ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

UNIVERSITY OF PIRAEUS

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Απαλλακτική Εργασία
Τεχνολογίες Ανάπτυξης Ηλεκτρονικών Παιχνιδιών

Ονοματεπώνυμο: Αθανάσιος Αναστασόπουλος

ΑΜ: Π19009

e-mail: aathanasios68@gmail.com

Ακαδημαϊκό Έτος

2022 – 2023

Πίνακας Περιεχομένων

Εισαγωγή

Η δημιουργία των videogames έχει αναδειχθεί σε μια από τις πιο δυναμικές και καινοτόμες βιομηχανίες του παγκόσμιου πολιτισμού. Από τα πρώτα τους βήματα στις δεκαετίες του '70 και του '80, τα videogames έχουν εξελιχθεί σε σημαντικό κομμάτι της ψυχαγωγίας και της τεχνολογίας. Η δημιουργία ενός videogame απαιτεί ένα συνδυασμό αρτιότητας στον τομέα της τεχνολογίας και της δημιουργικότητας. Οι developers σχεδιάζουν, αναπτύσσουν και δοκιμάζουν πρωτότυπα gameplay και στοιχεία, ενώ οι artists και οι designers δημιουργούν τα γραφικά, τη μουσική και τον κόσμο του παιχνιδιού. Το αποτέλεσμα είναι μια συναρπαστική εμπειρία που μπορεί να εντυπωσιάσει και να καθηλώσει το κοινό του. Για να υλοποιήσει κάποιος ένα παιχνίδι πρέπει πρώτα να έχει λάβει τις απαραίτητες γνώσεις, δεξιότητες, τεχνικές αλλά και να διαθέτει τον απαιτούμενο χρόνο. Σε αυτές τις γνώσεις θα μπορούσαν να συμπεριληφθούν τα μαθηματικά, ο σχεδιασμός και προγραμματισμός γραφικών, η επεξεργασία ήχου, η αντίληψη για το πως θα μοιάζει το ίδιο το παιχνίδι γι’ αυτό και έχει δημιουργηθεί ο τομέας του προγραμματισμού παιχνιδιών. Η ραγδαία ανάπτυξη της βιομηχανίας αυτής είχε σαν αποτέλεσμα την καθιέρωση συγκεκριμένων τεχνικών και εργαλείων που αφορούν αποκλειστικά τον σκοπό αυτό. Με την ταχεία εξέλιξη των τεχνολογιών και της ψυχαγωγίας, η δημιουργία των videogames συνεχίζει να προκαλεί το ενδιαφέρον των ανθρώπων και να επηρεάζει την κουλτούρα μας. Με πολλά από τα πιο δημοφιλή videogames να έχουν γίνει παγκόσμια φαινόμενα, η δημιουργία τους συνεχίζει να αποτελεί μια από τις πιο ενδιαφέρουσες και δημιουργικές διαδικασίες της σύγχρονης εποχής. Είναι σίγουρο ότι τα videogames θα συνεχίσουν να αναπτύσσονται και να εξελίσσονται, δημιουργώντας νέες ευκαιρίες για την ψυχαγωγία και την παιδαγωγική αξία τους. Η δημιουργία των videogames είναι ένα συναρπαστικό και δημιουργικό πεδίο που προσελκύει το ενδιαφέρον και την προσοχή ανθρώπων από διαφορετικές ηλικιακές ομάδες και πολιτισμικά υπόβαθρα.

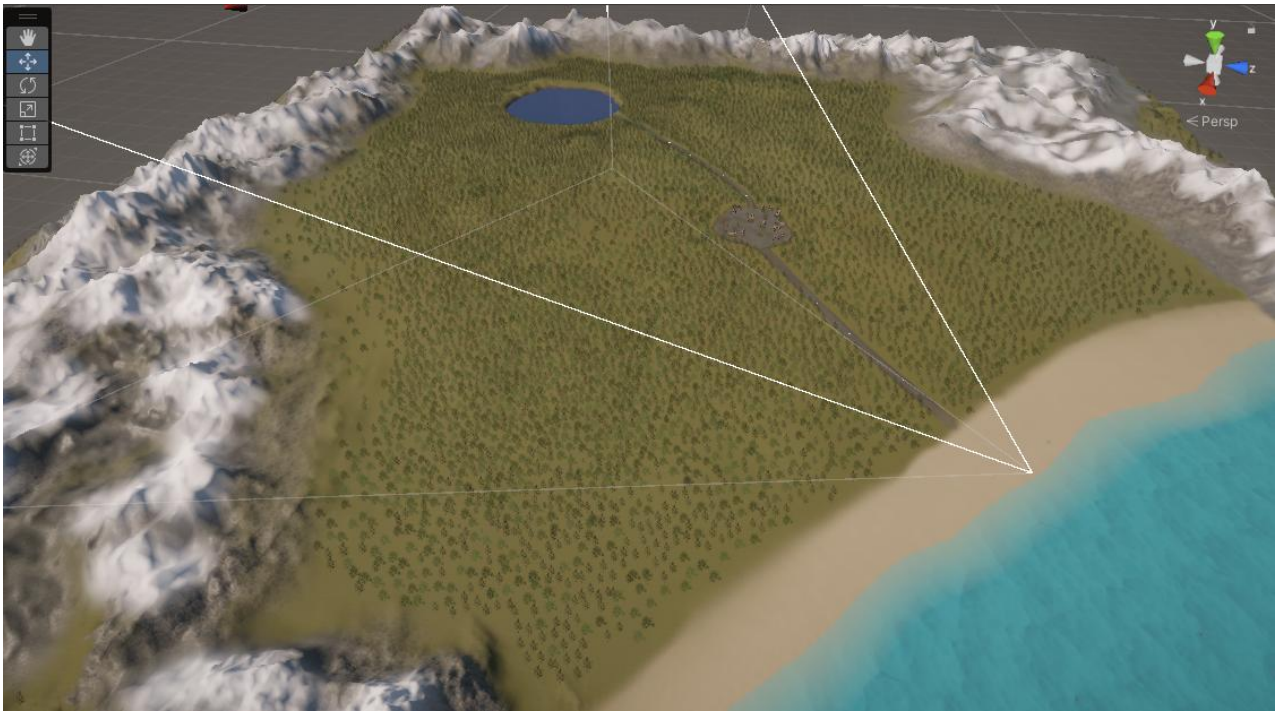
Περιγραφή του Προβλήματος

Σκοπός της παρούσας εργασίας είναι η υλοποίηση ενός διαδραστικού παιχνιδιού με τη βοήθεια ενός εργαλείου παραγωγής παιχνιδιού, του Unity. Χρησιμοποιούνται διάφορες τεχνικές υλοποίησης, οι οποίες βοηθούν το παιχνίδι να γίνει πιο δελεαστικό προκειμένου να κερδίσει τον χρήστη αλλά ταυτόχρονα και να αποτελέσει έναν ευχάριστο τρόπο εκμάθησης νέων τεχνικών για τους ίδιους τους φοιτητές. Πιο συγκεκριμένα ζητείται η δημιουργία ενός 2D ή 3D παιχνιδιού χρησιμοποιώντας την πλατφόρμα της Unity 3D. Το παιχνίδι πρέπει οπωσδήποτε να διαθέτει εισαγωγική οθόνη (Entry Screen), μενού εισαγωγικής οθόνης, help screen που θα κάνει pause το παιχνίδι και θα δείχνει την λειτουργικότητα των πλήκτρων, in-game οθόνη με μενού που θα κάνει pause το παιχνίδι και έξοδος από το παιχνίδι. Επιπλέον, μπορεί προαιρετικά να διαθέτει ένα εξελιγμένο HUD Interface, όπως minimap, health bar, stamina bar, level bar.

Αναλυτική παρουσίαση όλων των φάσεων ανάπτυξης

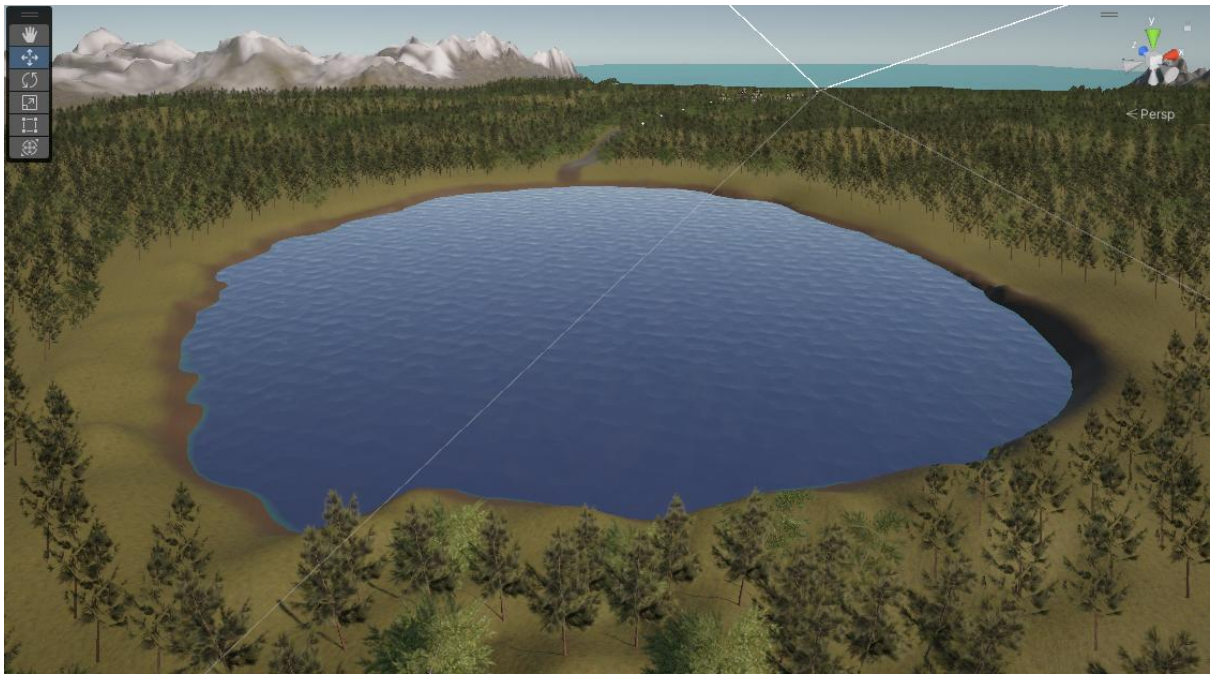
Αρχικά, το παιχνίδι δημιουργήθηκε για το μάθημα της Εικονικής Πραγματικότητας, οπότε η ανάπτυξη και η δημιουργία του βασικού κορμού του παιχνιδιού πραγματοποιήθηκε σε έναν υπολογιστή και σε ένα laptop με διαφορετικά χαρακτηριστικά προφανώς. Όσον αφορά το σύστημα μάχης του παιχνιδιού υλοποιήθηκε μόνο στον υπολογιστή. Το unity αποδείχθηκε εύκολο και σχετικά ελαφρύ στη χρήση του χωρίς να απαιτεί μεγάλο όγκο υπολογιστικής ισχύς ακόμα και στα τελικά στάδια του παιχνιδιού, όπου ο όγκος των assets ήταν μεγάλος. Όλο το project αναπτύχθηκε και υλοποιήθηκε σε Unity 2021.3.11f1. Το τελικό build του παιχνιδιού, το οποίο βρίσκεται στο φάκελο Project2022-2023 > Builds > Build_ver0.4 και δεν ξεπερνά τα 700MB, ωστόσο όλο το project ξεπερνάει τα 8GB. Τα builds δεν πήραν παραπάνω από 2 λεπτά.
Στο παιχνίδι αναλαμβάνεις τον ρόλο ενός ταξιδιώτη (traveller) ο οποίος έχει βρει στεριά σε μια ερημωμένη παραλία με μια βάρκα. Πηγαίνοντας προς στο χωριό, ο ταξιδιώτης στο δρόμο του θα πρέπει να αντιμετωπίσει εχθρούς μόνο με ένα σπαθί.

Η δημιουργία του περιβάλλον του παιχνιδιού έγινε με την χρήση του συστήματος Terrain του Unity το οποίο βοηθάει στην ανάπτυξη και επεξεργασία μεγάλων εκτάσεων ρεαλιστικών τοπίων με την παροχή αρκετών εργαλείων.




Persp


Persp


Persp

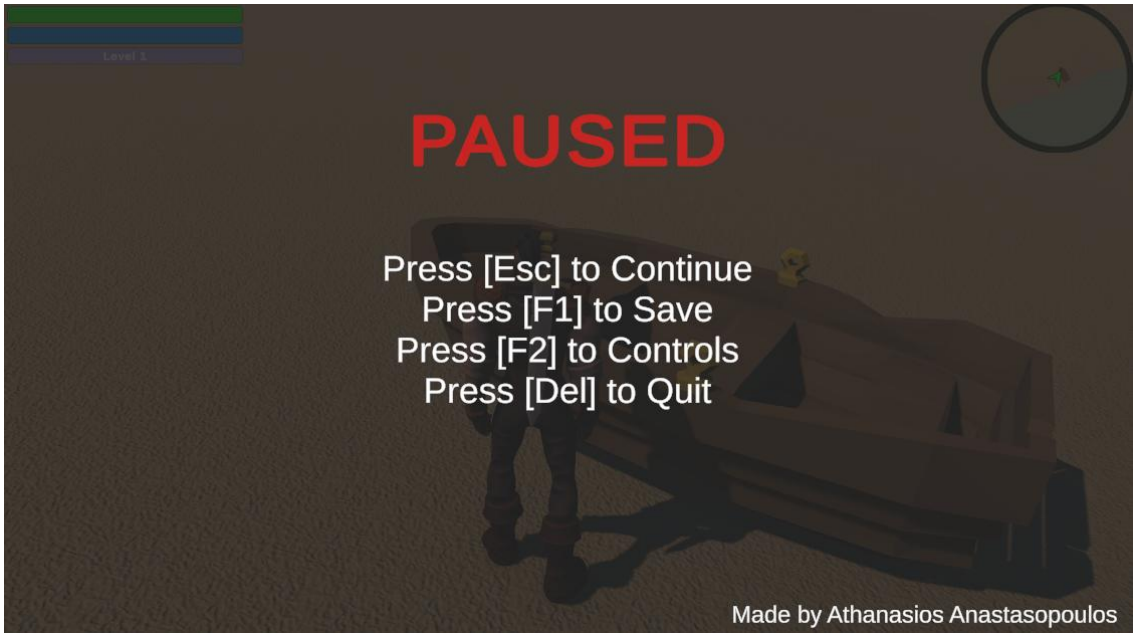
Σχεδίαση διεπαφής χρήστη μενού:

Ο παίκτης έχει την δυνατότητα να ξεκινήσει από την αρχή το παιχνίδι πατώντας “New Game”, να ξεκινήσει από εκεί που έγινε το τελευταίο save του παιχνιδιού πατώντας “Continue”. Επίσης μπορεί να πατήσει το κουμπί “Credits” και να δει τον δημιουργό του παιχνιδιού και τέλος να κλείσει το παιχνίδι.

Σχεδίαση διεπαφής χρήστη εντός παιχνιδιού:

Εντός παιχνιδιού το UI είναι απλό και εύκολο για να μπορέσει να καταλάβει ο χρήστης τι είναι το κάθε τι. Αναλυτικότερα η κύρια οθόνη περιέχει τα εξής:
1: Μπάρα ζωής (πράσινη μπάρα)
2: Μπάρα αντοχής (μπλε μπάρα)
3: Μπάρα επιπέδου (μωβ μπάρα)
4: Χάρτης, το οποίο δείχνει με πράσινο βελάκι τον παίκτη και με κόκκινη τελεία τους εχθρούς
5: Κουμπί ένδειξης αλληλεπίδρασης (Interact: F)
6: Ένδειξη για αλληλεπίδραση (κόκκινη σφαίρα)

Στην είσοδο του χωριού υπάρχει ένας χωρικός που όταν ο παίκτης τον πλησιάσει θα τον χαιρετήσει και πατώντας το F πάνω του θα εμφανίσει το παραπάνω κείμενο.
Μενού παύσης του παιχνιδιού:



Ο χρήστης πατώντας το Escape μπορεί να κάνει παύση/συνέχιση του παιχνιδιού. Πατώντας το F1 του επιτρέπεται η αποθήκευση των δεδομένων. Επιπροσθέτως μπορεί να δει τα πλήκτρα ενεργειών πατώντας το F2, φαίνονται στον παρακάτω πίνακα, και τέλος το κλείσιμο του παιχνιδιού πατώντας το Delete.

Πλήκτρα Ενεργειών	
W	Κίνηση προς τα μπροστά
S	Κίνηση προς τα πίσω
A	Κίνηση προς τα αριστερά
D	Κίνηση προς τα δεξιά
Space	Jump
Left Shift	Τρέξιμο (Sprint)
F	Interact
1	Equip/Unequip σπαθιού
Left Mouse Button (LMB)	Επίθεση

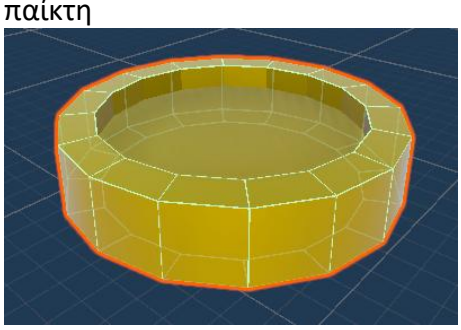
Game over:

Όταν ο χρήστης πεθάνει του εμφανίζεται στην οθόνη του ένα Game Over, με μια επιλογή μόνο, να κλείσει το παιχνίδι. Σημαντικό είναι να έχει κάνει save το παιχνίδι πριν πεθάνει ,αν δεν θέλει να ξεκινήσει το παιχνίδι από την αρχή.

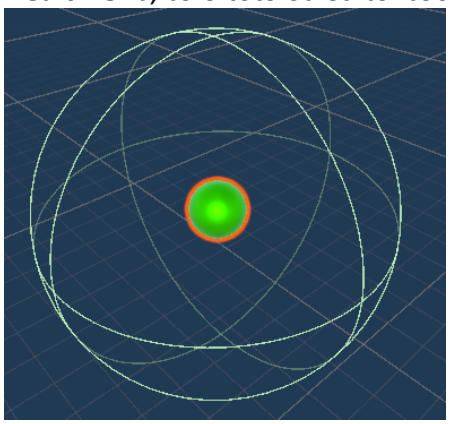
Prefabs

Τα prefabs που δημιουργήθηκαν είναι τα εξής:

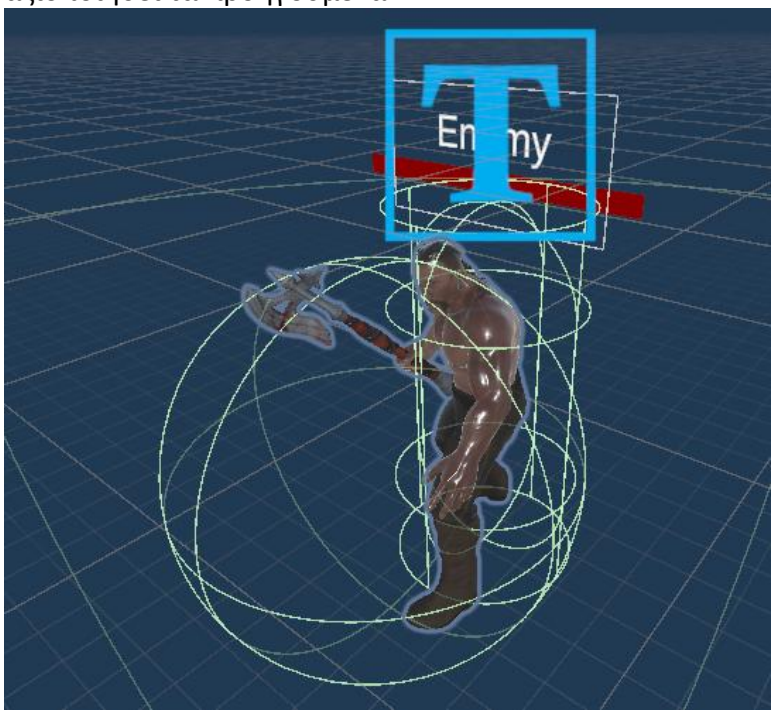
1. **Coin**, το οποίο δημιουργήθηκε με την χρήση του ProBuilder της Unity δίνει τα xp για την πρόοδο του παίκτη



2. **Health Orb**, το οποίο δίνει πόντους ζωής στον παίκτη



3. **Enemy**, που χωρίς αυτό θα έλειπε η μισή εμπειρία του παιχνιδιού και δεν θα μπορούσε να αξιοποιήσει τα προηγούμενα 2.



4. **DataPersistenceManager**, όπου χρησιμοποιείται για την αποθήκευση και φόρτωση δεδομένων μεταξύ σκηνών.

Πηγαίος κώδικας

Σύντομη παρουσίαση C# scripts

Health
Ορίζει ένα σύστημα υγείας για αντικείμενα παιχνιδιού στο Unity, επιτρέποντάς τους να υποστούν ζημιά και να πεθάνουν όταν η υγεία τους φτάσει στο μηδέν. Το σενάριο περιέχει μια συνάρτηση Start που προετοιμάζει την υγεία του αντικειμένου, καθώς και μια συνάρτηση TakeDamage που μειώνει την υγεία του αντικειμένου όταν χτυπιέται. Εάν η υγεία του αντικειμένου φτάσει στο μηδέν ή λιγότερο, καλείται η συνάρτηση Die, η οποία ορίζει το αντικείμενο ως "νεκρό".

PlayerHealth
Ορίζει τη συμπεριφορά ενός παίκτη που μπορεί να υποστεί ζημιά και να πεθάνει. Έχει μια μπάρα υγείας, μια μπάρα επιπέδου και μια μπάρα αντοχής και μπορεί να αποθηκευτεί και να φορτωθεί χρησιμοποιώντας τη διεπαφή IDataPersistence. Όταν το αντικείμενο του παιχνιδιού πεθαίνει, ενεργοποιεί ένα animation “Death” και αφαιρεί όλα τα component του εκτός από τα βασικά. Εμφανίζεται ένα game over όταν πεθάνει, επιτρέποντας στον παίκτη να εγκαταλείψει εντελώς το παιχνίδι.

PlayerAttack
Είναι υπεύθυνο για την ενεργοποίηση του animation επίθεσης του παίκτη όταν κάνετε κλικ στο αριστερό κουμπί του ποντικιού, αλλά μόνο εάν ο παίκτης έχει εξοπλισμένο όπλο. Χρησιμοποιεί το στοιχείο Animator για την αναπαραγωγή του animation επίθεσης και ορίζει μια boolean μεταβλητή (flag) σε true για να υποδείξει ότι ο παίκτης επιτίθεται αυτήν τη στιγμή.

EquipmentSystem
Είναι ένα σύστημα εξοπλισμού που επιτρέπει στον παίκτη να τραβήξει ένα σπαθί. Περιέχει αναφορές στη θήκη όπλου και τα αντικείμενα της θήκης, καθώς και έναν εμψυχωτή. Το σενάριο χρησιμοποιεί μια κορουτίνα για να περιμένει να τελειώσει η κινούμενη εικόνα προτού ρυθμίσει τη θήκη όπλου και τα

αντικείμενα θήκης σε ενεργά ή ανενεργά, ανάλογα με το εάν το όπλο είναι εξοπλισμένο ή όχι. Το σύστημα εξοπλισμού μπορεί να ενεργοποιηθεί πατώντας το πλήκτρο “1”.

Sword
Αντιπροσωπεύει ένα σπαθί σε ένα παιχνίδι και χειρίζεται συγκρούσεις με άλλα αντικείμενα παιχνιδιού. Εάν το σπαθί συγκρουστεί με ένα αντικείμενο που έχει την ετικέτα "Enemy", ελέγχει εάν το αντικείμενο έχει component "Health" και εάν ο παίκτης επιτίθεται αυτήν τη στιγμή. Εάν πληρούνται και οι δύο προϋποθέσεις, προκαλεί ζημιά στο αντικείμενο χρησιμοποιώντας τη συνάρτηση "TakeDamage()" από το στοιχείο Health. Τέλος, ορίζει τη σημαία "isAttacking" στο σενάριο PlayerAttack σε false για να υποδείξει ότι ο παίκτης τελείωσε την επίθεση.

Enemy
Αυτό είναι ένα σενάριο για ένα αντικείμενο Enemy σε ένα παιχνίδι που μπορεί να κινηθεί και να επιτεθεί. Διαθέτει ένα στοιχείο Animator και NavMeshAgent. Μπορεί να ειδοποιηθεί όταν δει ή χάσει την όρασή του έναν παίκτη και μπορεί να επιτεθεί στον παίκτη όταν βρίσκεται εντός εμβέλειας.

EnemyHealth
Αυτό είναι ένα script για το σύστημα υγείας ενός εχθρού που υλοποιεί τη διεπαφή IDataPersistence. Διαθέτει ιδιότητες για μπάρα υγείας, animator και λάφυρα. Δημιουργεί ένα GUID για αναγνώριση και ρίχνει λάφυρα μετά το θάνατο. Απενεργοποιεί επίσης τα στοιχεία του εχθρού και το αντικείμενο παιχνιδιού αφού πεθάνει.

EnemyAttack
Είναι υπεύθυνο για τον εντοπισμό του πότε ο παίκτης βρίσκεται εντός της εμβέλειας της επίθεσης του εχθρού και την ενεργοποίηση του κινούμενου σχεδίου επίθεσης. Στη μέθοδο Start(), λαμβάνεται μια αναφορά στο γονικό script Enemy. Η μέθοδος OnTriggerStay(Collider other) ονομάζεται κάθε καρέ όσο ο παίκτης βρίσκεται μέσα στο trigger collider. Εάν ο άλλος επιταχυντής έχει επισημανθεί ως "Player", η μέθοδος AttackPlayer() καλείται στο script Enemy. Εάν το άλλο collider έχει επισημανθεί ως "Player", η μέθοδος AttackPlayer() καλείται στο script Enemy. Υποθέτει ότι το animation επίθεσης χειρίζεται το script Enemy και ενεργοποιείται από τη μέθοδο AttackPlayer().

EnemySight
Ορίζει ένα component "EnemySight" που είναι συνδεδεμένο σε έναν trigger collider σε ένα παιχνίδι. Ακούει τα events OnTriggerEnter και OnTriggerExit για να ανιχνεύσει πότε ένα collider με την ετικέτα "Player" εισέρχεται ή εξέρχεται από την ενεργοποίηση. Με τον εντοπισμό αυτών των γεγονότων, καλεί τις αντίστοιχες μεθόδους στο γονικό component "Enemy", το οποίο επιτρέπει στον εχθρό να ανταποκριθεί με κάποιο τρόπο στην παρουσία του παίκτη.

EnemyDamageDeal
Είναι συνδεδεμένο σε ένα αντικείμενο παιχνιδιού και περιέχει δύο μεθόδους που ενεργοποιούν και απενεργοποιούν ένα GameObject που ονομάζεται "damageDealer". Η ενεργοποίηση εξαρτάται από το αν ο παίκτης επιτίθεται αυτήν τη στιγμή, και αν ναι, το GameObject παραμένει ανενεργό. Πιθανότατα επιτρέπει στον εχθρό να προκαλέσει ζημιά στον παίκτη υπό ορισμένες συνθήκες.

DamageTrigger
Προκαλεί ζημιά στον παίκτη όταν συγκρούεται. Το ποσό της ζημιάς καθορίζεται από μια μεταβλητή "damage" και το σενάριο βρίσκει το στοιχείο Health συνδεδεμένο στο γονικό αντικείμενο του χαρακτήρα του παίκτη για να εφαρμόσει τη ζημιά.

Coin
Είναι συνδεδεμένο σε ένα αντικείμενο παιχνιδιού που αντιπροσωπεύει ένα νόμισμα σε ένα παιχνίδι. Κινείται προς τον παίκτη όταν έρχεται σε επαφή μαζί του και μόλις φτάσει στον παίκτη, του δίνει XP και

αυτοκαταστρέφεται. Το σενάριο αναφέρεται σε ένα PlayerHealth component για να προσθέσει XP στο επίπεδο του προγράμματος αναπαραγωγής.

HealthOrb
Είναι συνδεδεμένο σε ένα αντικείμενο παιχνιδιού που αντιπροσωπεύει μια σφαίρα υγείας σε ένα παιχνίδι. Όταν ο παίκτης συγκρούεται με τη σφαίρα και πατήσει το πλήκτρο F, ο παίκτης θεραπεύεται και η σφαίρα καταστρέφεται μετά την ενεργοποίηση μιας κινούμενης εικόνας. Το σενάριο χρησιμοποιεί ένα στοιχείο Animator για την αναπαραγωγή μιας κινούμενης εικόνας κατά την αλληλεπίδραση. Η υγεία του παίκτη αυξάνεται κατά μια τιμή που καθορίζεται στο σενάριο. Το σενάριο διασφαλίζει ότι η σφαίρα μπορεί να ληφθεί μόνο μία φορά ελέγχοντας μια τιμή boolean.

BillboardEffect
Αυτό το script κάνει ένα αντικείμενο να βλέπει την κάμερα με ένα εφέ που μοιάζει με διαφημιστική πινακίδα. Ενημερώνει την περιστροφή του αντικειμένου ώστε να είναι πάντα στραμμένο προς την κύρια κάμερα.

MainMenuManager
Χειρίζεται τη συμπεριφορά των κουμπιών του κύριου μενού και της οθόνης φόρτωσης. Απενεργοποιεί τα κουμπιά ανάλογα με τη διαθεσιμότητα των δεδομένων του παιχνιδιού και φορτώνει τη σκηνή του παιχνιδιού ασύγχρονα. Το σενάριο περιλαμβάνει μεθόδους για την έναρξη ενός νέου παιχνιδιού, τη συνέχιση από ένα αποθηκευμένο παιχνίδι και την έξοδο από το παιχνίδι. Χρησιμοποιεί μια κορουτίνα για να ενημερώσει τη γραμμή φόρτωσης κατά τη φόρτωση της σκηνής του παιχνιδιού.

PauseManager
Διαχειρίζεται την παύση και την συνέχιση του παιχνιδιού. Αυτό το κάνει ορίζοντας διάφορα αντικείμενα ενεργά ή ανενεργά, απενεργοποιώντας ή ενεργοποιώντας το στοιχείο CinemachineBrain και θέτοντας παύση ή συνέχιση όλων των πηγών ήχου. Η οθόνη παύσης μπορεί να ανοίξει πατώντας το πλήκτρο Escape και το παιχνίδι μπορεί να αποθηκευτεί πατώντας το F1. Τα controls μπορούν να εμφανιστούν ή να κρυφτούν πατώντας το πλήκτρο F2. Μπορείτε να τερματίσετε το παιχνίδι πατώντας το πλήκτρο Delete.

NPCPatrol
Αυτό είναι ένα script για έναν NPC για περιπολία μεταξύ προκαθορισμένων σημείων. Χρησιμοποιεί τα στοιχεία NavMeshAgent και Animator για να μετακινήσει τον χαρακτήρα και να ζωντανέψει την κατάσταση του περπατήματος. Ο χαρακτήρας μετακινείται μεταξύ των σημείων σε έναν βρόχο, περιμένοντας 5 δευτερόλεπτα σε κάθε σημείο πριν μεταβεί στο επόμενο.

NPCWalking
Αυτό είναι ένα σενάριο για κάποιους NPCs να περπατούν ανάμεσα σε πολλά σημεία. Χρησιμοποιεί ένα NavMeshAgent για να μετακινήσει το NPC κατά μήκος μιας προκαθορισμένης διαδρομής, η οποία καθορίζεται από μια λίστα Transform objects. Όταν ο NPC φτάσει σε ένα σημείο, προχωρά στο επόμενο.

NPCWave
Χρησιμοποιείται σε έναν NPC που χαιρετάει τον παίκτη όταν εισέρχεται στη trigger περιοχή του NPC, ενεργοποιώντας το animation και απενεργοποιώντας το collider του NPC.

FootstepsSound
Αναπαράγει μια σειρά από αποσπάσματα ήχου βημάτων ασύγχρονα. Χρησιμοποιεί το Task.Delay για να ορίσει έναν χρόνο καθυστέρησης μεταξύ κάθε κλιπ και ορίζει τον χρόνο καθυστέρησης με βάση τη διάρκεια του τρέχοντος κλιπ συν ένα δευτερόλεπτο για να επιτρέψει το χρόνο φόρτωσης. Περιλαμβάνει επίσης έναν έλεγχο για ένα συμβάν παύσης.

Interactable
Ορίζει μια διεπαφή που ονομάζεται "IIinteractable" με τέσσερις αφηρημένες μεθόδους: OnInteract, OnEndInteract, OnReadyInteract και OnAbortInteract. Αυτές οι μέθοδοι χρησιμοποιούνται για την αλληλεπίδραση με ένα αντικείμενο, τον τερματισμό της αλληλεπίδρασης, την ένδειξη επιλογής και την αποεπιλογή ενός αντικειμένου, αντίστοιχα. Κάθε κλάση που υλοποιεί αυτήν τη διεπαφή πρέπει να παρέχει μια υλοποίηση για αυτές τις μεθόδους.

Interactor
Ο κώδικας είναι ένα Unity script για το χειρισμό αλληλεπιδράσεων αντικειμένων σε ένα παιχνίδι. Χρησιμοποιεί ακτινοβολία για να ανιχνεύσει αντικείμενα που έχουν επισημανθεί ως “Interactable” σε μια καθορισμένη απόσταση. Υλοποιεί επίσης μια διεπαφή για την αλληλεπίδραση με αυτά τα αντικείμενα και περιλαμβάνει στοιχεία διεπαφής χρήστη για ανάδραση του παίκτη.

InteractorUI
Η κλάση InteractorUI είναι ένα MonoBehaviour in Unity που ελέγχει έναν πίνακα διεπαφής χρήστη και ένα στοιχείο κειμένου. Έχει μια δημόσια μέθοδο που ονομάζεται ShowTextMessage που εμφανίζει ένα δεδομένο μήνυμα στο στοιχείο κειμένου της διεπαφής χρήστη και εμφανίζει τον πίνακα. Έχει επίσης μια δημόσια μέθοδο που ονομάζεται HideTextMessage που κρύβει το στοιχείο κειμένου και τον πίνακα διεπαφής χρήστη.

TextSign
Αυτό είναι ένα σενάριο C# για ένα αντικείμενο παιχνιδιού που υλοποιεί τη διεπαφή IIinteractable. Διαθέτει μια δημόσια μεταβλητή συμβολοσειράς για την αποθήκευση κειμένου και ένα αντικείμενο παιχνιδιού ένδειξης που μπορεί να εμφανιστεί ή να κρυφτεί. Όταν ένας interactor αλληλεπιδρά με αυτό το αντικείμενο, καλείται η μέθοδος ReceiveInteract του αλληλεπιδραστή με το αποθηκευμένο κείμενο ως παράμετρο. Οι μέθοδοι OnAbortInteract και OnReadyInteract εμφανίζουν και αποκρύπτουν το αντικείμενο του παιχνιδιού δείκτη, αντίστοιχα.

DoorController
Χειρίζεται τις πόρτες στο παιχνίδι. Χρησιμοποιεί το κλειδί "F" για να ανοίγει και να κλείνει την πόρτα όταν η συσκευή αναπαραγωγής βρίσκεται κοντά της. Το σενάριο περιλαμβάνει επίσης μια υπόδειξη που εμφανίζεται όταν ο παίκτης είναι κοντά στην πόρτα και είναι κλειστή. Η πόρτα περιστρέφεται ομαλά χρησιμοποιώντας τη μέθοδο Quaternion.Slerp.

DataPersistenceManager
Είναι ένας διαχειριστής διατήρησης δεδομένων που αποθηκεύει και φορτώνει δεδομένα παιχνιδιού προς και από ένα αρχείο. Χρησιμοποιεί έναν χειριστή δεδομένων αρχείων για τη διαχείριση της αποθήκευσης αρχείων και υποστηρίζει πολλαπλά αντικείμενα που υλοποιούν τη διεπαφή IDataPersistence για αποθήκευση και φόρτωση των δικών τους δεδομένων. Περιλαμβάνει επίσης λειτουργικότητα για την προετοιμασία νέων δεδομένων παιχνιδιού, εάν δεν υπάρχουν, και χειριστές συμβάντων για φόρτωση και εκφόρτωση σκηνής.

FileDataHandler
Αυτή είναι μια κλάση για το χειρισμό δεδομένων παιχνιδιού που είναι αποθηκευμένα σε ένα αρχείο. Έχει έναν κατασκευαστή που παίρνει τη διαδρομή του καταλόγου και του ονόματος αρχείου. Διαθέτει μεθόδους φόρτωσης και αποθήκευσης δεδομένων χρησιμοποιώντας το JsonUtility. Η μέθοδος Φόρτωση διαβάζει το αρχείο και επιστρέφει τα φορτωμένα δεδομένα και η μέθοδος Αποθήκευση εγγράφει τα δεδομένα στο αρχείο.

GameData
Αυτή η κλάση αντιπροσωπεύει δεδομένα παιχνιδιού που μπορούν να αποθηκευτούν και να φορτωθούν. Έχει πέντε δημόσια πεδία: health, level, levelUp, playerPosition και deadEnemies. Έχει έναν κατασκευαστή

που αρχικοποιεί κάθε πεδίο με προεπιλεγμένες τιμές. Η κλάση επισημαίνεται ως Serializable, ώστε να μπορεί να μετατραπεί σε μια μορφή που μπορεί να αποθηκευτεί, να μεταδοθεί ή να ανακατασκευαστεί αργότερα.

IDataPersistence
Είναι μια διεπαφή που ορίζει μεθόδους φόρτωσης και αποθήκευσης δεδομένων παιχνιδιού από ένα μόνιμο μέσο αποθήκευσης, όπως ένα αρχείο ή μια βάση δεδομένων.

SerializableDictionary
Είναι μια κλάση που επεκτείνει την κλάση Dictionary και υλοποιεί τη διεπαφή ISerializationCallbackReceiver για να ενεργοποιήσει το serialization και το deserialization του λεξικού.

Αναλυτική παρουσίαση των σημαντικότερων scripts

PlayerHealth
Αυτό είναι ένα script που αφορά κυρίως για τη διαχείριση της υγείας ενός παίκτη σε ένα βιντεοπαιχνίδι. Το σενάριο κληρονομείται από μια βασική κλάση που ονομάζεται "Health" και εφαρμόζει επίσης μια διεπαφή που ονομάζεται "IDataPersistence" για αποθήκευση και φόρτωση δεδομένων παιχνιδιού. Το script περιέχει πολλά private πεδία που είναι serialized, συμπεριλαμβανομένου ενός boolean για ανοσία, τρεις μπάρες για την εμφάνιση της υγείας, του επιπέδου και της αντοχής του παίκτη, ένα game over οθόνη, ένα animator για τον έλεγχο των animation του παίκτη και ένα αντικείμενο TextMeshProUGUI για την εμφάνιση του τρέχοντος επιπέδου του παίκτη. Επίσης, περιλαμβάνει πολλά προστατευμένα πεδία για τη διαχείριση της υγείας, του επιπέδου και της αντοχής του παίκτη, καθώς και έναν ακέραιο για την παρακολούθηση της εξέλιξης του επιπέδου του παίκτη. Περιέχει πολλές public μεθόδους για τη φόρτωση και αποθήκευση δεδομένων παιχνιδιού, καθώς και για τη θεραπεία του παίκτη, την αύξηση του επιπέδου του και τη διαχείριση της αντοχής του. Παρακάμπτει μια μέθοδο που ονομάζεται "TakeDamage" από τη βασική κλάση "Health", η οποία προκαλεί ζημιά στον παίκτη και ενεργοποιεί ένα animation “Hit” αν ο παίκτης δεν έχει ανοσία. Εάν η υγεία του παίκτη φτάσει στο μηδέν, το σενάριο παρακάμπτει μια μέθοδο που ονομάζεται "Die", η οποία ενεργοποιεί ένα animation θανάτου και καταστρέφει όλα τα component του παίκτη εκτός από τα SkinnedMeshRenderer, Transform, Animator και PlayerHealth. Επιπλέον, ειδοποιεί όλους τους εχθρούς στη σκηνή ότι ο παίκτης έχει χαθεί και ορίζει ένα στατικό boolean που ονομάζεται "gameover" σε true. Στη μέθοδο "Update", ελέγχει εάν το boolean gameover είναι αληθές και, αν ναι, εμφανίζει το παιχνίδι στην οθόνη και επιτρέπει στον παίκτη είτε να φορτώσει ένα αποθηκευμένο παιχνίδι είτε να κλείσει την εφαρμογή. Συνολικά, αυτό το σενάριο παρέχει ένα ολοκληρωμένο σύστημα για τη διαχείριση της υγείας, του επιπέδου και της αντοχής ενός παίκτη σε ένα βιντεοπαιχνίδι, καθώς και για την αποθήκευση και τη φόρτωση δεδομένων παιχνιδιού και τη διαχείριση σεναρίων παιχνιδιού.

PlayerAttack
Η κλάση PlayerAttack είναι υπεύθυνη για τον εντοπισμό του πότε ο παίκτης θέλει να επιτεθεί και την ενεργοποίηση του animation “Attack”. Η κλάση έχει μια αναφορά σε ένα εξάρτημα Animator, το οποίο χρησιμοποιείται για τον έλεγχο των animation του παίκτη. Η μεταβλητή isAttacking είναι μια δημόσια στατική μεταβλητή στην οποία μπορούν να έχουν πρόσβαση άλλες κλάσεις για να ελέγξουν εάν ο παίκτης επιτίθεται αυτήν τη στιγμή. Η μέθοδος Update() καλείται κάθε καρέ από τη μηχανή του παιχνιδιού. Σε αυτήν τη μέθοδο, η κλάση ελέγχει εάν έχει γίνει κλικ στο αριστερό κουμπί του ποντικιού (Input.GetMouseButtonDown(0)). Εάν έχει γίνει κλικ στο κουμπί και ο παίκτης διαθέτει ένα εξοπλισμένο όπλο (EquipmentSystem.isEquipped), τότε ο animator καλείται να ενεργοποιήσει το animation "Attack" καλώντας τη μέθοδο SetTrigger() στο animator και ρυθμίζοντας το isAttacking σε true. Άλλες κλάσεις μπορούν στη συνέχεια να ελέγξουν την τιμή του isAttacking για να προσδιορίσουν εάν ο παίκτης επιτίθεται αυτήν τη στιγμή.

Enemy

Σκοπό έχει να ελέγξει τη συμπεριφορά ενός εχθρού στο παιχνίδι. Η μεταβλητή animator είναι μια αναφορά στο component Animator που είναι προσαρτημένο στο αντικείμενο του παιχνιδιού, το οποίο είναι υπεύθυνο για την αναπαραγωγή animations. Η μεταβλητή agent είναι μια αναφορά στο στοιχείο NavMeshAgent που είναι προσαρτημένο στο αντικείμενο του παιχνιδιού, το οποίο είναι υπεύθυνο για τη κίνηση του χαρακτήρα στον κόσμο του παιχνιδιού. Η μεταβλητή target είναι μια αναφορά στο αντικείμενο παιχνιδιού που στοχεύει αυτήν τη στιγμή ο εχθρός, συνήθως ο παίκτης. Αυτό είναι ένα αντικείμενο Transform, το οποίο αντιπροσωπεύει τη θέση και τον προσανατολισμό του στόχου. Η running μεταβλητή είναι ένα boolean που καθορίζει εάν ο εχθρός τρέχει αυτήν τη στιγμή προς τον στόχο του. Η μέθοδος Start() καλείται όταν το script φορτώνεται για πρώτη φορά στο αντικείμενο του παιχνιδιού. Σε αυτή τη μέθοδο, οι μεταβλητές animator και agent αρχικοποιούνται για να παραπέμπουν στα κατάλληλα στοιχεία στο αντικείμενο του παιχνιδιού. Η μέθοδος Update() καλείται μία φορά ανά πλαίσιο. Εάν ο εχθρός έχει έγκυρο στόχο, ο πράκτορας κατευθύνεται να κινηθεί προς τη θέση του στόχου. Διαφορετικά, ο πράκτορας καλείται να παραμείνει στη θέση του. Η μέθοδος SawPlayer() καλείται όταν ο εχθρός εντοπίσει για πρώτη φορά τον παίκτη. Χρειάζεται μια παράμετρος GameObject, η οποία είναι το αντικείμενο του παιχνιδιού παίκτη. Ορίζει τη μεταβλητή target στο transform του παίκτη, ορίζει τη running σε true και ενεργοποιεί το animation “Running” στο animator. Η μέθοδος LostPlayer() καλείται όταν ο εχθρός χάσει «από τα μάτια του» τον παίκτη. Χρειάζεται μια παράμετρος GameObject, η οποία είναι το αντικείμενο του παιχνιδιού παίκτη. Εάν ο παίκτης είναι ο τρέχων στόχος, ορίζει τον στόχο σε null. Διαφορετικά, αφήνει τον στόχο αμετάβλητο. Ενημερώνει επίσης το running ανάλογα και ενεργοποιεί το animation “Running” στο animator. Η μέθοδος AttackPlayer() καλείται όταν ο εχθρός είναι αρκετά κοντά στον παίκτη για να επιτεθεί. Ενεργοποιεί το animation “Attack” στο animator. Συνολικά, αυτό το σενάριο παρέχει μια βασική υλοποίηση ενός εχθρικού χαρακτήρα που μπορεί να εντοπίσει και να κυνηγήσει τον παίκτη, καθώς και να του επιτεθεί όταν πλησιάσει αρκετά.

EnemyHealth

Η κλάση EnemyHealth επεκτείνει μια κλάση Health και υλοποιεί τη διεπαφή IDataPersistence. Είναι υπεύθυνο για τη διαχείριση της υγείας ενός εχθρού και για την αποθήκευση και τη φόρτωση των δεδομένων που σχετίζονται με το εχθρό. Η κλάση περιέχει πολλά serialized πεδία συμπεριλαμβανομένου ενός αναγνωριστικού συμβολοσειράς που είναι ένα μοναδικό αναγνωριστικό για το εχθρό, ένα slider που εμφανίζει την υγεία του εχθρού, ένα animator που ελέγχει τα animation του εχθρού και μια λίστα από αντικείμενα, που αντιπροσωπεύουν τα αντικείμενα που θα πέσουν μετά το θάνατο του εχθρού. Η μέθοδος LoadData φορτώνει τα αποθηκευμένα δεδομένα που σχετίζονται με τον εχθρό από ένα αντικείμενο GameData. Ελέγχει εάν ο εχθρός έχει επισημανθεί ως νεκρός στα αποθηκευμένα δεδομένα και αν είναι, τότε καλεί τη μέθοδο Dead η οποία είναι υπεύθυνη για την απενεργοποίηση του εχθρού και την αφαίρεση των collider του. Η μέθοδος SaveData αποθηκεύει την τρέχουσα κατάσταση του εχθρού σε ένα αντικείμενο GameData. Προσθέτει το id και τη κατάσταση του εχθρού στο λεξικό των νεκρών εχθρών στο αντικείμενο GameData. Η μέθοδος GenerateGuid δημιουργεί ένα μοναδικό αναγνωριστικό για κάθε εχθρό. Η μέθοδος Start ορίζει την αρχική τιμή του αντικειμένου healthBar με βάση την τρέχουσα και τη συνολική υγεία του εχθρού. Η μέθοδος TakeDamage καλείται όταν ο εχθρός υποστεί ζημιά. Μειώνει την υγεία του εχθρού, ενημερώνει την τιμή HealthBar και ενεργοποιεί το animation"Hit" του animator. Η μέθοδος Die καλείται όταν εχθρός πεθάνει. Ορίζει τη dead μεταβλητή σε true, ενεργοποιεί το animation "Death", ρίχνει τα loot και καλεί τη μέθοδο Dead για να απενεργοποιήσει τον εχθρό και να αφαιρέσει τους collider του. Η μέθοδος Invoke χρησιμοποιείται για την καθυστέρηση της κλήσης στη μέθοδο Dead κατά 6 δευτερόλεπτα. Η μέθοδος DropLoot δημιουργεί στιγμιότυπα των αντικειμένων GameObject στη λίστα loots και εφαρμόζει μια δύναμη έκρηξης σε αυτά. Η μέθοδος Dead απενεργοποιεί τη δέσμη ενεργειών Enemy του εχθρού, απενεργοποιεί το στοιχείο NavMeshAgent και αφαιρεί όλους τους collider από το εχθρό. Απενεργοποιεί επίσης τα component CanvasScaler και GraphicRaycaster του αντικειμένου Canvas που είναι συνδεδεμένο

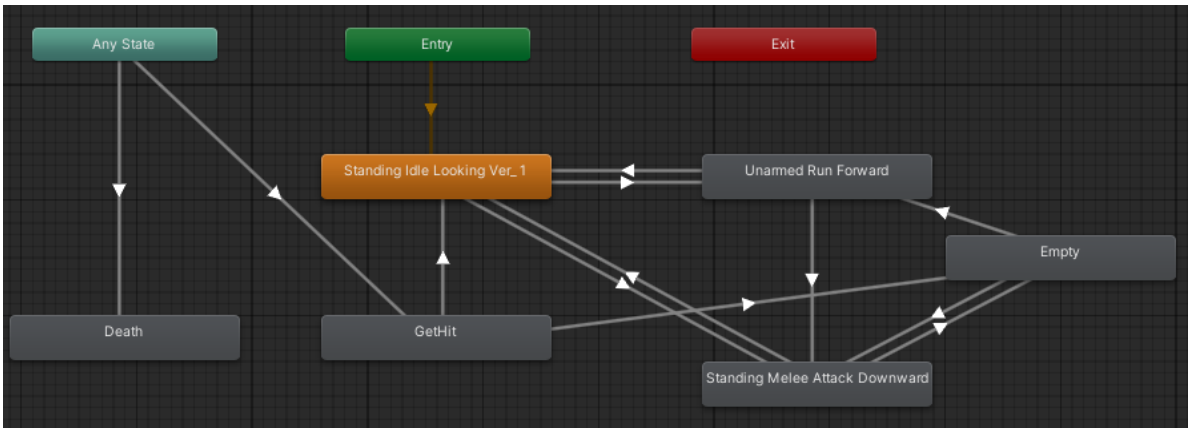
στο εχθρό και καταστρέφει το αντικείμενο Canvas. Τέλος, καταστρέφει το εχθρό μετά από καθορισμένη καθυστέρηση.

Animations που χρησιμοποιήθηκαν

Τα animations πάρθηκαν από το Mixamo και αυτά που χρησιμοποιηθήκαν είναι τα εξής:

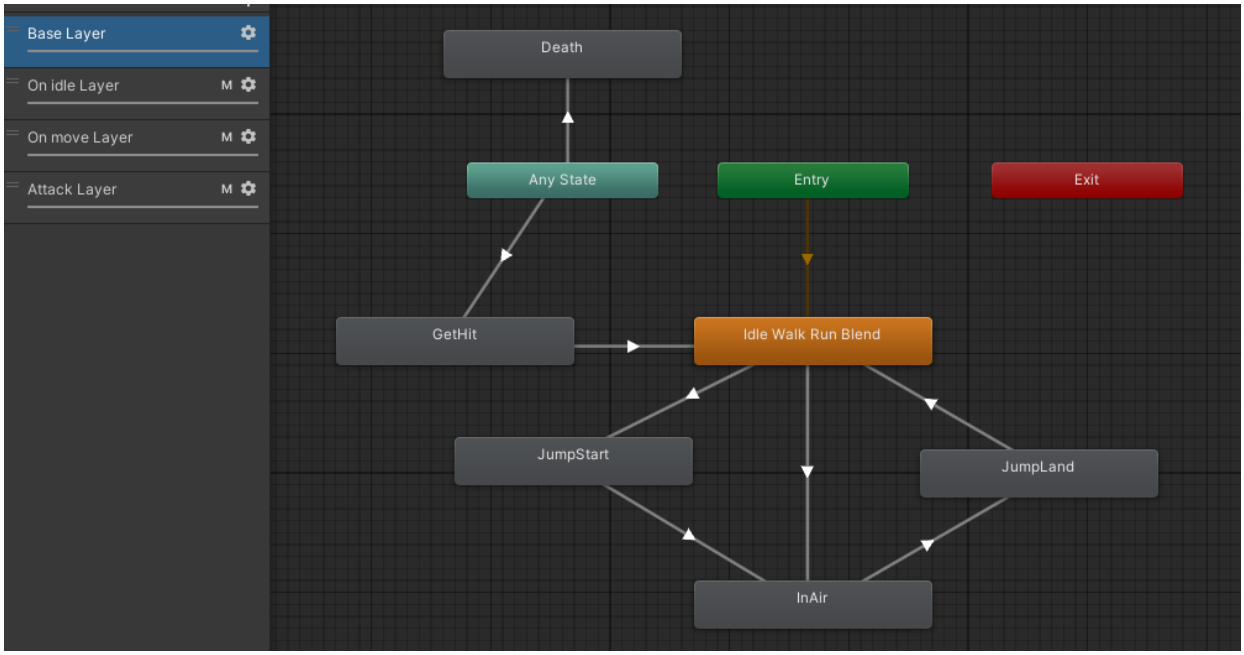
Για τον εχθρό (Brute):

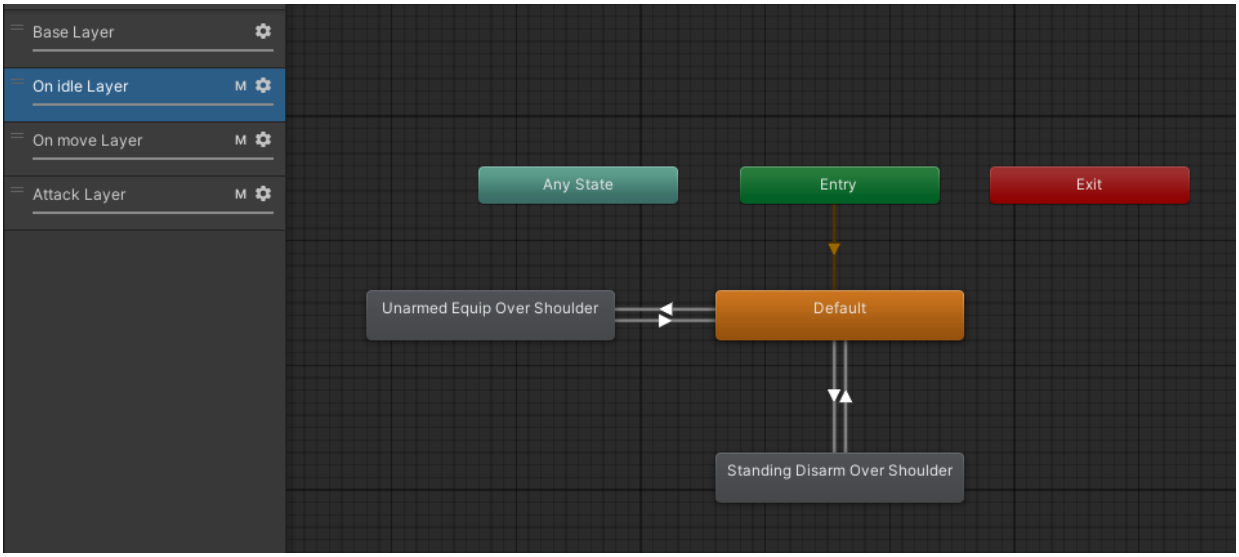
- Standing Idle Looking Ver.1
- Unarmed Run Forward
- Standing Melee Attack
- Standing React Large Gut
- Sword and Shield Death

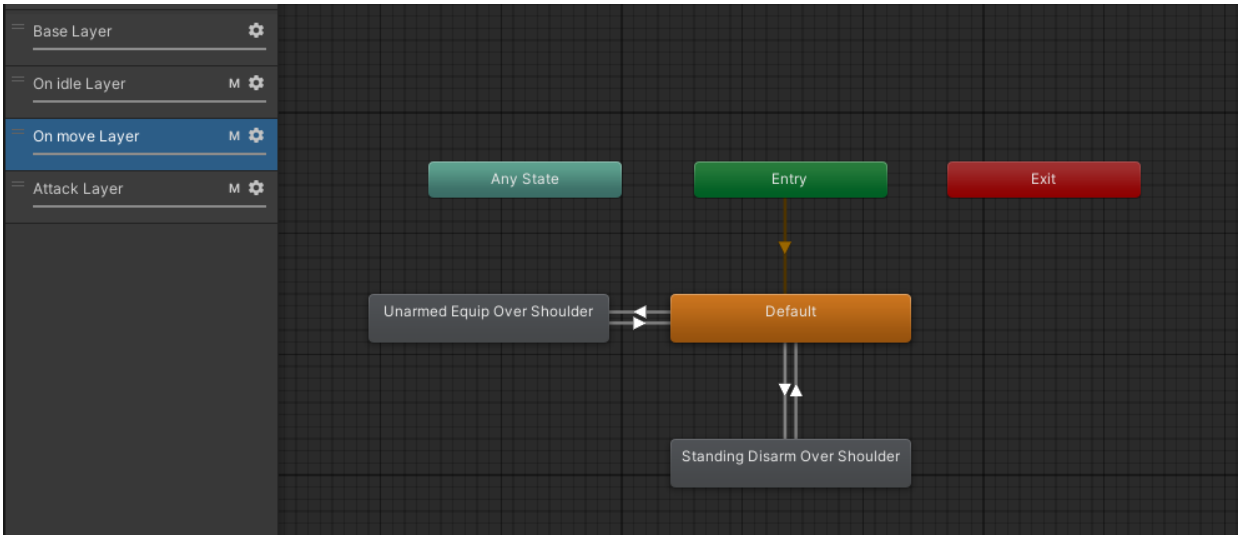


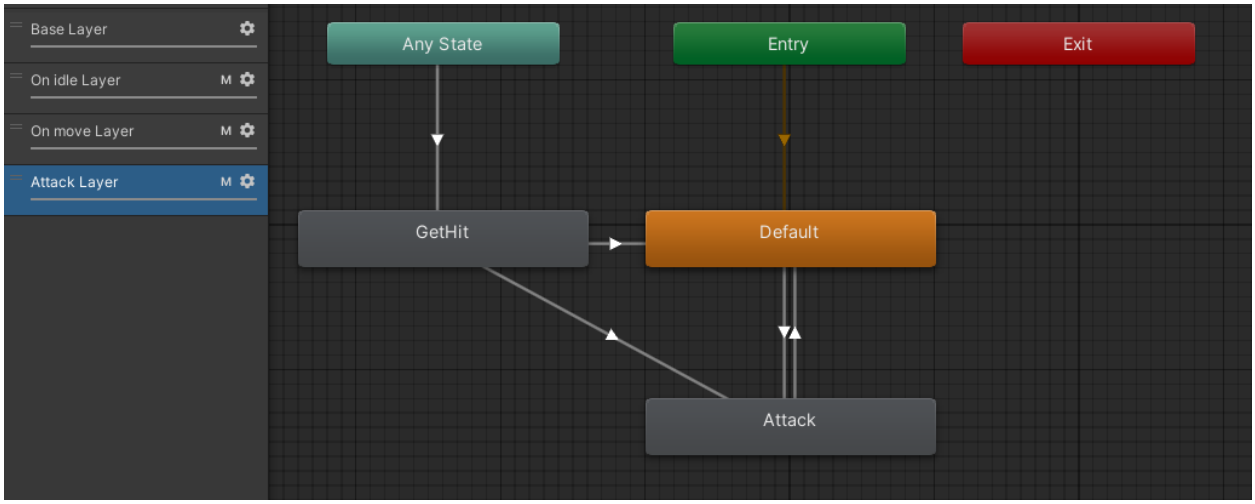
Για τον παίκτη:

- Unarmed Equip Over
- Standing Disarm Over



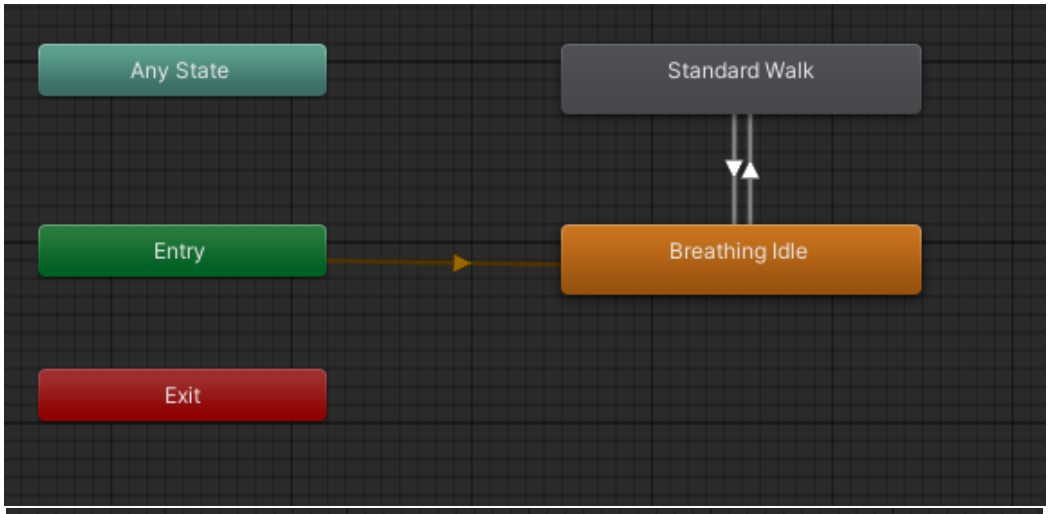


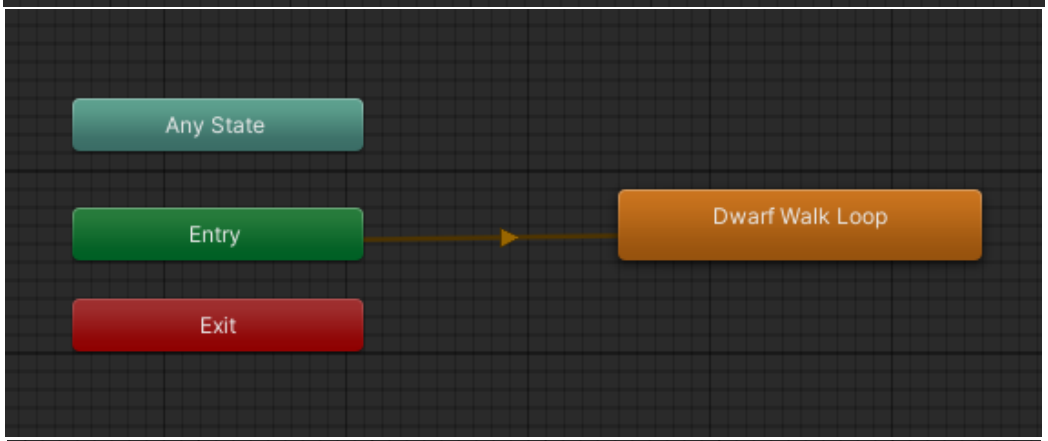


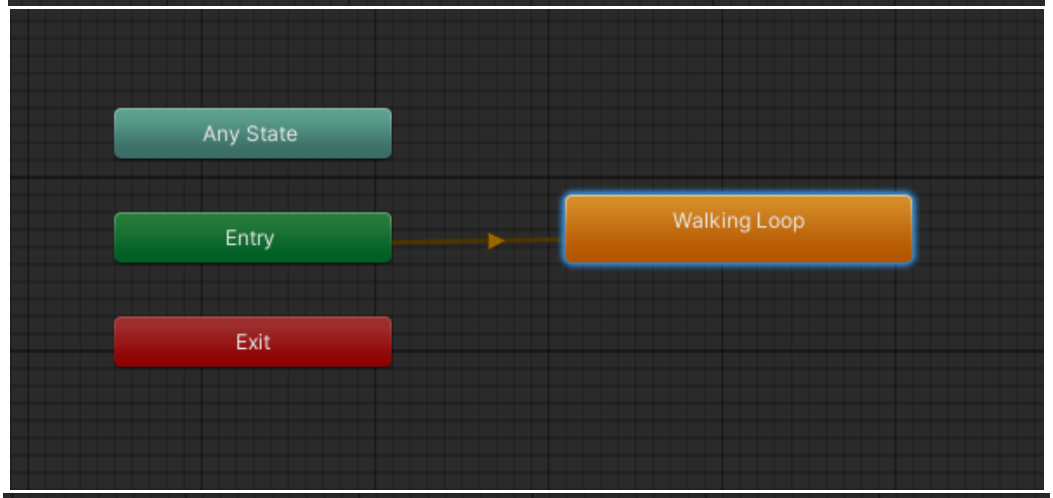


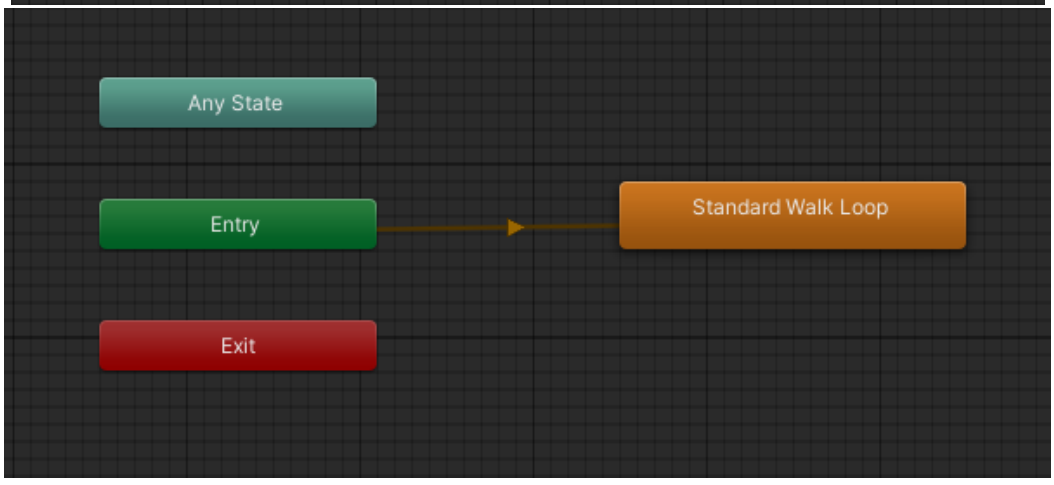
Για τους NPCs:

- Standard Walk
- Dwarf Walk
- Walking
- Breathing Idle


Any State
Standard Walk
Entry
Breathing Idle
Exit


Any State
Entry
Dwarf Walk Loop
Exit


Any State
Entry
Walking Loop
Exit


Any State
Entry
Standard Walk Loop
Exit

Κάλυψη Κριτηρίων

Αληθοφάνεια : Στο παιχνίδι γίνεται χρήση 3D μοντέλων προκειμένου ο παίκτης να νιώθει ότι τόσο τα έμψυχα (NPCs) όσο και τα άψυχα αντικείμενα (κτίρια) είναι πραγματικά στο περιβάλλον που διαδραματίζεται το παιχνίδι. Αξίζει να σημειωθεί ότι ορισμένα αντικείμενα κινούνται ή ακόμη και αλληλοεπιδρούν με τον χρήστη, το οποίο χαρίζει μια πιο αληθοφανή εμπειρία στο παιχνίδι.

Περιεχόμενο : Στο παιχνίδι το περιβάλλον έχει αναπαρασταθεί με περιεχόμενο του οποίου τα στοιχεία συνάδουν με τα δεδομένα του πραγματικού κόσμου.

Πληρότητα : Στο παιχνίδι έχουν χρησιμοποιηθεί κινούμενα στοιχεία (πχ γρασίδι, δέντρα, νερό, characters) τα οποία συμβάλλουν στο να καταλαβαίνει ο χρήστης ότι βρίσκεται και αποτελεί κομμάτι ενός εικονικού περιβάλλοντος.

Σχεδιασμός : Στο παιχνίδι το περιβάλλον έχει σχεδιαστεί με στοιχεία του πραγματικού κόσμου (πχ γρασίδι, δέντρα, νερό, κτίρια) προκειμένου να δίνει την αίσθηση του ρεαλισμού και της ομοιομορφίας της πραγματικής ζωής μέσω του εικονικού κόσμου.

Αισθητική : Τα στοιχεία του παιχνιδιού έχουν σχεδιαστεί με τέτοιο τρόπο ώστε το αποτέλεσμα να προσφέρει μια όμορφη αισθητική στον χρήστη.

Πρωτοτυπία : Χρησιμοποιώντας μόνο τα 3D μοντέλα το αποτέλεσμα θα ήταν πολύ στατικό και βαρετό, όμως συνδυάζοντας τα με τα εργαλεία το Unity το τοπίο του παιχνιδιού είναι αναμφίβολα πιο ζωντανό.

Χρηστικότητα : Στο παιχνίδι το τοπίο έχει σχεδιαστεί με τέτοιο τρόπο ώστε με την ύπαρξη δρόμων, χαρακτήρων, μετακίνησης και γενικά ερεθισμάτων ο χρήστης να απολαμβάνει την ενασχόλησή του με αυτό.

Κίνηση (animation) : Στο παιχνίδι μέσω της κίνησης, στην οποία δώσαμε ιδιαίτερη βάση, το εικονικός κόσμος γίνεται πιο διαδραστικός και ελκυστικός.

Λειτουργικότητα (functionality) : Στο παιχνίδι έχουν τοποθετηθεί τόσο έμψυχα όσο και άψυχα αντικείμενα όπως αναφέρθηκε και παραπάνω. Έτσι σε συνδυασμό με τα Unity Events, τα οποία ξεκινούν με ή χωρίς την επίγνωση του χρήστη, η εμπειρία του γίνεται πιο ενδιαφέρουσα. Ένα τέτοιο παράδειγμα αποτελεί η αλληλεπίδραση (interact) με κάποιον συγκεκριμένο NPC ή ο χαιρετισμός από κάποιον NPC όταν βρεθείς κοντά του.

Ανάπτυξη : Με τη χρήση συγκεκριμένων scripts και διαφόρων τεχνικών αναπαράστασης το αποτέλεσμα ήταν το μέγεθος του παιχνιδιού να είναι όσο το δυνατόν πιο χαμηλό και ταυτόχρονα η απόδοση του υψηλή για να προσφέρει την ιδανική εμπειρία στον χρήστη.

Αστοχίες κώδικα, Δυσκολίες, Μελλοντικές επεκτάσεις και Συμπεράσματα

Το συγκεκριμένο παιχνίδι υπήρξε το πρώτο μεγάλο παιχνίδι που έχει δημιουργηθεί από εμένα. Η δημιουργία βάσεων και θεμέλιων στα οποία θα βασίζονται κύρια κομμάτια του παιχνιδιού όπως οι εχθροί είναι περίπλοκα και απαιτούν χρόνια εξάσκησης και θεωρίας έτσι ώστε να μπορεί να γίνει συντήρηση στον κώδικα αλλά και επεκτασιμότητα χωρίς να χρειάζεται να γίνει η δημιουργία καινούργιων συστημάτων για καινούργια αντικείμενα διότι θα χρησιμοποιούνται συστήματα που ήδη υπάρχουν.
Ένα μεγάλο κομμάτι των προβλημάτων που αντιμετώπισα ήταν η “σύγκρουση” ορισμένων scripts. Πολλές από τις λειτουργίες που πραγματοποίησα κυρίως στο σύστημα μάχης επειδή ήταν προγραμματισμένες από εμένα και δεν είχα προηγούμενη εμπειρία προκαλούσαν ασυμβατότητα όταν δεν ολοκληρώνεται σωστά. Επίσης συμπεριφορά των πορτών. Πιο συγκείμενα όταν ο παίκτης πήγαινε να ανοίξει μια πόρτα, άνοιγαν και όσες πόρτες είχαν το script DoorController. Το ChatGPT βοήθησε να ανακαλυφθούν και να λυθούν πολλά από τα προβλήματα.
Σε τέτοια μεγάλα project δυσκολίες αντιμετωπίζουν αρκετοί οι οποίοι δουλεύουν ατομικά στον τομέα του planning. Σε project πολλών ατόμων υπάρχουν ρόλοι και εργασίες τα οποία μοιράζονται ανάμεσα σε μέλη τους στα οποία συνήθως ο καθένας αναλαμβάνει και ένα συγκεκριμένο κομμάτι. Κάνοντας αυτό, τα άτομα έχουν την ευκαιρία να δουλέψουν σε ένα κομμάτι του project το οποίο κάνει το άτομο να εξασκήσει τις ικανότητες τους και να γίνει καλύτερος στο συγκεκριμένο κομμάτι. Αυτο διασφαλίζει μια ποιότητα αφού το άτομο θα μπορεί να συγκεντρωθεί σε ένα μέρος του project και όχι σε πολλαπλά την ίδια στιγμή.
Είναι σημαντικό να τονίσω ότι είναι ταυτόχρονα ευκολότερο και δυσκολότερο από όσο πιστεύει ο καθένας να δημιουργήσει μια τέτοια εμπειρία, καθώς αρχικά φαίνεται τεράστιο αλλά με υπομονή και όρεξη μπορεί να δημιουργηθεί ένα πολύ καλό αποτέλεσμα. Οι δυσκολίες είναι σίγουρο ότι θα υπάρξουν αλλά το unity σου δίνει μια πληθώρα εργαλείων για να τις ξεπεράσεις και το community που το στηρίζει είναι έτοιμο ανά πάσα στιγμή να σε βοηθήσει να πετύχεις τον στόχο σου.

Βιβλιογραφία - πηγές των assets

- Scripts από το μάθημα
- ChatGTP για την επίλυση προβλημάτων
- Youtube:
  - https://youtu.be/9ytQK2QFRQo
  - https://youtu.be/aUi9aijvpgs
  - https://youtu.be/ijVA5Z-Mbh8
- Characters
  - https://www.mixamo.com/#/?limit=96&page=1&query=peasant&type=Character
  - https://www.mixamo.com/#/?limit=96&page=1&query=erica&type=Character
  - https://www.mixamo.com/#/?page=1&query=paladin&type=Character
  - https://www.mixamo.com/#/?limit=96&page=1&query=guard&type=Character
  - https://www.mixamo.com/#/?limit=96&page=1&query=knight+p&type=Character
  - https://www.mixamo.com/#/?limit=96&page=1&query=Brute+&type=Character
- Animations
  - https://www.mixamo.com/#/?limit=96&page=1&query=breathing+idle&type=Motion%2CMotionPack
  - https://www.mixamo.com/#/?limit=96&page=1&query=walk&type=Motion%2CMotionPack
  - https://www.mixamo.com/#/?limit=96&page=1&query=death&type=Motion2CMotionPack
  - https://www.mixamo.com/#/?limit=96&page=1&query=Brute&type=Motion2CMotionPack
- Assets
  - Starter Assets – Third Person Character Controller https://assetstore.unity.com/packages/essentials/starter-assets-third-person-character-controller-196526
  - Mobile Tree Package https://assetstore.unity.com/packages/3d/vegetation/trees/mobile-tree-package-18866
  - FREE – Modular Character – Fantasy RPG Human Male https://assetstore.unity.com/packages/3d/characters/humanoids/humans/free-modular-character-fantasy-rpg-human-male-228952
  - FREE Medieval Structure Kit https://assetstore.unity.com/packages/3d/environments/fantasy/free-medieval-structure-kit-141700
  - RPG Medieval Themes https://assetstore.unity.com/packages/audio/music/orchestral/rpg-medieval-themes-196607
  - Simple Water Shader URP https://assetstore.unity.com/packages/2d/textures-materials/water/simple-water-shader-urp-191449
  - Terrain Sample Asset Pack https://assetstore.unity.com/packages/3d/environments/landscapes/terrain-sample-asset-pack-145808
  - Voices – Essentials https://assetstore.unity.com/packages/audio/sound-fx/voices/voices-essentials-214441
  - PolygonKnights