



ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΠΡΟΤΥΠΟΥ ΓΙΑ 3D POINT AND CLICK
ADVENTURE ΠΑΙΧΝΙΔΙΑ ΣΤΗ UNITY

Διπλωματική Εργασία

του

Θεοφάνη Σ. Αγραφιώτη

(Α.Μ. 22001)

Θεσσαλονίκη, Οκτώβριος 2023

ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΦΑΡΜΟΓΗ ΠΡΟΤΥΠΟΥ ΓΙΑ 3D POINT AND CLICK
ADVENTURE ΠΑΙΧΝΙΔΙΩΝ ΣΤΗ UNITY

Θεοφάνης Σ. Αγραφιώτης

Πτυχίο Οικονομικών Επιστημών, Πανεπιστήμιο Μακεδονίας, 2022

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ
ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής
Ξυνόγαλος Στυλιανός - Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30/10/2023

Ξυνόγαλος
Στυλιανός

Σαρατζέμη
Μάγια

Χατζηγεωργίου
Αλέξανδρος

.....

...

.....

...

.....

...

Περίληψη

Η βιομηχανία βιντεοπαιχνιδιών είναι ένας κλάδος της οικονομίας που γνώρισε μεγάλη ανάπτυξη τις τελευταίες δεκαετίες. Τα **παιχνίδια περιπέτειας (adventure games)** είναι ένα δημοφιλές είδος βιντεοπαιχνιδιού, όπου ο παίκτης αναλαμβάνει την καθοδήγηση ενός κεντρικού χαρακτήρα μέσα σε μια διαδραστική ιστορία. Τα κύρια χαρακτηριστικά του είδους είναι η αφήγηση μιας ιστορίας, η εξερεύνηση και η επίλυση γρίφων. Το είδος γνώρισε μεγάλη άνθηση κατά την δεκαετία του 1990 και τα τελευταία χρόνια γίνονται προσπάθειες για την αναβίωσή του. Στη παρούσα διπλωματική εργασία σχεδιάστηκε και αναπτύχθηκε στη μηχανή ανάπτυξης βιντεοπαιχνιδιών (game development engine) **Unity** και σε γλώσσα προγραμματισμού **C#** ένα πρότυπο πάνω στο οποίο μπορεί να βασιστεί η δημιουργία βιντεοπαιχνιδιών του είδους **“3D Point and click adventure”**. Για να γίνει η παρουσίαση του ανωτέρω πρότυπου, δημιουργήθηκε ένα παιχνίδι μικρής διάρκειας και έκτασης, με τίτλο **“Επίσκεψη στο μουσείο”** που αποτελείται από ένα αρχικό μενού και δύο σκηνές. Η πρώτη σκηνή είναι εξωτερικά ενός αρχαιολογικού μουσείου ενώ η δεύτερη είναι εσωτερικά του μουσείου, σε έναν χώρο με εκθέματα. Το παιχνίδι αυτό χωρίς να διαθέτει σενάριο ολοκληρωμένου παιχνιδιού ενσωματώνει τους περισσότερους μηχανισμούς ενός σύγχρονου εμπορικού παιχνιδιού του είδους “3D Point and Click adventure”. Διαθέτει ρεαλιστικά τρισδιάστατα γραφικά και ήχους (εφέ και διαλόγους). Οι μηχανισμοί του παιχνιδιού παρατίθενται και οι σημαντικότεροι από αυτούς αναλύονται. Στη συνέχεια, γίνεται αξιολόγηση της ποιότητας του λογισμικού έργου και τέλος παρουσιάζονται οι μέθοδοι της εφαρμογής του προτύπου για την δημιουργία νέων παιχνιδιών του είδους “3D Point and Click adventure”.

Λέξεις Κλειδιά: Ηλεκτρονικό βιντεοπαιχνίδι περιπέτειας, 3D point and click Adventure game, Unity

Abstract

The video game industry is a sector of the economy that has seen great growth in recent decades. Adventure games are a popular genre of video games where the player takes on the role of a central character in an interactive story. The main features of an **adventure game** are the story-telling, exploration and puzzle-solving. This genre flourished in the 1990s and in recent years efforts have been made to revive it. In this thesis, a template was designed and developed using the game development engine **Unity** and the programming language **C#** , while this template can be used for the creation of **"3D Point and click adventure"** video games. In order to present the template, a short game was created, entitled "Visit to the museum" consisting of an initial menu and two scenes. The first scene is outside an archaeological museum while the second is inside the museum, in an area with exhibits. This game without having a full game script incorporates most of the mechanics of a modern commercial 3D Point and Click adventure game. It has realistic 3D graphics and sounds (effects and dialogues). The mechanics of the game are listed and the most important of them are analyzed. Then the quality of the software project is measured and finally the methods of applying the template for the creation of new games of the "3D Point and Click adventure" genre are presented.

Keywords: Adventure video game, 3D point and click Adventure game, Unity

Ευχαριστίες

Ευχαριστώ τους καθηγητές μου στο μεταπτυχιακό πρόγραμμα για τις γνώσεις που με προσέφεραν στον προγραμματισμό, ιδιαίτερος τον επιβλέποντα καθηγητή μου κ. Στέλιο Ξυνόγαλο, για την βοήθεια και τις συμβουλές που μου προσέφερε κατά την ανάπτυξη του προτύπου και για τον χρόνο που αφιέρωσε για την διόρθωση της εργασίας. Ακόμη ευχαριστώ την οικογένειά μου και όλους τους φίλους και φίλες που δοκίμασαν το παιχνίδι μου, με συμβούλεψαν και με στηρίζουν σε κάθε προσπάθεια.

Θεοφάνης Σ. Αγραφιώτης.



Στιγμιότυπα του παιχνιδιού "Επίσκεψη στο μουσείο"

Βίντεο παρουσίασης του παιχνιδιού υπάρχει στο κανάλι:

<https://www.youtube.com/@theofanisagrafiotis8779>

email: mai22001@uom.edu.gr
theoagras@hotmail.com

Περιεχόμενα

1. Εισαγωγή.....	1
1.1. Η βιομηχανία βιντεοπαιχνιδιών.....	1
1.2. Σκοπός – Στόχοι.....	2
1.3. Συνεισφορά.....	3
1.4. Διάρθρωση της μελέτης.....	4
2. Βιβλιογραφική Επισκόπηση – Θεωρητικό Υπόβαθρο.....	5
2.1. Σύνοψη adventure παιχνιδιών - Ιστορία.....	5
2.2. Επισκόπηση παιχνιδιών 3D point and click adventures.....	8
2.3. Επισκόπηση αρχών ανάπτυξης παιχνιδιών.....	11
2.4. Ιστορία ανάπτυξης βιντεοπαιχνιδιών.....	13
2.5. Επισκόπηση μηχανής παιχνιδιών.....	16
2.6. Επισκόπηση του UNITY.....	17
3. Μεθοδολογία.....	21
3.1. Εισαγωγή.....	21
3.2. Προσωπικό υπόβαθρο.....	21
3.3. Unity και άλλα εργαλεία ανάπτυξης του προτύπου.....	22
4. Σχεδίαση προτύπου και υλοποίηση παιχνιδιού “Επίσκεψη στο μουσείο”..	25
4.1. Περιγραφή του παιχνιδιού “Επίσκεψη στο μουσείο”.....	25
4.2. Η φάση της σχεδίασης του λογισμικού του προτύπου - Αρχές σχεδίασης.....	30
4.2.1. Οργάνωση των στοιχείων του έργου και ονοματοδοσία.....	30
4.2.2. Managers του προτύπου.....	32
4.2.3. Υποσύστημα κίνησης του κεντρικού χαρακτήρα στον χώρο.....	34
4.2.4. Υποσύστημα αυτόματης επιλογής ενέργειας.....	35
4.2.5. Καταστάσεις του παιχνιδιού.....	40
4.2.6. Υποσύστημα ελέγχου καμερών.....	41
4.2.7. Υποσύστημα διάδρασης με αντικείμενα.....	42
4.2.8. Υποσύστημα διαλόγων.....	43
4.2.9. Υποσύστημα ελέγχου animation χαρακτήρων.....	46
4.3. Συνολικό μέγεθος του έργου.....	49
4.4. Μηχανισμοί προτύπου που υλοποιούνται στο παιχνίδι.....	49
4.5. Γραφικά του παιχνιδιού.....	58
4.6. Ήχοι του παιχνιδιού.....	61
5. Τεχνική αξιολόγηση του προτύπου και του παιχνιδιού.....	63
5.1. Περιορισμοί και προαπαιτούμενα.....	63
5.2. Τεχνικές Βελτιστοποίησης.....	64
5.2.1. Επιλογή ποιότητας γραφικών και ανάλυσης.....	64
5.2.2. Αρχές σχεδίασης και ανάπτυξης του λογισμικού για βελτιστοποίηση...	67
5.2.3. Τεχνικές Βελτιστοποίησης κατά την δημιουργία σκηνών στο γραφικό περιβάλλον της Unity.....	68
5.2.3.1. Μοντέλα με λιγότερα πολύγωνα.....	68
5.2.3.2. Occlusion Culling.....	68
5.2.3.3. Περιορισμός στην ανίχνευση σύγκρουσης (collision detection).....	69
5.3. Μετρήσεις (benchmark) και αξιολόγηση αποδόσεων.....	70

5.4. Συμπεράσματα αξιολόγησης.....	75
6. Εφαρμογή του προτύπου για δημιουργία adventure παιχνιδιών.....	76
6.1. Εισαγωγή.....	76
6.2. Προσθήκη διαδραστικού αντικειμένου σε υπάρχουσα σκηνή.....	76
6.3. Προσθήκη σκηνής.....	81
7. Επίλογος.....	85
7.1. Σύνοψη και συμπεράσματα.....	85
7.2. Μελλοντικές προεκτάσεις.....	86

Κατάλογος Εικόνων

1.1. Διάγραμμα με τις προβλέψεις εσόδων της βιομηχανίας βιντεοπαιχνιδιών ανά τομέα πηγή - PwC (www.pwc.com).....	1
1.2. Στιγμιότυπο του παιχνιδιού "Επίσκεψη στο μουσείο".....	3
2.1. Το παιχνίδι "Colossal Cave Adventure".....	5
2.2. Στο παιχνίδι Maniac Mansion (1987) οι εντολές βρίσκονται στο κάτω μέρος της οθόνης και εισάγονται με κλικ του ποντικιού	6
2.3. Στιγμιότυπα από τις σειρές "Monkey island" και "King's Quest".....	6
2.4. Στιγμιότυπο από παιχνίδι "Grim Fandango".....	7
2.5. Εμφάνιση μενού στο adventure Beyond a Steel Sky - Revolution Software Ltd 2020.....	9
2.6. Στιγμιότυπο από το σύγχρονο 3D p&c adventure "The Book of Unwritten Tales 2".....	9
2.7. Φωτογραφία καμπίνας και στιγμιότυπο του παιχνιδιού "Pong" (1972) της ATARI	14
2.8. Το παιχνίδι "Manic Miner" έχει μοναδικό developer τον Matthew Smith.....	15
2.9. Ο visual editor της Unity.....	18
2.10. Στο παράθυρο Inspector με επιλεγμένο το gameObject ΚιβώτιοΑ εμφανίζονται τα component που έχουν προστεθεί σε αυτή.....	19
3.1. Στιγμιότυπο από το σύγχρονο 3D p&c adventure "Syberia 3".....	22
3.2. Επεξεργασία ήχου με το online πρόγραμμα "Twisted Wave".....	24
4.1. Η αρχική οθόνη του παιχνιδιού με το κεντρικό μενού.....	25
4.2. Οθόνη κατά την φόρτωση των σκηνών.....	26
4.3. Η πρώτη σκηνή του παιχνιδιού, εξωτερικά του μουσείου.....	27
4.4. Το αποθετήριο (inventory) με τα αντικείμενα που έχουν περισυλλεχθεί.....	28
4.5. Στιγμιότυπο από την εσωτερική σκηνή του μουσείου.....	29
4.6. Επιφάνειες που μπορεί να κινείται ο χαρακτήρας στην εσωτερική σκηνή.....	34
4.7. Διάγραμμα ροής για το υποσύστημα αυτόματης επιλογής ενέργειας.....	36
4.8. Διάγραμμα UML για την "Μέθοδο Υπόδειγμα".....	39
4.9. Τα keyframes του animation της κίνησης του χαρακτήρα όταν είναι σταθερός σε ένα σημείο (idle anim).....	47
4.10. Αναζήτηση Humanoid anim στην ιστοσελίδα mixamo.com	47
4.11. Ο animator του ανθρώπου που αθλείται στην εξωτερική σκηνή.....	48
4.12. Εισαγωγική οθόνη με λογότυπο δημιουργού.....	58
4.13. British παλέτα χρωμάτων με δημιουργό τον Jan Losert από την ιστοσελίδα https://flatuicolors.com/	59
4.14. Εξέλιξη της κατασκευής του κτηρίου με το Pro builder.....	61
5.1. Ιδιότητες LowURP.....	64
5.2. Ιδιότητες MediumURP.....	64
5.3. Ιδιότητες HighURP.....	65
5.4. Στιγμιότυπο με την χαμηλή ρύθμιση ποιότητας γραφικών.....	65
5.5. Στιγμιότυπο με την μέτρια ρύθμιση ποιότητας γραφικών.....	66
5.6. Στιγμιότυπο με την υψηλή ρύθμιση ποιότητας γραφικών.....	66

5.7. Ρύθμιση ανάλυσης της οθόνης.....	67
5.8. Occlusion Culling στην εξωτερική σκηνή του μουσείου.....	68
5.9. Ο collider σε σχήμα κυλίνδρου γύρω από τον κεντρικό χαρακτήρα.....	69
5.10. Μετρήσεις με το εργαλείο Statistics στην εξωτερική σκηνή.....	71
5.11. Μετρήσεις με το εργαλείο Statistics στην εσωτερική σκηνή.....	71
5.12. Μετρήσεις με το εργαλείο Statistics σε χαμηλή ποιότητα γραφικών.....	72
5.13. Μετρήσεις με το εργαλείο διαχείριση εργασιών των Windows.....	72
6.1. Αναζήτηση αντικειμένου αγάλματος αλόγου στο Asset Store.....	77
6.2. Εισαγωγή του αντικειμένου αγάλματος στο project.....	77
6.3. Αλλαγή των ιδιοτήτων του αντικειμένου.....	78
6.4. Ο inspector του αντικειμένου με προσθήκη της κλάσης InteractObject και των ιδιοτήτων της.....	78
6.5. Σύνολο των ιδιοτήτων του αντικειμένου στον inspector.....	79
6.6. Στιγμιότυπο από το παιχνίδι, όταν ο δείκτης κινείται πάνω στο αντικείμενο.....	80
6.7. Προσθήκη του component PickupObject.....	80
6.8. Η σκηνή μετά την δημιουργία εδάφους με το εργαλείο terrain.....	82
6.9. Η επιφάνεια παν-mesh στην σκηνή που υπάρχει ένα εμπόδιο.....	82
6.10. Οι ιδιότητες του αντικειμένου της κάμερας στον inspector.....	83
6.11. Οι ιδιότητες του αντικειμένου SceneLoader στον inspector.....	84
6.12. Η DemoScene σκηνή κατά την διάρκεια του παιχνιδιού.....	84

Κατάλογος Πινάκων

2.1. Βασικά component της Unity και οι λειτουργίες τους.....	18
4.1. Οι φάκελοι αρχείων με assets του λογισμικού έργου.....	30
4.2. Μέγεθος του κώδικα του έργου.....	49
4.3. Μηχανισμοί των τίτλων εισαγωγής.....	50
4.4. Μηχανισμοί αρχικού μενού	50
4.5. Μηχανισμοί διεπαφής (User Interface - UI).....	51
4.6. Μηχανισμοί κίνησης και ενεργειών κεντρικού χαρακτήρα	52
4.7. Μηχανισμοί για τις Σκηνές (scenes)	55
4.8. Μηχανισμοί Καμερών	56
4.9. Μηχανισμοί εκμάθησης και εξέλιξης παιχνιδιού.....	57
4.10. Δωρεάν Αντικείμενα 3D που έχουν ληφθεί από την ιστοσελίδα https://www.turbosquid.com/	60
5.1. Πίνακας: Αποτελέσματα μετρήσεων των fps σε διαφορετικούς υπολογιστές.....	73

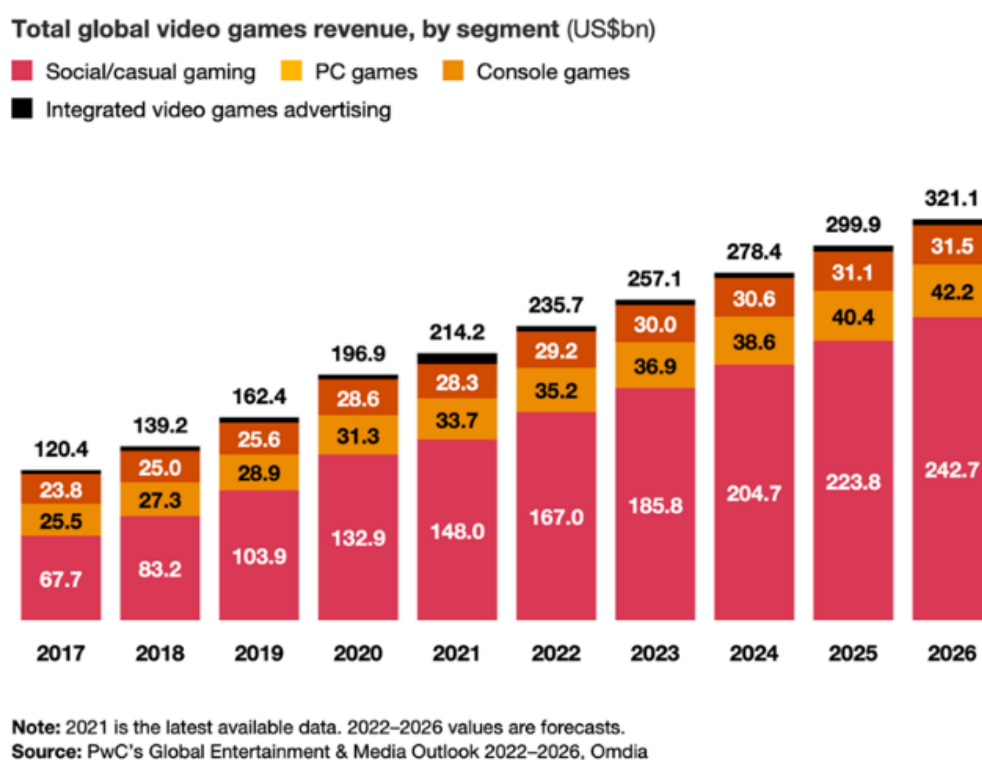
Κατάλογος Κώδικα

2.1. Το script CrateFall.cs που έχει προστεθεί στο αντικείμενο κιβώτιοΓ.....	20
4.1. Το script Managers.cs - ο κεντρικό μάντζερ του έργου.....	32
4.2. Δημόσια μέθοδος getWaypointIfExistsOtherwiseObjectsPos σε κάθε διαδραστικό αντικείμενο.....	37
4.3. Ορισμός καταστάσεων στην Scene_Manager.cs.....	40
4.4. Δημόσια διασύνδεση της gameState μέσα στην Scene_Manager.cs.....	40
4.5. Στην κλάση CameraScript.cs η κάμερα τίθεται πάνω από τον χαρακτήρα.....	41
4.6. Στην κλάση CameraScript.cs η μετακίνηση της κάμερας γίνεται ομαλά.....	42
4.7. Κώδικας: Οι ιδιότητες και ο κατασκευαστής της μάντζερ κλάσης Dialogue.cs	44
4.8. Κώδικας: Δήλωση πίνακα με κλάσεις Dialogue.cs και δημόσια διασύνδεση	44
4.9. Κώδικας: Αρχικοποίηση στην DialogueManager.cs.....	45

1. Εισαγωγή

1.1. Η βιομηχανία βιντεοπαιχνιδιών

Τα βιντεοπαιχνίδια είναι τα παιχνίδια που παίζονται με την χρήση ηλεκτρονικής συσκευής, όπως ο ηλεκτρονικός υπολογιστής, κινητό τηλέφωνο ή κονσόλα βιντεοπαιχνιδιών. Η βιομηχανία των βιντεοπαιχνιδιών είναι τομέας της βιομηχανίας ψυχαγωγίας και τα τελευταία χρόνια αποτελεί έναν από τους πιο ταχέως αναπτυσσόμενους κλάδους της παγκόσμιας οικονομίας. Τα έσοδα της πρόκειται να εκτοξευτούν στα 321 δισ.\$ το 2026, σύμφωνα με τις προβλέψεις που αποτυπώνονται στην Εικόνα 1.1.



Εικόνα 1.1: Διάγραμμα με τις προβλέψεις εσόδων της βιομηχανίας βιντεοπαιχνιδιών ανά τομέα
πηγή: PwC (www.pwc.com)

Χαρακτηριστικό είναι ότι πλέον τα έσοδα της βιομηχανίας των βιντεοπαιχνιδιών σε όλο τον κόσμο έχουν ξεπεράσει κατά πολύ αυτά της βιομηχανίας ταινιών. Αποτελείται από επιμέρους τομείς εκ των οποίων οι κυριότεροι είναι η **ανάπτυξη παιχνιδιών (game development - gamedev)** που αφορά τη σχεδίαση, τον προγραμματισμό και την παραγωγή περιεχομένου του παιχνιδιού και η **έκδοση των παιχνιδιών (game publishing)** που αφορά την κυκλοφορία, το μαρκετινγκ και την χρηματοδότηση του παιχνιδιού.

Δυστυχώς, η Ελλάδα δεν είναι μία χώρα που να φημίζεται για το game development και την παραγωγή πολλών επιτυχημένων εμπορικών βιντεοπαιχνιδιών. Διαχρονικά ελάχιστοι είναι οι ποιοτικοί τίτλοι από Έλληνες δημιουργούς που κατάφεραν και έγιναν παγκόσμιες εμπορικές επιτυχίες και μπόρεσαν να ανταγωνιστούν ξένα studios παραγωγής, όπως το **Dark Fall** (2007) της **Aventurine** που θεωρείται το μεγαλύτερο ελληνικό video game που φτιάχτηκε ποτέ. Υπάρχουν πολλοί λόγοι που συμβαίνει αυτό, που μπορούν να συνοψιστούν σε οικονομικούς-χρηματοδοτικούς, κουλτούρας, έλλειψης εκπαίδευσης και άλλους δομικούς που καθιστούν μέχρι τώρα την επαγγελματική ενασχόληση με το game development μη βιώσιμο, αλλά δεν θα επεκταθούμε στο θέμα αυτό στην παρούσα εργασία.

Από την άλλη πλευρά παρατηρείται τα τελευταία χρόνια στη χώρα μας μία τάση αντιστροφής του κλίματος και σε αυτό συμβάλλουν η δημιουργία κοινοτήτων δημιουργών για ανταλλαγή απόψεων, εύρεση συνεργατών και παρουσίαση έργων, η διοργάνωση μεγάλων εκδηλώσεων όπως η **Gameathon** και τα προγράμματα χρηματοδότησης της ανάπτυξης ηλεκτρονικών παιχνιδιών από την ελληνική πολιτεία όπως το πρόγραμμα Cash Rebate 40% του **Κέντρου Μελετών Οπτικοακουστικών Παραγωγών**. Αποτέλεσμα είναι ότι ήδη άρχισαν να εμφανίζονται κάποιες οργανωμένες προσπάθειες από μικρά studios και ανεξάρτητους παραγωγούς για κυκλοφορία ολοκληρωμένων παιχνιδιών στις παγκόσμιες αγορές, που να μπορούν να ανταγωνιστούν αντίστοιχες μικρές παραγωγές του εξωτερικού.

1.2. Σκοπός – Στόχοι

Ο σκοπός αυτής της εργασίας είναι η δημιουργία ενός προτύπου για την ανάπτυξη βιντεοπαιχνιδιών του είδους **“3D Point And Click Adventure”** και η παρουσίαση του μέσω ενός μικρού παιχνιδιού με τίτλο **“Επίσκεψη στο μουσείο”** (Εικόνα 1.2). Με την προϋπόθεση της εύρεσης ενός καλού σεναρίου για παιχνίδι adventure, που να περιέχει ποιοτικούς γρίφους και με την προσθήκη ρεαλιστικών γραφικών και ήχων το πρότυπο θα μπορεί να χρησιμοποιηθεί για την ανάπτυξη παιχνιδιών που θα μπορούν να ανταγωνιστούν άλλα υπάρχοντα adventure games σε παγκόσμιες αγορές. Το πρότυπο που δημιουργήθηκε περιέχει τις περισσότερες από τις λειτουργίες και τους μηχανισμούς που χαρακτηρίζουν τα Point And Click Adventure παιχνίδια και επιπλέον αποτελείται από αποτελεσματικό, επεκτάσιμο και μη ευάλλωτο κώδικα. Κατά την εκτέλεση του παιχνιδιού επίδειξης που βασίστηκε στο πρότυπο δεν προκύπτουν προβλήματα (bugs), ενώ η απόδοσή του σύμφωνα με μετρήσεις είναι πολύ ικανοποιητική. Η ομαλότητα στην κίνηση κατά την εκτέλεση εγγυάται μια συνολική ευχάριστη εμπειρία σε οποιονδήποτε παίξει το παιχνίδι.

Το πρότυπο και το παιχνίδι σχεδιάστηκαν και αναπτύχθηκαν αποκλειστικά με την προσωπική εργασία του συντάκτη, δουλεύοντας από τον Νοέμβριο του 2022 έως τον Μάιο του 2023 δύο με τρεις ώρες την ημέρα.



Εικόνα 1.2: Στιγμιότυπο του παιχνιδιού "Επίσκεψη στο μουσείο"

1.3. Συνεισφορά

Με το πρότυπο που αναπτύχθηκε στο πλαίσιο της παρούσας εργασίας, ο συντάκτης ευελπιστεί να συμμετέχει ενεργά στην προσπάθεια που γίνεται από τους Έλληνες δημιουργούς, για την εγχώρια παραγωγή αξιόλογων βιντεοπαιχνιδιών. Το πρότυπο προσφέρει στους δημιουργούς ένα εργαλείο για την ευκολότερη ανάπτυξη παιχνιδιών του είδους **"3D Point And Click Adventure"**. Τους επιτρέπει να εστιάσουν την δημιουργικότητά τους στην ιστορία του παιχνιδιού και στη παραγωγή γραφικών και ήχων και όχι σε συγγραφή κώδικα για την υλοποίηση των μηχανισμών που περιέχει ένα παιχνίδι του είδους. Όμως, η πλήρης εκμετάλλευση του προτύπου έχει προαπαιτούμενο την γνώση των βασικών αρχών προγραμματισμού, της γλώσσας C# και της μηχανής παιχνιδιών Unity.

Οι αναγνώστες της εργασίας θα λάβουν γνώση για την υλοποίηση των βασικών μηχανισμών ενός μοντέρνου adventure game σε στο περιβάλλον την Unity. Ακόμα, θα διαβάσουν για τις βασικές αρχές λογισμικού και για τα πρότυπα σχεδίασης που εφαρμόστηκαν, και θα δουν τον τρόπο που χρησιμοποιήθηκαν κάποια από τα εργαλεία που προσφέρει η μηχανή παιχνιδιών Unity για την ανάπτυξη παιχνιδιών.

1.4. Διάρθρωση της μελέτης

Το περιεχόμενο της εργασίας είναι χωρισμένο σε κεφάλαια ως εξής:

Στο δεύτερο κεφάλαιο γίνεται μία περιγραφή του όρου **adventure game** και μία ιστορική ανασκόπηση από την πρώτη εμφάνισή του. Ακολουθεί η περιγραφή μίας υποκατηγορίας παιχνιδιών adventure που είναι τα **3D Point And Click**. Στη συνέχεια, γίνεται μία ανασκόπηση της ανάπτυξης ηλεκτρονικών παιχνιδιών και των εξελίξεων που διαδραματίστηκαν μέσα στις τελευταίες δεκαετίες. Μετά γίνεται αναφορά στις μηχανές γραφικών και του ρόλου τους στην ανάπτυξη ηλεκτρονικών παιχνιδιών. Γίνεται ιδιαίτερη αναφορά στην Unity.

Το τρίτο κεφάλαιο αναφέρεται στη μεθοδολογία που ακολουθήθηκε κατά τη συγγραφή της εργασίας και στα εργαλεία ανάπτυξης του προτύπου και του παιχνιδιού που χρησιμοποιήθηκαν.

Το τέταρτο κεφάλαιο ξεκινάει με την περιγραφή του παιχνιδιού επίδειξης "Επίσκεψη στο μουσείο". Αναλύονται μερικά από τα υποσυστήματα που αποτελούν το πρότυπο και οι αρχές ποιοτικού λογισμικού που εφαρμόστηκαν κατά τη σχεδίαση και δημιουργία του. Έπειτα, παρατίθενται οι μηχανισμοί που υλοποιούνται στο παιχνίδι με μια σύντομη περιγραφή για τον καθένα. Το κεφάλαιο κλείνει με αναφορά στα γραφικά και στους ήχους του παιχνιδιού.

Στο πέμπτο κεφάλαιο αναφέρονται τα προαπαιτούμενα για να εκτελεστεί το παιχνίδι και αναλύονται οι τεχνικές βελτιστοποίησης που εφαρμόστηκαν κατά την ανάπτυξή του. Στη συνέχεια, παρουσιάζονται μετρήσεις της απόδοσης του παιχνιδιού, η αξιολόγησή τους και τέλος τα συμπεράσματα που εξάγονται από αυτές.

Στο έκτο κεφάλαιο περιγράφονται τα βήματα που πρέπει να κάνει ένας δημιουργός για την εφαρμογή του προτύπου στο δικό του παιχνίδι και συγκεκριμένα για την εισαγωγή ενός διαδραστικού αντικειμένου και για την προσθήκη νέας σκηνής.

Στο έβδομο κεφάλαιο παρατίθενται η σύνοψη με τα συμπεράσματα από την εργασία καθώς και οι μελλοντικές προεκτάσεις του προτύπου.

2. Βιβλιογραφική Επισκόπηση – Θεωρητικό Υπόβαθρο

2.1. Σύνοψη adventure παιχνιδιών - Ιστορία

Τα παιχνίδια περιπέτειας (adventure games) είναι το είδος βιντεοπαιχνιδιού όπου ο παίκτης αναλαμβάνει τον ρόλο του πρωταγωνιστή μιας διαδραστικής ιστορίας καθοδηγούμενης από την εξερεύνηση και από την επίλυση γρίφων (Rollings και Adams, 2003). Δεν περιέχουν δράση και συμπλοκές με εχθρούς, ενώ συνήθως ο ρυθμός που εξελίσσεται η ιστορία είναι αργός. Επιπλέον, τα παιχνίδια αυτά είναι σχεδιασμένα να παίζονται από μόνο ένα παίκτη και συνήθως δεν απαιτούν σύνδεση στο διαδίκτυο (offline). Η καθηλωτική και συναρπαστική εμπειρία που προσφέρουν πολλά από τα adventure games, έχει οδηγήσει στο να υπάρχει μεγάλη απήχηση στον κόσμο διαχρονικά, μεγάλο μέρος του οποίου προτιμάει να παίζει αυτό το είδος βιντεοπαιχνιδιών για την ψυχαγωγία του.

Τα πρώτα adventures games αναπτύχθηκαν την δεκαετία του 1970, ως παιχνίδια περιπέτειας κειμένου, στερούμενα γραφικών. Η εξέλιξη της ιστορίας τους βασιζόνταν στην καθοδήγηση του πρωταγωνιστή με εντολές που εισάγονταν από το πληκτρολόγιο όπως **get axe, use keys, w (go west)** κλπ. Το πρώτο παιχνίδι του είδους θεωρείται το **Colossal Cave Adventure** που αναπτύχθηκε από τον **Will Crowthe** το 1976, το οποίο ουσιαστικά ονομάτησε και το είδος, αφού ονομάζονταν συχνά απλά adventure (Εικόνα 2.1).

```
In West Side Chamber
>n
You can't go that way.

A threatening little dwarf comes out of
the shadows!

>throw axe at dwarf
You killed a little dwarf! The body
vanishes in a cloud of greasy black
smoke.

>get axe
Taken.

>|
```

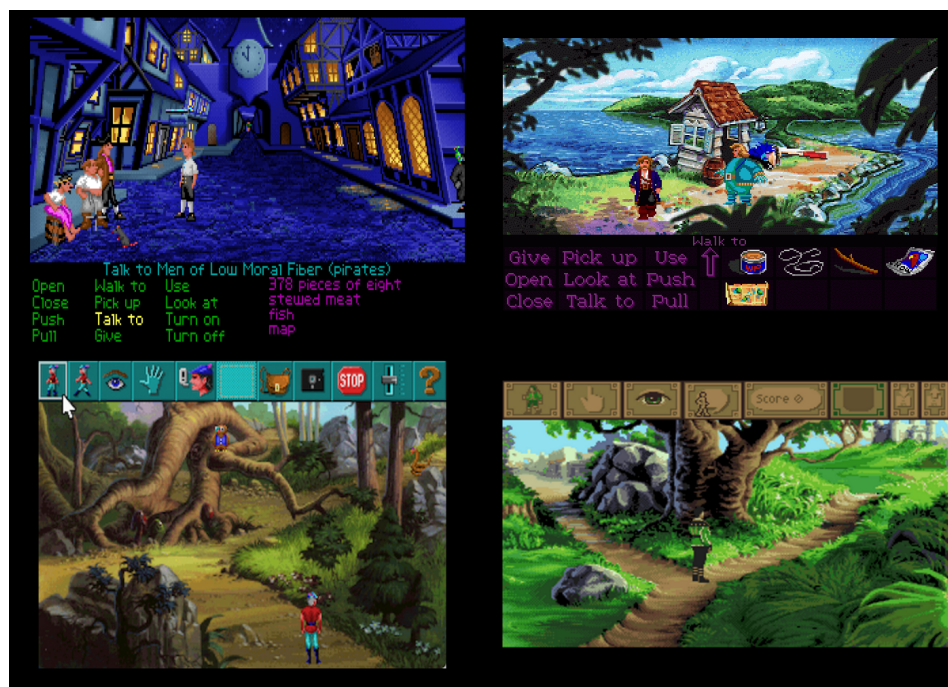
Εικόνα 2.1: Το παιχνίδι "Colossal Cave Adventure"

Το είδος φυσικά εξελίχθηκε παράλληλα με την εξέλιξη της τεχνολογίας υπολογιστών. Στις αρχές της δεκαετίας του 1980 προστέθηκαν γραφικά, αρχικά στατικές εικόνες και αργότερα animations. Σταδιακά ο χειρισμός με το πληκτρολόγιο αντικαταστάθηκε με το ποντίκι και πλέον οι εντολές δεν εισάγονταν πληκτρολογώντας αλλά με απλά κλικ του ποντικιού πάνω σε μια λίστα εντολών στο κάτω μέρος της οθόνης (Εικόνα 2.2). Τη δεκαετία του 90 η αύξηση στην χωρητικότητα της

αποθηκευτικής μνήμης των υπολογιστών οδήγησε στη δημιουργία μεγαλύτερων ιστοριών και πιο πλούσιων κόσμων μέσα στα adventures games. Αυτή η εποχή χαρακτηρίζεται από πολλούς (και τον συντάκτη) ως χρυσή εποχή των adventure games, καθώς απέκτησαν μεγάλη δημοφιλία ανάμεσα στα άλλα είδη βιντεοπαιχνιδιών. Τότε κυκλοφόρησαν πολλοί επιτυχημένοι εμπορικά τίτλοι που επηρέασαν την ανάπτυξη των παιχνιδιών του είδους από τότε μέχρι και σήμερα. Δύο εταιρείες ανάπτυξης ξεχώριζαν, η **Sierra** και η **LucasArts** με τις σειρές παιχνιδιών όπως τα **King's Quest** και τα **Monkey island** (Εικόνα 2.3). Αξιοσημείωτο είναι ότι τα περιοδικά για τους υπολογιστές της δεκαετίας του 90, σχεδόν όλα αφιέρωναν ξεχωριστές πολυσέλιδες ενότητες με άρθρα παρουσίασης των adventure games αλλά και άρθρα με λύσεις γρίφων που περιλαμβάνονταν σε γνωστούς adventure τίτλους της εποχής.



Εικόνα 2.2: Στο παιχνίδι Maniac Mansion (1987) οι εντολές βρίσκονται στο κάτω μέρος της οθόνης και εισάγονται με κλικ του ποντικιού



Εικόνα 2.3: Στιγμιότυπα από τις σειρές "Monkey island" και "King's Quest"

Καθώς το είδος εξελίσσονταν, η επιλογή της εντολής από το κάτω μέρος της οθόνης άρχισε να αντικαθιστάται με την διεπαφή στην οποία με ένα κουμπί κάνεις όλες τις ενέργειες, όπως για παράδειγμα στα adventures **broken sword** (1996) και **Toonstruck** (1996). Αυτή η διεπαφή έχει υιοθετηθεί στα περισσότερα adventure game μέχρι και σήμερα. Επίσης, η κυκλοφορία των παιχνιδιών σε CD αντί για δισκέτες, επέτρεψε την είσοδο της φωνητικής των διαλόγων αλλά και την είσοδο ολόκληρων ταινιών (Full Motion Video) που συμπλήρωναν την πλοκή του παιχνιδιού.

Επιπρόσθετα, όπως και σε όλα τα είδη video παιχνιδιών, προς το τέλος της δεκαετίας του 1990, τα τρισδιάστατα γραφικά άρχισαν να εμφανίζονται και στα adventure games. Με την είσοδο στην αγορά όλο και πιο ισχυρών επεξεργαστών γραφικών (GPU) οι δυνατότητες για καλύτερη τρισδιάστατη απεικόνιση μεγεθύνθηκαν. Το **Grim Fandango** (1998) της **LucasArts** είναι από τα πρωτοπόρα 3D adventure παιχνίδια και πολύ δημοφιλές ως τις μέρες μας, που περιέχει 3D κινούμενα γραφικά πάνω σε pre-rendered στατικό φόντο. Η Εικόνα 2.4 είναι ένα στιγμιότυπο από το παιχνίδι όπως είχε κυκλοφορήσει τότε. Άξιο αναφοράς είναι και το adventure **MYST** που αναπτύχθηκε από την **Cyan** και εκδόθηκε από την **Broderbund** το 1993, το οποίο επίσης, είχε pre-rendered 3D στατικά γραφικά, αλλά ήταν παιχνίδι πρώτου προσώπου, δηλαδή η περιπλάνηση στον χώρο γίνονταν από την προοπτική των ματιών του πρωταγωνιστή και δεν εμφανίζονταν αυτός στην οθόνη. Το παιχνίδι έκανε ρεκόρ πωλήσεων βιντεοπαιχνιδιών για υπολογιστές που κράτησε επτά χρόνια, ενώ, επειδή το περιεχόμενό του δεν χωρούσε σε δισκέτες και κυκλοφόρησε μόνο σε CD, ώθησε πολλούς κατόχους υπολογιστών να αγοράσουν συσκευές CDROM για να μπορέσουν να παίξουν το συγκεκριμένο παιχνίδι. Γενικά, την δεκαετία του 1990 το πιο εμπορικό είδος βιντεοπαιχνιδιού ήταν το adventure.



Εικόνα 2.4: Στιγμιότυπο από παιχνίδι "Grim Fandango"

Την δεκαετία του 2000, όμως η δημοφιλία του είδους άρχισε να μειώνεται. Λίγοι εμπορικοί τίτλοι κυκλοφόρησαν και το μερίδιο αγοράς σε σχέση με άλλα είδη βιντεοπαιχνιδιών συρικνώθηκε αρκετά. Ακόμα και η μεγάλη κατασκευάστρια εταιρεία των adventure **Sierra** έκλεισε διότι δεν κατάφερε να κυκλοφορήσει τίτλο με εμπορική επιτυχία. Ελάχιστα ποιοτικά adventures κυκλοφόρησαν, όπως το **The Longest Journey** της **Funcom** και το **Syberia** της **Microids**, τα οποία επίσης, διέθεταν στατικό pre-rendered φόντο.

Τέλος, τα τελευταία χρόνια υπάρχει μία αναβίωση του είδους, που απέκτησε ξανά μεγάλη δημοφιλία, η οποία μπορεί να οφείλεται στο αίσθημα της νοσταλγίας αυτών που έζησαν την δεκαετία του 90 και έπαιξαν adventures της εποχής. Χαρακτηριστικό είναι ότι το 2014 ο **Ron Gilbert**, developer του βραβευμένου adventure **Thimbleweed Park**, έλαβε χρηματοδότηση από κοινό ύψους 626.000 δολαρίων μέσω **Kickstarter**, με τους χρηματοδότες που ήταν λάτρεις των κλασικών p&c adventures να αναλαμβάνουν και ενεργό ρόλο στην ανάπτυξη του παιχνιδιού. Ένας άλλος λόγος είναι η δυνατότητα που υπάρχει πλέον σε ανεξάρτητες ομάδες δημιουργών, ανάμεσα στους οποίους υπάρχουν πολλοί λάτρεις των p&c adventures, να δημιουργήσουν δικά τους καινούργια adventures.

2.2. Επισκόπηση παιχνιδιών 3D point and click adventures

Μια υποκατηγορία των adventure games είναι και τα **3D point and click (p&c)** τρίτου προσώπου. Χαρακτηριστικό αυτού του είδους βιντεοπαιχνιδιού είναι ότι ο χειρισμός του παιχνιδιού γίνεται με το ποντίκι ή με απλό άγγιγμα στην οθόνη αφής (όταν παίζεται σε κινητό ή tablet), χωρίς να χρειάζεται ιδιαίτερη δεξιότητα για την κίνηση του παίκτη όπως στα παιχνίδια δράσης. Κατά κάποιο τρόπο τα 3D p&c είναι η εξέλιξη των κλασικών **p&c graphic adventures** της δεκαετίας του 1980 και αρχών του 1990, με την προσθήκη της τρισδιάστατης προοπτικής, των βελτιωμένων γραφικών και ήχου, της μεγαλύτερης εξερεύνησης και της περισσότερης διαδραστικότητας στην επίλυση γρίφων, χωρίς όμως αυτό να σημαίνει ότι δεν υπάρχουν μοντέρνα adventure δύο διαστάσεων όπως το **Larry: Wet Dreams Don't Dry (2018)**. Πλέον δεν υπάρχει κατάλογος εντολών, γιατί αυτός έχει αντικατασταθεί είτε από ένα σύστημα αυτόματης ανίχνευσης ενέργειας αλληλεπίδρασης με το αντικείμενο, πάνω στο οποίο ο χρήστης έκανε κλικ του ποντικιού (ή πάτημα στην οθόνη αφής) είτε ενός μικρού μενού στο σημείο όπου γίνεται κλικ, με δύο μέχρι το πολύ τέσσερις ενέργειες αλληλεπίδρασης με το αντικείμενο αυτό (Εικόνα 2.5). Ενδεικτικά, η Εικόνα 2.6 έχει ληφθεί από το 3D p&c adventure **The Book Of**

Unwritten Tales 2 της **King Arts Game** και κυκλοφόρησε από την **Nordic Games** το 2015.



Εικόνα 2.5: Εμφάνιση μενού στο adventure Beyond a Steel Sky - Revolution Software Ltd 2020



Εικόνα 2.6: Στιγμιότυπο από το σύγχρονο 3D p&c adventure "The Book of Unwritten Tales 2"

Υπάρχουν κάποια κοινού είδους υποσυστήματα σε όλα τα p&c adventure games που τα κάνουν να ξεχωρίζουν από τα άλλα είδη βιντεοπαιχνιδιών και αναφέρονται παρακάτω:

Το σύστημα χειρισμού με δείκτη (cursor): Ο παίκτης του παιχνιδιού αλληλεπιδρά με το παιχνίδι μέσω της κίνησης του δείκτη με το ποντίκι και κάνοντας κλικ για να πραγματοποιήσει διάφορες ενέργειες. Κάνοντας αριστερό κλικ σε σημείο στο περιβάλλον όπου δεν υπάρχει αντικείμενο ενδιαφέροντος ο χαρακτήρας που

ελέγχει ο παίκτης κατευθύνεται σε αυτό. Αν ο δείκτης μετακινηθεί σε σημείο ενδιαφέροντος ή πάνω σε άλλον χαρακτήρα τότε ο δείκτης αλλάζει ανάλογα σε άλλο σχήμα και με το κλικ αλληλεπιδρά με αυτό το αντικείμενο, δηλαδή είτε το εξερευνά, ή το περισυλλέγει, ή το χρησιμοποιεί ή ξεκινάει διάλογος αν το αντικείμενο είναι άλλος χαρακτήρας. Επίσης, μπορεί να επιλεγθεί σημείο εξόδου ή εισόδου σε άλλη σκηνή, όπως μία πόρτα αίθουσας, όπου ο χαρακτήρας μεταφέρεται σε άλλη σκηνή παιχνιδιού ή σε άλλη τοποθεσία. Πολλά παιχνίδια πλέον υποστηρίζουν οθόνες αφής, οπότε οι προαναφερθείσες ενέργειες γίνονται με άγγιγμα στην οθόνη πάνω στα σημεία ενδιαφέροντος

Η κίνηση του χαρακτήρα: Η τρισδιάστατη κίνηση του χαρακτήρα στον χώρο γίνεται αποφεύγοντας τα εμπόδια, επιλέγοντας την πιο σύντομη διαδρομή ανάμεσα στο σημείο που βρίσκεται και στο σημείο που θέλει να τον κατευθύνει ο παίκτης, ενώ υπάρχουν και σημεία στο περιβάλλον που ο χαρακτήρας δεν μπορεί να τα προσεγγίσει. Η κίνηση αυτή πρέπει να γίνεται με ομαλό 3D animation του χαρακτήρα που να εξομοιώνει όσο το δυνατόν καλύτερα την κίνηση ενός πραγματικού ανθρώπου.

Το σύστημα inventory (απόθεμα πραγμάτων): Ο παίκτης συλλέγει χρήσιμα πράγματα από το περιβάλλον, τα οποία τοποθετούνται στο inventory και μπορεί να τα χρησιμοποιήσει αργότερα για επίλυση γρίφων. Για να αποκτήσει πρόσβαση σε αυτά, υπάρχει ειδική διεπαφή που ανοίγει με κάποιο κουμπί και καλύπτει μέρος ή ολόκληρη την οθόνη. Εκεί εμφανίζονται τα αντικείμενα σε μικρογραφίες και κάνοντας κλικ σε αυτά ο παίκτης μπορεί να τα επιλέγει για να τα χρησιμοποιήσει κάπου στο περιβάλλον ή να τα δώσει σε άλλον χαρακτήρα ή και να τα συνδυάσει μεταξύ τους.

Το σύστημα διαλόγου: Καθώς εξελίσσεται η ιστορία ενός adventure ο χαρακτήρας του παίκτη έρχεται σε επικοινωνία με άλλους χαρακτήρες. Αυτό γίνεται συνήθως κάνοντας ο παίκτης αριστερό κλικ πάνω τους. Τότε ταυτόχρονα ακούγονται και εμφανίζονται οι διάλογοι στην οθόνη. Ο παίκτης σε κάποια σημεία συμμετέχει στον διάλογο επιλέγοντος αυτός πως θα εξελιχθεί, ανάμεσα από λίγες διαφορετικές ερωτήσεις ή αποκρίσεις. Σε αυτό το σημείο η ροή του παιχνιδιού σταματάει και εμφανίζονται στην οθόνη οι διαφορετικές επιλογές των φράσεων που μπορεί να πει ο χαρακτήρας. Κάνοντας κλικ σε μία από αυτές το παιχνίδι και ο διάλογος συνεχίζεται, ενώ από αυτή την επιλογή μπορεί να εξαρτηθεί και η εξέλιξη της ιστορίας.

Το σύστημα επίλυσης γρίφων: Προκειμένου να εξελιχθεί η ιστορία, ο παίκτης αφενός πρέπει να εξερευνά όλους τους χώρους και αφετέρω πρέπει να επιλύσει μια σειρά γρίφων που στηρίζονται ως επί των πλείστον στην λογική, αλλά και στη παρατηρητικότητα. Οι γρίφοι αυτοί μπορεί να είναι ενσωματωμένοι στους διαλόγους, όπου πρέπει να επιλεγθεί η σωστή φράση ενώ κάποιοι επιλύονται με την χρήση αντικειμένων που έχουν συλλεχθεί στο inventory. Πολλές φορές οι γρίφοι

αυτοί ανοίγουν σε ξεχωριστή οθόνη, όπου ο χαρακτήρας παύει να εμφανίζεται και σταματάει η κανονική ροή του παιχνιδιού.

Το σύστημα αφήγησης: Το σύστημα αυτό εισάγει τον παίκτη στον κόσμο του παιχνιδιού, τον τοποθετεί στα μέρη που εξελίσσεται η ιστορία του και δίνει πληροφορίες για τον κεντρικό ήρωα και τους άλλους χαρακτήρες που συμμετέχουν. Όπως προαναφέρθηκε η ιστορία ξεδιπλώνεται μέσω εξερεύνησης και επίλυσης γρίφων και το σύστημα αφήγησης είναι αυτό που ελέγχει αυτήν την εξέλιξη και διατηρεί την συνοχή της. Η ιστορία πέρα από το κυρίως παιχνίδι που ο παίκτης κινεί τον χαρακτήρα του και αλληλεπιδρά με άλλα αντικείμενα, περιέχει και cutscenes τα οποία είναι βίντεο σαν ταινία με σκοπό να παρουσιαστούν στον παίκτη στοιχεία από το σενάριο του παιχνιδιού και να συλλέξει πληροφορίες για την ιστορία. Συνήθως, στα 3D point and click adventures είναι τρισδιάστατα βίντεο και χρησιμοποιούνται για να τονίσουν ένα σημαντικό κομμάτι της ιστορίας ή για να εισαγάγουν τον παίκτη σε μία καινούργια τοποθεσία. Αντί για cutscene για τον ίδιο λόγο, μπορεί και να χρησιμοποιηθεί ένας μακρύς διάλογος ή μονόλογος, κατά τη διάρκεια του οποίου σταματάει και πάλι το gameplay. Η αφήγηση μπορεί να είναι σειριακή από την αρχή μέχρι το τέλος αλλά και να διακλαδώνεται και να υπάρχουν διαφορετικά τελειώματα της ιστορίας, ανάλογα με τις επιλογές που έκανε ο παίκτης.

2.3. Επισκόπηση αρχών ανάπτυξης παιχνιδιών

Το ηλεκτρονικό παιχνίδι είναι λογισμικό (software) με γραφικά, ήχο και χειρισμό παιχνιδιού (gameplay), το οποίο λειτουργεί σε υπολογιστή, κονσόλα, κινητά, τάμπλετ ή σε οποιαδήποτε ηλεκτρονική μηχανή. Η ανάπτυξη παιχνιδιού (game developing, gamedev) είναι η διαδικασία δημιουργίας του ηλεκτρονικού παιχνιδιού που διαφέρει από τα άλλα είδη λογισμικού υπολογιστή, κυρίως διότι στοχεύει στην ψυχαγωγία και διασκέδαση του χρήστη μέσω του διαδραστικού χειρισμού, των γραφικών και του ήχου, και όχι στην παραγωγή έργου.

Η ανάπτυξη ενός βιντεοπαιχνιδιού (game developing) είναι μια πολύπλοκη διαδικασία, που μπορεί να χωριστεί στα παρακάτω επιμέρους τμήματα:

Σχεδιασμός παιχνιδιού (Game Design): Είναι οι γενικές ιδέες για τη δημιουργία του παιχνιδιού, για το θέμα και για το είδος του, για το πως θα είναι ο χειρισμός του, για τη δυσκολία του και για το πως αυτές οι ιδέες θα μπορέσουν να υλοποιηθούν σε βιντεοπαιχνίδι.

Προγραμματισμός (programming). Η συγγραφή κώδικα σε γλώσσα προγραμματισμού μέσω της χρήσης μηχανών παιχνιδιών (game engines) και άλλων

εργαλείων, για την υλοποίηση όλων των μηχανισμών του παιχνιδιού, του ελέγχου του χαρακτήρα, της διαδραστικότητας, της συνδεσιμότητας στο διαδίκτυο, της τεχνητής νοημοσύνης και των άλλων χαρακτηριστικών που σχεδιάστηκαν από τον designer. Οι προγραμματιστές δεν αρκεί να παράγουν λειτουργικό κώδικα αλλά πρέπει και να τον βελτιστοποιούν (optimize) ώστε να τρέχει ομαλά και αποτελεσματικά χωρίς κλώματα (bugs) σε ευρύ φάσμα από hardware και όχι μόνο σε πολύ δυνατούς υπολογιστές.

Σχεδιασμός γραφικών: Σχεδιασμός των χαρακτήρων, των αντικειμένων του περιβάλλοντος αλλά και των μενού του παιχνιδιού. Τα γραφικά μπορεί να είναι σε δύο ή σε τρεις διαστάσεις, ή συνδυασμός τους, ανάλογα με τον σχεδιασμό του παιχνιδιού.

Σχεδιασμός κίνησης (animation): Το βιντεοπαιχνίδι βασίζεται στα κινούμενα γραφικά των χαρακτήρων που απαρτίζουν την ιστορία του, αλλά και των αντικειμένων στις διάφορες σκηνές του. Επίσης, μπορεί να περιέχει διάφορα οπτικά εφέ αλλά και ολόκληρες κινηματογραφικές σκηνές (cutscenes). Υπάρχουν τρισδιάστατα και δισδιάστατα animation και η τεχνική του καθενός είναι διαφορετική.

Σχεδιασμός επιπέδων (level design): Το παιχνίδι χωρίζεται σε διάφορα επίπεδα μέσα στα οποία κινείται ο χαρακτήρας που ελέγχει ο παίκτης. Για κάθε επίπεδο σχεδιάζεται η διάταξη και η δομή του και καθορίζονται οι δρόμοι που θα κινείται ο χαρακτήρας και οι αντίπαλοι, τα όρια μέσα στα οποία θα κινούνται, τα εμπόδια στον δρόμο τους, και οι διάφορες λειτουργίες κατά την αλληλεπίδραση του παίκτη με στοιχεία του επιπέδου.

Παραγωγή ήχου (sound design). Ενσωμάτωση στοιχείων ήχου στο παιχνίδι, όπως είναι οι διάλογοι, τα ηχητικά εφέ και οι μουσικές.

Δοκιμή (tester): Πριν την κυκλοφορία του παιχνιδιού και κατά την φάση της ανάπτυξης κάθε μέρος του παιχνιδιού δοκιμάζεται σχολαστικά, με σκοπό την εύρεση τυχόν προβλημάτων στον κώδικα (bugs) ή άλλων θεμάτων που μπορεί να προκύψουν κατά την εκτέλεση που μπορεί να προκαλέσουν προβλήματα στην εμπειρία του τελικού χρήστη, όπως η μειωμένη απόδοση και τα χαμηλά καρτέ ανά δευτερόλεπτο (**fps**). Αυτά αναφέρονται στους developers και διορθώνονται πριν την τελική κυκλοφορία του παιχνιδιού.

Υπάρχει και ο ρόλος του **παραγωγού παιχνιδιού** (game producer) που επιβλέπει και καθοδηγεί όλη την ανάπτυξη και τη συνεργασία των επιμέρους τμημάτων.

Ο **game developer** μπορεί να είναι ένα μοναδικό άτομο που υλοποιεί όλα τα επιμέρους τμήματα της ανάπτυξης μόνος του, είτε μια ομάδα από developers που απαρτίζουν ένα game studio, είτε με διακριτούς ρόλους είτε συνεργαζόμενοι στο ίδιο τμήμα (π.χ. δύο προγραμματιστές, τρεις γραφίστες κλπ.). Τα άτομα ενός studio μπορεί να δουλεύουν στον ίδιο χώρο ή απομακρυσμένα.

Όταν ολοκληρωθεί από τους developers το βιντεοπαιχνίδι, οι εταιρείες διανομείς (**publishers**) αναλαμβάνουν την κυκλοφορία του, ώστε αυτό να φτάσει στους τελικούς χρήστες που θα παίξουν το παιχνίδι. Οι διανομείς, επίσης, αναλαμβάνουν το marketing και την προώθηση με στόχο την εμπορική επιτυχία του παιχνιδιού.

Η παραγωγή του μπορεί να διαρκέσει από αρκετούς μήνες μέχρι και δύο, τρία ή περισσότερα χρόνια για να ολοκληρωθεί.

Η χρηματοδότηση της ανάπτυξης του παιχνιδιού γίνεται με διάφορους τρόπους, από τους οποίους οι πιο συνηθισμένοι είναι η χρηματοδότηση από τους εκδότες, από επενδυτές, από πλήθος μέσω πλατφορμών χρηματοδότησης (crowdfunding) όπως το kickstarter, από κρατική χρηματοδότηση, από σπόνσορες που θα χρησιμοποιήσουν το παιχνίδι για διαφημιστικούς σκοπούς ή από αυτοχρηματοδότηση όταν αφορά ανεξάρτητους παραγωγούς (indie developers).

Παράδειγμα κρατικής χρηματοδότησης στην Ελλάδα είναι το **ταμείο εγγυοδοσίας δανείων επιχειρήσεων παραγωγής οπτικοακουστικών έργων**.

Η ανάπτυξη παιχνιδιού είναι συνδυασμός πολλών τεχνών αλλά και επιστήμης. Τα βιντεοπαιχνίδια θεωρούνται ένα από τα σπουδαιότερα μέσα καλλιτεχνικής δημιουργίας στην ιστορία (Melissinos, 2015).

2.4. Ιστορία ανάπτυξης βιντεοπαιχνιδιών

Η ανάπτυξη βιντεοπαιχνιδιών ξεκίνησε την δεκαετία του 1950 και μέχρι τις αρχές της δεκαετίας του 1970 βρισκόταν σε πολύ νηπιακό στάδιο. Τα πρώτα βιντεοπαιχνίδια δημιουργήθηκαν για ογκώδεις υπολογιστές mainframe και απαιτούσαν γνώσεις ηλεκτρονικών κυκλωμάτων για την παραγωγή τους.

Από τα μέσα της δεκαετίας του 1970 οι μικροεπεξεργαστές αντικατέστησαν την παλαιότερη τεχνολογία που βασιζόταν στα συνδυασμένα τρανζίστορ. Οι υπολογιστές μίκρυναν σε μέγεθος και άρχισαν να κάνουν μαζικά την εμφάνισή τους στα σπίτια των ανθρώπων σε όλο τον κόσμο. Σταδιακά η χρήση των υπολογιστών για ψυχαγωγία παίζοντας βιντεοπαιχνίδια ξεπέρασε την επαγγελματική τους χρήση, όπως η συγγραφή κειμένου, τα υπολογιστικά φύλλα και ο προγραμματισμός.

Την ίδια εποχή άρχισαν να εμφανίζονται στα σπίτια και διάφορες κονσόλες βιντεοπαιχνιδιών που είχαν υποδοχή για κασέτα ROM (ROM cartridge), μέσα στην οποία ήταν γραμμένος ο κώδικας των παιχνιδιών. Τότε, επίσης, γνώρισαν τεράστια άνθηση και οι παιχνιδομηχανές με κερματοδέκτη σε αίθουσες επιχειρήσεων (arcade). Ορόσημο αυτών των παιχνιδομηχανών είναι το παιχνίδι **Pong** που αναπτύχθηκε από την **ATARI** το 1972 (Εικόνα 2.7).

Τότε ήταν σύνηθες ένας απλός game developer να αναπτύσσει μόνος του ένα ολόκληρο παιχνίδι, δηλαδή να ασχολείται και με το προγραμματιστικό μέρος και με τη σχεδίαση των γραφικών, των ήχων, της κίνησης κλπ. Για παράδειγμα, το **Manic Miner** δημιουργήθηκε μόνο από τον **Matthew Smith** και κυκλοφόρησε από την **Bug-Byte** το 1983 για τους οικιακούς υπολογιστές zx spectrum (Εικόνα 2.8). Τα παιχνίδια προγραμματίζονταν σε χαμηλού επιπέδου γλώσσες προγραμματισμού, συνήθως σε κώδικα μηχανής (assembly). Οι προγραμματιστές παιχνιδιών τότε, δεν προτιμούσαν τη χρήση υψηλότερου επιπέδου γλωσσών προγραμματισμού, διότι υστερούσαν σε ταχύτητα και η διαφορά ήταν πολύ αισθητή. Μεγάλο μειονέκτημα ήταν επίσης, ότι ο κώδικας παιχνιδιών έτρεχε αποκλειστικά για τον επεξεργαστή και το μηχάνημα για το οποίο ήταν γραμμένος και για να τρέξει ένα παιχνίδι σε άλλο μηχάνημα έπρεπε ο κώδικάς του να ξαναγραφτεί από την αρχή. Τα γραφικά περιοριζόνταν σε μικρές αναλύσεις οθόνης, με χρήση χρωματικών παλετών λίγων χρωμάτων και συνήθως σχεδιάζονταν με το κάθε εικονοστοιχείο ξεχωριστά (pixel graphics design).



Εικόνα 2.7: Φωτογραφία καμπίνας και στιγμιότυπο του παιχνιδιού "Pong" (1972) της ATARI



Εικόνα 2.8: Το παιχνίδι "Manic Miner" έχει μοναδικό developer τον Matthew Smith

Όμως, με την περαιτέρω ανάπτυξη της υπολογιστικής ισχύος και την αύξηση της ζήτησης για καλύτερα βιντεοπαιχνίδια από το κοινό, ξεκίνησε και η ανάπτυξη βιντεοπαιχνιδιών από ομάδες και studios παραγωγής. Πλέον, αντί για μόνο έναν developer που αναλάμβανε εξ'ολοκλήρου την ανάπτυξη του παιχνιδιού μέχρι την κυκλοφορία του, ομάδες ατόμων που ειδικεύονται σε διαφορετικό τομέα ανάπτυξης ο καθένας, (προγραμματιστές, γραφίστες, μουσικοί κλπ.) συνεργάζονται για να φτιάξουν μεγαλύτερα σε μέγεθος και εντυπωσιακότερα βιντεοπαιχνίδια.

Επίσης, ήδη από τη δεκαετία του 1990 ο κώδικας των παιχνιδιών γραφόταν σε υψηλότερου επιπέδου γλώσσες, κυρίως C και C++, ενώ τα γραφικά σχεδιάζονταν σε λογισμικά επεξεργασίας εικόνας όπως το photoshop.

Λόγω της μεγάλης κερδοφορίας του κλάδου, πολλές εταιρείες παιχνιδιών δημιουργήθηκαν σε όλο τον κόσμο, οι οποίες παρήγαγαν παιχνίδια για υπολογιστές, κονσόλες και τα arcade. Κάποιες από τις εταιρείες ανέπτυξαν και τις δικές τους μηχανές παιχνιδιών, όπως η **Id** που ανέπτυξε την **Doom Engine** το 1993 και την **Quake engine** το 1996, για παιχνίδια shooter πρώτου προσώπου. Η χρήση των game engine ήταν ένα σημαντικό βήμα στην ιστορία του game developing, διότι πλέον οι εταιρείες παραγωγής δεν χρειάζονταν να ξοδεύουν πολύ χρόνο για να φτιάχνουν βιντεοπαιχνίδια από την αρχή, αλλά χρησιμοποιούσαν έτοιμο επαναχρησιμοποιήσιμο κώδικα.

Επιπρόσθετα και τα προγράμματα ανάπτυξης λογισμικού έγιναν πιο φιλικά στο χρήστη με πιο ανεπτυγμένα γραφικά περιβάλλοντα και οι βιβλιοθήκες κώδικα ήταν διαθέσιμες στους προγραμματιστές βιντεοπαιχνιδιών.

Από τότε μέχρι σήμερα η τεχνολογία των υπολογιστών εξελίχθηκε και ο καθένας μπορεί να έχει έναν υπολογιστή με αρκετή ισχύ όχι μόνο για να παίζει βιντεοπαιχνίδια αλλά και για να μπορεί να τα δημιουργεί αν έχει τις γνώσεις. Υπάρχει πληθώρα βιβλίων που διδάσκουν προγραμματισμό βιντεοπαιχνιδιών, ενώ το διαδίκτυο

είναι αστείρευτη πηγή για μαθήματα. Επίσης, σε πολλά πανεπιστήμια στον κόσμο η ανάπτυξη παιχνιδιών έχει μπει στο πρόγραμμα σπουδών, ενώ οι μηχανές παιχνιδιών έχουν γίνει προσβάσιμες σε όλους. Αυτοί είναι και οι λόγοι που την τελευταία δεκαετία πέρα από τις μεγάλες εταιρείες παραγωγής βιντεοπαιχνιδιών, έχουν εμφανιστεί και πάρα πολλά μικρότερα studio και ανεξάρτητοι game developers, με χιλιάδες αξιοσημείωτες παραγωγές.

2.5. Επισκόπηση μηχανής παιχνιδιών

Μία μηχανή παιχνιδιού (game engine) όπως η **Unity engine**, η **Unreal**, η **Crytek**, η **Game Maker** παρέχει στον game developer τις βασικές διεργασίες - λειτουργίες και στοιχεία που απαιτούνται και χρησιμοποιούνται με τον ίδιο τρόπο σε πολλά βιντεοπαιχνίδια, όπως την απεικόνιση αντικειμένων, την εξομίωση φυσικών νόμων, την ανίχνευση συγκρούσεων των αντικειμένων, τους φωτισμούς και τις σκιάσεις, τις εισόδους από το πληκτρολόγιο και το ποντίκι κ.α. Με αυτό τον τρόπο ο developer δεν ασχολείται με τον προγραμματισμό αυτών των βασικών κοινών στοιχείων των βιντεοπαιχνιδιών, τα οποία προϋποθέτουν βαθιές γνώσεις προγραμματισμού, μαθηματικών και φυσικής, αλλά προσθέτει πάνω σε αυτά τα δικά του, ειδικά για το παιχνίδι που σχεδιάζει, στοιχεία όπως γραφικά, 3D ή 2D αντικείμενα, ήχους, κώδικα και άλλα.

Επίσης, μία game engine παρέχει χρήσιμα εργαλεία για ανάπτυξη κώδικα αλλά και δημιουργίας στοιχείων, όπως 3D modeling και animations.

Η ανάπτυξη παιχνιδιών με την χρήση game engine προϋποθέτει πολλά πράγματα, τα βασικότερα εκ των οποίων είναι:

- Γνώση των βασικών αρχών του game developing.
- Επιλογή και απόκτηση της κατάλληλης game engine που να ταιριάζει στις ανάγκες, απαιτήσεις και ικανότητες του δημιουργού. Υπάρχουν ελεύθερες και επί πληρωμή.
- Υπολογιστή με μεγάλες δυνατότητες όσον αναφορά την ισχύ του επεξεργαστή και της κάρτας γραφικών, και με μεγάλη χωρητικότητα σε μνήμη RAM και αποθηκευτική μνήμη για να μπορεί να υποστηρίξει την game engine που έχει επιλεγεί.
- Εξοικείωση με την διεπαφή της (τα μενού, τον editor, τα εργαλεία της).
- Υποστηρικτικό Λογισμικό.
- Δημιουργικότητα.
- Γνώσεις προγραμματισμού.

2.6. Επισκόπηση του UNITY

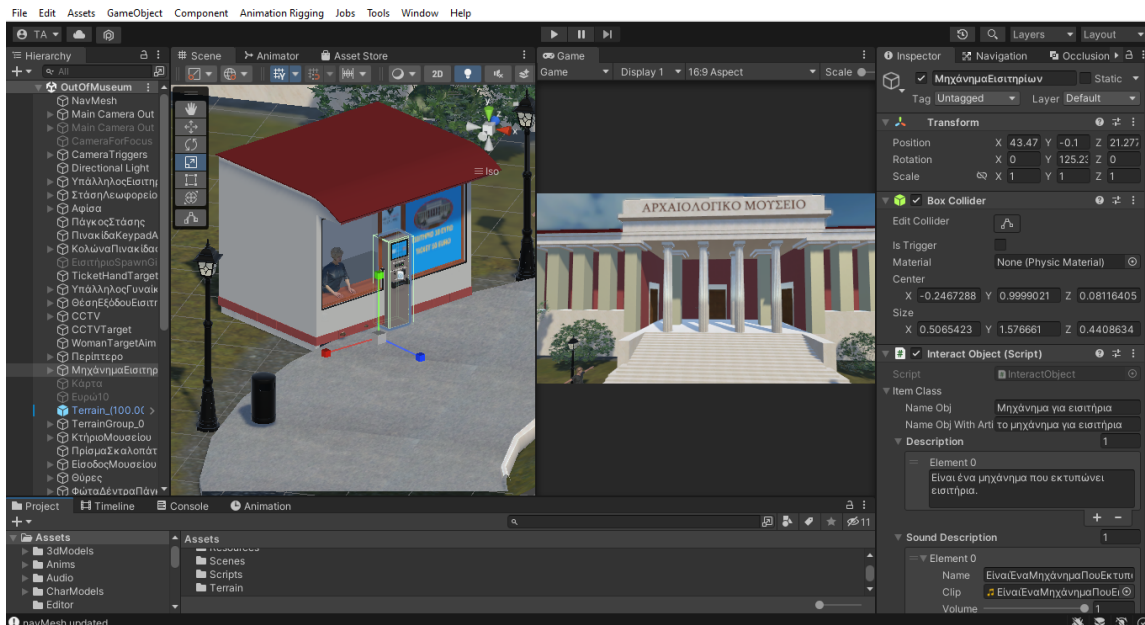
Η πρώτη έκδοση του Unity κυκλοφόρησε το 2005 από την **Unity Technologies** αποκλειστικά για MAC και από τότε συνεχώς εξελίσσεται, με καινούργιες εκδόσεις να κυκλοφορούν κάθε χρόνο με όνομα όπως **Unity 2023.X**. Στην αρχή δεν είχε ιδιαίτερη απήχηση στο κοινό αλλά από το 2010 και μετά απέκτησε μεγάλη δημοφιλία και χρησιμοποιείται από σημαντική μερίδα δημιουργών στη βιομηχανία βιντεοπαιχνιδιών.

Το Unity είναι ένα πολύ ισχυρό εργαλείο και για τους αρχάριους που μόλις ξεκινάνε την ενασχόλησή τους με την ανάπτυξη παιχνιδιών αλλά και για έμπειρους επαγγελματίες που αναπτύσσουν μεγάλες παραγωγές.

Μεγάλα στούντιο χρησιμοποίησαν το Unity για την παραγωγή πολύ γνωστών τίτλων, **Ori and the Blind Forest** (2015) **Moon Studios**, **Microsoft Studios** το 2015, αλλά και μικροί ανεξάρτητοι δημιουργοί το επιλέγουν για να αναπτύξουν τα παιχνίδια τους.

Οι βασικοί παράγοντες αυτής της διάδοσης είναι ο πολύ βολικός οπτικός συντάκτης (visual editor), η υποστήριξη πολλών πλατφορμών για να εξαχθεί το παιχνίδι (PC, Mac, κονσόλες, κινητά) και η πολύ ενεργή κοινότητα που το υποστηρίζει και διαμοιράζει assets μέσω asset store. Η Unity υποστηρίζει εξίσου καλά τόσο την δημιουργία τρισδιάστατων παιχνιδιών όσο και δισδιάστατων παιχνιδιών.

Με τον visual editor του Unity η ανάπτυξη παιχνιδιών γίνεται μια πιο εύκολη διαδικασία σε σχέση με τα κλασικά περιβάλλοντα ανάπτυξης, όπως το visual studio, καθώς το μεγαλύτερο μέρος γίνεται μέσω οπτικής ροής εργασίας (visual workflow). Στον visual editor εμφανίζεται η σκηνή παιχνιδιού (game scene), μέσα στην οποία μπορούμε να προσθέσουμε αντικείμενα παιχνιδιού (game objects) που συνήθως είναι οπτικά αντικείμενα, όπως sprites ή τρισδιάστατα μοντέλα. Το παιχνίδι μπορεί να περιέχει πολλές σκηνές, αλλά κάθε φορά γίνεται επεξεργασία μίας μόνο σκηνής, ενώ οι άλλες είναι αποθηκευμένες. Τα οπτικά αντικείμενα αν είναι βασικά γεωμετρικά σχήματα μπορούν να κατασκευαστούν και μέσα στο Unity, αλλά μπορούν να κατασκευαστούν και σε ειδικά προγράμματα εκτός Unity όπως το blender και στη συνέχεια να εισαχθούν μέσα στην σκηνή. Μετά την εισαγωγή στην σκηνή ενός αντικειμένου, μέσα από την διεπαφή του visual editor μπορούμε πολύ εύκολα να μεταβάλλουμε την θέση του, την γωνία του και το μέγεθός του μέσα στο παράθυρο που εμφανίζεται η σκηνή. Αυτά μπορούν να μεταβληθούν και μέσω του παραθύρου inspector και τις ιδιότητες του component transform (Εικόνα 2.9).



Εικόνα 2.9: Ο visual editor της Unity

Η Unity εφαρμόζει ένα σύστημα αρθρωτό με στοιχεία - εξαρτήματα (modular component) πάνω στα gameobjects. Το σύστημα αυτό παρέχει μεγάλη ευελιξία και επεκτασιμότητα, χαρακτηριστικά των καλά δομημένων έργων λογισμικού και είναι κάποιες μορφής ευέλικτου αντικειμενοστρεφούς προγραμματισμού.

Τα components αυτά μπορούν να θεωρηθούν σαν πακέτα που προστίθενται στα gameobjects και καθορίζουν τη λειτουργικότητα και την εμφάνισή τους. Έτσι το κάθε game object έχει μια συλλογή από components που συνδυάζονται μεταξύ τους, μέσω αναφορών ή γεγονότων (events), μετατρέποντάς το gameobject σε μια πλήρως θεμελιώδης λειτουργική μονάδα μέσα στην σκηνή. Η πολυπλοκότητα της μονάδας αυτής εξαρτάται από τα components που έχουν προστεθεί σε αυτή.

Επιλέγοντας ένα game object είτε από το παράθυρο της σκηνής είτε από το παράθυρο ιεραρχίας (Hierarchy) που εμφανίζονται σε λίστα όλα τα αντικείμενα της σκηνής θα εμφανιστούν τα components που αυτό περιέχει σε ένα άλλο παράθυρο που ονομάζεται επιθεωρητής (inspector). Κάτω από το όνομα του κάθε component εμφανίζονται οι ιδιότητές του, οι οποίες είναι και επεξεργάσιμες.

Τα βασικά διαθέσιμα component της Unity και οι λειτουργίες τους παρουσιάζονται στον Πίνακα 2.10.

Component	Λειτουργία
Transform	Καθορίζει την θέση του αντικειμένου στην σκηνή, την γωνία και το μέγεθός του.

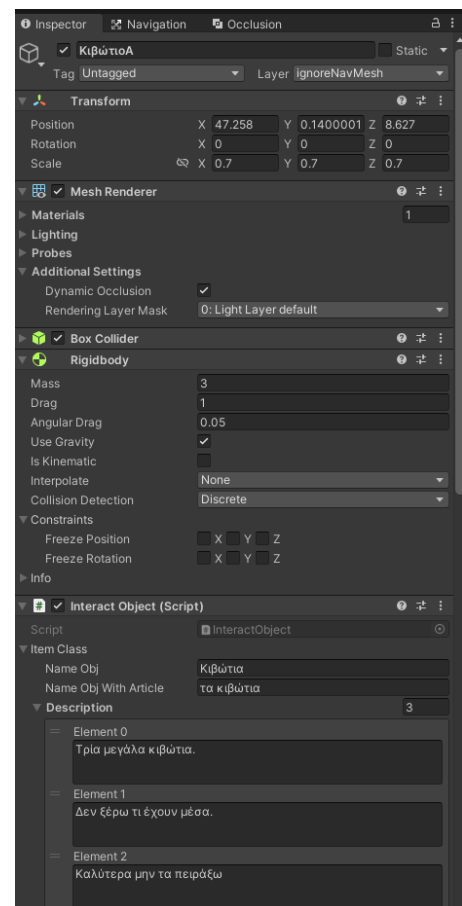
Mesh Renderer	Εμφανίζει το αντικείμενο στην σκηνή, σύμφωνα με τις ιδιότητες του υλικού και του φωτισμού.
Rigidbody	Προσδίδει φυσικές ιδιότητες στο αντικείμενο όπως, μάζα και βάρος που επιτρέπουν φυσικές κινήσεις μέσα στην σκηνή.
Collider	Δίνει το σχήμα στο αντικείμενο για την ανίχνευση συγκρούσεων.
Animator	Δίνει animation (κίνηση) στο αντικείμενο.
Audiosource	Παίζει ήχους και έχει ιδιότητες ήχου όπως την ένταση και τον τόνο.

Πίνακας 2.1: Βασικά component της Unity και οι λειτουργίες τους

Εκτός από τα ανωτέρω βασικά, η Unity διαθέτει εκατοντάδες δικά της components, όμως, το σημαντικό σε αυτό το σύστημα είναι ότι ο προγραμματιστής μπορεί να δημιουργήσει δικό του κώδικα (component script), σε γλώσσα προγραμματισμού C# ώστε να καθορίσει την δική του λειτουργικότητα και λογική, την οποία προσθέτει στο gameObject. Τυπικά τα script είναι κλάσεις της C# που κληρονομούν από μια ανώτερη στην ιεραρχία εσωτερική κλάση της Unity που ονομάζεται MonoBehaviour και αποτελείται από όλες τις ιδιότητες και λειτουργίες που επιτρέπουν να χειρίζεσαι αντικείμενα της Unity. Στην Εικόνα 2.10 εμφανίζονται τα component που έχουν προστεθεί σε ένα game Object με όνομα **ΚιβώτιοΑ** και οι ιδιότητές τους.

Η συγγραφή κώδικα στη C# και η δημιουργία script component, επιτρέπει στον developer να χρησιμοποιήσει και το κλασσικό σύστημα των ιεραρχιών, και τα γνωστά σχεδιαστικά πρότυπα (design patterns) που εφαρμόζονται στον αντικειμενοστρεφή προγραμματισμό.

Η προσθήκη ή η αφαίρεση των component στα αντικείμενα υλοποιείται εύκολα με τον visual editor ενώ το ίδιο component μπορεί να επαναχρησιμοποιηθεί σε πολλά game objects.



Εικόνα 2.10: Στο παράθυρο Inspector με επιλεγμένο το gameObject ΚιβώτιοΑ εμφανίζονται τα component που έχουν προστεθεί σε αυτό.

Πρέπει να σημειωθεί ότι αν και η δημιουργία στη Unity γίνεται κυρίως μέσω του visual editor, στον πυρήνα του ο κώδικάς του έχει πολλές διασυνδέσεις με βιβλιοθήκες κώδικα, όπως άλλα κλασικά περιβάλλοντα ανάπτυξης, για παράδειγμα το eclipse, γεγονός, όμως, που δεν απασχολεί τον τελικό χρήστη.

Ο Κώδικας 2.1 δείχνει το Script **CrateFall.cs** από το παιχνίδι που αναπτύχθηκε και θα παρουσιαστεί στα επόμενα κεφάλαια. Το script αυτό έχει προστεθεί στο gameObject **ΚιβώτιοΓ**, που προσδίδει λειτουργικότητα σε ένα κιβώτιο που βρίσκεται στη σκηνή. Συγκεκριμένα, όταν το κιβώτιο πέσει και βρεθεί κάτω από ένα συγκεκριμένο ύψος, ο κεντρικός χαρακτήρας θα ακινητοποιηθεί και θα πει μία φράση. Η κλάση βλέπουμε ότι κληρονομεί από την **MonoBehaviour** και διαθέτει τρεις μεθόδους. Οι μέθοδοι **Start()** και **Update()** δημιουργούνται αυτόματα κάθε φορά που στη Unity δημιουργούμε καινούργιο Script. Βλέπουμε επίσης ότι στην αρχή της κλάσης με την εντολή **using** συμπεριλαμβάνουμε κλάσεις και μεθόδους από έτοιμες βιβλιοθήκες.

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
public class CrateFall : MonoBehaviour  
{  
    private bool hasFallen;  
    void Start()  
    {  
    }  
    void Update()  
    {  
        if ((!hasFallen) && (transform.position.y < 0.8f))  
        {  
            hasFallen = true;  
            Managers.Player.playerControl.stopMoving();  
            Managers.Scene.setCutsceneOrPlayerAnim(true);  
            Managers.Dialogue.StartDialogue(25);  
        }  
    }  
    public void setHasFallen()  
    {  
        hasFallen = true;  
    }  
}
```

Κώδικας 2.1: Το script *CrateFall.cs* που έχει προστεθεί στο αντικείμενο κιβώτιοΓ

3.Μεθοδολογία

3.1. Εισαγωγή

Όπως προαναφέρθηκε στόχος της διπλωματικής εργασίας είναι η ανάπτυξη ενός προτύπου στη μηχανών παιχνιδιών Unity και σε γλώσσα προγραμματισμού C# που να περιέχει τους μηχανισμούς που συνθέτουν ένα adventure βιντεοπαιχνίδι 3D p&c τρίτου προσώπου. Πάνω σε αυτό το πρότυπο, μπορεί να βασιστεί η δημιουργία ενός σύγχρονου παιχνιδιού adventure, που οι δημιουργοί του έχουν ως στόχο την εμπορική επιτυχία. Για να μπορέσει να παρουσιαστεί αυτό το πρότυπο και οι μηχανισμοί που περιέχει, αναπτύχθηκε ένα απλό παιχνίδι, μικρό σε μέγεθος και χωρίς πολύπλοκο σενάριο. Όμως, το παιχνίδι αυτό με τίτλο "Επίσκεψη στο μουσείο" είναι ικανό να επιδείξει σε όσους το δοκιμάσουν, τις δυνατότητες που μπορεί να προσφέρει το πρότυπο.

Ο λόγος που δεν αναπτύχθηκε ένα ολοκληρωμένο εμπορικό παιχνίδι, είναι ότι αυτό προϋποθέτει κατά πρώτον την ύπαρξη ενός καλού σεναρίου, δηλαδή μιας μεγάλης ιστορίας που να περιέχει ενδιαφέροντες ανθρώπινους χαρακτήρες, έξυπνους γρίφους, διαλόγους, τοπία, και αφηγήσεις και κατά δεύτερον, μιας ομάδας studio παραγωγής, με άτομα που έχουν ειδικευση στα 3D γραφικά, στο 3D animation, στον ήχο, στην σκηνοθεσία κλπ.

Το παιχνίδι αναπτύχθηκε αποκλειστικά από τον συντάκτη, με τα περισσότερα από τα γραφικά και τους ήχους να είναι φτιαγμένα από τον ίδιο ή να έχουν ληφθεί από ελεύθερες πηγές στο διαδίκτυο. Ένα μικρό μόνο μέρος από τα γραφικά είναι αγορασμένο από αγορές στοιχείων (assets) στο διαδίκτυο. Για όλα τα στοιχεία του παιχνιδιού έγινε προσπάθεια να υπάρχουν τα κατάλληλα δικαιώματα χρήσης τους στο πλαίσιο αυτής της εργασίας.

3.2. Προσωπικό υπόβαθρο

Ο συντάκτης της παρούσας εργασίας είναι λάτρης των ηλεκτρονικών παιχνιδιών και ιδιαίτερα των adventure από νεαρή ηλικία. Έχει βιώσει την χρυσή εποχή των παιχνιδιών adventure και έχει καταφέρει να ολοκληρώσει πολλά από τα γνωστά adventure όλων των εποχών.

Εκτός από την εμπειρία αυτή, μελετήθηκαν οι μηχανισμοί των σύγχρονων P&C adventure παιχνιδιών και έγινε προσπάθεια το πρότυπο που παρουσιάζεται να περιέχει πανομοιότυπους με αυτούς τους μηχανισμούς, ώστε ένας απλός παίκτης που στο παρελθόν έχει παίξει adventure games να είναι ήδη εξοικειωμένος με τον τρόπο χειρισμού του παιχνιδιού που θα αναπτυχθεί με βάση το υπό κατασκευή πρότυπο. Παρόμοιους χειρισμούς και μηχανισμούς έχουν τα εμπορικά adventure της σειράς

Syberia της **Microids** (Εικόνα 3.1), **Trüberbrook** της **Headup Games** (2019), της σειράς **Longest Journey** της **Funcom** .

Σχετικά με το προγραμματιστικό κομμάτι ο συντάκτης ασχολείται γενικά με την ανάπτυξη παιχνιδιών από το 2010, και ειδικά με τη Unity από το 2017, έχοντας δημιουργήσει διάφορα μικρά παιχνίδια τα οποία έχουν ανεβεί σε marketplace για κινητά και υπολογιστές. Για την εκμάθηση της Unity και βελτίωση των τεχνικών ανάπτυξης παιχνιδιού, μελετήθηκε το βιβλίο του **Alex Okita**, "Learning C# Programming with Unity 3D" (2017) και παρακολούθησε η σειρά "How to make a Video Game" του καναλιού **Brackeys** στο youtube.



Εικόνα 3.1: Στιγμιότυπο από το σύγχρονο 3D r&c adventure "Syberia 3"

3.3. Unity και άλλα εργαλεία ανάπτυξης του προτύπου

Για την ανάπτυξη του προτύπου και στη συνέχεια την υλοποίηση του παιχνιδιού "**Επίσκεψη στο μουσείο**" επιλέχθηκε να χρησιμοποιηθεί η μηχανή παιχνιδιών Unity και συγκεκριμένα η ελεύθερη έκδοση **2021,3,12f1**. Η Unity εγκαταστάθηκε στον προσωπικό μου φορητό υπολογιστή με λειτουργικό win 10, επεξεργαστή Intel(R) Core(TM) i5-3210M 2.50GHz, 8GB RAM και κάρτα γραφικών NVIDIA GeForce 610M και δεν χρησιμοποιήθηκε κανένας άλλος υπολογιστής.

Για τη συγγραφή και επεξεργασία του κώδικα εγκαταστάθηκε η ελεύθερη έκδοση 16.11.20 του **Microsoft Visual Studio Community 2019**, η οποία συνεργάζεται άριστα με την Unity. Το Visual studio εκτός από την υποβοήθηση που δίνει στη συγγραφή κώδικα της γλώσσας C#, με αυτόματες συμπληρώσεις, αυτόματες μορφοποιήσεις, διορθώσεις συντακτικών λαθών κ.λ.π., δίνει και τη δυνατότητα

εκτέλεσης λειτουργιών αποσφαλμάτωσης (Debugging), όπως είναι η εκτέλεση κώδικα βήμα βήμα και τοποθέτηση διακοπών (breakpoint) μέσα στον κώδικα, όπου η εκτέλεση παύει και δίνεται η δυνατότητα παρακολούθησης των μεταβλητών του κώδικα.

Εντός του unity εκτός από τα προεγκατεστημένα εργαλεία και πακέτα που χρησιμοποιήθηκαν, όπως το **terrain tools 4.0.3**, κατεβάστηκαν και εγκαταστάθηκαν κάποια πακέτα που παρείχαν επιπλέον χρήσιμες λειτουργίες και εργαλεία για την ανάπτυξη του έργου:

Animation rigging v.1.1.1.. Χρησιμοποιήθηκε για την ξεχωριστή κίνηση μερών του σώματος των χαρακτήρων από το βασικό animation τους, όπως η κίνηση του κεφαλιού προς την κατεύθυνση ενός αντικειμένου.

Post Processing v3.2.2. Περιέχει εφέ και φίλτρα εικόνας και χρησιμοποιήθηκε για την προσθήκη του εφέ Depth of field.

Pro builder v5.0.6. Με το εργαλείο αυτό σχεδιάστηκε το κτίριο του μουσείου και διάφορες άλλες κατασκευές.

TextMeshPro v3.0.6. Βελτιώνει την εμφάνιση του κειμένου μέσα στο παιχνίδι.

Timeline v1.6.4. Διευκόλυνε την κατασκευή κινηματικών σκηνών του παιχνιδιού.

Universal RP v12.1.7. Υπάρχουν τρεις τρόποι απεικόνισης τρισδιάστατων γραφικών στην Unity. Ο **Unity Standard Pipeline** που είναι ο πιο απλός και ενσωματωμένος και προεπιλεγμένος, ο **Universal Render Pipeline** που δίνει βελτιωμένα γραφικά και υποστηρίζεται από πολλές συσκευές και ο βέλτιστος HDRP που δίνει ρεαλιστικά γραφικά, αλλά υποστηρίζεται μόνο από τελευταίας τεχνολογίες συσκευές, όπως το playstation 5. Στο πρότυπο χρησιμοποιήθηκε το **Universal Render Pipeline**.

Unity NavMesh. Το εργαλείο που προσφέρει η Unity για την κίνηση των χαρακτήρων του παιχνιδιού, την εύρεση των βέλτιστων διαδρομών και αποφυγή εμποδίων με χρήση τεχνητής νοημοσύνης. Από την έκδοση του 2021 έχει ενσωματωθεί στην Unity.

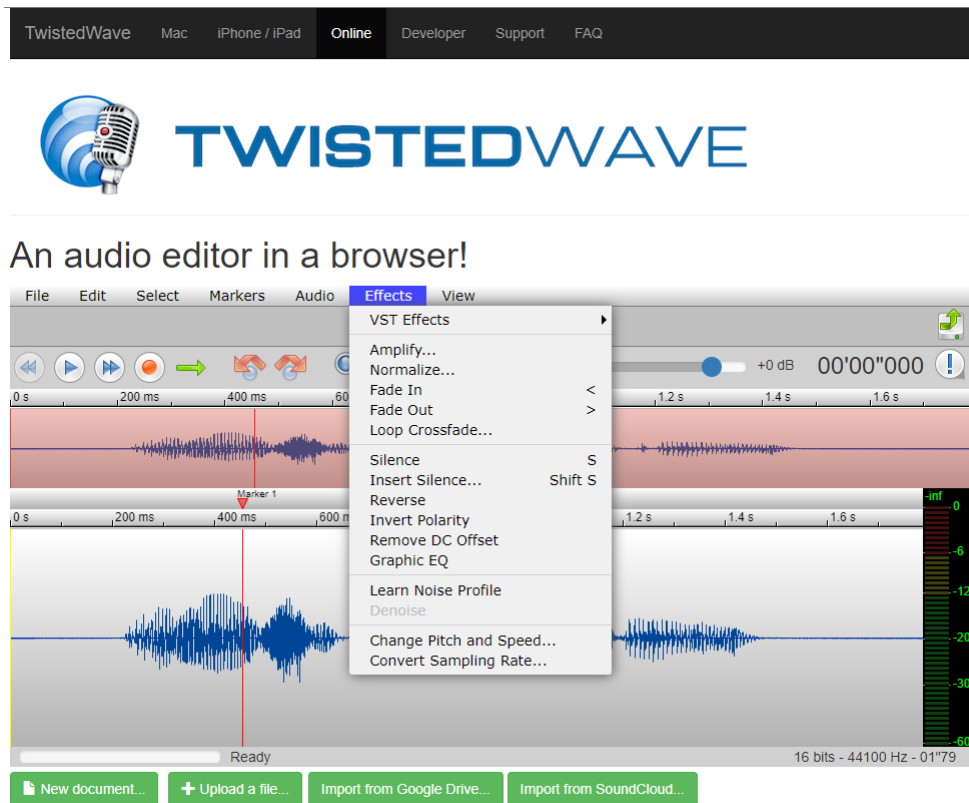
Quick Outline του **Chris Nolet.** Εργαλείο για τη δημιουργία περιγράμματος σε 3D αντικείμενα.

Για τη σχεδίαση και επεξεργασία δισδιάστατων εικόνων και φωτογραφιών, κυρίως της μορφής PNG που υποστηρίζει διαφάνειες, εγκαταστάθηκε η ελεύθερη έκδοση **5.0.7** του **paint.net**.

Επίσης, έγινε σε πολύ μικρό βαθμό χρήση του ελεύθερου λογισμικού **blender v.2.93.5** για δημιουργία και επεξεργασία 3D μοντέλων και 3D animation, η χρήση του οποίου απαιτεί πολύ εξειδικευμένες γνώσεις, και για αυτό το λόγο προτιμήθηκε τα

περισσότερα 3D αντικείμενα να κατασκευαστούν εντός της Unity, που η χρήση της είναι πιο απλή.

Τέλος, για την επεξεργασία όλων των ήχων χρησιμοποιήθηκε η online εφαρμογή επεξεργασίας ήχου **Twisted Wave** (Εικόνα 3.2).



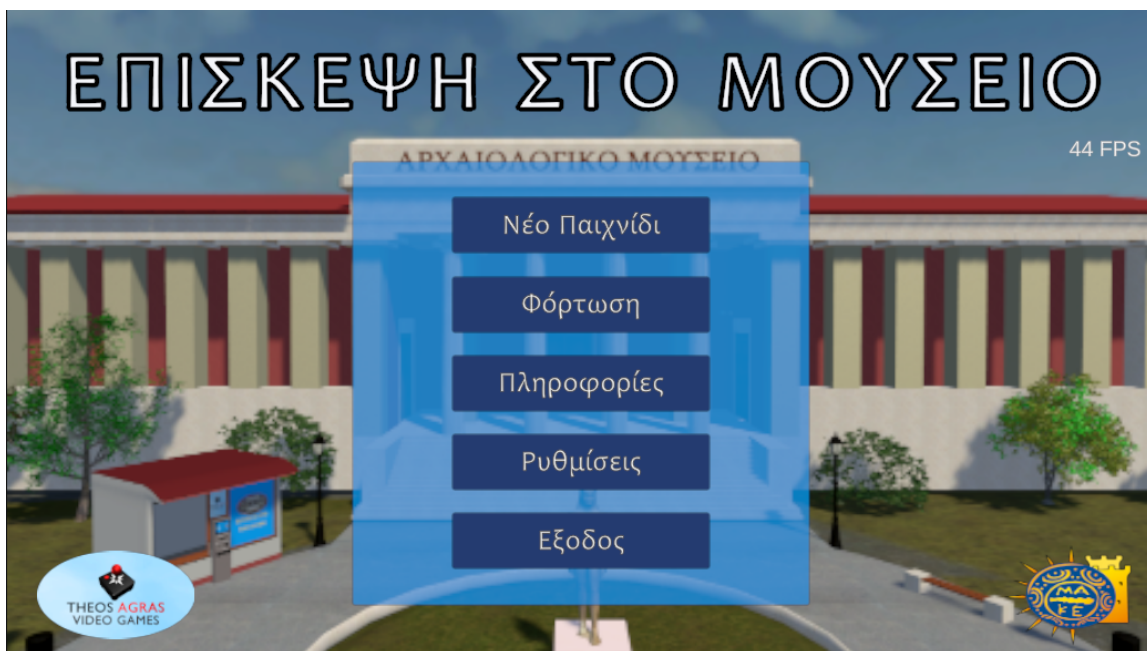
Εικόνα 3.2: Επεξεργασία ήχου με το online πρόγραμμα “Twisted Wave”

4. Σχεδίαση προτύπου και υλοποίηση παιχνιδιού “Επίσκεψη στο μουσείο”

4.1. Περιγραφή του παιχνιδιού “Επίσκεψη στο μουσείο”

Το παιχνίδι όπως αναφέρθηκε παραπάνω, δημιουργήθηκε με αποκλειστικό σκοπό να κάνει επίδειξη των μηχανισμών του προτύπου που αναπτύχθηκε για **3D point and click adventure** παιχνίδια.

Το παιχνίδι ξεκινάει με την οθόνη εισαγωγής, η οποία αποτελείται από ένα τρισδιάστατο λογότυπο του δημιουργού και από το σήμα του Πανεπιστημίου Μακεδονίας. Στη συνέχεια, εμφανίζεται το κεντρικό μενού. Σε αυτό υπάρχει μουσική υπόκρουση και κινούμενος φόντος, με εικόνες που εναλλάσσονται. Οι επιλογές στο κεντρικό μενού είναι οι εξής: **Νέο παιχνίδι**, η **Φόρτωση** σε προηγούμενο αποθηκευμένο σημείο, οι **Πληροφορίες** για το παιχνίδι, οι **Ρυθμίσεις** και η **Έξοδος** από το παιχνίδι (Εικόνα 4.1).

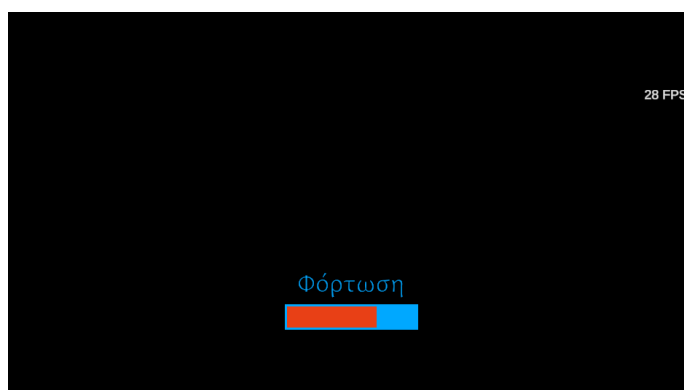


Εικόνα 4.1: Η αρχική οθόνη του παιχνιδιού με το κεντρικό μενού

Η επιλογή των ρυθμίσεων οδηγεί σε ένα δεύτερο μενού επιλογών, όπου ο χρήστης μπορεί να προσαρμόσει την ποιότητα των γραφικών και την ανάλυση στις αποδόσεις του υπολογιστή του και μπορεί να ρυθμίσει την ένταση της μουσικής του παιχνιδιού.

Με την επιλογή νέο παιχνίδι ξεκινάει το κυρίως παιχνίδι. Κατά τη φόρτωση της αρχικής σκηνής αλλά και κατά την εναλλαγή των σκηνών, εμφανίζεται οθόνη με

μπάρα φόρτωσης, που γεμίζει ανάλογα με το πόσα δεδομένα της σκηνής έχουν φορτώσει στη μνήμη και πόσα απομένουν ακόμα για να ολοκληρωθεί η φόρτωση (Εικόνα 4.2).



Εικόνα 4.2: Οθόνη κατά την φόρτωση των σκηνών

Η πρώτη σκηνή του παιχνιδιού είναι εξωτερικά ενός μουσείου μέσα σε ένα αστικό περιβάλλον. Στην είσοδο του μουσείου υπάρχουν μαρμάρινα σκαλοπάτια και μπροστά από το κτίριο του μουσείου υπάρχει ένα μεγάλο προαύλιο με έναν κυκλικό διάδρομο για τους επισκέπτες. Στο κέντρο του διαδρόμου υπάρχει μια πράσινη πλατεία με το άγαλμα ενός κούρου. Υπάρχουν στον χώρο κολώνες φωτισμού, πάγκοι, κάδοι σκουπιδιών και ένα μεγάλο περίπτερο για την έκδοση εισιτηρίων, μέσα στο οποίο κάθετα μία υπάλληλος. Υπάρχουν επίσης, τρία κιβώτια παρατημένα στο αριστερό μέρος του διαδρόμου. Εξωτερικά των διαδρόμων ο χώρος είναι καλυμμένος με γκαζόν και είναι γεμάτος με δέντρα και θάμνους. Η σχεδίαση του κτιρίου και του χώρου μπροστά του είναι εμπνευσμένη από το εθνικό αρχαιολογικό μουσείο της Αθήνας.

Μπροστά από το προαύλιο βρίσκονται το πεζοδρόμιο και ο δρόμος, τον οποίο διασχίζουν αυτοκίνητα. Στο πεζοδρόμιο υπάρχουν προστατευτικά κολωνάκια, μία στάση λεωφορείου και μία πινακίδα δρόμου με πληροφορίες. Δεξιά από το προαύλιο υπάρχει χώρος στάθμευσης με σταθμευμένα αυτοκίνητα και ακόμα δεξιότερα είναι μια διασταύρωση με φανάρι πέρα από την οποία υπάρχουν πολυκατοικίες. Η Εικόνα 4.3 δείχνει την εξωτερική σκηνή του παιχνιδιού.

Το παιχνίδι ξεκινάει με μια κινηματική σκηνή όπου ο κεντρικός χαρακτήρας καταφθάνει στο μουσείο με λεωφορείο. Η κάμερα είναι πανοραμική και κάνει μία κίνηση από το κτίριο του μουσείου μέχρι την στάση του λεωφορείου όπου μετά την αποβίβαση του χαρακτήρα εστιάζει σε αυτόν και τον ακολουθεί. Υπάρχει δυνατότητα παράλειψης της κινηματικής σκηνής με αριστερό κλικ ή με esc.



Εικόνα 4.3: Η πρώτη σκηνή του παιχνιδιού, εξωτερικά του μουσείου

Ο χειρισμός του παιχνιδιού γίνεται όπως και σε όλα τα **point and click** παιχνίδια μετακινώντας τον δείκτη του ποντικιού και κάνοντας αριστερό και δεξιό κλικ σε διάφορα σημεία στην οθόνη.

Πριν αρχίσει ο κανονικός χειρισμός του παιχνιδιού, ο παίκτης επιλέγει αν θα ξεκινήσει το παιχνίδι με εκμάθηση ή χωρίς. Με την εκμάθηση ο παίκτης μαθαίνει τους χειρισμούς του παιχνιδιού με μια σειρά από βήματα, πηγαίνοντας στο επόμενο αφού ολοκληρώσει το προηγούμενο, η περιγραφή των οποίων εμφανίζεται με μήνυμα στην οθόνη. Τα βήματα εκμάθησης είναι τα εξής:

Αριστερό κλικ: Μετακινείται ο χαρακτήρας στο σημείο αν το σημείο είναι προσβάσιμο, αλλιώς πηγαίνει στο πιο κοντινό επιτρεπτό σημείο. Η κίνηση γίνεται από τον συντομότερο δρόμο και αποφεύγοντας τα εμπόδια. Ακριβώς στο σημείο που έγινε κλικ εμφανίζεται μία γκρίζα σφαίρα που μεγαλώνει και εξαφανίζεται, σαν σημάδι.

Διπλό αριστερό κλικ: Μετακίνηση στο σημείο τρέχοντας.

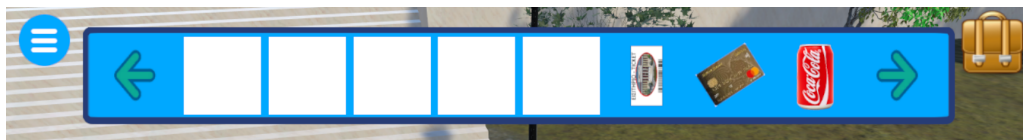
Μετακίνηση του δείκτη πάνω σε σημεία ενδιαφέροντος: Όταν ο δείκτης μετακινηθεί σε ένα αντικείμενο ενδιαφέροντος, το αντικείμενο κοκκινίζει περιμετρικά και ο δείκτης αλλάζει σχήμα ανάλογα την ενέργεια που θα γίνει με αριστερό κλικ στο αντικείμενο. Η ενέργεια αυτή αναγράφεται στο κεντρικό μέρος της οθόνης και το κεφάλι του χαρακτήρα στρέφεται προς την κατεύθυνση που βρίσκεται το αντικείμενο. Αν το σημείο ενδιαφέροντος είναι ένα αντικείμενο προς εξερεύνηση, ο δείκτης γίνεται μεγεθυντικός φακός και με το κλικ ο χαρακτήρας το πλησιάζει και το περιγράφει. Η περιγραφή εμφανίζεται στην οθόνη και ακούγεται φωνητικά.

Χρησιμοποίηση αντικειμένων του περιβάλλοντος: Αν ο χρήστης μετακινήσει τον δείκτη σε αντικείμενο που μπορεί να χρησιμοποιηθεί, ο δείκτης

αλλάζει σε γρανάζι και με αριστερό κλικ ο χαρακτήρας το πλησιάζει και το χρησιμοποιεί.

Συλλογή αντικειμένων: Αν το αντικείμενο ενδιαφέροντος είναι συλλέξιμο, ο δείκτης αλλάζει σε χεράκι και με αριστερό κλικ ο χαρακτήρας το συλλέγει και το τοποθετεί στο αποθετήριο (inventory).

Αποθετήριο: Η προβολή των αντικειμένων που έχει στην κατοχή του ο παίκτης, γίνεται με αριστερό κλικ στο σάκο πάνω δεξιά της οθόνης. Αυτά εμφανίζονται στο πάνω μέρος της οθόνης (Εικόνα 4.4).



Εικόνα 4.4: Το αποθετήριο (inventory) με τα αντικείμενα που έχουν περισυλλεχθεί

Εξερεύνηση αντικειμένων: Με δεξί κλικ σε αντικείμενα ενδιαφέροντος στο περιβάλλον, ή στα πράγματα που υπάρχουν στο αποθετήριο γίνεται η εξερεύνηση τους και η περιγραφή τους εμφανίζεται στην οθόνη.

Χρησιμοποίηση αντικειμένων από αποθετήριο: Με αριστερό κλικ στα αντικείμενα που έχει στην κατοχή του ο παίκτης, ο δείκτης αλλάζει σε γρανάζι μαζί με την εικόνα του αντικειμένου, και με αριστερό κλικ πάνω σε άλλο αντικείμενο ενδιαφέροντος γίνεται συνδυασμός τους ή χρησιμοποίηση του ενός αντικειμένου πάνω στο άλλο. Αν ο συνδυασμός είναι αδύνατος τότε εμφανίζεται και ακούγεται αντίστοιχο μήνυμα.

Συνομιλία με άλλους χαρακτήρες: Αν ο παίκτης μετακινήσει τον δείκτη πάνω σε έτερο χαρακτήρα, αυτός αλλάζει σε εικονίδιο διαλόγου και με αριστερό κλικ, ο κεντρικός χαρακτήρας θα πλησιάσει τον έτερο και θα ξεκινήσει διάλογος. Στην μέση του διαλόγου μπορεί να εμφανιστούν επιλογές με το ποια φράση θα ειπωθεί. Όλοι οι διάλογοι εμφανίζονται πάνω στον χαρακτήρα που μιλάει, ενώ ακούγονται τα φωνητικά αντίστοιχα (voice over). Οι χαρακτήρες όταν μιλούν, κάνουν και χειρονομίες, ενώ η γυναίκα χαρακτήρας ανοιγοκλείνει και το στόμα συγχρονισμένα με τη φωνή της (**lip sync**). Ο συγχρονισμός της κίνησης του στόματος του κεντρικού χαρακτήρα με την φωνή του παραλήφθηκε σε αυτό το παιχνίδι, διότι ενώ είναι εφικτός τεχνικά, ήθελε πολύ χρόνο για να επεξεργαστεί το 3D μοντέλο του στόματος.

Πάνω αριστερά στην οθόνη υπάρχει εικονίδιο για πρόσβαση στο μενού εντός παιχνιδιού το οποίο αποτελείται από τις ακόλουθες επιλογές: **Αποθήκευση** του παιχνιδιού στο σημείο ακριβώς που βρίσκεται, η **φόρτωση** σε προηγούμενο αποθηκευμένο σημείο, οι **ρυθμίσεις** που είναι όμοιες με τις ρυθμίσεις του κεντρικού μενού, η **επιστροφή** στην κανονική ροή του παιχνιδιού και η **έξοδος** στην αρχική

οθόνη του παιχνιδιού. Με κλικ στην αποθήκευση, στην φόρτωση ή στην έξοδο, εμφανίζεται ερώτηση επιβεβαίωσης της ενέργειας.

Κατά τη διάρκεια του παιχνιδιού υπάρχει η δυνατότητα προσαρμογής της απόστασης της κάμερας από τον χαρακτήρα πατώντας τα πλήκτρα + και -. Με το πλήκτρο * επανέρχεται στην προκαθορισμένη απόσταση.

Αφού αποβιβαστεί ο κεντρικός χαρακτήρας από το λεωφορείο μπορεί να περιπλανηθεί στους εξωτερικούς χώρους του μουσείου, να συνομιλήσει με έναν αθλούμενο που τρέχει μπροστά από το μουσείο και να πάρει πληροφορίες από μια υπάλληλο που βρίσκεται μέσα σε ένα περίπτερο. Μπορεί επίσης να εκδώσει εισιτήριο με μετρητά ή με κάρτα από το μηχάνημα έκδοσης.

Στη συνέχεια, ανεβαίνει τα σκαλοπάτια του κτιρίου και η κάμερα λήψης αλλάζει σε διαφορετική γωνία, ενώ εμφανίζεται και η κεντρική είσοδος του μουσείου. Δίπλα από την είσοδο υπάρχει μία κάμερα ασφαλείας που ακολουθεί τον χαρακτήρα ενώ η είσοδος από την πλαϊνή πόρτα είναι αδύνατη. Αν επιχειρήσει να εισέλθει χωρίς εισιτήριο από την κεντρική είσοδο, ο φύλακας τον σταματάει και του ζητάει να του επιδείξει το εισιτήριο. Μετά την επίδειξη είναι δυνατό το άνοιγμα της πόρτας και η είσοδος στο μουσείο.

Το εσωτερικό του μουσείου είναι η δεύτερη σκηνή του παιχνιδιού όπως φαίνεται στην Εικόνα 4.5, όπου ο παίκτης μπορεί να περιπλανηθεί εντός μιας μεγάλης αίθουσας και να εξερευνήσει διάφορα αρχαία αντικείμενα που εκτίθενται, παίρνοντας πληροφορίες για αυτά.



Εικόνα 4.5: Στιγμιότυπο από την εσωτερική σκηνή του μουσείου

4.2. Η φάση της σχεδίασης του λογισμικού του προτύπου - Αρχές σχεδίασης

Σε αυτή την φάση που προηγείται της ανάπτυξης του κώδικα του έργου, έγινε προσπάθεια να εφαρμοστούν κάποιες αρχές μεγάλου έργου λογισμικού που θα διέπουν όλο το υπό κατασκευή πρότυπο adventure game, ώστε στο τέλος το αποτέλεσμα να διακρίνεται από “καλή ποιότητα λογισμικού” (Χατζηγεωργίου, 2005, κεφ 5). Δόθηκε μεγάλη βαρύτητα στη σχεδίαση, καθώς σε μεγάλο βαθμό αυτή καθορίζει την ευκολία να πραγματοποιούνται αλλαγές και προσθήκες στην φάση της ανάπτυξης, πράγμα που είναι βασική επιδίωξη για έναν καλό προγραμματιστή.

Κατά τη σχεδίαση το έργο τμηματοποιείται σε επιμέρους υποσυστήματα (ή μονάδες λογισμικού) ώστε να καθίσταται πιο εύκολη η ανάπτυξη του και η συντήρησή του. Κατά τη σχεδίαση προσδιορίζεται το τι θα κάνει το κάθε υποσύστημα και ο τρόπος που θα συνεργάζονται μεταξύ τους ώστε να παραχθεί το τελικό αποτέλεσμα.

Επειδή το πρότυπο αυτό έχει πολύ μεγάλο μέγεθος και περιέχει πολλά υποσυστήματα παρακάτω θα εξεταστούν μόνο τα βασικότερα από αυτά, καθώς και οι βασικές αρχές που το διέπουν.

4.2.1. Οργάνωση των στοιχείων του έργου και ονοματοδοσία

Από την αρχή του έργου στη unity τηρήθηκε ένα σύστημα οργάνωσης των στοιχείων (assets) που το αποτελούν, με την τοποθέτηση των αρχείων αυτών σε φακέλους, ανάλογα με το είδος τους. Δημιουργήθηκαν ξεχωριστοί φάκελοι για κάθε είδους στοιχείου και μέσα σε αυτούς υποφάκελοι με ομάδες σχετιζόμενων στοιχείων. Οι βασικοί φάκελοι του έργου φαίνονται στον Πίνακα 4.1.

Φάκελος	Assets που περιέχονται στον φάκελο
3dModels	Όλα τα αντικείμενα 3D τύπου fbx ή 3ds που δημιουργήθηκαν εκτός unity. Κάθε 3D μοντέλο είναι σε δικό του υποφάκελο ο οποίος περιέχει επίσης και τα αρχεία των υλικών και των υφών τους.
Anims	Όλα τα αρχεία κίνησης μαζί με τα αρχεία που κατασκευάστηκαν από το εργαλείο timeline.
Audio	Τα ηχητικά εφέ, τα φωνητικά και οι μουσικές, όλα σε μορφή mp3.
CharModels	Τα 3D μοντέλα των ανθρώπινων χαρακτήρων μαζί με τους διαχειριστές κίνησης τους (animators).

Graphics	Διάφορα αρχεία εικόνας που χρησιμοποιούνται για τις δισδιάστατες απεικονίσεις στα μενού του παιχνιδιού αλλά και εντός του παιχνιδιού.
Materials	Στοιχεία για τα υλικά και τις υφές των τρισδιάστατων αντικειμένων που δημιουργήθηκαν εντός του unity.
Prefabs	Είναι τα αντικείμενα της unity τα οποία εκτός από την 3D μορφή τους έχουν προσκολλημένα πάνω τους διάφορα components που τους προσδίδουν ιδιότητες και μπορούν να φορτωθούν ξεχωριστά στην σκηνή.
Resources	Ο ειδικός αυτός φάκελος στην unity περιέχει στοιχεία που φορτώνονται γρηγορότερα κατά την εκτέλεση του παιχνιδιού, και όχι κατά την φόρτωση της σκηνής, όπως τα αρχεία εικόνας για τον δείκτη του ποντικιού και για τα αντικείμενα στο αποθετήριο (inventory).
Scenes	Τα αρχεία για τις τρεις σκηνές (scenes) του παιχνιδιού: Την αρχική με την εισαγωγή και το κεντρικό μενού, την σκηνή εξωτερικά του μουσείου και την σκηνή εντός του μουσείου.
Scripts	Όλα τα αρχεία κώδικα του παιχνιδιού γραμμένα σε γλώσσα C# και με κατάληξη cs.
VariusData	Περιέχονται αρχεία που δημιουργήθηκαν από την ίδια την unity με τα δεδομένα που χρησιμοποιούνται από τα διάφορα εργαλεία της, όπως το terrain, URP, ProBuilder.

Πίνακας 4.1: Οι φάκελοι αρχείων με assets του λογισμικού έργου

Αυτή η πρακτική οργάνωσης που σχεδιάστηκε και εφαρμόστηκε από την αρχή του έργου είναι πολύ σημαντική, γιατί ενισχύει την συντηρησιμότητα του έργου και την παραγωγικότερη χρησιμοποίηση των στοιχείων του.

Μια άλλη αρχή που εφαρμόστηκε στο έργο, σχετίζεται με την ονοματοδοσία των κλάσεων και των ιδιοτήτων τους. Όλα τα ονόματα περιγράφουν τον σκοπό ή την λειτουργικότητα τους και για αυτό το λόγο τις περισσότερες φορές είναι μακροσκελή και αποτελούνται από πάνω από μία λέξεις. Επίσης, χρησιμοποιήθηκε η τεχνική **camelCase** όπου τα ονόματα είναι γραμμένα με πεζά γράμματα εκτός από το πρώτο γράμμα των λέξεων που συναποτελούν το όνομα. Οι λέξεις αυτές εννοείται πως δεν χωρίζονται με κενό, αλλά το κεφαλαίο γράμμα τις κάνει διακριτές. Για παράδειγμα η μεταβλητή **initialAgentsSpeed** περιγράφει την αρχική ταχύτητα του πράκτορα, δηλαδή του κεντρικού χαρακτήρα. Για ευκολία στην ανάγνωση του κώδικα, τα ονόματα όλων των κλάσεων ξεκινούν με κεφαλαίο γράμμα, ενώ των ιδιοτήτων τους (μεταβλητές και μέθοδοι) με μικρό.

4.2.2. Managers του προτύπου

Στον κώδικα του έργου εφαρμόστηκε το σχεδιαστικό πρότυπο λογισμικού (**design pattern**) που προτείνει στο βιβλίο του ο **Joseph Hocking (2015, σ.178)**. Η βασική ιδέα συνοψίζεται στο ότι το λογισμικό έργο περιέχει κάποιες καλά ορισμένες μονάδες λογισμικού (modules) οι οποίες διαχειρίζονται τον δικό τους τομέα αρμοδιοτήτων. Αυτές οι μονάδες λογισμικού, που στην C# είναι απλές κλάσεις, ονομάζονται μάνατζερ του παιχνιδιού (game manager).

Στο έργο υπάρχει μία διασύνδεση κλάσης (interface) που ονομάστηκε **GameManager** την οποία εφαρμόζουν οι εξής κλάσεις μάνατζερ:

- **PlayerManager**
- **Scene_Manager**
- **InventoryManager**
- **UIManager**
- **DialogueManager**
- **AudioManager**

Ένας υψηλότερου επιπέδου κεντρικός μάνατζερ είναι διαχειριστής των υπολοίπων και βασική του δουλειά είναι να τους εκκινεί. Αυτός ονομάζεται **Managers** και μαζί με τις κλάσεις μάνατζερ έχουν προστεθεί σαν component σε ένα αντικείμενο στην αρχική σκηνή του παιχνιδιού, το οποίο είναι ενεργό καθόλη την διάρκεια του παιχνιδιού. Μέρος του κώδικα στη γλώσσα C# της κλάσης **Managers** παρουσιάζεται με ενσωματωμένα σχόλια παρακάτω (Κώδικας 4.1):

```
public class Managers : MonoBehaviour
{
    //ορισμός στιγμιοτύπων όλων των κλάσεων μάνατζερ ως στατικές
    public static PlayerManager Player { get; private set; }
    public static InventoryManager Inventory { get; private set; }
    public static Scene_Manager Scene { get; private set; }
    public static UIManager UI { get; private set; }
    public static DialogueManager Dialogue { get; private set; }
    public static AudioManager Audio_M { get; private set; }
    //με τον ορισμό των μεθόδων get; private set; μέσα στα άγκιστρα
    εφαρμόζεται η Αρχή της Ενσωμάτωσης όπου ενώ το αντικείμενο είναι προσβάσιμο
    από άλλες κλάσεις η τροποποίησή του γίνεται μόνο εσωτερικά αυτής της κλάσης.
    ...
    void Awake() //εκτελείται μόνο στην εκκίνηση
    {
        DontDestroyOnLoad(gameObject); //το αντικείμενο που περιέχει την
        κλάση δεν θα καταστρέφεται όταν φορτώνεται άλλη σκηνή, δηλαδή θα παραμένει
        ενεργό καθόλη την διάρκεια του παιχνιδιού

        // φόρτωση των αντικειμένων των κλάσεων μάνατζερ
        Player = GetComponent<PlayerManager>();
    }
}
```

```

Inventory = GetComponent<InventoryManager>();
Scene = GetComponent<Scene_Manager>();
UI = GetComponent<UIManager>();
Dialogue = GetComponent<DialogueManager>();
Audio_M = GetComponent<AudioManager>();
... // υπόλοιπος κώδικας της κλάσης
}

```

Κώδικας 4.1: Το script *Managers.cs* - ο κεντρικό μάνατζερ του έργου

Οι ανωτέρω **μάνατζερ** κλάσεις, κατά κάποιο τρόπο αποτελούν τον κορμό του κώδικα και οι βασικότερες λειτουργίες του προτύπου εμπεριέχονται σε αυτές. Οι υπόλοιπες κλάσεις του έργου, έχουν άμεση πρόσβαση σε αυτές τις λειτουργίες μέσω του κεντρικού μάνατζερ, καθώς σε αυτόν έχουν οριστεί ως στατικές όλες οι κλάσεις μάνατζερ. Και αυτό είναι που κάνει το συγκεκριμένο πρότυπο σχεδίασης ιδιαίτερα χρήσιμο. Οποιαδήποτε κλάση στο έργο, μπορεί να καλέσει μια μέθοδο ενός από τους μαντζερ, απλά με το πρόθεμα του. Για παράδειγμα, αν σε οποιαδήποτε κλάση εκτελεστεί η εντολή:

Managers.Dialogue.StartDialogue(x); // όπου x το id του διαλόγου

θα εμφανιστεί ο διάλογος με αριθμό Id x στην οθόνη, ενώ αν εκτελεστεί η εντολή:

Managers.UI.setCursorToUse();

θα αλλάξει το εικονίδιο του δείκτη του ποντικιού σε γρανάζι.

Παράλληλα, σε όλο το έργο, αλλά ιδιαίτερα στον κώδικα των μαντζερ του παιχνιδιού εφαρμόστηκε η **Αρχή της Ενσωμάτωσης**, σύμφωνα με την οποία “η εσωτερική κατάσταση ενός αντικειμένου πρέπει να είναι τροποποιήσιμη μόνο μέσω της δημόσιας διασύνδεσης του”. Έτσι η πρόσβαση και τροποποίηση των μεταβλητών της κάθε κλάσης του έργου γίνεται μόνο μέσω της πρόσβασης που παρέχουν δημόσια διαθέσιμες μέθοδοι αυτής της κλάσης. Η αρχή εγγυάται την εγκυρότητα και συνέπεια των κλάσεων και των ιδιοτήτων τους, ομαδοποιώντας τα δεδομένα και επιπλέον διευκολύνει την τροποποίηση και επέκταση του κώδικα (Χατζηγεωργίου, 2005 κεφ. 5).

Εκτός από τις κλάσεις μαντζερ, την κλάση για τον χειρισμό και έλεγχο του χαρακτήρα, ***PlayerController.cs*** και της κλάσης των διαδραστικών αντικειμένων ***InteractableObject.cs*** οι οποίες συγκεντρώνουν μεγάλη λειτουργικότητα και αποτελούνται από πολλές μεθόδους και μεταβλητές, στις υπόλοιπες κλάσεις του παιχνιδιού ακολουθήθηκε η **αρχή της μοναδιαίας αρμοδιότητας**, όπου κάθε κλάση του έργου αναλαμβάνει να εκτελέσει μία αρμοδιότητα και έχει υψηλή συνεκτικότητα. Με αυτό τον τρόπο η κλάση “έχει μόνο ένα λόγο να αλλάξει.” (DeMarco, 1979, Martin, 2003). Η εφαρμογή της αρχής διευκόλυνε την ανάπτυξη του έργου, καθώς όταν χρειάστηκαν αλλαγές στο παιχνίδι, το σημείο του κώδικα που έπρεπε να αλλάξει εντοπιζόταν εύκολα και η τροποποίησή του δεν επηρέαζε άλλες λειτουργίες

του παιχνιδιού, ούτε προκαλούσε σφάλματα. Δηλαδή, μειώθηκε η δυσκαμψία και ευθραυστότητα του έργου.

4.2.3. Υποσύστημα κίνησης του κεντρικού χαρακτήρα στον χώρο

Για την κίνηση του κεντρικού χαρακτήρα που ελέγχει ο παίκτης, την εύρεση της βέλτιστης διαδρομής και την αποφυγή εμποδίων, χρησιμοποιήθηκε το πολύ δυνατό εργαλείο **nav-mesh** που κάνει χρήση τεχνητής νοημοσύνης και έχει ενσωματωθεί στις νεότερες εκδόσεις της Unity. Μετά την ολοκλήρωση της κατασκευής της κάθε σκηνής που χρησιμοποιήθηκε το nav-mesh, δηλαδή την προσθήκη όλων των αντικειμένων που την αποτελούν, ορίστηκε αυτόματα η επιφάνεια στην οποία θα μπορούν να κινούνται οι χαρακτήρες με την επιλογή **Bake** του component **NavMeshSurface**. Με τη χρήση στρωμάτων (layers), όπου το κάθε αντικείμενο ανήκει σε μία στρώση και την επιλογή μέσα στις ιδιότητες του NavMeshSurface για το ποια στρώματα θα συμπεριληφθούν, έγινε η προσθήκη και η αφαίρεση εμποδίων σχετικά με τον περιορισμό της επιτρεπτής επιφάνειας κίνησης του χαρακτήρα.

Στον κεντρικό χαρακτήρα προστέθηκε το component **NavMeshAgent**. Οι ρυθμίσεις του NavMeshAgent για το ύψος του, την ακτίνα του νοητού κυλίνδρου που τον περιβάλλει και το μέγιστο ύψος που μπορεί να ανέβει με ένα βήμα. Αυτές οι ρυθμίσεις παίζουν ρόλο για τον αυτόματο υπολογισμό της επιτρεπτής επιφάνειας. Η Εικόνα 4.6 δείχνει το αποτέλεσμα της εφαρμογής του NavMeshSurface στην εσωτερική σκηνή του μουσείου, όπου με μπλε χρώμα είναι οι επιτρεπτές επιφάνειες.



Εικόνα 4.6: Επιφάνειες που μπορεί να κινείται ο χαρακτήρας στην εσωτερική σκηνή

Στη συνέχεια, προστέθηκε στον χαρακτήρα το script **PlayerController.cs** που περιέχει τους μηχανισμούς ελέγχου που κατασκευάστηκαν στο πρότυπο. Η κλάση αυτή δέχεται σαν είσοδο το σημείο της σκηνής όπου έχει γίνει αριστερό κλικ του ποντικιού και αυτό το σημείο μπαίνει σαν παράμετρος στην εντολή **agent.SetDestination()**. Αν αυτό το σημείο είναι εντός της επιτρεπτής επιφάνειας Nav-Mesh, αυτομάτως ο χαρακτήρας ακολουθεί τη συντομότερη διαδρομή, αν δεν είναι σε επιτρεπτή επιφάνεια, καταλήγει στο κοντινότερο επιτρεπτό σημείο.

Το 3D animation του χαρακτήρα προσαρμόζεται και συγχρονίζεται κάθε φορά στη διεύθυνση της κίνησης και στην ταχύτητα του αντικειμένου του χαρακτήρα που κινείται στον χώρο. Αυτό επιτεύχθηκε μέσω έτοιμων εργαλείων που προσφέρει η Unity. Συγκεκριμένα, στα **standard assets** της Unity υπάρχει η κλάση **ThirdPersonCharacter.cs**, η οποία προστέθηκε ως component στο αντικείμενο του χαρακτήρα, μαζί με την **playerController.cs** που αναφέρθηκε προηγουμένως και το component **animator** που χειρίζεται τα animation του αντικειμένου. Στον animator προστίθεται ο ελεγκτής **HumanoidPlayer.controller** που επίσης προσφέρεται από τη unity μέσω των standard assets.

Στον controller της κίνησης προστίθενται τα animation για το περπάτημα, το τρέξιμο, τις στροφές αριστερά και της αδράνειας (idle) όταν δηλαδή ο χαρακτήρας είναι στο ίδιο σημείο και απλά αναπνέει. Αυτά τα animation όπως και η μορφή του ίδιου του χαρακτήρα είναι δωρεάν προσβάσιμα στην ιστοσελίδα **Mixamo** Adobe (Mixamo, 2013). Με τη χρήση όλων αυτών των αυτοματισμών της Unity, το αποτέλεσμα είναι να αλλάζει αυτόματα το animation σε αυτό της αδράνειας, όταν το αντικείμενο του χαρακτήρα είναι ακίνητο, στο animation του περπατήματος όταν κινείται αργά, στο animation του τρεξίματος όταν κινείται γρήγορα κλπ.

4.2.4. Υποσύστημα αυτόματης επιλογής ενέργειας

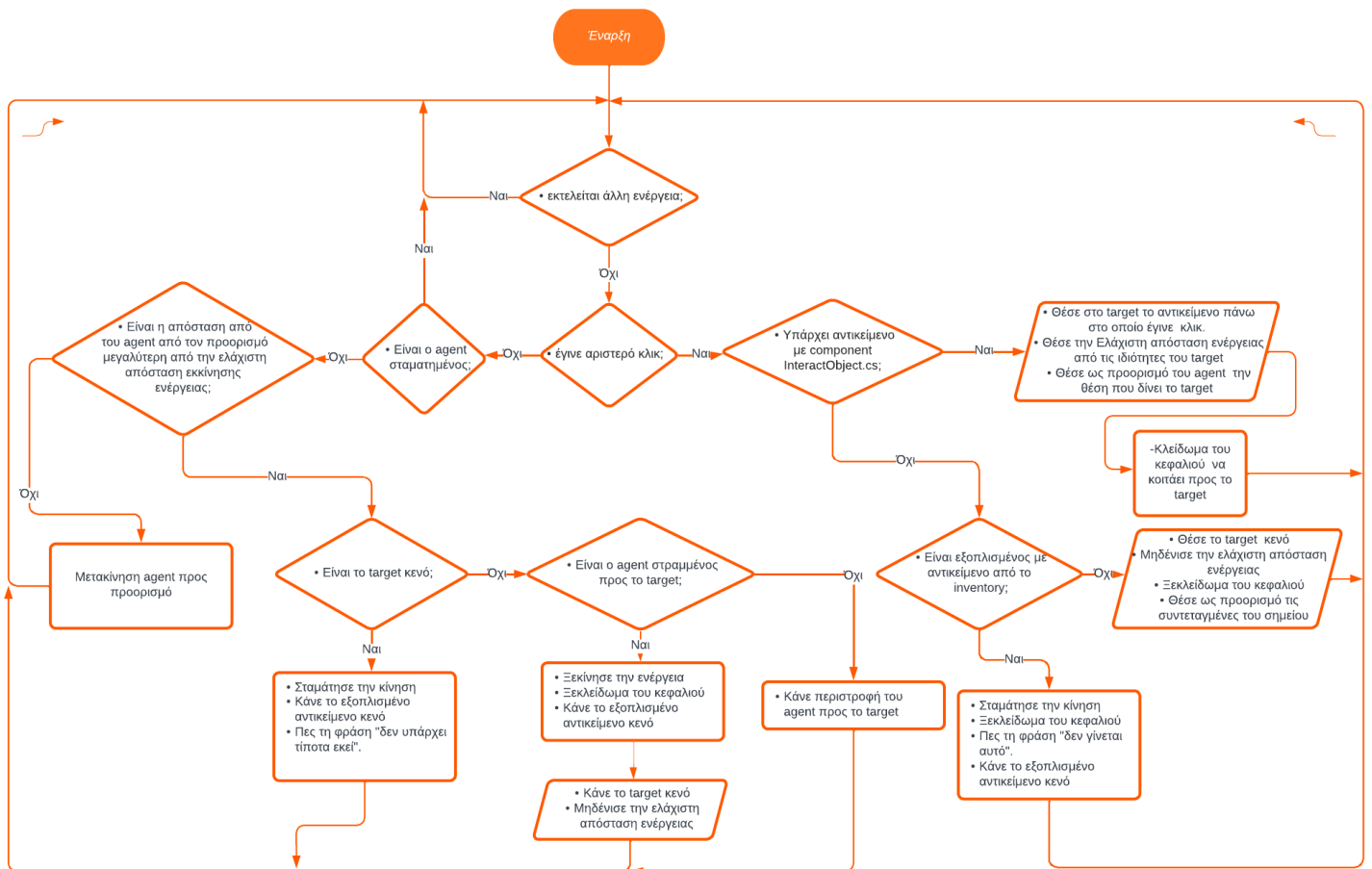
Ένα άλλο μεγάλο μέρος της φάσης της σχεδίασης αφορά τη σχεδίαση των υποσυστημάτων για την αυτόματη επιλογή ενέργειας που θα κάνει ο κεντρικός χαρακτήρας, κατά τη διάρκεια του κυρίως παιχνιδιού, μετά το αριστερό κλικ σε κάποιο σημείο ή πάνω σε κάποιο αντικείμενο. Σε ορισμένα σύγχρονα adventure game εφαρμόζεται η αυτόματη επιλογή ενέργειας, ενώ σε άλλα εμφανίζεται μικρό μενού στο σημείο του κλικ, με ενέργειες ανάμεσα στις οποίες μπορεί να επιλέξει ο χρήστης με νέο κλικ.

Επισημαίνεται ότι κάθε gameObject που υπάρχει μέσα σε σκηνή του κυρίως παιχνιδιού και με το οποίο ο παίκτης μπορεί να αλληλεπιδρά, έχει component το script **InteractObject.cs**. Οι κύριες ιδιότητες που δίνει στο GameObject αυτή η κλάση είναι οι εξής:

- όνομα
- περιγραφή του
- την ελάχιστη απόσταση εκκίνησης ενέργειας
- Την αλλαγή του δείκτη όταν τοποθετηθεί πάνω του
- καθορισμό της ενέργειας που θα γίνει μετά το κλικ πάνω σε αυτό.

Αριστερό κλικ μπορεί να γίνει πάνω σε αντικείμενα που αλληλεπιδρούν ή σε σημεία του περιβάλλοντος χωρίς αντικείμενα. Στην πρώτη περίπτωση ο χαρακτήρας κατευθύνεται προς το αντικείμενο και όταν η απόσταση είναι μικρότερη από ένα όριο κάνει κάποια ενέργεια, ενώ στην δεύτερη περίπτωση απλά κατευθύνεται προς το πλησιέστερο επιτρεπτό σημείο. Η ενέργεια και η ελάχιστη απόσταση εξαρτάται από το αντικείμενο.

Η σχεδίαση αυτού του υποσυστήματος περιγράφεται στο απλοποιημένο διάγραμμα ροής της Εικόνας 4.9.



Εικόνα 4.7: Διάγραμμα ροής για το υποσύστημα αυτόματης επιλογής ενέργειας

Η υλοποίησή του έγινε στην κλάση **playerController.cs**, που έχει προστεθεί ως component στο GameObject του χαρακτήρα.

Στην κλάση αυτή όπως και σε όλες τις κλάσεις που δημιουργούνται στην Unity υπάρχει η μέθοδος **Update()**. Η μέθοδος αυτή επειδή επανεκτελείται σε κάθε καρτέ του παιχνιδιού, μπορεί να θεωρηθεί ατέρμων βρόχος.

Στην αρχή της μεθόδου ελέγχεται αν ο χαρακτήρας εκτελεί κάποια ενέργεια από προηγούμενο αριστερό κλικ σε ένα αντικείμενο. Όταν δεν εκτελείται άλλη ενέργεια γίνεται ο έλεγχος αν πατήθηκε το αριστερό κουμπί του ποντικιού. Αν το κλικ έγινε πάνω σε ένα `interactObject` τότε η μεταβλητή αντικειμένου **target** θα γίνει αναφορά αυτού του αντικειμένου. Η **target** δηλαδή είναι ένα αντικείμενο της κλάσης **interactObject.cs** που μπορεί είτε να πάρει την αναφορά από ένα αντικείμενο της σκηνής ή να είναι κενό (null). Τότε εκτελείται η εντολή:

```
agent.SetDestination(target.getWaypointIfExistsOtherwiseObjectsPos());
```

Το παραπάνω όρισμα είναι η μια δημόσια μέθοδος που υπάρχει σε όλα τα αντικείμενα με την κλάση **interactObject.cs** και ο κώδικας της φαίνεται αναλυτικά παρακάτω (Κώδικας 4.2).

```
public Vector3 getWaypointIfExistsOtherwiseObjectsPos() {  
    foreach (Transform childTransform in transform)  
    {  
        if (childTransform.CompareTag("wayPoint"))  
        {  
            return childTransform.position;  
        }  
    }  
    return transform.position;  
}
```

Κώδικας 4.2: Δημόσια μέθοδος `getWaypointIfExistsOtherwiseObjectsPos` σε κάθε διαδραστικό αντικείμενο

Αν σε ένα διαδραστικό αντικείμενο έχει προστεθεί σαν αντικείμενο παιδί στην ιεραρχία, ένα `GameObject` που έχει την ετικέτα (tag) **"wayPoint"** τότε η μέθοδος επιστρέφει το διάνυσμα θέσης του αντικειμένου παιδιού. Αν δεν υπάρχει παιδί `wayPoint` τότε επιστρέφεται το διάνυσμα θέσης του ίδιου του διαδραστικού αντικειμένου. Με αυτό τον τρόπο επιτυγχάνεται σε ορισμένα αντικείμενα ο χαρακτήρας να τα προσεγγίσει σε σημείο διαφορετικό από το σημείο που βρίσκονται. Για παράδειγμα, όταν γίνει αριστερό κλικ στην μαρμάρινη πινακίδα του αγάλματος, ο χαρακτήρας δεν θα πάει πάνω στην πινακίδα αλλά σε σημείο δύο μέτρα μπροστά της.

Από τις ιδιότητες της `interactObject` λαμβάνεται και η ελάχιστη απόσταση της ενέργειας. Με το κλικ αυτομάτως ο χαρακτήρας στρέφει το κεφάλι του προς το αντικείμενο αυτό. Με τον ίδιο τρόπο όπως προηγουμένως αν υπάρχει αντικείμενο παιδί με ετικέτα **"AimToLook"** κλειδώνει το κεφάλι προς αυτό, αλλιώς κλειδώνει στο κέντρο του αντικειμένου.

Σε περίπτωση που γίνει αριστερό κλικ σε σημείο εκτός διαδραστικού αντικειμένου, γίνεται προορισμός το σημείο αυτό, η **target** γίνεται κενή και ξεκλειδώνει το κεφάλι, ενώ αν έχει επιλεγθεί πρώτα αντικείμενο από το Inventory και γίνει κλικ τότε θα αποεπιλεγθεί το αντικείμενο, θα σταματήσει η κίνηση του χαρακτήρα και θα πει την φράση "**δεν γίνεται αυτό**".

Εάν δεν έχει γίνει αριστερό κλικ και ο χαρακτήρας δεν είναι σταματημένος, τότε μετακινείται προς τον προορισμό του. Αν δεν γίνει νέο κλικ, θα συνεχίσει να κινείται μέχρι η απόσταση του από τον προορισμό γίνει μικρότερη από την ελάχιστη απόσταση ενέργειας.

Όταν φτάσει στον προορισμό, αν η target είναι κενή θα σταματήσει την κίνηση και θα πει την φράση "**δεν υπάρχει τίποτα εδώ**".

Αν η target έχει αναφορά του interactObject, πάνω στο οποίο έγινε κλικ, θα γίνει έλεγχος αν ο χαρακτήρας είναι στραμμένος προς το αντικείμενο. Αυτό γίνεται γιατί αν υπάρχει αντικείμενο "**wayPoint**", μπορεί όταν φτάσει στον προορισμό ο χαρακτήρας να έχει άλλη κατεύθυνση από αυτή προς το αντικείμενο. Τότε θα κάνει περιστροφή προς το αντικείμενο. Τελικά θα κάνει την ενέργεια που είναι ορισμένη στην ιδιότητα **whichaction** του interactObject.

Σε αυτό το σημείο, εφαρμόζεται μια τεχνική με χρήση αφηρημένων μεθόδων και παράγωγων κλάσεων που προσδίδει επεκτασιμότητα στο έργο. Έστω για παράδειγμα ότι έχει γίνει αριστερό κλικ σε ένα αντικείμενο στη σκηνή που μπορεί να χρησιμοποιηθεί, όπως μία πόρτα. Το αντικείμενο αυτό θα έχει σαν component την κλάση **OpenDoor.cs**, που ορίζεται ως παράγωγη κλάση της γονικής **UseObject.cs** ως ακολούθως:

```
public class OpenDoor : UseObject
```

Στην γονική κλάση **UseObject.cs** περιέχονται όλες οι ιδιότητες και μέθοδοι που έχουν τα αντικείμενα που μπορούν να χρησιμοποιηθούν σε μια σκηνή, όπως είναι και η πόρτα.

Περιέχεται, επίσης, μία αφηρημένη μέθοδος, η:

```
public virtual void specialAct(){}
```

η οποία είναι δημόσια και εκτελείται αφού εκτελεστούν οι άλλες μέθοδοι που έχει η γονική κλάση **UseObject.cs**. Δεν έχει σώμα εντολών, διότι η υλοποίησή της βρίσκεται μόνο στις παράγωγες κλάσεις, όπως η **OpenDoor.cs** και ορίζεται ως:

```
public override void specialAct()
```

Με αυτόν τον τρόπο μπορούμε να προσθέτουμε διαφορετικό κώδικα σε κάθε ξεχωριστό αντικείμενο που μπορεί να χρησιμοποιηθεί στη σκηνή, διατηρώντας όμως, τις βασικές ιδιότητες κοινές σε όλα αυτά τα αντικείμενα.

Το ίδιο συμβαίνει και με τα αντικείμενα της σκηνής που μπορούν να συλλεχθούν και να μπουν στο inventory. Υπάρχει αντίστοιχα η γονική κλάση

PickupObject.cs που περιέχει όλες τις κοινές ιδιότητες και μεθόδους αυτών των αντικειμένων και επιπλέον περιέχει και την αφηρημένη μέθοδο:

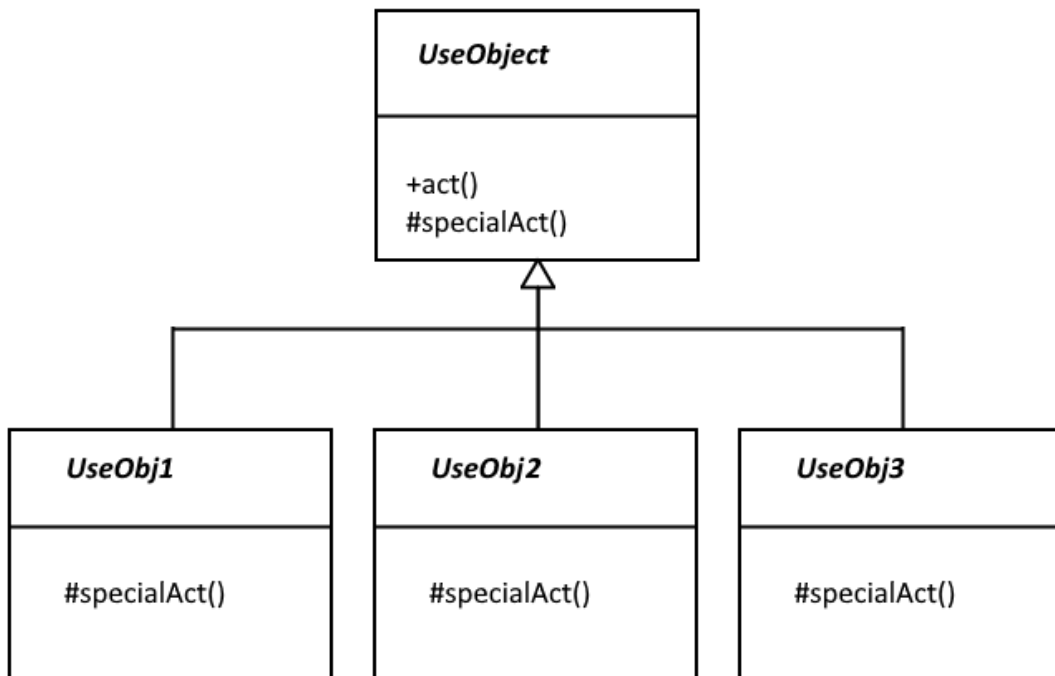
```
public virtual void specialAct(){}
```

και οι αντίστοιχες υποκλάσεις με την μέθοδο:

```
public override void Specialpickup()
```

όπου εφαρμόζεται ο συγκεκριμένος κώδικας για αυτό το αντικείμενο όπου συλλέχθηκε.

Η γενική ιδέα του υποσυστήματος με τις αφηρημένες μεθόδους περιγράφεται στο **design pattern "Μέθοδος Υπόδειγμα"** και εκμεταλλεύεται το μηχανισμό της κληρονομικότητας και του πολυμορφισμού της γλώσσας C#. Διαγραμματικά παρουσιάζεται στην Εικόνα 4.8.



Εικόνα 4.8: Διάγραμμα UML για την "Μέθοδο Υπόδειγμα"

Τις περισσότερες φορές στα αντικείμενα της σκηνής προστίθενται κατευθείαν οι παράγωγες κλάσεις που κληρονομούν τις ιδιότητες και τις μεθόδους από την γονική και όχι η ίδια η γονική. Μπορούν, όμως, και να χρησιμοποιηθούν απευθείας οι γονικές, όταν δεν γίνεται κάποια ιδιαίτερη ενέργεια, η οποία να ξεχωρίζει το αντικείμενο από τα άλλα.

4.2.5. Καταστάσεις του παιχνιδιού

Το πρότυπο σχεδιάστηκε ώστε κάθε στιγμή το παιχνίδι να βρίσκεται σε μία μόνο κατάσταση (game state). Οι καταστάσεις ορίζονται στην μάνατζερ κλάση **Scene_Manager.cs** όπως δείχνει ο Κώδικας 4.3.

```
[SerializeField] private EnumGameState gameState;  
// με την [SerializeField] η private μεταβλητές εμφανίζονται στο παράθυρο  
inspector όταν επιλέγεται το game Object  
public enum EnumGameState  
{  
    intro1State,  
    intro2State,  
    mainMenuState,  
    loadingScene,  
    tutorialQuest,  
    inGamePlayState,  
    inGameMenuState,  
    inGameSettingsState,  
    inGameAreYouSure,  
    inGameCutScene,  
    inGameFocus  
}
```

Κώδικας 4.3: Ορισμός καταστάσεων στην Scene_Manager.cs

Η κατάσταση μπορεί να ανιχνευτεί μέσω της δημόσιας διασύνδεσης της **Scene_Manager.cs** με τον Κώδικα 4.4.

```
public EnumGameState getEnumGameState()  
{  
    return gameState;  
}
```

Κώδικας 4.4: Δημόσια διασύνδεση της gameState μέσα στην Scene_Manager.cs

Αυτή η ανίχνευση γίνεται από οποιαδήποτε κλάση του έργου, καθώς οι μέθοδοι μιας μάνατζερ κλάσης είναι καθολικά προσβάσιμες.

Με αυτό το σύστημα επιτυγχάνεται κάποιες λειτουργίες να εκκινούν μόνο όταν το παιχνίδι βρίσκεται σε συγκεκριμένη κατάσταση ή να εκκινούν διαφορετικά αν βρίσκεται σε άλλη. Για παράδειγμα, το πλήκτρο esc αν πατηθεί κατά τη διάρκεια του κανονικού παιχνιδιού εμφανίζει το μενού επιλογών, αν είναι ανοιχτό το μενού επιλογών το κλείνει, ενώ αν πατηθεί κατά την διάρκεια ενός κινηματικού το τερματίζει.

4.2.6. Υποσύστημα ελέγχου καμερών

Η Unity παρέχει ένα πολύ εύκολο στη χρήση σύστημα καμερών, όπου αφήνει στον χρήστη να ασχοληθεί με τις βασικές ιδιότητες της, όπως την θέση, την γωνία και το εύρος του φακού, ενώ οι πολύπλοκοι υπολογισμοί, όπως ο τρόπος εμφάνισης των 3D αντικειμένων (rendering) γίνονται αυτόματα.

Σε κάθε σκηνή (scene) μπορεί να υπάρχουν μία ή παραπάνω κάμερες αλλά πάντα μόνο μία είναι ενεργή. Στη σκηνή του παιχνιδιού στο εξωτερικό του μουσείου, υπάρχουν τέσσερις κάμερες, μία για την κινηματική σκηνή (cutscene), δύο που ακολουθούν τον χαρακτήρα στο προαύλιο του μουσείου και στο χώρο της εισόδου αντίστοιχα και μία κάμερα που χρησιμοποιείται για τις εστιάσεις σε διάφορα αντικείμενα. Στην εσωτερική σκηνή του μουσείου υπάρχει μία κάμερα που ακολουθεί τον χαρακτήρα και μία για τις εστιάσεις.

Η κίνηση της κάμερας για την κινηματική σκηνή είναι προσχεδιασμένη μέσω του animation που δημιουργήθηκε με το εργαλείο **timeline**. Για τον έλεγχο των καμερών που ακολουθούν τον κεντρικό χαρακτήρα έχει προστεθεί στην κάθε μία το script component **CameraScript.cs**.

Σε αυτή την κλάση δηλώνεται και αρχικοποιείται μία μεταβλητή διανύσματος με όνομα **offsetCameraFromTarget** που παριστάνει την επιθυμητή απόσταση της κάμερας από τον χαρακτήρα. Δηλώνεται και μία μεταβλητή **target**, που ορίζεται ως αντικείμενο που αποτελεί τον κεντρικό χαρακτήρα, οπότε η `target.position` είναι το διάνυσμα θέσης του χαρακτήρα. Κατά την κανονική διάρκεια του παιχνιδιού εκτελείται συνεχώς μία επανάληψη (loop), στην οποία το διάνυσμα **offsetCameraFromTarget** προστίθεται στο διάνυσμα θέσης του κεντρικού χαρακτήρα και δίνει την θέση της κάμερας. Στη συνέχεια, μπαίνουν κάποια όρια στους άξονες κίνησης της κάμερας, πέρα από τα οποία δεν μπορεί να κινηθεί. Αυτό επιτυγχάνεται με μια μέθοδο που παίρνει σαν παράμετρο το διάνυσμα θέσης και επιστρέφει πάλι ένα διάνυσμα θέσης αλλά εντός ορίων. Έπειτα, η κάμερα στρέφεται προς την κατεύθυνση του χαρακτήρα. Η υλοποίηση γίνεται με τον Κώδικα 4.5.

```
Vector3 cameraPos = target.position + offsetCameraFromTarget;  
targetCamTransform.position = InsertClamps(cameraPos);  
targetCamTransform.LookAt(target);
```

Κώδικας 4.5: Στην κλάση CameraScript.cs η κάμερα τίθεται πάνω από τον χαρακτήρα

Τα διανύσματα θέσης **targetCamTransform** δεν μεταβάλλουν αμέσως την θέση της κάμερας. Για να έχει η κάμερα μια ομαλή κίνηση όταν μετακινείται ο

χαρακτήρας και όχι μια απότομη κοπή κίνηση, έχουν προστεθεί οι δύο γραμμές στην CameraScript.cs (Κώδικας 4.6):

```
transform.position = Vector3.Lerp (transform.position,  
targetCamTransform.position, timeCamToFinalPos *  
Time.deltaTime);  
transform.rotation = Quaternion.Lerp(transform.rotation,  
targetCamTransform.rotation, timeCamToFinalPos *  
Time.deltaTime);
```

Κώδικας 4.6: Στην κλάση CameraScript.cs η μετακίνηση της κάμερας γίνεται ομαλά

Η **Vector3.Lerp** μετακινεί ομαλά ένα αντικείμενο από μία αρχική σε μία τελική θέση μέσα σε μια συγκεκριμένη χρονική περίοδο που καθορίζεται από τον τρίτο παράγοντα στην παρένθεση. Εδώ η χρονική περίοδος καθορίστηκε από το γινόμενο **timeCamToFinalPos * Time.deltaTime**, όπου η **timeCamToFinalPos** είναι ένας αριθμός που ρυθμίζει την ταχύτητα, ενώ η **Time.deltaTime** είναι μια μεταβλητή που παριστάνει τον χρόνο που πέρασε από την εμφάνιση του προτελευταίου καρέ μέχρι του τελευταίου. Η τελευταία χρησιμοποιείται σε διάφορα σημεία στο έργο γιατί εξασφαλίζει ότι οι κινήσεις του θα είναι ομαλές και ανεξάρτητες του fps, άρα και της υπολογιστικής ισχύος του συστήματος.

Οι δύο αυτές εντολές *Lerp* χρησιμοποιούνται παρόμοια και όταν η κάμερα της εστίασης (που είναι διαφορετική από αυτή που ακολουθεί τον παίκτη) εστιάζει σε διάφορα αντικείμενα. Όταν πατήσει ο παίκτης σε ένα αντικείμενο στην σκηνή που μπορεί να εστιαστεί, γίνεται εναλλαγή σε αυτήν την κάμερα εστίασης. Για αρχική θέση στην εντολή **Lerp** μπαίνει η τελευταία θέση της κάμερας παρακολούθησης του χαρακτήρα, ενώ σαν τελική θέση μπαίνει ένα διάνυσμα θέσης που έχει οριστεί στις ιδιότητες του αντικειμένου.

4.2.7. Υποσύστημα διάδρασης με αντικείμενα

Όλα τα διαδραστικά αντικείμενα που υπάρχουν σε μία σκηνή εντός του παιχνιδιού, έχουν κάποιες κοινές ιδιότητες και λειτουργίες που καθορίζονται από το script component **InteractObject.cs**.

Σε αυτήν την κλάση καθορίζονται κάποιες βασικές ιδιότητες του αντικειμένου, όπως είναι το όνομα, η περιγραφή του, ποια ενέργεια θα γίνει με το κλικ πάνω του, το σημείο προσέγγισης του χαρακτήρα με το αντικείμενο και η ελάχιστη απόσταση για να ξεκινήσει αυτή η ενέργεια.

Εντός της κλάσης ορίζονται οι μέθοδοι για την αλλαγή της εικόνας του δείκτη του ποντικιού, όταν αυτός εισέρχεται και κινείται πάνω σε ένα διαδραστικό αντικείμενο, και της δημιουργίας έντονου περιβλήματος, μέσω των μεθόδων **OnMouseEnter()**, **OnMouseOver()**, **OnMouseExit()** που παρέχει η Unity.

Το σχήμα που θα πάρει ο δείκτης όταν μετακινηθεί πάνω στο αντικείμενο, καθώς και η ενέργεια που θα γίνει όταν γίνει κλικ πάνω του, καθορίζονται από την μεταβλητή:

```
private EnumWhichActions whichAction;
```

που παίρνει τις ακόλουθες τιμές:

```
public enum EnumWhichActions  
{  
    look, use, pickUp, talk, notAllowed, exit  
}
```

Παράλληλα με την αλλαγή του δείκτη εμφανίζεται μία φράση με την ενέργεια που πρόκειται να γίνει στο κέντρο της οθόνης.

Όταν γίνει τελικά αριστερό κλικ πάνω στο αντικείμενο, ο χαρακτήρας θα προσεγγίσει το σημείο προσέγγισης του συγκεκριμένου αντικειμένου. Όταν φτάσει στην ελάχιστη απόσταση θα εκτελεστεί ανάλογα με την τιμή της **whichaction**, η μέθοδος **act()** από μία από τις παρακάτω κλάσεις:

ExamineObject.cs

PickupObject.cs

UseObject.cs

TalkChar.cs

NotAllowedObject.cs

ExitObject.cs

Η κλάση αυτή θα πρέπει απαραίτητα να έχει προστεθεί στο αντικείμενο σαν component. Η μεταβλητή **whichaction** μπορεί να αλλάζει και κατά τη διάρκεια του παιχνιδιού, δηλαδή ένα αντικείμενο την πρώτη φορά που θα γίνει κλικ πάνω του μπορεί να κάνει την ενέργεια **act()** της κλάσης **ExamineObject.cs**, έπειτα να αλλάξει η **whichaction** και τη δεύτερη φορά να εκτελεστεί η **act()** της **PickupObject.cs**.

4.2.8. Υποσύστημα διαλόγων

Στο πρότυπο έχει αναπτυχθεί ένα υποσύστημα διαλόγου, όπου η ομιλία του κεντρικού και των άλλων χαρακτήρων ακούγεται και ταυτόχρονα εμφανίζεται στην οθόνη. Κάθε διάλογος στο παιχνίδι αποτελείται από μία σειρά φράσεων που έχουν κάποιες ιδιότητες. Οι φράσεις αυτές εισάγονται στο πρότυπο μέσω της κλάσης **Dialogue.cs**, της οποίας ο κατασκευαστής και οι ιδιότητες παρουσιάζονται παρακάτω (Κώδικας 4.7).

```

public Dialogue(int _num, string _name, int _nextnum, int _nextChoice, string[]
                _sentences, Sound[] _soundDialogue, float[] _durationExtend, int[]
                _numAnimOfDialogue, int[] _SyncLipIntsA, int[] _SyncLipIntsO)
{
    num = _num;           //αύξων αριθμός προτάσεων
    name = _name;        //όνομα χαρακτήρα που μιλάει
    nextnum = _nextnum;  //επόμενη στη σειρά φράση
    nextChoice = _nextChoice; //επόμενη στη σειρά επιλογή προτάσεων
    sentences = _sentences; //πίνακας με τα κείμενα των προτάσεων
    soundDialogue = _soundDialogue; //πίνακας με τα φωνητικά των προτάσεων
    durationExtend = _durationExtend; //πίνακας με την επιπλέον διάρκεια των
    προτάσεων
    numAnimOfDialogue = _numAnimOfDialogue; //πίνακας με τις κινήσεις σώματος
    SyncLipIntsA = _SyncLipIntsA; //πίνακας με συγχρονισμό του στόματος για το a
    SyncLipIntsO = _SyncLipIntsO; //πίνακας με συγχρονισμό του στόματος για το o
}

```

Κώδικας 4.7: Οι ιδιότητες και ο κατασκευαστής της μάντζερ κλάσης Dialogue.cs

Σε κάθε σκηνή υπάρχει ένα GameObject που ονομάζεται **DialogueOfScene** και σε αυτό έχει προστεθεί το component **DialogueOfScene.cs**. Εντός της κλάσης δηλώνεται ένας πίνακας με τα στιγμιότυπα των κλάσεων **Dialogue.cs** και μία δημόσια μέθοδος για την διαβίβαση των προτάσεων (Κώδικας 4.8).

```

public Dialogue[] dialogue;
...
public Dialogue getDialogue(int numDialogue)
{
    return dialogue[numDialogue];
}

```

Κώδικας 4.8: Δήλωση πίνακα με κλάσεις Dialogue.cs και δημόσια διασύνδεση εντός της DialogueOfScene.cs

Πρέπει να σημειωθεί ότι κάθε στιγμιότυπο κλάσης αυτού του πίνακα αποτελείται από ένα πίνακα προτάσεων και κάθε φορά εμφανίζεται μόνο μία φράση στην οθόνη.

Η εισαγωγή των δεδομένων των διαλόγων γίνεται στο γραφικό περιβάλλον της Unity μέσω του παραθύρου inspector επιλέγοντας το αντικείμενο **DialogueOfScenes**. Αυτός είναι και ο λόγος που ο πίνακας δηλώνεται δημόσια, για να εμφανίζεται στον inspector.

Ο κώδικας της υλοποίησης του μηχανισμών εδράζεται στην κλάση **DialogueManager.cs**. Στην έναρξη της κλάσης μανάτζερ αρχικοποιούνται πέντε ουρές δεδομένων (Κώδικας 4.9).

```
void Start()
{
    sentences = new Queue<string>();
    sounds = new Queue<Sound>();
    sentenceDuration = new Queue<float>();
    numAnimOfDialogue = new Queue<int>();
    LipSyncQueueA = new Queue<int>();
    LipSyncQueueO = new Queue<int>();
}
```

Κώδικας 4.9: Αρχικοποίηση στην DialogueManager.cs

Κάθε φορά που ξεκινάει ο διάλογος καλείται η δημόσια μέθοδος **StartDialogue(int DialogueNumber)**. Το όρισμα της είναι απλά ο αύξων αριθμός του στιγμιότυπου της **Dialogue.cs** στον πίνακα, δηλαδή των προτάσεων που θα εμφανιστούν. Ολόκληρο το στιγμιότυπο της κλάσης **Dialogue.cs**, με τα δεδομένα των προτάσεων, θα ληφθεί από τον πίνακα κλάσεων, οι οποίες είχαν εισαχθεί στο αντικείμενο **DialogueOfScene**. Δεν λαμβάνεται ολόκληρος ο πίνακας, αλλά μόνο ένα στιγμιότυπο της κλάσης **Dialogue.cs** που έχει τη θέση του ορίσματος **DialogueNumber** στον πίνακα.

Στη συνέχεια, ενεργοποιείται το αντικείμενο **PlayerDialogueGameObject**, το οποίο είναι ένα κείμενο πάνω στην οθόνη. Η θέση του στη σκηνή ορίζεται πάνω από τον χαρακτήρα που μιλάει, και εξαρτάται από την τιμή της μεταβλητής **Dialogue.name**. Αν η τελευταία είναι **"Player"** το κείμενο τοποθετείται πάνω από τον κεντρικό χαρακτήρα, αν είναι όνομα άλλου χαρακτήρα τοποθετείται πάνω από αυτόν.

Ακολουθώς, προστίθενται τα δεδομένα της Dialogue στις ουρές (Queue) **sentences, sounds, sentenceDuration, numAnimOfDialogue, LipSyncQueueA, LipSyncQueueB** και εκτελείται η μέθοδος **DisplayNextSentence()**.

Σε αυτή τη μέθοδο λαμβάνονται τα δεδομένα από τις ουρές με την σειρά εισόδου τους και γίνονται τα παρακάτω:

- Το κείμενο από την ουρά **sentence** φορτώνεται στο κείμενο του αντικειμένου **PlayerDialogueGameObject**.
- Ακούγεται ο ήχος από την ουρά **sounds**.
- Το κείμενο παραμένει ανοιχτό για χρόνο δυο δευτερόλεπτα συν την τιμή από την ουρά **sentenceDuration**.

- Ξεκινάει το animation του χαρακτήρα που έχει τον αριθμό της τιμής από την ουρά **numAnimOfDialogue**.
- Ξεκινάει ο συγχρονισμός του στόματος σύμφωνα με τα δεδομένα από τις ουρές **LipSyncQueue**.

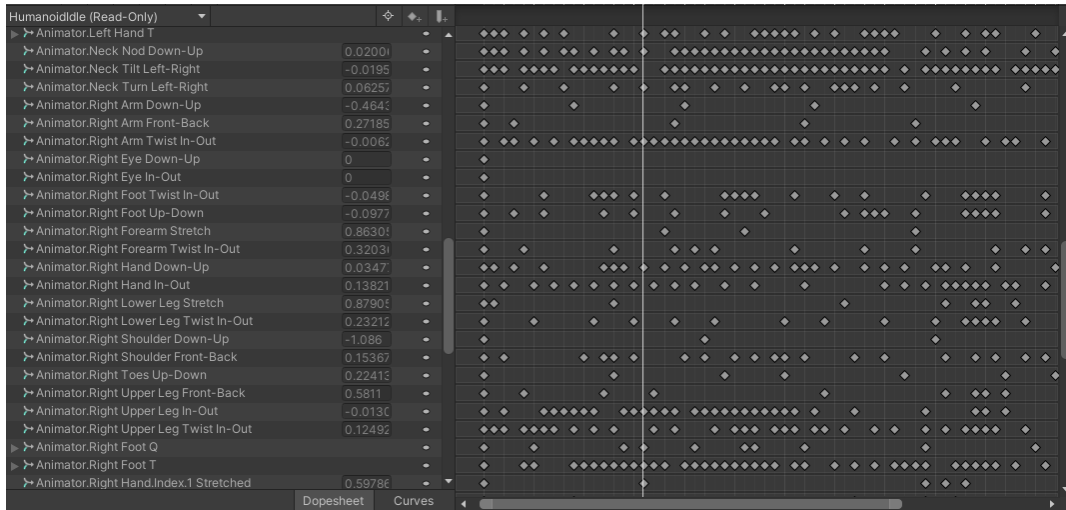
Με την πάροδο του χρονικού διαστήματος των δύο δευτερολέπτων συν την τιμή από την **sentenceDuration** η **DisplayNextSentence()** επανεκτελείται με τα επόμενα δεδομένα των σειρών.

Στην περίπτωση που οι ουρές είναι άδειες, δηλαδή δεν υπάρχει άλλη φράση για εμφάνιση, ελέγχεται η τιμή **dialogue.nextnum** και αν είναι μηδέν ο διάλογος σταματάει. Αν η τιμή της είναι διαφορετική εκτελείται πάλι από την αρχή η **StartDialogue(dialogue.nextnum)** με όρισμα τον αύξοντα αριθμό κάποιου άλλου στιγμιότυπου dialogue από τον πίνακα. Εάν η τιμή της **dialogue.nextnum** είναι μηδέν αλλά η τιμή της **dialogue.nextChoice** δεν είναι, τότε δεν εμφανίζεται διάλογος, αλλά πλαίσιο επιλογών όπου ο παίκτης επιλέγει ποια φράση θα πει. Επισημαίνεται ότι η ανάλυση αυτή αφορά μόνο τον κορμό της **DialogueManager.cs** και ότι σε αυτή υπάρχει επιπλέον κώδικας που υλοποιεί περισσότερες λειτουργίες. Για παράδειγμα, όταν εμφανίζεται μία φράση στην οθόνη υπάρχει η δυνατότητα στον παίκτη να παρακάμψει όλη τη διάρκειά της (δύο δευτερόλεπτα συν **sentenceDuration**) με αριστερό κλικ, ώστε να εμφανιστεί κατευθείαν η νέα φράση. Επίσης, κατά την διάρκεια των διαλόγων ο παίκτης δεν μπορεί να δώσει εντολή στον χαρακτήρα να μετακινηθεί σε άλλο σημείο.

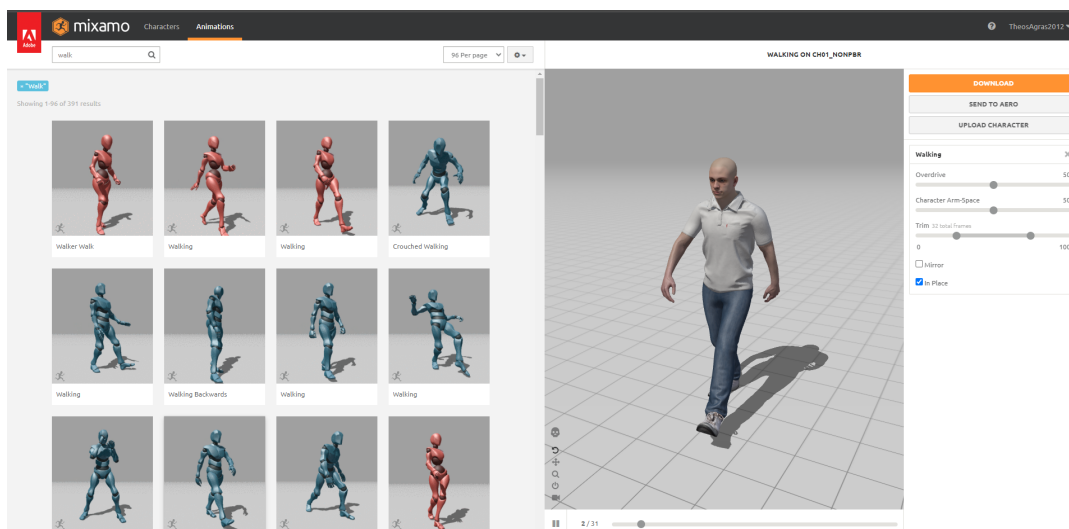
4.2.9. Υποσύστημα ελέγχου animation χαρακτήρων

Οι χαρακτήρες στο παιχνίδι δεν είναι ακίνητοι αλλά εκτελούν διαρκώς κάποιο animation. Το animation αυτό δεν αποτελείται από απλά καρτέ που εναλλάσσονται με κάποιο ρυθμό όπως τα κλασικά κινούμενα σχέδια (Sprite Sheet animation) αλλά αποτελείται από 3D animation clips, που είναι δεδομένα κίνησης της τρισδιάστατης μορφής του χαρακτήρα. Τα δεδομένα κίνησης περιέχουν ένα σύνολο από καρτέ κλειδιά (keyframes), τα οποία είναι οι στιγμιαίες θέσεις των μερών του χαρακτήρα (χέρια, πόδια, κεφάλι κλπ) όταν αυτά αλλάζουν θέση και κατεύθυνση (Εικόνα 4.9). Για παράδειγμα, για μια τρισδιάστατη κίνηση ανάτασης ενός χεριού, δεν χρειάζεται να σχεδιαστούν πολλά καρτέ, αλλά αρκούν δύο keyframes. Στο πρώτο, το χέρι πρέπει να βρίσκεται σε χαμηλή θέση στην αρχή της κίνησης και στο δεύτερο να βρίσκεται στην ανάταση μετά από κάποιο χρόνο. Συγκεκριμένα, για τα humanoid anim υπάρχουν keyframes για όλα τα μεγάλα κόκαλα του ανθρώπινου σκελετού, από το κεφάλι μέχρι και τη τελική φάλαγγα του κάθε δακτύλου. Τα animation clips έχουν εισαχθεί στο

Unity μαζί με το 3D μοντέλο του χαρακτήρα από το site **mixamo.com** (Εικόνα 4.10). Επειδή πρόκειται για δεδομένα κίνησης ανθρώπινων χαρακτήρων τα clips ονομάζονται humanoid anim clips. Σημαντικό είναι ότι οποιοδήποτε humanoid anim clip μπορεί να χρησιμοποιηθεί σε όλα τα 3D humanoid μοντέλα χαρακτήρων, όπως αυτά που προσφέρει δωρεάν το **mixamo.com**.



Εικόνα 4.9: Τα keyframes του animation της κίνησης του χαρακτήρα όταν είναι σταθερός σε ένα σημείο (idle anim)



Εικόνα 4.10: Αναζήτηση Humanoid anim στην ιστοσελίδα mixamo.com

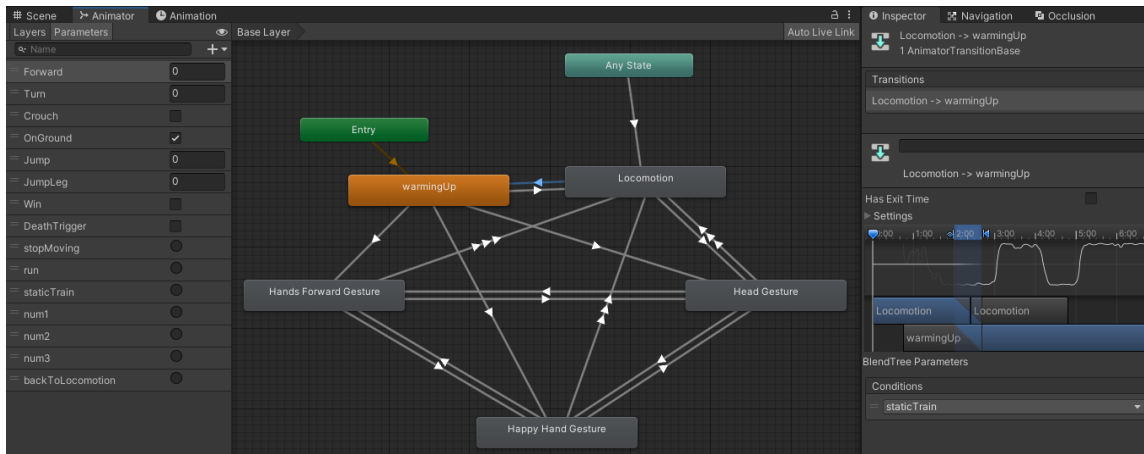
Τα animation clips είναι το θεμελιώδες στοιχείο του συστήματος animation του Unity. Ένα άλλο στοιχείο είναι ο ελεγκτής animator (**animator controller**) που προστίθεται στο component **animator**. Ο ελεγκτής αυτός προσφέρει μία οπτική διεπαφή για να ελέγχεται ποιο animation εκτελείται και πως γίνεται η μετάβαση σε ένα άλλο, ρυθμίζοντας κάποιες παραμέτρους.

Για να παρουσιαστεί ο τρόπος που εφαρμόστηκε ο ελεγκτής στο πρότυπο, θα παρουσιαστεί ο ελεγκτής **humanoid athletic** του χαρακτήρα του αθλούμενου άντρα που βρίσκεται στην εξωτερική σκηνή του παιχνιδιού.

Όπως φαίνεται στην Εικόνα 4.20, αριστερά έχουν προστεθεί οι παράμετροι πυροδότησης (trigger) **stopMoving**, **staticTrain**, **num1**, **num2**, **num3**, **backToLocomotion**. Δεξιά υπάρχει ένα διάγραμμα καταστάσεων, με βελάκια που δείχνουν την αλλαγή της κατάστασης του anim όταν πυροδοτείται κάποια παράμετρος. Η Locomotion κατάσταση περιλαμβάνει όλα τα anim clips που αφορούν τις βασικές κινήσεις του χαρακτήρα, όπως περπάτημα, τρέξιμο, στροφή αλλά και την αδράνεια. Οι υπόλοιπες καταστάσεις αφορούν διάφορες άλλες κινήσεις, όπως την προθέρμανση και τον χαιρετισμό. Για να γίνει η μετάβαση από την μία κατάσταση στην άλλη, πυροδοτείται η ανάλογη παράμετρος που βρίσκεται στον κώδικα της κλάσης **AthleticManController.cs** που έχει προστεθεί ως component στο αντικείμενο του χαρακτήρα. Για παράδειγμα, στον κώδικα υπάρχει η εντολή:

```
athleticManAnimator.SetTrigger("staticTrain");
```

όπου **athleticManAnimator** είναι η αναφορά στο component ελεγκτή του χαρακτήρα. Όταν εκτελεστεί αυτή η εντολή θα αλλάξει το anim στην προθέρμανση του παίκτη, σύμφωνα με τις παραμέτρους αυτής της μετάβασης, οι οποίες φαίνονται στην Εικόνα 4.11.



Εικόνα 4.11: Ο animator του ανθρώπου που αθλείται στην εξωτερική σκηνή

Μέσα στο πρότυπο υπάρχουν και κάποιες κινήσεις μερών του σώματος, που κινούνται διαφορετικά από το animation της κατάστασης του ελεγκτή, στην οποία βρίσκεται ο χαρακτήρας. Για παράδειγμα, το κεφάλι του κεντρικού χαρακτήρα όταν ο δείκτης του ποντικιού βρίσκεται πάνω από ένα αντικείμενο στρέφεται προς αυτό. Αυτή η κίνηση δεν ενσωματώνεται μέσα σε anim clip και δεν υλοποιείται με **keyframes**. Σε αυτή την περίπτωση, χρησιμοποιείται το εργαλείο **animation rig**, όπου η κίνηση μερών του σώματος ανεξαρτητοποιείται από το anim clip.

4.3. Συνολικό μέγεθος του έργου

Ο κώδικας του παιχνιδιού “Επίσκεψη στο μουσείο” που εμπεριέχει και τον κώδικα του προτύπου είναι δομημένος σε 91 κλάσεις της γλώσσας C# και συνολικά αποτελείται από 4960 γραμμές καθαρού κώδικα (LOC) όπως δείχνει ο Πίνακας 4.2. Βέβαια, σε αυτό το μέγεθος δεν προσμετράται ο κώδικα από τις βιβλιοθήκες της Unity που χρησιμοποιούνται, παρά μόνο ο κώδικας που γράφτηκε για την υλοποίηση του προτύπου. Ανάμεσα στις γραμμές κώδικα υπάρχουν και γραμμές με σχόλια (comments) που επιτρέπουν την ευκολότερη κατανόηση του κώδικα και κατά επέκταση την ευκολότερη συντήρησή του.

language	files	code	comment	blank	total
C#	91	4,960	192	974	6,126

Πίνακας 4.2: Μέγεθος του κώδικα του έργου

Το μέγεθος του φακέλου που εξάγεται από την Unity, με τα αρχεία του παιχνιδιού στην τελική εκτελέσιμη μορφή του είναι περίπου 1GB.

Το λογισμικό έργο θεωρείται μέτριου προς μεγάλου μεγέθους, όμως, αν το πρότυπο χρησιμοποιηθεί για την ανάπτυξη ενός ολοκληρωμένου adventure παιχνιδιού, σίγουρα το μέγεθος του κώδικα θα μεγαλώσει αρκετά και το λογισμικό έργο θα περάσει στα όρια ενός μεγάλου έργου.

4.4. Μηχανισμοί προτύπου που υλοποιούνται στο παιχνίδι

Στη συνέχεια θα γίνει παράθεση σε μορφή πίνακα όλων των μηχανισμών που συνθέτουν το πρότυπο του 3D r&c adventure που κατασκευάστηκε και υλοποιούνται στο παιχνίδι “Επίσκεψη στο μουσείο”. Οι περισσότεροι από αυτούς του μηχανισμούς υλοποιούνται σε όλα τα παιχνίδια του είδους, εντούτοις μερικοί μπορεί να διαφέρουν σημαντικά στην υλοποίησή τους, ενώ σε μερικά παιχνίδια κάποιοι μηχανισμοί μπορεί να παραλείπονται ή μπορεί να υπάρχουν και κάποιοι επιπλέον. Σε κάθε περίπτωση όμως, τουλάχιστον οι βασικοί από τους παρακάτω μηχανισμούς είναι κοινοί, και ουσιαστικά καθορίζουν ένα κοινό gameplay σε όλα τα r&c adventure με τέτοιο τρόπο που ένας παίκτης έστω και με μικρή εμπειρία σε αυτού του είδους τα βιντεοπαιχνίδια δεν χρειάζεται ξεχωριστές οδηγίες για να παίξει ένα καινούργιο παιχνίδι. Οι μηχανισμοί αυτοί χωρίστηκαν σε επτά ομάδες:

- α) Μηχανισμοί των τίτλων εισαγωγής (Πίνακας 4.3).
- β) Μηχανισμοί αρχικού μενού (Πίνακας 4.4).

- γ) Μηχανισμοί διεπαφής (User Interface - UI) (Πίνακας 4.5).
- δ) Μηχανισμοί κίνησης και ενεργειών κεντρικού χαρακτήρα (Πίνακας 4.6).
- ε) Μηχανισμοί για τις Σκηνές (scenes) (Πίνακας 4.7).
- στ) Μηχανισμοί Καμερών (Πίνακας 4.8).
- ζ) Μηχανισμοί εκμάθησης και εξέλιξης παιχνιδιού (Πίνακας 4.9).

Στις προηγούμενες ενότητες κάποιοι από αυτούς τους μηχανισμούς αναλύθηκαν λεπτομερώς.

Πίνακας 4.3: Μηχανισμοί των τίτλων εισαγωγής

Μηχανισμός	Σχόλια
i. 3D animation logo του δημιουργού και Slide εικόνα του σήματος του πανεπιστημίου	Το 3D animation έγινε με χρήση του εργαλείου timeline της Unity. Το 3D μοντέλο του κατασκευάστηκε στο πρόγραμμα blender ενώ προστέθηκαν και ηχητικά εφέ. Υπάρχει η δυνατότητα να γίνει άμεση προσπέραση με το κουμπί esc ή με αριστερό κλικ, διαφορετικά η πρώτη σκηνή με το 3D logo διαρκεί 6'' και αλλάζει στην επόμενη εικόνα με το σήμα του πανεπιστημίου που επίσης διαρκεί 6''.
ii. Fade in και Fade out	Στους τίτλους εισαγωγής, αλλά και σε κάθε αλλαγή σκηνής στο παιχνίδι, η εμφάνιση ξεκινάει με εφέ fade in και η έξοδος με fade out, δηλαδή με σταδιακή αύξηση του φωτισμού ή μείωση αντίστοιχα.
iii. Απεικόνιση των fps (καρέ ανά δευτερόλεπτο) και πληροφοριών του υπολογιστή	Στα δεξιά της εικόνας εμφανίζονται τα fps. Σαν στόχος στο έργο έχουν οριστεί τα 60 fps, που σημαίνει ότι δεν μπορούν να ξεπεραστούν. Στα σύγχρονα παιχνίδια που παίζονται από ένα παίκτη θεωρείται standard να τρέχουν στα 60 fps. Επειδή, όμως, το έργο αποτελείται από πολύπλοκες 3D σκηνές για να επιτευχθεί ο στόχος αυτός απαιτείται να υπάρχει κάρτα γραφικών με υψηλές αποδόσεις. Σε περίπτωση που οι αποδόσεις του υπολογιστή δεν είναι υψηλές και τα fps είναι λιγότερα, δύναται να μειωθεί η ποιότητα των γραφικών ή η ανάλυση από μενού που θα εξεταστεί παρακάτω και έτσι ανεβαίνουν τα fps. Σημειώνεται ότι στα εμπορικά παιχνίδια δεν εμφανίζεται η ένδειξη αυτή, αλλά εδώ προστέθηκε στο πλαίσιο της διπλωματικής. Επίσης, με το πλήκτρο "C" εμφανίζονται πληροφορίες - χαρακτηριστικά για το σύστημα (CPU και GPU) στα δεξιά της οθόνης.

Πίνακας 4.4: Μηχανισμοί αρχικού μενού

i. Κινούμενος φόντος στο αρχικό μενού	Στο φόντο του αρχικού μενού υπάρχει κινούμενη εικόνα από αριστερά στα δεξιά (scroll) η οποία εναλλάσσεται μεταξύ τριών εικόνων μετά από 7 δευτερόλεπτα.
ii. Μουσική αρχικού μενού	Ακούγεται μουσική επαναλαμβανόμενα, η ένταση της οποίας μπορεί να αυξομειωθεί από το μενού των ρυθμίσεων.
iii. Κουμπιά επιλογής.	Με την μετακίνηση του δείκτη πάνω σε οποιοδήποτε κουμπί επιλογής, αυτό αλλάζει χρώμα από μπλε σε κόκκινο και ακούγεται χαρακτηριστικός ήχος. Αν ο δείκτης απομακρυνθεί από το κουμπί, επανέρχεται το χρώμα του σε μπλε. Με αριστερό κλικ ακούγεται

	βαθύτερος ήχος και ενεργοποιείται η λειτουργία του πλήκτρου.
iv. Επιλογή "Νέο παιχνίδι"	Ξεκινάει η φόρτωση της πρώτης σκηνής του παιχνιδιού. Εμφανίζεται μία μπάρα φόρτωσης σε μαύρο φόντο που δείχνει τι μέρος της σκηνής φορτώθηκε στη μνήμη του υπολογιστή. Μόλις ολοκληρωθεί η φόρτωση ξεκινάει η πρώτη σκηνή του κυρίως παιχνιδιού. Παρόμοια μπάρα εμφανίζεται σε κάθε εναλλαγή σκηνών και εντός του παιχνιδιού.
v. Επιλογή "Φόρτωση"	Αρχικά αν δεν έχει αποθηκευτεί καμία εξέλιξη του παιχνιδιού η επιλογή είναι απενεργοποιημένη. Αν έχει γίνει αποθήκευση τότε η επιλογή μας επιτρέπει να ξεκινήσουμε το παιχνίδι ακριβώς από το σημείο που είχαμε κάνει την αποθήκευση. Μετά το κλικ ξεκινάει η φόρτωση της σκηνής όπου είχε γίνει η αποθήκευση.
vi. Επιλογή "Πληροφορίες"	Εμφανίζεται καινούργια οθόνη με διάφορες πληροφορίες για τον δημιουργό και το παιχνίδι ενώ στο κάτω μέρος βρίσκεται ένα κουμπί για επιστροφή στο κεντρικό μενού.
vii. Επιλογή "Ρυθμίσεις" Υπομενού: <ul style="list-style-type: none"> ○ Ανάλυση οθόνης ○ Πλήρης εικόνα ○ Ποιότητα γραφικών ○ Ένταση μουσικής 	Οι ρυθμίσεις εμφανίζονται σε ξεχωριστή οθόνη. Ο χρήστης μπορεί να επιλέξει ανάμεσα σε υποστηριζόμενες από την οθόνη και την κάρτα γραφικών αναλύσεις (resolutions). Όσο μεγαλύτερη ανάλυση τόσο πιο μεγάλη ευκρίνεια, αλλά και τόσο μεγαλύτερη απαίτηση για υπολογιστική ισχύς. Στις μεγαλύτερες αναλύσεις μπορεί να υπάρχει μείωση των fps. Υπάρχει κουμπί που ενεργοποιεί και απενεργοποιεί την πλήρη εικόνα (fullscreen) ή τη λειτουργία παραθύρου. Η επιλογή της ποιότητα γραφικών γίνεται ανάμεσα σε τρεις τιμές: χαμηλή (LowURP) - μέτρια (MediumURP) - υψηλή (HighURP). Ουσιαστικά υπάρχουν στο πρότυπο τρεις αποθηκευμένες ρυθμίσεις του Universal Render Pipeline που αλλάζουν το antialiasing, τις φωτοσκιάσεις (cast shadow) το PostProcessing και μερικές ιδιότητες ακόμη. Το HighURP απαιτεί περισσότερους πόρους κυρίως από την κάρτα γραφικών. Η ένταση της μουσικής ρυθμίζεται από μία μπάρα. Οι επιθυμητές ρυθμίσεις μπορούν να αποθηκευτούν και όταν επανεκκινήσει το παιχνίδι να φορτωθούν αυτόματα.
viii. Επιλογή "Έξοδος"	Γίνεται έξοδος από το παιχνίδι.

ΚΥΡΙΩΣ ΠΑΙΧΝΙΔΙ

Πίνακας 4.5: Μηχανισμοί διεπαφής (User Interface - UI)

i. Κουμπί μενού	Κατά τη διάρκεια του κυρίως παιχνιδιού υπάρχει πάνω αριστερά κουμπί που αν πατηθεί παγώνει το παιχνίδι και εμφανίζει ένα μενού στο κέντρο της οθόνης.
ii. Επιλογές μενού εντός του παιχνιδιού. <ul style="list-style-type: none"> ○ Αποθήκευση ○ Φόρτωση ○ Ρυθμίσεις ○ Επιστροφή ○ Έξοδος στην αρχική 	Πατώντας το κουμπί "Αποθήκευση" μπορεί να γίνει αποθήκευση της μέχρι εκείνης της στιγμής εξέλιξης του παιχνιδιού. Επειδή, υπάρχει μόνο μία θέση αποθήκευσης, προηγείται μία ερώτηση επαλήθευσης. Με την επιλογή "Φόρτωση" το παιχνίδι επανέρχεται σε προηγούμενη αποθηκευμένη θέση. Όταν πατηθεί, εμφανίζεται ερώτηση επαλήθευσης, γιατί η μέχρι εκείνη τη στιγμή εξέλιξη θα χαθεί. Όταν φορτωθεί το παιχνίδι, ο χαρακτήρας θα είναι στη θέση και γωνία όπως όταν αποθηκεύτηκε το παιχνίδι, τα πράγματα στο inventory θα είναι ίδια, και διάφορα αντικείμενα

	<p>της σκηνής θα αλλάξουν σε κατάσταση όμοια με αυτή όταν έγινε η αποθήκευση.</p> <p>Οι ρυθμίσεις είναι όμοιες με αυτές του αρχικού μενού.</p> <p>Το κουμπί επιστροφής επαναφέρει το παιχνίδι στην κανονική ροή του, ενώ με την "Εξοδο στην αρχική" τερματίζεται το παιχνίδι και επιστρέφει στο αρχικό μενού, αφού ο χρήστης επαληθεύσει την επιλογή του.</p>
iii. Inventory	<p>Πάνω δεξιά της οθόνης κατά την κανονική διάρκεια του παιχνιδιού υπάρχει ένα εικονίδιο σάκου που αν πατηθεί κατεβαίνει το παράθυρο του inventory στο πάνω μέρος της οθόνης. Αυτό αποτελείται από επτά τετράγωνα που γεμίζουν με τα αντικείμενα που συλλέγει ο χαρακτήρας. Αν έχει συλλέξει πάνω από επτά αντικείμενα ενεργοποιούνται και τα βελάκια για την κύλιση αριστερά δεξιά ώστε να είναι προσβάσιμα όλα τα αντικείμενα του inventory. Όταν ο δείκτης μετακινείται πάνω τους, αυτά αλλάζουν χρώμα. Τα τετράγωνα γεμίζουν με αντικείμενα σε μορφή στίβας από δεξιά στα αριστερά.</p>
<p>iv. Αλλαγή σχήματος του δείκτη του ποντικιού όταν μετακινείται:</p> <ul style="list-style-type: none"> ○ Σε σημείο που δεν υπάρχει αντικείμενο ○ Πάνω σε διαδραστικό αντικείμενο ○ Πάνω σε αντικείμενο του inventory ○ Όταν μπορεί να γίνει παράκαμψη (skip) 	<p>Όταν ο δείκτης κινείται σε σημείο όπου δεν υπάρχουν διαδραστικά αντικείμενα ή άλλοι χαρακτήρες στο περιβάλλον της σκηνής, έχει το κλασικό σχήμα ενός δείκτη ποντικιού. Το ίδιο σχήμα έχει και όταν βρίσκεται πάνω στα κουμπιά και τις επιλογές που υπάρχουν στην οθόνη ή όταν ο παίκτης πλοηγείται στα μενού.</p> <p>Όταν μετακινηθεί πάνω σε ένα διαδραστικό αντικείμενο μέσα στη σκηνή, ο δείκτης αλλάζει ανάλογα με την ενέργεια που θα γίνει μετά το αριστερό κλικ. Θα πάρει σχήμα φακού, για να εξετάσει το αντικείμενο, γραναζιού για να το χρησιμοποιήσει, φούσκας κειμένου για να ξεκινήσει διάλογο και εξόδου αν πρόκειται να βγει από την σκηνή. Αν ο δείκτης μετακινηθεί σε σημείο που ο χαρακτήρας δεν επιτρέπεται να πάει θα αλλάξει σε απαγορευτικό σχήμα. Η επιλογή της ενέργειας ρυθμίζεται μέσω του γραφικού περιβάλλοντος του Unity, στο inspector του κάθε αντικειμένου.</p> <p>Αν επιλεχθεί κάποιο αντικείμενο από το inventory, τότε ο δείκτης παίρνει το σχήμα του αντικειμένου μαζί με ένα γρανάκι που σημαίνει ότι πρόκειται να το χρησιμοποιήσει στην σκηνή.</p> <p>Στη μέση ενός διαλόγου, ή όταν παίζει κάποιο cutscene, το σχήμα του δείκτη γίνεται μπλε βελάκι με γραμμή, που υποδεικνύει ότι με αριστερό κλικ ή με esc μπορεί να παρακαμφθεί ο διάλογος ή το cutscene.</p>
v. Σημάδι στο σημείο εντός της σκηνής όπου έγινε κλικ	<p>Κάθε φορά που γίνεται αριστερό ή δεξί κλικ εντός της σκηνής, εμφανίζεται μία γκρι σφαίρα σαν καπνός για μερικά κλάσματα του δευτερολέπτου, που υποδεικνύει σε ποιο σημείο έγινε το κλικ.</p>
vi. Εμφάνιση στο κέντρο της οθόνης, της ενέργειας που πρόκειται να γίνει με αριστερό κλικ	<p>Όταν ο δείκτης του ποντικιού μετακινείται πάνω σε ένα διαδραστικό αντικείμενο εμφανίζεται κείμενο με κίτρινα γράμματα, το οποίο περιγράφει την ενέργεια που πρόκειται να γίνει με το αριστερό κλικ, όπως "κοίταξε τα κιβώτια", "πάρε το κουτάκι"</p>

Πίνακας 4.6: Μηχανισμοί κίνησης και ενεργειών κεντρικού χαρακτήρα

i. Αριστερό κλικ σε σημείο που δεν υπάρχει διαδραστικό αντικείμενο. Διπλό αριστερό κλικ για τρέξιμο	<p>Ο χαρακτήρας θα κινηθεί περπατώντας προς το σημείο. Θα επιλέξει την πιο σύντομη διαδρομή, αποφεύγοντας τα εμπόδια, με χρήση του εργαλείου nav-mesh surface. Αν το σημείο βρίσκεται εκτός επιτρεπτής επιφάνειας, ο χαρακτήρας θα σταματήσει στο πιο κοντινό επιτρεπτό σημείο. Αν γίνει διπλό αριστερό κλικ ο χαρακτήρας θα κινηθεί τρέχοντας.</p>
--	---

ii. Αριστερό κλικ πάνω σε διαδραστικό αντικείμενο	<p>Ο χαρακτήρας θα κινηθεί προς το αντικείμενο και όταν φτάσει σε μια ελάχιστη απόσταση, που υπάρχει στις ιδιότητές του, θα εκτελέσει κάποια ενέργεια ανάλογα το αντικείμενο. Κάποια διαδραστικά αντικείμενα έχουν συγκεκριμένο σημείο εκτός του σώματός τους, στο οποίο θα κινηθεί ο χαρακτήρας και όταν το πλησιάσει θα στραφεί προς το αντικείμενο και μετά θα εκτελέσει την ενέργεια.</p>
iii. Δεξί κλικ πάνω σε διαδραστικό αντικείμενο ή σε κενό σημείο	<p>Με δεξί κλικ πάνω σε διαδραστικό αντικείμενο ο χαρακτήρας στρέφεται προς αυτό και ξεκινάει την περιγραφή του. Αυτή εμφανίζεται στην οθόνη ενώ ακούγεται ταυτόχρονα. Μπορεί να αποτελείται από μία ή παραπάνω φράσεις που εμφανίζονται μία μία. Κάθε περιγραφή εισάγεται στο πρότυπο μέσω του γραφικού περιβάλλοντος του Unity, στο inspector του αντικειμένου. Όταν γίνεται δεξί κλικ σε σημείο όπου δεν υπάρχει διαδραστικό αντικείμενο, ο χαρακτήρας θα στραφεί προς το σημείο και θα πει την φράση "δεν υπάρχει τίποτα εκεί" κάνοντας ταυτόχρονα και μια χειρονομία.</p>
iv. Αριστερό κλικ σε άλλο χαρακτήρα. Έναρξη διαλόγου, επιλογές διαλόγου.	<p>Αν μετακινηθεί ο δείκτης πάνω σε άλλο χαρακτήρα, αλλάζει σχήμα σε φούσκα κειμένου. Με το αριστερό κλικ ο κεντρικός χαρακτήρας θα πλησιάσει τον άλλο χαρακτήρα και στην ελάχιστη απόσταση ενεργοποίησης θα ξεκινήσει ο διάλογος. Κατά την διάρκεια του διαλόγου οι χαρακτήρες μπορεί να κάνουν ξεχωριστές κινήσεις σώματος. Το κείμενο του διαλόγου εμφανίζεται ακριβώς πάνω από τον χαρακτήρα που μιλάει ενώ ταυτόχρονα ακούγεται και το φωνητικό. Το πρότυπο υποστηρίζει και τον συγχρονισμό της κίνησης του στόματος με τη φωνή. Κάθε φράση του διαλόγου μπορεί να παρακαμφθεί με κλικ του ποντικιού. Κάποιοι διάλογοι διακόπτονται για να εμφανιστούν στο κάτω μέρος της οθόνης επιλογές με φράσεις, ανάμεσα στις οποίες ο παίκτης μπορεί να διαλέξει. Από την επιλογή της φράσης εξαρτάται η συνέχεια του διαλόγου. Οι διάλογοι εισάγονται στο πρότυπο μέσω του γραφικού περιβάλλοντος του Unity, σε κάθε σκηνή ξεχωριστά. Οι διάλογοι έχουν κάποια αρίθμηση, που επιτρέπει να υπάρχει σύνδεση ανάμεσα στον προηγούμενο και στον επόμενο.</p>
v. Αριστερό κλικ σε αντικείμενο για εξέταση.	<p>Αν μετακινηθεί ο δείκτης σε διαδραστικά αντικείμενα με ιδιότητα ενέργειας "εξέταση" αλλάζει σε μεγεθυντικό φακό και με αριστερό κλικ, ο χαρακτήρας τα πλησιάζει και στην ελάχιστη απόσταση ενεργοποίησης ξεκινάει η εξέτασή τους. Αυτή εξαρτάται από την ιδιότητα του αντικειμένου. Σε περίπτωση που υπάρχει απλά το component ExamineObject.cs τότε απλά εμφανίζεται και ακούγεται η περιγραφή του. Αν υπάρχει κλάση παιδί της ExamineObject.cs τότε μπορεί να γίνουν ξεχωριστές για κάθε αντικείμενο ενέργειες, όπως η εστίαση της κάμερας, ή να αλλάξει η ενέργεια στο επόμενο κλικ σε χρησιμοποίηση του αντικειμένου.</p>
vi. Αριστερό κλικ σε αντικείμενο για την χρησιμοποίησή του	<p>Αν μετακινηθεί ο δείκτης σε διαδραστικά αντικείμενα με ιδιότητα ενέργειας "χρησιμοποίηση" αλλάζει σε γρανάζι και με αριστερό κλικ, ο χαρακτήρας τα πλησιάζει και στην ελάχιστη απόσταση ενεργοποίησης ξεκινάει η χρησιμοποίησή τους. Η ενέργεια που θα γίνει η προγραμματισμένη για κάθε αντικείμενο ξεχωριστά σε κλάση που κληρονομεί από την UseObject.cs. Συνήθως, ο χαρακτήρας εκτελεί ένα animation που δείχνει να χρησιμοποιεί το αντικείμενο.</p>
vii. Αριστερό κλικ σε αντικείμενο για τον συνδυασμό	<p>Ανοίγοντας το inventory υπάρχει η δυνατότητα να επιλεγθεί ένα αντικείμενο εντός αυτού με αριστερό κλικ πάνω του και να συνδυαστεί με αντικείμενο της σκηνής κάνοντας στη συνέχεια</p>

αντικειμένου από το inventory με αντικείμενο σκηνής	αριστερό κλικ σε αντικείμενο της σκηνής. Ο χαρακτήρας θα κάνει τις ίδιες ενέργειες όπως όταν χρησιμοποιεί ένα αντικείμενο αλλά αυτή τη φορά θα κάνει συνδυασμό αντικειμένων.
viii. Αριστερό κλικ σε αντικείμενο για την συλλογή του	Αν μετακινηθεί ο δείκτης σε διαδραστικά αντικείμενα με ιδιότητα ενέργειας "συλλογή" , αλλάζει σε χέρι και με αριστερό κλικ, ο χαρακτήρας τα πλησιάζει. Όταν φτάσει στην ελάχιστη απόσταση ενεργοποίησης τα συλλέγει και τα τοποθετεί στο inventory. Για αυτή την ενέργεια ο χαρακτήρας απλώνει το δεξί του χέρι στο αντικείμενο το οποίο εξαφανίζεται από την σκηνή. Το animation αυτό υλοποιείται με animation rig του χεριού. Αν το αντικείμενο είναι στο έδαφος ο χαρακτήρας σκύβει για να το συλλέξει.
ix. Αριστερό κλικ σε αντικείμενο για έξοδο σε άλλη σκηνή	Αν μετακινηθεί ο δείκτης σε σημεία εξόδου, όπως σε μία πόρτα, αλλάζει σε σήμα εξόδου. Με αριστερό κλικ πάνω του, ο χαρακτήρας εισέρχεται σε αυτό το σημείο και αλλάζει η ενεργή σκηνή. Πρώτα εμφανίζεται η οθόνη φόρτωσης σκηνής και στη συνέχεια η καινούργια σκηνή με τον χαρακτήρα να εισέρχεται σε αυτήν.
x. Animation χαρακτήρα	Για να γίνει το παιχνίδι ρεαλιστικό, έχουν προστεθεί στον χαρακτήρα 3D animation για κάθε κίνηση ή ενέργεια που κάνει. Για την κίνηση του υπάρχουν ξεχωριστά animation για το περπάτημα (ευθεία,αριστερά και δεξιά), το τρέξιμο (ευθεία, αριστερά, δεξιά) την επιτόπια στροφή (αριστερά, δεξιά) και την αδράνεια (idle). Αυτά απαρτίζουν την κατάσταση (state) <i>Locomotion</i> του "humanoid animator" που βρίσκεται στα standard assets της Unity. Όταν ο χαρακτήρας βρίσκεται σε αυτή την κατάσταση τα animation εναλλάσσονται μεταξύ τους και προσαρμόζουν την ταχύτητά τους αυτόματα, ανάλογα της κατεύθυνσης και της ταχύτητας του χαρακτήρα. Σε κάθε βήμα ακούγεται και ένας χαρακτηριστικός ήχος. Υπάρχουν πολλά άλλα animation που το καθένα αποτελεί μια άλλη κατάσταση στον animator. Οι μεταβάσεις στις καταστάσεις γίνονται με ενεργοποίηση παραμέτρων triggers μέσα από τον κώδικα του προτύπου, ενώ έχουν παραμετροποιηθεί έτσι που να δείχνουν ομαλές. Για παράδειγμα όταν ο χαρακτήρας πάει να συλλέξει ένα αντικείμενο από το έδαφος αλλάζει η κατάσταση από τη <i>Locomotion</i> που είναι μέχρι να φτάσει στο σημείο στη <i>PickUp</i> και όταν τελειώσει το anim που σηκώνει το αντικείμενο τότε επιστρέφει στην κατάσταση <i>Locomotion</i> . Αν μείνει ακίνητος ο χαρακτήρας στο ίδιο σημείο μετά από 20" αλλάζει η κατάσταση σε άλλη με anim που δείχνει τον χαρακτήρα να τεντώνει τον λαιμό και τα χέρια του.
xi. Animation rigs για το κεφάλι και το χέρι	Ο χαρακτήρας μπορεί να στρέφει το κεφάλι του σε αντικείμενα ενδιαφέροντος. Αυτή η κίνηση του κεφαλιού γίνεται ανεξάρτητα του animation που εκτελεί μέσα στη συγκεκριμένη κατάσταση του animator με τεχνική που ονομάζεται animation rig. Επίσης, το δεξί χέρι έχει την δυνατότητα να κινείται ανεξάρτητα από το anim του χαρακτήρα, όταν για παράδειγμα συλλέγει πράγματα ή όταν ενεργοποιεί ή όταν δίνει αντικείμενα. Με αυτό τον τρόπο επιτυγχάνεται η ακρίβεια της κίνησης του χεριού, ανεξάρτητα από την θέση που έχει ο χαρακτήρας.
xii. Κόκκινο περίγραμμα του χαρακτήρα	Όταν παρεμβάλλεται ανάμεσα στον χαρακτήρα και την κάμερα ένα εμπόδιο, που δεν επιτρέπει σε αυτόν να είναι ορατός, ενεργοποιείται ένα κόκκινο περίγραμμα στον χαρακτήρα, που επιτρέπει στον παίκτη να γνωρίζει την ακριβή του θέση.

Πίνακας 4.7: Μηχανισμοί για τις Σκηνές (scenes)

<p>i. κινηματικές (cinematic) σκηνές</p>	<p>Το κυρίως παιχνίδι ξεκινάει με μία κινηματική τρισδιάστατη σκηνή που δείχνει την άφιξη του κεντρικού χαρακτήρα στο μουσείο με λεωφορείο. Χρησιμοποιήθηκε το εργαλείο timeline που περιέχει τα εξής κινούμενα αντικείμενα: ξεχωριστή κάμερα που χρησιμοποιείται αποκλειστικά για κινηματικές σκηνές, το λεωφορείο με δύο κινούμενες ρόδες, δύο αυτοκίνητα και ο κεντρικός χαρακτήρας. Υπάρχει δυνατότητα παράλειψης κάθε κινηματικής σκηνής με το κουμπί ESC ή αριστερό κλικ, ενώ κατά την διάρκεια της, ο δείκτης αλλάζει σε μορφή play. Κατά τη διάρκεια της κινηματικής σκηνής καμία λειτουργία δεν είναι δυνατή πλην της παράλειψης.</p>
<p>ii. Πηγές φωτός και σκιές</p>	<p>Κάθε σκηνή έχει τα δικά της αντικείμενα για πηγές φωτός. Από την θέση τους και τον τρόπο που παράγουν φως δημιουργούνται οι σκιές όλων των άλλων αντικειμένων. Στην εξωτερική σκηνή μοναδική πηγή φωτός είναι ένα Directional light που προσομοιώνει το φως του ήλιου. Στη εσωτερική σκηνή του μουσείου υπάρχουν δεκατέσσερα point lights που η θέση τους είναι μέσα στα αντικείμενα των προβολέων που υπάρχουν στην οροφή της αίθουσας. Αυτά δημιουργούν ρεαλιστικές σκιές, οι ένταση των οποίων εξαρτάται από την απόσταση των αντικειμένων από όλους τους φωτισμούς.</p>
<p>iii. Μη διαδραστικά στατικά αντικείμενα</p>	<p>Οι σκηνές του κυρίως παιχνιδιού περιέχουν πολλά μη διαδραστικά αντικείμενα, πάνω στα οποία όταν μετακινείται ο δείκτης δεν αλλάζει σχήμα και ο κεντρικός χαρακτήρας δεν κάνει κάποια ξεχωριστή ενέργεια όταν γίνεται κλικ επάνω τους. Τέτοια αντικείμενα είναι τα δέντρα, οι κολώνες φωτισμού, τα πεζοδρόμια, τα κολωνάκια και άλλα. Όσα από τα αντικείμενα είναι μέσα στον χώρο όπου μπορεί να κινείται ο χαρακτήρας, περιέχουν το component collider, που είναι ένα γεωμετρικό σχήμα, συνήθως ορθογώνιο παραλληλεπίπεδο, γύρω από το αντικείμενο, που δεν επιτρέπει στον χαρακτήρα να εισέλθει εντός αυτού. Το collider του κάθε αντικειμένου χρησιμοποιείται από το εργαλείο nav mesh surface για να αναγνωριστούν τα αντικείμενα ως εμπόδια στην κίνηση του χαρακτήρα. Επίσης, χρησιμοποιείται και σε όλες τις επιφάνειες που μπορεί να κινούνται πάνω τους οι χαρακτήρες του παιχνιδιού, όπως τα πεζοδρόμια, τα σκαλοπάτια, το δάπεδο του μουσείου και λοιπά.</p>
<p>iv. Μη διαδραστικά κινούμενα αντικείμενα</p>	<p>Στην εξωτερική σκηνή υπάρχουν και αντικείμενα του σκηνικού που κινούνται. Σε καθένα από αυτά έχει προστεθεί το component animator που ελέγχει τα 3D animations που έχουν δημιουργηθεί εντός της Unity για την κίνηση του αντικειμένου αυτού. Για την κίνηση των αυτοκινήτων έχει κατασκευαστεί ειδικό script που ενεργοποιεί την κίνηση τους κατά μήκος του δρόμου σε τυχαίους χρόνους. Επίσης, μέσα από τον κώδικα ενεργοποιούνται και τα animation στην έκδοση εισιτηρίου. Όλα αυτά τα animation είναι απλές μετακινήσεις της θέσης και αλλαγές της γωνίας ολόκληρων των αντικειμένων και υλοποιούνται απλά. Αντίθετα, οι τεχνικές του animation rig και του blendshapes που υλοποιούνται σε άλλα σημεία του παιχνιδιού είναι πιο πολύπλοκες.</p>
<p>v. Διαδραστικά αντικείμενα</p>	<p>Όπως έχει ήδη περιγραφεί, υπάρχουν διαδραστικά αντικείμενα στις κυρίως σκηνές του παιχνιδιού, που όταν μετακινείται ο δείκτης πάνω τους αλλάζει σχήμα και με το αριστερό κλικ πάνω τους εκτελείται κάποια ενέργεια. Κάθε διαδραστικό αντικείμενο διαθέτει το component InteractObject.cs που του δίνει τις ιδιότητες του ονόματος, της περιγραφής και του είδους της</p>

	ενέργειας που θα κάνει ο χαρακτήρας, όπως η περισυλλογή του, η χρησιμοποίησή του ή η περιγραφή του. Επίσης, διαθέτουν το component collider που πέρα από τις ιδιότητες που δίνει όπως στα μη διαδραστικά αντικείμενα, είναι απαραίτητος για την αναγνώριση δείκτη της μετακίνησης του δείκτη πάνω σε αυτά.
vi. Άλλοι χαρακτήρες	Στην σκηνή εκτός από τον κεντρικό χαρακτήρα, υπάρχουν άλλοι δύο άντρες, ένας που αθλείται και ένας ελεγκτής εισιτηρίου, και μία γυναίκα υπαλληλος στην έκδοση εισιτηρίων. Με αριστερό κλικ πάνω τους ξεκινάει διάλογος και ο καθένας έχει διαφορετική φωνή. Επίσης, καθένας έχει τα δικά του animation. Επιπλέον, η γυναίκα υπαλληλος και ο ελεγκτής έχουν animation rig για την κίνηση του κεφαλιού που στρέφεται προς τον κεντρικό χαρακτήρα. Στον ελεγκτή έχει προστεθεί και animation rig του χεριού που παραλαμβάνει το εισιτήριο ενώ ο αθλούμενος κάνει μια επαναλαμβανόμενη κίνηση: διατάσεις τρέξιμο προς δεξιά, διατάσεις τρέξιμο αριστερά. Όταν γίνει αριστερό κλικ πάνω του, διακόπτει την άθληση και στρέφεται προς τον χαρακτήρα για να μιλήσει, ενώ όταν τελειώνει ο διάλογος συνεχίζει από το σημείο που βρισκόταν την άθληση.
vii. Φυσικές ιδιότητες σε αντικείμενα	Στην σκηνή εξωτερικά του μουσείου υπάρχουν τρεις κούτες που έχουν το component RigidBody , το οποίο προσθέτει φυσικές ιδιότητες μάζας στα αντικείμενα. Όταν ο παίκτης οδηγήσει τον χαρακτήρα πάνω στα κιβώτια, αυτά θα πέσουν λόγω της βαρύτητας τους στο έδαφος με απόλυτα φυσικό τρόπο.
viii. Εξωτερικό κόκκινο περίγραμμα	Σε όλα τα διαδραστικά αντικείμενα έχει προστεθεί το component Outline για την εμφάνιση κόκκινου περιγράμματος του τρισδιάστατου αντικειμένου όταν ο παίκτης μετακινήσει τον δείκτη πάνω τους. Αυτό το έντονο κόκκινο περίγραμμα κάνει το αντικείμενο να ξεχωρίζει από το περιβάλλον και δείχνει στον παίκτη ότι μπορεί να γίνει κάποια ενέργεια με αυτό. Για την χρήση του εργαλείου Outline πρέπει να εγκατασταθεί το πακέτο Quick Outline που δεν εμπεριέχεται στην αρχική έκδοση της Unity.
ix. Μουσική εντός παιχνιδιού	Κατά την κανονική διάρκεια του παιχνιδιού ακούγονται μουσικά κομμάτια επαναλαμβανόμενα. Σε κάθε σκηνή το μουσικό κομμάτι είναι διαφορετικό, ενώ η ένταση της μουσικής μπορεί να αυξημειωθεί από το μενού των ρυθμίσεων εντός παιχνιδιού.
x. Μουσικά εφέ	Εκτός από την μουσική κατά τη διάρκεια του παιχνιδιού ακούγονται και διάφοροι ήχοι από το περιβάλλον της σκηνής, καθώς και ήχοι από διάφορα αντικείμενα. Στην εξωτερική σκηνή του μουσείου ακούγεται το κελήθισμα των πουλιών, ο ήχος από τα αυτοκίνητα, ο ήχος από την εκτύπωση του εισιτηρίου και άλλοι πολλοί ήχοι. Η ένταση από τα ηχητικά εφέ είναι ανάλογη της απόστασης της πηγής από την ενεργή κάμερα. Δηλαδή, όσο πιο κοντά στο αυτοκίνητο είναι η κάμερα, τόσο δυνατότερος ο ήχος της μηχανής του. Αυτό επιτυγχάνεται με την προσθήκη του component AudioSource στα GameObjects.

Πίνακας 4.8: Μηχανισμοί Καμερών

i. Προοπτική τρίτου προσώπου	Σε κάθε σκηνή υπάρχει τουλάχιστον μία βασική κάμερα και μία βοηθητική, αλλά μόνο μία είναι ενεργή. Κατά τη διάρκεια του παιχνιδιού, η βασική κάμερα που είναι ενεργή, βρίσκεται πάνω από τον κεντρικό χαρακτήρα σε γωνία περίπου 45 μοιρών και τον ακολουθεί, διατηρώντας τον στο κέντρο του πλάνου. Η κίνησή της περιορίζεται από κάποια όρια που ορίζονται στις ιδιότητες της κάθε κάμερας ξεχωριστά.
-------------------------------------	---

ii. Εστιάσεις της βοηθητικής κάμερας	Σε ορισμένα αντικείμενα όταν γίνεται αριστερό κλικ πάνω τους, ενεργοποιείται η βοηθητική κάμερα, η οποία εστιάζει πάνω σε αυτά. Τότε παύει να είναι ο κεντρικός χαρακτήρας στο κέντρο του πλάνου. Όταν ενεργοποιείται η βοηθητική κάμερα, υπάρχει μία κίνηση κατά την οποία αλλάζει ομαλά η εστίαση από τον χαρακτήρα στο αντικείμενο. Η επιστροφή στη βασική κάμερα, γίνεται απλά με αριστερό κλικ σε σημείο όπου δεν υπάρχει άλλο διαδραστικό αντικείμενο και ο δείκτης έχει πάρει την μορφή εξόδου.
iii. Αλλαγή βασικών καμερών	Όταν ο χαρακτήρας περάσει σε τομέα της σκηνής που παρακολουθείται από άλλη κάμερα, γίνεται αλλαγή της ενεργής κάμερας και ο χαρακτήρας ακολουθείται από την καινούργια. Για παράδειγμα όταν ο χαρακτήρας ανέβει τα σκαλιά της εισόδου του μουσείου, αλλάζει η ενεργή κάμερα σε αυτή που βρίσκεται στην είσοδο.
iv. Εστίαση βασικής κάμερας	Με τα πλήκτρα + και - αυξομειώνεται η απόσταση της βασικής κάμερας από τον χαρακτήρα, ενώ με το πλήκτρο * επανέρχεται στη προκαθορισμένη απόσταση.
v. Εφέ post processing	Σε κάθε ενεργή κάμερα μπορούν να προστεθούν διάφορα εφέ εικόνας, που στη Unity ονομάζονται post processing effects . Τέτοια εφέ είναι η ομίχλη, η παραμόρφωση του φακού, η βινιέτα και πολλά άλλα. Στη συγκεκριμένη υλοποίηση εφαρμόζεται στην κάμερα εξωτερικά του μουσείου το εφέ Depth of Field , όπου τα αντικείμενα στο φόντο φαίνονται πιο θολά από τα αντικείμενα μπροστά στην κάμερα.

Πίνακας 4.9 Μηχανισμοί εκμάθησης και εξέλιξης παιχνιδιού

i. Εκμάθηση (tutorial)	Όταν ξεκινάει ένα καινούργιο παιχνίδι, και μετά το πρώτο cutscene, γίνεται ερώτηση προς τον παίκτη μέσω ενός παραθύρου, εάν επιθυμεί να ξεκινήσει η εκμάθηση του παιχνιδιού ή όχι. Σε θετική περίπτωση, κατά τη διάρκεια του παιχνιδιού εμφανίζονται μηνύματα στο κάτω μέρος της οθόνης, με οδηγίες που πρέπει να εκτελέσει ο παίκτης. Οι οδηγίες είναι χωρισμένες σε βήματα και για να προχωρήσει η εκμάθηση στο επόμενο βήμα πρέπει να ολοκληρωθεί το προηγούμενο. Σε κάθε βήμα υπάρχει περιορισμός στο χώρο που μπορεί να κινείται ο χαρακτήρας, με χρήση απαγορευτικής κορδέλας που δεν επιτρέπει την διέλευση πριν ολοκληρωθεί το βήμα. Οι οδηγίες αφορούν την εκμάθηση του χειρισμού του παιχνιδιού, δηλαδή της κίνησης, της εξερεύνησης, της συλλογής και χρησιμοποίησης των αντικειμένων, της χρήσης του inventory και των διαλόγων.
ii. Εξέλιξη του παιχνιδιού	Σε κάθε adventure game, υπάρχουν μηχανισμοί που ξεκλειδώνουν την εξέλιξη του παιχνιδιού. Αυτοί οι μηχανισμοί συνήθως εμπεριέχουν κάποιους γρίφους που πρέπει να λύσει ο παίκτης. Στη συγκεκριμένη υλοποίηση που είναι για επίδειξη των μηχανισμών και το σενάριο είναι πολύ απλό χωρίς γρίφους, αρχικά η είσοδος του μουσείου είναι κλειστή. Αν ο παίκτης προσπαθήσει να την ανοίξει χωρίς εισιτήριο θα του απαγορευτεί η είσοδος από τον υπάλληλο του μουσείου. Πρέπει πρώτα να οδηγήσει τον κεντρικό χαρακτήρα στο μηχανήμα εισιτηρίων, και αφού ελέγξει το πορτοφόλι του να αγοράσει ένα εισιτήριο. Έπειτα με την επίδειξη του στον υπάλληλο θα του επιτραπεί η είσοδος.
iii. Αποθήκευση της εξέλιξης	Το παιχνίδι ελέγχει την μέχρι στιγμής εξέλιξη του και αποθηκεύει σε μεταβλητές ποιοι μηχανισμοί έχουν ενεργοποιηθεί και ποια

	<p>σημεία έχουν ξεκλειδώσει. Έτσι κάθε φορά που φορτώνεται μια σκηνή, ελέγχονται οι μεταβλητές και τελικά η σκηνή εμφανίζεται με τις μέχρι τότε αλλαγές που έχουν επέλθει από την αρχή του παιχνιδιού. Για παράδειγμα, αν μια πόρτα έχει ξεκλειδώσει και αν ένα αντικείμενο έχει συλληχθεί από μία σκηνή, αν ξαναφορτωθεί αυτή η σκηνή, η πόρτα θα παραμένει ξεκλειδωμένη και το αντικείμενο δεν θα εμφανιστεί. Αυτή η εξέλιξη μπορεί επίσης να αποθηκευτεί και σε αρχείο (save file) που μπορεί να φορτωθεί όταν τερματιστεί και ξαναξεκινήσει το παιχνίδι.</p>
--	--

4.5. Γραφικά του παιχνιδιού

Όλα τα 3D γραφικά του παιχνιδιού χρησιμοποιούν το **Universal Render Pipeline** αντί του προεπιλεγμένου στη Unity, Standard Pipeline, με αποτέλεσμα την βελτιστοποίηση των απεικονίσεων των επιφανειών, των φωτισμών και των αντανακλάσεων, αλλά και την απαίτηση για μεγαλύτερη υπολογιστική ισχύ. Υπάρχει, όμως, πάντα η δυνατότητα, μέσω των ρυθμίσεων των γραφικών από το κεντρικό μενού, να μειωθεί η ποιότητα των γραφικών προς όφελος της ταχύτητας του παιχνιδιού.

Στην εισαγωγική οθόνη του παιχνιδιού με το animation του λογότυπου (Εικόνα 4.12), το μοντέλο του χειριστηρίου σχεδιάστηκε εντός Unity από απλά γεωμετρικά σχήματα, ενώ τα τρισδιάστατα γράμματα δημιουργήθηκαν στο πρόγραμμα blender. Ο φόντος του ουρανού δημιουργήθηκε με το εφέ απόδοσης σύννεφων του **paint.net**.



Εικόνα 4.12: Εισαγωγική οθόνη με λογότυπο δημιουργού

Στο κεντρικό μενού της αρχικής οθόνης αλλά και σε ολόκληρη την διεπαφή (UI) παιχνιδιού, για τα κείμενα στις λεζάντες και στα κουμπιά χρησιμοποιήθηκε το εργαλείο του Unity **TextMeshPro** και η γραμματοσειρά **CANDARA** που αναπτύχθηκε από την Microsoft. Για τους χρωματισμούς επιλέχθηκε η παλέτα χρωμάτων "**british palette**" από την ιστοσελίδα "**Flat UI Colors**" (Ahmet Sulek, 2023) (Εικόνα 4.13). Στον φόντο εναλλάσσονται εικόνες από το εξωτερικό και το εσωτερικό του μουσείου που παρουσιάζονται παρακάτω.



Εικόνα 4.13: British παλέτα χρωμάτων με δημιουργό τον Jan Losert από την ιστοσελίδα <https://flatuicolors.com/>

Ο κεντρικός και οι υπόλοιποι χαρακτήρες έχουν ληφθεί από την ιστοσελίδα **Mixamo** (Adobe, 2023) και είναι ελεύθεροι για κάθε χρήση. Εκτός από την μορφή έχουν ληφθεί και οι κινήσεις τους (3D animation). Οι μορφές τους δεν έχουν δεχτεί καθόλου επεξεργασία, εκτός από το πρόσωπο της γυναίκας υπαλλήλου, όπου έχουν προστεθεί άλλες δύο εκφράσεις προσώπου με τη μέθοδο *blendshapes* στο πρόγραμμα *blender*. Η μία απεικονίζει το πρόσωπο όταν εκφωνείται το φωνήεν “α” και η άλλη όταν εκφωνείται το φωνήεν “ο” και χρησιμοποιούνται για τον συγχρονισμό του στόματος με την ομιλία (*lip Sync*). Οι κινήσεις, επίσης, δεν έχουν υποστεί επεξεργασία, πλην κάποιων περικοπών για χρήση μέρους της ληφθείσας κίνησης και ενώσεων των κομματιών μεταξύ τους, που έγιναν επίσης με την χρήση του *blender*.

Στην σκηνή εξωτερικά του μουσείου πολλά από τα αντικείμενα στον χώρο έχουν ληφθεί δωρεάν από την ιστοσελίδα **TurboSquid**. Συγκεκριμένα, τα 3D αντικείμενα που έχουν κατεβαστεί δωρεάν περιγράφονται στον Πίνακα 4.10.

Περιγραφή Αντικειμένου	Όνομα Asset	Δημιουργός
Στάση λεωφορείου	3D Bus Stop model	Harmony Lands
Λεωφορείο	3D model Indonesian Bus Ecoline Rahayu Santosa Free 3D model	Agung Kuncoro
Αυτοκίνητο μάρκας Volkswagen μοντέλο Touran	Volkswagen Touran 2007 lowpoly model	Igor Shmidt
Αυτοκίνητο μάρκας Volkswagen μοντέλο Beetle	3D model Volkswagen Beetle 3D model	ProPolyModels
Αυτοκίνητο μάρκας Renault μοντέλο Sandero	3D Renault Sandero II model	wther
Αυτοκίνητο μάρκας Mercedes-Benz μοντέλο CL600 2007	Free Mercedes-Benz CL600 2007	ABCity3D

Αλουμινένιο κουτάκι κόλα	Metal Can 3D model	DanJ08
Κάδος σκουπιδιών	Park Bin 3D model	Aesterial Arts
Κολώνα φωτισμού	3D Street lamp single	Annet007
Μηχάνημα έκδοσης εισιτηρίου	Bitcoin ATM model	BitModel
Κάμερα παρακολούθησης CCTV	Altcam DCF51IR surveillance camera 3D model	DM_Pyatkin
Σηματοδότες κυκλοφορίας οχημάτων	Traffic Lights - Sign 3D model	hockenzie

Πίνακας 4.10: Δωρεάν Αντικείμενα 3D που έχουν ληφθεί από την ιστοσελίδα <https://www.turbosquid.com/>

Τα ανωτέρω αντικείμενα έχουν υποστεί μικρή επεξεργασία εντός **Unity**, κυρίως ως προς τους χρωματισμούς και τις υφές τους.

Για το γρασίδι και τους θάμνους έγινε χρήση του εργαλείου **terrain** εντός της Unity. Με το ίδιο εργαλείο χρωματίστηκε και η άσφαλτος και το χαλικόστρωτο μονοπάτι μέσα στο πράσινο.

Για τα δέντρα προτιμήθηκε να μην χρησιμοποιηθούν πολύ ρεαλιστικά μοντέλα, γιατί οι λεπτομερείς απεικονίσεις των φύλλων απαιτούν μεγάλους υπολογιστικούς πόρους και ρίχνει την απόδοση. Για τον εμπλουτισμό της σκηνής με δέντρα κατεβάστηκαν μερικά από τα δωρεάν πακέτα με 3D μοντέλα δέντρων που προσφέρει το Unity Asset Store:

Realistic Tree Pack Vol.1 (δημιουργός **PolyFix**),

Realistic Tree 9,10, (δημιουργός **Pixel Games**)

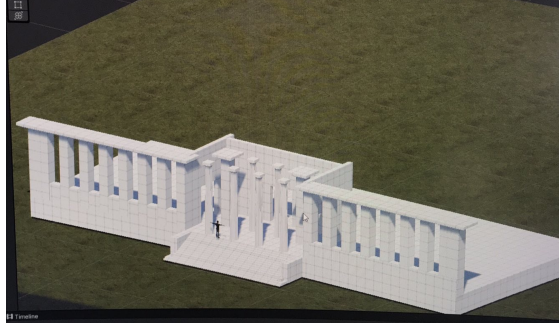
Mobile Tree Package, (δημιουργός **Laxer**)

Από το asset store, επίσης κατεβάστηκε και το μοντέλο του ξύλινου κουτιού:

Wood Crate YN, (δημιουργός **Yuji Nagata**)

Αντικείμενα που αποτελούνται από απλά γεωμετρικά σχήματα φτιάχτηκαν εντός της Unity. Τέτοια είναι τα κολωνάκια, οι πολυκατοικίες, τα πεζοδρόμια οι παγκοι, το μάρμαρο της πλατείας, το περίπτερο οι θύρες εισόδου του μουσείου, το εισιτήριο, οι θύρες εισόδου και η μαρμαρίνη στήλη στην πλατεία. Η οριζόντια σήμανση στην άσφαλτο έγινε με εισαγωγή εικόνων που σχεδιάστηκαν στο **paint.net** και χρήση τους ως **sprite**.

Το κτήριο του μουσείου κατασκευάστηκε από την αρχή με το εργαλείο **Pro builder**. Οι αρχαιοελληνικές κολώνες έχουν ληφθεί από το μοντέλο του παρθενώνα, το οποίο αγοράστηκε από την ιστοσελίδα της **Sketchfab** και δημιουργός του είναι ο **Myles Zhang** (Εικόνα 4.14).



Εικόνα 4.14: Εξέλιξη της κατασκευής του κτηρίου με το Pro builder

Η εσωτερική σκηνή του μουσείου μαζί με όλα τα εκθέματα, είναι αγορασμένη από το asset store: **Museum VR Asset Pack Vol. I**, δημιουργός **IGIIID**.

Τέλος, οι διάφορες δισδιάστατες εικόνες είναι δημιουργημένες στο paint.net εκτός από την εικόνα του σακιδίου που έχει ληφθεί από την ιστοσελίδα <https://iconduck.com> και είναι ελεύθερη για χρήση.

4.6. Ήχοι του παιχνιδιού

Υπάρχουν διάφορα ηχητικά εφέ που ακούγονται στο παιχνίδι, στοιχείο που είναι σημαντικό για κάθε βιντεοπαιχνίδι, γιατί εμπλουτίζει τη συνολική εμπειρία και ευχαρίστηση του παίκτη. Στα μενού όταν ο δείκτης του ποντικιού περνάει πάνω από μία επιλογή ή όταν γίνεται κλικ πάνω της ακούγεται χαρακτηριστικό κλικ. Επίσης, κατά την κανονική διάρκεια του παιχνιδιού ακούγονται ήχοι κλικ, όταν γίνεται αριστερό ή δεξί κλικ με το ποντίκι ή όταν ο δείκτης μετακινείται στο Inventory ή όταν ο χαρακτήρας κάνει μία ενέργεια. Όλοι οι ήχοι των κλικ έχουν ληφθεί από το βίντεο στο youtube με τίτλο "Free UI Click Sound Effects".

Κατά την κανονική διάρκεια του παιχνιδιού ακούγονται διάφοροι ήχοι από το περιβάλλον, όπως οι μηχανές των αυτοκινήτων, το λεωφορείο που σταματάει, το κελήδισμα πουλιών, τα βήματα του κεντρικού χαρακτήρα κ.λ.π. Τα ηχητικά εφέ είναι κατεβασμένα από ελεύθερες πηγές του διαδικτύου και συγκεκριμένα από τα site: <https://pixabay.com/sound-effects> και <https://freesound.org>.

Υπάρχει και επιγλώττιση (voice over) όταν ο κεντρικός χαρακτήρα κάνει διάλογο με άλλους, όταν περιγράφει αντικείμενα και όταν αφηγείται. Η φωνή ακούγεται ταυτόχρονα με την εμφάνιση του κειμένου στην οθόνη. Οι ανδρικές φωνές του κεντρικού χαρακτήρα, και των άλλων δύο ανδρών έξω από το μουσείο προέρχονται από την φωνή του συντάκτη, με αλλαγή της οξύτητας (pitch) ενώ για την ομιλία της γυναίκας χρησιμοποιήθηκε γυναικεία φωνή. Όλες οι φωνές

ηχογραφήθηκαν στο κινητό apple iphone 7 Plus με χρήση ακουστικών και μικροφώνου.

Τα αρχεία των ηχητικών εφέ και της επιγλώττισης είναι όλα σε μορφή mp3 με την πλειονότητά τους να έχει ρυθμό μετάδοσης - ποιότητα 128kbps. Για την επεξεργασία τους και κυρίως για την περικοπή (crop) χρησιμοποιήθηκε η online εφαρμογή επεξεργασίας ήχου **Twisted Wave**.

Όλη η παραγωγή του ήχου του παιχνιδιού έγινε με ερασιτεχνικό τρόπο, καθώς ο συντάκτης που είναι και μοναδικός developer του παιχνιδιού, δεν διαθέτει τις εξειδικευμένες γνώσεις που έχει ένας τεχνικός ήχου, ούτε τα κατάλληλα μέσα. Επίσης, η καλύτερη ηχοληψία θα απαιτούσε περισσότερο χρόνο και ξεφεύγει από τον σκοπό δημιουργίας αυτού του παιχνιδιού.

Το παιχνίδι διαθέτει και μουσική υπόκρουση. Στην οθόνη εισαγωγής ακούγεται το μουσικό κομμάτι του **"Suite - Badinerie"** του Γιόχαν Σεμπάστιαν Μπαχ σε μοντέρνα έκδοση, ενώ κατά τη διάρκεια του κανονικού παιχνιδιού ακούγεται μία επίσης μοντέρνα έκδοση του Ουγγρικού Χορού νούμερο 5 του Γιόχαν Μπραμς. Και τα δύο μουσικά κομμάτια είναι δωρεάν κατεβασμένα από την ιστοσελίδα της **Pixabay** (Pixabay GmbH, 2023) σε μορφή Mp3, και το κείμενο με τα δικαιώματα χρήσης βρίσκεται μέσα στα αρχεία του παιχνιδιού. Στο μενού των ρυθμίσεων του παιχνιδιού υπάρχει επιλογή για την ένταση της μουσικής.

5. Τεχνική αξιολόγηση του προτύπου και του παιχνιδιού

5.1. Περιορισμοί και προαπαιτούμενα

Όπως κάθε 3D παιχνίδι έτσι και το "Επίσκεψη στο μουσείο" για να καταφέρει να προσφέρει μία ομαλή και ευχάριστη εμπειρία στον παίκτη, το σύστημα στο οποίο εκτελείται πρέπει να ξεπερνάει κάποιους περιορισμούς (**Koulaxidis and Xinogalos, 2022**). Η έκδοση του παιχνιδιού στο πλαίσιο της διπλωματικής εργασίας είναι αποκλειστικά για υπολογιστές με Windows, πλην όμως με μικρές τροποποιήσεις στο πρότυπο, όπως ο χειρισμός με οθόνες αφής, θα μπορούσε να εξαχθεί (export) από την Unity εκτελέσιμο λογισμικό για tablets και κινητά, αλλά και σε παιχνιδιομηχανές.

Για να τρέξει το παιχνίδι σε ένα υπολογιστή απαιτούνται οι εξής ελάχιστες απαιτήσεις:

- Λειτουργικό σύστημα windows 7 ή νεότερο.
- Ελεύθερος χώρος στον σκληρό δίσκο τουλάχιστον 1Gb.
- Μνήμη RAM 2Gb, την οποία μεγάλο μέρος δεσμεύει το λειτουργικό σύστημα και περίπου το 0,5 Gb καταναλώνεται για τις ανάγκες του παιχνιδιού.
- Εγκατεστημένο το DirectX 9 ή νεότερη έκδοση.
- Κάρτα γραφικών με υποστήριξη WDDM (Windows Display Driver Model) 1.0.

Βέβαια συστήνεται η εγκατάσταση του παιχνιδιού σε υπολογιστές με καλύτερα χαρακτηριστικά για μεγαλύτερες αποδόσεις κατά την εκτέλεσή του, δηλαδή περισσότερα **fps**, καλύτερα γραφικά και χωρίς κωλύματα.

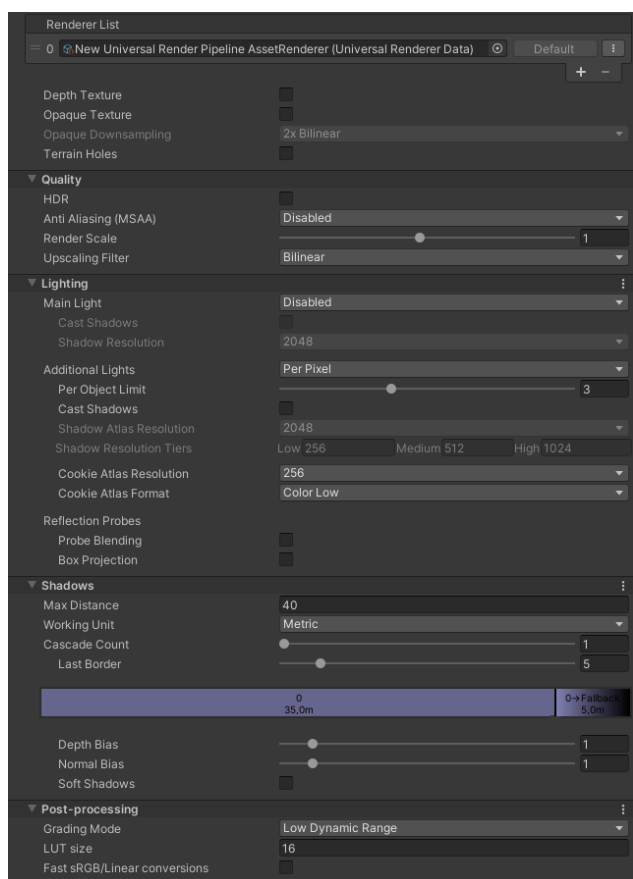
Πριν την σχεδίαση του παιχνιδιού, η οποία στηρίχθηκε στο πρότυπο για 3D point and click adventure παιχνιδιών που αναπτύχθηκε, τέθηκε ο στόχος να τρέχει ομαλά σε ένα μεγάλο εύρος υπολογιστών, από παλιούς με χαμηλές επιδόσεις μέχρι καινούργιους υπολογιστές που προορίζονται για παιχνίδια με πολύ υψηλές επιδόσεις. Ο στόχος αυτός επιτεύχθηκε με τρεις τρόπους. Πρώτον, με την προσθήκη επιλογών από τον χρήστη για την ποιότητα των γραφικών και της ανάλυσης, ώστε να προσαρμόζονται στις δυνατότητες του υπολογιστή που εκτελείται το παιχνίδι, ενώ ταυτόχρονα προσφέρεται ομαλή εμπειρία στον παίκτη κατά τον χειρισμό του, με γρήγορες ανταποκρίσεις στις ενέργειες του, αλλά και στην κίνηση των γραφικών του. Κατά δεύτερον με την εφαρμογή αρχών στη σχεδίαση και στην ανάπτυξη του λογισμικού. Και κατά τρίτων με την εφαρμογή μεθόδων βελτιστοποίησης (**Optimization techniques**) κατά την σχεδίαση και τη δημιουργία των σκηνών του παιχνιδιού στο γραφικό περιβάλλον της Unity. Υιοθετώντας από την αρχή αυτές τις αρχές και τεχνικές επιτεύχθηκε η βελτίωση των επιδόσεων του παιχνιδιού, αναφορικά με την χρήση του κεντρικού επεξεργαστή (**CPU usage**), την απεικόνιση των

αντικειμένων (**rendering**) και τη χρήση του επεξεργαστή της κάρτας γραφικών (**GPU usage**) καθώς και τη χρήση της μνήμης (**memory usage**).

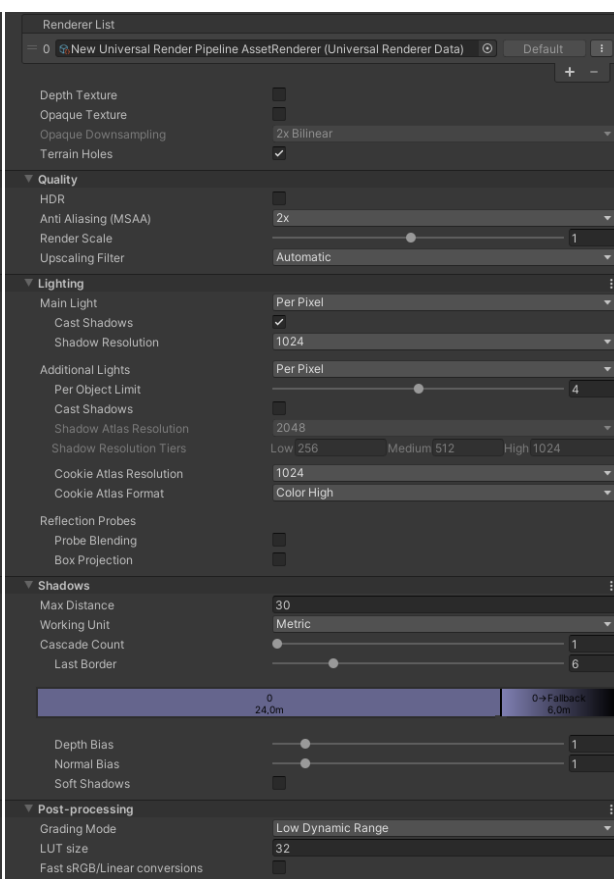
5.2. Τεχνικές Βελτιστοποίησης

5.2.1. Επιλογή ποιότητας γραφικών και ανάλυσης

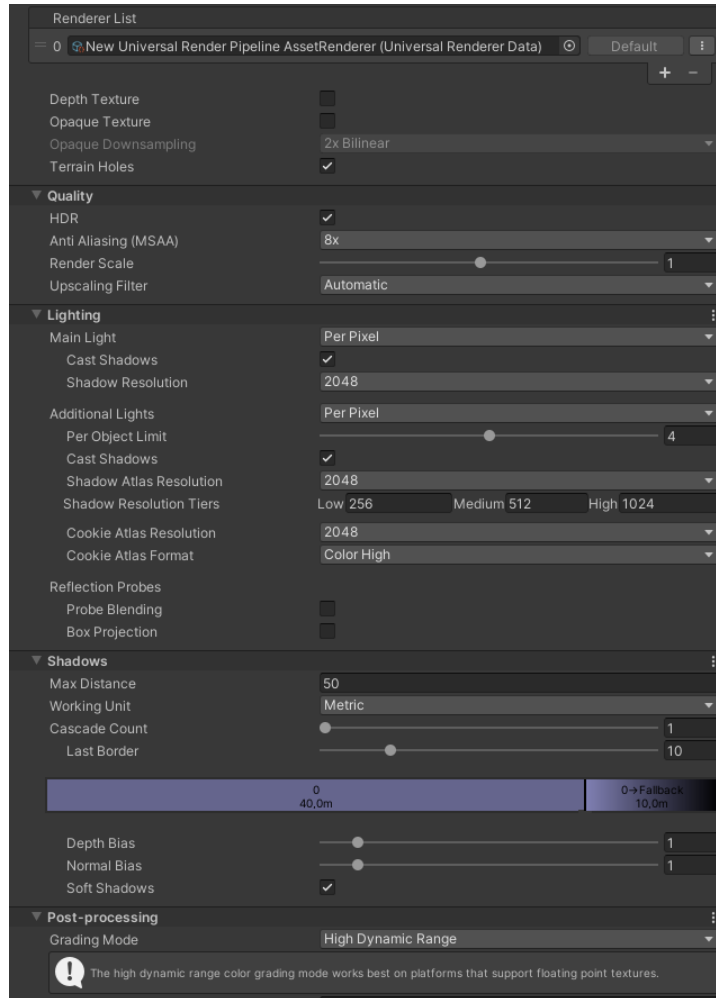
Όπως έχει ήδη αναφερθεί στο κεντρικό μενού του παιχνιδιού, ανάμεσα στις ρυθμίσεις βρίσκονται και οι επιλογές της ποιότητας των γραφικών και της ανάλυσης. Η επιλογή μεταξύ χαμηλής, μέτριας και υψηλής ποιότητας γραφικών, καθορίζει ποιο από τα τρία Universal Render Pipeline Asset που έχουν δημιουργηθεί (**LowURP** - **MediumURP** - **HighURP**) θα χρησιμοποιηθεί για την απεικόνιση. Το καθένα έχει τις δικές του αποθηκευμένες ρυθμίσεις οι οποίες εμφανίζονται στις Εικόνες 5.1, 5.2 και 5.3.



Εικόνα 5.1: Ιδιότητες LowURP



Εικόνα 5.2: Ιδιότητες MediumURP

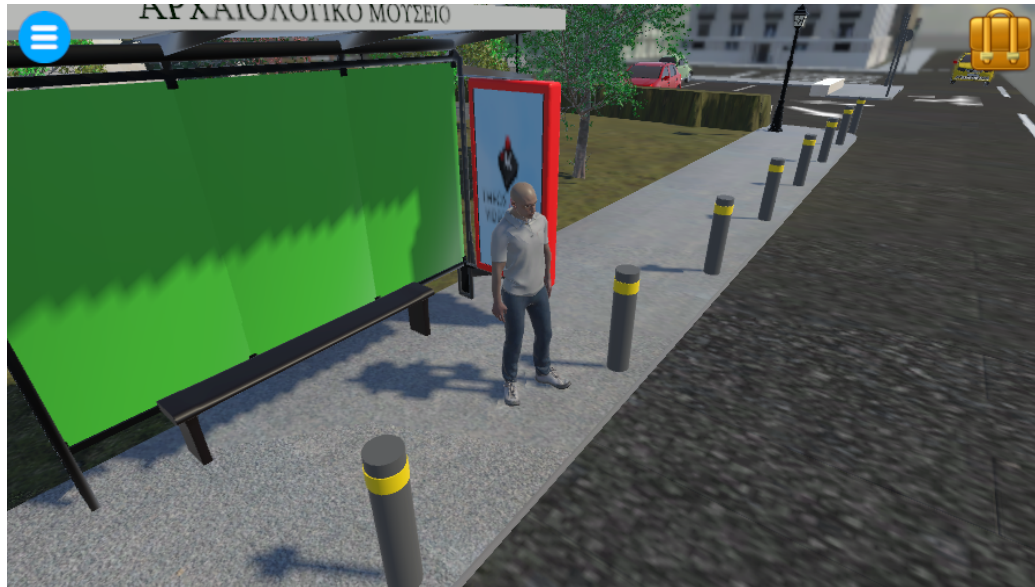


Εικόνα 5.3: Ιδιότητες HighURP

Τα αποτελέσματα των διαφορετικών ρυθμίσεων της ποιότητας των γραφικών, φαίνονται στις **Εικόνες 5.4, 5.5, 5.6**.



Εικόνα 5.4: Στιγμιότυπο με την χαμηλή ρύθμιση ποιότητας γραφικών



Εικόνα 5.5: Στιγμιότυπο με την μέτρια ρύθμιση ποιότητας γραφικών



Εικόνα 5.6: Στιγμιότυπο με την υψηλή ρύθμιση ποιότητας γραφικών

Η επιλογή της ανάλυσης του παιχνιδιού γίνεται ανάμεσα από αναλύσεις που υποστηρίζονται από το λειτουργικό σύστημα και από την οθόνη του χρήστη. Αυτές οι αναλύσεις πρέπει να έχουν μία αναλογία (**aspect**) **16:9** ή **16:10** για να εμφανιστούν στην επιλογή, ενώ ανιχνεύονται αυτόματα με την έναρξη του παιχνιδιού. Παραδείγματα με αναλύσεις που υποστηρίζονται βρίσκονται στην **Εικόνα 5.7**.



Εικόνα 5.7: Ρύθμιση ανάλυσης της οθόνης

Οι υψηλότερες αναλύσεις και η καλύτερη ποιότητα γραφικών απαιτούν περισσότερους υπολογιστικούς πόρους από το σύστημα. Αν ο χρήστης δεν διαθέτει υπολογιστή με δυνατά χαρακτηριστικά, προκειμένου το παιχνίδι να τρέχει ομαλά χωρίς κωλύματα, πρέπει να επιλέξει χαμηλότερες αναλύσεις και χαμηλής ποιότητας γραφικά. Ο συνδυασμός αυτών των ρυθμίσεων και της υπολογιστικής ισχύος του υπολογιστή καθορίζει τα καρτέ ανά δευτερόλεπτο (**fps**) του παιχνιδιού και κατ' επέκταση την ομαλή εμπειρία του παίκτη.

5.2.2. Αρχές σχεδίασης και ανάπτυξης του λογισμικού για βελτιστοποίηση

Οι εφαρμογές αρχών σχεδίασης στον αντικειμενοστρεφή προγραμματισμό καθώς και η υιοθέτηση προτύπων σχεδίασης (design patterns) απέδωσε στο πρότυπο οφέλη όχι μόνο στην οργάνωση, συντήρηση και ευελιξία του λογισμικού αλλά και στις επιδόσεις του, αυξάνοντας την αποτελεσματικότητα. Αυτό επιτεύχθηκε έμμεσα, αφενός με την μείωση της διπλοεγγραφής ίδιου κομματιού κώδικα (code duplication), αλλά και με την αποσύνθεση του συστήματος σε τμήματα κώδικα που κάνουν ξεχωριστές λειτουργίες το καθένα. Σε αυτά τα κομμάτια κώδικα, επειδή εφαρμόστηκαν οι αρχές, ήταν δυνατόν να ανιχνευτούν ευκολότερα προβλήματα και καθυστερήσεις και να γίνουν αλλαγές βελτιστοποίησης, χωρίς να χρειαστεί να αλλάξει το υπόλοιπο πρόγραμμα.

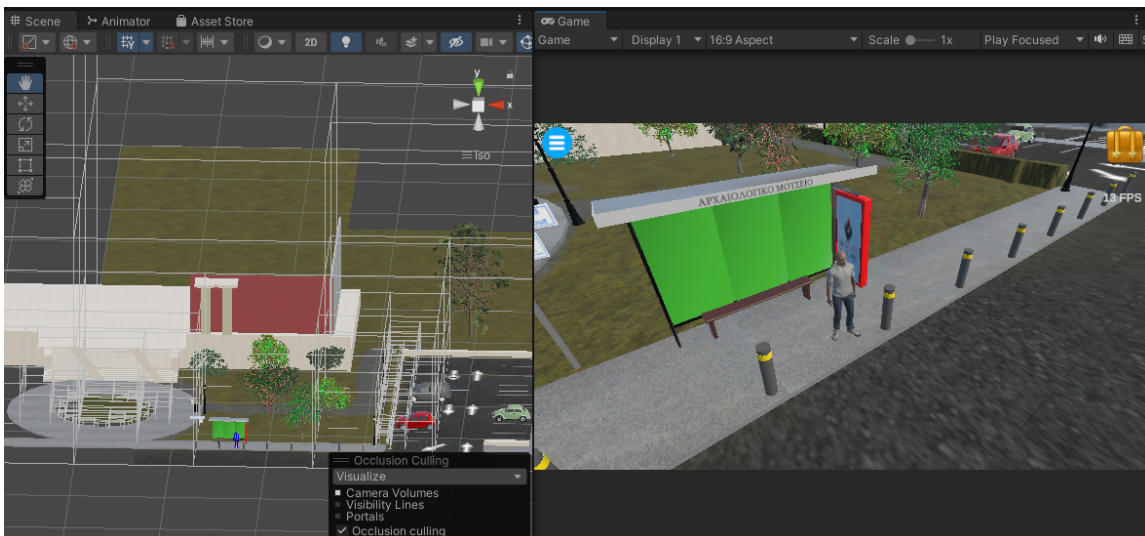
5.2.3. Τεχνικές Βελτιστοποίησης κατά την δημιουργία σκηνών στο γραφικό περιβάλλον της Unity

5.2.3.1. Μοντέλα με λιγότερα πολύγωνα

Η εξωτερική σκηνή του μουσείου είναι μια μεγάλη σε μέγεθος σκηνή που αποτελείται από πολλά 3D αντικείμενα. Κάθε 3D μοντέλο αντικειμένου αποτελείται από τρίγωνα (triangles), τα οποία αποτελούνται από τρεις κορυφές που είναι σημεία στον τρισδιάστατο χώρο (verts). Ενωμένα τρίγωνα δημιουργούν πολύγωνα (polygons) τα οποία όταν ενώνονται μεταξύ τους συνθέτουν το σχήμα του σαν πλέγμα (mesh). Περισσότερα πολύγωνα προσδίδουν μεγαλύτερη λεπτομέρεια στην απεικόνισή του και ρεαλιστικότητα, αλλά κοστίζουν σε υπολογιστική ισχύ κατά την διάρκεια των υπολογισμών της απεικόνισης (rendering). Στην αναζήτηση 3D μοντέλων για το γέμισμα αυτής της σκηνής, όπως αυτοκίνητα, δέντρα, κολώνες κλπ προτιμήθηκαν και χρησιμοποιήθηκαν μοντέλα με λιγότερα πολύγωνα (Low Poly).

5.2.3.2. Occlusion Culling

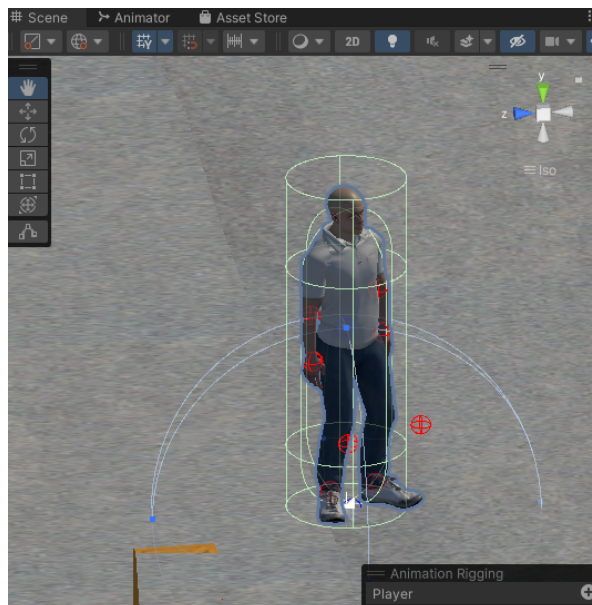
Στην εξωτερική σκηνή χρησιμοποιήθηκε το ενσωματωμένο στην Unity εργαλείο **Occlusion Culling**. Με τη χρήση αυτού του εργαλείου τα αντικείμενα που δεν απεικονίζονται στην κάμερα, παύουν να υπολογίζονται κατά την απεικόνιση εξοικονομώντας πόρους της κάρτας γραφικών. Παράδειγμα εφαρμογής Occlusion Culling φαίνεται στην Εικόνα 5.8, όπου στο scene view της Unity δεν υπολογίζονται και δεν εμφανίζονται τα αντικείμενα που δεν είναι στο πλάνο της ενεργής κάμερας (game view).



Εικόνα 5.8: Occlusion Culling στην εξωτερική σκηνή του μουσείου

5.2.3.3. Περιορισμός στην ανίχνευση σύγκρουσης (collision detection)

Στις δύο σκηνές του παιχνιδιού, υπάρχουν πολλά αντικείμενα που όταν δημιουργούνται ή όταν εισάγονται στην σκηνή περιβάλλονται από γεωμετρικούς χώρους γύρω τους για την ανίχνευση των συγκρούσεων που ονομάζονται **colliders**. Ο κάθε χώρος γύρω από ένα αντικείμενο μπορεί να έχει το σχήμα του αντικειμένου (mesh collider) ή να είναι ένα απλό γεωμετρικό σχήμα όπως ένα παραλληλεπίπεδο ή ένα κύλινδρος (box - Cylinder collider). Οι ανιχνεύσεις αφορούν κυρίως δύο συγκρούσεις. Η μία είναι η σύγκρουση του χαρακτήρα με διάφορα αντικείμενα του περιβάλλοντος κατά την κίνησή του. Για παράδειγμα, το έδαφος πρέπει να έχει σαν component έναν collider για να μπορεί να κινείται πάνω του ο χαρακτήρας. Μία κολώνα πρέπει να έχει collider για να μην περνάει από μέσα της, αλλά να την αναγνωρίζει σαν εμπόδιο. Και κατά κύριο λόγο ο ίδιος ο χαρακτήρας περιβάλλεται από ένα κυλινδρικό collider για να ανιχνεύεται η σύγκρουσή του με τα άλλα αντικείμενα, όπως φαίνεται στην Εικόνα 5.9. Ο άλλος τύπος σύγκρουσης είναι αυτός του δείκτη του ποντικιού με τα αντικείμενα. Για να ανιχνευτεί πάνω σε ποιο αντικείμενο βρίσκεται ο δείκτης, στέλνεται μία ακτίνα (Raycast) κάθετα από το σημείο που βρίσκεται ο δείκτης προς την σκηνή. Όταν αυτή η ακτίνα συγκρουστεί με τον collider ενός αντικειμένου, τότε ο κώδικας του προτύπου αναγνωρίζει το αντικείμενο και το σημείο πάνω στο οποίο βρίσκεται ο δείκτης.



Εικόνα 5.9: Ο collider σε σχήμα κυλίνδρου γύρω από τον κεντρικό χαρακτήρα

Στη σχεδίαση του παιχνιδιού χρησιμοποιήθηκαν απλοί γεωμετρικοί colliders αντί για τους σχηματικούς (mesh collider) καθώς οι τελευταίοι ξοδεύουν περισσότερους πόρους της CPU για τον υπολογισμό τους και η λεπτομέρεια που

προσφέρουν στην ανίχνευση της σύγκρουσης κρίθηκε περιττή. Επίσης, στα αντικείμενα της σκηνής που δεν βρίσκονται σε επιτρεπτές περιοχές κίνησης του χαρακτήρα και δεν είναι διαδραστικά, αφαιρέθηκε τελείως το component collider.

5.3. Μετρήσεις (benchmark) και αξιολόγηση αποδόσεων

Για την μέτρηση της απόδοσης του παιχνιδιού (game performance) χρησιμοποιήθηκε το ενσωματωμένο στην Unity εργαλείο **Statistics** στο παράθυρο Game View. Η μέτρηση των υπολογιστικών πόρων που χρησιμοποιεί το παιχνίδι κατά την εκτέλεσή του έγινε από το εργαλείο του λειτουργικού των windows 10, που λέγεται **Διαχείριση Εργασιών** και εμφανίζεται με το γνωστό **Ctrl + Alt + Del**. Το Unity διαθέτει, επίσης, ένα πολύπλοκο εργαλείο μετρήσεων, το **profiler**, που μετράει και αναλύει ξεχωριστά και με κάθε λεπτομέρεια όλες τις επιμέρους λειτουργίες του παιχνιδιού και τις χρήσεις των πόρων του συστήματος. Όμως, η παρουσίαση των μετρήσεων μέσω profiler ξεφεύγει από το πλαίσιο αυτής της εργασίας και προτιμήθηκε η χρήση των παραπάνω απλών εργαλείων.

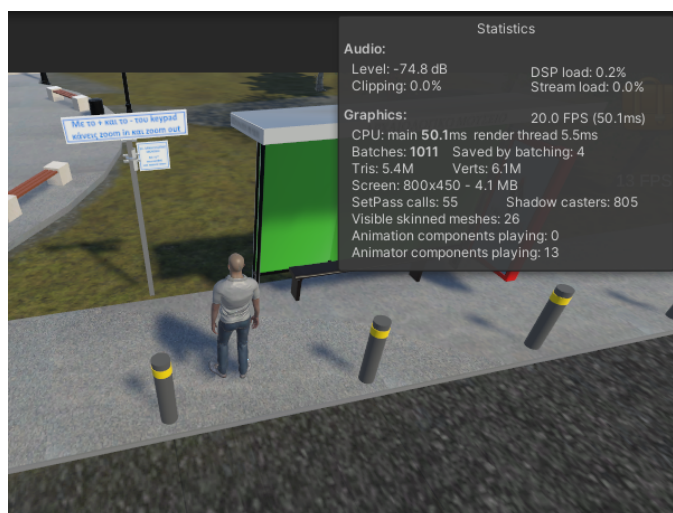
Όλες οι προηγούμενες τεχνικές βελτιστοποίησης εφαρμόστηκαν από την αρχή της σχεδίασης του προτύπου και του παιχνιδιού. Κατά συνέπεια δεν υπάρχει έκδοση χωρίς αυτές τις βελτιστοποιήσεις για να γίνει σύγκριση και μέτρηση αποκλειστικά της δικής τους απόδοσης.

Η σημαντικότερη μέτρηση που χρησιμοποιείται για τη μέτρηση της ομαλότητας της εμπειρίας του χρήστη είναι η μέτρηση των καρτέ ανα δευτερόλεπτο (Frames per second - **FPS**). Σαν στόχος στο πρότυπο έχουν οριστεί τα 60 fps που σημαίνει ότι όταν η υπολογιστική ισχύς το επιτρέπει τα καρτέ είναι σταθερά στα 60 fps και δεν ανεβαίνουν παραπάνω. Έχει καθιερωθεί στα παιχνίδια υπολογιστή, τα ελάχιστα επιτρεπτά fps να φτάνουν τα 30 fps. Κάτω από αυτό το όριο, το παιχνίδι παύει να προσφέρει ομαλή εμπειρία στον παίκτη.

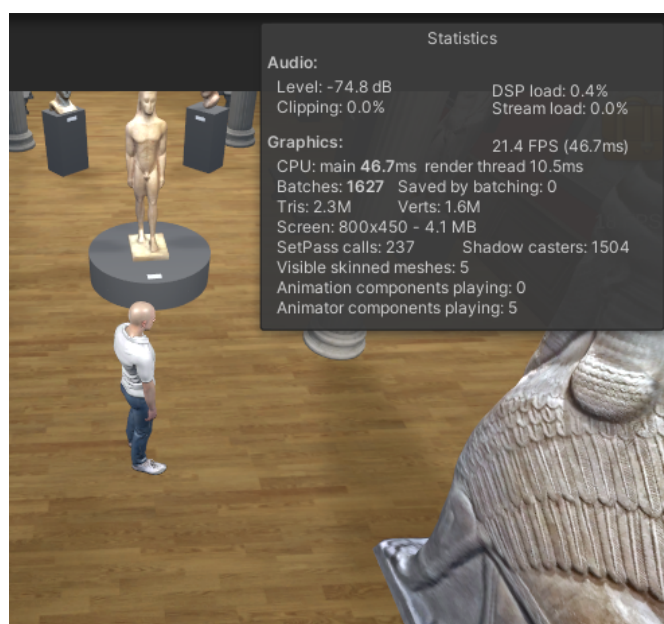
Αρχικά με το **statistics** μετρήθηκαν διάφορα χαρακτηριστικά της απόδοσης του παιχνιδιού κατά την ώρα εκτέλεσής του (runtime) με ανοιχτό τον editor. Οι μετρήσεις αντλήθηκαν από την εκτέλεση του παιχνιδιού στον φορητό υπολογιστή όπου έγινε και η ανάπτυξη του, ο οποίος έχει χαρακτηριστικά:

LENOVO G580, Επεξεργαστής Intel(R) Core(TM) i5-3210M 2.50GHz, 8GB RAM και κάρτα γραφικών NVIDIA GeForce 610M με έτος κατασκευής το 2012. Θεωρείται παλιός υπολογιστής με όχι καλά χαρακτηριστικά.

Έγιναν δύο μετρήσεις σε δύο διαφορετικά σημεία του παιχνιδιού. Το πρώτο σημείο είναι η αρχή του κυρίως παιχνιδιού δηλαδή η εξωτερική σκηνή όπου ο παίκτης βρίσκεται μπροστά στην στάση και το δεύτερο είναι εσωτερικά του μουσείου. Η κάμερα ήταν ρυθμισμένη στην αρχική (default) απόσταση της από τον χαρακτήρα, η ρύθμιση της ποιότητας των γραφικών ήταν στην υψηλή ενώ η ανάλυση του παραθύρου ήταν **800x450**. Οι μετρήσεις παρουσιάζονται στις Εικόνες 5.10, 5.11.



Εικόνα 5.10: Μετρήσεις με το εργαλείο Statistics στην εξωτερική σκηνή

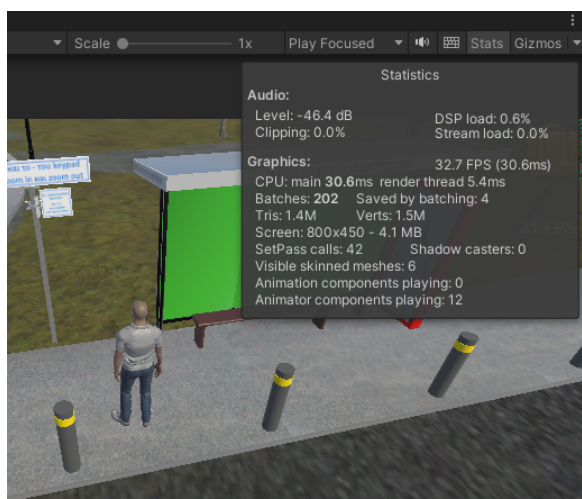


Εικόνα 5.11: Μετρήσεις με το εργαλείο Statistics στην εσωτερική σκηνή

Από τις δύο μετρήσεις συμπεραίνουμε ότι η εξωτερική σκηνή είναι ιδιαίτερα επιβαρυσμένη λόγω του μεγέθους της, με τον αριθμό των τριγώνων που την συνθέτουν να φτάνουν τα 5,4 εκατομμύρια. Αυτό δικαιολογείται και από το μέγεθος και την πολυπλοκότητά της. Επίσης, υπάρχουν 12 κινούμενα αντικείμενα και 805

φωτοσκιάσεις. Η εσωτερική του μουσείου σκηνή έχει λιγότερα τρίγωνα (2,3 εκ.), anim(5) και φωτοσκιάσεις (1504). Η πολυπλοκότητα της πρώτης σκηνής έχει σαν αποτέλεσμα το χαμηλό fps (20).

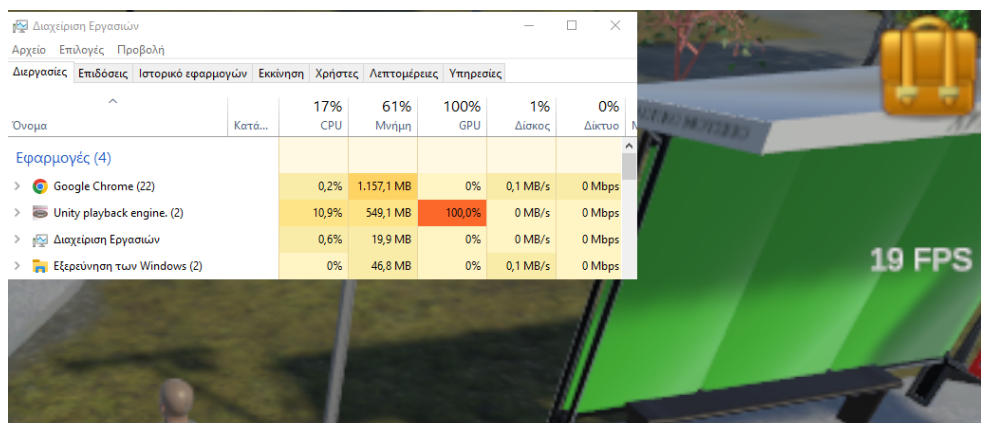
Μια νέα δοκιμή αυτή τη φορά με την χαμηλότερη ποιότητα γραφικών έδωσε τα αποτελέσματα της Εικόνας 5.12.



Εικόνα 5.12: Μετρήσεις με το εργαλείο Statistics σε χαμηλή ποιότητα γραφικών

Τα τρίγωνα μειώθηκαν σε μόλις 1,4 εκ από 5,4 εκ, οι φωτοσκιάσεις μηδενίστηκαν, ενώ τα fps αυξήθηκαν σε 32. Εύκολα διαπιστώνεται η σχέση ανταλλαγής ανάμεσα στην λεπτομέρεια - ρεαλιστικότητα των γραφικών με την ομαλότητα του παιχνιδιού.

Με τη **διαχείριση εργασιών** μετρήθηκαν οι πόροι που καταναλώνει το παιχνίδι στο ίδιο πρώτο σημείο με προηγούμενως και με την ποιότητα γραφικών στην ρύθμιση "υψηλή". Όμως, η εκτέλεση του παιχνιδιού έγινε σε ξεχωριστό παράθυρο, χωρίς την λειτουργία του editor της Unity. Τα fps εμφανίζονται πάνω δεξιά της οθόνης την ώρα της εκτέλεσης του παιχνιδιού, μέσω μιας κλάσης του κώδικα στο πρότυπο. Η μέτρηση έδειξε τα ακόλουθα αποτελέσματα ως προς την χρήση της CPU της RAM και της GPU (Εικόνα 5.13).



Εικόνα 5.13: Μετρήσεις με το εργαλείο διαχείριση εργασιών των Windows

Παρατηρείται ότι γίνεται μικρή χρήση της CPU, χρησιμοποιούνται 550 Mb της μνήμης RAM ενώ γίνεται αποκλειστική χρήση της GPU σε ποσοστό 100% για την εκτέλεση του παιχνιδιού. Τα fps είναι 19. Άρα, οι μικρές υπολογιστικές δυνατότητες της κάρτας γραφικών περιορίζουν την ομαλότητα του παιχνιδιού. Παρόμοια συμπεριφορά στην χρήση της GPU έχει κάθε μοντέρνο παιχνίδι με τρισδιάστατα γραφικά στα windows όταν προσπαθεί να ανεβάσει τα fps του.

Στη συνέχεια έγιναν μετρήσεις των fps στα δύο σημεία του παιχνιδιού που αναφέρθηκαν προηγουμένως σε οκτώ υπολογιστές με διαφορετικά χαρακτηριστικά. Στο κάθε υπολογιστή οι μετρήσεις έγιναν και με τις τρεις επιλογές γραφικών και σε τρεις αναλύσεις οθόνης σε πλήρη οθόνη (fullscreen). Τα χαρακτηριστικά των υπολογιστών και τα αποτελέσματα των μετρήσεων καταγράφηκαν στον Πίνακα 5.1.

1ος	Intel(R) Core(TM) i5-3210M 2.50GHz, 8GB RAM και κάρτα γραφικών NVIDIA GeForce 610M								
Επιλογή Γραφικών	Χαμηλή ποιότητα			Μέτρια ποιότητα			Υψηλή ποιότητα		
Ανάλυση (60Hz)	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080
fps στο σημείο A	58	30	17	29	19	11	20	14	9
fps στο σημείο B	60	60	30	60	60	30	30	19	9

2ος	AMD FX(tm)-8120 Eight-Core Processor 3.10 GHz, 8GB RAM και κάρτα γραφικών NVIDIA GeForce GTX 650 Ti BOOST								
Επιλογή Γραφικών	Χαμηλή ποιότητα			Μέτρια ποιότητα			Υψηλή ποιότητα		
Ανάλυση (144Hz)	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080
fps στο σημείο A	ΔΥ	60	60	ΔΥ	60	60	60	60	60
fps στο σημείο B	ΔΥ	60	60	ΔΥ	60	60	60	60	60

3ος	Amd Athlon(™) 5350 APU with Radeon (™) R3 2.05 GHz, 4GB RAM και κάρτα γραφικών AMD Radeon HD 8400 / R3 Series								
Επιλογή Γραφικών	Χαμηλή ποιότητα			Μέτρια ποιότητα			Υψηλή ΠΟΙΟΤΗΤΑ		
Ανάλυση (60Hz)	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080
fps στο σημείο A	55	35	ΔΥ	23	19	ΔΥ	14	12	ΔΥ
fps στο σημείο B	60	60	ΔΥ	60	56	ΔΥ	36	22	ΔΥ

4ος	Intel(R) Core(TM) i5-4570T CPU @ 2.90 GHz, 16GB RAM και κάρτα γραφικών Intel(R) HD Graphics 4600								
Επιλογή Γραφικών	Χαμηλή ποιότητα			Μέτρια ποιότητα			Υψηλή ποιότητα		
Ανάλυση (60Hz)	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080
fps στο σημείο A	60	60	30	44	30	20	30	20	11
fps στο σημείο B	60	60	60	60	60	30	60	60	20

5ος	Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz, 6GB RAM και κάρτα γραφικών NVIDIA GeForce GT540M								
Επιλογή Γραφικών	Χαμηλή ποιότητα			Μέτρια ποιότητα			Υψηλή ποιότητα		
Ανάλυση (60Hz)	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080
fps στο σημείο A	40	32	25	16	18	17	13	14	13
fps στο σημείο B	60	55	44	60	43	36	28	19	13

6ος	Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz, 16GB RAM και κάρτα γραφικών NVIDIA GeForce GTX 1660 Ti win Max-Q Design								
Επιλογή Γραφικών	Χαμηλή ποιότητα			Μέτρια ποιότητα			Υψηλή ποιότητα		
Ανάλυση (60Hz)	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080
fps στο σημείο A	60	60	60	60	60	60	60	60	60
fps στο σημείο B	60	60	60	60	60	60	60	60	60

7ος	Intel(R) Celeron(R) N4000 CPU @ 1.1GHz 1.10GHz, 4GB RAM και κάρτα γραφικών Intel ® UHD Graphics 600								
Επιλογή Γραφικών	Χαμηλή ποιότητα			Μέτρια ποιότητα			Υψηλή ποιότητα		
Ανάλυση (000Hz)	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080
fps στο σημείο A	25	20	9	13	10	8	9	8	2
fps στο σημείο B	37	20	14	20	22	9	22	12	5

8ος	AMD Ryzen 3 3250U with Radeon Graphics 2.60 GHz, 8 GB RAM και κάρτα γραφικών Integrated AMD Radeon R4 Graphics								
Επιλογή Γραφικών	Χαμηλή ποιότητα			Μέτρια ποιότητα			Υψηλή ποιότητα		
Ανάλυση (000Hz)	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080	640 x 400	1280 x 720	1980 x 1080
fps στο σημείο A	ΔΥ	60	49	ΔΥ	52	35	ΔΥ	30	19
fps στο σημείο B	ΔΥ	60	60	ΔΥ	60	60	ΔΥ	40	29

Πίνακας 5.1: Αποτελέσματα μετρήσεων των fps σε διαφορετικούς υπολογιστές

Όλοι οι υπολογιστές των δοκιμών διέθεταν σκληρό δίσκο SSD και δεν παρατηρήθηκαν καθυστερήσεις στο φόρτωμα των σκηνών. Το παιχνίδι φορτώνοταν σε μόλις λίγα δευτερόλεπτα.

5.4. Συμπεράσματα αξιολόγησης

Το παιχνίδι όντως τρέχει ικανοποιητικά σε ένα μεγάλο εύρος υπολογιστών με Windows. Προσφέρει την καλύτερη εμπειρία παιχνιδιού στους παίκτες με σύγχρονους υπολογιστές που διαθέτουν κάρτες γραφικών υψηλών αποδόσεων και συστήνεται να εκτελείται σε μηχανήματα με δυνατά χαρακτηριστικά. Στους παλαιότερους υπολογιστές με χαμηλές δυνατότητες, προκειμένου να επιτευχθεί η ομαλότητα και να έχει ικανοποιητικό αριθμό fps (>30), πρέπει να μειωθεί η ανάλυση της οθόνης και να ρυθμιστούν τα γραφικά στο χαμηλό επίπεδο. Σε γενικές γραμμές δεν διαφέρει από τους υπόλοιπους σύγχρονους 3D τίτλους του είδους που κυκλοφορούν στην αγορά.

6. Εφαρμογή του προτύπου για δημιουργία adventure παιχνιδιών

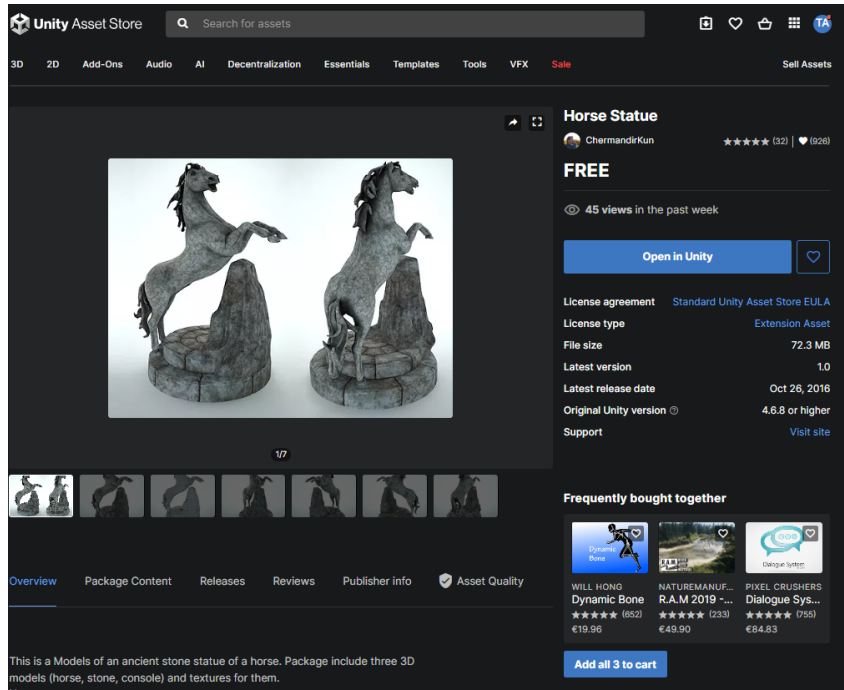
6.1. Εισαγωγή

Το πρότυπο που αναπτύχθηκε στο πλαίσιο αυτής της διπλωματικής αποτελείται από κώδικα σε γλώσσα C# και μηχανισμούς στη Unity που μπορούν να χρησιμοποιηθούν ως βάση για την ανάπτυξη ενός μοντέρνου 3D point and click adventure. Ο developer για να χρησιμοποιήσει τους περισσότερους μηχανισμούς δεν χρειάζεται να γράψει καθόλου κώδικα. Απλά πρέπει να ακολουθήσει κάποια βήματα στο γραφικό περιβάλλον της Unity και να προσθέσει τα δικά του γραφικά, animation και ήχους. Δυστυχώς, υπάρχουν κάποιοι μηχανισμοί, που δεν μπορούσαν να κατασκευαστούν στο πρότυπο εκ των προτέρων για όλα τα παιχνίδια adventure. Αυτοί σχετίζονται, κυρίως, με τις συγκεκριμένες αντιδράσεις των αντικειμένων όταν ενεργεί πάνω τους ο παίκτης. Για αυτούς τους μηχανισμούς πρέπει να γραφτεί κώδικας. Στις επόμενες δύο ενότητες θα εξεταστεί ο τρόπος που χρησιμοποιούνται οι μηχανισμοί του προτύπου, με δύο παραδείγματα.

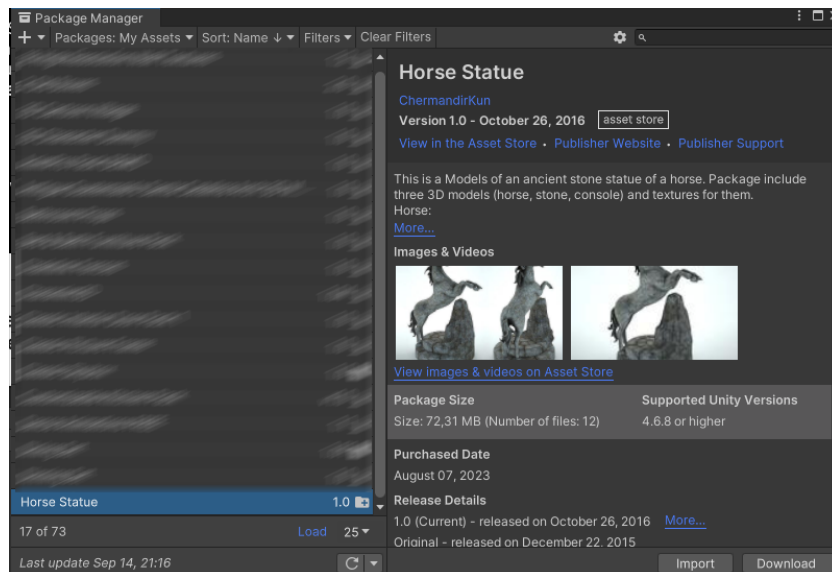
6.2. Προσθήκη διαδραστικού αντικειμένου σε υπάρχουσα σκηνή

Έστω ότι στην εσωτερική σκηνή του μουσείου θέλουμε να προσθέσουμε ένα νέο αντικείμενο έκθεμα, με ιδιότητες παρόμοιες με αυτές των υπολοίπων. Για να επιτευχθεί αυτό πρέπει να ακολουθήσουμε μια σειρά βημάτων στο γραφικό περιβάλλον της Unity.

Αρχικά, πρέπει να υπάρχει η 3D μορφή του αντικειμένου σε αρχείο. Το αντικείμενο μπορεί να δημιουργηθεί σε άλλο πρόγραμμα μοντελοποίησης, όπως το blender, ή να ληφθεί από το διαδίκτυο ή αν η μορφή του αποτελείται από απλά γεωμετρικά σχήματα, μπορεί να κατασκευαστεί εντός της Unity. Εδώ θα κατεβάσουμε ένα δωρεάν 3D αντικείμενο από το **asset store**. Κάνοντας αναζήτηση για ένα άγαλμα και βάζοντας σαν φίλτρο το free asset βρίσκουμε το άγαλμα ενός αλόγου με δημιουργό τον **ChermandirKun** (Εικόνα 6.1). Το κατεβάζουμε και το εισάγουμε στην σκηνή με την επιλογή **"Add to my Assets"**, **"Open in unity"**. Έπειτα στο παράθυρο package manager επιλέγουμε import και αυτό εμφανίζεται στο παράθυρο Project ανάμεσα στα άλλα Assets (Εικόνα 6.2).



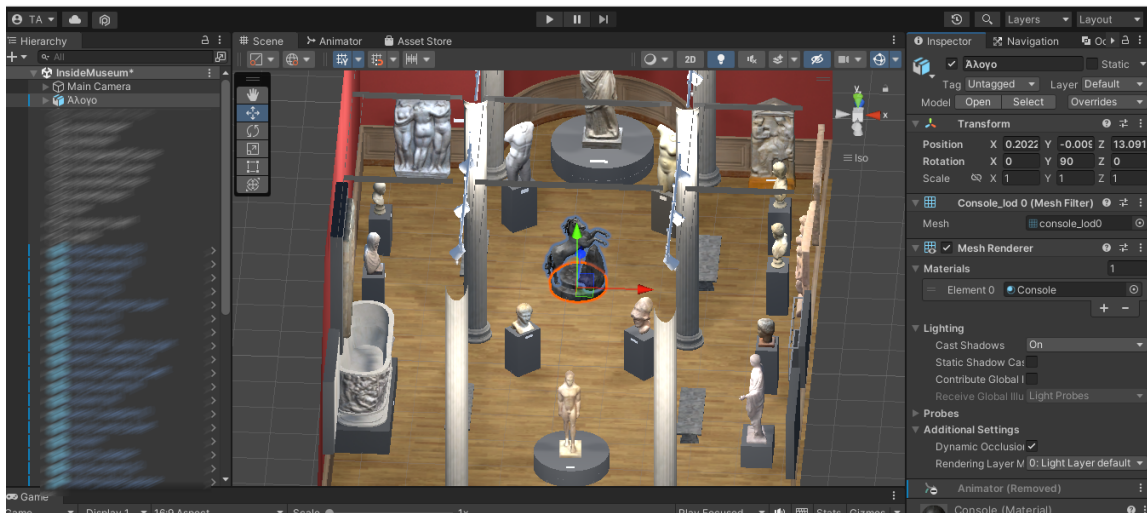
Εικόνα 6.1: Αναζήτηση αντικειμένου αγάλματος αλόγου στο Asset Store



Εικόνα 6.2: Εισαγωγή του αντικειμένου αγάλματος στο project

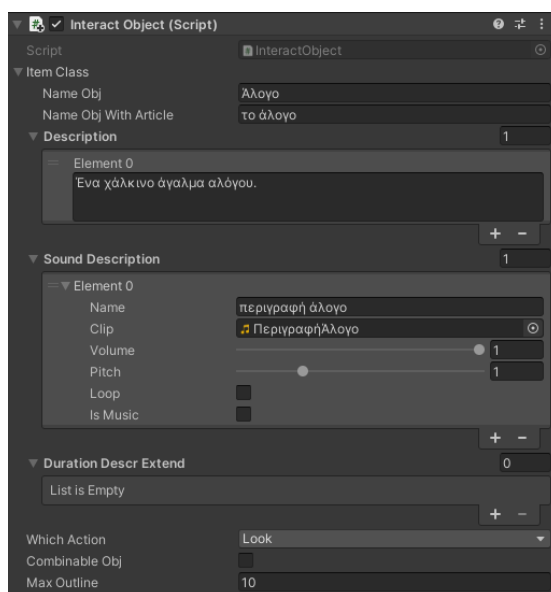
Από την καρτέλα project Επιλέγουμε το μοντέλο του αλόγου, και αλλάζουμε το **scale factor** σε **0.005** και materials location σε **"Use Embedded Materials"**, αφού βεβαιωθούμε ότι οι shaders των υλικών του υποστηρίζονται στο **URP**. Σε διαφορετική περίπτωση τα χρώματα έχουν μωβ χρώματα και θα πρέπει οι shaders του κάθε υλικού να αλλάξουν σε **URP/Lit**. Στη συνέχεια, εισάγουμε το μοντέλο στην σκηνή και στην καρτέλα Hierarchy αλλάζουμε το όνομά του σε Άλογο. Το επιλέγουμε και μέσω των ιδιοτήτων του component transform στο παράθυρο inspector, αλλάζουμε το μέγεθος την θέση και την γωνία του όπως φαίνεται στην Εικόνα 6.3. Η αλλαγή μπορεί να γίνει

και με τα χειριστήρια γύρω από το αντικείμενο μέσα στο παράθυρο Scene. Στην εκτέλεση του παιχνιδιού, το αντικείμενο που έχει προστεθεί θα είναι σαν μέρος του σκηνικού χωρίς διαδραστικότητα. Δεν θα είναι ούτε εμπόδιο στην κίνηση του χαρακτήρα, ο οποίος θα μπορεί να περάσει από μέσα του, καθώς η ρύθμιση στο **nav mesh surface** έχει οριστεί σε **Use Geometry: PhysicsColliders** και σε αυτό δεν έχει προστεθεί ακόμα **collider**.



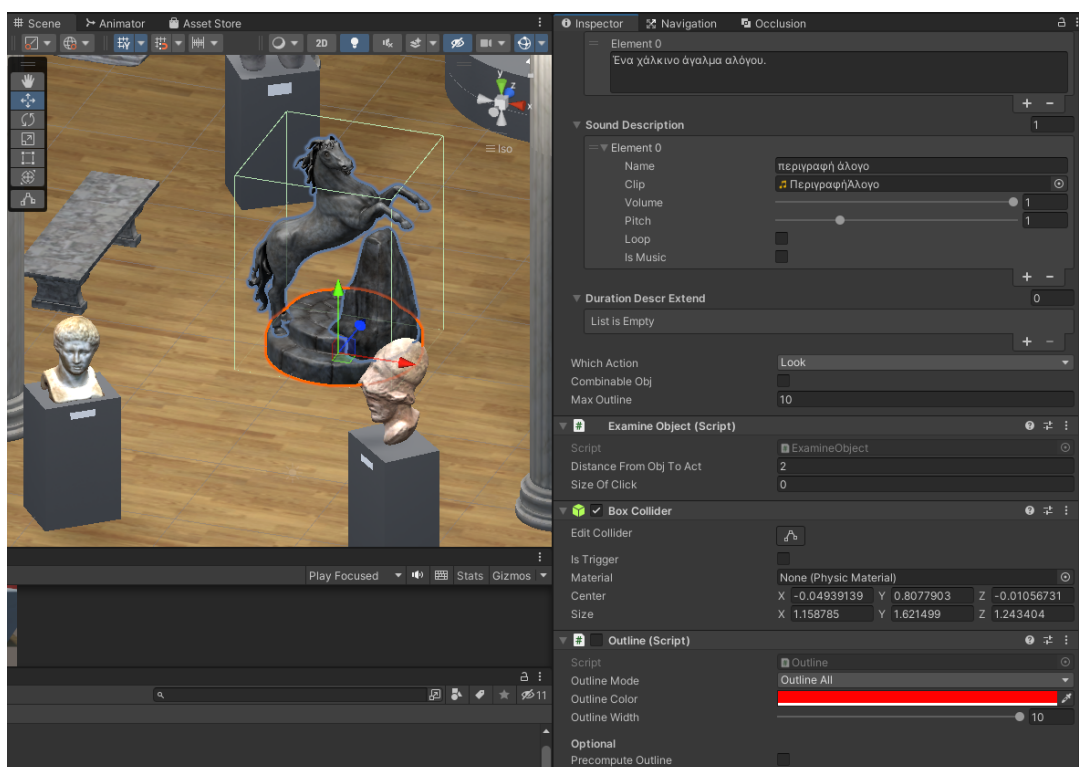
Εικόνα 6.3: Αλλαγή των ιδιοτήτων του αντικειμένου

Για την προσθήκη διαδραστικότητας επιλέγουμε το αντικείμενο και κάνουμε προσθήκη του component **InteractObject.cs**. Στις ιδιότητες του δίνουμε όνομα, όνομα με άρθρο, περιγραφή που αποτελείται από μία ή παραπάνω φράσεις, τα ηχητικά της περιγραφής που έχουμε εισάγει στο project από πριν και την ενέργεια που θα γίνει με αριστερό κλικ πάνω του (Εικόνα 6.4).



Εικόνα 6.4: Ο inspector του αντικειμένου με προσθήκη της κλάσης InteractObject και των ιδιοτήτων της

Επειδή η ενέργεια που επιλέχθηκε είναι η **look**, προσθέτουμε το component **ExamineObject.cs** και δίνουμε στην ιδιότητα **distanceFromObjectToAct** σε **2** που σημαίνει η εξέταση του αντικειμένου θα γίνει σε απόσταση το πολύ 2 μέτρα από το κέντρο του. Προσθέτουμε ακόμα το component **outline** με την ιδιότητα **color outline** σε κόκκινο και το απενεργοποιούμε. Τέλος, προσθέτουμε το component **Box Collider** και πατώντας **Edit Collider** διαμορφώνουμε τον παραλληλεπίπεδο χώρο, μέσα στον οποίο θα ενεργοποιείται το αντικείμενο όταν ο δείκτης του ποντικιού εισέλθει σε αυτόν. Επίσης, με τον collider δεν θα μπορεί ο χαρακτήρας να περνάει πλέον από μέσα σου, αλλά θα το αναγνωρίζει ως εμπόδιο. Προσθέτουμε επίσης κάτω στην καρτέλα Hierarchy ένα καινούργιο αντικείμενο μέσα στο αντικείμενο του αλόγου σαν τέκνο (child object), το οποίο θα ονομάσουμε AimToLook και το τοποθετούμε λίγο πάνω από το κέντρο του αλόγου. Αλλάζουμε την ετικέτα του (tag) στις ιδιότητες του σε AimToLook. Με αυτό τον τρόπο, ο χαρακτήρας θα κοιτάει στο κέντρο του αλόγου και όχι στην βάση του όταν επιλέγεται. Όλες οι τελευταίες ιδιότητες φαίνονται στην Εικόνα 6.5.



Εικόνα 6.5: Σύνολο των ιδιοτήτων του αντικειμένου στον inspector

Αν δοκιμάσουμε το παιχνίδι διαπιστώνουμε ότι πλέον το αντικείμενο έχει γίνει διαδραστικό. Αν ο δείκτης κυλήσει επάνω στο άλογο, ο χαρακτήρας στρέφει το κεφάλι του προς αυτό, ο δείκτης αλλάζει σε μεγεθυντικό φακό, εμφανίζεται κόκκινο περίγραμμα και το κείμενο **“κοίταξε το άλογο”** (Εικόνα 6.6). Αν γίνει αριστερό κλικ

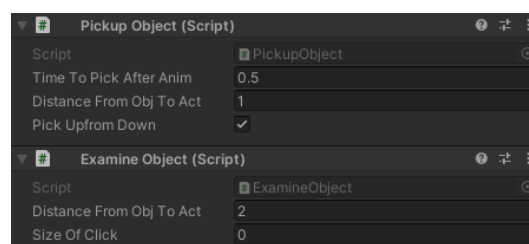
πάνω σε αυτό, ο χαρακτήρας θα το προσεγγίσει και σε απόσταση δύο μέτρων θα το περιγράψει.



Εικόνα 6.6: Στιγμιότυπο από το παιχνίδι, όταν ο δείκτης κινείται πάνω στο αντικείμενο

Υπάρχει η δυνατότητα η προσέγγιση να γίνει σε ένα διαφορετικό σημείο, προσθέτοντας ένα άδειο Game Object με ετικέτα **WayPoint** κάτω από το αντικείμενο στο παράθυρο των ιεραρχιών, και μεταφέροντας το στο συγκεκριμένο σημείο της σκηνής που θέλουμε. Τότε ο χαρακτήρας μετά το αριστερό κλικ στο αντικείμενο θα κινηθεί προς το σημείο αυτό, και όταν φτάσει στα δύο μέτρα θα στραφεί προς το άλογο και θα το περιγράψει.

Έστω ότι θέλουμε να αλλάξουμε την ενέργεια μετά το κλικ, και αντί για εξέταση να γίνεται συλλογή του αλόγου και προσθήκη στο inventory. Αλλάζουμε την ιδιότητα **whichAction** της **interactObject.cs** σε **pickup** και προσθέτουμε το component **PickUpObject**. Επιλέγουμε αν θα το πάρει από το έδαφος (αν θα σκύψει), την απόσταση και τον χρόνο που πρέπει να περάσει για να προστεθεί στο inventory όπως βλέπουμε στην Εικόνα 6.7. Στο φάκελο **Resources** στα assets πρέπει να προσθέσουμε δύο εικονίδια. Το ένα θα απεικονίζει πως θα εμφανίζεται το αντικείμενο στο inventory και το άλλο πως θα αλλάζει ο δείκτης του ποντικιού όταν επιλέγεται από το inventory. Η ονομασία πρέπει να είναι ίδια με το όνομα του αντικειμένου, στην περίπτωση αυτή "Άλογο".



Εικόνα 6.7: προσθήκη του component PickUpObject

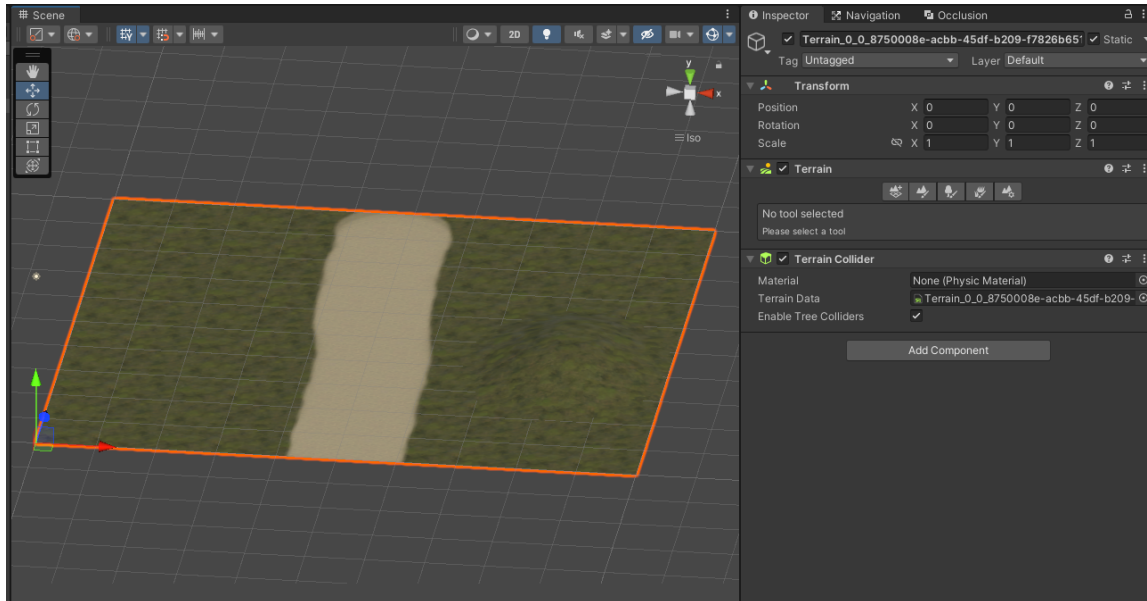
Στην δοκιμή αν γίνει αριστερό κλικ πάνω στο άλογο, ο χαρακτήρας θα σκύψει και θα σηκώσει το άλογο, το οποίο θα εξαφανιστεί μετά από μισό δευτερόλεπτο και θα προστεθεί στο inventory.

Αν θελήσουμε να προσθέσουμε παραπάνω ενέργειες όταν ο χαρακτήρας μαζεύει ένα αντικείμενο, εκτός από αυτές τις βασικές, όπως για παράδειγμα να πει κάτι, ή να αλλάξει κάτι στο περιβάλλον, θα πρέπει να γραφτεί μία καινούργια κλάση που θα κληρονομεί από την **PickUpObject.cs** και να περιέχει τον κώδικα για αυτές τις ενέργειες. Το ίδιο πρέπει να γίνει και όταν η ενέργεια που επιλέγεται στην **interactObject** είναι η **Use**. Πρέπει να γραφτεί και να προστεθεί στο αντικείμενο μία καινούργια κλάση, που να κληρονομεί από την UseObject και να περιγράφει τις ενέργειες που θα γίνουν όταν γίνει αριστερό κλικ πάνω στο αντικείμενο.

6.3. Προσθήκη σκηνής

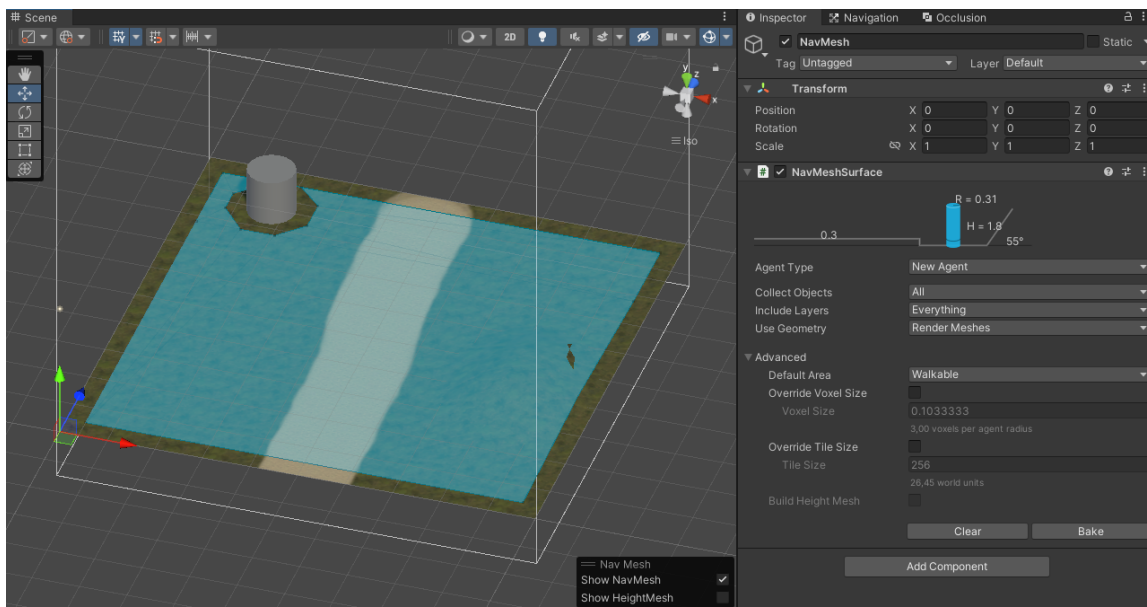
Θα δημιουργήσουμε μία σκηνή από την αρχή και θα την προσθέσουμε μέσα στο παιχνίδι. Η σκηνή αυτή θα είναι πλήρως λειτουργική, όπως οι δύο σκηνές που διαθέτει ήδη το παιχνίδι και θα φορτώνεται όταν ο κεντρικός χαρακτήρας ανοίξει την είσοδο του μουσείου και εισέλθει μέσα. Αρχικά θα δημιουργήσουμε μια νέα σκηνή με την επιλογή **Assets > Create > Scene** και θα την ονομάσουμε **DemoScene**.

Ανοίγουμε την σκηνή και με το εργαλείο **Terrain** δημιουργούμε μια νέα επιφάνεια **terrain**. Θα μπορούσαμε να δημιουργήσουμε ένα απλό επίπεδο από ένα απλό παραλληλεπίπεδο, αλλά αντί για αυτό, θα εκμεταλλευτούμε τις δυνατότητες που προσφέρει το εργαλείο **terrain**. Επιλέγουμε από το μενού **Window - terrain - terrain toolbox** και στο καινούργιο παράθυρο θα δώσουμε διαστάσεις **10x10** και θα πατήσουμε το κουμπί **Create New Terrain**. Θα δημιουργηθεί καινούργιο αντικείμενο στην σκηνή με όνομα **TerrainGroup_0**. Έπειτα θα το γεμίσουμε με γρασίδι και θα φτιάξουμε ένα μικρό λόφο στο κέντρο του. Στο παράθυρο του inspector αφού επιλέξουμε το αντικείμενο στη σκηνή που δημιουργήσουμε υπάρχει το tab **paint terrain**. Πατάμε στο **"Add Layer"** και για στρώση επιλέγουμε το **"Grass_Moss_TerrainLayer"** και όλη η επιφάνεια μετατρέπεται σε γρασίδι. Προσθέτουμε και ένα καινούργιο Layer για να φτιάξουμε ένα δρομάκι από άμμο ανάμεσα στο γρασίδι, επιλέγοντας **"Sand_TerrainLayer"**. Επιλέγουμε βούρτσα και μέγεθος και σχεδιάζουμε το δρομάκι σχεδιάζοντας στο παράθυρο **Scene**. Για να φτιάξουμε ένα λοφάκι αλλάζουμε την επιλογή στον inspector σε **Raise or Lower Terrain** και ρυθμίζοντας κατάλληλα το μέγεθος και την ένταση της βούρτσας, σχεδιάζουμε στο παράθυρο **Scene**. Η σκηνή που δημιουργήθηκε φαίνεται στην Εικόνα 6.8.



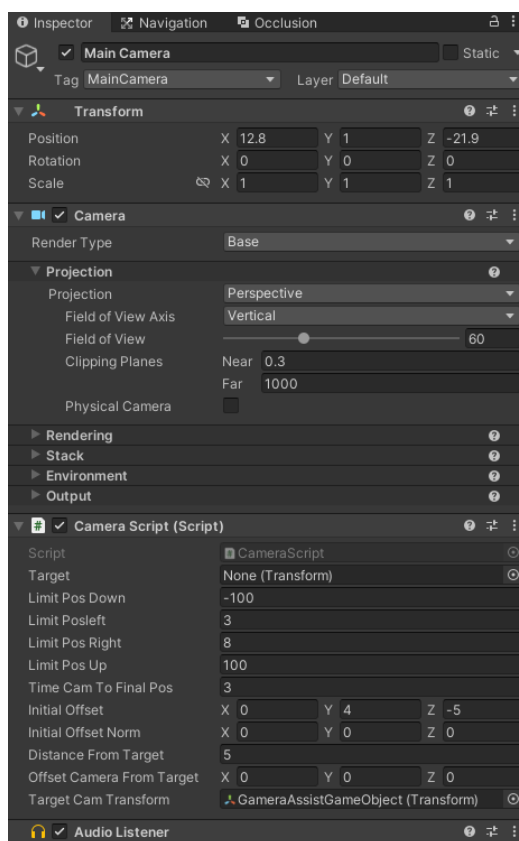
Εικόνα 6.8: Η σκηνή μετά την δημιουργία εδάφους με το εργαλείο terrain

Στην σκηνή θα δημιουργήσουμε και ένα αντικείμενο για εμπόδιο στην κίνηση του παίκτη, με την επιλογή **GameObject > 3DObject > Cylinder**. Για να ορίζουμε την επιφάνεια πάνω στην οποία θα μπορεί να κινείται ελεύθερα ο χαρακτήρας, δημιουργούμε ένα κενό GameObject και το ονομάζουμε NavMesh. Σε αυτό προσθέτουμε το component **NavMeshSurface** και στο παράθυρο inspector, αφού ορίσουμε τον τύπο του agent σε new agent, πατάμε στην επιλογή **"Bake"**. Η επιφάνεια **nav-mesh** που δημιουργήθηκε παρουσιάζεται στην Εικόνα 6.9.



Εικόνα 6.9: Η επιφάνεια nav-mesh στην σκηνή που υπάρχει ένα εμπόδιο

Στο αντικείμενο **MainCamera** της σκηνής, προσθέτουμε το Script Component **CameraScript.cs** και στην ιδιότητα του Initial Offset αλλάζουμε τις τιμές **X,Y,Z σε 0 , 4 , -5** και την τιμή της **DistanceFromTarget** σε 5, ώστε να ορίσουμε την απόσταση και της κατεύθυνσης της κάμερας από τον χαρακτήρα. Επίσης, ορίζουμε την χρονική καθυστέρηση της κάμερας βάζοντας στην ιδιότητα **timeCamToFinalPos** την τιμή 3. Για όρια της κάμερας ορίζουμε τις τιμές public float **limitPosDown, limitPosleft, limitPosRight** και **limitPosUp** σε -100, 3 , 8 , 100, ώστε όταν φτάνει στο αριστερό και δεξί άκρο να παραμένει η κάμερα στην ίδια θέση και να αλλάζει την γωνία της, πράγμα που έχει ωραίο οπτικό αποτέλεσμα. Οι ιδιότητες της **MainCamera** φαίνονται στην Εικόνα 6.10.

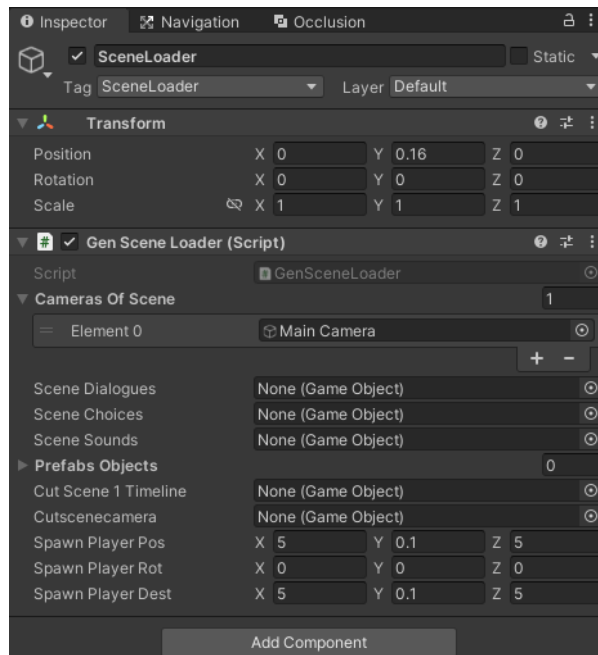


Εικόνα 6.10: Οι ιδιότητες του αντικείμενου της κάμερας στον inspector

Προσθέτουμε ένα κενό αντικείμενο και το ορίζουμε στην ιεραρχία κάτω από την κάμερα. Το ονομάζουμε **CameraAssistGameObject** και στις ιδιότητες της κάμερας, τοποθετούμε αυτό το αντικείμενο στην ιδιότητα **CameraAssistGameObject**

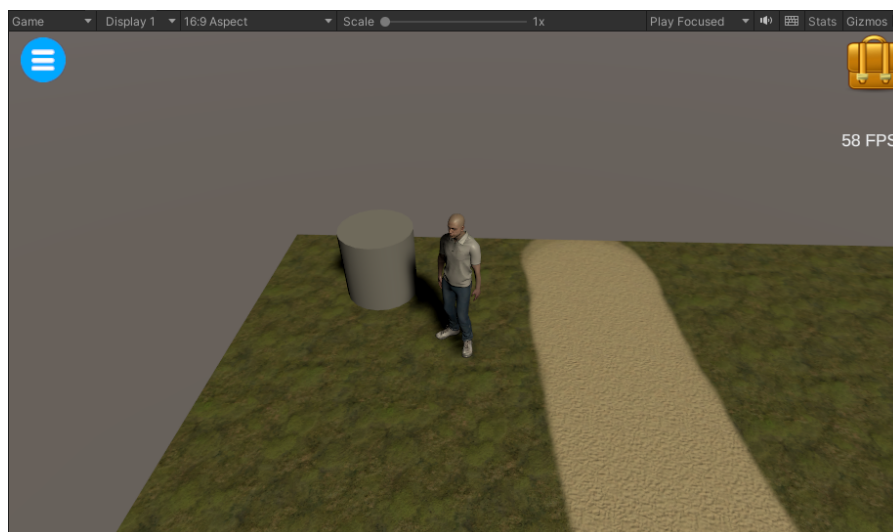
Προσθέτουμε ένα καινούργιο κενό αντικείμενο και το ονομάζουμε **SceneLoader** και του ορίζουμε την ετικέτα **sceneloader**. Σε αυτό προσθέτουμε σαν **component** την κλάση **GenSceneLoader** και στις ιδιότητες της προσθέτουμε την μοναδική

κάμερα της σκηνής, την θέση που θα εμφανιστεί ο χαρακτήρας, την γωνία του και την θέση που θα κινηθεί αρχικά όταν φορτώσει η σκηνή όπως στην Εικόνα 6.11.



Εικόνα 6.11: Οι ιδιότητες του αντικειμένου SceneLoader στον inspector

Για να φορτώσουμε την σκηνή, θα αλλάξουμε ιδιότητα στο component **ExitToOtherScene** του αντικειμένου **ΕίσοδοςΜουσείου** στην σκηνή εξωτερικά του μουσείου. Στην ιδιότητα **SceneNameToGo** βάζουμε την τιμή **"DemoScene"**, όπως ονομάσαμε την σκηνή που θέλουμε να φορτώσουμε. Παίζοντας το παιχνίδι, όταν εισέλθουμε στην είσοδο του μουσείου θα φορτωθεί η σκηνή που μόλις φτιάξαμε. Ο χαρακτήρας μπορεί να ελέγχεται κανονικά και να κινείται εντός του επιπέδου που φτιάξαμε, αποφεύγοντας το κυλινδρικό εμπόδιο. Το τελικό αποτέλεσμα παρουσιάζεται στην Εικόνα 6.12.



Εικόνα 6.12: Η DemoScene σκηνή κατά την διάρκεια του παιχνιδιού

7. Επίλογος

7.1. Σύνοψη και συμπεράσματα

Στο πλαίσιο της διπλωματικής εργασίας σχεδιάστηκε και αναπτύχθηκε στη μηχανή παιχνιδιών Unity και σε γλώσσα προγραμματισμού C# ένα πρότυπο για δημιουργία παιχνιδιών του είδους **“3d Point and click adventure”**. Στη συνέχεια, εφαρμόστηκε το πρότυπο για την ανάπτυξη του παιχνιδιού **“Επίσκεψη στο μουσείο”**.

Μέσω της παρουσίασης του παιχνιδιού αναλύθηκαν ο βασικός σχεδιασμός και οι περισσότεροι μηχανισμοί του προτύπου. Επίσης, περιγράφηκαν και οι τρόποι εφαρμογής του προτύπου σε ένα καινούργιο παιχνίδι 3D Point and Click adventure. Κάποιος τρίτος developer μπορεί εύκολα να χρησιμοποιήσει όλους αυτούς τους μηχανισμούς για την ανάπτυξη του δικού του adventure game.

Στην εργασία περιγράφηκαν κάποιες από τις αρχές σχεδίασης ποιοτικού λογισμικού που εφαρμόστηκαν στο πρότυπο, καθώς και κάποιες τεχνικές βελτιστοποίησης που εφαρμόστηκαν κατά τη διάρκεια της ανάπτυξης του παιχνιδιού στη Unity.

Εκτελέστηκαν benchmark σε μια οκτάδα υπολογιστών PC με λειτουργικό σύστημα Windows και διαφορετικά χαρακτηριστικά ο καθένας. Τα αποτελέσματα έδειξαν ότι το παιχνίδι εκτελείται ομαλά και χωρίς κωλύματα σε κάθε υπολογιστή, με την προϋπόθεση της προσαρμογής της ποιότητας των γραφικών και της ανάλυσης στις εκάστοτε δυνατότητες του υπολογιστή, όπως, γίνεται στα περισσότερα μοντέρνα 3D παιχνίδια που κυκλοφορούν σε υπολογιστές.

Το συμπέρασμα είναι ότι το πρότυπο μπορεί να εφαρμοστεί στην ανάπτυξη ενός αξιόλογου παιχνιδιού 3D r&c adventure, αλλά το γεγονός αυτό δεν αρκεί για μια εμπορική επιτυχία που θα μπορέσει να ανταγωνιστεί άλλους τίτλους του είδους στην παγκόσμια αγορά και να ευχαριστηθεί όποιος το δοκιμάσει. Απαιτείται η συμπλήρωση μιας ολόκληρης ομάδας από developers, που ο καθένας να ειδικεύεται σε ιδιαίτερους τομείς του game developer, όπως γραφίστες, animator, σεναριογράφοι κ.λ.π.

Στην Ελλάδα που μόλις τα τελευταία χρόνια άρχισε να μεγεθύνεται η gaming βιομηχανία με τη δημιουργία μικρών ομάδων game-dev, αλλά και κοινοτήτων που στηρίζουν κάθε νέα προσπάθεια ίσως η εκμετάλλευση του προτύπου να είναι μία ευκαιρία για τη δημιουργία ενός αξιόλογου εμπορικού τίτλου για τις παγκόσμιες αγορές βιντεοπαιχνιδιών.

7.2. Μελλοντικές προεκτάσεις

Κατά τη διάρκεια της σχεδίασης και της ανάπτυξης του προτύπου δόθηκε ιδιαίτερα βαρύτητα στην δυνατότητα επέκτασής του. Είναι δυνατή η προσθήκη περισσότερων μηχανισμών και λειτουργιών, χωρίς να παρουσιάζονται προβλήματα στο υπάρχον έργο. Σε αυτή την δυνατότητα συνεισέφεραν πολλά οι αρχές σχεδίασης προγραμματισμού και τα σχεδιαστικά πρότυπα που εφαρμόστηκαν.

Μία σημαντική επέκταση, είναι και η διεύρυνση των συστημάτων που θα μπορέσει να εκτελεστεί το παιχνίδι. Η ίδια η Unity προσφέρει την δυνατότητα εξαγωγής του υπάρχοντος κώδικα σε κονσόλες παιχνιδιών, κινητά και tablets. Για την εξαγωγή, όμως, σε αυτά τα μηχανήματα πρέπει να επεκταθούν και να προσαρμοστούν οι μηχανισμοί εισόδου. Καθένα από αυτά τα μηχανήματα έχει τις δικές του εισόδους, όπως είναι τα χειριστήρια, και οι οθόνες αφής. Επίσης, αν η εξαγωγή προορίζεται για κονσόλες πρέπει να καταργηθούν η επιλογή της ποιότητας των γραφικών και η επιλογή της ανάλυσης της οθόνης καθώς οι δυνατότητες της κάθε κονσόλας είναι δεδομένες.

Άλλη σημαντική επέκταση που δεν καλύφθηκε από το πρότυπο στην παρούσα εργασία, είναι η προσθήκη επιλογής γλώσσας κειμένου και φωνητικών. Όταν ένα εμπορικό παιχνίδι κυκλοφορεί για την παγκόσμια αγορά, περιέχει σχεδόν πάντοτε στις ρυθμίσεις του επιλογή μεταξύ γλωσσών για όλα τα κείμενα που εμφανίζονται στο παιχνίδι. Επίσης, διαθέτει επιλογή για την ενεργοποίηση ή μη των υπότιτλων και για το μέγεθός τους. Σε πολλά παιχνίδια υπάρχει ξεχωριστή επιλογή για την γλώσσα που θα ακούγεται στους διαλόγους.

Βιβλιογραφία

Εθνικό Κέντρο Οπτικοακουστικών Μέσων και Επικοινωνίας (2023) Υπουργείο Ψηφιακής Διακυβέρνησης.

<https://www.ekome.media/>

Χατζηγεωργίου Α. (2015). Αντικειμενοστραφής σχεδίαση UML, Αρχές, Πρότυπα και ευρετικοί κανόνες. Κλειδάριθμος.

Adobe. (2023). *Mixamo - Animate 3D characters for games, film, and more.*

<https://www.mixamo.com/>

Ahmet Sulek. (2023). *Flat UI Colors - Palettes, Flat UI Colors 280 handpicked colors.*

<https://flatuicolors.com/>

Aventurine SA (2007). *Darkfall* [Video Game]. Audiovisual Enterprises SAT.

<https://www.aventurine.gr/about/darkfall.html>

Atari, Inc. (1972). *Pong* [Video Game]. Atari, Inc.

Ballhaus, W., Chow, W., & Rivet, E. (2022). *Perspectives from the Global Entertainment & Media Outlook 2022–2026: Fault lines and fractures: Innovation and growth in a new competitive landscape.* PwC.

https://www.pwc.com/gx/en/industries/entertainment-media/outlook/downloads/PwC_Outlook22.pdf

Blender Foundation. (2021). Blender (Version 2.93) [Computer software].

<https://www.blender.org/>

Burst Studios. (1996). *Toonstruck* [Video Game]. Virgin Interactive.

Brackeys. (2017). *How to Make a Video Game* [Video]. YouTube.

<https://www.youtube.com/c/Brackeys/>

BTF/Headup Games. (2019). *Trüberbrook* [Video Game]. Headup Games.

<http://trueberbrook.com/>

CrazyBunch. (2018). *Leisure Suit Larry: Wet Dreams Don't Dry* [Video Game]. Assemble Entertainment.

<https://www.crazybunch.biz/games/leisure-suit-larry-wet-dreams-dont-dry/>

Crowther, W. (1976). *Colossal Cave Adventure* [Video Game]. Crowther.

Crytek.(2023) Crytek - video game developer.

<https://www.crytek.com/>

DeMarco, T. (1979). *Structured Analysis and System Specification*. Prentice Hall.

Doom Wiki. (n.d.). *Doom Engine*. Διαθέσιμο:

https://doomwiki.org/wiki/Doom_engine

Epic Games. (2023) *Unreal Engine: The most powerful real-time 3D creation tool*

<https://www.unrealengine.com/>

Funcom. (1999). *The Longest Journey* [Video Game]. Funcom.

<http://www.longestjourney.com/>

Gameathlon (2023), The No.1 Gaming &Technology EXPO in Greece. Επίσημη ιστοσελίδα: <https://gameathlon.eu/>

Rollings, A., & Adams, E. (2003). *Andrew Rollings and Ernest Adams on Game Design* (p. 43). New Riders.

Gilbert, R., & Winnick, G. (2014). *Thimbleweed Park* [Kickstarter campaign page].

Kickstarter. Διαθέσιμο στον σύνδεσμο::

<https://www.kickstarter.com/profile/thimbleweedpark/>

Hocking, J. (2018). *Unity in Action: Multiplatform Game Development in C#* (2nd ed.). Manning

Koulaxidis G, Xinogalos S. (2022). Improving Mobile Game Performance with Basic Optimization Techniques in Unity. *Modelling*. 3(2), 201-223.

<https://www.mdpi.com/2673-3951/3/2/14>

LucasArts (1998). *Grim Fandango* [Video Game]. LucasArts.

<https://grimremastered.com/>

LucasArts. (1991). *Monkey Island 2: LeChuck's Revenge*. [Video Game]. LucasArts.

Lucasfilm Games. (1987). *Maniac Mansion* [Video Game]. Lucasfilm Games.

Lucasfilm Games. (1990). *The Secret of Monkey Island*. [Video Game]. Lucasfilm Games.

Martin, R. (2003). *Agile Software Development: Principles, Patterns, and Practices* (p. 95). Prentice Hall.

Melissinos, C. (2015, 09, 22). Video Games Are the Most Important Art Form in History. *TIME*.

<https://time.com/collection-post/4038820/chris-melissinos-are-video-games-art/>

Microïds. (1999). *Syberia* [Video Game]. Microïds.

<https://www.microïds.com/game-syberia/>

Moon Studios. (2015). *Ori and the Blind Forest* [Video Game]. Microsoft Studios.

<https://www.orithegame.com/blind-forest/>

Nordic Games GmbH. (2014). *The Book of Unwritten Tales 2* [Video Game]. KING Art Games.

<https://www.thqnordic.com/games/book-unwritten-tales-2>

Okita, A. (2017). *Learning C# Programming with Unity 3D* (1st ed.). A K Peters/CRC Press.

Paint.NET (Version 5.0.7) [Computer software]. dotPDN LLC.

<https://www.getpaint.net/>

Pixabay GmbH. (2023). Pixabay - *Stunning royalty-free images & royalty-free stock*

<https://pixabay.com/>

Unity Technologies. (2023). *Unity Real-Time Development Platform*.

<https://unity.com/>

Revolution Software. (1996). *Broken Sword: The Shadow of the Templars* [Video Game]. Virgin Interactive.

https://revolution.co.uk/games_catalog/broken-sword-1/

Revolution Software/Revolution Software Ltd. (2020). *Beyond a Steel Sky* [Video Game]. Revolution Software Ltd.

https://revolution.co.uk/games_catalog/beyond-a-steel-sky/

Shutterstock. (2023). *TurboSquid | online 3D Model Store*.

<https://www.turbosquid.com/>

Sierra On-Line. (1990). *King's Quest V: Absence Makes the Heart Go Yonder!* [Video Game]. Sierra On-Line.

Sierra On-Line. (1992). *King's Quest VI: Heir Today, Gone Tomorrow* [Video Game]. Sierra On-Line.

Sketchfab. (2023). *Sketchfab - The leading platform for 3D & AR on the web*.

<https://sketchfab.com/>

Smith, M. (1983). *Manic Miner* [Video Game]. Bug-Byte

Thomas Thiriez. (2023). *TwistedWave - An audio editor Mobile application* . TwistedWave.

<https://twistedwave.com/online>

YoYo Games. (2023) *GameMaker Studio - video game developer*

<https://gamemaker.io/>