

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Πελοποννήσου

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Μηχανικών Πληροφορικής

ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
Ι Δ Ρ Υ Μ Α



ΠΕΛΟΠΟΝΝΗΣΟΥ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής: Ανάπτυξη ενός Android Game μέσω της πλατφόρμας δημιουργίας παιχνιδιών Unity και έρευνα στους τρόπους χρηματοδότησης του και αποκόμισης εσόδων.

Τίτλος Παιχνιδιού: *Soulbringer*

Επιβλέπων καθηγητής: Γκατζιώλης Κλεάνθης

Σύνταξη από : Νίκα Αναστάσιο **AM :** 2012050

Σπάρτη 2018

ΕΥΧΑΡΙΣΤΙΕΣ

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

ΠΕΡΙΛΗΨΗ

Η πτυχιακή εργασία αποσκοπεί στη σχεδίαση και ανάπτυξη ενός παιχνιδιού με λειτουργικό σύστημα Android χρησιμοποιώντας την πλατφόρμα Unity. Ο χρήστης χειρίζεται έναν χαρακτήρα και σκοπός του είναι να φτάσει στον τερματισμό αποφεύγοντας τα ηρσ. Το παιχνίδι αποτελείται από τρισδιάστατα (3D) γραφικά και τα επίπεδά του έχουν αυξανόμενο βαθμό δυσκολίας. Έπειτα θα αναλυθούν οι τρόποι που μπορεί ο developer να προσκομίσει έσοδα μέσα από τα παιχνίδια (την διαφήμιση, τις δωρεές, πώληση των εκάστοτε τίτλων παιχνιδιών.)

Επιπλέον προγράμματα που χρησιμοποιήθηκαν: Visual Studio (IDE), Blender
Επιπλέον προγράμματα που χρειάστηκαν στην πορεία: Audacity (επεξεργασία ήχου), Photoshop (επεξεργασία εικόνας. Επίσης, χρησιμοποιήθηκαν assets με ελεύθερη άδεια .

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

Όνομα και Επώνυμο Συγγραφέων (Με Κεφαλαία):

ΑΝΑΣΤΑΣΙΟΣ ΝΙΚΑΣ

Υπογραφές (Ολογράφως, χωρίς μονογραφή):

ΑΝΑΣΤΑΣΙΟΣ ΝΙΚΑΣ

Ημερομηνία (Ημέρα – Μήνας – Έτος):

18/09/2018

Περιεχόμενα

ΕΥΧΑΡΙΣΤΙΕΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
Κεφάλαιο 1ο : Εισαγωγή.....	6
1.1 Περίληψη	7
1.2 Κίνητρο Για Την Διεξαγωγή Της Εργασίας	7
1.3 Σκοπός Και Στόχοι Εργασίας.....	8
Κεφάλαιο 2ο Η ιστορία της ανάπτυξης ηλεκτρονικών παιχνιδιών	8
2.1 Τα πρώτα βήματα της βιομηχανίας παιχνιδιών	8
2.2 Σύντομη ιστορική αναδρομή και εξέλιξη των βιντεοπαιχνιδιών.....	10
2.3 Είδη ηλεκτρονικών παιχνιδιών	13
2.4 Γιατί επιλέγουμε το Unity.....	21
Κεφάλαιο 3ο Unity 3d Game Engine.....	26
3.1 Πρόλογος.....	26
3.2 Ιστορική Αναδρομή.....	27
3.3 Το εργαλείο σήμερα.....	27
3.4 Unity Editor.....	28
3.4.1 Τα παράθυρα του Editor.....	28
3.4.2 Η Μπάρα Μενού.....	32
3.4.3 Build Settings.....	36
Κεφάλαιο 4ο : Η δημιουργία της C#.....	39
4.1 Χαρακτηριστικά της C που κληρονόμησε η C#.....	39
4.2 Τι είναι το .NET Framework	40
4.3 Αντικειμενοστραφής Προγραμματισμός.....	41
4.4 Ενθυλάκωση.....	42
4.5 Πολυμορφισμός	42

4.6 Κληρονομικότητα.....	42
4.7 Γενικά περί Κλάσεων.....	43
Κεφάλαιο 5ο Παράδειγμα	44
5.1 Γενική λογική C# στην Unity.....	44
5.2 Βασικός τρόπος δήλωσης και αρχικοποίησης των μεταβλητών.....	45
5.3 Η δήλωση του θα γίνει ως εξής.....	45
5.4 Βιβλιοθήκες της Unity.....	46
5.5 Βασική επικοινωνία μεταξύ scripts.....	47
Κεφάλαιο 6ο MonoDevelop.....	48
Κεφάλαιο 7ο Ανάπτυξη Παιχνιδιού.....	51
7.1 Σύλληψη Ιδέας.....	52
7.2 Επιλογή χαρακτήρα.....	52
7.3 Επιλογή αντικειμένων.....	53
7.4 Παρουσίαση επιπέδων.....	58
7.5 Δημιουργία του Main Menu.....	60
7.6 Ο κώδικας του χαρακτήρα.....	62
7.7 Το Animation των χαρακτήρων.....	63
7.8 Δημιουργία του περιβάλλοντος του παιχνιδιού.....	64
7.9 Η μουσική του παιχνιδιού.....	67
7.10 Το τελικό αποτέλεσμα.....	68
Κεφάλαιο 8ο : Έσοδα από την δημιουργία παιχνιδιών.....	72
8.1 Οικονομία των παιχνιδιών στον κόσμο.....	72
8.2 Τρόποι Εσόδων On-Line Παιχνιδιών	75
8.3 Πληρωμή Για Πλήρη Χρήση Του Παιχνιδιού.....	78
8.4 Θετικά Και Αρνητικά Των Free - To - Play Παιχνιδιών.....	78
Ανακεφαλαίωση.....	82
Επίλογος.....	83

Κεφάλαιο 1ο : Εισαγωγή

1.1 Περίληψη

Η πτυχιακή εργασία αναφέρεται στην διαδικασία ανάπτυξης ενός ηλεκτρονικού παιχνιδιού, το οποίο για να δημιουργηθεί, βασικό ρόλο έπαιξε η μηχανή παιχνιδιών Unity με την χρήση της γλώσσας C#. Για την δημιουργία του παιχνιδιού χρειάστηκαν πέντε σκηνές. Στην πρώτη σκηνή είναι το μενού στο οποίο ο χρήστης μπορεί ή να ξεκινήσει το παιχνίδι ή να κάνει ρύθμιση στον ήχο και αν επιθυμεί μπορεί να κάνει έξοδο από το παιχνίδι. Η δεύτερη σκηνή είναι το πρώτο επίπεδο του παιχνιδιού στο οποίο ο χρήστης μπορεί να απολαύσει το παιχνίδι με την αντίστοιχη μουσική του επιπέδου. Η τρίτη σκηνή είναι το δεύτερο επίπεδο στο οποίο η δυσκολία είναι μεγαλύτερη. Στην τέταρτη σκηνή που είναι και η τελευταία επίπεδο είναι και η πιο μεγάλη πρόκληση για τον χρήστη. Τέλος η Πέμπτη δείχνει το σκορ του χρήστη και αν επιθυμεί να συνεχίσει ή την έξοδο του από το παιχνίδι. Για να πετύχουν όλα αυτά χρησιμοποιήθηκαν πολλά εργαλεία της Unity αλλά επίσης χρησιμοποιήθηκαν διάφορα scripts και από C#. Τέλος τα μοντέλα που χρησιμοποιήθηκαν ήταν έτοιμα είτε από το asset store της Unity είτε από διάφορες ιστοσελίδες.

1.2 Κίνητρο Για Την Διεξαγωγή Της Εργασίας

Το κίνητρο για την διεξαγωγή της εργασίας αυτής είναι ένα μεταπτυχιακό στο εξωτερικό πάνω στο game developing στο οποίο είναι πολύ σημαντικό να γνωρίζει κάποιος Unity και C#. Θέλησα να κάνω το συγκεκριμένο παιχνίδι διότι πέρα από την προετοιμασία που μου προσφέρει είναι και η πρώτη μου εμπειρία στον τομέα δημιουργίας παιχνιδιών και είναι ένα καλό ξεκίνημα για κάποιον φοιτητή στον τομέα του gaming και του android development.

1.3 Σκοπός Και Στόχοι Εργασίας

Σκοπός της συγκεκριμένης εργασίας είναι να δημιουργηθεί ένα ηλεκτρονικό παιχνίδι μέσω της μηχανής παιχνιδιών Unity και ανάλυση των τρόπων που μπορεί ο δημιουργός να προσκομίσει έσοδα . Στόχος είναι να αποτελέσει ένα λειτουργικό παιχνίδι και να κινήσει το ενδιαφέρον των χρηστών. Επίσης σαν στόχο έχει κάτι διαφορετικό πέρα από την ψυχαγωγία, έχει και την χαλάρωση που δημιουργεί η μουσική του παιχνιδιού και το περιβάλλον του παιχνιδιού.

Κεφάλαιο 2ο : Η ιστορία της ανάπτυξης ηλεκτρονικών παιχνιδιών

2.1 Τα πρώτα βήματα της βιομηχανίας παιχνιδιών

Η ανάπτυξη ηλεκτρονικών παιχνιδιών αποτελεί ένα μεγάλο κομμάτι της ψηφιακής βιομηχανίας, αλλά και ένα σημαντικό κεφάλαιο στον τομέα της πληροφορικής και του προγραμματισμού. Αξίζει να αναφέρουμε πως στην ανάπτυξη ενός μεγάλου project που αφορά ένα video game, συνήθως, συμμετέχει ένας μεγάλος αριθμός ατόμων με τον κάθε έναν να έχει διαφορετικές αρμοδιότητες. Ενδεικτικά μερικές από τις θέσεις που υπάρχουν σε μια ομάδα ανάπτυξης ηλεκτρονικών παιχνιδιών είναι: σχεδιαστής του παιχνιδιού, υπεύθυνος του καλλιτεχνικού μέρους, προγραμματιστής, σχεδιαστής επιπέδων δυσκολίας υπεύθυνος ήχου και άλλα.

Η ιστορία της ανάπτυξης ηλεκτρονικών παιχνιδιών ξεκινάει από το 1952 όταν και κατασκευάστηκε το πρώτο video game, που ήταν βέβαια πολύ περιορισμένων δυνατοτήτων και πολύ λίγοι μπορούσαν να το χρησιμοποιήσουν. Οι πρώτες γενιές ηλεκτρονικών παιχνιδιών που κυκλοφόρησαν στην αγορά, κατασκευάστηκαν τη δεκαετία του 1970. Από τότε μέχρι σήμερα, βεβαίως, η βιομηχανία των ηλεκτρονικών παιχνιδιών έχει παρουσιάσει αλματώδη αύξηση με τα video games να μην περιορίζονται μόνο

στους υπολογιστές και στις κονσόλες, αλλά να εισχωρούν δυναμικά στο κόσμο των κινητών συσκευών (smartphones, tablets).

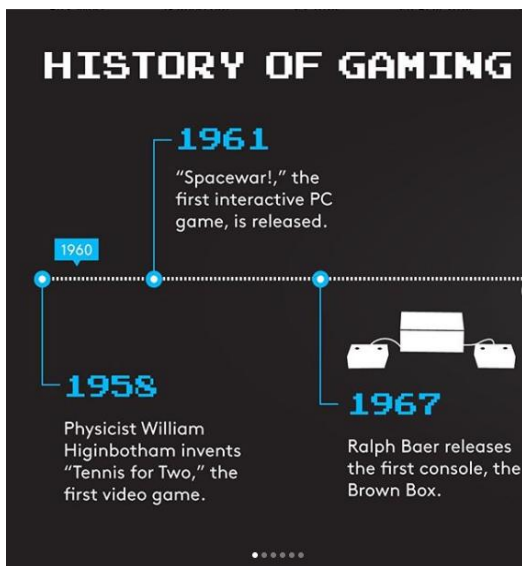
Τα πρώτα παιχνίδια κατασκευάζονταν ως μοναδικές οντότητες, με τους προγραμματιστές να δημιουργούν τα πάντα κυριολεκτικά από το μηδέν και με τη σχεδίαση των γραφικών να αποτελεί μια πολύ επίπονη διαδικασία. Σήμερα τα εργαλεία ανάπτυξης παιχνιδιών (game engines) αποτελούν μια πολύ ισχυρή βάση πάνω στην οποία μπορεί κάποιος να στηριχτεί για την κατασκευή ενός video game. Ο όρος game engine ήρθε στην επιφάνεια στα μέσα της δεκαετίας του 1990. Τότε, εταιρείες που είχαν κατασκευάσει δικά τους εργαλεία ανάπτυξης πουλούσαν άδειες χρήσης των εργαλείων τους σε άλλους δημιουργούς. Τα τελευταία χρόνια, έχουν δημιουργηθεί πολλά εργαλεία, τα οποία προσφέρουν και κάποιες δωρεάν εκδόσεις τους, δίνοντας τη δυνατότητα σε ακόμα περισσότερους developers να σχεδιάσουν και να δημιουργήσουν τα δικά τους ηλεκτρονικά παιχνίδια.

Η γενικότερη τάση που προωθείται, στο χώρο των υπολογιστικών συστημάτων, είναι το cloud computing, δηλαδή η μεταφορά όλων των δεδομένων από τους τοπικούς υπολογιστές, σε απομακρυσμένους αποθηκευτικούς χώρους που είναι προσβάσιμη μέσω του διαδικτύου. Ακολουθώντας αυτή τη τάση, πολλά video games κατασκευάζονται ούτως ώστε να μην απαιτούν από το χρήστη να κάνει κάποια εγκατάσταση στον υπολογιστή του, αλλά να είναι προσβάσιμα και εκτελέσιμα απ'ευθείας από τον browser. Τα παιχνίδια αυτά είναι γνωστά και ως browser games. Αυτό φυσικά επιτυγχάνεται με τη χρήση κάποιου plug-in. Κάποια από τα πιο γνωστά plug-in είναι το Flash Player, το Java, το Shockwave αλλά και το Unity web player.



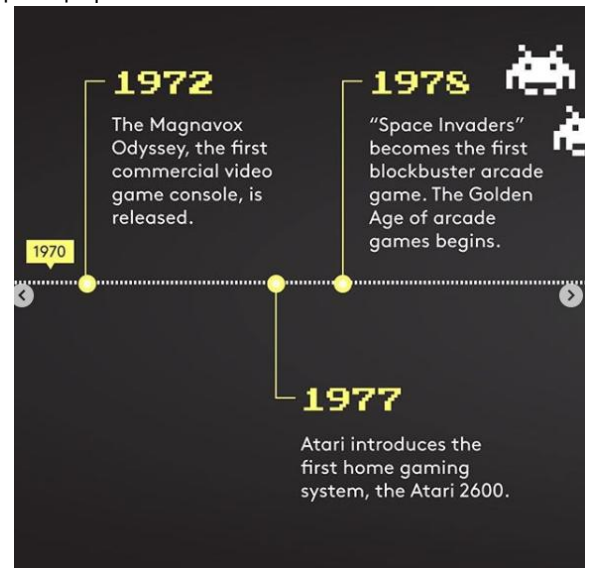
Video Games

2.2 Σύντομη ιστορική αναδρομή και εξέλιξη των βιντεοπαιχνιδιών

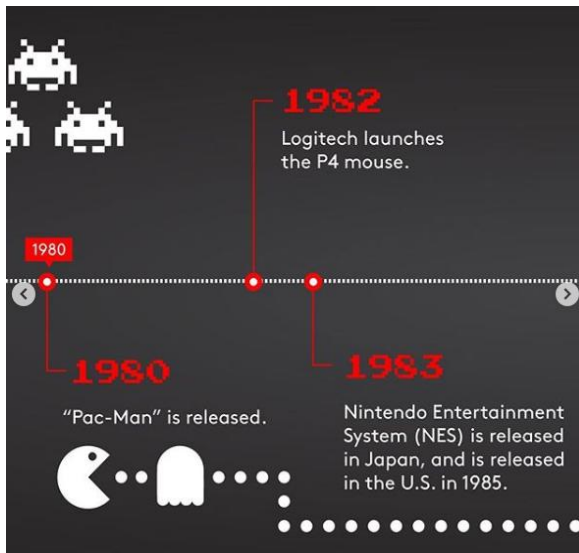


Τα πρώτα παιχνίδια που δημιουργήθηκαν δεν ήταν ιδιαίτερα ψυχαγωγικά, και ο τρόπος ανάπτυξής τους δεν εστίαζε στην εμπειρία χρηστών. Τα παιχνίδια απαιτούσαν ισχυρούς κεντρικούς υπολογιστές για να «τρέξουν». Το 1952, το «OXO» (τριλιζα), ήταν από τα πρώτα ηλεκτρονικά παιχνίδια, όπου για την απεικόνισή του, έγινε χρήση μιας ψηφιακής συσκευής προβολής. Ένας φυσικός, το 1958, έφτιαξε ένα παιχνίδι με τίτλο «tennis for two», χρησιμοποιώντας ως συσκευή αναπαράστασης, ένα παλμογράφο. Το 1961, μια ομάδα σπουδαστών έφτιαξε το «Spacewar!».

Η χρυσή εποχή των arcades ξεκίνησε το 1978 με την έκδοση του παιχνιδιού «Space Invaders» της εταιρίας Taito. Η τεράστια επιτυχία του, ενέπνευσε πολλούς κατασκευαστές να εισχωρήσουν στην αγορά. Το 1979, τα arcades έγιναν ακόμα πιο δημοφιλή με την άφιξη τίτλων, όπως το γνωστό «Pac-man» της Namco. Οι σχεδιαστές πολλών πρώιμων παιχνιδιών, αργότερα μεταβίβασαν την δουλειά τους στη βιομηχανία. Ογκώδης ηλεκτρονικές κονσόλες με αυτονομία, πρόσφεραν εύκολη και γρήγορη ψυχαγωγία, παρέχοντας ένα έγχρωμο παιχνίδι η κάθε μια. Το κόστος και το μέγεθός τους, τις καθιστούσε απαγορευτικές για οικιακή χρήση. Ήταν τοποθετημένες σε πολλά μέρη όπως μπαρ, εστιατόρια, ξενοδοχεία και χώρους ψυχαγωγίας και εμπορικά κέντρα, όπου οι χρήστες αντάλλαζαν ένα κέρμα ανά παιχνίδι. Τα παιχνίδια τύπου arcade ήταν αρκετά λαοφιλή τις δεκαετίες '80 και '90, βαθμιαία όμως, η δημοτικότητά τους μειώθηκε.

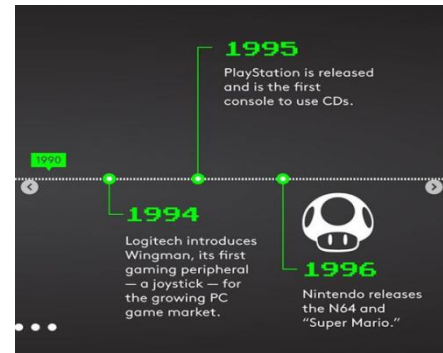


Ένα ουσιαστικό επιχειρηματικό πλάνο σχεδιάστηκε, όταν οι κονσόλες πρώτης γενιάς εισέβαλαν στην αγορά προορισμένες για οικιακή χρήση. Στη βιομηχανία δεν υπήρχε ιδιαίτερη καινοτομία σε σχέδια παιχνιδιών, με αποτέλεσμα ένας μεγάλος αριθμός συσκευών, να έχει πολλά πανομοιότυπα παιχνίδια. Συγχρόνως, οι προσωπικοί υπολογιστές έκαναν την εμφάνισή τους και πλέον μεμονωμένα άτομα μπορούσαν να προγραμματίσουν δικά τους παιχνίδια. Απλοϊκά παιχνίδια παράγονταν εύκολα και γρήγορα, δεδομένου ότι

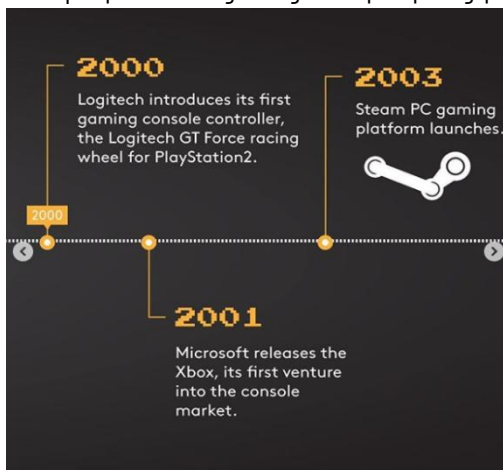


οι σχεδιαστές είχαν περιορισμούς μνήμης και γραφικών, οι οποίοι δεν επέτρεπαν πολυσύνθετη ανάπτυξη. Μεγαλύτερες επιχειρήσεις προσλάμβαναν μικρές ομάδες προγραμματιστών, που η κάθε μία αφιερωνόταν στην ανάπτυξη ενός τίτλου. Έτσι το 1984, οι κονσόλες παραμερίστηκαν και τα ηνία στις πωλήσεις πήραν, οι προσωπικοί υπολογιστές. Ένα χρόνο αργότερα, η ιαπωνική εταιρία Nintendo ζωντάνεψε ξανά την αγορά κονσόλας με μια νέα 8-bit συσκευή (NES) συνοδευμένη με τίτλους όπως, Super Mario Bros.

Η δεκαετία 1990 ξεχώρισε για την καινοτομία βιντεοπαιχνιδιών και τη μετάβαση γραφικών από 2d σε 3d. Νέα είδη, δημιουργήθηκαν όπως first-person-shooter, στρατηγικής και παιχνίδια διαδικτύου. Παράλληλα προστέθηκαν στην αγορά μικρές φορητές κονσόλες, όπως το Gameboy, με τεράστια απήχηση στο νεανικό κοινό. Ακολούθησαν οι κονσόλες 5ης γενιάς 32-64 bit και οι κασέτες αντικαταστάθηκαν με compact disks (cd).



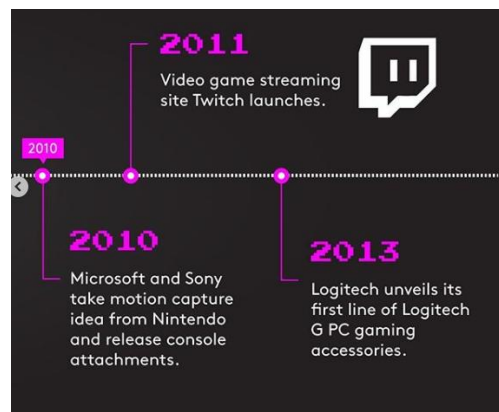
Τα κινητά τηλέφωνα μετατράπηκαν, το 1998, σε μίνι πλατφόρμες βιντεοπαιχνιδιών, όταν η φιλανδική Νοκία εγκατέστησε στα μοντέλα της, το ασπρόμαυρο παιχνίδι Snake. Από τότε πολλές μεγάλες εταιρίες έκαναν το ίδιο προσθέτοντας, απλά παιχνίδια στα κινητά τους τηλέφωνα. Τέλη δεκαετίας του 1990, οι ογκώδεις κονσόλες που λειτουργούσαν με κέρμα, αποσύρθηκαν, παραμένοντας στις αναμνήσεις μικρών και μεγάλων. Οι συνεχής αυξανόμενες



επεξεργαστικές και γραφικές ικανότητες των προσωπικών υπολογιστών σε συνδυασμό με άλλες δυνατότητες που παρείχαν, όπως η σύνδεση στο διαδίκτυο, τους κατέστησε μια αρκετά πιο δημοφιλή επιλογή. Η πρώτη δεκαετία του 2000 παρουσίασε καινοτομία στις κονσόλες, αλλά και στους προσωπικούς υπολογιστές. Συνάμα, η αγορά φορητών παιχνιδιομηχανών έγινε περισσότερο ανταγωνιστική. Το 2001 πάνω από 3000 παιχνίδια εκδόθηκαν για PC, όμως μόνο 50 με 100 παιχνίδια απέφεραν κέρδη. Το 2005 το κόστος παραγωγής παιχνιδιών για

κονσόλες έφτανε τα 3 με 6 εκατομμύρια δολάρια. Η βιομηχανία παιχνιδιών συνεχώς επεκτεινόταν. Τα εισοδήματα της βιομηχανίας πενταπλασιάστηκαν σε σύγκριση με τη δεκαετία του '90. Το 2007 η μερίδα του εισοδήματος ήταν \$9,5 δισεκατομμύρια, υπερβαίνοντας αυτή, της βιομηχανίας κινηματογράφου. Όσο οι ευρυζωνικές συνδέσεις διαδίδονταν, πολλοί εκδότες στράφηκαν στην ανάπτυξη βιντεοπαιχνιδιών διαδικτύου για έναν χρήστη και παιχνιδιών για πολλούς χρήστες όπου ταυτόχρονα συνδέονται και αλληλεπιδρούν μεταξύ τους (MMO). Το ενδιαφέρον των χρηστών επικεντρώθηκε και σε δωρεάν περιστασιακά παιχνίδια, που προσφέρονται σε διαδικτυακές κοινότητες.

Εν έτη 2010, την 7η γενιά χαρακτηρίζουν κονσόλες, που υποστηρίζουν πρόσβαση στο διαδίκτυο (playstation 3, Nintendo wii), αλλά και ισχυρές φορητές συσκευές (playstation psp, Nintendo ds), με καταπληκτικά γραφικά. Τέλος, τα κινητά τηλέφωνα 4ης γενιάς λαμβάνουν μεγάλο μερίδιο στην αγορά, όχι μόνο ως συσκευές τηλεπικοινωνίας αλλά και ως μίνι υπολογιστές με δυνατότητα ασύρματης πρόσβασης, και εγκατάστασης ψυχαγωγικών παιχνιδιών. Τώρα οι υπολογιστές και οι κονσόλες κατασκευάζονται για να υποδεχτούν βίντεο παιχνίδια φτιαγμένα με στερεοσκοπική τεχνική. Με αυτή την τεχνική, γίνεται χρήση διπλών εικόνων, που παράγονται από ελαφρώς διαφορετικές γωνίες προκειμένου να δημιουργηθεί μια τρισδιάστατη ψευδαισθήση. Αυτό απαιτεί συνήθως χρήση ειδικών γυαλιών.



Η ανάπτυξη βιντεοπαιχνιδιών όπως και άλλες μορφές ψυχαγωγίας, έχει μετατραπεί σε διεπιστημονικό τομέα. Για τη δημιουργία ενός παιχνιδιού τα συμβαλλόμενα μέλη χωρίζονται σε δύο κύριες κατηγορίες, σε προγραμματιστές και σχεδιαστές γραφικών. Με το πέρασμα των χρόνων ο τομέας έχει εξελιχθεί τόσο, ώστε για μια ολοκληρωμένη δουλειά είναι απαραίτητοι, ειδικοί και από άλλους τομείς, όπως της μουσικής, του κινηματογράφου, της τηλεόρασης, μουσικοί συνθέτες, τεχνικοί ήχου κ.α. Αυτή η ομάδα ανθρώπων διευθύνεται από παραγωγούς όπου ρόλος τους είναι να παρέχουν τελευταίας τεχνολογίας μηχανήματα και να προωθούν το τελικό προϊόν.

2.3 Είδη ηλεκτρονικών παιχνιδιών

Τα ηλεκτρονικά παιχνίδια χωρίζονται σε διαφορετικά είδη ανάλογα με την εμπειρία παιχνιδιού που προσφέρουν (gameplay). Κάθε είδος με τη σειρά του χωρίζεται σε κατηγορίες ανάλογα με την πλήρωση διαφόρων κριτηρίων, όπως για παράδειγμα ο αριθμός συμμετεχόντων παικτών, η προοπτική απεικόνισης, η χρήση τριών ή δύο διαστάσεων, η θεματική τους, ο τρόπος χειρισμού τους. Άξιο αναφοράς είναι το γεγονός ότι ένα παιχνίδι μπορεί να ανήκει ταυτόχρονα

σε δύο είδη, ενώ ένα είδος μπορεί να περιλαμβάνει δύο εκ πρώτης όψεως εντελώς ξένα μεταξύ τους παιχνίδια, όπως ένα παιχνίδι ποδοσφαίρου κι ένα παιχνίδι εξομοίωσης πτήσης, τα οποία ανήκουν στο είδος των εξομοιωτών. Στη συνέχεια ακολουθούν τα σημαντικότερα είδη ηλεκτρονικών παιχνιδιών, χαρακτηριστικές κατηγορίες που συναντάμε σε αυτά καθώς και μία μικρή “φωτογραφική” αναφορά στους κυριότερους εκπροσώπους κάθε είδους.

Παιχνίδια Δράσης (Action Games)

Πρόκειται για παιχνίδια στα οποία ο παίκτης υποβάλλεται σε δοκιμασίες (challenges) κατά πλειοψηφία σωματικής ικανότητας. Πολλές φορές συναντώνται και άλλου τύπου δοκιμασίες όπως αγώνες ταχύτητας, γρίφοι ή συλλογή αντικειμένων αλλά αυτές δεν αποτελούν το κυρίαρχο είδος προκλήσεων. Ο παίκτης μπορεί να συναντήσει ακόμα δοκιμασίες τακτικής ή εξερεύνησης αλλά τα παιχνίδια αυτά πρώτα και κύρια απαιτούν από τον παίκτη υψηλή ταχύτητα αντίδρασης και καλό συντονισμό χεριού-ματιού. Οι παίκτες βρίσκονται συνήθως υπό την πίεση χρόνου με αποτέλεσμα αυτός να μην είναι αρκετός για πολύπλοκο στρατηγικό σχεδιασμό. Τα παιχνίδια δράσης μπορεί να απαιτούν τη λύση γρίφων, οι οποίοι όμως είναι απλοί εξαιτίας του χρονικού αυτού περιορισμού (Rollings, Adams, 2006). Πρόκειται ίσως για το βασικότερο κι ευρύτερο σε υποκατηγορίες είδος ηλεκτρονικών παιχνιδιών. Υποκατηγορίες που συναντώνται είναι οι εξής:

- Σκόπευσης (Shooter): Στα παιχνίδια ο παίκτης χρησιμοποιώντας κάποιο όπλο μακρινής απόστασης (συνήθως πιστόλι και σχετικά με αυτό είδη) και πυρομαχικά, έχει σκοπό στοχεύοντας και πυροβολώντας τους ανταγωνιστές που συναντάει να φτάσει δια μέσου διαφορετικών επιπέδων στο τέλος του παιχνιδιού. Πρόκειται για ένα είδος με μεγάλη εξάπλωση και πολλές υποκατηγορίες με κυριότερη τα παιχνίδια Στόχευσης Πρώτου Προσώπου (First-Person Shooter). Άλλες υποκατηγορίες μεταξύ άλλων είναι τα Στόχευσης Τρίτου Προσώπου (Third-Person Shooter), Τακτικής (Tactical Shooter) και τα Shoot'em Up.



Αντιπροσωπευτικά παιχνίδια FPS: Wolfenstein 3D (id Software), DOOM 2 (id Software), Quake 2 (id Software), Half Life 2 (Valve), Call of Duty Modern Warfare (Infinity Ward)

- Μάχης (Fighting): Γνωστά και ως Παιχνίδια Ξύλου στην Ελλάδα. Πρόκειται για παιχνίδια στα οποία συνήθως δύο χαρακτήρες, εκ των οποίων ο ένας συνήθως ελέγχεται από τον υπολογιστή, μάχονται μεταξύ τους. Ο παίκτης μέσω ενός χειριστηρίου εισάγει κινήσεις οι οποίες όταν είναι επιτυχημένες δημιουργούν μία αλυσίδα κινήσεων (combo) ή οποία όσο περισσότερο μεγαλώνει τόσο μεγαλύτερη ζημιά προκαλεί στον αντίπαλο. Χαρακτηρίζονται από την ιδιαίτερη σημασία που δίνεται στην εικονογράφηση και την ομαλή κίνηση των χαρακτήρων (animation), στην άμεση ανταπόκριση στις κινήσεις που εισάγει ο παίκτης και στη σωστή εξισορρόπηση των δυνατοτήτων των διαφορετικών χαρακτήρων που έχει ο παίκτης στη διάθεσή του. Διακρίνονται και αυτά σε πολλές υποκατηγορίες ανάλογα με την τεχνοτροπία των γραφικών τους, τη χρήση δύο ή τριών διαστάσεων για τα μοντέλα του παίκτη και των περιβαλλόντων, τον αριθμό των παικτών που μάχονται ταυτόχρονα στην οθόνη.



Αντιπροσωπευτικά παιχνίδια Fighting: Virtua Fighter 1 (SEGA), Mortal Kombat (Midway), TEKKEN (NamcoBandai), Street Fighter IV (Capcom), Skullgirls (Reverge Labs)

- Πλατφόρμας (Platform): Στα παιχνίδια αυτά ο ελεγχόμενος από τον παίκτη χαρακτήρας χρειάζεται να τρέχει και να πηδάει επάνω από εμπόδια και από πλατφόρμα σε πλατφόρμα (εξού και το όνομα) ενώ παράλληλα αποφεύγει εμπόδια που οφείλονται είτε στο περιβάλλον, είτε στους ανταγωνιστές του παιχνιδιού, είτε στους νόμους της φυσικής (βαρύτητα). Ιστορικά η προοπτική απεικόνισης της συγκεκριμένης κατηγορίας ξεκίνησε ως διςδιάστατη πλευρική (2D side scrolling), ενώ εξελίχθηκε κατά τα μέσα της δεκαετίας του 1990 σε τρισδιάστατη. Εξακολουθούν να κυκλοφορούν επιτυχημένα παιχνίδια πλατφόρμας -πρόκειται κυρίως για παιχνίδια τα οποία έχουν καταφέρει με μεγάλη επιτυχία το πέρασμα από τις δύο στις τρεις διαστάσεις. Για κάποια χρονική περίοδο (1990-μέσα 2000) η κατηγορία αυτή παιχνιδιών ήταν ίσως η πιο δημοφιλής, κάτι που σήμερα δεν ισχύει στον ίδιο βαθμό.



Αντιπροσωπευτικά παιχνίδια Platform: Super Mario (Nintendo), Sonic (SEGA), Rayman (Ubisoft), Pandemonium (Crystal Dynamics), Crash Bandicoot (Naughty Dog)

- Beat'em Up: Στα παιχνίδια αυτά τα οποία επίσης γνώρισαν μεγάλη ανάπτυξη και επιτυχία την δεκαετία του 1990 ο παίκτης μέσω του χαρακτήρα που ελέγχει τίθεται αντιμέτωπος με μία μεγάλη και φαινομενικά ατελείωτη ορδή αντιπάλων χαμηλότερης δυναμικής από τον ίδιο καθώς διασχίζει κατά μήκος και πλάτος μία δισδιάστατη πίστα. Στόχος είναι να φτάσει στο τέλος της και να νικήσει τον τελικό αντίπαλο του επιπέδου, περνώντας έτσι στην επόμενη πίστα. Πρόκειται για παιχνίδια που μέσω επαναλαμβανόμενων κινήσεων ακονίζουν τις ικανότητες του παίκτη ξεκλειδώνοντας νέες κινήσεις και βελτιώνοντας το τελικό σκορ του παίκτη που αποτελεί άλλωστε και τον απώτερο στόχο. Ξεκίνησαν και αυτά ως δισδιάστατα και πέρασαν μετέπειτα στις τρεις διαστάσεις.



Αντιπροσωπευτικά παιχνίδια Beat'em Up: Golden Axe (SEGA), Streets of Rage (SEGA), Double Dragon (Technos Japan), Devil May Cry (Capcom), Ninja Gaiden (Tecmo)

Παιχνίδια Περιπέτειας (Adventure Games):

Στα παιχνίδια αυτής της κατηγορίας ο ήρωας που χειρίζεται ο παίκτης είναι ο πρωταγωνιστής της ιστορίας ο οποίος λύνοντας γρίφους προχωράει την ιστορία. Οι γρίφοι είναι διαφόρων ειδών και συνήθως σχετίζονται με την αλληλεπίδραση του παίκτη με χαρακτήρες και αντικείμενα του περιβάλλοντος. Κι εδώ συναντάμε πολλές υποκατηγορίες που έχουν να κάνουν με την προοπτική απεικόνισης, με την τεχνοτροπία των γραφικών αλλά και με τις παραλλαγές στην εμπειρία του παιχνιδιού (gameplay) που συναντάμε. Θα ακολουθήσει εκτενέστερη αναφορά στο είδος αυτό στο δεύτερο κεφάλαιο της εργασίας.



Αντιπροσωπευτικά παιχνίδια Adventure: Monkey Island (Lucasfilm), Gabriel Knight Sins of the Fathers (Sierra), The Dig (Lucasarts), Grim Fandango (Lucasarts), The Longest Journey (Funcom)

Παιχνίδια Δράσης-Περιπέτειας (Action-Adventure Games):

Πρόκειται για παιχνίδια που όπως φαίνεται και από τον τίτλο τους συνδυάζουν τα χαρακτηριστικά των δύο αντίστοιχων ειδών, δηλαδή την εξερεύνηση και την επίλυση (ευκολότερων γενικά) γρίφων σε συνδυασμό με σκηνές δράσης και μάχης. Πρόκειται για ένα είδος παιχνιδιών με μεγάλη ανάπτυξη τελευταία. Αποτελεί επίσης την κατηγορία που κατατάσσεται κάποιο παιχνίδι το οποίο δεν εμπίπτει εύκολα σε κάποια άλλη γνωστή κατηγορία.



Αντιπροσωπευτικά παιχνίδια Action-Adventure: Tomb Raider (EIDOS Interactive), Metal Gear Solid (Konami), Assassin's Creed 2 (Ubisoft), Batman Arkham Asylum (Rocksteady), Uncharted 3 Drake's Deception (Naughty Dog)

Παιχνίδια Ρόλων (Role-playing Games):

Πρόκειται για ένα είδος το οποίο βασίστηκε στα επιτραπέζια παιχνίδια ρόλων τα λεγόμενα pen and paper role playing games. Στην μετάβασή τους από το χαρτί στους υπολογιστές και στις κονσόλες διατήρησαν πολλά χαρακτηριστικά τους, όπως για παράδειγμα τους κόσμους φαντασίας που διαδραματίζονται και την ανάλογη θεματολογία. Βασικό ζητούμενο είναι η αύξηση του επιπέδου του χαρακτήρα (level up) μέσω της εμπειρίας που κερδίζει μαχόμενος, αναλαμβάνοντας και φέρνοντας σε πέρας αποστολές ή πραγματοποιώντας άλλες δραστηριότητες (εξάσκηση ικανοτήτων του). Χαρακτηριστικά που συναντώνται στα παιχνίδια αυτά είναι η ιδιαίτερη φροντίδα και σημασία που δίνεται στην ιστορία όπως επίσης και στη δημιουργία ενός ολοκληρωμένου κόσμου με δικούς του αυστηρά ορισμένους κανόνες. Το είδος αυτό διακρίνεται σε υποκατηγορίες με βάση τη θεματολογία τους, την προοπτική απεικόνισης, τον αριθμό των παικτών και το εάν διαθέτουν δυνατότητα για παιχνίδι χωρίς να απαιτείται σύνδεση στο διαδίκτυο, ή μπορούν να παιχτούν μόνο σε αυτό.



Αντιπροσωπευτικά παιχνίδια RPG: Ultima (Origin Systems), Might And Magic (New World Computing), Baldur's Gate (BioWare), Final Fantasy (SquareSoft), Elder Scrolls V: Skyrim (Bethesda)

Παιχνίδια Εξομοίωσης (Simulation Games):

Τα παιχνίδια αυτά έχουν ως στόχο την εξομοίωση δραστηριοτήτων, είτε αυτές είναι φυσικές και σωματικές (πέταγμα αεροπλάνων, ποδόσφαιρο, γκολφ είναι ορισμένα παραδείγματα), είτε αυτές αποτελούν μορφές διαχείρισης (προπονητής ομάδας ποδοσφαίρου, δήμαρχος μίας πόλης), με όσο το δυνατόν περισσότερο ρεαλισμό. Δίνεται τεράστια σημασία στην ακρίβεια, στο ρεαλισμό και στην αυθεντικότητα που αποτελούν χαρακτηριστικά στοιχεία της εμπειρίας παιχνιδιού.



Αντιπροσωπευτικά παιχνίδια Simulation: SIMCITY (Maxis), The Sims (Electronic Arts), SOCOM (SONY), Flight Simulator (Microsoft), Train Simulator (Railsimulator.com)

Παιχνίδια Στρατηγικής (Strategy Games):

Στα παιχνίδια αυτά ο παίκτης αναλαμβάνει την διοίκηση ενός στρατού. Στις αρμοδιότητές του περιλαμβάνονται η οργάνωση, ανάπτυξη, εκπαίδευση και καθοδήγηση του κατά τη διάρκεια μαχών. Εξίσου σημαντικό μέρος της εμπειρίας παιχνιδιού είναι και η διαχείριση των πόρων του παιχνιδιού που προσφέρουν τη δυνατότητα για απόκτηση επιπλέον μονάδων και αναβάθμισή τους. Οι αντίπαλοι του παίκτη έχουν τον ίδιο στόχο και πρόσβαση στους ίδιους πόρους. Χωρίζονται σε δύο βασικές κατηγορίες ανάλογα με τον χρόνο που έχουν τη δυνατότητα οι συμμετέχοντες να πραγματοποιήσουν τις ενέργειές τους. Εάν ο κάθε παίκτης περιμένει τη σειρά του για να δράσει (έχοντας συνήθως και κάποιο χρονικό περιορισμό), τότε μιλάμε για παιχνίδια στρατηγικής “με βάση τη σειρά” (turn based), ενώ εάν όλοι οι παίκτες μπορούν να δράσουν ταυτόχρονα και χωρίς χρονικούς περιορισμούς, τότε μιλάμε για παιχνίδια στρατηγικής “πραγματικού χρόνου” (real time). Ανεξάρτητα της παραπάνω κατηγοριοποίησης, τα παιχνίδια αυτά παίζονται είτε στο διαδίκτυο, είτε χωρίς να απαιτείται σύνδεση σε αυτό. Άλλη κατηγοριοποίηση είναι με βάση την προοπτική απεικόνισης.



Αντιπροσωπευτικά παιχνίδια Strategy: Dune II (Westwood Studios), Red Alert (Westwood Studios), Age of Empires: Age of Kings (Microsoft), Starcraft 2, Wings of Liberty (Blizzard), Total War (Creative Assembly)

Παιχνίδια Αγώνων Ταχύτητας (Racing Games):

Όπως γίνεται κατανοητό και από την ονομασία τους πρόκειται για παιχνίδια στα οποία ο παίκτης ελέγχοντας κάποιο όχημα (όχι αποκλειστικά αυτοκίνητο) τρέχει σε πίστες με σκοπό να επικρατήσει των αντιπάλων του. Η κατηγορία αυτή πολλές φορές μπορεί ως κάποιο βαθμό να μπλέκεται και με το είδος της εξομοίωσης που αναφέρθηκε προηγουμένως. Κι εδώ έχουμε διάφορες κατηγορίες ανάλογα με μία πληθώρα κριτηρίων, από την προοπτική απεικόνισης, τη σημασία στο ρεαλισμό (τείνουν προς την εξομοίωση) ή όχι (arcade), το εάν παίζονται από περισσότερους του ενός παίκτη στο ίδιο σύστημα και μοιράζονται την οθόνη απεικόνισης ή εάν οι παίκτες παίζουν μέσω του δικτύου κλπ.



Αντιπροσωπευτικά παιχνίδια Racing: Ridge Racer (Namco), Sega Rally (SEGA), Need for Speed (Electronic Arts), Collin McRae Rally (Codemasters), Gran Turismo 6 (Polyphony Digital)

Αθλητικά Παιχνίδια (Sport Games):

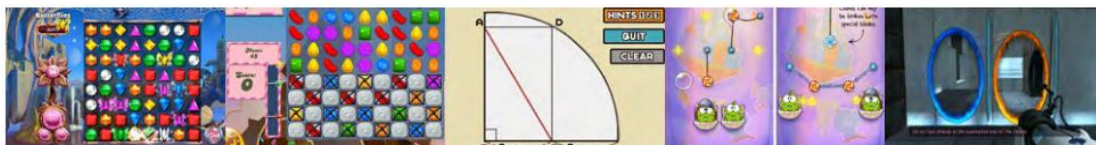
Στο είδος αυτό περιλαμβάνονται όλα τα παιχνίδια τα οποία έχουν ως αντικείμενο κάποιο άθλημα, με δημοφιλέστερα τα παιχνίδια ποδοσφαίρου. Πρόκειται για ένα είδος στο οποίο ανήκει μεγάλη γκάμα παιχνιδιών στα οποία περιλαμβάνονται και ατομικά και ομαδικά αθλήματα. Όπως συμβαίνει και με την κατηγορία παιχνιδιών αγώνων ταχύτητας κι εδώ εμφανίζεται το φαινόμενο να υπάρχουν και στοιχεία εξομοίωσης στην προσφερόμενη εμπειρία παιχνιδιού.



Αντιπροσωπευτικά παιχνίδια Sports: Pro Evolution Soccer (KONAMI), MLB The Show (SCE San Diego), NHL 2014 (Electronic Arts), NBA Live (Electronic Arts), FIFA 2014 (Electronic Arts)

Παιχνίδια Γρίφων (Puzzle Games):

Συγγενής κατηγορία με τα περιπέτειας. Στόχος του παίκτη είναι η επίλυση γρίφων μεγάλης ποικιλίας κατηγοριών (μνήμης, μαθηματικών, παρατηρητικότητας, ταχύτητας αντιδράσεων, αντιστοίχιση σχημάτων μεταξύ άλλων). Συχνά υπάρχει ένα υποτυπώδες σενάριο που λειτουργεί ως πρόφαση για την ύπαρξη των γρίφων οι οποίοι μπορεί να είναι τελείως ασύνδετοι με το περιβάλλον του παιχνιδιού, χωρίς αυτό όμως να αποτελεί τον κανόνα.



Αντιπροσωπευτικά παιχνίδια Puzzle: Bejeweled (Popcap), Candy Crush Saga (King), Professor Layton (Level-5), Cut the Rope (Zeptolab), Portal (Valve)

Παιχνίδια Ρυθμού και Παιχνίδια Μουσικής (Rhythm-Music Games):

Στα παιχνίδια αυτά η εμπειρία παιχνιδιού (gameplay) συνίσταται στο να δράσει ο παίκτης (να κινηθεί ή να ακολουθήσει ένα ρυθμό) ανάλογα με κάποια διέγερση διαφορετική για κάθε παιχνίδι (μουσική διέγερση στα παιχνίδια μουσικής). Πολλές φορές τα παιχνίδια αυτά απαιτούν την ύπαρξη ειδικών χειριστηρίων.



Αντιπροσωπευτικά παιχνίδια Rhythm: Frequency (Harmonix), Parappa the Rapper (NanaOn-Sha), Dance Dance Revolution (KONAMI), Rock Band (Harmonix), Singstar (SONY)

Casual Games:

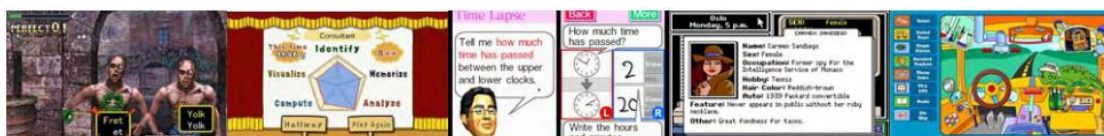
Πρόκειται για ένα είδος παιχνιδιών που περιλαμβάνει πολλά διαφορετικά είδη και που γνωρίζει μεγάλη άνθηση τα τελευταία χρόνια με αποτέλεσμα την ενασχόληση όλο και μεγαλύτερου μέρους του πληθυσμού με τα ηλεκτρονικά παιχνίδια. Η μεγάλη ανάπτυξη τους οφείλεται κυρίως στα έξυπνα κινητά νέας γενιάς για τα οποία έχουν αναπτυχθεί πολλά ηλεκτρονικά παιχνίδια τέτοιου είδους. Ορισμένες ενδεικτικές υπό κατηγορίες παιχνιδιών είναι οι εξής:

- Παραδοσιακά Παιχνίδια (Traditional Games): Πρόκειται για καθημερινά παραδοσιακά παιχνίδια (παιχνίδια τράπουλας, τάβλι, scrabble) τα οποία έχουν μετατραπεί κατάλληλα ώστε να παίζονται σε υπολογιστές και σε ηλεκτρονικές συσκευές γενικά.



Αντιπροσωπευτικά παραδοσιακά παιχνίδια: Σκάκι (Chessmaster-Ubisoft), Scrabble, Mahjong, Poker, Backgammon

- Εκπαιδευτικά Παιχνίδια (Educational Games): Πρόκειται για παιχνίδια με εκπαιδευτικό σκοπό, πιο συγκεκριμένα την εκμάθηση νέων αλλά και την εξάσκηση υπάρχόντων ικανοτήτων των παικτών. Τέτοια παιχνίδια ποικίλουν σε θέμα και δυσκολία αλλά και στις ηλικίες και το μορφωτικό επίπεδο που απευθύνονται



Αντιπροσωπευτικά εκπαιδευτικά παιχνίδια: The Typing of the Dead (WOW Entertainment), Big Brain Academy (Nintendo), Dr. Kawashima's Brain Training (Nintendo), Carmen Sandiego Series (Broderbound), The Magic School Bus (Microsoft)

2.4 Γιατί επιλέγουμε το Unity

Υπάρχει μια πληθώρα εργαλείων για την ανάπτυξη 3D παιχνιδιών. Η επιλογή του κατάλληλου εργαλείου αποτελεί το πρώτο και ιδιαίτερα σημαντικό βήμα ενός developer που θέλει να δημιουργήσει το δικό του παιχνίδι. Ενδεικτικά, μερικά από τα κορυφαία εργαλεία είναι το Unity, το Unreal Engine,

το Cry Engine, το Game Maker και άλλα πολλά. Δεν μπορεί κανείς να πει με ευκολία ποια είναι η καλύτερη επιλογή, αφού αυτό εξαρτάται από πολλούς παράγοντες, καθώς και από το ίδιο το παιχνίδι που θέλουμε να δημιουργήσουμε. Είναι χαρακτηριστικό πως αν κοιτάξει κανείς μια λίστα με τους γνωστότερους τίτλους παιχνιδιών θα παρατηρήσει πως το κάθε ένα έχει δημιουργηθεί με διαφορετικό εργαλείο.



Ποιο γνωστές μηχανές ανάπτυξης

Το Unity 3D είναι ένα πολύ ισχυρό εργαλείο για την ανάπτυξη 3D και 2D παιχνιδιών για ένα μεγάλο σύνολο από πλατφόρμες. Το χαρακτηρίζει η ιδιαίτερη φιλικότητα προς το χρήστη, ενώ είναι αρκετά εύκολο στην εκμάθηση για κάποιον αρχάριο και παράλληλα προσφέρει πολλές δυνατότητες σε κάποιον έμπειρο χρήστη. Είναι ενδεικτικό πως υπάρχει μια πληθώρα παιχνιδιών, με γνωστούς τίτλους ανάμεσά τους, που έχουν δημιουργηθεί μέσω του Unity. Ας δούμε τώρα πιο αναλυτικά τα πλεονεκτήματα του Unity:



Unity Logo

-Δημιουργία για πολλές πλατφόρμες: Ένα από τα κύρια πλεονεκτήματα του Unity σε σχέση με άλλα εργαλεία ανάπτυξης παιχνιδιών είναι η δυνατότητα που προσφέρει στον προγραμματιστή να αναπτύξει ένα παιχνίδι για οποιαδήποτε πλατφόρμα επιθυμεί. Ενδεικτικά κάποιες από αυτές τις πλατφόρμες είναι: iOS, Android, Windows Phones, Macs, Windows PCs, Linux, Playstation, Xbox και Wii. Επίσης είναι πολύ σημαντική η ευκολία με την οποία μπορεί κάποιος να μεταφέρει ένα παιχνίδι που δημιουργήθηκε για μια πλατφόρμα, σε μια άλλη, κάτι το οποίο σε άλλη περίπτωση θα χρειαζόταν πολλές ώρες αλλαγών και ρυθμίσεων.

-Υπαρξη μεγάλης και αναπτυσσόμενης κοινότητας χρηστών: Υπάρχουν πάνω από 2 εκατομμύρια developers οι οποίοι χρησιμοποιούν το λογισμικό του Unity. Αυτό έχει ως αποτέλεσμα την ύπαρξη μιας μεγάλης ενεργής κοινότητας, η οποία προσφέρει σημαντική βοήθεια σε χρήστες όλων των επιπέδων. Εδώ αξίζει να αναφέρουμε πως μια ιδιαίτερα σημαντική πηγή βοήθειας, ειδικά για τους νέους χρήστες, είναι μια πληθώρα από tutorials, τα οποία υπάρχουν, είτε από την επίσημη ομάδα του Unity στο <http://unity3d.com/learn> είτε από απλούς χρήστες του λογισμικού.

-Εκπαιδευτικά Σεμινάρια ζωντανά σε απευθείας σύνδεση. Η Unity προσφέρει τη δυνατότητα στους developers της, να παρακολουθήσουν μαθήματα τύπου σεμιναρίων δωρεάν, κατευθείαν από το browser του ηλεκτρονικού τους υπολογιστή. Τα online αυτά μαθήματα εκπονούνται από ειδικούς της Unity και παρουσιάζουν ένα ολοκληρωμένο project δημιουργίας παιχνιδιών στο κοινό, αναλύοντας σε διάφορα topics τις διάφορες ενότητες που απαρτίζουν το εκάστοτε project. Τα μαθήματα αυτά περιλαμβάνουν ολοκληρωμένα παιχνίδια, κομμάτια κώδικα, μέχρι και διάφορα tips για τη δημιουργία ενός παιχνιδιού. Τα περισσότερα από αυτά έχουν να κάνουν με το editor της Unity και το πως μπορεί να αξιοποιηθεί κατά τη διάρκεια ενός παιχνιδιού, όπως για παράδειγμα τα 2D-Scrolling Background, Lights, Camera, Game Objects, Parallax Backgrounds, κ.λ.π.

-Η κοινότητα της Unity με το Forum & Answers. Στη συγκεκριμένη ενότητα του ιστοχώρου της παιχνιδομηχανής υπάρχει ένας ξεχωριστός χώρος, όπου η

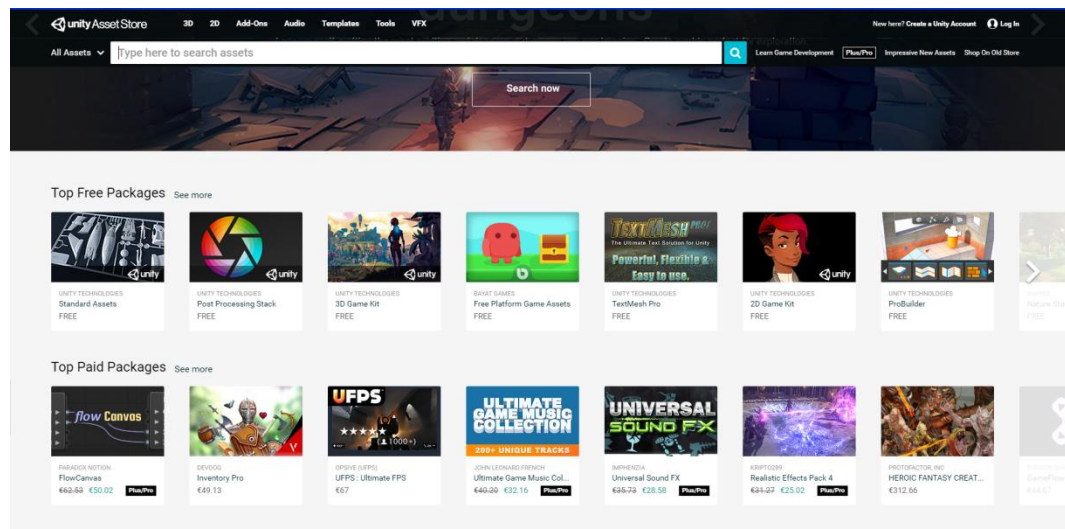
Unity επιτρέπει στον κόσμο της να εκθέσει και να αναλύει τις απορίες του, όποτε υπάρχει ανάγκη. Στην ενότητα Forums λοιπόν, μπορεί κάποιος να ανοίξει μια συζήτηση περί αποριών, ή να ζητήσει συμβουλές από άλλα εγγεγραμμένα μέλη του site σε ότι αφορά το παιχνίδι που επιθυμεί να κατασκευάσει ή κατασκευάζει ήδη. Ιδιαίτερο χώρο επίσης, αποτελεί και η ενότητα Answers. Εκεί το κάθε άτομο που έχει απορία σχετικά με το οτιδήποτε πάνω σε θέμα developing, μπορεί να εκθέσει την ερώτηση του. Ο σκοπός της συγκεκριμένης ενότητας, είναι να αποθηκεύονται μαζικά όλες οι απορίες – ερωτήσεις και εν συνεχεία να είναι ανοιχτές προς το κοινό, είτε για να απαντήσει σε κάποιες από αυτές, είτε να ψάξει και να λύσει τη δικιά του απορία σε μία ήδη απαντημένη ερώτηση.

The screenshot displays the Unity Answers forum interface. At the top, there's a navigation bar with links for Unity, Services, Made with Unity, Learn, and Community. Below that is a search bar and a 'Find posts and topics...' input field. The main content area is divided into two columns. The left column lists several questions with their titles, authors, and timestamps. The right column contains a 'Welcome to Unity Answers' message, a 'Popular Topics' section with tags like 'animation', 'unity 5', and 'javascript', and a 'Recent Badges' section showing user avatars and names like 'Tinoxt' and 'AlienNova'.

Unity – Forum & Answers

-Asset Store: Το Asset Store της Unity, είναι μια online βιβλιοθήκη στην οποία περιλαμβάνονται εμπορικά και ελεύθερα αρχεία παιχνιδιού που δημιουργούνται είτε από την ίδια τη Unity Technologies, είτε από τα μέλη της κοινότητας. Στην ουσία αποτελεί ένα market, στο οποίο μπορεί να εισέλθει ο καθένας και να αποκτήσει είτε δωρεάν, είτε με κάποιο αντίτιμο, όποιο στοιχείο επιθυμεί για να χρησιμοποιήσει για τα παιχνίδια του. Τέτοια στοιχεία είναι, είτε models, αντικείμενα δηλαδή, (άνθρωποι, ζώα, αυτοκίνητα, σύννεφα) σε ψηφιακή μορφή, κομμάτια κώδικα, ακόμα και ολόκληρα παιχνίδια. Τέλος το store παρέχει τη δυνατότητα στα αρχεία που κατεβάζει ο κάθε ενδιαφερόμενος,

να ανοίγουν κατευθείαν στον editor της Unity και να τα κάνει import κατευθείαν στα διάφορα μέρη του παιχνιδιού του.



Unity asset store

-Εύκολη ενσωμάτωση scripts: Το Unity προσφέρει τη δυνατότητα ενσωμάτωσης scripts γραμμένα σε C# ή Javascript τα οποία μπορούν εύκολα να αντιστοιχηθούν με συγκεκριμένα αντικείμενα δίνοντάς τους ξεχωριστές ιδιότητες.

-Είναι δωρεάν: Στο Unity 3D υπάρχει μια Free version και μια Pro version. Φυσικά η Pro version περιέχει κάποια extra χαρακτηριστικά. Όμως, σε αντίθεση με άλλα λογισμικά, η Free version προσφέρει ένα πλήρως λειτουργικό εργαλείο για την ανάπτυξη ενός παιχνιδιού. Σε σχέση με την κανονική έκδοση της Unity, η pro έκδοση δεν είναι δωρεάν. Η Unity Pro περιλαμβάνει την ευκολία και την απλή χρήση της κανονικής έκδοσης, αλλά περιέχει επιπλέον μία πλήρη γκάμα ισχυρών δυνατοτήτων, οι οποίες αφήνουν ικανοποιημένο κάθε 3D - προγραμματιστή και εταιρεία. Η διαφορά με την κανονική έκδοση της Unity βρίσκεται στο ότι με την Pro μπορούν να δημιουργηθούν πιο σύνθετα, πιο εξελιγμένα και πιο όμορφα σε γραφικό τομέα παιχνίδια. Επίσης, οι δυνατότητες και τα χαρακτηριστικά των παιχνιδιών με αυτήν την έκδοση για τις κονσόλες και

κυρίως για τα κινητά τηλέφωνα αυξάνονται. Οι διαφορές μεταξύ της ελεύθερης έκδοσης και της Pro έκδοσης της Unity φαίνονται στο πίνακα παρακάτω.

UNITY		PERSONAL EDITION	PROFESSIONAL EDITION
What's included			
Engine with all features	?	✓	✓
Royalty-free	?	✓	✓
All platforms (limitations apply)	?	✓	✓
Beta access	?	✓	✓
Customizable splash screen	?	✗	✓
Unity Cloud Build Pro - 12 Months	?	✗	✓
Unity Analytics Pro	?	✗	✓
Team License	?	✗	✓
Prioritized bug handling	?	✗	✓
Game Performance Reporting	?	✗	✓
Unlimited Revenue and Funding	?	✗	✓
Future platforms included	?	✗	✓
Professional editor skin	?	✗	✓
Asset Store Level 11	?	✗	✓
Professional Community Features	?	✗	✓
Source code access	?	✗	?
Premium Support	?	?	?
		FREE DOWNLOAD	FROM \$75/MONTH

Διαφορές personal & pro version

Κεφάλαιο 3ο : Unity 3d Game Engine

3.1 Πρόλογος

Είναι για πολλούς το καλύτερο Game Engine διότι είναι μια ολοκληρωμένη μηχανή ανάπτυξης παιχνιδιών, έχει ένα μεγάλο Forum και ένα Fan Club υποστηρικτών όπου μπορούν να σε βοηθήσουν πολύ γρήγορα. Ακόμα είναι η

πιο διαδεδομένη μηχανή ανάπτυξης σε ότι αφορά δημιουργία παιχνιδιών σε Smart phones & Tablet, έχει μέτρια προς καλά γραφικά και το ατού είναι τα Script που τρέχουν πιο γρήγορα από άλλες μηχανές.

3.2 Ιστορική Αναδρομή

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

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



Unity πλατφόρμες

3.3 Το εργαλείο σήμερα

Το Unity σήμερα βρίσκεται σήμερα στην έκδοση 2018.2.7 και οι Developers που χρησιμοποιούν το Unity φτάνουν το 45%.



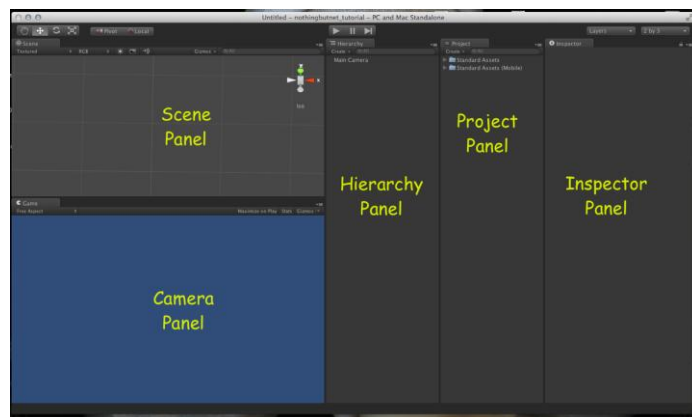
Ποσοστό χρηστών

3.4 Unity Editor

3.4.1 Τα παράθυρα του Editor

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

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



Unity Panel

•Scene View

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

Πλήκτρο Q(Navigation Tool):



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

Πλήκτρο W(Translate Tool):



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

Πλήκτρο E(Rotation Tool):



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

Πλήκτρο R(Scale Tool):



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

Rendering Options

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

•Game View

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

Play :



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

Pause : 

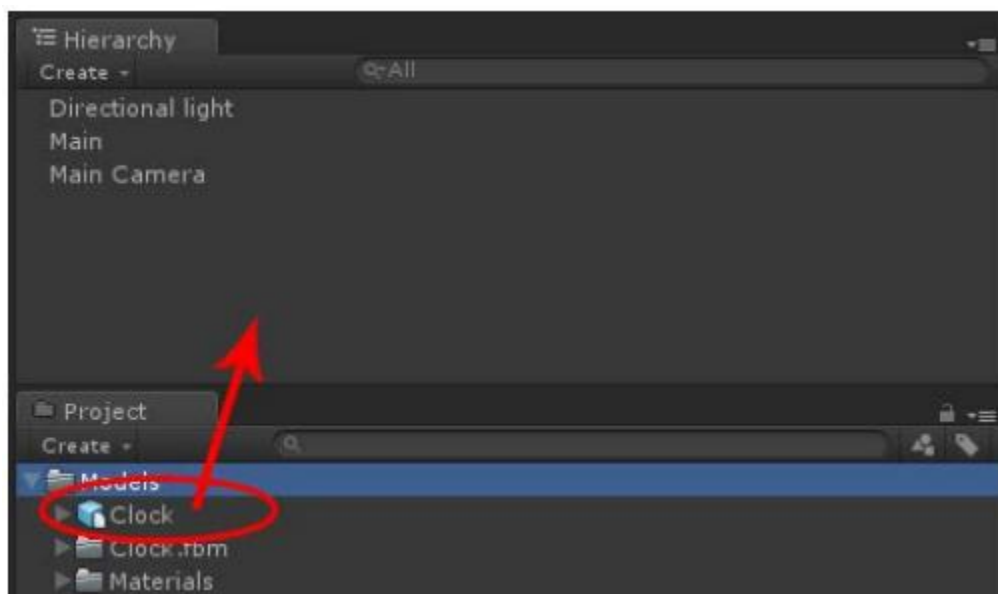
Με το κουμπί αυτό παγώνουμε το παιχνίδι σε ένα συγκεκριμένο Frame ώστε να μπορέσουμε να διακρίνουμε μια λεπτομέρεια, καθώς και να επεξεργαστούμε τιμές που επηρεάζουν την λειτουργία του (ταχύτητα χαρακτήρα-βαρύτητα-φωτισμός κ.α)

Next Frame : 

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

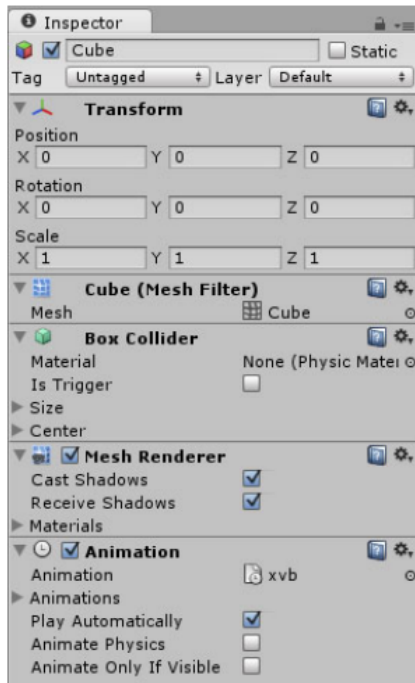
•Hierarchy and Project

Το Hierarchy είναι ένα παράθυρο όπου περιέχει όλα τα αντικείμενα που βρίσκονται στην σκηνή με τα ονόματά τους και από εκεί μπορούμε να αλλάξουμε τις τελικές ρυθμίσεις του, με λίγα λόγια το παιχνίδι μας παίρνει πληροφορίες από τον συγκεκριμένο φάκελο. Το παράθυρο Project από την άλλη περιέχει όλα τα Asset του παιχνιδιού με απλά λόγια οτιδήποτε θα χρειαστούμε στο παιχνίδι είναι εκεί (textures - scripts - ήχοι - prefabs κ.α)



Εισαγωγή Prefab από το Project στο Hierarchy

•Inspector



inspector

Στον Inspector εμφανίζονται όλα τα χαρακτηριστικά των αντικειμένων μας όλου του παιχνιδιού είτε αυτά βρίσκονται στο Hierarchy είτε στο Project. Από αυτό το παράθυρο μπορούμε να επεξεργαστούμε τα χαρακτηριστικά του ανάλογα με το είδος του. Βασικά χαρακτηριστικά είναι η θέση - position, η περιστροφή-rotation αλλά και το μέγεθος scale. Επιπλέον μπορούμε να του εισάγουμε νέα χαρακτηριστικά στο αντικείμενο μας όπως είναι ένα ήχος, φωτισμός, βαρύτητα, animations, scripts και άλλα που επηρεάζουν στην αντίδραση του Object μας.

3.4.2 Η Μπάρα Μενού

Η μπάρα μενού περιέχει στην ουσία όλες τις επιλογές για το περιβάλλον ανάπτυξης παιχνιδιού στον Editor της Unity. Η μπάρα αυτή περιέχει τις εξής καρτέλες:

- File
- Edit
- Assets
- Game Object
- Component
- Window
- Help

• File

Από την καρτέλα File στην ουσία ξεκινάει το project του παιχνιδιού. Εδώ δημιουργείται ένα καινούριο project, οι σκηνές του παιχνιδιού, οι επιλογές πλατφόρμας που θα κυκλοφορήσει το οποιοδήποτε παιχνίδι και φυσικά η καρτέλα αυτή παρέχει τη δυνατότητα να αποθηκευθεί το παιχνίδι κάθε φορά που κλείνει ο Editor για να συνεχιστεί η κατασκευή του παιχνιδιού όταν ξανανοίξει.

• Edit

Η καρτέλα Edit δίνει τη δυνατότητα στο χρήστη να αξιοποιεί διάφορες εντολές πληκτρολογίου (copy, paste, duplicate, κλπ.) και επιλογές για τις προτιμήσεις γενικά του Project και του Editor.

• Assets

Η καρτέλα Asset βρίσκεται στη μπάρα του μενού και στο παράθυρο Project του "Editor" της Unity. Αυτή παρέχει την δυνατότητα εισαγωγής καινούριων στοιχείων στο project καθώς και την εισαγωγή πακέτων με στοιχεία τα οποία παρέχει η ίδια η Unity γ, όπως για παράδειγμα νερό, δέντρα, φώτα κλπ.

• Game Object

Η καρτέλα GameObject είναι η κύρια βιβλιοθήκη αντικειμένων που μπορεί να χρησιμοποιήσει ο κάθε developer. Στην ουσία από εδώ εισάγονται στο παιχνίδι αντικείμενα όπως:

- Particle System: όπως υπονοεί και το ίδιο το όνομα, πρόκειται για ένα σύστημα σωματιδίων που ανήκει στην κατηγορία των γραφικών και χρησιμοποιείται σε ένα παιχνίδι όταν σχετίζονται σε αυτό φυσικά φαινόμενα

όπως φωτιά, εκρήξεις, νερό, καπνός, αύρες χαρακτήρων, μαλλιά, κίνηση κλπ. Είναι είτε δισδιάστατα είτε τρισδιάστατα αντικείμενα.

- Camera: ένα αντικείμενο που χρησιμοποιείται για να μπορέσει ο χρήστης να δει τον εικονικό κόσμο.

- GUI Text / GUI Texture/ 3D Text: αντικείμενα που εμφανίζουν το επιθυμητό κείμενο όταν καλούνται.

- Direction Light/ Point Light/ Spot Light/ Area Light: αντικείμενα που σκοπός τους είναι να φωτίζουν άλλα αντικείμενα, ένα χώρο ή σημείο ή και ολόκληρη τη σκηνή του παιχνιδιού.

- Cube/ Sphere/ Capsule/ Cylinder/ Plane/ Quad: αποτελούν τρισδιάστατα αντικείμενα με τα οποία μπορεί να αλληλεπιδράσει ο χρήστης τοποθετώντας τα μέσα στις σκηνές του παιχνιδιού, αλλάζοντάς τους διαστάσεις, ονόματα, ιδιότητες, κλπ.

- Sprites: πρόκειται για δισδιάστατα (2D) αντικείμενα γραφικών που χρησιμοποιούνται για χαρακτήρες και άλλα αντικείμενα κυρίως δισδιάστατων παιχνιδιών.

- Cloth: Χρησιμοποιούνται για την εμφάνιση γραφικών πάνω σε αντικείμενα, όπως ρούχα, σημαίες κλπ.

-Audio Reverb Zone: Το αντικείμενο αυτό σχετίζεται με ήχους τους παιχνιδιού και προσφέρει τη δυνατότητα να υπάρχουν εφέ όπως ηχώ κλπ

-Terrain/ Ragdoll/ Tree/ Wind Zone: Αντικείμενα για εδάφη, μηχανισμούς κίνησης χαρακτήρων, δέντρα και ζώνες με αέρα για να αλληλεπιδρούν με τα υπόλοιπα αντικείμενα στη σκηνή.

•Component

Η καρτέλα του Μενού Component περιέχει μια γκάμα στοιχείων τα οποία προστίθενται στα διάφορα αντικείμενα του παιχνιδιού. Τα στοιχεία αυτά, μαζί με τις ιδιότητές τους είναι τα εξής:

- Mesh: Είναι το πλέγμα που αποτελείται από πολλά τρίγωνα τοποθετημένα σε τρισδιάστατο χώρο και ρόλος τους είναι να δίνουν μια τρισδιάστατη στερεή όψη στα αντικείμενα που θέλουμε να χρησιμοποιήσουμε.

- Effects: Είναι τα εφέ που προστίθενται σε διάφορα αντικείμενα και μπορεί να είναι είτε συστήματα σωματιδίων (Particle Effects), είτε γραμμές που αφήνουν τα αντικείμενα κατά την κίνησή τους, για να δώσουν έναν πιο ρεαλιστικό τόνο στο παιχνίδι.

-Physics/ Physics 2D: Είναι οι ιδιότητες που παίρνει ένα αντικείμενο για να θεωρηθεί φυσικό σώμα και να μπορεί να κινείται και να αλληλεπιδρά με άλλα αντικείμενα του χώρου. Χωρίς αυτά οι χαρακτήρες του παιχνιδιού είναι στην ουσία άυλοι και δε μπορούν να υπάρξουν συγκρούσεις, καταστροφές και άλλα φαινόμενα στο παιχνίδι. Τα colliders είναι οι ιδιότητες που μετατρέπουν ένα αντικείμενο σε υλικό για να μπορεί αυτό να συγκρούεται με άλλα αντικείμενα με την ίδια ιδιότητα. Η ιδιότητα Rigidbody προσφέρει σε ένα αντικείμενο κίνηση μέσα στο χώρο, βαρύτητα και σύγκρουση με άλλα αντικείμενα. (Οι ιδιότητες της κατηγορίας Physics 2 σχετίζονται με δισδιάστατα στοιχεία του παιχνιδιού και μόνο.

-Navigation: Ο ρόλος του συστήματος αυτού είναι για να δείξει στον χρήστη ποια μέρη/ χώροι στον κόσμο του παιχνιδιού είναι πιο προσβάσιμα καθώς κινείται ο κύριος χαρακτήρας του παιχνιδιού για να μην υπάρξουν προβλήματα γραφικών και κίνησης.

-Audio: Παρέχει τη δυνατότητα σε διάφορα αντικείμενα του παιχνιδιού να αναπαραγάγουν αρχεία μουσικής που εισάγουμε στο παιχνίδι.

-Rendering: Η επιλογή αυτή προσθέτει στοιχεία όπως Camera, κείμενα και φώτα στο αντικείμενο που θέλουμε.

- Miscellaneous: Το κύριο μέρος αυτού του component είναι ότι παρέχει τη δυνατότητα στα αντικείμενα να αναπαραγάγουν animations που έχουν ήδη φτιαχτεί για οποιοδήποτε σκοπό.

-Scripts: Αυτή η επιλογή επιτρέπει την προσθήκη ενός script κώδικα σε κάποιο αντικείμενο με αποτέλεσμα αυτό να αποκτήσει κι άλλες ιδιότητες που δε μπορούμε να βάλουμε μέσω του editor. Η καρτέλα Window περιλαμβάνει ως επιλογές όλα τα παράθυρα του editor που αναφέρθηκαν στην προηγούμενη υποενότητα. Hierarchy , Scene , Inspector.

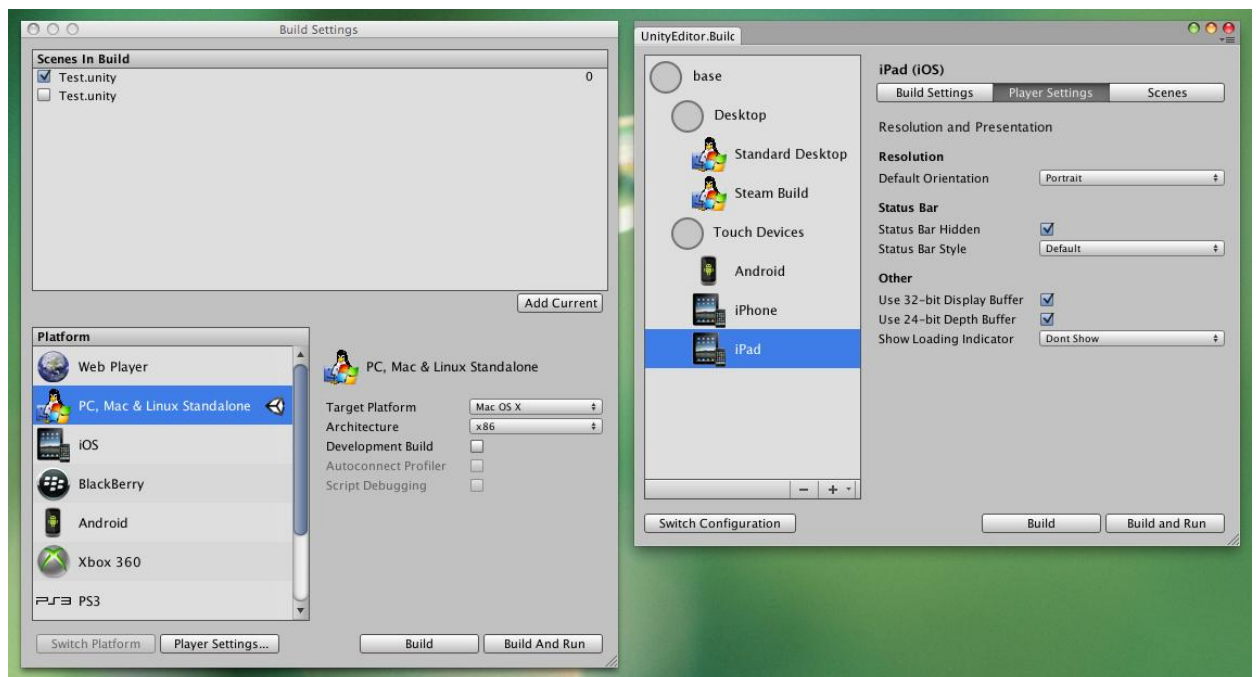
3.4.3 Build Settings

Στην καρτέλα File του μενού, βρίσκεται η επιλογή Build Settings. Επιλέγεται μετά το τέλος ανάπτυξης κάθε παιχνιδιού, με σκοπό να αλλάξουμε κάποιες γενικές ρυθμίσεις και αποτελεί το τελευταίο βήμα πριν την κυκλοφορία του παιχνιδιού σε κάποια πλατφόρμα.

Στο παράθυρο των Build Settings υπάρχει ένας μικρός χώρος στον οποίο μπαίνουν τα Scenes του παιχνιδιού είτε σέρνοντας με το ποντίκι τις σκηνές από το παράθυρο project σε αυτό, είτε πατώντας το κουμπί Add Current με τη σειρά που είναι επιθυμητή. Αυτό σημαίνει πως το παιχνίδι που θα εξαχθεί από

τη Unity αποτελείται από τα Scenes που τοποθετήθηκαν στα Build Settings. Επίσης, εκεί ο developer επιλέγει ποια κονσόλα επιθυμεί να «υιοθετήσει» το παιχνίδι του. Οι επιλογές είναι:

- Web Player
- Pc, Mac & Linux Standalone
- iOS
- Android
- BlackBerry
- Windows Store
- Windows Phone 8
- Xbox 360
- Xbox One
- PS3
- PS Vita
- PS4
- Playstation Mobile



Build Settings

Πατώντας την επιλογή Player Settings στο παράθυρο Build Settings, ανοίγει ένα άλλο παράθυρο στον Inspector το οποίο προσφέρει τη δυνατότητα να υιοθετηθεί ένα όνομα για το συγκεκριμένο παιχνίδι και για την εταιρεία παραγωγής του, καθώς και άλλες λεπτομέρειες ανάλογα με την επιλεγμένη πλατφόρμα και την έκδοση της Unity.

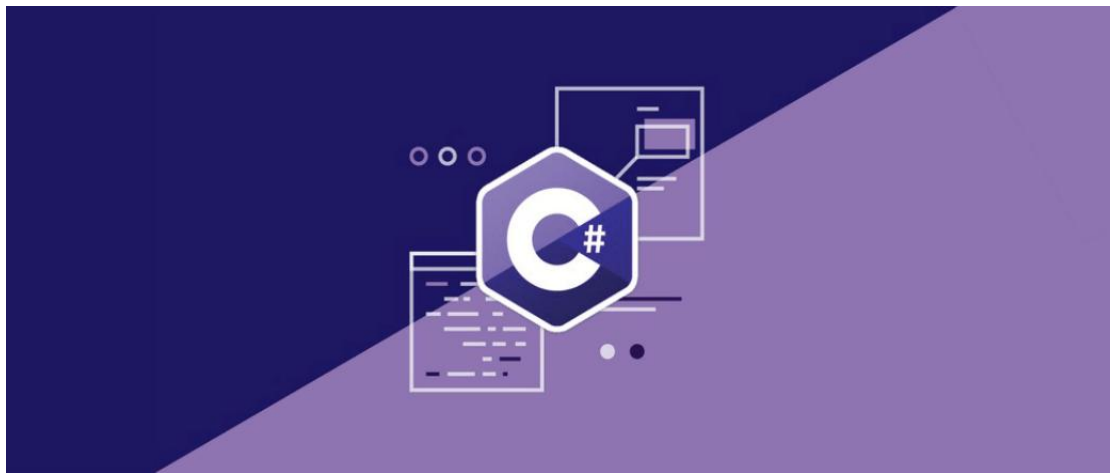


Player Settings

Κεφάλαιο 4ο Η δημιουργία της C#

4.1 Χαρακτηριστικά της C που κληρονόμησε η C#

1. Σύνταξη
2. Λέξεις κλειδιά
3. Τελεστές



C#

Σχέση Java και C#

1. μεταφερότος κώδικας
2. εκτέλεση προγραμμάτων σε ασφαλές περιβάλλον

Σχέση της C# με το .NET Framework

1. Το .NET Framework είναι το περιβάλλον χρόνου εκτέλεσης της C#.
2. Οι βιβλιοθήκες της γλώσσας ορίζονται από το .NET Framework. 25

4.2 Τι είναι το .NET Framework

Είναι ένα περιβάλλον που υποστηρίζει την ανάπτυξη και την εκτέλεση άκρως καταναμημένων εφαρμογών που βασίζονται σε συστατικά. Επιτρέπει στις γλώσσες προγραμματισμού να συνεργάζονται και παρέχει χαρακτηριστικά ασφαλείας, μεταφερτότητας και ένα κοινό μοντέλο προγραμματισμού για τα Windows. Παρέχει τα εξής:

1. Σύστημα Common Language Runtime: διαχειρίζεται την εκτέλεση ενός προγράμματος σε C#.
2. Βιβλιοθήκη Κλάσεων: παρέχει στα προγράμματα δυνατότητες πρόσβασης στο περιβάλλον χρόνου εκτέλεσης.



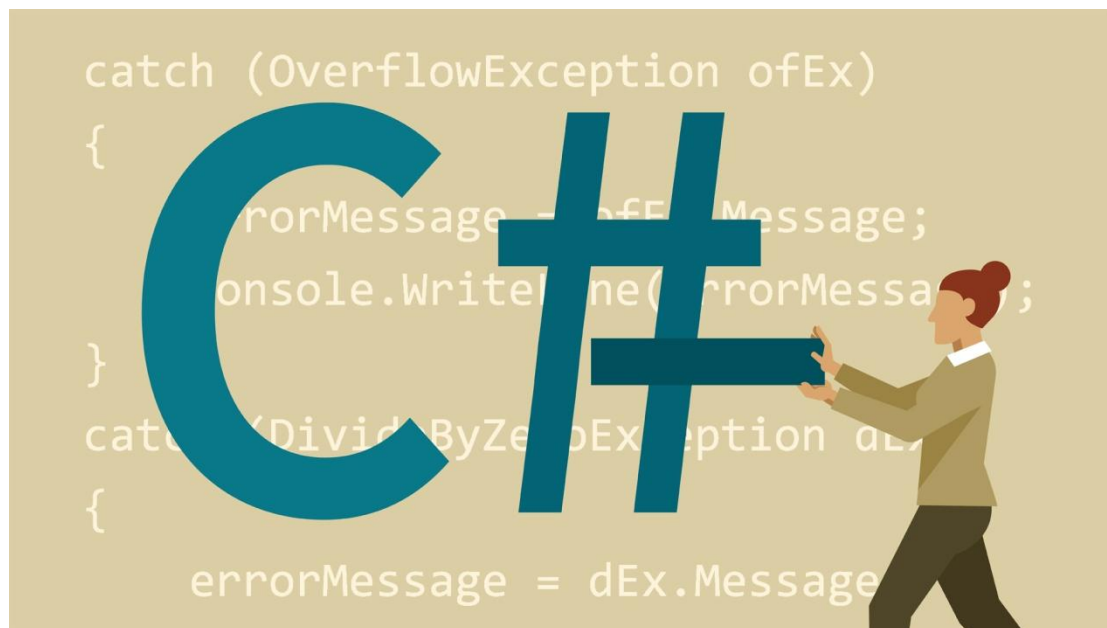
.Net Framework

4.3 Αντικειμενοστραφής Προγραμματισμός

Τα αντικειμενοστραφή προγράμματα οργανώνονται γύρω από τα δεδομένα, δηλαδή όταν προγραμματίζουμε στην ουσία ορίζουμε τα δεδομένα και τις ρουτίνες που επιτρέπεται να δρουν επί αυτών των δεδομένων. Έτσι ένας τύπος δεδομένων ορίζει τι ακριβώς τύπου λειτουργίες μπορούν να εφαρμοστούν επί αυτών των δεδομένων.

Χαρακτηριστικά

1. ενθυλάκωση
2. πολυμορφισμός
3. κληρονομικότητα



C#

4.4 Ενθυλάκωση

Είναι ένας μηχανισμός προγραμματισμού που συνδέει το κώδικα με τα δεδομένα που χειρίζεται και τα κρατά και τα δύο ασφαλή από εξωτερικές παρεμβολές και κακομεταχείριση. Ο κώδικας και τα δεδομένα συνδέονται κατά τέτοιο τρόπο ώστε να δημιουργείται ένα "μαύρο κουτί". Μέσα σε αυτό βρίσκονται όλα τα απαραίτητα δεδομένα και ο απαραίτητος κώδικας. Έτσι δημιουργείται ένα αντικείμενο. Μέσα σε αυτό ο κώδικας και τα δεδομένα μπορεί να είναι ιδιωτικά (private) - που σημαίνει ότι μπορούν να προσπελαστούν μόνο από άλλα μέρη του αντικειμένου ή δημόσια (public) - που σημαίνει ότι μπορούν να προσπελαστούν και από τμήμα του προγράμματος που βρίσκεται έξω από το αντικείμενο. Η βασική μονάδα ενθυλάκωσης είναι η κλάση. Μια κλάση ορίζει τη μορφή θα έχει το αντικείμενο της, δηλαδή καθορίζει τα δεδομένα και το κώδικα που δρα επί αυτών των δεδομένων. Τόσο τα δεδομένα όσο και ο κώδικας ονομάζονται μέλη της κλάσης. Κατά αντιστοιχία, τα δεδομένα ονομάζονται μεταβλητές μέλους ενώ, ο κώδικας μέθοδοι μέλους.

4.5 Πολυμορφισμός

Είναι το χαρακτηριστικό που επιτρέπει το ίδιο όνομα μεθόδου να προκαλεί την εκτέλεση διαφορετικού κώδικα ανάλογα με τον τύπο του αντικειμένου στο οποίο καλείται

Πλεονέκτημα : Μείωση πολυπλοκότητας κώδικα

4.6 Κληρονομικότητα

Είναι η διαδικασία κατά την οποία ένα αντικείμενο μπορεί να πάρει τις ιδιότητες ενός άλλου αντικειμένου.

Πλεονεκτήματα :

1. ιεραρχική δομή κλάσεων
2. κάθε αντικείμενο πρέπει να ορίζει μόνο εκείνες τις ιδιότητες που το κάνουν μοναδικό μέσα στη κλάση του και όχι αυτές που κληρονομεί

4.7 Γενικά περί Κλάσεων

Κλάση είναι ένα πρότυπο που ορίζει τη μορφή ενός αντικειμένου. Καθορίζει τα δεδομένα και το κώδικα που δρα πάνω σε αυτά τα δεδομένα. Τα αντικείμενα είναι στιγμιότυπα της κλάσης στην οποία ανήκουν. Ότι ανήκει μέσα σε μία κλάση αποτελεί μέλος της κλάσης. Οι περισσότερες κλάσεις περιέχουν κώδικα και δεδομένα. Γενικότερα μια κλάση μπορεί να έχει ως μέλη τα εξής: μεταβλητές στιγμιότυπου, στατικές μεταβλητές, σταθερές, μεθόδους, κατασκευαστές, καταστροφείς, δεικτοδότες, συμβάντα, τελεστές και ιδιότητες. Ένας απαραίτητος μηχανισμός για όλες τις κλάσεις είναι ο καθοριστής προσπέλασης μέλους, ο οποίος καθορίζει ποιος μπορεί να χρησιμοποιήσει τη μεταβλητή ή τη μέθοδο μιας κλάσης. Υπάρχουν 4 καθοριστές προσπέλασης:

1. `public` (δημόσια): το μέλος μπορεί να προσπελαύνεται από κάθε κώδικα που βρίσκεται μέσα σε ένα πρόγραμμα, φυσικά και εκτός της κλάσης.
2. `private` (ιδιωτική): το μέλος μπορεί να προσπελαστεί μόνο από άλλα μέλη της κλάσης στην οποία βρίσκεται.
3. `protected` (προστατευμένη)
4. `internal` (εσωτερική)

Επίσης, οι μέθοδοι είναι ένα σημαντικό ατού της αντικειμενοστρέφειας. Οι μέθοδοι κλάσης είναι υπορουτίνες που χειρίζονται τα δεδομένα της και σε πολλές περιπτώσεις παρέχουν πρόσβαση σε αυτά. Ουσιαστικά, άλλα μέρη του προγράμματος θα αλληλεπιδρούν με μία κλάση μέσω των μεθόδων της. Τέλος, μια μέθοδος υλοποιεί μια συγκεκριμένη λειτουργία, μπορεί να χρειάζεται ορίσματα για να δουλέψει και μετά το πέρας λειτουργίας της, να επιστρέφει κάτι σε αυτόν που τη κάλεσε. Ακόμη, ο κατασκευαστής είναι μία μέθοδος, η οποία αρχικοποιεί ένα αντικείμενο της κλάσης όταν αυτό δημιουργείται, δηλαδή δίνει αρχικές τιμές στα δεδομένα του. Έχει το ίδιο όνομα με τη κλάση αλλά δεν έχουν ρητό επιστρεφόμενο τύπο. Συνήθως η προσπέλαση είναι `public` για να μπορεί να κληθεί εκτός της κλάσης και μπορεί να περιέχει ή όχι ορίσματα. Όλες οι κλάσεις έχουν κατασκευαστές χωρίς ορίσματα, ακόμα και αν δε δηλώσουμε ρητά εμείς κάποιον (σε αυτή τη περίπτωση το κάνει ο μεταγλωττιστής και αρχικοποιεί όλες τις μεταβλητές-μέλη με τις προεπιλεγμένες τιμές τους). Επιπλέον, μια καθοριστική μέθοδος μπορεί να θεωρηθεί και ο καταστροφείας. Είναι μια μέθοδος-μέλος, η οποία απελευθερώνει τη μνήμη που

δεσμεύθηκε για τη δημιουργία ενός αντικειμένου (από το τελεστή new). Στην ουσία καταστρέφει το αντικείμενο. Ο καταστροφέας καλείται πριν από τη λεγόμενη συλλογή απορριμμάτων, κάτι που γίνεται στο παρασκήνιο (χωρίς τη συμμετοχή του προγραμματιστή), έτσι ώστε να απελευθερωθεί η μνήμη.

Κεφάλαιο 5^ο Παράδειγμα

5.1 Γενική λογική C# στην Unity

Η Unity μπορεί να χρησιμοποιήσει ένα σύνολο script, τα οποία αλληλεπιδρούν μεταξύ τους αλλάζοντας μεταβλητές. Κατά τη δημιουργία ενός script, μετά τη δήλωση της κλάσης, η οποία πάντα πρέπει να έχει το ίδιο όνομα με το όνομα του script (και γίνεται αυτόματα), δηλώνονται όλες τις μεταβλητές που θέλουμε να χρησιμοποιήσουμε για το αντικείμενο στο οποίο θα προσκολλήσουμε το script. Ένα script μπορεί να περιέχει πολλές κλάσεις, καθώς και υποκλάσεις οι οποίες κληρονομούν ιδιότητες από άλλες κλάσεις, άλλων, του ίδιου script ή του πηγαίου κώδικα της Unity.

Σε κάθε script που δημιουργούμε υπάρχουν αρχικά δύο default συναρτήσεις, οι οποίες χρησιμοποιούνται για τις περισσότερες λειτουργίες των αντικειμένων. Η Start και η Update. Ο κώδικας που εμπεριέχεται στην Start εκτελείται μόνο μία φορά, όταν αρχίζει το παιχνίδι (ή όταν εμφανίζεται το αντικείμενο που έχει το script στο παιχνίδι, αν δεν υπάρχει το αντικείμενο από την αρχή στην σκηνή). Συνήθως στην Start, πέρα άλλων λειτουργιών που μπορεί να χρειαστεί να εκτελεστούν μια φορά μόνο, αρχικοποιούνται οι μεταβλητές που έχουμε δηλώσει, ή αρχικοποιούνται (αντιστοιχούνται) με ένα αντικείμενο ή στοιχείο αντικειμένου που υπάρχει ήδη στην σκηνή. Η Update τις περισσότερες φορές περιέχει την πλειοψηφία του κώδικα ενός script και τον εκτελεί σε κάθε frame (καρέ), δηλαδή πολλές φορές το δευτερόλεπτο, ανάλογες με το frame rate, κατά την διάρκεια του παιχνιδιού. Για αυτό το λόγο στην Update προσθέτουμε λειτουργίες με παραμέτρους που θέλουμε να αλλάζουν σε πραγματικό χρόνο όσο λειτουργεί το παιχνίδι. Για παράδειγμα, οι κινήσεις και οι αλληλεπιδράσεις των αντικειμένων. Ένα loop που αλλάζει ένα στοιχείο, π.χ. ένα while που αλλάζει τις συνθήκες σε έναν animator ενός αντικειμένου, μέσα στην Update είναι αρκετό για να “κρασάρει” την Unity, εφόσον ακόμη και αν δεν είναι “αέναο”.

5.2 Βασικός τρόπος δήλωσης και αρχικοποίησης των μεταβλητών

Όπως αναφέρθηκε νωρίτερα, το πρώτο πράγμα που κάνει ο χρήστης μετά την δημιουργία ενός script είναι η δήλωση των μεταβλητών που θα χρησιμοποιηθούν (η οποία φυσικά μπορεί να συμπληρωθεί και αργότερα αν ο χρήστης κρίνει ότι χρειάζεται περισσότερους μεταβλητές). Πέρα από τους κλασικούς τύπους μεταβλητών (int, float, bool κλπ) υπάρχουν και τύποι (κλάσεις) μεταβλητών της βιβλιοθήκης της Unity. Έστω ότι δηλώνουμε μια μεταβλητή στην οποία θέλουμε να αποθηκεύσουμε ένα αντικείμενο, το οποίο εκπροσωπεί τον παίκτη μας. Στην θέση του τύπου μεταβλητής, συμπληρώνουμε τον όρο τον οποίο η Unity χρησιμοποιεί για να αναφερθεί στα αντικείμενα (GameObject).

5.3 Η δήλωση του θα γίνει ως εξής:

- **public GameObject knight;**

Έστω ότι δηλώνουμε και μια άλλη μεταβλητή στην οποία θέλουμε να αποθηκεύσουμε ένα στοιχείο του παίκτη μας – στην συγκεκριμένη περίπτωση το Transform του παίκτη. Ο τύπος δήλωσης θα είναι το όνομα του στοιχείου. Οπότε η δήλωση του θα γίνει ως εξής.

- **public Transform knightLocation;**

Με αυτές τις δηλώσεις η Unity καταλαβαίνει ότι θέλουμε να δημιουργήσουμε μια μεταβλητή τύπου GameObject και μια μεταβλητή τύπου Transform με τα αντίστοιχα ονόματα που τους δώσαμε, ωστόσο, εφόσον δεν τις έχουμε αρχικοποιήσει στη Start, δεν την έχει αντιστοιχίσει με τον παίκτη μας ή το στοιχείο του παίκτη μας. Για να αντιστοιχιστούν οι μεταβλητές μας με το αντικείμενο που εκπροσωπεί τον παίκτη και το στοιχείο του παίκτη Transform, πρέπει να τις αρχικοποιήσουμε στην Start χρησιμοποιώντας συναρτήσεις από τις βιβλιοθήκες της Unity. Για να αντιστοιχίσουμε το αντικείμενο που εκπροσωπεί

τον παίκτη με την μεταβλητή `knight` που φτιάξαμε νωρίτερα θα χρησιμοποιήσουμε την εξής συνάρτηση `GameObject.Find()`, η οποία βρίσκει το αντικείμενο στη σκηνή χρησιμοποιώντας το όνομα του. Έστω ότι το όνομα του στη σκηνή (ιεραρχία) είναι `Player`.

- `knight = GameObject.Find("Player");`

Τώρα που αντιστοιχίστηκε η μεταβλητή με το αντικείμενο που εκπροσωπεί τον παίκτη, πρέπει να αντιστοιχιστεί και το στοιχείο του αντικειμένου αυτού με την δεύτερη μεταβλητή. Θα χρησιμοποιηθεί μια άλλη συνάρτηση – `GetComponent()`, η οποία αντλεί στοιχεία από τα αντικείμενα τους.

- `knightLocation = knight.GetComponent();`

-Το κομμάτι `"knight."` στην παραπάνω εντολή δεν είναι απαραίτητο εφόσον το `script` στο οποίο είναι η εντολή, είναι προσκολλημένο στο συγκεκριμένο αντικείμενο που περιέχει το στοιχείο.

5.4 Βιβλιοθήκες της Unity:

Η Unity διαθέτει κάποιες βιβλιοθήκες με έτοιμες συναρτήσεις/λειτουργίες που μπορούμε να χρησιμοποιήσουμε στον κώδικα. Όταν ο χρήστης θέλει να πετύχει κάτι συγκεκριμένο μέσω κώδικα, δεν έχει παρά να ανατρέξει στο `manual` της Unity (<https://docs.unity3d.com/Manual/index.html>) όπου παρέχονται όλες η συναρτήσεις με παραδείγματα. Κάποια παραδείγματα συναρτήσεων/λειτουργιών που χρησιμοποιήθηκαν πολύ στην παρούσα πτυχιακή είναι:

- `knight = GameObject.FindWithTag("Player");`

- Εντοπίζει το αντικείμενο με το `tag "Player"` και το αντιστοιχεί στην μεταβλητή `"knight"` που έχουμε δηλώσει στην αρχικοποίηση.

- **mycollider = GetComponent();**

- Εντοπίζει το στοιχείο του αντικειμένου (στην συγκεκριμένη περίπτωση το collider του), στο οποίο βρίσκεται το script που την περιέχει και το αποθηκεύει σε μια μεταβλητή. Οποιοδήποτε στοιχείο μπορεί να αποθηκευτεί (π.χ. ένα άλλο script, ώστε μετά να μπορούμε να αλλάξουμε τις μεταβλητές του από το παρόν script – περισσότερο στην “ανάλυση δειγμάτων κώδικα”).

- **void OnTriggerEnter3D(Collider3D other)**

- Μια συνάρτηση η οποία εκτελείται μόνο την στιγμή που ένας άλλος collider ενός αντικειμένου στην σκηνή, έρθει σε επαφή με τον collider του αντικειμένου που έχει το script με αυτή.

- **void OnTriggerStay3D(Collider3D other)**

- Σαν την παραπάνω συνάρτηση, που ωστόσο εκτελείται συνεχώς όσο οι colliders των αντικειμένων βρίσκονται σε επαφή. Επίσης σημειώνεται ότι για να αναγνωριστούν συγκεκριμένες συναρτήσεις π.χ. για κατασκευή UI elements, πρέπει να προσθέσουμε την κατάλληλη βιβλιοθήκη στην αρχή του script.

5.5 Βασική επικοινωνία μεταξύ scripts

Όπως αναφέρθηκε νωρίτερα, όταν θέλουμε να δηλώσουμε μια μεταβλητή για ένα στοιχείο, χρησιμοποιούμε τον τύπο του στοιχείου. Για τα scripts χρησιμοποιούμε το όνομα που τους δώσαμε. Έστω ότι έχουμε έναν παίκτη που ονομάζεται example στην σκηνή - του έχουμε προσθέσει collider, rigidbody

για μάζα και ένα script για την ζωή του εν ονόματι "Health", το οποίο περιέχει μια μεταβλητή τύπου int που ονομάζεται currentHealth και έχει αρχικοποιηθεί με το νούμερο 100. Τώρα, έστω ότι θέλουμε να φτιάξουμε μια παγίδα που να κάνει ζημιά στον παίκτη όταν την πατάει. Σέρνουμε το γραφικό της παγίδας στη σκηνή, προσθέτουμε collider που μετατρέπουμε σε trigger collider από τις επιλογές του, δημιουργούμε ένα script που ονομάζουμε "Trap" και το προσθέτουμε στο αντικείμενο της παγίδας στην ιεραρχία, μέσω του Inspector. Εφόσον θέλουμε η παγίδα να κάνει ζημιά στην ζωή του παίκτη, ανοίγουμε το script Trap και δηλώνουμε μεταβλητή που θα εκπροσωπήσει το script Health.

- **public Health life;**

Για να βρούμε το στοιχείο/script Health στην σκηνή όμως , πρέπει πρώτα να βρούμε το αντικείμενο στο οποίο βρίσκεται, δηλαδή τον παίκτη και να τον αποθηκεύσουμε σε μια άλλη μεταβλητή (αφού πρώτα τη δηλώσουμε τη μεταβλητή ως GameObject. Θα τον βρούμε βάσει του ονόματος που του είχαμε δώσει. Στην Start γράφουμε:

- **player = GameObject.Find("example");**

Τώρα, αφού έχουμε βρει το αντικείμενο του παίκτη, πρέπει να πάρουμε το στοιχείο του αντικειμένου, το Health script και να το αντιστοιχίσουμε με την μεταβλητή life.

- **life = example.GetComponent();**

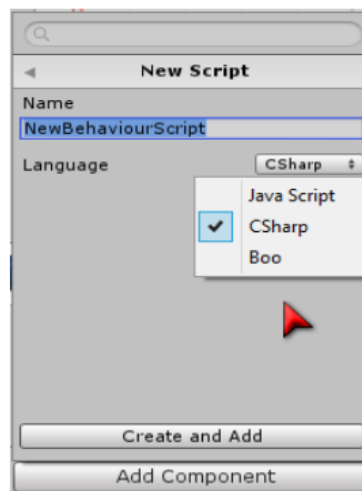
Εφόσον βρέθηκε το script component Health του example και αποθηκεύτηκε στη "life", πρέπει να αλλάξουμε την μεταβλητή τύπου int, το currentHealth μέσα σε αυτό το script. Αναφερόμαστε σε αυτή τη μεταβλητή με αυτόν τον τρόπο (το script που ανήκει και το όνομα της μεταβλητής μετά από τελεία):

- **life.currentHealth**

Κεφάλαιο 6° MonoDevelop

Η Unity προσφέρει γενικά πολλές δυνατότητες. Όμως για να βγει ένα καλό αποτέλεσμα όσον αφορά το σύνολο ενός παιχνιδιού θα πρέπει να υπάρχει και κώδικας που να εξυπηρετεί τις διάφορες απαιτήσεις του παιχνιδιού αυτού. Το MonoDevelop είναι το πρόγραμμα το οποίο εξυπηρετεί αυτές τις απαιτήσεις και προσφέρει στο developer τη δυνατότητα να γράψει τον κώδικα για το παιχνίδι του.

Κώδικας μπορεί να προστεθεί στο εκάστοτε Project είτε προσθέτοντας ως Component κάποιο Script όπως ειπώθηκε πιο πάνω ή μέσω δημιουργίας ενός script από το παράθυρο Project του editor και στη συνέχεια αυτό να προστεθεί στο επιθυμητό αντικείμενο. Έπειτα το script ανοίγει κατ' επιλογήν με το πρόγραμμα Monodevelop για να γίνει δυνατή η επεξεργασία του.



Add new Script

Η Unity σε συνδιασμό με το Monodevelop υποστηρίζει τρεις γλώσσες προγραμματισμού για την κατασκευή των παιχνιδιών της. Αυτές είναι:

- C# :Αντικειμενοστραφής γλώσσα προγραμματισμού, ανεπτυγμένη από τη Microsoft.
- JavaScript: Διερμηνευμένη γλώσσα προγραμματισμού επηρεασμένη από τη C και ανεπτυγμένη από τη Netscape.
- Boo: Στατική γενικού σκοπού γλώσσα προγραμματισμού εμπνευσμένη από τη σύνταξη της Python και ανεπτυγμένη από τον Rodrigo B. De Oliveira.

Ο λόγος που αυτές οι τρεις γλώσσες προγραμματισμού έχουν υιοθετηθεί από τη Unity και αποτελεί το βασικό τους κοινό χαρακτηριστικό, είναι πως και οι τρεις είναι αντικειμενοστραφείς.

Το Monodevelop είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment – IDE) για πλατφόρμες με χαρακτηριστικά παρόμοια με αυτά του Visual Studio, όπως η αυτόματη συμπλήρωση κώδικα, τον έλεγχο πηγής, το γραφικό περιβάλλον χρήστη (Graphical User Interface - GUI) και σχεδιασμό ιστοχώρων.

Το πρόγραμμα αυτό καθιστά δυνατό στους developers τη γρήγορη ανάπτυξη εφαρμογών για λειτουργικά συστήματα Linux, Windows και MacOS X. Υποστηρίζει τις παρακάτω γλώσσες προγραμματισμού :

- Boo
- C
- C++
- C#
- CIL
- D
- F#

7.1 Σύλληψη Ιδέας

Την δημιουργία ενός παιχνιδιού την αναλαμβάνει συνήθως μια ομάδα στην οποία ο καθένας είναι υπεύθυνος για κάτι διαφορετικό. Έτσι αποφάσισα τα αντικείμενα του παιχνιδιού και ο χαρακτήρας που συνήθως τα φτιάχνει ο graphic designer να τα πάρω από το Asset Store. Το επόμενο βήμα είναι η δημιουργία ενός πλάνου όσο αναφορά ποιος είναι ο στόχος του παιχνιδιού. Στόχος είναι η δημιουργία μιας δωρεάν εκδόσεις η οποία θα παρουσιαστεί στη πτυχιακή και στο μέλλον μιας επί πληρωμής. Στην δωρεάν έκδοση του παιχνιδιού ο σκοπός του είναι ο χρήστης να απολαύσει το παιχνίδι χωρίς να στρεσαριστεί και να χαλαρώσει με την μουσική.

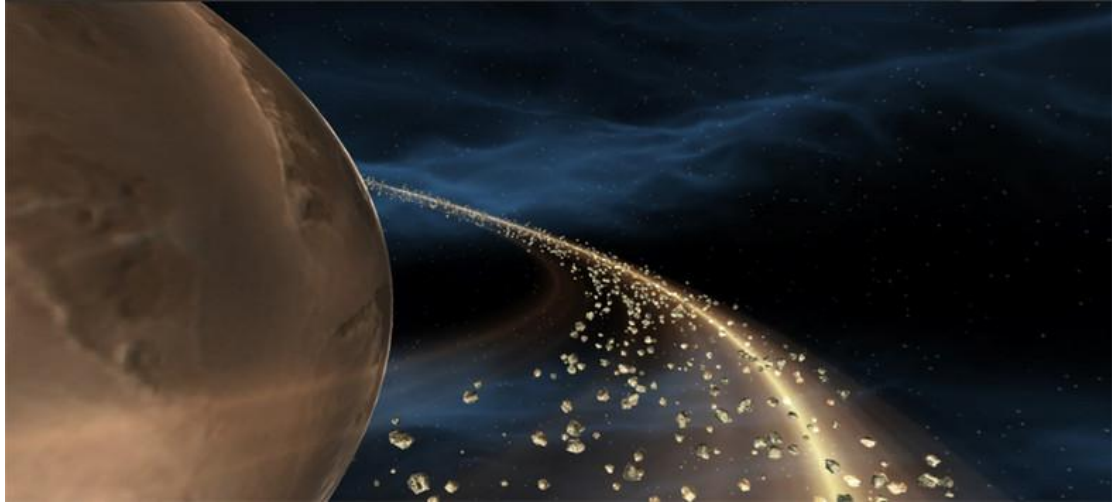
7.2 Επιλογή χαρακτήρα

Στο κομμάτι της επιλογής χαρακτήρα υπήρχαν διάφορες επιλογές. Υπάρχει η επιλογή αναζήτησης του χαρακτήρα από το Asset Store του Unity στο οποίο υπάρχει μια ποικιλία επιλογών. Επίσης υπάρχει και η δημιουργία από κάποιο πρόγραμμα όπως το blender.



7.3 Επιλογή αντικειμένων

Ως ουρανός στο πρώτο επίπεδο χρησιμοποιήθηκε:



Sky level 1

Ως ουρανός στο δεύτερο επίπεδο χρησιμοποιήθηκε:



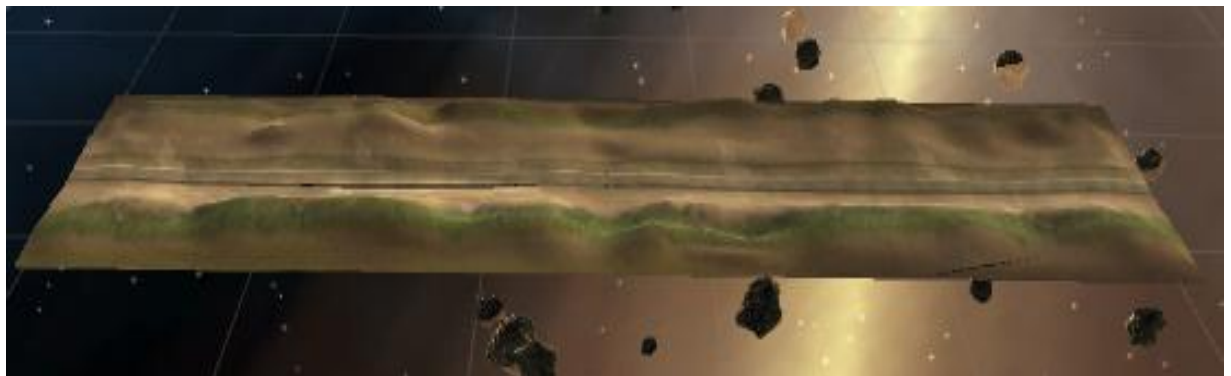
Sky level 2

Ως ουρανός στο τρίτο επίπεδο χρησιμοποιήθηκε:



Sky level 3

Ως έδαφος χρησιμοποιήθηκε:



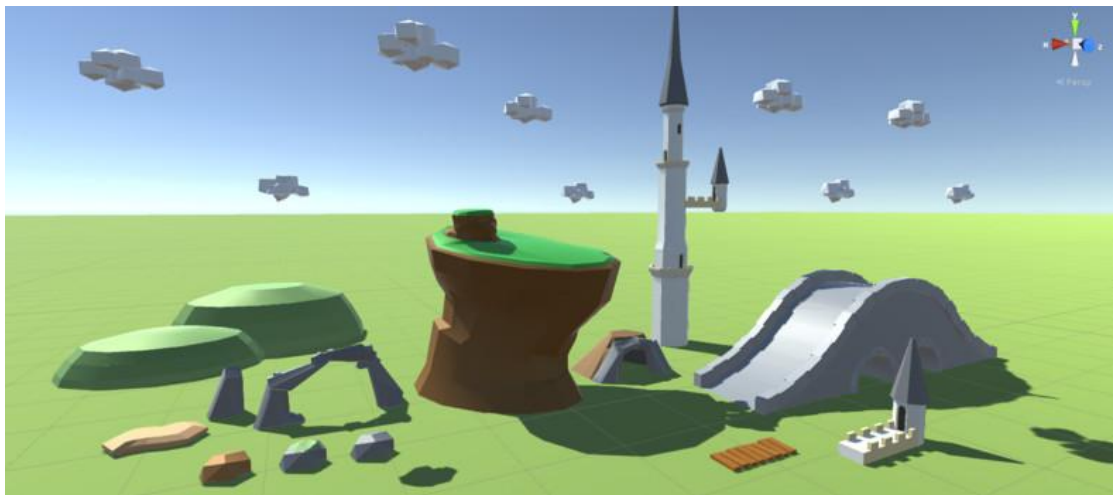
Terrain

Τα κτήρια που χρησιμοποιήθηκαν είναι από asset store :



Medieval Town Exteriors

Χρησιμοποιήθηκαν επίσης :





Low PolyPack

Οι αντίπαλοι :

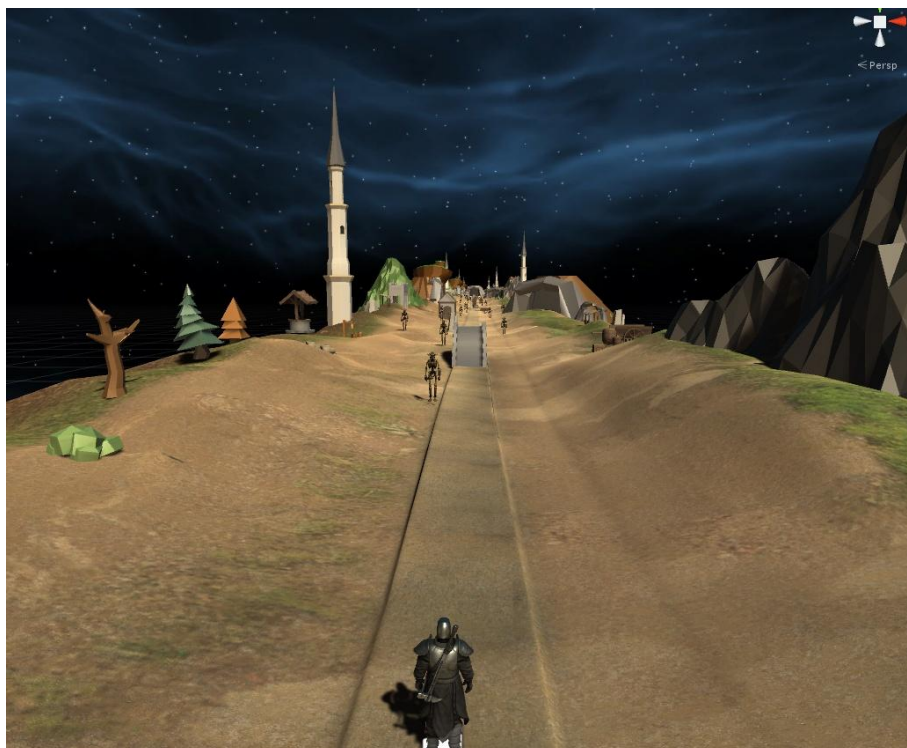




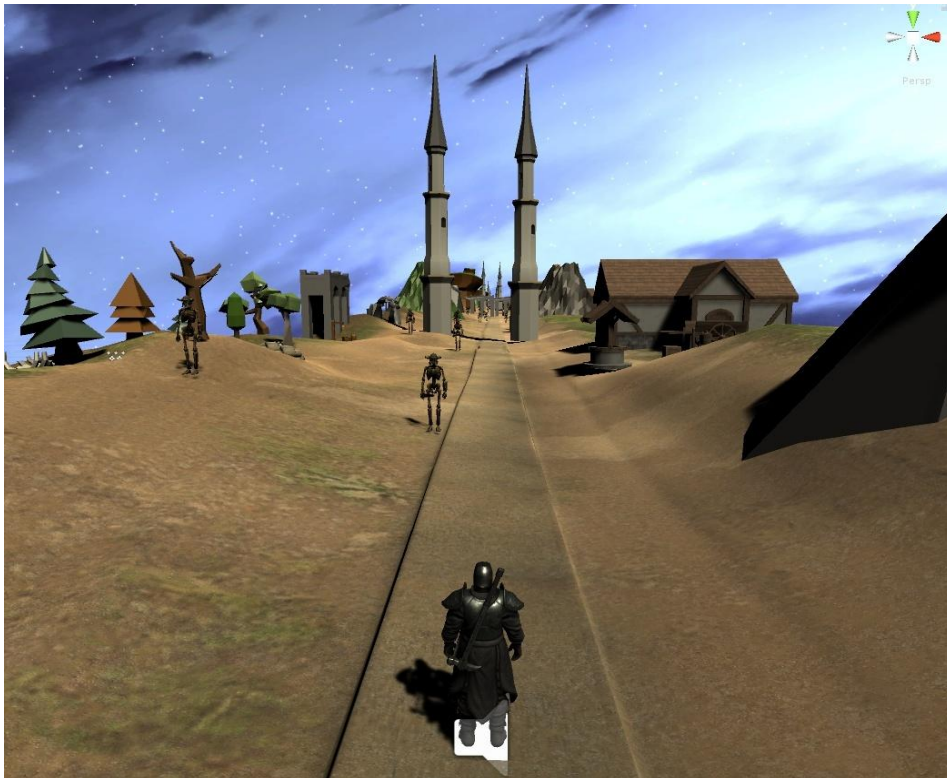
Dungeon Skeletons

7.4 Παρουσίαση επιπέδων:

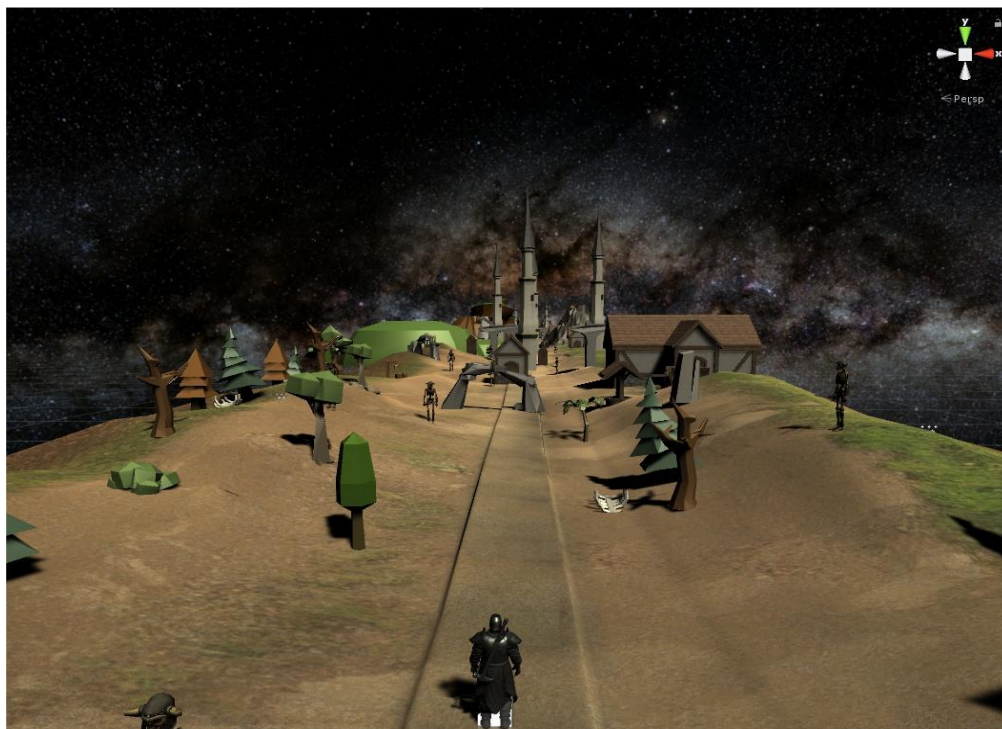
1^ο επίπεδο



2° επίπεδο

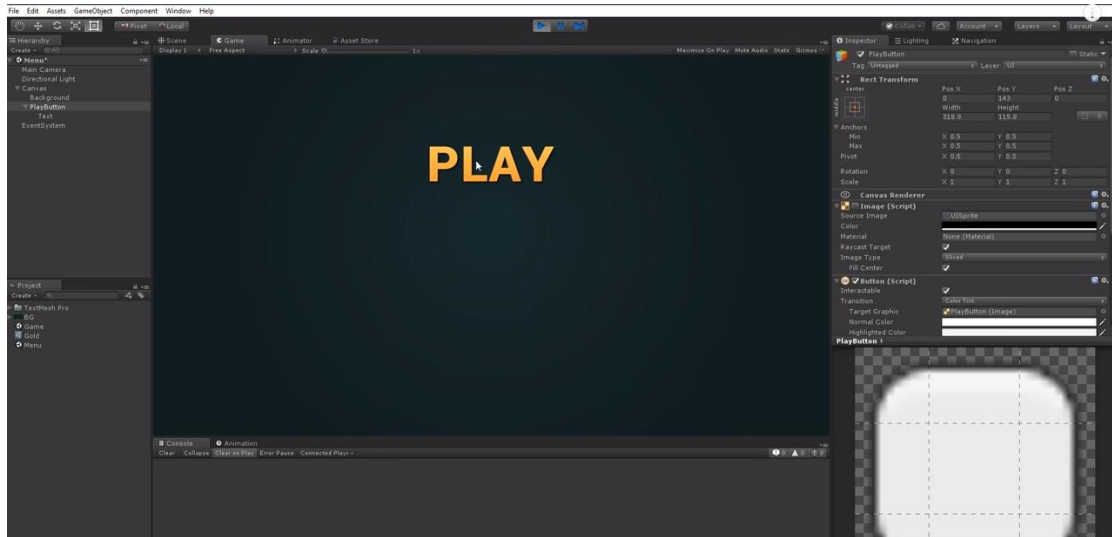


3° επίπεδο

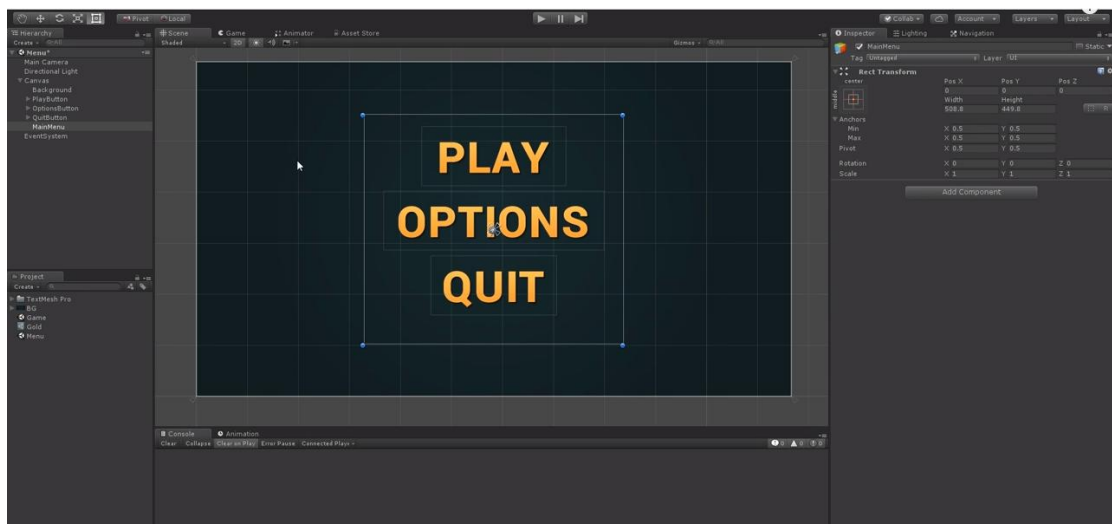


7.5 Δημιουργία του Main Menu

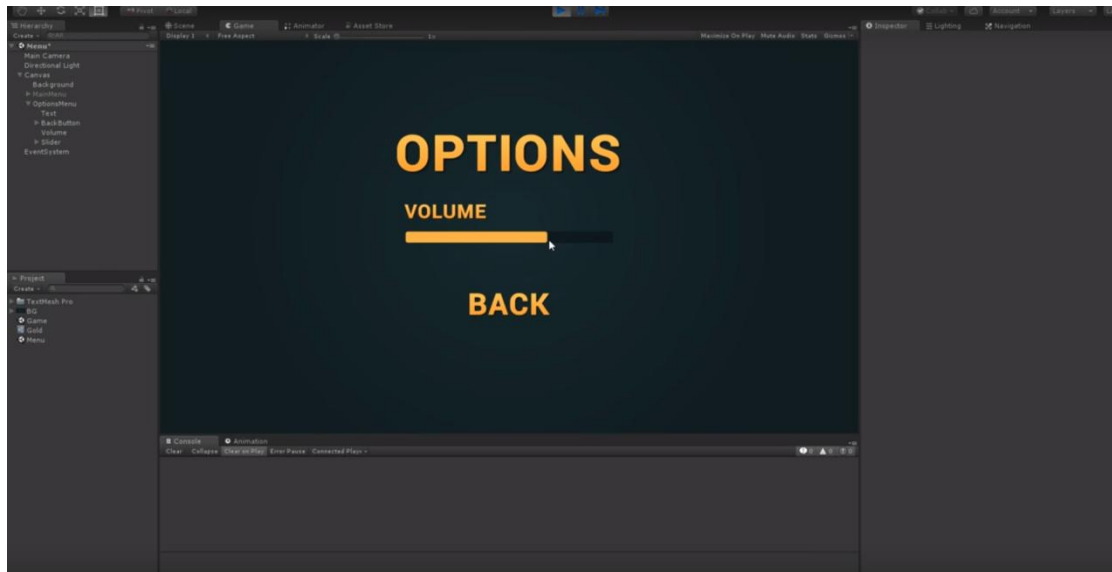
Στην αρχή επιλέγουμε στο hierarchy την επιλογή -> UI -> Panel αυτό θα δημιουργήσει ένα Canvas το οποίο θα περιέχει ένα panel. Μετά επιλέγουμε πάλι -> UI -> Text για να γράψουμε την επιλογή που θέλουμε στην περίπτωση μας το Play. Το επόμενο βήμα είναι να κάνουμε το play κουμπί. Οπότε -> UI -> Button



Το ίδιο κάνουμε και για το options και για το quit



Μετά φτιάχνουμε την επιλογή options που ρυθμίζει τον ήχο.



Για να δουλέψει το main menu πρέπει να φτιάξουμε τον κώδικα

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class MainMenu : MonoBehaviour {
7
8      public void PlayGame ()
9      {
10         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
11     }
12
13     public void QuitGame ()
14     {
15         Application.Quit();
16     }
17 }
18
19
```

Η βιβλιοθήκη στην γραμμή 4 είναι απαραίτητη όταν θέλουμε να αλλάξουμε σκηνές στο unity.

Η γραμμή 8 εκτελείτε όποτε πατάμε το κουμπί play. Για αυτό το λόγο η μέθοδος θα είναι δημόσια.

Η γραμμή 10 όταν εκτελείται μας πάει στην επόμενη σκηνή

Η γραμμή 15 είναι μια καινούργια μέθοδος η οποία είναι για την έξοδο από το παιχνίδι.

7.6 Ο κώδικας του χαρακτήρα

```
running.cs* X
infinity knight runner -enemys - 1.1 - Copy
running
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class running : MonoBehaviour {
7
8     private Rigidbody rb;
9     public float speed = 5f;
10
11
12     void Start()
13     {
14         rb = GetComponent<Rigidbody>();
15     }
16
17
18
19     void Update()
20     {
21         rb.velocity = new Vector3(Input.acceleration.x*5f, rb.velocity.y, speed);
22     }
23
24
25
26 }
```

Η γραμμή 6 είναι το όνομα που έχουμε δώσει στην κλάση

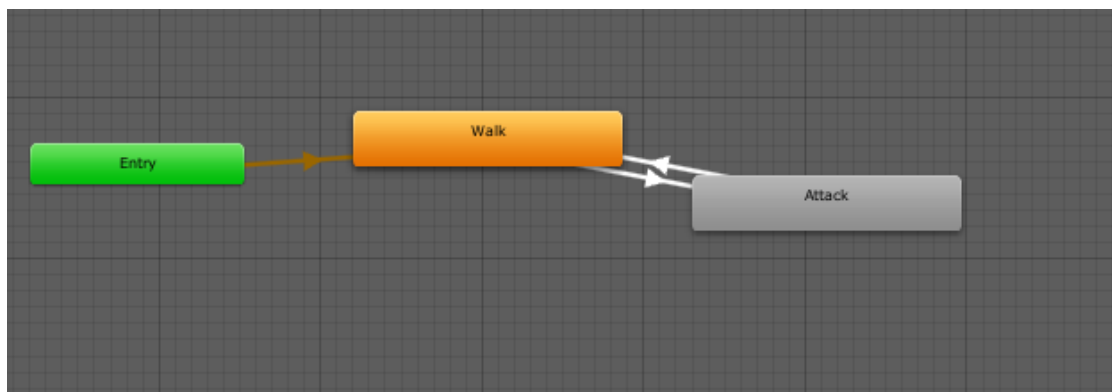
Στην γραμμή 8 φτιάξαμε μια ιδιωτική μεταβλητή τύπου rigidbody

Στην γραμμή 9 δημιουργήσαμε μια δημόσια μεταβλητή speed. Ο λόγος που το κάναμε αυτό είναι για να μπορούμε να αλλάζουμε την ταχύτητα από το unity editor .

Στην γραμμή 14 αντλείται το στοιχείο Rigidbody (από τον παίκτη) και αντιστοιχείται με την μεταβλητή rb.

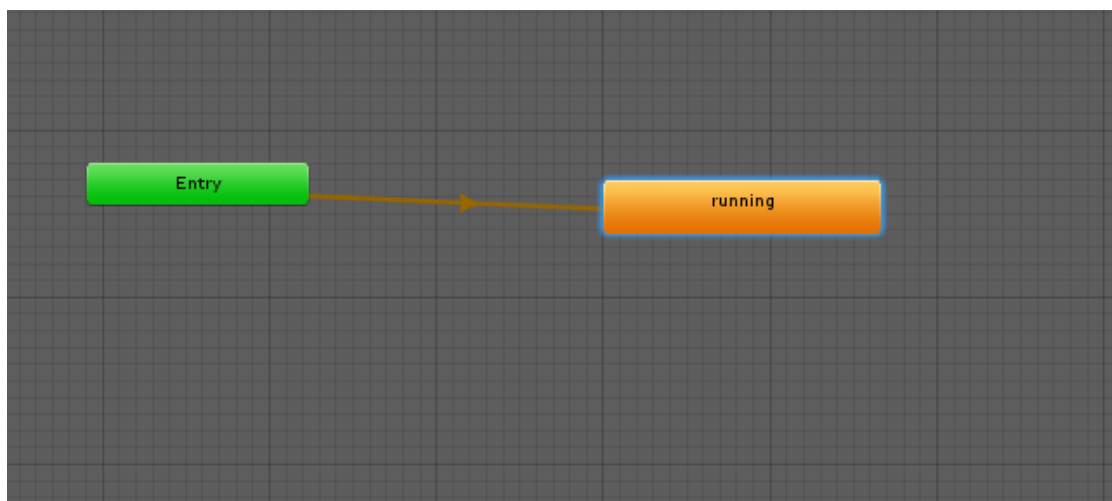
7.7 Το Animation των χαρακτήρων

Για τα npc



Τα npc όταν δεν τα πλησιάζει ο παίχτης απλά περπατάνε στον χώρο. Όταν τα πλησιάσει ο παίχτης του επιτίθενται.

Για τον παίχτη

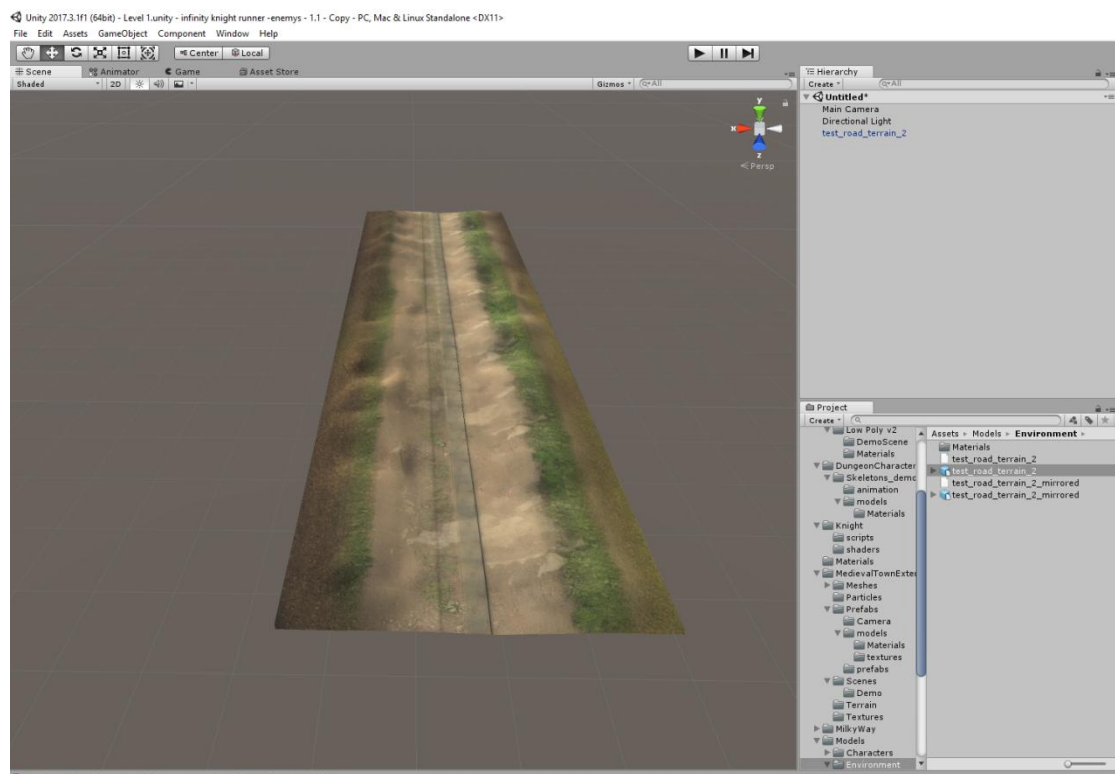


Ο παίχτης το μόνο animation που χρειάζεται είναι αυτό που τρέχει.

7.8 Δημιουργία του περιβάλλοντος του παιχνιδιού

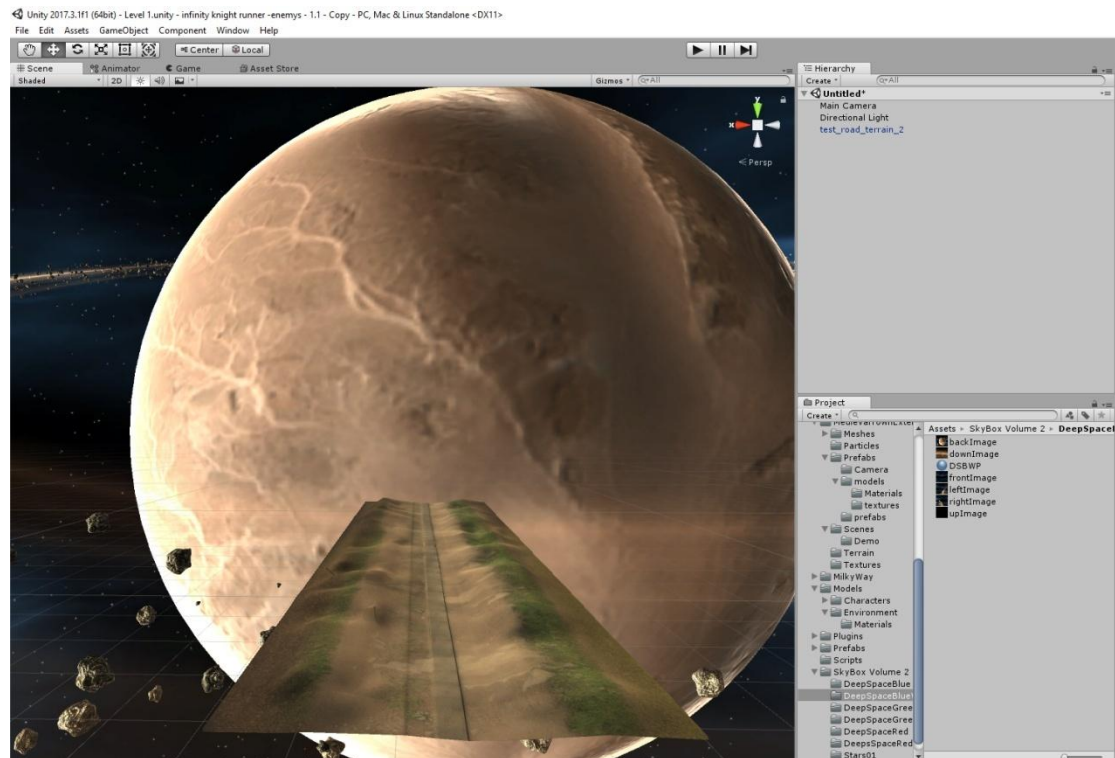
Τοποθέτηση terrain

Πρώτα αρχίζουμε τοποθετώντας το έδαφος για να μπορούμε να διαμορφώσουμε τον χώρο με βάση τις επιθυμίες μας. Οπότε σέρνουμε το prefab του terrain στην σκηνή μας.

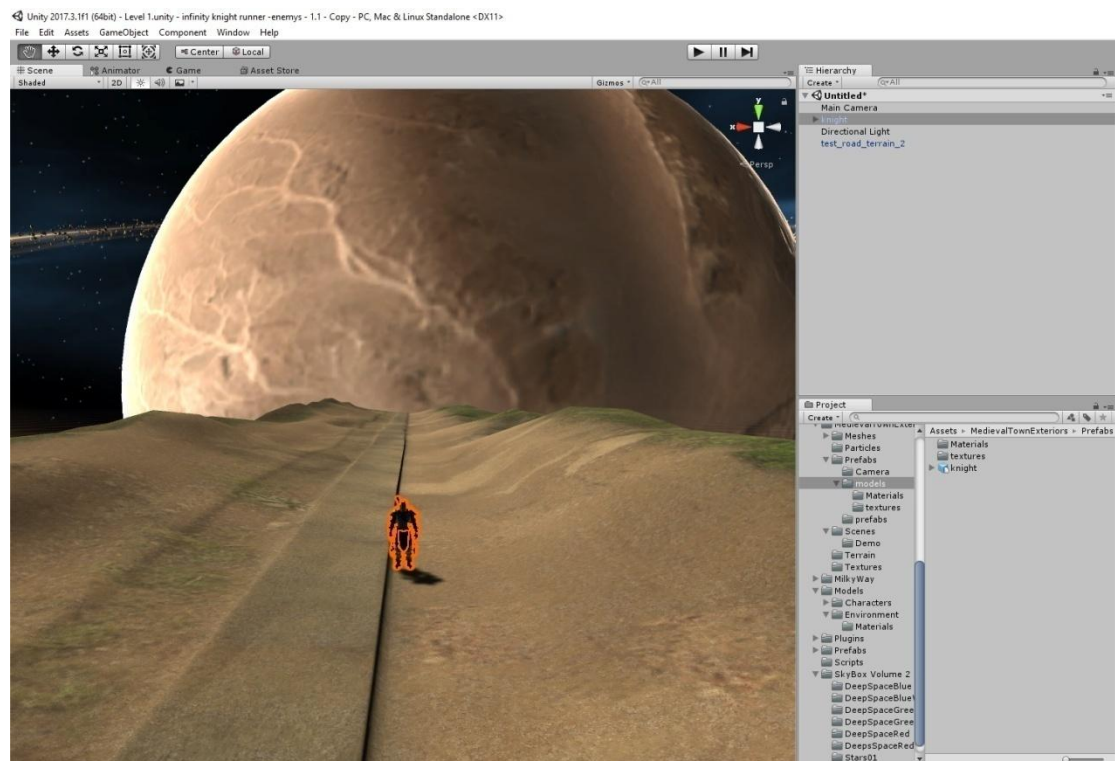


Τοποθέτηση ουρανού

Η τοποθέτηση γίνεται με τον ίδιο τρόπο.

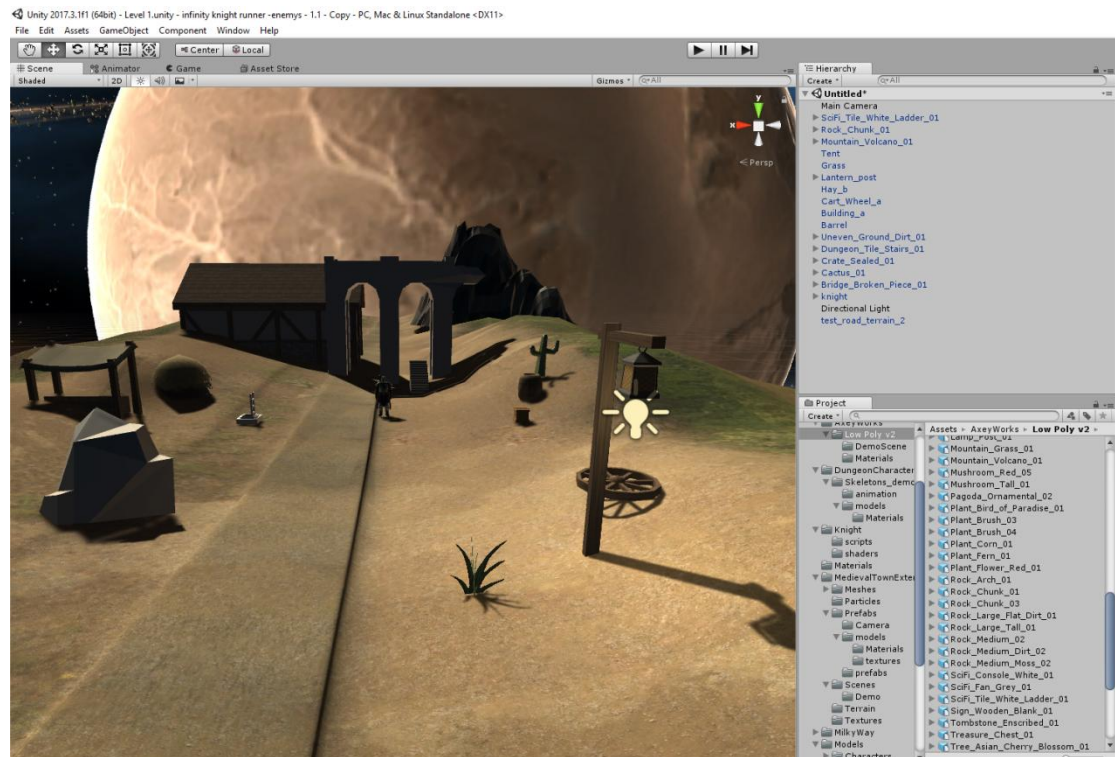


Εισαγωγή χαρακτήρα



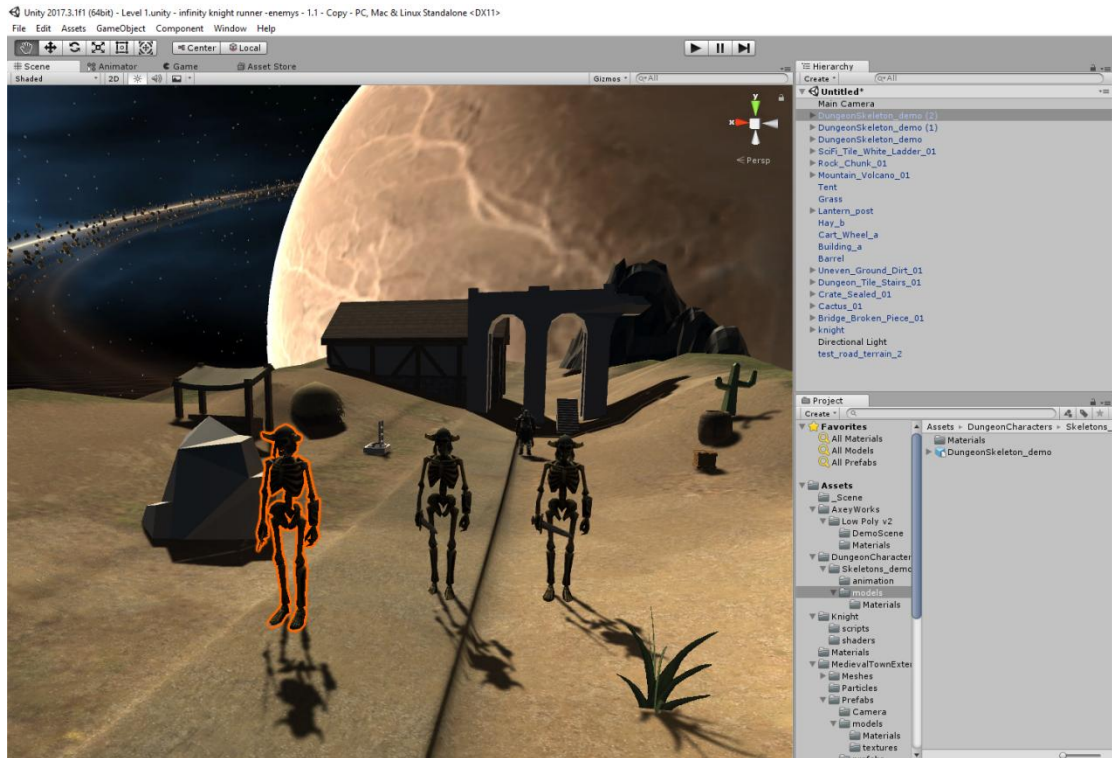
Τοποθέτηση αντικειμένων στον χώρο

Η διαμόρφωση του χώρου είναι στο χέρι του δημιουργού. Στα πρώτα λέβελ ήταν λιγότερα έτσι ώστε να είναι πιο εύκολα για τον χρήστη.



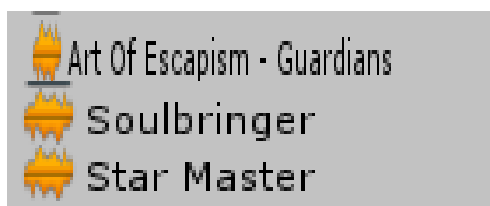
Εισαγωγή των αντιπάλων

Το αντίπαλο δέος του χαρακτήρα μας. Το πλήθος των αντιπάλων αυξάνεται όσο αλλάζουμε λέβελ. Οι δυνατότητες του είναι να κινηθεί μέσα στον χώρο και να μας επιτεθεί όταν μπούμε στην εμβέλεια του.



7.9 Η μουσική του παιχνιδιού

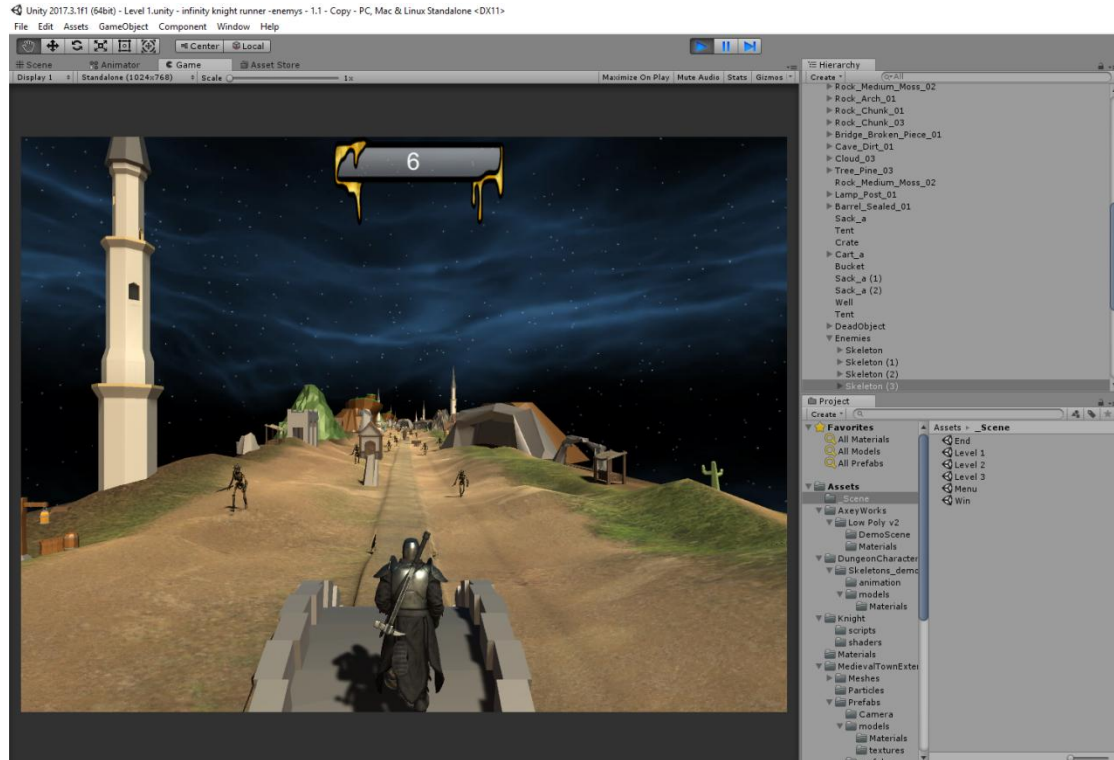
Η μουσική ήταν ένα από τα σημαντικά κομμάτια του παιχνιδιού γιατί είχε σκοπό την ψυχαγωγία και την χαλάρωση του χρήστη.



Όλα τα τραγούδια είναι δωρεάν απο ίντερνετ . Η εύρεση των τραγουδιών ήταν ένα από τα πιο χρονοβόρα κομμάτια του σχεδιασμού γιατί ήταν πολύ δύσκολη η εύρεση του κατάλληλου τραγουδιού.

7.10 Το τελικό αποτέλεσμα

Το πρώτο λέβελ είναι και το πιο εύκολο η ταχύτητα είναι μικρότερη, τα αντικείμενα και τα ηrc είναι λιγότερα. Το τραγούδι αυτού του λέβελ είναι το Art of Escarpism .



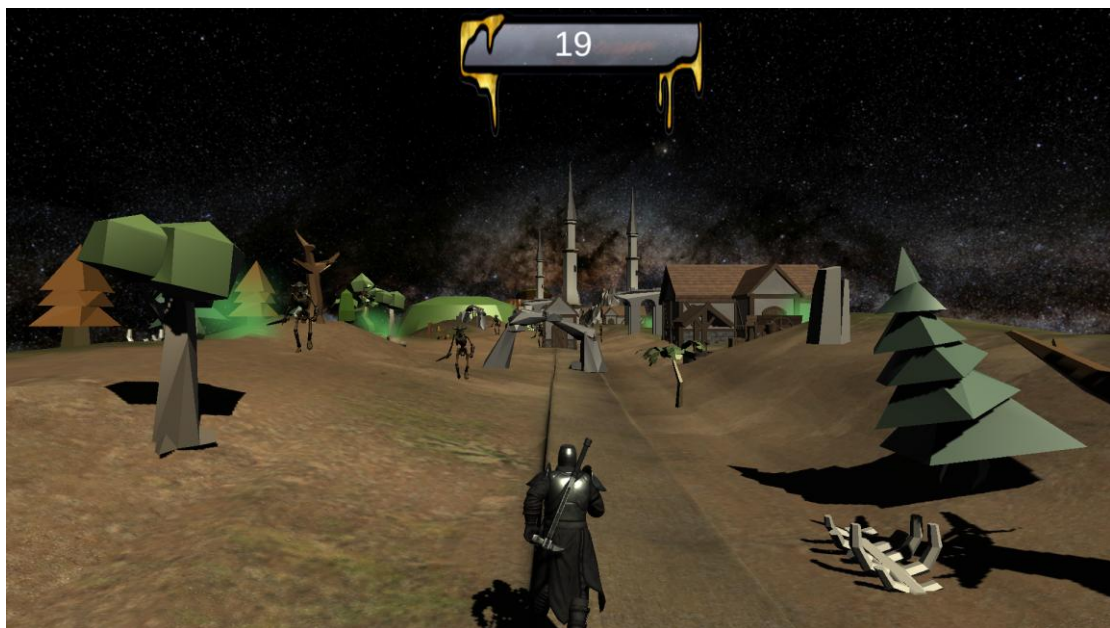


Το δεύτερο λήβελ είναι μεγαλύτερης δυσκολίας και έχει διαφορετικό τραγουδι Soulbringer .





Τέλος το τρίτο λήβελ το οποίο είναι το πιο γρήγορο και πιο δύσκολο έχει τραγούδι Star Master.





Κεφάλαιο 8° Έσοδα από την δημιουργία παιχνιδιών

8.1 Οικονομία των παιχνιδιών στον κόσμο

Τα ψηφιακά παιχνίδια εντάσσονται στις λεγόμενες Δημιουργικές Βιομηχανίες (Creative Industries), ένας οικονομικός κλάδος ο οποίος περιλαμβάνει την κινηματογραφική παραγωγή, τις οπτικοακουστικές και ψηφιακές τέχνες, την αρχιτεκτονική, το design, τη μουσική, τη διαφήμιση κ.α. και του οποίου τα έσοδα αντιστοιχούν στο 3% του παγκόσμιου εγχώριου προϊόντος (GDP) με συνεχείς αυξητικές τάσεις. Στην Ελλάδα ο τομέας του ψηφιακού παιχνιδιού υπήρξε για πολλά χρόνια ένας κοιμώμενος γίγαντας, αφού μέχρι πρόσφατα αποτελούσε ένα αχαρτογράφητο πεδίο, εγκλωβισμένο στις προθεσμίες των ευρωπαϊκών προγραμμάτων και απομονωμένο στα studios μεμονωμένων ομάδων ανάπτυξης ψηφιακών παιχνιδιών.

Και όλα αυτά, όταν σε παγκόσμιο επίπεδο τα ψηφιακά παιχνίδια αντιμετωπίζονται ως όχημα ανάπτυξης, καθώς η βιομηχανία των video games αντιστοιχεί σε περισσότερα από 100 δισεκατομμύρια δολάρια: όσον αφορά στην Αγγλία για παράδειγμα, η αγορά των video games υπολογίζεται ότι άγγιξε το 2016, τις 1.72 δις λίρες, ενώ η Γαλλία σύμφωνα με πρόσφατα στοιχεία, έχει στο ενεργητικό της πάνω από 300 εταιρείες game development, από τις οποίες το 77% έχει έσοδα πάνω από 1 εκατ. ευρώ, ενώ ένα μικρό ποσοστό σημειώνει έσοδα πάνω από 10 εκατ. ευρώ.

Τα video games αποτελούν ένα ταχύτατα αναπτυσσόμενο κλάδο ο οποίος προσελκύει ολοένα και περισσότερο το ενδιαφέρον των διεθνών επενδυτών. Τα ηλεκτρονικά αθλήματα (eSports) και οι εφαρμογές εικονικής πραγματικότητας (Virtual Reality) φαίνεται μάλιστα να σηματοδοτούν τη νέα κερδοφόρα τάση στο χώρο των ψηφιακών παιχνιδιών. Η βιομηχανία των video games αποκτά βαρύνουσα σημασία για την οικονομική ζωή μιας χώρας, η οποία επιθυμεί να επενδύσει στρατηγικά στην ανάπτυξη της τεχνολογίας.

Η κυρίαρχη αντίληψη θέλει τα video games να ταυτίζονται με παιχνίδια πολεμικών αναμετρήσεων και στρατηγικής τα οποία προσελκύουν σχεδόν κατα αποκλειστικότητα εφήβους. Αντίθετα, τα ψηφιακά παιχνίδια αποτελούν ένα διαδραστικό μέσο το οποίο τείνει επίσης να περικλείει πολιτιστικά στοιχεία. Είναι ενδεικτικό ότι το μουσείο MOMA της Νέας Υόρκης έχει συμπεριλάβει στη συλλογή του 14 κλασικά και καινούργια ψηφιακά παιχνίδια όπως το Pac-man (1980) και το Flow (2006), αναγνωρίζοντας κατα αυτόν τον τρόπο την αξία τους από άποψη σχεδιασμού, απεικόνισης χώρου, καλλιτεχνικής έκφρασης κτλ. Είναι επίσης ενδεικτικό ότι σήμερα λειτουργούν 14 μουσεία ψηφιακών παιχνιδιών- μουσεία που εκθέτουν κονσόλες, παιχνίδια, παιγνιώδεις εφαρμογές, αντικείμενα και προϊόντα που σχετίζονται με την παιγνιώδη κουλτούρα. Τα ψηφιακά παιχνίδια χρησιμοποιούνται πια για να εκφράσουν και να διαδώσουν στοιχεία πολιτισμού: το παιχνίδι Never Alone, το οποίο κέρδισε βραβείο BAFTA (2015) για «Best Debut Game» είναι αφιερωμένο στην κουλτούρα και τον πολιτισμό των Ινουίτ, των Εσκιμώων της Αλάσκας. Πρόκειται για ένα παιχνίδι το οποίο έχει ενσωματώσει πληροφορίες και τεκμήρια του πολιτισμού των Ινουίτ.

Τα ψηφιακά παιχνίδια συστήνουν πλέον ένα πολύπλοκο σύστημα και μια σύνθετη διανοητική προσπάθεια, η οποία απαιτεί ψηφιακές δεξιότητες και διαθέτει αφήγηση, σενάριο, γραφιστικά στοιχεία με στόχο να υποστηρίξει σε διαρκή βάση αναρίθμητα κοινωνικά δίκτυα και κοινότητες παικτών σε παγκόσμιο επίπεδο. Στη βάση των δεδομένων αυτών και έχοντας υπόψη μας το γεγονός ότι τα video games έχουν διεθνικό χαρακτήρα, με πολλά studios να συνεργάζονται με επαγγελματίες οι οποίοι εργάζονται και διαμένουν σε διαφορετικές χώρες, με εταιρείες οι οποίες έχουν μεταφέρει τη έδρα τους σε κάποια άλλη χώρα με ευνοϊκότερο οικονομικό και φορολογικό σύστημα, το στοίχημα της κάθε χώρας που επιθυμεί να επενδύσει στον τομέα ανάπτυξης ψηφιακών παιχνιδιών (συμπεριλαμβάνοντας όλες τις ειδικότητες που εμπλέκονται στη διαδικασία αυτή όπως πχ. Μουσικοί, artists, σχεδιαστές, προγραμματιστές, σκηνοθέτες, animators, σεναριογράφοι κτλ.) είναι: πώς θα συγκρατήσει το brain drain, εξειδικευμένους επαγγελματίες που θα αναζητήσουν εργασία σε άλλη χώρα πώς θα προσελκύσει μεγάλα studios τα οποία και θα ανοίξουν κάποιο τοπικό παράρτημα στη χώρα.

Χαρακτηριστικό παράδειγμα αποτελεί η εταιρεία Wargaming με έδρα τη Λευκωσία, η οποία διαθέτει 15 studios σε διαφορετικές χώρες και με έσοδα που για το 2015 και μόνο για το παιχνίδι World of Tanks, έφθασαν τα \$446 εκατ.

Ο ανταγωνισμός είναι μεγάλος και σημαντικό ρόλο στις παραπάνω εξελίξεις διαδραμάτισαν τα φορολογικά και επενδυτικά κίνητρα που υιοθέτησαν οι περισσότερες χώρες. Στην Αγγλία και στον Καναδά, η βιομηχανία των ψηφιακών παιχνιδιών λειτουργεί υπό καθεστώς tax relief, ενώ στην Αμερική εφαρμόζεται ένας συνδυασμός tax relief και cash rebate το οποίο διαφοροποιείται από πολιτεία σε πολιτεία (πχ. στη Φλόριντα, στο Κολοράντο, στο Μίσιγκαν και στο Αρκάνσας έχει εφαρμοστεί η τακτική του cash rebate). Η Γαλλία από το 2008 υιοθέτησε φορολογικές ελαφρύνσεις της τάξεως του 20% για την παραγωγή video games, στην Αγγλία το αντίστοιχο ποσοστό κυμαίνεται στο 25%, στον Καναδά και συγκεκριμένα στο Οντάριο, αγγίζει το 40%. Στη Γαλλία επίσης παρέχονται οικονομικές κρατικές ενισχύσεις για την ενίσχυση του pre-production, καθώς και για την ανάδειξη των ψηφιακών παιχνιδιών μέσω φεστιβάλ, ενώ στη Γερμανία και τη Βρετανία σημαντικό ρόλο διαδραματίζουν και τα Περιφερειακά Ταμεία Στήριξης με στόχο την προώθηση καινοτόμων ψηφιακών παιχνιδιών που χαρακτηρίζονται από υψηλή πολιτιστική ή εκπαιδευτική αξία ή εμφανίζουν μεγάλη εμπορική δυνατότητα. Τέλος, στην Ισπανία διεξάγεται συζήτηση για εισαγωγή φορολογικών και χρηματοοικονομικών κινήτρων tax rebate και tax relief παρόμοια με εκείνα της τηλεόρασης, του κινηματογράφου και του θεάτρου, με σκοπό την προσέλκυση ξένων επενδυτών.

Τα στοιχεία αυτά αναδεικνύουν το γεγονός ότι τα video games όπως άλλοι τομείς υψηλού επενδυτικού ενδιαφέροντος όπως κατεξοχήν είναι ο κινηματογράφος, αποτελούν μια συνεχώς αναπτυσσόμενη βιομηχανία όπου η προσέλκυση επενδύσεων αλλά και η παραγωγή ολοένα και πιο ανταγωνιστικών προϊόντων, θέτει ως επιτακτική ανάγκη το ζήτημα κρατικής ενίσχυσης.

Και ενώ η κινηματογραφική/ τηλεοπτική παραγωγή διαθέτει υψηλό κόστος λόγω εξοπλισμού και γυρισμάτων, όπου η φυσική παρουσία ενός επαγγελματικού συνεργείου σε επιλεγμένα σημεία ενδιαφέροντος κρίνεται σχεδόν αναπόφευκτη, στην περίπτωση των ψηφιακών παιχνιδιών η κατάσταση είναι διαφορετική. Η ανάπτυξη ενός ψηφιακού παιχνιδιού μπορεί να έχει από μηδενικό κόστος, ως εκατοντάδες εκατομμύρια δολάρια. Χαρακτηριστική είναι η περίπτωση του παιχνιδιού Destiny με κόστος παραγωγής 500 εκατ. δολάρια.

Η διαφορά μεταξύ των δύο μέσων, έγκειται στο γεγονός ότι στη περίπτωση των ψηφιακών παιχνιδιών, η επένδυση αφορά στην τεχνολογία και στην

ανάπτυξη περιεχομένου, χωρίς να αναγνωρίζονται εδαφικές δεσμεύσεις. Στην ανάπτυξη ψηφιακών παιχνιδιών δεν υπεισέρχονται ζητήματα εδαφικότητας. Η αναπαράσταση και προσομοίωση των τοπίων και του εικονικού περιβάλλοντος στο πλαίσιο ενός ψηφιακού παιχνιδιού, δεν απαιτεί τη φυσική παρουσία του δημιουργού/σχεδιαστή στο φυσικό χώρο. Κατ' επέκταση, τα επενδυτικά κίνητρα που μπορεί μια χώρα να παράσχει σε επενδυτές, σχετίζονται κατά κύριο λόγο με παραμέτρους όπως ελκυστικό φορολογικό περιβάλλον, ολοκληρωμένο σύστημα επενδυτικών κινήτρων, διαθεσιμότητα δεξιοτήτων και εγχώριου επαγγελματικού δυναμικού και γιατί όχι, ποιοτικές συνθήκες διαβίωσης.

8.2 Τρόποι Εσόδων On-Line Παιχνιδιών

-Διαφήμιση.Θα μπορούσε να αναρωτηθεί βέβαια, ο οποιοσδήποτε, πως συντηρούνται, πολλαπλασιάζονται και μερικώς αναβαθμίζονται αυτά τα παιχνίδια, από τη στιγμή που είναι δωρεάν στην πλειοψηφία τους (αρκετά ζητούν να καταβάλλει ο παίκτης ένα μικρό αντίτιμο που διακυμαίνεται από 5\$ έως 20\$ για την deluxe έκδοση του εκάστου παιχνιδιού). Η δημοτικότητα τους αυξήθηκε ραγδαία σε εκατομμύρια χρήστες ανά τον κόσμο και αυτό συνέβαλε στην ιδέα του In-game advertising, εισαγωγής δηλαδή διαφημίσεων σε πολλαπλά και συγκεκριμένα σημεία της ιστοσελίδας που φιλοξενεί το παιχνίδι.



Παράδειγμα in-game advertising

Ιστορικά η ιδέα αυτή πρωτοέγινε πράξη το 1978 με το παιχνίδι Adventureland και αργότερα το 1998 με το παιχνίδι James bond-Robocod, που και τα δύο ήταν games σε πλατφόρμα ηλεκτρονικού υπολογιστή.

Πιο αναλυτικά η βιομηχανία IGA (In-game advertising) αναφέρει έσοδα ανά τα έτη:

2004	34 εκατομμύρια \$
2005	56 εκατομμύρια \$
2006	80 εκατομμύρια \$
2007	296 εκατομμύρια \$
2009	669 εκατομμύρια \$
2014	Με τελική εκτίμηση εσόδων να φτάνει το <u>1 δισεκατομμύριο \$</u>

Έσοδα του κλάδου από το in-game advertising

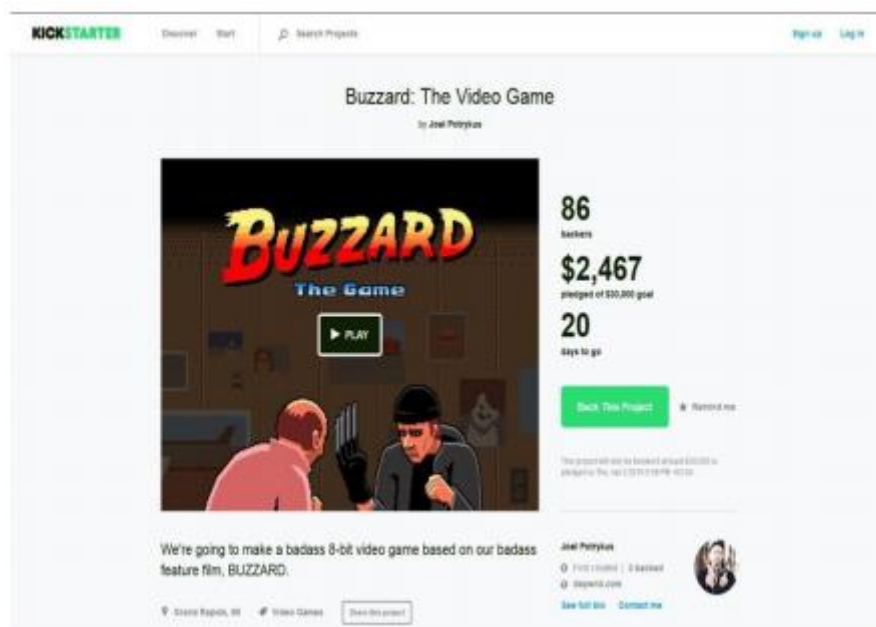
Άλλη πηγή εσόδων σε αρκετές περιπτώσεις παιχνιδιών είναι η χρηματοδότηση τους από διάφορους χορηγούς (sponsors), που απώτερο σκοπό έχουν την πλήρη διαφήμιση μέσω του παιχνιδιού που χορηγούν. Στοχευόμενη και προσωποποιημένη διαφήμιση στο συγκεκριμένο κοινό, που αποτελεί τους παίκτες του εκάστοτε game.

- Δωρεά (Donation). Σε πολλά από τα casual games, εμφανίστηκε η λειτουργία της δωρεάς (donation), δηλαδή σε κάποιο πλαίσιο της ιστοσελίδας υπάρχει η επιλογή όπου μπορεί ο παίκτης να κάνει δωρεά όποιο ποσό επιθυμεί για το αγαπημένο του παιχνίδι.

Στατιστική έρευνα στα 100 άτομα έδειξε ότι περίπου οι 70 στους 100 δεν θα δώσουν ποτέ κάποιο αντίτιμο, από τους 30 που μένουν οι 10 θα κάνουν δωρεά από 0,50€ έως 1€, ενώ στους υπόλοιπους 20, οι 5 θα δώσουν από 1€ έως 5€, οι 10 θα δώσουν από 10€ έως 20€ και οι τελευταίοι 5 θα καταβάλλουν από 50€ έως 100 €, σημαντικά έσοδα δηλαδή για τις εταιρίες ή τους ατομικούς προγραμματιστές από αυτήν την λειτουργία.

Επίσης, υπάρχουν διάφορες ιστοσελίδες όπως π.χ. το Kickstarter (43, Kickstarter, 2015), οι οποίες βοηθούν τους δημιουργούς ενός project, να αποκτήσουν κεφάλαιο, μέσω της δωρεάς ανθρώπων από όλο τον κόσμο, για την τελειοποίηση των ιδεών τους (μεταξύ άλλων και video games). Ο επισκέπτης, βλέπει μία σύντομη παρουσίαση του project και αποφασίζει εάν και τι ποσό θα κάνει δωρεά σε αυτό.

Ενδεικτικά αναφέρουμε ότι μόνο το έτος 2009, 8.400.000 άνθρωποι έκαναν δωρεές, που στο σύνολό τους, ξεπέρασαν τα 1,7 δισεκατομμύρια \$ και αφορούσαν την χρηματοδότηση 83.000 νέων project.



Εικόνα 15: ΠΑΡΑΔΕΙΓΜΑ ΔΩΡΕΑΣ ΣΕ ΠΑΙΧΝΙΔΙ, ΣΤΗΝ ΙΣΤΟΣΕΛΙΔΑ KICKSTARTER.

Παράδειγμα δωρεάς σε παιχνίδι Kickstarter

8.3 Πληρωμή Για Πλήρη Χρήση Του Παιχνιδιού

Η δημιουργία πολλών παιχνιδιών ανά είδος, που φέρουν τον όρο free-to-play, έχει εξαπλωθεί σε όλο τον πλανήτη όπως αναφέραμε και πιο πάνω. Πιο αναλυτικά, τα παιχνίδια που φέρουν τον ορισμό free-to-play αφήνουν τον παίκτη να γίνει μέλος του παιχνιδιού δωρεάν, χωρίς να ζητάει κάποιο αντίτιμο από αυτόν, να μπορεί να παίξει όσο θέλει, να τερματίσει το παιχνίδι, χωρίς απλά να έχει στην διάθεσή του το 100% των περιεχομένων του τίτλου (features) όπως π.χ. στο παιχνίδι Planet Side 2: όπλα, πανοπλίες, συγκεκριμένα οχήματα, κ.τ.λ.

Κάποια άλλα επιτρέπουν στον παίκτη να φτάσει μέχρι ένα σημείο στην συνολική πλοκή του παιχνιδιού και μετά ζητούν κάποιο αντίτιμο ώστε ο παίκτης να μπορέσει να συνεχίσει και να ανέβει παραπάνω κατηγορία (level) π.χ. το World of Warcraft. Σε αυτό το σημείο η εξέλιξη των τραπεζικών συστημάτων και η ευελιξία του internet στις διατραπεζικές συναλλαγές βοηθούν στην δημιουργία καινούργιων τραπεζικών προϊόντων που δρουν καταλυτικά στην επιβίωση αυτών των παιχνιδιών.

Οι πιο διαδεδομένες μορφές ηλεκτρονικών πληρωμών είναι:

- Η πληρωμή από πιστωτική κάρτα (credit card).
- Η πληρωμή από προπληρωμένη κάρτα (pre-paid card).
- Η πληρωμή μέσω της υπηρεσίας PayPal.

8.4 Θετικά Και Αρνητικά Των Free - To - Play Παιχνιδιών

Όλοι οι άνθρωποι, ψάχνουν την δωρεάν, ή τουλάχιστον την πιο φθηνή λύση, στο να αποκτήσουν κάτι που επιθυμούν. Πόσο μάλλον στην ηλεκτρονική ψυχαγωγία. Τις περισσότερες φορές όμως, η δωρεάν λύση έχει διάφορες συνέπειες, θετικές και αρνητικές. Κατ' αρχήν, ο ορισμός "free-to-play" είναι στην ουσία λάθος, διότι ναι μεν ο χρήστης έχει το παιχνίδι που επιθυμεί δωρεάν, αλλά όχι στο 100%, όπως αναφέραμε παραπάνω. Πιο πολύ του ταιριάζει ο όρος "πλήρωσε όσο θέλεις". Ο εκδότης του κάθε παιχνιδιού, θέλει να πληρωθεί για την δουλειά του και ένας τρόπος να το πραγματοποιήσει, είναι να διαθέσει την βασική έκδοση του παιχνιδιού δωρεάν στο κοινό, αλλά να δώσει πλεονέκτημα σε αυτούς που θα πληρώσουν για αυτό.

Έτσι μέσω μικρο-συναλλαγών της τάξεως των 5\$ και πάνω, που γίνονται μέσα από το παιχνίδι, ένας χρήστης ανάλογα με το ποσό που θέλει να δώσει, έχει πλεονέκτημα έναντι κάποιου άλλου, που εξακολουθεί να παίζει το παιχνίδι δωρεάν. Και οι δύο μπορούν να το τερματίσουν αλλά, ο χρήστης που δεν θα πληρώσει, θα χρειαστεί παραπάνω ώρες και αυτό αποτελεί δικλείδα διαχωρισμού μεταξύ τους. Τα παιχνίδια αυτά, στην πλειοψηφία τους, προτρέπουν τον χρήστη να κάνει μία αγορά για να συνεχίσει την στιγμή που το επιθυμεί, ώστε να παραμείνει το παιχνίδι διασκεδαστικό για αυτόν παραπάνω ώρες και αυτό αποτελεί δικλείδα διαχωρισμού μεταξύ τους. Τα παιχνίδια αυτά, στην πλειοψηφία τους, προτρέπουν τον χρήστη να κάνει μία αγορά για να συνεχίσει την στιγμή που το επιθυμεί, ώστε να παραμείνει το παιχνίδι διασκεδαστικό για αυτόν παραπάνω ώρες και αυτό αποτελεί δικλείδα διαχωρισμού μεταξύ τους. Τα παιχνίδια αυτά, στην πλειοψηφία τους, προτρέπουν τον χρήστη να κάνει μία αγορά για να συνεχίσει την στιγμή που το επιθυμεί, ώστε να παραμείνει το παιχνίδι διασκεδαστικό για αυτόν παραπάνω ώρες και αυτό αποτελεί δικλείδα διαχωρισμού μεταξύ τους. Τα παιχνίδια αυτά, στην πλειοψηφία τους, προτρέπουν τον χρήστη να κάνει μία αγορά για να συνεχίσει την στιγμή που το επιθυμεί, ώστε να παραμείνει το παιχνίδι διασκεδαστικό για αυτόν μπορούν να το τερματίσουν αλλά, ο χρήστης που δεν θα πληρώσει, θα χρειαστεί



Μενού αγοράς εικονικού χρήματος στα F2P παιχνίδια

Η παραπάνω εικόνα αφορά τιμολόγιο αγοράς εικονικού χρήματος, σε αυτήν την περίπτωση Meteor Credits, του free-to-play παιχνιδιού Hawken της εταιρείας Meteor Entertainment, το οποίο είναι είδους shoot 'em up (πρώτου προσώπου). Ο κάθε χρήστης, μπορεί να παίξει το παιχνίδι δωρεάν, αλλά για να αυξήσει τις πιθανότητες του κάνοντας το πιο εύκολο, η Meteor του δίνει προσφορές με αγορά εικονικού χρήματος για ξεκλείδωμα και απόκτηση περεταίρω λειτουργιών (πανοπλία του ήρωα, καμουφλάζ, όπλα κ.τ.λ.).

Δίνοντας 5\$ πραγματικό χρήμα, ο παίκτης μπορεί να αγοράσει 720 μονάδες εικονικού χρήματος (Meteor Credits) και έτσι μπορεί να αποκτήσει π.χ. μία καινούρια πανοπλία, που να είναι πιο ανθεκτική σε χτυπήματα αντιπάλων, διαθέτοντας κάποια εικονικά χρήματα. Ανάλογα με το τι ποσό πραγματικού χρήματος διαθέσει ο χρήστης, του δίνονται μπόνους. Αγοράζοντας π.χ. 14.400 Meteor Credits με 100 \$, του δίνει μπόνους 20% και μπορεί να αποκτήσει 17.280 Meteor Credits τελικά.

Υπάρχει βέβαια, σε πάρα πολλά τέτοιου είδους παιχνίδια η συλλογή πόντων εμπειρίας (xp), που μπορεί να συλλέξει ο χρήστης σε κάθε επίπεδο ή πίστα, νικώντας. Εξαργυρώνοντας επίσης αυτούς τους πόντους μέσα στο παιχνίδι, δίνεται και πάλι η δυνατότητα στον χρήστη να ξεκλειδώσει ή/και να αγοράσει σπάνια αντικείμενα, καινούρια όπλα κ.α.



Επιβράβευση Του Χρήστη Με Πόντους XP Στο Επίπεδο 2 Του Παιχνιδιού Starcraft 2

Ο χρήστης, όπως φαίνεται στην παραπάνω εικόνα, επιβραβεύτηκε με 21.182 πόντους εμπειρίας (XP), που μπορεί να τους χρησιμοποιήσει στο συγκεκριμένο παιχνίδι π.χ. για επέκταση των στρατευμάτων, για να κάνει τα υπάρχοντα στρατεύματα του πιο ισχυρά ή αν τον φτάνουν, από αυτά που του ζητά η κάθε επιλογή και τα δύο.

Υπάρχει η δυνατότητα όμως, να πληρώσει με πραγματικό χρήμα ο παίκτης και να αγοράσει πόντους XP, με την ίδια διαδικασία που έχουν τα credits. Αυτό δεν σημαίνει ότι μόλις τους αγοράσει θα ανέβει επίπεδο, αλλά θα μπορεί να έχει τις υπόλοιπες επιλογές. Πολλοί χρήστες, ειδικά αν δεν μπορούν να βγάλουν εις πέρας ένα συγκεκριμένο επίπεδο, είτε μία συγκεκριμένη πίστα, αγοράζουν credits ή/και πόντους XP. Οι δύο αυτές μέθοδοι απόκτησης διευκολύνσεων όμως, ειδικά σε εθισμένους παίκτες που δεν έχουν μέτρο, τους δημιουργούν ακόμη και οικονομικά προβλήματα.

Παρακάτω παραθέτονται τα θετικά και τα αρνητικά στοιχεία των free-to-play (F2P) παιχνιδιών:

Θετικά των F2P:

Ο παίκτης μπορεί να παίξει το παιχνίδι για ένα μεγάλο διάστημα πριν δαπανήσει χρήματα για αυτό. Μπορούν να απολαύσουν ένα παιχνίδι περιοδικά, χωρίς να πληρώσουν κάποια μηνιαία συνδρομή, λόγω περιορισμένου χρόνου που μπορεί να έχει ο χρήστης σε κάποια χρονική στιγμή. Κατά κάποιο τρόπο δίνεται στον χρήστη ο έλεγχος των χρημάτων που θέλει να ξοδεύει για αυτό. Τα F2P είναι πιο πρακτικά, γιατί μπορεί ο χρήστης να παίξει όσα θέλει χωρίς να πληρώσει τίποτα ώσπου να φτάσει στην τελική απόφασή του.

Τα αρνητικά των F2P:

Τα F2P διαφέρουν δραματικά στην τιμολόγηση τους από τα κανονικά παιχνίδια, είναι δύσκολο να υπολογίσουμε ακριβώς πόσα θα πρέπει να δαπανηθούν επιπλέον, ώστε το παιχνίδι να παραμείνει απολαυστικό και ανταγωνιστικό ως προς τα άλλα. Οι μικρο-συναλλαγές που γίνονται στο παιχνίδι δεν είναι πάντα τόσο μικρές, υπάρχει κοινό που μπορεί να βρει ένα παιχνίδι πολύ συναρπαστικό και να παρασυρθούν από τον εθισμό τους, σε ποσά πολύ

παραπάνω από μια μηνιαία συνδρομή. Δηλαδή, ένα κανονικό παιχνίδι που κοστίζει 60\$, παρέχει στον παίκτη όλο το περιεχόμενο του τίτλου αυτούσιο. Από την άλλη μεριά στα F2P, ο παίκτης για να φτάσει το κανονικό παιχνίδι στο ίδιο υλικό, θα πρέπει να ξοδέψει πάνω από 500\$.

Τα F2P κινδυνεύουν να γίνουν Pay-to-win, δηλαδή οι παίκτες να πληρώσουν, ώστε να έχουν ανταγωνιστικό πλεονέκτημα ή να μπορούν να προηγηθούν στο παιχνίδι πιο γρήγορα, προσβάλλοντας έτσι όλους τους υπόλοιπους παίκτες που δεν έχουν την ίδια χρηματική ευχέρεια. Τα σχέδια πληρωμής ενός παιχνιδιού μπορεί να γίνουν πολύπλοκα, απατηλά και εύκολα να αυξηθούν, αν το παιχνίδι βρεθεί σε δυσχερή οικονομικά προβλήματα. Οι αποκλεισμένοι (banned) παίκτες, λόγω κακής συμπεριφοράς ή αλλοίωση του λογισμικού του παιχνιδιού (hacked), μπορούν πολύ εύκολα να δημιουργήσουν έναν άλλο λογαριασμό, χωρίς να χρειαστεί να ξαναπληρώσουν για τον τίτλο.

Ανακεφαλαίωση

Σε αυτή την πτυχιακή μιλήσαμε για τα πρώτα βήματα της βιομηχανίας παιχνιδιών. Κάναμε μια σύντομη ιστορική αναδρομή και είδαμε πως εξελίχθηκαν τα παιχνίδια από την τρίλιζα στην σημερινή τους μορφή. Αναλύσαμε όλα τα είδη παιχνιδιών που υπάρχουν και είδαμε κάποια από αυτά τα παιχνίδια. Μιλήσαμε για τους λόγους που επιλέξαμε το Unity και τις δυνατότητες που προσφέρει σε σχέση με τις άλλες game engines. Αναφέραμε πως το Unity έγινε αυτό που είναι σήμερα και περιγράψαμε τις λειτουργίες του unity editor. Μιλήσαμε για την γλώσσα που χρησιμοποιήθηκε για την δημιουργία του παιχνιδιού την C# και είδαμε κάποια πράγματα που χρειάζεται να γνωρίζει για τις γλώσσες προγραμματισμού. Είδαμε πως φτιάχτηκε το παιχνίδι και τι αντικείμενα χρησιμοποιήθηκαν για αυτό. Τέλος μιλήσαμε για τους τρόπους που μπορεί ο δημιουργός του παιχνιδιού να έχει έσοδα και τα θετικά και αρνητικά ενός free to play παιχνιδιού.

Επίλογος

Η Unity προσφέρει στον ενδιαφερόμενο “developer” την ευκαιρία να μπει στη θέση των δημιουργών διάφορων αγαπημένων του παιχνιδιών και να δημιουργήσει ο ίδιος τον αγαπημένο του χαρακτήρα, να του δώσει ζωή και να δημιουργήσει το δικό του φανταστικό περιβάλλον. Η Unity παρέχει οτιδήποτε χρειάζεται κάποιος για την ανάπτυξη ένα παιχνιδι, από οδηγούς στον ιστοχώρο της, μέχρι και απίστευτα χαρακτηριστικά και επιλογές στο πρόγραμμά της. Τέλος, η πολλαπλή επιλογή πλατφορμών για την κυκλοφορία των παιχνιδιών την κάνει πιο προσιτή αφού οι κονσόλες, οι υπολογιστές και γενικά μία συσκευή αναπαραγωγής παιχνιδιών δεν λείπει από τις ζωές των ανθρώπων, με τα βιντεοπαιχνίδια να αποτελούν μία από τις πιο δυνατές μορφές ψυχαγωγίες στις μέρες μας, όπου η τεχνολογία κιάλας αυξάνεται συνεχώς.

Βιβλιογραφία

- [1] Unity3d.com - <https://unity3d.com/> -
- [2] Forum Unity - <https://forum.unity.com/>
- [3] Unity Tutorials - <https://unity3d.com/learn/tutorials>
- [4] Wikipedia Unity Technologies - https://en.wikipedia.org/wiki/Unity_Technologies
- [5] A History of the Unity Game - Engine https://web.wpi.edu/Pubs/E-project/Available/E-project-030614-143124/unrestricted/Haas_IQP_Final.pdf
- [6] Κανάλι στο Youtube σχετικά με το Unity <https://www.youtube.com/channel/UCmtyQOKKmrMVaKuRXz02jbQ>
- [7] Κανάλι στο Youtube σχετικά με το Unity <https://www.youtube.com/user/Brackeys>
- [8] Ιστότοπος σχετικά με το Unity <http://pixelnest.io/tutorials/2d-game-unity/>

[9] ιστότοπος σχετικά με το Unity <https://gamedevelopment.tutsplus.com/>

[10] εισαγωγή στη C#- <http://studentguru.gr/w/tutorials/01-c>

[11] το κανάλι της Unity στο Youtube

https://www.youtube.com/channel/UCG08EqOAXJk_YXPDsAvReSg

Παράρτημα

```
Infinity Knight runner -enemys - 1.1 - Copy - chase Update()
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class chase : MonoBehaviour {
7
8     public Transform player;
9     static Animator anim;
10    [SerializeField] private string loadLevel;
11
12    // Use this for initialization
13    void Start () {
14        anim = GetComponent();
15    }
16
17    // Update is called once per frame
18    void Update () {
19        Vector3 direction = player.position - this.transform.position;
20        float angle = Vector3.Angle(direction, this.transform.forward);
21        if (Vector3.Distance(player.position, this.transform.position) < 20 && angle < 90)
22        {
23
24            direction.y = 0;
25
26            this.transform.rotation = Quaternion.Slerp(this.transform.rotation, Quaternion.LookRotation(direction), 0.1f);
27
28
29            anim.SetBool("isWalking", false);
30            if (direction.magnitude > 5)
31            {
32                this.transform.Translate(0,0,0.07f);
33                anim.SetBool("isAttacking",true);
34            }
35            else
36            {
37                anim.SetBool("isWalking", true);
38                anim.SetBool("isAttacking",false);
39            }
40        }
41    }
42
43    private void OnTriggerEnter(Collider other)
44    {
45        if (other.CompareTag("die"))
46        {
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
```

```
50 private void OnTriggerEnter(Collider other)
51 {
52     if (other.CompareTag("die"))
53     {
54         SceneManager.LoadScene(loadLevel);
55     }
56 }
57
58
59
60
61
```

```
DeathZone.cs | chase.cs* | Score.cs | roadTrigger.cs | running.cs |
infinity knight runner - enemies - 1.1 - Copy | DeathZone | LoadLevel
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class DeathZone : MonoBehaviour {
7
8     // Use this for initialization
9     [SerializeField] private string LoadLevel;
10
11     private void OnTriggerEnter(Collider other)
12     {
13         if (other.CompareTag("die"))
14         {
15             SceneManager.LoadScene(LoadLevel);
16         }
17     }
18
19 }
20
21
```

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class MainMenu : MonoBehaviour {
7
8      public void PlayGame ()
9      {
10         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
11     }
12
13     public void QuitGame ()
14     {
15         Application.Quit();
16     }
17
18 }
19
```

```
running.cs* X
infinity knight runner -enemys - 1.1 - Copy  running
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class running : MonoBehaviour {
7
8      private Rigidbody rb;
9      public float speed = 5f;
10
11
12     void Start()
13     {
14         rb = GetComponent<Rigidbody>();
15     }
16
17
18     void Update()
19     {
20         rb.velocity = new Vector3(Input.acceleration.x*5f, rb.velocity.y, speed);
21     }
22
23
24
25
26 }
```