

Ελληνικό Μεσογειακό Πανεπιστήμιο

Σχολή Μηχανικών

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Πρόγραμμα Σπουδών Μηχανικών Πληροφορικής

Πτυχιακή Εργασία

Τίτλος: Development of a round-based zombie survival game με Unity3D

Ονοματεπώνυμο: Χατζηδάκης Κωνσταντίνος
Επιβλέπων Εκπαιδευτικός: Παχουλάκης Ιωάννης
Επιτροπή Αξιολόγησης:
Ημερομηνία παρουσίασης:

Ηράκλειο, 2024

Abstract

This thesis concerns a creation of a video game due to a personal interest in the gaming section. More specifically, i wanted to focus on how an electronic video game can be created and operated. How can we connect the different elements given to us for a game, with the possible codes in order to get a functional and user-friendly result. The central idea of this thesis is the creation of a first-person survival game. The player's goal is to survive as many rounds as possible with weapons given to him while the opponents he encounters consist of numerous hordes of zombies (SombieZ). Some of the tools that were used that interested me were Unity3D and C#. The first one is a software development kit for building video games while the other is a programming language that is highly compatible with Unity3D.

Σύνοψη

Η παρούσα πτυχιακή εργασία αφορά τη δημιουργία ενός βιντεοπαιχνιδιού λόγω προσωπικού ενδιαφέροντος για τον τομέα του gaming. Πιο συγκεκριμένα ήθελα να εστιάσω στο πως μπορεί να δημιουργηθεί και να λειτουργήσει ένα ηλεκτρονικό βιντεοπαιχνίδι. Με ποιο τρόπο μπορούμε να συνδέσουμε τα διαφορετικά στοιχεία που μας δίνονται για ένα παιχνίδι, με τους πιθανούς κώδικες ώστε να βγει κάποιο λειτουργικό αποτέλεσμα και ευχάριστο στο χρήστη. Κεντρική ιδέα της πτυχιακής εργασίας είναι η δημιουργία ενός παιχνιδιού επιβίωσης πρώτου προσώπου. Σκοπός του παίκτη είναι να επιβιώσει όσο δυνατόν περισσότερους γύρους με τα όπλα που το δίνονται, ενώ οι αντίπαλοι που συναντά αποτελούνται από πολυάριθμες ορδές zombies (SombieZ). Κάποια από τα εργαλεία που χρησιμοποιήθηκαν και μου κίνησαν το ενδιαφέρον ήταν το UNITY3D και η χρήση της γλώσσας C#. Το πρώτο αποτελεί πλαίσιο ανάπτυξης λογισμικού για την κατασκευή βιντεοπαιχνιδιών ενώ το δεύτερο είναι μια γλωσσά προγραμματισμού η οποία έχει υψηλή συμβατότητα με το UNITY3D.

Περιεχόμενα:

Abstract..... σ. 3
Σύνοψη..... σ. 4
Περιεχόμενα..... σ. 5
Λίστα εικόνων..... σ. 6
Κεφάλαιο 1: Εισαγωγή..... σ. 8
1.1. Περίληψη..... σ. 8
1.2 Κίνητρα/ Στόχοι..... σ. 8
1.3 Δομή Πτυχιακής Εργασίας..... σ. 9
Κεφάλαιο 2: Μεθοδολογία Υλοποίησης..... σ. 10
2.1 Μέθοδος Ανάλυσης και Ανάπτυξης..... σ. 10
Κεφάλαιο 3: Θεωρητικό Υπόβαθρο..... σ. 11
3.1 Ψηφιακά Παιχνίδια..... σ. 11
3.2 Ιστορική Αναδρομή Ψηφιακών Παιχνιδιών..... σ. 12
3.3 Κατηγορίες Ψηφιακών Παιχνιδιών..... σ. 15
3.4 Στοιχεία Ψηφιακών Παιχνιδιών..... σ. 18
3.5 Game Engine Unity..... σ. 19
3.5.1 C# (C Sharp)..... σ. 20
Κεφάλαιο 4: Υλοποίηση..... σ. 21
4.1 Μηχανισμοί του παιχνιδιού..... σ. 21
4.2 SombieZ..... σ. 21
4.2.1 Κεντρικό Μενού..... σ. 21
4.2.2 User Interface (UI)..... σ. 24
4.2.3 Weapons (Όπλα)..... σ. 26
4.2.4 Weapon Switch (Εναλλαγή των όπλων)..... σ. 30
4.2.5 Zombies..... σ. 32
4.2.6 Game Management..... σ. 38
4.2.7 Pick Up Items (AmmoBox)..... σ. 40
4.2.8 Player/ Character..... σ. 42
4.2.9 Κάμερα..... σ. 48
4.2.10 Map/ Χάρτης..... σ. 49
Βιβλιογραφία..... σ. 51
Παραρτήματα.....

Λίστα Εικόνων:

Image 1: Tennis For Two σε παλμογράφο
Image 2: Παιχνιδομηχανή Arcade
Image 3: GameBoy
Image 4: Κονσόλα Playstation 2
Image 5: “Mortal Kombat”
Image 6: “The Last Of Us”
Image 7: “Fifa”
Image 8: “Call Of Duty”
Image 9: Λογότυπο της μηχανής παιχνιδιών Unity
Image 10: Κεντρικό μενού SombieZ
Image 11: Κεντρικό μενού: Canvas
Image 12: Κεντρικό μενού: Μουσική (Background music)
Image 13: Κεντρικό μενού: Κώδικας
Image 14: UI: Canvas
Image 15: UI: Κώδικας No. 1
Image 16: UI: Κώδικας No. 2
Image 17: UI: Κώδικας στο παιχνίδι
Image 18: Κώδικας όπλων No. 1
Image 19: Κώδικας όπλων No. 2
Image 20: Κώδικας όπλων No. 3
Image 21: Κώδικας για γέμισμα όπλου
Image 22: Καραμπίνα
Image 23: Χαρακτηριστικά καραμπίνας
Image 24: Αυτόματο (AK-47)
Image 25: Χαρακτηριστικά αυτόματου (AK-47)
Image 26: Πιστόλι
Image 27: Χαρακτηριστικά πιστολιού
Image 28: Κώδικας εναλλαγής όπλων
Image 29: Σειρά τοποθέτησης όπλων
Image 30: Μοντέλο ζόμπι
Image 31: Σύστημα πλοήγησης (navigation system)
Image 32: Rigidbody & Capsule Collider
Image 33: Χαρακτηριστικά ζόμπι
Image 34: Εναλλαγή animation
Image 35: Κώδικας ζόμπι No. 1
Image 36: Κώδικας ζόμπι No. 2
Image 37: Κώδικας ζόμπι No. 3
Image 38: Κώδικας ζόμπι No. 4
Image 39: Κώδικας ζόμπι No. 5
Image 40: Κώδικας Game Management No. 1
Image 41: Κώδικας Game Management No. 2

Image 42: Κώδικας Game Management No. 3
Image 43: AmmoBox
Image 44: Κώδικας Pick Up Item No. 1
Image 45: Κώδικας Pick Up Item No. 2
Image 46: Κώδικας Pick Up Item No. 3
Image 47: Ιεραρχία του χαρακτήρα
Image 48: Χαρακτηριστικά του χαρακτήρα No. 1
Image 49: Χαρακτηριστικά του χαρακτήρα No. 2
Image 50: Rigidbody του χαρακτήρα
Image 51: Κώδικας χαρακτήρα No. 1
Image 52: Κώδικας χαρακτήρα No. 2
Image 53: Κώδικας χαρακτήρα No. 3
Image 54: Κώδικας χαρακτήρα
Image 55: Κώδικας μέτρησης της ζωής No. 1
Image 56: Κώδικας μέτρησης της ζωής No. 2
Image 57: Κώδικας θανάτου
Image 58: Κώδικας Game Over
Image 59: Game Over View
Image 60: Χαρακτηριστικά κάμερας
Image 61: Map view No. 1
Image 62: Map view No. 2
Image 63: Map view No. 3

Κεφάλαιο 1: Εισαγωγή

1.1 Περίληψη

Η παρούσα πτυχιακή εργασία αφορά την ανάπτυξη ενός παιχνιδιού κατηγορίας βολών πρώτου προσώπου με τη χρήση της μηχανής Unity. Η Unity αποτελεί ένα εξειδικευμένο λογισμικό σχεδίασης ψηφιακών παιχνιδιών, το οποίο παρέχει χρήσιμα εργαλεία και χρησιμοποιείται ευρύτερα από σχεδιαστές ψηφιακών παιχνιδιών. Αυτή, έχει τη δυνατότητα ανάπτυξης 2D και 3D γραφικών. Ακόμα, χρησιμοποιήθηκε η γλώσσα προγραμματισμού C#, μία αντικειμενοστραφής γλώσσα, για την δημιουργία των scripts, απαραίτητα για την τελειοποίηση του παιχνιδιού. Το ενδιαφέρον εστίασε στην ανάπτυξη του κώδικα με τη χρήση εξειδικευμένου λογισμικού και σκοπός η σύνθεση των διαφόρων στοιχείων για την ανάδειξη ενός ολοκληρωμένου αποτελέσματος.

Το παιχνίδι όπως αναφέρθηκε αφορά κατηγορίας βολών πρώτου προσώπου, όπου ο παίκτης προσπαθεί να ξεφύγει από ζόμπι που τον κυνηγάνε και να ανέβει επίπεδο. Στην εργασία αναφέρεται η μεθοδολογία υλοποίησης, στην αναζήτηση βιβλιογραφίας και στους στόχους και τα κίνητρα της. Φυσικά, γίνεται εκτενής αναφορά για την δημιουργία του παιχνιδιού και των εργαλείων που χρησιμοποιήθηκαν. Ακόμα, γίνεται αναφορά στο ορισμό των βιντεοπαιχνιδιών, στην ιστορία και τις κατηγορίες τους, στη μηχανή παιχνιδιών και γλώσσα προγραμματισμού που χρησιμοποιήθηκαν και στα μέρη του κώδικα.

Λέξεις κλειδιά: Ψηφιακά παιχνίδια, Unity, C#, παιχνίδια βολών πρώτου προσώπου.

1.2 Κίνητρα / Στόχοι

Αυτό που αποτέλεσε έναυσμα για την συγγραφή αυτής της εργασίας είναι το πως δημιουργείτε ένα ψηφιακό παιχνίδι από το μηδέν. Τα ψηφιακά παιχνίδια αποτελούν πλέον δραστηριότητες και μέρη της καθημερινής ζωής του σύγχρονου κόσμου. Αυτό, καθώς προσφέρει ψυχαγωγία στα άτομα και στις ομάδες, αποτελεί μέσο κοινωνικοποίησης ατόμων από διαφορετικά σημεία του πλανήτη, χρησιμοποιούνται ως μέσα για μάθηση κ.α. Πως, λοιπόν, ένα τόσο ευρύ διαδεδομένο μέσο και στοιχείο δημιουργείτε; Ποιες είναι οι ιδιαιτερότητες που κρύβει στη διαδικασία ανάπτυξης του; Επίσης, υπήρχε ενδιαφέρον για την ανάπτυξη αυτών, πότε ξεκίνησαν να κυκλοφορούν, ποια είναι η ιστορία των ψηφιακών παιχνιδιών και φυσικά τι αποτελεί ψηφιακό παιχνίδι. Αυτά και άλλα ερωτήματα είναι που με παρακίνησαν να ασχοληθώ με την ανάπτυξη αυτής της εργασίας.

Έπειτα, για την ανάπτυξη ψηφιακών παιχνιδιών είναι απαραίτητη η χρήση μηχανών και κάποιας γλώσσας προγραμματισμού. Έτσι, ένα επιπλέον ερώτημα που δημιουργείτε αφορά τη χρήση αυτών και τα διάφορα χαρακτηριστικά αυτών. Τι ακριβώς είναι αυτό που περιέχει μία μηχανή παιχνιδιών, ποιες είναι οι λειτουργίες και τα εργαλεία που αυτή προσφέρει και πως αυτά είναι χρήσιμα; Τέλος, πως οι διάφορες γλώσσες προγραμματισμού συμβάλλουν στην ανάπτυξη ψηφιακών παιχνιδιών, με ποιο τρόπο χρησιμοποιούνται και

επιλέγονται ως κατάλληλες για αυτά;

Στόχος της παρούσας εργασίας καταλήγει να είναι η επιτυχής ανάπτυξη ενός ψηφιακού παιχνιδιού με τη χρήση των παραπάνω, της μηχανής παιχνιδιών Unity και της γλώσσας προγραμματισμού C#, αναφερόμενος πρώτα σε ένα θεωρητικό υπόβαθρο αυτού.

1.3 Δομή Πτυχιακής Εργασίας

Η παρούσα πτυχιακή χωρίστηκε σε τέσσερα κεφάλαια όπου καθένα από αυτά έχει κάποια ύπο-κεφάλαια. Αρχικά υπάρχουν η σύνοψη, τα περιεχόμενα και η λίστα εικόνων, ενώ μετά ακολουθούν τα κεφάλαια που περιέχουν το κυρίως θέμα της εργασίας.

Το πρώτο κεφάλαιο αφορά την εισαγωγή του αναγνώστη στο θέμα της εργασίας. Μία σύντομη περίληψη του τι πρόκειται να διαβάσει στις επόμενες σελίδες, αναφέροντας εν συντομία το παιχνίδι που δημιουργήθηκε, κάποιους μηχανισμούς, η κατηγορία στην οποία ανήκει. Έπειτα, γίνεται αναφορά στα κίνητρα και τους στόχους που κίνησαν το ενδιαφέρον για την ενασχόληση με αυτού του είδους εργασία, ενώ τέλος αναφέρεται συνοπτικά η δομή των λοιπών κεφαλαίων.

Στο δεύτερο κεφάλαιο ο αναγνώστης θα δει την μεθοδολογία που ακολουθήθηκε για την έρευνα, συγγραφή και ανάπτυξη του αντικειμένου. Γίνεται αναφορά στην βιβλιογραφία και την αναζήτηση αυτή και στη μελέτη των εργαλείων που χρησιμοποιήθηκαν.

Στο αμέσως επόμενο κεφάλαιο, το ενδιαφέρον εστιάζει στο θεωρητικό υπόβαθρο της εργασίας. Η πρώτη αναφορά του κεφαλαίου αυτού έχει να κάνει με τον ορισμό των ψηφιακών παιχνιδιών, τι αποτελεί στην ουσία ένα ψηφιακό παιχνίδι. Ακόμα, ποια είναι η αναγνώριση των ψηφιακών παιχνιδιών στο κοινό και ποια η θέση τους στην καθημερινή ζωή των ατόμων. Στη συνέχεια, ο αναγνώστης θα μπορεί να διαβάσει μία ιστορική αναδρομή των ψηφιακών παιχνιδιών από την περίοδο που κυκλοφόρησαν για πρώτη φορά έως και σήμερα, με σύντομες αναφορές σε παιχνίδια και εταιρείες του παρελθόντος και του σήμερα. Ακολουθεί μία περιγραφή των πιο συχνών κατηγοριών ψηφιακών παιχνιδιών, όπως φυσικά και της κατηγορίας παιχνιδιού που αναπτύχθηκε (παιχνίδι βολών πρώτου προσώπου). Ακόμα, περιγράφονται κάποια απαραίτητα στοιχεία για την ανάπτυξη ψηφιακών παιχνιδιών.

Στο ίδιο κεφάλαιο γίνεται αναφορά στα εργαλεία που χρησιμοποιήθηκαν, όπου πρώτα φαίνεται η μηχανή παιχνιδιών της Unity ενώ αμέσως μετά η γλώσσα προγραμματισμού C#.

Το τελευταίο κεφάλαιο εστιάζει στο παιχνίδι που αναπτύχθηκε, ονόματι SombieZ. Πρώτα γίνεται μία περιγραφή των λειτουργιών που διαθέτει το παιχνίδι αρά και ποιες είναι οι κινήσεις που ο παίκτης μπορεί να κάνει. Στη συνέχεια γίνεται μία εκτενής αναφορά σε κάθε ξεχωριστό στοιχείο της ανάπτυξης. Γίνεται αναφορά στο κεντρικό μενού του παιχνιδιού, στο Ui, στον χαρακτήρα και τους αντιπάλους, τα όπλα, το χάρτη κ.α., ενώ αυτά συνοδεύονται και από εικόνες/ στιγμιότυπα του παιχνιδιού. Τέλος, ακολουθεί αναλυτικά η βιβλιογραφία.

Κεφάλαιο 2: Μεθοδολογία Υλοποίησης

2.1 Μέθοδος ανάλυσης και ανάπτυξης

Η ανάπτυξη της πτυχιακής εργασίας και του ίδιου του παιχνιδιού αποτελεί μία μεγάλη διαδικασία. Βασικό μέλημα ήταν η αναζήτηση επαρκούς βιβλιογραφίας και πηγών για μελέτη του αντικειμένου. Έτσι, οι πρώτες κινήσεις αφορούσαν στην αναζήτηση αυτή, αντίστοιχες εργασίες, μελέτες και αναφορές του αντικειμένου. Αποκτήθηκε μία θεωρητική γνώση ως προς τα ψηφιακά παιχνίδια, το ιστορικό τους υπόβαθρο, τις κατηγορίες, τις διάφορες χρήσεις τους κ.α. , ενώ στη συνέχεια έγινε εκτενής ανάλυση αυτών των πτυχών.

Έπειτα, η έρευνα εστίασε στην μελέτη των εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη του ψηφιακού παιχνιδιού. Πρώτα, η μελέτη ξεκίνησε στη μηχανή της Unity, με μία περιήγηση στη πλατφόρμα της αναζητώντας χρήσιμα εργαλεία και έτοιμα μοντέλα από το Unity Asset Store. Φυσικά, απαραίτητο ήταν και η εκμάθηση της γλώσσας προγραμματισμού C#, η βασική γλώσσα που χρησιμοποιεί η Unity. Ακολούθησε η ανάπτυξη του παιχνιδιού, έπειτα ο έλεγχος ορθότητας του και τέλος η αναλυτική συγγραφή της παρούσας εργασίας.

Κεφάλαιο 3: Θεωρητικό Υπόβαθρο

3.1 Ψηφιακά Παιχνίδια

Για να ορίσουμε ένα ψηφιακό παιχνίδι/ Βιντεοπαιχνίδι πρέπει πρώτα να ορίσουμε την έννοια του παιχνιδιού. Ένα παιχνίδι αποτελεί μία δραστηριότητα με κανόνες, στόχους, ρόλους και επιδιωκόμενα αποτελέσματα. Μπορεί να συμμετέχει ένα άτομο, ενώ άλλα παιχνίδια απαιτούν περισσότερα μέλη, περισσότερους παίκτες και διάφορες μορφές αλληλεπίδρασης (συνεργασία, άμυνα, ανταγωνισμό κ.α.). Ακόμα, οι Ernest Adams και Andrew Rollings αναφέρονται στο παιχνίδι ως ένα σύμπλεγμα προκλήσεων που εντάσσεται σε ένα περιβάλλον προσομοίωσης (Τέρπος, χ.χ.). Έτσι, ένα ψηφιακό παιχνίδι αφορά μία τέτοια δραστηριότητα με τη διαφορά ότι αυτή βρίσκεται σε ένα εικονικό περιβάλλον. Απαιτεί ένα πρόγραμμα λογισμικού όπου ο παίκτης ελέγχει του πόρους που του δίνονται στο παιχνίδι, λαμβάνει αποφάσεις με το χειρισμό ηλεκτρονικών μέσων για την εκπλήρωση ενός σκοπού (Σουσαμλής, 2017). Τέλος, απαιτεί συσκευές οι οποίες μπορούν να υποστηρίξουν δισδιάστατα ή τρισδιάστατα γραφικά.

Ένα βιντεοπαιχνίδι αφορά ένα σύστημα εισόδου και εξόδου, όπου η είσοδος αναφέρεται στην εισαγωγή δεδομένων και πληροφοριών με τη χρήση της κονσόλας, πληκτρολογίου, οθόνης κ.α., ενώ το σύστημα εξόδου αναφέρεται στην αισθητηριακή πληροφορία που δέχεται το άτομο μέσω του ήχου (ηχεία) και της εικόνας (οθόνη), όπου επιτυγχάνει ένα υπαρκτό βίωμα του ατόμου στο παιχνίδι. Αυτό αποτελεί τη αλληλεπίδραση του εμπλεκόμενου ατόμου με το παιχνίδι, καθώς το ένα τροφοδοτεί το άλλο. Χρειάζεται να έχει καλή αισθητική, φαντασία, πρόκληση, να είναι διασκεδαστικό είτε να χρησιμοποιείται ως μέσο για μάθηση. Ένα ψηφιακό παιχνίδι ανάλογα με τα γραφικά στοιχεία, το ρεαλισμό που διαθέτει, την αισθητηριακή ή νοητική διέγερση που προκαλεί μπορεί να ταυτίσει το άτομο που παίζει με το ρόλο που του δίνεται σε διάφορα επίπεδα, μικρή ή μεγάλη ταύτιση και κατά συνέπεια, μικρή ή μεγάλη ευχαρίστηση. Τέλος, αποτελείται από εντολές που γίνονται πράξη μέσω των χειρισμών του ατόμου σε αυτό, μπορεί να περιέχει στοιχεία του πραγματικού κόσμου ή στοιχεία του μη πραγματικού, όπως στοιχεία εμπνευσμένα από το μυθικό λόγο, το παραμύθι, τη φαντασία. Στα ψηφιακά παιχνίδια δεν απαιτείται φυσική εμπλοκή του ατόμου όπως ένα φυσικό παιχνίδι (λ.γ. επιτραπέζια), ενώ ακόμα και σε αυτά, υπάρχει η δυνατότητα χρήσης από έναν ή περισσότερους παίκτες. Ιδιαίτερα με τη χρήση του διαδικτύου, πλέον, υπάρχει η δυνατότητα διασύνδεσης μεταξύ ατόμων από διαφορετικά σημεία, χωρίς την ανάγκη να βρίσκονται στον ίδιο χώρο.

Τα βιντεοπαιχνίδια για κάποιους μπορεί να θεωρηθούν ως κάποια νέα μορφή τέχνης, με την ανάπτυξη της τεχνολογίας και των νέων μορφών δημιουργίας ενώ το άτομο παράγει και συμμετέχει ταυτόχρονα σε αυτή τη διαδικασία. Ακόμα, μπορούν να αποτελέσουν μέσω αφήγησης με διττό τρόπο: είτε μία αφήγηση που προέρχεται από τα μάτια του δημιουργού, είτε την αφήγηση μέσα από τα μάτια του παίκτη. Έτσι ένα βιντεοπαιχνίδι μπορεί να κατέχει πολλά διαφορετικά νοήματα, εξαρτώμενο από διαφορετικές οπτικές. Φυσικά, η έννοια του παιχνιδιού και το τι πραγματικά αποτελεί παιχνίδι αλλάζει με βάση το κοινωνικό και πολιτισμικό πλαίσιο στο οποίο διαδραματίζεται. Ενώ, υπάρχουν περιπτώσεις όπου σε πλευρές της καθημερινότητας υπάρχουν χαρακτηριστικά παιχνιδιών, όπως η συλλογή πόντων στο

super market ή τα like στα μέσα κοινωνικής δικτύωσης, που όμως δεν αποτελούν παιχνίδι. Η συλλογή πόντων και η ποσοτικοποίηση αυτή αποτελεί χαρακτηριστικό των παιχνιδιών και χρησιμοποιείται ως μέσο παιχνιδοποίησης πλευρών της καθημερινής ζωής ώστε να προκαλούνται αισθήματα ψυχαγωγίας, ανταγωνισμού, επίτευξης κ.α.

3.2 Ιστορική Αναδρομή Ψηφιακών Παιχνιδιών

Το πρώτο ψηφιακό παιχνίδι έκανε την εμφάνιση του κατά τα τέλη της δεκαετίας του ‘40 από τους T. Goldsmith Jr. και Estle Ray Mann που δημιούργησαν τη “Συσκευή Διασκέδασης Καθοδικού Σωλήνα”, όπου οι παίκτες καλούνταν να πετύχουν ένα στόχο, εμπνευσμένοι από την τεχνολογία που χρησιμοποιούσαν τα ραντάρ και αφορούσε την προσομοίωση πυραύλων. Το αμέσως επόμενο ήρθε από τον A.S. Douglas κατά το έτος 1952, με την ονομασία “OXO”, μία μορφή του παιχνιδιού “Τρίλιζα”, όπου αποτέλεσε το πρώτο ψηφιακό παιχνίδι με γραφικά. Η ιδιαιτερότητα αυτού του παιχνιδιού ήταν η δυνατότητα του ατόμου να παίξει ενάντια του υπολογιστή. Ακόμα, άξιο αναφοράς είναι το παιχνίδι “Tennis for Two” που δημιουργήθηκε το 1958 από τον William Higinbotham σε παλμογράφο (Κρέζιας, 2016).

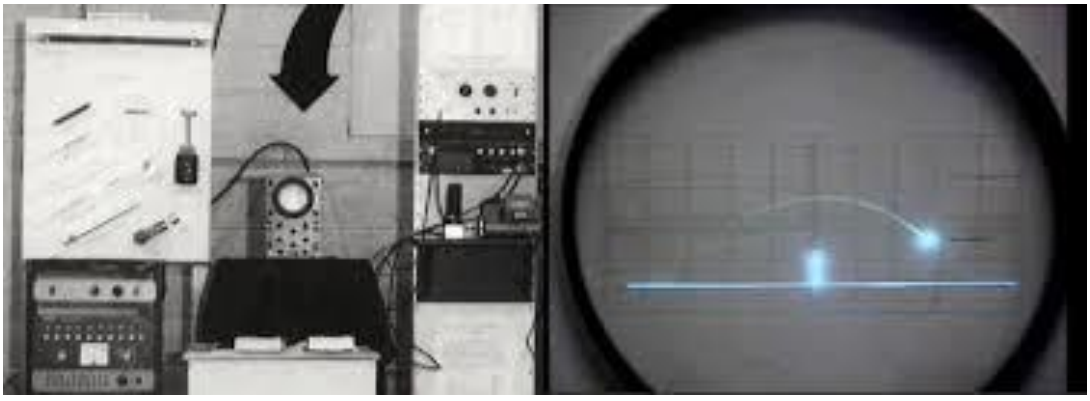


Image 1: Tennis For Two σε παλμογράφο.

Τα πρώτα παιχνίδια που κυκλοφόρησαν και δόθηκε πρόσβαση στο ευρύτερο κοινό ήταν με τις παιχνιδομηχανές Arcade, κατά τις δεκαετίες του ‘70 και ‘80, όπου αποτέλεσαν και ένα από τα βασικότερα μέσα ψυχαγωγίας σε Ευρώπη, Αμερική και Ιαπωνία. Οι παιχνιδομηχανές Arcade λειτουργούσαν με κέρματα, είχαν ενσωματωμένη οθόνη και χειριστήριο, γνώρισαν μεγάλη άνθιση κατά τα έτη 1978 έως 1982, που αποτέλεσε και τη χρυσή εποχή για αυτά. Βασικό χαρακτηριστικό τους ήταν η ηλεκτρονική μουσική και τα γραφικά, με συνέπεια να αποτελέσουν σημαντικό κομμάτι της pop κουλτούρας. Οι παιχνιδομηχανές αυτές βρίσκονταν πλέον παντού, από καταστήματα ψηλικών μέχρι εμπορικά κέντρα. Συγκεκριμένα, το 1971 έχουμε το πρώτο Arcade παιχνίδι από τους Nolan Bushnell και Ted Dabney ονόματι “Computer Space” βασισμένο στο “SpaceWar του Steve Russel. Κατά τη χρυσή εποχή των Arcade Games έχουμε τα: Space Invaders της Taito, το Asteroids και Luna Lander της Atari, Pac-Man και Ms Pac-Man της Namco και το Donkey Kong της Nitendo, ενώ οι πωλήσεις έφτασαν τα 900 εκατομμύρια δολάρια.



Image 2: Παιχνιδομηχανή Arcade.

Το 1985 κυκλοφόρησαν από τον Alexey Pajitnov το παιχνίδι Tetris και από την Nitendo το Super Mario, δύο παιχνίδια με παγκόσμια επιτυχία, τα οποία αναγνωρίζονται και απασχολούν τον κόσμο ακόμα και σήμερα. Στα τέλη του 1988 υποδέχθηκαν τη νέα εταιρεία Sega με την κονσόλα Mega Drive/ Genesis. Το 1989 έχουμε από την Nitendo την παιχνιδοκονσόλα χειρός GameBoy, μία κονσόλα όπου για πρώτη φορά μπορούσε να χρησιμοποιείται εκτός σπιτιού και χωρίς ρεύμα. Για να λειτουργήσει χρειάζονταν μπαταρίες, με ενσωματωμένο χειριστήριο και οθόνη. Από τη Sony το παιχνίδι Sonic The Hedgedog που γνώρισαν εξίσου μεγάλη επιτυχία (Πανανός, 2022).


Image 3: GameBoy

Στις επόμενες δεκαετίες τα Arcade παιχνίδια ξεκίνησαν να εγκαταλείπονται και αναδύθηκε η χρήση ηλεκτρονικών υπολογιστών και κονσόλων, ενώ πλέον δόθηκε η δυνατότητα οικιακής χρήσης, συνεπώς η διασκέδαση και ψυχαγωγία έλαβαν μέρος στον προσωπικό χώρο. Κυκλοφόρησαν τότε κονσόλες όπως Playstation από τη Sony το 1995, Nitendo και Xbox, ενώ με την εμφάνιση του internet κυκλοφόρησαν παιχνίδια με τη δυνατότητα πολλαπλών παικτών από διαφορετικούς χώρους. Οι κονσόλες αυτές ήταν οι πρώτες που είχαν τη δυνατότητα να τρέξουν τρισδιάστατα παιχνίδια. Έγιναν, επίσης, δημοφιλή τα παιχνίδια που βασίζονταν σε ταινίες ή παιχνίδια με πιο σοβαρό περιεχόμενο (Σπέντζας, 2015).

Κατά την επόμενη δεκαετία, του 2000, δημιουργήθηκε πρώτα η κονσόλα Sega Dreamcast, η οποία όμως δεν γνώρισε επιτυχία με συνέπεια η Sega να χάσει χώρο από την αγορά. Η επόμενη κονσόλα την δεκαετίας εκείνης ήταν το Playstation 2, όπου σε αντίθεση με τη Sega, είχε μεγάλη αναγνώριση καθώς ήταν η πρώτη κονσόλα που δέχονταν DVD, με

χαμηλό κόστος και περισσότερο εξελιγμένα γραφικά. Ακολούθησε η Nitendo με το GameCube όπου χρησιμοποιούσε [1]οπτικούς δίσκους. Κατά το 2005 εμφανίζεται για πρώτη φορά κονσόλα με ασύρματα χειριστήρια, το Xbox360 και ένα χρόνο αργότερα κυκλοφόρησε και το Playstation 3με παρόμοια χαρακτηριστικά, προσθέτοντας τη δυνατότητα προβολής ταινιών. Τέλος, ξεκίνησαν τότε με μεγάλη επιτυχία τα παιχνίδια για κινητά τηλέφωνα, τα οποία πήραν τη θέση κονσολών χειρός όπως το Nitendo DS ή το Sony PSP.



Image 4: Κονσόλα Playstation 2

Από τη δεκαετία του 2010 και μετά, η βιομηχανία των βιντεοπαιχνιδιών αναπτύχθηκε σε μεγάλο βαθμό, καθιστώντας τα βιντεοπαιχνίδια ένα βασικό μέσο ψυχαγωγίας στον κόσμο για μικρούς και μεγάλους. Σε αυτό συνέβαλε η χρήση του internet και η δυνατότητα παικτών να επικοινωνήσουν μέσω του ίδιου παιχνιδιού σε διαφορετικούς χώρους. Ακόμα, ύστερα από την επιτυχία του Nitendo Wii το 2007, δόθηκε έμφαση σε περισσότερο αληθοφανή παιχνίδια με τη χρήση της κίνησης και αργότερα της εικονικής πραγματικότητας (Σπέντζας, 2015). Τη δεκαετία αυτή υπήρξε τεράστια αναγνώριση των παιχνιδιών που χρησιμοποιούσαν υπολογιστή, αφού πλέον το μεγαλύτερο μέρος του πληθυσμού διέθετε ένα δικό του υπολογιστή, ενώ, ακόμα, δόθηκε η δυνατότητα σε άτομα να δημιουργήσουν από μόνοι τους ένα βιντεοπαιχνίδι. Δικαίως, λοιπόν, απέκτησαν μία θέση στην καθημερινότητα των ανθρώπων, στην ψυχαγωγία και στην ζωή των ανεπτυγμένων πολιτισμών.

Από τις αρχές της εμφάνισης των βιντεοπαιχνιδιών, είχε φανεί πως θα αποτελέσουν σημαντικό κομμάτι των επόμενων γενεών. Τα βιντεοπαιχνίδια αποτέλεσαν και συνεχίζουν να αποτελούν ένα μέσο κοινωνικοποίησης, μάθησης και εξερεύνησης. Ακόμα, αποτέλεσε ένα μέσο που βοήθησε τα άτομα να σκέφτονται με διαφορετικό τρόπο, ενώ έδωσε νέους τρόπους ώστε να γίνει η μάθηση πιο διασκεδαστική. Στοιχεία που βοήθησαν στην εξέλιξη τους ήταν η δυνατότητα σύνδεσης με το διαδίκτυο, η τεχνητή νοημοσύνη και η επαυξημένη πραγματικότητα. Αφορούν μία νέα πλευρά της πραγματικότητας, αφού πλέον δίνεται η επιλογή του ρίσκου και της προσπάθειας, χωρίς όμως τη φυσική καταπόνηση του ατόμου (Βασιλόπουλος, 2021).

[1] Οι οπτικοί δίσκοι αφορούν επίπεδους κυκλικούς δίσκους, που μέσω μικροσκοπικών βαθουλωμάτων ή προεξοχών δίνουν ηχιτικές ή εικονικές πληροφορίες σε ψηφιακή μορφή. Οι πιο γνωστοί οπτικοί δίσκοι είναι το CD, το DVD και το Ble-Ray.

3.3 Κατηγορίες Ψηφιακών Παιχνιδιών

Τα βιντεοπαιχνίδια με βάση τα ιδιαίτερα χαρακτηριστικά του καθενός χωρίζονται σε κατηγορίες. Αυτά τα χαρακτηριστικά διαφοροποιούνται βάση των στόχων κάθε παιχνιδιού, του σεναρίου, τον αριθμό παικτών και διάφορων άλλων λειτουργιών. Χωρίστηκαν σε κατηγορίες με στόχο να αναγνωρίζονται τα χαρακτηριστικά του καθενός. Οι βασικότερες κατηγόριες είναι οι εξής:

Παιχνίδια Δράσης (Action Games): Στα παιχνίδια δράσης ο παίκτης χρειάζεται να ελέγξει το χαρακτήρα/ πρωταγωνιστή του παιχνιδιού. Στις περισσότερες περιπτώσεις συλλέγει κάποια αντικείμενα και δύναται να αποφύγει επιθέσεις από χαρακτήρες που αποτελούν εχθρούς ή απειλητικές καταστάσεις στο περιβάλλον όπου βρίσκεται. Στις περισσότερες περιπτώσεις απαιτεί γρήγορα αντανακλαστικά και συντονισμό. Η κατηγορίας αυτή διαθέτει παρακλάδια όπως είναι τα παιχνίδια Shooter ή Fighting.



Image 5: "Mortal Kombat"

Παιχνίδια Περιπέτειας (Adventure Games): Σε αυτά συνήθως διαδραματίζεται μία ιστορία και ο παίκτης έχει σαν στόχο την εξερεύνηση και την επίλυση γρίφων ώστε να φτάσει στο τέλος της. Διαθέτει στοιχεία αφήγησης, ενώ στις περισσότερες περιπτώσεις αφορούν έναν παίκτη καθώς δίνεται η έμφαση σε έναν πρωταγωνιστή, το δικό του βίωμα και αφηγηματική πορεία. Γνώρισαν μεγάλη επιτυχία μετά το 2010, με πολλές εταιρείες να εστιάζουν στην ανάπτυξη αυτού του είδους παιχνιδιών (Σπέντζας, 2015).



Image 6: "The Last Of Us"

Παιχνίδια Προσομοίωσης (Simulation Games): Τα παιχνίδια προσομοίωσης αφορούν βιντεοπαιχνίδια με δραστηριότητες του πραγματικού κόσμου, μέσα από έναν εικονικό. Θα μπορούσε για παράδειγμα να είναι κάποιο παιχνίδι όπου ο παίκτης καλείται να εκτελέσει κάποια συνταγή μαγειρικής. Τέτοια παιχνίδια πολλές φορές χρησιμοποιούνται στα πλαίσια της εκπαίδευσης και μάθησης, όπως για παράδειγμα στην ειδική διαπαιδαγώγηση, για την κατανόηση και εκμάθηση καθημερινών δραστηριοτήτων.

Παιχνίδια Στρατηγικής (Strategy Games): Τα παιχνίδια αυτά δίνουν έμφαση σε μία πορεία από κινήσεις και δεξιότητες του παίκτη για την επίτευξη ενός στόχου. Είναι παιχνίδια μεθοδικά, χρειάζονται σκέψη και αποφασιστικότητα στις κινήσεις. Έμφαση δίνεται στις κινήσεις και όχι στους χαρακτήρες του παιχνιδιού, αυτοί καθορίζονται από τις κινήσεις του παίκτη. Ακόμα, έχει τη δυνατότητα να ελέγξει το πεδίο στο οποίο διεξάγεται η μάχη.

Αθλητικά Παιχνίδια (Sport Games): όπως φαίνεται καθαρά από την ονομασία τους, τα αθλητικά παιχνίδια αφορούν παιχνίδια προσομοίωσης αθλητικών παιχνιδιών της πραγματικότητας. Αθλήματα όπως το ποδόσφαιρο, το μπάσκετ, η πάλη, κ.α. Πολλές φορές μπορεί να περιέχουν στοιχεία, όπως ονομασίες αθλητικών ομάδων ή αθλητών, του πραγματικού κόσμου και να αποτυπώνονται στον εικονικό μέσω του βιντεοπαιχνιδιού.



Image 7: "Fifa"

Παιχνίδια Ρόλων (Role Playing Games): Τα παιχνίδια ρόλων αφορούν έναν χαρακτήρα, μέσα από κάποιο σενάριο και τις επιλογές και κινήσεις του παίκτη, αυτός ο χαρακτήρας αναπτύσσεται, άλλο τόσο αναπτύσσονται και οι αντίπαλοι με τους οποίους έρχεται αντιμέτωπος. Έχουν παρόμοια χαρακτηριστικά με επιτραπέζια παιχνίδια ρόλων, όπου κάθε

εμπλεκόμενος έχει κάποιες συγκεκριμένες ικανότητες με στόχο να τις αναπτύξει

Εκπαιδευτικά Παιχνίδια (Educational Games): Τα εκπαιδευτικά παιχνίδια αποσκοπούν στη διδασκαλία και αυτοδιδασκαλία του χρήστη. Χρησιμοποιούνται ως ένα πιο ευχάριστο και ψυχαγωγικό μέσο για μάθηση. Μπορούν να αφορούν παιχνίδια εκμάθησης ξένων γλωσσών, μαθηματικών, ιστορίας, πληροφορικής κ.α. Μπορούν να έχουν τη μορφή κουίζ, πολλαπλών ερωτήσεων, ενώ στόχος του παίκτη είναι να ανέβει επίπεδο, μαθαίνοντας όλο και περισσότερες πληροφορίες για το αντικείμενο που αφορά το παιχνίδι.

Παιχνίδια Βολών Τρίτου Προσώπου (Third Person Shooter): Τα παιχνίδια βολών τρίτου προσώπου αφορούν παιχνίδια στα οποία ο παίκτης έρχεται αντιμέτωπος με προκλήσεις μέσα σε έναν κόσμο που τον απειλεί. Η ιδιαιτερότητα των παιχνιδιών αυτών κρύβεται στην γραφική προοπτική του παιχνιδιού, όπου ο παίκτης ελέγχει έναν χαρακτήρα σε μία σταθερή απόσταση πίσω από το είδωλο του. Ακόμα ένα χαρακτηριστικό βρίσκεται στην τρισδιάστατη προοπτική του παιχνιδιού, που σημαίνει πως ο παίκτης δεν βασίζεται μονάχα σε δύο διαστάσεις και οι κινήσεις του δεν περιορίζονται εκεί, αλλά έχει τη δυνατότητα να επεξεργαστεί έναν πιο σφαιρικό εικονικό κόσμο.

[23]**Παιχνίδια Βολών Πρώτου Προσώπου (First Person Shooter):** Τέλος, τα παιχνίδια βολών πρώτου προσώπου μοιάζουν σε μεγάλο βαθμό με τα παιχνίδια βολών τρίτου προσώπου ως προς την τρισδιάστατη προοπτική τους, διαφέρουν όμως στο χαρακτήρα. Ο χαρακτήρας του παιχνιδιού είναι ο ίδιος ο παίκτης, αφορά μία προέκταση του παίκτη, αφού η οπτική του απορρέει από τα εικονικά μάτια του παίκτη. Έτσι, εκείνος δεν ελέγχει ακριβώς έναν χαρακτήρα όπως στα προηγούμενα, αλλά γίνεται ο ίδιος πρωταγωνιστής του παιχνιδιού και καλείται να επιβιώσει μέσα από τις προκλήσεις που του δίνονται με τη χρήση εικονικών όπλων.

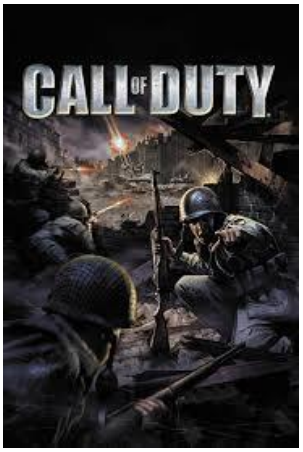


Image 8: "Call of Duty"

[2] Έχουν σαν στόχο την ψευδαίσθηση ότι ο παίκτης βρίσκεται μέσα στον εικονικό κόσμο, λόγω αυτού, πολλές φορές τα παιχνίδια βολών πρώτου προσώπου παρομοιάζονται με τα παιχνίδια επαυξημένης πραγματικότητας.

[3] Με τη δημιουργία μίας τέτοιας κατηγορίας παιχνιδιού ασχολήθηκε η συγκεκριμένη πτυχιακή εργασία.

Άλλα χαρακτηριστικά κατά τα οποία μπορούν να ταξινομηθούν τα βιντεοπαιχνίδια είναι η χρήση τους μέσω της τηλεόρασης και κάποιας κονσόλας, η χρήση μέσα από τον υπολογιστή, η δυνατότητα χρήσης του διαδικτύου ή όχι κ.α. (Μαραγκοπούλου, 2018). Αυτό έχει οδηγήσει στην εξίσωση των όρων "βιντεοπαιχνίδι", "ψηφιακό παιχνίδι", "ηλεκτρονικό παιχνίδι".

3.4 Στοιχεία Ψηφιακών Παιχνιδιών

Βασική απαίτηση για ένα παιχνίδι είναι να προσφέρει την απόλαυση και ψυχαγωγία που αναζητεί ο παίκτης. Η ευχαρίστηση αυτή πηγάζει μέσα από κάποια στοιχεία που ο σχεδιαστής καλείται να εφαρμόσει. Για αρχή, λοιπόν, θα πρέπει να έχει θέσει με σαφή τρόπο το σύστημα κανόνων που διακατέχει το παιχνίδι. Με αυτό εννοείται οι κανόνες να γίνονται κατανοητοί απέναντι στο χρήστη ώστε να έχει τη δυνατότητα να αναγνωρίζει τους στόχους που πρέπει να επιτελέσει στο παιχνίδι και το πως θα αλληλεπιδρά με τα ερεθίσματα.

Το αμέσως επόμενο στοιχείο των βιντεοπαιχνιδιών με μεγάλη σημασία στην απόδοση του είναι η ιστορία κατά την οποία εκτυλίσσεται. Αυτή μπορεί να είναι προσχεδιασμένη, είτε να παίρνει διαφορετική τροπή με βάση την πορεία που ακολουθεί ο παίκτης. Ανεξάρτητα την πορεία που ακολουθεί όμως, θα πρέπει να αποτελείται από ένα ελκυστικό σενάριο όπου ο παίκτης δεν θα κουράζεται αλλά, αντ΄ αυτού θα επιθυμεί να συνεχίσει μέχρι τέλους.

Μεγάλο ρόλο στην ελκυστικότητα της ιστορίας και στοιχείο των βιντεοπαιχνιδιών αποτελεί η αισθητική του παιχνιδιού, οι χαρακτήρες που ελέγχονται από τον παίκτη, η μουσική που ακούγεται, τα ιδιαίτερα καλλιτεχνικά στοιχεία ενός παιχνιδιού. Αυτά, θα πρέπει να ενσωματώνονται στο παιχνίδι με τρόπο που θα ταιριάζουν με την ιστορία, την εποχή, το στυλ και την κατηγορία του παιχνιδιού. Εάν, για παράδειγμα, το παιχνίδι είναι της κατηγορίας δράσης, η μουσική που επιλέγεται θα είναι μουσική που προκαλεί αδρεναλίνη στο χρήστη.

Ακόμα, ένα βασικό στοιχείο είναι η τεχνολογία που χρησιμοποιείται για τη χρήση του παιχνιδιού. Κάθε τεχνολογία έχει δικούς της περιορισμούς ή δυνατότητες. Διαφορετικά χρησιμοποιείται ένας υπολογιστής από μία κονσόλα Playstation, διαφορετικά εργαλεία χρειάζονται για τη χρήση τους (λ.γ. πληκτρολόγια, κονσόλες, διαδίκτυο κ.λπ), άρα και με διαφορετικό τρόπο σχεδιάζονται τα παιχνίδια για να χρησιμοποιηθούν στο εκάστοτε τεχνολογικό μέσο.

Τέλος, το περισσότερο απαραίτητο στοιχείο είναι ο ίδιος ο σχεδιαστής. Χρειάζεται να διαθέτει γνώσεις στο κομμάτι του προγραμματισμού, φαντασία, ιδέες και όρεξη για να φέρει κάποιο καλό αποτέλεσμα. Ο σχεδιασμός ενός βιντεοπαιχνιδιού για όποια κονσόλα αποτελεί μία πολυσύνθετη διαδικασία, απαιτεί χρόνο και προσπάθεια. Ακόμα, τα διάφορα προγράμματα που χρησιμοποιούνται προσφέρουν λίγες ή πολλές δυνατότητες που χρειάζονται για τη διαδικασία αυτή. Στις μέρες μας πολλοί έχουν τη δυνατότητα να δημιουργήσουν κάτι δικό τους. Υπάρχει δωρεάν υλικό στο διαδίκτυο, μηχανές εκμάθησης κ.α.

3.5 Game Engine Unity



Image 9: Λογότυπο της μηχανής παιχνιδιών Unity

Για την ανάπτυξη των βιντεοπαιχνιδιών χρειάζονται κάποιες μηχανές παιχνιδιών. Οι μηχανές παιχνιδιών αφορούν εξειδικευμένα λογισμικά σχεδίασης παιχνιδιών, σχεδιασμένα να λειτουργούν σε κονσόλες και λειτουργικά συστήματα υπολογιστών και δίνουν πρόσβαση στο χρήστη σε εργαλεία όπως 2D, 3D γραφικά, ήχους, animation, γράφο σκηνής, σύστημα διαχείρισης μνήμης κ.α. (Μετρεβέλι, 2023). Για το παιχνίδι με το οποίο ασχολείται η παρούσα πτυχιακή χρησιμοποιήθηκε η μηχανή παιχνιδιών Unity. Η Unity αποτελεί μία μηχανή παιχνιδιών που περιέχει πληθώρα εργαλείων, εκπαιδευτικά εγχειρίδια και πιστοποιήσεις γνώσης της μηχανής. Δημιουργήθηκε το 2004 στη Δανία με όνομα Over The Edge Entertainment ενώ το 2007 άλλαξε η ονομασία σε Unity. Πλέον αποτελεί μία παγκοσμίως αναγνωρισμένη μηχανή παιχνιδιών με εκατομμύρια σχεδιαστές να την επιλέγουν. Παρέχει τη δυνατότητα σχεδίασης βιντεοπαιχνιδιών για πολλές πλατφόρμες όπως υπολογιστές, κονσόλες, κινητά και αυτό της επιτρέπει να προσελκύει όλο και περισσότερους χρήστες.

Η πρόσβαση στη μηχανή Unity μπορεί να είναι δωρεάν ή επί πληρωμή, παρέχοντας διαφορετικές δυνατότητες ανάλογα με την εκάστοτε επιλογή. Οφείλουμε να προσθέσουμε πως η Unity παρέχει την επιλογή στους χρήστες να πουλήσουν και να αγοράσουν προϊόντα, χρήσιμα στη σχεδίαση των ψηφιακών παιχνιδιών, μέσω του Unity Asset Store. Στη παρούσα πτυχιακή εργασία, από το Unity Asset Store χρησιμοποιήθηκαν έτοιμα μοντέλα για τα ζόμπι (zombie), τα όπλα, το χάρτη και κάποιους από τους ήχους.

Ακόμα, υπάρχει μία μεγάλη κοινότητα όπου υπάρχει διάλογος μεταξύ των χρηστών και προγραμματιστών, όπου παρέχουν συμβουλές, ανταλλάσσουν ιδέες, οδηγίες χρήσης κ.α. Για την ολοκλήρωση ενός σχεδίου υπάρχουν βασικές πτυχές που πρέπει να έχουν ολοκληρωθεί. Αυτές είναι η όψη σκηνής, η όψη έργου, η όψη παιχνιδιού, η όψη ιεραρχίας, η όψη κονσόλας και η όψη ελέγχου. Η όψη σκηνής είναι η πιο βασική, καθώς σε αυτή τοποθετούνται τα αντικείμενα και δημιουργούνται οι σκηνές του παιχνιδιού. Η όψη έργου είναι εκείνη που διαθέτει τα στοιχεία του παιχνιδιού, όπως ο ήχος ή τα μοντέλα. Όψη παιχνιδιού είναι ο όψη που βλέπει ο παίκτης ενώ του δίνονται διάφορες επιλογές όπως η μεγιστοποίηση. Αμέσως μετά, η όψη ιεραρχίας αφορά το πως φαίνονται τα αντικείμενα στη σκηνή σε συγκεκριμένο χρονικό πλαίσιο. Έπειτα, η όψη ελέγχου αφορά τα στοιχεία κάθε αντικειμένου, όπως ο ρόλος του στο παιχνίδι, η τοποθεσία του στη σκηνή κ.λπ. Τέλος, η όψη κονσόλας αφορά τον κώδικα και σφάλματα που προκύπτουν στον κώδικα κατά τη διαδικασία

ανάπτυξης του από τον προγραμματιστή (Κρεζίας, 2016).

Οι μηχανές παιχνιδιών είναι εργαλεία που βοηθούν τον χρήστη να μειώσει κατά πολύ το χρόνο και τον κόπο για την ανάπτυξη παιχνιδιών ενώ του δίνεται η δυνατότητα να επαναχρησιμοποιήσει δεδομένα. Ένα παιχνίδι πριν από τη δημιουργία των μηχανών αυτών, έπρεπε να γραφτεί και να σχεδιαστεί εξ ολοκλήρου από το μηδέν. Πλέον με τη χρήση των μηχανών υπάρχουν έτοιμα στοιχεία και δεδομένα, έτσι ο σχεδιαστής να αφιερώνει λιγότερο χρόνο σε ορισμένα στοιχεία, δίνοντας περισσότερο έμφαση σε στοιχεία όπου θα βελτιώσουν το παιχνίδι ακόμα πιο πολύ. Η Unity είναι πλέον μία από τις περισσότερο χρησιμοποιούμενες μηχανές παιχνιδιών, αυτή, σε αντίθεση με άλλες μηχανές παιχνιδιών, εστιάζει στη δημιουργία αντικειμένων και όχι στον κώδικα. Η δημιουργία των αντικειμένων αυτών εντάσσεται στο scripting και αφορά τη συμπεριφορά των αντικειμένων αυτών, ενώ, τέλος, η Unity υποστήριζε τις γλώσσες C#, Javascript και Boo, πλέον όμως μονάχα της C#.

3.5.1 C# (C Sharp)

Για την ανάπτυξη ψηφιακών παιχνιδιών, η χρήση μηχανών απαιτεί και τη χρήση κάποιας γλώσσας προγραμματισμού. Η Unity κατά κύριο λόγο χρησιμοποιεί τη γλώσσα C#, μία αντικειμενοστραφής γλώσσα, γενικού σκοπού. Δημιουργήθηκε από τη Microsoft το 2000, υπεύθυνος της ομάδας ήταν ο Άντερς Χάιλσμπέργκ, και η γλώσσα αναγνωρίστηκε από την Ecma και την ISO το 2003 (Μετρεβέλι, 2023). Με τη χρήση της C# ο χρήστης δημιουργεί μία σειρά από scripts τα οποία αλληλεπιδρούν μεταξύ τους. Στην ουσία προσδίδει εντολές στα αντικείμενα για το πως θα αντιδράσουν ανάλογα με τα εκάστοτε ερεθίσματα.

Κεφάλαιο 4: Υλοποίηση

4.1 Λειτουργίες του παιχνιδιού

Στο παρόν παιχνίδι, για κάθε στοιχείο έχουν οριστεί και κάποιες συγκεκριμένες λειτουργίες που μπορεί να επιτελέσει. Σε κάθε παιχνίδι αυτές οι λειτουργίες μπορεί να είναι περιοριστικές ή μη, ανάλογα με τι θα επιλέξει να επιτρέψει ο κάθε προγραμματιστής και με αυτές τις λειτουργίες θα πορευθεί και κάθε χρήστης. Ξεκινώντας με τον χαρακτήρα του παιχνιδιού SombieZ, έχει τη δυνατότητα να τρέξει ή απλά να περπατήσει, μπορεί να αναπηδήσει και να σκύψει, να γυρίσει σε κάθε κατεύθυνση του χάρτη, να πυροβολήσει, να πιάσει αντικείμενα από κάτω και να αμυνθεί.

Επόμενο, βασικό, στοιχείο του παιχνιδιού είναι οι εχθροί, τα ζόμπι. Όλα τα ζόμπι έχουν τις ίδιες λειτουργίες, με τη διαφορά πως όσο περνάνε οι γύροι γίνονται πιο δυνατά από τους προηγούμενους. Έτσι, τα ζόμπι έχουν τη δυνατότητα να τρέξουν ή να περπατήσουν, να επιτεθούν στον χαρακτήρα του παιχνιδιού, μπορούν να προχωρήσουν προς οποιαδήποτε κατεύθυνση του χάρτη και να κυνηγήσουν τον πρωταγωνιστή.

Τα υπόλοιπα στοιχεία που διαθέτει το παιχνίδι συνεργάζονται με βάση τις λειτουργίες αυτές. Έτσι, η εικόνα του χάρτη που έχει ο χρήστης στην οθόνη του καθορίζεται από την κίνηση του μέσα σε αυτόν. Αντίστοιχα, η κάμερα είναι στραμμένη προς το μέρος που ο χρήστης κατευθύνει τον χαρακτήρα του. Τα όπλα, έχουν τη δυνατότητα να γεμίσουν ξανά με σφαίρες, να αλλάξουν, να πυροβολήσουν ή να μείνουν σε θέση άμυνας στο χέρι του χαρακτήρα. Κάθε λειτουργία τους όμως καθορίζεται επίσης από τον χρήστη. Τέλος, σκοπός του παιχνιδιού είναι να φτάσεις σε ένα όσο το δυνατόν μεγαλύτερο επίπεδο. Αυτό σημαίνει πως το παιχνίδι δεν ολοκληρώνεται με κάποιο συγκεκριμένο τρόπο αλλά κάθε χρήστης παίζει έως ότου να χάσει, ενώ δεν παρέχει τη δυνατότητα παύσης κατά τη διάρκεια χρήσης του.

4.2 SombieZ

Η συγκεκριμένη πτυχιακή εργασία αφορά την ανάπτυξη ενός παιχνιδιού βολών πρώτου προσώπου, ονόματι “SombieZ”, με τη χρήση της μηχανής παιχνιδιών Unity3D, όπου έγινε αναφορά στο προηγούμενο κεφάλαιο, και τη χρήση της γλώσσας προγραμματισμού C# (C Sharp). Το ενδιαφέρον εστίασε στο πως γράφετε ο κώδικας, πως δημιουργούνται τα στοιχεία και τα αντικείμενα, τι απαιτήσεις υπάρχουν κ.α. Στο παιχνίδι αυτό, πρωταγωνιστής είναι ο ίδιος ο παίκτης, χαρακτηριστικό -όπως προαναφέρθηκε- των παιχνιδιών κατηγορίας βολών πρώτου προσώπου, με στόχο να επιβιώσει όσο το δυνατόν περισσότερους γύρους μπορεί, εξοντώνοντας ζόμπι με τα όπλα που διαθέτει. Θα γίνει εκτενέστερη αναφορά για τα μέρη του κώδικα.

4.2.1 Κεντρικό Μενού

Το κεντρικό μενού του παιχνιδιού αποτελείται από πέντε στοιχεία, αυτά είναι η φωτογραφία εξωφύλλου, ο τίτλος/ ονομασία του, δύο κουμπιά, εισόδου και εξόδου και η μουσική.



Image 10: Κεντρικό Μενού SombieZ

Το βασικό μενού έχει δημιουργηθεί σαν ξεχωριστό [4]scene το οποίο παραπέμπει στο παιχνίδι (στο scene του παιχνιδιού). Χρησιμοποιώ απλές UI τεχνικές του Unity που έχω περάσει μέσα στο ήδη υπάρχων canvas. Αυτές είναι: Image, Buttons και TextMeshPro.

Το Image είναι για το background για να ομορφύνει το μενού μας, τα buttons βοηθάνε ώστε να μπορούμε να δώσουμε εντολές για να ξεκινήσει το παιχνίδι ή να κλείσει, ενώ ταυτόχρονα έχουν ενσωματωμένα δικά τους Texts για να μπορούμε να τα ονομάσουμε. Επίσης χρησιμοποιώ τα TextMeshPro στον τίτλο γιατί δίνουν περισσότερες δυνατότητες να ”παίξω” με τον τίτλο ως προς τα χρώματα, το μέγεθος κ.λπ. Τέλος, έχω προσθέσει μουσική στο μενού για να βάλω τον παίκτη στο κλίμα του παιχνιδιού.

[4] Το scene στην ουσία αφορά το χάρτη πάνω στον οποίο αναπτύσσεται το παιχνίδι. Είναι το μέρος στο οποίο τοποθετούνται τα στοιχεία και αντικείμενα του παιχνιδιού. Σε πιο απλά παιχνίδια χρειάζεται ένα scene, ενώ σε κάποια πιο περίπλοκα μπορεί να χρειαστούν περισσότερα από ένα.



Image 11: Κεντρικό μενού: Canvas.

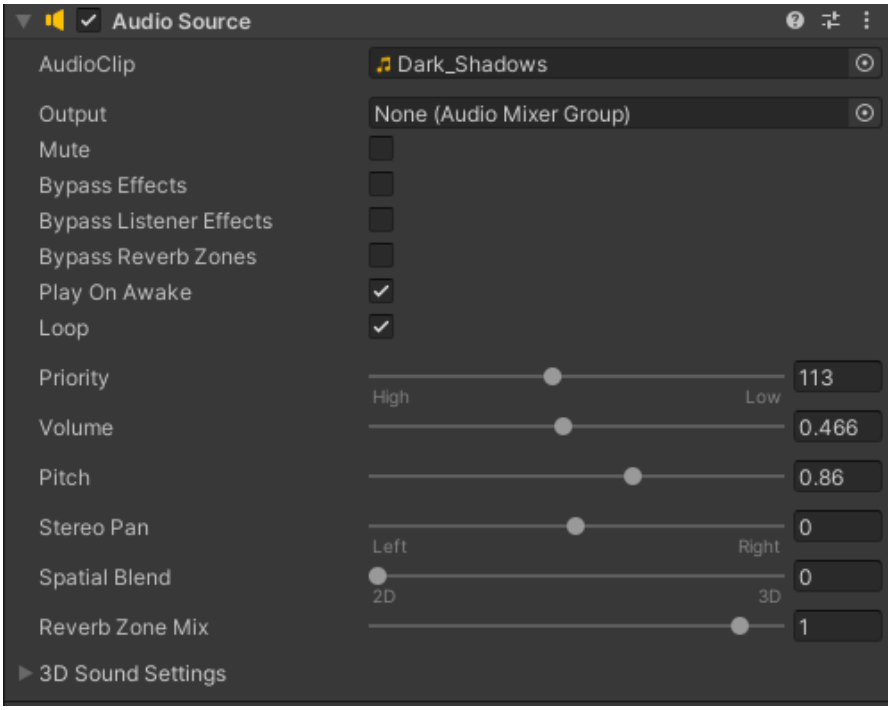


Image 12: Κεντρικό μενού: Μουσική (Background music).

Τέλος, χρησιμοποίησα κώδικα για να δώσω λειτουργίες στα κουμπιά στο κεντρικό μενού.

```
using System.Net.Mime;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;



public class MainMenu : MonoBehaviour
{
    public void Play()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +1);
       //SceneManager.LoadScene("Demo_Scene");//Allos tropos tou apo pano
    }
    public void Quit()
    {
        Application.Quit();
    }
}

```

Image 13: Κεντρικό μενού: Κώδικας.

4.2.2 User Interface (UI)

Το UI (user interface) είναι ο τρόπος επικοινωνίας του παιχνιδιού με τον χρήστη-παίκτη. Είναι ο τρόπος με τον οποίο βλέπει ο παίκτης τους μηχανισμούς και λειτουργίες του παιχνιδιού. Στη περίπτωση αυτού του παιχνιδιού ο παίχτης μέσα από κάποιους μηχανισμούς του δίνεται η δυνατότητα να ελέγχει σε τι κατάσταση βρίσκετε στο παιχνίδι. Αυτά είναι: ο τρέχον γύρος, τα υπολειπόμενα ζόμπι, η ζωή που του απομένει, το σκόπευτρο του και τέλος οι σφαίρες του.

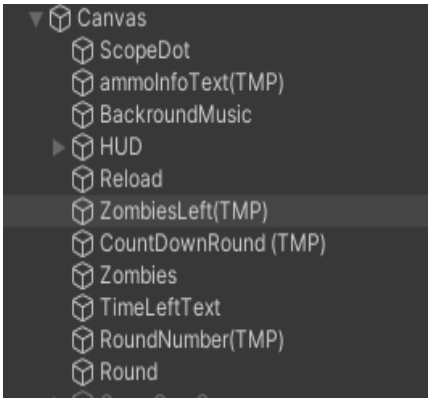


Image 14: UI: Canvas.

```
1 reference
public TextMeshProUGUI zombiesLeftText;
1 reference
public TextMeshProUGUI countdownText;
1 reference
public TextMeshProUGUI roundText;

2 references
public GameObject gameOverUI;
0 references
void FixedUpdate()
{
    countdownText.text = countdown.ToString();
    zombiesLeftText.text= zombiesLeftInRound.ToString();
    roundText.text = round.ToString();
```

Image 15: UI: Κώδικας No. 1.

```
0 references
public class PlayerHUD : MonoBehaviour
{
    1 reference
    [SerializeField] private TextMeshProUGUI  currentHealthText;
    1 reference
    [SerializeField] private TextMeshProUGUI  maxHealthText;

    0 references
    public void UpdateHealth(float currentHealth , float maxHealth)
    {
        currentHealthText.text = currentHealth.ToString();
        maxHealthText.text = maxHealth.ToString();
    }
}
```

Image 16: UI: Κώδικας No. 2.

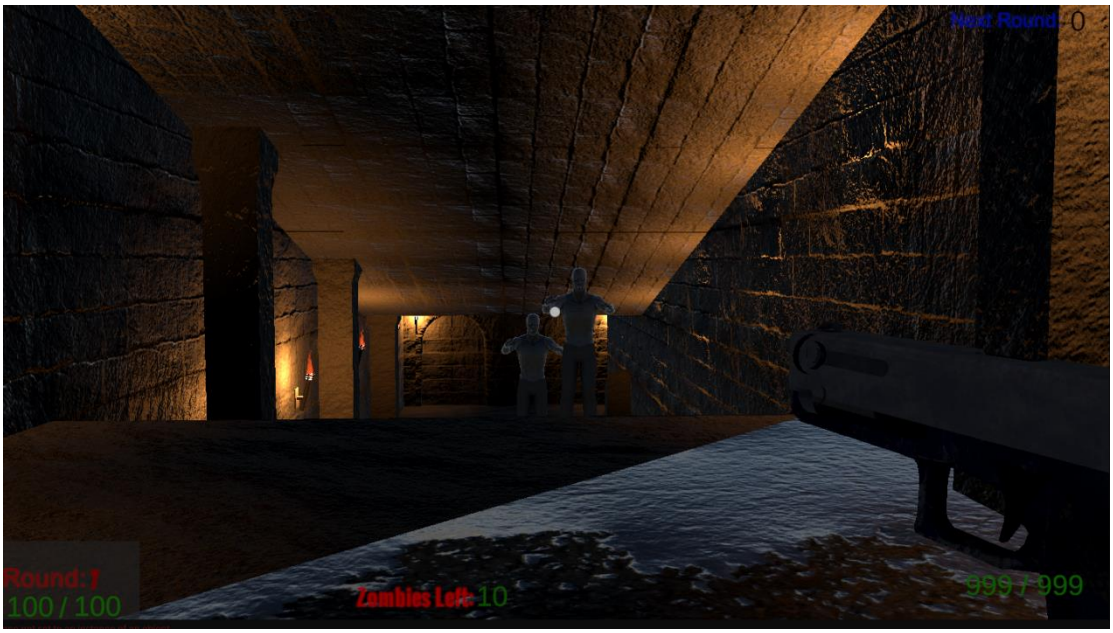


Image 17: UI: Εμφάνιση στο παιχνίδι.

Στις παραπάνω φωτογραφίες βλέπουμε τον τρόπο με τον οποίο έχουν οριστεί στο unity, τον κώδικα που έχει χρησιμοποιηθεί για να δημιουργηθούν και τον τρόπο που εμφανίζονται στο παιχνίδι.

4.2.3 Weapons (Όπλα)

Τα όπλα υπάρχουν για να μπορεί ο χαρακτήρας να αμύνεται απέναντι στους αντιπάλους του. Στο παιχνίδι αυτό αντίπαλοι είναι τα ζόμπι. Ο χαρακτήρας κρατάει πάνω του μόνιμα τρία όπλα, ένα αυτόματο, ένα μικρό πιστόλι και μια καραμπίνα. Το καθένα χρησιμοποιείται για διαφορετικό σκοπό και έχει διαφορετικά χαρακτηριστικά. Στις παρακάτω εικόνες φαίνεται ο βασικός κώδικας για τα όπλα.

```
using System.Net.WebSockets;
using System.Net;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.InputSystem;
using StarterAssets;
using TMPro;
//Auto to script einai gia tp Automato
2 references
public class Guns : MonoBehaviour
{
    //Fire kai Damage
    2 references
    public Transform FpsCam;
    1 reference
    public float range = 20;
    1 reference
    public float impactForce = 150;
    1 reference
    public int fireRate = 10;
    2 references
    private float nextTimeToFire;
    1 reference
    public float Damage;
//Effects kai hud
    1 reference
    public ParticleSystem muzzleflush;
    1 reference
    public GameObject impactEffect;

    2 references
    AudioSource AssaultSound;
    2 references
    AudioClip reload;
    1 reference
    public AudioSource audioSource;
    1 reference
    public TextMeshProUGUI ammoInfoText;

    //Reload kai sfaires
    9 references
    private int currentAmmo;
    6 references
    private int maxAmmo= 30;
    7 references
    private int magSize = 150;
    3 references
    private bool isReloading=false;
    1 reference
    private int reloadTime=2;
```

Image 18: Κώδικας όπλων No.1.

```
10  public class Guns : MonoBehaviour
36      void Start()
37      {
38          currentAmmo=maxAmmo;
39          AssaultSound = GetComponent<AudioSource>();
40          reload = GetComponent<AudioClip>();
41      }
42
        0 references
43      void Update()
44      {
45          Guns currentGun = FindObjectOfType<Guns>(false);
46          //print(currentGun.name);
47          ammoInfoText.text = currentGun.currentAmmo + " / " + currentGun.magSize;
48          if(isReloading)
49          {
50              return;
51          }
52          if(currentAmmo<=0)
53          {
54              audioSource.PlayOneShot(reload,3f);
55              StartCoroutine(Reload());
56              return;
57          }
58          if(Input.GetButton("Fire1") && Time.time>=nextTimeToFire)
59          {
60              nextTimeToFire= Time.time + 1f/fireRate;
61              Fire();
62          }
63
64          if(Input.GetKeyDown(KeyCode.R )&& currentAmmo<magSize && currentAmmo<maxAmmo)
65          {
66              StartCoroutine(Reload());
67              return;
68          }
```

Image 19: Κώδικας όπλων No. 2.

```
    }
    2 references
    IEnumerator Reload()
    {
        //audioSource.PlayOneShot(reload,3f);
        isReloading=true;
        //Debug.Log("Reloading");
        yield return new WaitForSeconds(reloadTime);
        if(magSize>=maxAmmo){
            currentAmmo=maxAmmo;
            magSize-=maxAmmo;
        }
        else{
            currentAmmo=magSize;
            magSize=0;
        }

        isReloading=false;
    }

    1 reference
    private void Fire()
    {
        AssaultSound.Play();
        RaycastHit hit;
        muzzleflush.Play();
        currentAmmo--;
        if(Physics.Raycast(FpsCam.position,FpsCam.forward,out hit,range));
        {
            //Debug.Log(hit.transform.name);
            Enemy target = hit.transform.GetComponent<Enemy>();
            if(target!=null){
                target.TakeDamage(Damage);
            }
           if(hit.rigidbody!= null)
           {
            hit.rigidbody.AddForce(-hit.normal * impactForce);
           }
           Quaternion impactRotation = Quaternion.LookRotation(hit.normal);
           GameObject impact=Instantiate(impactEffect,hit.point,impactRotation);
           Destroy(impact,5);

        }
    }
```

Image 20: Κώδικας όπλων No. 3.

```
    }

    0 references
    public void FillAmmo()
    {
        currentAmmo=maxAmmo;
        magSize=150;
    }
}

```

Image 21: Κώδικας για γέμισμα του όπλου.

Στον κώδικα αρχικά ξεκινάω με όσες μεταβλητές πιστεύω ότι θα χρειαστούν. Μέτα φαίνεται το πως χρησιμοποιώ τις συναρτήσεις που έχω φτιάξει. Οι συναρτήσεις είναι οι εξής: Reload, Fire, FillAmmo. Η πρώτη χρειάζεται για να γεμίσουμε το όπλο με σφαίρες όταν ξεμείνουμε, η δεύτερη δίνει τη δυνατότητα στο όπλο να πυροβολήσει και η τρίτη αφορά κομμάτι που θα εξεταστεί στη συνέχεια.

Η καραμπίνα είναι για σκοτώνει πιο γρήγορα τα ζόμπι λόγω μεγάλου damage (μπορεί να προκαλέσει μεγαλύτερη ζημιά στον αντίπαλο), αλλά έχει λίγες σφαίρες και είναι αργή.



Image 22: Καραμπίνα.

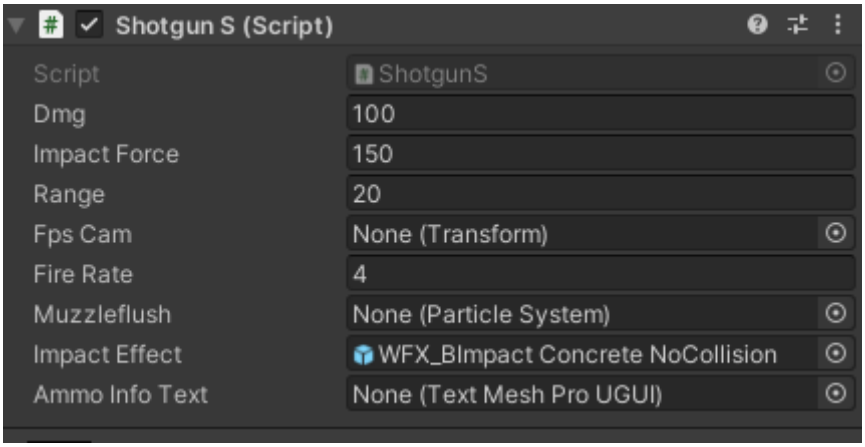


Image 23: Χαρακτηριστικά καραμπίνας.

Έπειτα το αυτόματο (AK-47) υπάρχει για να σκοτώνει μαζικά ορδές με το σχετικά μεγάλο damage (δυνατότητα μεγαλύτερης ζημιάς στον αντίπαλο), καλό fire rate (ταχύτητα βολών) και αρκετές σφαίρες.



Image 24: Αυτόματο (AK-47).

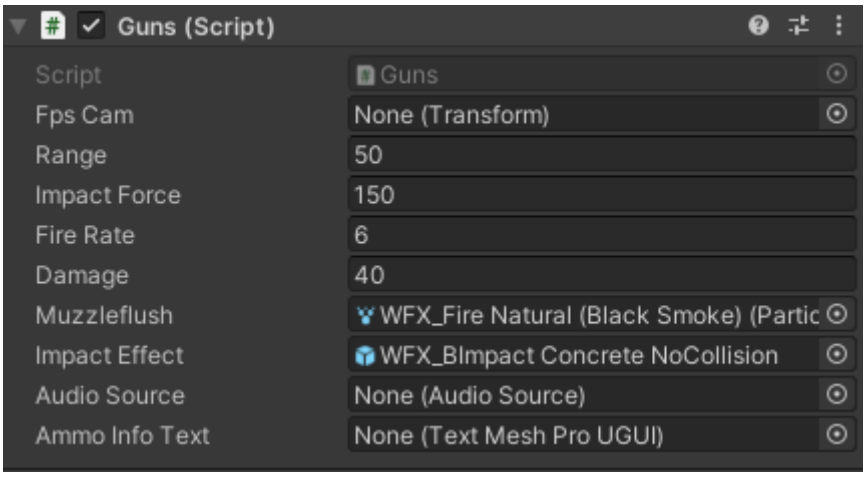


Image 25: Χαρακτηριστικά αυτόματου (AK-47).

Τέλος, το πιστολάκι είναι η τελευταία επιλογή σε περίπτωση με δεν έχουμε άλλες σφαίρες στα υπόλοιπα όπλα. Χαμηλό damage, αργό fire rate, αλλά άπειρες σφαίρες.


Image 26: Πιστόλι.

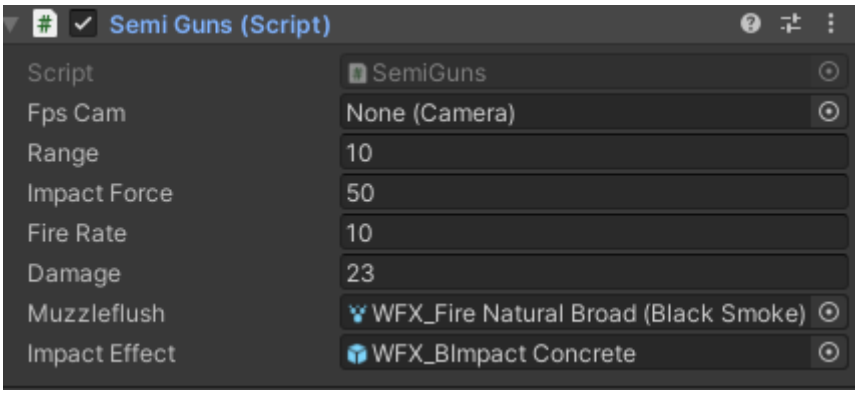


Image 27: Χαρακτηριστικά πιστολιού.

4.2.4 Weapon Switch (Εναλλαγή των όπλων)

Το weapon switching mechanic υπάρχει για να μπορεί ο χαρακτήρας να αλλάζει μεταξύ των όπλων που διαθέτει. Ο παίχτης πρέπει να χρησιμοποιήσει την ροδέλα για να

αλλάξει όπλο.

```
public class WeapponSwitch : MonoBehaviour
{
    public int selectedWeapon = 0;
    // Start is called before the first frame update
    void Start()
    {
        SelectWeapon();
    }

    // Update is called once per frame
    void Update()
    {
        int previousSelectedWeapon = selectedWeapon;

        if(Input.GetAxis("Mouse ScrollWheel") > 0f)
        {
            if(selectedWeapon>=transform.childCount- 1)
             selectedWeapon = 0;
             else
            selectedWeapon ++;
        }
         if(Input.GetAxis("Mouse ScrollWheel") < 0f)
        {
            if(selectedWeapon<= 0)
             selectedWeapon = transform.childCount- 1;
             else
               selectedWeapon --;
        }
        if(previousSelectedWeapon != selectedWeapon)
        {
            SelectWeapon();
        }
    }
    void SelectWeapon()
    {
         int i=0;
        foreach(Transform weapon in transform)
        {
            if(i == selectedWeapon)
              weapon.gameObject.SetActive(true);
              else
              weapon.gameObject.SetActive(false);
            i++;
        }
    }
}
```

Image 28: Κώδικας εναλλαγής όπλων.

Ο κώδικας ουσιαστικά ενεργοποιεί και απενεργοποιεί τα όπλα με βάση την σειρά που έχουν τοποθετηθεί.

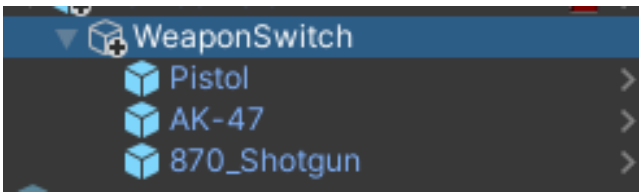


Image 29: Σειρά τοποθέτησης όπλων.

Εδώ φαίνεται ο τρόπος με τον οποίο δουλεύει το weapon switch στο unity, καθώς και την σειρά που έχουν τοποθετηθεί τα όπλα.

4.2.5 Zombies

Ο εχθρός σε αυτό το παιχνίδι είναι ορδές από ζόμπι. Τα ζόμπι έρχονται κατά αριθμούς σε γύρους. Όσο εμείς κερδίζουμε τους γύρους τόσα περισσότερα και δυνατότερα έρχονται. Τα ζόμπι αποτελούνται από κάποια χαρακτηριστικά: Ζωή, χτυπήματα και ταχύτητα. Επίσης η unity προσφέρει ένα δικό της navigation system το οποίο χρησιμοποιώ στα ζόμπι για να μπορούν να εντοπίζουν τον χαρακτήρα όπου και να βρίσκεται στον χάρτη.



Image 30: Μοντέλο Ζόμπι.

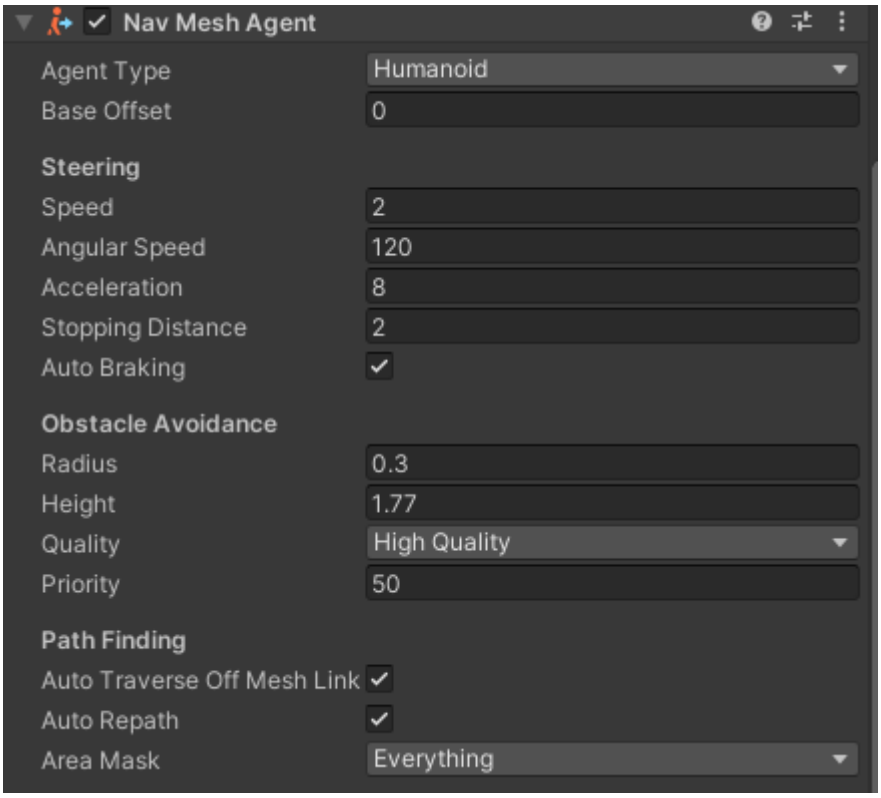


Image 31: Σύστημα πλοήγησης (navigation system).

Επίσης τα ζόμπι χρησιμοποιούν rigidbody και Capsule Collider, με αυτά τα δυο προσθέτουμε την δυνατότητα να μπορεί να επιτεθεί, αλλά και ο χρήστης να αμυνθεί επίσης.



Image 32: Rigidbody & Capsule Collider.



Image 33: Χαρακτηριστικά Ζόμπι.

Το ζόμπι έχει επίσης κάποια animation τα οποία δουλεύουν όταν χτυπάνε τον παίχτη, στο θάνατο τους και στο περπάτημα.

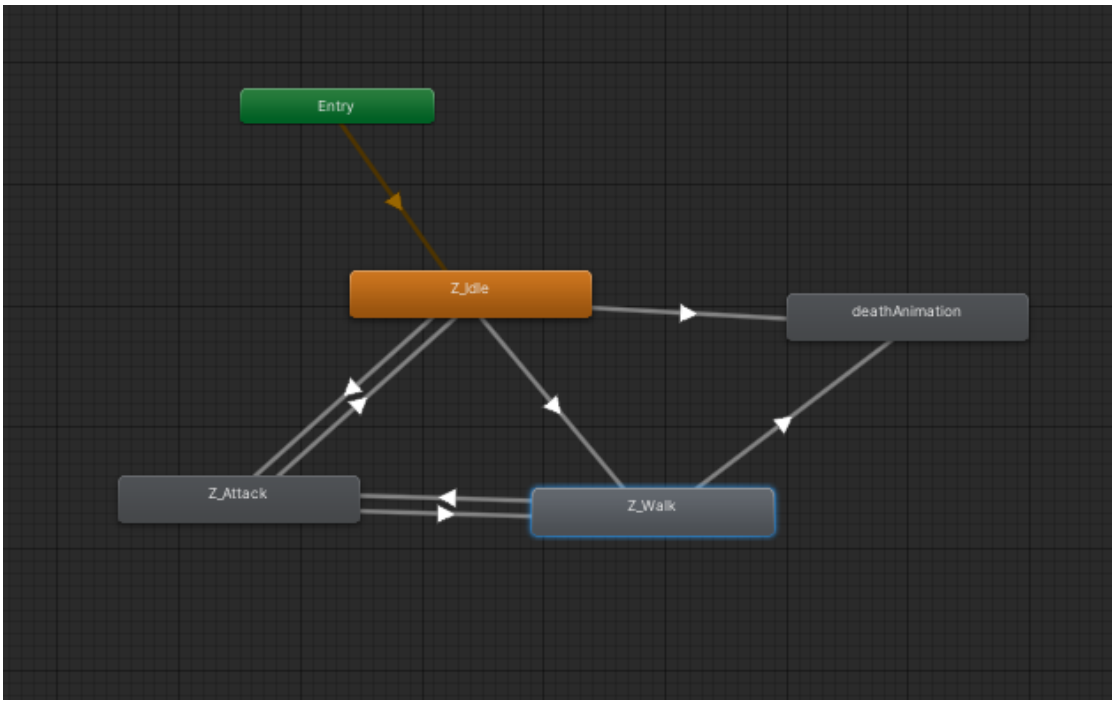


Image 34: Εναλλαγή animation.

Τέλος χρησιμοποιείται κώδικας για να δώσει στα ζόμπι την δυνατότητα να χρησιμοποιούν τα παραπάνω χαρακτηριστικά και να έχουν κάποια συγκεκριμένα όρια πάνω σε αυτά, όπως για παράδειγμα συγκεκριμένος αριθμός ζωής σε κάθε γύρο.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

0 references
public class EnemyStats : PlayerStats
{
    2 references
    public float damage;
    1 reference
    public float atkSpeed;
    1 reference
    public bool canAttack;
    1 reference
    GameManagment game;


    0 references
    private void Start()
    {
        Init();
    }

    0 references
    public void DealDamage(PlayerStats Dmg)
  {
    Dmg.TakeDamage(damage);
  }

    0 references
    public override void Die  () {
        base.Die();
        Destroy(gameObject);
    }

    1 reference
    public override void Init()
    {
        MaxHealth = 50f + (1.25f * game.round);
        SetHealth(MaxHealth);
        death = false;

        damage = 25;
        atkSpeed = 1.5f;
        canAttack = true;
    }
```

Image 35: Κώδικας Ζόμπι No. 1.

```
using System.Collections;
using UnityEngine;

0 references
public class Enemy: MonoBehaviour
{
  //Navigation
    6 references
    UnityEngine.AI.NavMeshAgent nav;
    4 references
    Transform player;
    2 references
    Animator controller;
    2 references
    Animator atkAnim;

    0 references
    public float speed;
    2 references
    CapsuleCollider capsuleCollider;
    2 references
    GameManagment game;

    //Health kai death
     3 references
    public float health;
     2 references
    Animator death;
     2 references
    bool isDead = false;

     2 references
    public GameObject[] pickups;

```

Image 36: Κώδικας Ζόμπι No. 2.

```
    //Attack
     0 references
    public float TimeToAttack;
     0 references
    public float attackTimer;

     0 references
    public bool canAttack = true;

     1 reference
    public int attackDmg=25;
     2 references
    private EnemyStats stats = null;

    1 reference
    public float attackCooldown = 2.0f;
    1 reference
    public float attackRange = 1.5f;
    2 references
    private float lastAttackTime;
```

Image 37: Κώδικας Ζόμπι No. 3.

```
38
39
40      // Start is called before the first frame update
        0 references
41      void Awake()
42      {
43          nav = GetComponent<UnityEngine.AI.NavMeshAgent>();
44          player = GameObject.FindGameObjectWithTag("Player").transform;
45          controller = GetComponentInParent<Animator>();
46          game = FindObjectOfType<GameManagment>();
47          health = 50f + (10f * game.round);
48          capsuleCollider=GetComponent<CapsuleCollider>();
49          death= GetComponentInParent<Animator>();
50          stats = GetComponent<EnemyStats>();
51          atkAnim= GetComponentInParent<Animator>();
52          nav.speed=0.5f +Random.Range(0f,4f);
53
54      }
55
56      // Update is called once per frame
        0 references
57      void Update()
58      {
59          if(!isDead)
60          {
61          nav.SetDestination(player.position);
62          controller.SetFloat("speed",Mathf.Abs(nav.velocity.x+nav.velocity.z));
63          //float distance = Vector3.Distance(player.position,transform.position);
64          float distance = Vector3.Distance(transform.position, player.transform.position);
65          if (distance <= attackRange && Time.time - lastAttackTime >= attackCooldown)
66          {
67              //atkAnim.SetFloat("Speed",0f);
68             PlayerStats targerstats = player.GetComponent<PlayerStats>();
69              Attack(targerstats);
70              lastAttackTime = Time.time;
71          }
72
```

Image 38: Κώδικας Ζόμπι No. 4.

```
0 references
public void TakeDamage(float amount)
{
 health-= amount;
 if(health<= 0f)
 {
     Die();
 }
}

1 reference
void Die()
{
  isDead = true;
  //death Animation
  death.SetTrigger("isDead");
  nav.isStopped = true;//stop navigation
  //bullet won't stop if hit
 capsuleCollider.enabled = false;
 GameManagment.zombiesLeftInRound --;
  Destroy(gameObject,1f);

  int randomDrop = Random.Range(0,10);
  int randomPickUps = Random.Range(0,pickups.Length-1);

  if(randomDrop < 4)
  {
   Instantiate(pickups[randomPickUps],transform.position + new Vector3(0,0.25f,0),Quaternion.Euler(0,0,0));
  }
 }
0 references
public void DealDamage(PlayerStats Dmg)
{
  Dmg.TakeDamage(attackDmg);
}
1 reference
public void Attack(PlayerStats Atk)
{
     atkAnim.SetTrigger("attack");
     stats.DealDamage(Atk);

}
}
```

Image 39: Κώδικας Ζόμπι No. 5.

Το πρώτο κομμάτι κώδικα χρησιμοποιείται για να δώσει χαρακτηριστικά στα ζόμπι ενώ το δεύτερο χρησιμοποιείται για να δώσει ζωή στο ζόμπι και να ενώσει όλα τα παραπάνω κομμάτια. Navigation, speed, animations, κ.λπ. Όλα ελέγχονται από τον δεύτερο κώδικα.

4.2.6 Game Management

Χρησιμοποιείται για να δώσει ζωή στο ίδιο το παιχνίδι. Είναι ο τρόπος με τον οποίο τα ζόμπι εμφανίζονται στο χάρτη, μετράει τον τωρινό γύρο στο παιχνίδι και μέσω αυτού βλέπουμε το πότε χάσαμε.

```
using System.Collections;
using UnityEngine;
using TMPro;
using UnityEngine.SceneManagement;


0 references
public class GameManagment: MonoBehaviour
{

    3 references
    public int round=1;
    2 references
    int zombiesInRound=10;
    3 references
    public static int zombiesLeftInRound=10;
    3 references
    int zombiesSpawnInRound=0;
    3 references
    float zombiesSpawnTimer=0;
    6 references
    float countdown=0;
    2 references
    public Transform[] zombiesSpawnPoint;
    1 reference
    public GameObject zombieEnemy;
    1 reference
    public TextMeshProUGUI zombiesLeftText;
    1 reference
    public TextMeshProUGUI countdownText;
    1 reference
    public TextMeshProUGUI roundText;

    2 references
    public GameObject gameOverUI;
```

Image 40: Κώδικας Game management No. 1.

```
    void FixedUpdate()
    {
        countdownText.text = countdown.ToString();
        zombiesLeftText.text= zombiesLeftInRound.ToString();
        roundText.text = round.ToString();
        if(zombiesSpawnInRound < zombiesInRound && countdown==0)
        {
            if(zombiesSpawnTimer>30)
            {
                SpawnZombie();
                zombiesSpawnTimer=0;
            }
            else{
                zombiesSpawnTimer++;
            }
        }
        else if(zombiesLeftInRound==0)
        {
            StartNextRound();
        }
        if(countdown > 0)
        {
            countdown-=Time.deltaTime;
        }
        else
         countdown=0;
    }

```

Image 41: Κώδικας Game Management No. 2.

```
        1 reference
51      void SpawnZombie()
52      {
53          Vector3 randomSpawnPoint = zombiesSpawnPoint[Random.Range(0,zombiesSpawnPoint.Length)].position;
54          Instantiate(zombieEnemy,randomSpawnPoint,Quaternion.identity);
55          zombiesSpawnInRound++;
56
57      }
58
        1 reference
59      void StartNextRound()
60      {
61          zombiesInRound = zombiesLeftInRound = 10 + (round*2);
62          zombiesSpawnInRound=0;
63          countdown=7;
64          round++;
65      }
66
        0 references
67      public void gameOver()
68      {
69          //Time.timeScale = 0;
70          gameOverUI.SetActive(true);
71          //Screen.showCursor=true;
72      }
73
        0 references
74      public void restart()
75      {
76          //Debug.Log("restart");
77          gameOverUI.SetActive(false);
78          //Time.timeScale = 1;
79          SceneManager.LoadScene("Demo_Scene");
80        }
81      //public void mainMenu()
82      //{
83          //SceneManager.LoadScene("MainMenu");
84      // }
85
        0 references
86      public void quit()
87      {
88       Application.Quit();
89      }
90  }
```

Image 42: Κώδικας Game Management No. 3.

4.2.7 Pick Up Items (AmmoBox)

Η γενική ιδέα είναι ότι τα ζόμπι θα ρίχνουν αντικείμενα όταν πεθαίνουν, τα οποία θα βοηθούν το παίχτη. Στη περίπτωση του παιχνιδιού αυτού, τα ζόμπι ρίχνουν ένα κουτί με σφαίρες το οποίο γεμίζει σφαίρες σε όλα τα όπλα του παίχτη.

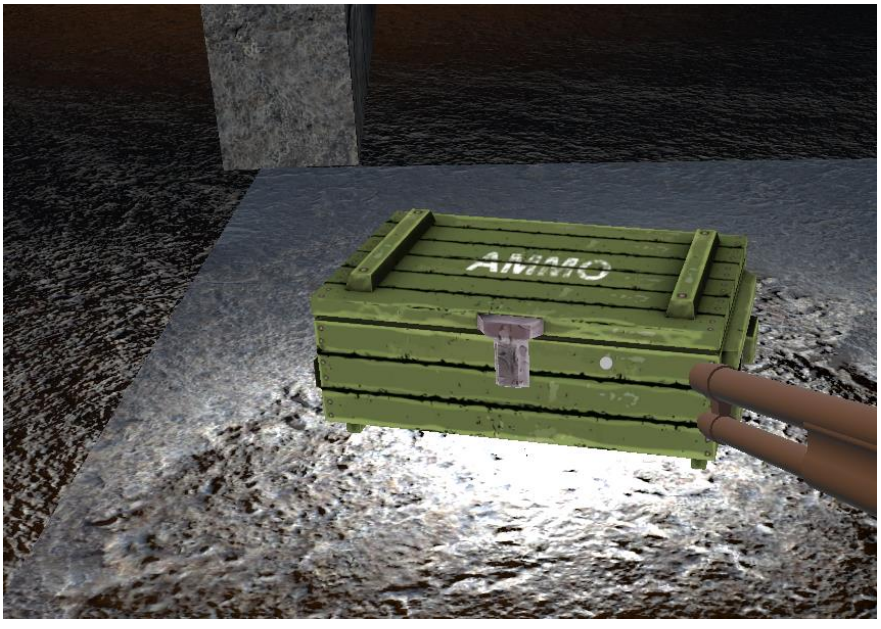


Image 43: AmmoBox.

```
1   using System;
2   using System.Collections;
3   using UnityEngine;
4
    0 references
5   public class PickUpItem : MonoBehaviour
6   {
        3 references
7       public float timer;
        3 references
8       public float flashTimer;
        1 reference
9       public float flashDuration;
        2 references
10      public float spinSpeed;
        3 references
11      public float rotation;
        0 references
12      void Start()
13      {
14          //start
15
16      }
17
```

Image 44: Κώδικας Pick Up Item No. 1.

```
17
    0 references
18  void Update()
19  {
20      timer -= Time.deltaTime;
21
22      if(timer>10f)
23      {
24          rotation += spinSpeed * Time.deltaTime;
25
26      }
27      else if(timer>0)
28      {
29          flashTimer -= Time.deltaTime;
30          if(flashTimer<=0)
31          {
32              flashTimer=flashDuration;
33              Flash();
34          }
35          rotation += 2* spinSpeed * Time.deltaTime;
36
37      }
38      else
39      {
40          Destroy(gameObject);
41      }
42      transform.rotation = Quaternion.Euler(0,rotation,0);
43
44  }
45
    1 reference
46  void Flash()
47  {
48      MeshRenderer renderer= GetComponentInParent<MeshRenderer>();
49      renderer.enabled = !renderer.enabled;
50      Light light =GetComponentInParent<Light>();
51      light.enabled = !light.enabled;
52  }
```

Image 45: Κώδικας Pick Up Item No. 2.

```
        1 reference
46      void Flash()
47      {
48          MeshRenderer renderer= GetComponentInParent<MeshRenderer>();
49          renderer.enabled = !renderer.enabled;
50          Light light =GetComponentInParent<Light>();
51          light.enabled = !light.enabled;
52      }
        0 references
53      public void DisablePickUp()
54      {
55          MeshRenderer renderer= GetComponentInParent<MeshRenderer>();
56          renderer.enabled = false;
57          Light light =GetComponentInParent<Light>();
58          light.enabled = false;
59          BoxCollider collider =GetComponentInParent<BoxCollider>();
60          collider.enabled=false;
61          Destroy(gameObject);
62      }
63  }
64
```

Image 46: Κώδικας Pick Up Item No. 3.

4.2.8 Player/Character

Στα χέρια του παίχτη δίνεται ένας χαρακτήρας. Ο χαρακτήρας είναι αυτός που ελέγχουμε για να παίζουμε το παιχνίδι. Ο χαρακτήρας για είναι ως ένα σημείο ολοκληρωμένος χρειάζεται πολλά χαρακτηρίστηκα.

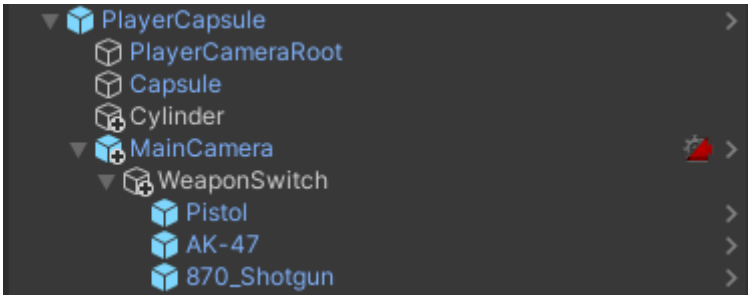


Image 47: Ιεραρχία του χαρακτήρα.

Όλα τα χαρακτηριστικά βρίσκονται κάτω από ένα Player Capsule. Το Player Capsule από μόνο του έχει: Την κίνηση του παίχτη, ένα RigidBody, κώδικες για την τωρινή και συνολική ζωή του παίχτη και ένα character controller για την πιο εύκολη κίνηση του χαρακτήρα.



Image 48: Χαρακτηριστικά χαρακτήρα No. 1.

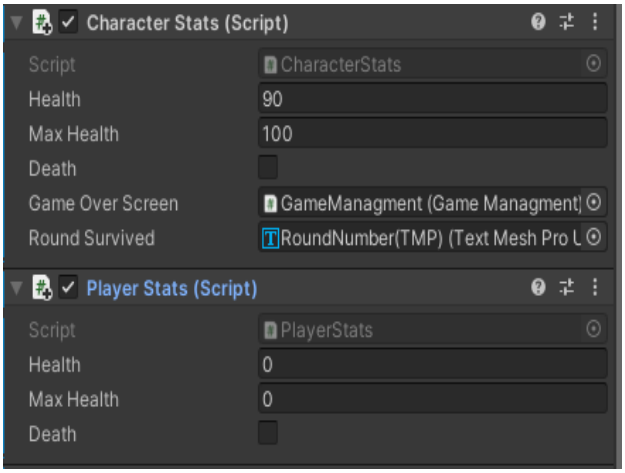


Image 49: Χαρακτηριστικά χαρακτήρα No. 2.

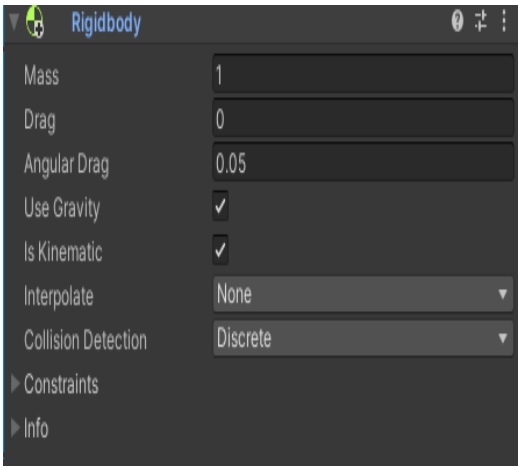


Image 50: Rigidbody χαρακτήρα.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[RequireComponent(typeof(CharacterController))]

0 references
public class SC_FPSController : MonoBehaviour
{
    2 references
    public float walkingSpeed = 7.5f;
    2 references
    public float runningSpeed = 11.5f;
    1 reference
    public float jumpSpeed = 8.0f;
    1 reference
    public float gravity = 20.0f;
    1 reference
    public Camera playerCamera;
    2 references
    public float lookSpeed = 2.0f;
    2 references
    public float lookXLimit = 45.0f;

    4 references
    CharacterController characterController;
    6 references
    Vector3 moveDirection = Vector3.zero;
    4 references
    float rotationX = 0;

    [HideInInspector]
    4 references
    public bool canMove = true;

    0 references
    void Start()
    {
        characterController = GetComponent<CharacterController>();

        // Lock cursor
        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
    }

```

Image 51: Κώδικας χαρακτήρα No. 1.

```
    void Update()
    {
        Vector3 forward = transform.TransformDirection(Vector3.forward);
        Vector3 right = transform.TransformDirection(Vector3.right);
        // Press Left Shift to run
        bool isRunning = Input.GetKey(KeyCode.LeftShift);
        float curSpeedX = canMove ? (isRunning ? runningSpeed : walkingSpeed) * Input.GetAxis("Vertical") : 0;
        float curSpeedY = canMove ? (isRunning ? runningSpeed : walkingSpeed) * Input.GetAxis("Horizontal") : 0;
        float movementDirectionY = moveDirection.y;
        moveDirection = (forward * curSpeedX) + (right * curSpeedY);

        if (Input.GetButton("Jump") && canMove && characterController.isGrounded)
        {
            moveDirection.y = jumpSpeed;
        }
        else
        {
            moveDirection.y = movementDirectionY;
        }

```

Image 52: Κώδικας χαρακτήρα No. 2.

```
56          if (!characterController.isGrounded)
57          {
58              moveDirection.y -= gravity * Time.deltaTime;
59          }
60
61          // Move the controller
62          characterController.Move(moveDirection * Time.deltaTime);
63
64          // Player and Camera rotation
65          if (canMove)
66          {
67              rotationX += -Input.GetAxis("Mouse Y") * lookSpeed;
68              rotationX = Mathf.Clamp(rotationX, -lookXLimit, lookXLimit);
69              playerCamera.transform.localRotation = Quaternion.Euler(rotationX, 0, 0);
70              transform.rotation *= Quaternion.Euler(0, Input.GetAxis("Mouse X") * lookSpeed, 0);
71          }
72      }
73  }
```

Image 53: Κώδικας χαρακτήρα No. 3.

Στην πρώτη φωτογραφία βλέπουμε τις μεταβλητές μας καθώς τον τρόπο με τον οποίο αφαιρούμε το ποντίκι από την οθόνη όσο παίζουμε.

Στη δεύτερη φωτογραφία βλέπουμε το πώς περπατάει ο χαρακτήρας, τον τρόπο με τον οποίο τρέχει μαζί με το κουμπί που πατάμε για να τρέξει (Left Shift) καθώς και το πώς πηδάει στον αέρα επίσης μαζί με το κουμπί (Jump Button/Space).

Τέλος στην τρίτη φωτογραφία βλέπουμε το πώς ο χαρακτήρας κατεβαίνει ξανά από τον αέρα μετά το jump καθώς και το τρόπο με τον οποίο φτιάχνουμε το ποντίκι να κουνάει την κάμερα.

```
1   using System.Net.Mime;
2   using System.Collections;
3   using System.Collections.Generic;
4   using UnityEngine;
5   using UnityEngine.UI;
6   using TMPro;
7
    0 references
8   public class PlayerHUD : MonoBehaviour
9   {
        1 reference
10      [SerializeField] private TextMeshProUGUI  currentHealthText;
        1 reference
11      [SerializeField] private TextMeshProUGUI  maxHealthText;
12
        0 references
13      public void UpdateHealth(float currentHealth , float maxHealth)
14      {
15          currentHealthText.text = currentHealth.ToString();
16          maxHealthText.text = maxHealth.ToString();
17      }
18  }
19
```

Image 54: Κώδικας χαρακτήρα.

Εδώ βλέπουμε τον κώδικα με τον οποίο φαίνεται η ζωή του χαρακτήρα στην οθόνη.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

0 references
public class PlayerStats : MonoBehaviour
{
    6 references
    public float health;
    4 references
    public float MaxHealth;
    2 references
    public bool death;


    0 references
    void Start()
    {
       //MaxHealth=100;
       //SetHealth(MaxHealth);
       //death=false;
     Init();
    }
    1 reference
    public virtual void CheckHealth()
    {
     if(health <=0f)
     {
        health = 0f;
        Die();

     }
     if(health>= MaxHealth)
     {
        health = MaxHealth;
     }
    }
    1 reference
```

Image 55: Κώδικας μέτρησης της ζωής No. 1.

```
    public virtual void Die()
    {
        death=true;
    }
    2 references
    public void SetHealth(float healSet)
    {
     health  = healSet;
     CheckHealth();
    }

    0 references
    public void TakeDamage(float Damage)
    {
      float remainHp;
      remainHp = health-Damage;
      SetHealth(remainHp);
    }
    1 reference
    public virtual void Init()
    {
       MaxHealth=100f;
       SetHealth(MaxHealth);
       death=false;
    }
}
```

Image 56: Κώδικας μέτρησης της ζωής No. 2.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using TMPro;
0 references
public class CharacterStats : PlayerStats
{
    2 references
    private PlayerHUD hud;

    1 reference
    public GameManagment gameOverScreen;
    0 references
    public TextMeshProUGUI roundSurvived;


    0 references
    private void Start()
    {
        GetReferences();
        Init();
    }
    1 reference
    private void GetReferences()
    {
        hud = GetComponent<PlayerHUD>();
    }

    0 references
    public override void CheckHealth()
    {
        base.CheckHealth();
        hud.UpdateHealth(health,MaxHealth);
    }
```

Image 57: Κώδικας θανάτου No. 1.

```
    void Update()
    {
    if(death)
        {
         gameOverScreen.gameOver();
         if(Input.GetKeyDown(KeyCode.N))
        {
          SceneManager.LoadScene("Demo_Scene");
        }
        if(Input.GetKeyDown(KeyCode.E))
        {
          Application.Quit();
        }
        }
         }
}
```

Image 58: Κώδικας Game Over.



Image 59: Game Over view

Στον πρώτο κώδικα βλέπουμε τον πως φτιάχνεται η ζωή του χαρακτήρα, τον τρόπο που χάνει την ζωή και το ότι πεθαίνει όταν τελειώσει.

Στον δεύτερο κώδικα βλέπουμε ότι παίρνει τα βασικά στοιχεία του πρώτου κώδικα και τα μεταφέρει στην οθόνη.

4.2.9 Κάμερα

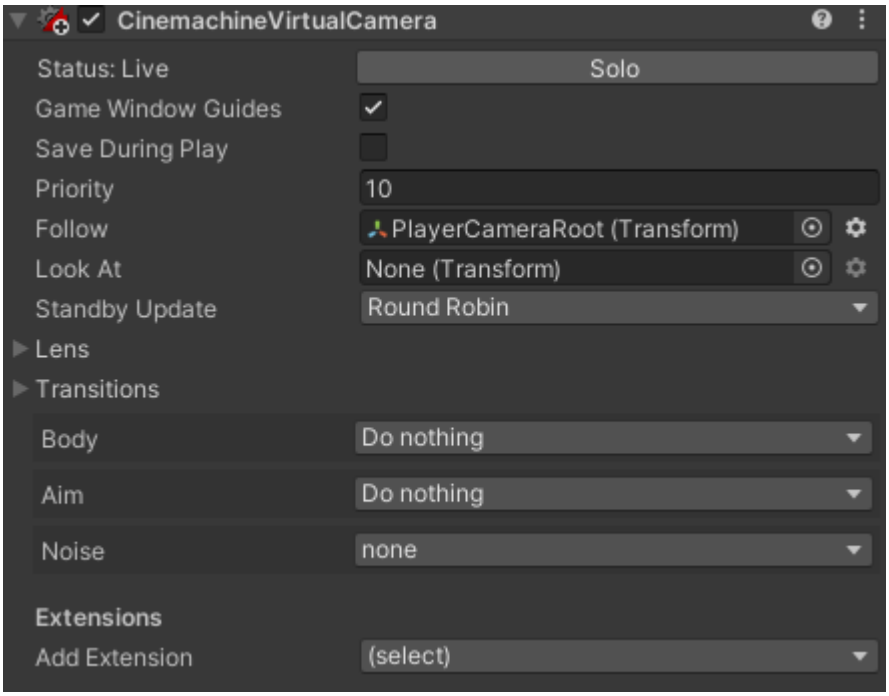


Image 60: Χαρακτηριστικά Κάμερας.

Η κάμερα είναι τοποθετημένη έτσι ώστε να χρησιμοποιείται σαν το κεφάλι του χαρακτήρα και προχωράει μαζί με αυτόν. Επίσης το Weapon Switch είναι τοποθετημένο μαζί της για να μπορούμε να βλέπουμε τα όπλα και την αλλαγή τους.

4.2.10 Map/ Χάρτης

Το map ή αλλιώς χάρτης είναι το μέρος στο οποίο διαδραματίζεται το παιχνίδι. Λειτουργεί σαν scene. Ο παίκτης αλλά και οι αντίπαλοι του μπορούν να μετακινηθούν σε σχεδόν οποιοδήποτε σημείο του χάρτη. Για χάρτη χρησιμοποίησα ένα έτοιμο asset της Unity, το decrepit dungeon lite και το επεξεργάστηκα για να έρθει στα μετρά που χρειαζόμουν για το παιχνίδι.



Image 61: Map view No. 1


Image 62: Map view No. 2

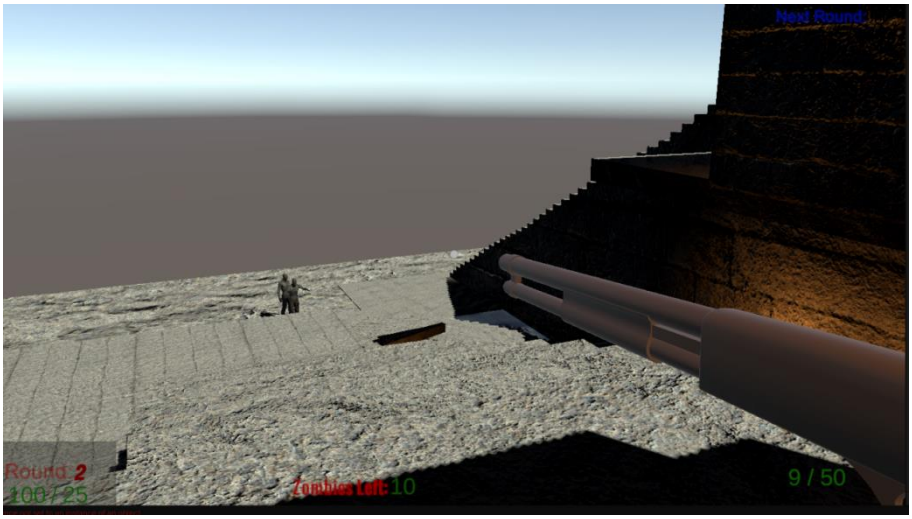


Image 63: Map view No. 3

Βιβλιογραφία:

- Μετρεβέλι, Ν., 2023, *Ανάπτυξη 3D σκοπευτικού παιχνιδιού τρίτου προσώπου πολλαπλών παικτών τύπου Roguelite*, Διπλωματική εργασία [online], Κοζάνη.
Διαθέσιμο στο: https://dspace.uowm.gr/xmlui/bitstream/handle/123456789/4199/diplwmatikh_metreveli.pdf?sequence=1&isAllowed=y
Πρόσβαση στις: 12/ 12/ 2023

- Κρεζίας, Χ., 2016, *Ανάπτυξη βιντεοπαιχνιδιού στρατηγικής με χρήση της μηχανής Unity*, Πτυχιακή εργασία [online], Αθήνα.
Διαθέσιμο στο: https://scholar.google.com/scholar?hl=el&as_sdt=0%2C5&q=+%CE%91%CE%BD%CE%AC%CF%80%CF%84%CF%85%CE%BE%CE%B7+%CE%B2%CE%B9%CE%BD%CF%84%CE%B5%CE%BF%CF%80%CE%B1%CE%B9%CF%87%CE%BD%CE%B9%CE%B4%CE%B9%CE%BF%CF%8D+%CF%83%CF%84%CF%81%CE%B1%CF%84%CE%B7%CE%B3%CE%B9%CE%BA%CE%AE%CF%82+%CE%BC%CE%B5+%CF%87%CF%81%CE%AE%CF%83%CE%B7+%CF%84%CE%B7%CF%82+%CE%BC%CE%B7%CF%87%CE%B1%CE%BD%CE%AE%CF%82+Unity&btnG=
Πρόσβαση στις: 12/ 12/ 2023

- Τέρπος, Ι., χ.χ., *Ψηφιακό παιχνίδι εικονικής πραγματικότητας με Unity και C# για την εκμάθηση Logo*, Διπλωματική εργασία [online], χ.τ.
Διαθέσιμο στο: https://scholar.google.com/scholar?hl=el&as_sdt=0%2C5&q=%CE%A8%CE%B7%CF%86%CE%B9%CE%B1%CE%BA%CF%8C+%CF%80%CE%B1%CE%B9%CF%87%CE%BD%CE%AF%CE%B4%CE%B9+%CE%B5%CE%B9%CE%BA%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE%CF%82+%CF%80%CF%81%CE%B1%CE%B3%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CF%8C%CF%84%CE%B7%CF%84%CE%B1%CF%82+%CE%BC%CE%B5+Unity+%CE%BA%CE%B1%CE%B9+C%23+%CE%B3%CE%B9%CE%B1+%CF%84%CE%B7%CE%BD+%CE%B5%CE%BA%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7+Logo%2C+%CE%94%CE%B9%CF%80%CE%BB%CF%89%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%AE+%CE%B5%CF%81%CE%B3%CE%B1%CF%83%CE%AF%CE%B1&btnG=
Πρόσβαση στις: 16/ 12/ 2023

- Βούλγαρη, Η., Ροϊνιώτη, Ε., Κουτρομάνος, Γ. Ι., Σιντόρης, Χ., και Μάνεσης, Δ., (2024), *Ψηφιακά παιχνίδια και μάθηση*, [προπτυχιακό εγχειρίδιο], Κάλλιπος, Ανοικτές Ακαδημαϊκές Εκδόσεις.

- Πανανός, Μ., 2022, *Game design with Unity using machine learning agents, Πτυχιακή εργασία*, [online], Πάτρα.

Διαθέσιμο στο: https://scholar.google.com/scholar?hl=el&as_sdt=0%2C5&q=+Game+design+with+Unity+using+machine+learning+agents&btnG=
Πρόσβαση στις: 17/ 1/ 2024

- Μαραγκοπούλου, Ε., 2018, *Στα μονοπάτια της αρχαίας Ολύνθου... με τη βοήθεια του Unity 3D*, Διπλωματική εργασία, [online], Θεσσαλονίκη.
Διαθέσιμο στο: https://scholar.google.com/scholar?hl=el&as_sdt=0%2C5&q=+%CE%A3%CF%84%CE%B1+%CE%BC%CE%BF%CE%BD%CE%BF%CF%80%CE%AC%CF%84%CE%B9%CE%B1+%CF%84%CE%B7%CF%82+%CE%B1%CF%81%CF%87%CE%B1%CE%AF%CE%B1%CF%82+%CE%9F%CE%BB%CF%8D%CE%BD%CE%B8%CE%BF%CF%85...+%CE%BC%CE%B5+%CF%84%CE%B7+%CE%B2%CE%BF%CE%AE%CE%B8%CE%B5%CE%B9%CE%B1+%CF%84%CE%BF%CF%85+Unity+3D&btnG=
Πρόσβαση στις: 17/ 1/ 2024

- Φατσέα, Α., Αντωνίου, Π., (2012), *"Καταγραφή της επίδρασης των ψηφιακών διαδραστικών παιχνιδιών στους χρήστες"*, Πρακτικά 2ου Πανελλήνιου Εκπαιδευτικού Συνεδρίου Ημαθίας, Βέροια.

- Σουσαμλής, Π., 2011, *Υλοποίηση τρισδιάστατων παιχνιδιών σε πολλαπλές πλατφόρμες, Διπλωματική διατριβή*, [online], χ.τ.
Διαθέσιμο στο: https://scholar.google.com/scholar?hl=el&as_sdt=0%2C5&q=%CE%A5%CE%BB%CE%BF%CF%80%CE%BF%CE%AF%CE%B7%CF%83%CE%B7+%CF%84%CF%81%CE%B9%CF%83%CE%B4%CE%B9%CE%AC%CF%83%CF%84%CE%B1%CF%84%CF%89%CE%BD+%CF%80%CE%B1%CE%B9%CF%87%CE%BD%CE%B9%CE%B4%CE%B9%CF%8E%CE%BD+%CF%83%CE%B5+%CF%80%CE%BF%CE%BB%CE%BB%CE%B1%CF%80%CE%BB%CE%AD%CF%82+%CF%80%CE%BB%CE%B1%CF%84%CF%86%CF%8C%CF%81%CE%BC%CE%B5%CF%82%2C+%CE%94%CE%B9%CF%80%CE%BB%CF%89%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%AE+%CE%B4%CE%B9%CE%B1%CF%84%CF%81%CE%B9%CE%B2%CE%AE&btnG=
Πρόσβαση στις: 20/ 1/ 2024

- Σπέντζας, Ι., 2015, *Ανάπτυξη βιντεοπαιχνιδιών με χρήση της Game Engine Unity*, Διπλωματική εργασία, [online], Αθήνα.
Διαθέσιμο στο: https://scholar.google.com/scholar?hl=el&as_sdt=0%2C5&q=+%CE%91%CE%BD%CE%AC%CF%80%CF%84%CF%85%CE%BE%CE%B7+%CE%B2%CE%B9%CE%BD%CF%84%CE%B5%CE%BF%CF%80%CE%B1%CE%B9%CF%87%CE%BD%CE%B9%CE%B4%CE%B9%CF%8E%CE%BD+%CE%BC%CE%B5+%CF%87%CF%81%CE%AE%CF%83%CE%B7+%CF%84%CE%B7%CF%82+Game+Engine+Unity&btnG=
Πρόσβαση στις: 21/ 1/ 2024

- Βασιλόπουλος, Στ., 2021, *"Σχεδιασμός και ανάπτυξη 2D video game σε Unity Engine με χρήση C#"*, Διπλωματική εργασία, [online], Πάτρα.
Διαθέσιμο στο: https://scholar.google.com/scholar?hl=el&as_sdt=0%2C5&q=%E2%80%9C%CE%A3%CF%87%CE%B5%CE%B4%CE%B9%CE%B1%CF%83%CE%BC%CF%8C%CF%82+%CE%BA%CE%B1%CE%B9+%CE%B1%CE%BD%CE%AC%CF%80%CF%84%CF%85%CE%BE%CE%B7+2D+video+game+%CF%83%CE%B5+Unity+Engine+%CE%BC%CE%B5+%CF%87%CF%81%CE%AE%CF%83%CE%B7+C%23&btnG=
Πρόσβαση στις: 2/ 2/ 2024

- Cheliotis, K., *ABMU: An Agent-Bases Modelling Framework for Unity 3D*, 2021, University College London.

- Okita, A., *Learning C# Programming with Unity 3D,* AK Peters/CRC Press, 2020.

- Ferrone, Harrison. *Learning C# by Developing Games with Unity 2021: Kickstart your C# programming and Unity journey by building 3D games from scratch*. Packt Publishing Ltd, 2021.

- Φωτογραφίες: Google Photos