ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	Ανάπτυξη RPG Εφαρμογής C++ στην Unreal Engine 5 Development of C++ Unreal Engine 5 RPG Application
Ονοματεπώνυμο Φοιτητή	ΧΑΡΑΛΑΜΠΟΣ ΚΟΥΣΙΑΒΕΛΟΣ
Πατρώνυμο	ΘΕΟΔΩΡΟΣ
Αριθμός Μητρώου	Π20102
Επιβλέπων	ΠΑΝΑΓΙΩΤΟΠΟΥΛΟΣ ΘΕΜΙΣΤΟΚΛΗΣ, ΚΑΘΗΓΗΤΗΣ

Ημερομηνία Παράδοσης 09/2024

Copyright ©

Περίληψη

Η πτυχιακή εργασία αφορά την ανάπτυξη ενός open world fantasy RPG action παιχνιδιού για υπολογιστές, το οποίο έχει δημιουργηθεί στην πλατφόρμα της Epic, Unreal Engine 5, με κύρια γλώσσα προγραμματισμού την C++. Το παιχνίδι χαρακτηρίζεται από έναν ανοιχτό κόσμο που προσφέρει στους παίκτες ελευθερία εξερεύνησης και αλληλεπίδρασης με το περιβάλλον, εχθρούς και NPCs. Μέσω της Unreal Engine 5, αξιοποιούνται σύγχρονες τεχνολογίες γραφικών, όπως η Lumen για δυναμικό φωτισμό και η Nanite για βελτιστοποιημένη απεικόνιση λεπτομερών αντικειμένων, ώστε να αποδίδεται ένα εντυπωσιακό και ρεαλιστικό οπτικό αποτέλεσμα. Το σύστημα μάχης του παιχνιδιού συνδυάζει μηχανισμούς real-time δράσης, με έμφαση στην τακτική προσέγγιση ενώ παράλληλα παρέχεται ένα σύστημα εξέλιξης. Το σενάριο του παιχνιδιού τοποθετείται σε έναν φανταστικό κόσμο με πλούσιο lore, όπου οι παίκτες εμπλέκονται σε επικές μάχες και μυθολογικές αφηγήσεις.

Η C++ χρησιμοποιείται για την ανάπτυξη της βασικής λογικής και της τεχνητής νοημοσύνης των εχθρών, ενώ αξιοποιούνται και τα ενσωματωμένα εργαλεία scripting της Unreal Engine 5 για την υλοποίηση διαδραστικών συστημάτων και μηχανισμών.

Abstract

The thesis is about the development of an open world fantasy RPG action game for computers, which has been created on Epic's platform, Unreal Engine 5, with C++ as the main programming language. The game is characterized by an open world that offers players freedom to explore and interact with the environment, enemies and NPCs. Through Unreal Engine 5, state-of-the-art graphics technologies such as Lumen for dynamic lighting and Nanite for optimized rendering of detailed objects are utilized to deliver an impressive and realistic visual effect. The game's combat system combines real-time action mechanics with an emphasis on a tactical approach while providing a progression system. The game's scenario is set in a fantasy world with a rich lore, where players engage in epic battles and mythological narratives.

C++ is used to develop the basic logic and artificial intelligence of the enemies, while Unreal Engine 5's built-in scripting tools are also utilized to implement interactive systems and mechanics

Πίνακας Περιεχομένων

Πίνακας Εικόνων

Εισαγωγή

Η συνεχής πρόοδος στον τομέα της τεχνολογίας λογισμικού και η αύξηση της πολυπλοκότητας των προγραμμάτων δημιουργούν την ανάγκη για νέες μεθοδολογίες και προσεγγίσεις στην ανάπτυξη εφαρμογών. Η γλώσσα προγραμματισμού C++ αναφέρεται συχνά ως μια αντικειμενοστρεφής γλώσσα σχεδιασμού υψηλής απόδοσης, ενώ παράλληλα δίνει τη δυνατότητα να συμμετέχει σε χαμηλού επιπέδου χειρισμό των πόρων του συστήματος Το κύριο χαρακτηριστικό της γλώσσας είναι η ευέλικτη σύνταξή της, και βλέπει εκτεταμένη χρήση στον ανταγωνιστικό προγραμματισμό από την ανάπτυξη λογισμικού συστήματος έως την ανάπτυξη παιχνιδιών, πράγμα που σημαίνει ότι μπορείτε να γράψετε κώδικα πολύ υψηλής απόδοσης και να λύσετε πολύ δύσκολα προβλήματα.

Στο πλαίσιο αυτό, το παρόν project στοχεύει στην ανάπτυξη και παρουσίαση συγκεκριμένων εφαρμογών που αναδεικνύουν τις δυνατότητες της C++ σε διάφορους τομείς. Ακολουθώντας μια συστηματική μεθοδολογία, αναλύονται και σχεδιάζονται προγράμματα που αντιμετωπίζουν προκλήσεις όπως η επεξεργασία δεδομένων, η αλγοριθμική επίλυση προβλημάτων και η βελτιστοποίηση υπολογιστικών διεργασιών. Τα επιμέρους τμήματα του project καλύπτουν διαφορετικές πτυχές της ανάπτυξης λογισμικού, με στόχο να προσφέρουν μια ολοκληρωμένη κατανόηση της λειτουργίας της γλώσσας και των δυνατοτήτων που προσφέρει.
Η προσέγγιση που ακολουθείται βασίζεται στη σταδιακή ανάπτυξη και την εμπεριστατωμένη μελέτη διαφορετικών τεχνικών που σχετίζονται με τη διαχείριση δεδομένων, την αλγοριθμική πολυπλοκότητα και την αποτελεσματική χρήση των διαθέσιμων πόρων. Τα αποτελέσματα των εφαρμογών παρουσιάζονται αναλυτικά, αποδεικνύοντας την αξία της C++ ως εργαλείο για την ανάπτυξη αποδοτικών και αξιόπιστων λύσεων σε προβλήματα που απαιτούν λεπτομερή προγραμματισμό και ακριβή έλεγχο των λειτουργιών του συστήματος.
Το project αυτό δεν περιορίζεται μόνο στην απλή παρουσίαση των δυνατοτήτων της C++, αλλά επιχειρεί να επιδείξει τη χρησιμότητά της και την ευρύτερη εφαρμογή της σε πραγματικά σενάρια. Η τεκμηριωμένη ανάλυση των επιμέρους λειτουργιών, η παρουσίαση των αλγοριθμικών προσεγγίσεων και η συζήτηση των αποτελεσμάτων συμβάλλουν στη βαθύτερη κατανόηση των πλεονεκτημάτων που προσφέρει η συγκεκριμένη γλώσσα προγραμματισμού σε ένα ευρύ φάσμα εφαρμογών.

1. Προσχεδιασμός

1.1Είδος game και περιεχόμενα

Ένας συνδυασμός τεχνικών και καλλιτεχνικών εκτιμήσεων που ανταποκρίνονταν στους στόχους και τις απαιτήσεις του project οδήγησε στην επιλογή της C++ ως γλώσσας προγραμματισμού, της Unreal Engine 5 ως μηχανής ανάπτυξης και του είδους RPG δράσης για το παιχνίδι. Οι κυριότεροι λόγοι για τους οποίους χρησιμοποιείται η C++ στην ανάπτυξη παιχνιδιών είναι οι γρήγορες επιδόσεις της και η προσαρμοστικότητά της σε πόρους χαμηλού επιπέδου. Η C++ είναι η καλύτερη επιλογή για τη δημιουργία παιχνιδιών με πολύπλοκα γραφικά και περίπλοκες διαδικασίες, όπως η απόδοση σε πραγματικό χρόνο και η τεχνητή νοημοσύνη, επειδή παρέχει πρόσβαση στη μνήμη και στις δομές δεδομένων με τρόπο που διευκολύνει την αποτελεσματική διαχείριση των πόρων του συστήματος.

Επιπλέον, η C++ και η μηχανή Unreal Engine συνεργάζονται άψογα, καθιστώντας απλή την ενσωμάτωση κώδικα απευθείας στη μηχανή του παιχνιδιού και δίνοντάς σας πλήρη έλεγχο του τρόπου λειτουργίας του, ενώ παράλληλα βελτιώνουν τις επιδόσεις. Αυτό εγγυάται ότι το παιχνίδι μπορεί να χειριστεί γρήγορα και αποτελεσματικά τις ανάγκες ενός ανοιχτού περιβάλλοντος.

Ένα από τα πιο ισχυρά και προσαρμόσιμα συστήματα παραγωγής παιχνιδιών είναι η Unreal Engine 5. Οι τεχνολογίες της, όπως το Lumen για ρεαλιστικό δυναμικό φωτισμό και το Nanite για βελτιωμένη απόδοση λεπτομερών γραφικών, είναι ιδανικές για τη δημιουργία ενός ανοιχτού περιβάλλοντος με πλούσια γραφικά. Επιπλέον, η Unreal Engine 5 προσφέρει εξελιγμένες δυνατότητες για το σχεδιασμό χαρακτήρων και περιβάλλοντος που απελευθερώνουν τους πόρους του συστήματος, ώστε οι προγραμματιστές να μπορούν να δημιουργήσουν εκτεταμένα, διαδραστικά περιβάλλοντα.

Επειδή η μηχανή διαθέτει ενσωματωμένη υποστήριξη για C++, οι προγραμματιστές μπορούν να αξιοποιήσουν πλήρως τα εργαλεία που έχουν στη διάθεσή τους και να ελέγχουν απευθείας τη λειτουργικότητα της μηχανής, δίνοντάς τους πλήρη έλεγχο του παιχνιδιού.

Από την άλλη πλευρά, η φύση και η κλίμακα του είδους RPG δράσης το καθιστούν μια δημοφιλή επιλογή για την εμπειρία του παίκτη. Αυτού του είδους τα παιχνίδια διακρίνονται από ένα πλούσιο και συναρπαστικό περιβάλλον που επιτρέπει στους χρήστες να εξερευνούν ελεύθερα, να αλληλεπιδρούν με διαφορετικούς χαρακτήρες και να λαμβάνουν μέρος σε συναρπαστικές μάχες.

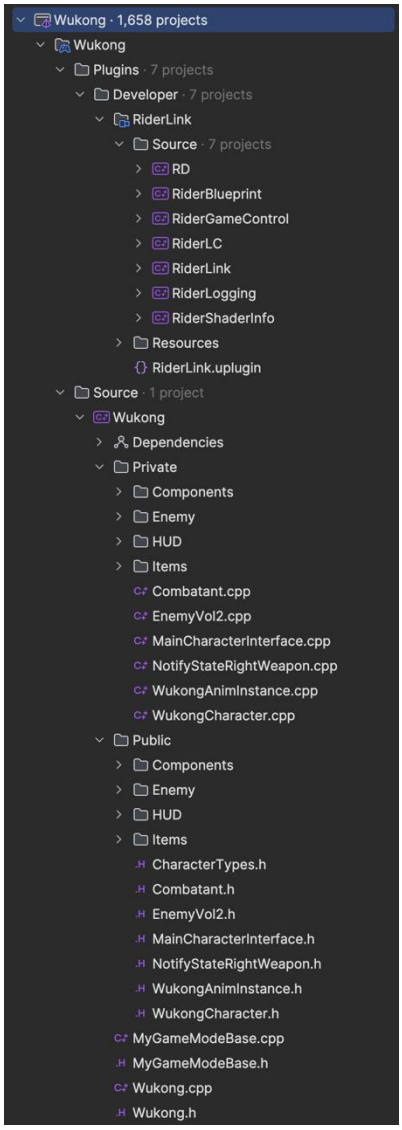
Ο ανοιχτός σχεδιασμός ενός RPG επιτρέπει ευέλικτο και στρατηγικό παιχνίδι, προσφέροντας στους παίκτες τη δυνατότητα να προσεγγίζουν τα προβλήματα με πολλαπλούς τρόπους. Η έμφαση στην εξερεύνηση, την αφήγηση ιστοριών και την αλληλεπίδραση με τον κόσμο καθιστά το παράδειγμα δράσης των RPG τέλειο ακόμη και σε αυτό το project χωρίς σύστημα εξέλιξης χαρακτήρα ή αντικειμένων. Επιπλέον, το είδος RPG συνυπάρχει με στοιχεία ανοιχτού κόσμου, επιτρέποντας την ανάπτυξη ενός ζωντανού περιβάλλοντος.

1.2 Τεχνολογίες Ανάπτυξης

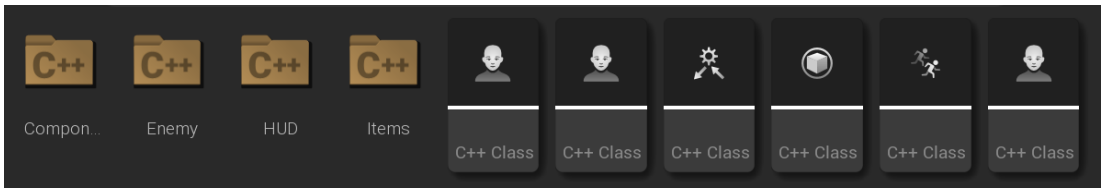
1.2.1 Rider

Για την ανάπτυξη του κώδικα χρησιμοποιήθηκε το πρόγραμμα Rider της JetBrains, το οποίο συνεργάζεται αποτελεσματικά με την Unreal Engine 5, παρέχοντας ένα περιβάλλον που ενισχύει την παραγωγικότητα και την αποδοτικότητα κατά την ανάπτυξη εφαρμογών και παιχνιδιών. Η λειτουργία *Live Coding* αξιοποιήθηκε για να επιτρέπεται η άμεση επανεγγραφή και εκτέλεση του κώδικα χωρίς την ανάγκη πλήρους επανεκκίνησης του προγράμματος, διευκολύνοντας έτσι τη διαδικασία δοκιμών και επιλύσεων σφαλμάτων σε πραγματικό χρόνο. Επιπλέον, χρησιμοποιήθηκε ένα αρχείο *header*, το οποίο περιέχει ορισμένα macros για την απλοποίηση των διαδικασιών *debugging*.

Είναι σημαντικό να σημειωθεί πως ο κώδικας C++, που υλοποιεί όλες τις λειτουργίες έχει αναπτυχθεί εξ ολοκλήρου από την αρχή και δεν προέρχεται από κάποιο έτοιμο asset pack του unreal marketplace.



Εικόνα 1. Κλάσεις C++



Εικόνα 2. Κλάσεις C++ Files στο Game Engine

2. Ανάπτυξη του χαρακτήρα του παίκτη

Εικόνα 3. Wukong Files στο Game Engine

2.1 AWukongCharacter Class

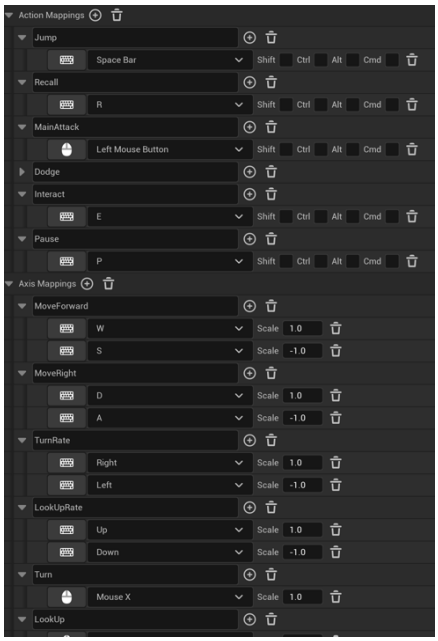
Το αρχιτεκτονικό μοτίβο που χρησιμοποιείται στον AWukongCharacter είναι βασισμένο σε κάποιες αρχές σχεδιασμού και προγραμματιστικών μοτίβων που συναντώνται συχνά σε παιχνίδια και ειδικά σε αυτά που χρησιμοποιούν την Unreal Engine και τη C++.

Η AWukongCharacter κληρονομεί από την ACharacter η οποία παρέχεται από την Unreal Engine και αυτή είναι η κλάση που εξυπηρετεί όλες τις βασικές ανάγκες των χαρακτήρων σε κάθε παιχνίδι όσον αφορά την κίνηση, τη φυσική και τις σχετικές αλληλεπιδράσεις. Με αυτόν τον τρόπο ο χαρακτήρας αποκτά όλες τις προϋπάρχουσες, σχεδιασμένες και αναπτυγμένες επιλογές της Unreal Engine. Επιπλέον, υλοποιεί το IMainCharacterInterface.

2.1.1 Απαρίθμιση Δυνατοτήτων

- Κίνηση προς τα εμπρός και πίσω - MoveForward()
- Κίνηση προς τα δεξιά και αριστερά - MoveRight()
- Περιστροφή – TurnRate()
- Μετατόπιση οπτικής γωνίας προς τα πάνω και κάτω - LookUpRate()

- Άλμα - Jump() και StopJumping()
- Βασική επίθεση (Main Attack) - MainAttack(), με υποστήριξη για combo επιθέσεις
- Επίθεση με άλμα (Air Attack)



Εικόνα 4. Action & Axis Mappings

Είναι σημαντικόνα αναφερθεί πως, για να μπορέσει ο παίκτης να χειριστεί τον χαρακτήρα Wukong μέσω των εισόδων (inputs) του, είναι απαραίτητο να οριστεί το κατάλληλο GameMode, το οποίο πρέπει να είναι τύπου Character. Το GameModeκαθορίζει τους κανόνες και τη λογική του παιχνιδιού, συμπεριλαμβανομένου του ποιος χαρακτήρας θα ελέγχεται από τον παίκτη και πώς θα γίνεται αυτός ο έλεγχος.
Έτσι ώστε να υποστηρίζονται οι λειτουργίες που σχετίζονται με την κίνηση και τις ενέργειες του χαρακτήρα, όπως το τρέξιμο, το άλμα και οι μαχητικές ικανότητες που καθορίζονται μέσα στην κλάση AWukongCharacter. Ο σωστός ορισμός του Player Controller στο GameMode επιτρέπει τη μετάφραση των εντολών του παίκτη σε ενέργειες του χαρακτήρα εντός του παιχνιδιού. Χωρίς τον κατάλληλο τύπο GameMode, δεν θα μπορούσε να καθοριστεί το πώς ο Wukong ανταποκρίνεται στα inputs του παίκτη, κάτι που θα εμπόδιζε τη σωστή λειτουργία του παιχνιδιού.

2.1.2 Κίνηση και Περιστροφή στον χώρο

Το σύστημα κίνησης του χαρακτήρα που ενσωματώνεται στην κλάση AWukongCharacter, ένα από τα χαρακτηριστικά του παιχνιδιού, είναι σχεδιασμένο με τέτοιο τρόπο ώστε να ανταποκρίνεται αρκετά στις εισόδους του παίκτη και έτσι να παρέχει εξατομικευμένο παιχνίδι. Αυτό, κατ' αρχάς, περιλαμβάνει την κίνηση προς τα εμπρός και προς τα πίσω, η οποία γίνεται σε μεγάλο βαθμό με τη μέθοδο MoveForward(). Με αυτόν τον τρόπο, υπάρχει ισορροπία της εστίασης της κάμερας και της κατεύθυνσης ενός χαρακτήρα.

Ο χαρακτήρας εστιάζει σε μια καθορισμένη κατεύθυνση και κινείται μέσα ή έξω από τον άξονα της κάμερας. Κατά συνέπεια, αυτό φέρνει περισσότερη βοήθεια στην κωδικοποίηση, καθώς οι παίκτες μπορούν να μετακινούν τους χαρακτήρες τους προς το στόχο ή μακριά από αυτόν, όπως απαιτείται, ενώ ο χαρακτήρας παραμένει στο σωστό άξονα και την προοπτική του παιχνιδιού.

Η μέθοδος MoveRight() χρησιμοποιείται επίσης για την εκτέλεση πλευρικής κίνησης, η οποία είναι πολύ σημαντική όσο και οι κινήσεις προς τα εμπρός και προς τα πίσω. Αυτό δίνει στον χαρακτήρα τη δυνατότητα να κινείται πλάγια διευρύνοντας το πεδίο των κινήσεων στο οριζόντιο επίπεδο.

Επιπλέον, η περιστροφή του χαρακτήρα ελέγχεται μέσω της μεθόδου TurnRate(), η οποία είναι πολύ σημαντική καθώς είναι ένα από τα μέσα για να «υπακούουν» οι χαρακτήρες στις εντολές του παίκτη.

Αυτή η μέθοδος καθιστά δυνατή τη στροφή των χαρακτήρων με βάση έναν καθορισμένο ρυθμό που επιτρέπει στους χαρακτήρες να στρέφονται προς την κίνηση του παίκτη Όταν η κατεύθυνση της κίνησης είναι εξ ολοκλήρου σχετική με τους στόχους, εγγυάται ότι ο χαρακτήρας μπορεί να στρέφεται προς το στόχο, να αποφεύγει τα εμπόδια και να αποκρούει τους εχθρούς όταν δέχεται επίθεση. Εφόσον ο παίκτης είναι σε θέση να διαχειριστεί πόσο γρήγορα και προς ποια κατεύθυνση θα στρέφεται ο χαρακτήρας, θα ήταν εύκολο να αντιμετωπιστούν ακόμη και τα πιο περίπλοκα γεγονότα που συμβαίνουν στο παιχνίδι.

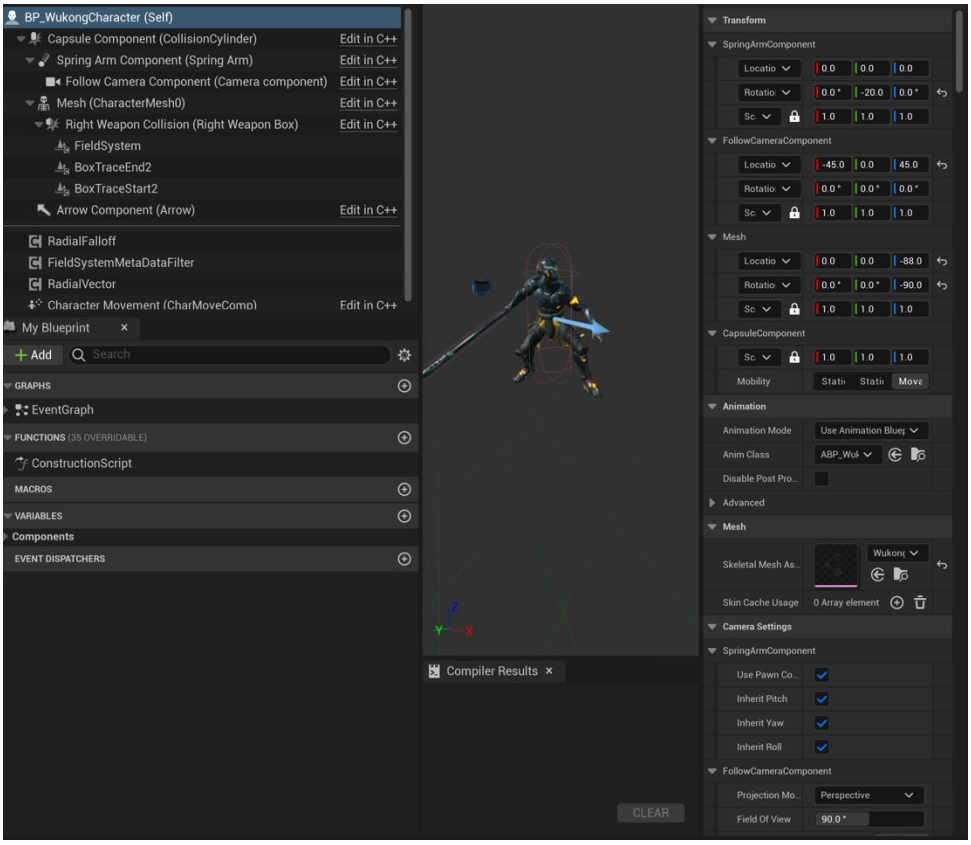
2.1.3 Λειτουργικότητα Επιθέσεων

Η μέθοδος SetupPlayerInputComponent χρησιμοποιείται για να ενώσει τις ενέργειες του παίκτη με τα πλήκτρα εισόδου. Μέσα από αυτή τη μέθοδο, ορίζονται τα πλήκτρα για την επίθεση, την άμυνα, την αλλαγή όπλων ή άλλες ενέργειες, οι οποίες καθορίζονται στο InputComponent. Αυτή η διαδικασία επιτρέπει στον παίκτη να αλληλεπιδρά με τον χαρακτήρα μέσω των controls που έχουν υλοποιηθεί στο παιχνίδι.

Η μέθοδος Attack αναλαμβάνει την έναρξη μίας επίθεσης, καθορίζοντας ποια animation και ποια όπλα θα χρησιμοποιηθούν. Κατά τη διάρκεια αυτής της μεθόδου, μπορεί να ελεγχθεί η κατάσταση του χαρακτήρα, όπως το αν είναι ήδη σε επίθεση, ώστε να αποφεύγονται διπλές ενεργοποιήσεις της επίθεσης.

Η ενεργοποίηση του collision για το όπλο στο "WukongCharacter" υλοποιείται μέσω ενός notifies (UNotifyStateRightWeapon) μέσα από το animation του χαρακτήρα. Στην αρχή του animation, καλείται η μέθοδος NotifyBegin, η οποία ενεργοποιεί το collision του όπλου. Η μέθοδος αυτή ελέγχει αν το MeshComp και ο ιδιοκτήτης του είναι έγκυροι και στη συνέχεια, χρησιμοποιεί τη μέθοδο Cast για να βρει τον χαρακτήρα του Wukong, ο οποίος είναι υπεύθυνος για το όπλο. Εάν ο χαρακτήρας βρεθεί, καλείται η μέθοδος ActivateRightWeapon που βρίσκεται στην κλάση AWukongCharacter. Αυτή η μέθοδος ενεργοποιεί το collision του όπλου, επιτρέποντας έτσι στο όπλο να ανιχνεύει συγκρούσεις μόνο κατά τη διάρκεια της επίθεσης. Κατά τη λήξη του animation, η μέθοδος NotifyEnd χρησιμοποιείται για να απενεργοποιήσει το collision μέσω της αντίστοιχης μεθόδου DeactivateRightWeapon, που επαναφέρει το collision σε "NoCollision" κατάσταση. Αυτό διασφαλίζει ότι το όπλο δεν ανιχνεύει συγκρούσεις όταν δεν χρησιμοποιείται, μειώνοντας έτσι τυχόν ανεπιθύμητες επιδράσεις.

2.2 Blueprint του Main Character



Εικόνα 5. Wukong Blueprint; Viewport

2.2.1 Components

Capsule Component: Collision Cylinder

Γεωμετρικό σχήμα που περιγράφει μια περιοχή στην οποία ο κόσμος του παιχνιδιού δημιουργεί σημεία σύγκρουσης. Ως γενικός κανόνας, είναι συνήθως κυλινδρικό και συνήθως χρησιμοποιείται για να υπαγορεύει πού συμβαίνουν οι συγκρούσεις με άλλες επιφάνειες ή αντικείμενα. Είναι καλό για τον προσδιορισμό των αποτυχιών αλληλεπίδρασης αν ένας χαρακτήρας ή κάποιο αντικείμενο θέλει να αλληλεπιδράσει με το περιβάλλον του.

Spring Arm Component: Spring Arm
Θα συνδέσει την κάμερα με έναν χαρακτήρα και θα δημιουργήσει έναν «ελαστικό» δεσμό που θα τους κρατά σε σταθερή απόσταση μεταξύ τους. Αυτό θα βοηθήσει ώστε η κάμερα να μην «κολλάει» κάτι άλλο και να έχουμε μια πιο ευχάριστη εμπειρία όταν ο χαρακτήρας μας κινείται.

Flollow Camera Component: Camera Component
Το αντικείμενο που διαχειρίζεται τι είναι ορατό στον παίκτη στον κόσμο του παιχνιδιού. Καταγράφει και εμφανίζει τη σκηνή από την οπτική γωνία του χαρακτήρα ή από οποιαδήποτε συγκεκριμένη τοποθεσία. Είναι δυνατό να το αλλάξετε ώστε να μπορείτε να βλέπετε τα στοιχεία υπό διαφορετικές γωνίες θέασης και να προσθέσετε εφέ όπως το depth of field.

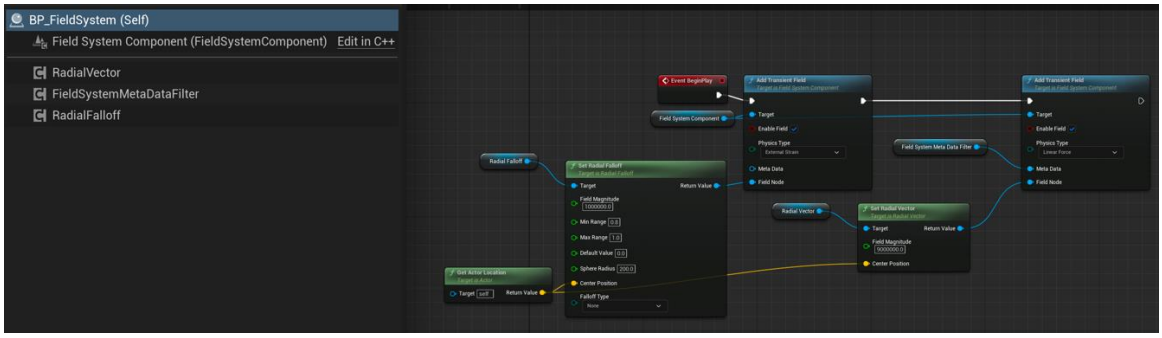
Mesh: Character Mesh
Επιτρέπει την ομαλή κίνηση της κάμερας σε σχέση με τις κινήσεις του χαρακτήρα. Συνδέει την κάμερα με έναν ηθοποιό, πράγμα που σημαίνει ότι η συγκεκριμένη κάμερα θα βρίσκεται πάντα σε μια καθορισμένη απόσταση από τον συγκεκριμένο ηθοποιό. Αυτό βοηθά στην αποφυγή του «κολλήματος» της κάμερας σε άλλα αλληλεπιδρώμενα αντικείμενα που βρίσκονται πιο κοντά, γεγονός που προκαλεί μια λιγότερο ομαλή εμπειρία θέασης κατά τη μετακίνηση ενός χαρακτήρα στον κόσμο του παιχνιδιού.

Right Weapon Collision: Collision Box
Μια απλή γεωμετρική περιοχή που χρησιμοποιείται για την ανίχνευση σύγκρουσης. Εμφανίζεται ως ένα ορθογώνιο πλαίσιο χρήσιμο για τον καθορισμό της περιοχής που μπορεί να οδηγήσει σε συγκρούσεις με άλλα αντικείμενα στο παιχνίδι. Χρήσιμο για περιοχές που πρέπει να διαχειρίζονται την είσοδο/έξοδο χαρακτήρων ή άλλων αντικειμένων.

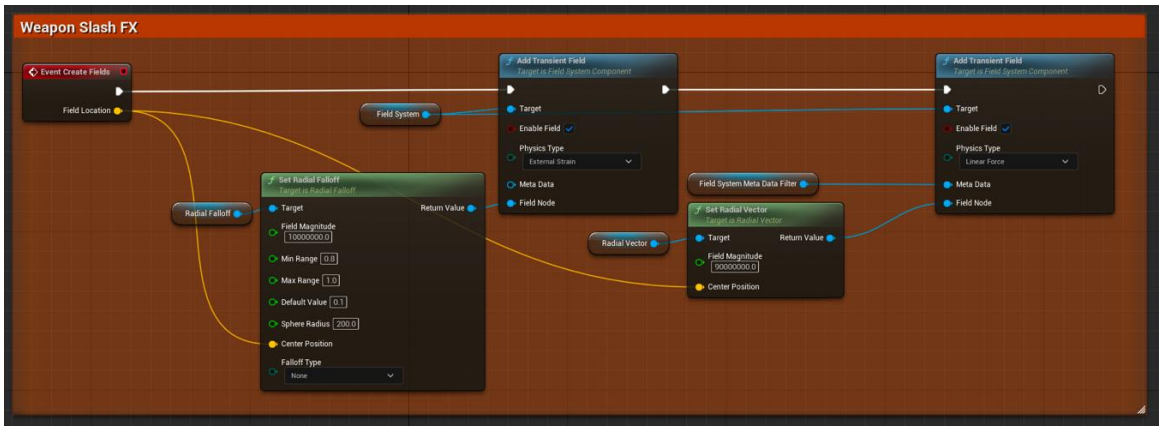
Field System: Scene Component
Θεμελιώδης τύπος component στον οποίο θα τοποθετηθούν όλα τα άλλα στοιχεία της σκηνής. Αυτό διευκολύνει την τοποθέτηση και τον προσανατολισμό διαφορετικών συστατικών, την οργάνωση και την ιεράρχηση μιας βιβλιοθήκης με κατάσταση για τον προγραμματιστή.

Arrow Component: Arrow
Είναι μια οπτική ένδειξη, η οποία δείχνει κυρίως προς ένα αντικείμενο ή μια κίνηση. Χρήσιμο για το κομμάτι της ανάπτυξης, όπου η οπτικοποίηση του προσανατολισμού του συστατικού καθιστά πιο ξεκάθαρη τη διαμόρφωση καθώς και εύκολη την κατανόηση του πώς θα φαίνεται η διάταξη του παιχνιδιού.

2.2.2 Field System



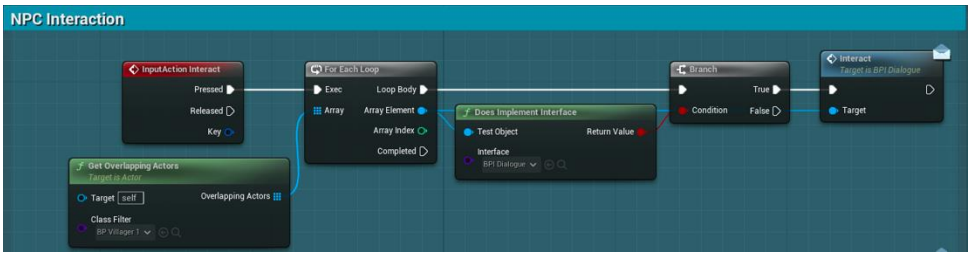
Εικόνα 6. Field System Blueprint; Event Graph: Field System λογική



Εικόνα 7. Wukong Blueprint; Event Graph: Field System λογική

Στο συγκεκριμένο Field System που έχει δημιουργηθεί, το εφέ trail εμφανίζεται πάνω στο όπλο του χαρακτήρα όταν εκτελείται μια επίθεση. Η όλη διαδικασία ξεκινά από το Blueprint του χαρακτήρα, όπου το event Create Fields ενεργοποιείται όταν ξεκινά η επίθεση. Αυτό το event κατευθύνεται προς το Add Transient Field, στο οποίο περνά ένα reference από το Field System. Το Add Transient Field είναι υπεύθυνο για τη δημιουργία ενός προσωρινού πεδίου, το οποίο διαρκεί για λίγο χρόνο και χρησιμοποιείται για την απόδοση του οπτικού εφέ που ακολουθεί το όπλο κατά τη διάρκεια της επίθεσης.
Στη συνέχεια, το Add Transient Field συνδέεται με το Set Radial Falloff, το οποίο ελέγχει τον τρόπο με τον οποίο η επίδραση του πεδίου μειώνεται όσο απομακρύνεται από το κέντρο του. Η Radial Falloff λειτουργεί ως ρύθμιση που καθορίζει την κλιμάκωση του εφέ ανάλογα με την απόσταση από το σημείο προέλευσης. Αυτή η διαδικασία εξασφαλίζει ότι το εφέ εμφανίζεται με πιο έντονη επίδραση κοντά στο όπλο και εξασθενεί όσο απομακρύνεται από αυτό.
Μετά από αυτή τη ρύθμιση, η διαδικασία επαναλαμβάνεται με ένα δεύτερο Add Transient Field, όπου αυτή τη φορά το node παίρνει ένα διαφορετικό reference από το πρώτο πεδίο, δημιουργώντας έτσι μια αλληλουχία εφέ. Αυτό το δεύτερο Add Transient Field επιτρέπει την προσθήκη επιπλέον πεδίων στο σύστημα, διασφαλίζοντας ότι το εφέ trail συνεχίζει να ενημερώνεται και να προσαρμόζεται με κάθε κίνηση του όπλου. Και σε αυτή την περίπτωση, το Set Radial Falloff χρησιμοποιείται για να ελέγξει τον τρόπο με τον οποίο μειώνεται η ένταση του εφέ σε σχέση με την απόσταση από το κέντρο.

2.2.3 NPC Interaction



Εικόνα 8: Wukong Blueprint; Event Graph: NPC Interaction λογική

Η παραπάνω διαδικασία αλληλεπίδρασης με NPC ξεκινά με το Input Action Interact, ένα node το οποίο αντιστοιχεί στη δράση του παίκτη όταν επιχειρεί να αλληλεπιδράσει με έναν χαρακτήρα. Όταν

ενεργοποιείται αυτή η ενέργεια, κατευθύνεται σε έναν For Each Loop, ο οποίος χρησιμοποιείται για να επεξεργαστεί όλους τους χαρακτήρες ή αντικείμενα που βρίσκονται γύρω από τον χαρακτήρα.

Ο For Each Loop συνδέεται με το Get Overlapping Actors, το οποίο έχει επιλεγμένη την κλάση των NPCs. Αυτό σημαίνει ότι το σύστημα εξετάζει όλους τους χαρακτήρες που βρίσκονται σε άμεση εγγύτητα με τον παίκτη και ανήκουν στην κλάση των NPCs, για να διαπιστώσει αν υπάρχουν πιθανοί στόχοι για αλληλεπίδραση.

Στη συνέχεια, κάθε χαρακτήρας που βρίσκεται από το Get Overlapping Actors ελέγχεται μέσω του Does Implement Interface, το οποίο χρησιμοποιείται για να διαπιστωθεί αν ο εκάστοτε NPC έχει υλοποιήσει το απαιτούμενο Interface. Εδώ γίνεται έλεγχος για το αν ο συγκεκριμένος NPC υποστηρίζει την αλληλεπίδραση μέσω του interface που έχει οριστεί για αυτήν τη λειτουργία. Το αποτέλεσμα αυτού του ελέγχου προωθείται σε ένα Branch, το οποίο λειτουργεί ως συνθήκη. Αν το Does Implement Interface επιστρέψει true, τότε σημαίνει ότι ο NPC υποστηρίζει την αλληλεπίδραση.

Όταν η συνθήκη στο Branch είναι true, τότε καλείται η συνάρτηση Interact, η οποία είναι δηλωμένη στο Interface και υλοποιείται από τον NPC. Η συνάρτηση αυτή καθορίζει την πραγματική συμπεριφορά του NPC όταν ο χαρακτήρας επιχειρεί να αλληλεπιδράσει μαζί του.

2.2.4 Pause Game

Εικόνα 9: Wukong Blueprint; Event Graph: Pause λογική

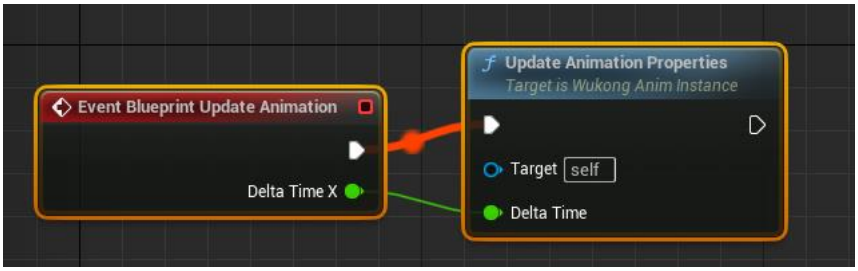
Για την επίλυση της ανάγκης παύσης του παιχνιδιού, έχει δημιουργηθεί ένα μενού παύσης το οποίο ενεργοποιείται όταν ο παίκτης πατήσει το κουμπί F. Όταν αυτό συμβαίνει, το F Pressed Node ανιχνεύει την ενέργεια και οδηγεί στη δημιουργία του Pause Menu Widget, το οποίο είναι ένας ορισμένος τύπος Blueprint Widget. Το widget αυτό στη συνέχεια προστίθεται στην οθόνη μέσω του Add to Viewport, ώστε να εμφανιστεί το μενού παύσης στον παίκτη.

Μόλις εμφανιστεί το μενού, το ποντίκι του παίκτη ενεργοποιείται μέσω της λειτουργίας Set Mouse Cursor True, επιτρέποντας στον παίκτη να χρησιμοποιήσει το ποντίκι για πλοήγηση στο μενού. Παράλληλα, γίνεται αλλαγή της λειτουργίας εισόδου σε UI Only μέσω του Set Input Mode UI Only, η οποία αποκλείει όλες τις άλλες εισαγωγές εκτός από τη διεπαφή χρήστη, διασφαλίζοντας ότι ο παίκτης αλληλεπιδρά αποκλειστικά με το μενού. Για να γίνει αυτό, χρησιμοποιείται ένας Get Player Controller, ο οποίος παρέχει τον έλεγχο του παίκτη στο UI. Τελικά, ενεργοποιείται η παύση του παιχνιδιού με το Set Game Paused True, σταματώντας όλη τη δράση και επιτρέποντας στον παίκτη να παραμείνει στο μενού μέχρι να αποφασίσει να συνεχίσει το παιχνίδι..

2.3 Animations του Main Character

2.3.1 Animation Blueprint

Στην Προεπιλεγμένη υποδοχή του Animation Blueprint πραγματοποιούνται animation montages - τα οποία συχνά περιλαμβάνουν animations δράσης, όπως επιθέσεις ή αλληλεπιδράσεις αντικειμένων.

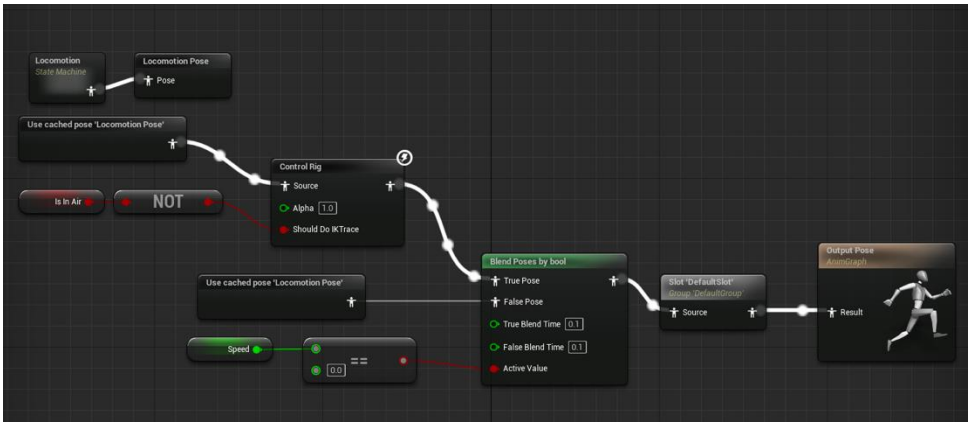


Εικόνα 10: Wukong Animation Blueprint Event Graph

Χρησιμοποιούμε τη συνάρτηση UpdateAnimationProperties(), η οποία χρησιμοποιείται για την ενημέρωση των ιδιοτήτων που συνδέονται με τα animations του χαρακτήρα, ώστε να ανταποκρίνονται καλύτερα στην τρέχουσα κατάσταση του παιχνιδιού. Κάνοντας την να καλείται κάθε frame για να ελέγχει την κατάσταση του χαρακτήρα και να αλλάζει τα animations του.

Η συνάρτηση μπορεί να διαχειριστεί την ταχύτητα ανάπτυξης του χαρακτήρα μας, η οποία είναι ο τρόπος με τον οποίο είμαστε σε θέση να η ενημέρωση του περπατήματος, του τρεξίματος ή της αδράνειας animation του αντιστοιχεί σε ένα μέτρο της ταχύτητάς του. Αν ο χαρακτήρας πέφτει ή βρίσκεται σε κατάσταση άλματος, ελέγχει αν η κατάσταση grounded είναι ψευδής και στη συνέχεια προσαρμόζει το animation ώστε να είναι άλμα/προσγείωση.

Με αυτόν τον τρόπο, η συνάρτηση UpdateAnimationProperties διασφαλίζει ότι οι κινήσεις του χαρακτήρα απεικονίζονται ομαλά και με ακρίβεια.

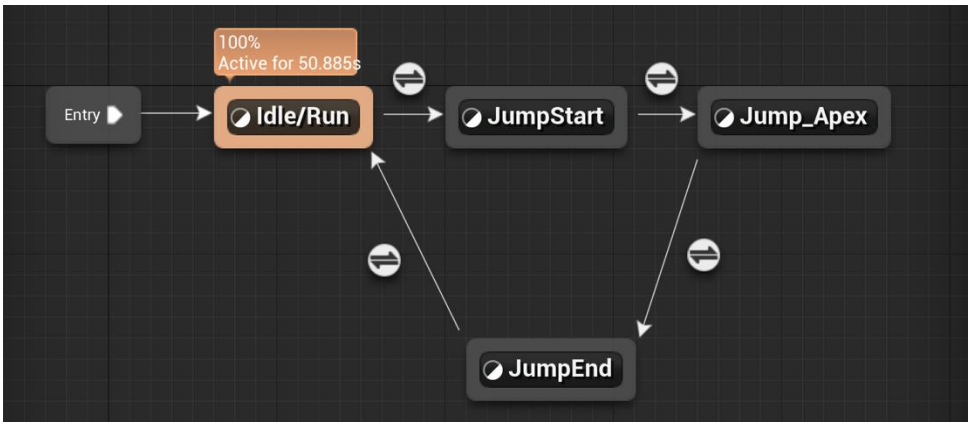


Εικόνα 11: Wukong AnimGraph

Για να αποθηκευτεί μια συγκεκριμένη στάση του χαρακτήρα ενώ εκτελείται μια κίνηση, χρησιμοποιήστε τη λειτουργία του AnimGraph Save pose στο Animation Blueprint της χαρακτήρα. Η δυνατότητα αποθήκευσης και ανάκτησης της αποθηκευμένης στάσης διευκολύνει τη διαχείριση περίπλοκων συνδυασμών animation και βελτιώνει την απόδοση εξαλείφοντας την ανάγκη επανειλημμένου επαναϋπολογισμού των ίδιων δεδομένων animations.

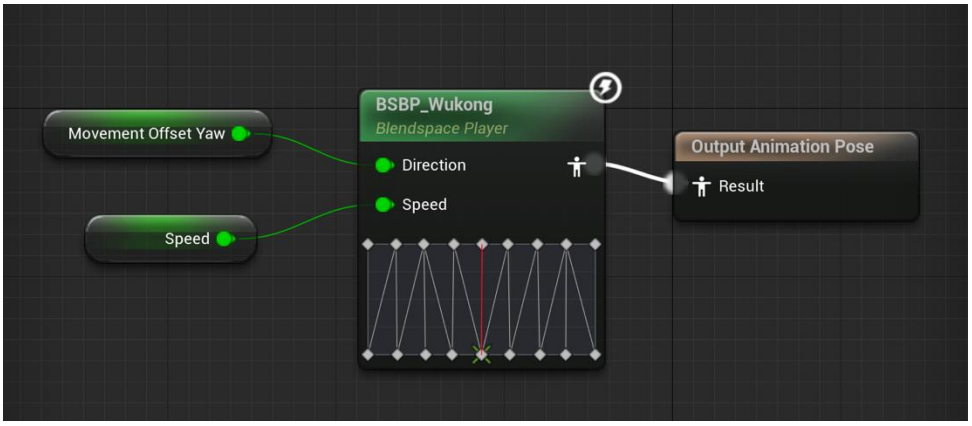
Ο κόμβος Save pose by Bool επιτρέπει την αποθήκευση μιας στάσης με βάση την τιμή μιας μεταβλητής boolean. Η τρέχουσα pose διατηρείται και μπορεί να χρησιμοποιηθεί ξανά εάν η τιμή boolean είναι αληθής.

Το Default Slot χρησιμεύει ως προορισμός για την αποθήκευση ή την αναπαραγωγή των animation montage, επιτρέποντας σε αυτά animations να προχωρούν μέσω του animation bueprints ανεξάρτητα από τις άλλες ρουτίνες animation του χαρακτήρα. Εξαιτίας αυτού, η οργάνωση και η εκτέλεση των animations ενός χαρακτήρα με τεράστια ευελιξία και δυναμική καθίσταται εφικτή από το συνδυασμό της αποθήκευσης μιας pose, του ελέγχου των τιμών boolean στα animations και του Default Slot, το οποίο παρέχει ακριβή έλεγχο των κινήσεων και των ενεργειών του χαρακτήρα.



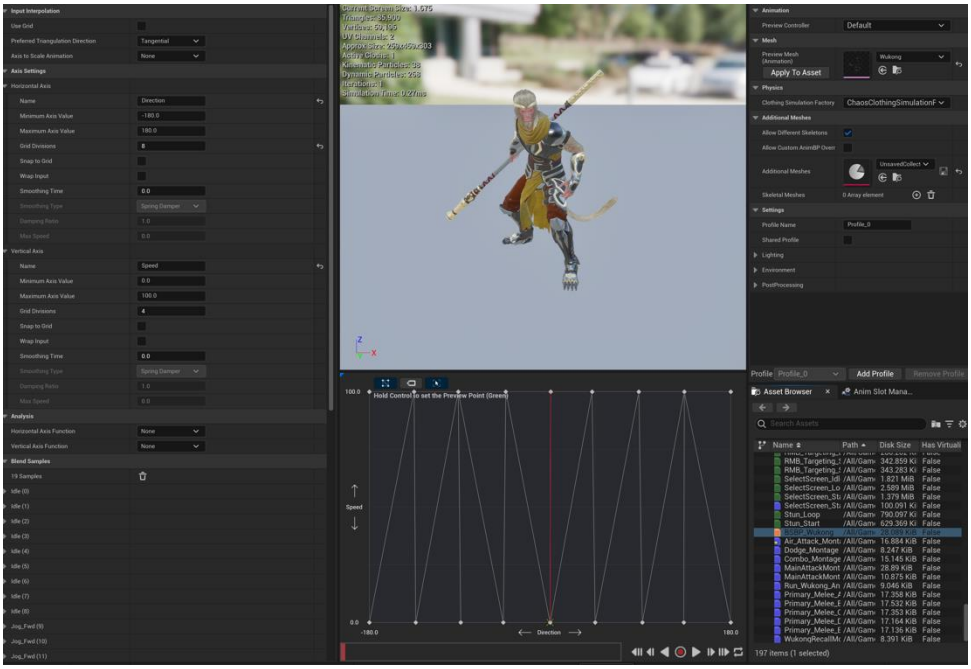
Εικόνα 12: Wukong AnimGraph Machine States

Στο AnimGraph υπαρχουν 4 machine states για την αναπαραγωγή animation του χαρακτήρα.
Idle/Run: Πραγματοποιείται η αναπαραγωγή του κατάλληλου animation με βάση την ταχύτητα του AWukongCharacter μέσω του BlendSpace που έχει δημιουργηθεί.



Εικόνα 13: Χρήση του Wukong BlendSpace

2.3.2 BlendSpace

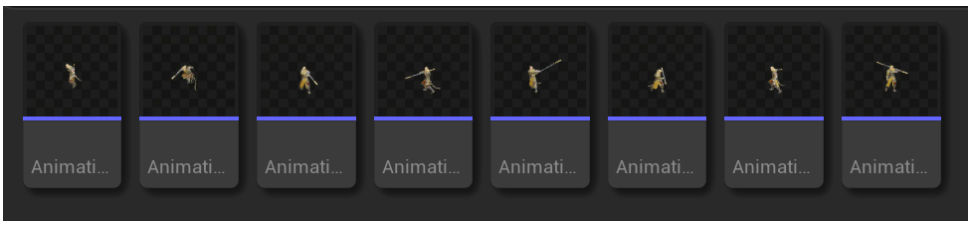


Εικόνα 13: Χρήση του Wukong BlendSpace

(Σημείωση: Το preview mesh των montage και blenspaces δεν είναι το ιδιο skin απαραιτητα με το κανονικό)

Το BlendSpace χρησιμοποιείται για τη δημιουργία ρευστών μεταβάσεων μεταξύ πολλαπλών animations, ανάλογα με παραμέτρους εισόδου, όπως η ταχύτητα ή η κατεύθυνση κίνησης του χαρακτήρα. Αντί ο χαρακτήρας να πηγαίνει κατευθείαν από ένα animation σε άλλο με απότομη αλλαγή, το BlendSpace επιτρέπει την ανάμειξη αυτών των animations, δημιουργώντας ομαλές και συνεχείς κινήσεις.Στην συγκεκριμένη περίπτωση πρόκειται για ένα BlendSpace δύο αξόνων, μπορείς να αναμείξεις animations βάδισης, τρεξίματος και στροφής ανάλογα με την κατεύθυνση και την ταχύτητα του χαρακτήρα, έτσι ώστε οι μεταβάσεις μεταξύ αυτών των κινήσεων να φαίνονται φυσικές.

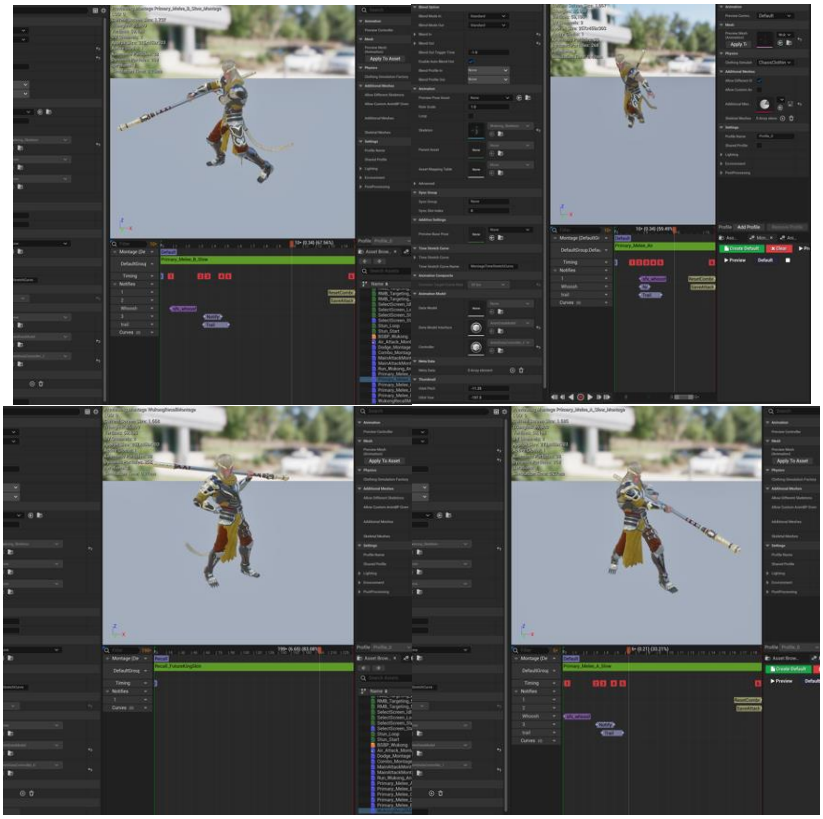
2.3.3 Animation Montage



Εικόνα 14: Wukong Animation Montage Files

Το Animation Montage είναι ένας πιο εξειδικευμένος τρόπος χειρισμού animations που χρησιμοποιείται συνήθως για συγκεκριμένες δράσεις ή αλληλεπιδράσεις ενός χαρακτήρα, όπως επιθέσεις, αλληλεπιδράσεις με αντικείμενα ή ειδικές κινήσεις. Ένα Montage μπορεί να περιλαμβάνει ένα ή περισσότερα animations, τα οποία εκτελούνται σε προκαθορισμένες χρονικές στιγμές και

τμήματα. Αυτό δίνει τη δυνατότητα να διαμερίσουμε ένα animation σε διαφορετικά τμήματα και να ελέγξουμε ποια μέρη του εκτελούνται ή διακόπτονται σε συγκεκριμένες καταστάσεις . Επιπλέον, τα Montages μπορούν να χρησιμοποιήσουν slots, όπως το Default Slot, για να καθορίσουν σε ποια περιοχή του animation blueprint θα εφαρμοστούν, διασφαλίζοντας ότι το συγκεκριμένο animation θα παιχτεί σε συνδυασμό με άλλες κινήσεις. Χρησιμοποιούνται συχνά για δράσεις που δεν πρέπει να διακόψουν το κύριο animation cycle του χαρακτήρα, όπως οι επιθέσεις ή οι ειδικές ενέργειες, οι οποίες απαιτούν μεγαλύτερο έλεγχο και ακρίβεια.



Εικόνα 15: Wukong Animation Montages

Έτσι ενώ το blend space επιλέγεται για την αναπαραγωγή movement animations, τα montages επιλέγονται για τις μαχιτικές κινήσεις του χαρακτήρα.

2.4 Attack Abilities

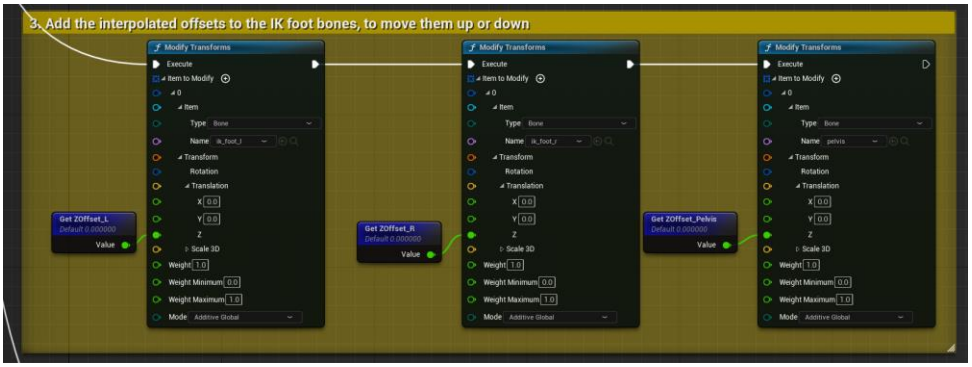
Στη μάχη, ο χαρακτήρας υποστηρίζει επιθέσεις combo. Όταν ο παίκτης ενεργοποιήσει τη βασική επίθεση μέσω της μεθόδου MainAttack(), ο χαρακτήρας εκτελεί την πρώτη κίνηση της επίθεσης. Εάν ο παίκτης συνεχίσει να πατά την επίθεση, το σύστημα combo επιτρέπει τη συνέχιση μιας ακολουθίας από επιθέσεις, αναπαράγοντας μια σειρά από animation montages που συνδέονται μεταξύ τους. Το σύστημα αυτό λειτουργεί με χρονομετρητή, όπου η μέθοδος PlayMontageAndSetupResetTimer() ενεργοποιεί τα διαφορετικά στάδια της επίθεσης, και το ResetCombo() επαναφέρει την αλληλουχία όταν τελειώσει το combo. Επίσης, ο χαρακτήρας μπορεί να εκτελέσει αεροπορική επίθεση όταν βρίσκεται στον αέρα. Αυτή η ειδική επίθεση ενεργοποιείται μέσω της MainAttack()εφόσον ο χαρακτήρας είναι εκτός εδάφους, προσθέτοντας ποικιλία στις κινήσεις του και επιτρέποντας επιθέσεις από τον αέρα

2.5 Control Rig

Εικόνα 16: Sphere trace from feet to ground logic

Εικόνα 17: Interpolate smoothly targets & prevent overextension

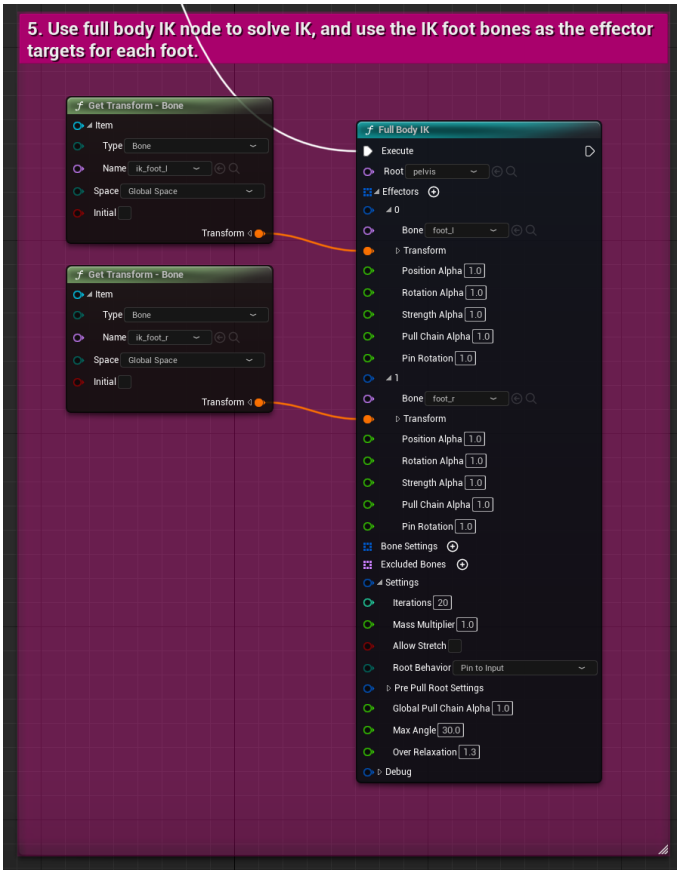
Στο Control Rig που έχει υλοποιηθεί για την προσαρμογή των ποδιών στο έδαφος, η χρήση του Alpha Interpolate σε συνδυασμό με τα Get Z Offset L Target και Z Offset R Target συμβάλλει καθοριστικά στη δημιουργία μιας ομαλής και φυσικής μετάβασης στις θέσεις των ποδιών του χαρακτήρα, καθώς αυτά προσαρμόζονται στις μεταβολές του εδάφους. Το Alpha Interpolate θεωρείται μια διαδικασία που επιτρέπει τη σταδιακή αλλαγή μιας τιμής, ορίζοντας την ανάμεσα σε μια αρχική και μια τελική κατάσταση. Στην περίπτωσή του, χρησιμοποιείται για να υπολογιστεί η ενδιάμεση θέση των ποδιών με βάση τις τρέχουσες και επιθυμητές θέσεις τους στον άξονα Z, δηλαδή το ύψος, εξασφαλίζοντας ότι οι κινήσεις των ποδιών προς τις νέες θέσεις στο έδαφος δεν γίνονται απότομα, αλλά με ομαλό τρόπο.



Εικόνα 18: Offset – IK bones Binding

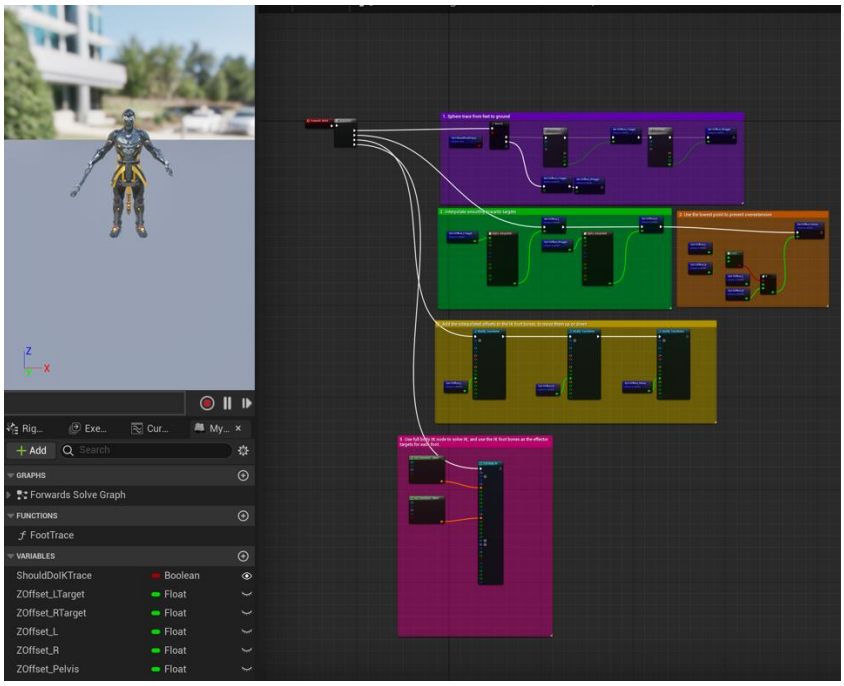
Τα Get Z Offset L Target και Z Offset R Target αντιπροσωπεύουν τους στόχους ύψους για το αριστερό και το δεξί πόδι αντίστοιχα, ανακτώντας τα απαραίτητα δεδομένα από τη γεωμετρία του εδάφους κάτω

από τα πόδια του χαρακτήρα. Με αυτόν τον τρόπο, καθορίζεται πόσο θα πρέπει να μετακινηθεί κάθε πόδι στον άξονα Z, είτε προς τα πάνω είτε προς τα κάτω, ώστε να επιτευχθεί πλήρης επαφή με το έδαφος. Στη συνέχεια, μέσω της διαδικασίας Alpha Interpolation, οι τιμές αυτές εφαρμόζονται σταδιακά και ρυθμίζουν τη θέση του κάθε ποδιού.



Εικόνα 19: Εφαρμογή Control Rig στο Full Mesh

Η χρήση του Alpha Interpolate εξασφαλίζει ότι η αλλαγή στο ύψος των ποδιών είναι ομαλή, αποτρέποντας τυχόν απότομες ή αφύσικες κινήσεις του χαρακτήρα. Αυτό, σε συνδυασμό με τη χρήση των Z Offsets για τον υπολογισμό της σωστής θέσης των ποδιών και τη δυνατότητα προσαρμογής κάθε ποδιού ξεχωριστά ανάλογα με το ύψος του εδάφους που πατάει, διασφαλίζει την αρμονική επαφή των ποδιών με οποιοδήποτε ανισόπεδο έδαφος. Έτσι, αν το ένα πόδι βρίσκεται πάνω σε μια επιφάνεια όπως μια πέτρα, το άλλο θα προσαρμόζεται αυτόνομα στο επίπεδο εδάφους, διατηρώντας τον χαρακτήρα σε φυσική στάση και αποφεύγοντας την αιώρηση του ποδιού. Αυτός ο συνδυασμός τεχνικών επιτυγχάνει ρεαλισμό στις κινήσεις και στη στάση του χαρακτήρα.



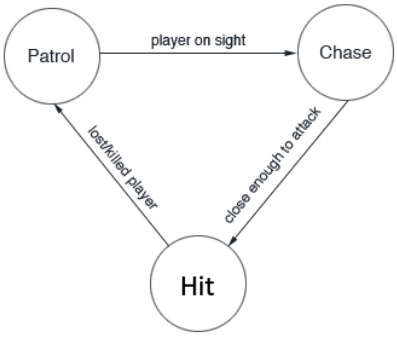
Εικόνα 20: Control Rig Overview

3. Κλάση Enemy – A.I. Controlled Εχθρός

3.1 Απαρίθμηση των δυνατοτήτων του εχθρού

Για την υλοποίηση του A.I driven hostile NPC (εχθρού), από μεριάς της c++, έχουν δημιουργηθεί κάποια states/καταστάσεις, στις οποίες μπορεί να βρίσκεται, βάσει ορισμένων εξωγενών παραγόντων. Συγκεκριμένα οι δυνατότητές του:

- Παρακολούθηση (Patrolling), όπου περιπολεί μεταξύ προκαθορισμένων στόχων.
- Ανίχνευση Παίκτη (Pawn Sensing) όταν αυτός έρχεται εντός της ακτίνας όρασης του εχθρού.
- Καταδίωξη Παικτών (Chasing) μόλις τον ανιχνεύσει, τον καταδιώκει.
- Επίθεση (Attacking) όταν πλησιάσει αρκετά.
- Health and Damage. Δηλαδή μπορεί να λαμβάνει ζημιά και να πεθάνει όταν το hp (health points) του πέσει στο 0.
- Παράβλεψη Παρεμβολών (Melee Collision Overlap): Ενεργοποιεί και απενεργοποιεί την περιοχή επίθεσης για να προσδιορίσει αν χτύπησε τον παίκτη. Όπως λειτουργεί και από μεριάς του AWukongCharacter.



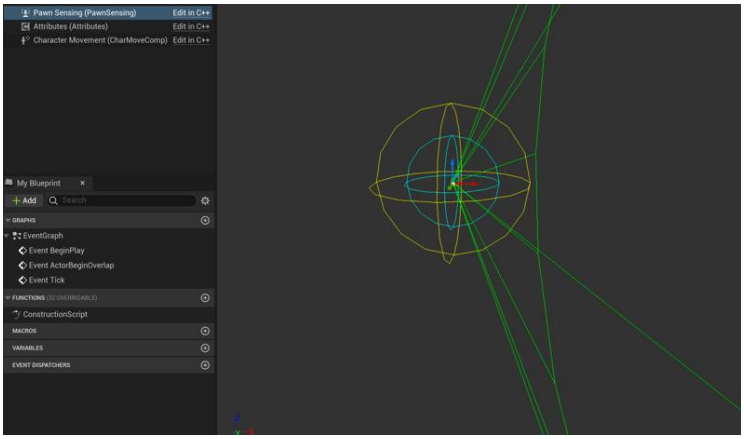
Διάγραμμα 1: Enemy State Logic

3.1.1 Enemy Movement States

Κατά την εκκίνηση του παιχνιδιού, ξεκινά σε κατάσταση περιπολίας (Patrolling) με βάση το σύστημα στόχων περιπολίας που υλοποιείται από τη μέθοδο ChoosePatrolTarget. Κατά τη διάρκεια της περιπολίας, ο εχθρός μετακινείται από το ένα σημείο στο άλλο, όπως έχει καθοριστεί από μια λίστα στόχων περιπολίας.

Μία μέθοδος MoveToTarget αναλαμβάνει να στείλει τον εχθρό προς τον επόμενο στόχο, χρησιμοποιώντας τον controller AI του εχθρού για να καθορίσει την πορεία. Εάν ο εχθρός φτάσει σε έναν στόχο, επιλέγει έναν νέο στόχο περιπολίας με τυχαία σειρά και περιμένει για ένα προκαθορισμένο χρονικό διάστημα πριν συνεχίσει την κίνησή του.

Επιπροσθέτως χρησιμοποιεί ένα PawnSensingComponent, το οποίο του επιτρέπει να ανιχνεύει τον παίκτη μέσα από την όραση (SightRadius). Όταν ο παίκτης βρεθεί εντός της ακτίνας ανίχνευσης, ενεργοποιείται η μέθοδος PawnSeen, η οποία αλλάζει την κατάσταση του εχθρού σε "καταδίωξη" (Chasing). Σε αυτή την κατάσταση, ο εχθρός κινείται προς τον παίκτη, αυξάνοντας την ταχύτητα του, ώστε να κλείσει την απόσταση. Ο εχθρός παρακολουθεί συνεχώς την απόσταση από τον παίκτη μέσω της μεθόδου CheckCombatTarget. Όταν ο παίκτης βρίσκεται έξω από την ακτίνα ανίχνευσης, ο εχθρός επιστρέφει στην κατάσταση περιπολίας, ενώ εάν ο παίκτης βρεθεί εντός ακτίνας επίθεσης, ο εχθρός προετοιμάζεται να επιτεθεί.



Εικόνα 20: Enemy Pawn Sensing Component

Όλες οι παραπάνω δυνατότητες κινητικότητας του εχθρού, μπορούν να πραγματοποιηθούν μόνο εντος των NavMeshBounds Volumes, εντός των οποίων εχουν τοποθετηθεί τα meshes των εχθρών.

Εικόνα 21: NavMeshBounds Volume

3.1.2 Enemy Combat States

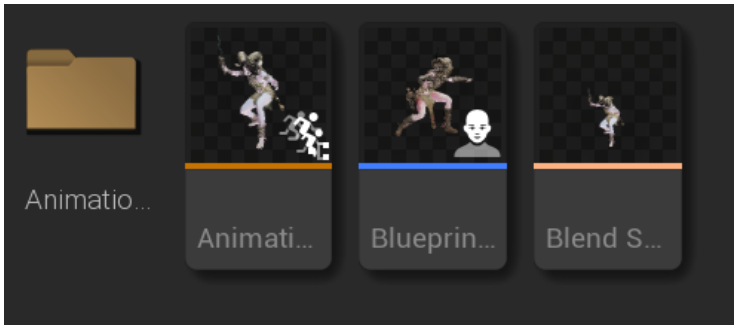
Η επίθεση του εχθρού είναι χρονικά καθορισμένη μέσω της μεθόδου StartAttackTimer, η οποία καθορίζει ένα τυχαίο χρονικό διάστημα για την εκτέλεση της επίθεσης. Όταν ξεκινά η επίθεση, ο εχθρός παίζει ένα animation επίθεσης μέσω του animation montage που έχει οριστεί, και η σύγκρουση του όπλου ενεργοποιείται προσωρινά, επιτρέποντας τη σύγκρουση με τον παίκτη. Αυτή η διαδικασία υλοποιείται μέσω των activateEnemyMelee και deactivateEnemyMelee, που ελέγχουν την κατάσταση του collision του όπλου. Εάν ο παίκτης χτυπηθεί κατά τη διάρκεια αυτής της περιόδου, η μέθοδος EnemyMeleeOverlap ενεργοποιείται και υπολογίζει τη ζημιά που πρέπει να επιβληθεί στον παίκτη. Παρομοίως με την αντίστροφη λειτουργικότητα.

Όταν ο εχθρός δεχθεί χτύπημα από τον παίκτη, η μέθοδος TakeDamage υπολογίζει τη «ζημιά» και μειώνει τα health points του. Εάν η υγεία του φτάσει στο μηδέν, ενεργοποιείται η μέθοδος die, η οποία αναπαράγει το animation θανάτου του και σταματά να αλληλεπιδρά με τον κόσμο, ενώ μετά από μερικά δευτερόλεπτα καταστρέφεται.

Η συμπεριφορά του εχθρού καθορίζεται από τις διαφορετικές καταστάσεις (states) που αναφέρονται ως EEnemyState. Ξεκινώντας λοιπόν, στην κατάσταση περιπολίας (Patrolling) και προχωρά στην κατάσταση καταδίωξης (Chasing) όταν εντοπίσει τον παίκτη. Μόλις φτάσει αρκετά κοντά για να επιτεθεί, περνά στην κατάσταση επίθεσης (Attacking) και, αφού ολοκληρώσει την επίθεση, μεταβαίνει στην κατάσταση ετοιμότητας (Engaged) ή επιστρέφει στην καταδίωξη εάν ο παίκτης παραμένει κοντά. Αντίστοιχα, όταν ο εχθρός πεθάνει, η κατάστασή του αλλάζει σε "νεκρός" (Dead), και όλες οι άλλες ενέργειες σταματούν.

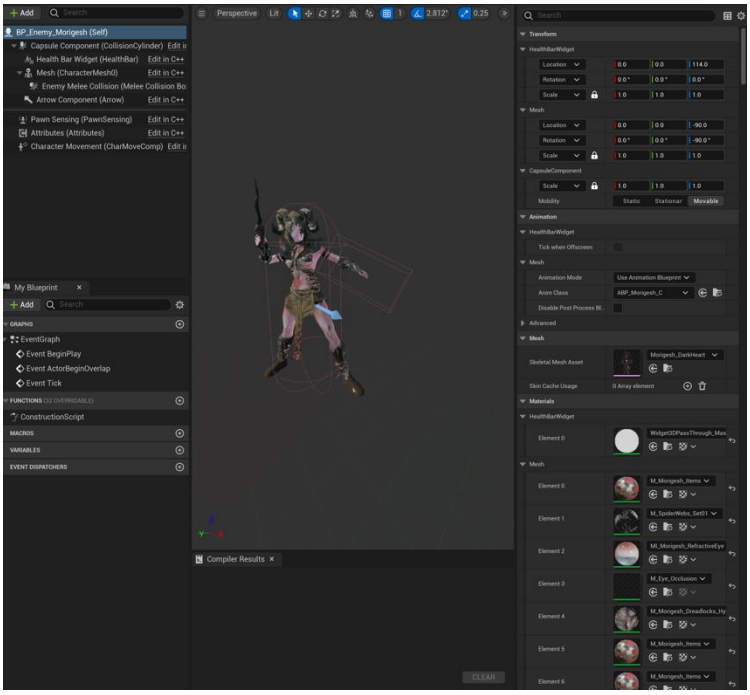
3.2 Enemy Blueprints

Από πλευράς των blueprints έχουν δημιουργηθεί, ένα animation blueprint ένα character blueprint καθώς και ένα blendspace τα οποια δρούν με παρόμοιο, αλλα απλούστερο, τρόπο με του AWukongCharacter.



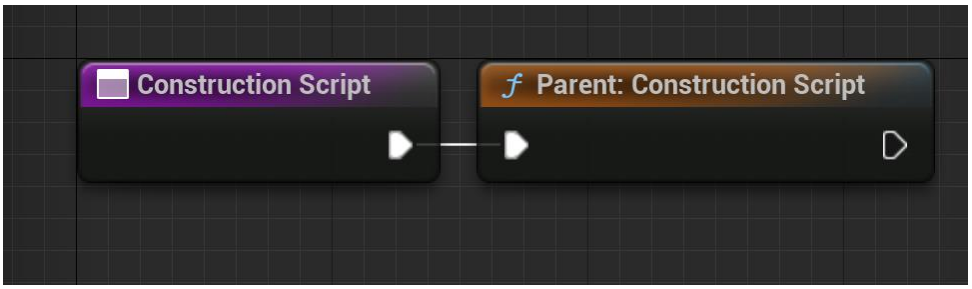
Εικόνα 22: Enemy Blueprint Files

3.2.1 Blueprint & Components



Εικόνα 23: Enemy Blueprint Viewport

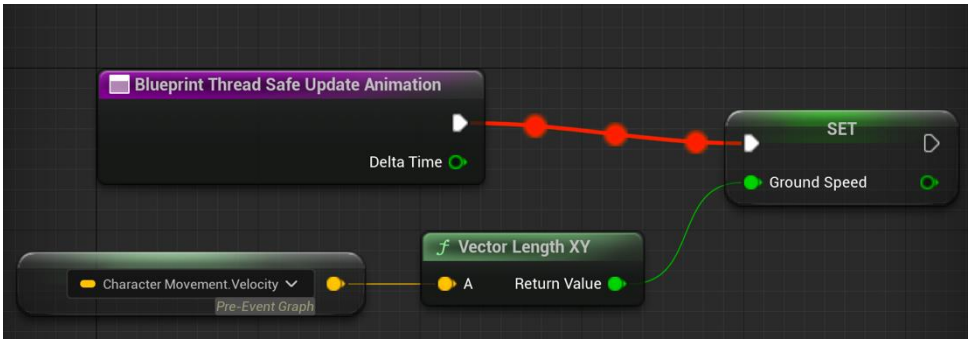
Τα components τα οποία κάνει χρήση το Enemy Blueprint αποτελούν ξανά όμοια και λιγότερο σύνθετη απεικόνιση εκείνων του βασικού χαρακτήρα του παίκτη. Με μόνες διάφορες το pawn sensing component για την περιήγηση του A.I., όπως αναλύθηκε παραπάνω, καθώς και την επιπρόσθετη εισαγωγή του component; attributes, για την κατάλληλη διαχειρηση των healthpoints του enemy character από την c++.



Εικόνα 24: Enemy Construction Script Initialization

3.2.2 Animation Blueprint

Το animation blueprint της παρούσας κλάσεις έχει διαφορετική υλοποίηση από εκείνη της προηγούμενης. Αν και επιτελούν αρκετά όμοιους στόχους, στην προκειμένη περίπτωση η διαχειριστική θέση που κατέχει το A.I. οδηγεί σε ορισμένες βασικές διαφοροποιήσεις.



Εικόνα 25: Update Animations

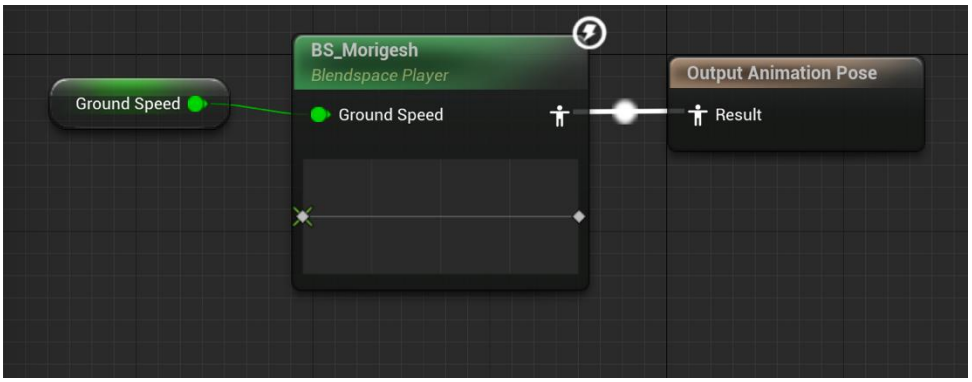
Ξεκινώντας από τον κόμβο "Thread Safe Update Animation Node", ο οποίος είναι σημαντικός για τη διαχείριση της ανανέωσης των animation states. Εξασφαλίζει ότι οι αλλαγές που γίνονται στις παραμέτρους του animation είναι ασφαλείς από πλευράς νήματος, δηλαδή ότι δεν υπάρχουν συγκρούσεις κατά τη διάρκεια της εκτέλεσης.

Στο επόμενο βήμα, η ταχύτητα του εχθρού ρυθμίζεται χρησιμοποιώντας την τιμή του Vector Length, η οποία προέρχεται από το Get Character Movement Velocity. Αυτή η λειτουργία αντλεί τις τρέχουσες ταχύτητες του χαρακτήρα σε τρεις διαστάσεις (X, Y, Z) και επιστρέφει το συνολικό μέγεθος του διανύσματος ταχύτητας. Έτσι, η ταχύτητα αυτή μπορεί να χρησιμοποιηθεί για να ρυθμίζει τη μετάβαση των animations του εχθρού, επιτρέποντας στον εχθρό να αντιδρά με φυσικό τρόπο στις κινήσεις του.

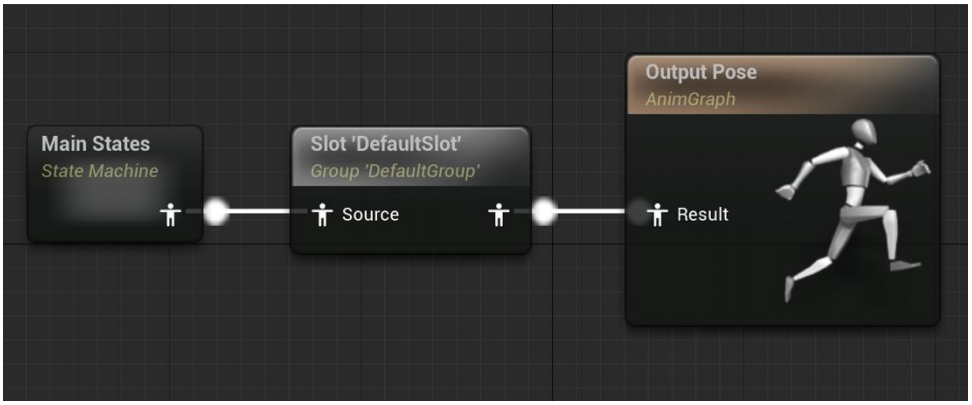
Για παράδειγμα, αν ο εχθρός κινείται γρήγορα, μπορεί να επιλέξει ένα animation που αντικατοπτρίζει τη γρήγορη κίνηση, όπως τρέξιμο ή επίθεση, ενώ αν είναι ακίνητος ή κινείται αργά, το animation μπορεί να είναι πιο ήρεμο, όπως η περιπολία ή η στάση.

Έτσι λοιπόν, η ρύθμιση της ταχύτητας του εχθρού μέσω του Vector Length επιτρέπει την άμεση σύνδεση ανάμεσα στη φυσική κίνηση του χαρακτήρα και την αναπαράσταση των animations, διασφαλίζοντας ότι οι εχθροί θα φαίνονται και θα συμπεριφέρονται όπως αναμένονται από τους παίκτες, χωρίς να αποσυνδέονται από την πραγματικότητα του gameplay.

Αντιθέτως το animgraph της κλάσης του εχθρού αποτελεί μια απλοποιημένη εκδοχή, εκείνης του χαρακτήρα.



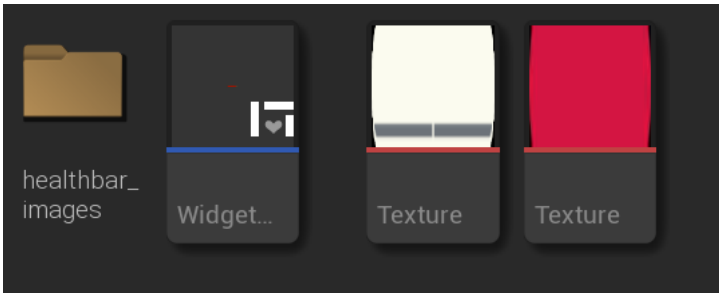
Εικόνα 26: Χρήση Enemy BlendSpace



Εικόνα 27: Enemy Animgraph

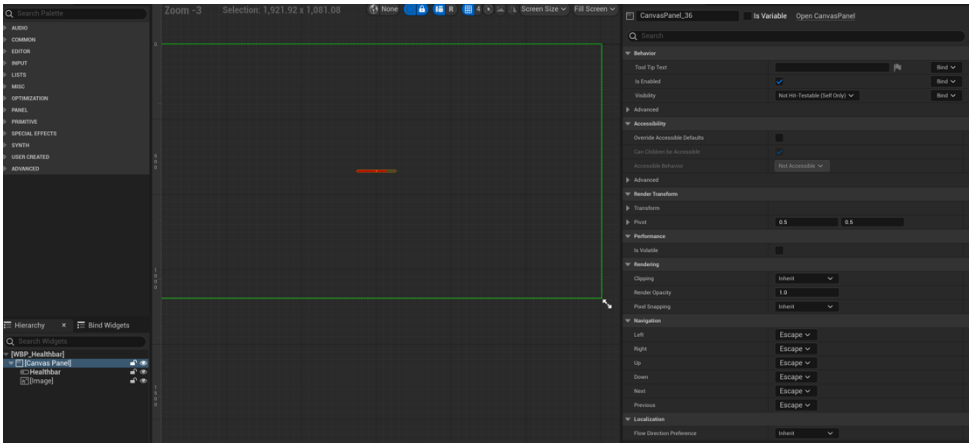
3.2 HUD (Heads Up Display)

Πρόκειται ουσιαστικά για το healthbar widget που εμφανίζεται ακριβώς επάνω από το mesh του εχθρού στον κόσμο, μόνο όταν και αν του επιτεθεί ο παίκτης.



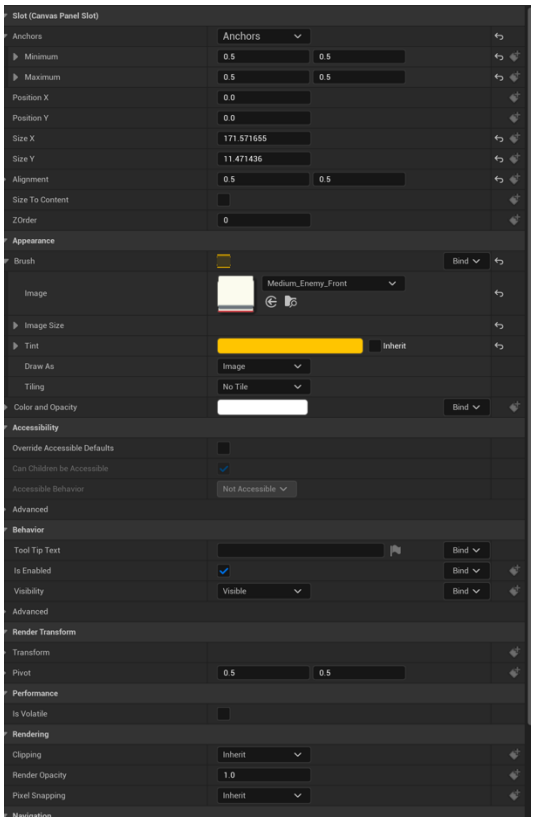
Εικόνα 28: Healthbar files

Έχει δημιουργηθεί λοιπόν το κατάλληλο widget blueprint και έχουν εισαχθεί 2 εικόνες ως textures. Η πρώτη απεικονίζει το περίβλημα της health bar και η δεύτερη το περιεχώμενό της.



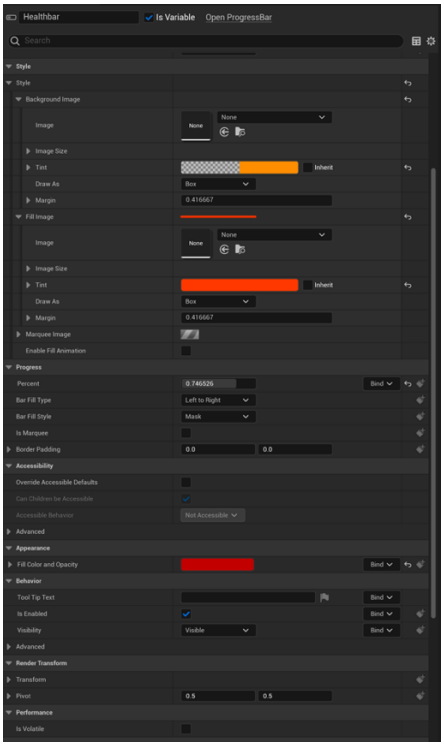
Εικόνα 29: Healthbar Widget

Στο healthbar widget blueprint έχει τοποθετηθεί ένα canvas panel που λειτουργεί ως η οθόνη του χρήστη.



Εικόνα 30: Healthbar Blueprint Settings 1

Ακολουθώντας, το περίβλημα του healthbar είναι απλά μία εικόνα η οποία περικλείει ένα component τύπου progress bar. Η progress bar αυτή έχει τεθεί ως μεταβλητή και συνδεθεί από C++ με την μεταβλητή health της κλασης AEnemy.



Εικόνα 30: Healthbar Blueprint Settings 2

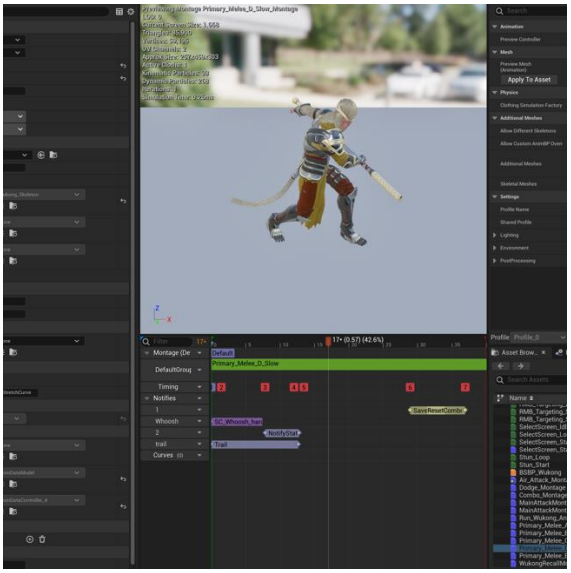
4. ΉΧΟΣ

Για την αναπαραγωγή ήχου, χρησιμοποιήθηκαν τόσο SoundWaves όσο και MetaSound. Τα SoundWaves είναι η πιο απλή μορφή ηχητικών αρχείων στο Unreal Engine, ενώ οι MetaSounds είναι ένα πιο σύνθετο σύστημα που σου επιτρέπει να δημιουργούμε δυναμικούς, προσαρμοστικούς ήχους χρησιμοποιώντας γραφήματα και κόμβους (nodes).

Εικόνα 31: Audio Files

4.1 Απλά Soundwaves

Τα απλής μορφής soundwaves χρησιμοποιήθηκαν σε ορισμένα animation montages όπου ήταν επιθυμητό να ακούγεται ο ήχος ακριβώς έτσι όπως είναι χωρίς κάποια μεταβολή στην συχνότητα ή και την έντασή του.

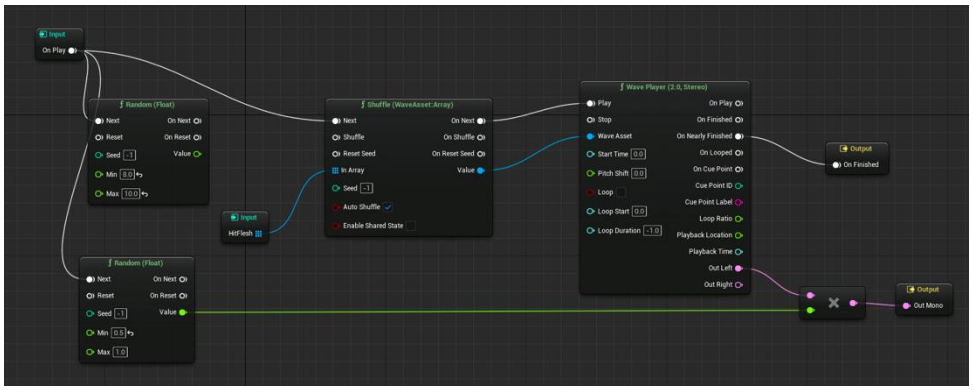


Εικόνα 32: Εφαρμογή Soundwaves

Συγκεκριμένο παράδειγμα αποτελεί το attack montage D (το 4ο σε σειρά motage του combo επιθέσεων του Wukong). Όπου πρόκειται για εμφανές heavy attack - δηλαδή επίθεση που θα έκανε περισσότερο θόρυβο από τις υπόλοιπες- και συνεπώς δεν θα ήταν επιθυμητό να είχε μικρότερο pitch από κάποια άλλη πιο αδύναμη.

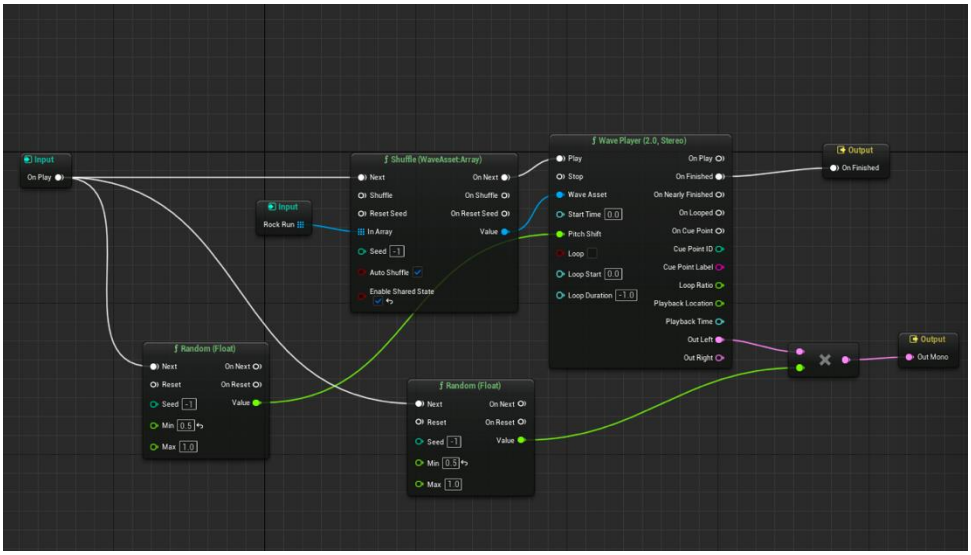
4.2 Metasounds

Αντιθέτως με τα απλά sound waves, όπως αναφέρθηκε, τα metasounds παρέχουν την ποικιλόμορφη επεξεργασία του ήχου μέσω των event graphs, με αποτέλεσμα πιο αληθοφανή και ζωντανό ηχητικό περιεχώμενο.



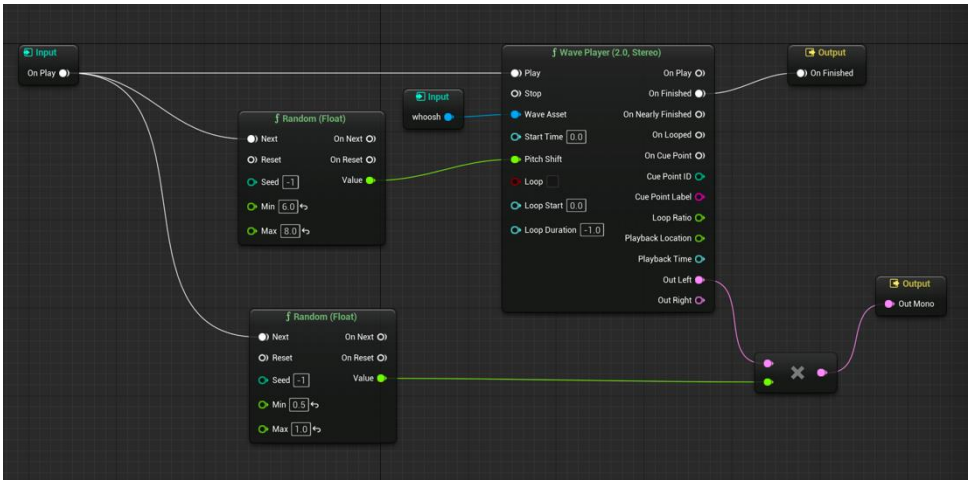
Εικόνα 33: "Whoosh" Metasound

Η εισαγωγή του event είναι συνδεδεμένη με την αναπαραγωγή ενός τυχαίου ήχου με τη χρήση της λειτουργίας Random Float για να επιλεγεί μία τυχαία τιμή μεταξύ δύο αριθμών. Αυτή η τυχαία τιμή καθορίζει πόσο συχνά και με ποια ποικιλία θα ακούγεται ο ήχος "whoosh", που είναι αναπαράσταση ενός συγκεκριμένου ηχητικού αρχείου. Ένας Random κόμβος δίνει μια τιμή μεταξύ 6 και 8 δευτερολέπτων για την αναπαραγωγή και ένας δεύτερος Random Float κόμβος, δίνει μια τυχαία τιμή για το Pitch Shift του ήχου, δηλαδή πόσο θα αλλάξει η συχνότητα, δίνοντας μεγαλύτερη παραλλαγή στον ήχο. Τέλος, ο κόμβος Wave Player αναπαράγει το ηχητικό αρχείο, το οποίο περνάει από έναν κόμβο μίξης και τελικά στέλνεται στην έξοδο (Output) ως Mono.



Εικόνα 34: "Steps" Metasound

Παρόμοιως με το προηγούμενο αλλά αφορά ένα event όπου επιλέγεται τυχαία ένα ηχητικό αρχείο μέσα από μια σειρά ήχων (Array) που σχετίζονται με βήματα. Η διαφορά είναι ότι εδώ χρησιμοποιείται ένας κόμβος Shuffle για την τυχαία αναπαραγωγή των ηχητικών αρχείων από μια λίστα (wave asset array), προσδίδοντας επιπλέον ποικιλία στον ήχο. Επίσης, ο κόμβος Random Float παρέχει τυχαία τιμή μεταξύ 0.5 και 1 για τον προσδιορισμό του pitch shift του ήχου.



Εικόνα 35: "Hit" Metasound

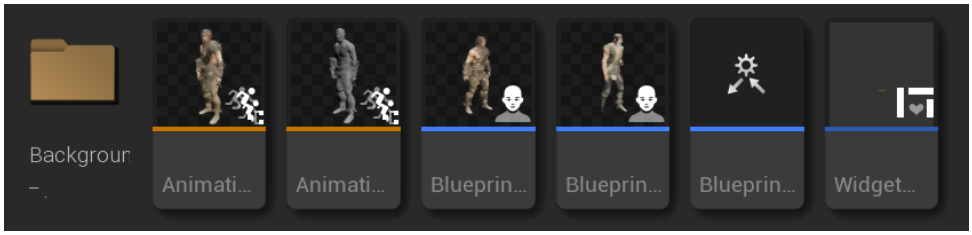
Στο τελευταίο metasound, το event αφορά έναν ήχο που σχετίζεται με το "HitFlesh". Ο ήχος αυτός αναπαράγεται με έναν παρόμοιο μηχανισμό, όπως στις προηγούμενες εικόνες, αλλά εδώ η επιλογή του ηχητικού αρχείου γίνεται επίσης τυχαία από μια λίστα με το shuffle node να παίζει τον ρόλο του τυχαίου αναμειγνύοντας διαφορετικά ηχητικά αρχεία κάθε φορά. Παράλληλα, δύο Random Float nodes ρυθμίζουν το pitch shift αλλά και το χρόνο καθυστέρησης πριν την αναπαραγωγή του ήχου, προσδίδοντας μεγαλύτερη δυναμική και ποικιλία στην αλληλεπίδραση του ήχου με τις ενέργειες του παίκτη.

5. NPC

Στον κόσμο περιλαμβάνονται δύο είδη NPC: οι χωρικοί και οι δράκοι.

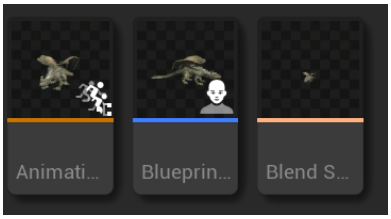
Εικόνα 36: NPC Files

Και οι δύο κατηγορίες είναι μη επιθετικοί χαρακτήρες. Οι χωρικοί διακρίνονται σε δύο υποκατηγορίες. Οι πρώτοι είναι οι χωρικοί που λειτουργούν ως background characters. Αυτοί δεν έχουν καμία λειτουργία αλληλεπίδρασης, υπάρχουν μόνο για να γεμίζουν τον χώρο, δημιουργώντας έτσι μια αίσθηση ζωντάνιας και πληθυσμού. Η δεύτερη υποκατηγορία αφορά τους κανονικούς χωρικούς, με τους οποίους ο παίκτης μπορεί να συνομιλήσει, δημιουργώντας έτσι μια αλληλεπίδραση που μπορεί να προσφέρει περισσότερες πληροφορίες ή άλλες δυνατότητες μέσα στο παιχνίδι.



Εικόνα 37: Villager Files

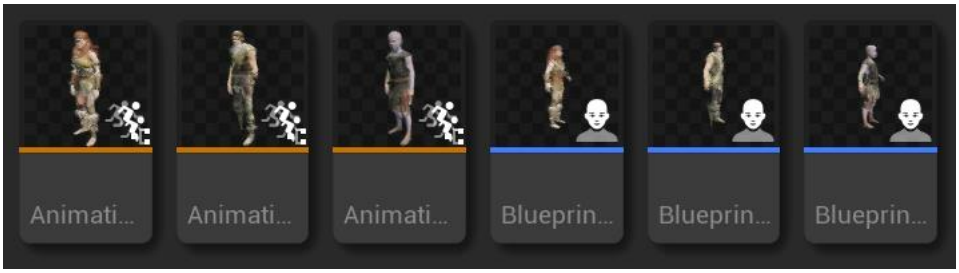
Από την άλλη, οι δράκοι μπορούν να περιηγούνται στον χώρο με τη χρήση του Nav Mesh Bounds Volume, όπως και ο εχθρικός χαρακτήρας. Ωστόσο, σε αντίθεση με τους χωρικούς, δεν υπάρχει η δυνατότητα αλληλεπίδρασης μεταξύ των δράκων και του παίκτη. Αν και είναι μη επιθετικοί, η παρουσία τους συμβάλλει στη δημιουργία του κόσμου, ενισχύοντας τη συνολική ατμόσφαιρα και κίνηση μέσα στο περιβάλλον του παιχνιδιού.



Εικόνα 38: Dragon Files

5.1 NPC Χωρικοί

5.1.1 Background Characters

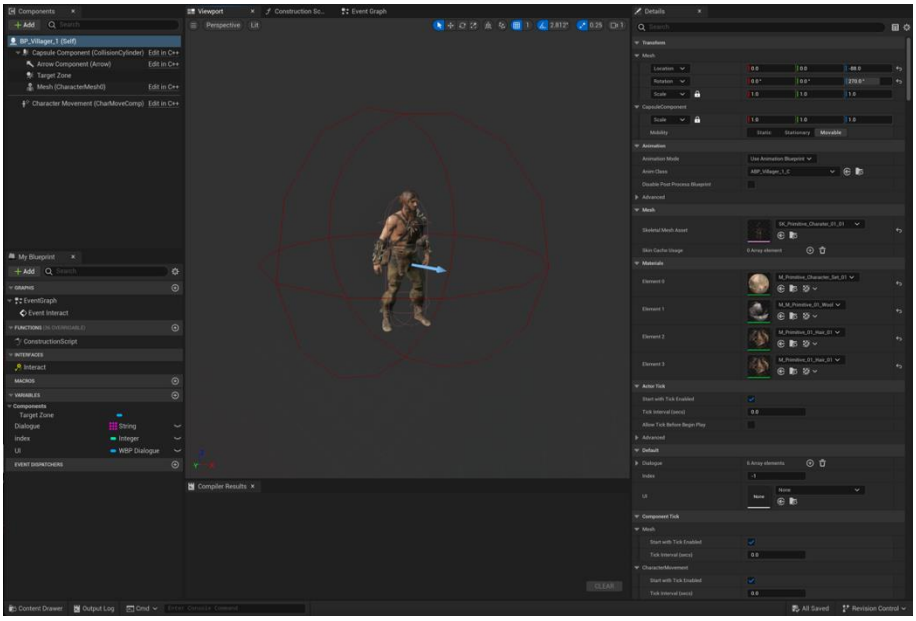


Εικόνα 39: Background Characters Files

Στην περίπτωση των background characters, η λειτουργία τους είναι απλή και περιορισμένη σε σχέση με τους κανονικούς χωρικούς ή τους διαλόγους. Αυτοί οι χαρακτήρες έχουν μόνο ένα character blueprint και ένα animation blueprint, και η κίνησή τους περιορίζεται στην αναπαραγωγή ενός απλού idle animation.

Ο σκοπός αυτής της διάταξης είναι να γεμίσει τον χώρο του παιχνιδιού με επιπλέον χαρακτήρες που δίνουν μια αίσθηση ζωντάνιας και πληρότητας στο περιβάλλον, χωρίς να επιβαρύνεται το σύστημα με περίπλοκες λειτουργίες αλληλεπίδρασης ή συμπεριφοράς. Οι background characters δεν συμμετέχουν ενεργά στην πλοκή ή τις δράσεις του παίκτη, αλλά συμβάλλουν στην αισθητική και την ατμόσφαιρα του παιχνιδιού.

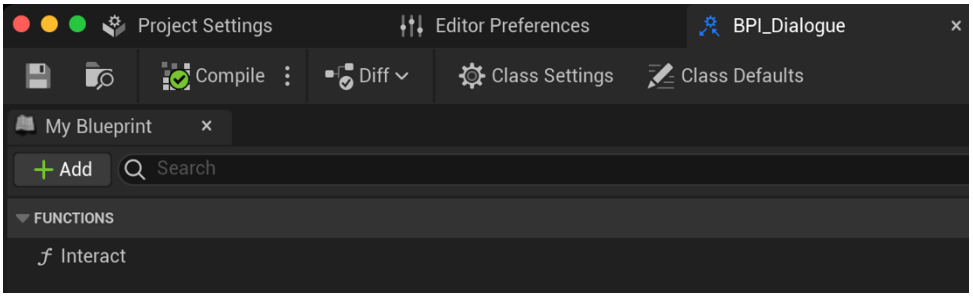
5.1.2 Χωρικοί (αλληλεπίδρασης)



Εικόνα 40: Villager1 Viewport

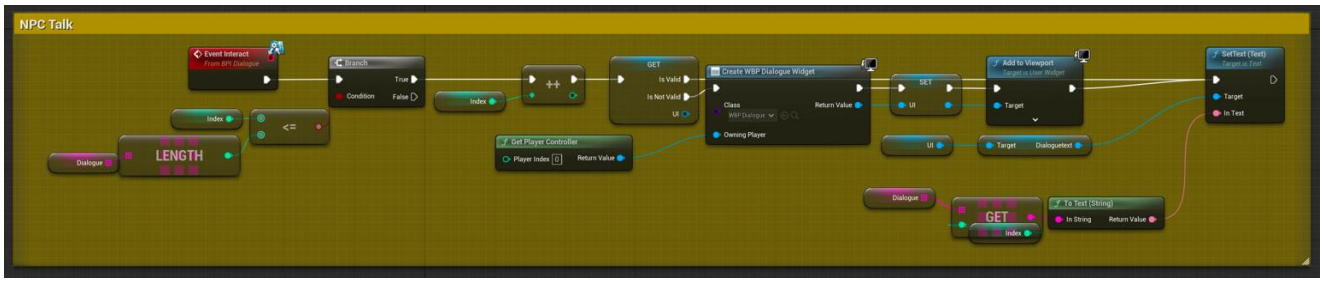
Οι χωρικοί, με τους οποίους ο παίκτης μπορεί να αλληλεπιδράσει πιέζοντας το πλήκτρο "E", διαθέτουν ένα επιπρόσθετο component που ονομάζεται "target zone". Όταν αυτό το component επικαλύπτεται από τον παίκτη και καλείται η κατάλληλη συνάρτηση, όπως περιγράφεται στην ενότητα 2.2.3, πραγματοποιείται η εκκίνηση της αλληλεπίδρασης μεταξύ των δύο οντοτήτων. Αυτή η διαδικασία επιτρέπει την ενεργοποίηση διαλόγου ή άλλης μορφής αλληλεπίδρασης, ενισχύοντας την εμπειρία του παίκτη και προσφέροντας πρόσθετες λειτουργίες στο gameplay.

Η προκείμενη κλάση κάνει χρήση της Interface BPI_Dialogue. Εκεί είναι δηλωμένη η συνάρτηση interact.



Εικόνα 41: Blueprint Dialogue Interface

Στο Event Graph των Villagers υλοποιείται η προαναφερόμενη συνάρτηση με τον ακόλουθο τρόπο.



Εικόνα 42: Villager Talk

Επεξηγώντας τα προαναφερόμενα το event interact, οδηγεί σε έναν κόμβο branch. Ο κόμβος αυτός ελέγχει αν η μεταβλητή index είναι μικρότερη ή ίση με το μήκος του πίνακα που περιέχει τα strings διαλόγου. Σε περίπτωση που η συνθήκη είναι αληθής, αυξάνεται η τιμή του index κατά ένα. Αμέσως μετά, γίνεται χρήση ενός getter του index, ο οποίος καθοδηγεί την επόμενη ενέργεια, που είναι η εκχώρηση του κειμένου σε έναν κόμβο set text με στόχο το UI σε μορφή διαλόγου. Το in text καθορίζεται μέσω ενός getter που αντλεί το string από τον πίνακα διαλόγου, αντιστοιχώντας το τρέχον στοιχείο του array στο UI.

Αν όμως, μετά την αύξηση του index, ο getter δεν είναι valid, η ροή οδηγείται στη δημιουργία ενός wbp dialogue widget, συνδεδεμένο με έναν κόμβο get player controller και επιλέγοντας ως κλάση το WP Dialogue. Αμέσως μετά, αυτό οδηγεί σε έναν setter του UI. Ακολουθεί η εντολή add to viewport, η οποία προσθέτει το widget στο παράθυρο προβολής. Τελικά, η διαδικασία

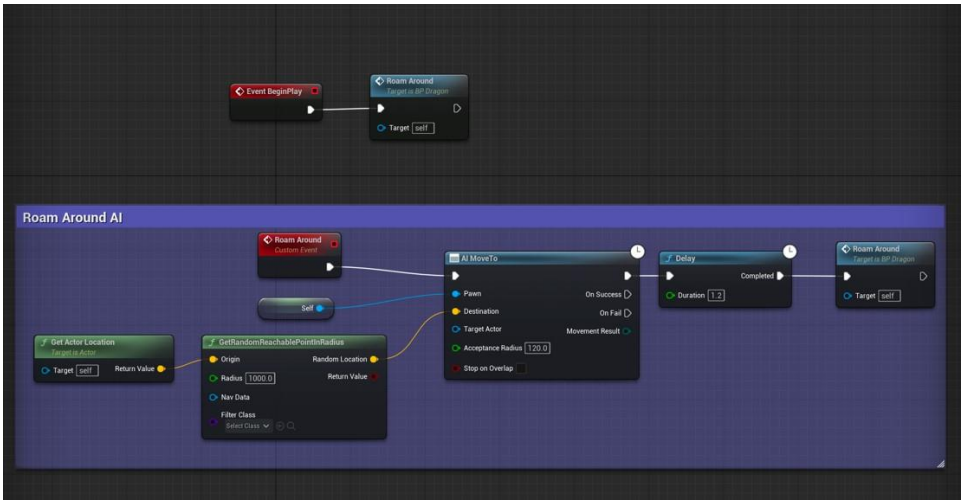
καταλήγει και πάλι στο set text, με τον ίδιο τρόπο που περιγράφηκε προηγουμένως, θέτοντας το κατάλληλο κείμενο από τον πίνακα διαλόγου στο UI.

Ουσιαστικά, η ροή αυτή επιτρέπει στον παίκτη να προχωρήσει μέσα σε μια σειρά διαλόγων, οι οποίοι αποθηκεύονται σε έναν πίνακα διαλόγου (dialogue array), με τρόπο που προσομοιώνει μια συνομιλία βήμα προς βήμα. Εάν λοιπόν υπάρχουν ακόμη διαθέσιμα στοιχεία στον πίνακα, το κείμενο ενημερώνεται στο UI, επιτρέποντας στον παίκτη να διαβάσει το επόμενο κομμάτι της συνομιλίας. Αν όμως ο *index* υπερβεί το μέγεθος του πίνακα, η ενέργεια προχωρά στη δημιουργία ενός νέου widget διαλόγου (*wbp dialogue widget*), το οποίο επιτρέπει την προβολή νέων μηνυμάτων ή άλλων σχετικών πληροφοριών.

5.2 Δράκοι

5.2.1 Dragon Blueprint

Ο δράκος αποτελείται επίσης, από ένα non-hostile ai npc, μέσα σε μία σπηλιά στην νοτιοδυτική γωνία του τοπίου και όπου και περιηγείται .



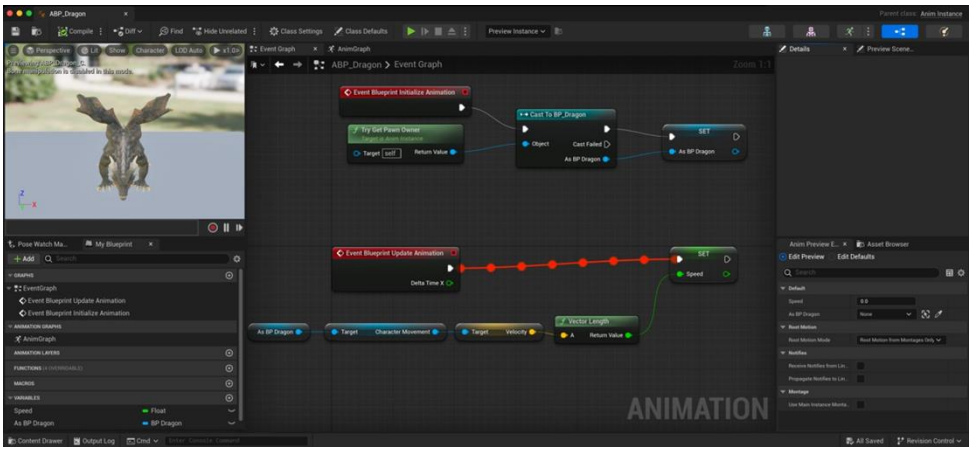
Εικόνα 43: Dragon Roam Around

Στο blueprint που έχει δημιουργηθεί, χρησιμοποιείται ένα custom event το οποίο ενεργοποιείται κατά την έναρξη του παιχνιδιού μέσω του κόμβου Begin Play. Το συγκεκριμένο custom event φέρει την ονομασία "roam around" και διαχειρίζεται τη συμπεριφορά της κίνησης του δράκου στον χώρο. Κατά την εκτέλεση αυτού του event, το σύστημα κατευθύνεται αρχικά προς τον κόμβο AI Move To, όπου καθορίζεται ότι το pawn είναι το ίδιο το AI (δηλαδή, ο δράκος). Ως προορισμός, χρησιμοποιείται η τιμή που επιστρέφεται από τη συνάρτηση Get Random Reachable Point in Radius, η οποία καθορίζεται βάσει της τρέχουσας τοποθεσίας του actor με τη χρήση της Get Actor Location.
Αυτός ο μηχανισμός εξασφαλίζει ότι το AI θα μετακινηθεί τυχαία μέσα σε έναν συγκεκριμένο ακτίνα γύρω από τη θέση του χαρακτήρα. Η τυχαία επιλογή τοποθεσίας προσδίδει μια φυσική και απρόβλεπτη κίνηση, γεγονός που ενισχύει τον ρεαλισμό της συμπεριφοράς περιπολίας του εχθρού. Μόλις ολοκληρωθεί η κίνηση, ακολουθεί ένα Delay διάρκειας 1.2 δευτερολέπτων, επιτρέποντας στον δράκο να παραμείνει για λίγο στάσιμος στη νέα του θέση, προσδίδοντας έναν πιο φυσικό ρυθμό στην κίνησή του. Αμέσως μετά το πέρας του delay, το σύστημα οδηγείται εκ νέου στο ίδιο custom event "roam around", το οποίο προκαλεί επανάληψη του κύκλου. Η

παραπάνω διαδικασία διαδραματίζεται εντός των ορίων ενός NavMeshBounds Volume, όπως συμβαίνει και στην περίπτωση του Enemy.

5.2.1 Dragon Animation Blueprint

Το animation blueprint του δράκου αποτελεί μια ακόμα απλούστερη εκδοχή εκείνης του enemy μιας και ο σκοπός του είναι η περιήγηση.
Έτσι λοιπόν στο παρόν animation event graph μας ενδιαφέρει να ενημερώνονται τα animation properties αναλόγως με την ταχύτητα που κατέχει το NPC

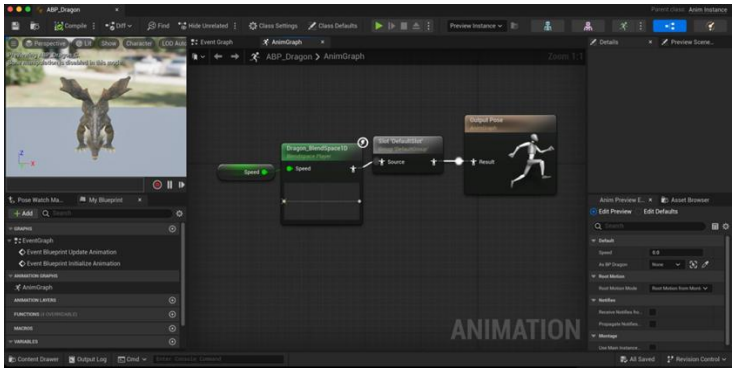


Εικόνα 44: Dragon EventGraph

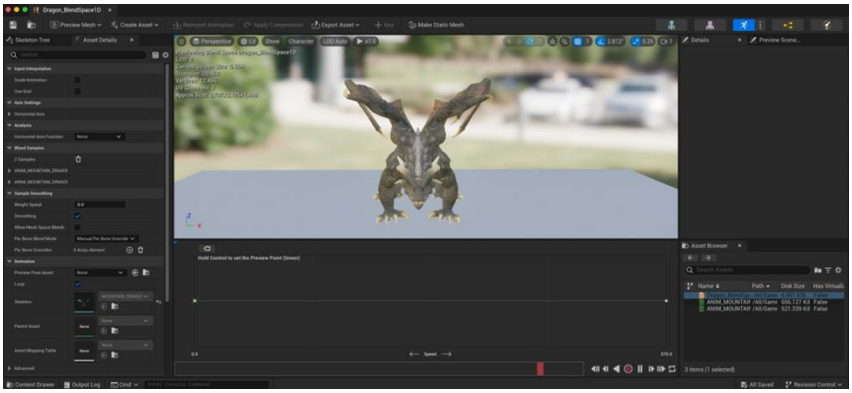
Η μεταβλητή speed καθορίζει την ταχύτητα με την οποία μπορεί να κινείται ο δράκος.

Το initialize animation event ξεκινά το animation ενώ το update animation το επαναλαμβάνει όταν ο δράκος έχει αναπτύξει ταχύτητα. Έτσι ώστε να μη λαμβάνει χώρα το walking animation ενώ ο δράκος να είναι idle.

Όσον αφορά το AnimGraph, η ταχύτητα εισάγεται σε ένα reference του blendspace, όπου -όπως και στις προηγούμενες 2 περιπτώσεις- αποτελεί τον οριζόντιο άξονά του.



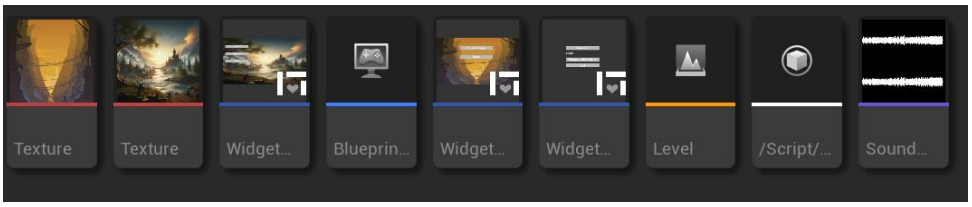
Εικόνα 45: Dragon Animgraph



Εικόνα 45: Dragon BlendSpace

6. ΜΕΝΟΥ

Απαραίτητο στοιχείο ενός παιχνιδιού είναι τα μενού του, τα οποία διαδραματίζουν σημαντικό ρόλο στην πλοήγηση και τη διαχείριση των επιλογών από τον χρήστη. Για το συγκεκριμένο project, έχουν δημιουργηθεί τρία βασικά μενού: το αρχικό μενού, το μενού παύσης και το μενού επιλογών.

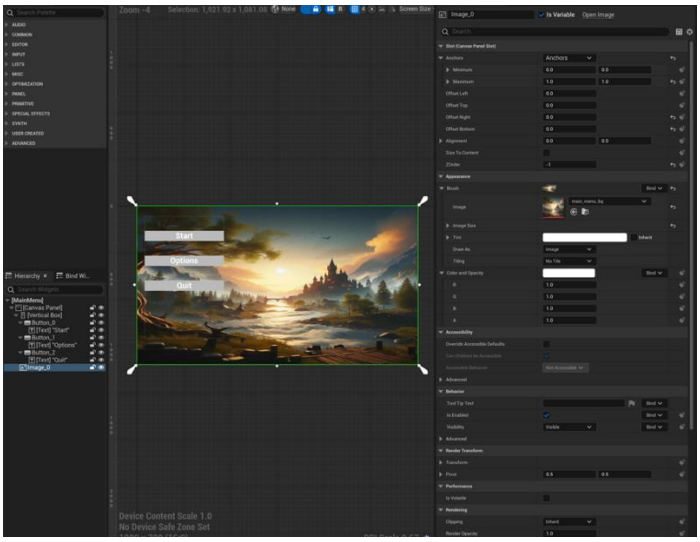


Εικόνα 47: Menu Files

Η ύπαρξη αυτών των μενού ενισχύει την συνολική εμπειρία του χρήστη, καθιστώντας το παιχνίδι περισσότερο διαδραστικό και φιλικό προς τον παίκτη, ενώ παράλληλα εξασφαλίζει την ομαλή ροή της διαδικασίας από την έναρξη ως την ολοκλήρωση της εμπειρίας.

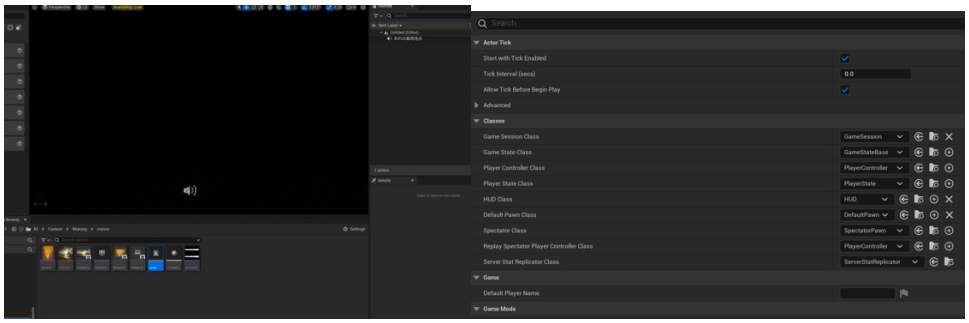
6.1 Αρχικό Μενού

Το αρχικό μενού αποτελεί το σημείο εκκίνησης του παιχνιδιού, επιτρέποντας στον χρήστη να ξεκινήσει το παιχνίδι, να επιλέξει διάφορες ρυθμίσεις ή να εξέλθει από την εφαρμογή.



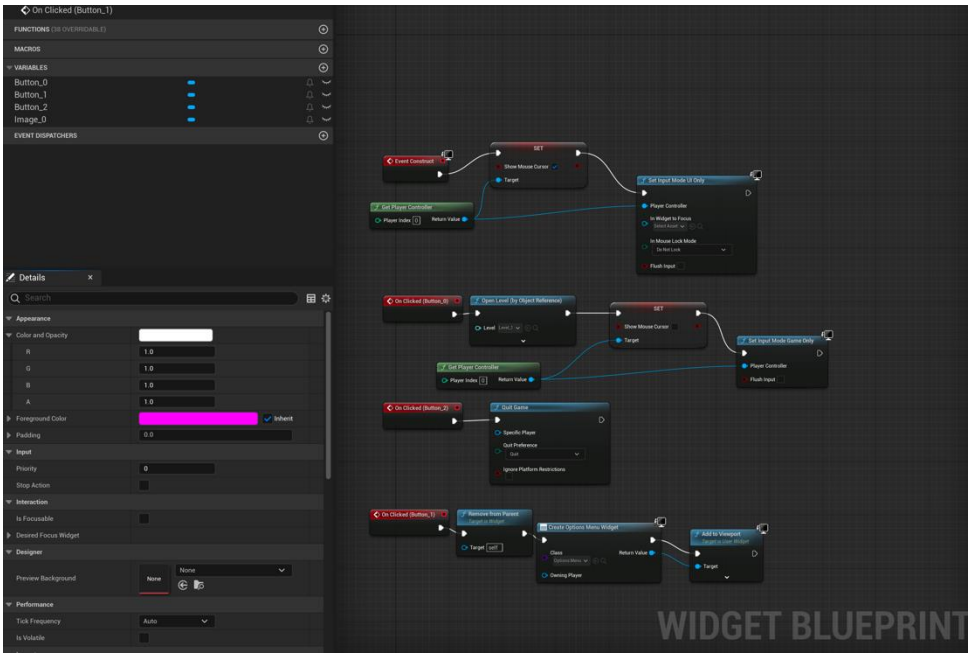
Εικόνα 48: Menu Widget

Για την υλοποίηση του αρχικού μενού του παιχνιδιού, δημιουργήθηκε ένα ειδικό level το οποίο δεν περιέχει τίποτα άλλο εκτός από έναν Player Sound Wave για την αναπαραγωγή μουσικής στο παρασκήνιο.



Εικόνα 49: Menu Level & Gamemode Settings

Επιπλέον, έχει δημιουργηθεί ένα ξεχωριστό GameMode, ώστε να αποτρέπεται το spawn του βασικού χαρακτήρα (AWukongCharacter) όταν ο παίκτης βρίσκεται είτε στο αρχικό μενού είτε στο μενού επιλογών.



Εικόνα 50: Main Menu EventGrpaph

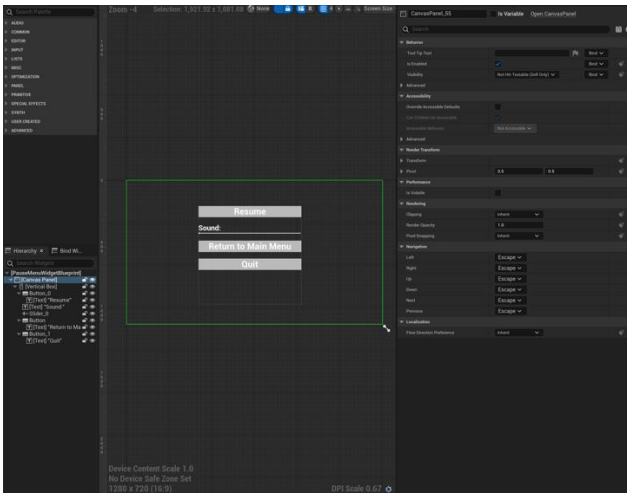
Η δομή του αρχικού μενού βασίζεται σε Widgets και περιλαμβάνει κουμπιά για τις επιλογές Start, Options, και Quit. Αυτά τα κουμπιά συνδέονται με κατάλληλες λειτουργίες που επιτρέπουν την έναρξη του παιχνιδιού, την εμφάνιση του μενού επιλογών, και την έξοδο από το παιχνίδι αντίστοιχα. Όπως φαίνεται στο blueprint, το κουμπί για την έναρξη του παιχνιδιού (Start) ανοίγει το επόμενο επίπεδο μέσω της εντολής Open Level, ενώ το κουμπί για τις επιλογές (Options) αφαιρεί το αρχικό μενού από την οθόνη και δημιουργεί το widget για το μενού επιλογών. Το κουμπί εξόδου (Quit) ενεργοποιεί την εντολή Quit Game, η οποία τερματίζει το παιχνίδι.

(Η προβολή των widgets στο viewport πραγματοποιείται όπως και στην αλληλεπίδραση με τα NPC.)

Αυτή η δομή εξασφαλίζει ότι κατά την αλληλεπίδραση του παίκτη με τα μενού, το σύστημα παραμένει ευέλικτο και δεν επηρεάζει την κύρια ροή του παιχνιδιού, δεδομένου ότι ο χαρακτήρας του παίκτη δεν κάνει spawn κατά την παραμονή στα μενού.

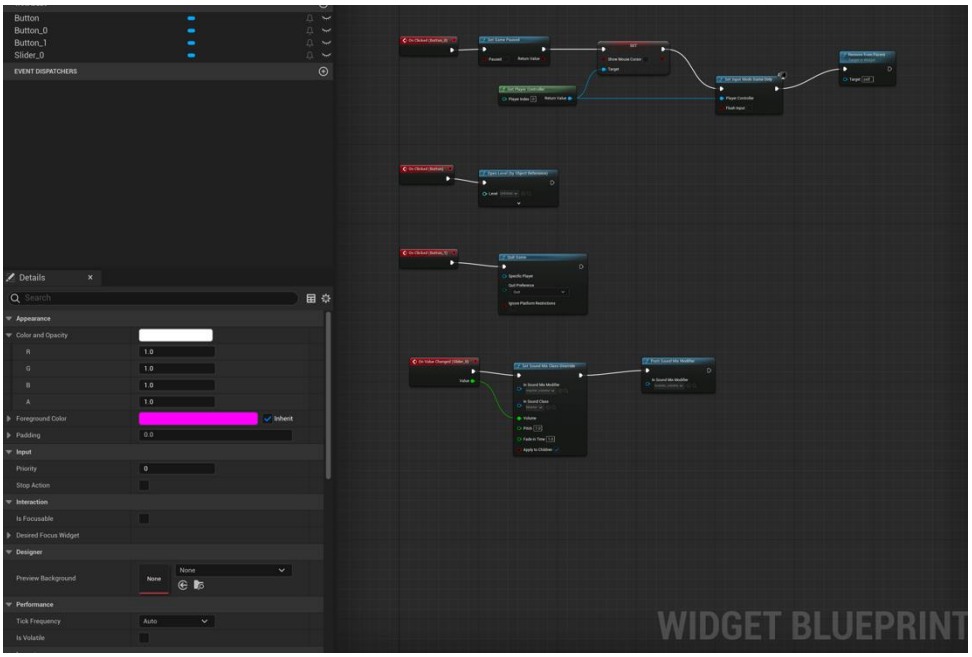
6.2 Μενού Παύσης

Το μενού παύσης εμφανίζεται όταν το παιχνίδι διακόπτεται προσωρινά από τον χρήστη, παρέχοντας δυνατότητες όπως συνέχιση του παιχνιδιού, επιστροφή στο κύριο μενού ή πρόσβαση σε ρυθμίσεις. Αυτό το μενού επιτρέπει στον παίκτη να κάνει ένα διάλειμμα χωρίς να χάσει την πρόοδο του, ενώ του δίνει τη δυνατότητα να προσαρμόσει διάφορες επιλογές αν χρειάζεται.



Εικόνα 51: Pause Menu Widget

Η διάταξη του μενού είναι οργανωμένη μέσω ενός Canvas Panel και ενός Vertical Box που φιλοξενεί τέσσερα κουμπιά, τα οποία εξυπηρετούν διαφορετικές λειτουργίες. Το κουμπί Resume επιτρέπει στον παίκτη να συνεχίσει το παιχνίδι από το σημείο όπου πάτησε παύση, ενώ το κουμπί Sound συνοδεύεται από ένα slider, το οποίο ρυθμίζει την ένταση του ήχου εντός του παιχνιδιού. Το κουμπί Return to Main Menu δίνει τη δυνατότητα επιστροφής στο αρχικό μενού, χωρίς να χρειαστεί να κλείσει το παιχνίδι. Τέλος, το κουμπί Quit επιτρέπει στον παίκτη να κλείσει πλήρως το παιχνίδι.

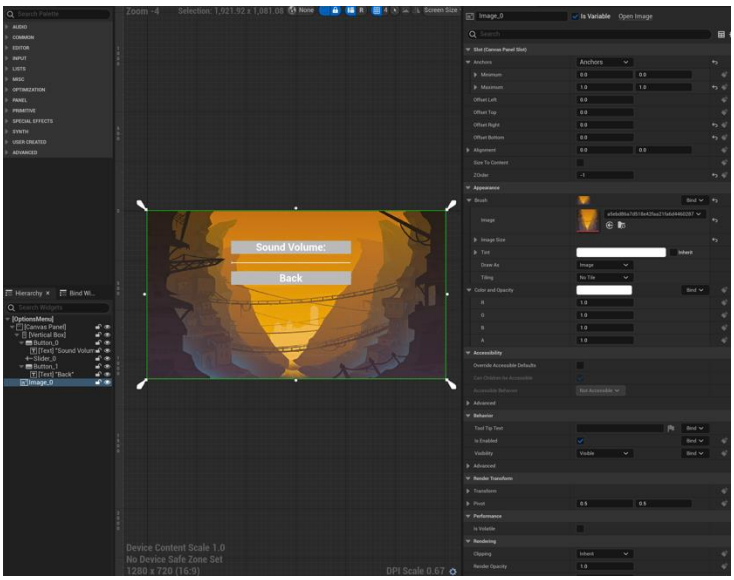


Εικόνα 52: Pause Menu Event Graph

Η λειτουργικότητα του μενού παύσης είναι σχεδιασμένη με τρόπο που δεν επηρεάζει την κύρια ροή του παιχνιδιού, επιτρέποντας στον παίκτη να διακόψει προσωρινά και να κάνει τις απαραίτητες ρυθμίσεις ή ενέργειες χωρίς να χάνει την πρόοδο που έχει κάνει.

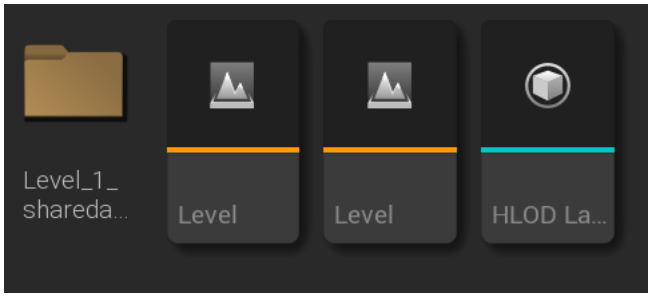
6.3 Μενού Επιλογών

Το μενού επιλογών επιτρέπει στον χρήστη να τροποποιήσει την ένταση της μουσικής. Είναι προσβάσιμο από το αρχικό μενού.



7. ΕΠΊΠΕΔΑ ΚΑΙ ΚΌΣΜΟΣ

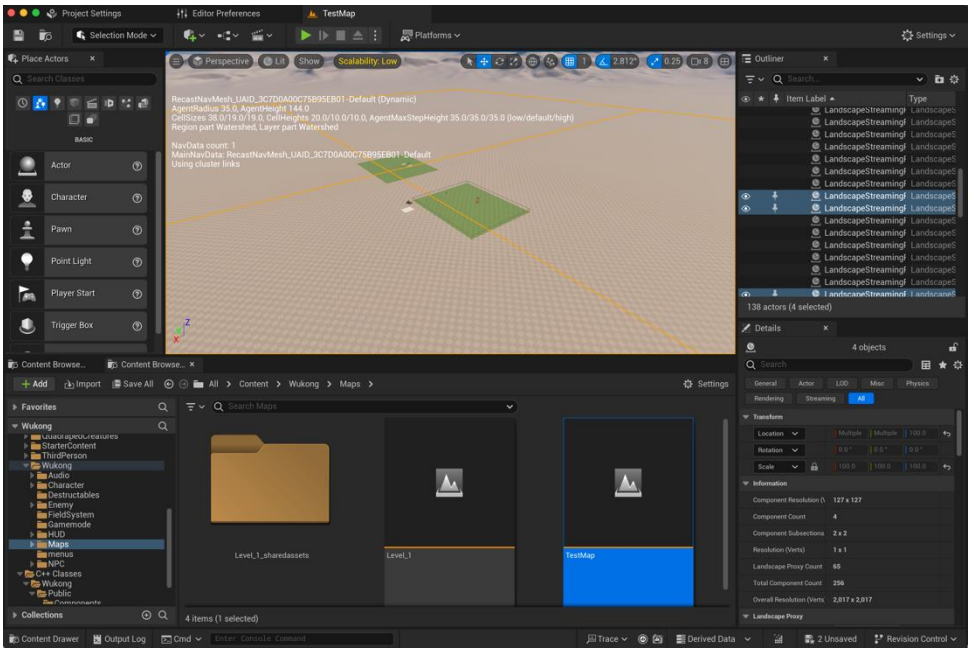
Καθ' όλη τη διάρκεια της υλοποίησης της περιγραφόμενης ιδέας, χρησιμοποιήθηκαν τρία διαφορετικά επίπεδα (levels). Το πρώτο επίπεδο είναι το κενό επίπεδο των μενού, το οποίο περιλαμβάνει αποκλειστικά τα διάφορα μενού του παιχνιδιού, όπως το αρχικό μενού, το μενού παύσης και το μενού επιλογών.



Εικόνα 53: Level Files

7.2 Test Map

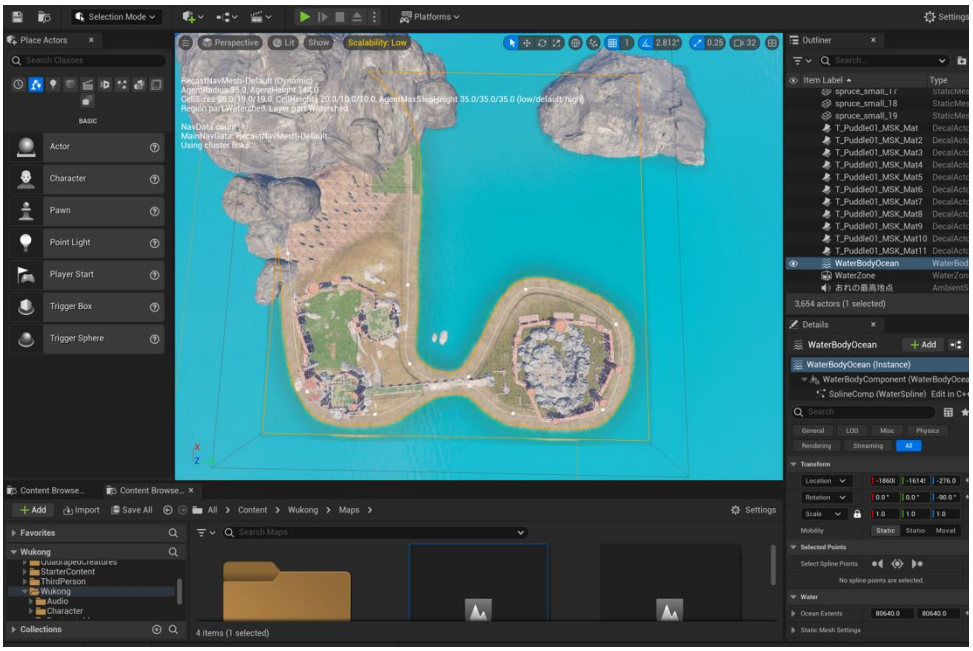
Στη συνέχεια, το test menu level χρησιμοποιήθηκε ως το επίπεδο όπου πραγματοποιήθηκε ολόκληρη η δοκιμαστική διαδικασία κατά τη διάρκεια του development stage. Σε αυτό το επίπεδο έγινε ο έλεγχος και η αξιολόγηση των διαφόρων λειτουργιών του παιχνιδιού.



Εικόνα 54: Test map

7.3 Level1 Map

Τέλος, το level1 αντιπροσωπεύει τον τελικό ανοιχτό κόσμο του παιχνιδιού, όπου οι παίκτες μπορούν να εξερευνήσουν και να αλληλεπιδράσουν με το περιβάλλον, αποτελώντας το κεντρικό επίπεδο της ολοκληρωμένης εμπειρίας παιχνιδιού.



Εικόνα 55: Level1 map

Το Level 1 map αποτελείται από ένα τοπίο διαστάσεων 8 επί 8, το οποίο περιλαμβάνει δύο κάστρα, με μια αρένα τύπου wasteland να βρίσκεται στο ενδιάμεσο τους. Το πρώτο από τα κάστρα συνδέεται μέσω μιας γέφυρας με ένα επιπλέον τείχος, το οποίο περιβάλλει ένα χωριό χτισμένο σε ένα βραχώδες περιβάλλον. Στην άκρη αυτού του χωριού, εκτός από το τείχος, υπάρχει και ένα μικρό δάσος, ενώ στο εσωτερικό του κατοικούν NPCs και άλλοι background characters που απεικονίζουν τους χωρικούς. Για να εισέλθει κάποιος στο χωριό, υπάρχει μια ειδική περιοχή με σκάλες, που επιτρέπει την πρόσβαση, αλλά και την έξοδο αν το επιθυμεί. Έξω από το τείχος του χωριού βρίσκονται μερικά επιπλέον σπίτια.

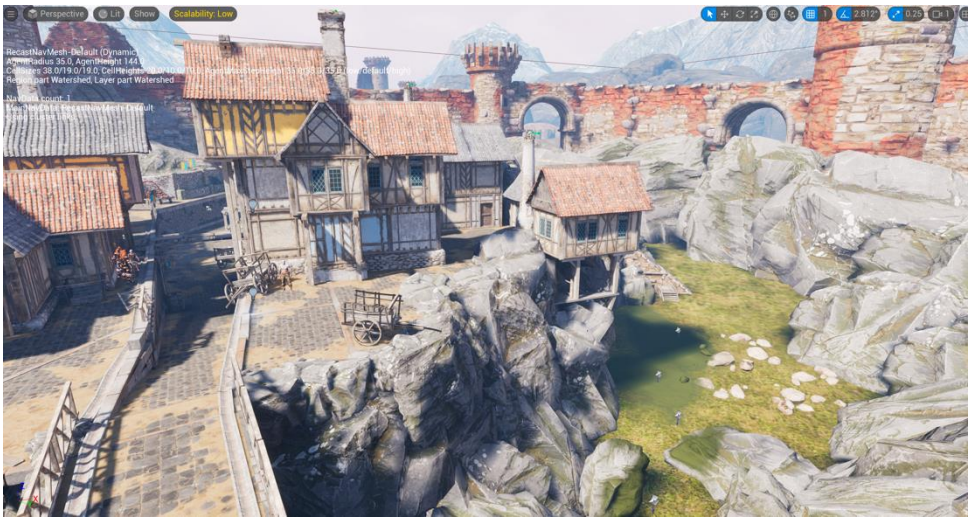
Το δεύτερο κάστρο περιβάλλεται από μια σειρά από ογκώδεις βράχους, οι οποίοι σχηματίζουν μια σπηλιά. Στη σπηλιά αυτή βρίσκονται οι δράκοι, ενώ οι εχθροί, οι μάγισσες, εμφανίζονται τόσο στα δύο κάστρα όσο και στο δεύτερο μισό της γέφυρας, καθώς και στην αρένα, σε μεγαλύτερη συγκέντρωση.

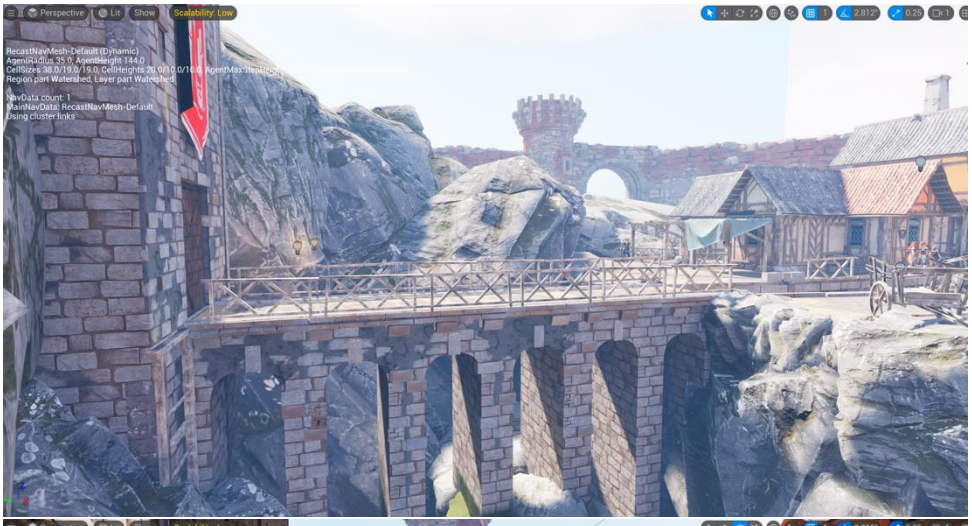
Ολόκληρο το τοπίο περιβάλλεται από έναν ωκεανό και κάποια βουνά, τα οποία προσθέτουν ένα επιπλέον επίπεδο ρεαλισμού στο περιβάλλον του παιχνιδιού, δίνοντας μια αίσθηση ανοιχτού κόσμου που περιβάλλεται από φυσικά εμπόδια και προσφέρει μια πιο ρεαλιστική εμπειρία στον παίκτη.

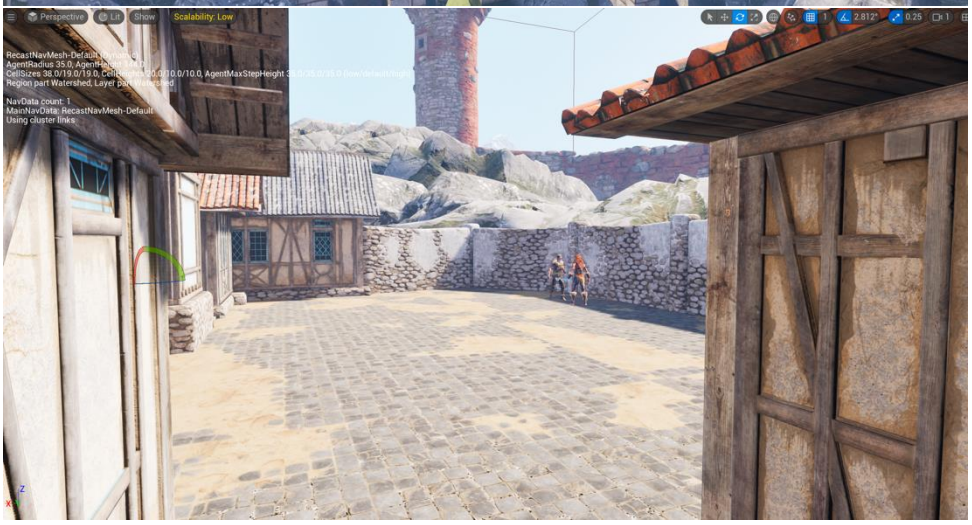
Ακολουθεί μία σειρά από screenshots του κόσμου:

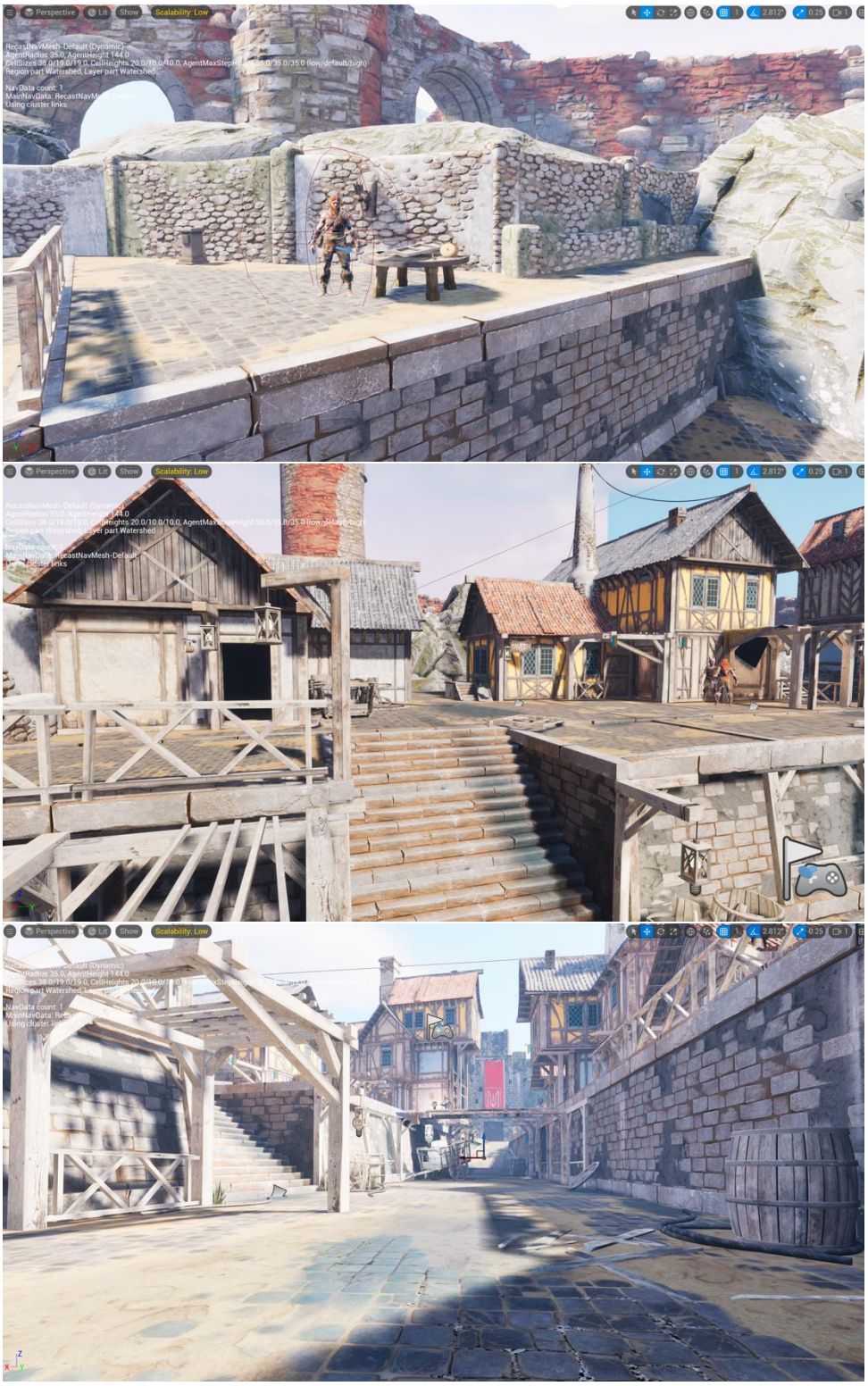
(actual gameplay)

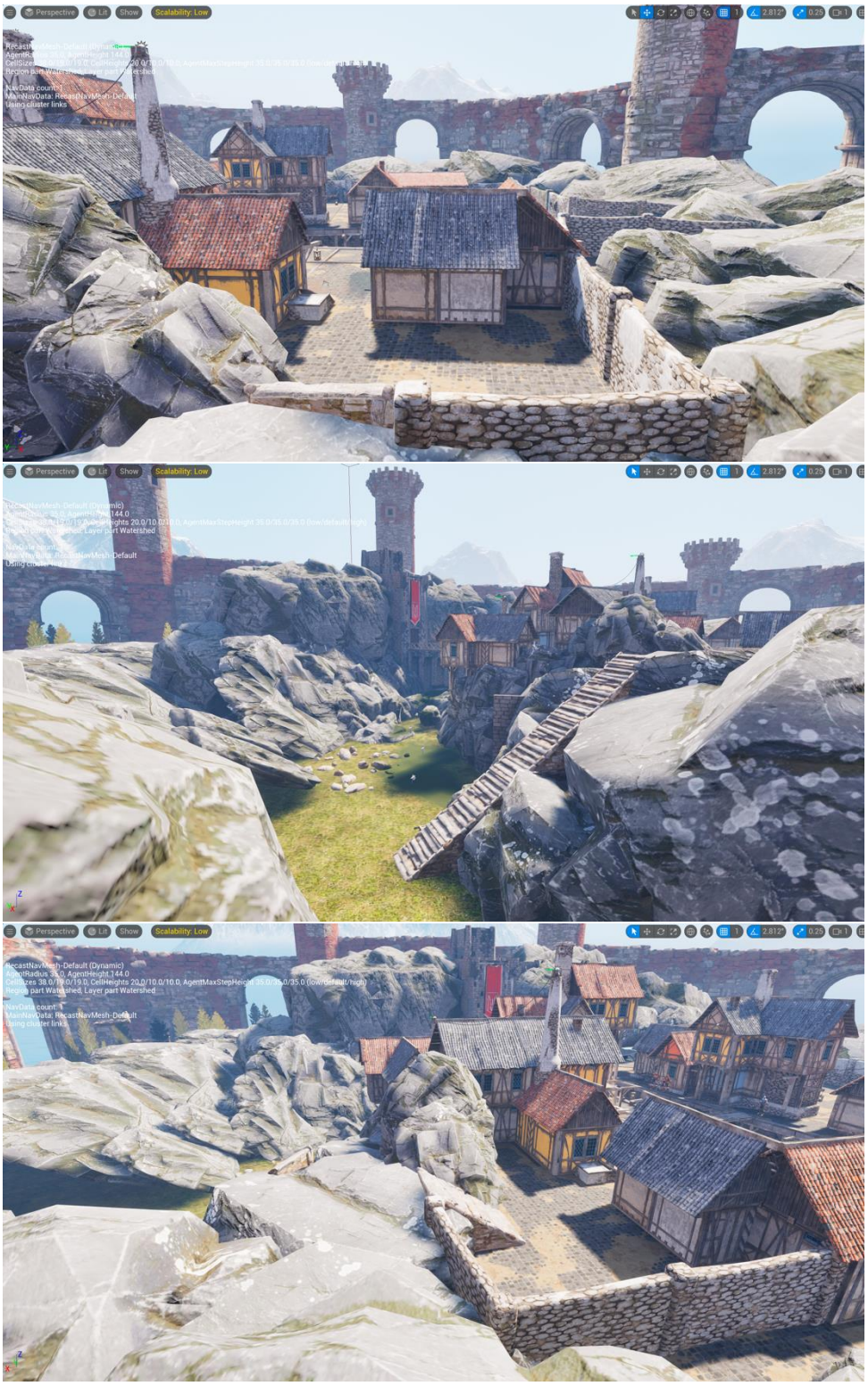
7.3.1 Χωριό με NPC

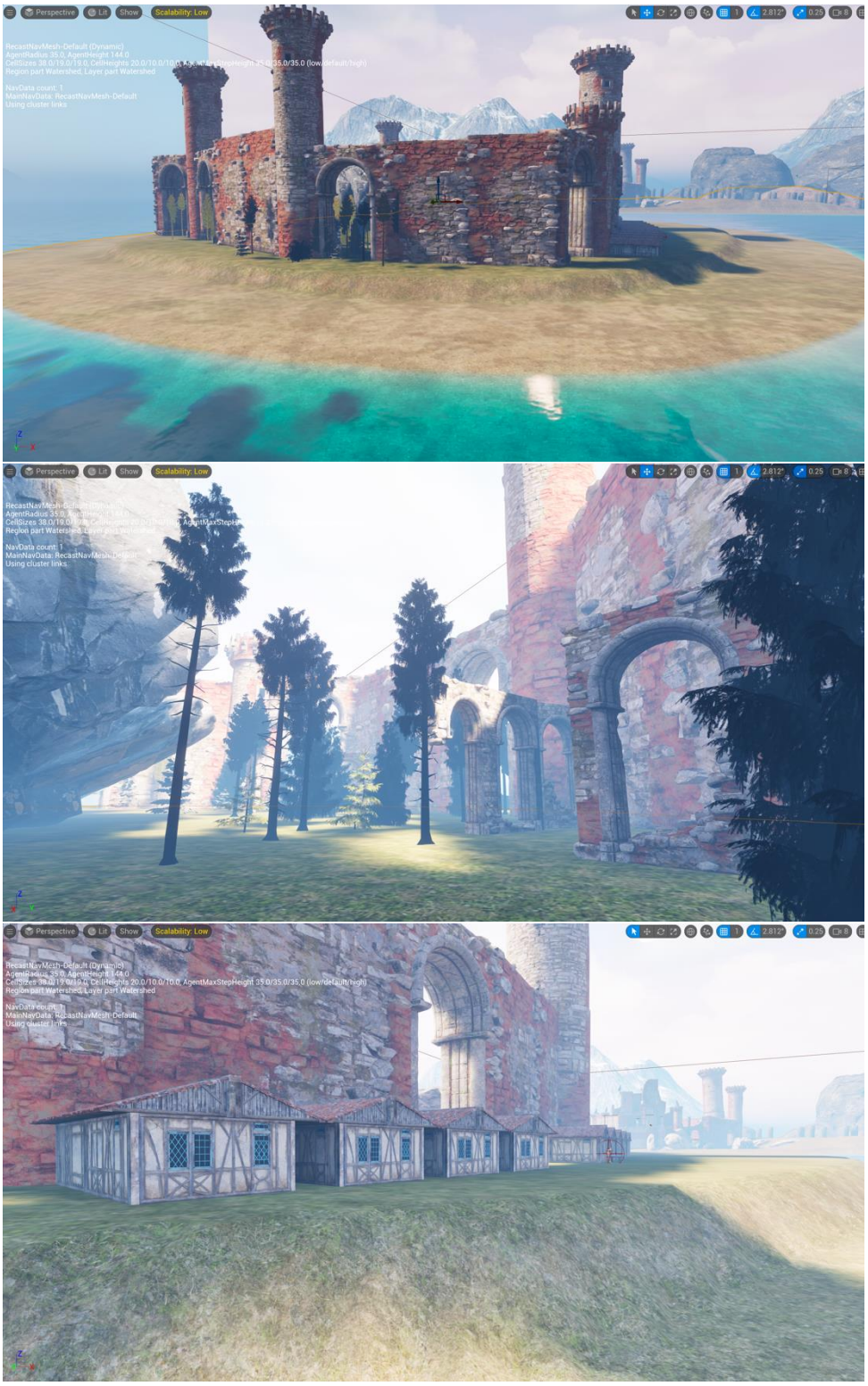




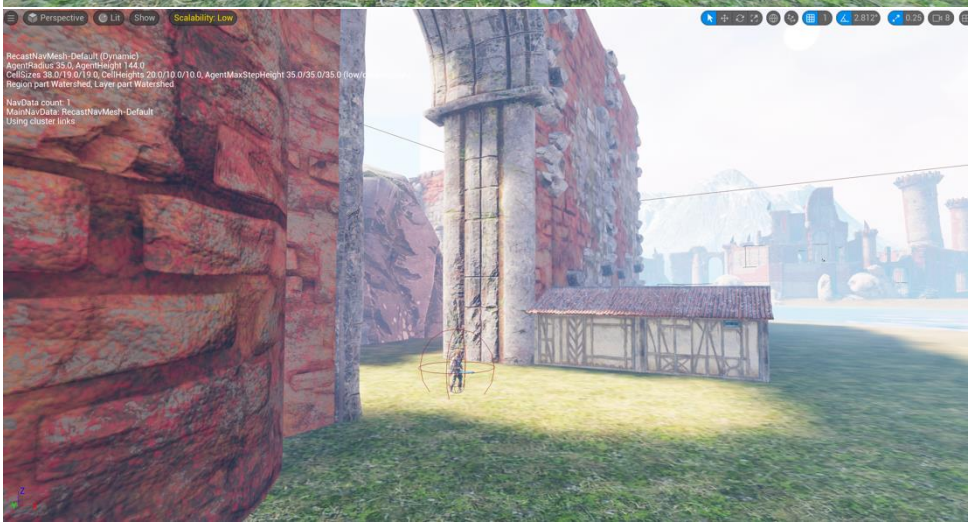




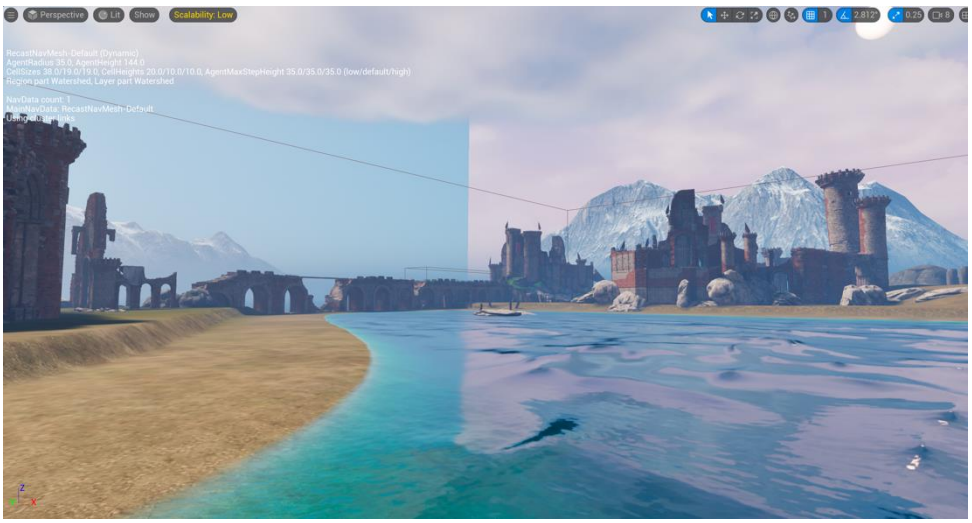


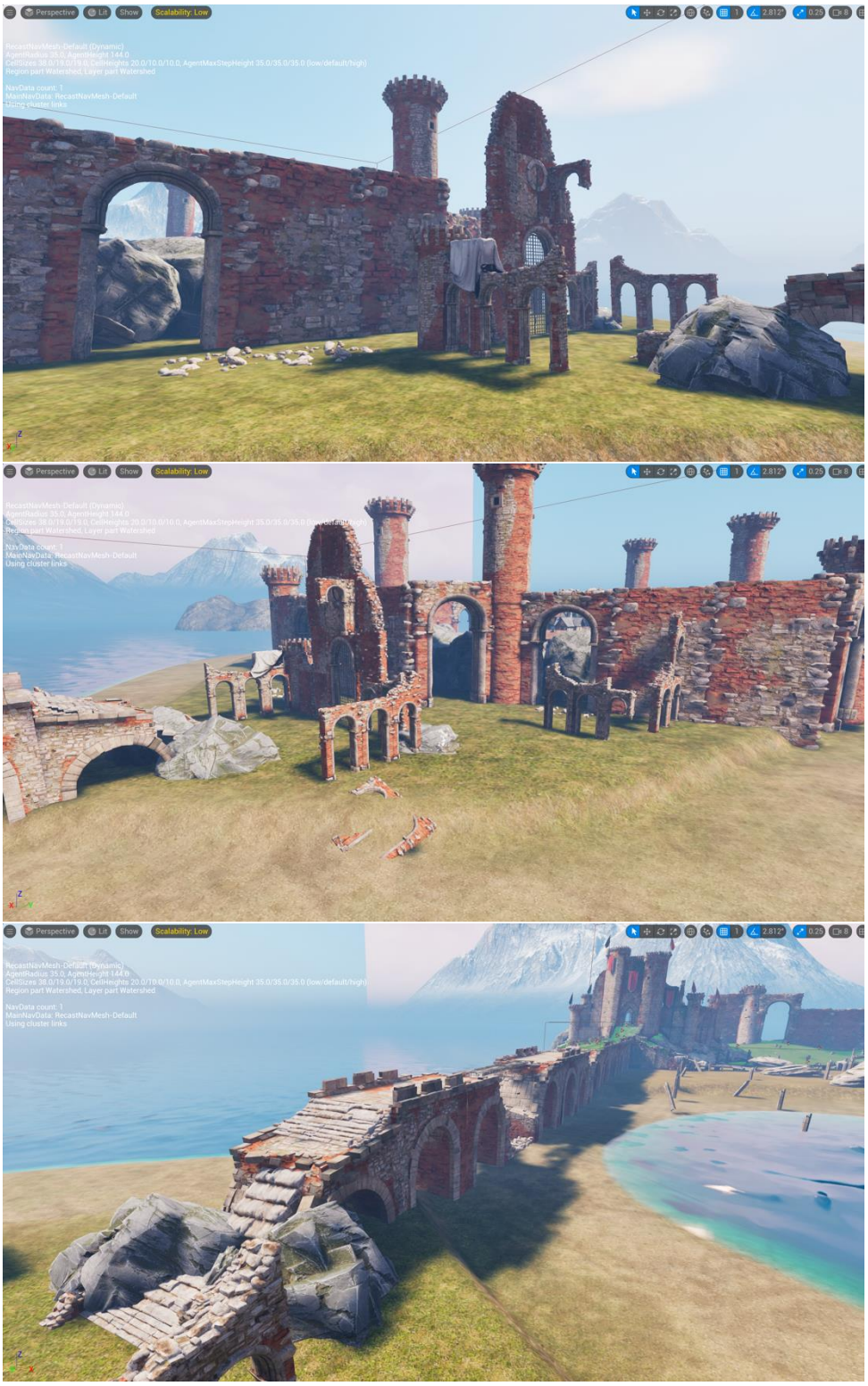



Perspective
Lit
Show
Scalability: Low
RecastNavMesh-Default (Dynamic)
AgentRadius 35.0, AgentHeight 144.0
CellSizes 38.0/19.0/19.0, CellHeights 20.0/10.0/10.0, AgentMaxStepHeight 35.0/35.0/35.0 (low/default/high)
Region part Watershed, Layer part Watershed
NavData count: 1
MainNavData: RecastNavMesh-Default
Using cluster links

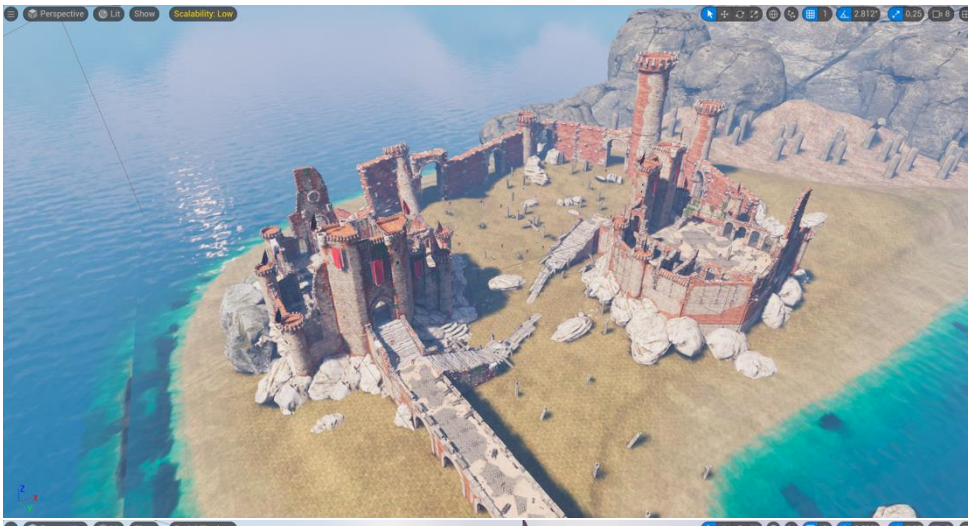

Perspective
Lit
Show
Scalability: Low
RecastNavMesh-Default (Dynamic)
AgentRadius 35.0, AgentHeight 144.0
Region part Watershed, Layer part Watershed
NavData count: 1
MainNavData: RecastNavMesh-Default
Using cluster links

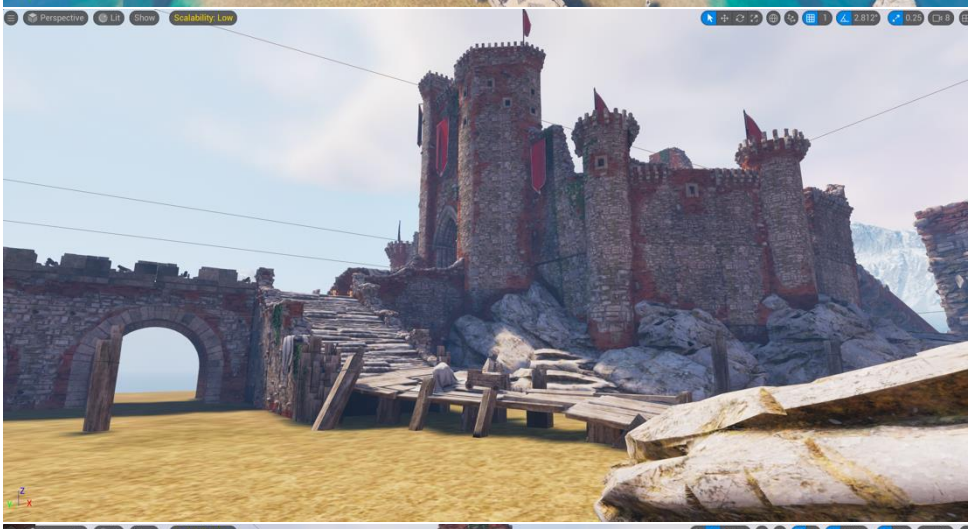
7.3.2 Γέφυρα

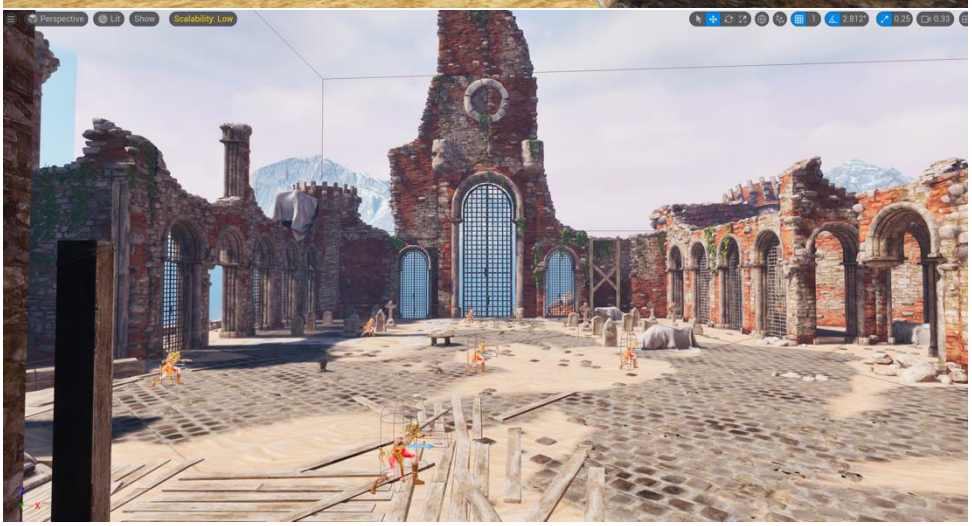

Perspective
Lit
Show
Scalability: Low
RecastNavMesh-Default (Dynamic)
AgentRadius 35.0, AgentHeight 144.0
CellSizes 38.0/19.0/19.0, CellHeights 20.0/10.0/10.0, AgentMaxStepHeight 35.0/35.0/35.0 (low/default/high)
Region part Watershed, Layer part Watershed
NavData count: 1
MainNavData: RecastNavMesh-Default

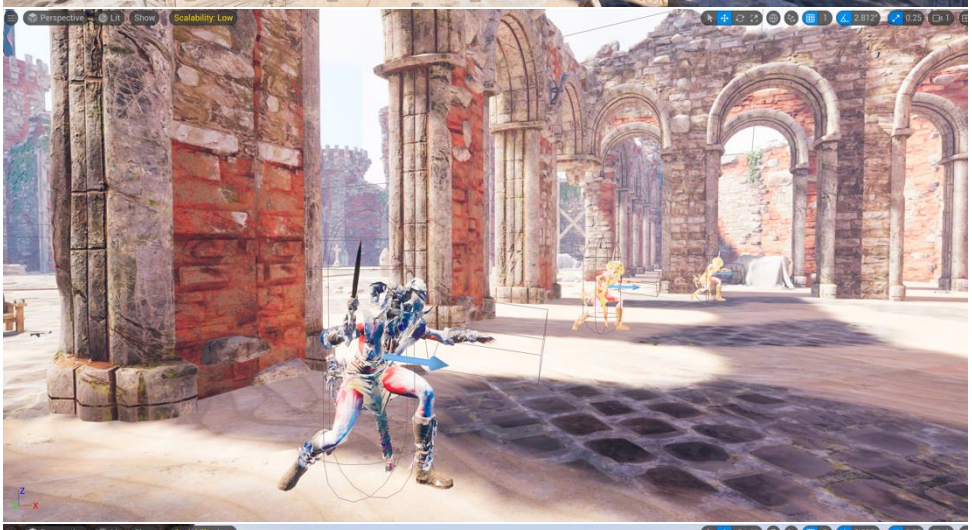


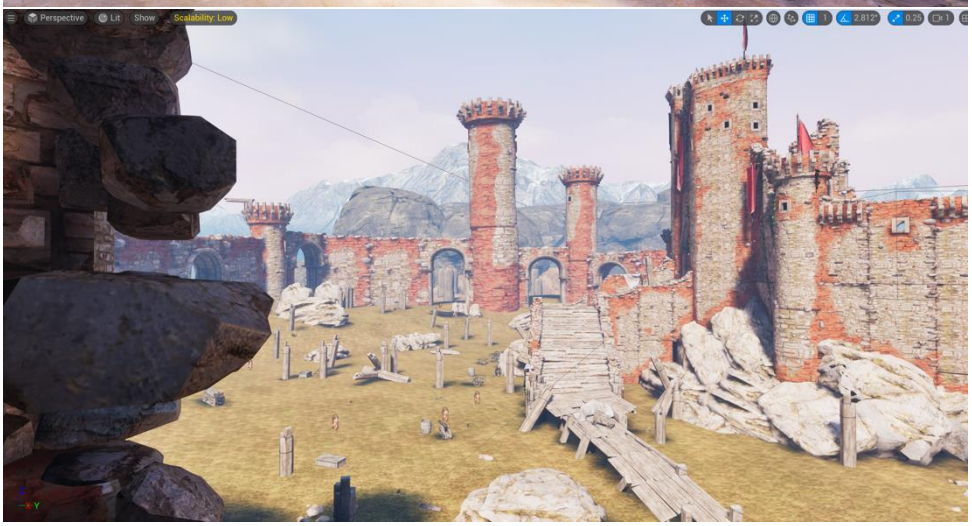
7.3.2 Εχθρικό Περιβάλλον

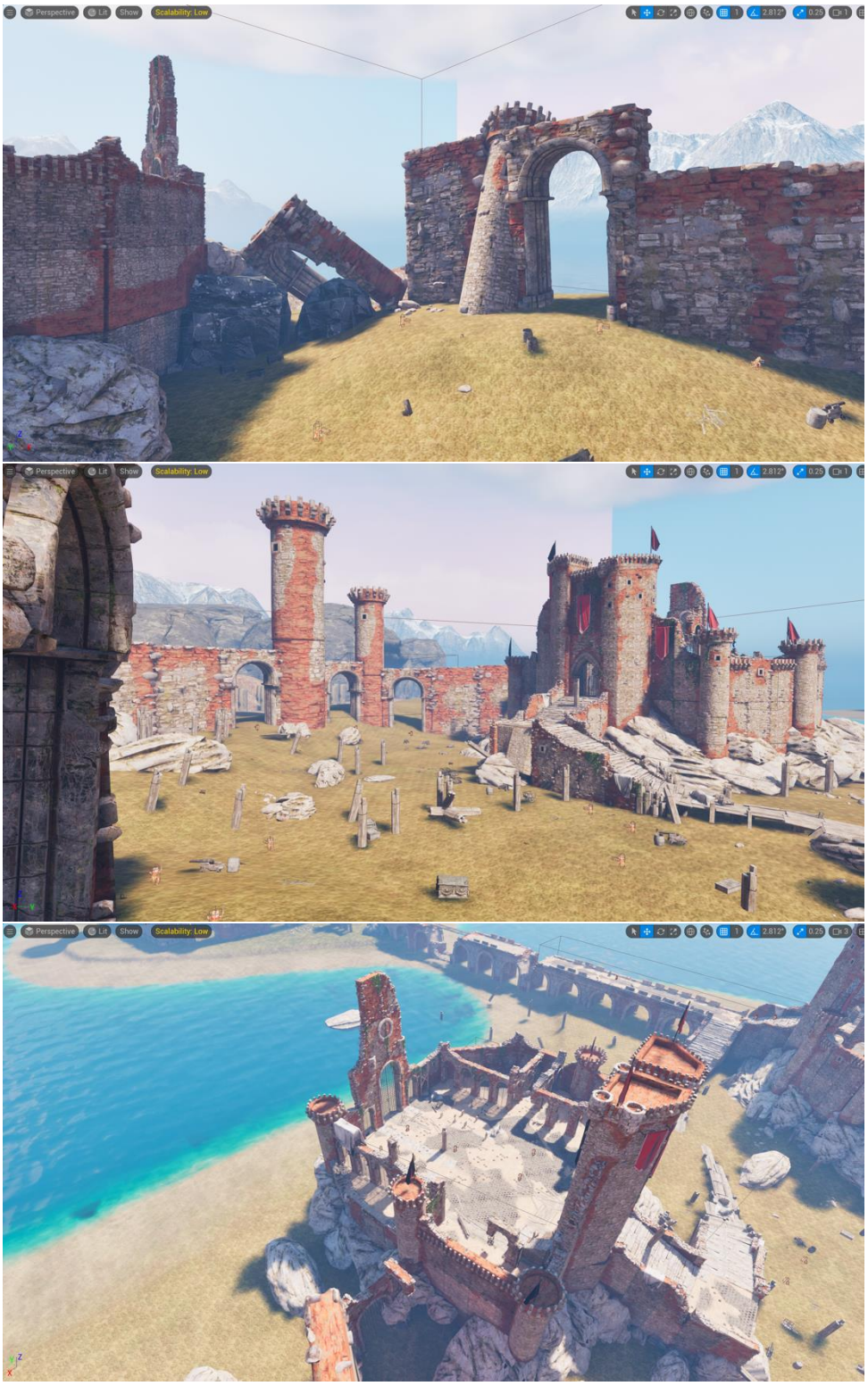


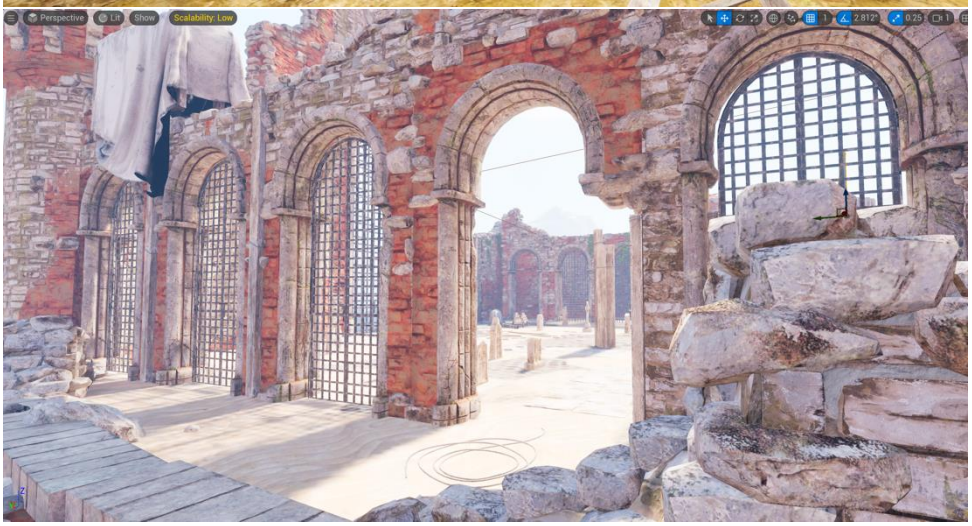




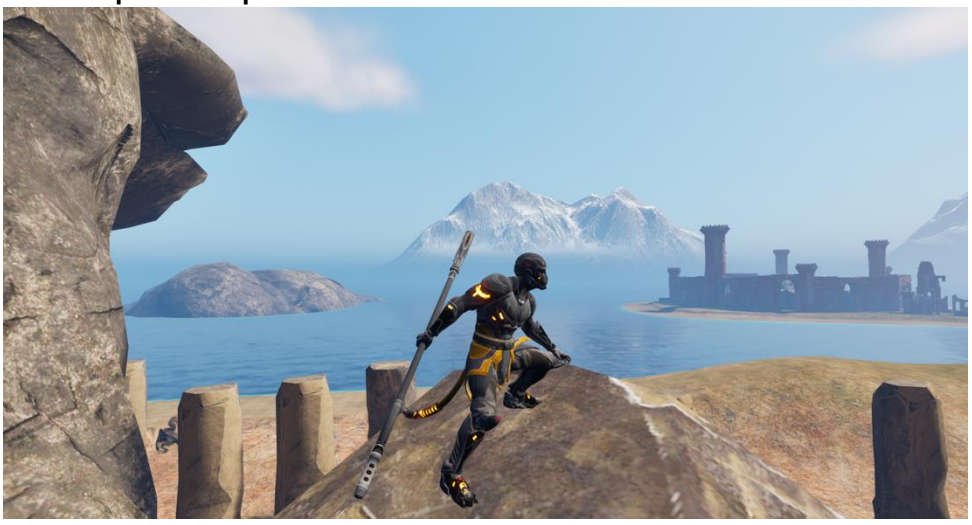


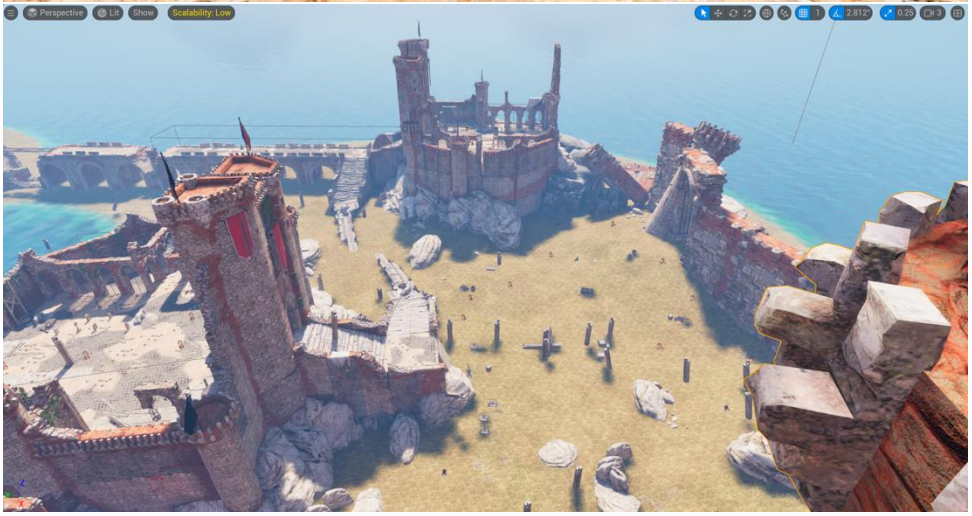




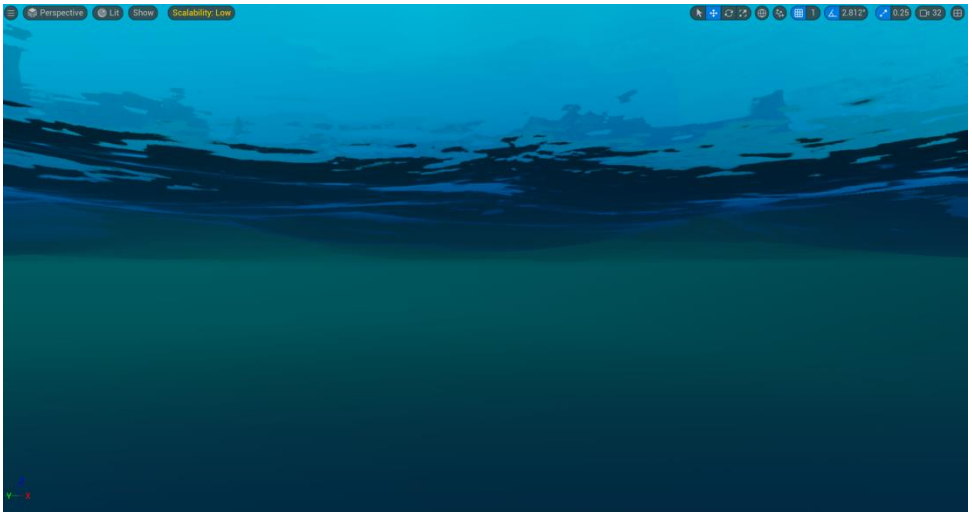


7.3.3 Σπηλιά των Δράκων





7.3.4 Water Body Zone



Μελλοντικές περεταίρω επεκτάσεις

Μελλοντικά, η ανάπτυξη του project μπορεί να επεκταθεί σε μερικούς τομείς, αξιοποιώντας τις δυνατότητες της C++ για πιο περίπλοκες και εξειδικευμένες εφαρμογές. Ένας τομέας πιθανής επέκτασης είναι η ανάπτυξη παραπάνω προγραμμάτων που σχετίζονται με την τεχνητή νοημοσύνη (AI) και τη μηχανική μάθηση. Με τη βοήθεια της C++, μπορούν να κατασκευαστούν αλγόριθμοι μηχανικής μάθησης που βελτιστοποιούν τη λήψη αποφάσεων σε πραγματικό χρόνο, κάτι που θα μπορούσε να εφαρμοστεί σε βιομηχανίες όπως η ρομποτική, η ανάλυση δεδομένων και η αυτοματοποιημένη επεξεργασία πληροφοριών.

Ένας άλλος τομέας μελλοντικής ανάπτυξης είναι η υλοποίηση συστημάτων προσομοίωσης υψηλής απόδοσης. Οι εφαρμογές αυτές μπορούν να χρησιμοποιηθούν στην επιστημονική έρευνα και στην εκπαίδευση, προσφέροντας ακριβείς προσομοιώσεις φαινομένων φυσικής, χημείας ή ακόμα και βιολογικών διεργασιών. Η C++ με την ισχυρή διαχείριση μνήμης και τις δυνατότητες πολυπλοκότητας επιτρέπει τη δημιουργία αυτών των προγραμμάτων με μεγάλη ακρίβεια και απόδοση.

Επιπλέον, μια άλλη επέκταση μπορεί να αφορά την ανάπτυξη εργαλείων για την ασφάλεια στον κυβερνοχώρο. Με την αυξανόμενη απειλή κυβερνοεπιθέσεων, υπάρχει συνεχώς αυξανόμενη ανάγκη για λογισμικά που μπορούν να εντοπίσουν και να εξουδετερώσουν κακόβουλες επιθέσεις σε πραγματικό χρόνο. Η C++ προσφέρει τη δυνατότητα δημιουργίας γρήγορων και αποτελεσματικών προγραμμάτων που εστιάζουν στην ανίχνευση ευπαθειών, ενισχύοντας την προστασία των δικτύων και των συστημάτων.

Τέλος, μια ενδιαφέρουσα προοπτική θα ήταν η ενσωμάτωση της γλώσσας C++ σε εφαρμογές εικονικής και επαυξημένης πραγματικότητας (VR/AR). Η αυξανόμενη χρήση αυτών των τεχνολογιών σε τομείς όπως τα παιχνίδια, η εκπαίδευση και η ιατρική προσφέρει ευκαιρίες για την ανάπτυξη εξειδικευμένων εργαλείων και πλατφορμών που θα βελτιώσουν την απόδοση και την ποιότητα της εμπειρίας του χρήστη. Η C++ μπορεί να χρησιμοποιηθεί για την ανάπτυξη αποδοτικών και γρήγορων μηχανών γραφικών, παρέχοντας βελτιστοποιημένες εφαρμογές για τις πλατφόρμες αυτές.

Συμπεράσματα

Η ανάπτυξη του project αυτού αναδεικνύει τη σημασία της C++ ως ένα ευέλικτο και ισχυρό εργαλείο για την υλοποίηση σύνθετων και αποδοτικών προγραμμάτων. Μέσα από τις εφαρμογές που παρουσιάστηκαν, καταδείχθηκε ότι η γλώσσα αυτή προσφέρει εξαιρετική απόδοση και δυνατότητες, καλύπτοντας ένα ευρύ φάσμα αναγκών στον τομέα του λογισμικού. Η ανάλυση και η επίλυση προβλημάτων μέσω των εφαρμογών αυτών ανέδειξαν τον ρόλο της C++ στην αλγοριθμική και συστηματική προσέγγιση της ανάπτυξης λογισμικού.

Η δυνατότητα διαχείρισης της μνήμης και των πόρων του συστήματος, σε συνδυασμό με την υποστήριξη αντικειμενοστραφούς προγραμματισμού, καθιστούν τη C++ μια από τις πιο αποτελεσματικές γλώσσες για την ανάπτυξη βελτιστοποιημένων προγραμμάτων. Η αξιοποίηση αυτών των χαρακτηριστικών στις εφαρμογές που αναπτύχθηκαν προσέφερε σημαντική απόδοση, δείχνοντας τις δυνατότητες που μπορεί να προσφέρει η C++ σε πραγματικές συνθήκες.

Πέρα από τις παρούσες εφαρμογές, η C++ διατηρεί τη θέση της ως μία από τις πλέον αξιόπιστες γλώσσες για την ανάπτυξη λογισμικού, με δυνατότητες που συνεχίζουν να εξελίσσονται και να προσαρμόζονται στις ανάγκες της σύγχρονης τεχνολογίας. Οι μελλοντικές επεκτάσεις που προτείνονται αποδεικνύουν την ευελιξία της γλώσσας και τη δυνατότητά της να προσαρμόζεται σε νέους τομείς, όπως η τεχνητή νοημοσύνη και η ασφάλεια στον κυβερνοχώρο.

Συνοψίζοντας, η συμβολή της C++ στην ανάπτυξη σύγχρονων εφαρμογών δεν περιορίζεται μόνο στις τεχνικές δυνατότητες που προσφέρει, αλλά και στη συνεχή εξέλιξή της για την επίλυση σύνθετων προβλημάτων. Με τη σωστή αξιοποίηση των εργαλείων και των δυνατοτήτων της γλώσσας, είναι εφικτό να παραχθούν προγράμματα υψηλής ποιότητας που να ανταποκρίνονται στις απαιτήσεις της σύγχρονης βιομηχανίας και τεχνολογίας.

Βιβλιογραφία

1. https://www.udemy.com/course/unreal-engine-5-the-ultimate-game-developer-course/learn/lecture/32521158#overview
   Unreal Engine 5 C++ The Ultimate Game Developer Course

2. https://www.youtube.com/@unrealengineindie
   UnrealEngineIndie | Youtube

3. https://unfgames.com/unreal-engine-5-ai-patrol-combat/
   Unreal Engine 5 tutorial for AI patrol and combat mechanics.

4. https://www.worldofleveldesign.com/categories/ue4/ue5-tutorial-melee-combat-system.php
   Guide to setting up a melee combat system in Unreal Engine.

5. **https://www.cgbookcase.com/tutorial/ue5-ai-combat-mechanics**
   Explains AI combat mechanics, including damage and health systems.

6. https://leveldesign.org/ai-sensing-attack-system-ue5
   A tutorial for AI sensing and attack systems in Unreal Engine 5.

7. https://www.game-creation.com/unreal-engine-melee-combat-tutorial
   Guide on implementing melee combat with animations and damage.

8. https://dev.epicgames.com/community/learning/tutorials/qyZr/how-to-make-a-combat-system-in-unreal-engine-5-2-tutorial
   Creating a combat system in Unreal Engine 5.

9. https://dev.epicgames.com/documentation/en-us/unreal-engine/ai-perception-in-unreal-engine
   Official Unreal Engine documentation for AI perception and sensing.

10. https://dev.epicgames.com/documentation/en-us/unreal-engine/implementing-health-and-damage-systems
    Guide to implementing health and damage systems in Unreal Engine 5.

11. https://dev.epicgames.com/documentation/en-us/unreal-engine/ai-patrol-guard-systems
    Documentation on setting up patrol and guard systems for AI in Unreal Engine 5

Πηγές

3D assets από το Unreal Engine Marketplace:

- Primitive Characters (Pack) from Bugrimov Maksim - Characters
- Paragon: Wukong from Epic Games - Epic Content
- Quixel Bridge
- Quadruped Fantasy Creatures from PROTOFACTOR INC - Characters
- Paragon: Morigesh from Epic Games - Epic Content
- temperate Vegetation: Spruce Forest from Project Nature - Props
- Ithris Cemetery from Rasmus Bagner - Environments
- Fantasy Bundle Environment Kit 3 in 1 from Denys Rutkovskyi - Environments