπως το Blender, το Photoshop ή το Unity και διάφορα άλλα. Για αυτό τον λόγο όλες οι μεγάλες εταιρίες δημιουργίας βιντεοπαιχνιδιών έχουν διάφορους προγραμματιστές με στόχο να είναι κάθε ένας υπεύθυνος για κάποιο συγκεκριμένο μέρος του παιχνιδιού όπως το τμήμα γραφικών, ήχου, videoediting κλπ.

2.2 UNITY

Μια από τις πιο γνωστές μηχανές κατασκευής παιχνιδιών είναι το πρόγραμμα Unity το οποίο έχει τη δυνατότητα να μπορεί να δημιουργήσει παιχνίδια για ένα πολύ μεγάλο αριθμό και μια εξίσου πολύ μεγάλη γκάμα πλατφόρμων (cross-platform). Έχει κυκλοφορήσει από την εταιρία Unity Technologies. Το Unity συμπεριλαμβάνει μια μηχανή κατασκευής βιντεοπαιχνιδιών καθώς επίσης και ένα ενοποιημένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE). Μέσω του Unity δημιουργούνται και έπειτα κυκλοφορούν κατά κύριο λόγο βιντεοπαιχνίδια για

- Smartphone και Tablets
- Σταθερούς και φορητούς ηλεκτρονικούς υπολογιστές
- Παιχνοδοκονσόλες (PlayStation, Xbox)
- Ιστοτόπους (Facebook κτλ.)

Το Unity όπως παρουσιάστηκε στο “Worldwide Developers Conference” το οποίο είχε διοργανωθεί από την Apple λειτουργούσε αποκλειστικά και μόνο σε λειτουργικά συστήματα Mac, η εταιρία όμως στην συνέχεια αποσκοπούσε στο να καταφέρει να επεκταθεί σε πάνω από δεκαπέντε πλατφόρμες. Εξίσου σημαντικό είναι ότι το Unity παρέχει στον καθένα developer να δημιουργήσει το βιντεοπαιχνίδι του και να το δημοσιεύσει σε οποία από όλες τις πλατφόρμες επιθυμεί αυτός. Τα αρχεία του συστήματος αποθηκεύονται σε ένα διακομιστή (server), που έχει το Unity όπως επίσης εκεί βρίσκεται και η μηχανή προσομοίωσης φυσικής

“PhysX” που παρέχεται από την εταιρία Nvidia. Η χρήση του “Direct 3D” απαιτείται για να χρησιμοποιηθούν στον καλύτερο βαθμό τα γραφικά των παιχνιδιών στα λειτουργικά των Xbox, OpenGL και Windows για τα λειτουργικά τους συστήματα Mac, Linux, Android και iOS. Με το Unity ο κάθε χρήστης έχει την δυνατότητα να μετατρέπει και να συμπιέζει την ανάλυση των γραφικών των εικόνων που χρησιμοποιεί στα παιχνίδια του για να μπορεί να τα κάνει να είναι συμβατά με την κάθε πλατφόρμα που στοχεύει. Το Unity έχει υλοποιηθεί βάση του “Mono” που είναι μια εφαρμογή ανοιχτού κώδικα από της “Net Framework”. Επίσης ο προγραμματιστής έχει τη δυνατότητα να συγγραφεί τον κώδικα πάνω σε τρεις διαφορετικές γλώσσες προγραμματισμού. Την C#, την Javascript και την Boo.

2.3 GAMEMAKER

Μια από τις πιο αναγνωρίσιμες εφαρμογές όπου η βασική της χρήση είναι η δημιουργία βιντεοπαιχνιδιών για υπολογιστές είναι το GameMaker. Η δημιουργία και η έκδοση του GameMaker έχει γίνει από τον λέκτορα του ιδρύματος πληροφοριών και επιστήμων πληροφορικής του πανεπιστημίου της Ουτρέχτης, Mark Overmars. Το GameMaker είναι σχεδιασμένο για να παρέχει στον προγραμματιστή διάφορες λειτουργίες οι οποίες θα του δώσουν την δυνατότητα να κατασκευάσει ένα δισδιάστατο 2D παιχνίδι κάνοντας χρήση ενός απλού ηλεκτρονικού υπολογιστή. Με την εκμάθηση κάποιων βασικών λειτουργιών του προγράμματος είναι εύκολο να χρησιμοποιηθεί και από κάποιον αρχάριο χρήστη. Το GameMaker είναι σχεδιασμένο ώστε να μπορεί να λειτουργήσει στα περισσότερα λειτουργικά συστήματα που κυκλοφορούν ενώ λειτουργεί και σε πιο παλιού τύπου λειτουργικά συστήματα, συγκεκριμένα από τα Windows 98 και μετά. Επιπλέον με το GameMaker ο χρήστης μπορεί να δημιουργήσει και τρισδιάστατα 3D παιχνίδια μέσα από μια πολύ μεγάλη γκάμα επιλογών που δίνονται από την Full έκδοση του προγράμματος. Παρόλο που το GameMaker είναι ένα εύχρηστο εργαλείο και για αρχάριους χρήστες, παρέχει τεράστιες δυνατότητες για την δημιουργία εξελιγμένων και ευπαρουσίαστων παιχνιδιών.

2.3 UNREAL ENGINE

Μια άλλη μηχανή δημιουργίας βιντεοπαιχνιδιών είναι η Unreal Engine που φτιάχτηκε από την εταιρία EpicGames. Παρουσιάστηκε το 1998 για πρώτη φορά στο first-person shooter παιχνίδι Unreal. Ενώ στο αρχικό του στάδιο το πρόγραμμα είχε δημιουργηθεί για την

ανάπτυξη first-person shooters παιχνιδιών, στην πορεία γνώρισε αρκετή επιτυχία όταν χρησιμοποιήθηκε και σε μια μεγάλη γκάμα διαφορετικών παιχνιδιών. Τέτοιοι τύποι παιχνιδιών είναι τα MMORPG και RPG. Στο Unreal Engine για να προγραμματίσει κάποιος πρέπει να χρησιμοποιήσει την C ++, με αυτόν τον τρόπο παρέχεται ένας υψηλός βαθμός φορητότητας. Με αυτόν τον τρόπο το Unreal Engine γίνεται ένα εργαλείο που έχει μεγάλη απήχηση σε έναν πολύ μεγάλο αριθμό προγραμματιστών βιντεοπαιχνιδιών. Η Unreal Engine 4 είναι πλέον η τρέχουσα έκδοση που κυκλοφορεί και έχει σχεδιαστεί ώστε να μπορεί να λειτουργήσει με το Microsoft DirectX 11 όπως επίσης και με το Microsoft DirectX 12 για τα λειτουργικά συστήματα των Microsoft Windows, Windows RT και Xbox One. Με το OpenGL είναι σχεδιασμένο για να λειτουργεί στα λειτουργικά συστήματα των OSX, Linux, PlayStation 4, iOS καθώς επίσης και Android. Με το Vulkan λειτουργεί σε λειτουργικό σύστημα Android, επίσης με το Metal λειτουργεί σε iOS. Κλείνοντας, με JavaScript / WebGL καταφέρνει να λειτουργεί για HTML5 Web Browsers.

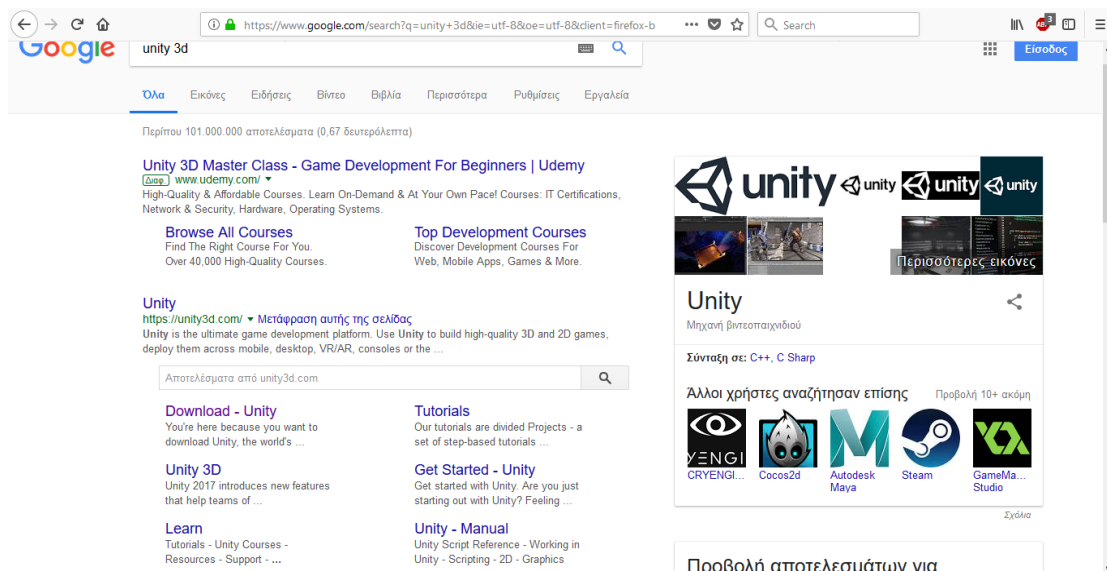
ΚΕΦΑΛΑΙΟ 3 : ΚΑΤΕΒΑΣΜΑ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ ΔΗΜΙΟΥΡΓΙΑΣ ΠΑΙΧΝΙΔΙΟΥ ΚΑΙ ΓΡΑΦΙΚΩΝ

3.1 ΠΡΟΛΟΓΟΣ

Έπειτα από αναζήτηση που έγινε στο διαδίκτυο για προγράμματα δημιουργίας παιχνιδιών και προγράμματα δημιουργίας τρισδιάστατων γραφικών, επιλέχθηκαν τα εξής προγράμματα. Για την δημιουργία του παιχνιδιού επιλέχθηκε το Unity 3D και για την δημιουργία τρισδιάστατων γραφικών και χαρακτήρων του παιχνιδιού επιλέχθηκε το Blender. Η επιλογή τους δεν ήταν τυχαία και αυτό διότι είναι δυο πολύ εύχρηστα προγράμματα για νέους χρήστες με μια ευρεία γκάμα δυνατοτήτων.

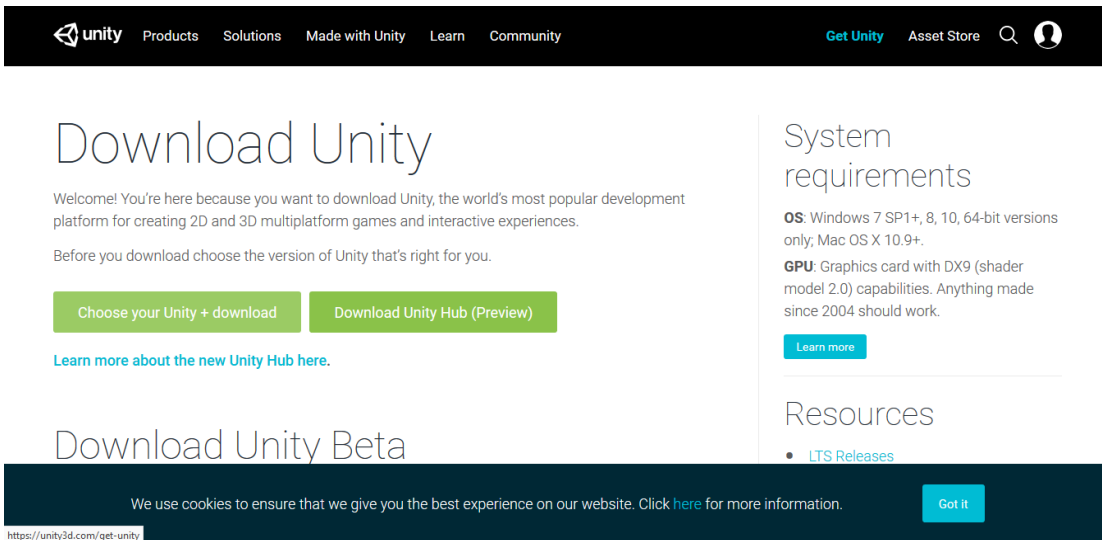
3.2 ΕΓΚΑΤΑΣΤΑΣΗ UNITY 3D

Μετά από αναζήτηση προγραμμάτων δημιουργίας παιχνιδιών επιλέχθηκε το Unity 3D για την υλοποίηση του παιχνιδιού.



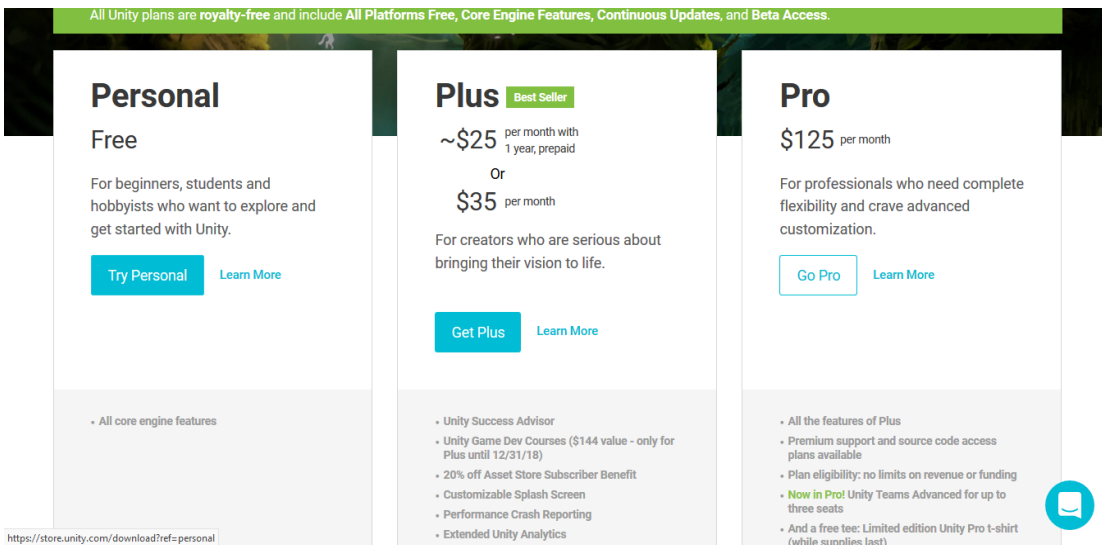
Εικόνα 1: Αναζήτηση για Unity.

Εφόσον γίνει είσοδος στην σελίδα του Unity, εκεί βρίσκετε η επιλογή του download.



Εικόνα 2: Επιλογή Κατεβάσματος Προγράμματος.

Μας δίνονται τρεις επιλογές κατεβάσματος του προγράμματος. Μια δωρεάν και δυο με πληρωμή. Επιλέχθηκε η πρώτη επιλογή.



Εικόνα 3: Επιλογή Προγράμματος Personal Free.

Στην συνέχεια επιλέγονται οι οροί χρήσης του προγράμματος και αφού διαβαστούν μπορεί να ξεκινήσει το κατέβασμα του προγράμματος.

Accept terms

By clicking, I confirm that I am eligible to use Unity Personal per the [Terms of Service](#), as I or my company meet the following criteria:

- Do not make more than \$100k in annual gross revenues, regardless of whether Unity Personal is being used for commercial purposes, or for an internal project or prototyping.
- Have not raised funds in excess of \$100K.
- Not currently using Unity Plus or Pro.

If you are not eligible to use Unity Personal, please [click here](#) to learn more about Unity Plus and Unity Pro.

Download Installer for Windows

Download Unity Hub (Preview)

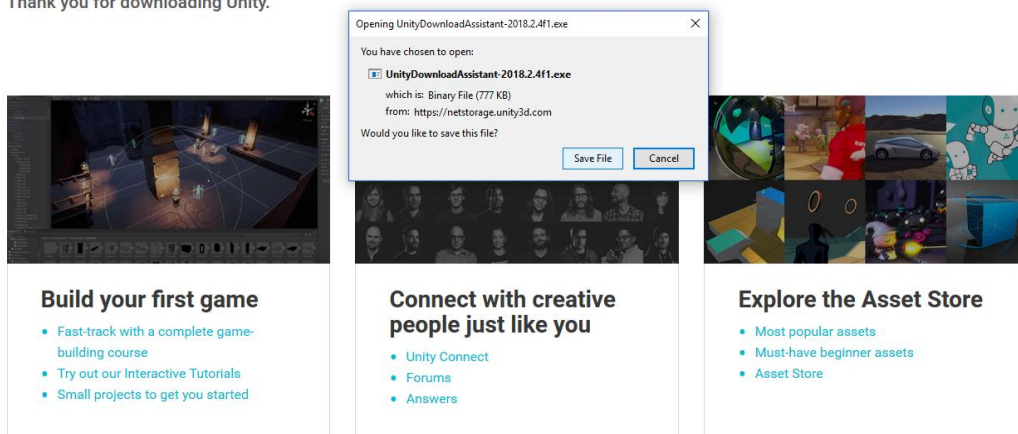
Looking to download the installer for Mac OS X?
[Choose Mac OS X](#)

Εικόνα 4: Επιλογή Όρων Χρήσης.

Έπειτα γίνεται η αποθήκευση του αρχείου στον υπολογιστή μας όπου από εκεί θα κατεβεί και το τελικό πρόγραμμα.

Let's get you started!

Thank you for downloading Unity.



Opening UnityDownloadAssistant-2018.2.4f1.exe

You have chosen to open:

UnityDownloadAssistant-2018.2.4f1.exe
which is: Binary File (777 KB)
from: <https://netstorage.unity3d.com>

Would you like to save this file?

Save File Cancel

Build your first game

- [Fast-track with a complete game-building course](#)
- [Try out our Interactive Tutorials](#)
- [Small projects to get you started](#)

Connect with creative people just like you

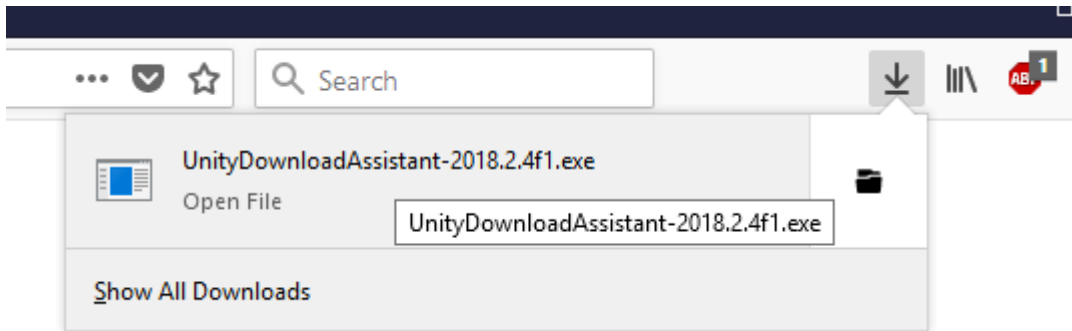
- [Unity Connect](#)
- [Forums](#)
- [Answers](#)

Explore the Asset Store

- [Most popular assets](#)
- [Must-have beginner assets](#)
- [Asset Store](#)

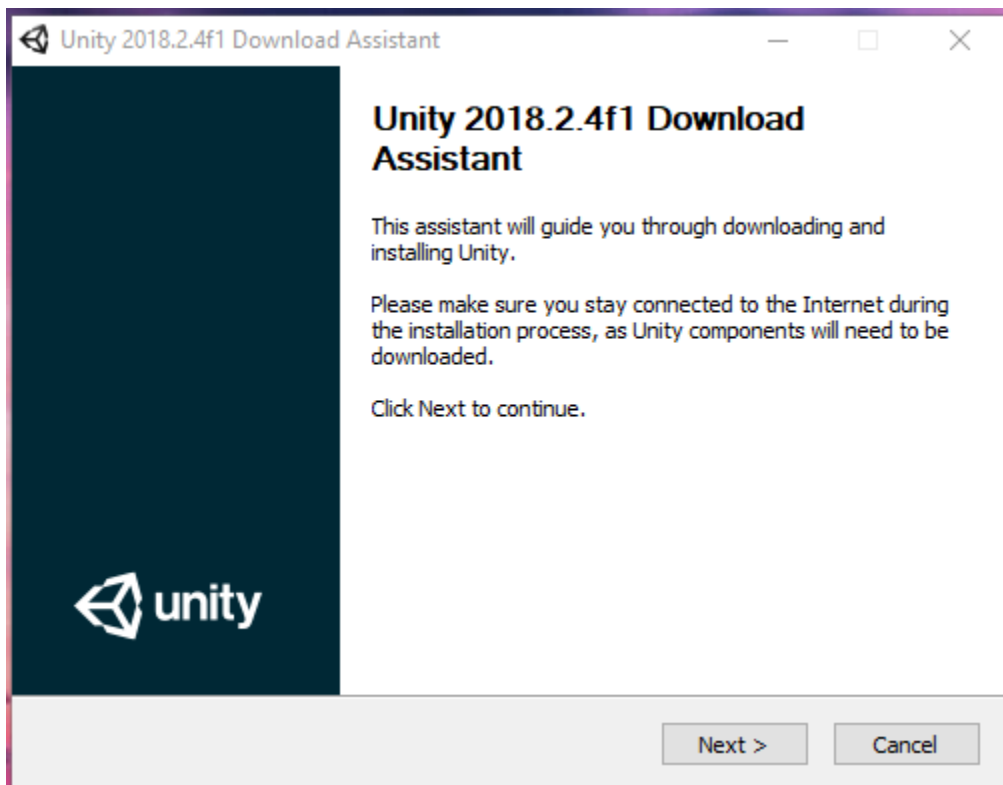
Εικόνα 5: Επιλογή Αποθήκευσης Αρχείου.

Αφού κατεβεί το αρχείο στον υπολογιστή με ένα κλικ μπορεί πλέον να αρχίσει η εγκατάσταση του προγράμματος.



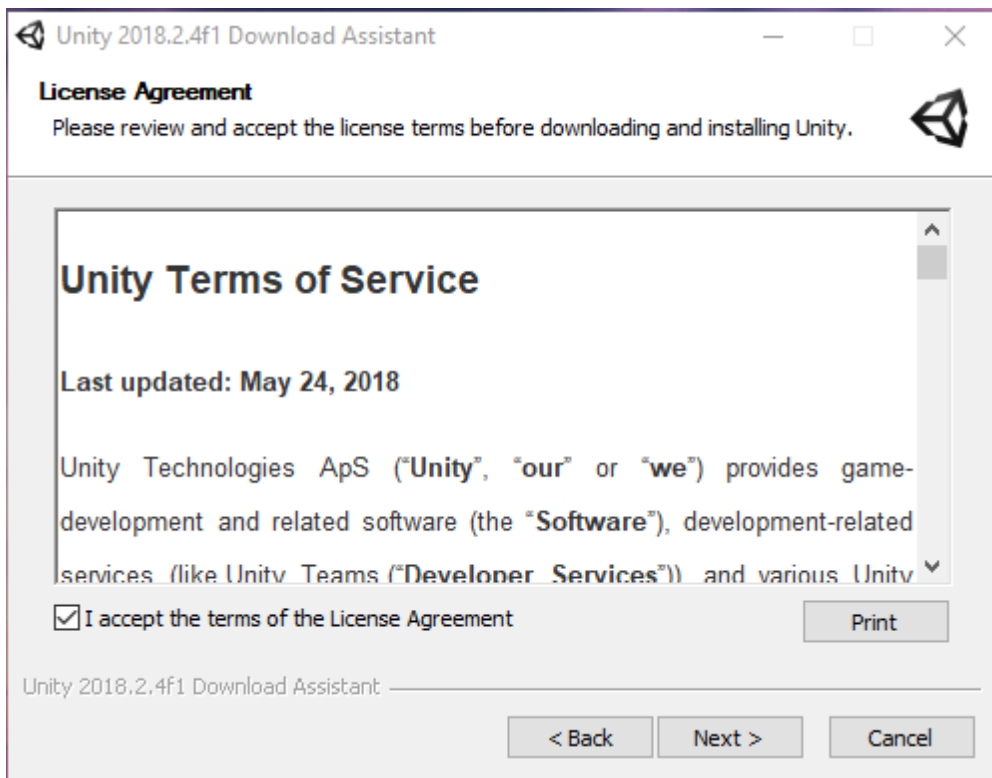
Εικόνα 6: Άνοιγμα Αρχείου Εγκατάστασης.

Στο πρώτο παράθυρο που εμφανίζεται στην οθόνη επιλέγετε η επιλογή next.



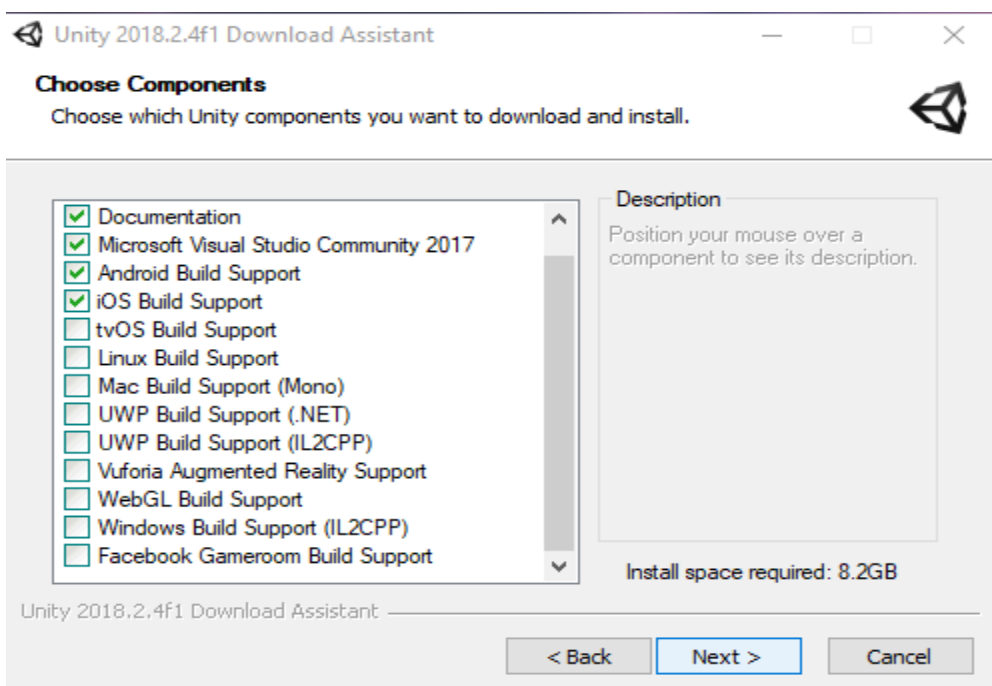
Εικόνα 7: Εκκίνηση Εγκατάστασης Unity.

Αφού διαβαστούν πάλι οι όροι χρήσης του προγράμματος επιλέγετε το checkbox και στην συνέχεια το πεδίο next.



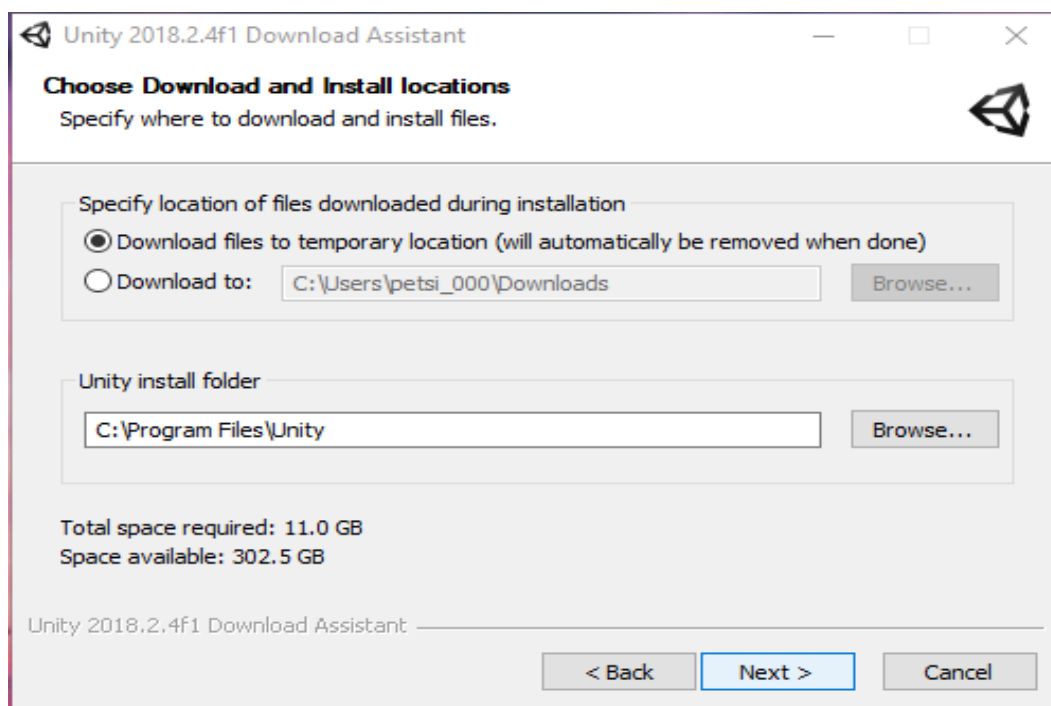
Εικόνα 8: Επιλογή Όρων Χρήσεως.

Στο επόμενο βήμα πρέπει να γίνει η επιλογή εξαρτημάτων που θα παρέχει το πρόγραμμα στον χρήστη. Αφού επιλεχτούν τα εξαρτήματα που χρειάζεται ο κάθε χρήστης, πατάει το πεδίο next.



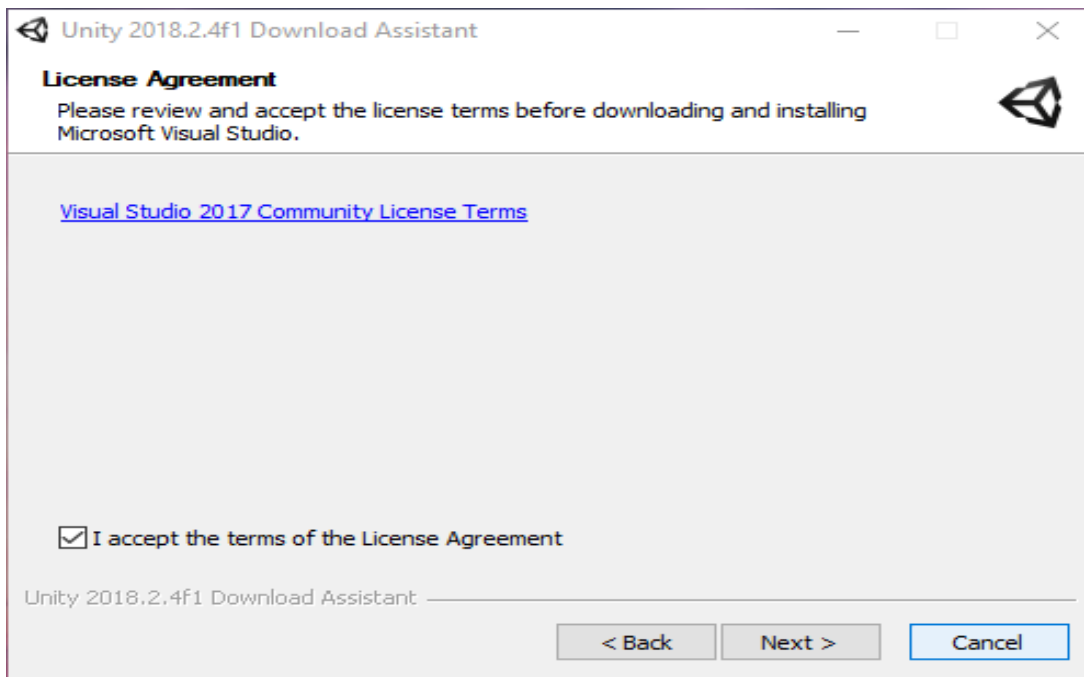
Εικόνα 9: Επιλογή Εξαρτημάτων Προγράμματος.

Στο επόμενο παράθυρο ζητείται από τον χρήστη να επιλέξει την διαδρομή αποθήκευσης του προγράμματος. Εφόσον διαλέξει την διαδρομή επιλέγετε πάλι το πεδίο next για να συνεχιστεί η εγκατάσταση.



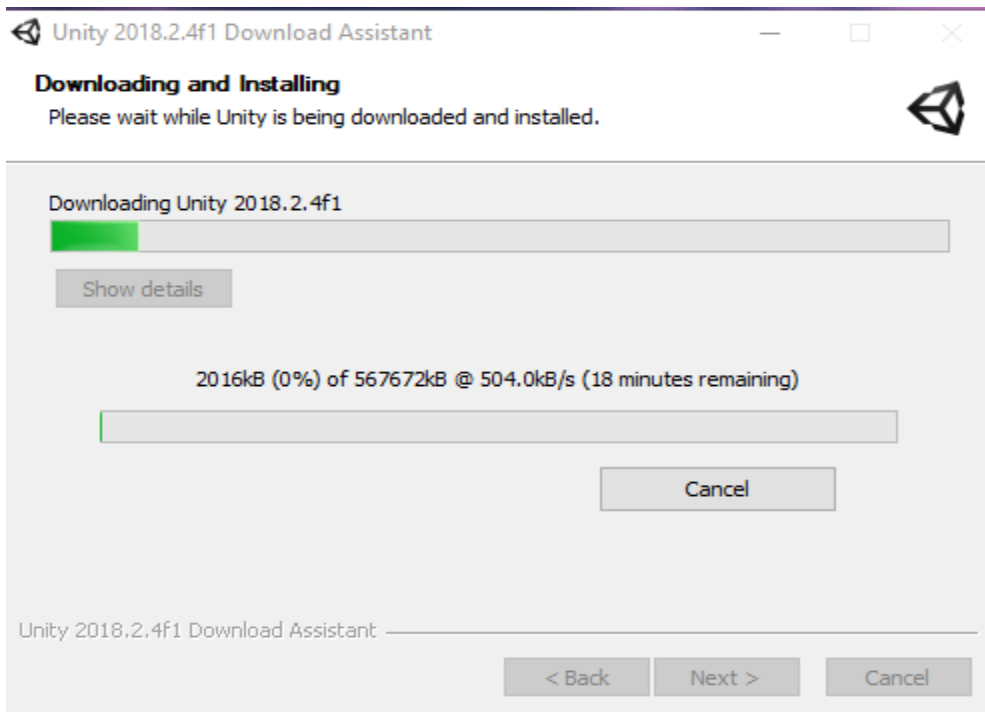
Εικόνα 10: Επιλογή Διαδρομής Αποθήκευσης.

Αμέσως μετά οριστικοποιούνται οι όροι χρήσης για τελευταία φορά και μέσω του πεδίου next μπορεί πλέον να ξεκινήσει η εγκατάσταση.



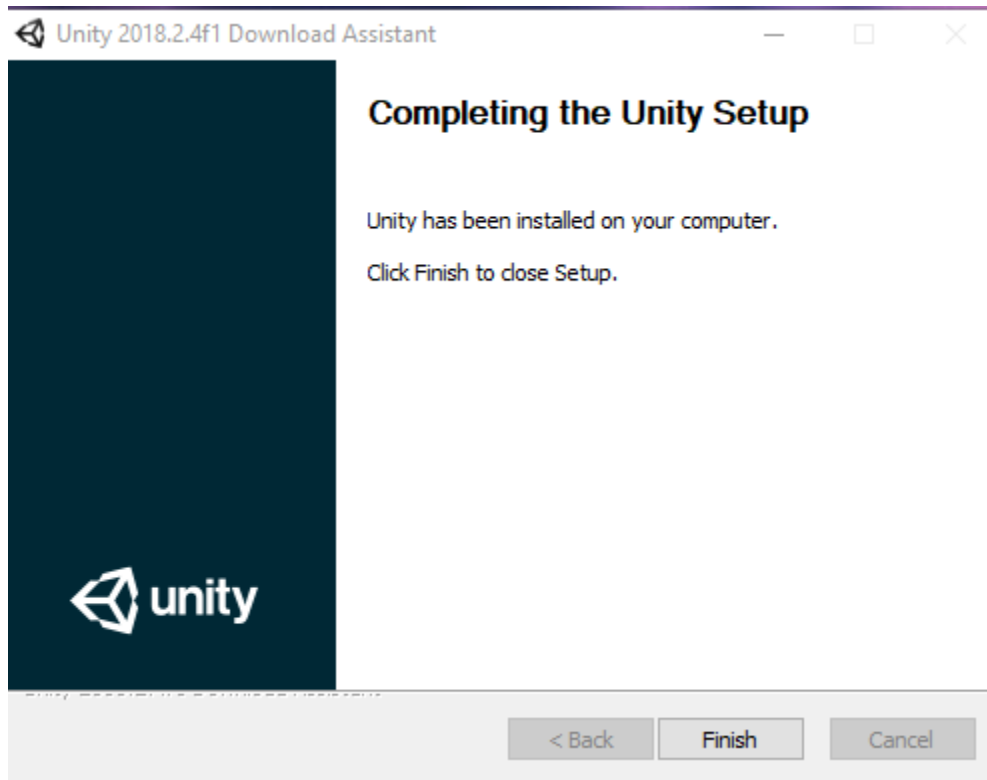
Εικόνα 11: Εκκίνηση Εγκατάστασης.

Στην συνέχεια φαίνεται η πρόοδος της εγκατάστασης.



Εικόνα 12: Πρόοδος εγκατάστασης.

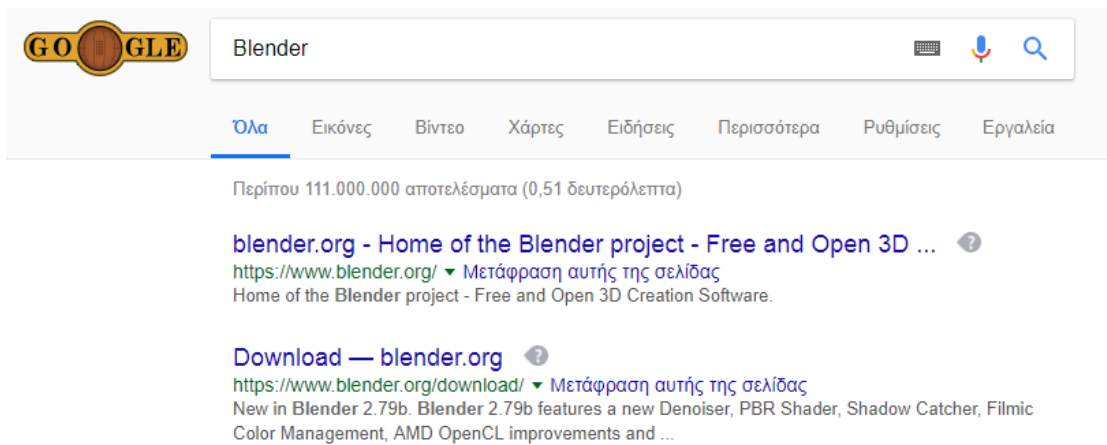
Αφού ολοκληρωθεί η εγκατάσταση του Unity εμφανίζεται το παράθυρο που δείχνει το τέλος της εγκατάστασης και μέσω του πεδίου Finish ο χρήστης μπορεί πλέον να κάνει εκκίνηση του προγράμματος.



Εικόνα 13: Ολοκλήρωση Εγκατάστασης.

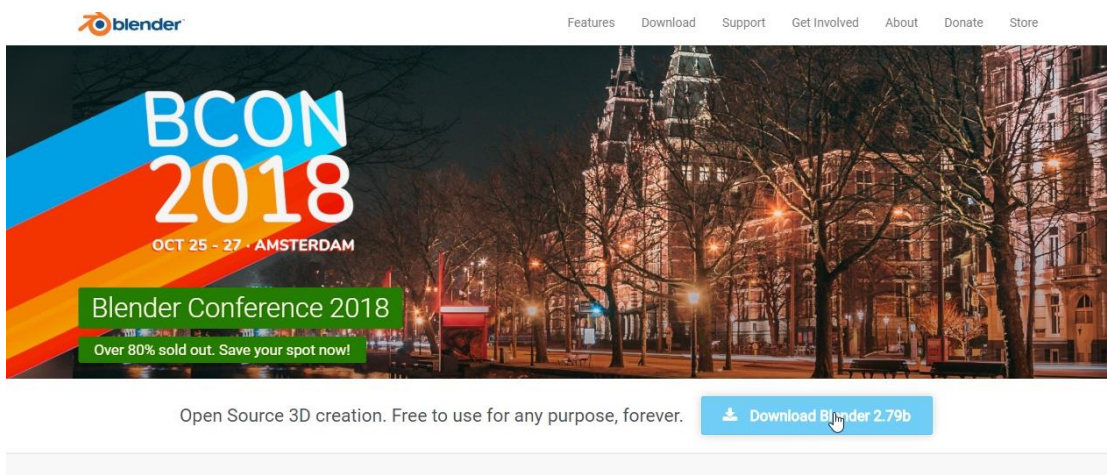
3.3 ΕΓΚΑΤΑΣΤΑΣΗ BLENDER

Προκειμένου να πραγματοποιηθεί η δημιουργία των μοντέλων έγινε αναζήτηση στο Google για μηχανές δημιουργίας τρισδιάστατων μοντέλων. Η μηχανή που επιλέχθηκε είναι η Blender.



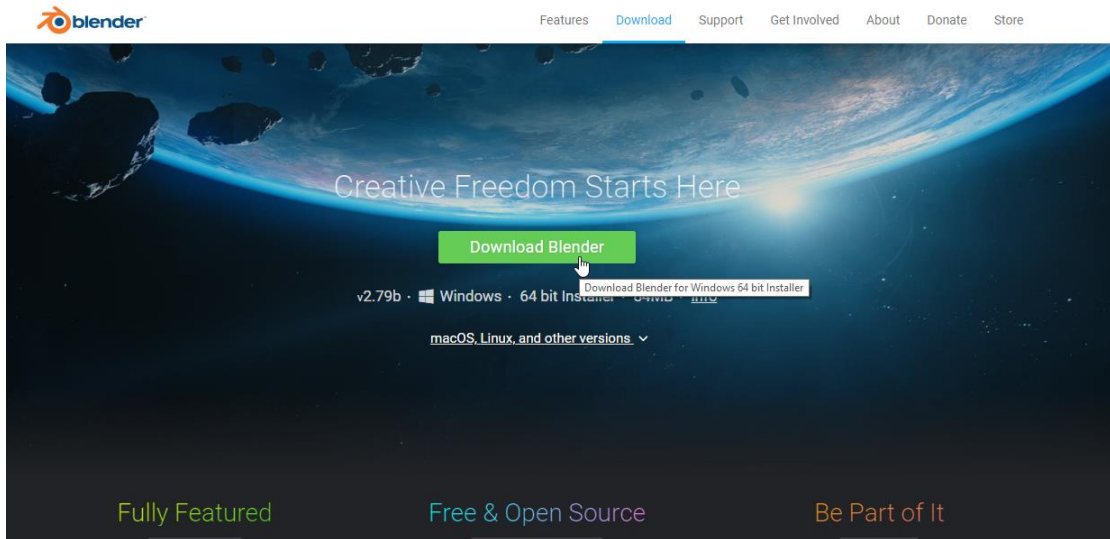
Εικόνα 1: Σελίδα του Blender.

Αφού γίνει είσοδος στην σελίδα της blender υπάρχει εκεί επιλογή κατεβάσματος της εφαρμογής.



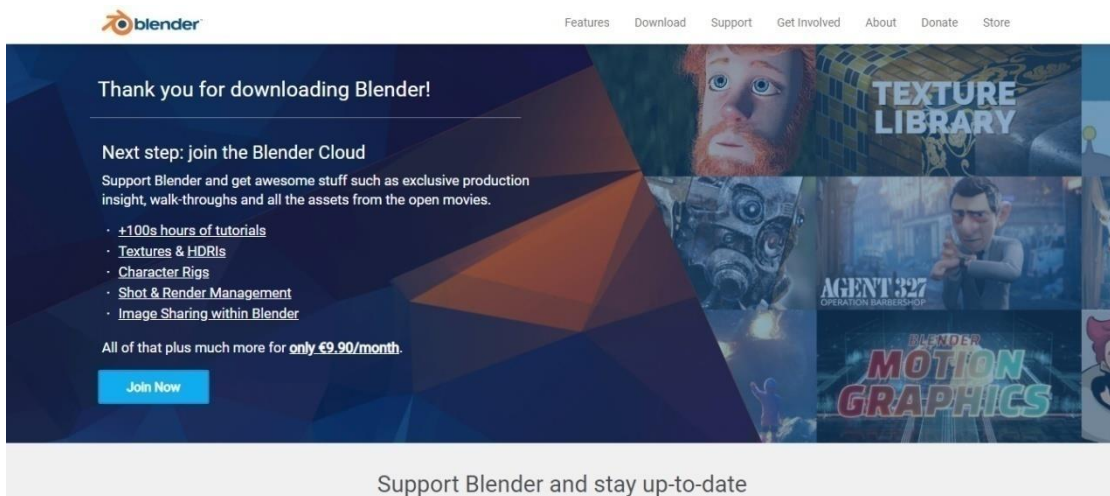
Εικόνα 2: Επιλογή κατεβάσματος εφαρμογής.

Στη συνέχεια δίνονται επιλογές όσο αφορά το λειτουργικό σύστημα που θα κατεβεί το αρχείο. Εκεί γίνεται επιλογή του πεδίου download.



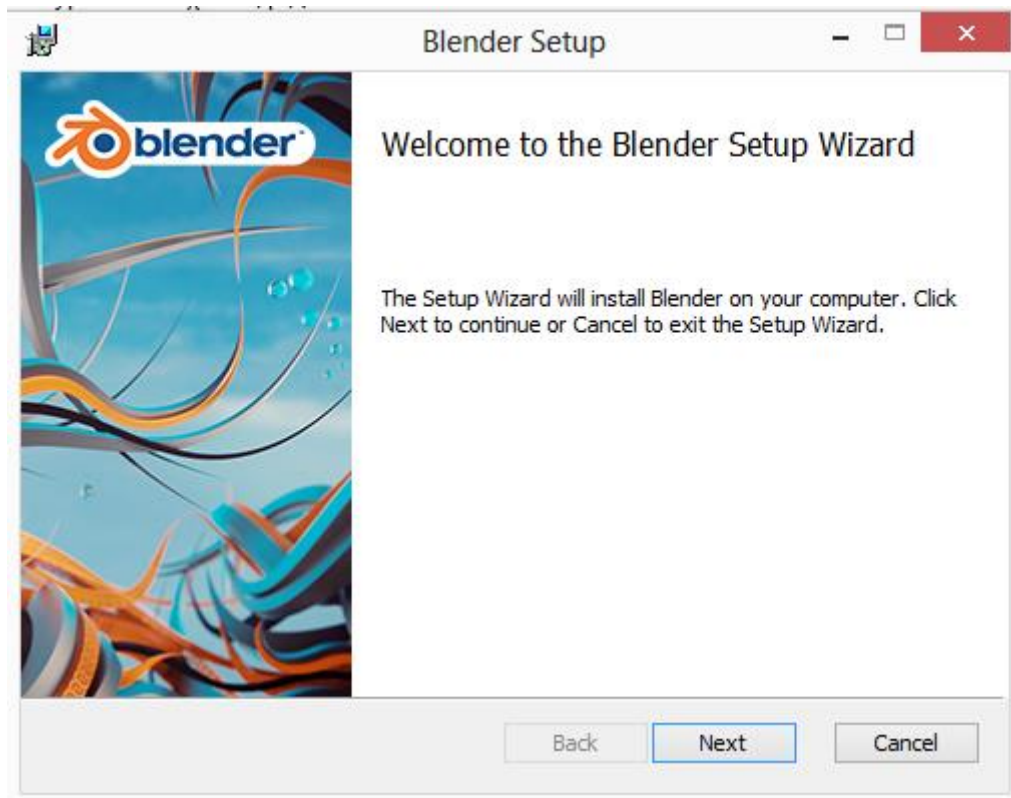
Εικόνα 3: Επιλογή download.

Εφόσον επιλεγθεί το προαναφερθέν πεδίο ανοίγει ένα παράθυρο το οποίο αναφέρει στον χρήστη ότι με ένα χρηματικό αντίτιμο μπορεί να γίνει μέλος στην κοινότητα του blender.



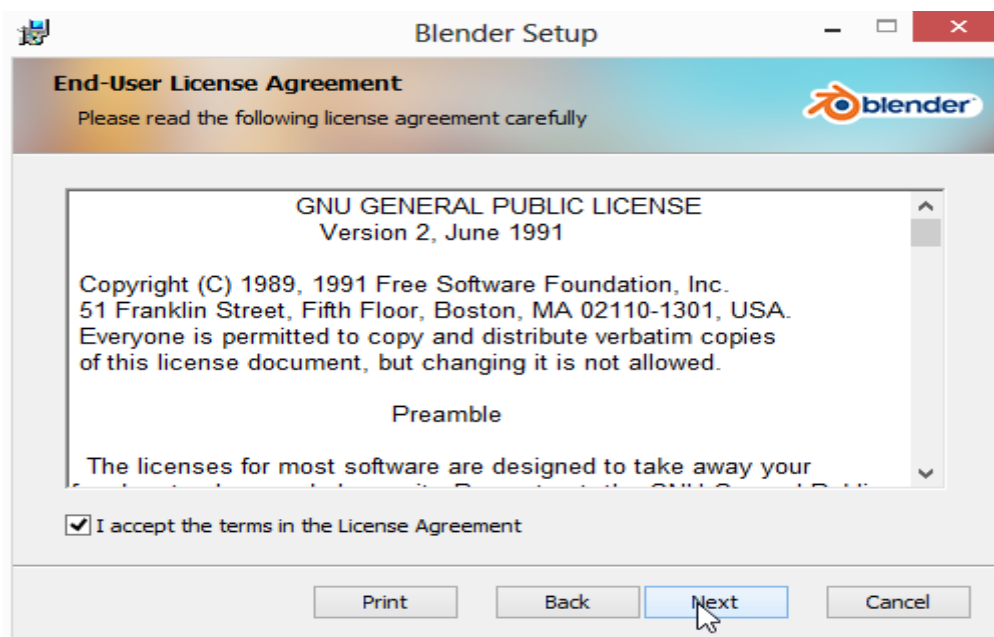
Εικόνα 4: Κοινότητα Blender.

Αφού ολοκληρωθεί η διαδικασία κατεβάσματος του προγράμματος, ξεκινάει η διαδικασία εγκατάστασης του. Από το πρώτο παράθυρο του οδηγού εγκατάστασης επιλέγετε το πεδίο next.



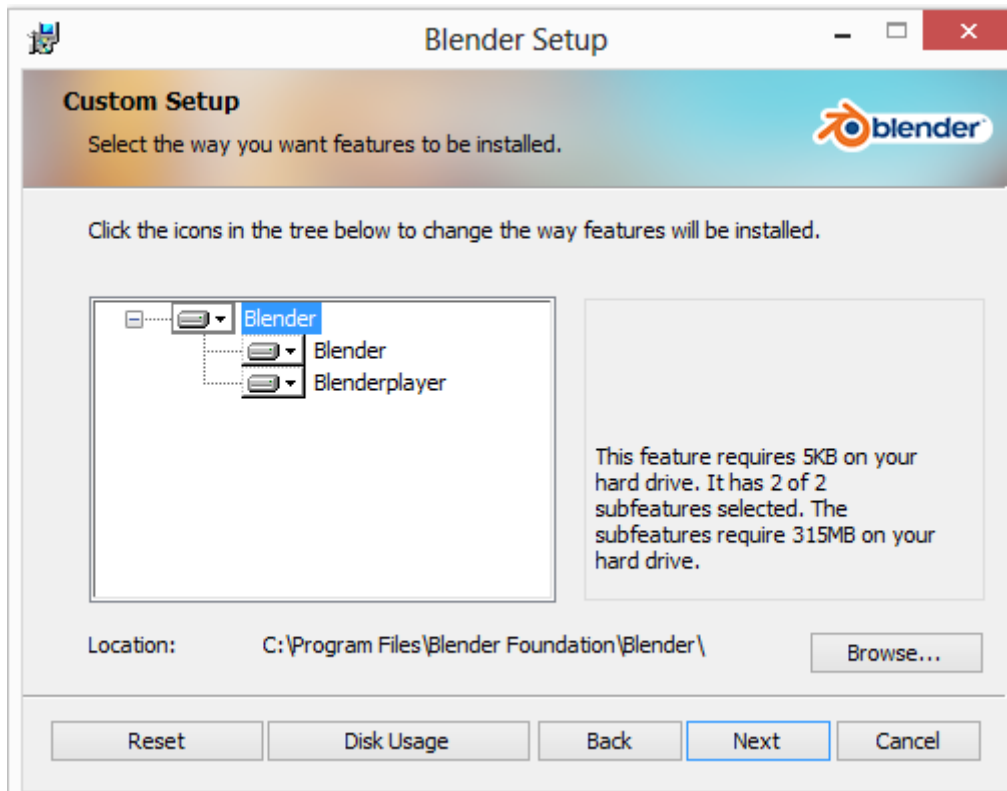
Εικόνα 5: Εκκίνηση εγκατάστασης Blender.

Στο επόμενο παράθυρο επιλέγετε τους όρους χρήσης του προγράμματος και μετά επιλέγετε το πεδίο next.



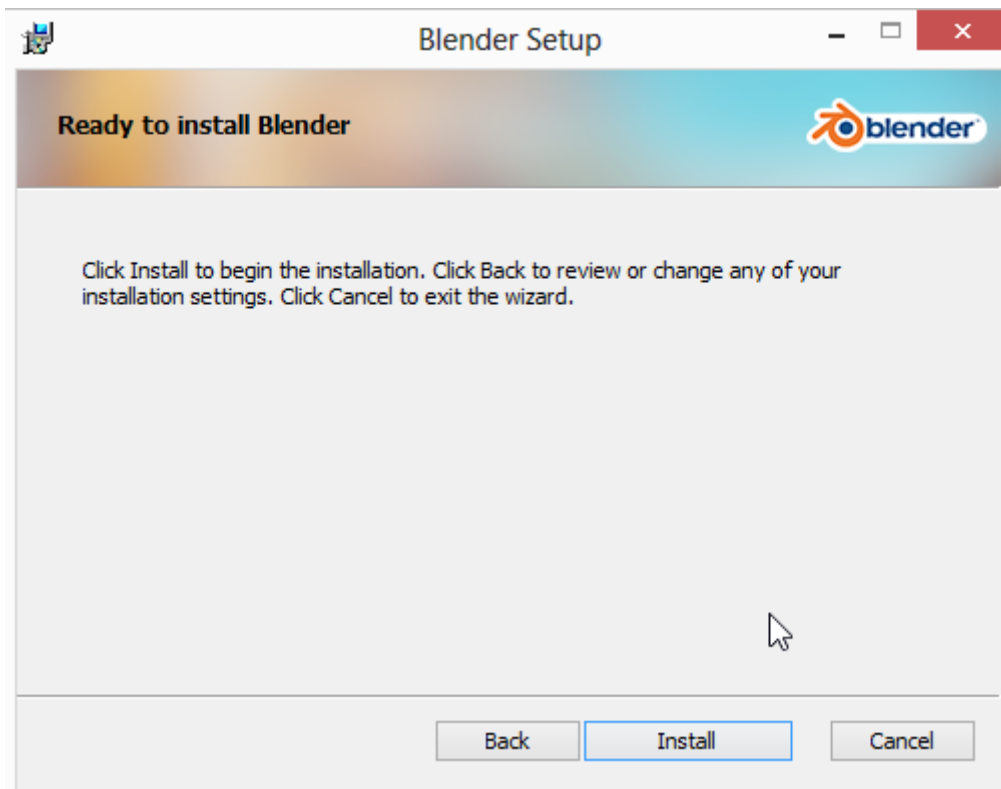
Εικόνα 6: Όροι χρήσης του προγράμματος.

Το επόμενο παράθυρο που εμφανίζεται καλεί το χρήστη να δηλώσει την διαδρομή εγκατάστασης του προγράμματος, ενώ και στο παρόν σημείο επιλέγεται το πεδίο Next προκειμένου να συνεχίσει η διαδικασία εγκατάστασης.



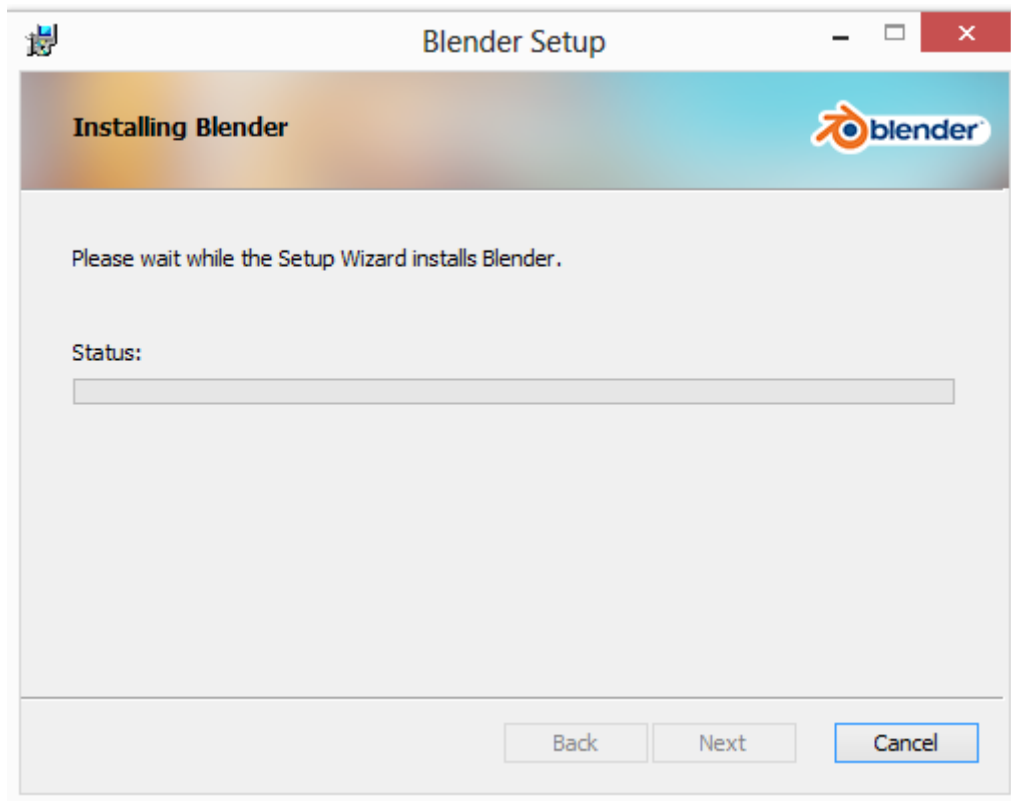
Εικόνα 7: Συνέχιση εγκατάστασης.

Το επόμενο παράθυρο του οδηγού εγκατάστασης οριστικοποιεί τη διαδικασία εγκατάστασης μέσω του πεδίου Install.



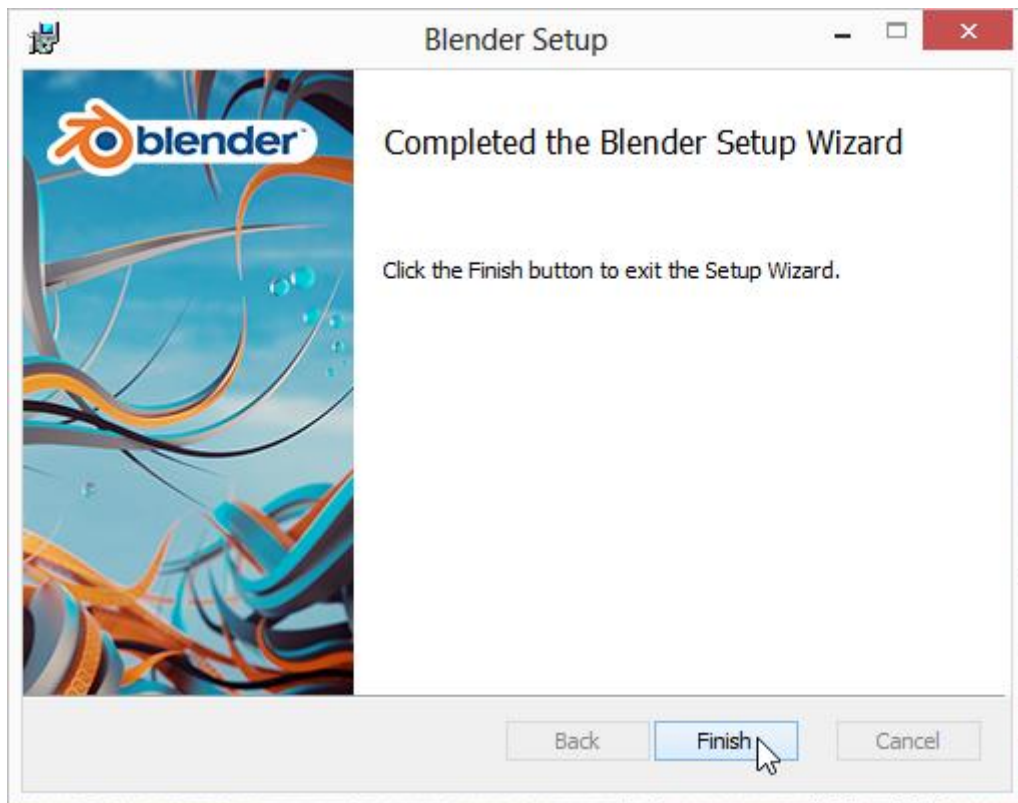
Εικόνα 8: Οριστικοποίηση έναρξης εγκατάστασης.

Ενώ στη συνέχεια ξεκινά η διαδικασία εγκατάστασης και παρουσιάζεται η πρόοδος της διαδικασίας.



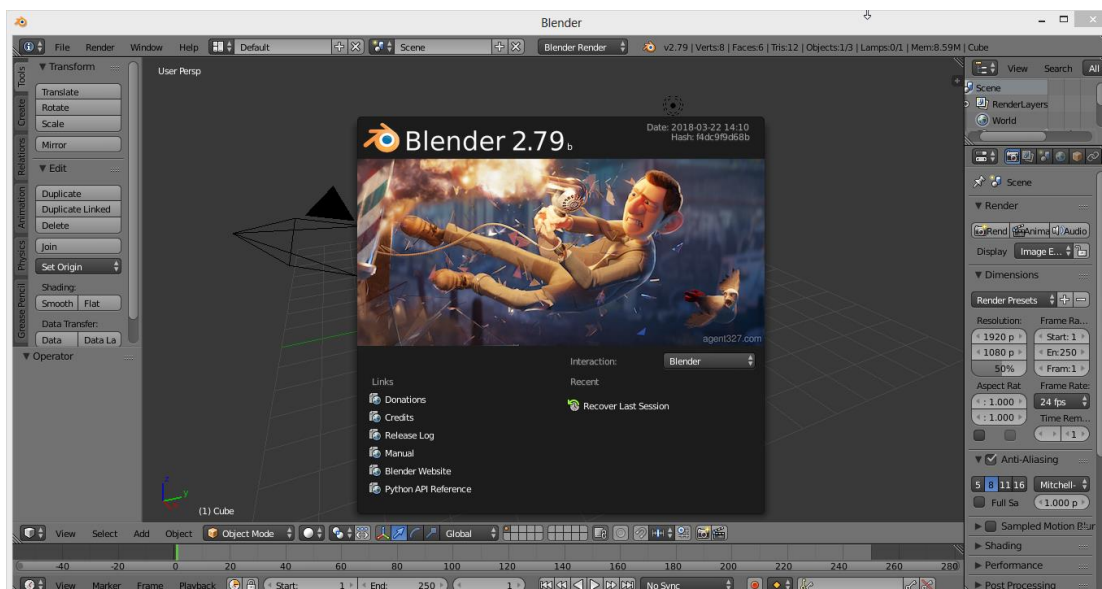
Εικόνα 9: Πρόοδος εγκατάστασης.

Όταν ολοκληρωθεί η εγκατάσταση του προγράμματος εμφανίζεται ένα παραθύρο που αναφέρει το πέρας της εγκατάστασης και μέσω του πεδίου Finish είναι δυνατή η εκκίνηση του προγράμματος.



Εικόνα 10: Ολοκλήρωση εγκατάστασης.

Εφόσον έχει εγκατασταθεί το πρόγραμμα μπορεί να ανοίξει.

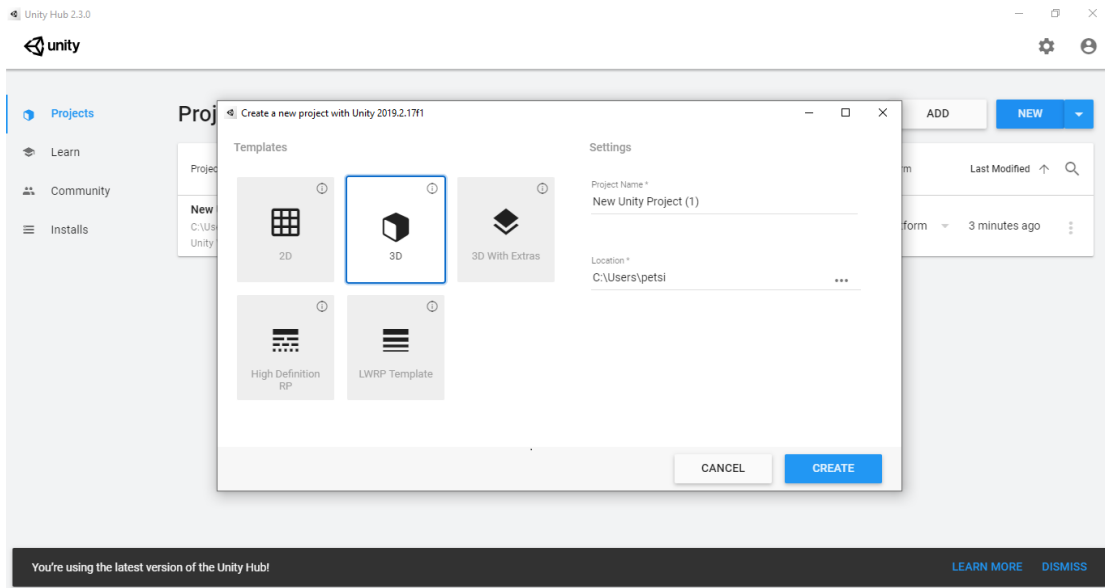


Εικόνα 11: Αρχική Οθόνη Blender.

ΚΕΦΑΛΑΙΟ 4 : ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΑΙΧΝΙΔΙΟΥ

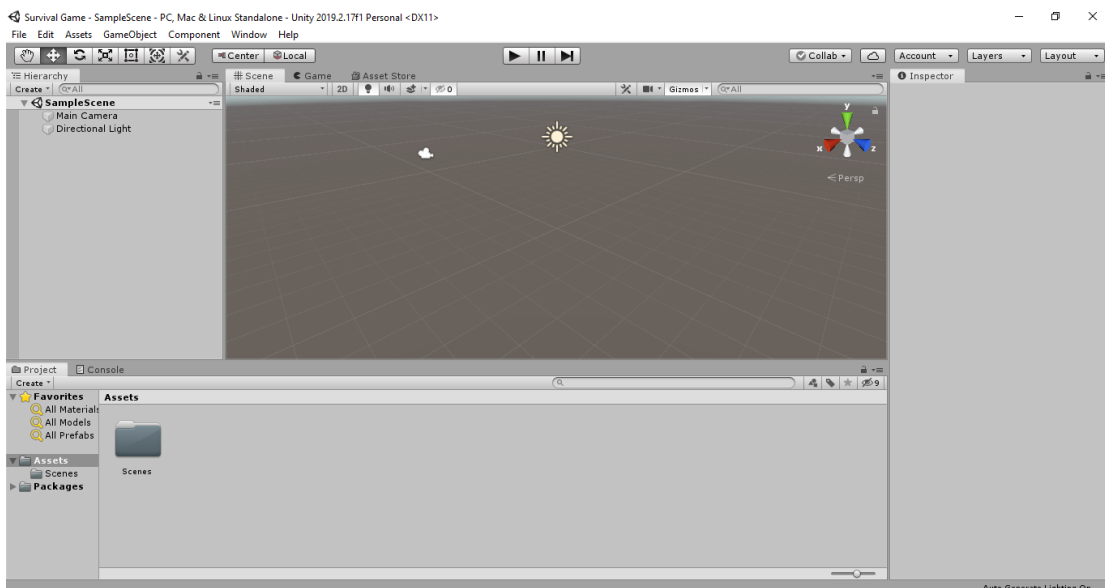
4.1 ΓΝΩΡΙΜΙΑ ΜΕ ΤΟ UNITY

Προκειμένου να δημιουργηθεί το παιχνίδι, αφότου άνοιξε η εφαρμογή του Unity επιλέχθηκε το πεδίο New Project. Έπειτα αφού επιλεγεί το όνομα του Project και ο χώρος αποθήκευσής του, κάνοντας κλικ στο πεδίο Create η εφαρμογή αρχίζει να δημιουργεί το Project.



Εικόνα 1: Επιλογή Νέου Project.

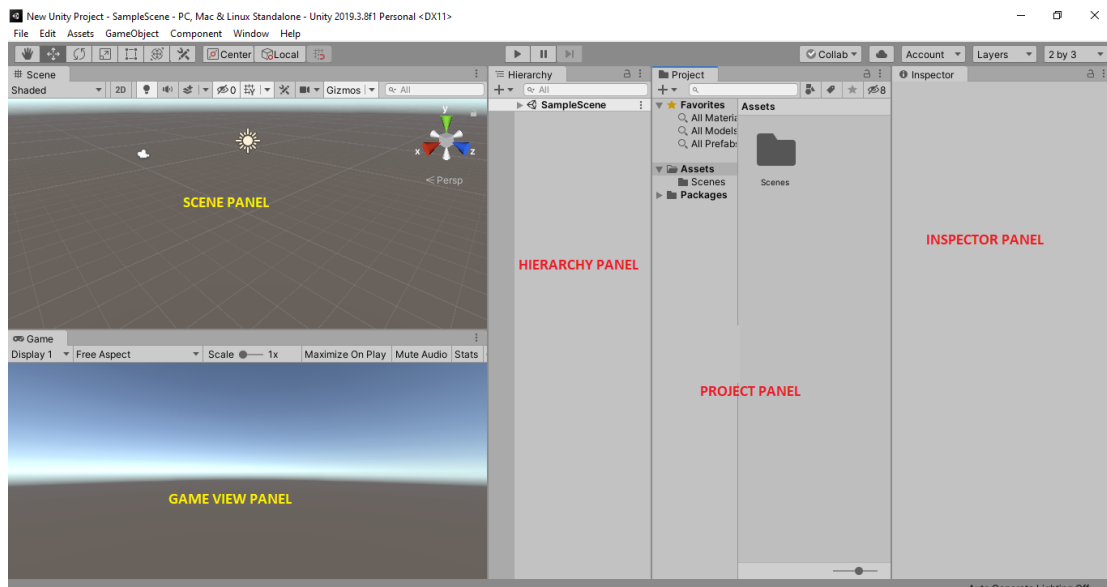
Μόλις δημιουργηθεί το Project ανοίγει η αρχική σκηνή της εφαρμογή πάνω στην οποία θα υλοποιηθεί το παιχνίδι.



Εικόνα 2: Αρχική σκηνή Unity.

Στην σκηνή αυτή υπάρχουν διάφορα Panel τα οποία ο κάθε χρήστης τα διαμορφώνει με τον τρόπο που αυτός κρίνει ότι τον βολεύει να δημιουργήσει το Project του καλύτερα,. Πατώντας επάνω δεξιά στο πεδίο Layout υπάρχουν διάφορες επιλογές για τον τρόπο διάταξης της σκηνής. Το κάθε Panel εμπεριέχει μέσα διάφορα στοιχεία που βοηθάνε τον χρήστη να διαχωρίζει το Project του. Τα πιο βασικά Panel είναι τα εξής:

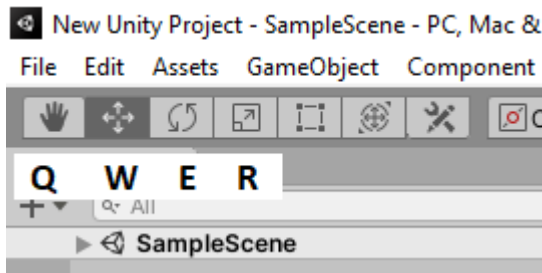
- **Scene ViewPanel**
- **Game View Panel**
- **Hierarchy Panel**
- **Project Panel**
- **Inspector Panel**



Εικόνα 3: Layout αρχικής σκηνής στο Unity.

4.1.1 Scene View

Το Scene View είναι ο κύριος χώρος κατασκευής του παιχνιδιού. Υπάρχουν πλήκτρα που κάνουν πιο γρήγορη την εναλλαγή εργαλείων και έτσι κάνουν το Unity πιο εύκολο στην χρήση. Τα πιο σημαντικά πλήκτρα είναι τα **Q,W,E,R** και βρίσκονται επάνω αριστερά στην αρχική σκηνή.



Εικόνα 4: Συντομεύσεις εργαλείων Unity.

Hand Tool (Q) : Μέσω αυτού του πλήκτρου ο χρήστης μπορεί πατώντας το αριστερό κλικ του ποντικιού, να περιηγηθεί στην σκηνή του και να δει από διαφορετικές γωνίες το Project που υλοποιεί. Κρατώντας πατημένο το δεξί κλικ μπορεί να μετακινήσει την κάμερα προς όποια μεριά θελήσει, αρκεί να μετακινήσει το ποντίκι του.

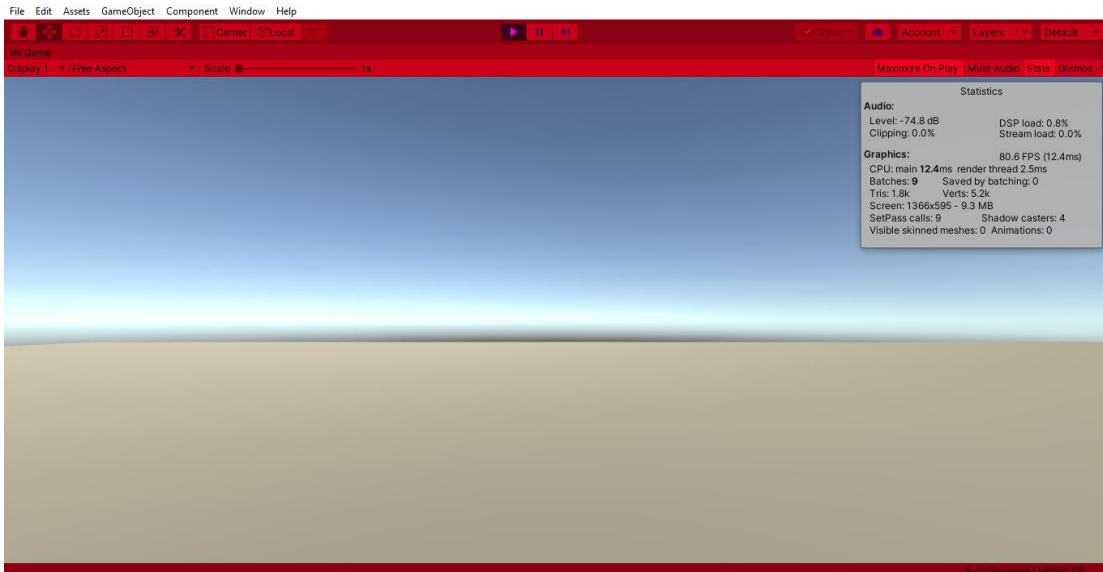
Move Tool (W) : Είναι το πλήκτρο όπου ο χρήστης πατώντας επάνω σε κάποιο αντικείμενο του Project του μπορεί να το μετακινήσει πάνω στους άξονες X, Y, Z.

Rotate Tool (E) : Επιτρέπει στον χρήστη να περιστρέψει το επιλεγμένο αντικείμενο προς τους τρεις άξονες που προαναφέραμε.

Scale Tool (R) : Το πλήκτρο το οποίο μπορεί ο χρήστης να αλλάξει τις διαστάσεις σε ένα επιλεγμένο αντικείμενο σύμφωνα πάλι με τις τρεις διαστάσεις.

4.1.2 Game View

Το GameView δείχνει στο χρήστη πως θα είναι το παιχνίδι που κατασκευάζει σύμφωνα με την κάμερα την οποία έχει ενσωματώσει. Διαθέτει τρεις επιλογές, το Play, το Pause και το Step. Υφίσταται η προσοχή του χρήστη διότι οι αλλαγές που κάνει όταν τρέχει το παιχνίδι, του είτε έχει πατήσει Play είτε Pause δεν αποθηκεύονται με συνέπεια όταν βγει από το GameMode να χάνονται. Για αυτό τον λόγο πολλοί χρήστες επιλέγουν να αλλάζουν το Background χρώμα έτσι ώστε να μην μπερδεύονται όταν έχουν πατημένο το πλήκτρο Play. Επίσης υπάρχουν τα πλήκτρα MaximizeOnPlay όπου όταν πατηθεί το πλήκτρο Play, το παιχνίδι θα ανοίγει σε πλήρη οθόνη. Βασικό εργαλείο είναι και το Stats το οποίο δείχνει τα στατιστικά γραφικών του παιχνιδιού.



Εικόνα 5: GameView όταν το πλήκτρο Play είναι επιλεγμένο.

Play :Το πλήκτρο PLAY μπορεί να χρησιμοποιηθεί οποιαδήποτε στιγμή θελήσει ο χρήστης. Το πάτημα του πλήκτρου μεταφέρει τον χρήστη στο παιχνίδι που δημιουργεί και ουσιαστικά τρέχει το παιχνίδι.

Pause :Με αυτό το πλήκτρο παγώνει το παιχνίδι έτσι ώστε ο χρήστης να μπορέσει να διακρίνει όποια λεπτομέρεια θελήσει ή να αλλάξει κάποια τιμή όπως για παράδειγμα η ταχύτητα, ο φωτισμός ή ότι άλλο κρίνει εκείνος απαραίτητο.

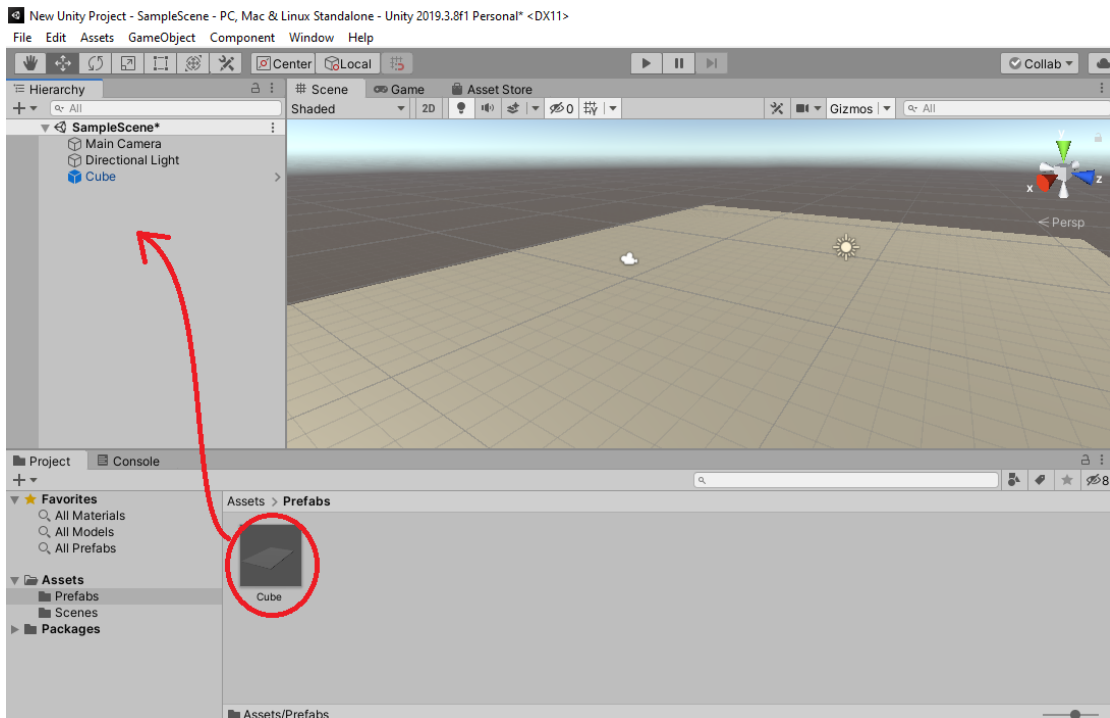
Step :Επιτρέπει στον χρήστη να πηγαίνει στο επόμενο Frame έτσι ώστε να κάνει ότι αλλαγή κρίνει απαραίτητη.

4.1.3 Hierarchy Panel

Είναι το παράθυρο που εμπεριέχει όλα τα αντικείμενα που βρίσκονται στο ScenePanel. Από εκεί ο χρήστης μπορεί να αλλάξει ονόματα ή τιμές ή οτιδήποτε θελήσει στο παιχνίδι του. Όλες οι πληροφορίες που χρειάζεται το παιχνίδι για να ξεκινήσει όταν πατηθεί το πλήκτρο Play αντλούνται από το συγκεκριμένο Panel.

4.1.4 Project Panel

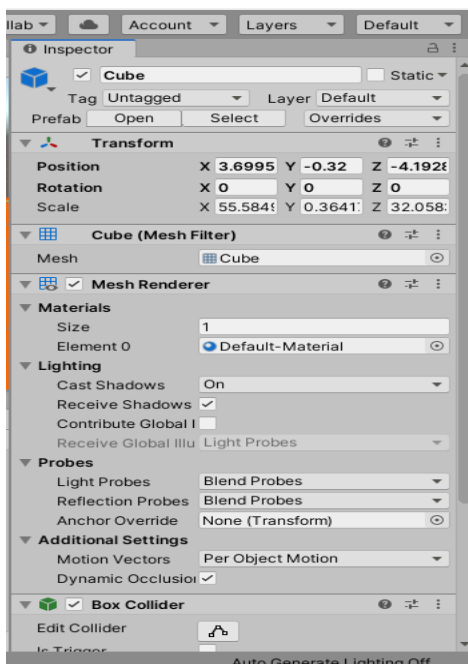
Υπάρχουν όλα τα Assets του παιχνιδιού. Ο χρήστης εκεί μπορεί να αποθηκεύει διάφορα αντικείμενα (Scripts, Materials, Prefabs, Animations κ.α.) και να τα χρησιμοποιεί όποτε αυτός κρίνει ότι χρειάζεται με τη μέθοδο Drag And Drop από το Project Panel στο Hierarchy Panel.



Εικόνα 6: Μεταφορά αντικειμένου απο Project Panel σε Hierarchy Panel.

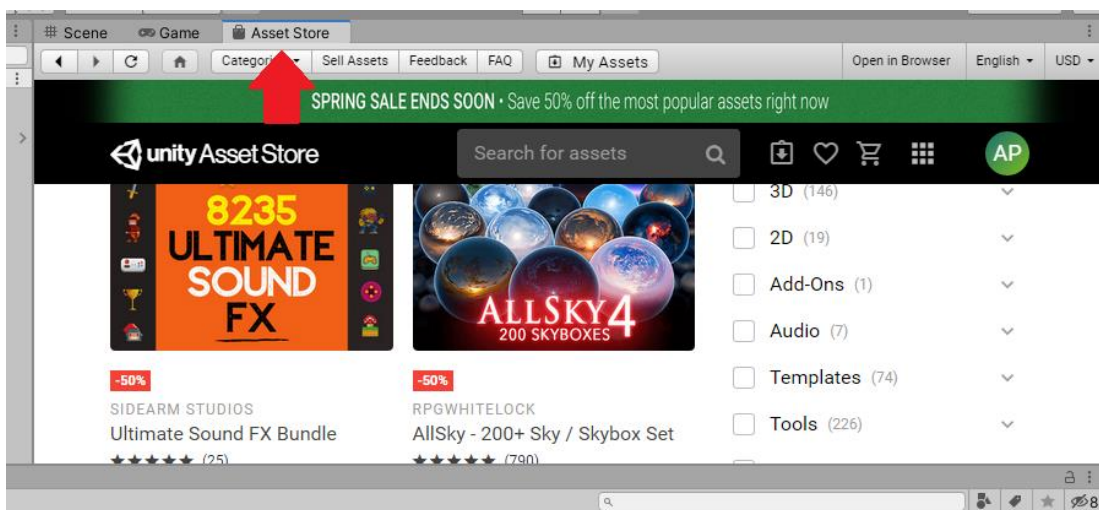
4.1.5 Inspector Panel

Ο χρήστης μπορεί να δει όλα τα χαρακτηριστικά του αντικειμένου που έχει δημιουργήσει, είτε αυτό είναι στο Project Panel είτε στο Hierarchy Panel. Επίσης μπορεί να αλλάξει ή να προσθέσει χαρακτηριστικά όπως Scripts στο αντικείμενο που έχει επιλέξει.



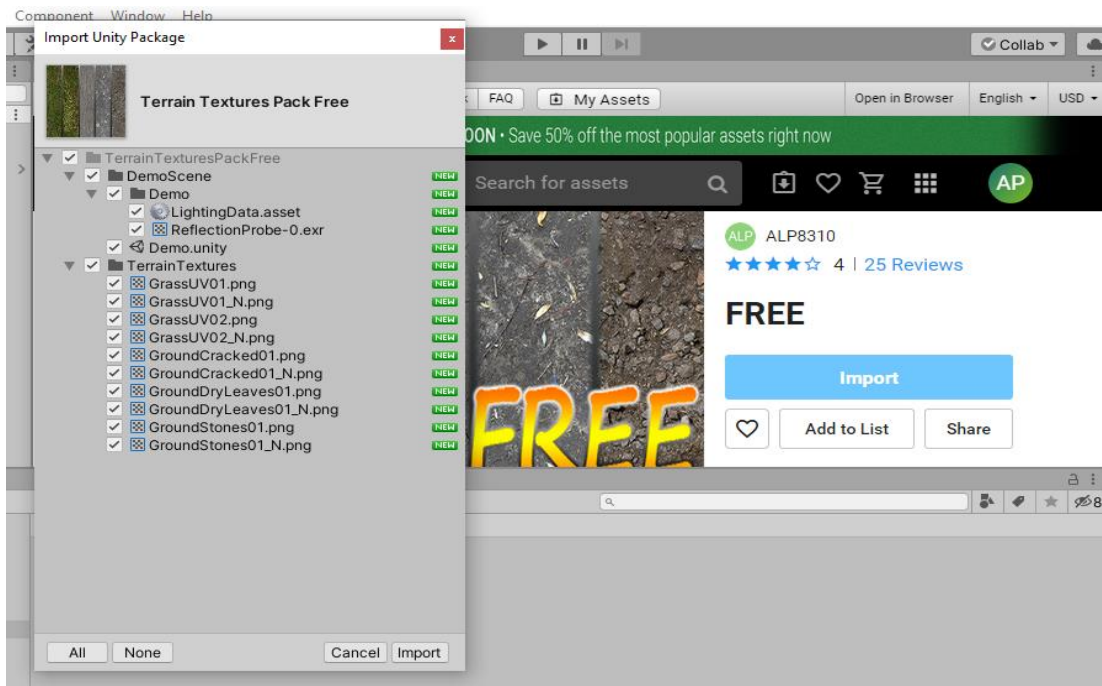
Εικόνα 7: Inspector Panel.

Στην αρχική σκηνή του Unity ο χρήστης θα παρατηρήσει μια καρτέλα που ονομάζεται Asset Store. Αυτή η επιλογή δίνει πρόσβαση στον κάθε χρήστη να μπει απευθείας από την σκηνή του στη σελίδα του Unity Asset Store στο ίντερνετ και να κατεβάσει εφόσον έχει κάνει Login, χιλιάδες Assets που έχουν δημιουργηθεί από άλλους χρήστες. Υπάρχουν δωρεάν Assets αλλά τα περισσότερα είναι έναντι κάποιου χρηματικού ποσού.



Εικόνα 8: Αρχική σελίδα Unity Asset Store.

Για κάποιον αρχάριο χρήστη το Asset Store είναι φοβερό εργαλείο και μπορεί να του λύσει τα χέρια από την στιγμή που περιέχει τόσες πολλές επιλογές για κατέβασμα. Επίσης είναι πολύ εύκολη η μεταφορά των Assets από την σελίδα στην σκηνή του κάθε χρήστη. Εφόσον βρει το Asset που τον ενδιαφέρει, το κατεβάζει και έπειτα πατώντας Import μπορεί να το φορτώσει κατευθείαν στην σκηνή του. Παράλληλα υπάρχει και η δυνατότητα αποεπιλογής ορισμένων Assets από αυτά που εμπεριέχονται στο πακέτο, έτσι ο χρήστης δεν φορτώνει την σκηνή του με άχρηστα για αυτόν Assets.



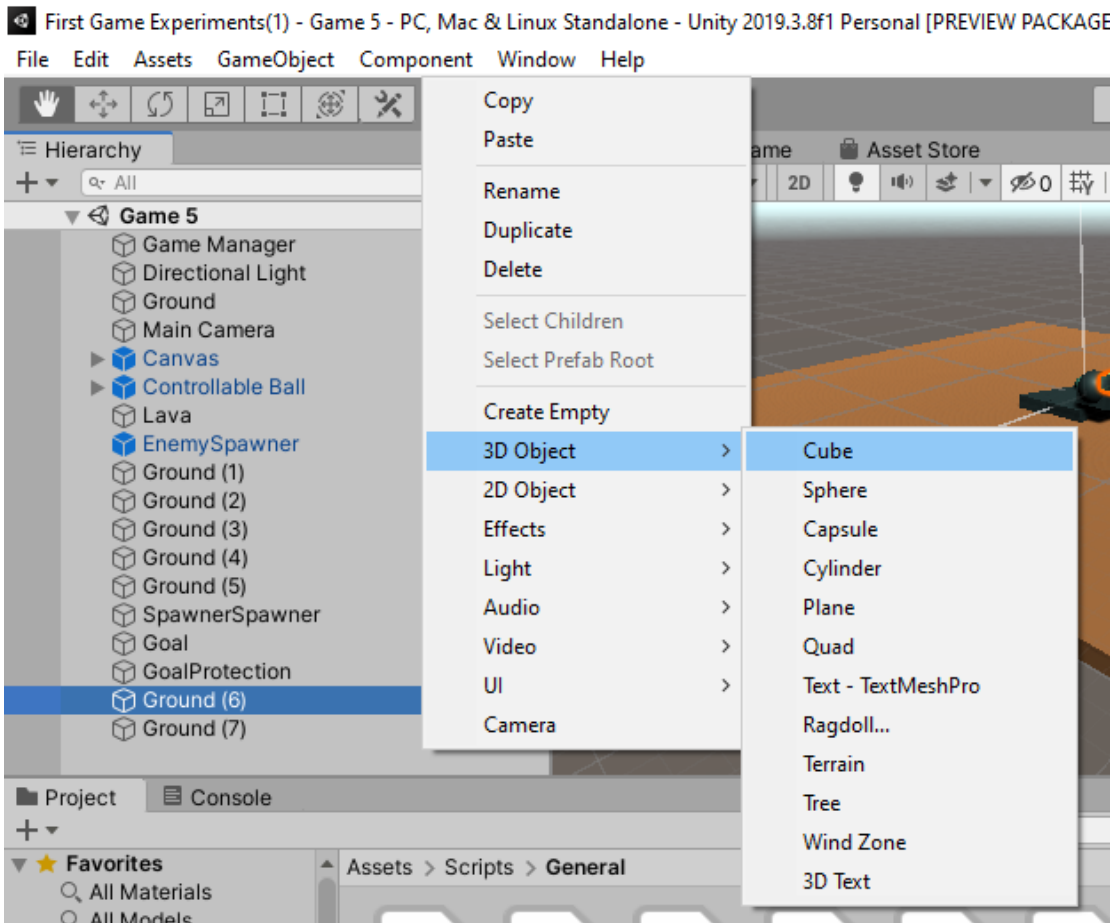
Εικόνα 9: Επιλογή Asset που θα περαστούν στην σκηνή.

4.2 ΠΡΟΛΟΓΟΣ ΠΑΙΧΝΙΔΙΟΥ

Εφόσον έχει ανοίξει η εφαρμογή του Unity και έχει δημιουργηθεί το αρχικό Project, ήρθε η ώρα να αρχίσουν να προστίθενται τα κατάλληλα Assets για τη δημιουργία του παιχνιδιού. Η ονομασία του παιχνιδιού είναι **Rolling Balls 3D** και αποτελείται από έξι σκηνές (Levels) όπου είναι single player και μια σκηνή (Level) που είναι 1 vs 1. Στόχος του παιχνιδιού είναι ο χρήστης να μετακινεί την μπάλα κατάλληλα περνώντας τα διάφορα εμπόδια που βρίσκει μέχρι να φτάσει στον τερματισμό. Στο τέλος κάθε Level υπάρχει ο τερματισμός, όπου επιτρέπει στο χρήστη να μεταβεί στην επόμενη σκηνή (Level).

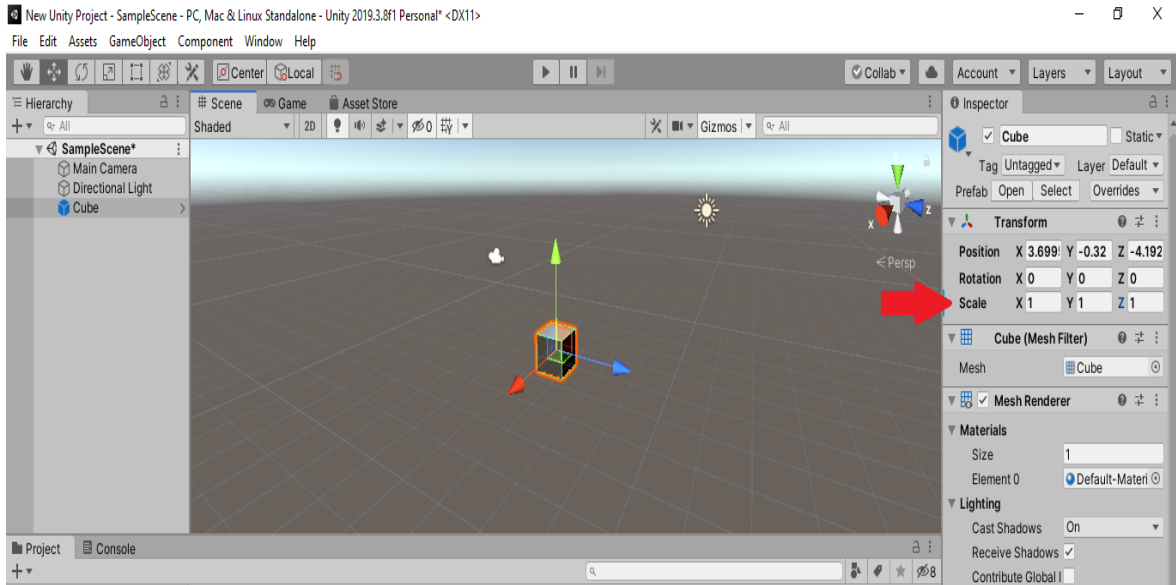
4.3 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΕΔΑΦΟΥΣ

Για τη δημιουργία του εδάφους του παιχνιδιού χρησιμοποιήθηκε ένα από τα έτοιμα αντικείμενα του Unity, ο κύβος. Το Unity δίνει στους χρήστες την ευκολία να δημιουργήσουν και να τροποποιήσουν διάφορα 3D αντικείμενα όπως είναι ο κύβος, η σφαίρα, η κάψουλα και πολλά άλλα. Κάνοντας δεξί κλικ στο Hierarchy Panel, στο μενού που παρουσιάζεται υπάρχει η επιλογή 3D Object, όταν επιλεγθεί η παραπάνω επιλογή ανοίγει και άλλο ένα μενού με τα διάφορα Object όπου μπορούν να δημιουργηθούν.



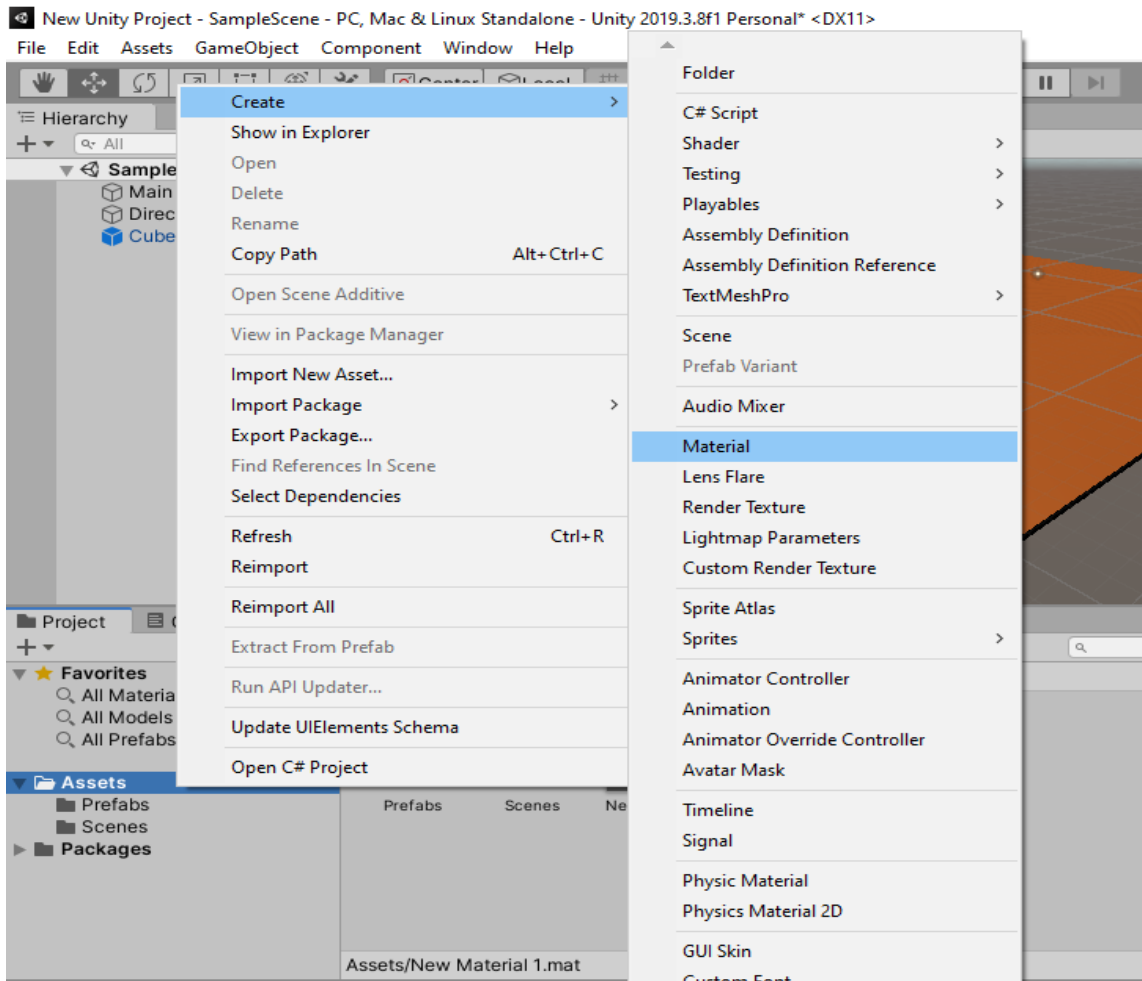
Εικόνα 10: Δημιουργία 3D Object.

Όπως αναφέρθηκε σαν έδαφος χρησιμοποιήθηκε ο κύβος. Μόλις δημιουργηθεί το αντικείμενο υπάρχει η δυνατότητα τροποποίησης του στην καρτέλα Inspector. Οι τιμές στο Scale θα πρέπει να είναι μεγάλες στους άξονες X, Z μιας και θα πρέπει να καλύψει μεγάλη επιφάνεια. Ο άξονας Y θα πρέπει να παραμείνει σε τιμές χαμηλότερες του 1, κυρίως για αισθητικούς λόγους μιας και όταν θα παίζεται το παιχνίδι δεν θα φαίνεται αυτή η διάσταση.



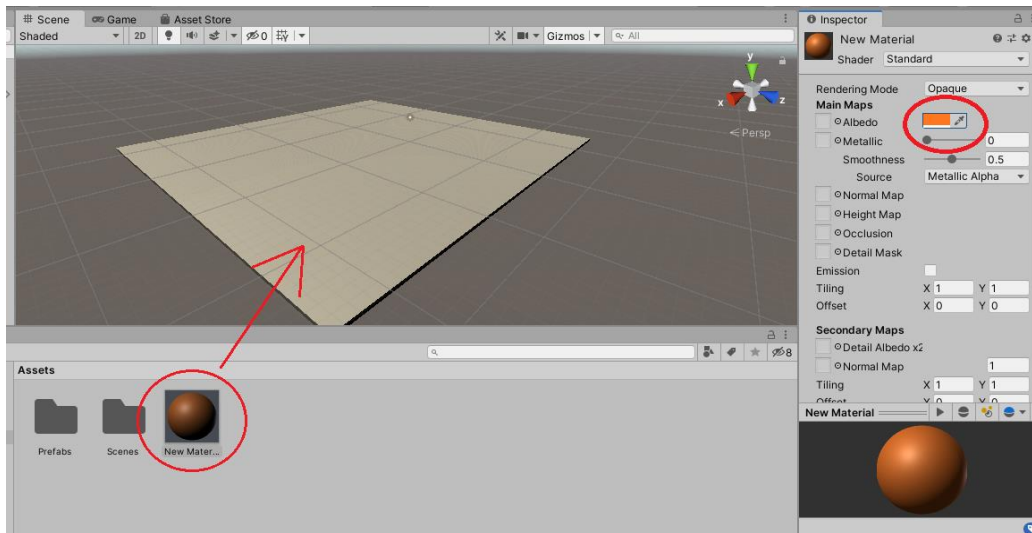
Εικόνα 11: Αλλαγή στοιχείων του εδάφους.

Για να αλλαχθεί χρώμα στο έδαφος και σε κάθε αντικείμενο που δημιουργείτε πρέπει να υπάρχει και το κατάλληλο Material. Τα Materials είτε βρίσκονται έτοιμα στο Unity Asset Store είτε τα δημιουργεί ο χρήστης κάνοντας δεξί κλικ στον φάκελο Assets που βρίσκεται στο Project Panel. Έπειτα στην επιλογή Create ανοίγει ένα νέο μενού επιλογών όπου ο χρήστης θα βρει τη λέξη Material. Πατώντας κλικ θα δημιουργηθεί το νέο Material όπου μπορεί να το τροποποιήσει μέσω του Inspector Panel. Ύστερα με την μέθοδο drag and drop μπορεί να το προσθέσει σε όποιο αντικείμενο επιθυμεί.



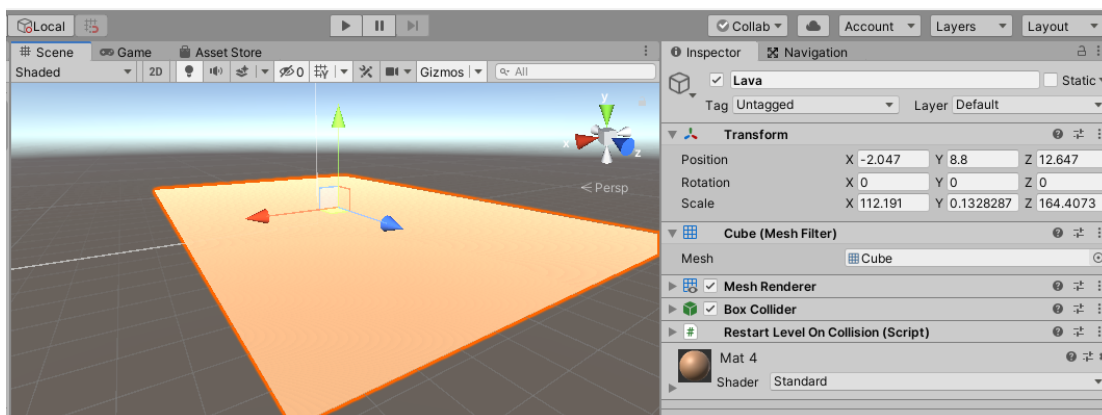
Εικόνα 14: Δημιουργία Material.

Αφού δημιουργηθεί το Material στο Inspector Panel υπάρχει η επιλογή Albedo. Από εκεί μπορεί να αλλαχθεί το χρώμα από το Material, επίσης υπάρχουν οι επιλογές Metallic και Smoothness που αλλάζοντας τιμές επηρεάζεται η όψη του Material.



Εικόνα 15: Προσθήκη Material στο αντικείμενο.

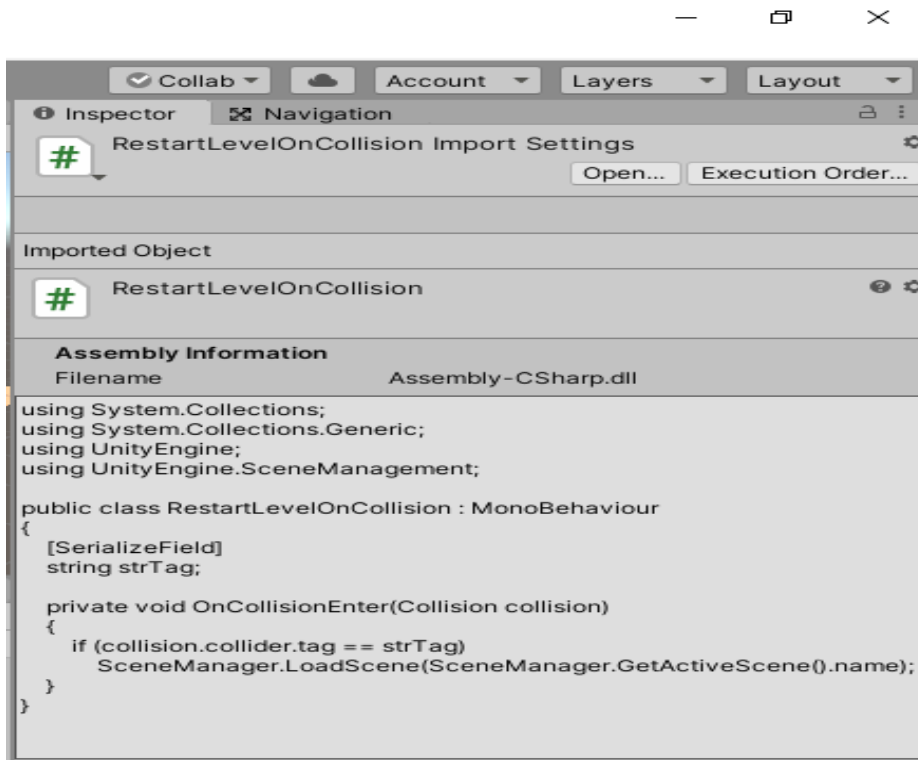
Από την καρτέλα του Inspector επίσης μπορεί να αλλαχθεί το όνομα του κύβου σε Lava, αυτό βοηθάει τον χρήστη να βρίσκει πιο γρήγορα τα αντικείμενα του στο Hierarchy Panel. Μπορεί επίσης να προστεθεί το κατάλληλο Script έτσι ώστε όταν ο παίχτης – μπάλα αποκτήσει επαφή με το έδαφος η σκηνή να αρχίζει από την αρχή.



Εικόνα 12: Τελική μορφή του εδάφους.

Ο χρήστης μπορεί να προσθέσει το Script στο αντικείμενο με δύο τρόπους. Είτε με drag and drop από το Project Panel στο Inspector Panel είτε με την επιλογή Add Component–New Script που βρίσκεται στο Inspector Panel. Στο ProjectPanel μπορεί να δημιουργήσει Script κάνοντας δεξιά κλικ Create–C#Script και έπειτα του δίνει το όνομα που αυτός επιθυμεί. Τα Scripts φτιάχνονται σε γλώσσα C# σε ξεχωριστό πρόγραμμα από το Unity. Εάν τα Scripts έχουν λάθη όταν ο χρήστης πατήσει Play το Unity δεν θα ξεκινήσει το παιχνίδι και θα του βγάλει Error, γι' αυτό χρειάζεται πολύ προσοχή ο κώδικας να είναι σωστά γραμμένος. Το

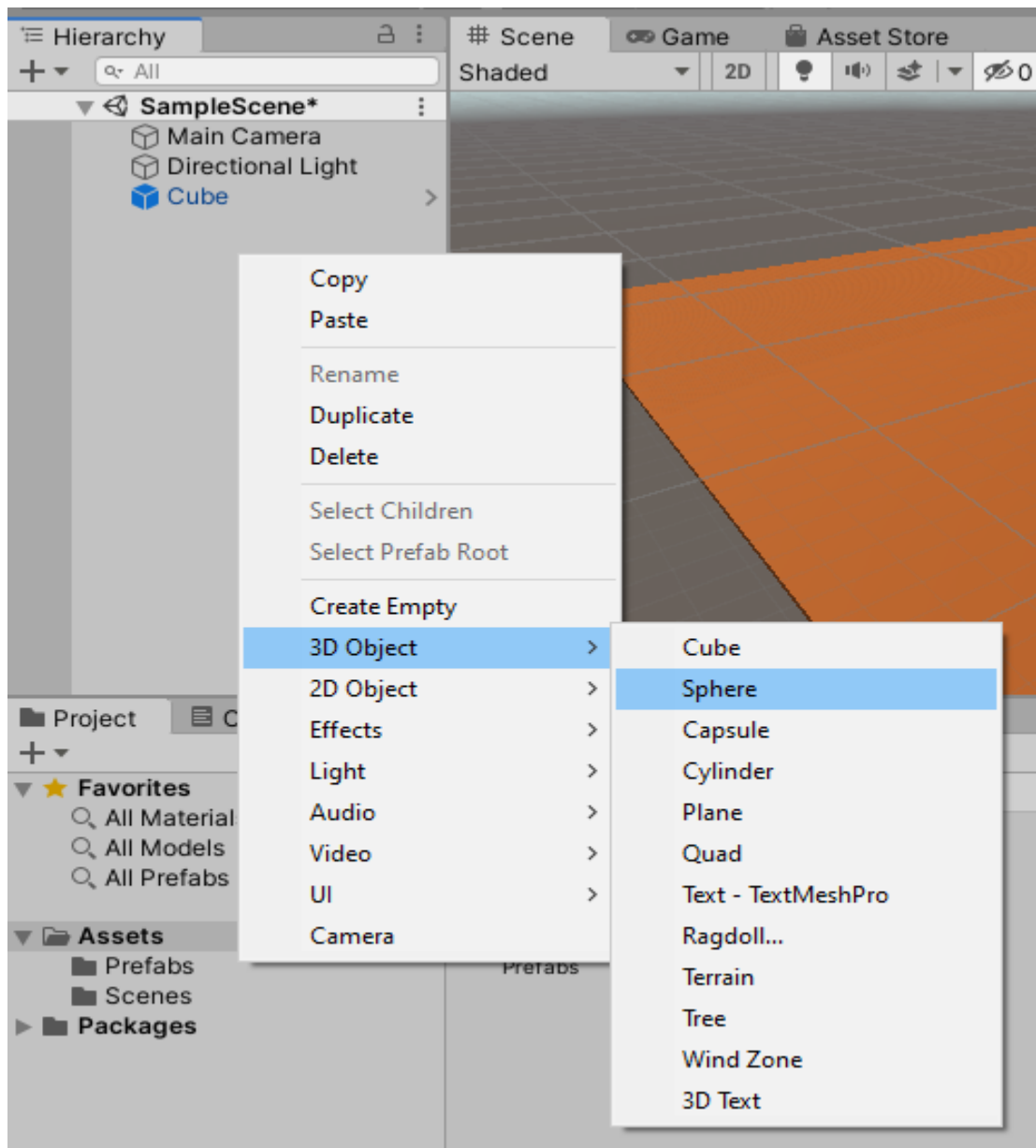
Script εδάφους είναι πολύ απλό και το μόνο που κάνει είναι να φορτώνει ξανά την σκηνή όταν ο παίχτης - μπάλα ακουμπήσει επάνω του.



Εικόνα 13: Script εδάφους.

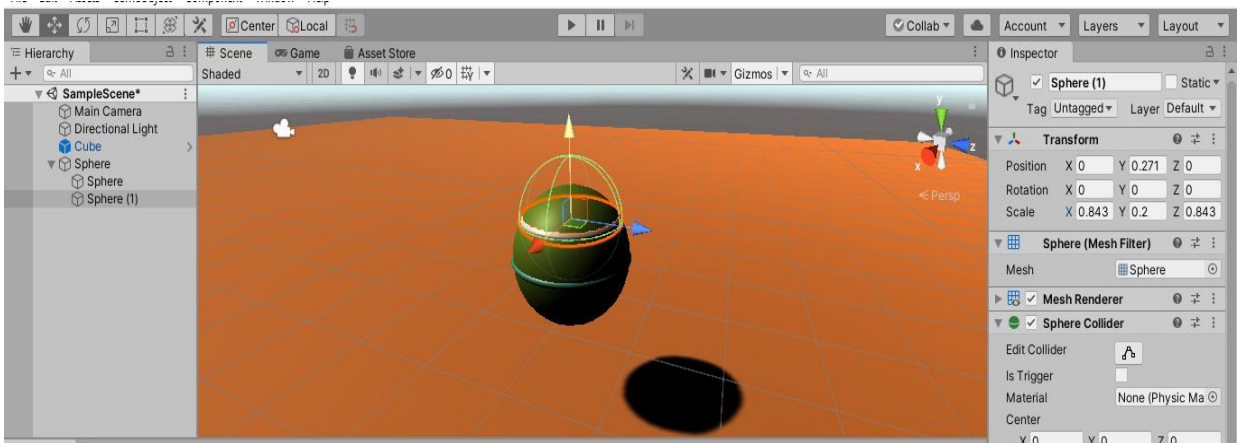
4.4 ΔΗΜΙΟΥΡΓΙΑ ΧΑΡΑΚΤΗΡΑ – PLAYER

Η διαδικασία δημιουργίας του βασικού παίχτη είναι σχεδόν παρόμοια με αυτή του εδάφους –terrain με την μόνη διαφορά ότι τώρα δημιουργήθηκε μια σφαίρα αντί για έναν κύβο. Όπως έχει προαναφερθεί η σφαίρα όπως και ο κύβος είναι στα έτοιμα αντικείμενα που δίνει στη διάθεση του χρήστη η εφαρμογή του Unity.



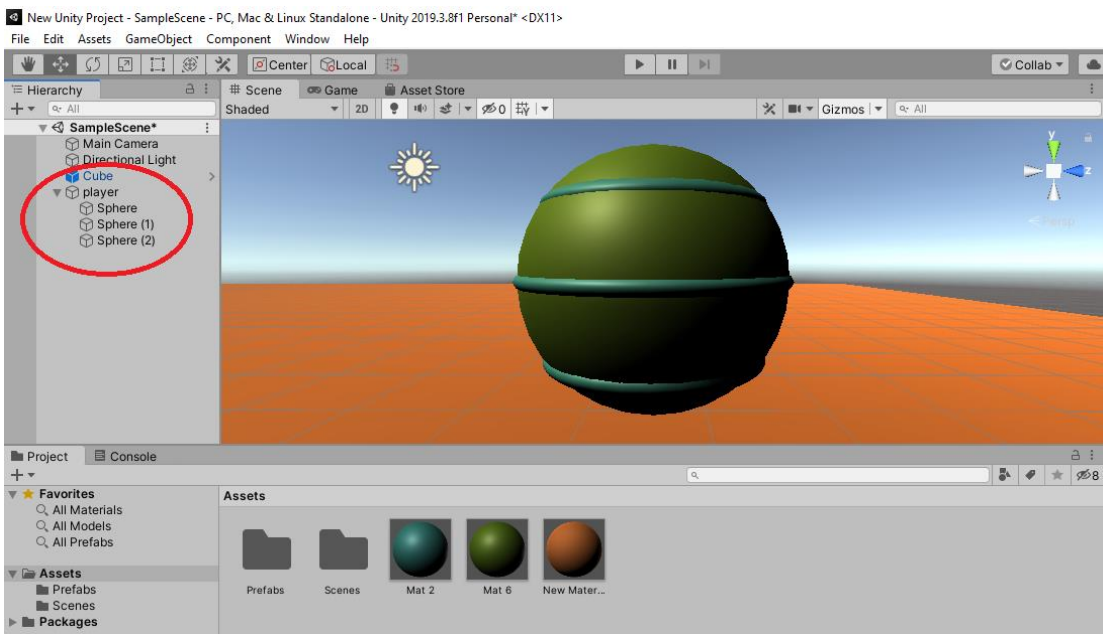
Εικόνα 1: Δημιουργία σφαίρας – player.

Για αισθητικούς λόγους κυρίως προστέθηκαν στην σφαίρα – παίχτη άλλες τρεις σφαίρες με πολύ μικρότερες διαστάσεις έτσι ώστε να φαίνεται πιο όμορφος ο παίχτης. Δημιουργήθηκαν δύο νέα Materials σε διαφορετικά χρώματα και προστέθηκαν και αυτά στον παίχτη.



Εικόνα 2: Προσαρμογή Details στον παίχτη.

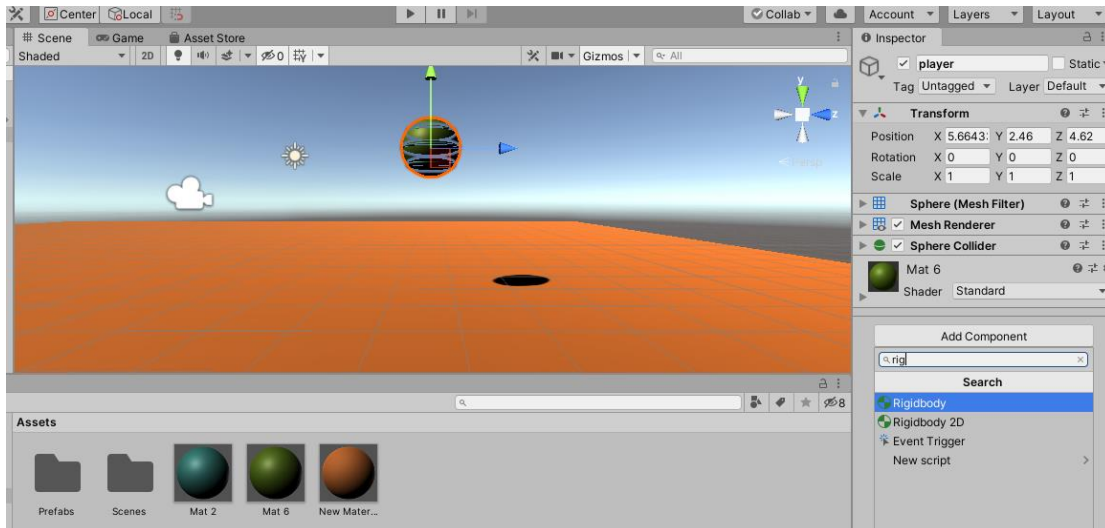
Οι τρεις νέες σφαίρες που δημιουργήθηκαν θα πρέπει να τοποθετηθούν στον φάκελο του παίχτη στο Hierarchy Panel. Με αυτόν τον τρόπο γίνονται « παιδιά » του παίχτη. Αυτή η μέθοδος ονομάζεται ‘ Parenting ’ και είναι μια από τις βασικές έννοιες της εφαρμογής του Unity. Επίσης αλλάχθηκε και το όνομα του αντικειμένου από σφαίρα σε Player για να υπάρχει μια συνοχή και να δείχνει πιο οργανωμένο το Hierarchy Panel.



Εικόνα 3: Παράδειγμα ‘ Parenting ’ και τελική μορφή παίχτη.

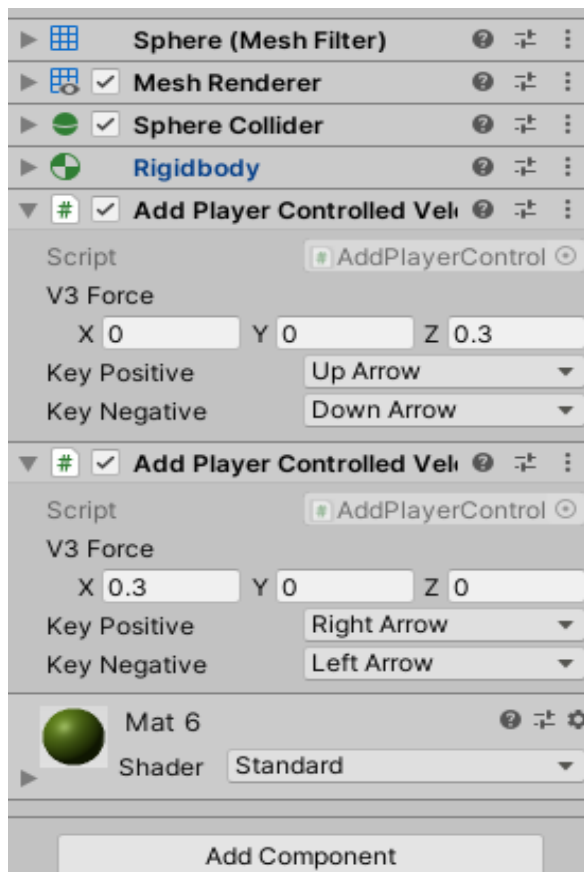
Όπως φαίνεται στην παραπάνω εικόνα (3) ο χαρακτήρας πήρε την τελική μορφή του αλλά δε διαφέρει σε τίποτα από ένα κοινό αντικείμενο του Unity. Αν πατηθεί το πλήκτρο Play ο χρήστης παρατηρεί ότι ο χαρακτήρας του παραμένει εκεί που βρίσκεται δίχως να επηρεάζεται από κάποια φυσική δύναμη. Για να μπορεί ο χρήστης να ασκεί δύναμη στον

παίχτη είτε βαρυτική είτε με κάποιο Script κίνησης θα πρέπει να περάσει στον χαρακτήρα του ένα σημαντικό Component που ονομάζεται Rigidbody και βρίσκεται στο Inspector Panel – Add Component – Rigidbody.



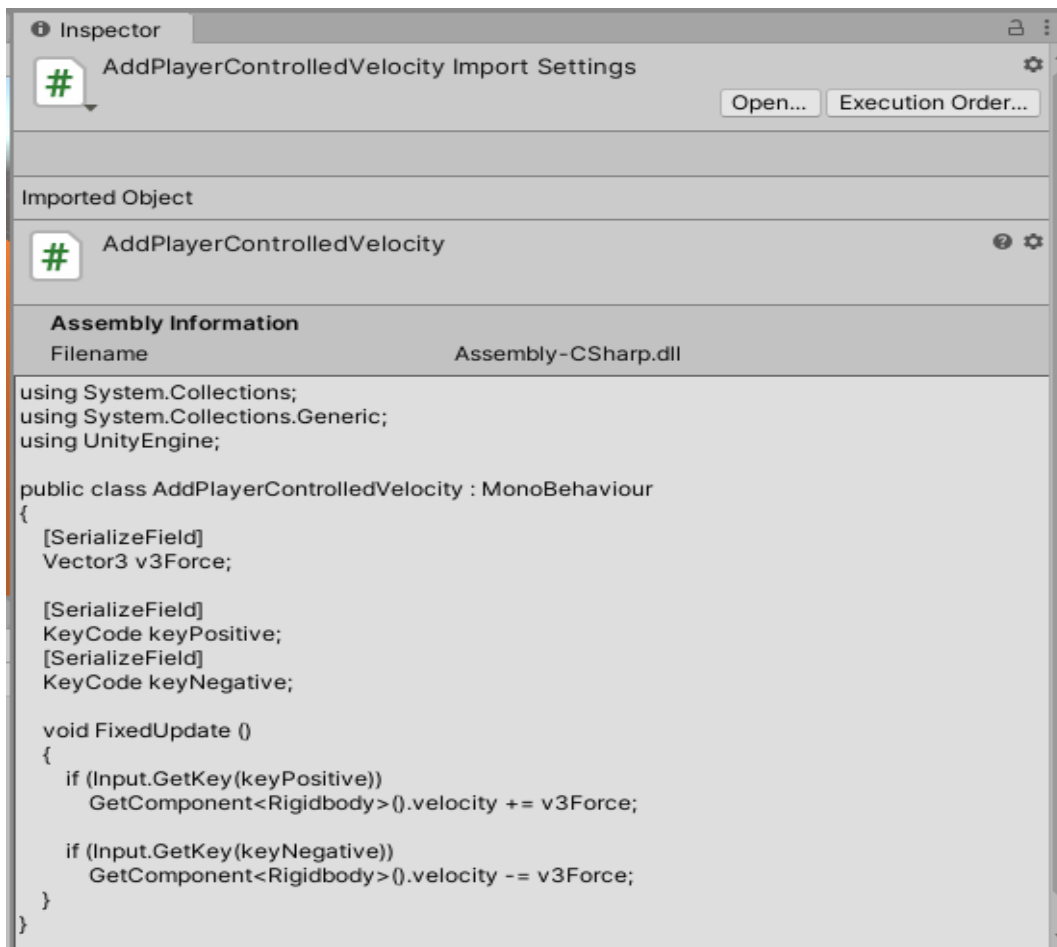
Εικόνα 4: Προσθήκη Component (Rigidbody).

Πλέον με το πάτημα του πλήκτρου Play το αντικείμενο – player θα δέχεται βαρυτική δύναμη και απλά θα πέφτει στο έδαφος που έχει δημιουργηθεί. Για να μπορεί ο χρήστης να κατευθύνει το αντικείμενο – player θα πρέπει να δημιουργήσει δύο ίδια Script. Ο παίχτης θα μετακινείται με τα βελάκια του πληκτρολογίου.



Εικόνα 5: Προσθήκη Script κίνησης.

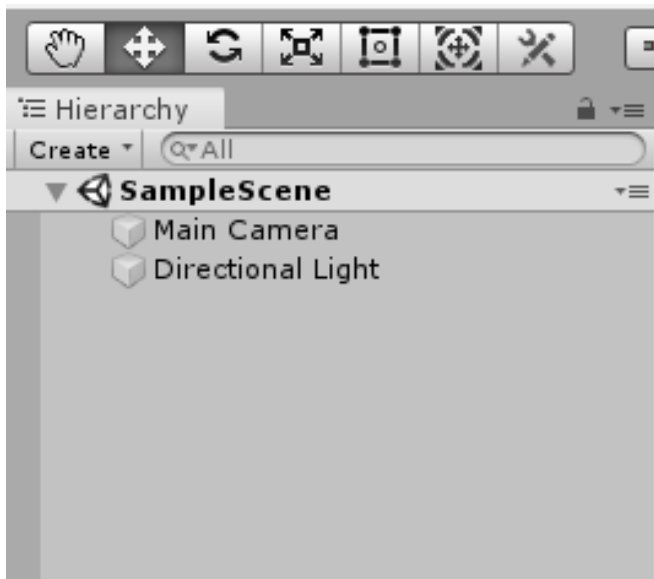
Το συγκεκριμένο Script ζητάει από το χρήστη να βάλει μια τιμή σε έναν από τους τρεις άξονες και με το πάτημα δύο πλήκτρων της αρεσκείας του να μετακινεί τον παίχτη επάνω στους συγκεκριμένους άξονες. Στον άξονα Z ο χαρακτήρας κινείται μπροστά και πίσω ενώ στον άξονα X ο χαρακτήρας κινείται δεξιά και αριστερά.



Εικόνα 6: Script κίνησης.

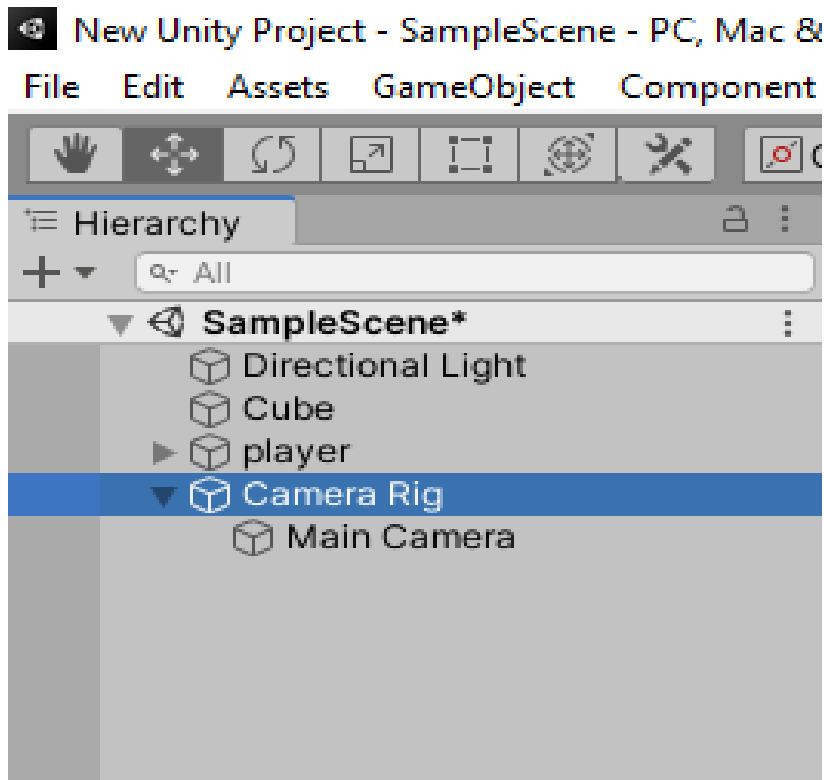
4.5 ΔΗΜΙΟΥΡΓΙΑ ΚΑΜΕΡΑΣ

Το πιο βασικό εργαλείο για τη δημιουργία του παιχνιδιού είναι η κάμερα διότι χωρίς αυτή το παιχνίδι δεν είναι λειτουργικό, γι' αυτό και το Unity με το που ανοίξει ο χρήστης το νέο του Project του έχει δημιουργήσει από μόνο του την MainCamera.



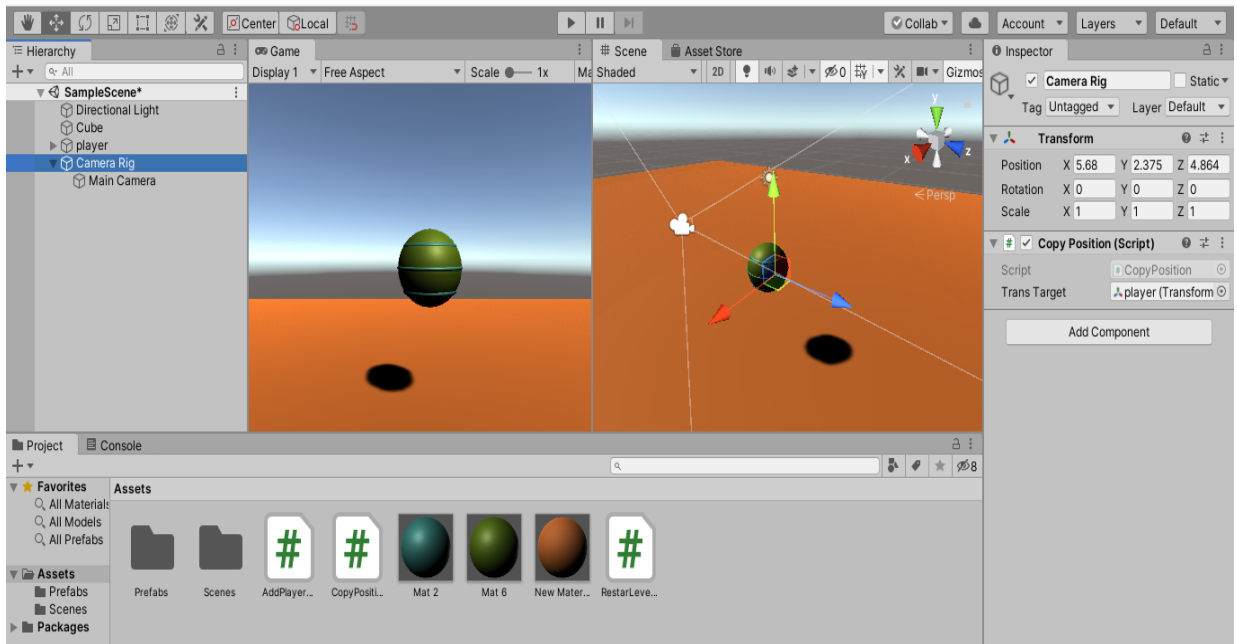
Εικόνα 1: MainCamera.

Σε κάθε παιχνίδι οι ρυθμίσεις της κάμερας διαφέρουν ανάλογα με το είδος παιχνιδιού που έχει σκοπό να δημιουργήσει ο κάθε χρήστης. Η διαμόρφωση της κάμερας γίνεται με διάφορους τρόπους, είτε με κάποιο Scriptόπου η κάμερα θα ακολουθεί τον παίχτη, είτε με το να εφαρμοστεί η κάμερα πάνω στον παίχτη με την βοήθεια του Parenting που αναφέρθηκε πιο πάνω, είτε να είναι κολλημένη απλά σε ένα σημείο της σκηνής και με διάφορους ακόμη τρόπους. Η κάμερα στο συγκεκριμένο παιχνίδι δημιουργήθηκε με την μορφή του Parenting αλλά με τη μόνη διαφορά ότι αντί να τοποθετηθεί μέσα στον φάκελο του Player στο Hierarchy Panel, τοποθετήθηκε σε ένα EmptyObject που ονομάστηκε CameraRig. Η δημιουργία του EmptyObject είναι πολύ απλή, όπως και στα υπόλοιπα αντικείμενα του Unity έτσι και σε αυτό ο χρήστης με ένα δεξί κλικ στο Hierarchy Panel θα δει την επιλογή CreateEmpty και επιλέγοντας το θα δημιουργηθεί. Με τη μέθοδο Drag And Drop μπορεί να βάλει την MainCamera μέσα στο φάκελο του CameraRig και έτσι να δημιουργήσει δεσμό Parenting.



Εικόνα 2: Δημιουργία CameraRig και δεσμός Parenting με την MainCamera.

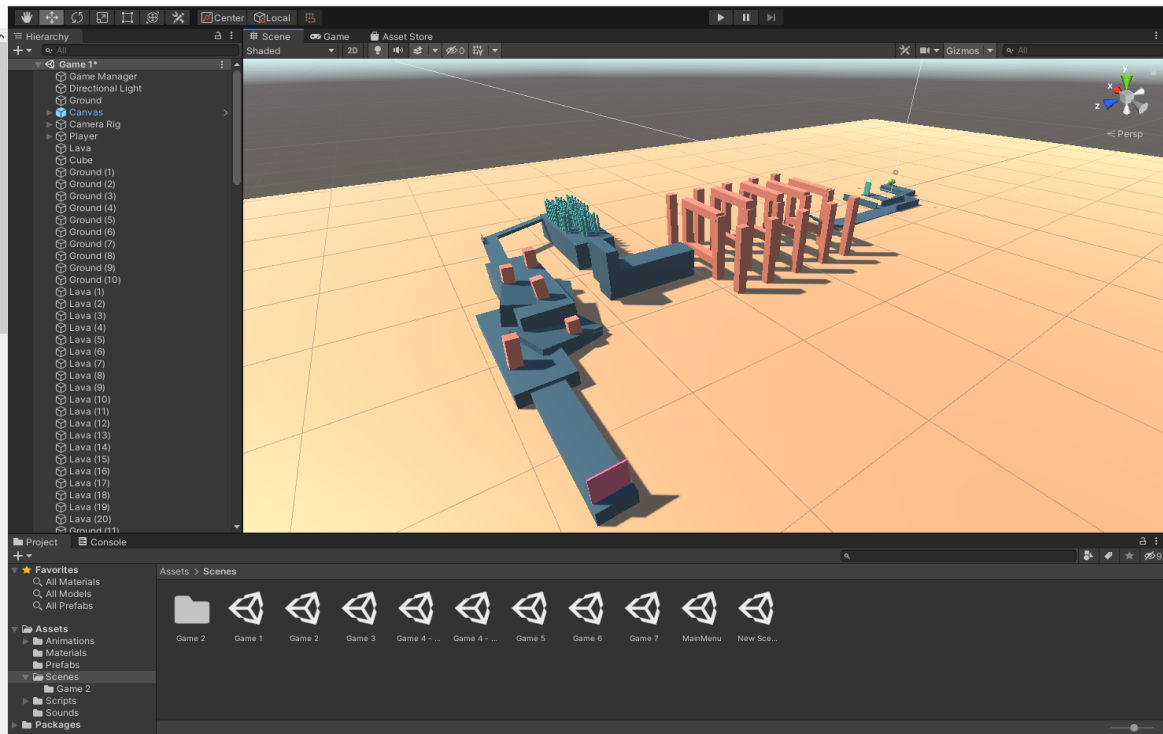
Το EmptyObject ουσιαστικά είναι ένα άρατο αντικείμενο το οποίο δεν έχει διαστάσεις ούτε είναι εμφανές από τον χρήστη του παιχνιδιού σε όποιο σημείο της σκηνής και να τοποθετηθεί. Η μόνη τροποποίηση που μπορεί να κάνει ο χρήστης είναι να του προσθέσει κάποιο Component όπως για παράδειγμα κάποιο Script ή μια πηγή ήχου. Στο συγκεκριμένο EmptyObject προστέθηκε το Script της κάμερας. Ονομάστηκε CopyPosition και το μόνο που κάνει το Script αυτό είναι να ακολουθεί ένα συγκεκριμένο αντικείμενο που θα του πει ο χρήστης. Στην προκειμένη περίπτωση το αντικείμενο που έχει οριστεί να ακολουθεί είναι ο Player. Αν η κάμερα έμπαινε στην σκηνή χωρίς το EmptyObject όταν ο χρήστης πάταγε το Play ο παίχτης θα προχώραγε ενώ η κάμερα θα έμενε σταθερή σε μια μεριά. Αν ο χρήστης έβαζε το Script κατευθείαν πάνω στην MainCamera τότε ανεξαρτήτως που θα τοποθετούσε την κάμερα στην σκηνή, αυτή θα ακολουθούσε τον παίχτη χωρίς να φαίνεται ο παίχτης διότι το Script αντιγράφει την θέση του κάθε φορά που κινείται γι' αυτό ονομάστηκε και CopyPosition. Οπότε η λύση ήταν να τοποθετηθεί το Script στο EmptyObject. Ο χρήστης κάνοντας Parenting την MainCamera στο EmptyObject μπορεί να τοποθετήσει όπου θέλει αυτός την κάμερα χωρίς πλέον να έχει προβλήματα.



Εικόνα 3: Τελικό αποτέλεσμα MainCamera.

4.6 ΔΗΜΙΟΥΡΓΙΑ ΣΚΗΝΗΣ – LEVEL

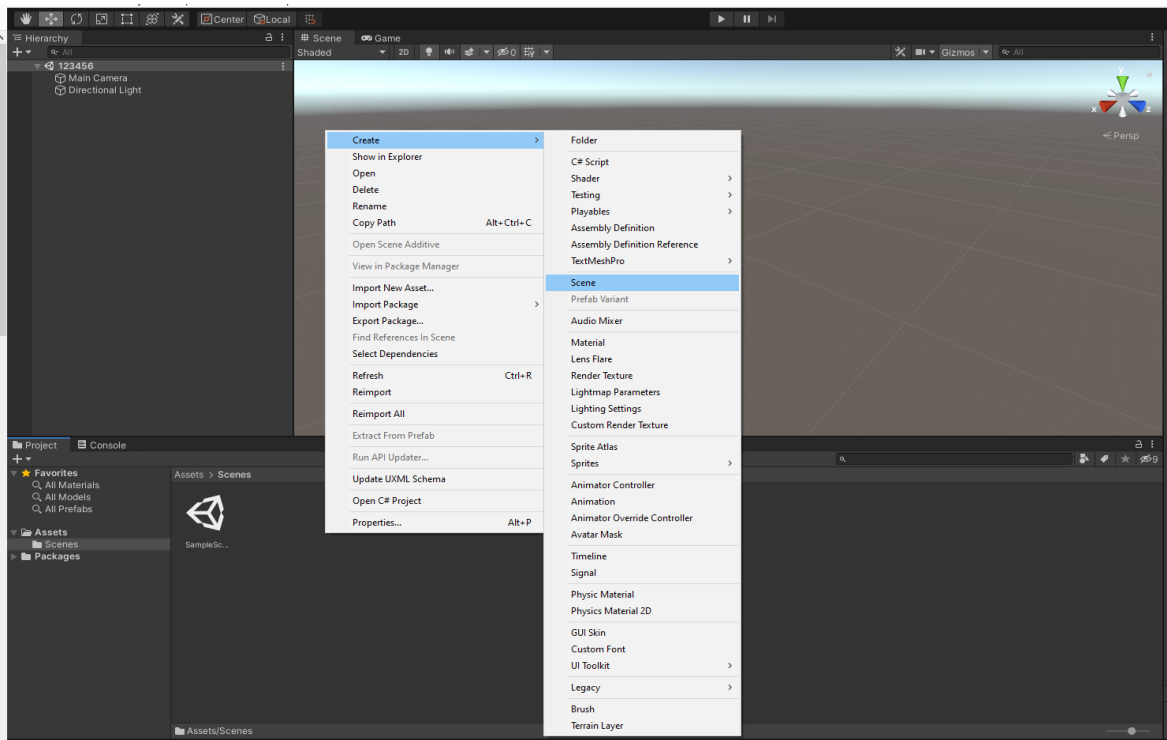
Εφόσον ο χρήστης έχει τελειώσει με την δημιουργία του βασικού χαρακτήρα (Player), την δημιουργία του εδάφους, της κάμερας και όλα όσα έχουν να κάνουν με τις βασικές λειτουργίες του παιχνιδιού, ήρθε η ώρα να αναπτύξει και την σκηνή (Level) του. Η δημιουργία της κάθε σκηνής (Level) είναι αρκετά χρονοβόρα διότι ο χρήστης θα πρέπει να σχεδιάσει το πως θέλει να είναι η σκηνή του και μετά να φτιάξει τα κατάλληλα Objects και Scripts για να έχει μια ομαλή λειτουργία. Ο πιο γρήγορος τρόπος για να υλοποιηθεί μια σκηνή είναι η δημιουργία βασικών Object. Αφού ο χρήστης φτιάξει τα Object που έχει σκεφτεί μπορεί να τα κάνει αντιγραφή - επικόλληση στο HierarchyPanel και να τροποποιεί ή να τα τοποθετεί οπουδήποτε έχει σκεφτεί μέσα στην σκηνή του. Για την ομαλή λειτουργία του παιχνιδιού κάποια αντικείμενα θα πρέπει να συνοδεύονται και από διάφορα Script όπως για παράδειγμα το Script εδάφους που χρησιμοποιήθηκε και παραπάνω.



Εικόνα 1 : Ολοκληρωμένη Σκηνή – Level.

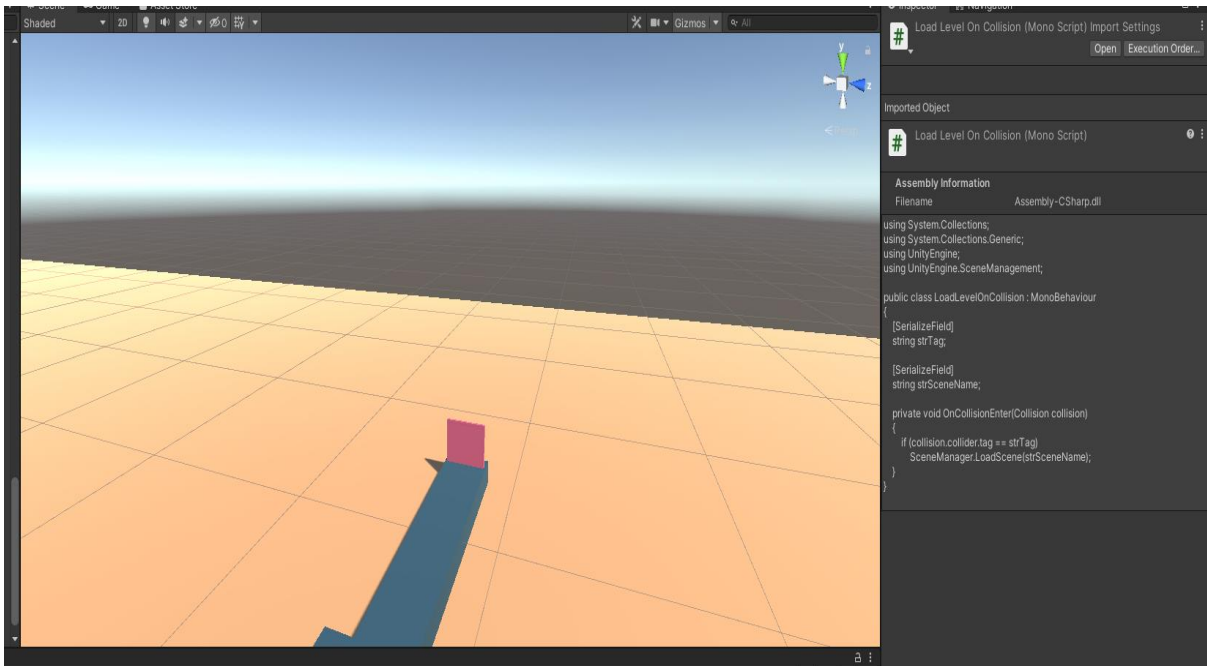
4.7 ΔΗΜΙΟΥΡΓΙΑ ΕΠΟΜΕΝΗΣ ΣΚΗΝΗΣ (LEVEL)

Η δημιουργία ενός νέου level στο Unity 3D είναι κάτι πολύ εύκολο γιατί στην ουσία ο χρήστης θα πρέπει να ακολουθήσει από την αρχή όλα αυτά που αναφέρθηκαν παραπάνω για την δημιουργία μιας νέας σκηνής (Level). Στο Project Panel ο χρήστης κάνοντας δεξί κλικ θα δει ένα μενού να εμφανίζεται στην οθόνη του. Πατώντας Create το μενού επεκτείνεται με επιπλέον επιλογές και εκεί θα βρει την επιλογή Scene. Επιλέγοντας την επιλογή αυτή ανοίγει στο Project του μια νέα σκηνή όπου εκεί μπορεί να δημιουργήσει το επόμενο Level.



Εικόνα 1: Δημιουργία νέας σκηνής.

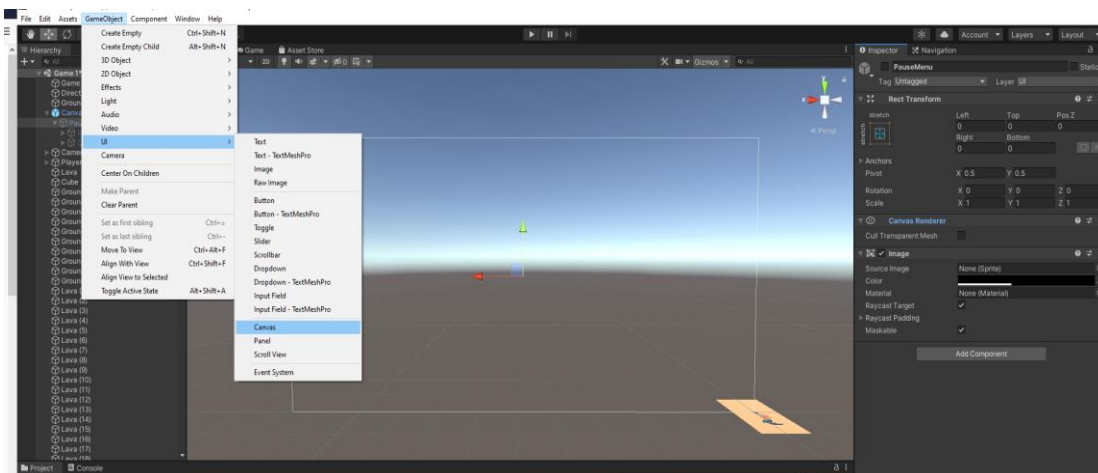
Για να δημιουργηθεί το επόμενο Level ο χρήστης θα πρέπει να σχεδιάσει από την αρχή την σκηνή του. Φτιάχνοντας το έδαφος, το χαρακτήρα του παιχνιδιού και όλα όσα αναφέρθηκαν πιο πάνω, με τη μόνη διαφορά ότι τα Scripts, τα Materials και τα Objects που έχουν ήδη δημιουργηθεί από πριν υπάρχουν στη βάση δεδομένων του Project οπότε δεν χρειάζεται να χρονοτριβεί τόσο. Το μόνο που έχει να κάνει είναι να τοποθετήσει τα κατάλληλα Objects στην σκηνή του και να της δώσει την μορφή που έχει φανταστεί. Για την εναλλαγή Level θα πρέπει να γραφτεί ένα νέο Script και να τοποθετηθεί σε ένα αντικείμενο που θα ορίσει ο χρήστης. Αυτό θα ουσιαστικά θα σημαίνει το τέλος του συγκεκριμένου Level και αυτόματα θα φορτώνει το επόμενο. Η λογική του Script είναι πολύ απλή, στην ουσία είναι σχεδόν ίδιο με το Script εδάφους με την μόνη διαφορά ότι τώρα αντί να φορτώνει την ίδια σκηνή, αυτό φορτώνει την επόμενη σκηνή που έχει ορίσει ο χρήστης.



Εικόνα 2: Τελικό Object και Script εναλλαγής σκηνής.

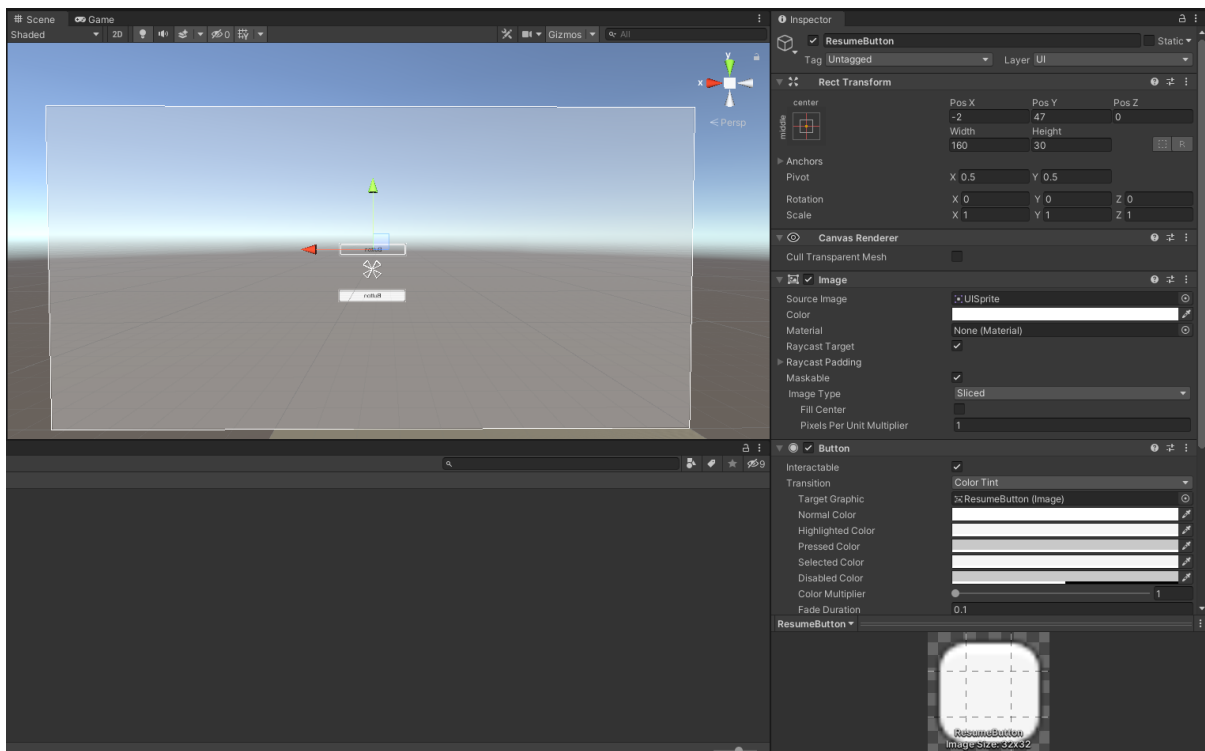
4.8 ΔΗΜΙΟΥΡΓΙΑ PAUSE MENU ΚΑΙ MAIN MENU

Ένα δύσκολο μέρος στη δημιουργία ενός παιχνιδιού είναι το Κύριο Μενού (MainMenu) του και το Μενού Παύσης (PauseMenu). Ξεκινώντας ο χρήστης θα πρέπει να προσθέσει στο project του έναν Canvas, τον οποίο μπορεί εύκολα να τον βρει στο GameObject - UI - Canvas. Αμέσως μετά τη δημιουργία του θα εμφανιστεί στην οθόνη του ο καμβάς που στην ουσία δεν είναι τίποτα άλλο από μια διάφανη επιφάνεια στην οποία ο χρήστης με κατάλληλα Scripts μπορεί να την καλέσει οποιαδήποτε στιγμή χρειαστεί στο παιχνίδι με το πάτημα ενός πλήκτρου που θα έχει ορίσει ο ίδιος.



Εικόνα 1: Αρχική μορφή Canvas.

Στη συνέχεια θα χρειαστεί ένα Panel. Με τον ίδιο τρόπο όπως δημιουργήθηκε ο καμβάς, έτσι θα φτιαχτεί και το Panel (GameObject - UI - Panel). Με τη μέθοδο Parenting ο χρήστης θα βάλει το Panel που δημιούργησε μέσα στον καμβά του. Για ευκολία του χρήστη και για πιο ωραίο οπτικό αποτέλεσμα μπορεί να ονομάσει τον καμβά «MainMenu» και το πάνελ «PauseMenu». Ουσιαστικά το πάνελ είναι μια μικρή συντόμευση που την χρειάζεται το Unity για να μπορέσει να ξεκινήσει την όποια διαδικασία του έχει αναθέσει ο χρήστης. Στην προκειμένη περίπτωση είναι μια συντόμευση για να μπορέσει το Unity να ξεκινήσει το χτίσιμο του PauseMenu. Στη συνέχεια θα δημιουργηθούν δύο Button (κουμπιά) τα οποία θα ονομαστούν ResumeButton και QuitButton. Στο Inspector Panel ο δημιουργός του παιχνιδιού έχει τη δυνατότητα να αλλάξει χρώμα, διαστάσεις και διάφορες άλλες παραμέτρους του Button που δημιούργησε.



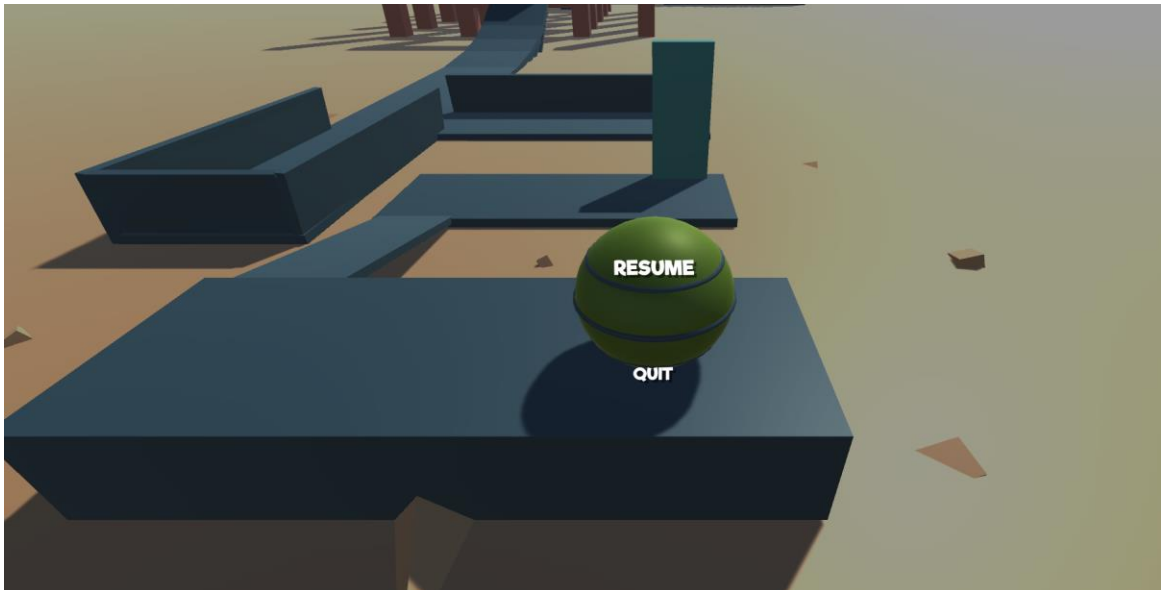
Εικόνα 2: Δημιουργία Resume και QuitButton.

Για να συνεργαστεί ο καμβάς με τα δυο κουμπιά που δημιούργησε ο χρήστης θα χρειαστεί ένα Script, για λόγους ευκολίας μπορεί να ονομαστεί PauseMenuScript. Στην ουσία το Script διαβάζει το γράμμα P και όταν ο χρήστης το πατήσει ανοίγει την καρτέλα του μενού. Το Script θα περαστεί στον καμβά μόνο.


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class PauseMenu : MonoBehaviour
7 {
8     public static bool GameIsPaused = false;
9
10    public GameObject pauseMenuUI;
11
12    [SerializeField]
13    string strSceneName;
14
15    void Update()
16    {
17        if(Input.GetKeyDown(KeyCode.P))
18        {
19            {
20                if (GameIsPaused)
21                {
22                    Resume();
23                }
24                else
25                {
26                    Pause();
27                }
28            }
29        }
30
31        public void Resume()
32        {
33            pauseMenuUI.SetActive(false);
34            Time.timeScale = 1f;
35            GameIsPaused = false;
36        }
37
38        void Pause()
39        {
40            pauseMenuUI.SetActive(true);
41            Time.timeScale = 0f;
42            GameIsPaused = true;
43        }
44
45
46        public void QuitGame()
47        {
48            SceneManager.LoadScene(strSceneName);
49        }
50    }
```

Εικόνα 3: Script καμβά.

Στα δυο κουμπιά μέσω του Inspector Panel, στην επιλογή OnClick που βρίσκει ο χρήστης, θα ρυθμίσει ανάλογα το κάθε κουμπί το τι θέλει να κάνει πατώντας το. Με το Resume συνεχίζει το παιχνίδι από εκεί που σταμάτησε ενώ με το Quit κλείνει το παιχνίδι και φορτώνει την σκηνή με το αρχικό μενού του παιχνιδιού.



Εικόνα 3: Τελικό αποτέλεσμα PauseMenu.

Με τη δημιουργία ενός GameManager στο Hierarchy Panel ο χρήστης μπορεί να προσθέτει στην σκηνή του Script για να κάνουν διάφορες λειτουργίες. Στην συγκεκριμένη περίπτωση ο παίχτης πατώντας το R μπορεί να κάνει Restart το επίπεδο (level) στο οποίο είναι και να το αρχίσει από την αρχή. Αυτό γίνεται με ένα πολύ απλό Script όπου διαβάζει το γράμμα R και μόλις το πατήσει ο παίκτης φορτώνει από την αρχή την σκηνή.

```
# Restart Level With Button (Mono Script)
Assembly Information
Filename Assembly-CSharp.dll
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

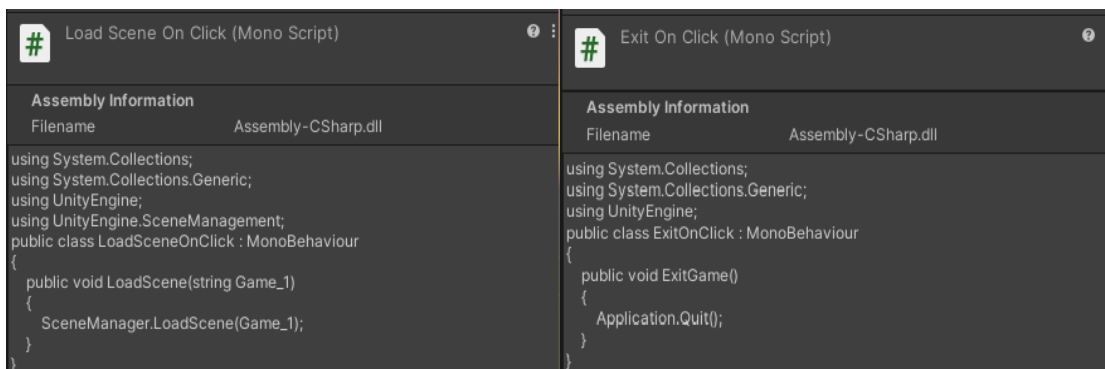
public class RestartLevelWithButton : MonoBehaviour
{
    [SerializeField]
    KeyCode keyRestart;

    void Update ()
    {
        if (Input.GetKey(keyRestart))
            SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }
}
```

Εικόνα 5: Restartscript.

Η δημιουργία του MainMenu είναι λίγο πιο σύνθετη διαδικασία. Θα πρέπει να φτιαχτεί μια νέα σκηνή και πάνω σε αυτή να σχεδιαστεί το Κύριο Μενού του παιχνιδιού. Ο χρήστης

εφόσον έχει δημιουργήσει και σχεδιάσει την νέα σκηνή του μπορεί πλέον να ασχοληθεί με τη διαμόρφωση του Μενού. Η διαδικασία είναι ίδια σχεδόν όπως και παραπάνω με τη μόνη διαφορά ότι πλέον ο καμβάς που δημιουργήθηκε θα φαίνεται στην σκηνή από την αρχή και δεν θα εμφανίζεται με το πάτημα κάποιου πλήκτρου. Ο χρήστης θα φτιάξει τρία κουμπιά, Play, Multiplayer και Exit. Πατώντας το Play ανοίγει η σκηνή με το πρώτο Level του παιχνιδιού. Με το κουμπί Multiplayer ανοίγει η σκηνή με το Level που μπορεί να υποστηρίξει δυο παίκτες ταυτόχρονα και τέλος με το Exit απλά κλείνει το παιχνίδι. Για να το πετύχει αυτό ο χρήστης χρειάζεται δυο Script, τα LoadSceneOnClick και ExitOnClick. Στα δυο πρώτα κουμπιά έβαλε το LoadSceneOnClickScript και στο κουμπί Exit έβαλε το ExitOnClickScript.



Εικόνα 6: Load Scene και Exit script.

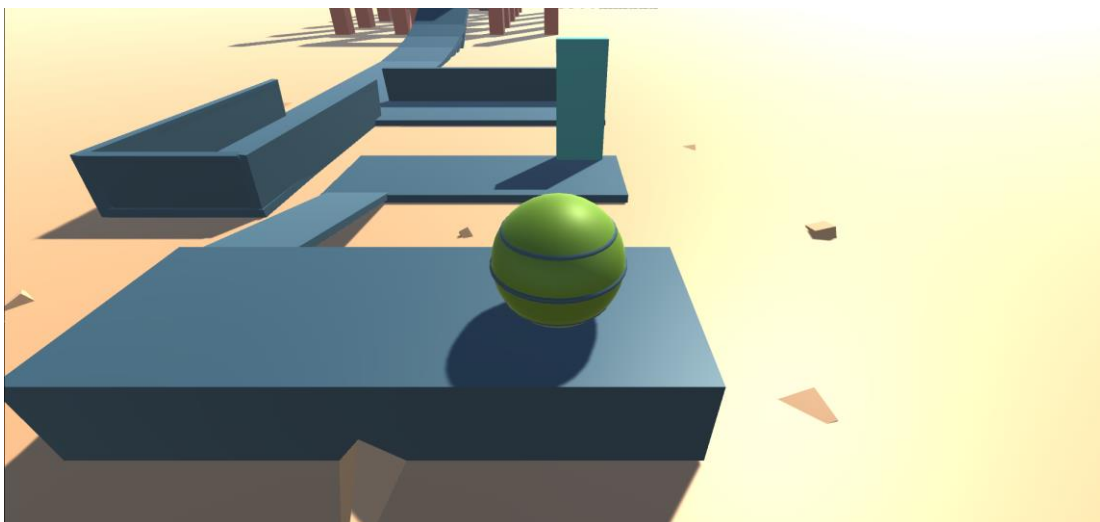
Παράλληλα όμως παραμετροποίηση τα κουμπιά έτσι ώστε πατώντας τα, να ανοίγει η συγκεκριμένη σκηνή που αυτός επιθυμεί. Επίσης ο χρήστης μπορεί να προσθέσει μουσική εάν επιθυμεί στην σκηνή του. Αυτό επιτυγχάνεται με την προσθήκη ενός AudioSource στο Hierarchy Panel. Σε αυτή την πηγή ήχου μπορεί να προστεθεί ένας ήχος και να διαμορφωθεί κατάλληλα και ανάλογα τις ανάγκες του εκάστοτε χρήστη.



Εικόνα7: Τελική μορφή MainMenu.

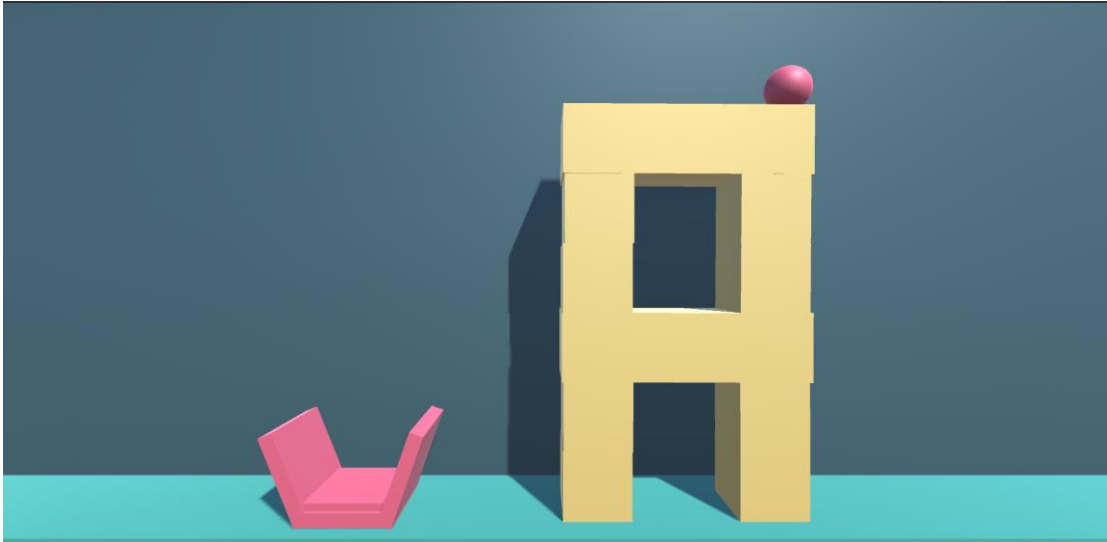
ΚΕΦΑΛΑΙΟ 5 : ROLLINGBALLS 3D

Το παιχνίδι αυτό έχει σαν βασικό χαρακτήρα μια μπάλα όπου έχει σαν προϋπόθεση, ο παίχτης να μετακινεί την μπάλα ανάμεσα στα εμπόδια και να φτάνει έτσι στο τέλος κάθε πίστας και να προωθείται στην επόμενη. Αποτελείται από πέντε πίστες Singleplayer και μια πίστα Multiplayer που μπορούν να παίξουν δυο παίχτες ταυτόχρονα. Ξεκινώντας από την πρώτη πίστα ο παίχτης θα πρέπει να ξεπεράσει όλα τα άλματα και τα εμπόδια που βρίσκονται στο δρόμο του έτσι ώστε να φτάσει στο επόμενο επίπεδο. Ο παίχτης έχει τη δυνατότητα να μετακινήσει την μπάλα με τα βελάκια του πληκτρολογίου του. Εάν η μπάλα ακουμπήσει το κίτρινο έδαφος τότε η πίστα αρχίζει από την αρχή. Το ίδιο συμβαίνει όταν η μπάλα ακουμπήσει κάποια κόκκινα εμπόδια που βρίσκονται σε προχωρημένο σημείο της πίστας.



Εικόνα 1 : Εικόνα έναρξης Level 1.

Το Level 2 και 3 είναι τελείως διαφορετικό από το Level 1. Ο παίχτης θα πρέπει με να καταστρέψει τους κύβους έτσι ώστε να καταφέρει να τοποθετήσει την μπάλα σε ένα ειδικά διαμορφωμένο καλάθι. Με το που ακουμπήσει η μπάλα στο καλάθι αυτομάτως περνάει στην επόμενη πίστα που είναι σχεδόν όμοια με αυτή απλά έχει ένα μεγαλύτερο βαθμό δυσκολίας. Οι κύβου καταστρέφονται κάνοντας αριστερό κλικ στο ποντίκι του ο χρήστης πάνω στον κύβο που διάλεξε να καταστρέψει.



Εικόνα 2: Εικόνα έναρξης Level 2.

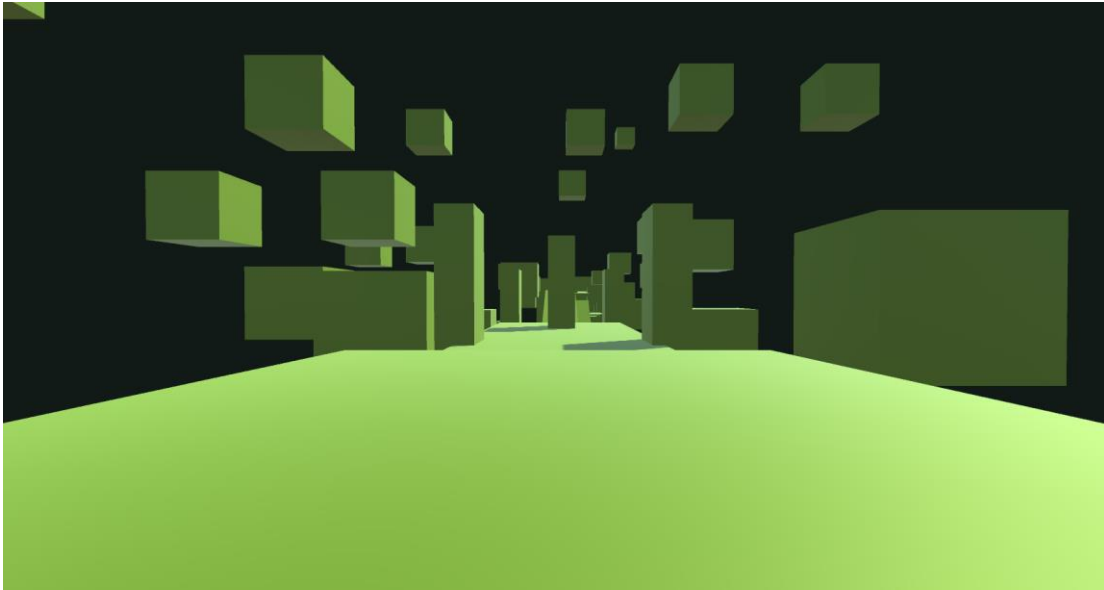
Το Level 3 όπως αναφέρθηκε είναι σχεδόν πανομοιότυπο με το προηγούμενο Level αλλά έχει μια δόση δυσκολίας παραπάνω για τον χρήστη. Και στα δυο αυτά Level ο χρήστης με το πλήκτρο R μπορεί να κάνει Reload το Level και να προσπαθήσει να το περάσει πάλι από την αρχή.



Εικόνα 3: Εικόνα έναρξης Level 3.

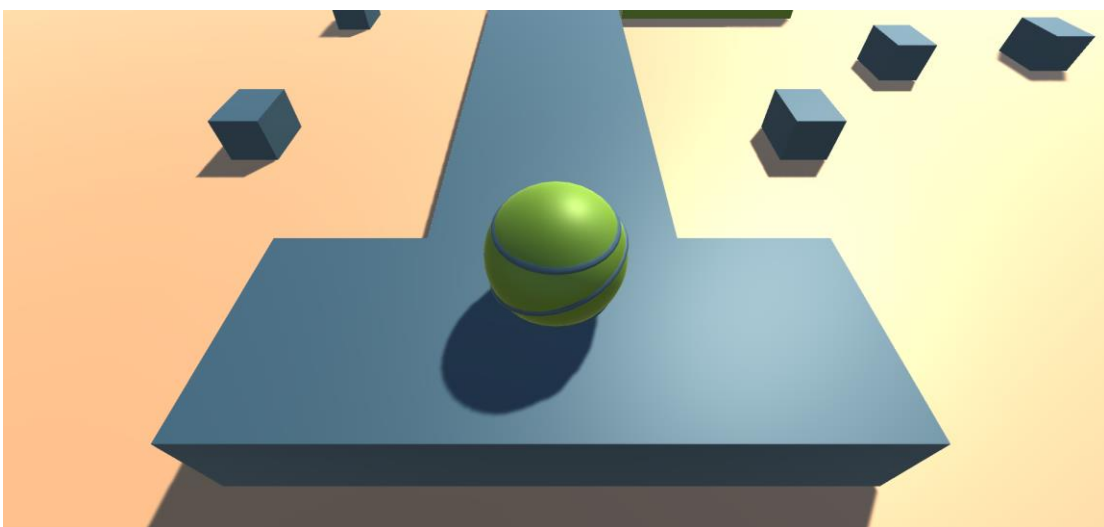
Το Level 4 ξεκινάει με την κάμερα να είναι τοποθετημένη πάνω στην μπάλα και την μπάλα να έχει αρχική ταχύτητα. Στόχος του κάθε παίχτη είναι να μετακινεί την μπάλα ανάμεσα στα εμπόδια και τα άλματα που θα βρει στο δρόμο του και να επιταχύνει όποτε χρειαστεί έτσι ώστε να φτάσει στον τερματισμό. Η αρχική ταχύτητα της μπάλας και τα

πολλαπλά άλματα και εμπόδια κάνουν αυτό το Level αρκετά δύσκολο για τον χρήστη, που θα πρέπει να βρίσκεται συνέχεια σε ετοιμότητα για να καταφέρει να τερματίσει.



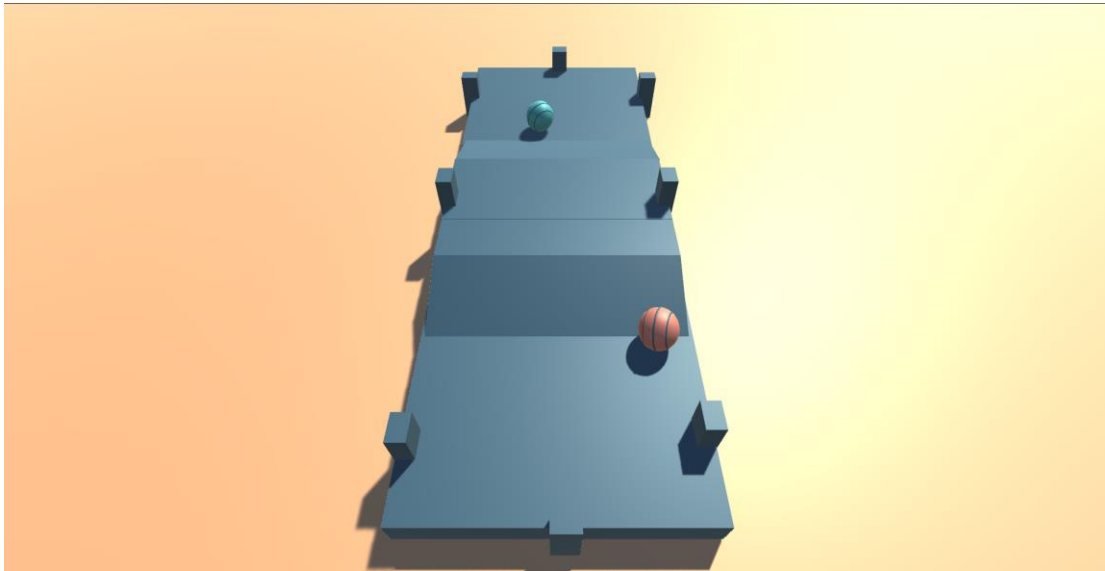
Εικόνα 4: Εικόνα έναρξης Level 4.

Το πέμπτο και τελευταίο Level του παιχνιδιού είναι εξαιρετικά δύσκολο καθώς απαιτεί από τον χρήστη άψογη κίνηση και τοποθέτηση της μπάλας ανάμεσα στα πολλαπλά εμπόδια που θα συναντήσει. Υπάρχουν κινούμενα εμπόδια και γέφυρες τις οποίες πρέπει ο χρήστης να περάσει για να καταφέρει να τερματίσει. Έχουν χρησιμοποιηθεί αρκετά Animation και στην μπάλα (παίχτης) αλλά και στις γέφυρες ή στα εμπόδια που υπάρχουν στην σκηνή αυτή.



Εικόνα 5: Εικόνα έναρξης Level 5.

Επίσης όπως αναφέρθηκε υπάρχει και ένα Multiplayer Level. Σε αυτό το Level μπορούν να παίξουν δύο χρήστες ταυτόχρονα. Αποτελείται από μια σκηνή με δύο χαρακτήρες (μπάλες) και ο κάθε χρήστης κατευθύνει από έναν. Στόχος της πίστας αυτής είναι ένας από τους δύο χρήστες να πέσουν από την πίστα και να καταλήξουν στην λάβα (έδαφος). Εκεί το παιχνίδι τελειώνει και φορτώνει η σκηνή με το αρχικό μενού.



Εικόνα 6: Multiplayer level.

Στον παρακάτω πίνακα ο χρήστης μπορεί να δει όλα τα χρήσιμα κουμπιά τα οποία πρέπει να ξέρει για να μπορέσει να παίξει το συγκεκριμένο παιχνίδι. Το παιχνίδι γενικά είναι πολύ απλό οπότε και τα πλήκτρα που χρησιμοποιήθηκαν είναι πολύ λίγα για να το κάνουν ακόμη πιο εύχρηστο στον κάθε χρήστη.

↑ ή W	Κίνηση Εμπρός
↓ ή S	Κίνηση Πίσω
→ ή D	Κίνηση Δεξιά
← ή A	Κίνηση Αριστερά
Αριστερό Κλικ	Καταστροφή Κουτιού
R	Restart Level
P	Pause Menu
Esc	Έξοδος

Πίνακας 1 : Πλήκτρα παιχνιδιού.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Λαμβάνοντας υπόψη την όλη διαδικασία που πραγματοποιήθηκε για τη δημιουργία του παιχνιδιού της παρούσας πτυχιακής εργασίας πρέπει να σημειωθεί πως ήταν εξαιρετικά επικοινωνιακή, λαμβάνοντας υπόψη το πλήθος των γνώσεων που λήφθηκαν. Εξαιρετικής σημασίας ήταν το γεγονός ότι υπάρχει μια μεγάλη κοινότητα χρηστών των προγραμμάτων που χρησιμοποιήθηκαν (blender, unity) οι οποίοι σε μεγάλο αριθμό περιπτώσεων έδιναν μέσω των σχολίων τους λύσεις σε πολλά προβλήματα. Επιπλέον όσον αφορά τα δύο προγράμματα που χρησιμοποιήθηκαν πρέπει να αναφερθεί πως υπήρχε μεγάλος αριθμός Βίντεο τα οποία χρησιμοποιήθηκαν και αυτά σε αρκετές περιπτώσεις για την διεκπεραίωση του παιχνιδιού.

Οι υπάρχουσες γνώσεις πάνω στον προγραμματισμό συνδυάστηκαν με πλήθος νέων γνώσεων οι οποίες επέφεραν νέες ιδέες και δεξιότητες. Σημαντικό είναι πως κατά την εκπόνηση της εργασίας μέσω της βιβλιογραφικής ανασκόπησης λήφθηκαν επιπρόσθετες γνώσεις και ερεθίσματα για πιθανές μελλοντικές καινοτομίες.

ΠΑΡΑΡΤΗΜΑ 1

Σε αυτό το παράρτημα θα παρουσιαστούν όλα τα Script τα οποία χρησιμοποιήθηκαν για την υλοποίηση του παιχνιδιού. Όλα τα Script δημιουργήθηκαν στο VisualStudio σε γλώσσα C#.

```
AccelerateTowards.cs - X
Miscellaneous Files - AccelerateTowards
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class AccelerateTowards : MonoBehaviour
6 {
7     [SerializeField]
8     Transform transTowards;
9
10    [SerializeField]
11    float fSpeed;
12
13    Rigidbody rigid;
14
15    // Start is called before the first frame update
16    void Start()
17    {
18        rigid = GetComponent<Rigidbody>();
19        if (transTowards == null)
20        {
21            transTowards = FindObjectOfType<AddPlayerControlledVelocity>().transform;
22        }
23    }
24
25    // Update is called once per frame
26    void Update()
27    {
28        Vector3 v3MeTowardsTarget = transTowards.position - transform.position;
29        rigid.velocity += v3MeTowardsTarget.normalized * fSpeed * Time.deltaTime;
30    }
31
32 }
```

```
AddConstantVelocity.cs - X AccelerateTowards.cs
Miscellaneous Files
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class AddConstantVelocity : MonoBehaviour
6 {
7     [SerializeField]
8     Vector3 v3Force;
9
10    void FixedUpdate ()
11    {
12        GetComponent<Rigidbody>().velocity += v3Force;
13    }
14
15 }
```

```
AddPlayerControlledVelocity.cs  AddConstantVelocity.cs  AccelerateTowards.cs
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class AddPlayerControlledVelocity : MonoBehaviour
6  {
7      [SerializeField]
8      Vector3 v3Force;
9
10     [SerializeField]
11     KeyCode keyPositive;
12     [SerializeField]
13     KeyCode keyNegative;
14
15     void FixedUpdate ()
16     {
17         if (Input.GetKey(keyPositive))
18             GetComponent<Rigidbody>().velocity += v3Force;
19
20         if (Input.GetKey(keyNegative))
21             GetComponent<Rigidbody>().velocity -= v3Force;
22     }
23
24
```

```
CopyPosition.cs  AddPlayerControlledVelocity.cs  AddConstantVelocity.cs  AccelerateToward
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class CopyPosition : MonoBehaviour
6  {
7      [SerializeField]
8      Transform transTarget;
9
10     void Update ()
11     {
12         transform.position = transTarget.position;
13     }
14
15
```

```
DestroyAfter.cs  CopyPosition.cs  AddPlayerControlledVelocity.cs  AddConstantVelocity.cs
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DestroyAfter : MonoBehaviour
6  {
7      [SerializeField]
8      float fTimer;
9
10     // Start is called before the first frame update
11     void Start()
12     {
13     }
14
15     // Update is called once per frame
16     void Update()
17     {
18         fTimer -= Time.deltaTime;
19         if (fTimer <= 0)
20             Destroy(this.gameObject);
21     }
22 }
23
24
```

```
DestroyOnClick.cs  DestroyAfter.cs  CopyPosition.cs  AddPlayerControlledVelocity.cs
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DestroyOnClick : MonoBehaviour
6  {
7     private void OnMouseDown()
8     {
9         Destroy(gameObject);
10    }
11 }
12
```

```
DestroyOnCollision.cs  DestroyOnClick.cs  DestroyAfter.cs  CopyPosition.cs
C# Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class DestroyOnCollision : MonoBehaviour
7  {
8      [SerializeField]
9      string strTag;
10
11     [SerializeField]
12     bool bDestroySelf = false;
13
14     [SerializeField]
15     bool bDestroyOther = false;
16
17     private void OnCollisionEnter(Collision collision)
18     {
19         if (collision.collider.tag == strTag)
20         {
21             if (bDestroySelf)
22                 Destroy(this.gameObject);
23             if (bDestroyOther)
24                 Destroy(collision.gameObject);
25         }
26     }
27 }
28
```

```
LoadLevelOnCollision.cs  DestroyOnCollision.cs  DestroyOnClick.cs  DestroyAfter.cs
C# Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class LoadLevelOnCollision : MonoBehaviour
7  {
8      [SerializeField]
9      string strTag;
10
11     [SerializeField]
12     string strSceneName;
13
14     private void OnCollisionEnter(Collision collision)
15     {
16         if (collision.collider.tag == strTag)
17             SceneManager.LoadScene(strSceneName);
18     }
19 }
20
```

```
RestartLevelOnCollision.cs  X LoadLevelOnCollision.cs DestroyOnCollision.cs DestroyOnClick
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class RestartLevelOnCollision : MonoBehaviour
7  {
8      [SerializeField]
9      string strTag;
10
11     private void OnCollisionEnter(Collision collision)
12     {
13         if (collision.collider.tag == strTag)
14             SceneManager.LoadScene(SceneManager.GetActiveScene().name);
15     }
16 }
17
```

```
RestartLevelWithButton.cs  X RestartLevelOnCollision.cs LoadLevelOnCollision.cs DestroyOnC
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class RestartLevelWithButton : MonoBehaviour
7  {
8      [SerializeField]
9      KeyCode keyRestart;
10
11     void Update ()
12     {
13         if (Input.GetKey(keyRestart))
14             SceneManager.LoadScene(SceneManager.GetActiveScene().name);
15     }
16 }
17
```

```
EscapeMenu.cs  SpawnOnCollision.cs  Spawn.cs  RestartLevelWithButton.cs
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class EscapeMenu : MonoBehaviour
7  {
8
9
10     [SerializeField]
11     string strSceneName;
12
13     private void Update()
14     {
15         if (Input.GetKey("escape"))
16             SceneManager.LoadScene(strSceneName);
17     }
18 }
19
```

```
ExitOnClick.cs  EscapeMenu.cs  SpawnOnCollision.cs
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  public class ExitOnClick : MonoBehaviour
5  {
6     public void ExitGame()
7     {
8         Application.Quit();
9     }
10 }
```

```
LoadSceneOnClick.cs × ExitOnClick.cs EscapeMenu.cs
Miscellaneous Files
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5 public class LoadSceneOnClick : MonoBehaviour
6 {
7     public void LoadScene(string Game_1)
8     {
9         SceneManager.LoadScene(Game_1);
10    }
11 }
```

```
MultiplayerScene.cs × LoadSceneOnClick.cs ExitOnClick.cs
Miscellaneous Files
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5 public class MultiplayerScene : MonoBehaviour
6 {
7     public void LoadScene(string Game_2)
8     {
9         SceneManager.LoadScene(Game_2);
10    }
11 }
```



```
PauseMenu.cs  X MultiplayerScene.cs  LoadSceneOnClick.cs  ExitOnClick.cs  EscapeM
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class PauseMenu : MonoBehaviour
7  {
8      public static bool GameIsPaused = false;
9
10     public GameObject pauseMenuUI;
11
12     [SerializeField]
13     string strSceneName;
14
15     void Update()
16     {
17         if(Input.GetKeyDown(KeyCode.P))
18         {
19             {
20                 if (GameIsPaused)
21                 {
22                     Resume();
23                 }
24                 else
25                 {
26                     Pause();
27                 }
28             }
29         }
30
31         public void Resume()
32         {
33             pauseMenuUI.SetActive(false);
34             Time.timeScale = 1f;
35             GameIsPaused = false;
36         }
37
38         void Pause()
39         {
40             pauseMenuUI.SetActive(true);
41             Time.timeScale = 0f;
42             GameIsPaused = true;
43         }
44
45         public void QuitGame()
46         {
47             SceneManager.LoadScene(strSceneName);
48         }
49     }
50 }
51
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

Nicholas A. Christakis, James H. Fowler, (2010), Connected: The spectacular power of social networks and the way they change our life, Athens :Katoptro

Goultidis, Christos, (2014), Parents, kids and Internet :Protect your child from the dangers of internet, Athens : Klidarithmos

Thebrian C.L. (2000), Network, Internet and the new social media, USA

Καλογρίδου – Κολυβά, Μ. (2011), Οι πολλές όψεις του διαδικτύου- Χρήση, Κατάχρηση, Εθισμός. Διάλογος-Αντίλογος. Αθήνα: Δρόμων

Courth, E. (2003), Communication and internet, Athens: Ellinika Grammata

Μαραγκός, Κ. & Γρηγοριάδου, Μ. (2004), Διερεύνηση των χαρακτηριστικών των κινήτρων και της δυναμικής χρήσης των ηλεκτρονικών παιχνιδιών στη μαθησιακή διαδικασία, Στο Μ. Γρηγοριάδου, Α. Ράπτης, Σ. Βοσνιάδου& Χ. Κυνηγός (Επιμ.), *Πρακτικά 4ου Πανελληνίου Συνεδρίου με διεθνή συμμετοχή «Οι τεχνολογίες της πληροφορίας και της επικοινωνίας στην εκπαίδευση»* (103-112) (τόμος Α), ΕΤΠΕ, Αθήνα: Εκδόσεις Νέων Τεχνολογιών

Μάτσα, Κ., (2009). Εξάρτηση από το Διαδίκτυο: η πιο σύγχρονη μορφή τοξικομανίας, Τετράδια Ψυχιατρικής

Σιώμος, Κ., Σφακιανάνακης, Ε. ,Φλώρος, Γ. (2012) Εθισμός στο διαδίκτυο και άλλες διαδικτυακές συμπεριφορές υψηλού κινδύνου. Αθήνα: Α.Α. Λιβάνη.

ΧΡΗΣΙΜΟΙ ΣΥΝΔΕΣΜΟΙ

- <https://unity.com/>
- <https://www.blender.org/>
- <https://forum.unity.com/>
- <https://el.wikipedia.org/wiki/%CE%A0%CF%8D%CE%BB%CE%B7:%CE%9A%CF%8D%CF%81%CE%B9%CE%B1>
- <https://visualstudio.microsoft.com/>
- <https://www.youtube.com/user/Unity3D>