



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Διπλωματική Εργασία

Ανάπτυξη Βιντεοπαιχνιδιού Εικονικής Πραγματικότητας
για τη Διδασκαλία του Δυναμικού Συστήματος

Απόστολος-Νικόλαος Ρήγας

Επιτροπή διπλωματικής εργασίας

Αικατερίνη Μανιά
Καθηγήτρια,
Πολυτεχνείο Κρήτης

Μιχαήλ Λαγουδάκης
Καθηγητής,
Πολυτεχνείο Κρήτης

Αντώνιος Δεληγιαννάκης
Καθηγητής,
Πολυτεχνείο Κρήτης

Χανιά, Νοέμβριος 2023



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Diploma Thesis

Development of a Virtual Reality Game to Reinforce Teaching of Binary Arithmetic System

Apostolos-Nikolaos Rigas

Thesis Committee

Prof. Aikaterini Mania
Technical University of
Crete

Prof. Michail Lagoudakis
Technical University of
Crete

Prof. Antonios Deligiannakis
Technical University of Crete

Chania, November 2023

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ρήγα Απόστολου-Νικολάου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο ΠΚ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίας στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.

ΕΥΧΑΡΙΣΤΙΕΣ...

Ευχαριστίες από καρδιάς οφείλω στην επιβλέπουσα καθηγήτρια κα. Μανιά για την καθοδήγηση της καθ' όλη την διάρκεια της συγγραφής της εργασίας καθώς και στον κ. Μουμουτζή για την έμπνευση και συνεχή επίβλεψη της εργασίας. Δεν θα μπορούσα να παραλείψω να ευχαριστήσω και τα μέλη της τριμελούς επιτροπής κ. Λαγουδάκη και κ. Δεληγιαννάκη για τις πολύτιμες υποδείξεις τους για βελτίωση της εργασίας.

Εξαιρετικά θα ήθελα να ευχαριστήσω τον πνευματικό μου πατέρα, Επίσκοπο Ευμενίας κ. κ. Ειρηναίο Βερυκάκη και την συνοδεία του, την οικογένειά μου, τον κ. Υφαντίδη, καθώς και τις κα. Γιαννουδάκη και κα. Πέτρου, χωρίς την υπομονή και στήριξη των οποίων δεν θα μπορούσα να φτάσω ως εδώ.

ΠΕΡΙΛΗΨΗ

Οι υπολογιστές, προκειμένου να εκτελέσουν τους διάφορους υπολογισμούς, βασίζονται στο δυαδικό σύστημα αρίθμησης. Πράγματι, για να εκτελεστεί κάθε πρόγραμμα που εκτελεί ο οποιοσδήποτε υπολογιστής, αυτό μεταγλωττίζεται σε γλώσσα μηχανής, η οποία βασίζεται στο δυαδικό σύστημα αρίθμησης. Οι χρήστες των ηλεκτρονικών υπολογιστών όμως δυσκολεύονται να το κατανοήσουν με αποτέλεσμα να περιορίζονται απλά στη χρήση των υπολογιστών και όχι στην εμβάθυνση του τρόπου λειτουργίας τους.

Στόχος της παρούσας πτυχιακής εργασίας είναι η δημιουργία ενός τρισδιάστατου παιχνιδιού με τη χρήση της δημοφιλούς μηχανής παιχνιδιών Unity που θα έχει σαν σκοπό την διδασκαλία του δυαδικού συστήματος σε παιδιά του Δημοτικού – Γυμνασίου. Η διδασκαλία θα πραγματοποιείται χρησιμοποιώντας ένα από τα πλέον αγαπητά μέσα των νέων, το τρισδιάστατο βιντεοπαιχνίδι, έτσι ώστε να εγείρεται το ενδιαφέρον των παιδιών και η εκμάθηση του συστήματος να γίνεται με τρόπο ευχάριστο. Πιο συγκεκριμένα, οι μαθητές θα χειρίζονται έναν ήρωα, ο οποίος θα πρέπει να προστατευτεί από διάφορα μυθικά πλάσματα που τον απειλούν. Ο ήρωας για να προστατευτεί θα πρέπει να παρασκευάσει ένα μαγικό φίλτρο, χρησιμοποιώντας το δυαδικό σύστημα. Το φίλτρο αυτό θα του δίνει την απαραίτητη δύναμη, ώστε να αντιμετωπίσει τους εχθρούς του.

Για την ανάπτυξη του βιντεοπαιχνιδιού χρησιμοποιήθηκε η μηχανή δημιουργίας παιχνιδιών (game engine) της Unity. Κατά την ανάπτυξη, χρησιμοποιήθηκαν διάφορα εργαλεία που παρέχονται από τη συγκεκριμένη πλατφόρμα. Τέτοια είναι το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE), το οποίο διευκόλυνε την δημιουργία των αρχείων πηγαίου κώδικα, η μηχανή προσομοίωσης φυσικής, η οποία παρείχε στο σύστημα την δυνατότητα αληθοφάνειας και εμβύθισης δημιουργώντας την αίσθηση ότι τα επιμέρους στοιχεία του παιχνιδιού υπακούν στους νόμους της φυσικής. Τέλος, άλλα εργαλεία της πλατφόρμας που αξιοποιήθηκαν στην εργασία ήταν η μηχανή γραφικών για την απεικόνιση και τον ορθό φωτισμό της σκηνής μέσω της διασωλήνωσης γραφιστικής απεικόνισης (rendering pipeline) και εξειδικευμένων αλγορίθμων φωτισμού, αλλά και η μηχανή animation, η οποία παρείχε την δυνατότητα δημιουργίας μηχανών πεπερασμένων καταστάσεων για τον έλεγχο των υπό συνθήκη μεταβάσεων των επιμέρους σεναρίων animation. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την συγγραφή του πηγαίου κώδικα ήταν η C#.

ABSTRACT

Computers store information and process instructions to perform various actions and calculations using the binary numbering system. Any program run by a computer, from games on a console to code on an embedded system, is compiled into a machine language to run, using the binary system. However, computer users find the binary system difficult to understand, which consequently limits their use of the available technology and computer capabilities. They are unable to delve into the respective principles of operation.

The aim of this thesis is to create a three dimensional game using the popular game engine Unity to teach the binary system to children from primary through secondary school. The teaching will be carried out using a 3D video game, specifically selected to pique the interest of the adolescents, while simultaneously engaging the student in active learning. The game specifically requires students to handle a hero that must protect himself from various mythical creatures that threaten him. The hero must prepare a magic potion using the binary system in the process of protecting himself. This potion gives him the required strength to face his enemies, incidentally teaching the user/child/student the binary system.

The Unity game engine and a plethora of its tools were employed in designing the video game. Such tools include the integrated development environment (IDE), which enabled the composition of game's scripts, and the physics engine that contributed to the full immersion of the player into the game's world. Physics engine makes each individual component obey the rules of physics, thus fortifying the plausibility of gameplay. Other tools that were utilized include the graphics engine with its rendering pipeline and the specialized algorithms for scene triangulation and lighting. The animation tool was crucial, as it enabled the implementation of state machines to control the conditional transition between animation clips. The C# programming language was used throughout the development process.

ΕΡΕΥΝΗΤΙΚΕΣ ΔΗΜΟΣΙΕΥΣΕΙΣ

- Moumoutzis, N., Rigas, N. A., Xanthaki, C., Maragkoudakis, Y., Christodoulakis, S., Paneva-Marinova, D., & Pavlova, L. (2020). Employing Theatrical Games to Establish and Support Onlife Learning Communities on Mathematical Principles of Informatics. *Digital Presentation and Preservation of Cultural and Scientific Heritage*, 10, 19-44.
- Moumoutzis, N., Rigas, N. A., Xanthaki, C., Maragkoudakis, I., Christodoulakis, S., Paneva-Marinova, D., & Pavlova, L. (2021). The Design of a Serious Game to Enable the Exploration of the Binary System. In *Interactive Mobile Communication, Technologies and Learning* (pp. 897-906). Cham: Springer International Publishing.
- Moumoutzis, N., Rigas, N. A., Xanthaki, C., Maragkoudakis, Y., Christodoulakis, S., Paneva-Marinova, D., & Pavlova, L. (2021, April). Using the PerFECt framework to invent playful learning activities for exploring the binary system. In *2021 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1237-1246). IEEE.

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΡΕΥΝΗΤΙΚΕΣ ΔΗΜΟΣΙΕΥΣΕΙΣ	vii
ΠΕΡΙΕΧΟΜΕΝΑ	viii
ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ	x
ΠΙΝΑΚΑΣ ΓΡΑΦΗΜΑΤΩΝ	xii
ΚΕΦΑΛΑΙΟ 1 – ΕΙΣΑΓΩΓΗ	1
1.1 ΚΙΝΗΤΡΑ - ΣΥΜΒΟΛΗ	1
1.2. ΠΕΡΙΛΗΨΗ	2
1.3 ΔΙΑΡΘΡΩΣΗ ΚΕΦΑΛΑΙΩΝ	4
ΚΕΦΑΛΑΙΟ 2 - ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ	5
2.1 ΠΑΙΔΑΓΩΓΙΚΗ ΠΡΟΣΕΓΓΙΣΗ	5
2.1.1 Εισαγωγή	5
2.1.2 Αρχές σχεδιασμού που υιοθετήθηκαν από το πλαίσιο PerFECT	6
2.2 ΣΥΣΤΗΜΑΤΑ ΑΡΙΘΜΗΣΗΣ	9
2.2.1 Εισαγωγή	9
2.2.2. Είδη συστημάτων αρίθμησης	9
2.2.2.1 Σύστημα αξίας συμβόλου	10
2.2.2.2. Καταμετρητικό σύστημα	11
2.2.2.3. Θεσιακό σύστημα αρίθμησης	12
2.2.2.4. Δυαδικό σύστημα αρίθμησης	13
2.3 ΣΥΓΚΡΙΣΗ ΜΕ ΠΑΡΕΜΦΕΡΗ ΠΑΙΓΝΙΑ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΟΥ ΔΥΑΔΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ	14
2.3.1. Cisco Binary Game	14
2.3.2. Binary bonanza	15
2.3.3. Binary Dots	16
ΚΕΦΑΛΑΙΟ 3 - ΜΗΧΑΝΕΣ ΠΑΙΧΝΙΔΙΩΝ	18
3.1. ΕΙΣΑΓΩΓΗ	18
3.2. UNREAL	19
3.3. CRYENGINE	20
3.4. UNITY	22
3.5. ΣΤΟΙΧΕΙΑ ΤΗΣ UNITY ΣΤΗΝ ΕΡΓΑΣΙΑ	23
3.5.1. Μηχανή γραφικών (rendering engine)	23
3.5.2. Διαχείριση εισόδου από τα περιφερειακά	24
3.5.3. Τεχνητή νοημοσύνη	25
3.5.4. Μηχανή εξομείωσης φυσικής	25
3.5.5. Μηχανή ήχου	26
3.5.6. Animation	26
3.5.7. Animator	26
3.5.8. Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE)	27
ΚΕΦΑΛΑΙΟ 4 - ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΑΙΧΝΙΔΙΟΥ	28
ΚΕΦΑΛΑΙΟ 5 - ΥΛΟΠΟΙΗΣΗ	34
5.1. FIRST PERSON CHARACTER	34
5.1.1. Εισαγωγή του μοντέλου	34
5.1.2. Δημιουργία animation	35
5.1.2.1. Animation Βαδίσματος	36
5.1.3. Χρωματισμός του μοντέλου	39
5.2. ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΑΡΧΙΚΗΣ ΠΙΣΤΑΣ	40
5.2.1. Δημιουργία Terrain	40
5.2.2. Χρωματισμός Terrain	41

5.2.3. Μορφολογία εδάφους	42
5.2.4. Διαμόρφωση περιβάλλοντος	44
5.2.5. Φωτισμός της σκηνής	46
5.3. ΔΗΜΙΟΥΡΓΙΑ ΦΙΛΤΡΟΥ	47
5.3.1. Συλλογή βοτάνων	47
5.4. ΕΠΙΛΟΓΗ ΟΠΛΟΥ	50
5.4.1. Εισαγωγή όπλων στη Unity	50
5.4.2. Εναλλαγή των όπλων	51
5.5. USER INTERFACE	53
5.5.1. Δημιουργία Καμβά	53
5.5.2. Εμφάνιση - Απόκρυψη Καμβά	56
5.6. ΑΠΟΣΤΟΛΕΣ	58
5.6.1. Αντικείμενο Κανάτα	59
5.6.2. Αντικείμενο Camera – Χειρισμός οθόνης	63
5.6.3. Αντικείμενο GameManager – Χειρισμός Αποστολών	63
5.6.4. Εμφάνιση μηνυμάτων στην οθόνη με συγκεκριμένη χρονική διάρκεια	65
5.7. ΕΠΙΠΕΔΟ ΠΡΩΤΟ	67
5.7.1. Terrain	67
5.7.2. Εισαγωγή στη σκηνή του εχθρικού χαρακτήρα	68
5.7.3. Επεξεργασία του μοντέλου	70
5.7.4. Νοημοσύνη στον χαρακτήρα του αντιπάλου	73
5.7.5. Animation στο χαρακτήρα	77
5.7.6. Καθορισμός Avatar	77
5.7.7. Animation Controller	78
5.7.8. Πρόσβαση στις μεταβλητές του Animation Controller	80
5.7.9. Επίθεση στον ήρωα - Πρόσβαση σε μεταβλητές διαφορετικού Script	81
5.7.10. Εντοπισμός χτυπήματος μέσω Raycast	83
5.8. PREFAB ΑΝΤΙΚΕΙΜΕΝΟ	84
5.9. SINGLETON ΑΝΤΙΚΕΙΜΕΝΟ	85
5.10. ΕΠΙΘΕΣΗ ΣΤΟ GOBLIN	86
5.10.1 Animation Controller	86
5.10.2. Υλοποίηση Επίθεσης	88
5.11. ΣΚΟΤΕΙΝΕΣ ΣΚΗΝΕΣ	91
5.12. ΕΠΙΠΕΔΟ ΔΕΥΤΕΡΟ	93
5.12.1. Η σκηνή	93
5.12.2 Διαχείριση πολλαπλών εχθρών	94
5.13. ΕΠΙΠΕΔΟ ΤΡΙΤΟ	97
5.14. ΜΕΝΟΥ ΚΑΙ ΔΙΣΔΙΑΣΤΑΤΑ ΓΡΑΦΙΚΑ ΠΙΣΤΑΣ	98
ΚΕΦΑΛΑΙΟ 6 - ΑΞΙΟΛΟΓΗΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ	102
6.1. ΕΙΣΑΓΩΓΗ	102
6.2. ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ	102
6.2.1. Περιγραφή	102
6.2.2. Αποτελέσματα	104
6.3. ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΕΛΕΓΧΟΥ ΓΝΩΣΕΩΝ	107
6.3.1. Περιγραφή	107
6.3.2. Αποτελέσματα	107
ΚΕΦΑΛΑΙΟ 7 - ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΙΛΟΓΟΣ	111
7.1. ΚΑΤΑΚΛΕΙΔΑ	111
7.2. ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ	112
ΒΙΒΛΙΟΓΡΑΦΙΑ	114
ΠΑΡΑΡΤΗΜΑ Α: ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ – UEQ	117

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 1 ΡΩΜΑΪΚΟ ΣΥΣΤΗΜΑ ΠΗΓΗ: HTTPS://WWW.SAUDIINSTITUTE.ORG/XXV-ROMAN-NUMERALS/	10
ΕΙΚΟΝΑ 2 ΣΥΣΤΗΜΑ ΑΡΙΘΜΗΣΗΣ "ΤΩΝ ΦΥΛΑΚΙΣΜΕΝΩΝ" ΠΗΓΗ: HTTPS://WWW.QUORA.COM/WHAT-IS-A-POSITIONAL-NUMBER-SYSTEM	11
ΕΙΚΟΝΑ 3 ΣΤΙΓΜΙΟΤΥΠΟ ΟΘΟΝΗΣ ΤΟΥ CISCO BINARY GAME	14
ΕΙΚΟΝΑ 4 ΣΤΙΓΜΙΟΤΥΠΟ ΟΘΟΝΗΣ BINARY BONANZA.	15
ΕΙΚΟΝΑ 5 Η ΠΛΑΚΕΤΑ ΔΡΑΣΤΗΡΙΟΤΗΤΩΝ BINARY DOTS.	16
ΕΙΚΟΝΑ 6 UNREAL IDE. ΠΗΓΗ: BY UNREAL ENGINE 4.20 RELEASED!, FAIR USE, HTTPS://EN.WIKIPEDIA.ORG/W/INDEX.PHP?CURID=57933430	20
ΕΙΚΟΝΑ 7 CRYENGINE SDK. ΠΗΓΗ: WWW.GEEKS3D.COM	21
ΕΙΚΟΝΑ 8 ΠΕΡΙΒΑΛΛΟΝ ΑΝΑΠΤΥΞΗΣ UNITY. ΠΗΓΗ: WWW.UNITY3D.COM	23
ΕΙΚΟΝΑ 9 ΔΙΕΠΑΦΗ ΕΡΓΑΛΕΙΟΥ ANIMATION	26
ΕΙΚΟΝΑ 10 ΑΠΟΨΗ ΤΗΣ ΚΑΛΥΒΑΣ ΚΑΙ ΤΟΥ ΚΗΠΟΥ - ΣΗΜΕΙΟ ΕΚΚΙΝΗΣΗΣ	28
ΕΙΚΟΝΑ 11 ΟΙ ΜΑΓΙΚΕΣ ΚΑΝΑΤΕΣ ΤΟΠΟΘΕΤΗΜΕΝΕΣ ΚΑΤΑ ΑΥΞΟΥΣΑ ΑΞΙΑ ΘΕΣΗΣ	29
ΕΙΚΟΝΑ 12 ΣΤΟ ΤΡΑΠΕΖΙ ΔΙΑΚΡΙΝΕΤΑΙ ΤΟ ΆΝΘΡΩΣ ΠΟΥ ΜΑΡΑΜΕΝΟ ΈΠΕΣΕ ΈΞΩ ΑΠΟ ΤΗΝ ΚΑΝΑΤΑ ΓΙΑΤΙ ΞΕΠΕΡΑΣΤΗΚΕ Η ΧΩΡΗΤΙΚΟΤΗΤΑ	30
ΕΙΚΟΝΑ 13 Ο ΧΡΗΣΤΗΣ ΣΥΛΛΕΓΕΙ ΤΑ ΜΑΓΙΚΑ ΆΝΘΗ ΑΠΟ ΤΟΝ ΚΗΠΟ ΤΟΥ	30
ΕΙΚΟΝΑ 14 ΚΑΘΩΣ Ο ΠΑΙΚΤΗΣ ΠΕΡΙΠΛΑΝΙΕΤΑΙ ΣΤΟ ΔΑΣΟΣ ΣΥΝΑΝΤΑ ΤΑ ΕΧΘΡΙΚΑ ΞΩΤΙΚΑ	31
ΕΙΚΟΝΑ 15 ΜΟΝΟΜΑΧΙΑ ΜΕ ΤΑ ΕΧΘΡΙΚΑ ΒΟΤ ΌΠΟΥ Ο ΠΑΙΚΤΗΣ ΕΊΝΑΙ ΚΡΙΣΙΜΑ ΤΡΑΥΜΑΤΙΣΜΕΝΟΣ	32
ΕΙΚΟΝΑ 16 Η ΔΙΕΠΑΦΗ ΤΟΥ ΧΡΗΣΤΗ ΓΙΑ ΤΗΝ ΕΠΙΣΤΡΟΦΗ ΣΤΟΝ ΚΗΠΟ ΚΑΙ ΠΑΡΑΣΚΕΥΗ ΦΙΛΤΡΟΥ ΟΤΑΝ ΧΡΕΙΑΣΤΕΙ	33
ΕΙΚΟΝΑ 17 ΕΙΣΑΓΩΓΗ ΜΟΝΤΕΛΟΥ ΧΑΡΑΚΤΗΡΑ.	34
ΕΙΚΟΝΑ 18 ΑΝΤΙΣΤΟΙΧΙΣΗ ΤΟΥ ANIMATOR CONTROLLER ΣΤΟ ΜΟΝΤΕΛΟ.	35
ΕΙΚΟΝΑ 19 Ο ANIMATOR CONTROLLER ΜΕ ΤΗ ΚΑΤΑΣΤΑΣΗ IDLE.	36
ΕΙΚΟΝΑ 20 Ο ANIMATION CONTROLLER ΜΕ ΤΗ ΝΕΑ ΚΑΤΑΣΤΑΣΗ ΚΑΙ ΤΙΣ ΣΥΝΘΗΚΕΣ ΜΕΤΑΒΑΣΗΣ.	37
ΕΙΚΟΝΑ 21 ΣΥΝΘΗΚΗ ΜΕΤΑΒΑΣΗΣ ΑΠΟ ΤΗΝ IDLE ΣΤΗ WALK.	37
ΕΙΚΟΝΑ 22 ΤΟ SCRIPT FIRSTPERSONKEYS.	38
ΕΙΚΟΝΑ 23 ΧΡΩΜΑΤΙΣΜΟΣ ΜΟΝΤΕΛΟΥ.	39
ΕΙΚΟΝΑ 24 ΤΟ ΜΟΝΤΕΛΟ ΣΤΗ ΤΕΛΙΚΗ ΤΟΥ ΜΟΡΦΗ.	39
ΕΙΚΟΝΑ 25 ΟΡΙΣΜΟΣ ΔΙΑΣΤΑΣΕΩΝ TERRAIN.	40
ΕΙΚΟΝΑ 26 ΧΡΩΜΑΤΙΣΜΟΣ ΤΟΥ TERRAIN.	41
ΕΙΚΟΝΑ 27 ΠΑΡΑΜΕΤΡΟΙ CHARACTER CONTROLLER.	42
ΕΙΚΟΝΑ 28 ΕΠΙΛΟΓΗ ΚΑΤΆΛΛΗΛΗΣ ΒΟΥΡΤΣΑΣ ΓΙΑ ΤΟΝ ΟΡΙΣΜΟ ΤΗΣ ΜΟΡΦΟΛΟΓΙΑΣ.	43
ΕΙΚΟΝΑ 29 ΟΛΟΚΛΗΡΩΣΗ ΤΗΣ ΜΟΡΦΟΛΟΓΙΑΣ.	43
ΕΙΚΟΝΑ 30 ΕΙΣΑΓΩΓΗ ΔΕΝΤΡΩΝ.	44
ΕΙΚΟΝΑ 31 ΤΟ TERRAIN ΜΕ ΤΑ ΔΕΝΤΡΑ.	45
ΕΙΚΟΝΑ 32 ΤΟ ΣΠΙΤΙ ΜΕ ΤΟΝ ΚΗΠΟ ΤΟΥ ΉΡΩΑ.	45
ΕΙΚΟΝΑ 33 ΦΩΤΙΣΜΟΣ ΣΚΗΝΗΣ.	46
ΕΙΚΟΝΑ 34 Ο ΚΗΠΟΣ ΤΟΥ ΉΡΩΑ.	47
ΕΙΚΟΝΑ 35 ΙΔΙΟΤΗΤΕΣ ΤΩΝ ΛΟΥΛΟΥΔΙΩΝ.	48
ΕΙΚΟΝΑ 36 Ο CONTAINER ΤΩΝ ΛΟΥΛΟΥΔΙΩΝ.	49
ΕΙΚΟΝΑ 37 ΤΟ SCRIPT ΠΟΥ ΜΕΤΑΦΕΡΕΙ ΤΑ ΛΟΥΛΟΥΔΙΑ.	49
ΕΙΚΟΝΑ 38 ΤΑ ΔΙΑΘΕΣΙΜΑ ΌΠΛΑ.	50
ΕΙΚΟΝΑ 39 ΤΟ ΣΥΝΟΛΟ ΤΩΝ ΌΠΛΩΝ ΜΕΣΑ ΣΤΟ ΑΝΤΙΚΕΙΜΕΝΟ ΣΥΛΛΟΓΗ.	50
ΕΙΚΟΝΑ 40 Όλα τα Όπλα Έχουν τοποθετηθεί στη κατάλληλη θέση.	51
ΕΙΚΟΝΑ 41 Λήψη τιμών από το χρήστη.	52
ΕΙΚΟΝΑ 42 ΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ ΚΑΤΆΛΛΗΛΟΥ ΌΠΛΟΥ ΚΑΙ ANIMATION.	52
ΕΙΚΟΝΑ 43 Ο ΝΕΟΣ ANIMATOR.	53
ΕΙΚΟΝΑ 44 ΔΗΜΙΟΥΡΓΙΑ ΑΝΤΙΚΕΙΜΕΝΟΥ CANVAS.	54
ΕΙΚΟΝΑ 45 Ο ΚΑΜΒΑΣ ΜΕ ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ TEXT ΣΤΗ ΣΚΗΝΗ.	54
ΕΙΚΟΝΑ 46 ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ CANVAS ΜΕ ΤΟ ΚΕΙΜΕΝΟ ΚΑΤΆ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΑΙΧΝΙΔΙΟΥ.	55

ΕΙΚΟΝΑ 47 ΚΑΘΟΡΙΣΜΟΣ ΤΟΥ ΤΡΟΠΟΥ ΜΕ ΤΟΝ ΟΠΟΙΟ ΘΑ ΕΜΦΑΝΙΖΕΤΑΙ Ο CANVAS.	55
ΕΙΚΟΝΑ 48 ΣΤΟ ΑΝΤΙΚΕΙΜΕΝΟ GAMEMANAGER, ΣΤΗΝ ΙΔΙΟΤΗΤΑ OBJCANVAS ΘΑ ΠΡΕΠΕΙ ΝΑ ΑΝΤΙΣΤΟΙΧΗΘΕΙ ΤΟ ΑΝΤΙΚΕΙΜΕΝΟ.	56
ΕΙΚΟΝΑ 49 SCRIPT ΓΙΑ ΤΗΝ ΕΜΦΑΝΙΣΗ ΤΟΥ ΚΑΜΒΑ.	57
ΕΙΚΟΝΑ 50 ΕΝΑΛΛΑΓΗ ΤΟΥ ΚΕΙΜΕΝΟΥ ΤΟΥ ΚΑΜΒΑ.	57
ΕΙΚΟΝΑ 51 ΠΡΟΒΟΛΗ ΜΗΝΥΜΑΤΩΝ ΑΠΟ ΤΟΝ ΚΑΜΒΑ.	57
ΕΙΚΟΝΑ 52 ΟΤΑΝ Η ΚΑΝΑΤΑ ΠΕΡΙΕΧΕΙ ΛΟΥΛΟΥΔΙ ΤΑ ΕΠΟΜΕΝΑ ΤΟΠΟΘΕΤΟΥΝΤΑΙ ΕΠΙ ΤΟΥ ΤΡΑΠΕΖΙΟΥ.	59
ΕΙΚΟΝΑ 53 ΑΝΤΙΚΕΙΜΕΝΑ ΤΡΑΠΕΖΙΑ ΜΕ ΤΙΣ ΚΑΝΑΤΕΣ.	59
ΕΙΚΟΝΑ 54 ΚΩΔΙΚΑΣ ΑΡΙΣΤΕΡΟΥ ΚΛΙΚ.	60
ΕΙΚΟΝΑ 55 ΚΩΔΙΚΑΣ ΓΙΑ ΤΟ ΔΕΞΙ ΚΛΙΚ.	61
ΕΙΚΟΝΑ 56 ΑΝΤΙΚΕΙΜΕΝΑ TEXT MESH ΓΙΑ ΤΗΝ ΕΜΦΑΝΙΣΗ ΤΟΥ ΑΡΙΘΜΟΥ ΤΗΣ ΚΑΘΕ ΚΑΝΑΤΑΣ.	62
ΕΙΚΟΝΑ 57 ΠΕΡΙΣΤΡΟΦΗ ΑΝΤΙΚΕΙΜΕΝΟΥ TEXT MESH PRO.	62
ΕΙΚΟΝΑ 58 Η ΣΥΝΑΡΤΗΣΗ SETOBJECTIVETEXT.	63
ΕΙΚΟΝΑ 59 ΈΛΕΓΧΟΣ ΠΕΡΑΙΩΣΗΣ Ή ΟΧΙ ΤΗΣ ΤΡΕΧΟΥΣΑΣ ΑΠΟΣΤΟΛΗΣ.	64
ΕΙΚΟΝΑ 60 ΠΡΟΣΒΑΣΗ ΣΤΗ ΜΕΤΑΒΛΗΤΗ ΤΟΥ SCRIPT CETTLCONTROLLER.	64
ΕΙΚΟΝΑ 61 ΥΠΟΡΟΥΤΗΝΑ ΕΜΦΑΝΙΣΗΣ ΤΩΝ ΜΗΝΥΜΑΤΩΝ.	65
ΕΙΚΟΝΑ 62 ΜΗΝΥΜΑ ΕΝΗΜΕΡΩΣΗΣ ΤΟΥ ΧΡΗΣΤΗ ΓΙΑ ΤΗΝ ΟΛΟΚΛΗΡΩΣΗ ΤΗΣ ΑΠΟΣΤΟΛΗΣ.	66
ΕΙΚΟΝΑ 63 2ο ΤΜΗΜΑ ΤΟΥ ΜΗΝΥΜΑΤΟΣ.	66
ΕΙΚΟΝΑ 64 ΤΟ PREFAB ΤΟΥ ΝΕΡΟΥ.	67
ΕΙΚΟΝΑ 65 Η ΠΙΣΤΑ ΤΟΥ ΕΠΙΠΕΔΟΥ 1.	67
ΕΙΚΟΝΑ 66 ΜΟΝΤΕΛΟ ΤΟΥ ΕΧΘΡΟΥ. ΠΗΓΗ: ASSETSTORE	68
ΕΙΚΟΝΑ 67 PACKAGE MANAGER ΤΗΣ UNITY.	69
ΕΙΚΟΝΑ 68 ΕΙΣΑΓΩΓΗ ΜΟΝΤΕΛΟΥ ΣΤΗ ΣΚΗΝΗ.	69
ΕΙΚΟΝΑ 69 Ο ΕΧΘΡΟΣ ΣΤΗ ΣΚΗΝΗ.	70
ΕΙΚΟΝΑ 70 ΤΑ ΑΝΤΙΚΕΙΜΕΝΑ ΑΣΠΙΔΑ ΚΑΙ ΟΠΛΟ ΤΟΥ ΜΟΝΤΕΛΟΥ.	71
ΕΙΚΟΝΑ 71 SCRIPT ΑΠΕΝΕΡΓΟΠΟΙΗΣΗΣ ΟΠΛΟΥ ΚΑΙ ΑΣΠΙΔΑΣ ΜΟΝΤΕΛΟΥ.	71
ΕΙΚΟΝΑ 72 ΤΟ ΜΟΝΤΕΛΟ ΜΕ ΤΟ SCRIPT ΚΑΙ ΤΙΣ ΑΝΑΦΟΡΕΣ ΣΤΑ ΑΝΤΙΚΕΙΜΕΝΑ.	72
ΕΙΚΟΝΑ 73 Ο ΕΧΘΡΟΣ ΧΩΡΙΣ ΟΠΛΟ ΚΑΙ ΑΣΠΙΔΑ.	72
ΕΙΚΟΝΑ 74 COLLIDER ΑΠΟΤΡΟΠΗΣ ΤΟΥ ΉΡΩΑ ΑΠΟ ΤΟ ΝΑ ΔΙΕΡΧΕΤΑΙ ΑΠΟ ΤΟ ΕΣΩΤΕΡΙΚΟ ΤΟΥ ΜΟΝΤΕΛΟΥ.	73
ΕΙΚΟΝΑ 75 ΑΚΤΙΝΑ ΕΝΤΟΣ ΤΗΣ ΟΠΟΙΑΣ ΤΟ ΜΟΝΤΕΛΟ ΕΝΤΟΠΙΖΕΙ ΕΧΘΡΟΥΣ.	74
ΕΙΚΟΝΑ 76 ΠΑΡΑΜΕΤΡΟΠΟΙΗΣΗ ΤΟΥ NAVIGATION.	75
ΕΙΚΟΝΑ 77 NAV MESH AGENT.	76
ΕΙΚΟΝΑ 78 ΟΡΙΣΜΟΣ ΚΑΤΕΨΘΥΝΣΗΣ ΣΤΟΝ NAV MESH AGENT.	76
ΕΙΚΟΝΑ 79 ΚΑΘΟΡΙΣΜΟΣ AVATAR ΓΙΑ ΤΗ ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ANIMATION.	77
ΕΙΚΟΝΑ 80 ΔΗΜΙΟΥΡΓΙΑ ANIMATOR CONTROLLER ΣΤΟ GOBLIN.	78
ΕΙΚΟΝΑ 81 ANIMATION CONTROLLER ΤΟΥ GOBLIN.	79
ΕΙΚΟΝΑ 82 ΕΠΕΞΕΡΓΑΣΙΑ ΤΟΥ ANIMATION.	79
ΕΙΚΟΝΑ 83 LOOP ΣΤΟ ANIMATION.	80
ΕΙΚΟΝΑ 84 ΠΡΟΣΒΑΣΗ ΣΤΙΣ ΜΕΤΑΒΛΗΤΕΣ ΤΟΥ ANIMATOR CONTROLLER.	81
ΕΙΚΟΝΑ 85 ΚΩΔΙΚΑΣ ΤΟΥ GAMEMANAGER SCRIPT.	82
ΕΙΚΟΝΑ 86 ΣΤΗ ΓΡΑΜΜΗ 40 Ο ΚΩΔΙΚΑΣ ΓΙΑ ΤΗ ΠΡΟΣΒΑΣΗ ΣΤΗ ΖΩΗ ΤΟΥ ΉΡΩΑ ΣΤΟ ΑΝΤΙΣΤΟΙΧΟ ΑΝΤΙΚΕΙΜΕΝΟ.	83
ΕΙΚΟΝΑ 87 ΕΠΕΞΕΡΓΑΣΙΑ ΤΗΣ ΜΕΤΑΒΛΗΤΗΣ ΔΙΑΦΟΡΕΤΙΚΟΥ SCRIPT ΣΤΗ ΓΡΑΜΜΗ 121.	83
ΕΙΚΟΝΑ 88 ΔΗΜΙΟΥΡΓΙΑ PREFABS.	84
ΕΙΚΟΝΑ 89 ΚΩΔΙΚΑΣ ΤΟΥ SINGLETON.	85
ΕΙΚΟΝΑ 90 ΜΟΛΙΣ ΟΛΟΚΛΗΡΩΘΕΙ Η 2η ΑΠΟΣΤΟΛΗ, ΣΤΗ ΓΡΑΜΜΗ 117 ΕΝΗΜΕΡΩΝΕΤΑΙ Η ΜΕΤΑΒΛΗΤΗ GUNS ΤΟΥ SINGLETON ΓΙΑ ΤΗ ΝΕΑ ΤΙΜΗ.	86
ΕΙΚΟΝΑ 91 ANIMATOR ΤΟΥ ΉΡΩΑ.	87
ΕΙΚΟΝΑ 92 ΈΛΕΓΧΟΣ ΤΟΥ ΑΝ Ο ΉΡΩΑΣ ΜΠΟΡΕΙ ΝΑ ΕΠΙΤΕΘΕΙ.	88
ΕΙΚΟΝΑ 93 ΤΜΗΜΑ ΚΩΔΙΚΑ ΤΗΣ ΕΠΙΘΕΣΗΣ ΤΟΥ ΉΡΩΑ ΣΤΟ GOBLIN.	89
ΕΙΚΟΝΑ 94 ΚΑΤΑΧΩΡΗΣΗ ΤΟΥ ΕΝΕΡΓΟΥ ΟΠΛΟΥ ΣΤΟ SINGLETON ΑΠΟ ΤΟ WEAPONCHANGE SCRIPT.	89
ΕΙΚΟΝΑ 95 ΚΑΘΟΡΙΣΜΟΣ ΤΟΥ ΕΠΙΠΕΔΟΥ ΖΗΜΙΑΣ ΠΟΥ ΠΡΟΚΑΛΟΥΝ ΤΑ ΧΤΥΠΗΜΑΤΑ ΣΤΟΝ ΕΧΘΡΟ.	90
ΕΙΚΟΝΑ 96 Ο ΚΑΜΒΑΣ ΠΟΥ ΕΜΦΑΝΙΖΕΙ ΤΑ ΥΠΟΛΟΙΠΑ ΤΗΣ ΖΩΗΣ ΤΩΝ ΜΑΧΟΜΕΝΩΝ.	90
ΕΙΚΟΝΑ 97 ΈΛΕΓΧΟΣ ΤΗΣ ΠΟΡΕΙΑΣ ΤΗΣ ΜΑΧΗΣ ΑΠΟ ΤΟ GAMEMANAGER.	91
ΕΙΚΟΝΑ 98 ΑΝΤΙΜΕΤΩΠΙΣΗ ΤΟΥ ΦΑΙΝΟΜΕΝΟΥ ΤΩΝ ΣΚΟΤΕΙΝΩΝ ΣΚΗΝΩΝ.	92
ΕΙΚΟΝΑ 99 ΠΙΣΤΑ 2ΟΥ ΕΠΙΠΕΔΟΥ.	93

ΕΙΚΟΝΑ 100 ΝΈΑ ΣΥΝΑΡΤΗΣΗ ΠΟΥ ΥΛΟΠΟΙΕΙ ΤΟ ΧΤΥΠΗΜΑ ΤΟΥ ΉΡΩΑ ΣΤΟ GOBLIN ΩΣΤΕ ΝΑ ΜΕΙΩΝΕΤΑΙ Η ΜΕΤΑΒΛΗΤΗ ΤΟΥ ΊΔΙΟΥ SCRIPT.	94
ΕΙΚΟΝΑ 101 ΤΟ PREFAB GAMEMANAGER ΜΕ ΤΟ ΝΕΟ SCRIPT.	95
ΕΙΚΟΝΑ 102 ΔΙΑΧΕΙΡΙΣΗ ΤΗΣ ΖΩΗΣ ΤΩΝ GOBLIN ΑΠΟ ΤΟ SCRIPT ΤΟΥ ΚΑΘΕ GOBLIN.	96
ΕΙΚΟΝΑ 103 ΤΜΗΜΑ ΚΩΔΙΚΑ ΤΗΣ UPDATE ΤΟΥ GAMEMANAGER.	96
ΕΙΚΟΝΑ 104 ΤΑ GOBLINS ΜΕΣΑ ΣΤΟ ΚΕΝΟ ΑΝΤΙΚΕΙΜΕΝΟ.	97
ΕΙΚΟΝΑ 105 ΠΙΣΤΑ ΤΟΥ 3ΟΥ ΕΠΙΠΕΔΟΥ.	97
ΕΙΚΟΝΑ 106 ΤΟ ΚΕΝΟ ΑΝΤΙΚΕΙΜΕΝΟ ΣΤΟ ΚΑΤΩ ΜΕΡΟΣ ΤΗΣ ΛΙΜΝΗΣ.	98
ΕΙΚΟΝΑ 107 Η ΑΡΧΙΚΗ ΠΙΣΤΑ ΠΟΥ ΕΜΦΑΝΙΖΕΙ ΤΟ MENU.	99
ΕΙΚΟΝΑ 108 ΕΜΦΑΝΙΣΗ ΤΟΥ ΚΕΡΣΟΡΑ ΣΤΗ ΠΙΣΤΑ ΤΟΥ GUI.	99
ΕΙΚΟΝΑ 109 ΥΛΟΠΟΙΗΣΗ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑΣ ΚΟΥΜΠΙΩΝ.	100
ΕΙΚΟΝΑ 110 ΣΥΝΑΡΤΗΣΕΙΣ ΤΟΥ LOADSCENE SCRIPT.	100
ΕΙΚΟΝΑ 111 ΧΤΙΣΙΜΟ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.	101

ΠΙΝΑΚΑΣ ΓΡΑΦΗΜΑΤΩΝ

ΓΡΑΦΗΜΑ 1 ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΤΑ ΑΠΟΛΥΤΗ ΤΙΜΗ	105
ΓΡΑΦΗΜΑ 2 ΑΠΟΤΕΛΕΣΜΑΤΑ ΣΕ ΣΧΕΣΗ ΜΕ ΠΡΟΪΟΝΤΑ ΤΗΣ ΑΥΤΗΣ ΑΓΟΡΑΣ	105
ΓΡΑΦΗΜΑ 3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ	106
ΓΡΑΦΗΜΑ 4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΩΤΗΜΑΤΟΛΟΓΙΟΥ ΓΝΩΣΕΩΝ	108
ΓΡΑΦΗΜΑ 5 ΚΑΜΠΥΛΗ ΚΑΤΑΚΡΑΤΗΣΗΣ ΓΝΩΣΤΙΚΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ ΓΙΑ ΤΟΝ ΜΕΣΟ ΜΑΘΗΤΗ , ΠΗΓΗ: HTTPS://WWW.RESEARCHGATE.NET/FIGURE/EBBINGHAUS-FORGETTING-CURVE-AND-REVIEW-CYCLE_FIG1_324816198	109

Κεφάλαιο 1 – Εισαγωγή

Η παρούσα εργασία εκπονήθηκε στα πλαίσια του προπτυχιακού προγράμματος σπουδών της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πολυτεχνείου Κρήτης.

1.1 Κίνητρα - Συμβολή

Βασική έμπνευση για την ενασχόληση με το θέμα της παρούσας εργασίας αποτέλεσε η παρατήρηση της νέας τάσης να ενσωματώνονται σε διάφορες πτυχές του κοινωνικού, επιχειρηματικού και εκπαιδευτικού κόσμου, τεχνικές και διαδικασίες που κλασσικά χρησιμοποιούνταν στα παιχνίδια (κυρίως βιντεοπαιχνίδια, αλλά και επιτραπέζια ή ομαδικά) ώστε να ενισχυθούν επιθυμητές συμπεριφορές από μία ομάδα ή να διορθωθούν μή επιθυμητά φαινόμενα. Αυτά τα μεταφέριμα σε άλλους τομείς στοιχεία των παιχνιδιών ονομάζονται συμπεριληπτικά με τον όρο «παιγνιοποίηση» (gamification). Όταν, τώρα, αντί για κάποια μεμονωμένα στοιχεία, κρατήσουμε εξ ολοκλήρου ένα παιχνίδι προσθέτοντας όμως επιπλέον στοιχεία τα οποία δύνανται να «μοχλεύσουν την δύναμη των ηλεκτρονικών παιχνιδιών για να γοητεύσει και να εμπλέξει τους τελικούς χρήστες για έναν συγκεκριμένο σκοπό, όπως η ανάπτυξη νέων γνώσεων και δεξιοτήτων» [1], τότε μιλάμε για «σοβαρά παίγνια» (serious games). Πρόσφατες μελέτες έχουν αποδείξει ότι τα σοβαρά παιχνίδια μπορούν να υποστηρίξουν την μάθηση των μαθηματικών (βλ. [2][3]) και δύνανται να βελτιώσουν αρκετά τις επιδόσεις των μαθητών στα μαθηματικά (βλ. [4]) υπό την προϋπόθεση ότι είναι σωστά σχεδιασμένα (βλ. [5]).

Παράλληλα με την ολοένα και αυξανόμενη χρήση των ως άνω σοβαρών παιγνίων είναι καταφανής και η διείσδυση της τεχνολογίας σε ποικίλες εκφάνσεις του καθημερινού βίου του σύγχρονου ανθρώπου. Το οξύμωρον του πραγματος είναι όμως, ότι παρά την εξοικείωση με την τεχνολογία, απουσιάζει η επαρκής γνώση των θεμελιωδών αρχών πάνω στις οποίες στηρίζεται το οικοδόμημα της τεχνολογίας και δή των ηλεκτρονικών υπολογιστών. Το γεγονός ότι οι τεχνολογικές εξελίξεις δεν συμβαδίζουν με την κατ' ουσίαν γνώση αυτών από τον γενικό πληθυσμό οφείλεται κατά ένα μέρος στην βραδύτητα των εκπαιδευτικών συστημάτων να ανανεώνουν τα προγράμματα σπουδών με βάση αυτές τις εξελίξεις είδη από τις πρώτες εκπαιδευτικές βαθμίδες.

Στα πλαίσια αυτά, η συνεισφορά της παρούσας εργασίας είναι αφενός η περαιτέρω καλλιέργεια της αναλυτικής σκέψης σε παιδιά δημοτικού -ικανότητα απαραίτητη στον

σύγχρονο τεχνολογικό κόσμο- και αφετέρου η εκμάθηση της πιο πρωταρχικής ιδέας στην οποία βασίζεται η λειτουργία των ηλεκτρονικών υπολογιστών, αυτής του δυαδικού συστήματος αρίθμησης.

Η εξοικίωση με το δυαδικό σύστημα προσδοκούμε να ανοίξει νέους ορίζοντες στους νεαρούς μαθητές οι οποίοι πλέον θα έχουν την ευκαιρία να καταδυθούν στο βάθος των πραγμάτων του τεχνολογικού γίνεσθαι, χρησιμοποιώντας αυτές τις γνώσεις για τον σχεδιασμό ηλεκτρονικών συστημάτων και δίνοντας απαντήσεις σε απορίες σχετικά με τις αρχές λειτουργίας των υπολογιστών, ώστε να αναδυθούν πλέον στις επόμενες εκπαιδευτικές βαθμίδες ως κριτικοί στοχαστές των τεκταινόμενων στον τεχνολογικό κόσμο ή ακόμη και να γίνουν ενεργά μέλη της επιστημονικής κοινότητας που οδηγεί αυτές τις εξελίξεις. Σε αυτό παίζει σημαντικό ρόλο η πλήρης ουδετερότητα στις όποιες ανομοιογένειες υπάρχουν στις σχολικές τάξεις, είτε αυτές αφορούν το φύλο του μαθητή ή το αναπτυξιακό του επίπεδο. Σε αυτό το πλαίσιο επιδιώχθηκε, η πλοκή του παιχνιδιού να είναι εξίσου ελκυστική για αγόρια και για κορίτσια, καθώς και να υπάρχει ευελιξία σχετικά με το βάθος εκμάθησης της διδακταίας ύλης. Έτσι επιτυγχάνεται ένα περιβάλλον ίσων ευκαιριών και ενισχύεται η ισότητα των φύλων στις θετικές επιστήμες.

1.2. Περίληψη

Το παιχνίδι που αναπτύχθηκε κατά την παρουσία εργασία λαμβάνει χώρα σε ένα μαγικό δάσος που κατοικείται από διάφορα μυθικά πλάσματα, τα οποία όμως δέχονται επίθεση από κάποια μοχθηρά ξωτικά. Η διάσωση αυτού του δάσους από τα ξωτικά βασίζεται στον μοναδικό άνθρωπο-κάτοικο του δάσους, που είναι ο παίκτης του παιχνιδιού. Ο παίκτης περιφέρεται μες στο δάσος ψάχνοντας τα ξωτικά για να τα αντιμετωπίσει. Για το σκοπό αυτό μπορεί να κάνει χρήση τριών διαφορετικών όπλων. Τα όπλα αυτά όμως τα αποκτάει σταδιακά, ολοκληρώνοντας με επιτυχία τις τρεις πίστεις του παιχνιδιού, ενώ αρχικά μονομαχεί με τις γροθιές του καταφέροντας ωστόσο λιγότερη ζημιά στον αντίπαλο σε σχέση με την περιπτωση που θα χρησιμοποιούσε κάποιο όπλο. Κατα τις μονομαχίες αυτές, ο παίκτης είναι φυσικό να χάνει μέρος από την δύναμή του με αποτέλεσμα να χρειάζεται να πει κάποιο μαγικό φίλτρο που επαναφέρει την δύναμή του στα αρχικά επίπεδα.

Για να παρασκευάσει αυτό το φίλτρο πρέπει να επιστρέψει στην καλύβα του (πατώντας ένα κουμπί για να εμφανιστεί το παράθυρο διαλόγου με την αντίστοιχη επιλογή), να πάει στον κήπο του όπου καλλιεργεί τα ειδικά μαγικά βότανα και αφού μαζέψει κάμποσα, να τα μοιράσει σε κάποιο από τα υπάρχοντα καζάνια. Ο τρόπος με τον οποίο τα βότανα μοιράζονται στα καζάνια ακολουθεί τους κανόνες του δυαδικού συστήματος αρίθμησης. Το κάθε καζάνι έχει διαφορετική χωρητικότητα (η οποία αποτελεί δύναμη του δύο) και εάν κατα την εκτέλεση της συνταγής τοποθετηθούν περισσότερα βότανα

στο καζάνι, τότε αυτά μαραίνονται και πέφτουν κάτω, οπότε ο παίκτης πρέπει να τα ξαναμαζέψει και να τα τοποθετήσει στο κατάλληλο καζάνι. Έτσι μαθαίνει την έννοια της αξίας της θέσης στις αναπαραστάσεις αριθμών. Για να ξέρει βέβαια, πόσα βότανα να τοποθετήσει σε κάθε καζάνι, στην οθόνη εμφανίζεται ένα παράθυρο διαλόγου που δείχνει την συνταγή για το φίλτρο. Δηλαδή, δείχνει πόσα βότανα χρειάζεται να χρησιμοποιήσει για την συνταγή και ακολούθως ο παίκτης πρέπει να σκεφτεί σε ποια καζάνια πρέπει να τα ρίξει με βάση την χωρητικότητά τους ώστε να χρησιμοποιηθεί ο συγκεκριμένος αριθμός βοτανών στο φίλτρο.

Όταν αρχίζει το παιχνίδι, ο παίκτης καλείται να ακολουθήσει τρεις συνταγές για την παρασκευή μαγικών φίλτρων στην αυλή του, εν είδη τριών αποστολών, μετά το πέρας των οποίων κερδίζει από ένα όπλο κάθε φορά και μόνο τότε είναι έτοιμος για να βγει στο δάσος, να αντιμετωπίσει τα εχθρικά πλάσματα και να προστατεύσει τους κατοίκους του μυθικού δάσους.

Η αντιμετώπιση των εχθρών είναι κλιμακούμενης δυσκολίας, με το παιχνίδι να έχει τρεις πίστες όπου υπάρχουν όλο και περισσότεροι εχθροί, με τον παίκτη ωστόσο να έχει όλο και περισσότερη δύναμη, οπότε να επιτυγχάνει όλο και μεγαλύτερη ζημιά στους αντιπάλους του.

Για την δημιουργία του παιχνιδιού χρησιμοποιήθηκε η πλατφόρμα ανάπτυξης παιχνιδιών της Unity. Η συγκεκριμένη πλατφόρμα μας παρείχε πληθώρα εργαλείων για την αποδοτική και ταχεία ανάπτυξη του βιντεοπαιχνιδιού. Τέτοια εργαλεία είναι η μηχανή φυσικής εξομίωσης (physics engine), η μηχανή γραφικής απεικόνισης και χειρισμού τρισδιάστατων και δισδιάστατων γραφικών (rendering engine και 3D/2D graphics engine), το ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) καθώς και πακέτα λογισμικού με αλγορίθμους τεχνητής νοῦμοσύνης και εργαλεία χειρισμού των animations. Επίσης, για τα τρισδιάστατα μοντέλα του παιχνιδιού χρησιμοποιήθηκαν έτοιμα πακέτα με χαρακτήρες και άλλα αντικείμενα (έπιπλα, δένδρα, βράχοι, γρασίδι) που διατίθενται στο επίσημο e-shop της Unity.

Με την περάτωση της δημιουργίας του βιντεοπαιχνιδιού, κάναμε μια πιλοτική εφαρμογή του παιχνιδιού για την διδασκαλία του δυαδικού συστήματος σε σχολικές αίθουσες των τελευταίων τάξεων του δημοτικού, σε ελληνικά σχολεία της Γερμανίας και της Αμερικής. Η δοκιμές έγιναν χωρίζοντας τους συμμετέχοντες σε δύο ομάδες, ελέγχου και πειραματική, σύμφωνα με τις αρχές της στατιστικής επιστήμης. Για την αξιολόγηση χρησιμοποιήθηκαν δύο ερωτηματολόγια με ξεχωριστούς στόχους. Το ένα αποσκοπούσε στην βαθμολόγηση της εμπειρίας χρήστη με βάση διάφορες παραμέτρους (πραγματιστικές και ηδονιστικές), ενώ το άλλο αφορούσε στην ανάλυση της απόδοσης της προτεινόμενης παιγνιώδους διδακτικής προσέγγισης στην κατ' ουσίαν απόκτηση θεωρητικών γνώσεων. Τα αποτελέσματα αξιολογήθηκαν, και με βάση αυτά προτείνονται κάποιες διορθώσεις και επεκτάσεις στο αντίστοιχο κεφάλαιο του παρόντος συγγράμματος.

1.3 Διάρθρωση κεφαλαίων

Το παρόν πόνημα χωρίζεται σε επτά κεφάλαια.

Το πρώτο κεφάλαιο αποτελεί μία εισαγωγή στο θέμα της εργασίας, όπου αναλύονται τα κίνητρα ενασχόλησης με αυτό, περιγράφονται οι προσδοκίες μας για την συμβολή της εργασίας στο γενικό σύνολο και έπειτα υπάρχει μία αναλυτική περιγραφή των θεωρητικών αρχών τι οποίες πραγματεύεται η εργασία, η οποία βαίνει σε πλείστα επιστημονικά πεδία. Αυτά είναι η παιδαγωγική επιστήμη, η επιστήμη των μαθηματικών, αλλά και αυτή της λογικής σχεδίασης και αρχιτεκτονικής υπολογιστών.

Στο δεύτερο κεφάλαιο παρουσιάζεται μία συγκριτική μελέτη των πιο κοινών μηχανών ανάπτυξης βιντεοπαιχνιδιών (αφού αυτές είναι το βασικό εργαλείο που χρησιμοποιήθηκε στην παρούσα εργασία). Ειδική μνεία, γίνεται στην συγκεκριμένη μηχανή που χρησιμοποιήθηκε, αυτή της Unity και αφού δοθούν συνοπτικά τα προτερήματά της έναντι των άλλων μηχανών της αγοράς, αναλύονται από τεχνικής πλευράς τα βασικά εργαλεία της Unity που βοήθησαν στην ανάπτυξη του βιντεοπαιχνιδιού.

Στη συνέχεια, ακολουθεί το τέταρτο κεφάλαιο, η περιγραφή του παιχνιδιού, το οποίο εξηγεί όλα όσα ο παίκτης θα ήθελε να δει και να μάθει για το παιχνίδι. Ξεκινώντας από το ότι το παιχνίδι λαμβάνει χώρα σε έναν δάσος με μυθικά πλάσματα το οποίο καλείται ο ίδιος να προστατεύσει, καθώς και πώς εμπλέκεται η εκμάθηση του δυαδικού συστήματος στην πορεία του παιχνιδιού. Επίσης, δίνει μία σαφή εικόνα των διαφόρων και της δοκιμαστικής πίστας εξοικείωσης με το περιβάλλον του παιχνιδιού.

Το πέμπτο κεφάλαιο είναι μία αναλυτική παρουσίαση της υλοποίησης από καθαρά τεχνική σκοπιά. Αυτό το κεφάλαιο περιέχει πληθώρα υπο-κεφαλαίων για την καλύτερη ομαδοποίηση των στοιχείων από τα οποία αποτελείται το βιντεοπαιχνίδι. Έτσι υπάρχει κεφάλαιο που εξηγείται οτιδήποτε αφορά το μοντέλο του χρήστη, άλλο κεφάλαιο αποκλειστικά για το φυσικό περιβάλλον των επιπέδων και ομοίως για τα μαγικά φίλτρα, το οπλοστάσιο του παίκτη, την διεπαφή του χρήστη (GUI), των αποστολών και τέλος ξεχωριστά υπο-κεφάλαια για την κάθεμία από τις τρεις πίστες αναλυτικά.

Ακολουθεί το έκτο κεφάλαιο με την επίδειξη των μεθόδων που χρησιμοποιήθηκαν για την αξιολόγηση της αποδοτικότητας του παιχνιδιού, όπως αυτή θεωρείται από δύο οπτικές γωνίες: της εμπειρίας χρήστη και της αποτελεσματικής διδακτικής

προσέγγισης των πραγματευομένων θεωρητικών εννοιών (δυναμικό σύστημα αρίθμησης). Για την κάθεμία οπτική υπάρχει ξεχωριστό υπο-κεφάλαιο.

Σαν κατακλείδα του κειμένου της παρούσας εργασίας βρίσκεται το έβδομο κεφάλαιο με τα συμπεράσματα από την ενασχόληση με το θέμα της εργασίας και πώς αυτή αποτελεί ένα χρήσιμο εργαλείο στα χέρια των εκπαιδευτικών της πρωτοβάθμιας και δευτεροβάθμιας εκπαίδευσης, αλλά και ένα ολοκληρωμένο σοβαρό παίγνιο διαθέσιμο και για χρήση από τους μαθητές στο σπίτι με ποικίλα ωφέλη για την καλλιέργεια της αναλυτικής τους σκέψης.

Η βιβλιογραφία και τα δύο παραρτήματα με τα ερωτηματολόγια αξιολόγησης του παιχνιδιού, ολοκληρώνουν τελικά το θεωρητικό μέρος της παρούσας εργασίας.

Κεφάλαιο 2 - Θεωρητικό υπόβαθρο

2.1 Παιδαγωγική προσέγγιση

2.1.1 Εισαγωγή

Η εξάπλωση των ψηφιακών τεχνολογιών τα τελευταία χρόνια είχε τεράστιο αντίκτυπο στην εκπαίδευση. Σε παγκόσμιο επίπεδο αναγνωρίζεται η ανάγκη εκπαίδευσης της νέας γενιάς στις ψηφιακές τεχνολογίες και, ειδικότερα, στις βασικές αρχές της πληροφορικής και του προγραμματισμού [6],[7], προκειμένου οι νέοι να αποκτήσουν τις βασικές δεξιότητες και ικανότητες που θα τους επιτρέψουν να αξιοποιήσουν τις ψηφιακές τεχνολογίες για να γίνουν δημιουργικοί και καινοτόμοι πολίτες. Η τάση αυτή μπορεί να αναχθεί στο πρωτοποριακό έργο του Seymour Papert, δημιουργού της LOGO, της πρώτης γλώσσας προγραμματισμού για παιδιά. Ο Papert έχει τονίσει τη σχέση μεταξύ του προγραμματισμού στον υπολογιστή και της κατασκευής ψηφιακών τεχνουργημάτων που αποτελούν τη βάση της θεωρίας του για τη μάθηση, δηλαδή του κονστрукτιβισμού, δηλώνοντας ότι η μάθηση είναι βαθύτερη όταν οι μαθητές κατασκευάζουν πράγματα. Στο θεμελιώδες βιβλίο του "Mindstorms: Children, Computers and Powerful Ideas" παρουσίασε τις μεγάλες ιδέες πίσω από τους υπολογιστές και πώς αυτές οι μεγάλες ιδέες μπορούν να γίνουν καλύτερα κατανοητές από τα μικρά παιδιά [8]. Σε μεταγενέστερα έργα του, ο Papert εξέφρασε την πεποίθησή του ότι οι "μεγάλες (θεμελιώδεις) ιδέες" της επιστήμης των υπολογιστών πρέπει να γίνουν μέρος των σχολικών προγραμμάτων σπουδών [9]. Μία από αυτές τις ισχυρές ιδέες είναι αναμφίβολα το δυναμικό σύστημα αριθμών, το οποίο αποτελεί τη βάση για την ανάπτυξη του υλικού των υπολογιστών και όλων των τύπων αναπαραστάσεων δεδομένων που χρησιμοποιούνται στη σύγχρονη πληροφορική. Η παρούσα εργασία

ανταποκρίνεται στην ανάγκη να προσφερθεί ένα αποτελεσματικό μαθησιακό περιβάλλον για τους μικρούς μαθητές, ώστε να κατανοήσουν τις έννοιες πίσω από το δυαδικό σύστημα και τον τρόπο με τον οποίο χρησιμοποιείται για την υλοποίηση της αριθμητικής.

Σήμερα, με την έλευση της εκπαιδευτικής ρομποτικής, της διδασκαλίας STEM/STEAM και την εφαρμογή παγκόσμιων πρωτοβουλιών για την προώθηση του προγραμματισμού, όπως η "Ωρα του Κώδικα" και η "Εβδομάδα του Κώδικα της ΕΕ" (Kyfonidis, Mousmoutzis, & Christodoulakis, 2017) (Mousmoutzis, et al., 2017), ο κίνδυνος να δοθεί υπερβολική έμφαση στη χρήση των υπολογιστών χωρίς να δοθεί η απαραίτητη προσοχή στις θεμελιώδεις ιδέες που βρίσκονται πίσω από αυτούς, είναι ακόμη μεγαλύτερος.

Το ζήτημα γίνεται ακόμη πιο σημαντικό όταν πρόκειται για μικρότερα παιδιά στις πρώτες τάξεις του δημοτικού σχολείου ή ακόμη και του νηπιαγωγείου. Υπάρχει ήδη μια πληθώρα σχετικών εφαρμογών και «ρομποτικών κιτ» που είναι διαθέσιμα για τα παιδιά αυτών των ηλικιών. Κατά συνέπεια, υπάρχει ο πειρασμός για τους εκπαιδευτικούς να επικεντρωθούν στο τεχνικό μέρος της εκπαιδευτικής πληροφορικής αντί να προσπαθήσουν να δώσουν έμφαση στις τεχνολογικά ανεξάρτητες μεγάλες ιδέες της πληροφορικής, ώστε τα μικρά παιδιά να μπορούν να κατανοήσουν σε βάθος και να χρησιμοποιήσουν αποτελεσματικά την ψηφιακή τεχνολογία

Η συνολική προσέγγιση βασίζεται στο PerFECt, ένα σχεδιαστικό πλαίσιο (design framework) που δίνει έμφαση στην «επιτελεσματικότητα», δηλαδή στο ότι η γνώση βασίζεται στην ανθρώπινη επίδοση και στις ενέργειες που γίνονται μέσα σε συγκεκριμένα κοινωνικά πλαίσια, παρά στην ανάπτυξη εννοιολογικών αναπαραστάσεων. Το πλαίσιο αυτό έχει ήδη χρησιμοποιηθεί για να καθοδηγήσει τις εξελίξεις σε διάφορα έργα που αφορούν τη διατήρηση της ψηφιακής κληρονομιάς και τη δημιουργική μάθηση (Mousmoutzis, Christoulakis, Christodoulakis, & Paneva-Marinova, 2018) (Mousmoutzis, Sifakis, Christodoulakis, & Paneva-Marinova, 2019). Στην παρούσα εργασία, το πλαίσιο PerFECt ερμηνεύεται εκ νέου και επαναχρησιμοποιείται για να επικεντρωθεί στην ιδέα του απο-σχεδιασμού. Ο απο-σχεδιασμός αντιμετωπίζει την ανάγκη να δοθεί έμφαση στην παράλειψη χαρακτηριστικών από έναν σχεδιασμό προκειμένου να δοθεί μεγαλύτερη ελευθερία στους χρήστες ενός συγκεκριμένου συστήματος [10]. Στην περίπτωση του παιχνιδιού εικονικής πραγματικότητας, ο απο-σχεδιασμός χρησιμοποιείται για να προσφέρει μια εναλλακτική λύση στη χρήση ψηφιακών εργαλείων για το υπό εξέταση θέμα μάθησης: το ανθρώπινο σώμα

ενεργοποιείται και τίθεται σε λειτουργία για να επιτρέψει όλες τις απαραίτητες αλληλεπιδράσεις που εμπλέκουν τα συμμετέχοντα παιδιά στη διαδικασία μάθησης.

2.1.2 Αρχές σχεδιασμού που υιοθετήθηκαν από το πλαίσιο PerFECt

Η τεχνολογία γενικότερα και οι ψηφιακές τεχνολογίες ειδικότερα αποτελούν καταλύτη για την εγκαθίδρυση και διατήρηση ορισμένων κοινωνικών δομών, όπως υπογραμμίζουν και παραδειγματίζουν οι Cabitza et al. (2014)[11]. Τονίζουν το γεγονός ότι οι τελικοί χρήστες γίνονται "παραγωγοί" περιεχομένου και λειτουργιών. Από την άλλη πλευρά, ο όρος «ειδικός χρήστης» προτείνεται για να υποδηλώσει έναν εμπειρογνώμονα σε έναν συγκεκριμένο τομέα με κύριο στόχο την ανάπτυξη των τεχνολογικών δυνατοτήτων που είναι διαθέσιμες σε αυτόν τον τομέα.

Ένας ειδικός χρήστης ασχολείται με δημιουργικές/συγγραφικές δραστηριότητες χωρίς να είναι επαγγελματίας προγραμματιστής λογισμικού. Συνήθως ο ρόλος του τελικού χρήστη και ο ρόλος του ειδικού χρήστη ασκούνται από διαφορετικά άτομα που μπορεί επίσης να ανήκουν σε διαφορετικές κοινότητες. Επιπλέον, οι Cabitza & Simone [12] προτείνουν τον ρόλο του «μετα-σχεδιαστή» για να περιγράψουν το έργο που επιτελείται από επαγγελματίες οι οποίοι δημιουργούν τις κοινωνικο-τεχνικές συνθήκες για την ενδυνάμωση των τελικών χρηστών ώστε να λειτουργούν ως ενεργοί συντελεστές περιεχομένου και λειτουργιών.

Ένας μετα-σχεδιαστής δημιουργεί ανοικτά συστήματα που μπορούν να αναπτυχθούν περαιτέρω από τους χρήστες τους που ενεργούν ως συν-σχεδιαστές. Στην παρούσα εργασία τον ρόλο του ειδικού χρήστη αναλαμβάνει ο, ενώ αυτόν του μετά-σχεδιαστή συνήθως αναλαμβάνει ο εκπαιδευτικός όταν το παιχνίδι χρησιμοποιείται σε κάποια σχολική αίθουσα ή ο ίδιος ο χρήστης στις άλλες περιπτώσεις. Ωστόσο, εκτός από τις τεχνικές συνθήκες που είναι απαραίτητες για τη δημιουργία τέτοιων περιβαλλόντων, υπάρχει ανάγκη να δημιουργηθούν αποτελεσματικά οι κοινωνικές συνθήκες που θα επιτρέψουν στους έμπειρους χρήστες να κατασκευάσουν και να προσαρμόσουν τα τεχνουργήματα που θα χρησιμοποιηθούν από τους τελικούς χρήστες. Για να ανταποκριθεί σε αυτή την ανάγκη, καθορίζεται ένας ειδικός ρόλος χρήστη: οι μαιευτές-σχεδιαστές (maieuta-designers) με βάση την προσέγγιση διδασκαλίας του

Σωκράτη. Ένας μαιευτής-σχεδιαστής δημιουργεί τις απαραίτητες προϋποθέσεις για τη διευκόλυνση των έμπειρων χρηστών. Στην περίπτωση μας θα μπορούσε να είναι ο εκπαιδευτικός που επιβλέπει την αλληλεπίδραση των μαθητών με την πλατφόρμα εικονικής πραγματικότητας.

Το πλαίσιο PerFECT χρησιμοποιεί την έννοια της καθολικότητας για να αντιμετωπίσει το ζήτημα της αιτιότητας στις ψηφιακές αναπαραστάσεις, όπως το θέτει η Brenda Laurel στο θεμελιώδες βιβλίο της "Computers as Theatre":

"Το γεγονός ότι οι άνθρωποι επιδιώκουν να κατανοήσουν την αιτιότητα σε κόσμους αναπαραστάσεων παρέχει τη βάση για τον ορισμό της καθολικότητας του Αριστοτέλη. Κατά την καθομιλουμένη άποψη, μια ενέργεια είναι καθολική αν όλοι μπορούν να την κατανοήσουν, ανεξάρτητα από τις πολιτισμικές και άλλες διαφορές μεταξύ των ατόμων. Αυτό φαίνεται να περιορίζει το σύνολο των καθολικών ενεργειών σε πράγματα που κάνουν όλοι στον πλανήτη: τρώνε, κοιμούνται, αγαπούν κ.λπ. Ο Αριστοτέλης υποστηρίζει ότι οποιαδήποτε ενέργεια μπορεί να "καθολικευτεί" απλώς αποκαλύπτοντας την αιτία της- δηλαδή, η κατανόηση της αιτίας αρκεί για την κατανόηση της ενέργειας, ακόμη και αν πρόκειται για κάτι ξένο προς τον πολιτισμό, το ιστορικό ή την προσωπική "πραγματικότητα" του ατόμου". [13]

Η προσέγγιση λοιπόν της παρούσας εργασίας είναι η φυγή από το κλασσικό μοντέλο διδασκαλίας της αριθμητικής με χρήση συμβόλων και η προσφυγή σε ποιό θεμελιώδεις αναπαραστάσεις, όπως αυτές δίδονται από την παρατήρηση του πλήθους αντικειμένων. Έτσι, για παράδειγμα, αντί να εμφανίσουμε στον χρήστη του παιχνιδιού την πληροφορία ότι στην κανάτα βρίσκονται «5 λουλούδια», του δείχνουμε μία αναπαράσταση με πέντε άνθη.

Η γνώση δημιουργείται μέσω των ενεργειών των μελών μιας κοινωνικής δομής. Από αυτή την άποψη, παρατηρείται μια στροφή προς "την ενεργό κοινωνική κατασκευή της πραγματικότητας και όχι προς την αναπαράστασή της" [14]. Με αυτή την σκέψη λήφθηκε ιδιαίτερη μέριμνα κατά τον σχεδιασμό του παιχνιδιού, ώστε να μπορεί να υποστηριχθεί συμμετοχή ολόκληρης ομάδας για την περάτωση των αποστολών και -σε κατοπινή βελτίωση του παιχνιδιού- η εκμάθηση της διδακτικής ύλης που το παιχνίδι καλείται να καλύψει να αποτελεί γεγονός σύμπασας της σχολικής τάξης ή εκπαιδευτικής κοινότητας, γενικότερα.

Μια πολύ ενδιαφέρουσα πτυχή του τρόπου με τον οποίο το παρόν παιχνίδι εικονικής πραγματικότητας ερμηνεύει το πλαίσιο PerFECt και ρίχνει νέο φως στη δυνατότητα εφαρμογής του στο σχεδιασμό συνεργατικών μαθησιακών εμπειριών, σχετίζεται με την ιδέα του απο-σχεδιασμού [10]. Ο απο-σχεδιασμός παραπέμπει στην ιδέα ότι η παράλειψη και η απομάκρυνση χαρακτηριστικών από έναν σχεδιασμό είναι εξίσου κρίσιμη για την επιτυχία ενός συστήματος όσο και η θετική συμπερίληψή τους. Αυτό συνδέεται με το γεγονός ότι κάθε χαρακτηριστικό γνώρισμα παρέχει και περιορίζει τις αλληλεπιδράσεις με το τεχνούργημα (παίγνιο, εν προκειμένω) και μέσω αυτού, αυτό που αφήνεται έξω από αυτό έχει τη δυνατότητα να είναι ακόμη πιο σημαντικό από αυτό που οι σχεδιαστές έβαλαν επίτηδες μέσα σε αυτό. Πρόκειται για μια σκόπιμη αδράνεια που υπερβαίνει τις κύριες σχεδιαστικές προσεγγίσεις, προσφέροντας ευκαιρίες για διαφορετικές ερμηνείες των πληροφοριών που πρέπει να ληφθούν υπόψη στις πρακτικές των χρηστών και αναγνωρίζοντας τη δημιουργική δύναμη της ασάφειας. Η συνάφεια του απο-σχεδιασμού με τη μάθηση και τη δημιουργικότητα, είναι επομένως προφανής.

Ακολουθώντας μια προσέγγιση απο-σχεδιασμού, το υπό ανάλυση παιχνίδι εικονικής πραγματικότητας, παίρνει την ιδέα της καθολικότητας, όπως χρησιμοποιείται στο πλαίσιο PerFECt, μαζί με την υποκείμενη έννοια της αιτιότητας, και τη χρησιμοποιεί πέρα από τις ψηφιακές τεχνολογίες για να εξηγήσει μία σειρά από ιδιόμορφες («μαγικές») κανάτες που συμπεριφέρεται σύμφωνα με ορισμένους κανόνες. Με αυτόν τον τρόπο, το παιχνίδι χρησιμοποιεί την σειρά των κανατών και την τοποθέτηση βοτάνων σε αυτές για να ενεργοποιήσει τη λειτουργία κυκλωμάτων που υλοποιούν μια δυαδική αριθμομηχανή.

2.2 Συστήματα αρίθμησης

2.2.1 Εισαγωγή

Από την αρχαιότητα ο άνθρωπος, από τα πρώτα κιόλας επίπεδα πολιτισμού, αντηλήφθηκε την ανάγκη της απαρίθμησης αντικειμένων (με εξαίρεση κάποιες φυλές του Αμαζονίου όπως οι Πιράχα όσον αφορά στους σύγχρονους μας πολιτισμούς [15]). Στις αυγές τις ιστορίας, με την ανακάλυψη της γραφής διαφάνηκε και η χρησιμότητα της αναπαράστασης του πλήθους των απαριθμούμενων ποσοτήτων. Αυτή η αναπαράσταση συμβάδιζε με την νέα τότε «τεχνολογία» της γραφής και καθώς οι πρώτες κοινότητες των ανθρώπων οργανώνονταν σε πιο περίπλοκα πολιτισμικά μοντέλα -με δομές καταγραφής και ελέγχου οικονομικών συναλλαγών, με εμπορικές συναναστροφές με γείτονες ομάδες και γενικότερα, με έναν πιο συγκροτημένο τρόπο

οργάνωσης πολυειδούς πληροφορίας- η αναπαράσταση μιας ποσότητας ήταν μείζονος σημασίας για την εύρυθμη λειτουργία του πολιτιστικού οικοδομήματος.

2.2.2. Είδη συστημάτων αρίθμησης

Στο διάβα της ιστορίας και σε διαφορετικές πάντα γεωγραφικές περιοχές χρησιμοποιήθηκαν διαφορετικοί τρόποι αναπαράστασης των αριθμών. Ο φορμαλισμός της σύγχρονης μαθηματικής επιστήμης επιτάσσει την ονοματοδοσία και κατηγοριοποίηση αυτών των αναπαραστάσεων στα λεγόμενα, συστήματα αρίθμησης.

Σύστημα αρίθμησης καλείται μία μαθηματική σημειολογία (σύστημα γραφής) για την αναπαράσταση αριθμών ενός συγκεκριμένου συνόλου χρησιμοποιώντας σύμβολα (ψηφία, γράμματα, εικονίδια) με συστηματικό τρόπο [16]. Ο αριθμός τον οποίο συμβολίζει η εκάστοτε αναπαράσταση ονομάζεται τιμή.

Τα συστήματα αρίθμησης αναπαριστούν εν γένει διαφορετικά σύνολα αριθμών. Για παράδειγμα, το «ρωμαϊκό σύστημα», που αναλύεται στην συνέχεια, αναπαριστά μόνο θετικούς αριθμούς, ενώ το «σύστημα των φυλακισμένων» χρησιμοποιείται μόνο για μή αρνητικούς αριθμούς.

Μερικά παραδείγματα συστημάτων αρίθμησης αποτελούν τα εξής:

- Αξίας-συμβόλου
- Καταμετρητικά
- Θεσιακά

2.2.2.1 Σύστημα αξίας συμβόλου

Τα συστήματα αξίας συμβόλου αναπαριστούν αριθμούς χρησιμοποιώντας σύμβολα τα οποία έχουν συγκεκριμένη αξία ανεξάρτητα από την θέση τους στην αναπαράσταση. Για τον υπολογισμό της τιμής του αριθμού βέβαια, μπορούν άλλες φορές να χρησιμοποιούν αθροιστική σημειολογία και άλλες φορές αφαιρετική σημειολογία [17].

Κατά την αθροιστική σημειολογία, οι αξίες όλων των συμβόλων απλώς προστίθενται για να μας δώσουν το τελικό αποτέλεσμα για την τιμή του αριθμού, ενώ κατά την αφαιρετική αντίστοιχα, όταν ένα σύμβολο μικρότερης αξίας βρίσκεται πριν από ένα μεγαλύτερης, τότε αυτά αφαιρούνται (π.χ. στο ρωμαϊκό σύστημα I=1 και X=10, ενώ IX=(10-1)=9).

Γνωστότερο σύστημα αξίας-συμβόλου είναι το «ρωμαϊκό» (Εικόνα 1), ενώ ένα άλλο, λιγότερο διαδεδομένο είναι το εφευρεθέν υπό του Τζόν Νάπιερ (ή Νέπερ) στο έργο του Ραβδολογία (1617), «σύστημα θέσης» ή «arithmeticæ localis» [18].

1	I	11	XI	50	L
2	II	12	XII	100	C
3	III	13	XIII	500	D
4	IV	14	XIV	1000	M
5	V	15	XV		
6	VI	16	XVI		
7	VII	17	XVII		
8	VIII	18	XVIII		
9	IX	19	XIX		
10	X	20	XX		

Εικόνα 1 Ρωμαϊκό Σύστημα Πηγή: <https://www.saudiinstitute.org/xxv-roman-numerals/>

2.2.2.2. Καταμετρητικό σύστημα

Στα καταμετρητικά συστήματα η αξία του αριθμού υποδηλώνεται με τον σχηματισμό αντίστοιχου πλήθους συμβόλων με τον αναπαριστούμενο αριθμό. Αυτός ο τρόπος αποτελεί την πιο διαισθητική αναπαράσταση αριθμών, αφού για την εξακρίβωση του αριθμού αρκεί κάποιος να καταμετρήσει το πλήθος των συμβόλων. Επιπλέον, ο αναγνώστης της αριθμητικής αναπαράστασης, μπορεί αμέσως να λάβει αίσθηση του μεγέθους του αριθμού με μια απλή επισκόπηση της αναπαράστασης: μεγάλοι αριθμοί – μεγάλο πλήθος συμβόλων, μικροί αριθμοί – λίγα σύμβολα. Ο Bricken στο έργο του σχετικά με την συμβολική αναπαράσταση αριθμών (πρβλ. [19]) επιχειρηματολογεί ότι αυτή η αναπαράσταση, ως πιο διαισθητική, θα πρέπει να είναι η πρώτη με την οποία θα έρχονται σε επαφή οι μαθητές των πρώτων σχολικών τάξεων. Μάλιστα, φτάνει μέχρι το σημείο να δημιουργήσει ολοκληρωμένο σύστημα μετατροπής της κλασσικής

αριθμητικής και κάποιων εννοιών του απειροστικού λογισμού (όπως το άπειρο) σε αυτή την πιο θεμελιώδη και αφαιρετική -όπως την αποκαλεί- αναπαράσταση.

Χαρακτηριστικό παράδειγμα καταμετρητικού συστήματος αποτελούν οι αριθμοί στις όψεις ενός ζαριού, ενώ επίσης γνωστό παράδειγμα τέτοιου συστήματος είναι και το «σύστημα των φυλακισμένων» (Εικόνα 2) όπου κάθε μονάδα συμβολίζεται με μία κάθετη γραμμή και κάθε τέσσερις γραμμές ομαδοποιούνται με μία οριζόντια γραμμή στο μέσον τους, αποτελώντας μια πεντάδα.

1	I	6	I
2	II	7	II
3	III	8	III
4	IIII	9	IIII
5		10	

Εικόνα 2 Σύστημα αρίθμησης "των φυλακισμένων" Πηγή: <https://www.quora.com/What-is-a-positional-number-system>

Ενώ τέτοιες καταμετρητικές αναπαραστάσεις έχουν το πλεονέκτημα της διαίσθησης που εξηγήσαμε παραπάνω -και για αυτό και χρησιμοποιούνται από πλήθος εκπαιδευτικών ειδικής αγωγής για την διδασκαλία αριθμητικής σε παιδιά στο φάσμα του αυτισμού- εντούτοις, αυτό είναι και το μειονέκτημά τους κατα κάποιον τρόπο, διότι το πλήθος των συμβόλων είναι γραμμικό ως προς το μέγεθος του αριθμού, όπως αναλύσαμε.

Αυτό μπορεί να δυσχαιράνει αρκετά την αναπαράσταση μεγάλων αριθμών, καθιστώντας την ακόμη και αδύνατη. Την λύση σε αυτό το πρόβλημα έρχεται να δώσει μία άλλη αναπαράσταση με λογαριθμική σχέση μεταξύ του πλήθους των συμβόλων και της αξίας του αριθμού. Το σύστημα αυτό αναλύεται στην συνέχεια.

2.2.2.3. Θεσιακό σύστημα αρίθμησης

Η εφεύρεση του θεσιακού συστήματος αρίθμησης αποδίδεται σε αρχαιο-ινδικούς πολιτισμούς, ενώ εσχάτως βρέθηκαν ενδείξεις για την χρήση του ήδη από αρχαίους

λαούς της Μεσοποταμίας, Σουμέριους και Ακκάδες και κατοπινά στους κατακτητές τους, Βαβυλώνιους [20].

Για την αναπαράσταση αριθμών στα θεσιακά συστήματα χρειαζόμαστε τον ορισμό δύο παραμέτρων: έναν ακέραιο αριθμό b που ονομάζεται «βάση του συστήματος» (και τότε το σύστημα ονομάζεται b -αδικό) και ένα πεπερασμένο σύνολο συμβόλων με προκαθορισμένη αξία έκαστο [21]. Ο πληθάριθος του συνόλου συμβόλων ισούται με την βάση του συστήματος. Το πιο συνηθισμένο σύνολο συμβόλων είναι αυτό των ινδο-αραβικών ψηφίων, προσαυξημένο -εάν χρειαστεί- με τα γράμματα του λατινικού αλφαβήτου. Παρόλα αυτά, σε χώρες τις Άπω Ανατολής χρησιμοποιείται ακόμη, σε πολλές περιπτώσεις, το σύνολο των ιδεογραμμάτων για τους αντίστοιχους αριθμούς.

Στα θεσιακά συστήματα αρίθμησης η τιμή του αριθμού καθορίζεται από το άθροισμα της αξίας του κάθε ψηφίου στην αναπαράσταση. Τα ψηφία όμως έχουν διαφορετική αξία, ανάλογα με την θέση στην οποία βρίσκονται. Συγκεκριμένα, η αξία είναι το αποτέλεσμα του πολλαπλασιασμού της αξίας του εκάστοτε ψηφίου με κάποιο συγκεκριμένο πολλαπλάσιο της βάσης. Για την ακρίβεια, κάποια δύναμη της βάσης. Η δύναμη στην οποία ωψώνεται η βάση είναι ακριβώς ο αύξων αριθμός (αρχίζοντας την αρίθμηση από το μηδέν) της θέσης στην οποία βρίσκεται το σύμβολο, του οποίου την αξία υπολογίζουμε. Κατά καιρούς έχουν χρησιμοποιηθεί θεσιακά συστήματα αρίθμησης με διαφορετικές βάσεις [17].

Έστω το b -αδικό σύστημα αρίθμησης (βάση το b) και αναπαριστούμε έναν αριθμό ως εξής:

$p_3 p_2 p_1 p_0$. Η τιμή αυτού του αριθμού τότε δίνεται από τον τύπο: $\sum_{i=0}^3 p_i \cdot b^i$.

Πιο αναλυτικά: $p_3 \cdot b^3 + p_2 \cdot b^2 + p_1 \cdot b + p_0$

2.2.2.4. Δυαδικό σύστημα αρίθμησης

Ιδιαίτερη περίπτωση των θεσιακών συστημάτων αποτελεί το δυαδικό σύστημα.

Στο δυαδικό σύστημα η βάση είναι το δύο. Αυτό σημαίνει αφενός ότι μπορούμε να χρησιμοποιήσουμε δύο σύμβολα για να αναπαραστήσουμε τους αριθμούς και αφετέρου καταδεικνύει ότι οι αξία της κάθε θέσης στην αναπαράσταση του αριθμού αποτελεί δύναμη του δύο. Έτσι, έχουμε το λιγότερο σημαντικό ψηφίο, αυτό με την μικρότερη αξία, να "ομαδοποιεί" τις μονάδες. Στην αμέσως επόμενη θέση έχουμε τις δυάδες. Έπειτα, το ψηφίο έχει αξία σε πλήθος τετράδων, στη συνέχεια οκτάδων, δεκαεξάδων κ.ο.κ.

Για να δηλώσουμε ότι ο αριθμός αναπαρίσταται στο δυαδικό σύστημα, εάν αυτό δεν είναι ήδη σαφές, περικλύουμε τον αριθμό σε παρενθέσεις και γράφουμε τον αριθμό δύο σε θέση δείκτη κάτω δεξιά.

Για παράδειγμα, ο αριθμός $(011)_2$, με το άκρως δεξιό ψηφίο να αποτελεί το λιγότερο σημαντικό ψηφίο, αποτελείται από μία (1) μονάδα, μία (1) δυάδα και καμία (0) τετράδα. Άρα, ο αριθμός αυτός είναι το τρία. Θα μπορούσαμε να γράψουμε: $(011)_2 = (3)_{10}$.

Όσον αφορά στην πρόσθεση δύο αριθμών στο δυαδικό σύστημα, αυτή ακολουθεί τους κανόνες της κλασσικής αριθμητικής [23]. Βέβαια, τώρα η υπερχείλιση (=το κρατούμενο) συμβαίνει όταν ξεπεράσουμε το αποτέλεσμα της πρόσθεσης ξεπεράσει το ένα. Άλλωστε, αυτό είναι χαρακτηριστικό της πρόσθεσης στα θεσιακά συστήματα αρίθμησης, όταν το άθροισμα ξεπεράσει την τιμή της βάσης του συστήματος μείον ένα να έχουμε κρατούμενο (το οποίο μπορεί να διαδοθεί στις επόμενες θέσεις του αριθμού κατά τη εκτέλεση της διαδικασίας-αλγορίθμου της πρόσθεσης εφόσον έχουμε κι άλλα ψηφία κατόπιν).

Έτσι η πρόσθεση μπορεί να γίνει με βάση τους επόμενους τέσσερις κανόνες:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 1 = 0$ (και έχουμε 1 κρατούμενο)
- $1 + 1 + 1 = 1$ (και έχουμε 1 κρατούμενο)

Παράδειγμα πρόσθεσης δύο αριθμών:

$$\begin{array}{r} 011011 \\ 011010 + \\ \hline 110101 \end{array}$$

Παρατηρείστε ότι στην δεύτερη θέση έχουμε υπερχείλιση και κατά συνέπεια το κρατούμενο μεταφέρεται στην τρίτη θέση του αποτελέσματος, ενώ έχουμε διαδοχικές

υπερχειλίσσεις στις θέσεις τέσσερα και πέντε και έτσι το κρατούμενο μεταφέρεται διαδοχικά από την θέση τέσσερα, στην πέντε και τέλος στην έξι του τελικού αποτελέσματος.

Στην παρούσα εργασία επιλέξαμε να ασχοληθούμε συγκεκριμένα με το δυαδικό σύστημα αρίθμησης αφού αυτό αποτελεί τον βασικό τρόπο μαθηματικού φορμαλισμού λογικής πρώτης τάξης (προτασιακή λογική) και ακολούθως της λογικής σχεδίασης (αρχές και θεωρία σχεδιασμού λογικών κυκλωμάτων με βάση την προτασιακή λογική). Σε αυτήν την σύντομη εισαγωγή παρουσιάστηκε η λογική πίσω από την αναπαράσταση των αριθμών στο δυαδικό, καθώς και η πράξη της πρόσθεσης. Οι υπόλοιπες αριθμητικές πράξεις στο δυαδικό σύστημα είναι πέρα από τους σκοπούς της παρούσας εργασίας αφού δεν διδάσκονται στο βιντεοπαιχνίδι της εικονικής πραγματικότητας που παρουσιάζεται στην συνέχεια.

2.3 Σύγκριση με παρεμφερή παίγνια για την εκμάθηση του δυαδικού συστήματος

2.3.1. Cisco Binary Game

Το Cisco Binary Game ([24], Εικόνα 3) είναι ένα σύνολο προκλήσεων που οργανώνονται ως πολυεπίπεδο κουίζ. Προσφέρεται ως εργαλείο για να βοηθήσει στην επιτυχία στις εξετάσεις CCENT ή CCNA Routing and Switching, και συγκεκριμένα στο μέρος εκείνο των εξετάσεων που αναφέρεται στη μετατροπή δεκαδικών αριθμών σε δυαδικούς αριθμούς και δυαδικών αριθμών σε δεκαδικούς αριθμούς και μάλιστα γρήγορα. Στην Εικόνα 3 παρουσιάζεται ένα χαρακτηριστικό στιγμιότυπο οθόνης αυτού του παιχνιδιού. Η βασική διαφορά όμως αυτού του παιχνιδιού με την δική μας πρόταση παιχνιδιού είναι ότι το Cisco Binary Game επικεντρώνεται αποκλειστικά στην εκμάθηση της δυαδικής αναπαράστασης των αριθμών χωρίς να έχει ως στόχο τη χρήση της δυαδικής αναπαράστασης για τη διερεύνηση των αριθμητικών πράξεων, σε αντίθεση με το παρουσιαζόμενο ηλ. παίγνιο.



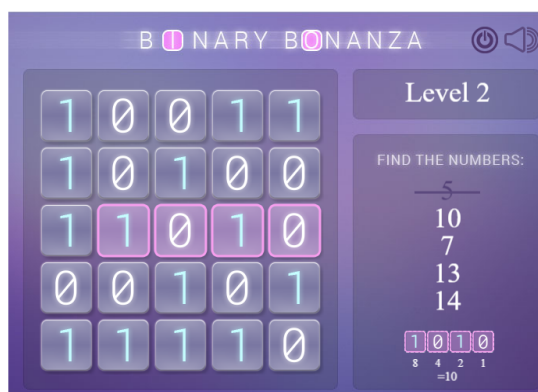
Εικόνα 3 Στιγμιότυπο οθόνης του Cisco Binary Game

Εν κατακλείδι, το παιχνίδι επικεντρώνεται μόνο στη δυαδική αναπαράσταση των αριθμών, χωρίς περαιτέρω στόχο να εξερευνήσει τις αριθμητικές πράξεις στο δυαδικό σύστημα, σε αντίθεση με το παιχνίδι που παρουσιάζεται στην παρούσα εργασία. Για τη μετατροπή από το δεκαδικό στο δυαδικό σύστημα, και το παιχνίδι που παρουσιάζεται στην παρούσα εργασία υιοθετεί μια πράξη καταμέτρησης.

Με άλλα λόγια, για να δημιουργηθεί οποιοσδήποτε αριθμός (μεταξύ 0 και 15) απλώς προσθέτουμε άνθη στην κανάτα μικρότερης αξίας -δηλαδή των μονάδων, πράγμα που ισοδυναμεί με το μέτρημα από το μηδέν μέχρι τον απαιτούμενο αριθμό.

2.3.2. Binary bonanza

Το Binary bonanza ([25], Εικόνα 4 Στιγμιότυπο οθόνης Binary Bonanza. Εικόνα 4) είναι ένα ακόμη ψηφιακό παιχνίδι για την εκμάθηση αποκλειστικά της δυαδικής αναπαράστασης των αριθμών. Ο παίκτης καλείται να σύρει-επιλέξει έναν αριθμό bits στον τετράγωνο πίνακα στα αριστερά για να δημιουργήσει τους ζητούμενους αριθμούς στη λίστα στα δεξιά, όπως φαίνεται στην Εικόνα 4. Το σημείο όπου υπολείπεται και αυτό το ηλ. παιχνίδι σε σχέση με το παρουσιαζόμενο είναι ότι δεν ασχολείται με τις βασικές αριθμητικές πράξεις στο δυαδικό σύστημα, αλλά μόνο την αναπαράσταση των αριθμών σε αυτό.



Εικόνα 4 Στιγμιότυπο οθόνης Binary Bonanza.

2.3.3. Binary Dots

Τέλος, ένα παιχνίδι με αρκετές ομοιότητες με το επιτραπέζιο παιχνίδι που περιγράφεται στην παρούσα εργασία είναι το Binary Dots ([26], Εικόνα 5). Η δυαδική αναπαράσταση ενός αριθμού παρουσιάζεται σε έναν πίνακα που παρουσιάζει τη σημασία ή κάθε δυαδικό ψηφίο. Μια κίτρινη μάρκα που τοποθετείται σε μια συγκεκριμένη θέση του ταμπλό αντιπροσωπεύει το 0, ενώ μια κόκκινη μάρκα αντιπροσωπεύει το 1 (Εικόνα 5). Μόνο οι δυαδικές αναπαραστάσεις των αριθμών εξετάζονται και από αυτό το παιχνίδι.

32	16	8	4	2	1
●	●	●	●	●	●

$$43 = 2^5 + 2^3 + 2^1 + 2^0 = 101011_2$$

Εικόνα 5 Η πλακέτα δραστηριοτήτων Binary Dots.

Κεφάλαιο 3 - Μηχανές Παιχνιδιών

Στο κεφάλαιο αυτό επιχειρείται μια παρουσίαση του χώρου των μηχανών δημιουργίας παιχνιδιών ώστε να αποκτήσει ο αναγνώστης μια εικόνα των διαθέσιμων επιλογών και να μπορεί να επιλέξει την κατάλληλη για εκείνον. Κατά το διάστημα συγγραφής της παρούσας πτυχιακής εργασίας ο αναγνώστης μπορούσε να εντοπίσει στο εμπόριο (αλλά και ανοιχτού κώδικα) πληθώρα μηχανών που βρίσκουν εφαρμογή όχι μόνο στο τομέα της κατασκευής παιχνιδιών αλλά και σε άλλους τομείς όπως στο κινηματογράφο, την αυτοκινητοβιομηχανία, την αρχιτεκτονική καθώς και άλλου. Ο συγγραφέας επέλεξε να παρουσιάσει τις πλέον καθιερωμένες και αντιπροσωπευτικές προτάσεις ώστε να

αποκτήσει ο αναγνώστης μια όσο το δυνατόν πιο πληρέστερη εικόνα. Δεδομένου ότι στην παρούσα εργασία χρησιμοποιήθηκε η Unity, στο τέλος αναλύονται διεξοδικά τα στοιχεία της Unity τα οποία βοήθησαν στην σχεδίαση του παιχνιδιού.

3.1. Εισαγωγή

Ο όρος μηχανή παιχνιδιού αναφέρεται σε ένα λογισμικό γενικού τύπου που έχει σαν στόχο την ανάπτυξη παιχνιδιών που θα εκτελούνται σε ηλεκτρονικό υπολογιστή. Φυσικά, ο όρος ηλεκτρονικός υπολογιστής είναι πλέον αρκετά ευρύς και περιλαμβάνει όχι μόνο τους προσωπικούς ηλεκτρονικούς υπολογιστές και τις παιχνιδομηχανές αλλά επιπλέον περιλαμβάνει συσκευές τύπου tablet, έξυπνα κινητά τηλέφωνα καθώς και άλλα. Το λογισμικό αποτελείται από μια συλλογή από βιβλιοθήκες, υποστηρικτικά προγράμματα καθώς και διάφορα εργαλεία για την κατασκευή των παιχνιδιών. Ενδεικτικά της λειτουργικότητας που μπορεί να προσφέρει ένα λογισμικό τέτοιου τύπου είναι η μηχανή ρεαλιστικής απόδοσης των διαφόρων μοντέλων και περιβαλλόντων εφαρμόζοντας κατάλληλα τα διάφορα χρώματα, τις υφές και τα φώτα που έχει επιλέξει ο κατασκευαστής (Render Engine), λογισμικό κατασκευής μοντέλων (Τριών διαστάσεων ή δύο), λογισμικό κατασκευής κινούμενων εικόνων (animation) καθώς και εργαλεία που σχετίζονται με τη πλατφόρμα, όπως εργαλεία διαχείρισης μνήμης, διαχείρισης των διαφόρων εργασιών κλπ. Τα εργαλεία αυτά προσφέρονται μέσω ενός Ολοκληρωμένου Συστήματος Ανάπτυξης Εφαρμογών (IDE) το οποίο διευκολύνει σημαντικά τη διαδικασία κατασκευής της εφαρμογής.

Οι μηχανές προσφέρουν την αίσθηση της γενίκευσής (Abstraction) με την έννοια ότι ο κατασκευαστής δημιουργεί μια εφαρμογή η οποία με μικρό αριθμό προσθηκών (ή και καθόλου) μπορεί να εκτελεστεί σε διάφορες πλατφόρμες. Με τον τρόπο αυτό ο κατασκευαστής εστιάζει σε θέματα που έχουν να κάνουν με την εφαρμογή και τις λεπτομέρειες που έχουν να κάνουν με τη πλατφόρμα (όπως η διαχείριση της μνήμης, των διεργασιών, καθώς και άλλα), τις διαχειρίζεται το εργαλείο. Η διαδικασία αυτή έχει σαν αποτέλεσμα να εξοικονομούνται σημαντικοί πόροι όχι μόνο οικονομικοί αλλά και σε χρόνο καθώς δεν απαιτείται η κατασκευαστική ομάδα να αποτελείται και από ειδικούς σε θέματα πλατφόρμων ούτε απαιτούνται εξαντλητικές δοκιμές της εφαρμογής στις διάφορες πλατφόρμες.

Από την άλλη πλευρά, η κάθε μηχανή παρουσιάζει τα δικά της πλεονεκτήματα και μειονεκτήματα, παρόλο που τα τελευταία χρόνια έχει καταβληθεί ιδιαίτερη προσπάθεια από τους κατασκευαστές να δημιουργήσουν προϊόντα που να ανταποκρίνονται στις

απαιτήσεις της πλειονότητας των κατασκευαστών εφαρμογών αυτού του είδους. Ένα από τα σημαντικότερα κριτήρια αποτελεί το κόστος χρήσης μιας τέτοιας μηχανής με τις περισσότερες εξ αυτών να εφαρμόζουν παρόμοια μοντέλα. Πιο συγκεκριμένα, οι περισσότερες μηχανές που δοκιμάστηκαν διατίθενται δωρεάν για εφαρμογές που αποφέρουν μικρά κέρδη στους κατασκευαστές τους. Άλλο κριτήριο επιλογής αποτελεί το είδος της εφαρμογής που θα φτιαχτεί, με μηχανές που στοχεύουν στη κατασκευή τρισδιάστατων ή δυσδιάστατων εφαρμογών -παιχνιδιών ή μη- καθώς και άλλων. Τέλος, σημαντικό ρόλο διαδραματίζουν και οι κατασκευαστές των εφαρμογών. Άλλες μηχανές είναι κατάλληλες για χρήση από ομάδες και άλλες από ένα και μόνο άτομο.

3.2. Unreal

Η παρουσίαση των μηχανών θα ξεκινήσει από την Unreal (, Εικόνα 6) η οποία εμφανίστηκε το 1998 από την εταιρία Epic Games. Δημοφιλές προϊόν της εταιρίας αποτελεί το ιδιαίτερα επιτυχημένο και ομώνυμο παιχνίδι Unreal το οποίο κυκλοφόρησε το 1998. Η μηχανή αρχικά εστίαζε στους προσωπικούς ηλεκτρονικούς υπολογιστές και στα παιχνίδια πρώτου προσώπου επίθεσης (First Person Shooter), στη πάροδο των ετών όμως διεύρυνε τα πεδία εφαρμογής της και πλέον χρησιμοποιείται για την παραγωγή διαφόρων ειδών παιχνιδιών. Παράλληλα επεκτάθηκε και σε άλλους τομείς όπως στο κινηματογράφο αλλά και στη μικρή οθόνη. Αρχικά είχε γραφεί σε Visual Basic, οι επόμενες εκδόσεις όμως (Unreal Engine 2) χρησιμοποίησαν τη C++ και όλα τα πλεονεκτήματα που αυτή είχε να προσφέρει (ταχύτητα, μεταφερσιμότητα, κλπ). Ο κώδικας της μηχανής είναι ελεύθερος στο δημοφιλή αποθετήριο κώδικα GitHub για τους εγγεγραμμένους χρήστες. Τέλος, για τη χρήση της μηχανής η εταιρία διαχωρίζει τη χρήση σε προσωπική και εμπορική. Η προσωπική χρήση είναι δωρεάν ενώ για την εμπορική η εταιρία ζητά το 5% των κερδών. Στη περίπτωση που η εφαρμογή που θα παραχθεί διατεθεί μέσω του καταστήματος της εταιρίας Epic Games Store τότε και η εμπορική χρήση είναι δωρεάν.



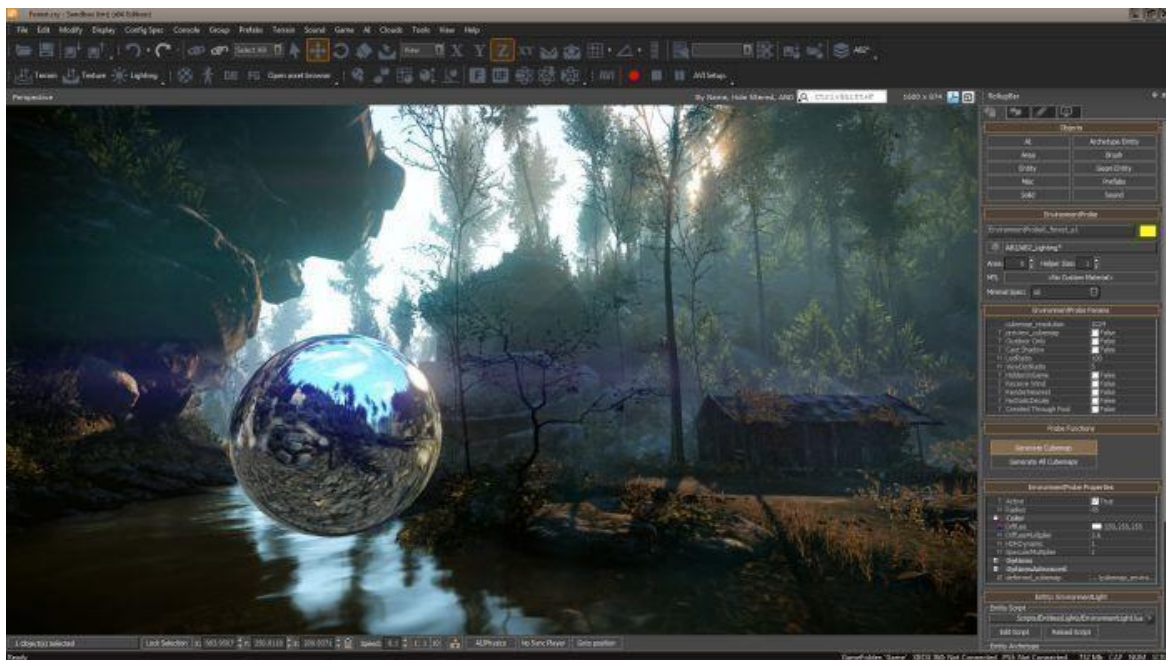
Εικόνα 6 Unreal Ide. Πηγή: By Unreal Engine 4.20 Released!, Fair use, <https://en.wikipedia.org/w/index.php?curid=57933430>

Στις αρχικές τις εκδόσεις οι προγραμματιστές χρησιμοποιούσαν την UnrealScript σαν γλώσσα προγραμματισμού για να συντάξουν τον κώδικα του παιχνιδιού καθώς και τα διάφορα συμβάντα του παιχνιδιού. Η γλώσσα παρουσίαζε αρκετά κοινά σημεία με τη Java όπως το ότι ήταν αντικειμενοστραφής και δεν υποστήριζε την πολλαπλή κληρονομικότητα που υποστηρίζει η C++, συνεπώς επρόκειτο για μια απλή και εύκολη στη χρήση γλώσσα προγραμματισμού. Από την άλλη πλευρά, διέφερε από τη Java στο ότι δεν διέθετε κλάσεις για τους βασικούς τύπους δεδομένων, ούτε υπερφόρτωση τελεστών και συναρτήσεων. Τέλος, η υποστήριξη διαπροσωπειών (interfaces) εμφανίστηκε από την έκδοση δύο και έπειτα. Η γλώσσα χρησιμοποιούνταν έως και το 2012 όπου αντικαταστάθηκε πλήρως από τη C++.

3.3. Cryengine

Από την παρουσίαση των μηχανών παιχνιδιών δεν θα μπορούσε να απουσιάζει το προϊόν Cryengine (, Εικόνα 7) της γερμανικής εταιρίας Crytek από το οποίο προέκυψαν τα ιδιαίτερα αγαπημένα παιχνίδια Far Cry, Crysis καθώς και πληθώρα άλλων. Η εταιρία ξεκίνησε το 2002 από τους αδελφούς Yerli στο Κομπουργκ της Βαυαρίας και η μηχανή χρησιμοποιήθηκε από την εταιρία για τα δικά της προϊόντα. Ακολούθως η μηχανή της χρησιμοποιήθηκε και από άλλες εταιρίες είτε αυτούσια είτε τροποποιημένη, για να δημιουργήσουν τα δικά τους επιτυχημένα παιχνίδια όπως το Sniper, το Ghost Warrior και το Snow.

Η εταιρία, μέχρι το διάστημα συγγραφής της παρούσας πτυχιακής εργασίας είχε δημιουργήσει έξι εκδόσεις της μηχανής της. Η αρχή έγινε με την πρώτη έκδοση η οποία χρησιμοποιήθηκε στη κατασκευή του παιχνιδιού Far Cry. Αρχικά ο στόχος την μηχανής δεν ήταν η κατασκευή του παιχνιδιού, αλλά να παρουσιάσει τις δυνατότητες της τότε NVIDIA. Η εταιρία όμως είδε τις δυνατότητες της μηχανής και αποφάσισε να δημιουργήσει το παιχνίδι. Η επόμενη έκδοση της μηχανής χρησιμοποιήθηκε για την κατασκευή του Crysis για υπολογιστή, ενώ η υποστήριξη των υπολοίπων πλατφόρμων (Xbox 360, PlayStation 3 και Wii U) έγινε εφικτή το 2009 με την έκδοση 3.



Εικόνα 7 Cryengine SDK. Πηγή: www.geeks3d.com

Το 2016 η εταιρία ανανέωσε το μοντέλο χρέωσης ώστε να ανταγωνίζεται και τις εταιρίες που κυριαρχούν στο χώρο σε ένα μοντέλο «Πλήρωσε αυτό που χρειάζεσαι». Σύμφωνα με το μοντέλο αυτό, οι χρήστες της μηχανής είναι υποχρεωμένοι να πληρώσουν μόνο όταν τα κέρδη της εφαρμογής ξεπερνούν τα 5.000 δολάρια ΗΠΑ. Το 2015 η εταιρία Amazon αγόρασε την μηχανή και πλέον κυκλοφορεί το δικό της προϊόν με το όνομα Amazon Lumberyard.

3.4. Unity

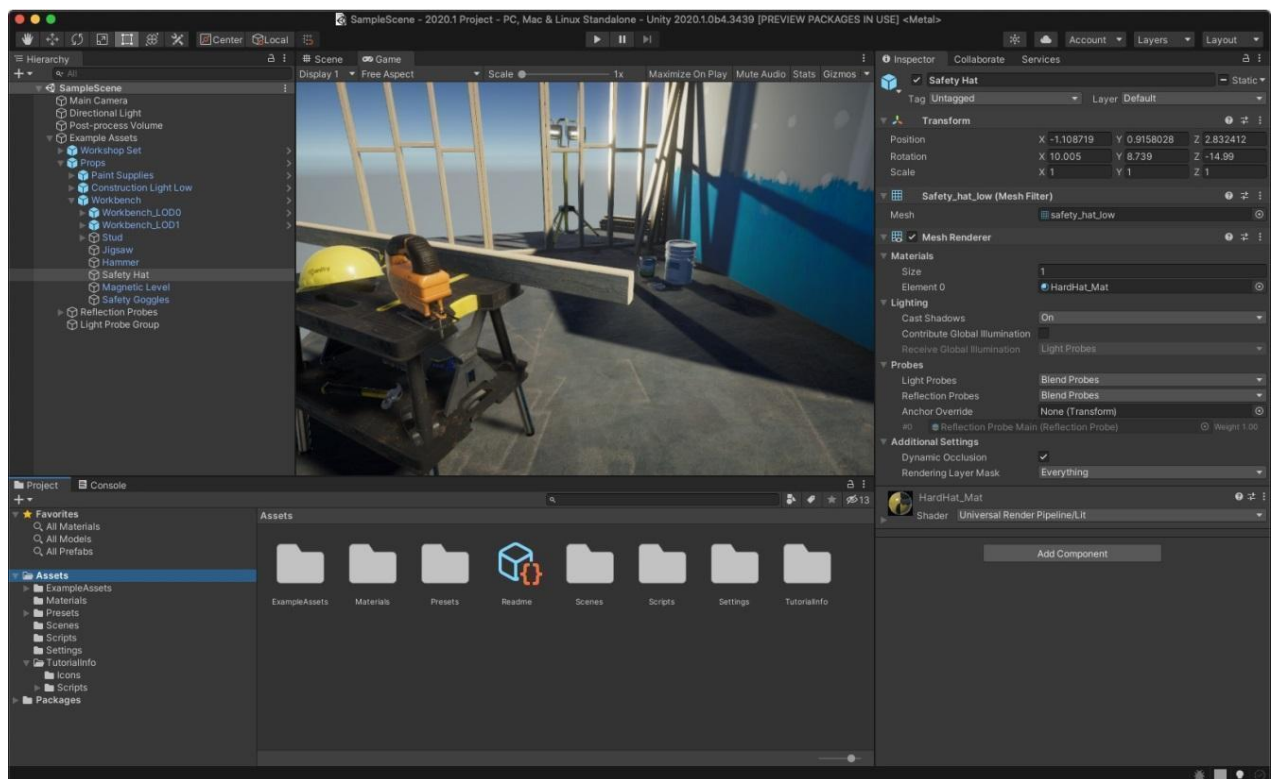
Η Unity (, Εικόνα 8) η οποία θα χρησιμοποιηθεί στη παρούσα πτυχιακή εργασία, παρουσιάστηκε από την ομώνυμη εταιρία Unity Technologies τον Ιούνιο του 2005 στο παγκόσμιο συνέδριο των κατασκευαστών εφαρμογών της Apple. Αρχικά η μηχανή στόχευε στη δημιουργία εφαρμογών μόνο για το λειτουργικό Mac OS X, όμως σταδιακά εξελίχθηκε και αυτό ώστε να υποστηρίζει την πλειονότητα των επικρατούμενων πλατφόρμων, όπως προσωπικοί υπολογιστές, έξυπνα κινητά τηλέφωνα, παιχνιδομηχανές αλλά και πλατφόρμες εικονικής πραγματικότητας. Πλέον, η μηχανή είναι αρκετά δημοφιλής στα λειτουργικά συστήματα iOS και Android και είναι κατάλληλη για την κατασκευή εφαρμογών τόσο τριών διαστάσεων όσο και δύο. Όπως και οι άλλες μηχανές που παρουσιάστηκαν στις προηγούμενες παραγράφους έτσι και η Unity δεν περιορίζεται μόνο στο τομέα των παιχνιδιών αλλά επεκτείνεται και σε άλλους τομείς όπως στο κινηματογράφο, στην μηχανική, στην αρχιτεκτονική, στην αυτοκινητοβιομηχανία, στο στρατό των ΗΠΑ αλλά και αλλού.

Για τη συγγραφή των διαφόρων προγραμμάτων των εφαρμογών η Unity αρχικά χρησιμοποίησε τη Boo η οποία το 2015 αντικαταστάθηκε από μία υλοποίηση της Javascript η οποία βασιζόταν επίσης στη Boo. Από το 2017 και έπειτα όμως η εταιρία χρησιμοποιεί τη C#.

Όσο αφορά το μοντέλο πληρωμής, η εταιρία τα πρώτα δέκα χρόνια παρείχε δωρεάν άδεια χρήσης περιορισμένων δυνατοτήτων και μία επί πληρωμής. Ακολούθως, το 2016 το μοντέλο τροποποιήθηκε και η δωρεάν άδεια μπορεί να χρησιμοποιηθεί από μικρές εταιρίες με ετήσιο κύκλο εργασίας τις 100.000 δολάρια ΗΠΑ, ποσόν το οποίο στην συνέχεια αυξήθηκε στα 200.000 δολάρια ΗΠΑ. Ακολούθως, το 2021 η εταιρία προσάρμοσε το μοντέλο της και πλέον απαιτείται η επαγγελματική άδεια για την ανάπτυξη εφαρμογών για κονσόλες, η οποία παρέχεται από τον κατασκευαστή της εκάστοτε κονσόλας.

Γενικώς, η συγκεκριμένη μηχανή θεωρείται από τις πλέον απλές στη χρήση και συνεπώς είναι κατάλληλη για προγραμματιστές που δεν διαθέτουν μεγάλη εμπειρία στη κατασκευή τέτοιου είδους εφαρμογών. Παράλληλα, λόγω της απλότητας της είναι κατάλληλη για εφαρμογές που αναπτύσσονται από μικρές ομάδες -ακόμα και από ένα

άτομο. Οι λόγοι αυτοί συνετέλεσαν στο να επιλεγθεί η συγκεκριμένη μηχανή για την παρούσα εργασία.



Εικόνα 8 Περιβάλλον ανάπτυξης Unity. Πηγή: www.unity3d.com

3.5. Στοιχεία της Unity στην εργασία

Τα στοιχεία της μηχανής ανάπτυξης βιντεοπαιχνιδιών Unity που χρησιμοποιήθηκαν αναφέρονται παρακάτω, ενώ παράλληλα αναλύεται η χρησιμότητά τους κατά την διαδικασία ανάπτυξης του παιχνιδιού.

3.5.1. Μηχανή γραφικών (rendering engine)

Το στοιχείο αυτό, που ονομάζεται επίσης μηχανή απεικόνισης, είναι υπεύθυνο για τη δισδιάστατη ή τρισδιάστατη απεικόνιση των οπτικών γραφικών του παιχνιδιού. Τα γραφικά ενός παιχνιδιού είναι ζωτικής σημασίας για την αισθητική του εμφάνιση και την ελληλεπίδραση με τον χρήστη. Η μηχανές γραφικών αποτελούνται από πολλά επιμέρους στοιχεία που αναγνωρίζονται με τον συλλογικό όρο σωλήνωση γραφικών

(graphics pipeline) με απαραίτητες λειτουργίες την τριγωνοποίηση των γραφικών, την σκίαση με βάση αλγόριθμους φωτισμού και η τελική ραστεροποίηση (rasterization) στο τέλος του rendering pipeline.

Το συγκεκριμένο εργαλείο χρησιμοποιήθηκε για μια σειρά από καλλιτεχνικά χαρακτηριστικά, όπως εφέ φωτισμού, σκιάσεις και χάρτες bump και ανάμειξης, ώστε να επιτυγχάνεται πραγματικά η εμβύθιση του παίκτη στον κόσμο του παιχνιδιού. Για την εργασία χρησιμοποιήθηκε η προκαθορισμένη μηχανή γραφικών της Unity με τους renderers που παρέχονται στο βασικό πακέτο.

3.5.2. Διαχείριση εισόδου από τα περιφερειακά

Περιλαμβάνει συσκευές που θα καταγράφουν τις ενέργειες του παίκτη, διασφαλίζοντας ότι το παιχνίδι μπορεί να παιχτεί πραγματικά λαμβάνοντας είσοδο από τον παίκτη. Η μηχανή πρέπει να υποστηρίζει την είσοδο συσκευών όπως ποντίκι, πληκτρολόγιο, gamepad ή joystick.

Η μηχανή του παιχνιδιού χειρίζεται την είσοδο με δύο τρόπους:

- Συμβάντα εισόδου - ο υπολογιστής ανιχνεύει τις ενέργειες του παίκτη και ενεργοποιεί έναν προσαρμοσμένο κώδικα.
- Σφυγμομετρηση (polling) εισόδου - καταγράφει τιμές θέσης, όπως η φυσική κλίση ενός χειριστηρίου.

Στό παρόν παιχνίδι μπορεί να χρησιμοποιηθεί είτε ένας συνδυασμός πληκτρολογίου και ποντικιού, είτε ένα χειριστήριο παιχνιδιού (joystick), με την Unity να αναλαμβάνει την διαφοροποίηση του είδους της συσκευής εισόδου, από τα πραγματικά σήματα που λαμβάνει ο κώδικας σαν συμβάντα-διακοπές. Έτσι δεν είναι απαραίτητη στον κώδικα η μέριμνα για έλεγχο της εισόδου, αν είναι από πληκτρολόγιο, ποντίκι, joystick ή ό,τι άλλο συμβατό. Βέβαια, αυτό είναι ένα πρόσφατο χαρακτηριστικό της Unity και για αυτό έπρεπε να χρησιμοποιήσουμε την πιο πρόσφατη έκδοση της βιβλιοθήκης εισόδου. Επιπλέον, με την χρήση του εργαλείου διαχείρισης εισόδου (Input Manager) κατέστη δυνατή η αντιστοίχιση των φυσικών πλήκτρων με τους κωδικούς πλήκτρων στη φυσική διάταξη του πληκτρολογίου και όχι στη διάταξη που καθορίζεται από τη γλώσσα, η οποία μπορεί να διαφέρει μεταξύ των χρηστών σε διαφορετικές περιοχές.

Για παράδειγμα, σε ορισμένα πληκτρολόγια η πρώτη σειρά γραμμάτων γράφει "QWERTY", ενώ σε άλλα γράφει "AZERTY". Αυτό σημαίνει ότι αν είχε δημιουργηθεί συγκεκριμένο σενάριο ελέγχου όπου απαιτείται η χρήση των γνωστών πλήκτρων "WASD" για την κίνηση του παίκτη, αυτά δεν θα είναι στη σωστή φυσική διάταξη (όπως η διάταξη των πλήκτρων βέλους) σε ένα πληκτρολόγιο με διάταξη AZERTY.

3.5.3. Τεχνητή νοημοσύνη

Η τεχνητή νοημοσύνη (AI) παίζει μεγαλύτερο ρόλο από ποτέ στην ανάπτυξη παιχνιδιών και στο ίδιο το gameplay. Η τεχνητή νοημοσύνη έχει επιτρέψει στα παιχνίδια να προχωρήσουν όχι μόνο στο σχεδιασμό του λογισμικού τους αλλά και στην αφήγηση της ιστορίας τους.

Στην παρούσα εργασία χρησιμοποιήθηκε το πακέτο AI Navigation το οποίο επιτρέπει την δημιουργία χαρακτήρων που μπορούν να κινούνται έξυπνα στον κόσμο του παιχνιδιού. Αυτοί οι χαρακτήρες χρησιμοποιούν πλέγματα πλοήγησης (NavMesh) που δημιουργούνται αυτόματα από τη γεωμετρία της Σκηνής με αλγορίθμους τριγωνοποίησης. Τα δυναμικά εμπόδια επιτρέπουν τις αλλαγές στην πλοήγηση των χαρακτήρων κατά την εκτέλεση, ενώ οι σύνδεσμοι εκτός πλέγματος επιτρέπουν την δημιουργία συγκεκριμένων ενεργειών, όπως το άνοιγμα θυρών ή το άλμα πάνω από κενά ή από ένα περβάζι.

Η χρήση του πακέτου AI Navigation έγινε για την αυτόνομη κίνηση των εχθρών στο δάσος (ενός είδους περιπολίας από σημείο σε σημείο περισσότερο, παρά τυχαίος περίπατος στο δάσος).

3.5.4. Μηχανή προσομοίωσης φυσικής

Η φυσική είναι συνήθως κάτι που θεωρούμε δεδομένο, ειδικά για ένα παιχνίδι, αλλά πρόκειται για ένα σημαντικό χαρακτηριστικό που παρέχει την εμβύθιση στον κόσμο ενός παιχνιδιού. Χωρίς μια μηχανή φυσικής, η κίνηση θα φαινόταν απλώς αλλόκοτη και αφύσικη.

Οι μηχανές παιχνιδιών εφαρμόζουν το φυσικό στοιχείο που περιμένουμε μέσα στη δραστηριότητα του παιχνιδιού. Αυτό περιλαμβάνει την βαρύτητα, την ταχύτητα, την επιτάχυνση, τη ρευστότητα και την κίνηση των βλημάτων.

Σημαντικό στοιχείο για να λειτουργήσει όπως αναμένεται η μηχανή φυσικής της Unity είναι ο ορισμός της περιοχής του μοντέλου που αυτό μπορεί να αλληλεπιδράσει με άλλα φυσικά αντικείμενα στο παιχνίδι. Αυτή η περιοχή οριοθετείται από τον προγραμματιστή χειροκίνητα και ονομάζεται Collider. Έτσι στο παιχνίδι μας ορίσαμε Colliders για κάθε χαρακτήρα (παίκτης και εχθρικά bots) καθώς και τα αντικείμενα με τα οποία καλείται να αλληλεπιδράσει ο χρήστης (μαγικά βότανα, καζάνια, τραπέζι, φράκτης, καλύβα κ.α.).

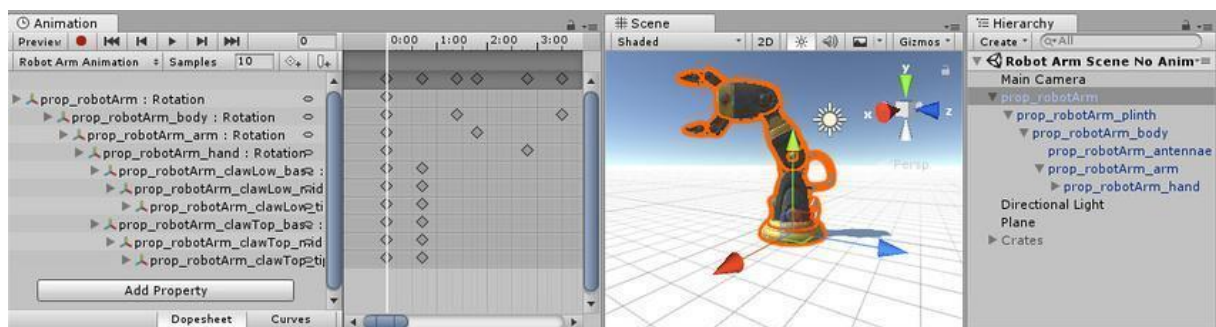
3.5.5. Μηχανή ήχου

Η μηχανή ήχου διαχειρίζεται τους ήχους ενός παιχνιδιού. Τα αρχεία ήχου ενσωματώνονται από τη μηχανή παιχνιδιού στον κώδικα και ενεργοποιούνται από συγκεκριμένες ενέργειες που αντιστοιχούν στο παιχνίδι. Για παράδειγμα, στην παρούσα εργασία χρησιμοποιήθηκαν αρχεία ήχου κατά τον βηματισμό του παίκτη και την προσγείωση έπειτα από άλμα.

3.5.6. Animation

Εδώ κατασκευάζονται και επεξεργάζονται τα animations χρησιμοποιώντας τα τρισδιάστατα μοντέλα του παιχνιδιού. Συγκεκριμένα, ορίζονται στιγμιότυπα του animation όπου το μοντέλο βρίσκεται σε συγκεκριμένη θέση (όπως καθορίζεται από διάφορες παραμέτρους που επιλέγει ο σχεδιαστής του animation). Η συλλογή αυτών των στιγμιότυπων δίνει την αίσθηση της μεταβολής σε κάθε καρέ στοιχείων και μεταβλητών του αντικειμένου.

Τα μοντέλα που χρησιμοποιήθηκαν για το παρόν παιχνίδι συμπεριλάμβαναν και τα αρχεία animation. Σε αυτά έγιναν μόνο κάποιες μικρές παρεμβάσεις με σκοπό την βελτίωση της αληθοφάνειας των κινήσεων των χαρακτήρων.



Εικόνα 9 Διεπαφή εργαλείου animation

3.5.7. Animator

Το εργαλείο Animation της Unity πάει 'χέρι-χέρι' με το εργαλείο Animator. Αυτό είναι μια στρωματοποιημένη μηχανή λογικών καταστάσεων με μεταβάσεις μεταξύ τους που γίνονται βάσει αλλαγών παραμέτρων (συνθηκών) που δηλώνονται μέσω κώδικα. Π.χ. Έστω ότι έχουμε δύο διαφορετικά animations, ένα για περπάτημα και ένα για το τρέξιμο του χαρακτήρα. Προσθέτουμε τα animations βάρδησης και τρεξίματος στον animator ως καταστάσεις (states), τα ενώνουμε με μια συνθήκη αλλαγής που επηρεάζεται από μία παράμετρο (π.χ. boolean), αλλάζουμε την παράμετρο μέσω κώδικα όταν θελήσουμε (π.χ. εάν πατήσουμε ένα κουμπί η boolean παράμετρος λαμβάνει την τιμή True και η συνθήκη γίνεται αληθής) και έτσι γίνεται η μετάβαση μεταξύ των δύο καταστάσεων.

Στην εργασία χρειάστηκε να δημιουργηθεί μία μηχανή καταστάσεων για την μετάβαση μεταξύ των καταστάσεων βάρδησης, τρεξίματος, άλματος και στάσης, αλλά και μία δεύτερη μηχανή καταστάσεων για την μετάβαση στο animation μονομαχίας και έξοδο από αυτό (και επιστροφή στο animation στάσης-βαδίσματος).

3.5.8. Ολοκληρωμένο περιβάλλον ανάπτυξης (IDE)

Ονομάζουμε "Περιβάλλον Ανάπτυξης" ή "Integrated Development Environment - Ολοκληρωμένο περιβάλλον ανάπτυξης " (IDE) ένα ειδικό λογισμικό που βοηθά τους προγραμματιστές να δημιουργήσουν λογισμικό. Το βασικό IDE περιλαμβάνει επεξεργαστή, έναν μεταγλωττιστή και ένα εργαλείο εντοπισμού σφαλμάτων, αν και τα σύγχρονα IDE διαθέτουν πολλά άλλα βοηθητικά εργαλεία.

Τα IDE ανάπτυξης παιχνιδιών έχουν διάφορα ειδικά χαρακτηριστικά, αλλά δύο είναι τα πιο σημαντικά. Το πρώτο είναι η ενσωμάτωση πολλαπλών εργαλείων για την παραγωγή διαφορετικών τμημάτων ενός παιχνιδιού (γραφικά, ήχοι, προγραμματισμός, δικτύωση κ.λπ.) και το δεύτερο σημαντικό χαρακτηριστικό είναι η εξειδίκευση στην ανάπτυξη παιχνιδιών, όπου τα περισσότερα στοιχεία ενός παιχνιδιού είναι προσχεδιασμένα, όπως επίπεδα, χάρτες, χαρακτήρες, animations, κλπ.

Για το συγκεκριμένο παιχνίδι χρησιμοποιήσαμε το IDE της Unity, κάνοντας εκτενή χρήση της βιβλιοθήκης της Unity, UnityEngine. Τα αρχεία του πηγαίου κώδικα ήταν γραμμένα σε C#, αφού πλέον αυτή είναι η μόνη γλώσσα προγραμματισμού που υποστηρίζει η Unity. Χαρακτηριστικό της Unity είναι ότι για τον προγραμματισμό των αντικειμένων του

παιχνιδιού που αλληλεπιδρούν με άλλα στοιχεία του παιχνιδιού είναι η απαραίτητη η ενσωμάτωση στο αντίστοιχο αντικείμενο ενός αρχείου-script το οποίο θα περιλαμβάνει το ορισμό μιας κλάσης που κληρονομεί από την MonoBehaviour. Μόνο απόγονοι της κλάσης MonoBehaviour είναι διαχειρίσιμα από την Unity.

Στην εργασία γράφτηκαν scripts για τον χειρισμό ποικίλων αντικειμένων όπως των μαγικών λουλουδιών, των κανατών, της εναλλαγής όπλων παίκτη, τις διεπαφές με το χρήστη (GUI), καθώς και της γενικότερης διαχείρισης της κάθε πίστας και τον έλεγχο της ροής των αποστολών σε αυτή.

Κεφάλαιο 4 - Περιγραφή του παιχνιδιού

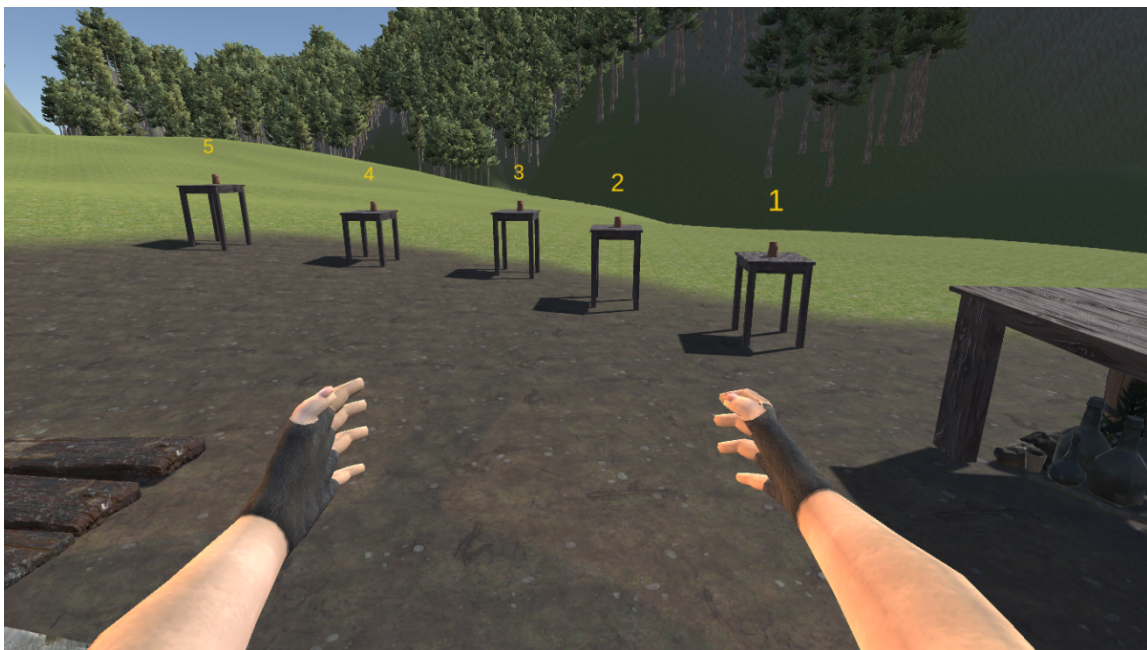
Η ιστορία στην οποία βασίζεται το παιχνίδι αφορά σε έναν χωρικό (παίκτης) που έχει μία καλύβα σε ένα μαγικό δάσος το οποίο απειλείται από μυθικά πλάσματα. Ο παίκτης πρέπει να προστατέψει την οικογένειά του, καθώς και τις καλλιέργειές μαγικών βοτάνων που έχει στον κήπο του, από αυτά τα μοχθηρά πλάσματα.



Εικόνα 10 Αποψη της καλύβας και του κήπου - σημείο εκκίνησης

Κατα την πορεία του παιχνιδιού καλείται να φέρει εις πέρας κάποιες αποστολές:

- Στην πρώτη πίστα, εν ειδει εξοικείωσης με το περιβάλλον του παιχνιδιού, του ζητείται να παρασκευάσει το πρώτο του μαγικό φίλτρο, ρίχνοντας ένα από τα μαγικά του λουλούδια σε ένα καζάνι συγκεκριμένης, προκαθορισμένης χωρητικότητας. Κάπου στην οθόνη του παιχνιδιού γίνεται γνωστό στον χρήστη η ποσότητα του φίλτρου που παρασκευάστηκε (με βάση την χωρητικότητα του καζανιού). Ενδέχεται αυτή η διαδικασία να επαναληφθεί κάμποσες φορές ώστε ο παίκτης να κατανοήσει την έννοια των διαφορετικών χωρητικοτήτων στα καζάνια (που έχει άμεση σχέση με την έννοια της διαφορετικής αξίας των ψηφίων στα θεσιακά συστήματα αρίθμησης).



Εικόνα 11 Οι μαγικές κανάτες τοποθετημένες κατά αύξουσα αξία θέσης

- Η δεύτερη πίστα έχει στόχο να εισάγει τον παίκτη στην λογική των “κρατούμενων” και της πεπερασμένης και συγκεκριμένης αξίας ενός ψηφίου σε κάθε θέση της αναπαράστασης του αριθμού. Για τον σκοπό αυτό, ο παίκτης προτρέπεται από τον εντολοδότη (bot του παιχνιδιού) να φτιάξει ένα φίλτρο ρίχνοντας σε κάποιο

καζάνι περισσότερα μαγικά άνθη από ό,τι χωράει το καζάνι (π.χ. στο 3ο καζάνι-των τετράδων να ρίξει 5 λουλούδια). Τότε, καθίσταται σαφές στον παίκτη ότι κάτι δεν πάει καλά (π.χ. το καζάνι κουνιέται ωσαν να πρόκειται να εκραγεί, ή τα λουλούδια πέφτουν κάτω από το καζάνι, ή βγαίνουν καπνοί από το μίγμα), οπότε ο χρήστης πρέπει να μετακινήσει το πιο πρόσφατο άνθος σε καζάνι μεγαλύτερης χωρητικότητας, ώσπου να μην υπάρχει καζάνι έτοιμο να ξεχειλίσει, οπότε και ολοκληρώνεται η πίστα. Πάλι, η πίστα μπορεί να επαναλλειφθεί κάμποσες φορές για διαφορετικούς αριθμούς.



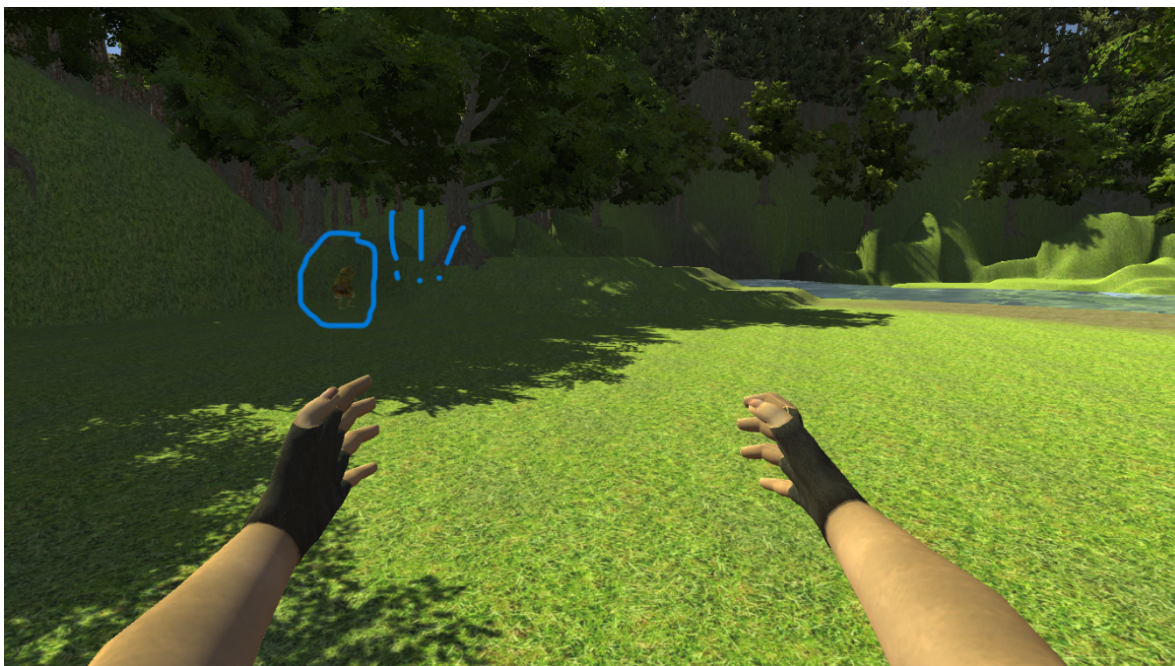
Εικόνα 12 Στο τραπέζι διακρίνεται το άνθος που μαραμένο έπεσε έξω από την κανάτα γιατί ξεπεράστηκε η χωρητικότητα



Εικόνα 13 Ο χρήστης συλλέγει τα μαγικά άνθη από τον κήπο του

- Από την τρίτη πίστα και εξής, οι αποστολές του παίκτη είναι να σκοτώσει συγκεκριμένο αριθμό τεράτων κάθε φορά. Βρίσκει τα τέρατα τυχαία, περιπλανώμενος στο δάσος και τα πλησιάζει για να μονομαχήσει ή τα τέρατα τον κυνηγούν και του επιτίθενται όταν εισέλθει στο οπτικό τους πεδίο.

Κάθε φορά που καταφέρνει ένα χτύπημα στον αντίπαλο, ένα “γραφικό” εμφανίζεται σύντομα πάνω απο τον αντίπαλο με πληροφορία για το πόση ζωή έχασε απο το χτύπημα του παίκτη, αλλά και το όπλο του παίκτη φθείρεται (πράγμα που αντανακλάται στην μείωση της αντοχής-stamina του). Εάν χτυπηθεί ο παίκτης, χάνει λίγη ζωή. Καθώς η stamina (αντοχή) μειώνεται, τα χτυπήματα του παίκτη είναι όλο και πιό αδύναμα, με τον εχθρό να χάνει λιγότερη ζωή. Όταν η αντοχή είναι σε κρίσιμα χαμηλά επίπεδα, ο παίκτης πρέπει να τρέξει πίσω στην καλύβα του, στον κήπο του και να παρασκευάσει κάποιο φίλτρο.



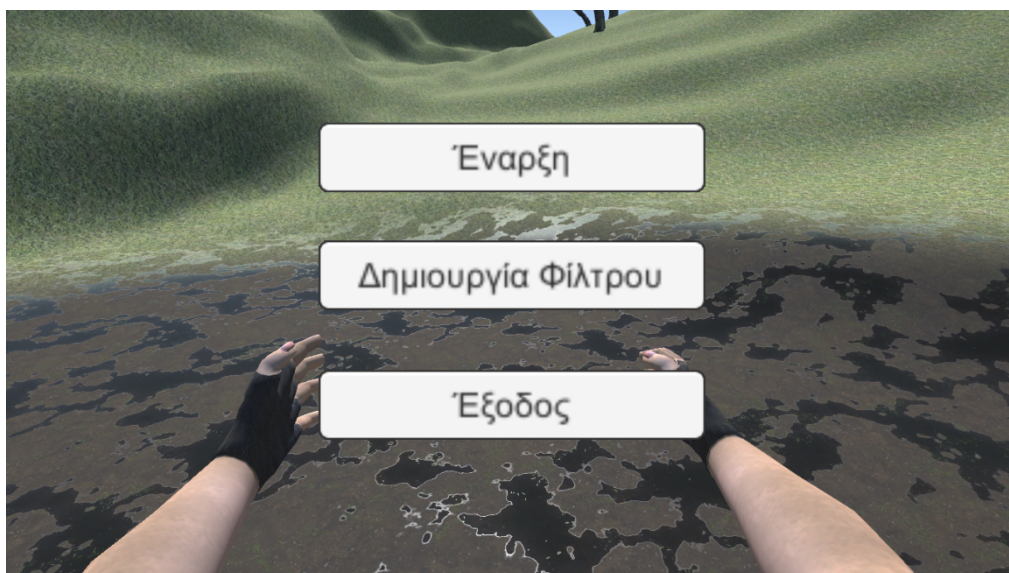
Εικόνα 14 Καθώς ο παίκτης περιπλανιέται στο δάσος συναντά τα εχθρικά ξωτικά

Όσο πιο δύσκολη είναι η συνταγή (αναπαράσταση αριθμού στο δυαδικό με πολλά ψηφία/ πρόσθεση στο δυαδικό με αναγκαστική συνένωση καζανιών κ.τ.λ.), τόσο πιο αποδοτικό είναι το φίλτρο. Δηλαδή τόσο πιο πολύ ανεβαίνει η στάμινα. Ειδικά, ενδέχεται να χρειαστεί να φτιάξει αρκετά τέτοια φίλτρα. Η καλλιέργεια των λουλουδιών για το φίλτρο παρουσιάζεται στον παίκτη ως μίνι-παιχνίδι που σταματά την κανονική ροή του παιχνιδιού.



Εικόνα 15 Μονομαχία με τα εχθρικά bot όπου ο παίκτης είναι κρίσιμα τραυματισμένος

Αυτά τα λουλούδια είναι μαγικά. Αυτό σημαίνει ότι φυλάσσονται σε ειδικά δοχεία που τα βοηθούν να αναπτυχθούν με συγκεκριμένο τρόπο. Πιο συγκεκριμένα, μόνο ένα λουλούδι μπορεί να κρατηθεί σε ένα δοχείο. Αν τύχει να βρεθούν δύο λουλούδια στο ίδιο δοχείο, αρχίζουν να μαραίνονται, αλλά αν μετακινηθούν δύο σε ένα μεγαλύτερο δοχείο (αριστερό μέρος της παρακάτω εικόνας) ανθίζουν διπλάσια άνθη από όσα είχαν προηγουμένως. Έτσι, στο πρώτο δοχείο όλα τα λουλούδια έχουν ένα άνθος, στο δεύτερο δοχείο δύο άνθη, στο τρίτο δοχείο τέσσερα άνθη κ.λπ. Για να έχει ένα λουλούδι με συγκεκριμένο αριθμό ανθέων, ο παίκτης μπορεί να τοποθετήσει λουλούδια στο πρώτο δοχείο (μοναδιαία λουλούδια) και να τα μετακινήσει κατάλληλα ώστε να μην μαραθούν. Εναλλακτικά, ο παίκτης μπορεί να τοποθετήσει αμέσως τα άνθη στα κατάλληλα αγγεία. Αυτή η μέθοδος για την παραγωγή των μαγικών λουλουδιών υπακούει στους κανόνες αναπαράστασης ποσοτήτων στο δυαδικό σύστημα και διδάσκεται στον παίκτη βήμα προς βήμα, σε μια "λειτουργία εκμάθησης", στις πίστες 1 και 2 όπως περιγράφησαν παραπάνω.



Εικόνα 16 Η διεπαφή του χρήστη για την επιστροφή στον κήπο και παρασκευή φίλτρου όταν χρειαστεί

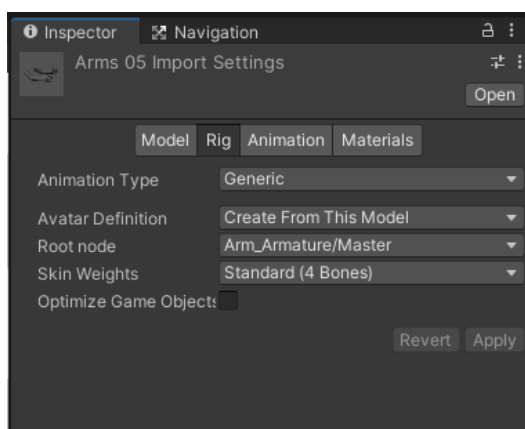
Κεφάλαιο 5 - Υλοποίηση

Στο κεφάλαιο αυτό αναλύεται η υλοποίηση της εφαρμογής. Πιο συγκεκριμένα, θα δημιουργηθεί μια εφαρμογή τύπου First Person Shooter η οποία θα παρουσιάζει ένα ήρωα ο οποίος θα πρέπει να αντιμετωπίσει τους εχθρούς του σε μια δοκιμασία τριών σταδίων. Στο αρχικό επίπεδο υπάρχει μόνο ένας εχθρός ο οποίος είναι άοπλος και συνεπώς είναι ευκολότερο για τον ήρωα να τον εξοντώσει. Στο 2ο στάδιο υπάρχουν τέσσερις εχθροί οι οποίοι φέρουν πλήρη εξάρτηση (όπλο και ασπίδα), συνεπώς είναι δυσκολότερο για τον ήρωα να τους αντιμετωπίσει. Τέλος, στο 3ο στάδιο υπάρχει ένας εχθρός ο οποίος είναι αρκετά ισχυρός και για να μπορέσει να τον αντιμετωπίσει ο ήρωας θα πρέπει να έχει αποκτήσει πρόσβαση σε όπλα. Τα όπλα μπορεί να τα αποκτήσει εφόσον ολοκληρώσει την παραγωγή των φίλτρων μέσω της ειδικής πίστας που είναι κατασκευασμένη για το σκοπό αυτό. Πιο συγκεκριμένα, στη πίστα αυτή, ο παίχτης θα πρέπει να αποδείξει τις γνώσεις του στο δυαδικό σύστημα τοποθετώντας κάποια άνθη μέσα σε κανάτες. Με τον τρόπο αυτό παράγει τα απαραίτητα φίλτρα που του δίνουν πρόσβαση στα όπλα. Όσα περισσότερα φίλτρα παράξει τόσο περισσότερα όπλα αποκτά και συνεπώς μπορεί να αντιμετωπίσει τους εχθρούς του ευκολότερα.

5.1. First Person Character

5.1.1. Εισαγωγή του μοντέλου

Το μοντέλο του ήρωα που θα χρησιμοποιηθεί προέρχεται από τον eafg [30] και για την εισαγωγή του εισάγεται στη Unity το αρχείο με κατάληξη fbx. Το αρχείο που περιέχει τα χρώματα των αντικειμένων είναι το ArmsDiffuse.jpg. Τα αρχεία μπορούν να εισαχθούν στην περιοχή Project της Unity (Εικόνα 17), σε ένα φάκελο με drag and drop. Ακολουθώντας, εισάγονται οι κατάλληλες παράμετροι για το animation. Με τον τρόπο αυτό η Unity μπορεί να αναπαράγει τα animations.

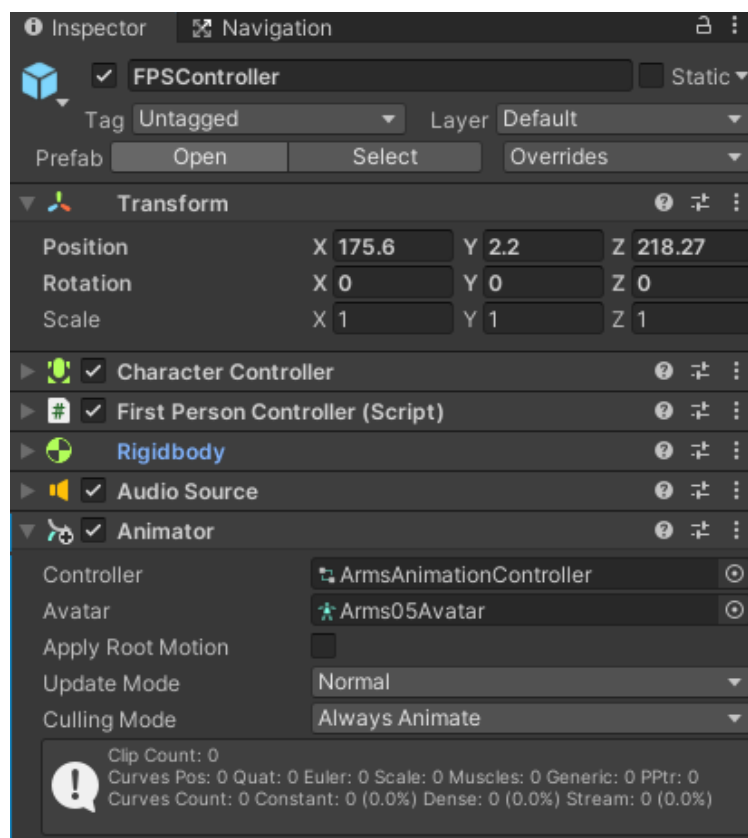


Εικόνα 17 Εισαγωγή μοντέλου χαρακτήρα.

5.1.2. Δημιουργία animation

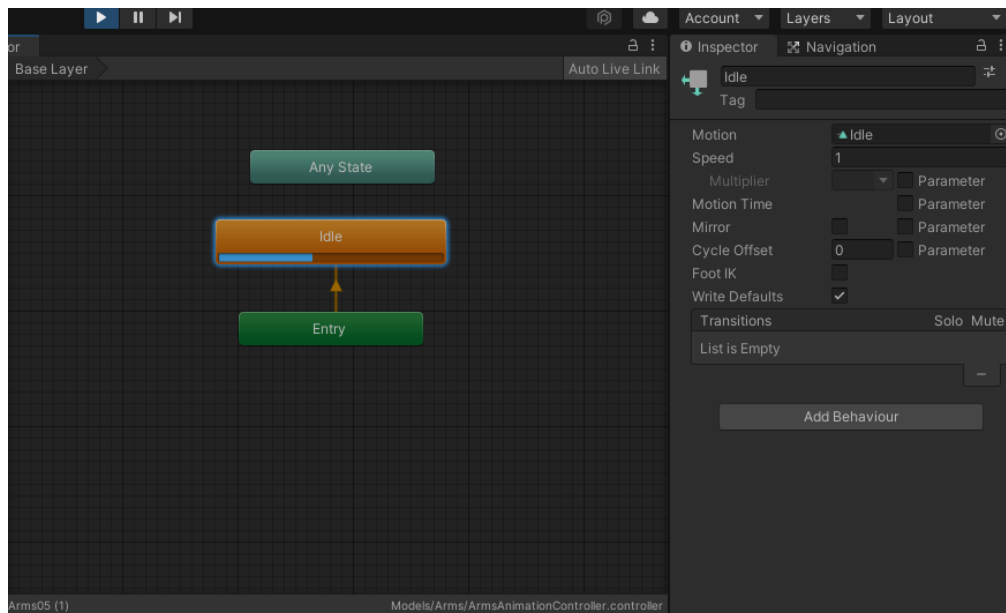
Για να μπορεί να χρησιμοποιήσει το μοντέλο τα διάφορα animations που το συνοδεύουν θα πρέπει να δημιουργηθεί ένας μηχανισμός που να αποφασίζει ποιο animation θα εκτελείται την κάθε στιγμή. Ο μηχανισμός αυτός καλείται «Animation Controller» και παρέχεται από την μηχανή της Unity. Η βάση για να αποφασιστεί ποιο animation θα εκτελεστεί κάθε φορά είναι η τιμή κάποιων μεταβλητών που ορίζονται από τον προγραμματιστή. Στην συνέχεια, αναλύεται η διαδικασία χρήσης του ως άνω ελεγκτή.

Η διαδικασία αυτή ξεκινά με δεξί κλικ στο φάκελο όπου βρίσκεται το **Prefab** του μοντέλου και ακολούθως επιλογή από το αναδυόμενο μενού **New Animation Controller**. Ακολούθως επιλογή του μοντέλου FPSController από την καρτέλα **Hierarchy** και αντιστοίχιση του **Animation Controller** στην ιδιότητα **Controller** της καρτέλας **Animator** του μοντέλου (Εικόνα 18).



Εικόνα 18 Αντιστοίχιση του Animator Controller στο μοντέλο.

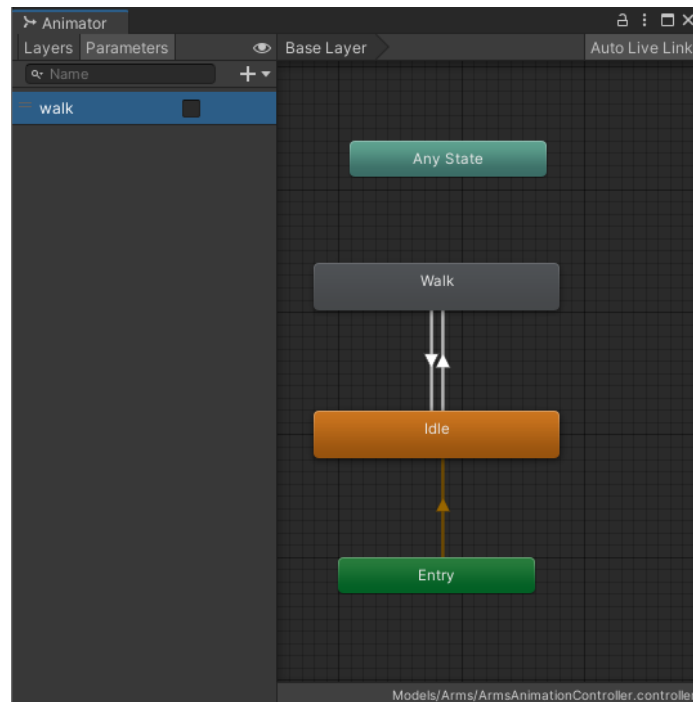
Ακολουθώντας με διπλό κλικ ο Animator Controller που μόλις δημιουργήθηκε ανοίγει για επεξεργασία (Εικόνα 19). Σε αυτόν θα δημιουργηθεί μια νέα κατάσταση η οποία θα ονομάζεται idle (αδρανής) και θα εκτελεί το ομώνυμο animation. Συνεπώς, με το που ξεκινά η εφαρμογή, ο animation controller θα εκτελεί το idle animation και ο χαρακτήρας θα παραμένει αδρανής, σε μία ουδέτερη στάση, αναμένοντας εντολή από τον χρήστη.



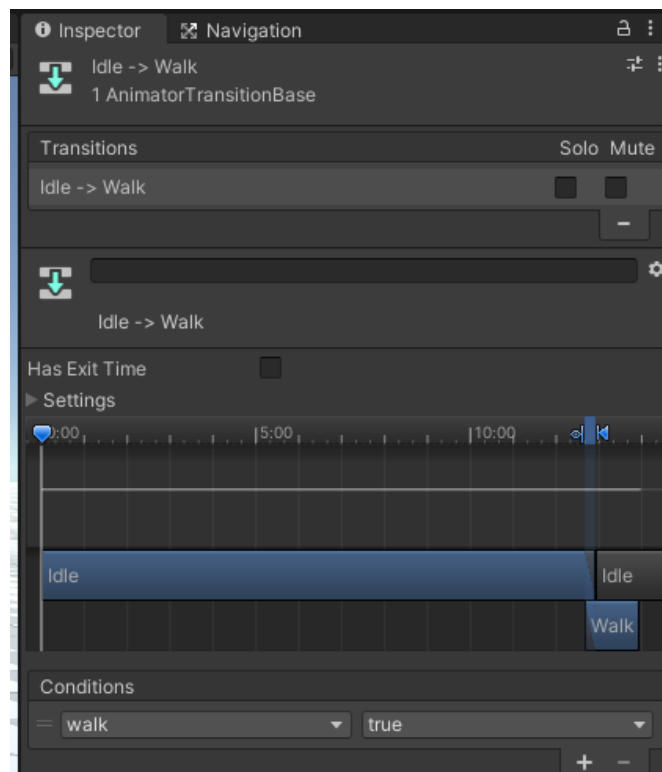
Εικόνα 19 Ο animator controller με τη κατάσταση idle.

5.1.2.1. Animation Βαδίσματος

Ο χαρακτήρας όμως δεν θα στέκεται απλώς, αλλά θα μπορεί να εκτελεί και άλλες κινήσεις. Συγκεκριμένα, για το βάδισμα θα δημιουργηθεί μια νέα κατάσταση, η walk. Όταν θα ενεργοποιείται η κατάσταση walk, θα εκτελείται το αντίστοιχο animation, Walk (Εικόνα 20). Η μετάβαση μεταξύ των καταστάσεων αδράνειας και βαδίσματος θα καθορίζεται από μια μεταβλητή τύπου Boolean με το όνομα walk (Εικόνα 21). Όταν ο χρήστης πατάει το πλήκτρο 'W' ή το 'S' η μεταβλητή θα ενεργοποιείται και θα τίθεται σε ισχύ η κατάσταση Walk η οποία θα ενεργοποιεί το ομώνυμο animation. Όταν το πλήκτρο απελευθερώνεται, η μεταβλητή θα γίνεται false και θα τίθεται σε ισχύ η κατάσταση Idle. Ταυτόχρονα, απενεργοποιείται η επιλογή 'Has Exit Time' ώστε το animation να μπορεί να διακοπεί.



Εικόνα 20 Ο Animation Controller με τη νέα κατάσταση και τις συνθήκες μετάβασης.



Εικόνα 21 Συνθήκη μετάβασης απο την Idle στη Walk.

Για την ενημέρωση της μεταβλητής walk θα δημιουργηθεί ένα Script (Εικόνα 22) το οποίο θα ανατεθεί στο αντικείμενο FPSController και η αποστολή του θα είναι να λαμβάνει την είσοδο από το πληκτρολόγιο και ακολούθως να ενημερώνει τη μεταβλητή walk του animator. Συνεπώς, επιλέγεται ο FPSController και ακολούθως από τη καρτέλα **Inspector** επιλέγεται το **Add Component** → **New Script**. Εντός της κλάσης δημιουργείται μια μεταβλητή τύπου Animator η οποία θα φιλοξενήσει τον **Animator** και εντός της μεθόδου Start καλείται η GetComponent με όρισμα το **Animator** (ώστε να επιστρέφει τον **Animator** που έχει αντιστοιχηθεί στο αντικείμενο). Ακολούθως, στην μέθοδο update (που εκτελείται σε κάθε κύκλο λειτουργίας -frame- του παιχνιδιού) ελέγχεται αν έχει πατηθεί το κατάλληλο πλήκτρο και ενημερώνεται η μεταβλητή του animator.

```
public class FirstPersonKeys : MonoBehaviour
{
    Animator anim;

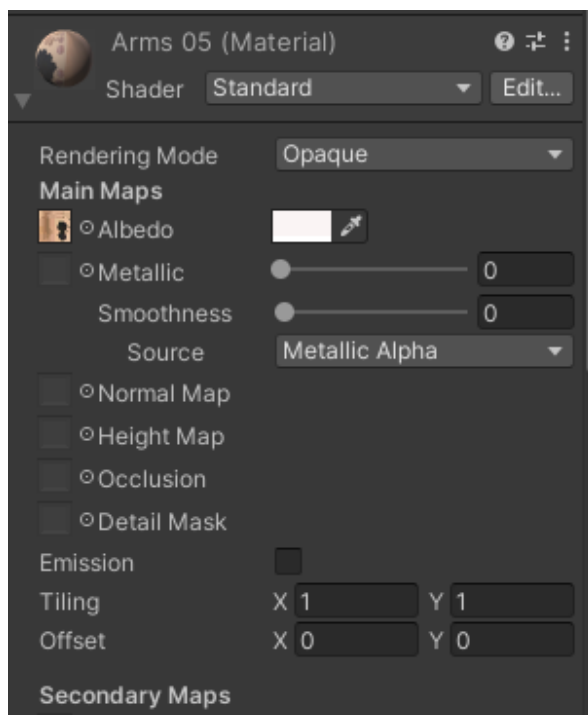
    // Start is called before the first frame update
    void Start()
    {
        anim = GetComponent<Animator>();
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetKey(KeyCode.W))
            anim.SetBool("walk", true);
        else if(Input.GetKey(KeyCode.S))
            anim.SetBool("walk", true);
        else
            anim.SetBool("walk", false);
    }
}
```

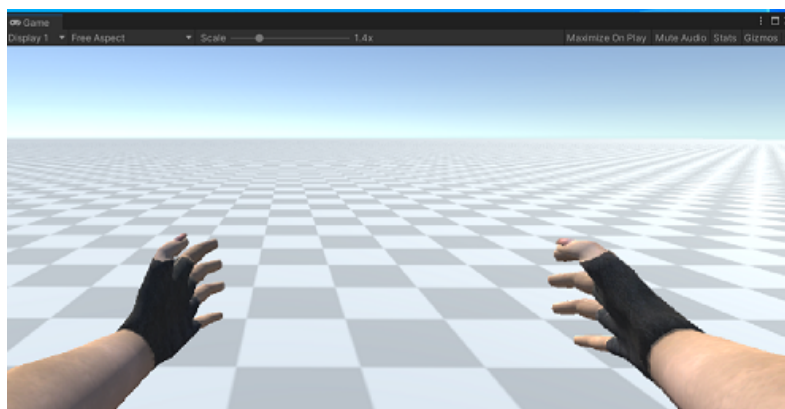
Εικόνα 22 Το Script FirstPersonKeys.

5.1.3. Χρωματισμός του μοντέλου

Το μοντέλο συνοδεύεται με μία εικόνα η οποία περιέχει τα υλικά (γάντια, δέρμα κλπ) τα οποία θα χρησιμοποιηθούν για να χρωματίσουν το μοντέλο (Εικόνα 23). Η εικόνα θα πρέπει να εισαχθεί στο αντικείμενο *Arm_Mesh*, στην ιδιότητα **Materials** → **Main Maps** → **Albedo** (Εικόνα 23). Στη περίπτωση που η ιδιότητα είναι απενεργοποιημένη, θα πρέπει να δημιουργηθεί μια νέα κάνοντας δεξί κλικ στη περιοχή και επιλέγοντας από το αναδυόμενο μενού την επιλογή **Create Material Preset**.



Εικόνα 23 Χρωματισμός μοντέλου.

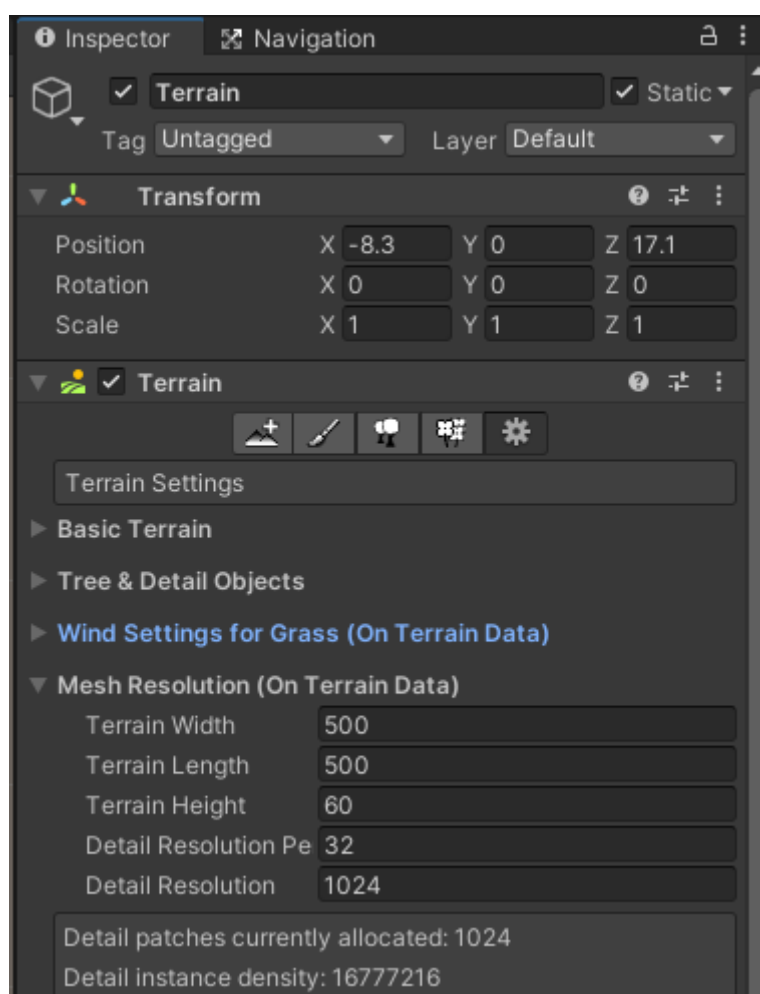


Εικόνα 24 Το μοντέλο στη τελική του μορφή.

5.2. Δημιουργία της αρχικής πίστας

5.2.1. Δημιουργία Terrain

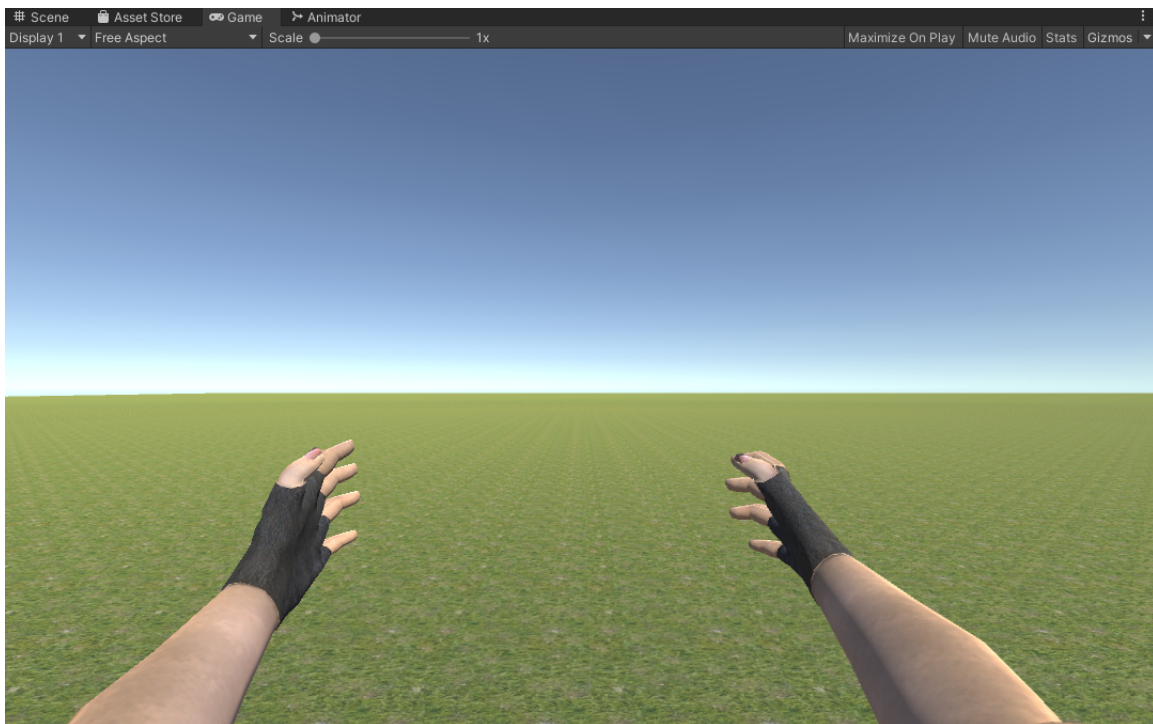
Το έδαφος πάνω στο οποίο θα δημιουργηθεί η πίστα θα αποτελείται από το αντικείμενο Terrain. Το αντικείμενο εισάγεται από το μενού **GameObject** → **3D Object** και ακολούθως επιλογή του **Terrain**. Αυτό τοποθετείται στο κέντρο του χώρου και αρχικά θα πρέπει να οριστούν οι κατάλληλες διαστάσεις από την επιλογή **Mesh Resolution** της καρτέλας **Terrain Settings** (Εικόνα 25).



Εικόνα 25 Ορισμός διαστάσεων Terrain.

5.2.2. Χρωματισμός Terrain

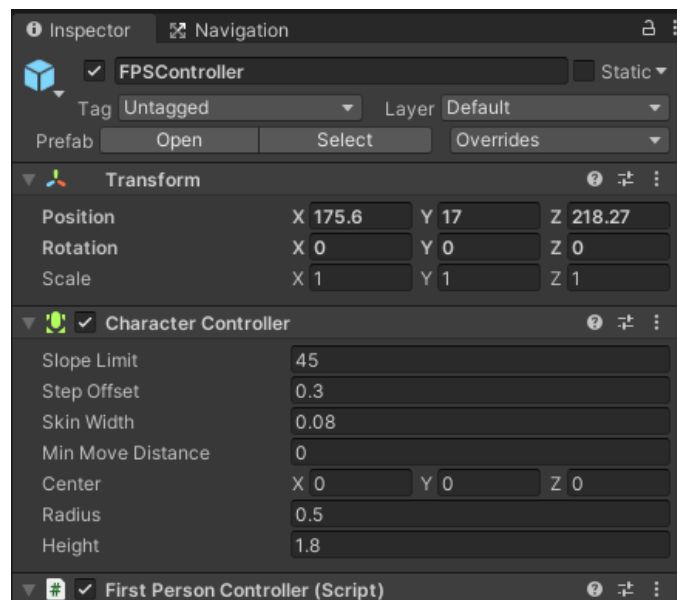
Το αντικείμενο εδάφους (Terrain) θα χρωματιστεί με ένα υλικό που θα παραπέμπει σε γρασίδι, το οποίο και θα αποτελέσει το βασικό υλικό (Εικόνα 26). Έπειτα, πάνω σε αυτό το υλικό θα προστεθούν τα υπόλοιπα αντικείμενα, όπως δένδρα, όρη κ.α. Η διαδικασία χρωματισμού του αντικειμένου του εδάφους επιτυγχάνεται επιλέγοντας την κατάλληλη εικόνα με μια αναζήτηση στο web του τύπου ‘free seamless ground textures’. Η αναζητούμενη εικόνα απαιτούμε να είναι χωρίς κοψίματα στις άκρες (seamless) ώστε να ταιριάζει ακριβώς με τις παρακείμενες εικόνες του αυτού αρχείου και έτσι να μην δίνεται στον χρήστη η εντύπωση ενός κατακερματισμένου, ψεύτικου δαπέδου. Στην συνέχεια, η εικόνα θα πρέπει να εισαχθεί στο φάκελο **Materials** της Unity είτε με drag and drop, είτε με απευθείας αποθήκευση στο φάκελο του **Project**. Για την εφαρμογή της εικόνας στο **Terrain** επιλέγεται το κουμπί **Paint Terrain** → **Paint Texture** από το Drop Down και έπειτα **Edit Terrain Layers** → **Create Layer**. Από το αναδυόμενο μενού επιλέγεται το κατάλληλο υλικό και ορίζονται οι κατάλληλες διαστάσεις του. Η Unity αναλαμβάνει να ‘τυλίξει’ το **Terrain** με το υλικό που επιλέχθηκε.



Εικόνα 26 Χρωματισμός του Terrain.

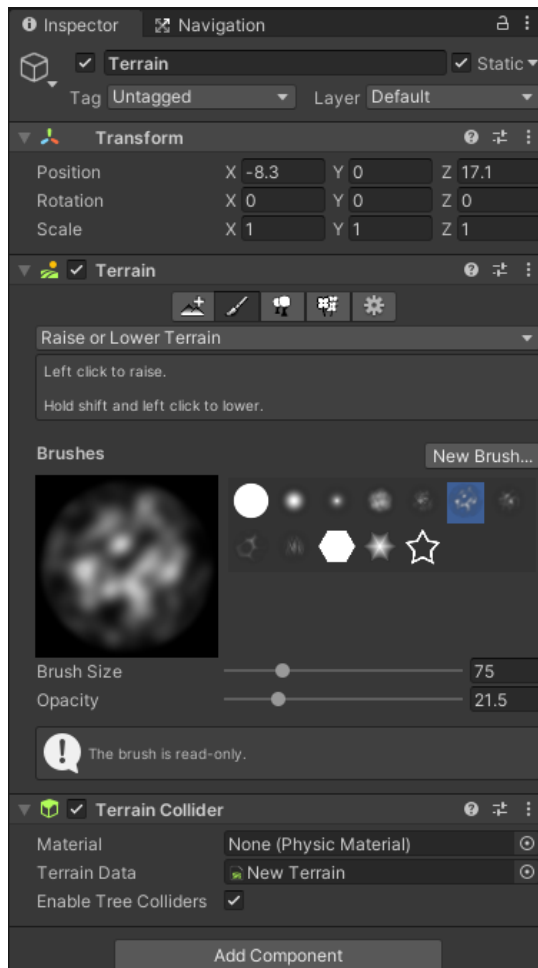
5.2.3. Μορφολογία εδάφους

Η πίστα θα περιβάλλεται από ψηλά βουνά τόσο για τη δημιουργία της αίσθησης αληθοφάνειας αλλά και για τον περιορισμό του χρήστη εντός του χώρου του παιχνιδιού. Εναλλακτικά, για τον περιορισμό του χρήστη μπορούν να εισαχθούν και αόρατοι κύβοι στα άκρα της πίστας οι οποίοι θα εμποδίζουν τον χρήστη από το να συνεχίσει να κινείται εκτός των ορίων της πίστας. Η προσέγγιση που ακολουθήθηκε ήταν αυτή της περικλείσης της πίστας με απότομα βουνά, ώστε να καθίσταται αδύνατη η επίβαση του παίκτη σε αυτά. Πιο συγκεκριμένα, η κλίση που ο ήρωας μπορεί να περπατήσει ορίζεται στη μεταβλητή **Slope Limit** (Εικόνα 27) και αυτό καθορίζει ποιās κλίσης επιφάνειες είναι βατές από τον χαρακτήρα.

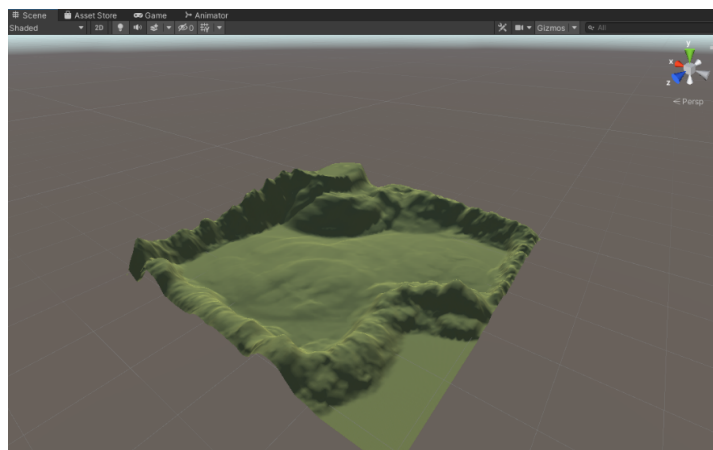


Εικόνα 27 Παράμετροι Character Controller.

Συνεχίζουμε εξετάζοντας την διαδικασία με την οποία μπορεί να δημιουργηθεί στην μηχανή το περιβάλλον της πίστας και δη το έδαφος. Για τη διαμόρφωση της μορφολογίας του εδάφους ο σχεδιαστής επιλέγει το κουμπί **Paint Terrain** και από το Drop Down List επιλέγει το **Raise or Lower Terrain**. Ακολούθως, επιλέγει την κατάλληλη βούρτσα (Εικόνα 28), το μέγεθος της βούρτσας αλλά και την διαφάνεια της. Έπειτα, κάνοντας κλικ πάνω στο αντικείμενο **Terrain**, το έδαφος ανυψώνεται έως το όριο που έχει οριστεί σαν ύψος του **Terrain**. Αντίθετα, για το χαμήλωμα του εδάφους θα πρέπει μαζί με το αριστερό κλικ να κρατείται πατημένο και το πλήκτρο Shift. Η σχεδίαση συνεχίζεται ώσπου να επιτευχθεί το επιθυμητό αποτέλεσμα (Εικόνα 29).



Εικόνα 28 Επιλογή κατάλληλης βούρτσας για τον ορισμό της μορφολογίας.

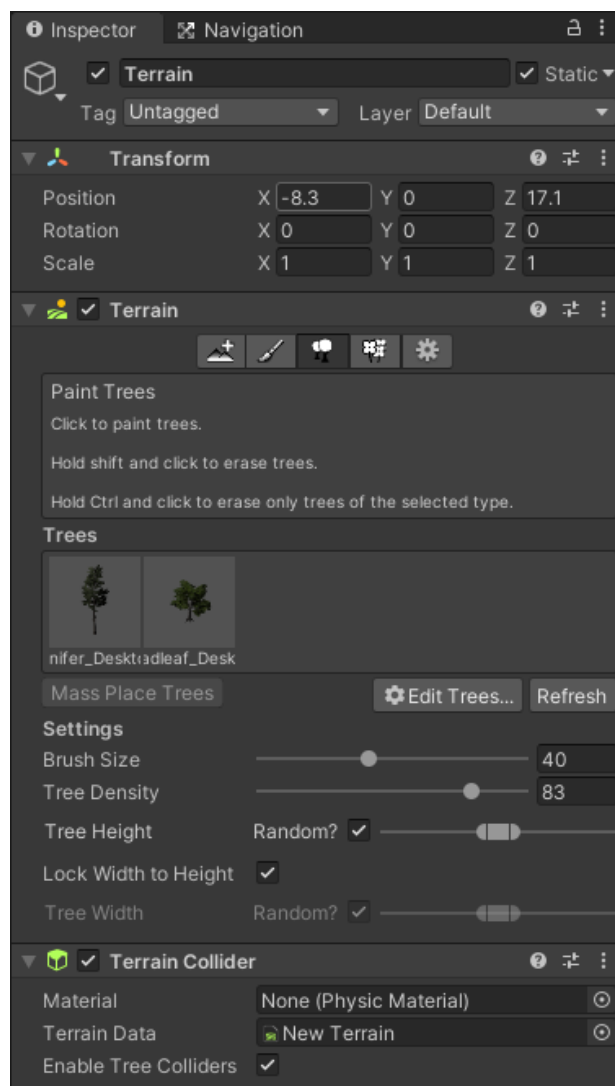


Εικόνα 29 Ολοκλήρωση της μορφολογίας.

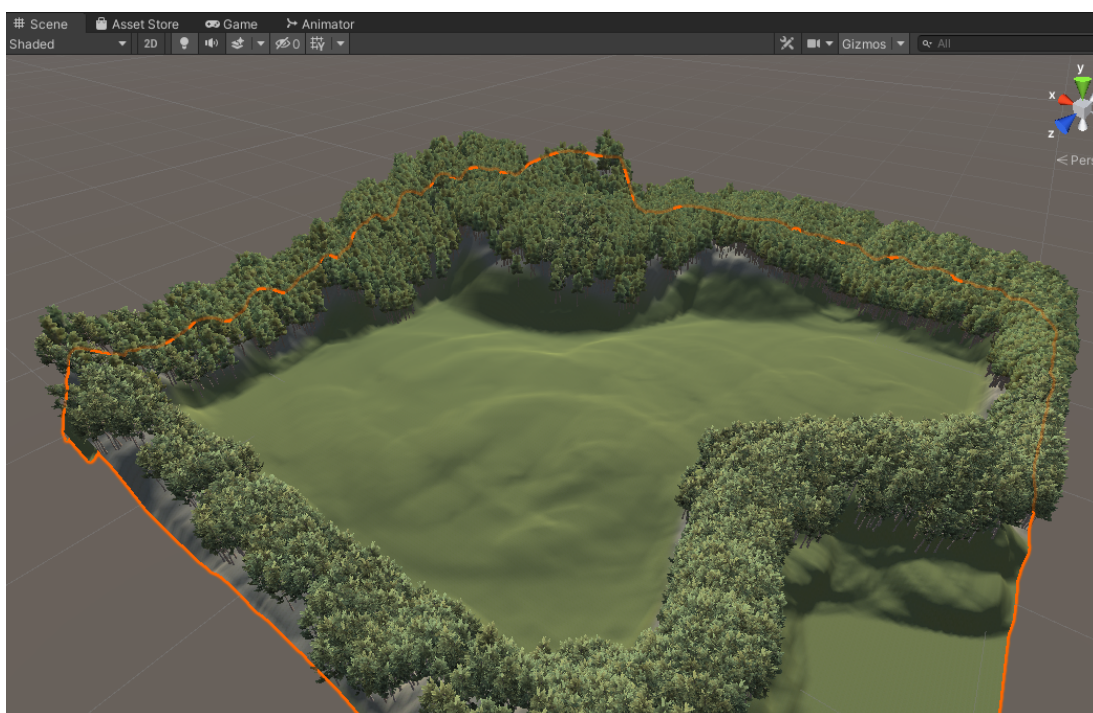
5.2.4. Διαμόρφωση περιβάλλοντος

Για την βελτίωση της αληθοφάνειας θα προστεθούν διάφορα αντικείμενα στη πίστα. Συγκεκριμένα τα βουνά θα μετατραπούν σε πυκνό δάσος εισάγοντας δέντρα. Τα δέντρα εισάγονται στο **Terrain** από την επιλογή **Paint Trees** (Εικόνα 30). Με το κουμπί **Edit Trees** επιλέγεται το είδος των δέντρων. Το πακέτο Standard Assets που παρέχεται με την εγκατάσταση της μηχανής της Unity περιέχει κωνοφόρα και πλατύφυλλα. Για το δάσος θα χρησιμοποιηθούν τα κωνοφόρα (Εικόνα 31).

Παράλληλα θα δημιουργηθεί το σπίτι του ήρωα (Εικόνα 32) το οποίο προέρχεται από το πακέτο Astrofish Games [31] ενώ τα λουλούδια του κήπου είναι από το chlyang [32].



Εικόνα 30 Εισαγωγή Δέντρων.



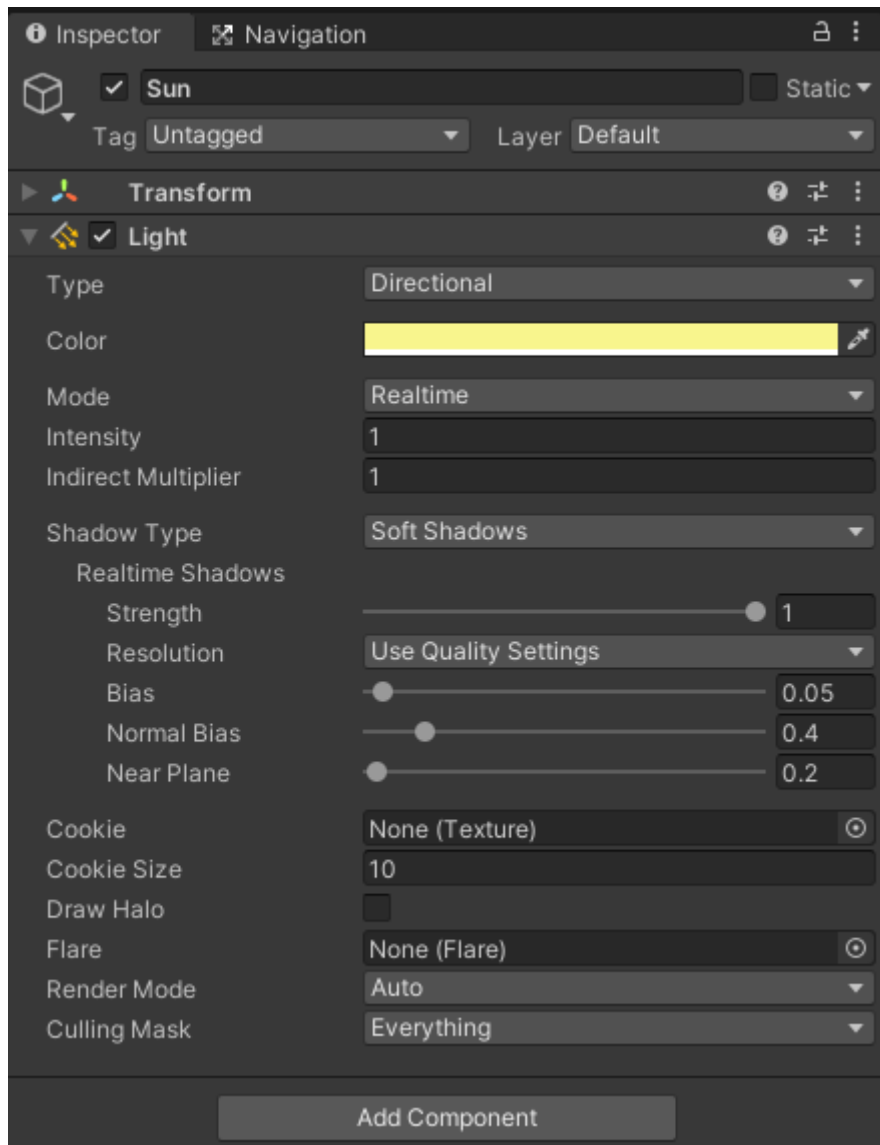
Εικόνα 31 Το Terrain με τα δέντρα.



Εικόνα 32 Το σπίτι με τον κήπο του ήρωα.

5.2.5. Φωτισμός της σκηνής

Για το φωτισμό της σκηνής εισάγεται ένα αντικείμενο Directional Light το οποίο θα διαδραματίζει το ρόλο του ήλιου. Το αντικείμενο δεν απαιτείται να τοποθετηθεί σε συγκεκριμένη θέση καθώς αυτή δεν έχει σημασία. Μόνο η περιστροφή του έχει αντίκτυπο στη σκηνή καθώς καθορίζει τη σκίαση. Τέλος, θα πρέπει να εισαχθεί το είδος της σκίασης από μία λίστα παρεχόμενων επιλογών (Εικόνα 33).

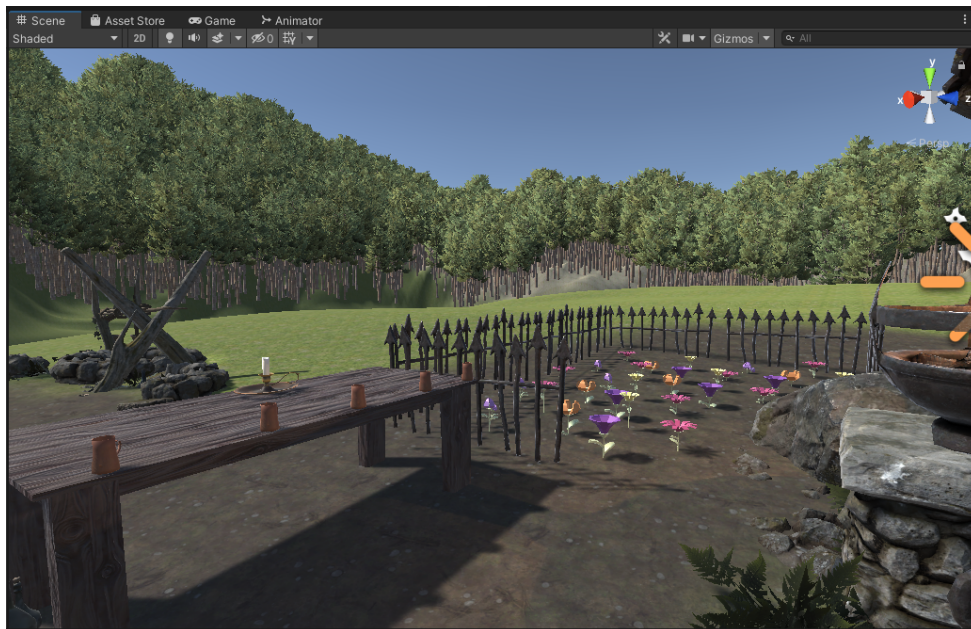


Εικόνα 33 Φωτισμός σκηνής.

5.3. Δημιουργία φίλτρου

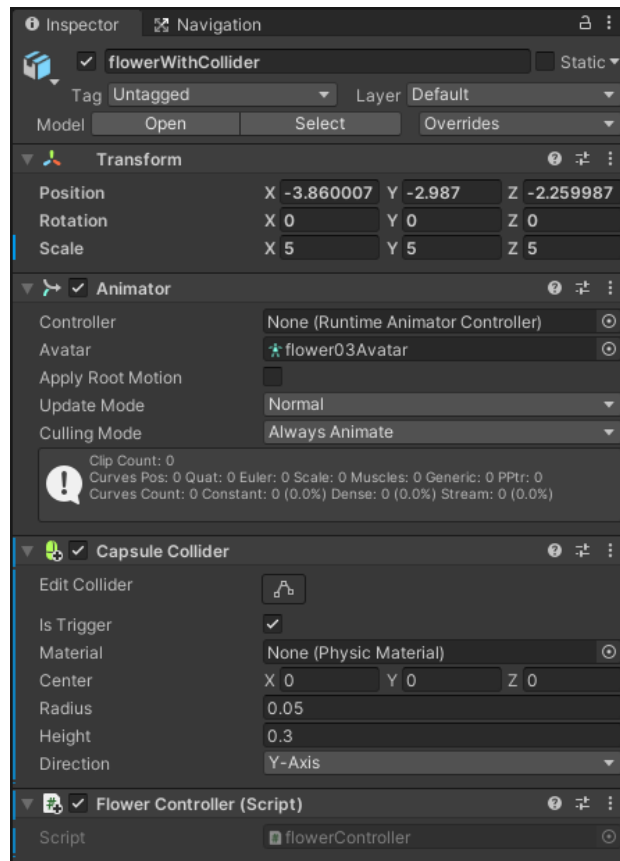
5.3.1. Συλλογή βοτάνων

Η παρασκευή του μαγικού φίλτρου γίνεται με τη συλλογή των βοτάνων που βρίσκονται στον κήπο του ήρωα (Εικόνα 34). Πιο συγκεκριμένα, στο κήπο του ήρωα καλλιεργούνται τα βότανα τα οποία θα αποτελέσουν το βασικό συστατικό του φίλτρου του. Τα βότανα αναπαρίστανται ως λουλούδια τα οποία ελήφθησαν από τον αντίστοιχο ιστότοπο της εταιρείας, Asset Store της Unity . Τα λουλούδια τοποθετούνται εντός των κανατών που βρίσκονται πάνω στα τραπέζια και παράγεται το φίλτρο. Οι κανάτες, το τραπέζι καθώς και τα υπόλοιπα κοντινά αντικείμενα προέρχονται από τη 3DLucas [34].



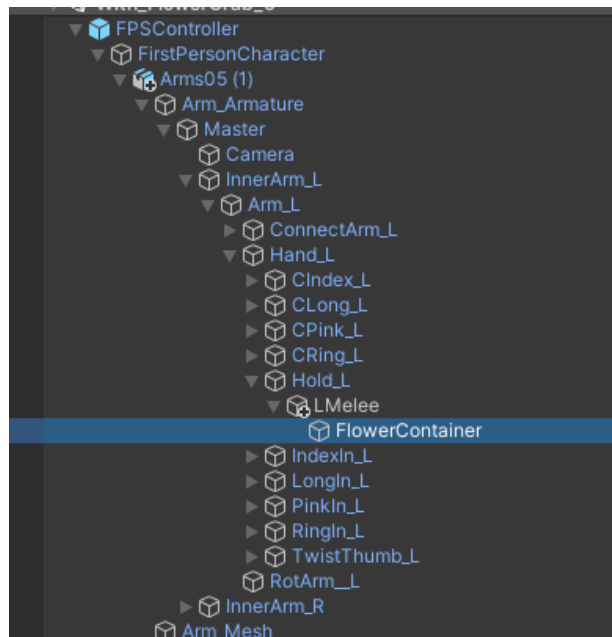
Εικόνα 34 Ο κήπος του ήρωα.

Το κάθε λουλούδι είναι εφοδιασμένο με ένα **Capsule Collider** (Εικόνα 35) στις διαστάσεις του αντικειμένου του λουλουδιού και η επιλογή **is Trigger** είναι ενεργοποιημένη ώστε να πυροδοτείται το αντίστοιχο γεγονός κάθε φορά που ο παίκτης αλληλεπιδρά (και συγκεκριμένα, έρχεται σε επαφή) με την γεωμετρία του αντικειμένου. Παράλληλα, κάθε αντικείμενο λουλουδιού διαθέτει και ένα “εκτελέσιμο,, αρχείο (script) το οποίο περιέχει τη συνάρτηση που θα κληθεί όταν ένα αντικείμενο εισέλθει εντός του αντικειμένου **Capsule Collider**.



Εικόνα 35 Ιδιότητες των λουλουδιών.

Μόλις ο χρήστης εισέλθει εντός των ορίων του **Capsule Collider** το λουλούδι θα μεταφέρεται στο αριστερό του χέρι και θα ακολουθεί τις κινήσεις του. Για να επιτευχθεί αυτό θα πρέπει το λουλούδι να αποκτήσει ως γονικό αντικείμενο το χέρι. Συνεπώς, δημιουργείται ένα αντικείμενο κενό με όνομα **Melee** το οποίο θα περιέχει ένα περιέκτη για το κάθε λουλούδι (Εικόνα 36). Ακολούθως το κάθε λουλούδι από το script (Εικόνα 37) μέσω της `localPosition` θα μεταφέρεται στη μηδενική θέση σε σχέση με το γονικό αντικείμενο ($x = 0$, $y = 0$ και $z = 0$). Δηλαδή, ακριβώς όπου βρίσκεται και το χέρι.



Εικόνα 36 Ο container των λουλουδιών.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class flowerController : MonoBehaviour
6  {
7
8      // reference to objects
9      public Transform flowerContainer;
10
11      // Start is called before the first frame update
12      void Start()
13      {
14
15      }
16
17      // Update is called once per frame
18      void Update()
19      {
20
21      }
22
23      void OnTriggerEnter(Collider other)
24      {
25          transform.SetParent(flowerContainer);
26          transform.localPosition = Vector3.zero;
27          transform.localRotation = Quaternion.Euler(Vector3.zero);
28          transform.localScale = Vector3.one;
29
30      }
31
32  }
33

```

Εικόνα 37 Το script που μεταφέρει τα λουλούδια.

5.4. Επιλογή όπλου

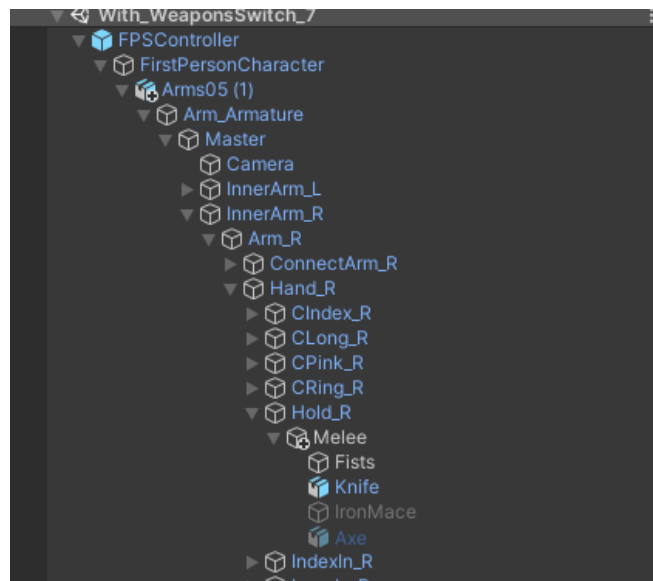
5.4.1. Εισαγωγή όπλων στη Unity

Τα μοντέλα που θα χρησιμοποιηθούν σαν όπλα περιέχονται στο ίδιο πακέτο με τα χέρια. Θα δημιουργηθεί ένας νέος φάκελος μέσα στο φάκελο Models ο οποίος θα περιέχει τα όπλα (Εικόνα 38). Για την εισαγωγή τους θα χρησιμοποιηθούν τα αρχεία με κατάληξη fbx για τα μοντέλα και τα png για το χρωματισμό τους.

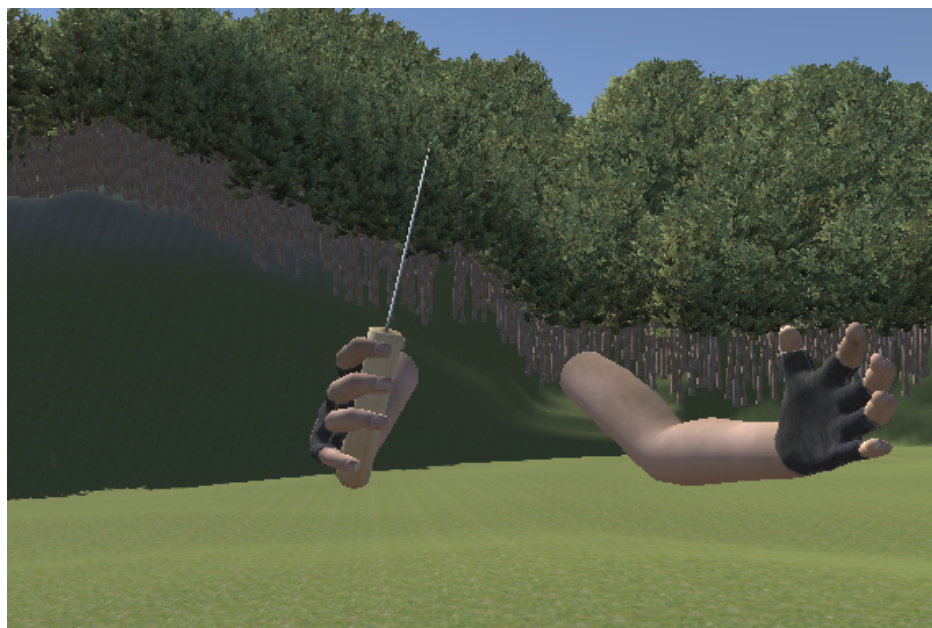


Εικόνα 38 Τα διαθέσιμα όπλα.

Πέρα από τα τρία όπλα που ο ήρωας έχει στη διάθεση του μπορεί να χρησιμοποιήσει και τα χέρια του ως γροθιές. Όλα τα όπλα, όπως και τα λουλούδια, για να ακολουθούν το animation θα εισαχθούν σε ένα άδειο αντικείμενο μέσα στο αντικείμενο του χεριού (Εικόνα 39). Στη σκηνή όλα έχουν τοποθετηθεί στη κατάλληλη θέση εντός της δεξιάς παλάμης (Εικόνα 40).



Εικόνα 39 Το σύνολο των όπλων μέσα στο αντικείμενο συλλογή.



Εικόνα 40 Όλα τα όπλα έχουν τοποθετηθεί στη κατάλληλη θέση.

5.4.2. Εναλλαγή των όπλων

Η εναλλαγή μεταξύ των όπλων γίνεται είτε με τη ρόδα του ποντικιού είτε με τη χρήση των αριθμών επι του πληκτρολογίου ενώ η σειρά αλληλουχίας τους είναι ανάλογη με το

μέγεθος της ζημιάς που το κάθε ένα μπορεί να προκαλέσει (Εικόνα 41). Πρώτα έχουν τοποθετηθεί οι γροθιές, ακολούθως το μαχαίρι, στη συνέχεια η σφαίρα και τέλος το τσεκούρι. Συνεπώς, αρχικά ο ήρωας θα εμφανίζεται με άδεια χέρια και θα μπορεί να χρησιμοποιήσει μόνο τις γροθιές του. Γυρνώντας την ρόδα του ποντικιού προς τα εμπρός θα επιλέγει το επόμενο όπλο στη σειρά, ενώ γυρνώντας τη προς τα πίσω θα επιλέγει το προηγούμενο. Όμοια και με τους αριθμούς του πληκτρολογίου, πατώντας το 1 επιλέγει τις γροθιές, το 2 το μαχαίρι κ.ο.κ ως το 4 που είναι το τσεκούρι.

Η υλοποίηση της λειτουργικότητας αυτής επιτυγχάνεται μέσω του script WeaponChange το οποίο έχει ανατεθεί στο αντικείμενο που περιέχει όλα τα όπλα. Το script, μετά την αρχικοποίηση μεταβλητών αρχικά λαμβάνει μία τιμή από την περιστροφή της ρόδας του ποντικιού και ανάλογα καθορίζει τη τιμή της μεταβλητής currWeapon η οποία έχει εύρος τιμών από 0 έως και 3. Με αντίστοιχο τρόπο μπορούν να ληφθούν και οι τιμές από το πληκτρολόγιο.

```
// Update is called once per frame
void Update()
{
    if (Input.GetAxis("Mouse ScrollWheel") > 0)
    {
        if (currWeapon + 1 <= maxWeapon)
            currWeapon++;
        else
            currWeapon = 0;
        selectWeapon(currWeapon);
    }
    else if (Input.GetAxis("Mouse ScrollWheel") < 0)
    {
        if (currWeapon - 1 >= 0)
            currWeapon--;
        else
            currWeapon = maxWeapon;
        selectWeapon(currWeapon);
    }

    if (currWeapon == maxWeapon + 1)
        currWeapon = 0;
    if (currWeapon == -1)
        currWeapon = maxWeapon;
}
```

Εικόνα 41 Λήψη τιμών από το χρήστη.

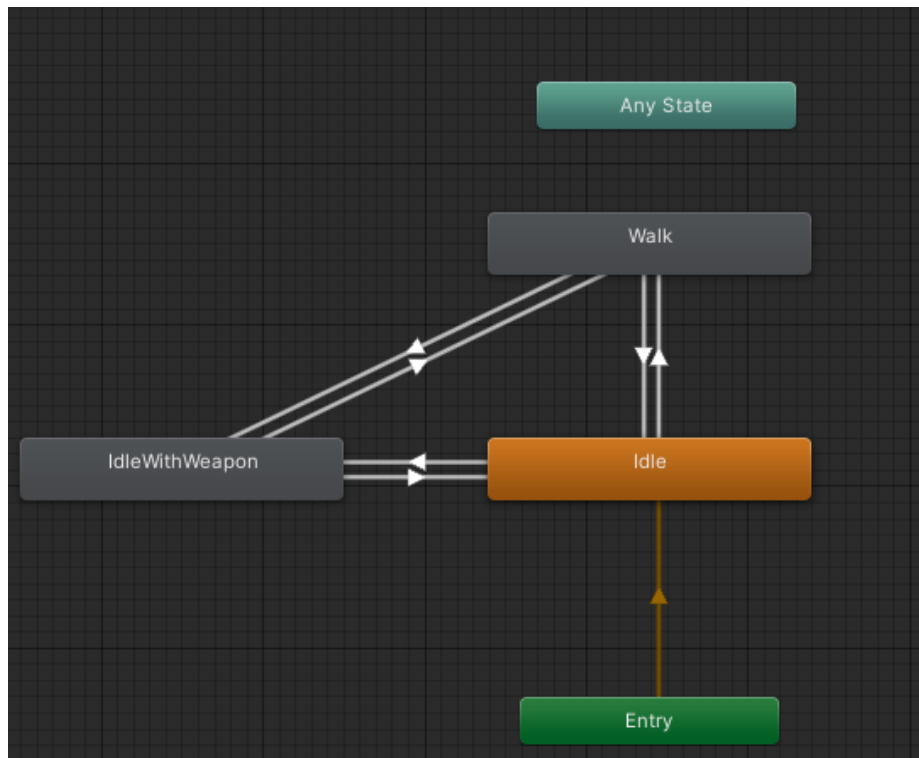
Η τιμή της μεταβλητής currWeapon περνάται ως όρισμα στην συνάρτηση selectWeapon η οποία ανακαλεί και επιστρέφει το αντικείμενο που βρίσκεται στην επιλεγθείσα από τον παίκτη θέση στο πίνακα με όλα τα αντικείμενα όπλων (Melee). Ποιο συγκεκριμένα, στον πίνακα αντικειμένων Melee, στη πρώτη θέση βρίσκονται οι γροθιές, στην δεύτερη το μαχαίρι κ.ο.κ. Συνεπώς, ένας βρόχος επανάληψης διαπερνάει όλες τις θέσεις του πίνακα με τα όπλα και αυτό που βρίσκεται στην ίδια θέση με τη μεταβλητή currWeapon γίνεται ενεργό μέσω της SetActive, ενώ τα υπόλοιπα απενεργοποιούνται.

Το μοντέλο των χεριών διαθέτει ένα animation με τα χέρια ανοικτά και ένα με κλειστά. Αυτό με τα ανοικτά χέρια θα χρησιμοποιηθεί όταν ο ήρωας δεν κρατά όπλο ενώ αυτό με κλειστά όταν κρατάει. Έτσι, αρχικά ο ήρωας θα εμφανίζεται με ανοικτά χέρια και θα μπορεί να χρησιμοποιήσει μόνο τις γροθιές του ενώ όταν παραλαμβάνει κάποιο όπλο θα ενεργοποιείται το animation που τον παρουσιάζει με τα χεριά κλειστά (Εικόνα 42).

```
void selectWeapon(int currWeapon)
{
    for (int i = 0; i < transform.childCount; i++)
    {
        if (i == currWeapon)
        {
            if (transform.GetChild(i).name == "Fists")
            {
                anim.SetBool("WeaponOn", false);
            }
            else
            {
                anim.SetBool("WeaponOn", true);
            }
            transform.GetChild(i).gameObject.SetActive(true);
        }
        else
        {
            transform.GetChild(i).gameObject.SetActive(false);
        }
    }
}
```

Εικόνα 42 Ενεργοποίηση του κατάλληλου όπλου και animation.

Για την υλοποίηση της λειτουργικότητας αυτής δημιουργείται μια νέα κατάσταση στον **Animator**, η *IdleWithWeapon*, η οποία θα εμφανίζει το ομώνυμο animation. Η επιλογή μεταξύ των καταστάσεων *Idle* και *IdleWithWeapon* επιτυγχάνεται μέσω μιας Boolean μεταβλητής *WeaponOn* την οποία το script θέτει σε *true* όταν το επιλεγμένο όπλο δεν έχει το όνομα *Fists* (Εικόνα 43).

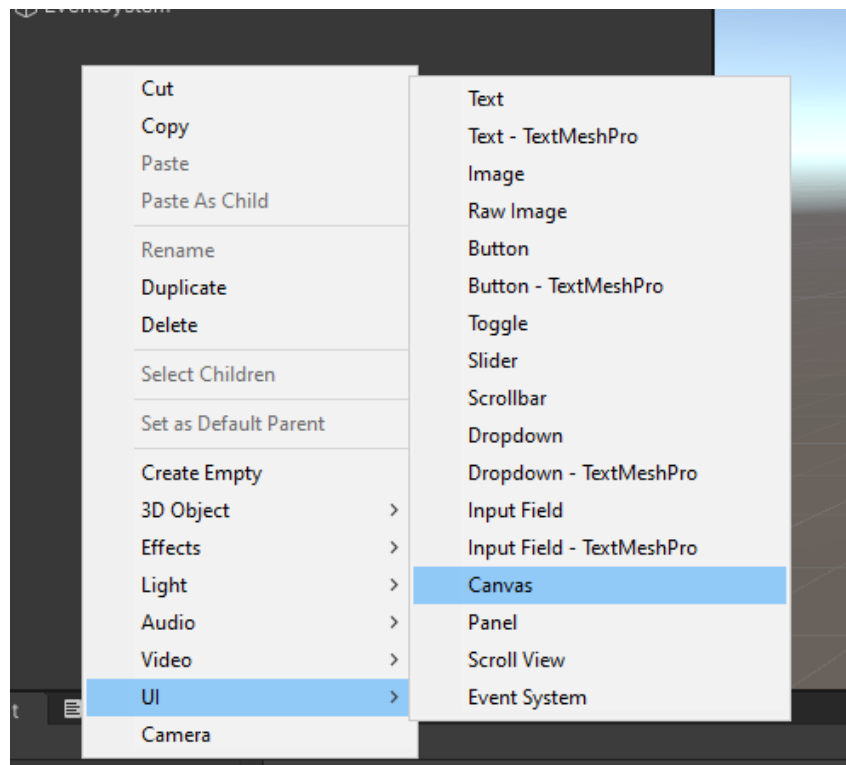


Εικόνα 43 Ο νέος Animator.

5.5. User Interface

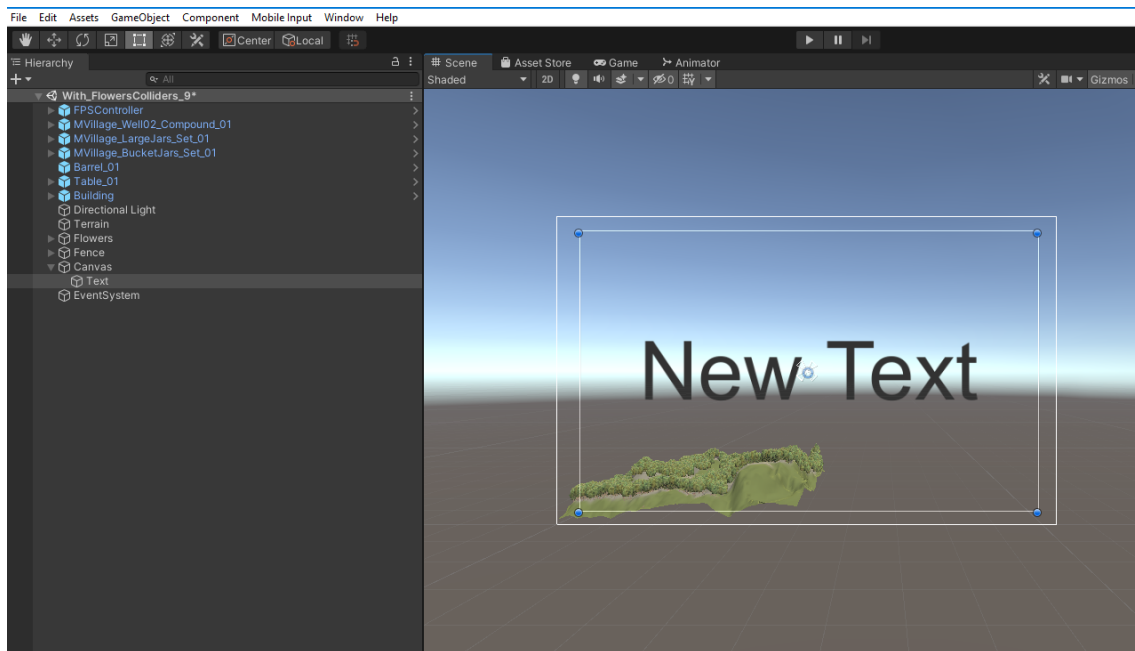
5.5.1. Δημιουργία Καμβά

Για την προβολή μηνυμάτων στο χρήστη θα χρησιμοποιηθεί το αντικείμενο **Canvas** στο οποίο θα προστεθούν διάφορα πεδία κειμένου, κουμπιά, κλπ. Το αντικείμενο **Canvas** δημιουργείται όπως και τα υπόλοιπα αντικείμενα: ένας τρόπος είναι επιλέγοντας δεξί κλικ στη καρτέλα **Hierarchy** και ακολούθως επιλογή του **UI → Canvas** (Εικόνα 44). Εναλλακτικά, ο χρήστης θα μπορούσε να επιλέξει και απευθείας αντικείμενο του καμβά (text, image, button) και το σύστημα μαζί με το αντικείμενο θα εισήγαγε και το καμβά πάνω στον οποίο θα εισαχθεί το αντικείμενο (Εικόνα 45).

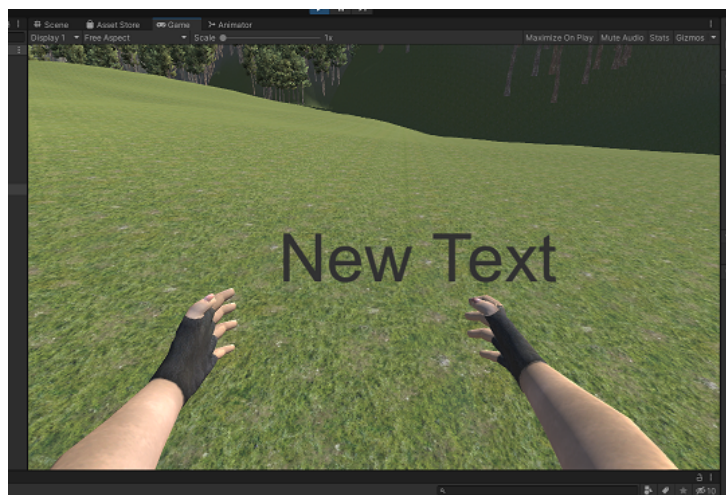


Εικόνα 44 Δημιουργία αντικειμένου Canvas.

Μετά την εισαγωγή του καμβά και του αντικειμένου που θα εμφανίζει (πχ ένα κείμενο) ο χρήστης θα πρέπει να μεγαλώσει το εύρος της σκηνής που είναι ορατό ώστε να μπορέσει να δει τον καμβά ο οποίος είναι αρκετά μεγαλύτερος από τη σκηνή.

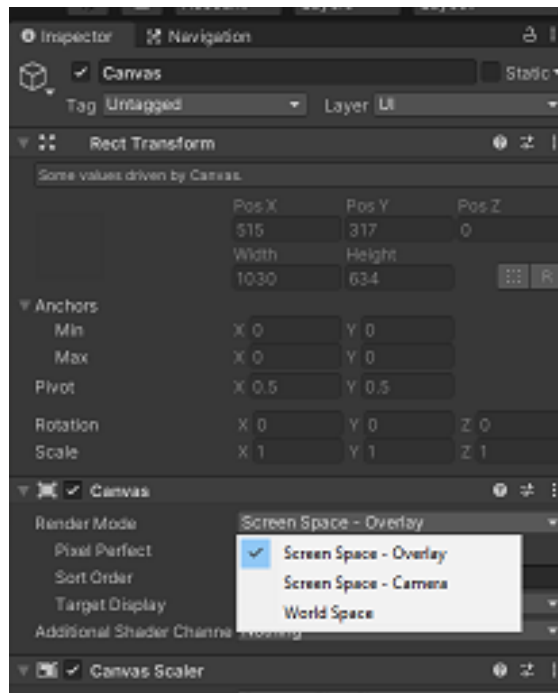


Εικόνα 45 Ο καμβάς με το αντικείμενο text στη σκηνή.



Εικόνα 46 Το αντικείμενο Canvas με το κείμενο κατά την εκτέλεση του παιχνιδιού.

Το αντικείμενο εμφανίζεται με τον τρόπο αυτό διότι ο τρόπος με τον οποίο η Unity το χειρίζεται είναι να το εμφανίζει η κάμερα μπροστά από οποιοδήποτε άλλο αντικείμενο (Εικόνα 46). Για το λόγο αυτό δεν έχει σημασία που έχει τοποθετηθεί στη σκηνή ή οι διαστάσεις του. Η Unity θα το εμφανίζει πάντα μπροστά από όλα τα αντικείμενα σαν να βρίσκεται μόνιμα μπροστά από τη κάμερα και την ακολουθεί σε κάθε της κίνηση.

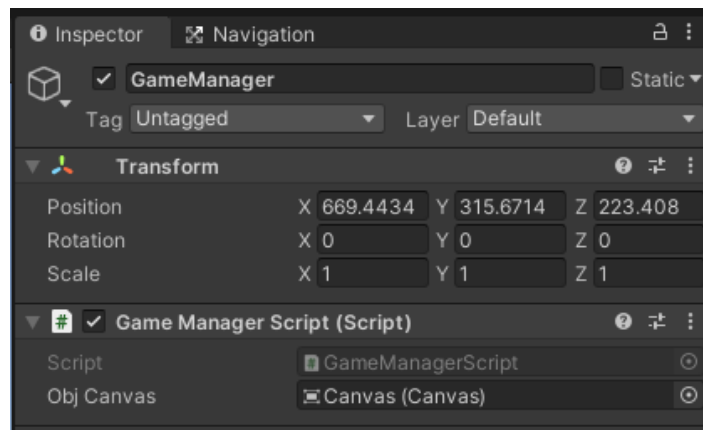


Εικόνα 47 Καθορισμός του τρόπου με τον οποίο θα εμφανίζεται ο Canvas.

Η επιλογή **Render Mode** καθορίζει τον τρόπο που θα εμφανίζεται ο καμβάς (Εικόνα 47). Με την πρώτη επιλογή **Screen Space – Overlay** επιτυγχάνεται η λειτουργικότητα που περιγράφηκε ενώ με την **Screen Space – Camera** επιλέγεται η κάμερα η οποία θα εμφανίζει τον **Canvas**. Με τον τρόπο αυτό είναι εφικτό να χρησιμοποιούνται πολλαπλές κάμερες καθώς και να διέρχονται αντικείμενα μεταξύ της κάμερας και του καμβά. Τέλος η τρίτη επιλογή **World Space** χειρίζεται τον καμβά σαν αντικείμενο του κόσμου (της σκηνής) και μπορεί να το τοποθετήσει όπου επιθυμεί, να του ορίσει διαστάσεις, κλπ.

5.5.2. Εμφάνιση - Απόκρυψη Καμβά

Ο καμβάς που εμφανίζει την αποστολή που θα πρέπει να εκτελέσει ο παίχτης θα εμφανίζεται και θα αποκρύπτεται με το πάτημα του πλήκτρου του πληκτρολογίου 'Ο' από το Αγγλικό Objective. Για το σκοπό αυτό θα δημιουργηθεί ένα κενό αντικείμενο το οποίο μεταξύ άλλων θα περιμένει να λάβει το πάτημα του κουμπιού και ανάλογα θα εκτελεί τη διαδικασία. Δημιουργείται λοιπόν ένα αντικείμενο με όνομα GameManager στο οποίο ανατίθεται ένα νέο Script GameManagerScript (Εικόνα 49). Το Script αυτό περιέχει μια μεταβλητή που αντιστοιχεί στο αντικείμενο καμβά, στην οποία το αντικείμενο Canvas θα πρέπει να αντιστοιχηθεί. Στη συνέχεια ορίζεται και μια Boolean μεταβλητή που αντιστοιχεί στο αν ο καμβάς είναι ορατός. Τέλος στη συνάρτηση Update ο κώδικας που εμφανίζει ή αποκρύπτει τον καμβά όταν πατηθεί το πλήκτρο O.



Εικόνα 48 Στο αντικείμενο GameManager, στην ιδιότητα ObjCanvas θα πρέπει να αντιστοιχηθεί το αντικείμενο.

```

4
5 public class GameManagerScript : MonoBehaviour
6 {
7
8     public Canvas objCanvas;
9     private bool objCamActive = false;
10
11     // Start is called before the first frame update
12     void Start()
13     {
14
15     }
16
17     // Update is called once per frame
18     void Update()
19     {
20         if (Input.GetKeyDown(KeyCode.O))
21         {
22             objCamActive = !objCamActive;
23             objCanvas.enabled = objCamActive;
24         }
25     }
26 }
27

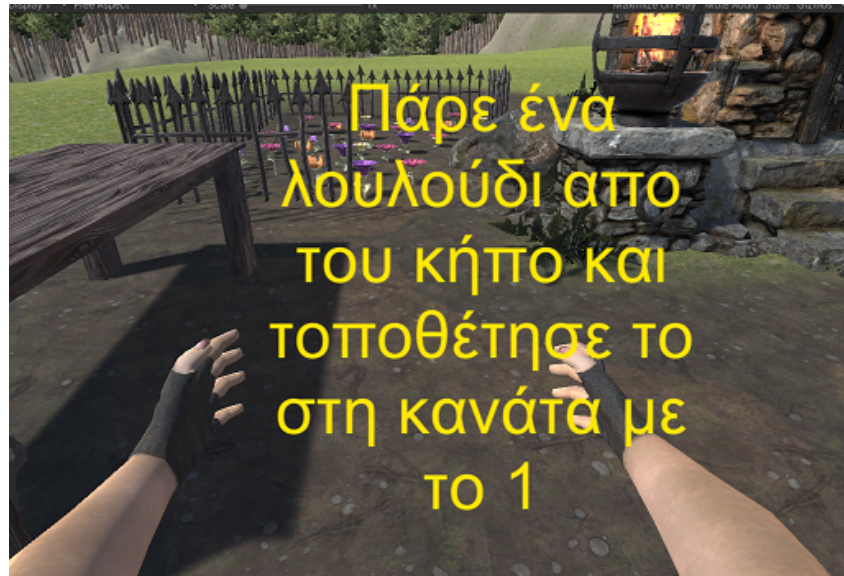
```

Εικόνα 49 Script για την εμφάνιση του καμβά.

Αντίστοιχα και για την εναλλαγή των μηνυμάτων που θα εμφανίζονται στην οθόνη το αντικείμενο GameManager θα περιέχει μια αναφορά στο αντικείμενο Text του καμβά (Εικόνα 48). Με τον τρόπο αυτό κάθε φορά που θα πρέπει ο καμβάς να εμφανίζει διαφορετικό μήνυμα αυτό θα αντιστοιχείται στη μεταβλητή text του αντικειμένου Text του καμβά (Εικόνα 50, Εικόνα 51).

```
objText.text = "Πάρε ένα λουλούδι απο του κήπο και τοποθέτησε το στη κανάτα με το 1";
```

Εικόνα 50 Εναλλαγή του κειμένου του Καμβά.



Εικόνα 51 Προβολή μηνυμάτων από τον Καμβά.

5.6. Αποστολές

Το παιχνίδι θα περιλαμβάνει διάφορες αποστολές μέσω των οποίων ο παίχτης θα πειραματίζεται με το δυαδικό σύστημα και σαν επιβράβευση θα λαμβάνει τα διάφορα εφόδια για να αντιμετωπίσει τους εχθρούς του. Πιο συγκεκριμένα, δίπλα στο σπίτι του ήρωα του παιχνιδιού υπάρχει ένας κήπος εντός του οποίου καλλιεργούνται λουλούδια. Παράλληλα στην αυλή του σπιτιού υπάρχουν πέντε τραπέζια όπου το κάθε ένα έχει πάνω του μία αριθμημένη κανάτα. Ο παίχτης θα μαζεύει τα λουλούδια από τον κήπο και θα τα τοποθετεί στις κανάτες σχηματίζοντας τους αριθμούς που η αποστολή ορίζει στο δυαδικό σύστημα. Για παράδειγμα για να σχηματίσει τον αριθμό 5 θα πρέπει να τοποθετήσει ένα λουλούδι στη κανάτα νούμερο ένα και άλλο ένα σε αυτή με το νούμερο τρία. Έχοντας τα λουλούδια στο χέρι και αν βρίσκεται πολύ κοντά σε μια κανάτα με το αριστερό κουμπί του ποντικιού ένα λουλούδι τοποθετείται στη κανάτα ενώ με το δεξί το λουλούδι της κανάτας επιστρέφει στο χέρι. Αν η κανάτα περιέχει ήδη λουλούδι δεν μπορεί να δεχθεί άλλο και συνεπώς το επόμενο λουλούδι που θα προσπαθήσει να τοποθετήσει εντός της κανάτας θα πέσει πάνω στο τραπέζι δίπλα στη κανάτα. Ακολούθως ο παίχτης μπορεί να το σηκώσει με το αριστερό κλικ του ποντικιού.

Η διαδικασία συλλογής των λουλουδιών από τον κήπο έχει περιγραφεί σε προηγούμενη παράγραφο. Αντίστοιχη διαδικασία θα εφαρμοστεί και για την τοποθέτηση των λουλουδιών στις κανάτες. Πιο συγκεκριμένα η κάθε κανάτα είναι εφοδιασμένη με ένα Collider ο οποίος ενεργοποιείται μόλις ο χρήστης εισέλθει στο

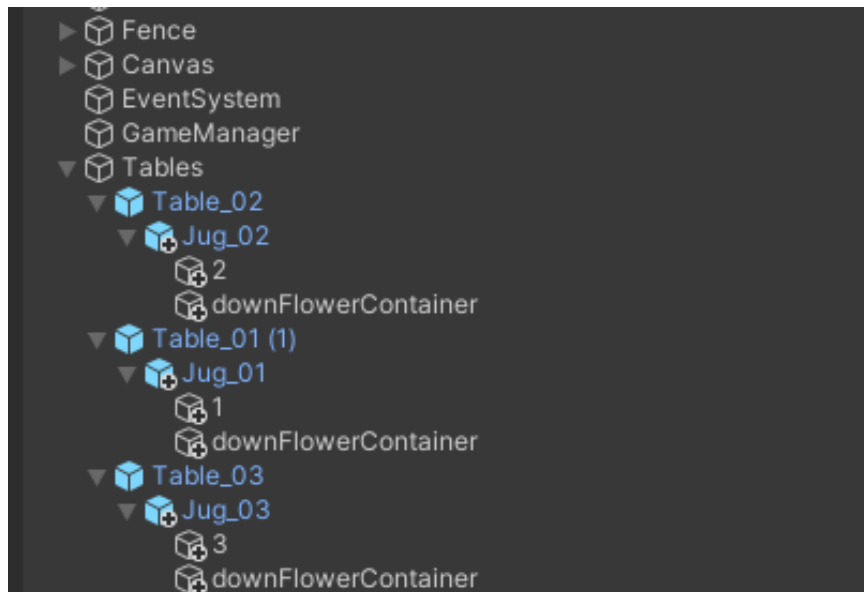
χώρο της. Με τον τρόπο αυτό μπορεί ο προγραμματιστής να γνωρίζει σε ποια από τις 5 κανάτες έχει πλησιάσει ο παίχτης προκειμένου να αποθέσει το λουλούδι. Ακολουθώντας η κάθε κανάτα περιέχει ένα Script με όνομα KettlerController το οποίο μεταξύ άλλων περιέχει και μια αναφορά στο περιέκτη λουλουδιών που βρίσκεται στο χέρι του χρήστη. Συνεπώς αν ο παίχτης βρίσκεται πολύ κοντά σε μία κανάτα και πατήσει το αριστερό πλήκτρο του ποντικιού ελέγχεται το κατά πόσον υπάρχει λουλούδι στο χέρι του. Αν υπάρχει τοποθετείται στην συγκεκριμένη κανάτα και ενεργοποιείται μια 'hasFlower' μεταβλητή ώστε να μπορεί εύκολα να αναγνωριστεί αν η κανάτα έχει λουλούδι ή όχι. Αντίστοιχα το δεξί κλικ μεταφέρει το λουλούδι στο χέρι.

Στη περίπτωση που η κανάτα ήδη διαθέτει λουλούδι τα νέα λουλούδια που θα προσπαθήσει να τοποθετήσει ο χρήστης στη κανάτα θα τοποθετούνται επί του τραπεζιού δίπλα στη κανάτα (Εικόνα 52). Για να υλοποιηθεί η λειτουργικότητα αυτή η κάθε κανάτα περιέχει ένα αντικείμενο που θα περιλάβει τα λουλούδια που θα τοποθετηθούν επί του τραπεζιού. Τα αντικείμενα αυτά βρίσκονται πάνω στο κάθε τραπέζι και δίπλα από την κανάτα. Έτσι κάθε φορά που πατιέται το αριστερό κλικ, ενώ ο χρήστης είναι κοντά σε μία κανάτα, ελέγχεται το κατά πόσο η κανάτα περιέχει λουλούδι. Αν έχει τότε το λουλούδι τοποθετείται εντός του EmptyObject που βρίσκεται δίπλα στη κανάτα και είναι παιδί της συγκεκριμένης κανάτας. Αντίστοιχα όταν πατιέται το δεξί κουμπί ελέγχεται πρώτα αν το EmptyObject περιέχει λουλούδι. Αν ναι τότε αυτό μεταφέρεται στο χέρι. Όταν αυτό δεν περιέχει πλέον άλλα λουλούδια τότε στο χέρι μεταφέρεται αυτό που υπάρχει εντός της κανάτας.



Εικόνα 52 Όταν η κανάτα περιέχει λουλούδι τα επόμενα τοποθετούνται επι του τραπεζιού.

5.6.1. Αντικείμενο Κανάτα



Εικόνα 53 Αντικείμενα τραπέζια με τις κανάτες.


```

void OnTriggerStay(Collider other)
{
    if (flowerContainer.transform.childCount > 0)
    {
        if (Input.GetMouseButtonDown(0)) {
            if (canPut)
            {
                if (!hasFlower)
                {
                    flowerTran = flowerContainer.transform.GetChild(0).transform;
                    Transform cetCotnain = transform;
                    flowerTran.SetParent(cetCotnain);
                    flowerTran.localPosition = new Vector3(0, 0.15f, 0);
                    hasFlower = true;
                    canPut = false;
                    StartCoroutine(PutDelay());
                }
                else
                {
                    flowerTran = flowerContainer.transform.GetChild(0).transform;
                    print(transform.GetChild(1).name);
                    flowerTran.SetParent(transform.GetChild(1).transform);
                    flowerTran.localPosition = new Vector3(0, 0, 0);
                    canPut = false;
                    StartCoroutine(PutDelay());
                }
            }
        }
    }
}

```

Εικόνα 54 Κώδικας αριστερού κλικ.

Το Script της κάθε κανάτας περιέχει μια αναφορά `flowerContainer` που αντιστοιχεί στο περιέκτη λουλουδιών που βρίσκεται στο χέρι του παίχτη (Εικόνα 53). Η συνάρτηση `childCount` του αντικειμένου `transform` του `flowerContainer` επιστρέφει τον αριθμό των αντικειμένων που βρίσκονται εντός του `flowerContainer`, δηλαδή τον αριθμό των λουλουδιών που βρίσκονται εντός του. Συνεπώς ο ακόλουθος κώδικας (Εικόνα 54) θα εκτελεστεί μόνο αν υπάρχουν λουλούδια στο χέρι του παίχτη. Η συνάρτηση `GetChild(0)` του αντικειμένου `transform` του περιέκτη επιστρέφει κάθε φορά που καλείται το πρώτο αντικείμενο λουλούδι που βρίσκεται εντός του περιέκτη. Από το αντικείμενο αυτό θα πρέπει να χρησιμοποιηθεί το αντικείμενο `transform` ώστε να τοποθετηθεί το λουλούδι στη νέα θέση. Οι θέσεις που μπορεί να τοποθετηθεί το λουλούδι με το αριστερό κλικ είναι δύο: ή εντός της κανάτας ή εντός του `EmptyObject` δίπλα από την κανάτα. Το κριτήριο είναι η μεταβλητή `hasFlower` η οποία αρχικά είναι `false`. Συνεπώς τη πρώτη φορά που θα πατηθεί το αριστερό κλικ θα εκτελεστεί το `if` της προηγούμενης εικόνας με στόχο το λουλούδι να τοποθετηθεί εντός της κανάτας. Μόλις ο κώδικας αποκτήσει πρόσβαση στο αντικείμενο `transform` του λουλουδιού που θα μεταφερθεί θα πρέπει να εντοπίσει το αντικείμενο `transform` της κανάτας που θα μπει το λουλούδι το οποίο είναι το τρέχον αντικείμενο `transform` (η τρέχουσα κανάτα). Συνεπώς για να αποκτήσει το λουλούδι πατρικό αντικείμενο την κανάτα αρκεί στη συνάρτηση `setParent` να δοθεί ως όρισμα το τρέχον `transform`. Τέλος για να μεταφερθεί το λουλούδι εντός της κανάτας (στο κέντρο της) και λίγο υψηλότερα για να είναι ορατό, αντιστοιχίζεται στη μεταβλητή `localPosition` ένα νέο διάνυσμα με τη τιμή που αντιστοιχεί στην άξονα `Y` ελαφρώς αυξημένη. Τέλος ενημερώνεται η μεταβλητή `hasFlower` ότι η κανάτα περιέχει λουλούδι

και οι επόμενες δύο γραμμές τοποθετούν μια καθυστέρηση στην ανάγνωση των κλικ διότι το σύστημα θα λαμβάνει πολλαπλά κλικ εντός της OnTriggerStay με αποτέλεσμα να μην λειτουργεί σωστά.

Στη περίπτωση που η κανάτα περιέχει λουλούδι η μεταβλητή hasFlower είναι true συνεπώς θα εκτελεστεί το else τμήμα του κώδικα της προηγούμενης εικόνας και το λουλούδι θα τοποθετηθεί εντός του EmptyObject δίπλα από την κανάτα. Συνεπώς το πατρικό του θα είναι το 2ο αντικείμενο που βρίσκεται εντός της κανάτας (το αντικείμενο με όνομα 'downFlowerContainer') καθώς το πρώτο είναι το αντικείμενο που εμφανίζει τον αριθμό πάνω από τη κάθε κανάτα.

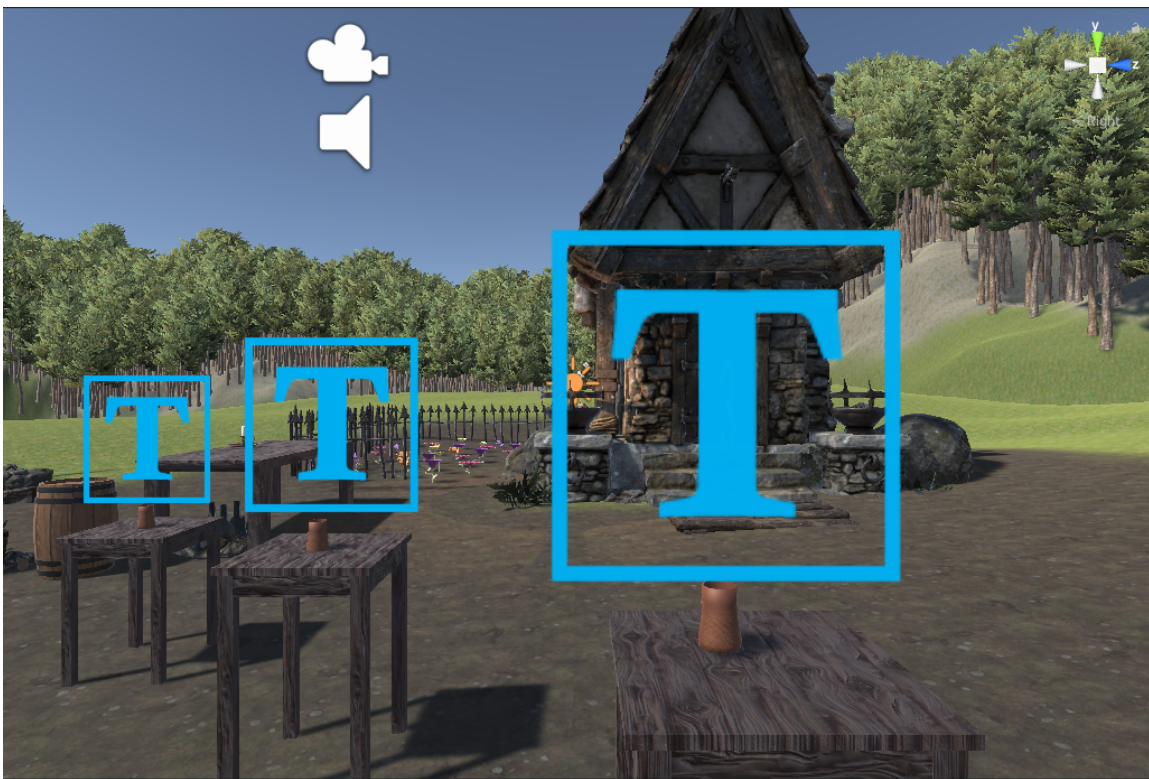
```
78 if (Input.GetMouseButtonDown(1))
79 {
80     if (canPut)
81     {
82         if (transform.GetChild(1).childCount > 0)
83         {
84             Transform flowerDown = transform.GetChild(1).GetChild(0).transform;
85             Transform handFlower = flowerContainer.transform;
86             flowerDown.SetParent(handFlower);
87             flowerDown.localPosition = new Vector3(0, 0, 0);
88             canPut = false;
89             StartCoroutine(PutDelay());
90         }
91     }
92     else
93     {
94         if (hasFlower)
95         {
96             Transform flowKett = transform.GetChild(2).transform;
97             flowKett.SetParent(flowerContainer.transform);
98             flowKett.localPosition = new Vector3(0, 0, 0);
99             hasFlower = false;
100             canPut = false;
101             StartCoroutine(PutDelay());
102         }
103     }
104 }
```

Εικόνα 55 Κώδικας για το δεξί κλικ.

Η αντίστοιχη διαδικασία ακολουθείται και στη περίπτωση του δεξιού κλικ (Εικόνα 55) με τη διαφορά ότι πλέον πρώτα μεταφέρονται τα λουλούδια από το EmptyObject του τραπεζιού στο χέρι αν υπάρχουν και αν όχι το λουλούδι από τη κανάτα. Συνεπώς αρχικά ελέγχεται ο αριθμός των αντικειμένων που περιέχονται εντός του 2ου αντικειμένου (με το όνομα downFlowerContainer) της κανάτας. Αν το αντικείμενο αυτό περιέχει αντικείμενα αυτά είναι λουλούδια και συνεπώς θα πρέπει να μεταφερθούν στο χέρι. Για το σκοπό αυτό λαμβάνεται το αντικείμενο transform του κάθε λουλουδιού και ορίζεται το πατρικό του ώστε να μην περιέχεται πλέον στο EmptyObject της κανάτας αλλά του χεριού. Ακολούθως ενημερώνεται η localPosition του λουλουδιού ώστε να

τοποθετηθεί στο κέντρο του χεριού. Στη περίπτωση που το EmptyObject του τραπέζιού δεν περιέχει λουλούδια τότε μεταφέρεται το λουλούδι που υπάρχει εντός της κανάτας και η μεταβλητή hasFlower αντιστοιχίζεται σε false.

Τέλος, για την εμφάνιση του αριθμού της κάθε κανάτας έχει χρησιμοποιηθεί το αντικείμενο **Text Mesh Pro** το οποίο έχει τοποθετηθεί ως απόγονος του αντικειμένου κανάτα. Στο χώρο έχει τοποθετηθεί πάνω από την κάθε κανάτα και για να μένει προσανατολισμένο στη κάμερα του έχει ανατεθεί ένα Script (Εικόνα 57) το οποίο στη συνάρτηση Update λαμβάνει τις τιμές rotation της κάμερας και τις αναθέτει στο αντικείμενο.



Εικόνα 56 Αντικείμενα Text Mesh για την εμφάνιση του αριθμού της κάθε κανάτας.

```

12
13 // Update is called once per frame
14 void Update()
15 {
16     transform.rotation = Camera.main.transform.rotation;
17 }
18
19 }
20

```

Εικόνα 57 Περιστροφή αντικειμένου Text Mesh Pro.

5.6.2. Αντικείμενο Camera – Χειρισμός οθόνης

Το αντικείμενο Camera.main είναι κατά κάποιο τρόπο κοινό και όλα τα αντικείμενα του παιχνιδιού έχουν πρόσβαση σε αυτό χωρίς να διαθέτουν άμεση αναφορά σε αυτό. Σε σχέση με αυτό το αντικείμενο παρουσιάζονται όλα τα δισδιάστατα γραφικά στην οθόνη.

5.6.3. Αντικείμενο GameManager – Χειρισμός Αποστολών

Το αντικείμενο αυτό θα χειρίζεται μεγάλο μέρος των λειτουργιών του παιχνιδιού. Για παράδειγμα το αντικείμενο θα περιέχει την μεταβλητή που θα περιέχει τις ζωές των παιχτών, θα την ενημερώνουν τα υπόλοιπα αντικείμενα και θα την ελέγχει ώστε αν εξαντληθεί κάποιου παίχτη να τον τερματίζει. Επίσης θα περιλαμβάνει και τη λειτουργία των αποστολών. Πιο συγκεκριμένα το παιχνίδι περιέχει για κάθε αποστολή δύο μεταβλητές όπου η πρώτη ενημερώνει ότι η συγκεκριμένη αποστολή είναι ενεργή και η δεύτερη ότι έχει ολοκληρωθεί ώστε να ξεκινήσει τη 2η. Το αν έχει ολοκληρωθεί η αποστολή ορίζεται από το ποιες κανάτες έχουν λουλούδι. Συνεπώς το Script περιέχει αναφορές για όλες τις κανάτες και στη συνάρτηση Update ελέγχει εάν η ενεργή αποστολή έχει ολοκληρωθεί.

Η ενεργοποίηση ή η απενεργοποίηση των αποστολών γίνεται από τη συνάρτηση setObjectiveText (Εικόνα 58) η οποία ανάλογα με το είδος των ορισμάτων που δέχεται

εισάγει το κείμενο της αποστολής στο Text του καμβά και ενεργοποιεί ή απενεργοποιεί αποστολές. Για παράδειγμα για να ενεργοποιηθεί η πρώτη αποστολή η συνάρτηση λαμβάνει το πρώτο όρισμα ως True και όλα τα υπόλοιπα False. Συνεπώς όταν η εφαρμογή ξεκινά στη συνάρτηση Start του GameManager εκτελείται η setObjectiveText με το πρώτο όρισμα True.

```
175 public void setObjectiveText(bool obj1, bool obj2, bool obj3, bool obj4, bool obj5)
176 {
177     if (obj1)
178     {
179         objText.text = "Πάρε ένα λουλούδι απο του κήπο και τοποθέτησε το στη κανάτα με το 1";
180         objec1Started = true;
181     }
182     if (obj2)
183     {
184         objText.text = "Αφαίρεσε το λουλούδι που τοποθέτησες στη πρώτη κανάτα.";
185         objec2Started = true;
186     }
187 }
188
```

Εικόνα 58 Η συνάρτηση setObjectiveText.

```
91 public void checkObjective()
92 {
93     if (objec1Started)
94     {
95         // 1 0 0 0 0
96         if(ketControlScr1.hasFlower & !ketControlScr2.hasFlower & !ketControlScr3.hasFlower & !ketControlScr4.hasFlower & !ketControlScr5.hasFlower)
97         {
98             objec1Complete = true;
99
100             StartCoroutine(MessageDisplay("Συγχαρητήρια, μόλις ολοκλήρωσες την πρώτη σου αποστολή.", "Ξεκίνα τη 2η..."));
101
102             setObjectiveText(false, true, false, false, false);
103
104             objec1Started = false;
105         }
106     }
107
108     if (objec2Started)
109     {
110         // 0 0 0 0 0
111         if (!ketControlScr1.hasFlower & !ketControlScr2.hasFlower & !ketControlScr3.hasFlower & !ketControlScr4.hasFlower & !ketControlScr5.hasFlower)
112         {
113             objec2Complete = true;
114
115             StartCoroutine(MessageDisplay("Συγχαρητήρια, μόλις ολοκλήρωσες την δεύτερη σου αποστολή.", "Ξεκίνα τη 3η..."));
116
117             setObjectiveText(false, false, true, false, false);
118
119             objec2Started = false;
120         }
121     }
122 }
123
124
```

Εικόνα 59 Έλεγχος περαίωσης ή όχι της τρέχουσας αποστολής.

Η περαίωση της κάθε αποστολής πραγματοποιείται από τη συνάρτηση checkObjective η οποία εκτελείται συνεχώς μέσω της Update (Εικόνα 59). Προκειμένου να αποκτήσει το

Script πρόσβαση στην μεταβλητή `hasFlower` του Script `CettlerController` του αντικειμένου κανάτα διαθέτει μια από μία αναφορά σε όλα τα αντικείμενα κανάτες και μέσω της `GetComponent<CettlerCotroller>()` αποκτά πρόσβαση στη κοινόχρηστη μεταβλητή `hasFlower` (Εικόνα 60).

```
49  
50 ketControlScr1 = kettler1.GetComponent<CettlerController>();  
51 ketControlScr2 = kettler2.GetComponent<CettlerController>();  
52 ketControlScr3 = kettler3.GetComponent<CettlerController>();  
53 ketControlScr4 = kettler4.GetComponent<CettlerController>();  
54 ketControlScr5 = kettler5.GetComponent<CettlerController>();  
55
```

Εικόνα 60 Πρόσβαση στη μεταβλητή του Script `CettlerController`.

5.6.4. Εμφάνιση μηνυμάτων στην οθόνη με συγκεκριμένη χρονική διάρκεια

Μόλις η εφαρμογή «αντιληφθεί» ότι η αποστολή έχει ολοκληρωθεί εμφανίζει στο χρήστη ενημερωτικό μήνυμα, σε ξεχωριστό αντικείμενο καμβά, το οποίο εξαφανίζεται μετά την πάροδο των 5 δευτερολέπτων. Η διαδικασία επιτυγχάνεται με τη χρήση της `StartCoroutine` (Εικόνα 61) η οποία λαμβάνει ως όρισμα τη συνάρτηση την οποία θα εκτελέσει σε διαφορετικό νήμα από αυτό της εφαρμογής.

Η συνάρτηση επιστρέφει αντικείμενο τύπου `IEnumerator` και στο σημείο επιστροφής (`yield return`) σταματά την εκτέλεση για το χρονικό διάστημα που λαμβάνει ως όρισμα η `WaitForSeconds`. Συνεπώς το τμήμα του κώδικα πριν την `return` θα εκτελεστεί άμεσα και παράλληλα με το κύριο νήμα της εφαρμογής, ενώ το τμήμα μετά θα εκτελεστεί μετά το πέρας του χρόνου.

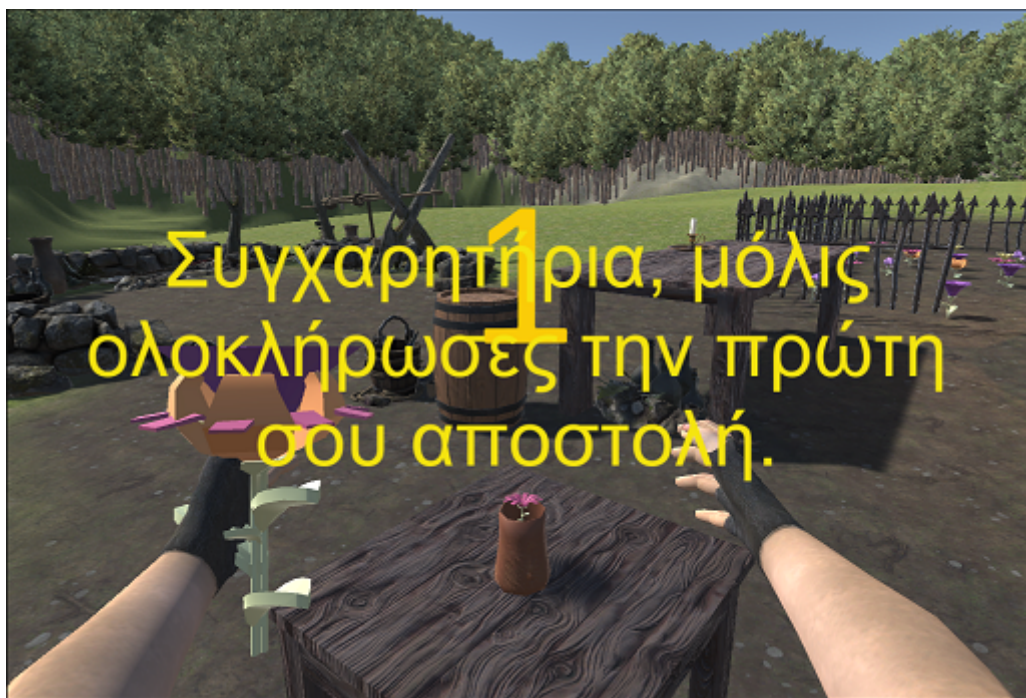
```

212 IEnumerator MessageDisplay( string msg1, string msg2)
213 {
214     objCanvas.enabled = false;
215     msgCanvas.enabled = true;
216     msgText.text = msg1;
217
218     yield return new WaitForSeconds(5);
219     if (msg2 != null)
220     {
221         msgText.text = msg2;
222         yield return new WaitForSeconds(5);
223     }
224     msgCanvas.enabled = false;
225
226 }

```

Εικόνα 61 Υπορουτίνα εμφάνισης των μηνυμάτων.

Η συνάρτηση λαμβάνει ως όρισμα τα δύο κείμενα που θα εμφανίσει. Συνεπώς η συνάρτηση αρχικά θα εξαφανίσει τον καμβά που δείχνει την αποστολή (στη περίπτωση που αυτός είναι ενεργός ώστε να μην μπλέκονται τα κείμενα) και ακολούθως θα εμφανίζει τον καμβά που εμφανίζει τα μηνύματα ενημερώνοντας κατάλληλα το αντικείμενο Text με το κείμενο που θα εμφανίσει (Εικόνα 62). Ακολούθως θα περιμένει 5 δευτερόλεπτα και στη συνέχεια θα εμφανίσει το 2ο μήνυμα που λαμβάνει ως όρισμα, αν αυτό έχει αρχικοποιηθεί (Εικόνα 63). Στη συνέχεια θα αποκρύψει τον καμβά και η λειτουργία της ολοκληρώνεται.



Εικόνα 62 Μήνυμα ενημέρωσης του χρήστη για την ολοκλήρωση της αποστολής.

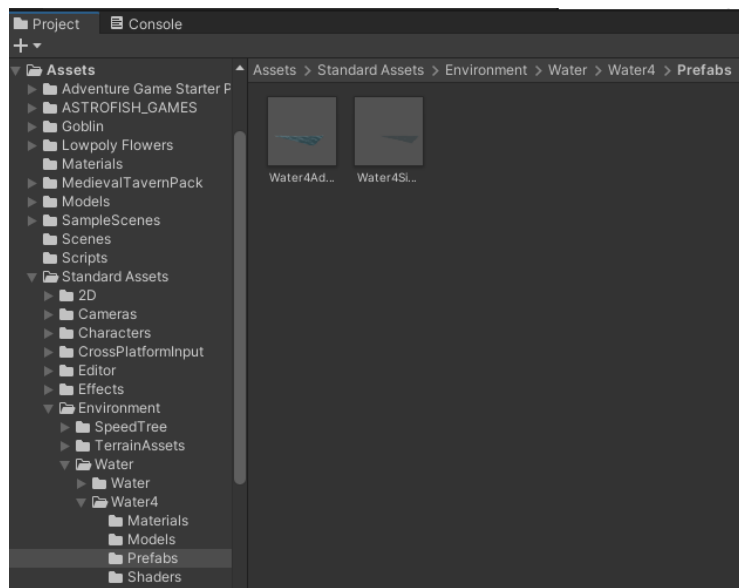


Εικόνα 63 2ο τμήμα του μηνύματος.

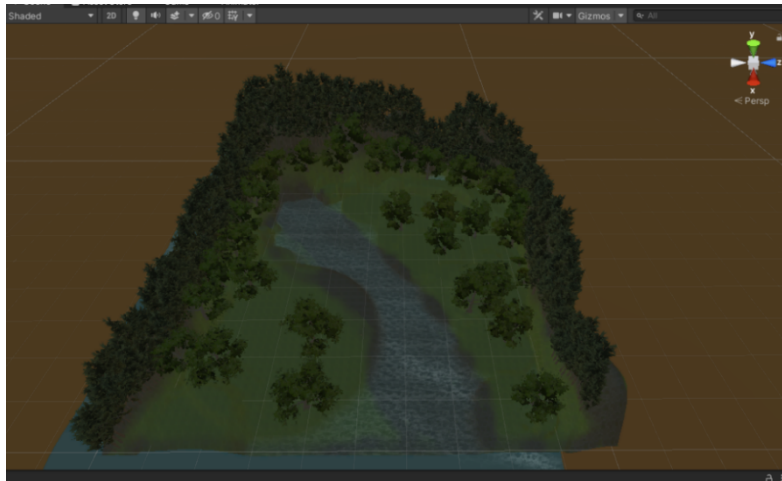
5.7. Επίπεδο πρώτο

5.7.1. Terrain

Η πίστα αυτή αποτελεί την πρώτη πίστα όπου ο ήρωας θα μάχεται με τους εχθρούς. Για το λόγο αυτό δημιουργείται μια νέα σκηνή στη Unity στην οποία τοποθετείται εκ νέου ένα αντικείμενο **Terrain** (Εικόνα 65) το οποίο παραμετροποιείται κατάλληλα με τις μεθόδους που παρουσιάστηκαν στις προηγούμενες παραγράφους. Επιπλέον στη πίστα τοποθετείται και νερό για να ενισχύσει την αληθοφάνεια (Εικόνα 64). Το αντικείμενο αυτό επιλέγεται από τα φάκελο *Assets \ Standard Assets \ Environment \ Water \ Water4 \ Prefabs \ Water4Advanced*.



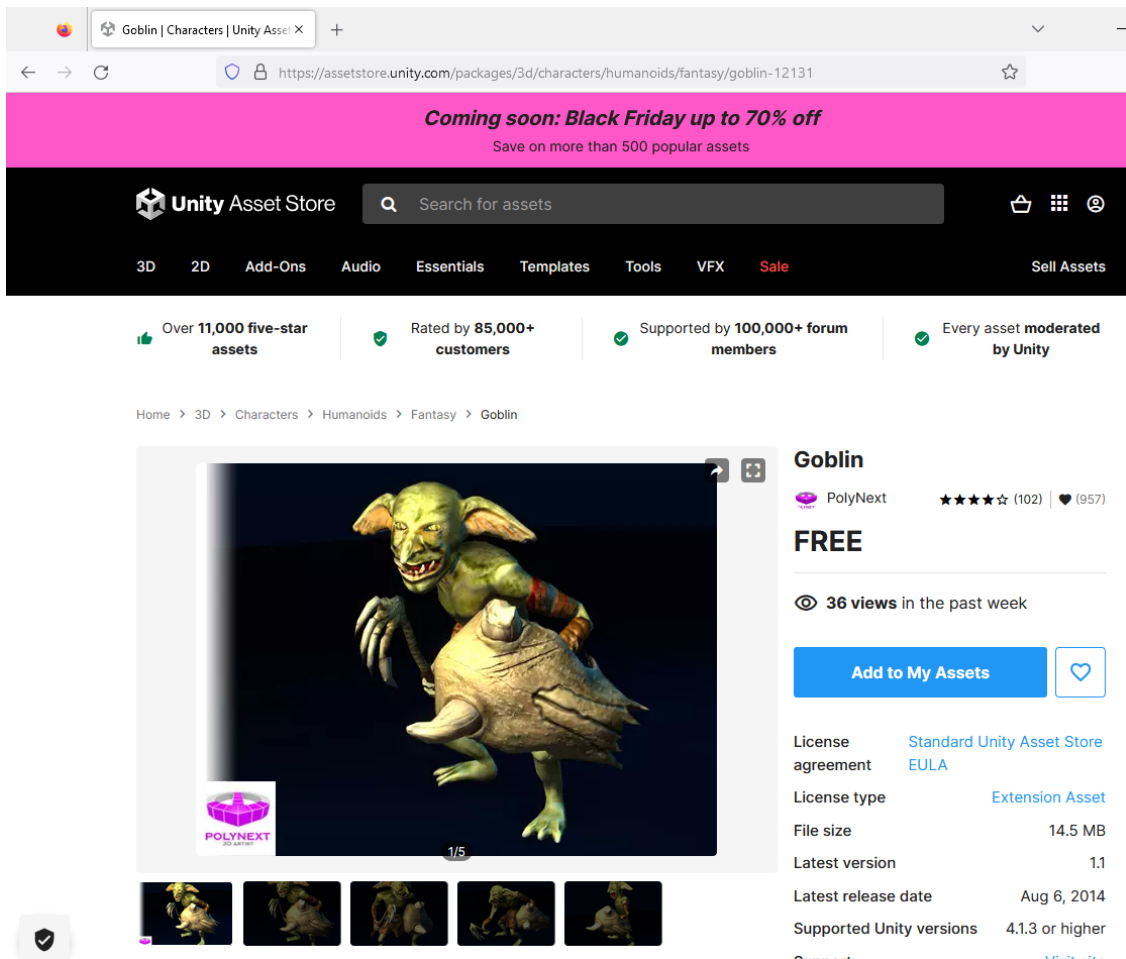
Εικόνα 64 Το Prefab του νερού.



Εικόνα 65 Η πίστα του επιπέδου 1.

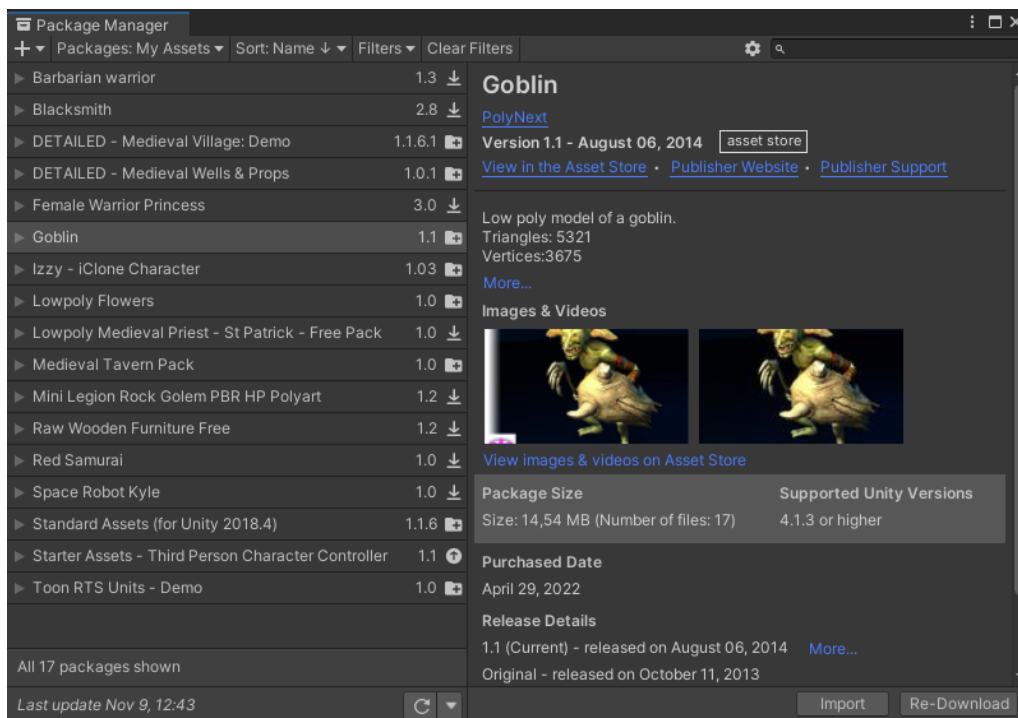
5.7.2. Εισαγωγή στη σκηνή του εχθρικού χαρακτήρα

Ο εχθρός που θα πρέπει να αντιμετωπίσει ο ήρωας είναι το μοντέλο Goblin (Εικόνα 66) από την εταιρία Polynext [35].



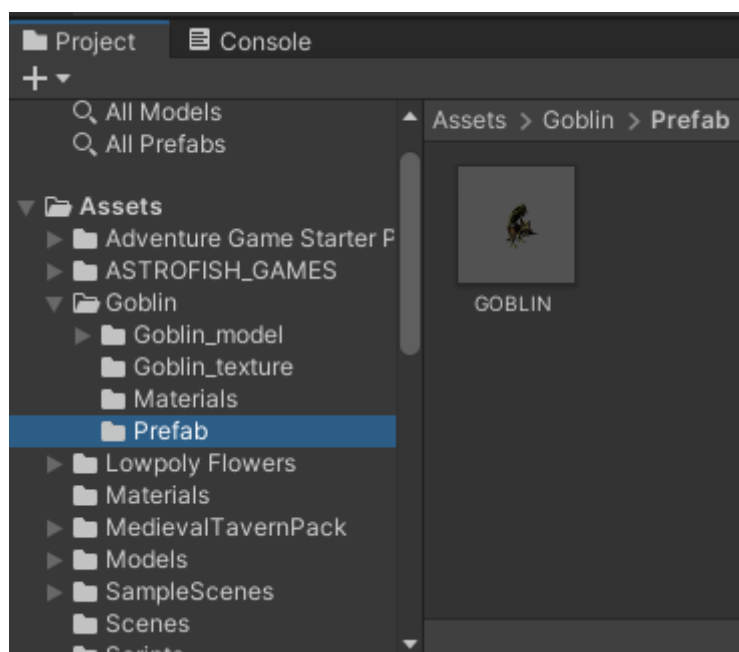
Εικόνα 66 Μοντέλο του εχθρού. Πηγή: assetstore

Για την εισαγωγή του μοντέλου ο χρήστης θα πρέπει να χρησιμοποιήσει το λογαριασμό του και ακολούθως να επιλέξει το κουμπί **Open to Unity**. Ακολούθως θα ανοίξει αυτόματα το παράθυρο **Package Manager** της Unity (Εικόνα 67) από όπου ο χρήστης θα κατεβάσει τα απαραίτητα αρχεία στον υπολογιστή του και θα τα εισάγει στις κατάλληλες διαδρομές της Unity.

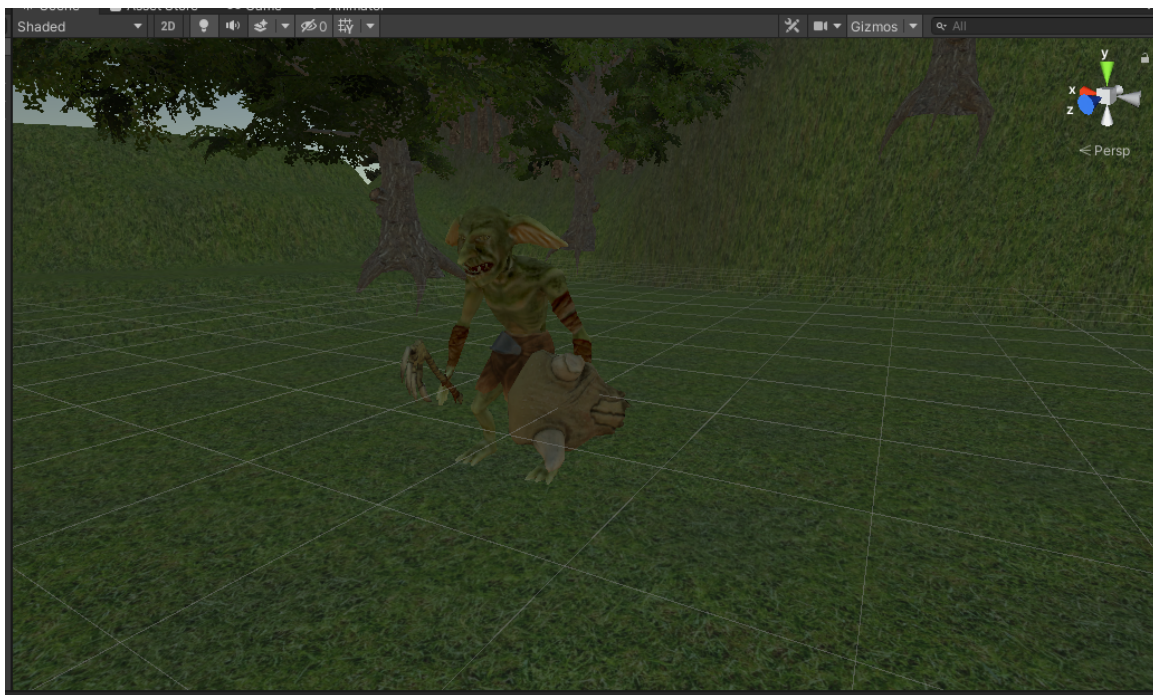


Εικόνα 67 Package Manager της Unity.

Ακολουθώντας το μοντέλο θα εμφανίζεται μέσα στο φάκελο **Assets** με το όνομα **Goblin** (Εικόνα 68). Για την εισαγωγή του στη πίστα ο σχεδιαστής θα επιλέξει το **Prefab Goblin** (Εικόνα 69).



Εικόνα 68 Εισαγωγή μοντέλου στη σκηνή.

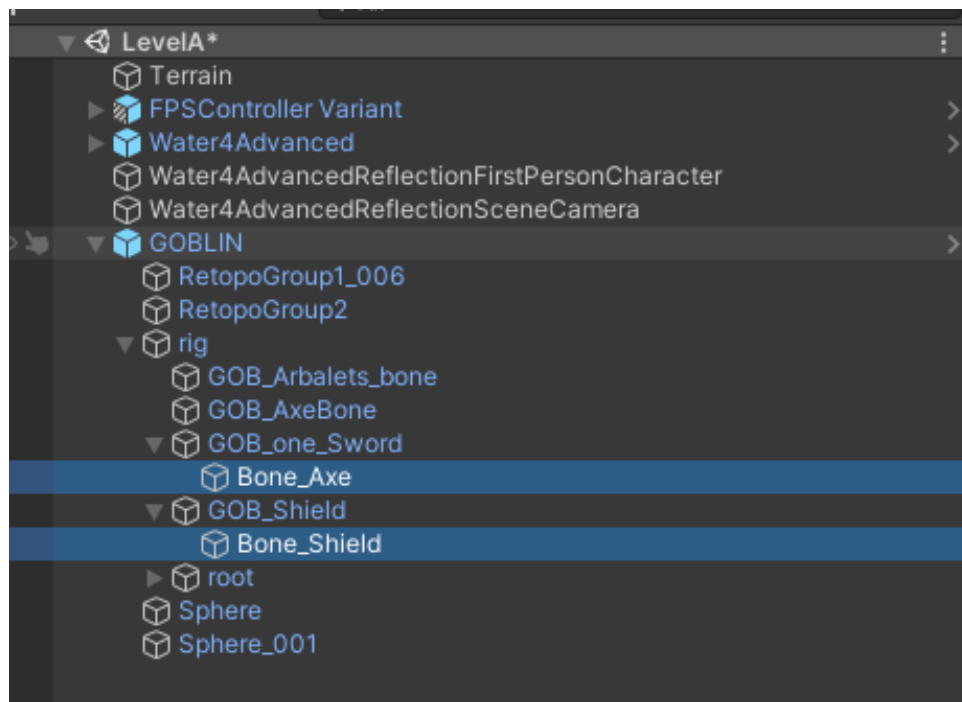


Εικόνα 69 Ο εχθρός στη σκηνή.

5.7.3. Επεξεργασία του μοντέλου

Ο εχθρός αρχικά θα είναι ιδιαίτερα ευάλωτος στα χτυπήματα του ήρωα καθώς και ο ήρωας θα είναι στο αρχικό επίπεδο και δεν θα διαθέτει μεγάλες δυνάμεις. Συνεπώς ο εχθρός θα πρέπει να μην είναι αρκετά δυνατός ώστε ο ήρωας να μπορεί εύκολα να τον αντιμετωπίσει. Για το σκοπό αυτό ο εχθρός θα εμφανίζεται μικρός σε διαστάσεις και δεν θα διαθέτει ασπίδα και όπλο (Εικόνα 70) ώστε να μην μπορεί ούτε να αμυνθεί στα χτυπήματα του ήρωα αλλά ούτε να προκαλέσει μεγάλη ζημιά στον ήρωα με τα δικά του χτυπήματα.

Για να απενεργοποιηθούν και να εξαφανιστούν τα αντικείμενα αυτά θα πρέπει να γραφεί ένα Script (Εικόνα 71) το οποίο θα ανατεθεί στο μοντέλο (Εικόνα 72). Το script θα διαθέτει αναφορές προς τα αντικείμενα αυτά (ασπίδα και όπλο) και στη συνάρτηση Start μέσω της συνάρτησης SetActive των αντικειμένων θα τα απενεργοποιεί. Έτσι δεν θα είναι πλέον ορατά (Εικόνα 73).



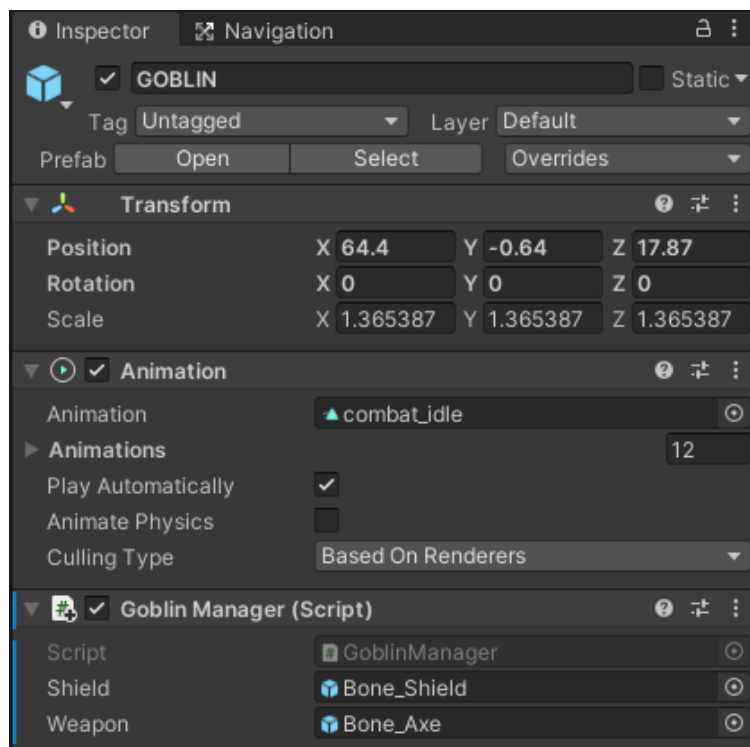
Εικόνα 70 Τα αντικείμενα ασπίδα και όπλο του μοντέλου.

```

6 public class GoblinManager : MonoBehaviour
7 {
8     public GameObject shield;
9     public GameObject weapon;
10
11     // Start is called before the first frame update
12     void Start()
13     {
14         shield.SetActive(false);
15         weapon.SetActive(false);
16     }
17
18     // Update is called once per frame
19     void Update()
20     {
21     }
22 }
23
24

```

Εικόνα 71 Script απενεργοποίησης όπλου και ασπίδας μοντέλου.

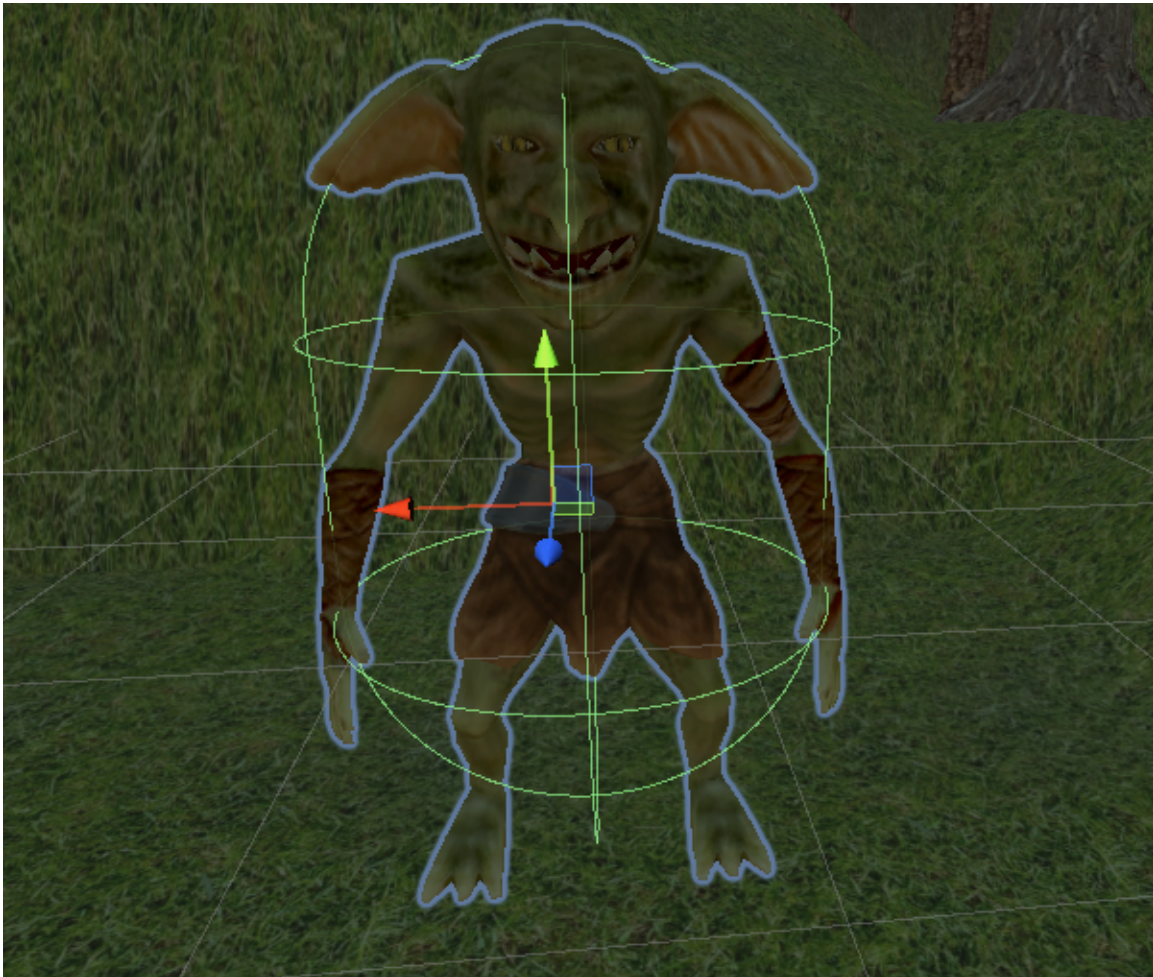


Εικόνα 72 Το μοντέλο με το Script και τις αναφορές στα αντικείμενα.



Εικόνα 73 Ο εχθρός χωρίς όπλο και ασπίδα.

Τέλος στο μοντέλο θα πρέπει να προστεθεί ένας **Collider** (Εικόνα 74) ο οποίος θα αποτρέπει τον ήρωα να μπορεί να διέλθει μέσα από το μοντέλο.

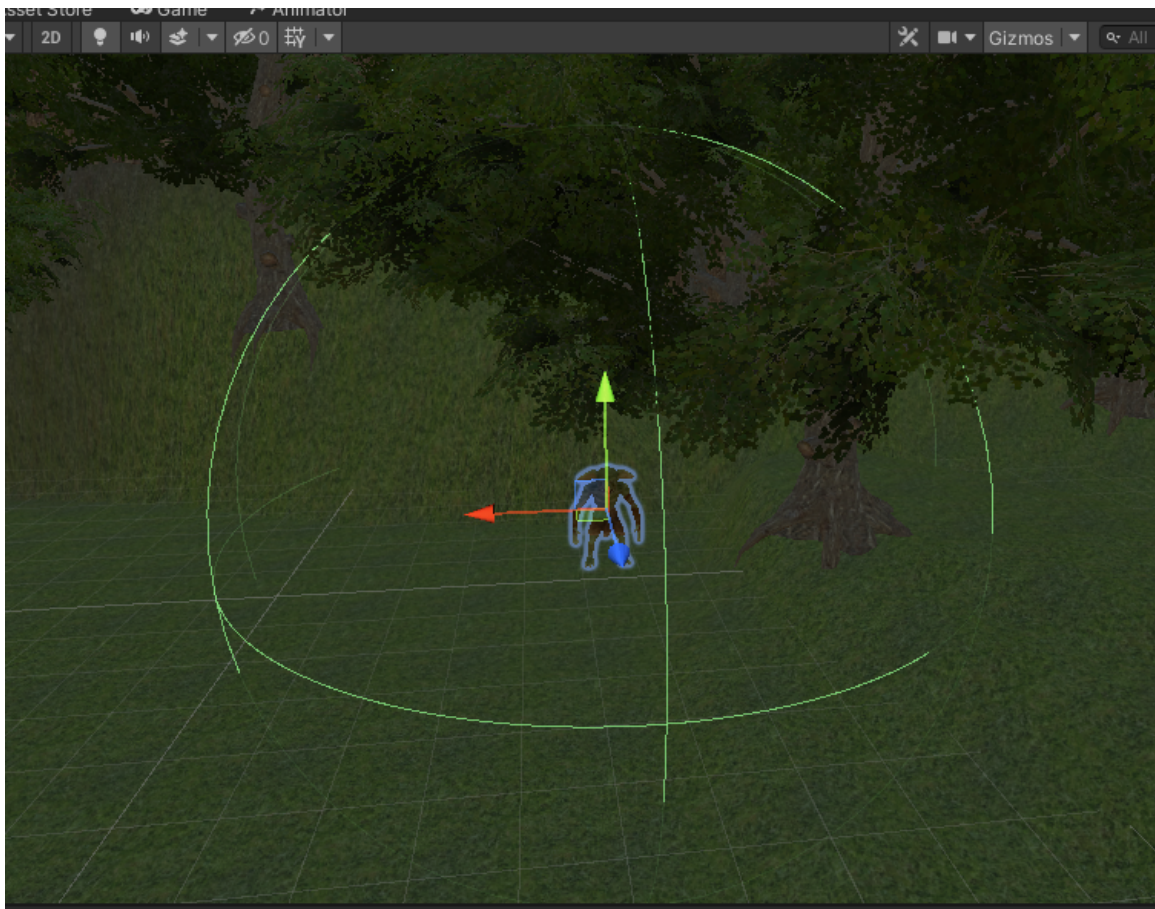


Εικόνα 74 Collider αποτροπής του ήρωα από το να διέρχεται από το εσωτερικό του μοντέλου.

5.7.4. Νοημοσύνη στον χαρακτήρα του αντιπάλου

Το μοντέλο μόλις εντοπίζει τον ήρωα θα του επιτίθεται. Πιο συγκεκριμένα μόλις ο ήρωας εισέρχεται στο οπτικό πεδίο του μοντέλου αυτό θα κινείται αυτόνομα προς τον ήρωα προσπερνώντας τα διάφορα εμπόδια. Τέλος μόλις φτάνει σε ακτίνα δράσης θα σταματά να κινείται και θα ξεκινά την επίθεση του. Συνεπώς το Script που δημιουργήθηκε στη προηγούμενη παράγραφο θα πλαισιωθεί με επιπλέον κώδικα ο οποίος θα υλοποιεί τη λειτουργικότητα αυτή.

Για να μπορεί να εντοπίζει το μοντέλο τον ήρωα θα πρέπει αρχικά αυτός να βρίσκεται ενός ορισμένης απόστασης από αυτόν με το σκεπτικό ότι το μοντέλο μπορεί να βλέπει και να εντοπίζει τον εχθρό έως μια συγκεκριμένη απόσταση (Εικόνα 75). Συνεπώς τοποθετείται ένα αντικείμενο τύπου **Sphere Collider**, με ενεργοποιημένο το **Is Trigger**, το οποίο έχει ακτίνα 5. Αντίστοιχα, ο ήρωας είναι εφοδιασμένος με ένα **Rigidbody**. Συνεπώς μόλις ο ήρωας πλησιάσει το μοντέλο σε απόσταση μικρότερη των 5 μέτρων θα ενεργοποιηθεί η συνάρτηση **OnTriggerStay** του μοντέλου.

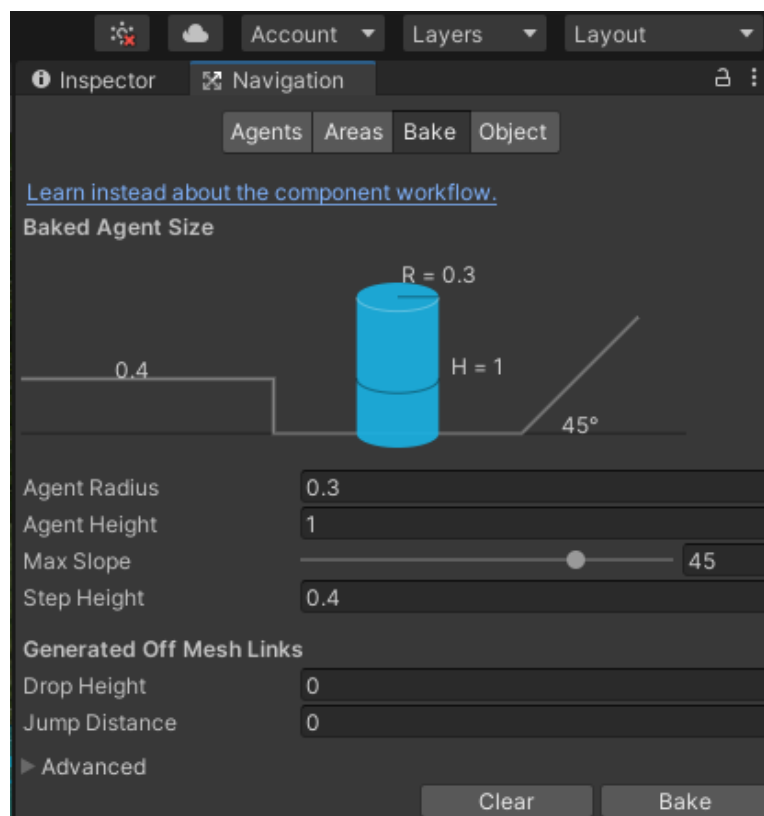


Εικόνα 75 Ακτίνα εντός της οποίας το μοντέλο εντοπίζει εχθρούς.

Ακολουθώντας θα ελέγξει το κατά πόσον ο χαρακτήρας βρίσκεται εντός του οπτικού πεδίου του μοντέλου το οποίο έχει εύρος 110 μοίρες. Στη περίπτωση που αυτό είναι αληθές η μεταβλητή **isPlayerInSight** θα τίθεται σε **True**. Το εύρος του οπτικού πεδίου μπορεί στις επόμενες πίστεις να διευρύνεται ώστε ο χαρακτήρας να μπορεί να εντοπίζει τον ήρωα ευκολότερα και το παιχνίδι να δυσκολεύει.

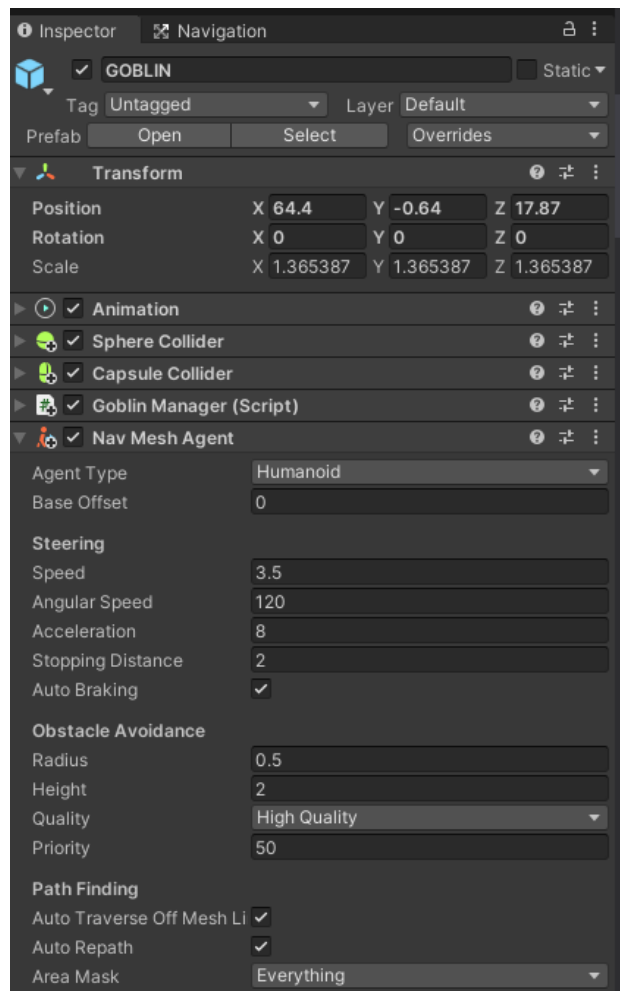
Από τη στιγμή που το μοντέλο εντοπίζει τον ήρωα θα κινείται επιθετικά εναντίων του. Για να μπορεί να κινηθεί το μοντέλο και να προσπεράσει τα διάφορα εμπόδια θα χρησιμοποιηθεί το αντικείμενο **Navigation**. Το αντικείμενο αυτό δημιουργεί ένα χάρτη της πίστας ο οποίος περιέχει τα σημεία που το μοντέλο μπορεί να κινηθεί ώστε να αποφεύγει αυτά που δεν μπορεί, όπως για παράδειγμα τα δέντρα. Για την ενεργοποίηση του αντικειμένου ο χρήστης επιλέγει από το μενού το κουμπί **Window** και ακολούθως **AI** και **Navigation**.

Επόμενο βήμα είναι να οριστούν τα αντικείμενα της πίστας που είναι σταθερά και ο χαρακτήρα θα πρέπει να τα προσπεράσει. Αντικείμενα τέτοιου τύπου είναι το **Terrain**, σπίτια καθώς και άλλα. Η πίστα της παρούσας παρουσίασης περιέχει μόνο **Terrain** συνεπώς δεν απαιτείται κάποια ενέργεια. Μόνο στη καρτέλα **Navigation**, στο κουμπί **Bake** θα πρέπει να οριστεί η γωνία και το ύψος του πράκτορα ώστε η κίνηση του χαρακτήρα να είναι όσο το δυνατόν αληθοφανής (Εικόνα 76). Τέλος ο χρήστης επιλέγει το κουμπί **Bake**.



Εικόνα 76 Παραμετροποίηση του Navigation.

Τέλος στο μοντέλο θα πρέπει να ανατεθεί ο **Agent** (Εικόνα 77) ο οποίος θα διαβάζει τον **Navigation** και θα κινείται. Η διαδικασία ολοκληρώνεται επιλέγοντας το Goblin και ακολούθως **Add Component Nav Mesh Agent**. Στο αντικείμενο αυτό ο προγραμματιστής θα ορίσει διάφορες παραμέτρους της κίνησης όπως τη ταχύτητα, την απόσταση από τον ήρωα που θα σταματά καθώς και άλλα.



Εικόνα 77 Nav Mesh Agent.

Το μόνο που απομένει πλέον για να ολοκληρωθεί η διαδικασία είναι η ενεργοποίηση της λειτουργικότητας αυτής μέσω του αρχείου Script. Αρχικά θα πρέπει να δηλωθεί και να αρχικοποιηθεί το αντικείμενο **Nav Mesh Agent** και ακολούθως, όταν θα πρέπει να κινηθεί, μέσω της `SetDestination` να του δίνεται η κατεύθυνση (Εικόνα 78). Η κατεύθυνση λαμβάνεται από την ιδιότητα **transform** του αντικειμένου που εισήλθε στο **Collider** του και είναι ο **Player**.

```

32 void OnTriggerStay(Collider other)
33 {
34     direction = other.transform.position - transform.position;
35     calculatedAngle = Vector3.Angle(direction, transform.forward);
36
37     if (calculatedAngle < fieldOfViewAngle * 0.5f)
38     {
39         isPlayerInSight = true;
40         nav.SetDestination(other.transform.position);
41     }
42 }

```

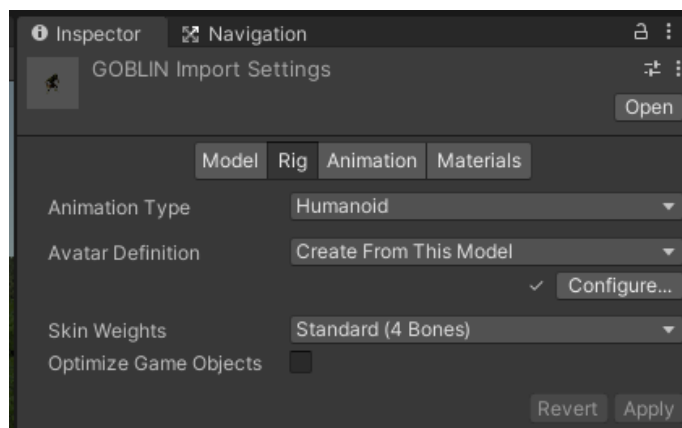
Εικόνα 78 Ορισμός κατεύθυνσης στον Nav Mesh Agent.

5.7.5. Animation στο χαρακτήρα

Το μοντέλο πλέον έχει τη δυνατότητα να κινηθεί προς τον ήρωα όμως δεν γνωρίζει πώς να κινήσει τα πόδια του. Συνεπώς θα πρέπει να ενεργοποιηθεί το **animation** που συνοδεύει το μοντέλο ώστε να ξεκινά να εκτελείται και να κινούνται τα πόδια του μοντέλου όταν αυτό ξεκινά να τρέχει. Ακολουθώντας και μόλις το μοντέλο φτάσει τον ήρωα θα μπαίνει σε κατάσταση επίθεσης όπου θα τον χτυπά περιοδικά. Συνεπώς η κατάσταση του θα εναλλάσσεται μεταξύ δύο καταστάσεων στάσης και επίθεσης.

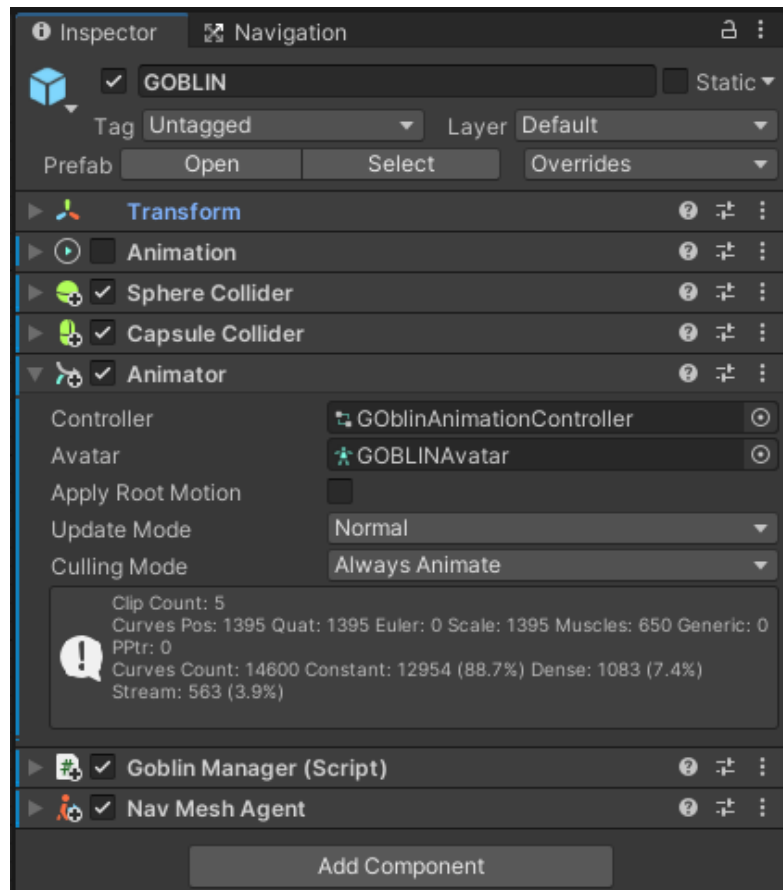
5.7.6. Καθορισμός Avatar

Για να εισαχθούν τα animation που συνοδεύουν το μοντέλο και να μπορέσει να τα χρησιμοποιήσει η Unity θα πρέπει να δημιουργήσει η Unity ένα σκελετό (**Avatar**) από το μοντέλο. Η διαδικασία αυτή επιτυγχάνεται επιλέγοντας το μοντέλο GOBLIN από το φάκελο *Goblin_model* και ακολουθώντας από τη καρτέλα **Inspector** την καρτέλα **Rig** επιλέγεται το είδος του **Animation** καθώς και ο τύπος του **Avatar** που θα δημιουργηθεί. Μετά τις απαραίτητες ρυθμίσεις επιλέγεται το κουμπί **Configure** (Εικόνα 79).



Εικόνα 79 Καθορισμός Avatar για τη λειτουργία του animation.

Μετά την ολοκλήρωση της διαδικασίας η Unity θα δημιουργήσει ένα **Avatar** στον ίδιο φάκελο το οποίο θα πρέπει να ανατεθεί στο αντικείμενο **Animator** που θα ανατεθεί στο Goblin. Το αντικείμενο αυτό θα περιέχει επιπλέον τον **Animation Controller** (Εικόνα 80) που θα έχει τις καταστάσεις οι οποίες θα εμφανίζουν τα **Animations**. Τέλος ο default Animation που συνοδεύει το μοντέλο θα πρέπει να απενεργοποιηθεί.

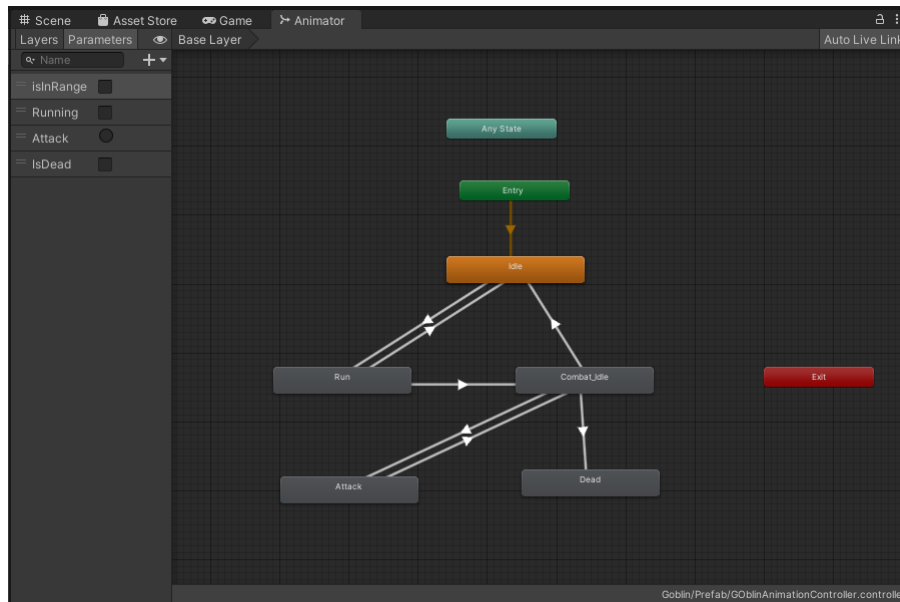


Εικόνα 80 Δημιουργία Animator Controller στο Goblin.

5.7.7. Animation Controller

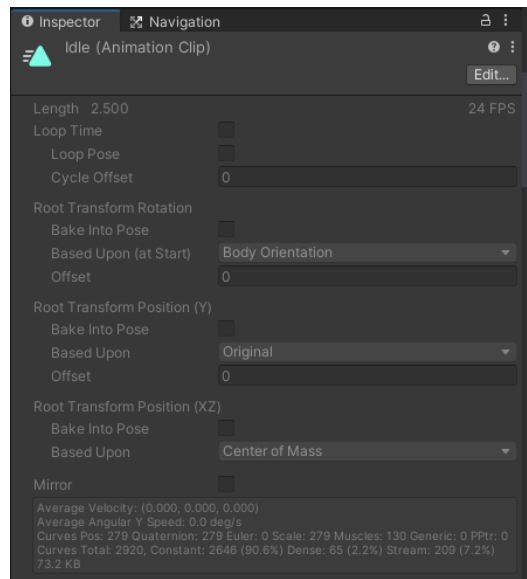
Στο Goblin μπορούν αρχικά να οριστούν οι καταστάσεις Idle, Run, Combat_Idle, Attack και Dead, όπου η κάθε κατάσταση εκτελεί το ομώνυμο animation (Εικόνα 81). Πιο συγκεκριμένα όταν το παιχνίδι ξεκινά το Goblin θα βρίσκεται σε κατάσταση Idle και μόλις εντοπίζει τον παίχτη θα ενεργοποιείται η μεταβλητή Running και θα μεταβαίνει στη κατάσταση Run. Μόλις φτάνει τον ήρωα θα ενεργοποιείται η κατάσταση Combat_idle

από την οποία μέσω της μεταβλητής τύπου trigger θα μεταβαίνει στη κατάσταση Attack, θα χτυπά τον ήρωα και ακολούθως θα επιστρέφει στην Combat_Idle. Τέλος όταν η ζωή του τελειώνει θα μεταβαίνει από την Combat_idle στη Dead. Αντίθετα αν ο ήρωας απομακρυνθεί και το Goblin δεν τον βλέπει πλέον θα επιστρέφει στη κατάσταση Idle.

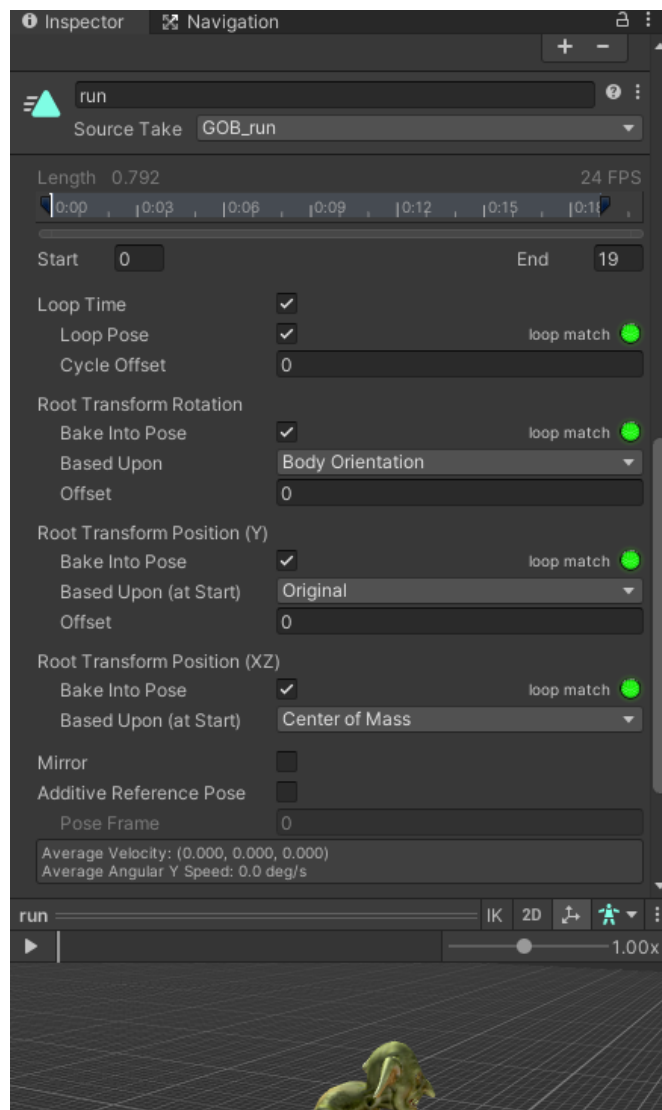


Εικόνα 81 Animation Controller του Goblin.

Όλα τα animations που θα χρησιμοποιηθούν θα πρέπει να έχει ενεργοποιηθεί η **επιλογή Loop Time** καθώς και το **Pose** ώστε να εκτελούνται συνεχώς και όχι να σταματάνε μετά το πέρας της μιας εκτέλεσης (Εικόνα 82, Εικόνα 83). Η ιδιότητα αυτή ορίζεται επιλέγοντας το animation από το φάκελο Goblin, Goblin_model και ακολούθως το κουμπί **Edit**.



Εικόνα 82 Επεξεργασία του Animation.



Εικόνα 83 Loop στο Animation.

5.7.8. Πρόσβαση στις μεταβλητές του Animation Controller

Ο προγραμματιστής μέσω του Script αρχείου που αναθέτει στο Goblin έχει πρόσβαση στις μεταβλητές του **Animator Controller** οι οποίες αποτελούν τα **Flags** που εναλλάσσουν τις καταστάσεις. Για το σκοπό αυτό δημιουργείται στο αρχείο μια μεταβλητή τύπου **Animator**. Η μεταβλητή αρχικοποιείται στην Start μέσω της GetComponent δίνοντας της ως τύπο αντικειμένου το Animator. Πλέον η μεταβλητή τύπου Animator διαθέτει συναρτήσεις τύπου Set για τη χρήση των αντίστοιχων μεταβλητών (Εικόνα 84).

```

41
42 if (calculatedAngle < fieldOfViewAngle * 0.5f)
43 {
44     isPlayerInSight = true;
45     nav.SetDestination(other.transform.position);
46     animator.SetBool("Running", true);
47 }
48
49 if (isPlayerInSight)
50 {
51     if (distance < 1.1f)
52     {
53         attack = true;
54         animator.SetBool("isInRange", true);
55     }
56     else
57     {
58         attack = false;
59         animator.SetBool("isInRange", false);
60     }
61 }

```

Εικόνα 84 Πρόσβαση στις μεταβλητές του Animator Controller.

5.7.9. Επίθεση στον ήρωα - Πρόσβαση σε μεταβλητές διαφορετικού Script

Το μοντέλο θα επιτίθεται στον ήρωα και θα προσπαθεί να τον εξοντώσει. Σε κάθε χτύπημα που θα του καταφέρει θα του αφαιρεί ένα ποσοστό από τη ζωή του και όταν αυτή εξαντλείται τότε το παιχνίδι θα ολοκληρώνεται και ο παίχτης θα χάνει. Στην αρχική πίστα το μοντέλο δεν θα έχει αρκετή δύναμη και τα χτυπήματά του θα αφαιρούν μικρό ποσοστό από τη ζωή του ήρωα (20) ενώ στις επόμενες πίστες το ποσοστό αυτό θα αυξάνεται.

Για το σκοπό αυτό θα δημιουργηθεί ένα κενό αντικείμενο, με όνομα GameManager, το οποίο θα διαθέτει ένα Script που μεταξύ άλλων θα φιλοξενεί και τη ποσότητα της ζωής του ήρωα (Εικόνα 85). Η τιμή αρχικά θα είναι 100 και σε κάθε χτύπημα θα μειώνεται κατά 20. Παράλληλα η πίστα θα διαθέτει ένα καμβά για την εμφάνιση του υπολοίπου της ζωής του ήρωα το οποίο θα εμφανίζεται μετά από κάθε χτύπημα. Τέλος το Script, στην Update θα παρακολουθεί την τιμή της ζωής και όταν αυτή εξαντληθεί θα ενημερώνει τον παίχτη ότι έχασε και θα τερματίζει το παιχνίδι.

```

5
6 public class LevelGameManager : MonoBehaviour
7 {
8     // Start is called before the first frame update
9
10    public int heroLife;
11    public Text heroLifeText;
12    public Canvas heroLifeCanvas;
13
14    void Start()
15    {
16        heroLife = 100;
17    }
18
19    // Update is called once per frame
20    void Update()
21    {
22        if (heroLife <= 0)
23        {
24            heroLifeText.text = "Εχασες!!!";
25            heroLifeCanvas.enabled = true;
26            Time.timeScale = 0;
27        }
28    }
29

```

Εικόνα 85 Κώδικας του GameManager Script.

Ακολουθώντας θα πρέπει το μοντέλο να έχει πρόσβαση στη τιμή της ζωής του ήρωα και να μπορεί να τη μειώνει σε κάθε χτύπημα (Εικόνα 86). Συνεπώς το Goblin, στο Script του διαθέτει μια μεταβλητή τύπου GameObject στην οποία, μέσω της Unity, έχει ανατεθεί το αντικείμενο GameManager που φιλοξενεί τη ζωή του ήρωα. Το GameObject αυτό διαθέτει το Script LevelGameManager και στο Script υπάρχει η μεταβλητή heroLife. Για να αποκτήσει πρόσβαση σε αυτή το Script του Goblin θα πρέπει από το αντικείμενο στο οποίο έχει ανατεθεί το GameManager να κληθεί η συνάρτηση GetComponent η οποία θα επιστρέψει το πρώτο αντικείμενο τύπου LevelGameManager που αντιστοιχεί στο Script που φέρει τη μεταβλητή heroLife (Εικόνα 87). Τέλος, για να σταματήσει να εκτελείται το παιχνίδι θα πρέπει να αντιστοιχηθεί στη μεταβλητή timescale του αντικειμένου Time η τιμή 0. Η τιμή αυτή αντιστοιχεί στην ροή του χρόνου του παιχνιδιού, συνεπώς η αντιστοίχιση τιμών μεγαλύτερων της μονάδας έχει σαν συνέπεια την αύξηση της ταχύτητας ροής του παιχνιδιού ενώ οι μικρότερες τιμές τη μείωση. Η ανάθεση της μηδενικής τιμής έχει σαν συνέπεια τη παύση του χρόνου και συνεπώς τη παύση του παιχνιδιού.

```

30
31 // Start is called before the first frame update
32 void Start()
33 {
34     heroLifeCanvas.enabled = false;
35     shield.SetActive(false);
36     weapon.SetActive(false);
37     nav = GetComponent<UnityEngine.AI.NavMeshAgent>();
38     animator = GetComponent<Animator>();
39
40     heroLifeInt = leveaGameManager.GetComponent<LaveIAGameManager>().heroLife;
41
42
43 }
44

```

Εικόνα 86 Στη γραμμή 40 ο κώδικας για τη πρόσβαση στη ζωή του ήρωα στο αντίστοιχο αντικείμενο.

Πλέον σε κάθε χτύπημα το αντικείμενο έχει τη δυνατότητα να μειώνει την τιμή και να την εμφανίζει στο χρήστη μέσω του GUI.

```

111 IEnumerator hitFunction()
112 {
113
114     canHit = false;
115     yield return new WaitForSeconds(3);
116     if (Physics.Raycast(transform.position + new Vector3(0, 1, 0), transform.TransformDirection(Vector3.forward), out hit, 2))
117     {
118         hitObject = hit.collider.gameObject;
119         if (hit.collider.gameObject.name.Equals("FPSController Variant"))
120             print("I hit the hero");
121         leveaGameManager.GetComponent<LaveIAGameManager>().heroLife -= 20;
122         StartCoroutine(MessageDisplay("Η ζωή είναι: " + leveaGameManager.GetComponent<LaveIAGameManager>().heroLife));
123     }
124     canHit = true;
125 }

```

Εικόνα 87 Επεξεργασία της μεταβλητής διαφορετικού Script στη γραμμή 121.

5.7.10. Εντοπισμός χτυπήματος μέσω Raycast

Για τον εντοπισμό του χτυπήματος θα χρησιμοποιηθεί το εργαλείο Raycast. Ουσιαστικά πρόκειται για την εξομοίωση μιας ακτίνας που εκτοξεύεται από ορισμένο σημείο προς μια συγκεκριμένη διεύθυνση, η οποία θα ταξιδέψει σε συγκεκριμένη απόσταση. Το αντικείμενο στο οποίο θα χτυπήσει επιστρέφεται μέσω της hit. Συγκεκριμένα για την περίπτωση του Goblin η ακτίνα θα εκτοξευθεί από το ύψος του στήθους του, καθώς εκεί θα πραγματοποιηθεί το χτύπημα, θα ταξιδέψει προς την κατεύθυνση που κοιτά ο χαρακτήρας και θα ταξιδέψει για απόσταση 2 μέτρων. Αν στην απόσταση αυτή έλθει σε επαφή με κάποιο αντικείμενο τότε θα το επιστρέψει μέσω της hit. Ο προγραμματιστής

στη συνέχεια μπορεί να ελέγξει το όνομα του αντικειμένου που χτυπήθηκε και αν αυτό είναι ο ήρωας να του αφαιρέσει ποσοστό από τη ζωή του. Η απόσταση που θα ταξιδέψει η ακτίνα μπορεί να αυξηθεί ώστε να βελτιώσει την στοχευτική ικανότητα του χαρακτήρα στις επόμενες πίστες.

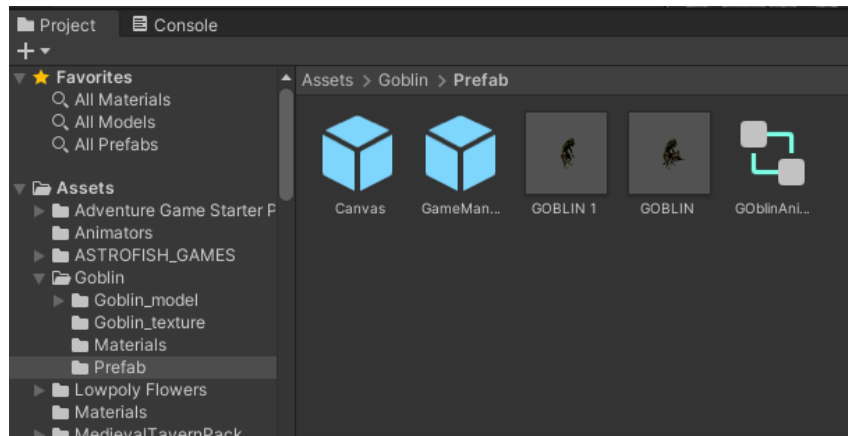
5.8. Prefab Αντικείμενο

Το αντικείμενο του εχθρού είναι πλέον έτοιμο για χρήση. Περιέχει το αρχικό μοντέλο που εισήχθη στη Unity ενισχυμένο με τον Animation Controller ο οποίος είναι υπεύθυνος για την εκτέλεση των animation τις κατάλληλες στιγμές, τα απαραίτητα Scripts, τους Collider καθώς και τον Agent που του επιτρέπει να κινείται. Τέλος έχει τις κατάλληλες διαστάσεις καθώς και έχουν απενεργοποιηθεί η ασπίδα και το όπλο του. Το μοντέλο αυτό θα χρησιμοποιηθεί και στις επόμενες πίστες με μικρές παραλλαγές.

Πιο συγκεκριμένα στις επόμενες πίστες το μοντέλο θα εμφανίζεται με αυξημένες διαστάσεις και θα φέρει και όπλο αλλά και ασπίδα ώστε να είναι δυσκολότερο στον ήρωα να το αντιμετωπίσει. Πέραν αυτών των διαφορών όλες οι υπόλοιπες λειτουργίες που έχουν προστεθεί στο αρχικό μοντέλο (Animator Controller, Colliders, Scripts κλπ) θα μείνουν ως έχουν.

Για να χρησιμοποιηθεί το μοντέλο αυτό του εχθρού όπως έχει ενισχυθεί με τις νέες αυτές δυνατότητες εκ νέου στην ίδια πίστα ή και σε άλλες θα πρέπει να δημιουργηθεί ένα Prefab (Εικόνα 88). Το Prefab έχει το νόημα της κλάσης, δηλαδή θα δημιουργηθεί ένα αντικείμενο που θα περιλαμβάνει το σύνολο των λειτουργιών που περιέχει το μοντέλο. Ακολούθως μπορούν να δημιουργηθούν άλλα αντικείμενα από αυτό (όπως ακριβώς τα αντικείμενα μιας κλάσης) τα οποία θα κληρονομούν το σύνολο των λειτουργιών και θα μπορούν να ενισχυθούν με επιπλέον δυνατότητες ή χαρακτηριστικά. Τέλος αν γίνει μια αλλαγή στο αρχικό Prefab η αλλαγή αυτή θα μεταφερθεί αυτομάτως σε όλα τα αντικείμενα του Prefab αυτού. Συνεπώς αν ο προγραμματιστής εισάγει στη σκηνή αντικείμενα του Prefab αυτού και στη συνέχεια αποφασίσει να αλλάξει κάποιο χαρακτηριστικό σε όλα τα αντίγραφα θα εφαρμόσει την αλλαγή του στο Prefab και αυτομάτως θα ενημερωθούν όλα τα αντίγραφα. Για τη δημιουργία του Prefab αρκεί η μεταφορά του αντικειμένου από τη καρτέλα Hierarchy στη καρτέλα Project, στο κατάλληλο φάκελο.

Ο χαρακτήρας όμως δεν είναι ανεξάρτητος αλλά χρησιμοποιεί τον καμβά για να εμφανίσει μηνύματα καθώς και το αντικείμενο GameManager από το οποίο ανακτά το επίπεδο της ζωής του. Παράλληλα τα αντικείμενα αυτά θα χρησιμοποιηθούν και στις επόμενες σκηνές συνεπώς θα γίνουν και αυτά Prefabs.



Εικόνα 88 Δημιουργία Prefabs.

5.9. Singleton Αντικείμενο

Τα χτυπήματα του ήρωα στο μοντέλο θα έχουν διαφορετική επίδραση ανάλογα με τη δύναμη του. Πιο συγκεκριμένα αν ο ήρωας δεν έχει δημιουργήσει τα φίλτρα δεν θα έχει όπλα και συνεπώς δεν θα μπορεί να επιφέρει σημαντικά χτυπήματα στον εχθρό του. Ο ήρωας, μόλις ξεκινά το παιχνίδι θα διαθέτει μόνο τις γροθιές του για να αντιμετωπίσει τον εχθρό, οι οποίες δεν θα του προκαλούν μεγάλη ζημιά. Μόλις ολοκληρώσει τη 2η του αποστολή, θα αποκτήσει πρόσβαση στο μαχαίρι. Ακολούθως η 4η αποστολή του δίνει πρόσβαση στο ρόπαλο και τέλος η 5η στο τσεκούρι. Θα πρέπει λοιπόν να δημιουργηθεί ένα αντικείμενο το οποίο να διατηρείται μεταξύ των σκηνών και να φιλοξενεί τις πληροφορίες αυτές ώστε να είναι διαθέσιμες σε όλες τις σκηνές. Το αντικείμενο αυτό θα είναι τύπου Singleton ώστε να υπάρχει μόνο ένα αντικείμενο καθόλη τη χρονική διάρκεια του παιχνιδιού. Ταυτόχρονα θα είναι δημόσιο ώστε κάθε αντικείμενο του παιχνιδιού να έχει πρόσβαση στα δεδομένα του χωρίς να απαιτούνται συναρτήσεις. Τέλος θα είναι static ώστε κάθε αντίγραφο του αντικειμένου θα μοιράζεται τις ίδιες μεταβλητές και όχι αντίγραφα αυτών. Το να είναι static το αντικείμενο έχει και το πλεονέκτημα ότι δεν απαιτείται η αρχικοποίηση του προτού γίνει διαθέσιμο σε κάποιο άλλο αντικείμενο.

Η αρχικοποίηση του Singleton γίνεται μέσω της Awake στο Script που θα του ανατεθεί (Εικόνα 89) στην οποία αρχικά ελέγχεται το κατά πόσον το αντικείμενο έχει ήδη δημιουργηθεί. Αν έχει δημιουργηθεί από κάποια άλλη σκηνή τότε το νέο διαγράφεται και η συνάρτηση επιστρέφει. Διαφορετικά, όταν δημιουργείται για πρώτη φορά κατά την εκκίνηση του παιχνιδιού, αντιστοιχεί στη μεταβλητή Instance το αντικείμενο και μέσω της DontDestroyOnLoad γνωστοποιεί στη Unity να μην το καταστρέψει όταν η σκηνή θα αφαιρεθεί από τη μνήμη.

```
5 public class SingletonScript : MonoBehaviour
6 {
7     public static SingletonScript Instance;
8
9     public int Guns;
10
11     private void Awake()
12     {
13         if (Instance != null)
14         {
15             Destroy(gameObject);
16             return;
17         }
18         Guns = 0;
19         Instance = this;
20         DontDestroyOnLoad(gameObject);
21     }
22 }
```

Εικόνα 89 Κώδικας του Singleton.

Πλέον ο προγραμματιστής μπορεί να αποθηκεύσει δεδομένα στο αντικείμενο και αυτά να είναι διαθέσιμα για ανάγνωση αλλά και ενημέρωση στις υπόλοιπες σκηνές του παιχνιδιού. Ειδικά για τα διαθέσιμα όπλα θα δημιουργηθεί μια public μεταβλητή τύπου int με το όνομα Guns η οποία θα αρχικοποιηθεί μέσω της Awake σε 0 καθώς αρχικά ο ήρωας δεν θα διαθέτει κανένα όπλο. Ακολουθώντας το Script του αντικειμένου GameManager της σκηνής των φίλτρων μπορεί να έχει πρόσβαση στο αντικείμενο και να ενημερώνει την τιμή του όταν ολοκληρώνεται μια πίστα (Εικόνα 90).

```

110     if (objec2Started)
111     {
112         // 0 0 0 0 0
113         if (!ketControlScr1.hasFlower & !ketControlScr2.hasFlower & !ketControlScr3.hasFlower & !ketControlScr4.hasFlower & !ketControlScr5.hasFlower)
114         {
115             objec2Complete = true;
116
117             SingletonScript.Instance.Guns = 1;
118
119             StartCoroutine(MessageDisplay("Συγχαρητήρια, μόλις ολοκλήρωσες την δεύτερη σου αποστολή.", "Εκείνα τη 3η..."));
120
121             setObjectiveText(false, false, true, false, false);
122
123             objec2Started = false;
124         }
125     }

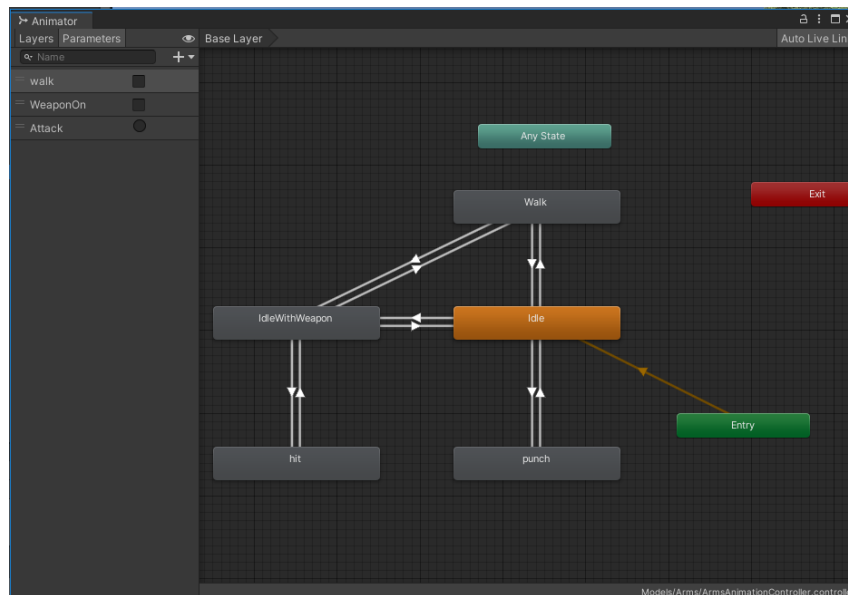
```

Εικόνα 90 Μόλις ολοκληρωθεί η 2η αποστολή, στη γραμμή 117 ενημερώνεται η μεταβλητή *Guns* του *Singleton* για τη νέα τιμή.

5.10. Επίθεση στο Goblin

5.10.1 Animation Controller

Πρώτο βήμα για την υλοποίηση της επίθεσης του ήρωα στον εχθρό είναι η υλοποίηση του **Animator** (Εικόνα 91). Το αντικείμενο που έχει ήδη δημιουργηθεί σε προηγούμενη παράγραφο θα ενισχυθεί με επιπλέον καταστάσεις για να εμφανίζει τα διαφορετικά animation που αντιστοιχούν στα διαφορετικά είδη χτυπημάτων. Πιο συγκεκριμένα όταν ο ήρωας δεν διαθέτει όπλο σε κάθε χτύπημα θα εκτελείται το animation του χτυπήματος της γροθιάς ενώ όταν φέρει όπλο θα εκτελείται μια κίνηση του χεριού κάθετη. Συνεπώς αρχικά ο ήρωας, πριν λάβει όπλο θα βρίσκεται στη κατάσταση *Idle* από την οποία κάθε φορά που πατιέται το αριστερό πλήκτρο του ποντικιού θα μεταβαίνει στην *Punch* στην οποία εκτελείται το animation *Punch*. Αν λάβει όπλο θα μεταβαίνει από την *Idle* στην *IdleWithWeapon* όπου από αυτή με το αριστερό πλήκτρο του ποντικιού θα μεταβαίνει στη *Hit* η οποία θα εκτελεί το animation που φέρει όπλο. Η μετάβαση από την *Idle* στην *IdleWithWeapon* γίνεται μέσω της Boolean μεταβλητής *WeaponOn* η οποία ενεργοποιείται από το Script *WeaponChange* που έχει ανατεθεί στο χέρι που φέρει το όπλο. Στο σημείο αυτό αξίζει να αναφερθεί ότι η τεχνική αυτή δεν είναι αποδοτική με την έννοια ότι οι πληροφορίες αυτές θα έπρεπε να συγκεντρώνονται σε ένα κεντρικό αντικείμενο και αυτό να αναλαμβάνει να εκτελεί τις απαραίτητες διαδικασίες. Διαφορετικά αν κάθε αντικείμενο εκτελεί ενέργειες, καθώς η ανάπτυξη της εφαρμογής θα επεκτείνεται, η διαδικασία θα γίνεται όλο και πιο δυσκολότερη καθώς θα αυξάνεται ο αριθμός των αντικειμένων που καθορίζουν την εκτέλεση της εφαρμογής.



Εικόνα 91 Animator του Ήρωα.

Τέλος, για την εκτέλεση του χτυπήματος εισάγεται μια μεταβλητή τύπου **Trigger** στον **Animator** με το όνομα **Attack**. Η μεταβλητή αυτή ενεργοποιείται από το Script του ήρωα όταν πατιέται το αριστερό κουμπί του πληκτρολογίου. Η μετάβαση έχει απενεργοποιημένη την επιλογή **Has Exit Time** συνεπώς δεν περιμένει να ολοκληρωθεί το προηγούμενο **Animation** προτού ξεκινήσει με συνέπεια το χτύπημα του ήρωα να είναι άμεσο. Για την επιστροφή όμως στη μετά το χτύπημα κατάσταση θα πρέπει η μετάβαση να έχει ενεργοποιημένη της **Has Exit Time** ώστε μόλις ολοκληρώνεται το animation του χτυπήματος ο animator να εμφανίζει την προηγούμενη κατάσταση.

Στη πίστα δημιουργίας του φίλτρου ο ήρωας δεν θα πρέπει να μπορεί να επιτεθεί καθώς το αριστερό πλήκτρο του ποντικιού χρησιμοποιείται για την απόθεση των λουλουδιών στις κανάτες. Συνεπώς προτού εκτελεστεί η λειτουργία της επίθεσης θα πρέπει να ελέγχεται κατά πόσον ο χρήσης βρίσκεται εκτός αυτής της πίστας. Για το σκοπό αυτό δημιουργείται μια μεταβλητή τύπου **Boolean** στο **Singleton** με το όνομα **canAttack**. Η μεταβλητή αυτή αρχικοποιείται σε **false** στο **Singleton** αλλά και στην πίστα των φίλτρων. Αντίθετα στις υπόλοιπες πίστες τίθεται σε **true**. Ακολουθώντας, στο πάτημα του αριστερού κλικ ελέγχεται παράλληλα και η τιμή της προτού εκτελεστεί η λειτουργία (Εικόνα 92).

```

26
27     if (Input.GetMouseButtonDown(0) && SingletonScript.Instance.canAttack)
28     {
29         print("Mouse clicked!!!");
30         anim.SetTrigger("Attack");
31     }
32
33

```

Εικόνα 92 Έλεγχος του αν ο ήρωας μπορεί να επιτεθεί.

5.10.2. Υλοποίηση Επίθεσης

Η επίθεση θα πραγματοποιείται μέσω του εργαλείου **Raycast** όπως και στη περίπτωση του Goblin. Πιο συγκεκριμένα, μόλις ο χρήστης πατήσει το αριστερό κουμπί του πληκτρολογίου εκτοξεύεται μια ακτίνα από το ύψος του στήθους του ήρωα με κατεύθυνση προς το σημείο που κοιτά ο ήρωας, ακτίνα η οποία θα ταξιδέψει για ένα μέτρο. Ταυτόχρονα ξεκινά να προβάλλεται και το αντίστοιχο animation που παρουσιάζει τη κίνηση του χεριού. Στη συνέχεια ελέγχεται το όνομα του αντικειμένου που χτύπησε και αν αυτό είναι το Goblin τότε αφαιρείται ένα ποσό από τη ζωή του η οποία είναι αποθηκευμένη στο GameManager αντικείμενο (Εικόνα 93). Για να συγχρονίζεται η επίθεση με το **animation** η όλη λειτουργικότητα εκτελείται με χρονοκαθυστέρηση σε ξεχωριστό νήμα.

```

35
36     if (Input.GetMouseButtonDown(0) && SingletonScript.Instance.canAttack)
37     {
38
39         if (canHit)
40             StartCoroutine(hitFunction());
41
42     }
43
44
45
46
47     IEnumerator hitFunction()
48     {
49
50         canHit = false;
51         anim.SetTrigger("Attack");
52         yield return new WaitForSeconds(1);
53
54         if (Physics.Raycast(transform.position + new Vector3(0, 0, 0), transform.TransformDirection(Vector3.forward), out hit, 1))
55         {
56             hitObject = hit.collider.gameObject;
57             if (hit.collider.gameObject.name.Equals("GOBLIN"))
58             {
59                 int power = checkWeapon();
60                 leveaGameManager.GetComponent<LaveaGameManager>().goblinLife -= power;
61                 StartCoroutine(MessageDisplay("Η ζωή του Goblin είναι: " + leveaGameManager.GetComponent<LaveaGameManager>().goblinLife));
62             }
63         }
64     }
65
66     canHit = true;
67

```

Εικόνα 93 Τμήμα κώδικα της επίθεσης του ήρωα στο Goblin.

Η δύναμη των χτυπημάτων του ήρωα εξαρτάται από το όπλο που φέρει και συνεπώς από το αν έχει ολοκληρώσει τις αποστολές του δυαδικού συστήματος. Πιο συγκεκριμένα αν ο ήρωας χτυπά με τις γροθιές του θα αφαιρεί 10 μονάδες από τη ζωή του εχθρού, με το μαχαίρι 20 μονάδες, με το ρόπαλο 30 και τέλος με το τσεκούρι 50. Το πιο όπλο φέρει τη κάθε στιγμή ο ήρωας αποθηκεύεται στο Singleton από όπου και ανακτάται μέσω της `checkWeapon` (Εικόνα 94).

```
63
64 if (Input.GetKeyDown(KeyCode.Alpha3) && (maxWeapon >= 2))
65 {
66     currWeapon = 2;
67     selectWeapon(currWeapon);
68 }
69
70 if (Input.GetKeyDown(KeyCode.Alpha4) && (maxWeapon >= 3))
71 {
72     currWeapon = 3;
73     selectWeapon(currWeapon);
74 }
75
76 SingletonScript.Instance.curWeapon = currWeapon;
77
```

Εικόνα 94 Καταχώρηση του ενεργού όπλου στο Singleton από το `weaponChange` script.

```
69 public int checkWeapon()
70 {
71     if (SingletonScript.Instance.curWeapon == 0)
72         return 10;
73     else if (SingletonScript.Instance.curWeapon == 1)
74         return 20;
75     else if (SingletonScript.Instance.curWeapon == 2)
76         return 30;
77     else
78         return 50;
79 }
80
```

Εικόνα 95 Καθορισμός του επιπέδου ζημιάς που προκαλούν τα χτυπήματα στον εχθρό.

Τέλος η ζημιά που προκαλείται στο Goblin (Εικόνα 95) εμφανίζεται σε ένα νέο Text του καμβά και για το λόγο αυτό ο καμβάς πλέον δεν εξαφανίζεται μετά την προβολή του κάθε μηνύματος αλλά μένει πάντα ενεργός (Εικόνα 96). Αντίθετα εξαφανίζεται το Text αντικείμενο που έχει ολοκληρώσει τη προβολή του μηνύματος. Πιο συγκεκριμένα, όταν

το Goblin δέχεται ένα χτύπημα ενεργοποιείται το **Text** που εμφανίζει με πράσινο χρώμα το υπόλοιπο της ζωής του Goblin και μετά τη πάροδο τριών δευτερολέπτων εξαφανίζεται. Αντίστοιχη διαδικασία πραγματοποιείται και όταν δέχεται χτύπημα ο ήρωας του οποίου το υπόλοιπο εμφανίζεται σε ένα ξεχωριστό **Text** με κόκκινο όμως χρώμα.



Εικόνα 96 Ο καμβάς που εμφανίζει τα υπόλοιπα της ζωής των μαχόμενων.

Μόλις ο ήρωας εξαντλήσει τη ζωή του Goblin τότε εκτελείται το animation του θανάτου του Goblin, ο ήρωας ενημερώνεται και ξεκινά η επόμενη πίστα. Η λειτουργικότητα αυτή πραγματοποιείται από το GameManager αντικείμενο στο Script του οποίου στη συνάρτηση Update ελέγχονται τα επίπεδα της ζωής των χαρακτήρων (Εικόνα 97). Αν εξαντληθεί αυτό του Goblin το **Text** που εμφανίζει τη ζωή του εμφανίζει ένα μήνυμα και ενημερώνει τον παίκτη ότι κέρδισε, ενημερώνει το Singleton ότι η πίστα έχει ολοκληρωθεί και ενεργοποιεί την υπορουτίνα που θα εκτελέσει το animation. Μόλις αυτό ολοκληρωθεί θα εξαφανίσει το Goblin και θα ξεκινήσει τη νέα πίστα.

```

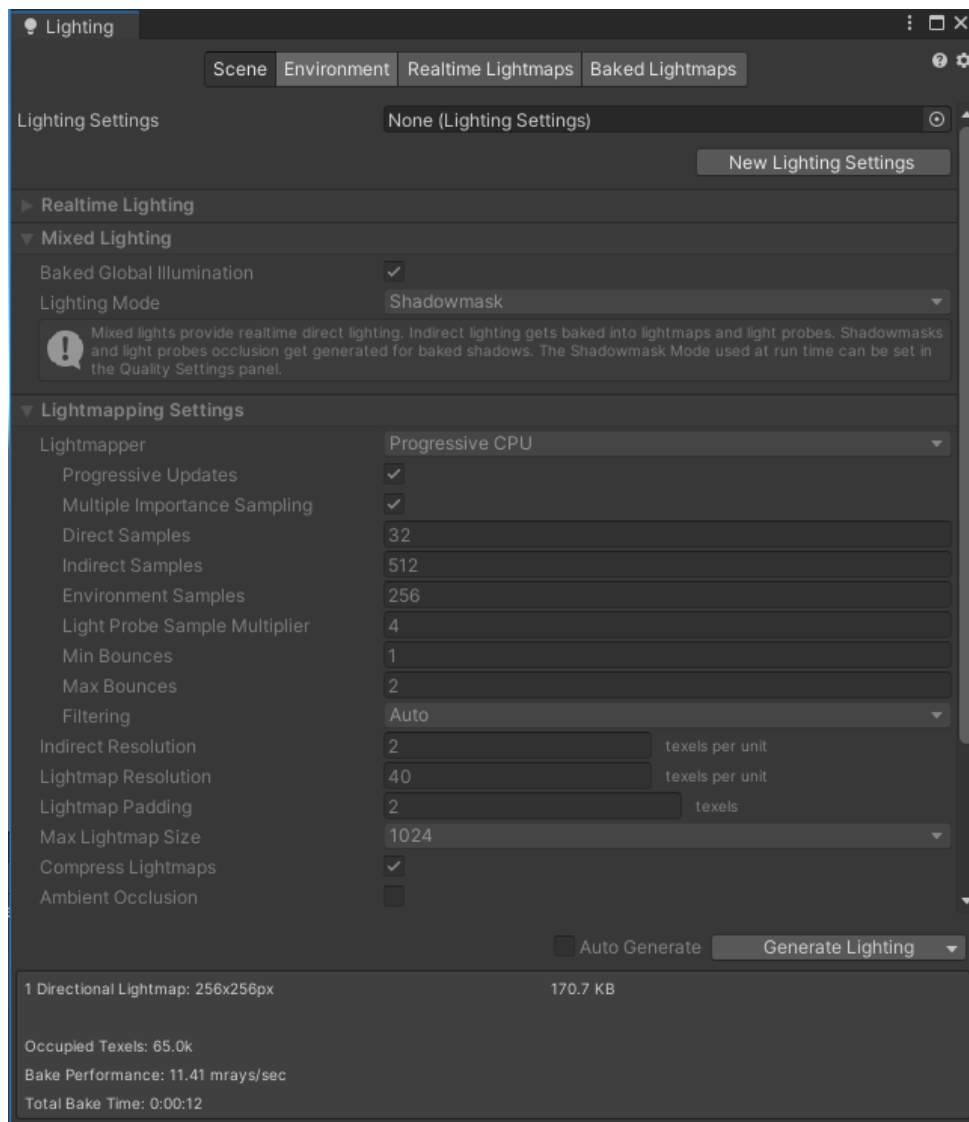
35 // Update is called once per frame
36 void Update()
37 {
38     if (heroLife <= 0)
39     {
40         heroLifeText.text = "Έχασες!!!";
41         heroLifeText.enabled = true;
42         Time.timeScale = 0;
43     }
44
45     if (goblinLife <= 0)
46     {
47         goblinLifeText.text = "Κέρδισες! Συνέχισε στην επόμενη πίστα.";
48         goblinLifeText.enabled = true;
49         SingletonScript.Instance.level++;
50         StartCoroutine(GoblinDie());
51     }
52 }
53
54 IEnumerator GoblinDie()
55 {
56     animator.SetBool("IsDead", true);
57     yield return new WaitForSeconds(5);
58     goblin.SetActive(false);
59     SceneManager.LoadScene("LevelB");
60 }

```

Εικόνα 97 Έλεγχος της πορείας της μάχης από το GameManager.

5.11. Σκοτεινές σκηνές

Ο προγραμματιστής θα παρατηρήσει ότι όταν φορτώνεται μια νέα σκηνή στη Unity, αυτή εμφανίζεται εμφανώς σκοτεινότερη από το φυσιολογικό. Ο λόγος για τον οποίο συμβαίνει αυτό είναι επειδή η Unity δεν διαθέτει τα απαραίτητα δεδομένα για το φωτισμό της σκηνής. Προκειμένου να αντιμετωπιστεί το φαινόμενο αυτό θα πρέπει ο προγραμματιστής, σε κάθε νέα σκηνή, να επιλέγει την καρτέλα **Window / Rendering / Lighting**, να απο-επιλέγει την επιλογή **Auto Generate** και ακολούθως να κάνει κλικ στο κουμπί **Generate Lighting** (Εικόνα 98).



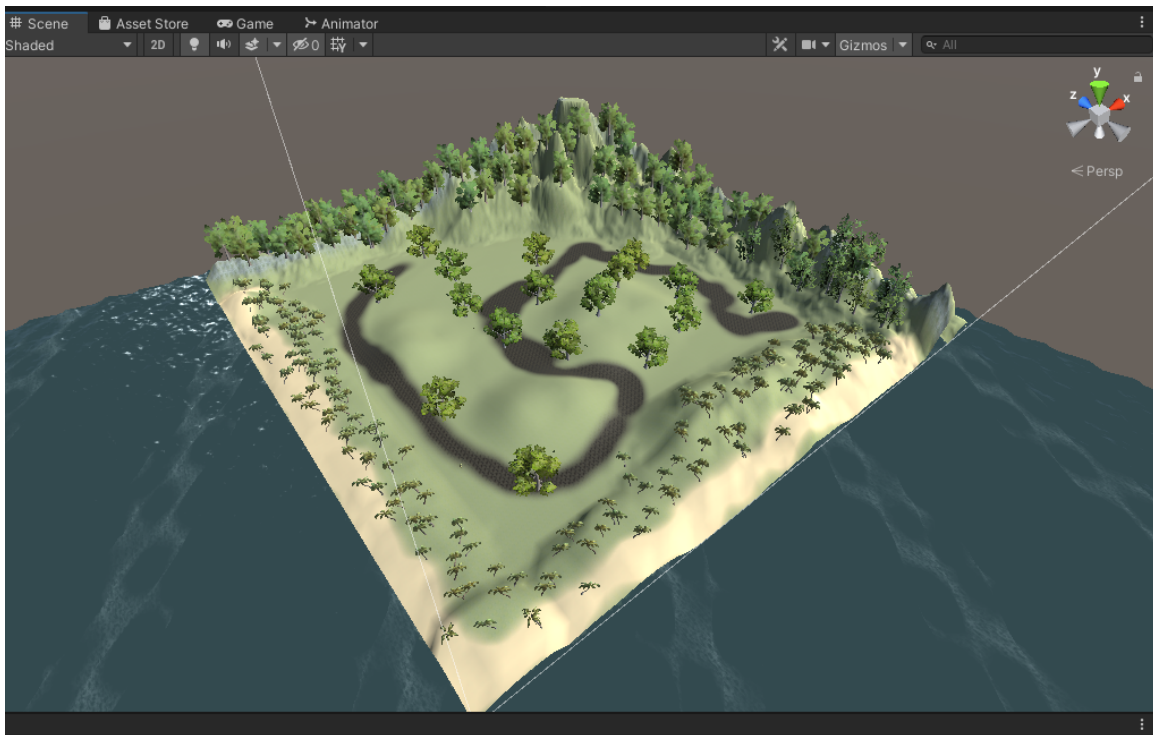
Εικόνα 98 Αντιμετώπιση του φαινομένου των σκοτεινών σκηνών.

5.12. Επίπεδο δεύτερο

5.12.1. Η σκηνή

Μόλις ο ήρωας αντιμετωπίσει με επιτυχία τον εχθρό τότε φορτώνεται η επόμενη πίστα (Εικόνα 99). Η πίστα αυτή περιέχει τέσσερις εχθρούς οι οποίοι φέρουν πλήρη εξάρτηση (όπλο και ασπίδα) και συνεπώς τα χτυπήματά τους είναι περισσότερο επιζήμια αλλά και λόγω της ασπίδας τα χτυπήματα που δέχονται δεν προκαλούν τόσο μεγάλη ζημιά. Η πίστα είναι διπλάσια σε έκταση από την αρχική και περιλαμβάνει δέντρα καθώς και αμμόλοφους οι οποίοι επιτρέπουν στους εχθρούς παραμένουν κρυμμένοι και να επιτίθενται στον ήρωα χωρίς αυτός να τους αντιληφτεί. Με τον τρόπο αυτό οι εχθροί

είναι ευκολότερο να αρχίσουν να χτυπούν τον ήρωα πρώτοι και συνεπώς να τον εξοντώσουν.



Εικόνα 99 Πίστα 2ου επιπέδου.

5.12.2 Διαχείριση πολλαπλών εχθρών

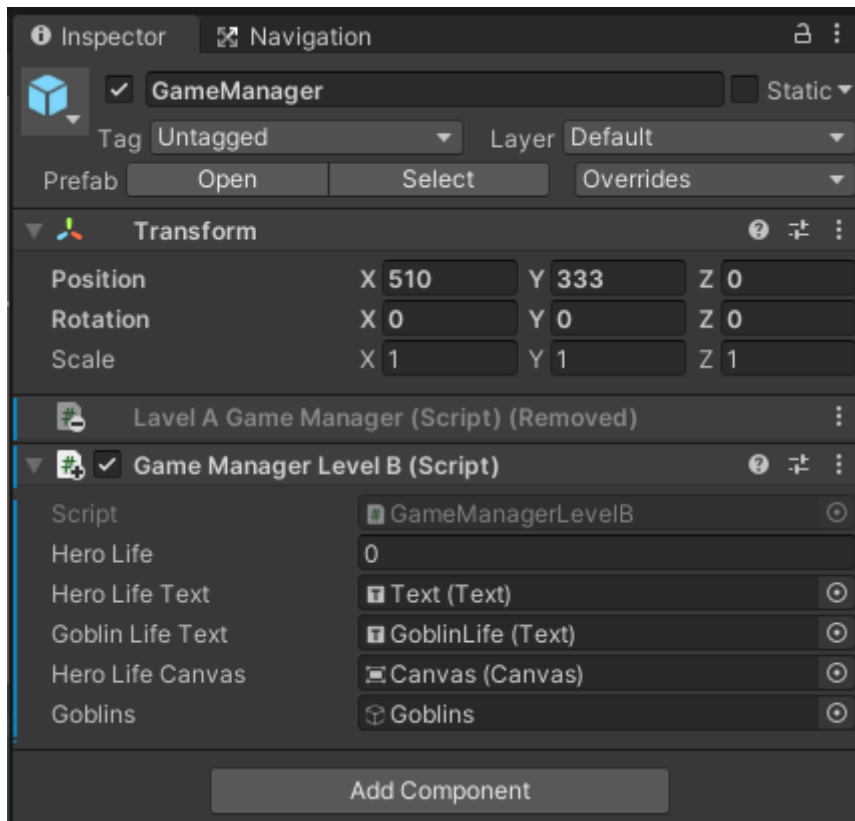
Στη πρώτη πίστα η ζωή του Goblin διατηρούνταν σε μια μεταβλητή η οποία φιλοξενούνταν στο αντικείμενο GameManager. Κάθε φορά που ο ήρωας χτύπαγε το Goblin αφαιρούσε ένα ποσοστό από τη ζωή του Goblin από το αντικείμενο GameManager. Τέλος αυτό, στην συνάρτηση Update, παρακολουθούσε τη ζωή του Goblin και μόλις αυτή εξαντλούνταν φόρτωνε την επόμενη πίστα. Η τεχνική αυτή στην αρχική πίστα ήταν ικανοποιητική, όμως δεν μπορούσε να εφαρμοστεί στην επόμενη όπου υπάρχουν περισσότερα του ενός Goblin. Ο λόγος για τον οποίο συμβαίνει αυτό είναι ότι θα πρέπει να αφαιρείται τμήμα της ζωής του Goblin το οποίο δέχεται το χτύπημα, συνεπώς η ζωή του Goblin δεν μπορεί να είναι στο κεντρικό GameManager.

Αντίθετα, στη παρούσα πίστα, η ζωή του κάθε Goblin διατηρείται σε μια μεταβλητή στο Script του ίδιου του Goblin και σε κάθε χτύπημα μειώνεται η μεταβλητή ζωή του ίδιου του αντικειμένου (Εικόνα 100). Στο GameManager διατηρείται το σύνολο των Goblin τα οποία υπάρχουν στη πίστα και κάθε φορά που κάποιο εξοντώνεται αφαιρείται από τον μετρητή. Στη συνάρτηση update παρακολουθείται το υπόλοιπο και μόλις τα Goblin εξαντληθούν η πίστα έχει ολοκληρωθεί και φορτώνεται η επόμενη.

```
49 IEnumerator hitFunction()  
50 {  
51  
52     canHit = false;  
53     anim.SetTrigger("Attack");  
54     yield return new WaitForSeconds(1);  
55  
56     if (Physics.Raycast(transform.position + new Vector3(0, 0, 0), transform.TransformDirection(Vector3.forward), out hit, 1))  
57     {  
58         hitObject = hit.collider.gameObject;  
59         int power = checkWeapon();  
60         hit.collider.gameObject.GetComponent<GoblinManagerLevelB>().goblinLife -= power;  
61         StartCoroutine(MessageDisplay("Η ζωή του Goblin είναι: " + hit.collider.gameObject.GetComponent<GoblinManagerLevelB>().goblinLife));  
62     }  
63  
64     canHit = true;  
65 }  
66
```

Εικόνα 100 Νέα συνάρτηση που υλοποιεί το χτύπημα του ήρωα στο Goblin ώστε να μειώνεται η μεταβλητή του ίδιου Script.

Για να επιτευχθεί η άνω λειτουργικότητα δημιουργήθηκαν νέα Script τα οποία ανατέθηκαν στα αντικείμενα του Goblin αλλά και του ήρωα τα οποία μεταφέρθηκαν ως prefab (Εικόνα 101) από την προηγούμενη πίστα. Πιο συγκεκριμένα στη νέα πίστα διαγράφηκε από το κάθε αντικείμενο το Script που είχε και προστέθηκε το νέο. Τα αντικείμενα, επειδή ακριβώς είναι prefab, μπορούν να δέχονται αλλαγές οι οποίες επηρεάζουν μόνο το συγκεκριμένο αντίγραφο του αντικειμένου. Τα αντικείμενα που έπρεπε να ανανεωθούν είναι το GameManager, το Goblin αλλά και του ήρωα.



Εικόνα 101 Το prefab GameManager με το νέο Script.

Ο αριθμός των Goblin δεν μπορεί να διατηρείται σαν μεταβλητή τύπου integer στο GameManager και ακολούθως από το Script του Goblin να μειώνεται όταν αυτό εξοντώνεται (Εικόνα 102). Πιο συγκεκριμένα στη συνάρτηση Update του Goblin ελέγχεται το αν η ζωή του έχει εξαντληθεί. Μόλις αυτό συμβεί θα μπορούσε να μειώνεται η τιμή του αριθμού των Goblin όμως η Update καλείται ανά frame συνεπώς η τιμή θα μειωθεί αρκετές φορές και όχι μόνο μία. Για να αντιμετωπισθεί το πρόβλημα αυτό δημιουργήθηκε ένα κενό αντικείμενο που θα φιλοξενεί όλα τα Goblines σαν παιδιά του. Μόλις ένα Goblin εξοντώνεται θα αφαιρείται από την ομάδα. Με τον τρόπο αυτό θα αφαιρείται μόνο ένα καθώς το ίδιο το Goblin από τη στιγμή που έχει αφαιρεθεί δεν επηρεάζει το αποτέλεσμα αν το ίδιο αφαιρεθεί εκ νέου. Ακολούθως στο GameManager ελέγχεται ο αριθμός των Goblin που υπάρχουν μέσα στο πατρικό αντικείμενο και όταν αυτά εξαντληθούν φορτώνει την επόμενη πίστα.


```

76     if (goblinLife <= 0)
77     {
78
79         StartCoroutine(GoblinDie());
80
81     }
82
83 }
84 IEnumerator GoblinDie()
85 {
86     animator.SetBool("IsDead", true);
87     yield return new WaitForSeconds(5);
88     heroLifeText.enabled = false;
89     gameObject.transform.parent = null;
90     gameObject.SetActive(false);
91
92 }
93

```

Εικόνα 102 Διαχείριση της ζωής των Goblin από το Script του κάθε Goblin.

Όπως θα παρατηρήσει ο αναγνώστης στη παραπάνω εικόνα στη συνάρτηση Update ελέγχεται το επίπεδο της ζωής του συγκεκριμένου Goblin. Μόλις αυτό εξαντληθεί ξεκινά η υπορουτίνα `GoblinDie` ώστε να υπάρχει ένας συγχρονισμός του **animation** που διαρκεί μερικά δευτερόλεπτα μέχρι και το θάνατο του Goblin. Μόλις το **animation** ολοκληρωθεί σταματά να εμφανίζεται το αντικείμενο **Text** το οποίο εμφανίζει τη ζωή του που ενεργοποιήθηκε όταν πραγματοποιήθηκε το χτύπημα του ήρωα και ακολούθως παύει να έχει πατρικό το Goblins. Τέλος απενεργοποιείται ώστε να εξαφανιστεί και από την πίστα.

```

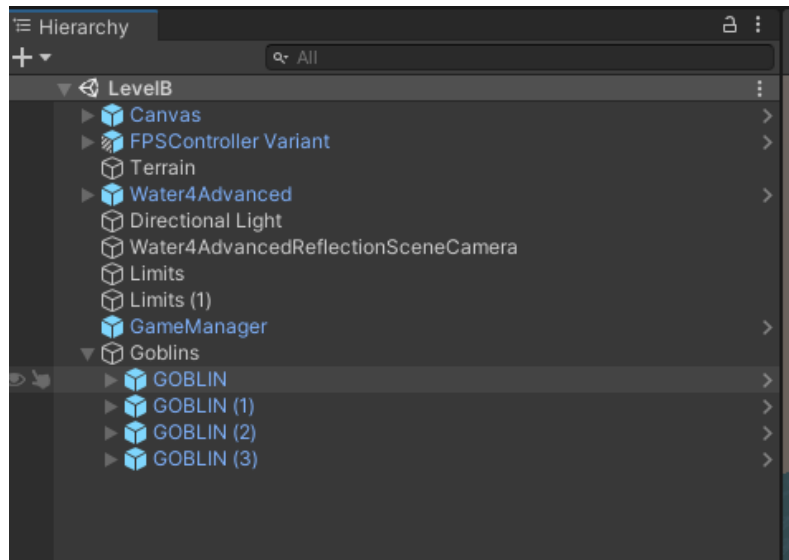
40
41
42     if(goblins.transform.childCount==0)
43     {
44         goblinLifeText.text = "Κέρδισες! Συνέχισε στην επόμενη πίστα.";
45         goblinLifeText.enabled = true;
46         SingletonScript.Instance.level++;
47     }
48

```

Εικόνα 103 Τμήμα κώδικα της Update του GameManager.

Το αντικείμενο `GameManager` περιέχει μια αναφορά τύπου `GameObject` με όνομα `goblins` η οποία αντιστοιχεί στο κενό αντικείμενο που περιέχει τα Goblin (Εικόνα 104). Μέσω της

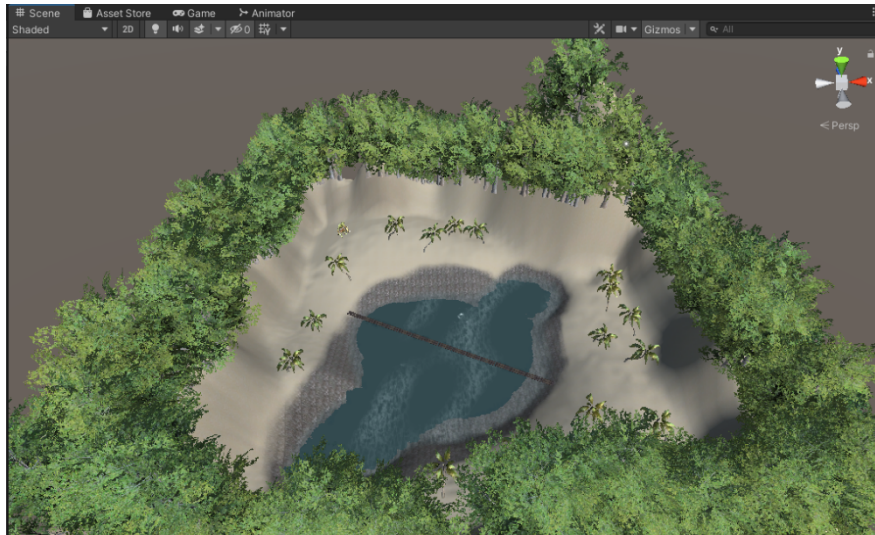
transform παρακολουθείται ο αριθμός των παιδιών και μόλις αυτά εξαντληθούν φορτώνεται η επόμενη πίστα (Εικόνα 103).



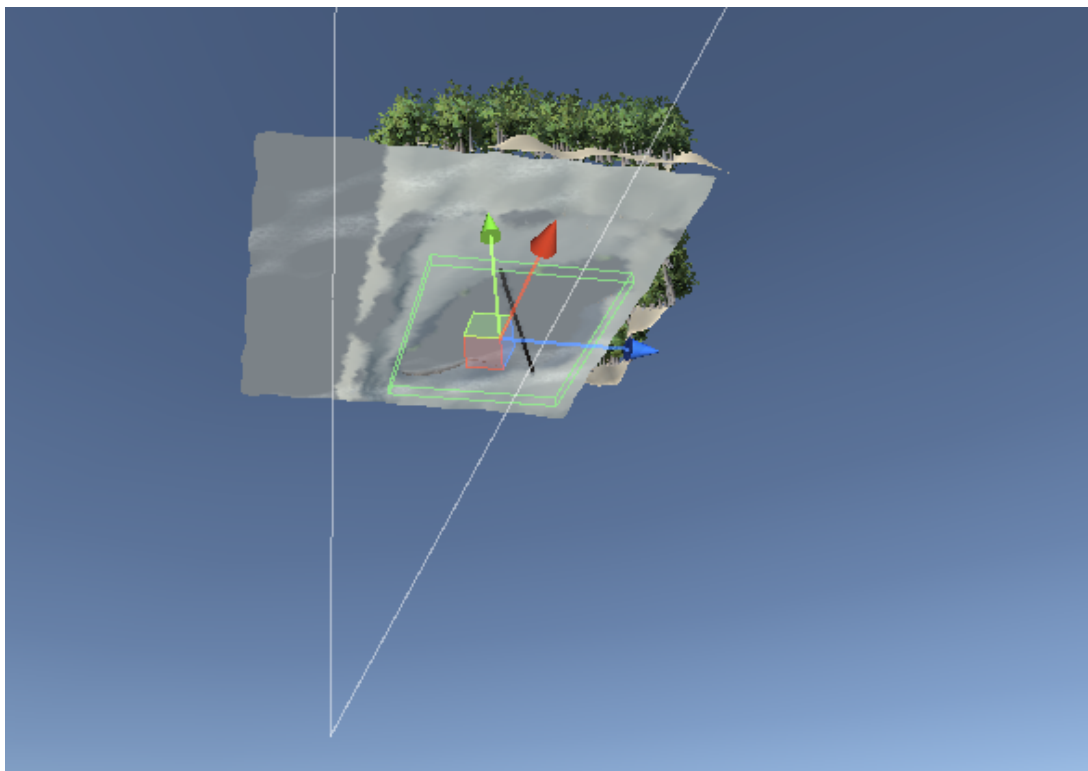
Εικόνα 104 Τα Goblings μέσα στο κενό αντικείμενο.

5.13. Επίπεδο τρίτο

Στη 3η και τελευταία πίστα ο ήρωας εμφανίζεται σε μία όαση στη μέση της οποίας υπάρχει μία λίμνη την οποία ο ήρωας θα πρέπει να διασχίσει μέσω μιας γέφυρας (Εικόνα 105). Αν πέσει από τη γέφυρα θα χάσει καθώς το νερό είναι ιδιαίτερα βαθύ και ο ήρωας θα πνιγεί. Για το σκοπό αυτό έχει δημιουργηθεί μια τρύπα στο **Terrain** στο σημείο του βυθού της λίμνης και κάτω από αυτό έχει δημιουργηθεί ένα κενό αντικείμενο το οποίο διαθέτει **Box Collider** διαστάσεων όσο το μέγεθος της λίμνης (Εικόνα 106). Ο **Collider** έχει ενεργοποιημένη την επιλογή **OnTrigger** και στο Script που του έχει ανατεθεί μηδενίζει τη ζωή του ήρωα.



Εικόνα 105 Πίστα του 3ου επιπέδου.



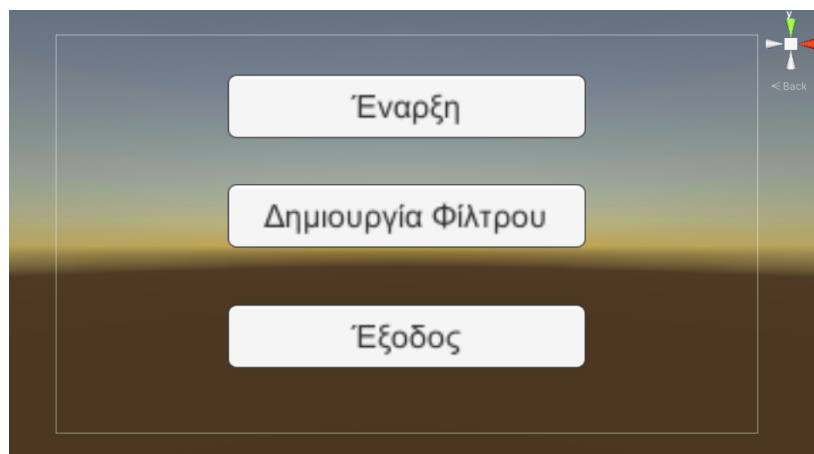
Εικόνα 106 Το κενό αντικείμενο στο κάτω μέρος της λίμνης.

Στη πίστα αυτή ο εχθρός ο οποίος εμφανίζεται στην αντίπερα όχθη της λίμνης είναι αισθητά μεγαλύτερος και δυσκολότερο αντιμετωπίσιμος. Συνεπώς για να καταφέρει να τον αντιμετωπίσει ο ήρωας θα πρέπει να έχει ολοκληρώσει τις αποστολές και να έχει αποκτήσει τα όπλα. Τέλος ο εχθρός δεν μπορεί να περάσει τη γέφυρα επιτρέποντας με

τον τρόπο αυτό των ήρωα να προετοιμαστεί για την επίθεση του καθώς και να έχει τη δυνατότητα να επιστρέψει στη βάση του ώστε να ανασυγκροτηθεί και να ξεκινήσει εκ νέου την επίθεση του.

5.14. Μενού και δισδιάστατα γραφικά πίστας

Πλέον απομένει να δημιουργηθεί μια σκηνή που θα παρουσιάζει το γραφικό περιβάλλον το οποίο θα εμφανίζεται στον παίχτη μόλις η εφαρμογή ξεκινήσει. Αυτό θα παρουσιάζει τρία κουμπιά όπου το πρώτο θα ξεκινά την πρώτη πίστα (Εικόνα 107), το δεύτερο θα εμφανίζει την πίστα παραγωγής των φίλτρων και το 3ο θα τερματίζει το παιχνίδι. Με τον τρόπο αυτό ο παίχτης θα μπορεί να επιλέξει είτε να ξεκινήσει το παιχνίδι απευθείας, χωρίς να δημιουργήσει φίλτρα το οποίο όμως θα έχει σαν συνέπεια να μην μπορεί να κερδίσει καθώς δεν μπορεί να αντιμετωπίσει τους εχθρούς χωρίς όπλα. Συνεπώς θα πρέπει πρώτα να δημιουργήσει τα φίλτρα απαντώντας στις ερωτήσεις μέσω της αντίστοιχης πίστας και ακολούθως να ξεκινήσει τις πίστες με τη μάχη.



Εικόνα 107 Η αρχική πίστα που εμφανίζει το Menu.

Για να εμφανίζεται σωστά ο κέρσορας στο γραφικό περιβάλλον θα πρέπει να δημιουργηθεί ένα Script και να ανατεθεί σε κάποιο αντικείμενο της πίστας, το οποίο στην Start θα πρέπει να ξεκλειδώνει τον κέρσορα και να τον κάνει ορατό (Εικόνα 108).

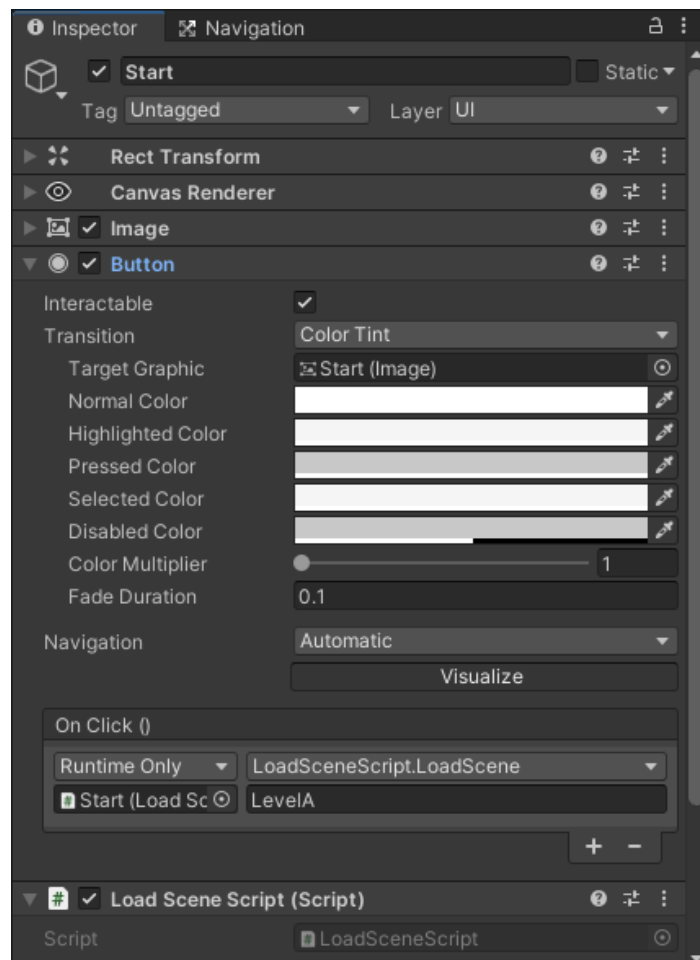
```

4
5 public class GuiScript : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     void Start()
9     {
10         Cursor.lockState = CursorLockMode.None;
11         Cursor.visible = true;
12     }
13

```

Εικόνα 108 Εμφάνιση του κέρσorra στη πίστα του GUI.

Τέλος για να λειτουργούν τα κουμπιά θα πρέπει να δημιουργηθεί ένα Script το οποίο να περιέχει τις συναρτήσεις που το κάθε κουμπί θα εκτελεί όταν θα πατιέται (Εικόνα 109). Το Script αυτό αφού ενσωματωθεί στο αντικείμενο ακολούθως πρέπει να επιλεγεί από το DropDown της **OnClick**. Συγκεκριμένα, για το κουμπί Έναρξη που ξεκινά τη πρώτη πίστα επιλέγεται από το DropDown το Script και η συνάρτηση LoadScene του Script ενώ σαν όρισμα δίνεται το όνομα της πίστας που θα εμφανίσει Εικόνα 110().



Εικόνα 109 Υλοποίηση λειτουργικότητας κουμπιών.

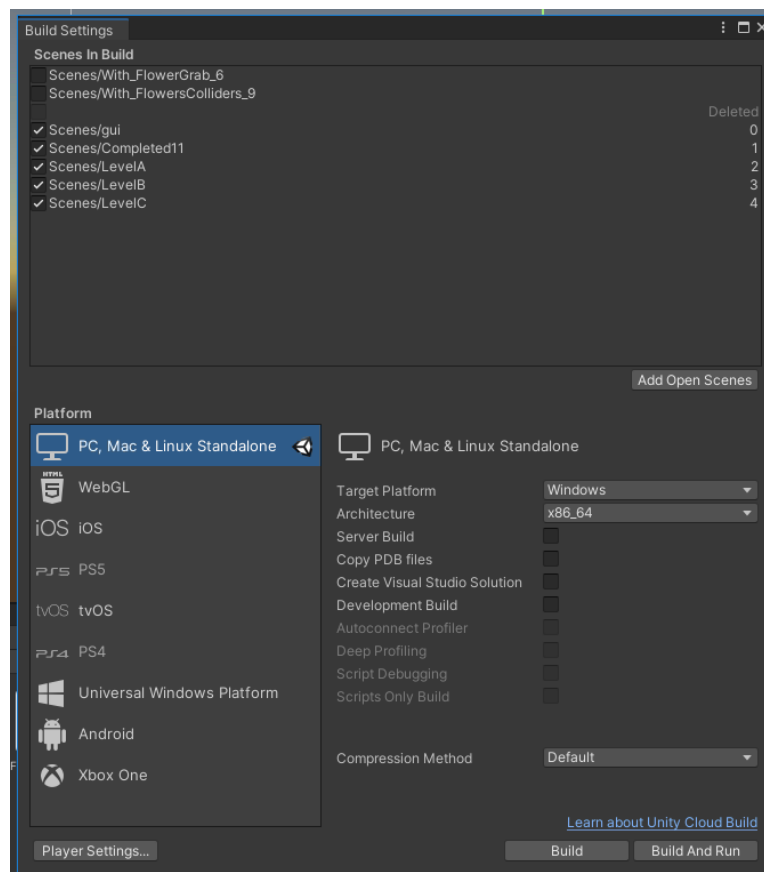
```

19
20 public void LoadScene(string sceneName)
21 {
22     SceneManager.LoadScene(sceneName);
23 }
24
25 public void QuitGame()
26 {
27     Application.Quit();
28 }
29
30 }

```

Εικόνα 110 Συναρτήσεις του LoadScene Script.

Πλέον απομένει το **Build** της εφαρμογής από την καρτέλα **File / Build Settings** (Εικόνα 111). Ο χρήστης θα πρέπει να επιλέξει τις σκηνές που επιθυμεί να ενσωματωθούν στη τελική έκδοση της εφαρμογής, την πλατφόρμα στην οποία η εφαρμογή θα εκτελείται, την ανάλυση της ή αν θα εκτελείται σε παράθυρο καθώς και άλλες λεπτομέρειες. Μόλις ολοκληρώσει την εισαγωγή παραμέτρων επιλέγει **Build** και η εφαρμογή τον ενημερώνει για τη διαδρομή που βρίσκονται τα εκτελέσιμα αρχεία της εφαρμογής. Η εφαρμογή είναι πλέον έτοιμη να διανεμηθεί στους χρήστες της.



Εικόνα 111 Χτίσιμο της εφαρμογής.

Κεφάλαιο 6 - Αξιολόγηση και Αποτελέσματα

6.1. Εισαγωγή

Για την εκτίμηση της αποδοτικότητας της προσέγγισης της παρούσας εργασίας όσον αφορά στην διδασκαλία του δυαδικού συστήματος, έγιναν πιλοτικές εφαρμογές του βιντεοπαιχνιδιού σε σχολικές αίθουσες με μαθητές των τελευταίων τάξεων του δημοτικού. Συγκεκριμένα, στο «πείραμα» έλαβαν μέρος δύο τάξεις (Πέμπτη και έκτη δημοτικού) του Αριστοτέλειου Ιδιωτικού Δημοτικού Σχολείου στο Μόναχο της Γερμανίας (Aristoteles Private Volksschule der Republik Griechenland) οι οποίες διδάχθηκαν το δυαδικό σύστημα μέσω του βιντεοπαιχνιδιού εικονικής πραγματικότητας. Μετά το πέρας του μαθήματος οι μαθητές κλήθηκαν να συμπληρώσουν ένα ερωτηματολόγιο σχετικό με την εμπειρία τους με την πλατφόρμα του παιχνιδιού. Το ίδιο ερωτηματολόγιο συμπλήρωσαν και οι δασκαλοι, ώστε να φανεί κατα πόσο είναι πρόθυμοι να χρησιμοποιήσουν το προτεινόμενο βιντεοπαιχνίδι στο μάθημά τους. Επιπλέον, οι μαθητές συμπλήρωσαν και ένα διαγώνισμα το οποίο έλεγχε τις γνώσεις που απέκτησαν πάνω στο δυαδικό σύστημα.

Για την ασφαλέστερη και ορθότερη ανάλυση των αποτελεσμάτων από στατιστικής άποψης δημιουργήθηκε και μία ομάδα ελέγχου (control group). Σε αυτήν την ομάδα μαθητών, το δυαδικό σύστημα αρίθμησης διδάχθηκε με την κλασσική μέθοδο: χωρίς το βιντεοπαιχνίδι - ο δάσκαλος παρέδιδε το μάθημα στον πίνακα, ενώ οι μαθητές κρατούσαν σημειώσεις. Η ομάδα ελέγχου αποτελούνταν από μαθητές της πέμπτης και έκτης δημοτικού, του ιδιωτικού δημοτικού σχολείου Αριζόνας Αμερικής (Three Hierarchs Academy of Arizona, US). Η ομάδα ελέγχου κλήθηκε στο τέλος του μαθήματος να γράψει ένα διαγώνισμα το οποίο αξιολογούσε την κατανόηση του δυαδικού συστήματος. Αυτό το διαγώνισμα ήταν ακριβώς το ίδιο με αυτό που συμπλήρωσε η πρώτη ομάδα μαθητών (που χρησιμοποίησε το βιντεοπαιχνίδι στο μάθημα). Σκοπός ήταν με την αντιπαραβολή των αποτελεσμάτων των δύο διαγωνισμάτων, να αξιολογηθεί η αποτελεσματικότητα της προτεινόμενης διδακτικής προσέγγισης.

Στη συνέχεια παρουσιάζεται το ερωτηματολόγιο που χρησιμοποιήθηκε για την αξιολόγηση της εμπειρίας του χρήστη και κατόπιν τα αποτελέσματα της έρευνας.

6.2. Ερωτηματολόγιο εμπειρίας χρήστη

6.2.1. Περιγραφή

Για την ανάλυση και διερεύνηση της εμπειρίας που είχαν οι νεαροί μαθητές με το παρουσιαζόμενο παιχνίδι εικονικής πραγματικότητας χρησιμοποιήθηκε το ερωτηματολόγιο UEQ (User Experience Questionnaire) του δρ. Μάρτιν Σρέππ (Martin Schrepp).

Η αρχική έκδοση του ερωτηματολογίου δημιουργήθηκε το 2005. Κάθε ερώτημα αποτελεί κατά την ορολογία της στατιστικής ένα σημασιολογικό διαφορικό (semantic differential). Με άλλα λόγια κάθε πτυχή του υπό εξέταση διαδραστικού προϊόντος εξετάζεται με δύο όρους με αντίθετες έννοιες σε μία κλίμακα από επτά στοιχεία (1 είναι το ένα άκρο της κλίμακας για τον έναν όρο, 7 είναι ο άλλος όρος και 4 είναι το αδιάφορο). Για παράδειγμα, η καινοτομία του προϊόντος αξιολογείται με τους όρους «συνηθισμένο» και «πρωτοπόρο» με την τιμή 1 να αποτελεί ένα απόλυτα συνηθισμένο προϊόν, ενώ το άλλο άκρο της κλίμακας, το 7 να αποδίδεται σε ένα εξαιρετικά πρωτότυπο προϊόν.

Μία σημαντική λεπτομέρεια του συγκεκριμένου ερωτηματολογίου είναι ότι δεν αποδίδεται πάντα στο θετικό χαρακτηριστικό η ακραία τιμή 7. Κάποιες φορές δίδεται τό αντίθετο άκρο, 1. Επίσης, υπάρχουν ερωτήσεις που εξετάζουν την ίδια παράμετρο με διαφορετικό τρόπο. Τα δύο παραπάνω στοιχεία του ερωτηματολογίου UEQ προσφέρουν την δυνατότητα ουσιαστικής ανάλυσης, φιλτράρωντας απαντήσεις στατιστικά μη-σημαντικές οι οποίες ενδεχομένως απαντήθηκαν τυχαία.

Οι πτυχές του προϊόντος οι οποίες αξιολογούνται από αυτό το ερωτηματολόγιο είναι οι εξής έξι (με 26 ερωτήσεις συνολικά):

- **Ελκυστικότητα:** Συνολική εντύπωση για το προϊόν. Αρέσει ή δεν αρέσει στους χρήστες το το προϊόν;
- **Ευκρίνεια:** Είναι εύκολο να εξοικειωθεί κανείς με το προϊόν; Είναι εύκολο να μάθει κανείς πώς να κάνει χρήση του προϊόντος;
- **Αποδοτικότητα:** Μπορούν οι χρήστες να επιλύσουν τις εργασίες τους χωρίς περιττή προσπάθεια;
- **Αξιοπιστία:** Αισθάνεται ο χρήστης ότι έχει τον έλεγχο της αλληλεπίδρασης;
- **Διέγερση:** Είναι συναρπαστική και παρακινητική η χρήση του προϊόντος;

- **Καινοτομία:** Είναι το προϊόν καινοτόμο και δημιουργικό; Το προϊόν τραβάει το ενδιαφέρον του το ενδιαφέρον των χρηστών;

Το ενδιαφέρον με το ερωτηματολόγιο UEQ είναι ότι παρέχει την δυνατότητα σύγκρισης του υπο εξέταση προϊόντος με μία πληθώρα άλλων διαδραστικών προϊόντων στην αγορά. Για το σκοπό αυτό, οι δημιουργοί του ερωτηματολογίου συγκέντρωσαν στοιχεία από 452 προϊόντα (με συνολικά 20.190 συμμετέχοντες στις έρευνες). Οι έρευνες αυτές επαναλαμβάνονται κάθε χρόνο, ώστε τα αποτελέσματα να είναι έγκυρα.

Με βάση αυτές τις έρευνες, το προϊόν μας, το βιντεο-παιχνίδι εν προκειμένω, δύναται να καταταγεί σε μία από τις επόμενες κατηγορίες:

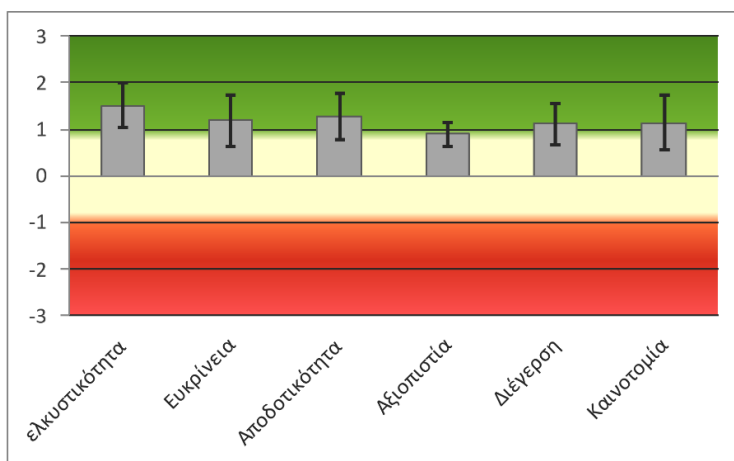
- **Εξαιρετικό:** Στο εύρος του 10% των καλύτερων αποτελεσμάτων.
- **Καλό:** Το 10% των αποτελεσμάτων στο σύνολο δεδομένων αναφοράς είναι καλύτερα και το 75% των αποτελεσμάτων είναι χειρότερα.
- **Πάνω από το μέσο όρο:** Το 25% των αποτελεσμάτων στο σύνολο αναφοράς είναι καλύτερα από το αποτέλεσμα για το αξιολογούμενο προϊόν, ενώ το 50% των αποτελεσμάτων είναι χειρότερα.
- **Κάτω του μέσου όρου:** Το 50% των αποτελεσμάτων στη συγκριτική αξιολόγηση είναι καλύτερα από το αποτέλεσμα για το αξιολογούμενο προϊόν, το 25% των αποτελεσμάτων είναι χειρότερα.
- **Κακή:** Στο εύρος του 25% των χειρότερων αποτελεσμάτων.

6.2.2. Αποτελέσματα

Όπως μπορεί να φανεί από το Γράφημα 1 με τα αποτελέσματα εμπειρίας χρήστη το παιχνίδι οι χρήστες βρήκαν αρκετά ελκυστικό το παιχνίδι («ελκυστικότητα»), προκλητικό των ενδιαφερόντός τους («διέγερση») και καινοτόμο («καινοτομία»). Επίσης, η άποψη των χρηστών για την «αποδοτικότητα» όσον αφορά στον στόχο του παιχνιδιού που είναι η διδασκαλία του δυαδικού συστήματος, κυμαίνεται στον μέσο όρο σε σχέση με παρόμοια προϊόντα.

Οι παράμετροι στις οποίες η αξιολόγηση έφερε μικρότερες τιμές ήταν η «ευκρίνεια» και «αξιοπιστία» του προϊόντος-παιχνιδιού. Αυτό σημαίνει ότι σε γενικές γραμμές, αν και το παιχνίδι φαίνεται διασκεδαστικό και αποτελεσματικό, ωστόσο έχει μία κάπως αμβλεία καμπύλη μάθησης και εξοικείωσης με το περιβάλλον του παιχνιδιού. Επιπλέον, το

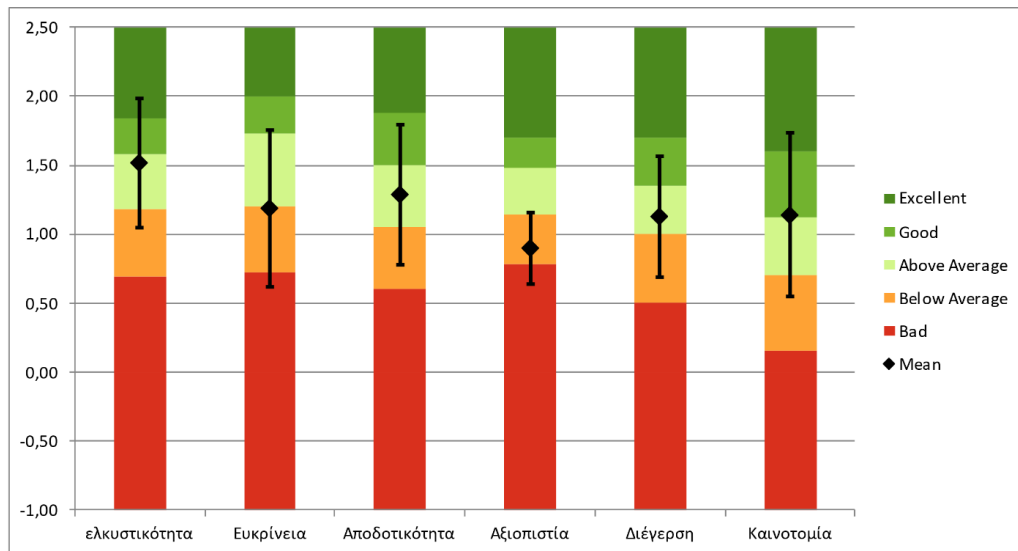
γεγονός ότι η «αξιοπιστία» δεν βρίσκεται στα ίδια επίπεδα με τις υπόλοιπες παραμέτρους υποδεικνύει ότι οι παίκτες αισθάνονταν σε μεγάλο βαθμό καθοδηγούμενοι κατά την ροή του παιχνιδιού. Αυτό βέβαια, είναι αναμενόμενο στα πλαίσια μιας εκπαιδευτικής διαδικασίας, όπου τα πράγματα πρέπει να έχουν κάποιο -έστω χαλαρό- πλαίσιο ροής εκτέλεσης των απαραίτητων διαδικασιών. Σκοπός του συγκερασμού της διδακτικής διαδικασίας με τα παίγνια ήταν απλώς η χαλάρωση αυτού του πλαισίου (όχι η εκμηδένιση) ώστε η μάθηση να είναι πιο ευχάριστη.



Γράφημα 1 αποτελέσματα κατά απόλυτη τιμή

Ενδιαφέρον έχει να συγκρίνουμε το παρόν βιντεοπαιχνίδι εικονικής πραγματικότητας με άλλα τεχνολογικά προϊόντα, δυνατότητα που μας παρέχεται από το ερωτηματολόγιο UEQ.

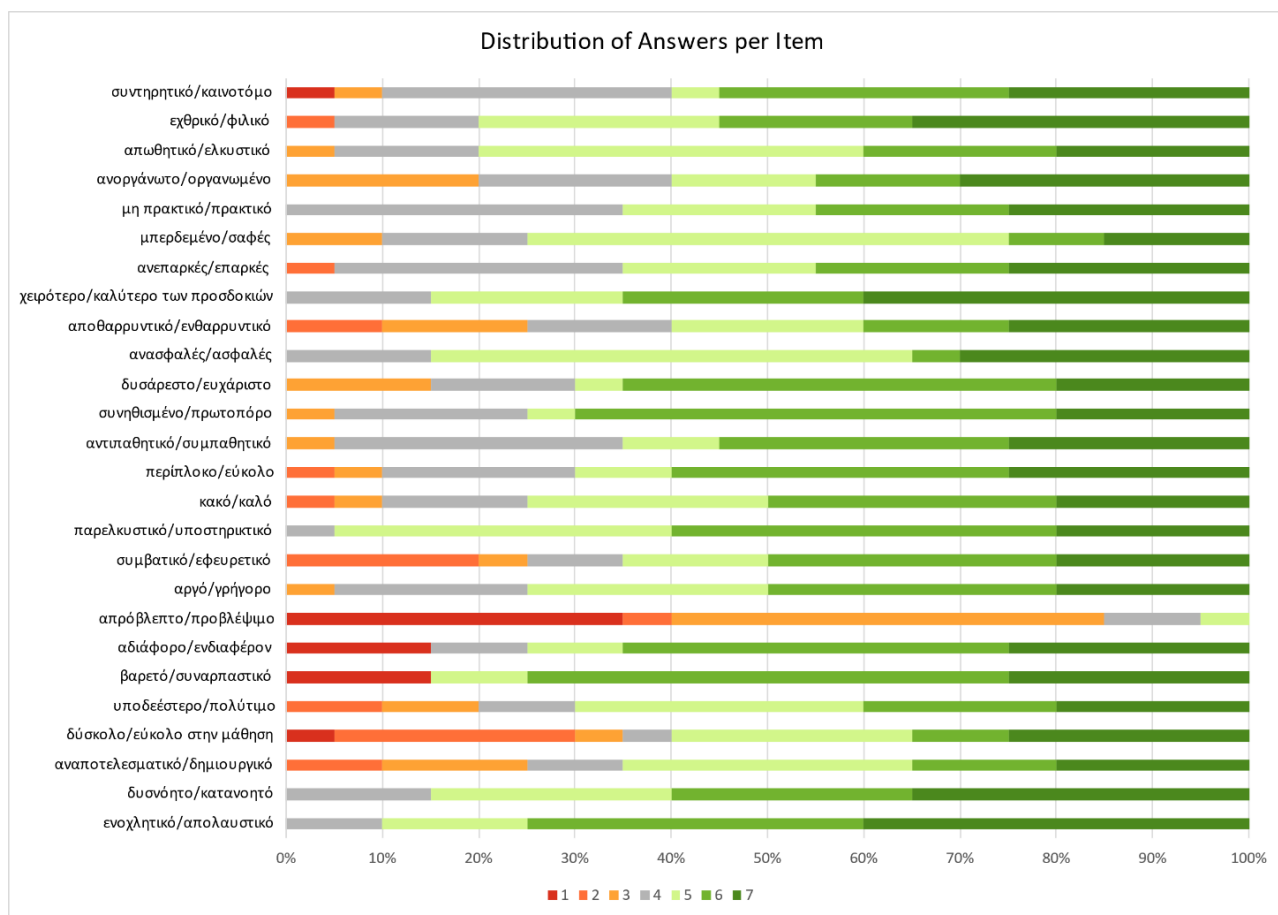
Στο Γράφημα 2 διαφαίνεται μία τάση για θετικές αξιολογήσεις όσον αφορά στην ελκυστικότητα και τις παραμέτρους ηδονικής ποιότητας (μη-στοχοστρεφείς παράμετροι) όπως είναι η διέγερση και καινοτομία. Οι παράμετροι ρεαλιστικής ποιότητας (στοχοστρεφείς παράμετροι), δηλαδή η ευκρίνεια, αποδοτικότητα και αξιοπιστία, βρίσκονται ως επί το πλείστον στο δεύτερο πεμπτημόριο (πορτοκαλί περιοχή) με μόνο την αξιοπιστία να βρίσκεται στο πρώτο πεμπτημόριο.



Γράφημα 2 αποτελέσματα σε σχέση με προϊόντα της αυτής αγοράς

Εδώ αξίζει να αναφέρουμε ότι δεδομένης της ποικιλίας των τεχνολογικών προϊόντων με τα οποία συγκρίθηκε το παρόν παιχνίδι και του εύρους της υφιστάμενης αγοράς παρόμοιων παιχνιδιών, τα παραπάνω αποτελέσματα αποτελούν μια πολύ ελπιδοφόρα απόδειξη ότι το προτεινόμενο βιντεοπαιχνίδι μπορεί να κερδίσει ένα ικανό μερίδιο στην αγορά των εκπαιδευτικών προϊόντων τεχνολογίας. Η παραπάνω διατύπωση γίνεται ακόμη πιο έγκυρη όταν αναλογιστούμε ότι είναι στατιστικά πιά σπάνια η απόδοση ακραίων τιμών (πράσινη και κόκκινη περιοχή στο Γράφημα 2) από τους χρήστες κατά τις έρευνες με ερωτηματολόγια.

Κλείνοντας αυτήν την παράγραφο, παραθέτουμε τις ερωτήσεις του ερωτηματολογίου UEQ μαζί με την κατανομή των απαντήσεων σε κάθε ερώτηση (Γράφημα 3).



Γράφημα 3 Αποτελέσματα ερωτηματολογίου εμπειρίας χρήστη

6.3. Ερωτηματολόγιο ελέγχου γνώσεων

6.3.1. Περιγραφή

Για την αξιολόγηση των αντικειμενικών γνώσεων που αποκτήθηκαν κατά την εκπαιδευτική διαδικασία δεν θα μπορούσε να λείπει και ένα σύντομο διαγώνισμα. Ωστόσο, δεδομένου ότι είχαμε δύο διαφορετικές ομάδες μαθητών (test και control groups) οι οποίοι διδάχθηκαν με διαφορετικό τρόπο την ύλη, κρίθηκε απαραίτητο να δημιουργηθούν δύο διαφορετικά διαγωνίσματα. Οι ερωτήσεις και στα δύο εξετάζουν το ίδιο ζήτημα, αλλά η αναπαράσταση των αριθμών διαφέρει. Έτσι, για τους μαθητές που δοκίμασαν το βιντεοπαιχνίδι εικονικής πραγματικότητας (test group), οι αριθμοί στο

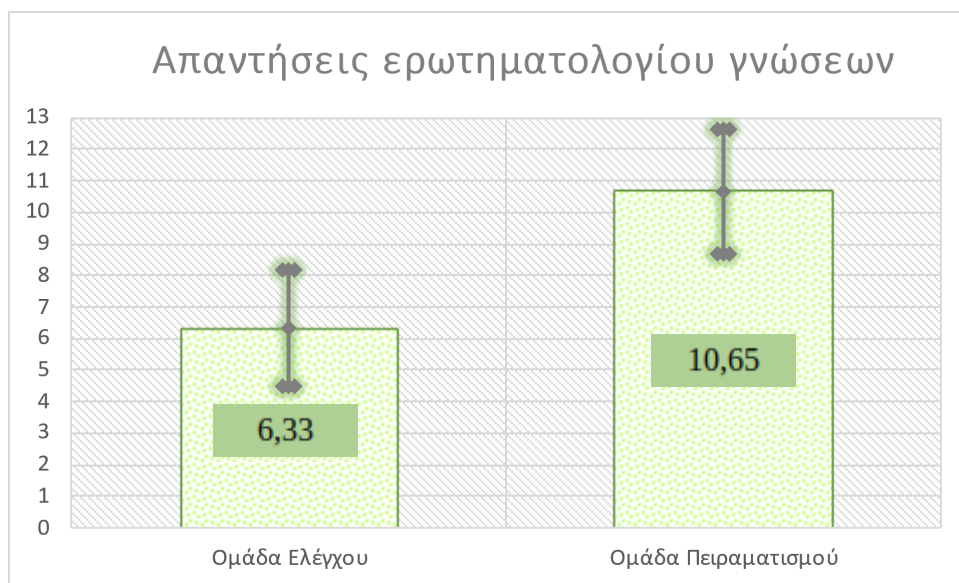
δυναμικό δίνονταν ως σειρά από καλάθια διαφορετικής αξίας/χωρητικότητας είτε άδεια ή με μήλα. Από την άλλη, για τους μαθητές της ομάδας ελέγχου (control group), οι αριθμοί δίνονταν με την κλασσική αναπαράσταση με τα ψηφία 0 και 1. Βέβαια, επειδή αυτή η ομάδα άκουσε έννοιες όπως μονάδα, δυάδα, λιγότερο σημαντικό ψηφίο, αξία θέσης κ.α. προστέθηκαν στο διαγώνισμα τέσσερις επιπλέον ερωτήσεις σωστού/λάθους όπου εξετάζεται η ικανότητα γλωσσικής κατανόησης των παραπάνω εννοιών και το τί αναπαριστούν στα μαθηματικά. Για παράδειγμα, υπάρχει η ερώτηση για το εάν το $(11)_2$ αναπαριστά τον αριθμό τρία. Μετά από μερικές ερωτήσεις όμως, ερωτάται πάλι η ίδια ερώτηση με διαφορετική διατύπωση -το $(11)_2$ περιγράφεται με λέξεις όπως μία μονάδα και μία δυάδα. Έτσι, μπορούμε να καταλάβουμε το σημείο αδυναμίας του μαθητή, εφόσον υπάρχει.

Σημαντικό στοιχείο, τέλος, της μεθόδου αξιολόγησης γνώσεων ήταν η επανάληψη του διαγώνισματος μετά από έναν και δύο μήνες, χωρίς να έχει προηγηθεί επανάληψη της διδασκαλίας. Σκοπός ήταν να εξαχθεί συμπέρασμα για την διατήρηση της γνώσης μακροπρόθεσμα και πώς η πρότασή μας για την παιγνιώδη μάθηση την επηρεάζει.

6.3.2. Αποτελέσματα

Από την ανάλυση των ερωτηματολογίων αντικειμενικών γνώσεων προέκυψαν δύο αποτελέσματα, αναφορικά προς τις δύο παραμέτρους που αξιολογήθηκαν: μέγεθος κατανόησης της ύλης και μακροπρόθεσμη διατήρηση της γνώσης.

Σχετικά με το μέγεθος της κατανόησης της διδαχθείσας ύλης, παρατηρήθηκε ξεκάθαρη βελτίωση της μαθηματικής ικανότητας κατά σχεδόν 68,2% στην δοκιμαστική ομάδα σε σχέση με την ομάδα ελέγχου. Με άλλα λόγια, η ομάδα ελέγχου απάντησε σωστά κατά μέσο όρο 6,33 από τις δώδεκα ερωτήσεις γνώσεων που είχε συνολικά το ερωτηματολόγιο (κατανεμημένες σε τρία ερωτήματα - χωρίς τις δημογραφικές ερωτήσεις), ενώ η ομάδα πειραματισμού κατάφερε το εντυπωσιακό μέσο αποτέλεσμα 10,65 στα δώδεκα (Γράφημα 4). Μάλιστα, σημειώθηκε σημαντική στατιστική διαφορά μεταξύ των δύο ομάδων.



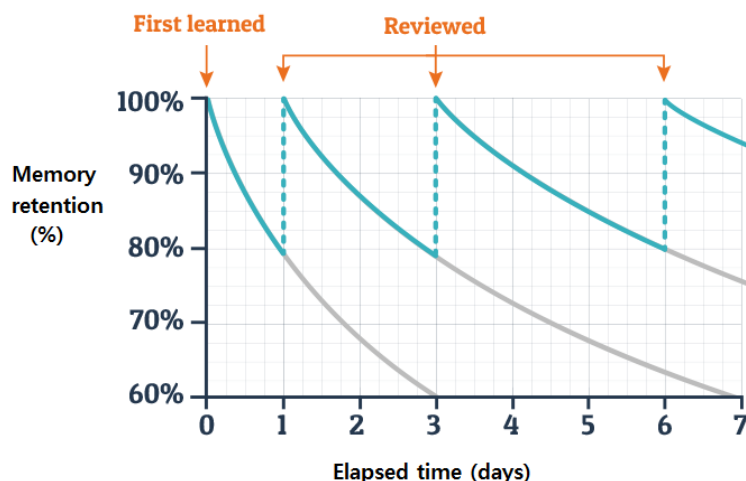
Γράφημα 4 Αποτελέσματα Ερωτηματολογίου γνώσεων

Συγκεκριμένα, παρατηρήθηκε ουσιαστικότερη κατανόηση της αξίας των θέσεων (βασικής έννοιας στα θεσιακά συστήματα αρίθμησης) καθώς και του τρόπου αναπαράστασης αριθμών. Το ενδιαφέρον είναι ότι στα παραπάνω οι μαθητές της πειραματικής ομάδας (test group) οι οποίοι δεν είχαν στέρεες γνώσεις μαθηματικών και ούτε ενδιαφέρονταν ιδιαίτερα για αυτά (όπως συμπεραίνεται από τις ερωτήσεις του ερωτηματολογίου 5 και 2, αντίστοιχα) έφεραν καλύτερα αποτελέσματα ακόμη και από τους μαθητές με κάποιο σχετικό μαθηματικό υπόβαθρο. Αναλύοντας σε ακόμη μεγαλύτερο βάθος την ομάδα μαθητών του test group όπου δεν είχαν μαθηματική ικανότητα και αρέσκεια στην αριθμητική, αλλά έφεραν αξιόλογα αποτελέσματα, βλέπουμε ότι οι μαθητές που ανήκουν στο 85ο ποσοστημόριο, χαρακτηρίζονταν από φιλομάθεια για τον τρόπο λειτουργίας των ηλ. υπολογιστών (ερώτηση 1 του ερωτηματολογίου). Το κομμάτι στο οποίο η πειραματική ομάδα υστερούσε ελάχιστα σε κατανόηση σε σχέση με την ομάδα ελέγχου ήταν αυτό των αριθμητικών πράξεων. Η διαφορά όμως αυτή παρατηρήθηκε μόνο για τους μαθητές που είχαν τις καλύτερες επιδόσεις μέσα στο γκρούπ τους, ενώ για τους υπολοίπους δεν υπήρχε αξιόλογη στατιστική διαφορά.

Από τα προηγούμενα, συνάγεται το συμπέρασμα ότι αφενός μεν η διδασκαλία με το προτεινόμενο παιχνίδι εικονικής πραγματικότητας είχε καταφανή και “χειροπιαστά” αποτελέσματα στην βελτίωση των αντικειμενικών γνώσεων στο δυαδικό σύστημα, αφετέρου δε ότι σημαντικός παράγοντας για την εγκόλπωση των διδασκόμενων εννοιών αποτελεί η ανάπτυξη ενός υποστηρικτικού περιβάλλοντος το οποίο θα εξάπτει και προκαλεί τα παιδιά αισθητηριακά (όπως είναι τα περιβάλλοντα εικονικής πραγματικότητας) και θα τα ελκύει, καλλιεργώντας παράλληλα και την φαντασία τους,

χαρακτηριστικά που η υποβοηθούμενη από την τεχνολογία εκπαίδευση δύναται απλόχερα να προσφέρει.

Εδώ, να σημειώσουμε ότι η αξιολόγηση των γνώσεων επαναλήφθηκε μετά από ένα και δύο μήνες, για την ομάδα ελέγχου μόλις που έφτασε το 15% του αρχικού σκόρ μετά από ένα μήνα και το 12% μετά από δύο μήνες. Το γεγονός ότι τα ποσοστά αυτά είναι παρόμοια δίνει μία επιπλέον πληροφορία: ότι η απώλεια της ουσιαστικής γνώσης είχε συμβεί ήδη πριν τον πρώτο μήνα μετά την διδασκαλία. Με άλλα λόγια, όσον αφορά στην κατακράτηση της πληροφορίας, οι μαθητές του τέστ γκρούπ είχαν απόκριση η οποία ακολουθεί την καμπύλη “κατακράτησης πληροφορίας” για τον μέσο μαθητή όπως φαίνεται στο Γράφημα 5, καθ’ όσα μπορούμε να συμπαιράνουμε από την ανάλυση των αποτελεσμάτων μας. Έτσι, η επανάληψη του μαθήματος είναι σχεδόν απαραίτητη για την διατήρηση της γνώσης. Η χρήση της εικονικής πραγματικότητας στην παρούσα εργασία στόχο έχει την προσέγγιση ενός είδους βιωματικής μάθησης η οποία θα ενίσχυε ακόμη περισσότερο την κατακράτηση πληροφορίας για μεγαλύτερο χρονικό διάστημα, χωρίς ενδιάμεσες επαναλήψεις. Ακόμη βέβαια και όταν χρειαστεί αυτή η επανάληψη, αυτή δεν φαίνεται να είναι πρόβλημα για τους νεαρούς μαθητές αφού αυτή έχει την μορφή παιχνιδιού και επιπλέον μπορεί να βασιστεί σε κάποιο μοντέλο αντεστραμμένης τάξης (flipped classroom model) ενισχυοντας ακόμη περισσότερο την διατήρηση γνώσης [36].



Γράφημα 5 Καμπύλη κατακράτησης γνωστικών αντικειμένων για τον μέσο μαθητή, Πηγή: https://www.researchgate.net/figure/Ebbinghaus-forgetting-curve-and-review-cycle_fig1_324816198

Τέλος, να αναφέρουμε ότι αναλύοντας συνδυαστικά τα ερωτηματολόγια εμπειρίας χρήστη και αξιολόγησης γνώσεων, παρατηρήθηκε ότι οι μαθητές τους οποίους το UEQ χαρακτήριζε ως αμφιλεγόμενης εγκυρότητας μέσω των μετρικών του (επειδή για παράδειγμα συμπλήρωσαν στην τύχη το ερωτηματολόγιο, βαρέθηκαν ή δεν έδωσαν προσοχή), ήταν αυτοί με το χαμηλότερο σκόρ στο αντίστοιχο ερωτηματολόγιο γνώσεων. Δηλαδή αυτοί οι μαθητές, γενικότερα, δεν είχαν διάθεση να συμμετάσχουν

ενεργά στο πείραμα και οι απαντήσεις τους ίσως και να μπορούσαν να παραληφθούν τελείως. Παρόλα αυτά κρατήθηκαν, γιατί σε μία τάξη υπάρχει όντως αυτή η ποικιλία μαθητών οπότε τα αποτελέσματα αφενός προσέγγιζαν καλύτερα την σχολική πραγματικότητα, αφετέρου μπορούν να συγκριθούν και με άλλες έρευνες που ενδεχομένους να μην είναι τόσο ενδεδειγμένες στον διαχωρισμό και “κατηγοριοποίηση” των συμμετεχόντων.

Κεφάλαιο 7 - Συμπεράσματα και Επίλογος

7.1. Κατακλείδα

Στόχος της παρούσας πτυχιακής εργασίας αποτέλεσε η διερεύνηση των δυνατοτήτων της μηχανής γραφικών Unity μέσα από την υλοποίηση μιας εφαρμογής εκπαιδευτικού χαρακτήρα. Πιο συγκεκριμένα, στα πλαίσια της παρούσης πτυχιακής εργασίας αναπτύχθηκε μια εφαρμογή 3D τύπου First Person Shooter η οποία διδάσκει στους μαθητές του Γυμνασίου το δυαδικό σύστημα. Η εφαρμογή περιλαμβάνει μία πίστα στην οποία ο μαθητής εξασκεί τις γνώσεις του στην αναπαράσταση αριθμών στο δυαδικό σύστημα σχηματίζοντας στο δυαδικό σύστημα τους αριθμούς του δεκαδικού που θα του ζητηθούν. Οι αριθμοί του δυαδικού σχηματίζονται τοποθετώντας λουλούδια σε κανάτες όπου η άδεια κανάτα συμβολίζει το 0 στο δυαδικό ενώ η κανάτα που περιέχει λουλούδι το 1 στο δυαδικό. Η διαδικασία παρουσιάζεται στον παίχτη ως η παρασκευή μαγικών φίλτρων τοποθετώντας βότανα στις κανάτες. Καθώς ο παίχτης απαντά σε όλο και πιο προχωρημένες ερωτήσεις αποκτά σταδιακά πρόσβαση σε όπλα τα οποία αυξάνουν τη ζημιά που προκαλούν στον εχθρό και συνεπώς τις πιθανότητες ολοκλήρωσης του παιχνιδιού. Η εφαρμογή περιλαμβάνει άλλες τρεις πίστες όπου στη πρώτη υπάρχει ένας εχθρός ο οποίος δεν φέρει οπλισμό και ασπίδα, συνεπώς αντιμετωπίζεται εύκολα, στην επόμενη υπάρχουν τέσσερις πλήρως εξοπλισμένοι εχθροί όπου για τους αντιμετωπίσει ο ήρωας θα πρέπει να έχει αποκτήσει το σύνολο των όπλων. Τέλος στη 3η και τελευταία πίστα υπάρχει ένας μόνο εχθρός ο οποίος όμως είναι αισθητά μεγαλύτερος από τους υπολοίπους προκαλώντας με τον τρόπο αυτό την εντύπωση ότι είναι πιο δύσκολο να αντιμετωπισθεί.

Προκειμένου ο αναγνώστης να αποκτήσει μια πιο πλήρη εικόνα του χώρου, στο πρώτο και εισαγωγικό κεφάλαιο παρουσιάζονται περιληπτικά οι πλέον καθιερωμένες μηχανές γραφικών της περιόδου συγγραφής της παρούσας πτυχιακής εργασίας. Από τις μηχανές αυτές μεταξύ άλλων παρουσιάζονται τα βασικά τους χαρακτηριστικά καθώς και ο λόγος για τον οποίο η Unity επιλέχθηκε για τη συγκεκριμένη παρουσίαση. Στη συνέχεια, στο επόμενο κεφάλαιο παρουσιάζονται οι τεχνικές διδασκαλίας στις οποίες η παρούσα εργασία έχει στηριχθεί. Ακολούθως αναλύεται η υλοποίηση της εφαρμογής ώστε ο αναγνώστης να αποκτήσει μια όσο το δυνατόν πληρέστερη εικόνα τόσο της ίδιας της μηχανής αλλά και του τρόπου με τον οποίο τα παιχνίδια αυτού του τύπου έχουν υλοποιηθεί.

Συμπερασματικά αξίζει να αναφερθεί ότι η Unity άξια φέρει τον τίτλο της μηχανής που επιτρέπει τη γρήγορη ανάπτυξη ικανοποιητικού επιπέδου εφαρμογών ακόμη και από ένα άτομο που δεν διαθέτει μεγάλη εμπειρία στο χώρο. Στο διαδίκτυο μπορείς κανείς να εντοπίσει εύκολα αρκετή υποστήριξη και να αντιμετωπίσει τις όποιες δυσκόλεues συναντήσει. Πέρα από υποστήριξη, ο προγραμματιστής θα εντοπίσει εύκολα πληθώρα

μοντέλων – έτοιμων αντικειμένων που μπορεί να εισάγει στη Unity και να τα χρησιμοποιήσει, χωρίς να απαιτείται να τα δημιουργήσει εξ αρχής. Το περιβάλλον εργασίας είναι αρκετά φιλικό προς το χρήστη ενώ διαθέτει πληθώρα εργαλείων, όπως animators, εισαγωγή μοντέλων, χρωματισμούς αντικειμένων κ.α. τα οποία θα διευκολύνουν ιδιαίτερα την εργασία του προγραμματιστή. Τέλος η C# που χρησιμοποιεί είναι ιδιαίτερα διαδεδομένη αλλά και εύκολη στη χρήση γεγονός που επιτρέπει τη χρήση της μηχανής χωρίς να απαιτείται η εκ νέου εκμάθηση μιας νέας γλώσσας προγραμματισμού.

7.2. Μελλοντικές επεκτάσεις

Για λόγους εξέτασης των γνώσεων που αποκομίσθηκαν από το παιχνίδι και για μεγαλύτερη ποικιλία, θα μπορούσαν κάποιες παρασκευές φίλτρων να έχουν ήδη γεμάτα κάποια καζάνια και ο παίκτης να καλείται να απαντήσει ποιά είναι η ποσότητα του μίγματος. Αυτό απαιτεί ευχέρεια στην μετατροπή από το δυαδικό σύστημα, στο δεκαδικό.

Εκτός από την εκμάθηση της αναπαράστασης ποσοτήτων στο δυαδικό σύστημα, ο χρήστης εξασκείται επίσης στην πρόσθεση και τον πολλαπλασιασμό δύο αριθμών στο δυαδικό σύστημα. Για το παιχνίδι πρόσθεσης, παρουσιάζονται δύο παράλληλες σειρές αγγείων με τα λουλούδια ήδη τοποθετημένα και ο παίκτης καλείται να βάλει τα λουλούδια κάθε αγγείου στο καζάνι που βρίσκεται κάτω από αυτό. Εάν δύο λουλούδια τοποθετηθούν στο ίδιο καζάνι, ο παίκτης πρέπει να τα μετακινήσει γρήγορα στο επόμενο καζάνι, αλλιώς το παρασκεύασμα θα εκραγεί.

Περαιτέρω βελτίωση της αποδοτικότητας του παιχνιδιού θα μπορούσε να επιτευχθεί με την συνεισφορά της τεχνητής νοημοσύνης, και πιο συγκεκριμένα της εκμάθησης μηχανής (machine learning). Τα αντίστοιχα μίνι-παιχνίδια θα μπορούσαν να καθορίζονται από αλγόριθμους που παρακολουθούν την πρόοδο του χρήστη και αξιολογούν την απόδοσή του. Έτσι, η τεχνητή νοημοσύνη θα αναλάμβανε ένα πολύ σημαντικό κομμάτι στην διαδικασία της μάθησης, του οποίου η αξία εσχάτως άρχισε να διαφαίνεται, αυτό της μετα-γνωστικής λειτουργίας (meta-cognition)[37], [38], [39].

Επιπλέον βελτιώσεις θα μπορούσαν να γίνουν προκειμένου να αντιμετωπιστεί η αδυναμία που διαφάνηκε κατά την αξιολόγηση των ερωτηματολογίων εμπειρίας χρήστη. Συγκεκριμένα οι παίκτες φάνηκε να δυσκολεύονται να καταλάβουν τις οδηγίες του παιχνιδιού και φαινόταν σαν να μην έχουν αυτοί τον έλεγχο του παιχνιδιού. Για να ξεπεραστούν αυτές οι δυσκολίες θα μπορούσε να επανασχεδιαστεί η πίστα παρασκευής φίλτρων σύμφωνα με τα παρακάτω.

Στην συνταγή του φίλτρου δίνεται συγκεκριμένος αριθμός λουλουδιών και αφού ο παίκτης τα συλλέξει από τον κήπο του πρέπει να τα τοποθετήσει στις κανάτες

προσέχοντας έναν και μοναδικό κανόνα, καμία κανάτα δεν μπορεί να μείνει μισο-γεμάτη. Οι κανάτες πρέπει να είναι είτε άδειες ή τελείως γεμάτες. Αυτός ο κανόνας εμποδίζει τον χρήστη από το να σχηματίσει έναν αριθμό στο δυαδικό λανθασμένα, κάτι που στην παρούσα έκδοση του παιχνιδιού είναι δυνατόν να γίνει, αλλά τότε δεν θα εμφανιστεί το μήνυμα επιτυχίας στην οθόνη. Έτσι, ο παίκτης ενδεχομένως αναρωτιέται εάν έκανε κάτι λάθος και μπορεί να αρχίσει να τοποθετεί τυχαία τα λουλούδια κάνοντας δοκιμές μέχρι να πετύχει την απάντηση. Βλέπουμε δηλαδή, ένα ξεκάθαρο σφάλμα στην επικοινωνία των οδηγιών στον χρήστη και αυτό υπάρχει περίπτωση να καθυστερήσει αρκετά την εκμάθηση της σκοπούμενης θεωρίας. Στην ίδια κατεύθυνση βελτίωσης της ευκρίνειας του παιχνιδιού, οι διάλογοι με τις οδηγίες θα μπορούσαν να εμπλουτιστούν. Επίσης, κάτι που θα βοηθούσε τον παίκτη να αντιληφθεί γρηγορότερα ότι έκανε κάποιο λάθος θα ήταν οι κανάτες να κοκκινίζουν και να πάλλονται ωσάν να πρόκειται να εκραγούν (ενώ στην παρούσα υλοποίηση τα άνθη απλώς πέφτουν έξω από τις κανάτες. Επιπλέον, εάν το μέγεθος των κανατών αντιστοιχούσε στην χωρητικότητά τους (που είναι δύναμη του δύο), θα γινόταν ακόμα πιο αισθητή η έννοια της αξίας θέσης στην αναπαράσταση των αριθμών.

Κλείνοντας θα μπορούσε κανείς να εντοπίσει πληθώρα προσθηκών που θα μπορούσαν να ενσωματωθούν στην εφαρμογή ώστε να γίνει περισσότερο δελεαστική στους χρήστες αλλά και να διδάσκει περισσότερη ύλη στους μαθητές. Όπως άλλωστε συμβαίνει με όλα τα έργα της πληροφορικής όπου το μοναδικό όριο αποτελεί η φαντασία, έτσι και στη συγκεκριμένη παρουσίαση θα μπορούσαν να δημιουργηθούν επιπλέον αποστολές για την μετατροπή μεταξύ δεκαδικού και δυαδικού καθώς και περισσότερες πίστες με εχθρούς. Με τον τρόπο αυτό οι μαθητές θα ανακαλύψουν ότι το δυαδικό σύστημα δεν είναι και τόσο δύσκολο καθώς και πόσο χρήσιμο είναι στην επιστήμη των υπολογιστών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Corti, K. (2006). Games-based Learning; a serious business application. PIXELearning Limited. [www.pixelearning.com/docs/games_basedlearning_pixelearning.pdf]
- [2] Baker, J. M., Martin, T., Aghababayan, A., Armaghanyan, A., & Gillam, R. (2015). Cortical activations during a computer-based fraction learning game: Preliminary results from a pilot study. *Technology, Knowledge and Learning*, 20(3), 339–355
- [3] Gaggi, O. & Petenazzi, G. (2019). A digital platform for teaching mathematics. In *EAI International Conference on Smart Objects and Technologies for Social Good (GoodTechs '19)*, September 25–27, 2019, Valencia, Spain. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3342428.3342666>
- [4] Mayo, M.J. (2009). Video games: A route to large-scale STEM education? *Science* 323(5910) 79–82. <https://doi.org/10.1126/science.1166900>
- [5] Pelton, T. & Francis Pelton, L. (2011). Design Principles for Making Meaningful Mathematics Apps. In M. Koehler & P. Mishra (Eds.), *Proceedings of SITE 2011--Society for Information Technology & Teacher Education International Conference* (pp. 2199-2204). Nashville, Tennessee, USA: Association for the Advancement of Computing in Education (AACE). Retrieved from <https://www.learntechlib.org/p/36631> (accessed May 2021)
- [6] Kyfonidis, C., Moumoutzis, N., Christodoulakis, S.: Block-C: A block-based programming teaching tool to facilitate introductory C programming courses. *2017 IEEE Global Engineering Education Conference (EDUCON)*, 2017, pp. 570-579.
- [7] Moumoutzis, N., Boukeas, G., Vassilakis, V., Pappas, N., Xanthaki, C., Maragkoudakis, I., Deligiannakis, A., Christodoulakis, S.: Design, Implementation and Evaluation of a Computer Science Teacher Training Programme for Learning and Teaching of Python Inside and Outside School. In *Interactive Mobile Communication, Technologies and Learning*, 2017, pp. 575-586
- [8] Papert, S.: *Mindstorms*. New York: Basic Books, 1980.
- [9] Papert, S.: *The children's machine: rethinking school in the age of the computer*, 1993.
- [10] Cabitza, Federico. (2014). De-designing the IT artifact. Drafting small narratives for the coming of the Socio-Technical Artifact.
- [11] Cabitza, F., Fogli, D., & Piccinno, A. (2014). Cultivating a Culture of Participation for the Co-Evolution of Users and Systems. *CoPDA@ AVI*, (pp. 1-6).
- [12] Cabitza, F., & Simone, C. (2015). Building socially embedded technologies: Implications about design. In *Designing Socially Embedded Technologies in the Real-world* (pp. 217-270). London: Springer.
- [13] Laurel, B. (2013). *Computers as theatre*. Addison-Wesley. (pp. 94)
- [14] Dirksmeier, P., & Helbrecht, I. (2008). Time, non-representational theory and the 'Performa-tive Turn' – Towards a new methodology in qualitative social research. *Forum Qualitative Social Research*, 9(2), Art. 55. (pp. 5)

- [15] Piraha people : <https://www.spiegel.de/international/spiegel/brazil-s-piraha-tribe-living-without-number-s-or-time-a-414291.html> (accessed Mai 6, 2021)
- [16] Numeral system : https://en.wikipedia.org/wiki/Numeral_system/ (accessed Jul. 25, 2023).
- [17] Sign-value notation : https://en.wikipedia.org/wiki/Sign-value_notation/ (accessed Jul. 25, 2023).
- [18] Location arithmetic : https://en.wikipedia.org/wiki/Location_arithmetic (accessed Jul. 27, 2023).
- [19] William Bricken Ph.D.: Iconic Arithmetic Volume I: The Design of Mathematics for Human Understanding. Unary press, 2019.
- [20] Friberg, J.: Three thousand years of sexagesimal numbers in Mesopotamian mathematical texts. Arch. Hist. Exact Sci. 73, 183–216 (2019). <https://doi.org/10.1007/s00407-019-00221-3> (accessed Jul. 20, 2023)
- [21] Positional notation : https://en.wikipedia.org/wiki/Positional_notation/ (accessed Jul. 30, 2023).
- [22] Decimal : <https://en.wikipedia.org/wiki/Decimal/> (accessed Jul. 30, 2023).
- [23] Addition - Wikipedia : <https://en.wikipedia.org/wiki/Addition/> (accessed Jul. 30, 2023).
- [24] Cisco Binary Game: <https://learningnetwork.cisco.com/s/binary-game> (accessed Dec. 10, 2020).
- [25] Binary Bonanza: <https://games.penjee.com/binary-bonanza/> (accessed Dec. 10, 2020).
- [26] Binary Dots: https://raft.net/idea_sheets/binary-dots/ (accessed Dec. 10, 2020).
- [27] Unreal. Unreal Engine. Ανακτήθηκε 10 Ιουνίου 2022 από: <https://www.unrealengine.com/en-US>
- [28] Cryengine. About. Ανακτήθηκε 10 Ιουνίου 2022 από: <https://www.cryengine.com>
- [29] Unity. Unity Real-Time Development Platform. Ανακτήθηκε 14 Ιουνίου 2022 από: <https://www.unity.com>
- [30] <http://www.blendswap.com/blends/view/42830>
- [31] Astrofish Games - Asset Store (unity.com)
- [32] chlyang - Asset Store (unity.com)
- [33] <https://assetstore.unity.com/publishers/16743>
- [34] <https://assetstore.unity.com/publishers/35660>
- [35] Goblin | Characters | Unity Asset Store
- [36] Chun, Bo Ae & hae ja, Heo. (2018). The effect of flipped learning on academic performance as an innovative method for overcoming ebbinghaus' forgetting curve. 56-60. DOI: 10.1145/3178158.3178206.
- [37] Zeynep Çiğdem Özcan (2016) The relationship between mathematical problem-solving skills and self-regulated learning through homework behaviours,

motivation, and metacognition, *International Journal of Mathematical Education in Science and Technology*, 47:3, 408-420, DOI: 10.1080/0020739X.2015.1080313

- [38] Muhammad Maqsd (1998) Effects of metacognitive instruction on mathematics achievement and attitude towards mathematics of low mathematics achievers, *Educational Research*, 40:2, 237-243, DOI: 10.1080/0013188980400210
- [39] Stephanie M. Lemley, Jessica T. Ivy, Dana Pomykal Franz & Seth F. Oppenheimer (2019) Metacognition and Middle Grade Mathematics Teachers: Supporting Productive Struggle, *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 92:1-2, 15-22, DOI: 10.1080/00098655.2018.1547264

Παράρτημα Α: Ερωτηματολόγιο Εμπειρίας Χρήστη – UEQ

	1	2	3	4	5	6	7		
ενοχλητικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	απολαυστικό	1
δυσνόητο	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	κατανοητό	2
δημιουργικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	αναποτελεσματικό	3
εύκολο στη μάθηση	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	δύσκολο στη μάθηση	4
πολύτιμο	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	υποδεέστερο	5
βαρετό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	συναρπαστικό	6
αδιάφορο	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ενδιαφέρον	7
απρόβλεπτο	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	προβλέψιμο	8
γρήγορο	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	αργό	9
εφευρετικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	συμβατικό	10
παρελκυστικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	υποστηρικτικό	11
καλό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	κακό	12
περίπλοκο	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	εύκολο	13
αντιπαθητικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	συμπαθητικό	14
συνηθισμένο	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	πρωτοπόρο	15
δυσάρεστο	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ευχάριστο	16
ασφαλές	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ανασφαλές	17
ενθαρρυντικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	αποθαρρυντικό	18
ανταποκρίνεται στις προσδοκίες	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	δεν ανταποκρίνεται στις προσδοκίες	19
ανεπαρκές	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	επαρκές	20
σαφές	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	μπερδεμένο	21
μη πρακτικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	πρακτικό	22
οργανωμένο	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	ανοργάνωτο	23
ελκυστικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	απωθητικό	24
φιλικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	εχθρικό	25
συντηρητικό	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	καινοτόμο	26

Παράρτημα Β: Ερωτηματολόγια Αξιολόγησης Αντικειμενικών Γνώσεων

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΑΞΙΟΛΟΓΗΣΗΣ ΔΙΔΑΚΤΙΚΗΣ ΜΕΘΟΔΟΥ

(για μαθητές στο control group)

Ηλικία:

Φύλλο: αγόρι , κορίτσι

Ερώτηση 1

Σου αρέσει να μαθαίνεις για το πώς λειτουργούν οι υπολογιστές; ναι , όχι

Ερώτηση 2

Σου αρέσει να κάνεις αριθμητικές πράξεις; ναι , όχι

Ερώτηση 3

Έχεις ξανα-ακούσει για το δυαδικό σύστημα; ναι , Όχι

Ερώτηση 4

Δώσε ένα παράδειγμα πού χρησιμοποιείται το δυαδικό σύστημα:

Ερώτηση 5

Υπολογίστε τις παρακάτω πράξεις:

α. $2 \cdot 2 =$

β. $2 \cdot 2 \cdot 2 =$

γ. $2 \cdot 2 \cdot 2 \cdot 2 =$

δ. $4 \cdot 2 =$

ε. $8 \cdot 2 =$

στ. $4 \cdot 4 =$

Ερώτηση 6

Αντιστοιχίστε τα κουτάκια με τις χωρητικότητές τους με βάση την αναπαράταση αριθμών στο δυαδικό σύστημα. Το πιο αριστερό κουτάκι είναι το πιο μικρό.



1.

2

3.

4.

.

A.

B.

Γ.

Δ.

Οκτάδες

Μονάδες

Δυάδες

Τετράδες

Ερώτηση 7

Γράψτε **Σ** εάν η πρόταση είναι σωστή και **Λ** εάν είναι λάθος.

- | | |
|--|-------|
| α. Το ένα στο δυαδικό σύστημα γράφεται ως 0. | |
| β. Το 11 στο δυαδικό είναι ο αριθμός τρία. | |
| γ. $0010 + 0001 = 0011$ | |
| δ. $0011 + 0001 = 0111$ | |
| ε. Το 0111 είναι μεγαλύτερο απ το 1000. | |
| στ. Μία μονάδα και μία δυάδα μας κάνει το τρία. | |
| ζ. Μία οκτάδα είναι ένα σύνολο από έξι αντικείμενα. | |
| η. Η πιο αριστερή θέση έχει αξία χιλιάδων στο δυαδικό σύστημα. | |

ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΑΞΙΟΛΟΓΗΣΗΣ ΔΙΔΑΚΤΙΚΗΣ ΜΕΘΟΔΟΥ

(για μαθητές στο test group)

Ηλικία:

Φύλλο: αγόρι , κορίτσι

Ερώτηση 1

Σου αρέσει να μαθαίνεις για το πώς λειτουργούν οι υπολογιστές; ναι , όχι

Ερώτηση 2

Σου αρέσει να κάνεις αριθμητικές πράξεις; ναι , όχι

Ερώτηση 3

Έχεις ξανα-ακούσει για το δυαδικό σύστημα; ναι , Όχι

Ερώτηση 4

Δώσε ένα παράδειγμα πού χρησιμοποιείται το δυαδικό σύστημα:

Ερώτηση 5

Υπολογίστε τις παρακάτω πράξεις:

α. $2 \cdot 2 =$

β. $2 \cdot 2 \cdot 2 =$

γ. $2 \cdot 2 \cdot 2 \cdot 2 =$

δ. $4 \cdot 2 =$

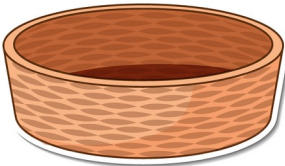
ε. $8 \cdot 2 =$

στ. $4 \cdot 4 =$

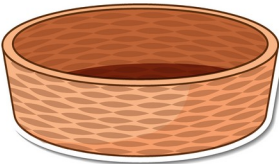
Ερώτηση 6

Τα καλάθια στο σχήμα μπορούν να έχουν τόσες μπαλίτσες όσες χωράνε και σε κάθε θέση ενός δυαδικού αριθμού.

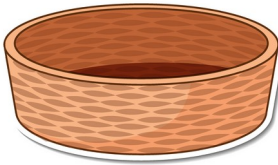
Πόσες μπαλίτσες χωράνε σε κάθε καλάθι; Αντιστοιχίστε!



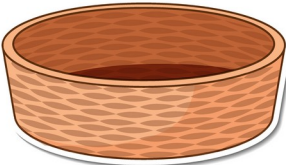
1.



2
.




3.




4.


A.




B.



Γ.

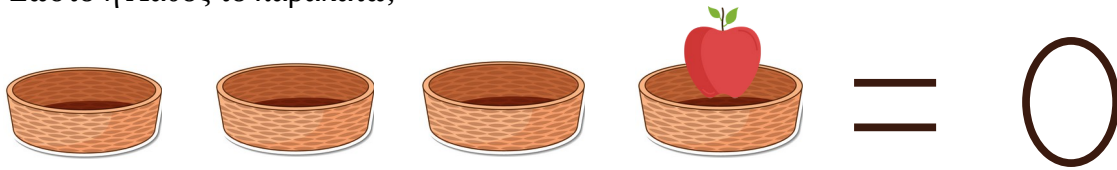


Δ.

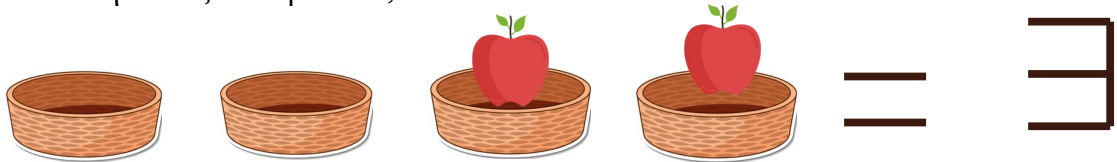


Ερώτηση 7

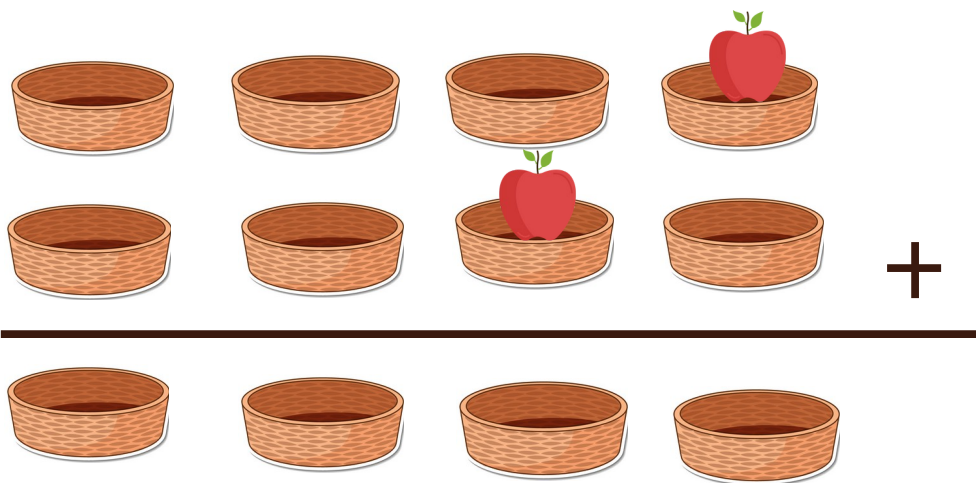
α. Σωστό ή Λάθος το παρακάτω;



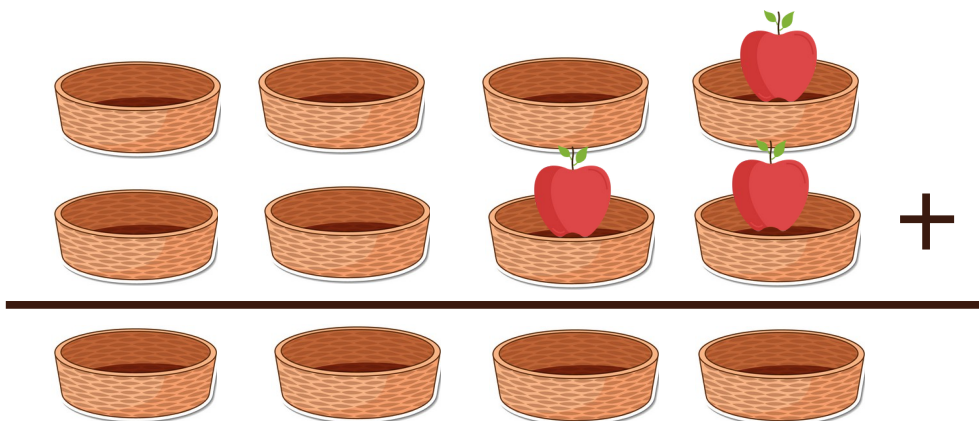
β. Σωστό ή Λάθος το παρακάτω;



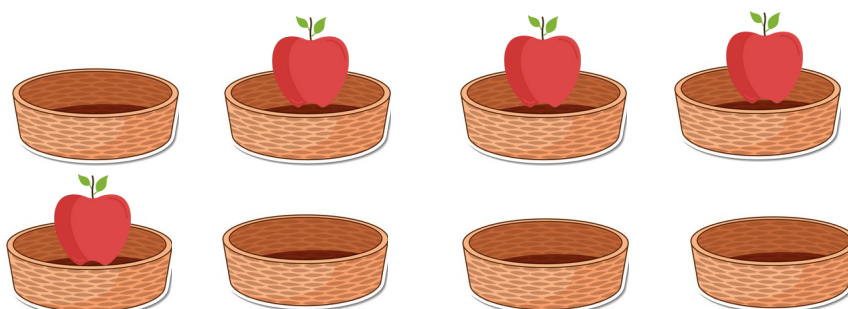
γ. Γράψτε κατευθείαν το αποτέλεσμα της πράξης παρακάτω:



δ. Γράψτε κατευθείαν το αποτέλεσμα της πράξης παρακάτω:



ε. Ποιά σειρά από καλάθια αναπαριστά τον μεγαλύτερο αριθμό μήλων;



ΕΡΩΤΗΜΑΤΟΛΟΓΙΟ ΑΞΙΟΛΟΓΗΣΗΣ ΔΙΔΑΚΤΙΚΗΣ ΜΕΘΟΔΟΥ

(για εκπαιδευτικούς)

Ερώτηση 1

Έχετε ξαναδιδάξει μαθηματικά σε παιδιά γυμνασίου;

Ερώτηση 2

Σε κλίμακα του επτά, πόσο όσο συχνά χρησιμοποιείτε ψηφιακά μέσα κατά την εκπαιδευτική διαδικασία;

(1- ποτέ , 7 - καθημερινά)

Ερώτηση 3

Σε κλίμακα του επτά, πόσο καλά γνωρίζετε το δυαδικό σύστημα αρίθμησης

(1- δεν το έχω ξανα-ακούσει , 7 - άριστα)

Ερώτηση 4

Πόσο πιθανόν είναι να χρησιμοποιούσατε το προτεινόμενο βιντεο-παιχνίδι στην τάξη, δεδομένου ότι σας παρέχεται ο εξοπλισμός;

(1 – καθόλου πιθανό , 7 – σίγουρο)

Ερώτηση 5

Κατα πόσο θεωρείτε ελκυστικό για τα παιδιά το βιντεοπαιχνίδι από άποψη γραφικών και πλοκής.

(1 – καθόλου ευχάριστο , 7 – πολύ ευχάριστο)

Ερώτηση 6

Κατα την γνώμη σας, πόσο πετυχαίνει τον στόχο της διδασκαλίας του δυαδικού συστήματος το προτεινόμενο βιντεο-παιχνίδι;

(1 – καθόλου , 7 – απόλυτα)

Ερώτηση 7

Πόσο αποτελεσματική θεωρείτε την χρήση εικονικής πραγματικότητας στο συγκεκριμένο βιντεοπαιχνίδι;

(1 – καθόλου αποτελεσματική, 7 – εξαιρετικά αποτελεσματική)

Πλάνο διδασκαλίας του δυαδικού συστήματος αρίθμησης σε μία διδακτική ώρα

1. Μονάδες, δεκάδες, εκατοντάδες, χιλιάδες (τί είναι;)	2´
2. Θα μπορούσαμε να ομαδοποιούμε σε μονάδες, δυάδες, τετράδες, οκτάδες...;	2´
3. Ανάλυση δεκαδικού συστήματος: αξία θέσης	7´
4. Αντιπαραβολή δεκαδικού με δυαδικό όσον αφορά την αξία θέσης	7´
5. Υπόδειγμα καλαθιού – μπαλών	13´
6. Εξάσκηση: πότε έχουμε περισσότερα αντικείμενα στα καλάθια συνολικά (1000 ή 0111 κ.α.)	13´
7. Παιχνίδι πρόσθεσης: βάζουμε μπάλες στο πρώτο καλάθι εκτελώντας αντίστοιχες υπερχειλίσεις όταν απαιτείται	10´
8. Μαθηματικός φορμαλισμός	10´