



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Σχολή Τεχνολογίας

Τμήμα Ψηφιακών Συστημάτων

ΑΝΑΠΤΥΞΗ ΒΙΝΤΕΟΠΑΙΧΝΙΔΙΟΥ ΜΕ UNITY ΓΙΑ

ANDROID/ IOS

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Χρήστος Μαραντίδης(ΑΜ: 3119130)

Επιβλέπων: Γεώργιος Κακαρόντζας, Αναπληρωτής Καθηγητής

ΛΑΡΙΣΑ, Οκτώβριος 2023



UNIVERSITY OF THESSALY

School of Technology

Digital Systems Department

**VIDEO GAME DEVELOPMENT WITH UNITY FOR
ANDROID/ IOS**

BACHELOR THESIS

Christos Marantidis (RN: 3119130)

Supervisor: *George Kakarontzas, Associate professor*

LARISSA, October 2023

Υπεύθυνη Δήλωση περί Ακαδημαϊκής Δεοντολογίας και Πνευματικών Δικαιωμάτων

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα πτυχιακή εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον, και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο/Η Δηλών/ούσα

(Υπογραφή)

Χρήστος Μαραντίδης

6/10/2023

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπων/πουσα	Ονοματεπώνυμο Επιβλέποντα Βαθμίδα/ιδιότητα επιβλέποντα, Τμήμα Ψηφιακών Συστημάτων, Πανεπιστήμιο Θεσσαλίας
Μέλος	Ονοματεπώνυμο Μέλους 1 Βαθμίδα/ιδιότητα μέλους 1, Τμήμα/Ιδρυμα μέλους 1
Μέλος	Ονοματεπώνυμο Μέλους 2 Βαθμίδα/ιδιότητα μέλους 2, Τμήμα/Ιδρυμα μέλους 2

Ημερομηνία έγκρισης: dd-mm-yyyy

Περίληψη

Η πτυχιακή εργασία εστίασε στην ανάπτυξη ενός βιντεοπαιχνιδιού με χρήση του προγράμματος Unity, στο πλαίσιο των λειτουργικών συστημάτων Android και iOS. Το παιχνίδι ανήκει στην κατηγορία top-down 2D RPG και περιλαμβάνει το μενού του παιχνιδιού καθώς και τρία επίπεδα με ποικιλομορφία εχθρών, με το τρίτο επίπεδο να περιλαμβάνει δύο boss. Για την επίτευξη των στόχων αυτών, ακολουθήθηκε μια συγκεκριμένη μεθοδολογία ανάπτυξης. Αρχικά, πραγματοποιήθηκε μια λεπτομερής ανάλυση των απαιτήσεων και του σχεδιασμού του παιχνιδιού. Ακολούθησε η ανάπτυξη του παιχνιδιού στο περιβάλλον του Unity, με έμφαση τόσο στην προγραμματιστική πλευρά του παιχνιδιού και τη δημιουργία των επιπέδων και των εχθρών όσο και στο σχεδιασμό του γραφικού κομματιού. Κατά τη διάρκεια της ανάπτυξης, παρακολούθηθηκε στενά η λειτουργία του παιχνιδιού για την ανίχνευση και την αποκατάσταση πιθανών σφαλμάτων. Το τελικό αποτέλεσμα ήταν ένα πλήρως λειτουργικό παιχνίδι που πληρούσε όλους τους στόχους της εργασίας. Τα τρία επίπεδα περιλάμβαναν μια ποικίλη εμπειρία για τον παίκτη, ενώ οι δύο τελικοί εχθροί στο τελευταίο επίπεδο αποτελούν μια δυναμική πρόκληση. Το μενού λειτουργούσε σε έναν ικανοποιητικό βαθμό, πληρώντας τις προϋποθέσεις για ένα μενού το οποίο προσφέρει βασικές λειτουργίες όπως η αποθήκευση και η φόρτωση της προόδου του παίκτη. Επιπλέον, κατά τη διάρκεια της ανάπτυξης, διεξήχθησαν δοκιμές και αναβαθμίσεις προκειμένου να διασφαλιστεί η σταθερή λειτουργία του παιχνιδιού χωρίς σφάλματα. Με βάση τα παραπάνω, η πτυχιακή εργασία παρουσιάζει επιτυχώς τη διαδικασία ανάπτυξης ενός βιντεοπαιχνιδιού με τη χρήση του Unity για τα λειτουργικά συστήματα Android και iOS, επιτυγχάνοντας όλους τους στόχους της εργασίας με μια λειτουργική και σταθερή απόδοση.

Abstract

The bachelor's thesis focused on the development of a video game using the Unity platform, within the Android and iOS operating systems. The game belongs to the top-down 2D RPG category and includes the game menu as well as three levels with a variety of enemies, with the third level featuring two bosses. To achieve these goals, a specific development methodology was followed. Initially, a detailed analysis of the game's requirements and design was conducted. This was followed by the development of the game in the Unity environment, with emphasis on both the programming side of the game and the creation of levels and enemies, as well as the design of the graphical elements. Throughout the development process, the game's functionality was closely monitored to detect and address any potential errors. The final result was a fully functional game that met all the objectives of the thesis. The three levels provided a diverse experience for the player, while the two bosses in the final level posed a dynamic challenge. The menu functioned satisfactorily, meeting the requirements for a menu that offered basic functions such as saving and loading player progress. Additionally, during development, testing and upgrades were conducted to ensure the stable operation of the game without errors. Based on the above, the bachelor's thesis successfully presents the process of developing a video game using Unity for Android and iOS operating systems, achieving all the objectives of the work with functional and stable performance.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Θεό πρώτα από όλα για το κουράγιο και την υπομονή που μου έδωσε για να μπορέσω να ολοκληρώσω την πτυχιακή εργασία. Επίσης είναι επιτακτική ανάγκη να δώσω θερμές ευχαριστίες στην οικογένειά μου για την ατελείωτη ψυχολογική στήριξη που μου δώσανε. Ακόμη ευχαριστώ τον κύριο Κακαρόντζα Γεώργιο για την βοήθεια και την καθοδήγηση που έδωσε. Τέλος είναι σημαντικό να δώσω τις ευχαριστίες μου στον συμφοιτητή-βοηθό στην πτυχιακή εργασία Αθανάσιο-Απόστολο Σύψα γιατί μαζί καταφέραμε να βγάλουμε εις πέρας ένα άρτια ολοκληρωμένο και ανεπτυγμένο παιχνίδι το οποίο είναι προσβάσιμο για συστήματα Android και IOS.

Χρήστος Μαραντίδης
ημερομηνία

Περιεχόμενα

ΠΕΡΙΛΗΨΗ	VII
ABSTRACT	IX
ΕΥΧΑΡΙΣΤΙΕΣ	XI
ΠΕΡΙΕΧΟΜΕΝΑ	XIII
1 ΕΙΣΑΓΩΓΗ	1
2 ΓΡΑΦΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ	ERROR! BOOKMARK NOT DEFINED.
2.1 ΜΕΘΟΔΟΙ ΓΡΑΦΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ	ERROR! BOOKMARK NOT DEFINED.
2.2 ΣΤΥΛ ΠΑΙΧΝΙΔΙΟΥ	5
2.3 ΜΕΝΟΥ	6
2.3.1 Εικόνα Μενου	7
2.4 ΕΠΙΠΕΔΟ 1 ^ο (FOREST)	8
2.4.1 Εικόνα 1 ^{ου} Επιπέδου	9
2.5 ΚΑΤΑΣΤΗΜΑ	11
2.5.1 Εικόνα Καταστήματος	12
2.6 ΕΠΙΠΕΔΟ 2 ^ο (CATACOMBS)	12
2.6.1 Εικόνα 2 ^{ου} Επιπέδου	13
2.7 ΕΠΙΠΕΔΟ 3 ^ο (BOSS)	18
2.7.1 Εικόνα 3 ^{ου} Επιπέδου	19
2.8 Εργαλία που χρησιμοποιήθηκαν	19
2.9 Διεπαφή Χρήστη (UI)	20
2.10 ANIMATIONS	21
2.10.1 Εχθροί	22
2.10.2 Συστατικά	22
3 ΑΠΟΘΗΚΕΥΣΗ ΚΑΙ ΦΟΡΤΩΣΗ ΠΡΟΟΔΟΥ	23
3.1 ΑΠΟΘΗΚΕΥΣΗ	23
3.1.1 κωδικας Αποθηκευσης	24
3.2 ΦΟΡΤΩΣΗ ΠΡΟΟΔΟΥ	26

3.2.1 κωδικας Φορτώσης Προόδου.....	27
4 ΣΧΕΔΙΑΣΜΟΣ ΕΧΘΡΩΝ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΛΟΓΙΚΗ	29
4.1 ΣΧΕΔΙΑΣΜΟΣ ΕΧΘΡΩΝ	29
4.2 ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΛΟΓΙΚΗ	30
4.2.1 κωδικας καταδίωξης και υπόδειξης ζημίας στον παίχτη	30
4.3 ΜΗΧΑΝΙΣΜΟΣ ΖΗΜΙΑΣ ΕΧΘΡΩΝ ΣΕ ΠΑΙΧΤΗ	33
4.3.1 κωδικας μηχανισμού ζημίας από εχθρούς σε παίχτη	34
5 ΑΝΑΛΥΣΕΙΣ ΟΘΟΝΗΣ	39
5.1 ΑΡΧΙΚΗ ΑΝΑΛΥΣΗ ΠΑΙΧΝΙΔΙΟΥ	40
5.2 ΑΝΑΛΥΣΗ ΘΕΣΕΩΝ ΑΝΤΙΚΕΙΜΕΝΩΝ	42
5.2.1 κωδικας ανάλυσης θέσεων αντικειμένων	43
6 ΔΟΚΙΜΗ ΛΕΙΤΟΥΡΓΙΑΣ ΠΑΙΧΝΙΔΙΟΥ ΣΕ ANDROID	45
7 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	47
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	49
ΠΑΡΑΡΤΗΜΑ Α	51
ΠΑΡΑΡΤΗΜΑ Β	55
ΠΑΡΑΡΤΗΜΑ Γ	57

1 Εισαγωγή

Η πτυχιακή είχε ως σκοπό την ανάπτυξη ενός βιντεοπαιχνιδιού με το πρόγραμμα Unity, στο πλαίσιο των λειτουργικών συστημάτων Android και iOS. Η ανάπτυξη βιντεοπαιχνιδιών έχει γνωρίσει μια εκρηκτική αύξηση τα τελευταία χρόνια καθώς οι τεχνολογικές εξελίξεις και η δημοτικότητα των κινητών συσκευών Android και IOS έχουν ανοίξει νέες πόρτες στον κόσμο των βιντεοπαιχνιδιών καθώς αμέτρητες εφαρμογές με πάρα πολλές κατηγορίες βιντεοπαιχνιδιών έχουν αναπτυχθεί με σκοπό την ψυχαγωγία εκατομμυρίων ανθρώπων ανά τον πλανήτη. Ένα από τα πιο δημοφιλή εργαλεία για την ανάπτυξη παιχνιδιών είναι το Unity, μια πλατφόρμα που επιτρέπει την ανάπτυξη βιντεοπαιχνιδιών σε οποιαδήποτε πλατφόρμα συμπεριλαμβανομένου του Android και του IOS με την γλώσσα ανάπτυξης C#.

Στο πλαίσιο αυτής της πτυχιακής εργασίας θα εστιάσουμε στον σχεδιασμό του παιχνιδιού, την μουσική που συνοδεύει το κάθε επίπεδο όπως και το μενού, θα εξερευνήσουμε τη λογική προγραμματισμού και τον σχεδιασμό των εχθρών, συμπεριλαμβανομένων και των δύο αρχηγών. Επίσης, θα εξετάσουμε την υλοποίηση του συστήματος ζημίας που καταφέρνουν οι εχθροί στον παίκτη, το σύστημα αποθήκευσης και φόρτωσης της προόδου του παίκτη, τις αναλύσεις που προσαρμόζονται για κάθε ανάλυση της κινητής συσκευής, τα γραφικά (assets) που χρησιμοποιούνται, τα hitbox των εχθρών, τις μεταβάσεις σε σκηνές, τις κινούμενες εικόνες (animations) των εχθρών και των αντικειμένων, καθώς και το χρήστη (UI) από όλα τα αντικείμενα.

2 Γραφικός Σχεδιασμός

Στον κόσμο των βιντεοπαιχνιδιών, τα γραφικά στοιχεία παίζουν καθοριστικό ρόλο στην εμπλοκή των παικτών και στη βύθισή τους στον κόσμο του παιχνιδιού. Αυτή η διατριβή εμβαθύνει στη γραφική σχεδίαση του "Epic Quest", top-down ενός 2D RPG που δημιουργήθηκε χρησιμοποιώντας το Unity. Οι οπτικές πτυχές του παιχνιδιού δεν είναι απλώς διακοσμητικές, αλλά χρησιμεύουν ως κρίσιμο εργαλείο για τη μετάδοση της αφήγησης, ενισχύοντας τη βύθιση του παίκτη και συμβάλλοντας στη συνολική ικανοποίηση του ίδιου για το παιχνίδι.

2.1 Μέθοδοι Γραφικού Σχεδιασμού

Υπάρχουν πάρα πολλές μέθοδοι γραφικού σχεδιασμού οι οποίες χρησιμοποιούνται για την κατασκευή παιχνιδιών. Κάθε μέθοδος έχει τα πλεονεκτήματά της και τα μειονεκτήματά της. Οπότε θα αναφερθούν κάποιες μέθοδοι που χρησιμοποιούνται και είναι δημοφιλή για την υλοποίηση παρόμοιων παιχνιδιών.

Επίπεδο πλακιδίων

Ο σχεδιασμός σε επίπεδο πλακιδίων είναι μια θεμελιώδης μέθοδος για τη δημιουργία περιβαλλόντων παιχνιδιών 2D RPG. Προσφέρει μια οργανωμένη και αποτελεσματική προσέγγιση για την κατασκευή λεπτομερών κόσμων παιχνιδιών. Διαιρώντας τον κόσμο του παιχνιδιού σε πλακίδια που βασίζονται σε πλέγμα, οι προγραμματιστές μπορούν εύκολα να τοποθετήσουν και να χειριστούν το έδαφος, τα αντικείμενα και τα εμπόδια. Αυτή η μέθοδος διευκολύνει τη δημιουργία περίπλοκων επιπέδων, όπως δάση με μονοπάτια, λίμνες, βουνά, δέντρα και διάφορα διαδραστικά στοιχεία όπως σεντούκια και κρυμμένους εχθρούς. Επιπλέον, ο σχεδιασμός που βασίζεται σε πλακίδια προάγει τη συνέπεια στην αισθητική επιπέδου και τη μηχανική του παιχνιδιού, καθιστώντας το ένα πολύτιμο εργαλείο για προγραμματιστές που στοχεύουν να εξισορροπήσουν την οπτική απήχηση και την πολυπλοκότητα του παιχνιδιού.

Κύλιση παράλλαξης

Η κύλιση παράλλαξης είναι μια ισχυρή τεχνική για τη βελτίωση του οπτικού βάθους και της εμβάπτισης σε παιχνίδια τέχνης pixel. Αυτό το επιτυγχάνει μετακινώντας τα στρώματα φόντου με διαφορετικές ταχύτητες, δημιουργώντας μια ψευδαίσθηση βάθους

και προοπτικής. Η κύλιση παράλλαξης μπορεί να δώσει ζωή στο επίπεδο του δάσους, κάνοντας λίμνες, βουνά και άλλα στοιχεία φόντου να φαίνονται μακρινά ή πιο κοντά. Αυτή η μέθοδος συμβάλλει σημαντικά στην ατμόσφαιρα του παιχνιδιού και στην αφοσίωση των παικτών προσθέτοντας ένα δυναμικό και αισθητικά ευχάριστο οπτικό στοιχείο στο περιβάλλον.

Δυναμικά εφέ φωτισμού και σκίασης

Η εφαρμογή δυναμικών εφέ φωτισμού και σκίασης προσθέτει ένα επίπεδο ρεαλισμού και ατμόσφαιρας στο παιχνίδι. Οι ανεπαίσθητες αλλαγές φωτισμού μπορούν να καθοδηγήσουν τους παίκτες στο μονοπάτι, ενώ δημιουργούν σασπένς σε σκιασμένες περιοχές όπου κρύβονται εχθροί. Ελέγχοντας το φωτισμό και τη σκίαση, μπορείτε να επηρεαστούν τα συναισθήματα και την εμπειρία παιχνιδιού του παίκτη. Αυτή η μέθοδος γίνεται κρίσιμη για τη δημιουργία μιας σαγηνευτικής και καθηλωτικής ατμόσφαιρας.

Αλγόριθμοι τυχαιοποίησης

Οι αλγόριθμοι τυχαιοποίησης παίζουν ζωτικό ρόλο στη διατήρηση του παιχνιδιού φρέσκου και ελκυστικού. Αυτοί οι αλγόριθμοι καθορίζουν την τοποθέτηση των εχθρών και αντικειμένων, διασφαλίζοντας ότι οι παίκτες αντιμετωπίζουν διαφορετικές προκλήσεις σε κάθε επανάληψη του παιχνιδιού. Αυτή η μέθοδος ενισχύει τη δυνατότητα αναπαραγωγής και δοκιμάζει διαφορετικά τα αντανakλαστικά του παίκτη κάθε φορά, συμβάλλοντας στη μακροζωία και την ψυχαγωγική αξία του παιχνιδιού.

Τεχνικές απόδοσης ωκεανών

Οι τεχνικές απόδοσης ωκεανών προσθέτουν ρεαλισμό και οπτική έλξη στο τελικό επίπεδο, όπου το μονοπάτι περικλείεται από θάλασσα. Η εφαρμογή αυτών των τεχνικών μπορεί να δημιουργήσει ζωντανά κύματα και εφέ νερού, συμβάλλοντας στη βύθιση των παικτών. Αυτή η μέθοδος βελτιώνει την οπτική ποιότητα του παιχνιδιού και κάνει το τελικό επίπεδο πιο σαγηνευτικό και οπτικά δυναμικό, ενισχύοντας έτσι τη συνολική εμπειρία παιχνιδιού.

Ποια μέθοδος Χρησιμοποιήθηκε

Στο αποκορύφωμα της διαδικασίας ανάπτυξης παιχνιδιών για το "Epic Quest", η μέθοδος που αναμφίβολα αναδείχθηκε καθοριστική για την επιτυχή υλοποίησή του ήταν ο σχεδιασμός σε επίπεδο πλακιδίων. Αυτή η μέθοδος επιλέχθηκε για τον αξιοσημείωτο συνδυασμό απλότητας και ευελιξίας στη δημιουργία επιπέδων παιχνιδιού από την αρχή, επιτρέποντάς να κατασκευαστούν περίπλοκα και οπτικά σαγηνευτικά περιβάλλοντα χωρίς κόπο. Η δομημένη προσέγγιση με βάση το πλέγμα επέτρεψε την ακριβή τοποθέτηση του εδάφους, των αντικειμένων και των διαδραστικών στοιχείων,

διευκολύνοντας τη δημιουργία διαφορετικών τοπίων όπως δάση, μουντρούμια και ωκεάνια μονοπάτια. Επιπλέον, ο σχεδιασμός σε επίπεδο πλακιδίων εξισορρόπησε απρόσκοπτα την οπτική ελκυστικότητα και την πολυπλοκότητα του παιχνιδιού που απαιτούσε το 2D RPG παιχνίδι από πάνω προς τα κάτω με τεχνολογία pixel, παρέχοντας μια συνεκτική και καθηλωτική εμπειρία στους παίκτες. Η αποτελεσματικότητά του και η ευρεία χρήση του στη βιομηχανία ενισχύουν τη σημασία του στο ταξίδι ανάπτυξης παιχνιδιών, υπογραμμίζοντας τον ρόλο του ως το θεμέλιο πάνω στο οποίο χτίστηκε το "Epic Quest".

Είναι απαραίτητο να τονιστεί ότι στο σχεδιασμό του επιπέδου υπάρχουν διάφορα στρώματα εδαφών για διαχωρισμό. Αυτό προσθέτει οπτική λεπτομέρεια στο παιχνίδι δημιουργώντας έτσι ένα πιο ελκυστικό κόσμο. Ακόμη βοηθάει στην βελτιστοποίηση στην απόδοση επιτρέποντας στη μηχανή παιχνιδιού να έχει σε προτεραιότητα την φόρτωση σχετικών αντικειμένων βελτιώνοντας έτσι την ομαλότητα του παιχνιδιού. Αναλυτικότερα το στρώμα εδάφους αντιπροσωπεύει το βασικό έδαφος, το ανώτερο στρώμα του εδάφους μπορεί να περιλαμβάνει διακοσμητικά στοιχεία όπως γρασίδι ή μικρά αντικείμενα, ενώ τα στρώματα σύγκρουσης χειρίζονται την αλληλεπίδραση μεταξύ των χαρακτήρων και του περιβάλλοντος. Αυτή η δομή για το σχεδιασμό του εδάφους και των ανωτέρων επιπέδων εφαρμόζονται σε όλα τα επίπεδα του παιχνιδιού.

2.2 Στυλ Παιχνιδιού

Το βιντεοπαιχνίδι "Epic Quest" είναι ένα pixel-style παιχνίδι που είχε ως στόχο με το στυλ παιχνιδιού να δώσει μια ζωντανή ατμόσφαιρα σε κάθε επίπεδο αλλά και τη δική του προσωπικότητα ως προς τον σχεδιασμό κάθε επιπέδου ξεχωριστά. Αυτό ενισχύει την ικανοποίηση του παίχτη και την εμβάθυνση του για το ίδιο το παιχνίδι το οποίο πάντα θα του προσφέρει κάτι το διαφορετικό σε εμφάνιση που έχει ως αποτέλεσμα τη μη κούραση του παίχτη ή την αδιαφορία του από το παιχνίδι λόγω επαναλαμβανόμενων και προβλέψιμων εμφανίσεων του παιχνιδιού. Η έμπνευση που πάρθηκε για την υλοποίηση του προήλθε από μεγάλους τίτλους εταιριών παρόμοιας κατηγορίας όπως το "Legend of Zelda" της Nintendo, το σύγχρονο δημοφιλές παιχνίδι "Undertale" και το "The Binding Of Isaac" στο οποίο εμπνεύστηκε και η προγραμματιστική λειτουργία του Shop εκτός από το σχεδιασμό που θα ειπωθεί παρακάτω. Λόγω του ξεχωριστού σχεδιασμού του κάθε επιπέδου μαζί με το μενού του παιχνιδιού και το κατάστημα. Παρακάτω θα αναλυθεί ο σχεδιασμός τους και η λογική πίσω από αυτόν τον σχεδιασμό.

2.3 Μενού

Ο γραφικός σχεδιασμός του συστήματος μενού στο " Epic Quest " είναι ένα κρίσιμο στοιχείο στη διαμόρφωση της αρχικής αντίληψης του παιχνιδιού από τον παίκτη. Χρησιμεύει ως πύλη προς τον καθηλωτικό κόσμο που δημιουργήσαμε, δίνοντας τον τόνο στο ταξίδι του παίκτη και ενσταλάζοντας μια αίσθηση ενθουσιασμού και προσμονής για μια ανάλογη η και μια ακόμα καλύτερη συνέχεια. Αυτή η ενότητα διερευνά τη λογική και τη διαδικασία σκέψης πίσω από τη σχεδίαση του μενού, εστιάζοντας στον ρόλο του να καλωσορίζει ομαλά τους παίκτες στο παιχνίδι, ενώ τους εμποτίζει με μια ζωνρή και αναζωογονητική αίσθηση.

Ο σχεδιασμός του μενού ξεκινά με μια απρόσκοπτη μετάβαση από την οθόνη τίτλου στο κύριο μενού. Εμφανίζεται ένα μαγευτικό φόντο με μια γραφική σκηνή από τον κόσμο του παιχνιδιού, δημιουργώντας μια άμεση σύνδεση με την αισθητική του παιχνιδιού. Αυτή η οπτική συνέχεια διασφαλίζει ότι οι παίκτες αισθάνονται ότι μπαίνουν στον κόσμο του παιχνιδιού από τη στιγμή που εκκινούν την εφαρμογή.

Η οπτική συνέπεια είναι ζωτικής σημασίας για τη διατήρηση της εμβάθυνσης του παίκτη. Ο συνδυασμός χρωμάτων, η τυπογραφία και τα γραφικά στοιχεία του μενού από τα κουμπιά τα γράμματα μέχρι και την εικόνα από πίσω αντικατοπτρίζουν αυτά που χρησιμοποιούνται σε όλο το παιχνίδι. Αυτή η συνοχή ενισχύει τη σύνδεση του παίκτη με τον κόσμο του παιχνιδιού και την αφήγηση, κάνοντας τη μετάβαση από το μενού στο παιχνίδι να είναι φυσική και αβίαστη.

Μία από τις βασικές προκλήσεις στον σχεδιασμό του μενού είναι η επίτευξη της σωστής ισορροπίας μεταξύ ζωντάνιας και κομψότητας. Στο "Epic Quest", επιτυγχάνουμε αυτήν την ισορροπία ενσωματώνοντας διακριτικά τον ήχο που θα ειπωθεί παρακάτω μαζί με το όλο γραφική εμφάνιση προσεφέροντας μια ισορροπία στον ενθουσιασμό αλλά και στην χαλάρωση του παίκτη. Για παράδειγμα, τα κουμπιά μενού εμφανίζουν ήπια χρώματα στο εξωτερικό τους, παρέχοντας οπτική ανατροφοδότηση στους παίκτες, ενώ τα γράμματα είναι ζωνρά, για να διατηρηθεί ένας κομψός και συνεκτικός σχεδιασμός.

2.3.1 Εικόνα Μενού



Εικόνα 1: Μενού γραφική εμφάνιση.

Όπως φαίνεται υπάρχει στο φόντο μια εικόνα που δείχνει ένα δάσος προσδίδοντας στον θεατή μια αίσθηση ηρεμίας η οποία συμβολίζεται μέσω των δασών λόγω του πρασίνου και των μεγάλων κορμών των δέντρων που στοιχίζονται σε σειρά το ένα με το άλλο και προθυμίας εξερεύνησης λόγω της περιέργειας του ανθρώπου να εξερευνάει το κάτι διαφορετικό ή νέο, που θα του κεντρίσει το ενδιαφέρον και θα τον ωθήσει σε περαιτέρω εξερεύνηση για να ικανοποιήσει την περιέργεια αυτή.

Παρατηρώντας παραπάνω θα διαπιστώσουμε ότι τα κουμπιά του μενού είναι ένα μια ανοιχτή απόχρωση του γκρι με μια αίσθηση άσπρου το οποίο θεωρείται ότι δίνει μια ισορροπία ανάμεσα στον ενθουσιασμό και την ηρεμία του παίχτη κάτι που ήταν ένας από τους στόχους της δημιουργίας του μενού που στέφθηκε με επιτυχία. Επίσης δίνει μια ευελιξία και προσαρμόζεται με ποικίλα χρώματα δίνοντας την δυνατότητα να υπάρχουν πολλές επιλογές για το χρώμα που θα διακοσμήσει τα γράμματα των κουμπιών. Για να επιτευχθεί το χρώμα των κουμπιών χρησιμοποιείται το component button του Unity το οποίο μπορείς να προσαρμόσεις το χρώμα, πόσο σκούρο ή ανοιχτό θα είναι το χρώμα που επιλέχθηκε, τι χρώμα θα είναι όταν πατάς το κουμπί και το πόσο διαφανές θα είναι. Στο δεκαεξαδικό σύστημα το ανοιχτό γκρι που χρησιμοποιήθηκε είναι το C8C8C8.

Προχωρώντας στα γράμματα των κουμπιών και του τίτλου ως χρώμα επιλέχθηκε μια απόχρωση του πορτοκαλί το οποίο στο δεκαεξαδικό σύστημα είναι το FF7C00. Αυτό το συγκεκριμένο χρώμα επιλέχθηκε διότι είναι ένα ενεργητικό και ενθουσιώδες χρώμα το

οποίο περνάει και στα συναισθήματα του παίχτη μόλις το βλέπει. Επίσης δίνει μια αίσθηση άνεσης και ζεστασιάς και δίνει στο παίχτη την δυνατότητα να εμβαθύνει στο παιχνίδι και να νιώσει μια οικειότητα με το παιχνίδι. Ακόμη το χρώμα αυτό είναι γνωστό ότι τραβάει την προσοχή των ανθρώπων και τους προσελκύει ώστε να δείξουν ενδιαφέρον για το παιχνίδι. Για την επίτευξη του χρώματος και των γραμμάτων χρησιμοποιήθηκε το TextMeshPro το οποίο είναι ένα εργαλείο του unity το οποίο δίνει την δυνατότητα στον χρήστη να γράφει το πως λέγεται το κουμπί αλλά και να αλλάζει χρώμα στα γράμματα.

2.4 Επίπεδο 1^ο(Forest)

Το επίπεδο του δάσους ανοίγει με μια απρόσκοπτη ενσωμάτωση του παιχνιδιού και της εκμάθησης του παίχτη με τον χειρισμό του παιχνιδιού. Καθώς ο παίκτης εισέρχεται στο καταπράσινο, ζωντανό δάσος, του ζητείται αμέσως να αλληλοεπιδράσουν με τον κόσμο μέσω ενός κουμπιού "Αλληλεπίδραση". Μια λεπτή προτροπή στην οθόνη καθοδηγεί τη συσκευή αναπαραγωγής, ενθαρρύνοντάς τον παίχτη να πατήσει το κουμπί. Όταν ενεργοποιηθεί, αυτός ο μηχανικός αλληλεπίδρασης προσφέρει μια δυναμική εμπειρία εκμάθησης. Κάνοντας τον παίχτη πιο οικείο με τον χειρισμό του παιχνιδιού. Αυτή η πρώιμη αλληλεπίδραση όχι μόνο διδάσκει στον παίκτη το βασικό χειρισμό αλλά δίνει επίσης τον τόνο στην ατμόσφαιρα του παιχνιδιού, τραβώντας τον πιο βαθιά στον μαγευτικό κόσμο.

Καθώς ο παίκτης συνεχίζει το ταξίδι του μέσα στο δάσος, το σχέδιο εισάγει στοιχεία προσοχής και κινδύνου. Το δάσος κρύβει κρυμμένους εχθρούς που χτυπούν όταν δεν περίμενε κανείς. Για παράδειγμα, το φαινομενικά ειρηνικό μονοπάτι που ακολουθεί ο παίκτης ή μια παράκαμψη που παίρνει είτε για να πάρει σεντούκια είτε για να περάσει πιο γρήγορα το επίπεδο μπορεί ξαφνικά να μετατραπεί σε μια επικίνδυνη ενέδρα από καμουφλαρισμένους εχθρούς οι οποίοι με την ταχύτητά τους τεστάρουν τα αντανakλαστικά του χρήστη. Ο παίκτης μαθαίνει να είναι συνεχώς σε εγρήγορση, αναπτύσσοντας μια αυξημένη αίσθηση επίγνωσης. Αυτή η λεπτή εισαγωγή στον κίνδυνο χρησιμεύει ως προοίμιο για πιο τρομερές προκλήσεις σε μεταγενέστερα επίπεδα, δημιουργώντας μια αίσθηση προσμονής και ενθουσιασμού.

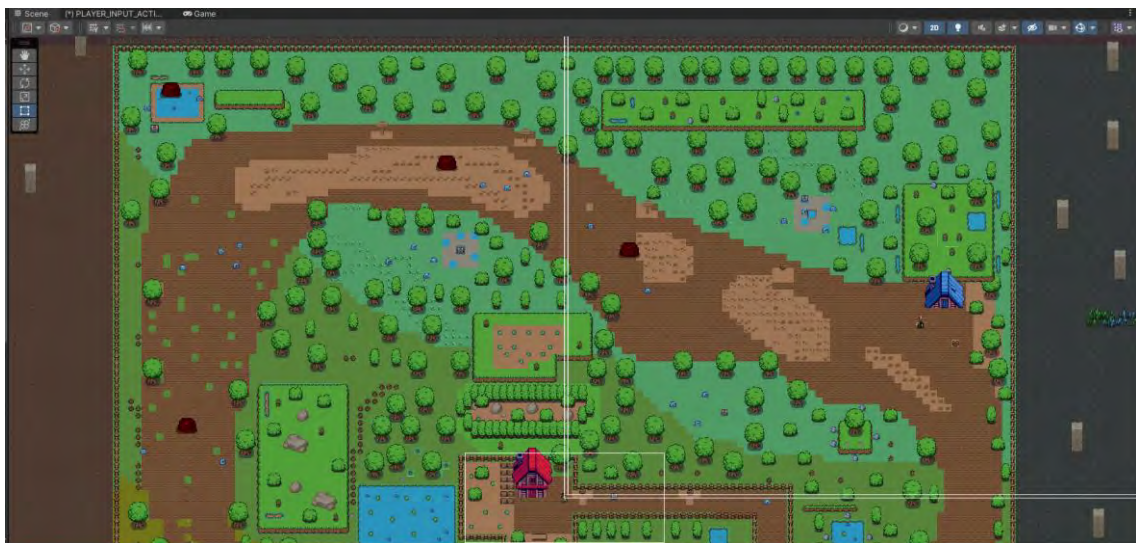
Ένα από τα εντυπωσιακά χαρακτηριστικά του δασικού επιπέδου είναι το συνεχώς μεταβαλλόμενο περιβάλλον του. Τα γραφικά περνούν απρόσκοπτα μεταξύ καταπράσινων λιβαδιών, πυκνών δέντρων σε κάποια σημεία, μικρά ποτάμια, λόφους και

μεγάλα βουνά. Κάθε τμήμα του δάσους προσφέρει διαφορετικές ευκαιρίες για εξερεύνηση και ανακάλυψη. Τα μονοπάτια, τα κρυμμένα σεντούκια και η εξερεύνηση του περιβάλλοντος ενθαρρύνουν τους παίκτες να βγουν από την πεπατημένη, ανταμείβοντας την περιέργεια με νέες εμπειρίες. Αυτή η ποικιλομορφία όχι μόνο προσθέτει βάθος στο πρώτο επίπεδο αλλά επίσης δελεάζει τους παίκτες να εξερευνήσουν και να βυθιστούν στον κόσμο του παιχνιδιού.

Ο γραφικός σχεδιασμός του δασικού επιπέδου εξυπηρετεί διπλό σκοπό, προμηνύοντας τις πιο δύσκολες προκλήσεις που βρίσκονται μπροστά. Μέσω του έξυπνου σχεδιασμού, το δάσος σταδιακά αλλάζει με την προσθήκη αντικειμένων κυρίως ερειπίων, με ολοένα και πιο δύσκολες συναντήσεις εχθρών. Καθώς οι παίκτες προχωρούν, γίνονται μάρτυρες του περιβάλλοντος που μεταμορφώνεται, υπονοώντας τις επικείμενες δοκιμασίες. Αυτή η ανεπαίσθητη αλλαγή δυσκολίας προετοιμάζει τους παίκτες για τις κλιμακούμενες προκλήσεις των μεταγενέστερων επιπέδων, διασφαλίζοντας μια ομαλή καμπύλη μάθησης. Το επίπεδο του δάσους, με την πλούσια διαδραστικότητα, τους κρυφούς κινδύνους και τα διαφορετικά περιβάλλοντα θέτει τις βάσεις για μια συναρπαστική και καθηλωτική εμπειρία παιχνιδιού, ενώ ενσωματώνει απρόσκοπτα τη μηχανική και την αφήγηση του παιχνιδιού.

2.4.1 Εικόνα 1^{ου} Επιπέδου

Εδώ παρουσιάζεται η έκταση ολόκληρου του πρώτου επιπέδου που παρουσιάζεται σε δύο εικόνες.



Εικόνα 2: 1ο μέρος του πρώτου επιπέδου.



Εικόνα 3: 2ο μέρος πρώτου επιπέδου.

Το πρώτο επίπεδο έχει στην αρχή ένα σπίτι το οποίο δίνει την αίσθηση στον παίχτη ότι είναι το σπίτι του χαρακτήρα τον οποίο παίζει. Όσο προχωράει ο παίχτης και εξοικειώνεται τον παίχτη με τις πινακίδες που εξυπηρετούν το σκοπό αυτό παρατηρείται ότι υπάρχουν κάποια σεντούκια διασκορπισμένα σε όλο το επίπεδο με σκοπό να ωθήσει το παίχτη στην εξερεύνηση του επιπέδου και στην εμβάθυνση του στο παιχνίδι. Κάποια σεντούκια έχουν φύλαξη από εχθρούς. Αυτό έχει ως σκοπό να δώσει στον παίχτη την αίσθηση στον παίχτη ότι πρέπει να πολεμήσει και με την αξία του να πάρει τα σεντούκια που περιέχουν μέσα κέρματα που χρησιμεύουν για την αγορά αντικειμένων.

Στην Συνέχεια υπάρχει ένας καταστηματάρχης ο οποίος σου δείχνει το κατάστημα του και σου προτρέπει να μπεις μέσω της πόρτας του μαγαζιού του η λογική της οποίας θα αναλυθεί σε παρακάτω κεφάλαιο. Όσο προχωράει ο παίχτης παρατηρείται η σταδιακή αλλαγή του εδάφους από ξύλινη επιφάνεια σε χωμάτινη σε κάποια σημεία ή σε επιφάνεια γρασιδιού είτε σε εξερεύνηση είτε στο μονοπάτι που ακολουθεί και στο τέλος της πίστας σε μια πέτρινη επιφάνεια η οποία δείχνει παλιά και σε κάποια σημεία αντικαταστάθηκε από το γρασίδι. Στο σημείο της τηλεμεταφοράς που είναι η πέτρινη κατασκευή στο τέλος της πίστας σε τηλεμεταφέρει στην επόμενη πίστα. Τριγύρω από την τηλεμεταφορά απλώνονται κάποιες κολώνες και ερείπια κυρίως τα οποία κεντρίζουν την φαντασία του παίχτη προσπαθώντας ο ίδιος να καταλάβει τη ιστορία του παιχνιδιού. Το παιχνίδι είναι φτιαγμένο με τέτοιο τρόπο ώστε ο παίχτης να φανταστεί και να βγάλει τα δικά του συμπεράσματα για την ιστορία του.

2.5 Κατάστημα

Στη σχεδίαση του καταστήματος εντός παιχνιδιού στο πλαίσιο του δασικού επιπέδου, ο πρωταρχικός στόχος είναι να δημιουργηθεί ένας οπτικά ελκυστικός και λειτουργικός χώρος που χρησιμεύει ως ένα απρόσκοπτο σημείο μετάβασης για τους παίκτες να ξεφύγουν στιγμιαία από τη δράση υψηλής έντασης του παιχνιδιού ενώ προετοιμάζοντάς τους επίσης για τις επερχόμενες περιπέτειές τους. Αυτός ο σχεδιασμός διπλής χρήσης βασίζεται στην προσεκτική ισορροπία αισθητικής και χρηστικότητας. Για να επιτευχθεί αυτό, τα γραφικά στοιχεία του καταστήματος είναι σχεδιασμένα στρατηγικά για να προκαλούν μια αίσθηση ηρεμίας και γαλήνης. Γήινες και ηρεμιστικές παλέτες χρωμάτων, όπως διάφορες αποχρώσεις του καφέ και του γκρι, χρησιμοποιούνται για να αντηχούν στο φυσικό περιβάλλον του επιπέδου του δάσους. Ο απαλός, ατμοσφαιρικός φωτισμός εντός του καταστήματος όχι μόνο τονίζει τη ζεστή και φιλόξενη ατμόσφαιρα, αλλά προσφέρει επίσης στους παίκτες μια σύντομη ανάπαυλα από τη γρήγορη δράση του παιχνιδιού.

Επιπλέον, η διάταξη του καταστήματος έχει σχεδιαστεί προσεκτικά για να διευκολύνει την εμπλοκή και τη βύθιση των παικτών. Η παρουσία διαφόρων αντικειμένων σχημάτων στη διακόσμηση βοηθά στη χαλάρωση και ηρεμία του παίχτη, εξασφαλίζοντας μια απρόσκοπτη μετάβαση. Εκτός από την αισθητική, η χρηστικότητα παίζει καθοριστικό ρόλο, με διαισθητική κατηγοριοποίηση αντικειμένων, σαφή σήμανση και σχεδιασμό που βελτιώνει την ικανότητα του παίκτη να περιηγείται και να αγοράζει αντικείμενα γρήγορα. Με αυτόν τον τρόπο, η γραφική σχεδίαση του καταστήματος όχι μόνο παρέχει ένα ήρεμο καταφύγιο μέσα στο παιχνίδι, αλλά χρησιμεύει επίσης ως ένας πρακτικός κόμβος όπου οι παίκτες μπορούν να εξοπλιστούν αποτελεσματικά για τις περιπέτειές τους, εξασφαλίζοντας έναν αρμονικό συνδυασμό χαλάρωσης και προετοιμασίας στην εμπειρία παιχνιδιού.

2.5.1 Εικόνα καταστήματος



Εικόνα 4: Κατάστημα γραφική εμφάνιση.

Όπως παρατηρείται κατά την είσοδο του παίχτη στο κατάστημα τριγύρω του υπάρχουν διακοσμητικά αντικείμενα διάφορων ειδών όπως κουτιά, βαρέλια και σεντούκια. Βοηθώντας στη εξοικείωση του παίχτη με το σκηνικό και για να δοθεί μια αίσθηση ρεαλισμού. Επίσης υπάρχουν τριγύρω πινακίδες οι οποίες εξηγούν το τι είναι και πόσο κάνει κάθε αντικείμενό παίχτης αγγίζοντας τα αντικείμενα αυτά, εξαφανίζονται από τη σκηνή, τα αγοράζει ο παίχτης και αποθηκεύονται στο inventory του.

2.6 Επίπεδο 2^ο (Catacombs)

Η μετάβαση στο επίπεδο των κατακομβών ξεκινά με μια ανεπαίσθητη αλλαγή στον τόνο και την ατμόσφαιρα του παιχνιδιού. Καθώς οι παίκτες κατεβαίνουν πιο βαθιά στον κόσμο του παιχνιδιού, αντιμετωπίζονται με μια απτή αίσθηση προαισθήματος. Το άλλοτε γαλήνιο και ειδυλλιακό τοπίο του πρώτου επιπέδου μεταμορφώνεται σε έναν ζοφερό και αμυδρά φωτισμένο υπόγειο λαβύρινθο. Οι τοίχοι είναι στολισμένοι με αρχαία σύμβολα και αχνοί απόηχοι μακρινών, απόκοσμων ήχων διαπερνούν τον αέρα. Αυτή η ξαφνική αλλαγή στην ατμόσφαιρα ουσιαστικά σηματοδοτεί στους παίκτες ότι πρόκειται να ξεκινήσουν ένα πολύ πιο απαιτητικό και μυστηριώδες ταξίδι.

Ο σχεδιασμός του επιπέδου κατακόμβες έχει δημιουργηθεί σκόπιμα για να προκαλεί τους παίκτες τόσο πνευματικά όσο και σωματικά. Είναι μια δομή που μοιάζει με λαβύρινθο με πολλούς διαδρόμους, περάσματα και διασυνδεδεμένους θαλάμους. Το επίπεδο έχει σχεδιαστεί για να αποπροσανατολίζει τους παίκτες, αναγκάζοντάς τους να

βασίζονται στις εξερευνητικές τους δεξιότητες και στη διαίσθησή τους για να προοδεύσουν. Οι παίκτες πρέπει να κάνουν στρατηγικές επιλογές για τα μονοπάτια που θα ακολουθήσουν, οδηγώντας σε πολλαπλές διακλαδώσεις με διαφορετικά δωμάτια μέσα στο λαβύρινθο. Κάθε διαδρομή προσφέρει τις δικές της μοναδικές προκλήσεις και ανταμοιβές, ενθαρρύνοντας τους παίκτες να πειραματιστούν και να εξερευνήσουν διεξοδικά.

Για να ενισχύσει την απόκοσμη ατμόσφαιρα, το επίπεδο των κατακομβών κοσμεύεται με μακάβριες λεπτομέρειες. Οι τοίχοι είναι επενδυμένοι με αρχαίους τάφους και κρύπτες. Μυστηριώδεις σκελετοί παραμονεύουν στις γωνίες και ακούγονται απόκοσμοι ήχοι καθώς οι παίκτες βυθίζονται βαθύτερα στις κατακόμβες.

Μία από τις πιο σημαντικές αλλαγές στο επίπεδο των Κατακομβών είναι η εισαγωγή πιο απαιτητικών αντιπάλων. Οι παίκτες θα συναντήσουν εχθρούς που όχι μόνο είναι περισσότεροι αλλά διαθέτουν και πιο δύσκολοι να νικηθούν. Η εμφάνιση τους ταιριάζει με το στυλ του επιπέδου προσθέτοντας βάθος αλλά και μυστήριο στο παιχνίδι.

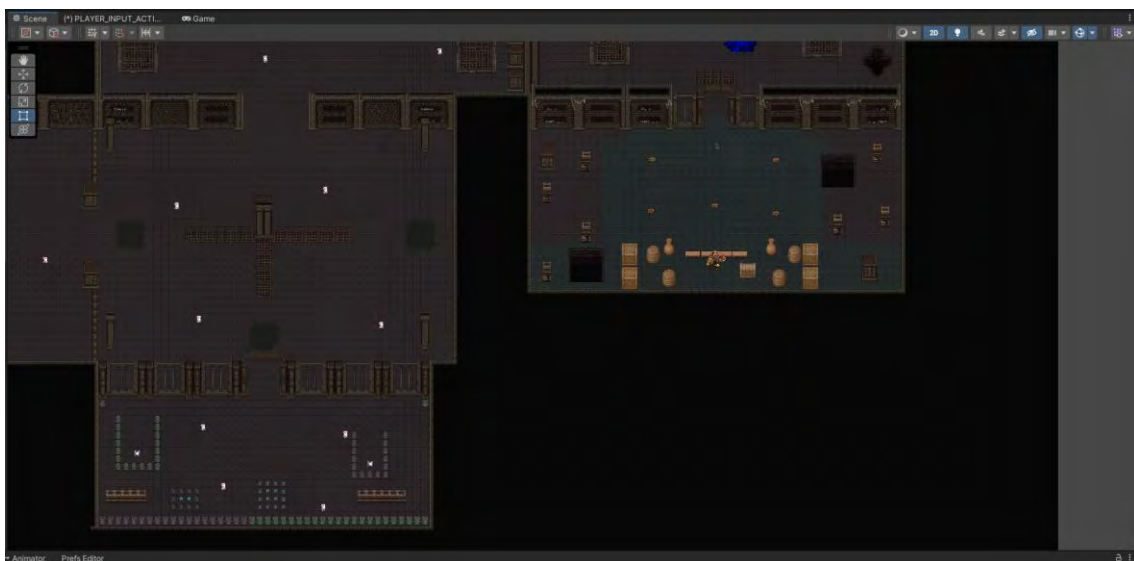
Συμπερασματικά, η μετάβαση από το πρώτο επίπεδο στις κατακόμβες στο "Epic Quest" είναι μια προσεκτικά δημιουργημένη εμπειρία που στοχεύει να αιχμαλωτίσει τους παίκτες με την απόκοσμη ατμόσφαιρα, το περίπλοκο επίπεδο σχεδιασμού και τις τρομερές προκλήσεις. Αυτή η αλλαγή του τόνου και του παιχνιδιού ενθαρρύνει τους παίκτες να εξερευνήσουν, να προσαρμοστούν, να βγάλουν τα δικά τους συμπεράσματα για το κόσμο στον οποίο βρίσκεται και να βυθιστούν στον κόσμο του παιχνιδιού, θέτοντας τη βάση για ένα συναρπαστικό και συναρπαστικό δεύτερο επίπεδο στην επική περιπέτεια.

2.6.1 Εικόνα 2^{ου} Επιπέδου

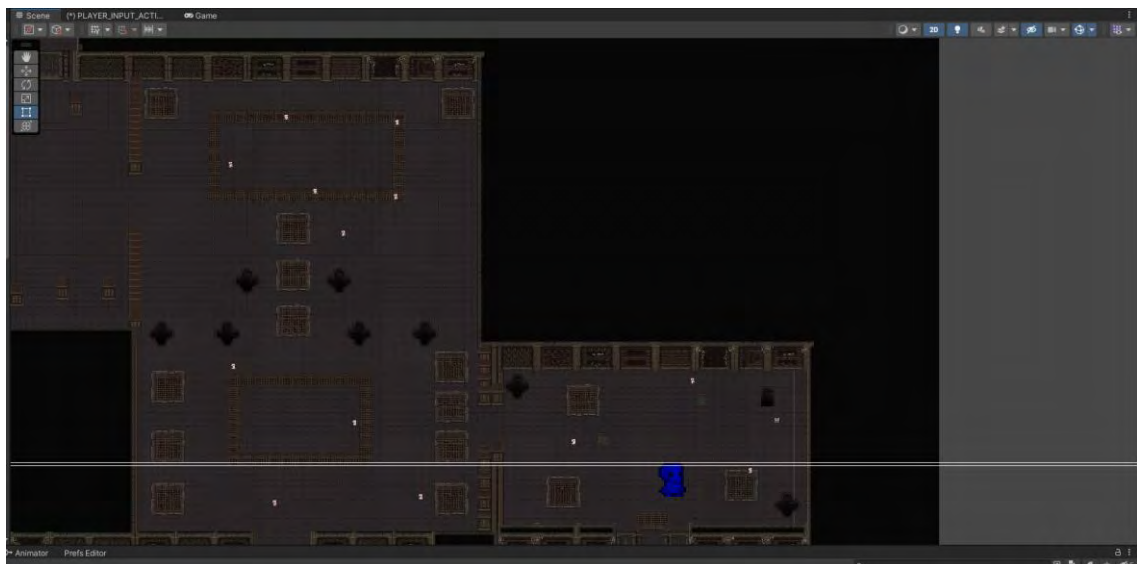
Εδώ παραπείθονται οι εικόνες του 2^{ου} επιπέδου. Είναι πολλές λόγω του μεγέθους του επιπέδου.



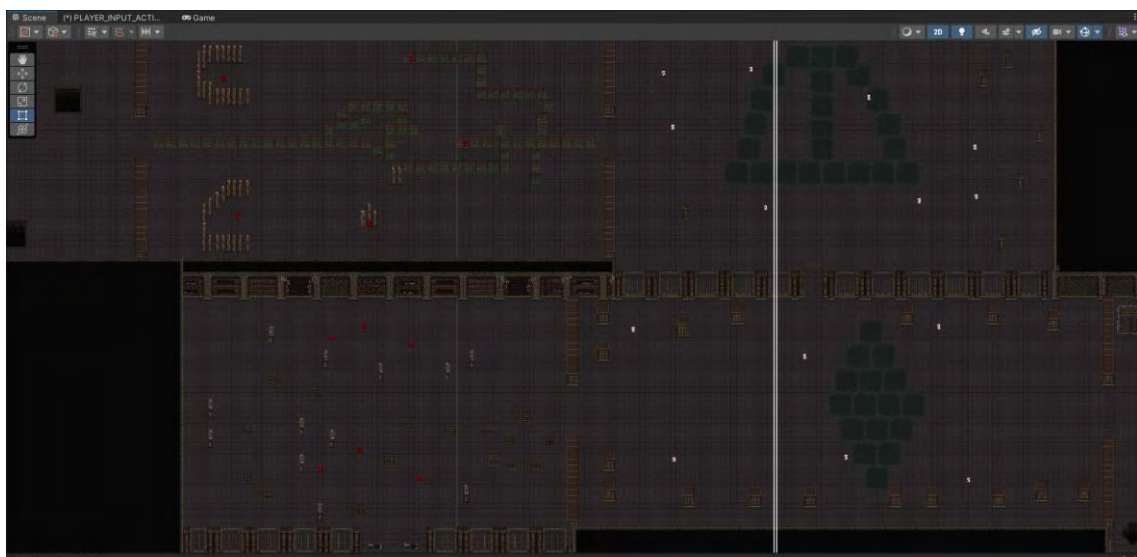
Εικόνα 5: 1ο μέρος δεύτερου επιπέδου.



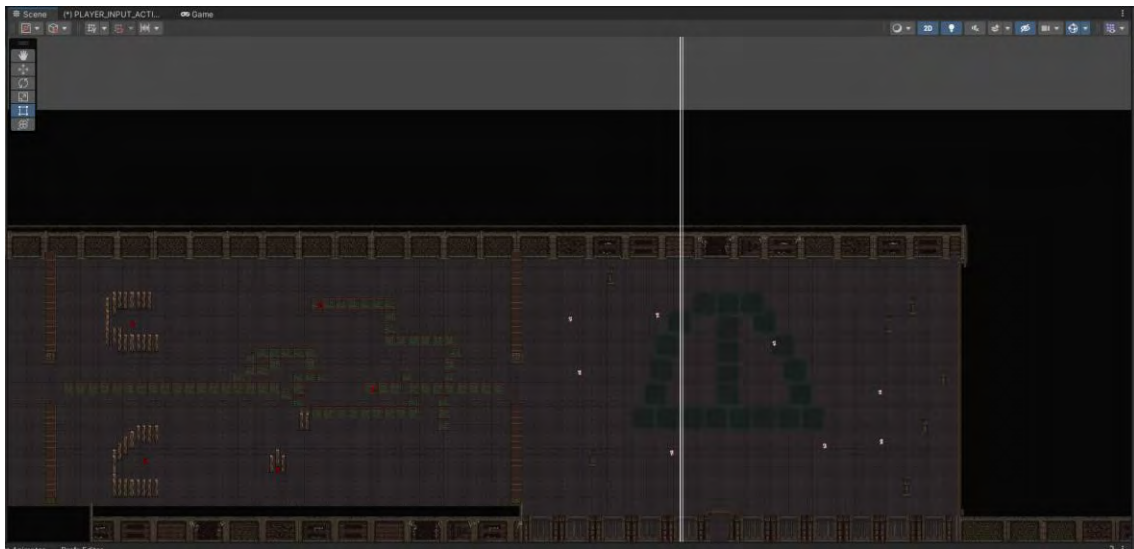
Εικόνα 6: 2ο μέρος δεύτερου επιπέδου.



Εικόνα 7: 3ο μέρος δεύτερου επιπέδου.



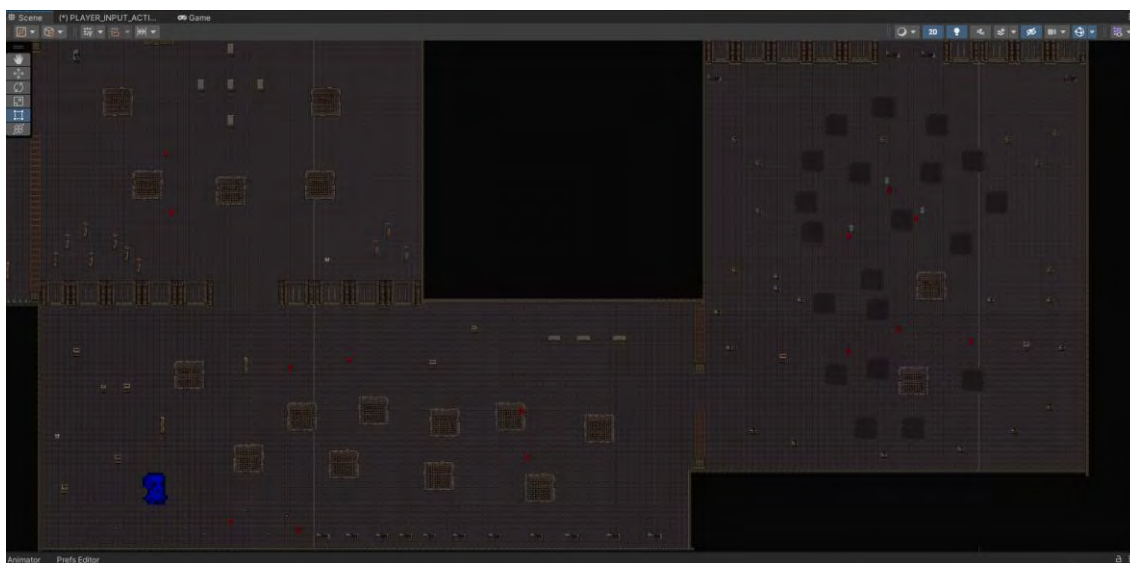
Εικόνα 8: 4ο μέρος δεύτερου επιπέδου.



Εικόνα 9: 5ο μέρος δεύτερου επιπέδου.



Εικόνα 10: 6ο μέρος δεύτερου επιπέδου.



Εικόνα 11: 7ο μέρος δεύτερου επιπέδου.



Εικόνα 12: 8ο μέρος δεύτερου επιπέδου.

Στο δεύτερο επίπεδο ξεκινάει ο παίχτης και βλέπει τριγύρω του διάφορους τοίχους και διακοσμητικά στοιχεία όπως τρύπες κολώνες και τάφοι. Όσο προχωράει αντικρίζει τους εχθρούς οι οποίοι είναι κανονικοί άσπροι σκελετοί για την ώρα. Αν προχωρήσει παραπέρα ο παίχτης παρατηρεί πολλές εισόδους στις οποίες μπορεί να μπει και να εξερευνήσει διάφορα δωμάτια. Κάποια έχουν εχθρούς σκελετούς οι οποίοι είναι κόκκινοι οι οποίοι είναι πιο γρήγοροι και οι μπλε που είναι πιο μεγάλοι και προκαλούν περισσότερη ζημιά από τους κανονικούς. Θα αναλυθεί σε παρακάτω κεφάλαιο η πρόκληση ζημιάς από εχθρούς σε παίχτη. Τα σεντούκια είναι διασκορπισμένα σε όλο το

επίπεδο αλλά υπάρχει και ένα ειδικό δωμάτιο το οποίο περιέχει 2 σεντούκια που είναι στην εικόνα 6. Επίσης υπάρχει και ένα κατάστημα χτισμένο σε αυτό το επίπεδο το οποίο λειτουργεί με τον ίδιο ακριβώς τρόπο όπως το κατάστημα στο πρώτο επίπεδο. Τέλος ο παίκτης εάν καταφέρει να φτάσει στο τελικό δωμάτιο και περάσει τους σκελετούς θα βρει μια πύλη τηλεμεταφοράς περιτριγυρισμένη από διάφορα κεριά για διακόσμηση η οποία σε οδηγεί στο τρίτο και τελευταίο επίπεδο του παιχνιδιού.

2.7 Επίπεδο 3^ο (Boss)

Το τρίτο και τελευταίο επίπεδο του "Epic Quest", που ονομάζεται "Boss", σηματοδοτεί την κορύφωση του παιχνιδιού, προσφέροντας στους παίκτες μια έντονη και συναρπαστική εμπειρία που ξεπερνά τις προκλήσεις των προηγούμενων επιπέδων. Σε αυτή το απόλυτη αναμέτρηση, οι παίκτες αντιμετωπίζουν όχι έναν αλλά δύο τρομερούς αντιπάλους: τον Dark Wizard και τον Fire Wizard. Αυτό το επίπεδο έχει σχεδιαστεί για να ωθεί τους παίκτες στα όριά τους με μοναδικούς μηχανισμούς, στρατηγικό παιχνίδι και ένα αξέχαστο σκηνικό.

Το επίπεδο Boss είναι μια κορύφωση όσον αφορά τη δυσκολία και την ένταση καθώς οι παίκτες πλησιάζουν το κτίριο στο οποίο θα αντιμετωπίσουν τους δύο αντιπάλους. Η προσμονή χτίζεται καθώς περπατούν το μονοπάτι για το κτίριο, που περιβάλλεται από μια απέραντη θάλασσα με σκόρπια βράχια, ενισχύοντας την αίσθηση της απομόνωσης. Στο ίδιο το νησί κυριαρχεί ένα δυσοίωνο κτίριο που χρησιμεύει ως το βήμα για την απόλυτη αναμέτρηση.

Η επιτυχία στο επίπεδο του Boss εξαρτάται από τη στρατηγική σκέψη και την ακρίβεια. Οι παίκτες πρέπει να αναλύσουν τα μοτίβα επίθεσης και των δύο αφεντικών, εκμεταλλευόμενοι τις αδυναμίες τους αποφεύγοντας καταστροφικές επιθέσεις. Ο συγχρονισμός είναι απαραίτητος, καθώς ο παίκτης μπορεί να χρειαστεί να αλλάξει μεταξύ στόχων ή να χρησιμοποιήσει το περιβάλλον προς όφελός του. Αυτό το επίπεδο απαιτεί τη μέγιστη ικανότητα και συγκέντρωση, διασφαλίζοντας ότι οι παίκτες συμμετέχει πλήρως στο επικό φινάλε.

Συμπερασματικά, το επίπεδο Boss του "Epic Quest" είναι το αποκορύφωμα του παιχνιδιού, προσφέροντας στους παίκτες μια σκληρή πρόκληση, μοναδικούς μηχανισμούς αφεντικού και ένα εκπληκτικό σκηνικό που περιβάλλεται από τη θάλασσα. Η αναμέτρηση με τον Dark Wizard και τον Fire Wizard είναι μια δοκιμασία δεξιοτήτων

και στρατηγικής, και η συνοδευτική μουσική του boss ενισχύει την καθηλωτική εμπειρία, κάνοντας αυτό το τελικό επίπεδο ένα αξέχαστο και επικό τέλος του παιχνιδιού.

2.7.1 Εικόνα 3^{ου} Επιπέδου



Εικόνα 13: Γραφική εμφάνιση τρίτου επιπέδου.

Όπως φαίνεται στην εικόνα ο παίχτης βρίσκεται σε ένα νησί το οποίο περιτριγυρίζεται από θάλασσα και βράχια. Υπάρχει το μονοπάτι που πρέπει να ακολουθήσει για να πάει στο κτίριο όπου γίνεται η αναμέτρηση. Τριγύρω από το κτίριο υπάρχουν πέτρες και ορυκτά όπου κάποια χρησιμοποιούνται ως διακοσμητικά και άλλα σαν αντικείμενα όπου συγκρούεται ο παίχτης και οι εχθροί. Εάν οι εχθροί νικηθούν εμφανίζεται ένα μήνυμα ότι ο παίχτης νίκησε το παιχνίδι.

2.8 Εργαλεία που χρησιμοποιήθηκαν

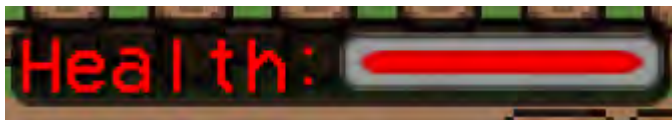
Τα εργαλεία που χρησιμοποιήθηκαν για τα Sprites είναι το Mystic Woods το οποίο περιέχει τα αντικείμενα για το 1^ο επίπεδο(βουνά, δέντρα, ποτάμια, χόρτα κ.ά.).Επίσης χρησιμοποιούνται από το Mystic woods το σεντούκι και την πινακίδα σαν αντικείμενο το οποίο υπάρχει στο 1^ο και 2^ο επίπεδο. Ακόμη υπάρχει το mini pack το οποίο χρησιμοποιείται για τα σπίτια που υπάρχουν στο 1^ο επίπεδο. Επιπλέον υπάρχουν τα tx-prors τα οποία περιέχουν την πύλη τηλεμεταφοράς που χρησιμοποιείται στο 1^ο και 2^ο επίπεδο και αντικείμενα(κολώνες παγκάκια, βαρέλια κ.α.) που χρησιμοποιούνται στο τέλος του πρώτου επιπέδου και στο κατάστημα .Επιπρόσθετα υπάρχει το RF-Catacombs

v1.0 το οποίο χρησιμοποιείται αποκλειστικά για την κατασκευή και τον σχεδιασμό του 2^{ου} επιπέδου. Τέλος χρησιμοποιήθηκε το RPGW_Caves_V2.1 το οποίο περιέχει ορυκτά πέτρες και το έδαφος για την κατασκευή του 3^{ου} και τελευταίου επιπέδου.

2.9 Διεπαφή Χρήστη (UI)

Για την διεπαφή χρήστη χρησιμοποιήθηκαν τα RPG Items 16x16 pack 1 τα οποία αποτελούν εικόνες για το inventory του παίχτη όταν έχει αντικείμενα τα οποία είναι είτε για εξοπλισμό (σπαθιά) είτε για κατανάλωση (κεράσια, φίλτρα).

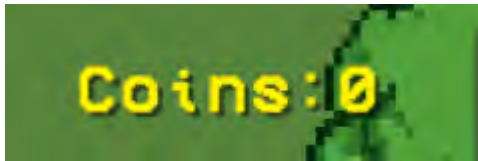
Απαραίτητο είναι να αναφερθεί η ζωή του παίχτη καθώς η διεπαφή της ζωής κατασκευάστηκε μέσω του Unity και ως σκοπό έχει την οπτική αναπαράσταση και ανατροφοδότηση της ζωής του παίχτη. Περαιτέρω ανάλυση για την λειτουργία της ζωής του παίχτη θα γίνει σε επόμενο κεφάλαιο.



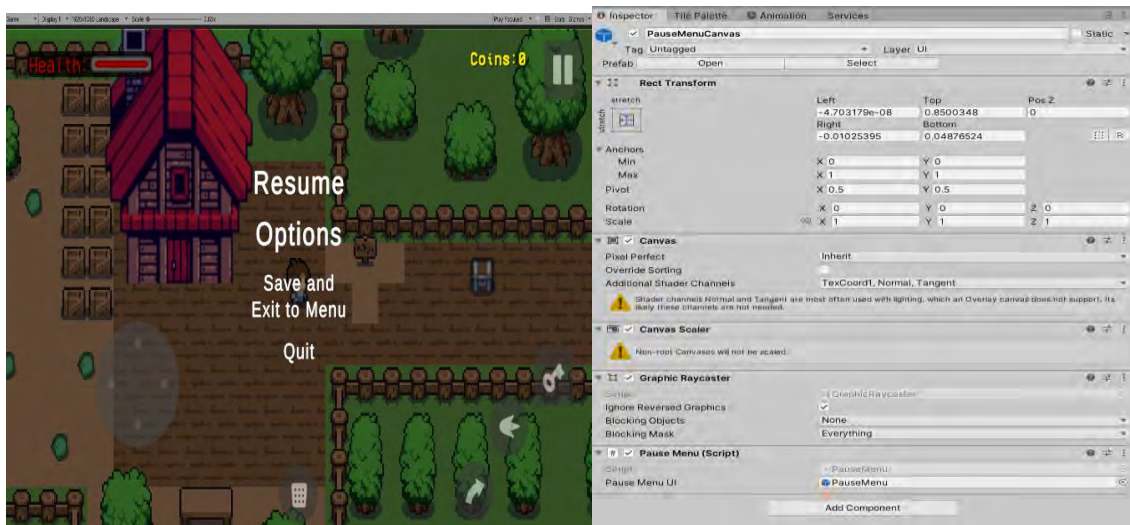
Εικόνα 14: Οπτική αναπαράσταση ζωής του παίχτη.

Είναι σημαντικό να αναφερθεί το ότι υπάρχουν οι καμβάδες οι οποίοι διαχειρίζονται την διεπαφή πολλών αντικειμένων όπως το Inventory, Pause Menu, WinScreenCanvas(διαχειρίζεται τη στιγμή που ο παίχτης ολοκληρώνει επιτυχώς το παιχνίδι) DeathScreenCanvas(διαχειρίζεται τη στιγμή που ο παίχτης ολοκληρώνει τη στιγμή που ο παίχτης πεθαίνει), τη ζωή του παίχτη και των τελικών εχθρών του 3^{ου} επιπέδου κ.α. Οι καμβάδες αυτοί είναι συγκεντρωμένοι σε ένα καμβά ο οποίος ονομάζεται General Canvas ο οποίος λειτουργεί κυρίως για την ανάλυση που θα έχουν όλοι οι υπόλοιποι καμβάδες, θα αναλυθεί το GeneralCanvas στο κεφάλαιο 5. Οι περισσότεροι καμβάδες ενεργοποιούνται όταν συμβαίνει μια συγκεκριμένη συνθήκη, όπως το WinScreenCanvas(ενεργοποιείται τη στιγμή που ο παίχτης ολοκληρώσει το παιχνίδι), DeathScreenCanvas(ενεργοποιείται τη στιγμή που ο παίχτης χάσει όλη του τη ζωή και πεθαίνει), PauseMenu(όταν πατηθεί το κουμπί της παύσης ενεργοποιείται), το Inventory(όταν πατηθεί το κουμπί του inventory), το DialogueCanvas(ενεργοποιείται τη στιγμή που ο παίχτης φτάνει στο NPC του καταστήματος πατώντας το κουμπί) το InteractTextCanvas(ενεργοποιείται όταν ο παίχτης μπει στο range των αντικειμένων όπως σεντούκια και πινακίδες προτρέποντάς τον μέσω κειμένου να πατήσει το κατάλληλο

κουμπι). Κάποιοι καμβάδες όμως υπάρχουν σε όλη τη διάρκεια του παιχνιδιού όπως το HealthUI(ζωή) του παίχτη και των τελικών εχθρών, και το ScoreCanvas(μετράει τα κέρματα που μάζεψε ο παίχτης κατά τη διάρκεια του παιχνιδιού).



Εικόνα 15: το ScoreCanvas.



Εικόνα 16 & 17: το PauseMenuCanvas.

2.10 Animations

Τα animations έχουν καθοριστικό ρόλο στη βελτίωση της συνολικής εμπειρίας παιχνιδιού στη σφαίρα των 2D RPG όπως το "Epic Quest". Λειτουργούν ως ένα οπτικό και διαδραστικό εργαλείο που δίνει ζωή στον κόσμο του παιχνιδιού, καθιστώντας το πιο καθηλωτικό και ελκυστικό για τους παίκτες. Τα κινούμενα σχέδια μπορούν να επικοινωνούν βασικές πληροφορίες, να μεταφέρουν συναισθήματα χαρακτήρων και να παρέχουν οπτική ανατροφοδότηση. Στο πλαίσιο του "Epic Quest", τα καλοφτιαγμένα κινούμενα σχέδια αποτελούν θεμελιώδες στοιχείο για τη δημιουργία ενός δυναμικού και σαγηνευτικού περιβάλλοντος παιχνιδιού. Χρησιμοποιούν όχι μόνο για να διασκεδάσουν

αλλά και στη βελτίωση των αλληλεπιδράσεων των παικτών και στην προσθήκη βάθους στο παιχνίδι.

2.10.1 Εχθροί

Όταν εμβαθύνουμε στον κόσμο των κινούμενων σχεδίων στο "Epic Quest", η εστίαση μετατοπίζεται προς τους εχθρούς του παιχνιδιού. Η ομαλότητα και η ρευστότητα αυτών των animation είναι υψίστης σημασίας για τη διασφάλιση ότι οι μάχες θα αισθάνονται απρόσκοπτες και ανταποκρίνονται. Τα ομαλά animation του εχθρού όχι μόνο κάνουν τις μάχες ελκυστικές, αλλά επηρεάζουν και τους μηχανισμούς του παιχνιδιού. Βοηθούν τους παίκτες να κατανοήσουν τις συμπεριφορές του εχθρού, να τηλεγραφούν επιθέσεις και να αντιδρούν στρατηγικά. Ένας εχθρός με καλά κινούμενα σχέδια μπορεί να είναι η διαφορά μεταξύ μιας απογοητευτικής συνάντησης και μιας συναρπαστικής μάχης βασισμένης σε δεξιότητες, καθιστώντας ζωτικής σημασίας την αφιέρωση της προσοχής στην ποιότητα και τη λεπτομέρεια των κινούμενων εικόνων του εχθρού στο "Epic Quest".

2.10.2 Συστατικά

Στο "Epic Quest", τα κινούμενα σχέδια εκτείνονται πέρα από απλώς χαρακτήρες και εχθρούς. Επεκτείνονται επίσης σε αντικείμενα, όπως σεντούκια και θησαυρούς. Το animation των αντικειμένων, όπως το animation που ανοίγει το σεντούκι, προσθέτει ένα στοιχείο προσμονής και ανταμοιβής στο παιχνίδι. Δημιουργεί μια αίσθηση ολοκλήρωσης και ενθουσιασμού όταν οι παίκτες αποκαλύπτουν το περιεχόμενο ενός σεντουκιού. Επιπλέον, αυτά τα κινούμενα σχέδια χρησιμεύουν ως οπτικό σύνθημα, καθοδηγώντας τους παίκτες σχετικά με τα αντικείμενα με τα οποία μπορούν να αλληλεπιδράσουν. Η κινούμενη εικόνα των αντικειμένων συμβάλλει στη συνολική εικόνα του παιχνιδιού και βοηθά στη διατήρηση της αφοσίωσης των παικτών προσθέτοντας ένα επίπεδο ικανοποίησης στις πτυχές εξερεύνησης και λεηλασίας του "Epic Quest". Συμπερασματικά, τα κινούμενα σχέδια, είτε αφορούν χαρακτήρες, εχθρούς ή αντικείμενα, είναι απαραίτητα για να γίνει το "Epic Quest" μια οπτικά εκπληκτική, συναισθηματικά συναρπαστική και απολαυστική εμπειρία 2D RPG.

3 Αποθήκευση και Φόρτωση προόδου

Ένα σύστημα αποθήκευσης και φόρτωσης είναι ζωτικής σημασίας για τη διατήρηση της προόδου και της αφοσίωσης των παικτών. Οι παίκτες επενδύουν χρόνο και προσπάθεια στο παιχνίδι και έχοντας τη δυνατότητα να σώσουν την πρόοδό τους διασφαλίζει ότι μπορούν να συνεχίσουν το ταξίδι τους από εκεί που το σταμάτησαν. Αυτό προάγει την αίσθηση της ολοκλήρωσης και τους παρακινεί να εξερευνήσουν περισσότερο τον κόσμο του παιχνιδιού. Ένα καλά σχεδιασμένο σύστημα αποθήκευσης δίνει στους παίκτες την ελευθερία να επιλέξουν πότε και πού θα αποθηκεύσουν το παιχνίδι τους. Αυτή η ευελιξία είναι απαραίτητη, καθώς επιτρέπει στους παίκτες να προσαρμόζουν τις συνεδρίες παιχνιδιών τους στα χρονοδιαγράμματα της πραγματικής ζωής τους. Είτε πρόκειται για μια γρήγορη αποθήκευση κατά τη διάρκεια ενός σύντομου διαλείμματος είτε για μια ολοκληρωμένη αποθήκευση στο τέλος μιας μακράς περιόδου παιχνιδιού, οι παίκτες εκτιμούν τον έλεγχο της εμπειρίας παιχνιδιού τους. Επίσης ένα σύστημα αποθήκευσης και φόρτωσης συμβάλλει στη διατήρηση και την ικανοποίηση των παικτών. Οι παίκτες είναι πιο πιθανό να επιστρέψουν και να προτείνουν ένα παιχνίδι που σέβεται τον χρόνο τους και παρέχει μια ομαλή εμπειρία χωρίς απογοητεύσεις. Οι θετικές εμπειρίες παικτών μπορούν να οδηγήσουν σε υψηλότερες κριτικές που μπορούν να οδηγήσουν στην άνοδο της δημοτικότητας για το "Epic Quest". Συμπερασματικά, η εφαρμογή ενός συστήματος αποθήκευσης και φόρτωσης στο "Epic Quest" δεν είναι απλώς μια ευκολία, αλλά ένα θεμελιώδες χαρακτηριστικό που ενισχύει την αφοσίωση των παικτών, τη διαχείριση κινδύνου και τη συνολική ικανοποίηση των παικτών. Είναι ένα βασικό συστατικό για τη δημιουργία μιας αξέχαστης και ευχάριστης εμπειρίας παιχνιδιού που θα έχει απήχηση στο κοινό και θα συμβάλει στην επιτυχία του παιχνιδιού.

3.1 Αποθήκευση

Η υλοποίηση της αποθήκευσης είχε ως στόχο ενός συστήματος το οποίο είναι αυτόματο που σημαίνει ότι ο παίχτης δεν χρειάζεται να αποθηκεύσει την πρόοδο του χειροκίνητα. Η κύρια λειτουργία του βασίζεται σε ένα script το οποίο ονομάζεται "SaveController"

το οποίο είναι υπεύθυνο για την αποθήκευση και φόρτωση της προόδου του χρήστη. Το script αυτό χρησιμοποιεί το μοτίβο σχεδίασης Singleton. Διασφαλίζει ότι υπάρχει μόνο μία παρουσία του SaveController σε όλο το παιχνίδι. Αυτό επιτυγχάνεται με τη μέθοδο Awake, όπου ελέγχει εάν υπάρχει ήδη ένα instance. Εάν συμβεί αυτό, το νέο στιγμιότυπο καταστρέφεται και εάν όχι, το νέο στιγμιότυπο ορίζεται ως ενεργό. Ακόμη η αποθήκευση γίνεται κρυπτογραφημένα διασφαλίζοντας ότι κανένας δεν μπορεί να πειράξει τα δεδομένα του παιχνιδιού προς όφελος του. Η υλοποίηση της μεθόδου αποθήκευσης αναλύεται παρακάτω καθώς και κάποια δευτερεύοντα κομμάτια από τον κώδικα αποθήκευσης θα παρατεθεί στο παράρτημα Α.

3.1.1 Κώδικας αποθήκευσης

```
3 references | Changed by thanossipsas21@gmail.com on Tuesday, October 10, 2023
public void SaveGame()
{
    SaveData saveData = new SaveData();

    if (ISaveControllerObjects != null)
    {
        ISaveControllerObjects = ISaveControllerObjects.Where(obj => obj != null).ToList();

        foreach (ISaveController saveControllerObj in ISaveControllerObjects)
        {
            if (saveControllerObj != null)
            {
                // Save the scene name
                saveData.savedSceneName = SceneManager.GetActiveScene().name;

                PlayerPrefs.SetString("SavedScene", SceneManager.GetActiveScene().name);
                Debug.Log("Saving data for: " + saveControllerObj.GetType().FullName);
                saveControllerObj.SaveGameData(ref saveData);
            }
            else
            {
                Debug.Log("ISaveController object is null.");
            }
        }

        dataHandler.Save(saveData);
    }
}
```

Εικόνα 18: Η μέθοδος SaveGame.

Τα αντικείμενα ISaveControllerObjects παίζουν κρίσιμο ρόλο στη διαδικασία αποθήκευσης. Είναι μια συλλογή αντικειμένων παιχνιδιού ή σεναρίων που υλοποιούν τη διεπαφή ISaveController και σκοπός τους είναι να παρέχουν προσαρμοσμένες μεθόδους για τη φόρτωση και την αποθήκευση δεδομένων παιχνιδιού. Αφού ληφθεί η λίστα των

αντικειμένων ISaveController, η κλάση SaveController επαναλαμβάνεται μέσω αυτών χρησιμοποιώντας έναν βρόχο foreach. Αυτή η επανάληψη είναι ένα κρίσιμο βήμα στη διαδικασία αποθήκευσης, επειδή επιτρέπει σε κάθε αντικείμενο ISaveController να συνεισφέρει τα συγκεκριμένα δεδομένα του όπως τοποθεσία, κέρματα, οι εχθροί που σκοτώθηκαν, τα σεντούκια που ανοίχθηκαν και τα αντικείμενα του inventory στη συνολική αποθήκευση του παιχνιδιού. Επίσης αποθηκεύεται το επίπεδο στο οποίο βρίσκεται ο παίχτης και τα αντικείμενα στο inventory του παίχτη μέσω της PlayerPrefs, η οποία είναι μια λειτουργία τύπου Unity που επιτρέπει την αποθήκευση και την ανάκτηση των προτιμήσεων του παίκτη, οι οποίες μπορούν να περιλαμβάνουν πληροφορίες κατάστασης παιχνιδιού. Είναι σημαντικό να αναφερθεί ότι για κάθε αντικείμενο ISaveController στον βρόχο, το σενάριο καλεί τη μέθοδο SaveGameData. Αυτό είναι το σημείο όπου αποθηκεύονται προσαρμοσμένα δεδομένα παιχνιδιού που σχετίζονται με το συγκεκριμένο αντικείμενο. Κάθε αντικείμενο ISaveController έχει τη δική του εφαρμογή της μεθόδου SaveGameData, προσαρμοσμένη στις ανάγκες του. Μπορεί να περιλαμβάνει διάφορες πληροφορίες με τα δεδομένα που έχει ο παίχτης να αποθηκεύσει στο παιχνίδι.

Τα δεδομένα αυτά αποθηκεύονται σε ένα construct το οποίο αποθηκεύει της πληροφορίες αυτές και βρίσκεται σε μια νέα κλάση η οποία ονομάζεται "SaveData".



```
[System.Serializable]
20 references | Moved from Inventory.Model.SaveData to SaveData by thanossipsas21@gmail.com on Thursday, September 21, 2023
public class SaveData
{
    public Vector3 playerPosition;
    public int coins;
    public SerializableDictionary<string, bool> chestsOpened;
    public SerializableDictionary<string, bool> enemiesKilled;
    public string savedSceneName;

    2 references | Changed by thanossipsas21@gmail.com on Tuesday, October 10, 2023
    public SaveData ()
    {
        savedSceneName = "Forest";
        coins = CoinCount.coins;
        chestsOpened = new SerializableDictionary<string, bool>();
        enemiesKilled = new SerializableDictionary<string, bool>();
    }
}
```

Εικόνα 19: Η κλάση SaveData.

Οι μεταβλητές αυτές είναι για την τοποθεσία του παίχτη Vector3. Για τα κέρματα είναι int, για τα σεντούκια που ανοίχθηκαν και τους εχθρούς που σκοτώθηκαν είναι SerializableDictionary τα οποία λειτουργούν ως έλεγχος εάν τα σεντούκια ανοίχθηκαν ή οι εχθροί σκοτώθηκαν και αν συμβαίνει αυτό τότε αποθηκεύονται με ένα key και value το οποίο αντιπροσωπεύει ότι ανοίχθηκαν τα σεντούκια ή σκοτώθηκαν οι εχθροί και το επίπεδο στο οποίο βρίσκεται ο παίχτης που είναι string. Ο κώδικας μαζί με την επεξήγηση θα παρατεθεί στο Παράρτημα Α.

Υπάρχει και ο μηχανισμός κρυπτογράφησης ο οποίος χρησιμοποιεί τη λέξη "mystical". Για την κρυπτογράφηση χρησιμοποιείται ο αλγόριθμος XOR για να κάνει την πράξη μεταξύ των χαρακτήρων της λέξης "mystical" σε δυαδική μορφή και των χαρακτήρων του αρχείου που είναι αποθηκευμένο σε JSON με την πράξη XOR. Αυτό έχει ως αποτέλεσμα το αρχείο να κρυπτογραφηθεί και για να αποκρυπτογραφηθεί θα πρέπει ο χρήστης να ξέρει τη λέξη και τον αλγόριθμο κρυπτογράφησης κάτι που είναι δύσκολο για τον μέσο χρήστη. Η μέθοδος που υλοποιεί την κρυπτογράφηση είναι η EncryptDecrypt. Η εξήγηση της μεθόδου θα παρατεθεί στο παράρτημα Α.

```
private readonly string encryptionCodeWord = "mystical";
```

Εικόνα 20: Η λέξη κρυπτογράφησης.

```
//decrypt data  
if(useEncryption){  
    dataToLoad = EncryptDecrypt(dataToLoad);  
}
```

Εικόνα 21: η εντολή κρυπτογράφησης-αποκρυπτογράφησης.

3.2 Φόρτωση Προόδου

Η Φόρτωση προόδου είναι σημαντική για τον παίχτη διότι ο παίχτης όταν αποθηκεύει την πρόοδό του, επιθυμεί να συνεχίσει από εκεί που σταμάτησε την τελευταία φορά. Για αυτό το λόγο το script "SaveController" το οποίο εκτός από την αποθήκευση διαχειρίζεται και την φόρτωση της προόδου. Η φόρτωση ουσιαστικά ανακτά τα

αποθηκευμένα δεδομένα από τον φάκελο στον οποίο βρίσκονται στην κάθε συσκευή. Περαιτέρω ανάλυση θα γίνει παρακάτω.

3.2.1 Κώδικας Φόρτωσης Προόδου

```
2 references | Changed by paokarateam01@gmail.com on Thursday, September 21, 2023
public void LoadGame(){

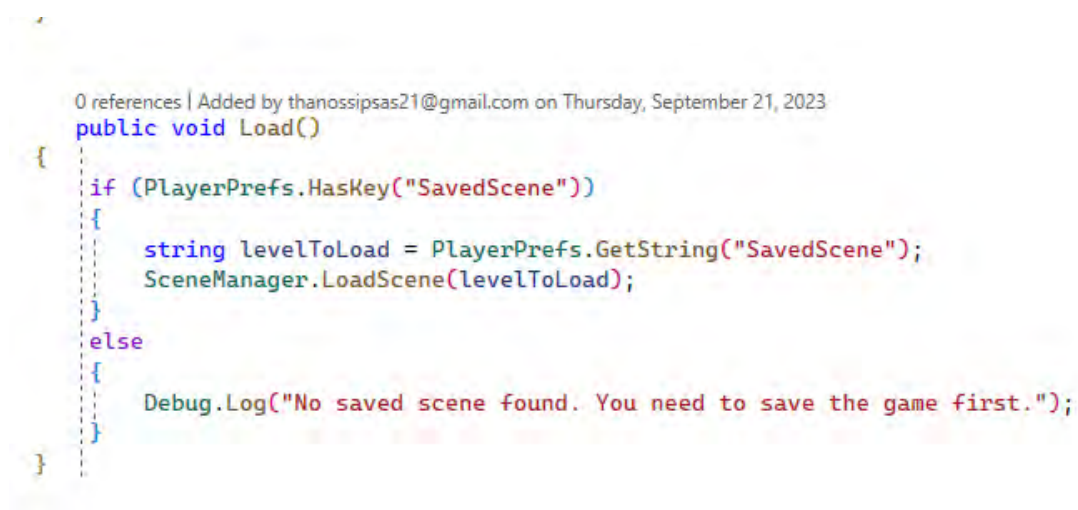
    //load any saved data from a file
    try
    {
        this.saveData = dataHandler.Load();
    }
    catch (Exception e)
    {
        Debug.Log("An error occurred while loading data: " + e.Message);
    }
    if (this.saveData == null && initializedataifnull)
    {
        NewGame();
    }

    // if no data can be loaded dont continue start a new game
    if(this.saveData==null)
    {
        Debug.Log("No data found.A new game needs to be started before data can be loaded");
        return;
    }
    if (ISaveControllerObjects != null)
    {
        foreach (ISaveController saveControllerObj in ISaveControllerObjects)
        {
            saveControllerObj.LoadGameData(saveData);
        }
    }
}
```

Εικόνα 22: Η μέθοδος LoadGame.

Η μέθοδος αυτή φορτώνει την πρόοδο από τον φάκελο στον οποίο αποθήκευσε ο παίχτης ελέγχοντας πρώτα αν υπάρχει ο φάκελος αυτός, διότι εάν δεν υπάρχει ξεκινάει καινούργιο παιχνίδι, αλλιώς φορτώνει όλες τις μεταβλητές που αποθηκεύτηκαν.

Επίσης είναι σημαντικό να αναφερθεί ότι με το κουμπί Load το οποίο βρίσκεται στο μενού, εφόσον πατηθεί, ανακτά το επίπεδο στο οποίο βρίσκεται ο παίχτης χρησιμοποιώντας το PlayerPrefs.



Εικόνα 23: Η μέθοδος Load για το κουμπί Load.

Συγκεκριμένα ψάχνει, εάν υπάρχει αποθηκευμένο, μέσω του PlayerPrefs το key SavedScene και εάν υπάρχει, φορτώνει το επίπεδο το οποίο είναι αποθηκευμένο στη μεταβλητή αυτή. Στην αντίθετη περίπτωση δεν φορτώνει τίποτα.

4 Σχεδιασμός εχθρών και προγραμματιστική λογική

Στο κόσμο του σχεδιασμού βιντεοπαιχνιδιών, οι εχθροί παίζουν καθοριστικό ρόλο στη διαμόρφωση της εμπειρίας του παίκτη. Κάθε επίπεδο εισάγει μια ποικιλία αντιπάλων, ο καθένας με μοναδικές συμπεριφορές και χαρακτηριστικά. Αυτοί οι εχθροί δεν είναι απλώς στατικά εμπόδια, αλλά δυναμικές οντότητες, προσεκτικά προγραμματισμένες για να εμπλέκουν τους παίκτες σε συναρπαστικές μάχες. Η βασική λογική που διέπει τις ενέργειές τους περιστρέφεται γύρω από μια αδυσώπητη καταδίωξη του παίκτη, χτυπώντας με ακρίβεια κατά την επαφή, προκαλώντας ζημιά. Αυτό το κεφάλαιο εμβαθύνει στις περιπλοκές του σχεδιασμού του εχθρού και στη λογική προγραμματισμού που ζωντανεύει αυτούς τους αντιπάλους, εξετάζοντας τις κινήσεις τους που βασίζονται στην AI, τα μοτίβα επιθέσεων τους και την ιεραρχία των αφεντικών, οι οποίοι έχουν τον δικό τους τρόπο επίθεσης θέτοντας μια ακόμη μεγαλύτερη απειλή στο ταξίδι του παίκτη στον κόσμο του παιχνιδιού.

4.1 Σχεδιασμός εχθρών

Στο πλαίσιο της ανάπτυξης παιχνιδιών, ο σχεδιασμός του εχθρού είναι ένας έντεχνος συνδυασμός δημιουργικότητας και ικανότητας προγραμματισμού. Κατά τη δημιουργία των εχθρών για κάθε επίπεδο, η επιλογή του οπτικού σχεδιασμού τους είναι μια κρίσιμη πτυχή. Αυτά τα σχέδια δεν είναι αυθαίρετα. Αντίθετα, έχουν επιλεγεί προσεκτικά για να εμποτίσουν κάθε επίπεδο με μια μοναδική ατμόσφαιρα και εμφάνιση. Είτε πρόκειται για τα απόκοσμα πλάσματα που κατοικούν σε ένα σκοτεινό, μυστηριώδες σπήλαιο είτε για τα γλοιώδη σφαιρικά πλάσματα που κατοικούν στο δασικό περιβάλλον τους. Αυτή η προσεκτική εξέταση επεκτείνεται και στη λογική που διέπει τη συμπεριφορά τους. Κάθε εχθρός είναι προγραμματισμένος να καταδιώκει ανελέητα τον παίκτη και όταν έρθει σε επαφή, τον χτυπάει προκαλώντας ζημιά. Οι περιπλοκές των μοτίβων επίθεσης ποικίλουν, προσθέτοντας πολυπλοκότητα στην εμπειρία παιχνιδιού. Επιπλέον, η ιεραρχία του παιχνιδιού περιλαμβάνει τρομερά αφεντικά, μερικά εξοπλισμένα με πολλαπλές καταστροφικές επιθέσεις, εισάγοντας έτσι στρατηγικά στοιχεία και εντείνοντας έτσι τις

προκλήσεις που πρέπει να κατακτήσει ο παίχτης. Παρακάτω, θα εμβαθύνουμε στις πολυπλοκότητες της λογικής προγραμματισμού, διερευνώντας τις διαδικασίες σκέψης και τις τεχνικές λεπτομέρειες που κάνουν αυτούς τους αντιπάλους αναπόσπαστο μέρος της καθηλωτικής διερεύνησης των παικτών του κόσμου.

4.2 Προγραμματιστική λογική

Η βάση του κόσμου του παιχνιδιού είναι μια λεπτομερώς ρυθμισμένη λογική προγραμματισμού που ζωντανεύει τους εχθρούς. Όταν ένας εχθρός ανιχνεύσει την παρουσία του παίκτη, αρχίζει η καταδίωξη. Αυτό επιτυγχάνεται μέσω της μεθόδου «FixedUpdate» που καθοδηγούν τους εχθρούς να μετακινηθούν στη θέση του παίκτη, παρακολουθώντας ακούραστα κάθε του κίνηση. Επίσης θα εξηγηθεί η οπτική αναπαράσταση της ζημιάς του παίκτη η οποία υπάρχει στην μέθοδο «OnCollisionEnter2D».

4.2.1 Κώδικας καταδίωξης και υπόδειξης ζημιάς στον παίκτη.

```
Unity Message | 0 references | Error: The auth token expired. User: paokarateam01@gmail.com.
void FixedUpdate() {

    //if enemy detects player
    if (damageableCharacter.Targetable && detectionZone.detectedObjs.Count > 0)
    {

        //calculate direction to target object
        Vector2 direction = (detectionZone.detectedObjs[0].transform.position - transform.position).normalized;
        if (!AudioSource.isPlaying)
        {
            AudioSource.Play();
        }

        anim.SetBool("isMoving", true);
        //move towards detected target
        rb.AddForce(direction * movespeed * Time.deltaTime);
        if (direction.x > 0)
        {
            spriteRenderer.flipX = false;
        }
        else if (direction.x < 0)
        {
            spriteRenderer.flipX = true;
        }
    }
    else
    {
        anim.SetBool("isMoving", false);
        AudioSource.Stop();
    }
}
```

Εικόνα 24: Η μέθοδος FixedUpdate.

Η μέθοδος «FixedUpdate» χρησιμοποιήθηκε διότι καλείται σε σταθερά χρονικά διαστήματα σε αντίθεση με την «Update» μέθοδο, η οποία καλείται μία φορά ανά frame. Στην μέθοδο μέσα συμβαίνει το εξής, εάν ο εχθρός εντοπίσει τον παίκτη τότε ο παίχτης με την εντολή «rb.AddForce» όπου το rb είναι η μεταβλητή για το rigidbody το

οποίο λειτουργεί ως το σώμα του εχθρού και το AddForce το οποίο ελέγχει το rigidbody του εχθρού και του προσθέτει κίνηση, η οποία είναι η κατεύθυνση(direction) πολλαπλασιασμένη με την ταχύτητα(movespeed) με την οποία προχωράει ο εχθρός, πολλαπλασιασμένο με τον χρόνο(Time.deltaTime) που υπάρχει μεταξύ των frames. Η κατεύθυνση υπολογίζεται εφόσον εντοπίσει ο εχθρός την θέση του παίχτη και την απόσταση που έχουν μεταξύ τους και μαζί με την ταχύτητα επιτυγχάνεται το κυνήγι από τον εχθρό στον παίχτη. Ακόμη εάν ο παίχτης αλλάξει κατεύθυνση είτε δεξιά είτε αριστερά το sprite του εχθρού με την εντολή «spriteRenderer.flipX» που παίρνει δύο τιμές true και false στο οποίο εάν είναι true τότε είναι δεξιά το sprite του εχθρού ενώ εάν είναι false τότε είναι αριστερά συνεχίζοντας το κυνήγι του παίχτη. Επίσης υπάρχει και το animation του εχθρού που κινείται το οποίο ονομάζεται «isMoving» και είναι Boolean όπου εάν ο εχθρός εντοπίσει το παίχτη τότε το Boolean είναι true αλλιώς είναι false. Η μέθοδος αυτή είναι ίδια για τους εχθρούς του πρώτου και του δεύτερου επιπέδου.

Είναι σημαντικό να αναφερθεί ότι τα δύο Bosses έχουν διαφορετικό μηχανισμό καταδίωξης από ότι οι εχθροί στο πρώτο και δεύτερο επίπεδο που θα αναλυθεί αμέσως από κάτω.



```

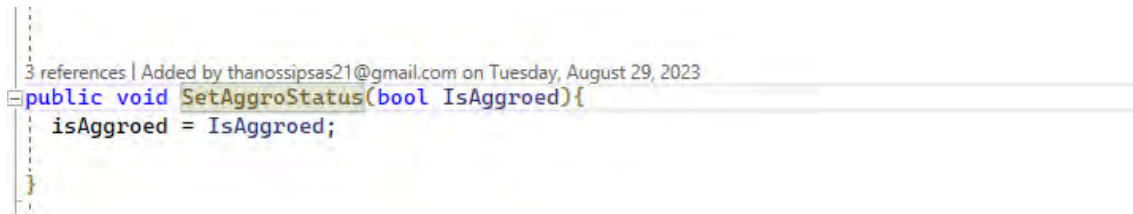
Unity Message | 0 references | Changed by thanossipsas21@gmail.com on Monday, October 9, 2023
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.tag == tagTarget)
    {
        BossHealthUI.SetActive(true);
        bossFightMusicStart.PlayAudio();
        flag=true;
        boss.SetAggroStatus(true);
        detectedObjs.Add(collision);
        anim.SetBool("inAggroRange", true);
        if (!AudioSource.isPlaying)
        {
            AudioSource.Play();
        }
    }
}

Unity Message | 0 references | Changed by paokarateam01@gmail.com on Saturday, September 9, 2023
private void OnTriggerExit2D(Collider2D collision)
{
    if (collision.gameObject.tag == tagTarget)
    {
        boss.SetAggroStatus(false);
        detectedObjs.Remove(collision);
        anim.SetBool("inAggroRange", false);
        AudioSource.Stop();
    }
}

```

Εικόνα 25: Οι μέθοδοι OnTriggerEnter2D και OnTriggerExit2D.

Οι μέθοδοι αυτοί λειτουργούν ως εξής, στην «OnTriggerEnter2D» εάν εντοπιστεί το collision με το tagtarget που είναι του παίχτη τότε προσθέτει το collision αυτό και ενεργοποιείται το animation «inAggroRange» κάνοντας το true όπως και την SetAggroStatus μέθοδο. Στην αντίθετη περίπτωση γίνονται false.



```
3 references | Added by thanossipsas21@gmail.com on Tuesday, August 29, 2023
public void SetAggroStatus(bool IsAggroed){
    isAggroed = IsAggroed;
}
```

Εικόνα 26: Η μέθοδος SetAggroStatus

Η μέθοδος αυτή έχει το Boolean Isaggroed μεταβλητή η οποία αναλόγως εάν ο παίχτης έχει εντοπιστεί η όχι, γίνεται true η false αντίστοιχα προκαλώντας, μαζί με το animation «inAggroRange», τη καταδίωξη η μη του παίχτη.



```
//if any character collides with player... damage
Unity Message | 0 references | Changed by thanossipsas21@gmail.com on Saturday, September 23, 2023
void OnCollisionEnter2D(Collision2D col)
{
    if(col.gameObject.tag == "Player"){
        Collider2D collider = col.collider;
        IDamageable damageable = col.collider.GetComponent<IDamageable>();
        if (damageable != null)
        {
            healthText.SetEnemyDamageText(damage);
            Vector3 parentPosition = transform.position;
            Vector2 direction = (Vector2)(col.gameObject.transform.position - transform.position).normalized;
            Vector2 knockBack = direction * knockBackForce;
            damageable.OnHit(damage, knockBack);
        }
    }
}
```

Εικόνα 27: Η μέθοδος OnCollisionEnter2D.

Η μέθοδος «OnCollisionEnter2D» ευθύνεται για την ζημιά η οποία θα αναλυθεί σε επόμενη υποενότητα και για την υπόδειξη της. Η υπόδειξη της ζημιάς γίνεται με την εντολή «healthText.SetEnemyDamageText(damage)» όπου γίνεται η οπτικοποίηση της ζημιάς που παίρνει ο παίχτης από τον εχθρό που γίνεται ανάλογα με τη ζημιά που προκαλεί ο εχθρός σε κάθε παίχτη, το οποίο γίνεται με την χρήση της μεθόδου «SetPlayerDamageText».


```
1 reference | Renamed from SetDamageText to SetPlayerDamageText by thanossipsas21@gmail.com on Saturday, September 23, 2023
public void SetPlayerDamageText(float damage)
{
    textMesh.text = damage.ToString(); // Display the damage as text
}
```

Εικόνα 28: Η μέθοδος SetPlayerDamageText.

Η μέθοδος αυτή μετατρέπει τη ζημιά που εφαρμόζεται από τον εχθρό στο παίκτη σε string, η οποία συνεργάζεται με την μέθοδο «Update» και εμφανίζει την ζημιά σαν ένα αριθμό, ο οποίος ανεβαίνει προς τα πάνω για λίγο με χρώμα μαύρο, δίνοντας στον παίκτη οπτική ανατροφοδότηση για το πόση ζημιά πήρε και μετά εξαφανίζεται.

```
void Update()
{
    timeElapsed += Time.deltaTime;
    rt.position += floatDirection * floatSpeed * Time.deltaTime;
    textMesh.color = new Color(startingcolor.r, startingcolor.g, startingcolor.b, 1 - (timeElapsed / timeToLive));
    if (timeElapsed > timeToLive) {
        Destroy(gameObject);
    }
}
```

Εικόνα 29: Η μέθοδος Update.

4.3 Μηχανισμός ζημιάς εχθρών σε παίκτη

Ο μηχανισμός ζημιάς από εχθρούς σε παίκτη αποτελεί ένα αναπόσπαστο κομμάτι για την υλοποίηση του παιχνιδιού. Ένα από τα βασικά στοιχεία αυτής της λογικής είναι η μέθοδος «OnCollisionEnter2D», η οποία παίζει κεντρικό ρόλο στη δυναμική της μάχης. Κατά την επαφή με τον παίκτη, αυτή η μέθοδος διευκολύνει την απρόσκοπτη αλληλεπίδραση μεταξύ των εχθρών και του χαρακτήρα του παίκτη. Εδώ είναι πιο εμφανής η ακρίβεια του εχθρού. Όταν ο εχθρός αγγίζει τον παίκτη, προκαλεί ένα συμβάν που οδηγεί στον παίκτη να υποστεί ζημιά. Αυτή η αλληλεπίδραση δεν είναι απλώς μια μονόπλευρη υπόθεση. Αντίθετα, υπογραμμίζει τη δυναμική φύση του παιχνιδιού, προσφέροντας στους παίκτες την ευκαιρία να αναπτύξουν στρατηγική και να κυριαρχήσουν στην τέχνη της υπεκφυγής και της αντεπίθεσης. Η καταδίωξη και η εφαρμογή της ζημιάς στον παίκτη μέσω της μεθόδου «OnCollisionEnter2D» δημιουργεί μια ομαλή εμπειρία παιχνιδιού, όπου κάθε συνάντηση εχθρού γίνεται μια δυναμική

πρόκληση που απαιτεί δεξιότητες και στρατηγική. Επίσης θα αναφερθεί και θα αναλυθεί η επίθεση των τελικών εχθρών στο τρίτο και τελευταίο επίπεδο διότι ο μηχανισμός επίθεσης και εφαρμογής ζημιάς είναι διαφορετικός.

4.3.1 Κώδικας μηχανισμού ζημιάς από εχθρούς σε παίχτη

Η μέθοδος «OnCollisionEnter2D» όπως φαίνεται στην εικόνα 24 διαχειρίζεται τη ζημιά από εχθρούς σε παίχτη. Συγκεκριμένα εάν η επαφή που κάνει ο εχθρός στον παίχτη αναγνωρίσει τον παίχτη με το tag του, τότε παίρνει το «IDamagable» component το οποίο είναι ένα interface και εφαρμόζεται η ζημιά στο παίχτη. Επίσης εάν η μεταβλητή damageable που δείχνει εάν ο παίχτης μπορεί να πάρει ζημιά και δεν έχει πεθάνει (δηλαδή να μην είναι null), τότε γίνεται η μέθοδος «OnHit» στον παίχτη που του περνάει τη ζημιά και την απώθηση προς τα πίσω που δημιουργείται κατά την επαφή παίχτη και εχθρού. Αυτή η μέθοδος ζημιάς «OnCollisionEnter2D» εφαρμόζεται στους εχθρούς του πρώτου και του δεύτερου επιπέδου.

```
//multiple
6 references | Changed by paokarateam01@gmail.com on Saturday, May 27, 2023
public void OnHit(float damage, Vector2 knockback)
{
    if(!Invincible){
        Health -= damage;
        rb.AddForce(knockback, ForceMode2D.Impulse);
        if(canTurnInvincible){
            //activate invincibility timer
            Invincible = true;
        }
    }
}

//single
2 references | Added by paokarateam01@gmail.com on Saturday, April 8, 2023
public void OnHit(float damage)
{
    if(!Invincible){
        Health -= damage;
        if(canTurnInvincible){
            //activate invincibility timer
            Invincible = true;
        }
    }
}
```

Εικόνα 30: η μέθοδος OnHit για πολλαπλούς και έναν εχθρό.

Η μέθοδος «OnHit» διαχειρίζεται τη ζημιά που εφαρμόζεται στον παίχτη όταν έρχεται σε επαφή με τον εχθρό. Οι δύο μέθοδοι είναι παρόμοιες και εκτελούν την ίδια λειτουργία με τη ειδοποιό διαφορά ότι η πρώτη μέθοδος χρησιμοποιεί την απώθηση προς τα πίσω. Στην παρούσα περίπτωση χρησιμοποιείται η πρώτη μέθοδος. Αναλυτικότερα, σε περίπτωση που ο παίχτης δεν έχει invincibility, το οποίο ενεργοποιείται αφότου ο παίχτης έχει πάρει ζημιά προστατεύοντάς τον από αλλεπάλληλα χτυπήματα, τότε

εφαρμόζεται η ζημιά, ενεργοποιείται το invincibility για ένα συγκεκριμένο χρονικό διάστημα του παίχτη και γίνεται η απώθησή του προς τα πίσω.

```
//if player inside the boss' attack distance...
Unity Message | 0 references | Changed by thanossipsas21@gmail.com on Monday, September 4, 2023
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.tag == tagTarget)
    {
        boss.SetAttackDistanceBool(true);
        detectedObjs.Add(collision);
    }
}

Unity Message | 0 references | Changed by thanossipsas21@gmail.com on Monday, September 4, 2023
private void OnTriggerExit2D(Collider2D collision)
{
    if (collision.gameObject.tag == tagTarget)
    {
        boss.SetAttackDistanceBool(false);
        detectedObjs.Remove(collision);
    }
}
```

Εικόνα 31: Ο έλεγχος απόστασης για να επιτεθούν τα αφεντικά.

Για να εξηγηθεί ο μηχανισμός ζημιάς θα πρέπει να αναλυθεί πως κάνουν επίθεση τα αφεντικά. Εάν το collision του παίχτη με το tagTarget εντοπιστεί μπαίνοντας στο range των αφεντικών τότε προστίθεται στα collisions και σταματάει ετοιμάζοντας την επίθεση του. Δίνοντας έτσι και χρόνο στον παίχτη να ξεφύγει έτσι ώστε να υπάρχει ισορροπία μεταξύ του χρόνου επίθεσης και του χρόνου αντίδρασης του παίχτη για την αποφυγή της επίθεσης. Χρησιμοποιείται η μέθοδος SetAttackDistanceBool η οποία θα αναλυθεί στη συνέχεια.

```
if (!isAttacking && timeSincePlayerEnteredRange >= timeBetweenAttacks)
{
    rand=Random.Range(0,2);
    if(rand==0){
        Debug.Log("Starting Attack1");
        anim.SetBool("isAttacking", true);
        isAttacking = true;
        attackStartTime = Time.time;
    }
    if(rand==1){
        Debug.Log("Starting Attack2");
        anim.SetBool("isAttacking2", true);
        isAttacking = true;
        attackStartTime = Time.time;
    }
}
```

Εικόνα 32: Ο μηχανισμός και η εναλλαγή των επιθέσεων του πρώτου αφεντικού(Dark Wizard).

Ο μηχανισμός επίθεσης του πρώτου αφεντικού λειτουργεί ως εξής. Λόγω της ύπαρξης δύο επιθέσεων που έχει, χρησιμοποιείται η εντολή rand. Αυτό δημιουργεί ένα τυχαίο ακέραιο rand που είναι είτε 0 είτε 1. Χρησιμοποιείται για να αποφασίσει ποια επίθεση θα εκτελέσει το αφεντικό.

Στη συνέχεια, ο κώδικας ελέγχει την τιμή του rand και ορίζει τις κατάλληλες παραμέτρους boolean και ενεργοποιεί τον animator του boss για να ξεκινήσει το αντίστοιχο animation επίθεσης, το οποίο προκαλεί ζημιά καλώντας την μέθοδο «OnHit». Μετά το πέρας της επίθεσης, εάν ολοκληρώθηκε, ελέγχεται η μεταβλητή isAttacking που είναι Boolean και αυτή, αν έχει πάρει την τιμή true για να γίνει η επίθεση, τότε η επίθεση σταματάει. Επίσης αν βγει ο παίχτης από το range της επίθεσης του αφεντικού, τότε σταματάει και πάλι η επίθεση και συνεχίζει να καταδιώκει τον παίχτη.

```
if (isAttacking && Time.time >= attackStartTime + attackDuration)
{
    if(rand==0){
        Debug.Log("Stopping Attack1");
        anim.SetBool("isAttacking", false);
        isAttacking = false;
        timeSincePlayerEnteredRange = 0f;
        boss.stateMachineManager.ChangeBossState(boss.idleState);
    }
    if(rand==1){
        Debug.Log("Stopping Attack2");
        anim.SetBool("isAttacking2", false);
        isAttacking = false;
        timeSincePlayerEnteredRange = 0f;
        boss.stateMachineManager.ChangeBossState(boss.idleState);
    }
}

exitTimer = 0f;
else
{
    if (isAttacking)
    {
        Debug.Log("Stopping Attack (out of range)");
        if(rand==0){
            boss.StopAttack();
        }
        if(rand==1){
            boss.StopAttack2();
        }
        timeSincePlayerEnteredRange = 0f; // Reset the timer when the attack is stopped
    }
}
```

Εικόνα 33: Μηχανισμός για διακοπή της επίθεσης.

Είναι σημαντικό να επισημανθεί ότι αυτός ο μηχανισμός που περιέχει 2 επιθέσεις, ισχύει μόνο για το ένα αφεντικό(Dark Wizard). Για το άλλο αφεντικό(Fire Wizard) λόγω της ύπαρξης μιας μόνο επίθεσης δεν υπάρχει η εντολή rand για την τυχαιοποίηση των επιθέσεων, αλλά λειτουργεί παρόμοια σαν μηχανισμός με το πρώτο αφεντικό.

```

3 references | Added by thanossipsas21@gmail.com on Tuesday, August 29, 2023
public void SetAttackDistanceBool(bool isWithinAttackDistance){
    isInAttackDistance = isWithinAttackDistance;
}

```

Εικόνα 34: Η μέθοδος SetAttackDistanceBool.

Η μέθοδος αυτή έχει το Boolean isWithinAttackDistance μεταβλητή η οποία αναλόγως εάν ο παίχτης έχει εντοπιστεί για να επιτεθεί στο αφεντικό ή όχι γίνεται true ή false αντίστοιχα, προκαλώντας έτσι το animation «PlayerInAttackRange» το οποίο προκαλεί την επίθεση στον παίχτη.

```

5 references | Changed by thanossipsas21@gmail.com on Thursday, July 27, 2023
public float Health
{
    set
    {
        if (currentHealth.Value < maxHealth)
        {
            animator.SetTrigger("hit");
            RectTransform textTransform = Instantiate(healthText).GetComponent<RectTransform>();
            textTransform.transform.position = Camera.main.WorldToScreenPoint(gameObject.transform.position);
            Canvas canvas = GameObject.FindObjectOfType<Canvas>();
            if (canvas != null)
            {
                textTransform.SetParent(canvas.transform);
            }
        }

        currentHealth.Value = value;

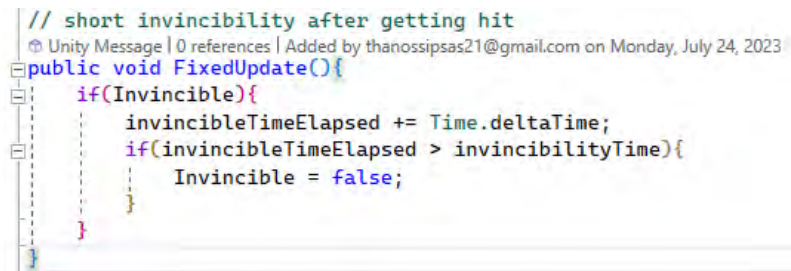
        if (currentHealth.Value <= 0)
        {
            animator.SetTrigger("Death");
            Targetable = false;
        }
    }

    get
    {
        return currentHealth.Value;
    }
}

```

Εικόνα 35: η διαχείριση της ζωής του παίχτη

Η διαχείριση της ζωής του παίχτη γίνεται από την ιδιότητα Health η οποία είναι ιδιότητα float με setter και getter. Συγκεκριμένα στον setter, εάν το currentHealth.Value που είναι η ζωή του παίχτη αυτήν την στιγμή, τότε ενεργοποιείται το animation hit του παίχτη. Στην περίπτωση που το currentHealth.Value είναι μικρότερο ή ίσο του 0, ενεργοποιείται το animation του θανάτου(Death) του παίχτη. Στον getter επιστρέφει το currentHealth.Value το οποίο δείχνει πόση ζωή έχει ο παίχτης αυτήν την στιγμή. Επίσης γίνεται οπτική ανατροφοδότηση της ζωής του παίχτη όταν έχει πλήρη ζωή, όταν χάνει ζωή και όταν δεν έχει καθόλου. Η εικόνα της ζωής του παίχτη είναι στην εικόνα 14.



```
// short invincibility after getting hit
Unity Message | 0 references | Added by thanossipsas21@gmail.com on Monday, July 24, 2023
public void FixedUpdate(){
    if(Invincible){
        invincibleTimeElapsed += Time.deltaTime;
        if(invincibleTimeElapsed > invincibilityTime){
            Invincible = false;
        }
    }
}
```

Εικόνα 36: Η μέθοδος FixedUpdate.

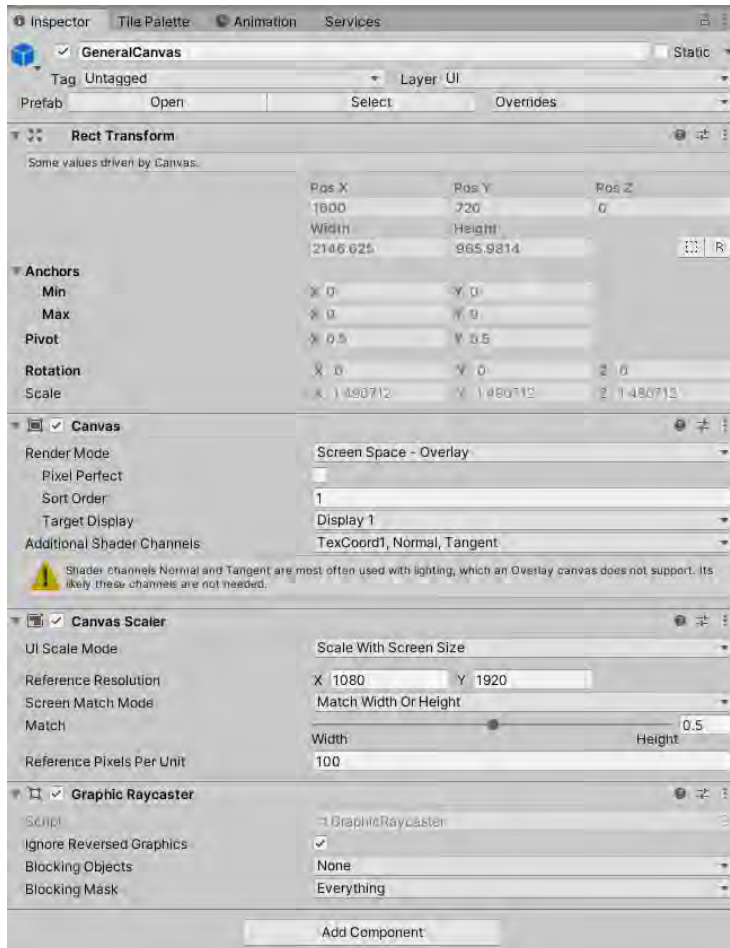
Είναι αναγκαίο να αναφερθεί ότι στην περίπτωση που ο παίχτης δεχτεί ζημιά από τους εχθρούς εκτελείται η μέθοδος FixedUpdate, η οποία διαχειρίζεται το invincibility, το οποίο ενεργοποιείται για ένα ορισμένο χρονικό διάστημα μόλις ο παίχτης δεχτεί ζημιά, προστατεύοντας τον από συνεχόμενα χτυπήματα των εχθρών.

5 Αναλύσεις Οθόνης

Οι αναλύσεις στο πλαίσιο της ανάπτυξης παιχνιδιών παίζουν καθοριστικό ρόλο στη διασφάλιση της βέλτιστης οπτικής εμπειρίας για τους παίκτες. Στο παιχνίδι 2D RPG σε στυλ pixel από πάνω προς τα κάτω, "Epic Quest", είναι απαραίτητο να ληφθεί υπόψη η σημασία των αναλύσεων σε διάφορες οθόνες, καθώς μπορεί να επηρεάσει σημαντικά τη συνολική ποιότητα και προσβασιμότητα του παιχνιδιού. Οι αναλύσεις επηρεάζουν την προβολή του παιχνιδιού. Όταν προσαρμόζεται σωστά σε διάφορες οθόνες, το παιχνίδι μπορεί να διατηρήσει την ποιότητα του, επιτρέποντας στους παίκτες να απορροφηθούν πλήρως στον φανταστικό κόσμο του "Epic Quest". Οι υψηλότερες αναλύσεις σε μεγαλύτερες οθόνες μπορούν να προσφέρουν μεγαλύτερη λεπτομέρεια και σαφήνεια, βελτιώνοντας τη συνολική εμπειρία του παίκτη. Αντίθετα, οι χαμηλότερες αναλύσεις σε μικρότερες οθόνες μπορούν να διατηρήσουν την απόδοση χωρίς να διακυβεύεται η γοητεία του παιχνιδιού. Διαφορετικοί παίκτες έχουν διαφορετικές συσκευές, που κυμαίνονται από υπολογιστές παιχνιδιών προηγμένης τεχνολογίας έως φορητές συσκευές χαμηλής κατανάλωσης. Βελτιστοποιώντας την ανάλυση του παιχνιδιού για μια ποικιλία οθονών, διασφαλίζεται η συμπερίληψη και η προσβασιμότητα, καλύπτοντας ένα ευρύτερο κοινό. Για παίκτες με παλαιότερο υλικό ή περιορισμένες επιλογές μεγέθους οθόνης, η προσαρμογή των αναλύσεων μπορεί να σημαίνει τη διαφορά μεταξύ μιας ευχάριστης και ευχάριστης εμπειρίας ή μιας γεμάτης τεχνικά προβλήματα. Οι αναλύσεις επηρεάζουν άμεσα την απόδοση και τον ρυθμό καρέ του παιχνιδιού. Ακόμη σε οθόνες υψηλής ανάλυσης, η απόδοση περισσότερων pixel μπορεί να καταπονήσει το υλικό και να μειώσει τους ρυθμούς καρέ, προκαλώντας ενδεχομένως μια απογοητευτική εμπειρία ή εμπειρία που δεν μπορεί να παίξει. Επίσης για συσκευές χαμηλού επιπέδου, η μείωση της ανάλυσης μπορεί να βοηθήσει στη διατήρηση ενός ομαλού ρυθμού καρέ, καθιστώντας το παιχνίδι προσβάσιμο σε ένα ευρύτερο φάσμα παικτών. Για αυτό σε αυτό κεφάλαιο θα αναλυθεί η επίτευξη της ανάλυσης σε διαφορετικές οθόνες καθώς και η ανάλυση των αντικειμένων σε διαφορετικές οθόνες με χρήση κώδικα.

5.1 Αρχική ανάλυση παιχνιδιού

Η ανάλυση του παιχνιδιού γίνεται από ένα καμβά που διαχειρίζεται ένα σύνολο καμβάδων, που ονομάζεται GeneralCanvas.

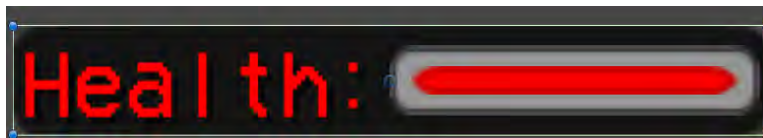
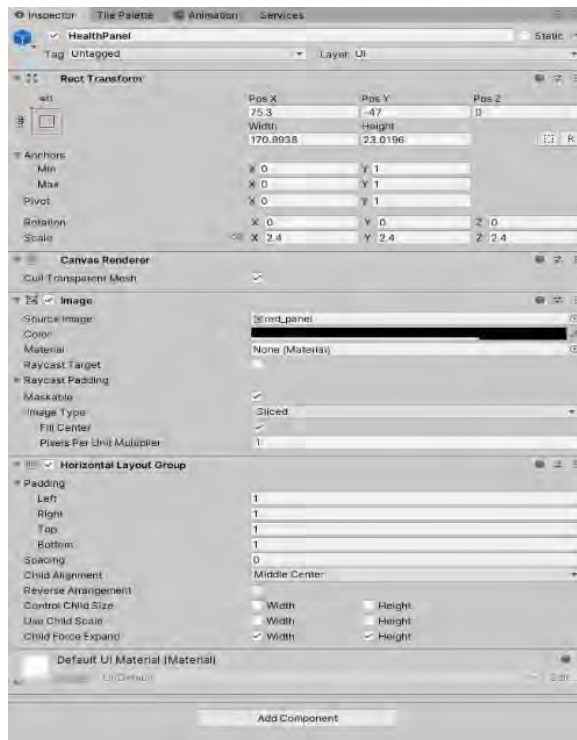


Εικόνα 37: Το GeneralCanvas.

Το συγκεκριμένο Canvas επειδή διαχειρίζεται όλους τους καμβάδες μπορεί και να διαχειριστεί και την ανάλυση τους πόσο θα είναι. Αυτό γίνεται μέσω του Canvas Scaler το οποίο προσαρμόζει το canvas ώστε να χωρέσει σε μια συγκεκριμένη ανάλυση. Στην συγκεκριμένη περίπτωση είναι 1920x1080, η οποία θεωρείται η full high definition ανάλυση. Η ανάλυση αυτή επιλέχθηκε ως βάση, διότι οι περισσότερες κινητές συσκευές στη σημερινή εποχή διαθέτουν τέτοια ανάλυση.

Όμως στις περιπτώσεις μικρότερων ή μεγαλύτερων αναλύσεων, χρησιμοποιήθηκε η αγκύρωση καμβάδων και UI αντικειμένων σε συγκεκριμένο σημείο. Με αυτόν τον τρόπο επιτυγχάνεται η σταθερή τοποθεσία όλων των καμβάδων και αντικειμένων σε μια συγκεκριμένη θέση. Σε αντίθετη περίπτωση θα υπήρχαν θέματα επικάλυψης

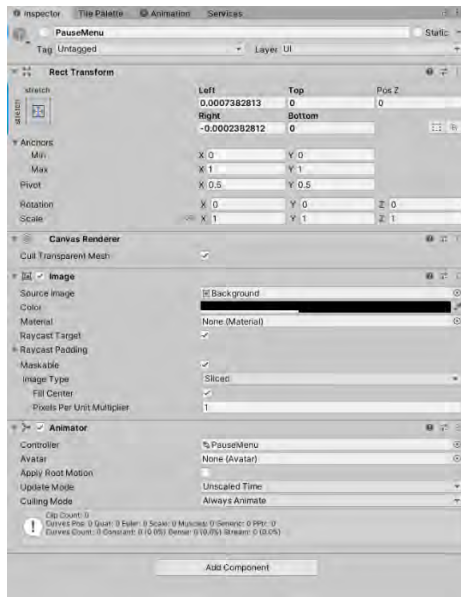
αντικειμένων παιχνιδιού και διεπαφών χρήστη τα οποία θα καθιστούσαν το παιχνίδι απογοητευτικό ή στην χειρότερη περίπτωση να μην μπορεί να παιχτεί.



Εικόνα 38 & 39: αγκύρωση του UI του HealthPanel.

Εδώ ως παράδειγμα υπάρχει η αγκύρωση του HealthPanel πάνω και αριστερά μέσω του component του Unity Rect Transform το οποίο στοιχίζει το αντικείμενο σε μια θέση διασφαλίζοντας ότι θα μείνει σε αυτή τη θέση για όλες τις αναλύσεις. Με τον ίδιο τρόπο γίνεται και για τις υπόλοιπες διεπαφές χρήστη καθώς και για τους καμβάδες.

Επίσης για την κάλυψη όλης της οθόνης για συγκεκριμένους καμβάδες όταν αλλάζει, χρησιμοποιείται το ίδιο εργαλείο με την λειτουργία Stretch. Με αυτό καλύπτονται όλες οι αναλύσεις οθόνης της κινητής συσκευής, εξασφαλίζοντας ότι δεν θα υπάρχουν προβλήματα κατά την αλλαγή των αναλύσεων, για την καλύτερη εμπειρία των χρηστών.



Εικόνα 40: η λειτουργία Stretch.

Στο παράδειγμα χρησιμοποιείται το PauseMenu το οποίο ευθύνεται για την παύση του παιχνιδιού. Η λειτουργία Stretch χρησιμοποιείται για να καλύψει όλη την εικόνα που χρησιμοποιείται ως φόντο την οθόνη της συσκευής ώστε να εμφανίζεται με τις σωστές αναλογίες σε κάθε οθόνη.

5.2 Ανάλυση Θέσεων Αντικειμένων

Πριν γίνει ανάλυση θέσεων αντικειμένων θα ήταν καλό να αναφερθεί το inventory του παίκτη. Το inventory στο "Epic Quest" χρησιμεύει ως αποθήκη για τα διάφορα αντικείμενα που συλλέγουν οι παίκτες σε όλη τη διάρκεια του ταξιδιού τους. Οι παίκτες μπορούν να αλληλοεπιδράσουν με αυτό το inventory, είτε για να εξοπλίσουν αντικείμενα και να βελτιώσουν τις ικανότητες του χαρακτήρα τους, είτε να καταναλώσουν αντικείμενα για να αναπληρώσουν την ζωή τους. Τα αντικείμενα αυτά περικλείονται από τις θέσεις του inventory. Στο "Epic Quest", η ανάλυση των slot του inventory είναι βασικός παράγοντας για τη βελτίωση της εμπειρίας του παίκτη, διασφαλίζοντας ότι οι παίκτες έχουν το κατάλληλο μέγεθος για την αλληλεπίδρασή τους με τα αντικείμενα αυτά. Για αυτό το λόγο παρακάτω θα παρέχει μια εις βάθος ανάλυση του κώδικα που εφαρμόζεται στο "Epic Quest" για να επιτευχθούν οι επιθυμητές αναλύσεις των θέσεων των αντικειμένων.

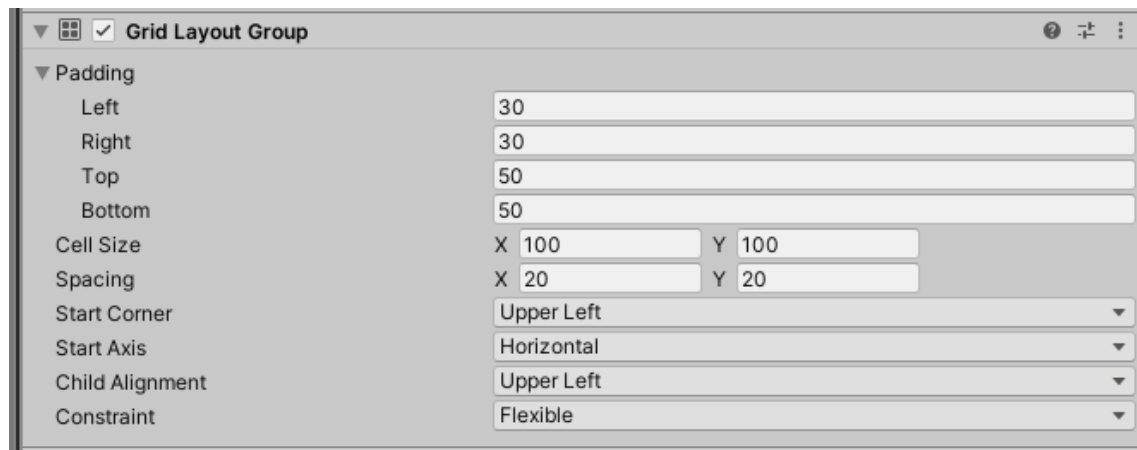
5.2.1 Κώδικας ανάλυσης θέσεων αντικειμένων.

```
if (screenWidth <= targetWidth && screenHeight <= targetHeight)
{
    // Screen resolution is smaller or equal to the target, use min cell size and max spacing
    slotSize = minSlotSize;
    spacing = maxSpacing;
}
//when resolution gets higher than 3200x1440
else if (screenWidth >= maxTargetWidth && screenHeight >= maxTargetHeight){
    slotSize = maxSlotSize2;
    spacing = minSpacing;
}
//when resolution is higher than 1920x1080 and smaller than 3200x1440
else
{
    // Screen resolution is larger than the target, use max cell size and min spacing
    slotSize = maxSlotSize;
    spacing = minSpacing;
}

// Set the cell size, padding, and spacing in the Grid Layout Group
gridLayout.cellSize = new Vector2(slotSize, slotSize);
gridLayout.spacing = new Vector2(spacing, spacing);
}
```

Εικόνα 41: Εντολές ελέγχου της μεθόδου "AdjustLayout".

Αυτό το κομμάτι του κώδικα περικλείεται από την μέθοδο "AdjustLayout" το οποίο με τη σειρά του εκτελείται στην μέθοδο "Start", η οποία παίρνοντας το Grid Layout Group, εφαρμόζει σε αυτό την μέθοδο "AdjustLayout" με την εκκίνηση του παιχνιδιού. Συγκεκριμένα η "AdjustLayout" διαχειρίζεται το component του Unity το Grid Layout Group, το οποίο διαχειρίζεται το μέγεθος και το κενό ανάμεσα στις θέσεις όπως και το κενό πάνω, κάτω, δεξιά και αριστερά από το inventory στη συγκεκριμένη περίπτωση. Οι μεταβλητές screenWidth και screenHeight είναι η όποια ανάλυση έχει ο παίχτης στη κινητή συσκευή του. Μέσω του targetWidth και targetHeight, τα οποία είναι ένα όριο από αναλύσεις που τέθηκε η οποία είναι 1920x1080 και χρησιμοποιείται στη συνθήκη «if». Η συνθήκη αυτή ελέγχει την ανάλυση του παίχτη η οποία αν είναι μικρότερη ή ίση από το targetWidth και το targetHeight τότε παίρνει για το μέγεθος της θέσης το μικρότερο μέγεθος και το μεγαλύτερο κενό ανάμεσα στις θέσεις. Αλλιώς εάν η ανάλυση είναι μεγαλύτερη από 1920x1080 αλλά μικρότερη από 3200x1440, τότε χρησιμοποιείται το μεγαλύτερο μέγεθος θέσης και ο μικρότερος χώρος. Τέλος εάν η ανάλυση είναι μεγαλύτερη ή ίση από 3200x1440 που έχει τις μεταβλητές maxTargetWidth και maxTargetHeight, τότε χρησιμοποιείται ένα ακόμα μεγαλύτερο μέγεθος θέσης το maxSlotSize2, λόγω της ακόμα μεγαλύτερης ανάλυσης και του μεγαλύτερου κενού ανάμεσα στις θέσεις. Ο διαχωρισμός αυτός έγινε με βάση δοκιμών σε διαφορετικές αναλύσεις και αντιμετωπίζοντας προβλήματα σε μικρότερες αναλύσεις(800x480) και μεγαλύτερες αναλύσεις(3200x1440).



Εικόνα 42: το component Grid Layout Group.

```

Unity Message | 0 references | Added by thanossipsas21@gmail.com on Friday, September 29, 2023
private void Start()
{
    // Get the GridLayoutGroup component
    GridLayout = GetComponent<GridLayoutGroup>();

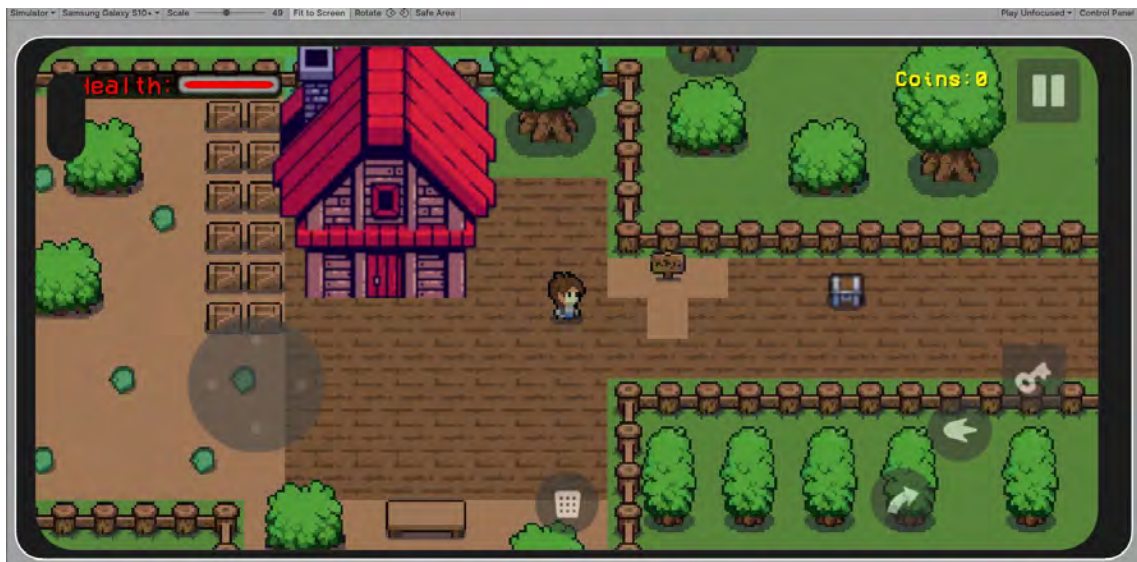
    // Adjust the cell size, padding, and spacing based on the screen resolution
    AdjustLayout();
}

```

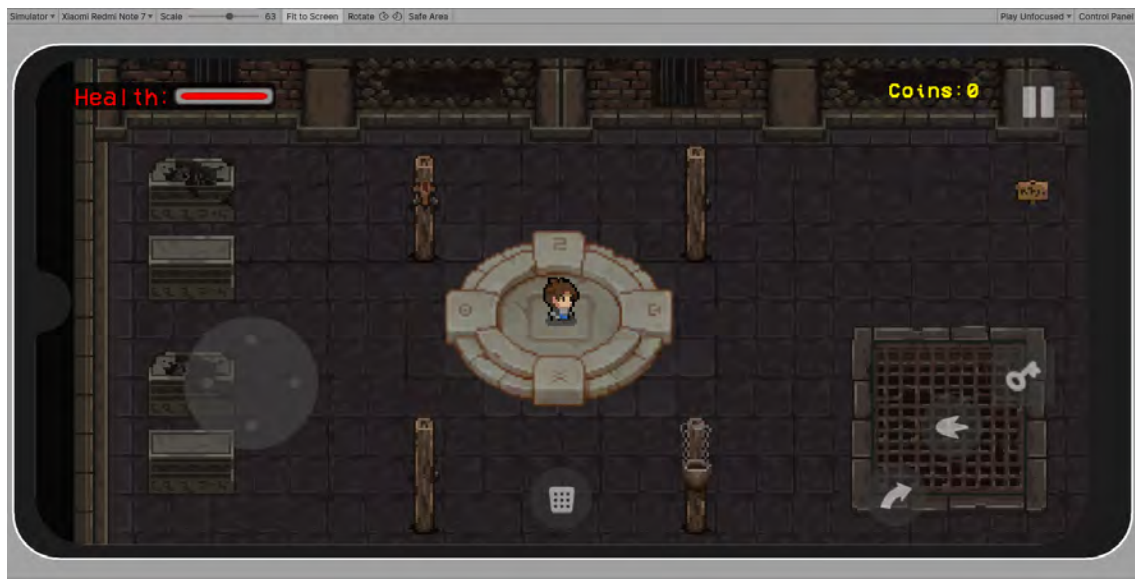
Εικόνα 43: Η μέθοδος Start.

6 Δοκιμή λειτουργίας παιχνιδιού σε Android

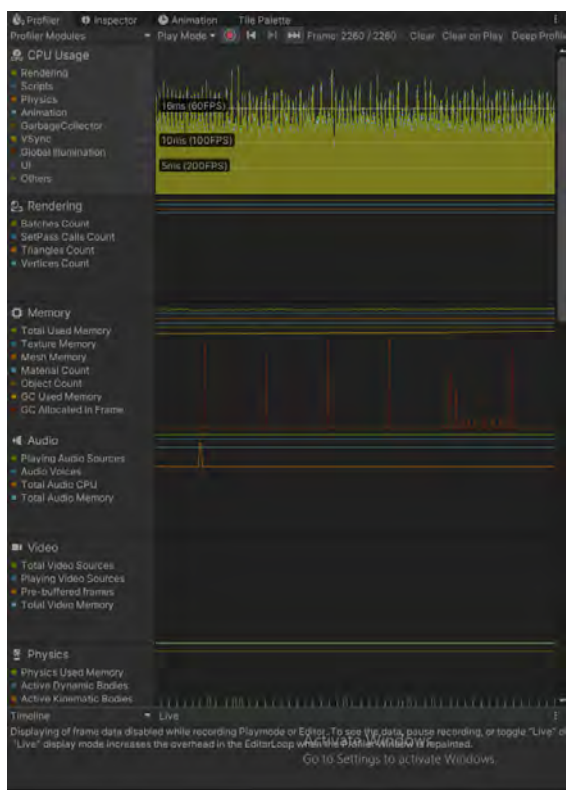
Μέσω ποικίλων δοκιμών στον editor του unity, στον προσομοιωτή και στην τελική κατασκευή, σε συνδυασμό με εκτεταμένο εντοπισμό και διόρθωσης σφαλμάτων, το παιχνίδι στις συσκευές Android έχει εξελιχθεί σε μια εξαιρετικά ομαλή εμπειρία για τους χρήστες. Η επαναληπτική διαδικασία δοκιμών μας επέτρεψε να βελτιώσουμε και να βελτιστοποιήσουμε κάθε πτυχή του παιχνιδιού, διασφαλίζοντας ότι αποδίδει άψογα σε ένα ευρύ φάσμα συσκευών Android. Με έντονη εστίαση στην ενίσχυση της ικανοποίησης των χρηστών, αυτή η ολοκληρωμένη φάση δοκιμών ήταν αναγκαία για την παροχή μιας ευχάριστης εμπειρίας παιχνιδιού για τους χρήστες κινητών συσκευών Android. Παρακάτω προβάλλονται δύο από τις πολλές δοκιμές από διαφορετικά μοντέλα Android, το ένα είναι το Samsung galaxy s10+ με ανάλυση 1440 x 3440 και το δεύτερο είναι το Xiaomi Redmi note 7 με ανάλυση 1080x2340. Η δοκιμή και στα δύο μοντέλα έγινε ομαλά με ικανοποιητικό framerate 60 και πάνω που θεωρείται ομαλή απόδοση και σε άλλα παρόμοια παιχνίδια σε κινητές συσκευές.



Εικόνα 44: Δοκιμή στο Samsung galaxy s10+.



Εικόνα 45: δοκιμή στο Xiaomi Redmi note 7.



Εικόνα 46: στατιστικά απόδοσης παιχνιδιού.

7 Συμπεράσματα

Συμπερασματικά, το ταξίδι για την ανάπτυξη αυτού του παιχνιδιού δεν ήταν τίποτα λιγότερο από μια αξιοσημείωτη εμπειρία εκμάθησης. Η διαδικασία χαρακτηρίστηκε από αμέτρητες προκλήσεις και εμπόδια, με χαρακτηριστικό παράδειγμα το επίμονο ζήτημα της εσφαλμένης μεταφοράς δεδομένων θέσης μεταξύ των σκηνών, εμποδίζοντας τον παίκτη να εμφανιστεί στην αρχή της επόμενης σκηνής και βάζοντάς τον σε μέρη τα οποία ήταν έξω από την σκηνή με αποτέλεσμα να κολλάει σε ένα σημείο χωρίς επιστροφή. Ωστόσο, μέσω του επίπονου εντοπισμού σφαλμάτων και της σχολαστικής επίλυσης σφαλμάτων, ξεπεράστηκαν τα εμπόδια και η εμπειρία κατασκευής του παιχνιδιού έγινε ευχάριστη. Αυτή η πτυχιακή εργασία έχει προσφέρει ανεκτίμητες γνώσεις για τον περίπλοκο κόσμο του γραφικού σχεδιασμού, της κωδικοποίησης και του εντοπισμού σφαλμάτων, προσφέροντας μια βαθύτερη κατανόηση των πολυπλοκοτήτων που αποτελούν τη βάση της ανάπτυξης παιχνιδιών. Με την ευκαιρία, θα ξεκινούσα ξανά με ενθουσιασμό αυτό το ταξίδι, οπλισμένος με νέες γνώσεις και προθυμία για τη δημιουργία παιχνιδιών. Κοιτάζοντας μπροστά, στο μέλλον θα μπορούσε να γίνει περαιτέρω επέκταση, με τη δυνατότητα εισαγωγής περισσότερων και γραφικά πολύπλοκων επιπέδων, προκλητικών εχθρών και διαφορετικών συστημάτων μάχης, προσφέροντας στους παίκτες ακόμη περισσότερες ευκαιρίες να ασχοληθούν, να απολαύσουν και να ψυχαγωγηθούν από τον εξελισσόμενο κόσμο του παιχνιδιού.

Βιβλιογραφία

- [1] SAVING THE GAME IS SHAPING THE GAME Aarseth, 2003, Van Vught & Glas, 2017.
- [2] Evaluating the use of the Unity engine for developing 2D mobile games in consideration of start-up/student developers Alain Simons 2017.
- [3] Design Implications for Context Aware Mobile Games Anders Liljedal 2002.
- [4] Combat in Games Joseph C. Osborn Lederle-Ensign, Dylan Wardrip-Fruin, NG 2022.
- [5] The Design and Implementation of a Mobile Game Engine for the Android Platform Jon Hammer 2012.
- [6] IMPLEMENTING USER INTERFACE FUNCTIONALITY FOR MOBILE GAMES IN UNITY GAME ENGINE Kasper Ekqvist 2017.
- [7] MOBILE GAME PROTOTYPE FOR SAVONIA Sini-Maaret Mustonen 2015.
- [8] DESIGNING A TABLETOP ROLEPLAYING GAME Miikka Virtanen 2018.
- [9] GAME DEVELOPMENT - FROM THE VISION TO THE FINAL PRODUCT Oriol Tobar 2016
- [10] MOBILE GAME USABILITY AND HEURISTICS Obi Ikemefuna Marizu 2015.
- [11] BOSS FIGHT ANIMATIONS. Structure of a Boss Fight Animation and How it Preconditions a Player's Actions Toivo Kylmä 2022.
- [12] A Comparison of Graphics Performance of Tile Based and Traditional Rendering Architectures Tsz Ho Wong 2010.
- [13] INTRODUCTION TO 2D GAME DEVELOPMENT WITH UNITY Tu Thanh Nguyen 2021.

Παράρτημα Α

```
1 reference | Changed by thanossipas21@gmail.com on Wednesday, August 16, 2023
public void Save(SaveData data){
    string fullPath = Path.Combine(dataDirectoryPath, dataFileName);

    try{

        //create the directory where the file will be stored when there isn't one
        Directory.CreateDirectory(Path.GetDirectoryName(fullPath));

        //serialization of c# save data to json
        string dataToStore = JsonUtility.ToJson(data,true);

        //encrypt data
        if(useEncryption){
            dataToStore = EncryptDecrypt(dataToStore);
        }

        //write the serialized data to file
        using(FileStream stream = new FileStream(fullPath, FileMode.Create)){
            using (StreamWriter writer = new StreamWriter(stream)){
                writer.Write(dataToStore);
            }
        }

        catch(Exception e){
            Debug.LogError("Error when trying to save the file : " + fullPath + "\n" + e);
        }
    }
}
```

Εικόνα 47: Η μέθοδος Save του κώδικα FileDataHandler.

Η μέθοδος Save αυτή βρίσκεται στον κώδικα FileDataHandler όπου διαχειρίζεται την αποθήκευση της προόδου να γίνει σε έναν φάκελο στην συσκευή του χρήστη. Πιο συγκεκριμένα η μέθοδος αυτή δημιουργεί ένα directory με το fullPath του αρχείου στο οποίο θα βρίσκεται η πρόοδος του παίχτη. Στη συνέχεια χρησιμοποιεί το filestream για να γράψει τα δεδομένα προς αποθήκευση στην τοποθεσία την οποία βρίσκεται το αρχείο αποθήκευσης που είναι σε JSON.Επίσης εάν χρησιμοποιηθεί κρυπτογράφηση, τότε εφόσον γραφτούν τα δεδομένα, στη συνέχεια κρυπτογραφούνται χρησιμοποιώντας τη λέξη κρυπτογράφησης, καθιστώντας το αρχείο ασφαλές από τυχόν προσπάθειες τροποποίησης του.

```

1 reference | Changed by thanossipsas21@gmail.com on Wednesday, August 16, 2023
public SaveData Load(){

    string fullPath = Path.Combine(dataDirectoryPath, dataFileName);
    SaveData loadedData = null;
    if(File.Exists(fullPath)){
    try{
        //load the serialized data from file
        string dataToLoad = "";
        using(FileStream stream = new FileStream(fullPath, FileMode.Open)){
        using (StreamReader reader = new StreamReader(stream)){
            dataToLoad = reader.ReadToEnd();
        }
    }
    }
}

```

Εικόνα 48: Η μέθοδος Load του κώδικα FileDataHandler.

Η μέθοδος Load παίρνει το path από directory και το όνομα του αρχείου στη μεταβλητή fullPath που είναι string και εάν υπάρχει, τότε φορτώνει την πρόοδο του παίχτη από το αρχείο που συμπεριλαμβάνει τα νομίσματα, θέση του, τα σεντούκια που ανοίχθηκαν και τους εχθρούς που σκοτώθηκαν, χρησιμοποιώντας το filestream που διαβάζει το αρχείο αυτό.

```

[System.Serializable]
4 references | Moved down 1 positions and Renamed from SerializableDictionary to SerializableDictionary by thanossipsas21@gmail.com on Tuesday, September 12, 2023
public class SerializableDictionary<TKey, TValue> : Dictionary<TKey, TValue>, ISerializationCallbackReceiver
{
    [SerializeField] private List<TKey> keys = new List<TKey>();
    [SerializeField] private List<TValue> values = new List<TValue>();

    0 references | Added by thanossipsas21@gmail.com on Sunday, August 13, 2023
    public void OnBeforeSerialize()
    {
        keys.Clear();
        values.Clear();
        foreach (KeyValuePair<TKey, TValue> pair in this)
        {
            keys.Add(pair.Key);
            values.Add(pair.Value);
        }
    }
}

```

Εικόνα 49: Η μέθοδος OnBeforeSerialize του κώδικα SerializableDictionary.

Η μέθοδος OnBeforeSerialize του κώδικα SerializableDictionary καλείται πριν από τη σειριοποίηση ενός αντικειμένου του Unity, συνήθως όταν το έργο αποθηκεύεται ή όταν το αντικείμενο έχει επισημανθεί για σειριοποίηση. Τα αντικείμενα προς σειριοποίηση στη περίπτωση μας είναι η κατάσταση των εχθρών(αν έχουν πεθάνει ή όχι) και των σεντουκιών(αν έχουν ανοίξει ή όχι). Αυτό γίνεται μέσω 2 λιστών που περιέχουν κλειδιά τα οποία χρησιμοποιούνται για να δώσουν σε κάθε αντικείμενο ένα μοναδικό κλειδί και το value το οποίο δείχνει την κατάσταση του εκάστοτε αντικειμένου.

```

0 references | Added by thanosipss21@gmail.com on Sunday, August 13, 2023
public void OnAfterDeserialize()
{
    this.Clear();

    if(keys.Count != values.Count){
        Debug.LogError("Tried to deserialize a Serializable dictionary but the amount of keys (" + keys.Count + ") which indicates that something went wrong ");
    }

    for(int i = 0; i<keys.Count; i++){
        this.Add(keys[i] , values[i]);
    }
}

```

Εικόνα 50: Η μέθοδος OnAfterDeserialize του κώδικα SerializableDictionary.

Η μέθοδος OnAfterDeserialize καλείται μετά τη σειριοποίηση ενός αντικειμένου Unity όταν το παιχνίδι έχει αποθηκευτεί. Χρησιμοποιείται για την επαναφορά του αντικειμένου μετά τη σειριοποίηση, ελέγχοντας αν είναι ίδια τα values και τα κλειδιά, φορτώνοντας την κατάσταση των αντικειμένων. Σε περίπτωση αποτυχίας του ελέγχου μέσω της συνθήκης το παιχνίδι δεν φορτώνει.

Παράρτημα Β

Παράρτημα Γ