



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΡΙΣΔΙΑΣΤΑΤΟ ΠΑΙΧΝΙΔΙ ΠΡΩΤΟΥ ΠΡΟΣΩΠΟΥ ΜΕ ΧΡΗΣΗ Α.Ι.

ΜΠΑΛΗΣ ΒΑΣΙΛΕΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

.....Κολομβάτσος Κωνσταντίνος.....

.....ΒαθμίδαΑναπληρωτής καθηγητής.....

Λαμία έτος ...4ο...



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΡΙΣΔΙΑΣΤΑΤΟ ΠΑΙΧΝΙΔΙ ΠΡΩΤΟΥ ΠΡΟΣΩΠΟΥ ΜΕ ΧΡΗΣΗ Α.Ι.

ΜΠΑΛΗΣ ΒΑΣΙΛΕΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

..... Κολομβάτσος Κωνσταντίνος

.....Βαθμίδα... Αναπληρωτής καθηγητής

Λαμία έτος ...4ο...



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE &
TELECOMMUNICATIONS

3D FIRST PERSON GAME USING A.I.

MPALHS VASILEIOS

FINAL THESIS

ADVISOR

.....Kolomvatsos Konstantinos.....

.....Position..... Associate professor

Lamia year 4o...

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφική. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία:/...../20.....

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση

του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων

σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Στο πλαίσιο αυτής της πτυχιακής εργασίας θα μελετήσουμε τον τρόπο με τον οποίο κατασκευάζεται ένα ηλεκτρονικό παιχνίδι καθώς και το πώς το Α.Ι. συμβάλει στην βελτίωση των δυνατοτήτων των παιχνιδιών. Αρχικά γίνεται αναφορά στις πιο γνωστές μηχανές παραγωγής παιχνιδιών που υπάρχουν ενώ παράλληλα αναλύονται τα πλεονεκτήματα και τα μειονεκτήματα τις κάθε μίας. Στην συνέχεια περιγράφεται η συμμετοχή του Α.Ι. στα παιχνίδια, εξηγώντας τον τρόπο λειτουργίας του, τον τρόπο εξέλιξης του και παραθέτονται παραδείγματα μεγάλων τίτλων της βιομηχανίας των παιχνιδιών όπου χρησιμοποιήθηκε. Επιπλέον γίνεται ανάλυση του παιχνιδιού όπου έφτιαξα με βάση το σενάριο και τους χαρακτήρες που έχει αλλά και περιγραφή και επεξήγηση των blueprints(ο τρόπος κωδικοποίησης). Τέλος αναφέρονται τα συμπεράσματα και οι μελλοντικές προεκτάσεις για τα παιχνίδια μέσω της εξέλιξης του Α.Ι.

ABSTRACT

In the context of this senior thesis, we will study how a game is made as well as how A.I. helps to improve the game's capabilities. First reference is made to the most well-known game production machines that exist while at the same time the advantages and disadvantages of each one are analyzed. After that describes the participation of A.I. in games, explaining how it works, how it evolved and examples of major games industry titles where it was used. In addition, there is an analysis of the game I made based on the script and the characters it has, as well as a description and explanation of the blueprints (the way of coding). Finally, the conclusions and future implications for games through the evolution of A.I.

Table of Contents

ΠΕΡΙΛΗΨΗ	5
<u>ABSTRACT</u>	<u>6</u>
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ.....	8
ΚΕΦΑΛΑΙΟ 2 ΜΗΧΑΝΕΣ ΠΑΡΑΓΩΓΗΣ ΠΑΙΧΝΙΔΙΩΝ.....	10
(Υποκεφάλαιο 2.1) Εισαγωγή.....	10
(Υποκεφάλαιο 2.2) Πλεονεκτήματα και μειονεκτήματα της Unity...11	
(Υποκεφάλαιο 2.3) Πλεονεκτήματα και μειονεκτήματα της Godot..15	
(Υποκεφάλαιο 2.4) Πλεονεκτήματα και μειονεκτήματα της Game maker....18	
(Υποκεφάλαιο 2.5) Πλεονεκτήματα και μειονεκτήματα της Unreal engine...22	
ΚΕΦΑΛΑΙΟ 3 Α.Ι. ΣΤΑ ΠΑΙΧΝΙΔΙΑ.....	26
(Υποκεφάλαιο 3.1) Εισαγωγή.....	26
(Υποκεφάλαιο 3.2) Τρόποι λειτουργίας του Α.Ι. στα παιχνίδια...28	
(Υποκεφάλαιο 3.3) Η εξέλιξη του Α.Ι. στην βιομηχανία των παιχνιδιών..29	
(Υποκεφάλαιο 3.4) Μεγάλοι τίτλοι παιχνιδιών με χρήση Α.Ι.....30	
ΚΕΦΑΛΑΙΟ 4 SUICIDE MISSION Z.....	32
(Υποκεφάλαιο 4.1) Σενάριο και ανάλυση χαρακτήρων	32
(Υποκεφάλαιο 4.2) Επεξήγηση κώδικα.....	34
ΚΕΦΑΛΑΙΟ 5 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	73

ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ

Τα ηλεκτρονικά παιχνίδια υπάρχουν στην καθημερινότητα μας και όλο και περισσότεροι άνθρωποι επιλέγουν την ενασχόληση τους με αυτά είτε για ψυχαγωγία, είτε για επαγγελματική ενασχόληση. Τα παιχνίδια έκαναν την εμφάνιση τους στις αρχές της δεκαετίας του 60 και πιο συγκεκριμένα το 1961 με το παιχνίδι Spacewar που ήταν ένα δημιούργημα των φοιτητών του MIT. Αρχικά τα παιχνίδια χωριζόντουσαν σε δυο κατηγορίες: τα single player (δηλαδή παιχνίδια που παίζει μόνο ένας παίχτης) και τα local multiplayer (δηλαδή τα παιχνίδια που μπορούσαν να παίξουν δυο ή περισσότερα άτομα μαζί αλλά να βρίσκονται στον ίδιο χώρο και πιο συγκεκριμένα στην ίδια κονσόλα). Με την πάροδο του χρόνου και την ανάπτυξη της τεχνολογίας έκανε την εμφάνιση της και μια τρίτη κατηγορία τα παιχνίδια online multiplayer (όπου ο παίχτης μπορούσε να παίξει με παραπάνω από ένα άτομα και χωρίς να βρίσκεται στον ίδιο χώρο με εκείνα αλλά ο κάθε ένας να παίζει από την δική του κονσόλα).

Η εξέλιξη των παιχνιδιών δεν έμεινε μόνο στην ανάπτυξη τρόπων για να μπορούν περισσότεροι παίκτες να παίξουν μαζί. Οι εταιρίες παραγωγής παιχνιδιών προσπάθησαν να βρουν τρόπους να κάνουν τα single player παιχνίδια να εξελιχθούν ώστε να μην είναι κουραστικά και βαρετά για τους παίκτες αφού μετά από τόσα χρόνια τα είχαν συνηθίσει. Όσο και να βοήθησε η βελτίωση των γραφικών και η ποιότητα ήχου δεν ήταν αρκετό. Χρειάζοντουσαν και κάτι άλλο για να βελτιωθεί ο τρόπος παιχνιδιού για τον παίκτη. Εκεί χρειάστηκε η χρήση τεχνητής νοημοσύνης. Με την χρήση της τεχνητής νοημοσύνης (AI - Artificial Intelligence) θέλησαν να λύσουν αυτό το πρόβλημα χρησιμοποιώντας την στους αντιπάλους με σκοπό να αυξήσουν τη δυσκολία για τον παίκτη αλλά και να προσφέρουν την αίσθηση του απρόσμενου δηλαδή ο παίχτης να μην γνωρίζει ποια θα είναι η επόμενη κίνηση του αντιπάλου, όπως γινόταν πιο παλιά όπου οι αντίπαλοι ήταν προγραμματισμένοι να κάνουν συγκεκριμένες κινήσεις και να εκτελούν συγκεκριμένες ενέργειες. Με το AI κατάφεραν να κάνουν τους αντιπάλους να μπορούν να πάρουν μόνοι τους αποφάσεις, σε πλαίσια τα οποία οι προγραμματιστές όρισαν και να υπάρχει ένα αποτέλεσμα το οποίο στα μάτια του παίκτη να είναι ευχάριστο και να αυξάνει την δυσκολία του παιχνιδιού με αποτέλεσμα να τον κάνει να θέλει να ασχοληθεί με τα single player παιχνίδια.

Η ερώτηση όμως είναι «πώς φτιάχνουμε ένα παιχνίδι;». Η δημιουργία ενός παιχνιδιού γίνεται με την χρήση μηχανών δημιουργίας παιχνιδιών. Αυτές οι μηχανές είναι στην ουσία σύστημα λογισμικού για την παραγωγή και την ανάπτυξη βιντεοπαιχνιδιών. Υπάρχουν διάφορες μηχανές παιχνιδιών οι οποίες έχουν διαφορετικά χαρακτηριστικά ανάλογα το είδος του παιχνιδιού που θέλει ο προγραμματιστής να φτιάξει αλλά και την ταχύτητα παραγωγής του. Σκοπός αυτής της πτυχιακής εργασίας είναι η διαδικασία δημιουργίας ενός ηλεκτρονικού single player παιχνιδιού, καθώς και πως χρησιμοποιείται το ΑΙ για να εκτελέσει τις εντολές που θέλει ο εκάστοτε προγραμματιστής.

ΚΕΦΑΛΑΙΟ 2 ΜΗΧΑΝΕΣ ΠΑΡΑΓΩΓΗΣ ΠΑΙΧΝΙΔΙΩΝ

(Υποκεφάλαιο 2.1) Εισαγωγή

Μια μηχανή παιχνιδιού είναι ένα σύστημα λογισμικού το οποίο χρησιμοποιείται για την δημιουργία αλλά και την ανάπτυξη ηλεκτρονικών παιχνιδιών. Στα πρώτα χρόνια εμφάνισης των ηλεκτρονικών παιχνιδιών κάθε παιχνίδι έπρεπε να προγραμματιστεί από την αρχή. Ακόμα και οι δημιουργοί ενός παιχνιδιού δεν ήταν εύκολο να χρησιμοποιήσουν τον κώδικα που είχαν ήδη γραμμένο για το παιχνίδι τους σε ένα άλλο. Για αυτό τον λόγο άρχισαν να δημιουργούνται οι μηχανές παιχνιδιών έτσι ώστε να μπορούν οι προγραμματιστές να έχουν έτοιμες τις βασικές εντολές για την ανάπτυξη και την δημιουργία του εκάστοτε παιχνιδιού και από εκεί και πέρα να πρέπει να τις συνδέσουν με τον κατάλληλο τρόπο ώστε να έχουν το επιθυμητό αποτέλεσμα. Με αυτό τον τρόπο γλιτώνουν πολύ χρόνο τον οποίο θα έπρεπε να σπαταλήσουν στο να προγραμματίζουν τις απλές εντολές κάθε φορά από την αρχή. Σε κάθε μηχανή παιχνιδιών θα βρούμε διαφορά εργαλεία όπως :εργαλεία για animation, για επεξεργασία ήχου, τεχνητής νοημοσύνης (Α.Ι.) κ.α. Οι μηχανές παιχνιδιών μπορεί να έχουν φτιαχτεί για έναν κοινό σκοπό(την ανάπτυξη και την δημιουργία παιχνιδιών) αλλά αυτό δεν σημαίνει πως έχουν και τα ίδια χαρακτηριστικά. Η κάθε μηχανή διαφέρει σε διαφορά χαρακτηριστικά από τις άλλες και μπορεί να αλλάξει η επιλογή της μηχανής που θα χρησιμοποιήσουμε ανάλογα το είδος του παιχνιδιού που θέλουμε να φτιάξουμε. Ενδεικτικά κάποιες από τις πιο γνωστές μηχανές παιχνιδιών είναι: Unity, Godot, Game Maker, Unreal Engine. Η κάθε μια από τις παραπάνω μηχανές έχει κάποια θετικά και κάποια αρνητικά σε σχέση με τις άλλες ως προς την δομή και τις δυνατότητες της. Ας εξετάσουμε κάθε μηχανή ξεχωριστά.

(Υποκεφάλαιο 2.2) Πλεονεκτήματα και μειονεκτήματα της Unity



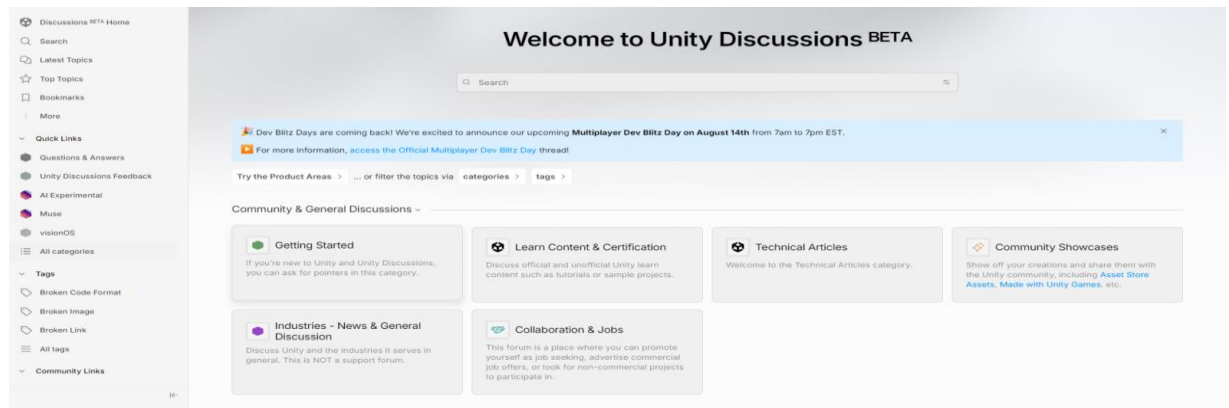
Εικόνα 1.1 Λογότυπο της Unity

Σύνδεσμος για την σελίδα : <https://unity.com/>

Η Unity είναι μία μηχανή δημιουργίας παιχνιδιών, η οποία απευθύνεται σε πολλαπλές πλατφόρμες και μέσω αυτού του χαρακτηριστικού της έγινε γνωστή διότι είχαν μεγάλη απήχηση τα παιχνίδια της για κινητά ios και android. Κυκλοφόρησε το 2005 και από τότε συνεχώς αναπτύσσεται.

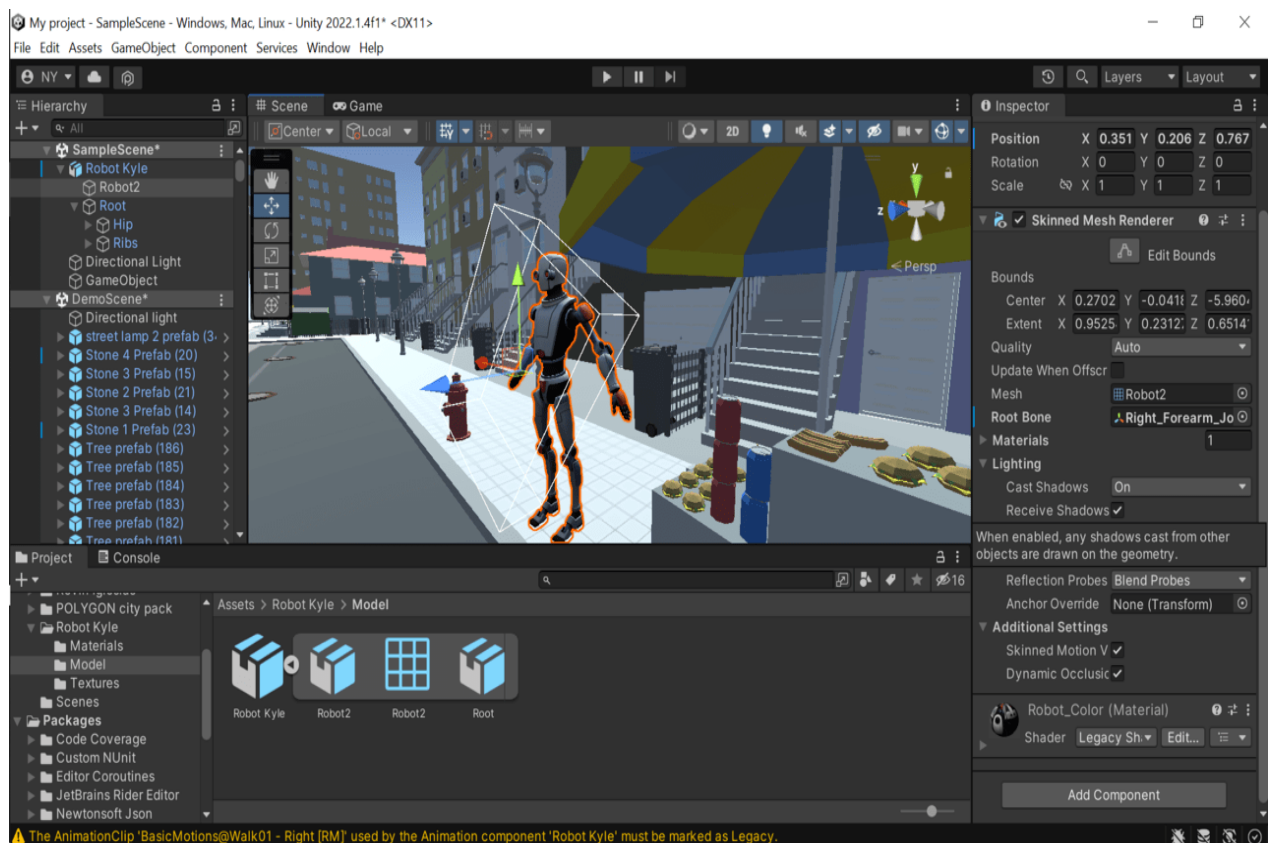
Πλεονεκτήματα

- Υπάρχει ενεργή ανάπτυξη και συνεχώς αποκτά περισσότερες δυνατότητες με κάθε νέα έκδοση.
- Έχει μια μεγάλη ποικιλία από πλατφόρμες που υποστηρίζονται από τον ιστό και τις κινητές συσκευές έως και τους υπολογιστές αλλά και τις κονσόλες.
- Η κοινότητα της Unity είναι μια από τις καλύτερες και αυτό φαίνεται από το feedback που παίρνει ο κάθε νέος προγραμματιστής από τους πιο έμπειρους σε απορίες που ενδεχομένως να έχει ως καινούργιος στον χώρο.

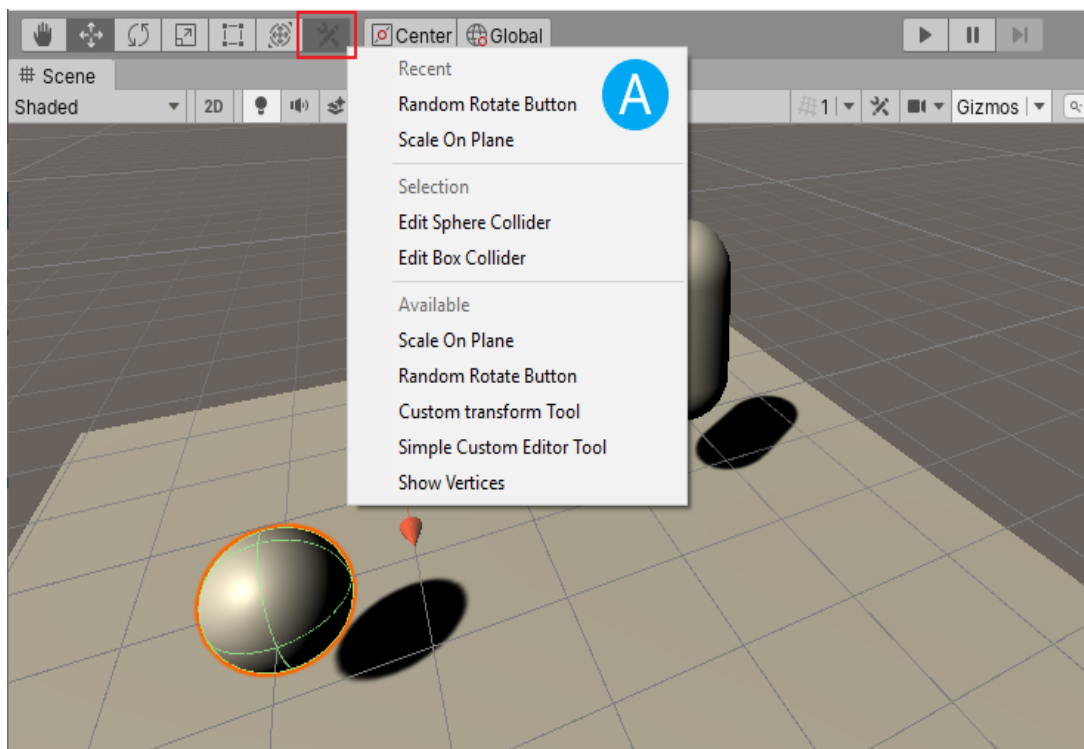
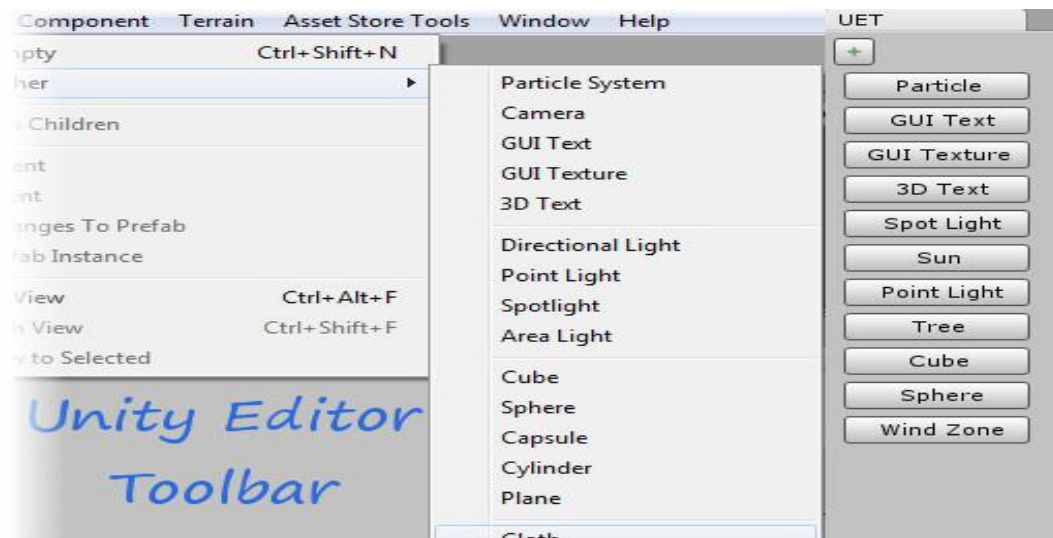


Εικόνα 1.2 Το official forum page της Unity

- Η Unity προσφέρει πολλές έτοιμες προς χρήση λύσεις αλλά και στοιχεία.



Εικόνα 1.3 Το workspace της Unity.



Εικόνα 1.4 Ενδεικτικό παράδειγμα έτοιμων εργαλείων από την Unity.

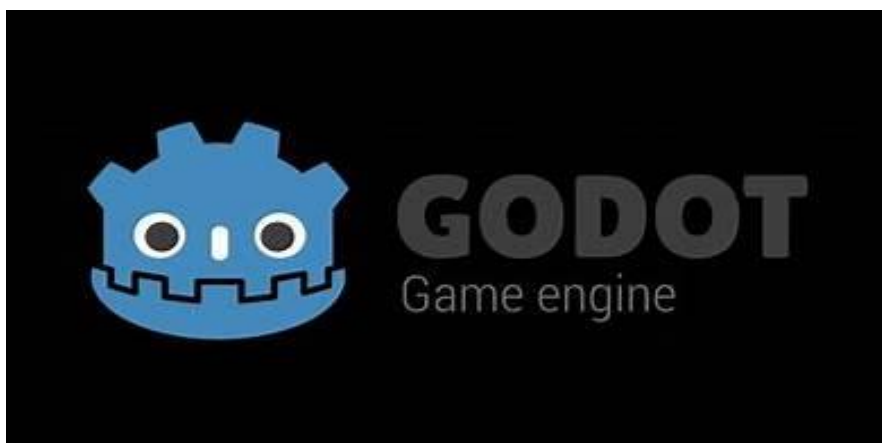
- Είναι εύκολο για έναν αρχάριο να την μάθει και να μπορεί να έχει ένα ικανοποιητικό αποτέλεσμα με αυτή. Προσφέρει απόλυτο έλεγχο στην δημιουργία, την εκτέλεση και

την βελτίωση του παιχνιδιού χωρίς να επιβάλει στον προγραμματιστή να κάνει αλλαγές μόνο με συγκεκριμένους τρόπους.

Μειονεκτήματα

- Το Unity παρόλο που είναι δωρεάν η βασική του έκδοση, για να αποκτήσει πρόσβαση ο χρήστης σε όλες τις λειτουργίες του θα πρέπει να αποκτήσει το Unity plus, το οποίο είναι η premium έκδοση του και είναι με χρέωση.
- Το Unity απαιτεί περισσότερη μνήμη Ram από άλλες μηχανές παιχνιδιών. Για να υπάρξει μια ικανοποιητική λειτουργία του χρειάζεται τουλάχιστον 16GB.
- Ένα από τα πιο κοινά προβλήματα που μπορεί να αντιμετωπίσουν οι χρήστες της Unity είναι ότι δεν υπάρχουν templates (template είναι ένας οδηγός για να φτιάξουμε μια ιστοσελίδα, ένα παιχνίδι, ένα έγγραφο κ.α.) για τις πιο σημαντικές δυνατότητες του παιχνιδιού το οποίο είναι ένα σημαντικό πρόβλημα ειδικά για τους νέους χρήστες διότι δεν έχουν ένα πρότυπο να ακολουθήσουν μέχρι να εξοικειωθούν με την μηχανή.

(Υποκεφάλαιο 2.3) Πλεονεκτήματα και μειονεκτήματα της Godot



Εικόνα 1.5 Λογότυπο της Godot.

Σύνδεσμος για την σελίδα : <https://godotengine.org/>

Παρομοίως με την Unity η Godot είναι επίσης μια μηχανή δημιουργίας παιχνιδιών που υποστηρίζει πολλαπλές πλατφόρμες. Επιπλέον μπορεί να αναπτύξει λογισμικό όπου δεν είναι παιχνίδι. Δημιουργήθηκε το 2001 με το όνομα Larvator και μέσα στα χρόνια άλλαξε αρκετά διαφορετικά ονόματα μέχρι που κατοχυρώθηκε το όνομα Godot.

Πλεονεκτήματα

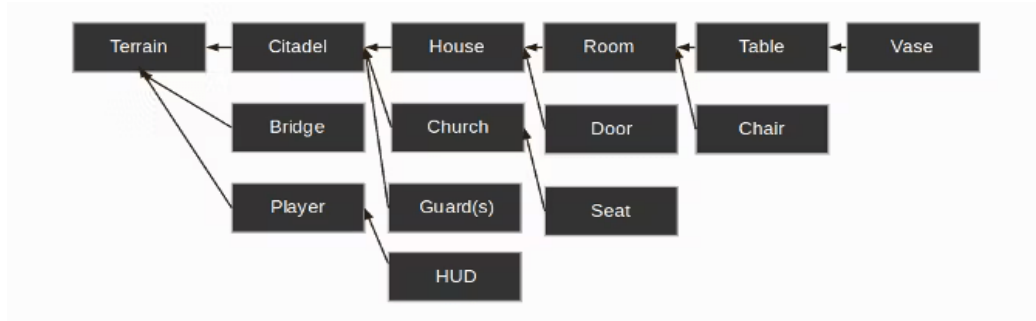
- Το Godot είναι μια μηχανή παιχνιδιών η οποία υποστηρίζει πολλές γλώσσες προγραμματισμού σε σχέση με άλλες αυτές είναι: C++, C# αλλά και GDScript η οποία είναι η γλώσσα που έχει δημιουργήσει η εταιρία της Godot.

```
1 extends Node2D
2 # Declare global variables with specific types to reduce
3 # bugs and errors, especially when the program gets so large
4 var base_input: float = 1.23
5 const ANS_STR: String = "The square of the base is: "
6
7 # Godot built-in function _ready() exists everytime a GDScript is created
8 func _ready():
9     >| # Setting the local variable with type float
10    >| var sqr: float
11    >|
12    >| sqr = squared(base_input)>| # Calling the function
13    >|
14    >| # Convert the results to 2 decimal places and into a string
15    >| # and print it in the Output console
16    >| print(ANS_STR + str(stepify(sqr, 0.01)))
17
18 # Declare a function with
19 # func functionName(some_variables: TYPE) -> TYPE:
20 # TYPE can be int, float, String, bool, etc...
21 func squared(base: float) -> float:
22    >| var sqr: float>| >| >| # Set local variable with type float
23    >|
24    >| sqr = base * base>| >| # Formula for getting the square of a base
25    >|
26    >| return sqr>| >| >| # Return variable as float
```

Εικόνα 1.6 Παράδειγμα δομής κώδικα σε GDScript.

- Οι περισσότερες γλώσσες χρησιμοποιούν σκηνές για να αναπαραστήσουν ένα level εντός του παιχνιδιού. Αυτές οι σκηνές συνήθως περιέχουν διαφορά αντικείμενα τα οποία διαφέρουν σε κάθε γλώσσα για παράδειγμα η Unity έχει τα game objects και η Unreal Engine τα actors. Στην Godot αυτό δεν συμβαίνει. Η κάθε σκηνή αναπαριστάται ως ένα σύνολο κόμβων όπου ο κάθε κόμβος είναι ένα ξεχωριστό αντικείμενο. Ο κάθε κόμβος επίσης έχει την δυνατότητα να κληρονομεί αντικείμενα από τους άλλους κόμβους αν έχουν κάποιον ριζικό κόμβο να τους συνδέει. Αυτό το σύστημα ίσως

χρειάζεται κάποιον χρόνο έτσι ώστε να κατανοηθεί πλήρως αλλά πολλοί από τους προγραμματιστές οι οποίοι εξοικειώνονται μαζί του το εκτιμούν ως ένα διαισθητικό και επεκτάσιμο εργαλείο σχεδίασης.



Εικόνα 1.7 Το σύστημα κόμβων της Godot αναπαριστώντας τον τρόπο με τον οποίο κληρονομούν δεδομένα οι κόμβοι. Ο κάθε κόμβος κληρονομεί δεδομένα από τους κόμβους όπου συνδέεται.

- Η Godot μπορεί να χρησιμοποιηθεί για τον σχεδιασμό και την ανάπτυξη δισδιάστατων (2D) αλλά και τρισδιάστατων παιχνιδιών (3D). Η Godot είναι πιο αποδοτική στην δημιουργία 2D παιχνιδιών βέβαια διότι η 3D λειτουργίες της είναι πιο νέες και όχι τόσο εξελιγμένες σε σχέση με αυτές των άλλων μηχανών. Αυτό δεν σημαίνει πως δεν μπορούν να υπάρξουν καλά αποτελέσματα και στην 3D σχεδίαση απλά αν ο προγραμματιστής θέλει να καλύτερα γραφικά δεν θα πρέπει να επιλέξει την συγκεκριμένη μηχανή.



1.8 Παραγόμενο 2D παιχνίδι με χρήση Godot.

- Η Godot είναι μια μηχανή ανοιχτού κώδικα. Αυτό σημαίνει πως τα δικαιώματα του παιχνιδιού ανοίξουν αποκλειστικά στον προγραμματιστή που το φτιάχνει. Αυτό σημαίνει διότι η μηχανή είναι δωρεάν και κάθε εργαλείο αλλά και τα προγράμματα επεξεργασίας που υπάρχουν μέσα σε αυτή υπάγονται στην άδεια του MIT.
- Η Godot είναι πολύ πιο «ελαφριά» σε σχέση με άλλες μηχανές παιχνιδιών. Ο χώρος που καταλαμβάνει είναι 60mb και δεν απαιτείται εγκατάσταση. Η μόνη διαδικασία που πρέπει να κάνει ο χρήστης είναι να κατεβάσει το συμπιεσμένο αρχείο της Godot, να κάνει αποσυμπίεση και μετά από αυτό μπορεί να ξεκινήσει κανονικά να προγραμματίζει με αυτή. Αυτό την καθιστά και πιο καλή επιλογή για προγραμματιστές που έχουν συστήματα χαμηλότερης ισχύος και αυτό γιατί τα χαρακτηριστικά που απαιτεί υπάρχουν στους περισσότερους υπολογιστές.
- Η Godot είναι μια μηχανή Cross Platform. Αυτό σημαίνει πως υποστηρίζει Mac, Windows αλλά και Linux. Η ίδια η Godot δεν έχει αναφέρει αν είναι cross platform μηχανή αλλά οι χρήστες κάθε διαφορετικού λογισμικού που την έχουν δοκιμάσει είχαν παρόμοιες εμπειρίες μεταξύ τους καθώς και τα παιχνίδια που έφτιαξαν δημιουργούνται για πολλά περιβάλλοντα.

Μειονεκτήματα

- Η Godot δεν είναι μια μηχανή παιχνιδιών που θα χρησιμοποιηθεί για το επόμενο “Big thing” της Gaming βιομηχανίας. Είναι για χρήστες οι οποίοι κάνουν τα πρώτα τους βήματα στην δημιουργία παιχνιδιών ή για εκείνους που θέλουν να σχεδιάσουν ένα παιχνίδι (συνήθως 2D) απευθυνόμενο σε πιο μικρό κοινό.
- Η Godot τουλάχιστον προς το παρόν δεν μπορεί άμεσα να κάνει παραγωγή παιχνιδιού που να είναι διαθέσιμο για κονσόλες. Αυτό συμβαίνει γιατί τα εργαλεία που χρειάζονται για την παραγωγή παιχνιδιών για κονσόλες είναι κλειστού κώδικα και εναντιώνονται με την δέσμευση που έχει η Godot η οποία είναι ανοιχτού κώδικα. Υπάρχουν εργαλεία τα οποία μπορούν να κάνουν την Godot διαθέσιμη για κονσόλες αλλά είναι μια διαδικασία χρονοβόρα και συχνά ένας λόγος για τους προγραμματιστές να μην την επιλέξουν.

(Υποκεφάλαιο 2.4) Πλεονεκτήματα και μειονεκτήματα της Gamemaker



Εικόνα 1.9 Λογότυπο της GameMaker.

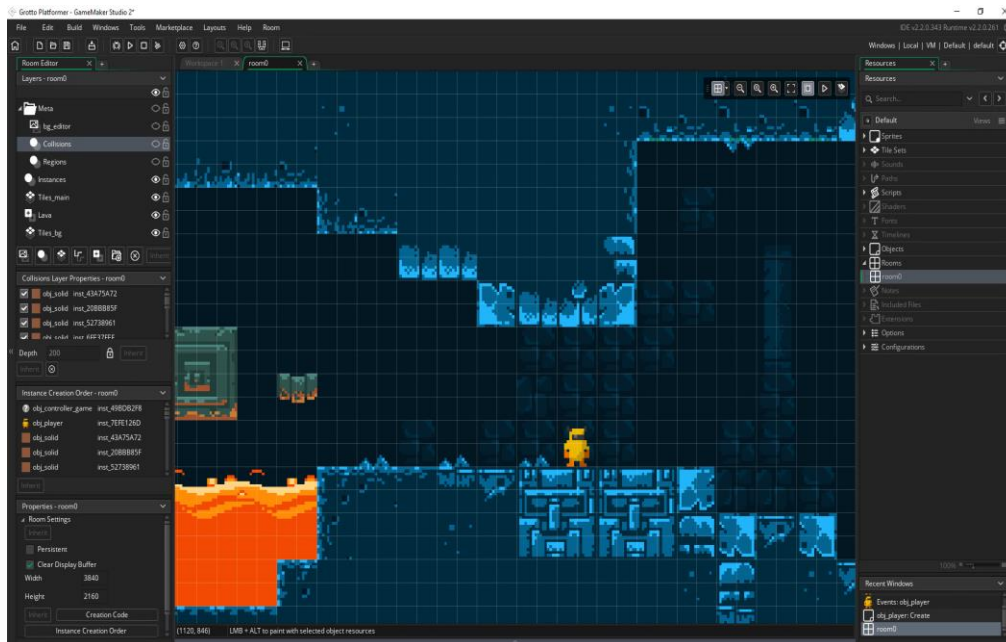
Σύνδεσμος για την σελίδα : <https://gamemaker.io/en>

Η GameMaker είναι ακόμη μια μηχανή δημιουργίας παιχνιδιών που υποστηρίζει πολλαπλές πλατφόρμες. Δημιουργήθηκε το 2007 έχοντας το όνομα Animo μέχρι το 2011 όπου μετονομάστηκε σε GameMaker.

Βασικός λόγος δημιουργίας της ήταν να βοηθήσει ανθρώπους με λίγες προγραμματιστικές γνώσεις να φτιάξουν δικά τους παιχνίδια. Σε αντίθεση με τις άλλες μηχανές η GameMaker χρησιμοποιείται για την δημιουργία 2D παιχνιδιών.

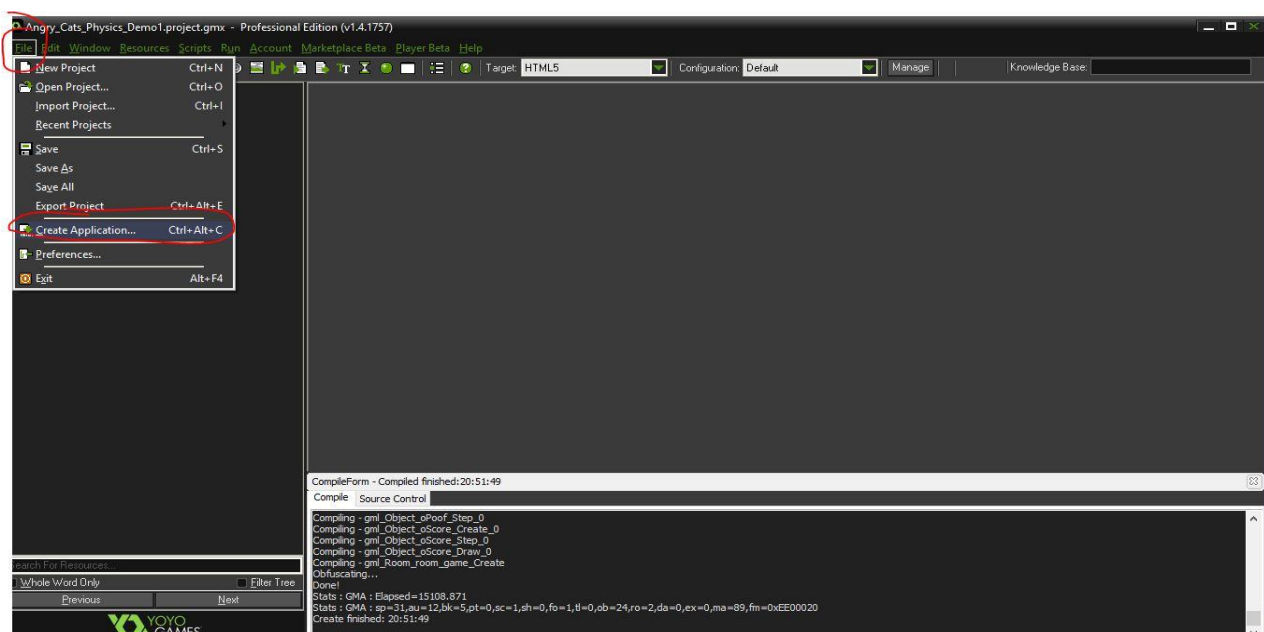
Πλεονεκτήματα

- Όπως και η Godot έτσι και η GameMaker είναι μια μηχανή για αρχάριους και μια μηχανή η οποία απευθύνεται σε στόμα τα οποία θέλουν να ασχοληθούν με την παραγωγή 2D παιχνιδιών. “Παρόλο που απευθύνεται σε αρχάριους όμως η GameMaker είναι η καλύτερη μηχανή σχεδίασης 2D παιχνιδιών και αυτός είναι και ο λόγος για τον οποίο κατασκευάστηκε” όπως ανέφερε και ο συνιδρυτής της Butterscotch Shenanigans, ο Seth Coster.

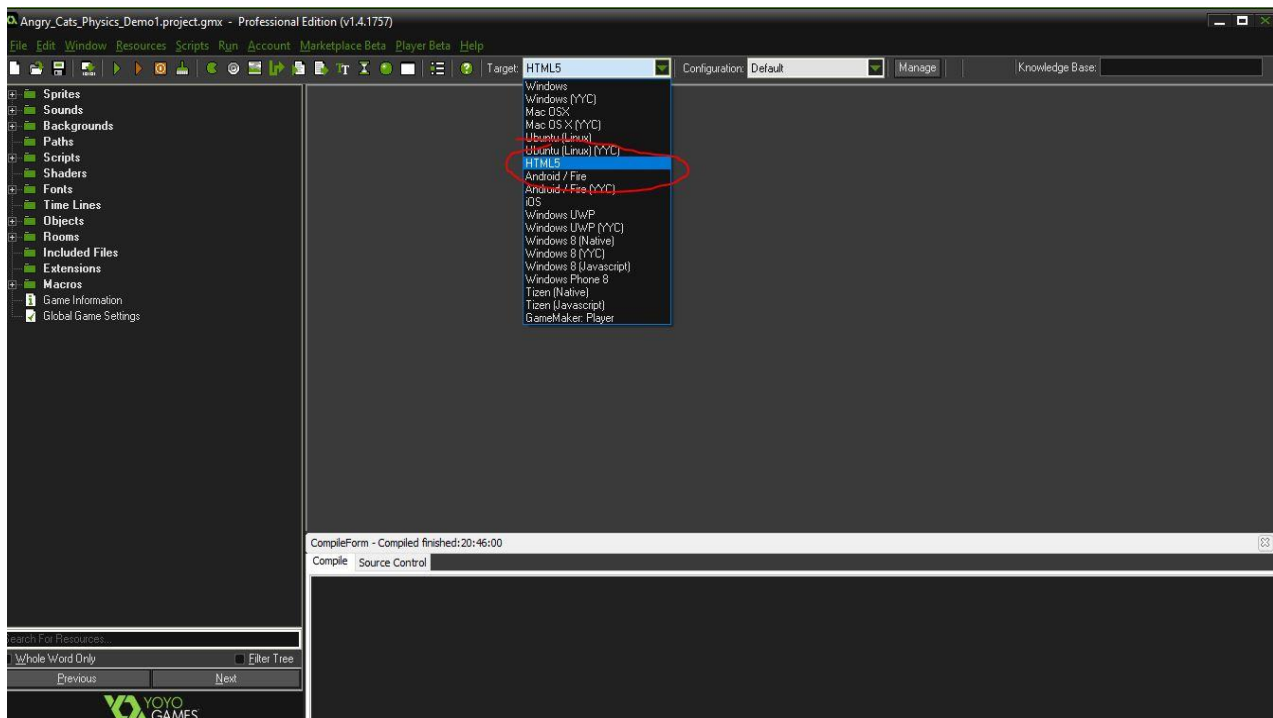


Εικόνα 1.10 WorkSpace της GameMaker κατά την διάρκεια κατασκευής 2D παιχνιδιού.

- Η GameMaker μπορεί να χειριστεί όλα τα ήδη και στυλ 2D παιχνιδιών. Η GameMaker είναι καλύτερη όταν χρησιμοποιείται για 2D παιχνίδια βασισμένα σε εικόνες (sprites). Βέβαια σύμφωνα με τον δημιουργό των Home, Alone with you και Worse than death τον Benjamin Rivers, "Η Game maker μπορεί να εφαρμοστεί σε διαφορά είδη παιχνιδιών εξίσου καλά ασχέτως το πόσο μεγάλο ή μικρό είναι".
- Άλλο ένα θετικό χαρακτηριστικό της GameMaker ειδικά για αρχάριους είναι ότι είναι πολύ γρήγορη αλλά και απλή. Μπορεί ο χρήστης να φτιάξει ένα παιχνίδι το οποίο να τρέχει πολύ γρήγορα και ομαλά. Το θέμα της απλότητας είναι βασισμένο στο σύστημα plug and play το οποίο δεν χρειάζεται σχεδόν καθόλου προγραμματισμός. Το μόνο που χρειάζεται να κάνει ο χρήστης είναι να ενώνει τις εκάστοτε εντολές μεταξύ τους έτσι ώστε να έχει το επιθυμητό αποτέλεσμα.
- Η GameMaker μπορεί να κάνει εξαγωγή του παιχνιδιού για πολλές πλατφόρμες με εύκολο τρόπο. Το μόνο που χρειάζεται είναι ο χρήστης να επιλέξει την πλατφόρμα που θέλει να εξαγάγει το παιχνίδι και να πατήσει εξαγωγή. Τα υπόλοιπα τα κάνει η μηχανή από μόνη της.



Εικόνα 1.11



Εικόνα 1.12 Τρόπος εξαγωγής παιχνιδιού της GameMaker. Διακρίνουμε την ευκολία εξαγωγής η οποία γίνεται κάνοντας click στο file -> create application -> επιλογή γλώσσας εξαγωγής.

Μειονεκτήματα

- Όπως προαναφέρθηκε η GameMaker είναι μόνο για σχεδίαση του 2D παιχνιδιών. Αυτό είναι ένα μειονέκτημα διότι στις μέρες μας μεγαλύτερη απήχηση έχουν τα παιχνίδια 3D.
- Η GML (GameMakerLanguage) μπορεί να οδηγήσει σε κακές συνήθειες. Αυτό σημαίνει πως λόγω της ελαστικότητας και της επιείκειας που έχει η γλώσσα. Όταν ο χρήστης αποφασίσει να αλλάξει μηχανή και γλώσσα προγραμματισμού θα τον εμποδίσει από το να μπορεί να δουλέψει σωστά.
- Αυξημένη δυσκολία εύρεσης προγραμματιστών. Παρόλο που μπορεί να είναι μια απλή μηχανή συχνά οι πιο έμπειροι προγραμματιστές αφήνουν την μηχανή αυτή ώστε να

μάθουν κάποια πιο σύγχρονη και ικανή μηχανή δημιουργίας παιχνιδιών. Αυτό καθιστά την εύρεση προσωπικού πολύ πιο δύσκολη.

- Οι δυνατότητες του GameMaker δεν είναι αρκετές. Αυτό οφείλετε στο ότι έχει πολύ λιγότερους πόρους από την Epic games (κατασκευαστική εταιρία της Unreal Engine) και την Unity Technologies (κατασκευαστική εταιρία της Unity) και αυτό φαίνεται στις δυνατότητες της μηχανής

(Υποκεφάλαιο 2.5) Πλεονεκτήματα και μειονεκτήματα της Unreal Engine



Εικόνα 1.13 Λογότυπο της Unreal Engine.

Σύνδεσμος για την σελίδα : <https://www.unrealengine.com/en-US?sessionInvalidated=true>

Η Unreal Engine ως μηχανή δημιουργίας παιχνιδιών εμφανίστηκε για πρώτη φορά το 1998 απευθυνόμενη σε shooter παιχνίδια πρώτου προσώπου για υπολογιστή. Με το πέρασμα του χρόνου και την αναβάθμιση των γραφικών και των δυνατοτήτων της είδαμε να χρησιμοποιείται και από την βιομηχανία του κινηματογράφου και της τηλεόρασης. Πλέον εκτός από παιχνίδια για υπολογιστές την βλέπουμε να χρησιμοποιείται για κινητά και άλλες συσκευές όπως ακριβώς οι προαναφερθείσες μηχανές δημιουργίας παιχνιδιών.

Πλεονεκτήματα

- Ένα βασικό πλεονέκτημα της Unreal Engine είναι τα γραφικά της. Είναι η μηχανή η οποία μπορεί να προσφέρει τα πιο ρεαλιστικά γραφικά στα παιχνίδια που φτιάχνονται μέσω αυτής. Βέβαια τα γραφικά δεν κρίνονται μόνο με βάση τον ρεαλισμό. Ένας προγραμματιστής μπορεί να δώσει έμφαση και σε ένα στυλ γραφικών cartoon. Το πιο συμβολικό παράδειγμα είναι αυτό του Fortnite από την Epic Games που χρησιμοποίησε την Unreal Engine έτσι ώστε να αποδώσει στα χρώματα στο cartoon style παιχνίδι της και από ότι φάνηκε από το αποτέλεσμα και τις πωλήσεις του παιχνιδιού πέτυχε. Το συγκεκριμένο αποτέλεσμα δεν θα μπορούσε να το πετύχει με καμία άλλη μηχανή παιχνιδιών διότι δεν έχουν τις δυνατότητες απόδοσης γραφικών και χρωμάτων της Unreal Engine. Αλλά παιχνίδια τα οποία χρησιμοποίησαν στο μέγιστο τις δυνατότητες της Unreal Engine στα γραφικά είναι το Gears of war το οποίο πούλησε πάνω από 26 εκατομμύρια αντίτυπα και ξεχώρισε για τα κατεστραμμένα περιβάλλοντα μέσα στην πόλη αλλά και το Hellblade: Senua's Sacrifice το οποίο ξεχώρισε για τους ρεαλιστικούς χαρακτήρες και το περιβάλλον του κερδίζοντας μάλιστα πολλά βραβεία οπτικού σχεδιασμού.

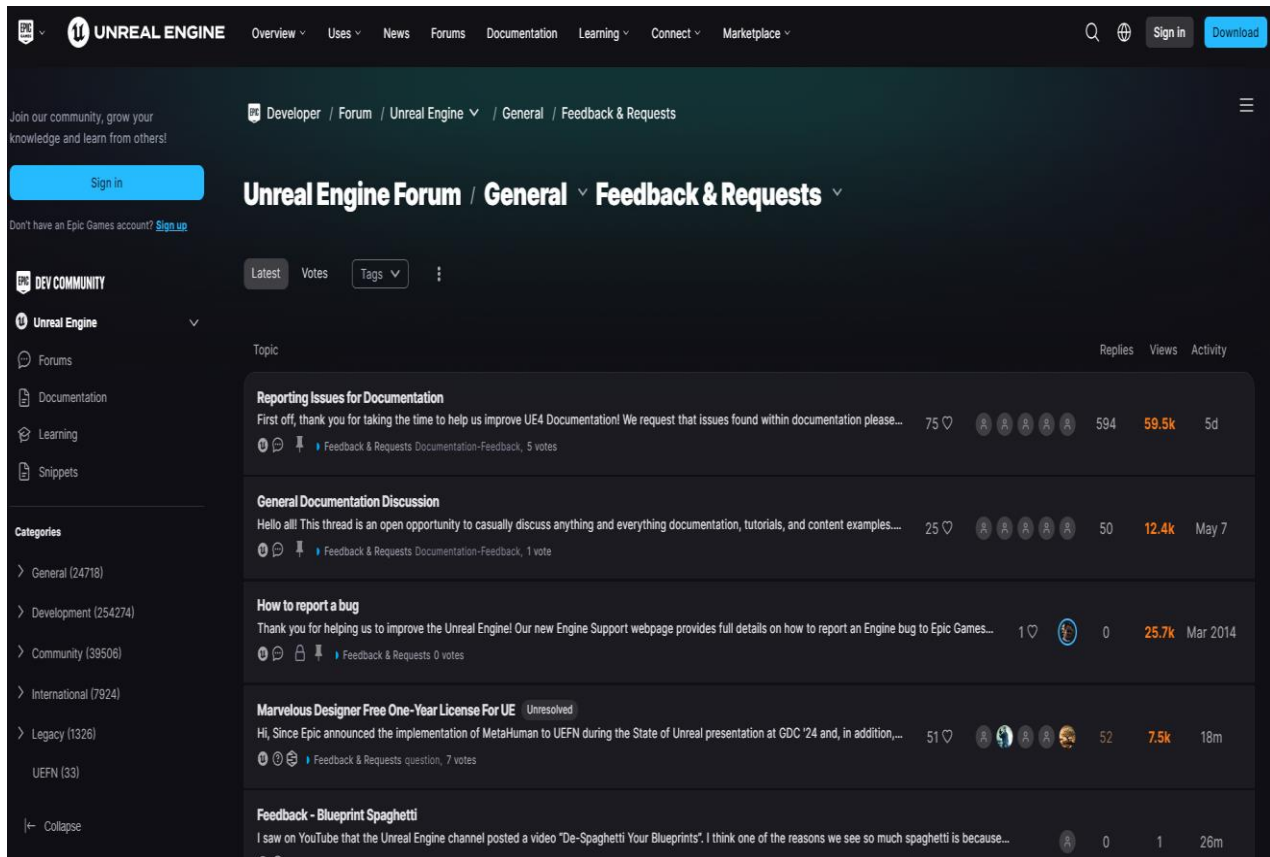


Εικόνα 1.14 Στιγμιότυπο από το παιχνίδι Fortnite όπου διακρίνουμε το cartoon style γραφικών.



Εικόνα 1.15 Στιγμιότυπο από το Hellblade: Senua's Sacrifice όπου διακρίνουμε τις δυνατότητες της Unreal Engine στον ρεαλισμό των γραφικών.

- Λόγω της τεράστιας εξέλιξης της η Unreal Engine έχει ένα τεράστιο κοινό το οποίο παρέχει συνεχή υποστήριξη σε νέους προγραμματιστές οι οποίοι έχουν δυσκολίες στα πρώτα τους βήματα με αυτή. Επιπλέον η ίδια η Epic(η εταιρία παραγωγής την Unreal Engine) παρέχει forums προωθώντας έτσι αυτόν τον τρόπο επίλυσης προβλημάτων για τους προγραμματιστές που αντιμετωπίζουν προβλήματα κάνοντας έτσι πιο εύκολη και άμεση την επικοινωνία μεταξύ των χρηστών.



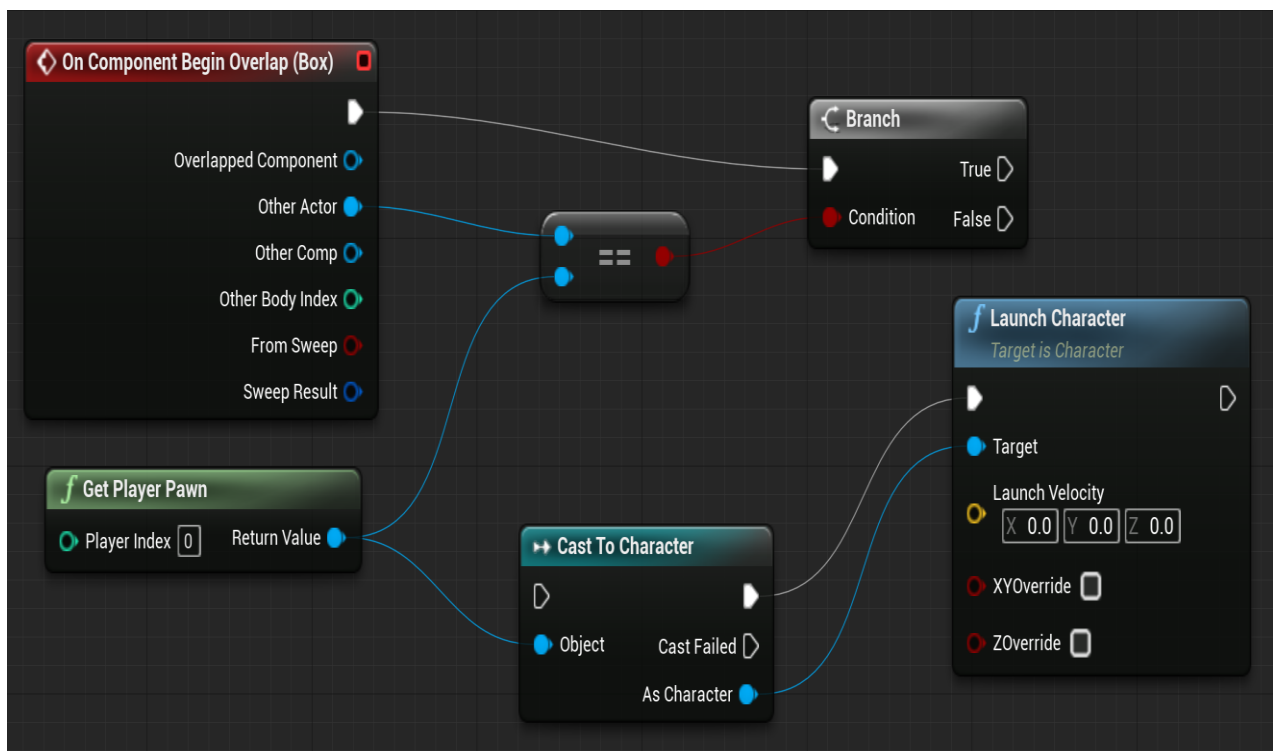
Εικόνα 1.16 Forum της Unreal Engine. Σε αυτό μπορούν όλοι οι χρήστες να ζητήσουν βοήθεια για τα προβλήματα που ενδεχομένως να αντιμετωπίζουν στον κώδικα τους.

- Η Unreal Engine παρέχει εργαλεία για την ανάπτυξη συνθετών παιχνιδιών. Παρόλο που η εφαρμογή είναι φιλική σε αρχάριους προγραμματιστές στα χέρια επαγγελματιών με χρόνια εμπειρίας αυτά τα εργαλεία μπορούν να ανεβάσουν τα παιχνίδια που θα παραχθούν στο επόμενο επίπεδο. Κάποια από τα βασικά εργαλεία είναι η αρχιτεκτονική δικτύωσης η οποία συνεισφέρει στην παραγωγή online παιχνιδιών αλλά και ζωντανές υπηρεσίες ώστε να μπορούν ταυτόχρονα οι παίχτες να βλέπουν τα ίδια πλάνα, στιγμιότυπα αλλά και τους ίδιους του αντιπάλους παίχτες στην οθόνη τους.

Επίσης παρέχει προηγμένα εργαλεία AI όπως για παράδειγμα δέντρα συμπεριφοράς (behaviour trees) και νευρωνικά δίκτυα (neural networks) που βοηθάνε στην παραγωγή πιο ρεαλιστικών εχθρών ως προς την συμπεριφορά τους.

Μειονεκτήματα

- Παρόλο που η Unreal Engine είναι φιλική προς αρχάριους ο τρόπος για να μάθεις την μηχανή είναι πιο δύσκολος από άλλες όπως η Unity, η Godot ή GameMaker. Οι χρήστες χρειάζονται μήνες για να μάθουν τα εργαλεία που τους προσφέρει η Unreal Engine πριν αρχίσουν να προγραμματίζουν.



Εικόνα 1.17 Η γλώσσα της Unreal Engine με χρήση blueprint.

- Η Epic Games κρατάει ένα 5% των δικαιωμάτων του παιχνιδιού αν τα κέρδη του ξεπεράσουν το 1.000.000€. Αυτό είναι ένα μειονέκτημα το οποίο δεν αφορά όλους τους

χρήστες διότι αν δεν βρίσκονται σε μια εταιρία ώστε να έχουν το κατάλληλο budget και την προώθηση της είναι πολύ δύσκολο να φτάσουν αυτά τα κέρδη αλλά είναι ένα μειονέκτημα για το οποίο υπάρχουν χρήστες που δεν την επιλέγουν.

- Η Unreal Engine βασίζεται κατά βάση σε παραγωγή παιχνιδιών 3D και οι λειτουργίες της για κινητά και 2D παιχνίδια δεν είναι το ίδιο ικανές καθώς τα εργαλεία της είναι 3D και απαιτούν πόρους κάνοντας την παραγωγή παιχνιδιού για κινητά και 2D παιχνίδια εξαιρετικά δύσκολη.

Αν συγκρίνουμε τα πλεονεκτήματα και τα μειονεκτήματα των παραπάνω μηχανών για την παραγωγή ενός 3D Single player παιχνιδιού με χρήση A.I., φτάνουμε στο συμπέρασμα πως η Unreal Engine είναι η πιο ικανή μηχανή για να ικανοποιήσει αυτόν τον σκοπό. Αυτός είναι και ο λόγος που την επέλεξα για την παραγωγή του παιχνιδιού αυτής της πτυχιακής εργασίας διότι παρέχει πολύ καλύτερη ποιότητα στην εικόνα του παιχνιδιού αλλά ταυτόχρονα και πιο πολλές δυνατότητες στο gameplay αλλά και στην συμπεριφορά των AI εχθρών.

ΚΕΦΑΛΑΙΟ 3 ΑΙ ΣΤΑ ΠΑΙΧΝΙΔΙΑ

(Υποκεφάλαιο 3.1) Εισαγωγή

Η χρήση AI είναι ένας μηχανισμός που τα τελευταία χρόνια παρατηρείται στα περισσότερα παιχνίδια. Το συναντάμε σε χαρακτήρες οι οποίοι δεν είναι πραγματικοί παίκτες και ονομάζονται NPC (nonplayer character).

Τα NPC σε ένα παιχνίδι μπορεί να είναι αντίπαλοι, σύμμαχοι, φίλοι του χαρακτήρα αλλά και απλοί περαστικοί στον δρόμο έτσι ώστε να υπάρχει η αίσθηση της ρεαλιστικότητας στο περιβάλλον του παιχνιδιού. Τα NPC δεν χρησιμοποιούνται απλά για να υπάρχουν ως παρουσίες στον χώρο αλλά έχουν συνομιλίες με τον χαρακτήρα και αλλαγές συμπεριφοράς και κινήσεων ανάλογα με τις ενέργειες που θα εκτελέσει ο παίχτης. Αυτό συμβαίνει διότι με την χρήση του AI τα NPC μαθαίνουν από την αλληλεπίδραση με τον παίχτη όποτε μπορούν να αντιδράσουν με τον

ανάλογο τρόπο. Επιπλέον το ΑΙ πέρα από χαρακτήρες μπορούμε να το βρούμε και στο περιβάλλον του παιχνιδιού με την εδάφους, κτηρίων, βουνών κ.α. Σε αυτή την περίπτωση διακρίνουμε την τεχνητή νοημοσύνη στον τρόπο αναδιαμόρφωσης του περιβάλλοντος ανάλογα με τις αποφάσεις του χαρακτήρα για παράδειγμα με την καταστροφή ενός τοίχου όταν ο χαρακτήρας τρακάρει με ένα αυτοκίνητο πάνω του. Αυτό έχει ως αποτέλεσμα το ΑΙ να προσφέρει μια πολυπλοκότητα στα παιχνίδια, να τα κάνει πιο ευχάριστα για τον παίκτη αλλά και να υπάρχει εξέλιξη στις δυνατότητες παραγωγής νέων μηχανισμών για τα παιχνίδια.



Εικόνα 2.1 Παράδειγμα NPC χαρακτήρα.

(Υποκεφάλαιο 3.2) Τρόποι λειτουργίας του ΑΙ στα παιχνίδια

Το ΑΙ με την χρήση διαφόρων τεχνολογιών και τεχνικών κάνει τα NPC να μοιάζουν με ανθρώπους και να μπορούν να πάρουν αποφάσεις. Αυτές οι τεχνολογίες είναι οι εξής:

Δέντρα απόφασης Τα δέντρα αποφάσεων είναι αλγόριθμοι μηχανικής μάθησης οι οποίοι κάνουν μετάφραση δεδομένων σε μεταβλητές που μπορούν να αξιολογηθούν. Με αυτές τις μεταβλητές βάζουν κανόνες στα NPC για να ακολουθούν έτσι ώστε να πάρουν αποφάσεις με βάση αυτούς τους κανόνες. Ένα παράδειγμα είναι σε ένα αντίπαλο NPC αν ο παίχτης μπει στην εμβέλεια όρασης του κρατώντας όπλο στο χέρι το NPC να πάρει αμυντική στάση απέναντι του διότι θα καταλάβει πως είναι εχθρός του.

Ενισχυτική μάθηση Η ενισχυτική μάθηση είναι μια μέθοδος βαθιάς μάθησης σε μορφή δοκιμής και λάθους κάνοντας έτσι μια αξιολόγηση με βάση τις ανταμοιβές και τις συνέπειες των ενεργειών που εκτελέστηκαν. Για παράδειγμα ένα NPC κατά την διάρκεια του παιχνιδιού μπορεί να πάρει μια απόφαση με την οποία να πεθάνει. Η συνέπεια αυτής της απόφασης αποθηκεύεται στην βάση δεδομένων του NPC έτσι ώστε την επόμενη φορά που θα φτάσει στο σημείο να πάρει μια παρόμοια απόφαση να μην εκτέλεση την συγκεκριμένη αλλά κάποια άλλη η οποία μπορεί να το οδηγήσει στο να ζήσει περισσότερο ή να νικήσει τον παίχτη. Μέσω της ενισχυτικής μάθησης τα NPC προσπαθούν να μεγιστοποιήσουν τις θετικές συνέπειες των αποφάσεων τους ώστε να παίρνουν πιο σωστές αποφάσεις.

Νευρωνικά δίκτυα Τα νευρωνικά δίκτυα είναι αλγόριθμοι που εκπαιδεύονται με την χρήση συγκεκριμένων συνόλων δεδομένων. Έχουν την ικανότητα προσαρμογής σε διαφορετικά σύνολα δεδομένων. Μέσω αυτής μπορούν να μαθαίνουν καθώς τρέχει το παιχνίδι βελτιώνοντας έτσι τα αποτελέσματα τους και να καλύπτουν μεγάλο αριθμό διαφορετικών σεναρίων. Η χρήση νευρωνικών δικτύων στα NPC τους δίνει την ικανότητα να αλλάζουν συμπεριφορές ανάλογα τις ενέργειες των παιχτών αυξάνοντας έτσι την δυσκολία του παιχνιδιού αλλά και την ρεαλιστικότητα του. Για παράδειγμα σε ένα παιχνίδι προσομοίωσης ποδοσφαίρου τα NPC παρακολουθούν τις κινήσεις των αντιπάλων παιχτών που καθοδηγούνται από τον χρήστη ώστε να πάρουν την καλύτερη απόφαση για να κάνουν πιο σωστό το μαρκάρισμα στους αντιπάλους παίχτες και να ανακτήσουν την κατοχή της μπάλας.

(Υποκεφάλαιο 3.3) Η εξέλιξη του ΑΙ στην βιομηχανία των παιχνιδιών

Με την εξέλιξη της τεχνολογίας τα παιχνίδια ΑΙ συνεχώς βελτιώνονται. Όπως ανέφερα και πιο πάνω τα NPC έχουν αρχίσει να παίρνουν δικές τους αποφάσεις ανάλογα με τις ενέργειες του χρήστη και τους κανόνες που τους δίνει ο προγραμματιστής. Δεν είναι καθόλου απίθανο στο μέλλον να δούμε τα NPC να έχουν μια μεγαλύτερη ελευθερία στις αποφάσεις τους. Ακόμα μπορεί να δούμε τα NPC να έχουν δικό τους χαρακτήρα και συναισθήματα μέσω του ΑΙ το οποίο θα δίνει μια αίσθηση ανθρωπιάς. Επιπλέον με την ανάπτυξη του ChatGPT και άλλες παρόμοιες τεχνολογίες που σιγά σιγά μπαίνουν στο Gaming για να υπάρχει μεγαλύτερη πολυπλοκότητα στις συνομιλίες με τον χρήστη. Αυτό δεν φαίνεται να είναι τόσο μακριά από εμάς διότι η NFT Gaming Company ανακοίνωσε πως στα σχέδια της είναι να προσπαθήσει να χρησιμοποιήσει το ChatGPT για την απόδοση διαλόγων. Ακόμη το Blockchain έχει ενταχθεί τα τελευταία χρόνια στο Gaming επιτρέποντας στους χρήστες να αλλάζουν την εμφάνιση των παιχτών τους παράγοντας διαφορά χρώματα για τα ρούχα, το δέρμα και τα όπλα του παίχτη. Τέλος το Generative AI είναι ένα εργαλείο που ήδη χρησιμοποιείται από πολλούς σχεδιαστές για να παραχθούν κτήρια, δάση και γενικότερα να τους βοηθήσουν να ολοκληρώσουν το περιβάλλον τον παιχνιδιών τους αλλά και να τους γλιτώσει χρόνο. Το επόμενο βήμα θα είναι το ΑΙ να μπορεί να δημιουργήσει ολόκληρα παιχνίδια μόνο του κάτι το οποίο δεν μοιάζει ουτοπικό αν δούμε το πόσο γρήγορα αναπτύσσεται η τεχνητή νοημοσύνη και τι δυνατότητες έχει.

(Υποκεφάλαιο 3.4) Μεγάλοι τίτλοι παιχνιδιών με χρήση ΑΙ

Αξίζει να αναφέρουμε κάποιους από τους σπουδαιότερους τίτλους που αξιοποιούν σε μεγάλο βαθμό το ΑΙ. Ένας από αυτούς είναι το The Last of Us:Part 1. Το The last of us είναι ένα παιχνίδι το οποίο είχε μεγάλη επιτυχία στο PlayStation 4 όταν είχε βγει και έτσι η Naughty Dog (η εταιρία παραγωγής του) αποφάσισε να κάνει re-mastering το παιχνίδι δηλαδή να το επανακυκλοφορήσει για το PlayStation 5. Μετά την επανακυκλοφόρηση του είδαμε βελτιώσεις στο ΑΙ με τα NPC που ήταν σύμμαχοι του χαρακτήρα να μπορούν να εντοπίζουν εχθρούς και να προσαρμόζουν τις ενέργειες τους ανάλογα την κατάσταση. Επίσης οι

κινήσεις τους ήταν πιο αθόρυβες το οποίο χρησιμεύει έτσι ώστε να μην μπορούν να σε ακούσουν οι εχθροί κάτι που στο αρχικό παιχνίδι δεν υπήρχε σε τέτοιο βαθμό. Άλλο ένα παιχνίδι είναι το Alien : Isolation. Σε αυτό ο βασικός εχθρός είναι ένας τεράστιος εξωγήινος ο οποίος μέσω του ΑΙ διαθέτει ισχυρή όσφρηση και ακοή αλλά και όραση που το βοηθά να προσαρμόζεται σε αυτό που βλέπει. Επιπλέον υπάρχει και ένας μετρητής απειλής με τον οποίο όταν ανεβαίνει σε υψηλά επίπεδα ο εξωγήινος υποχωρεί έτσι ώστε ο παίχτης να βιώνει μια ένταση που υποχωρεί και επανέρχεται συνεχώς στο παιχνίδι. Εν κατακλείδι το ΑΙ στα παιχνίδια έχει απεριόριστες δυνατότητες και καθημερινά οι προγραμματιστές ανακαλύπτουν και άλλες. Αυτό έχει ως αποτέλεσμα την συνεχή ανάπτυξη του Gaming όχι μόνο στον τομέα της εικόνας αλλά και του τρόπου παιχνιδιού.



Εικόνα 2.2



Εικόνα 2.3 Στιγμιότυπα από το The last of us



Εικόνα 2.4



Εικόνα 2.5 Στιγμιότυπα από το Alien: Isolation.

ΚΕΦΑΛΑΙΟ 4 – SUICIDE MISSION Z

(Υποκεφάλαιο 4.1) Σενάριο και ανάλυση χαρακτήρων

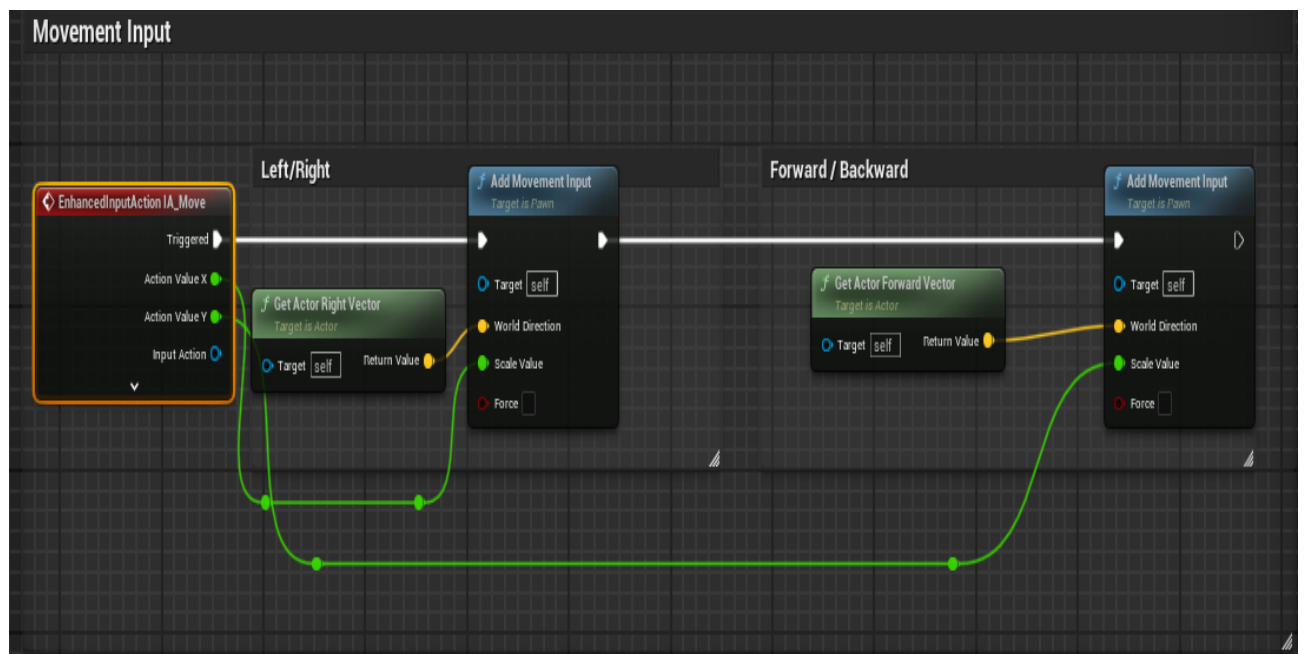
Στα πλαίσια αυτής της πτυχιακής εργασίας έφτιαξα ένα παιχνίδι με τίτλο SUICIDE MISSION Z. Είναι ένα παιχνίδι single player, πρώτου προσώπου που ανήκει στις κατηγορίες δράσης, φαντασίας, περιπέτειας και shooter. Στο παιχνίδι ο βασικός χαρακτήρας ονομάζεται Trey και είναι ένας στρατιώτης των ειδικών δυνάμεων της Αμερικής. Το παιχνίδι διαδραματίζεται 15 χρόνια μετά από ένα ατύχημα που έγινε σε ένα εργαστήριο όπου έκαναν πειράματα για να βρουν ένα αντίδοτο για τον καρκίνο. Το εργαστήριο ανατινάχτηκε και εξαπλώθηκε ένα αέριο κατά μήκος της πόλης που βρισκόταν κοντά σε αυτό. Αυτό το αέριο επηρεάζει τα ανθρώπινα εγκεφαλικά κύτταρα σκοτώνοντας τα και κάνοντας τους ανθρώπους ένα είδος Zombie. Στην πόλη που βρίσκεται κοντά στο εργαστήριο υπήρχαν επιστήμονες που προσπάθησαν να βρουν αντίδοτο για

το αέριο αυτό ώστε να επαναφέρουν τους ανθρώπους στην κατάσταση που ήταν πριν και τα κατάφεραν. Όμως το αέριο εξαπλώθηκε πιο γρήγορα από ότι περίμεναν με αποτελέσματα να μετατραπούν και αυτοί σε Zombie χωρίς να προλάβουν να χρησιμοποιήσουν το αντίδοτο. 15 χρόνια αργότερα λίγο πριν το σημείο όπου διαδραματίζεται η ιστορία του Trey οι ειδικές δυνάμεις αποφάσισαν να στείλουν μια ομάδα επιλεκτών στρατιωτών στην πόλη τώρα που το αέριο δεν υπήρχε πια ώστε να βρουν το αντίδοτο και να επαναφέρουν τους ανθρώπους όπου είχαν παραμείνει ζωντανοί στην κανονική τους κατάσταση. Ενώ υπήρξαν πληροφορίες ότι βρέθηκε η συνταγή για το αντίδοτο ξαφνικά η επαφή με την ομάδα χάθηκε. Έτσι ερχόμαστε στο σήμερα όπου ο Trey καλείται να πάει μια αυτός στην πόλη ώστε να βρει το αντίδοτο αλλά και να μάθει τι απέγιναν τα μέλη της ομάδας. Καθόλη την διάρκεια της αποστολής είναι συνεχώς σε επικοινωνία με την Master Plan όπου είναι η υπεύθυνη της αποστολής ώστε να του δίνει οδηγίες. Στην αποστολή του ο Trey ανακαλύπτει από την πρώτη στιγμή πως τα Zombie είναι επιθετικά ως προς τους ανθρώπους όποτε καλείται να τα αντιμετωπίσει για να επιβιώσει αλλά και να φέρει την αποστολή εις πέρας. Καθώς εξερευνά την πόλη ο Trey λαμβάνει συντεταγμένες από την Master plan ώστε να πάει να ελέγξει μια περιοχή που πιθανώς να βρίσκεται ένα μέλος από την ομάδα. Καθώς φτάνει στο σημείο συναντά τον Jake (ένα από τα πέντε μέλη της ομάδας) ο οποίος είναι βαριά τραυματισμένος και περικυκλωμένος από Zombies. Όταν ο Trey κατορθώνει να σκοτώσει τα Zombies, κατευθύνεται προς τον Jake ο οποίος του λέει να μην προσπαθήσει να τον βοηθήσει διότι δεν θα καταφέρει να επιβιώσει από τα τραύματα του και να πάει στην κοντινή παιδική χαρά όπου βρισκόταν ο Mike (το δεύτερο μέλλον της ομάδας) πριν χωριστούν μήπως καταφέρει και σώσει εκείνον. Ο Trey αποχαιρετά τον φίλο του και πάει να ψάξει για τον Mike. Όταν φτάνει στην παιδική χαρά παίρνοντας από έναν δρόμο γεμάτο Zombie βρίσκει τον Mike νεκρό και ζητάει από την Master Plan οδηγίες για την επόμενη αποστολή του ενώ πρώτα της αναφέρει πως οι 2 από τους 5 συνεργάτες του είναι νεκροί. Η Master Plan καθοδηγεί τον Trey στο τελευταίο σημείο όπου είχαν επαφή με την ομάδα και της είχαν δώσει την πληροφορία ότι κάτι βρήκαν. Όταν καταφθάνει στο σημείο ο Trey βρίσκει ένα σκισμένο χαρτί όπου είναι το πρώτο μισό του αντιδότου για τον ιό και κάτω δεξιά γραμμένες συντεταγμένες χωρίς καμία περαιτέρω πληροφορία. Ακολουθώντας τις συντεταγμένες φτάνει σε ένα γήπεδο μπάσκετ όπου για άλλη μια φορά έρχεται αντιμέτωπος με Zombies, ανακαλύπτει το άλλο μισό της συνταγής όπου είχε βρει η ομάδα και το είχε χωρίσει στα δυο για μεγαλύτερη ασφάλεια. Ενώ έδινε ενημέρωση στα κεντρικά ότι βρήκε το αντίδοτο η Master Plan, η οποία είχε λάβει σήμα κινδύνου από το γήπεδο που βρισκόταν κοντά στο σημείο όπου ήταν ο Trey, να πάει γρήγορα στην περιοχή ώστε να ελέγξει το σημείο και να δει αν το σήμα έρχεται από μέλη

της ομάδας. Με την άφιξη του στο σημείο ο Trey έμεινε άφωνος. Ένα γιγάντιο Zombie βρισκόταν εκεί μαζί με άλλα πολλά από τα Zombies τα οποία είχε έρθει αντιμέτωπος ήδη ενώ βρισκόταν στην περιοχή και 2 από τα μέλη της ομάδας του να είναι δεμένα πάνω στον φράχτη που περιτριγύριζε το γήπεδο. Δίνοντας την πιο δύσκολη μάχη που είχε δώσει μέχρι στιγμής όσο βρισκόταν εκεί καταφέρει να εξολοθρεύσει τα Zombies και να ελευθερώσει τους συναδέλφους του. Εκείνοι έκπληκτοι που το βλέπουν, αφού πίστευαν ότι δεν θα σωθούν, του εξηγούν την ιστορία για το πως βρέθηκαν δεμένοι στον φράχτη. Όπως φαίνεται κάποιοι ληστές όπου έκαναν επιδρομή στην περιοχή τους αφοπλίσαν και τους έδεσαν αλλά δεν κατάφεραν να γλιτώσουν από το γιγάντιο Zombie που φύλαγε την περιοχή. Ο Trey τους ρωτάει αν έχουν στοιχεία για το που βρίσκεται ο Ryan (το πέμπτο και τελευταίο μέλος της ομάδας) αλλά εκείνοι δεν είχαν ιδέα. Είπαν στον Trey μόνο το τελευταίο μέρος όπου είχαν ενημέρωση για τον Ryan και εκείνος κατευθύνθηκε αμέσως στο σημείο. Φτάνοντας εκεί συναντά μια μικρή καφετέρια και πάνω σε ένα από τα τραπέζια της ένα γράμμα. Έκπληκτος βλέπει πως το γράμμα είναι από τον Ryan προς την ομάδα του λέγοντας τους ότι δεν αντέχει άλλο όλη αυτή την κατάσταση με τα Zombie και ότι θα πάει σε ένα μέρος όπου θα είναι μόνος του να αυτοκτονήσει ενώ πρώτα ευχαρίστησε την ομάδα του για ότι πέρασαν μαζί και λέγοντας τους ότι κάποια στιγμή θα ξανά ειπωθούν. Ο Trey αποφασίζει να ψάξει για να βρει τι απέγινε ο Ryan και φτάνει σε μια γέφυρα όπου συναντά το πτώμα του Ryan να επιπλέει στο νερό. Ο Trey ενώ είναι συναισθηματικά φορτισμένος λόγω της κατάστασης των πρώην συνεργατών του καλείται από την Master Plan να πάει στο ελικόπτερο όπου τον περίμενε για να φύγει από την πόλη. Έτσι μπαίνοντας στο αυτοκίνητο που του είχαν αφήσει στην πόλη για την αποστολή μέσω ενός ελικόπτερο ο Trey καταφθάνει στο ελικόπτερο και η αποστολή του τελειώνει επίσημα ενώ φεύγει από την πόλη. Τέλος η Master Plan ευχαριστεί τον Trey για τις υπηρεσίες του και τον αποχαιρετά λέγοντας του ότι θα ξανά βρεθούν στην επόμενη αποστολή που θα του ανατεθεί.

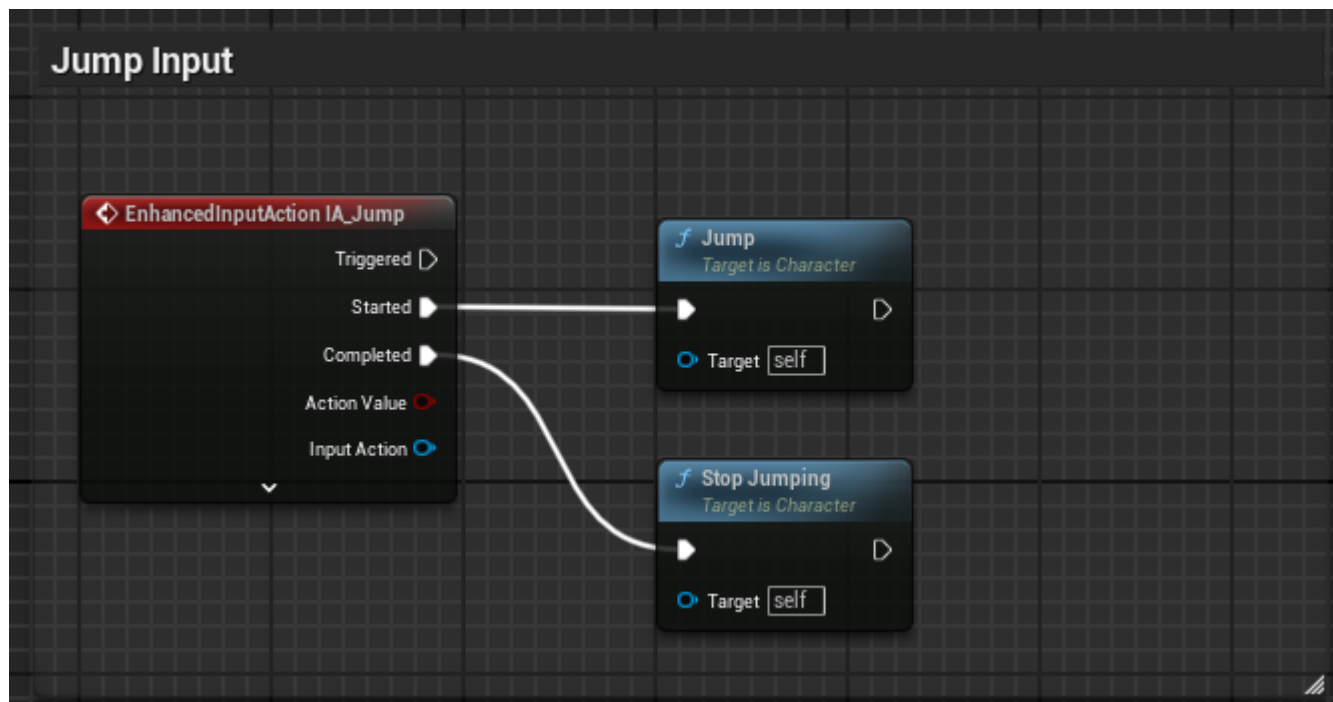
(Υποκεφάλαιο 4.2) Επεξήγηση κώδικα

Ας ξεκινήσουμε την επεξήγηση του κώδικα από τον βασικό χαρακτήρα του παιχνιδιού.



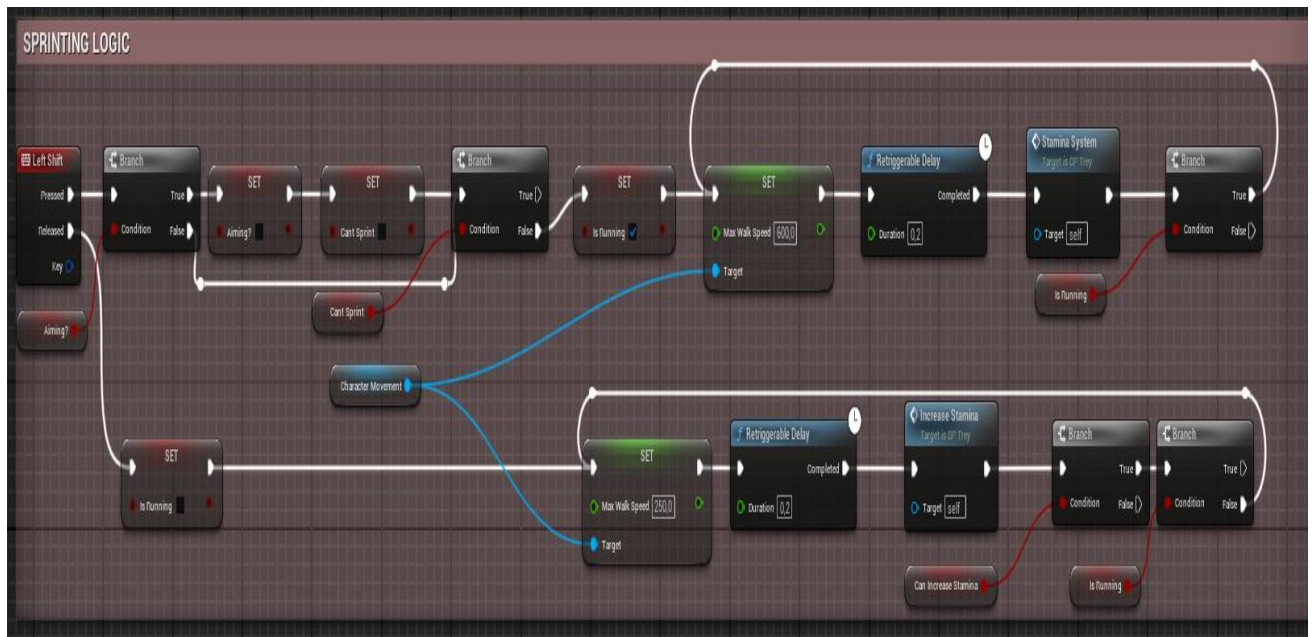
Εικόνα 3.1 Κώδικας για τον τρόπο κίνησης του χαρακτήρα.

Όπως διακρίνουμε στην παραπάνω εικόνα αυτός είναι ο κώδικας για τον τρόπο με τον οποίο κινείται ο βασικός χαρακτήρας στον χώρο. Αρχικά χρησιμοποιούμε ένα Custom event το οποίο γίνεται trigger όταν ο χρήστης πατήσει τα κουμπιά W S D A. Όταν γίνει trigger με τα κουμπιά A και D ενεργοποιείται η πρώτη εντολή Add movement input όπου εισάγοντας σε αυτή τον vector του άξονα y του χαρακτήρα τον κάνει να κινείται αριστερά αν ο παίχτης πατήσει το κουμπί A και αριστερά αν πατήσει το κουμπί D. Αντίστοιχα για το δεύτερο Add movement Input αν ο παίχτης πατήσει τα κουμπιά W S εισάγουμε τον άξονα x του χαρακτήρα ώστε να του δώσουμε κίνηση προς τα μπροστά ή προς τα πίσω. Και στις δυο περιπτώσεις το blueprint λειτουργεί με μια εντολή if όπου το Scale value έχει την τιμή 0 ως προκαθορισμένη και αυξάνεται αν ο παίχτης πατήσει τα κουμπιά D για δεξιά και W για πάνω και μειώνεται αν πατήσει τα κουμπιά A για αριστερά και S για κάτω. Αν το Scale έχει τιμή μεγαλύτερη του 0 τότε δίνει κίνηση στο actor του χαρακτήρα προς τα δεξιά αν είναι το πρώτο input και μπροστά αν είναι το δεύτερο. Αν έχει αρνητική τιμή τότε ξανά δίνει αριστερά κίνηση στο actor του χαρακτήρα αν είναι στο πρώτο Input και κάτω αν είναι στο δεύτερο.



Εικόνα 3.2 Κώδικας για το πώς ο χαρακτήρας εκτελεί άλμα.

Για το άλμα χρησιμοποιούμε ένα Custom event όπου γίνεται trigger όταν ο χρήστης πατήσει το spacebar. Σε αυτή την περίπτωση καλείται η εντολή Jump όπου αλλάζει την τιμή του άξονα z έτσι ώστε να δώσει την ψευδαισθηση ότι ο χαρακτήρας κάνει άλμα. Όταν ολοκληρωθεί το άλμα εκτελείται αυτόματα η εντολή stop jumping όπου επιστρέφει το actor του χαρακτήρα στην θέση όπου βρισκόταν πριν κάνει το άλμα. Ο αυτοματισμός αυτός συμβαίνει διότι έχουμε συνδέσει την εντολή stop jumping στο σημείο completed του custom event.

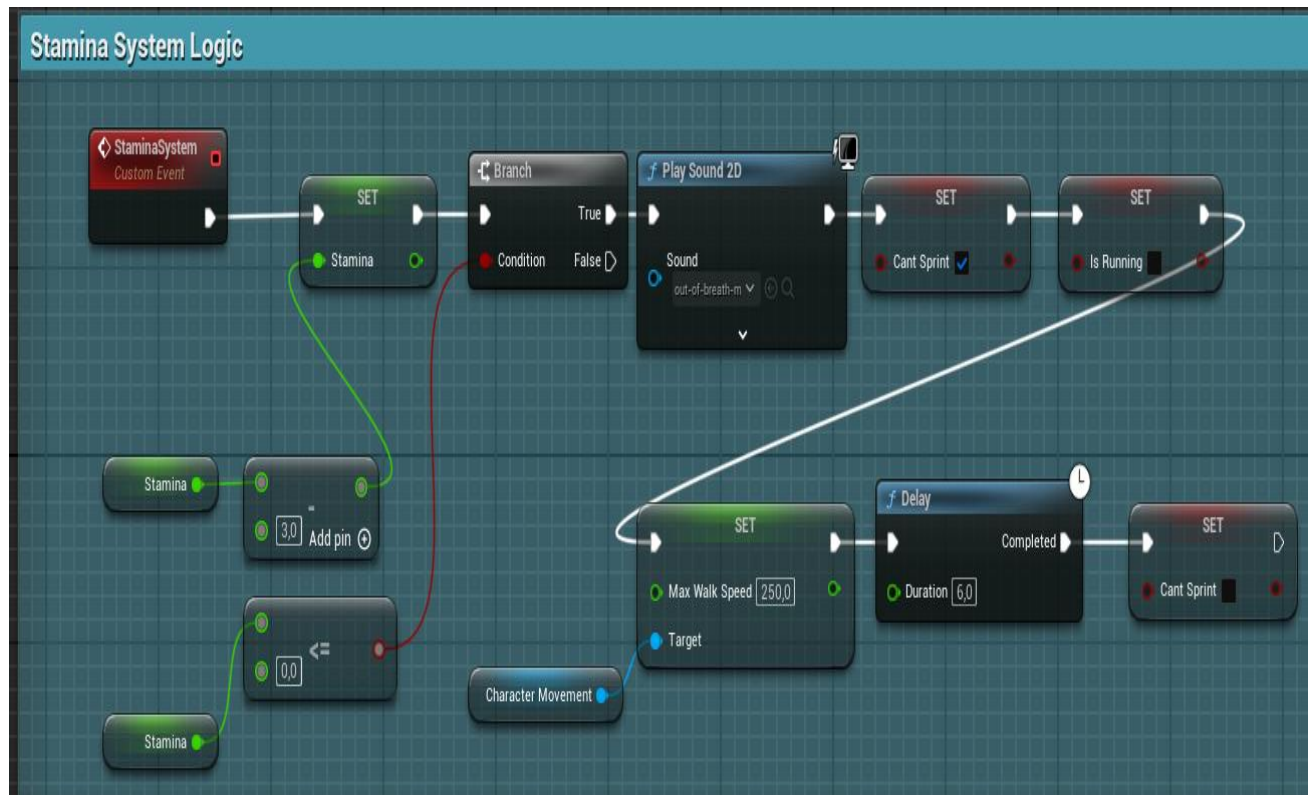


Εικόνα 3.3 Κώδικας ώστε να μπορεί ο χαρακτήρας να τρέξει γρήγορα.

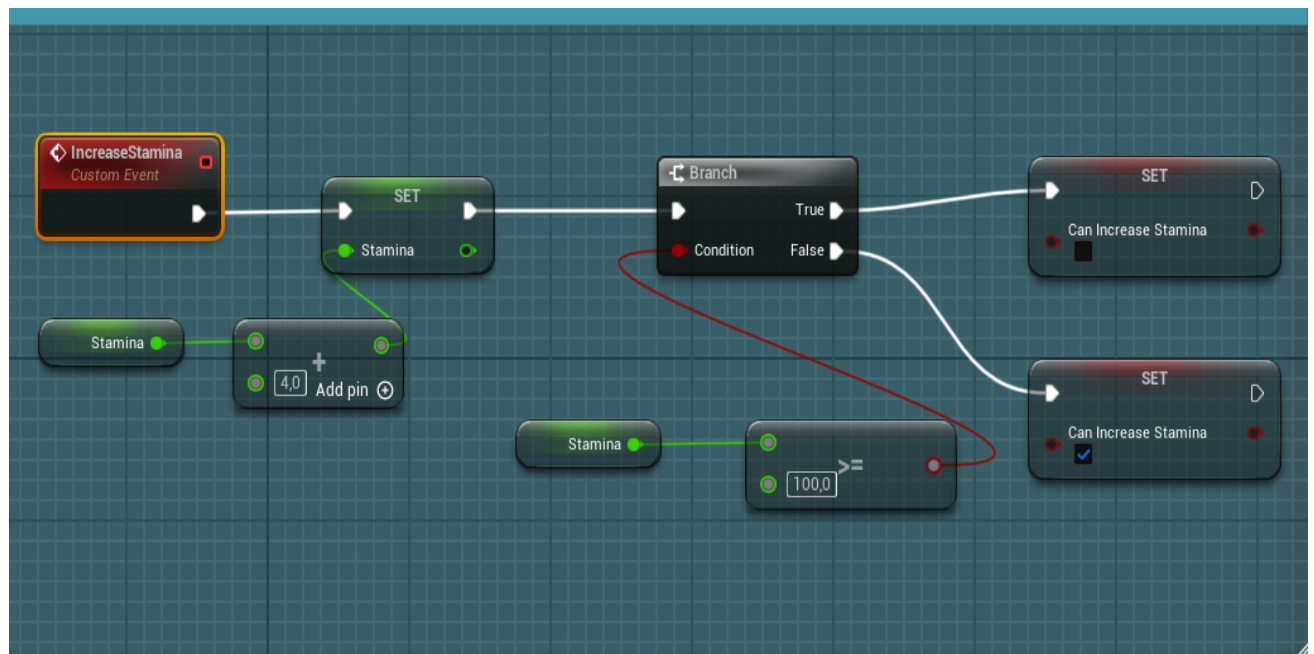
Σε αυτή την περίπτωση ορίζω το κουμπι left shift ως το κουμπι το οποίο θα κάνει trigger το event. Όταν το πατήσει ο παίχτης υπάρχει έλεγχος με μια συνθήκη branch το αν η τιμή του Aim (είναι η μεταβλητή όπου γίνεται αληθής όταν ο παίχτης σημαδεύει με το όπλο) είναι αληθής ή ψευδής. Στην περίπτωση όπου είναι αληθής σημαίνει ότι ο παίχτης σημαδεύει με το όπλο όποτε την ορίζουμε ως ψευδής ώστε να κατεβάσει το όπλο όπως και την μεταβλητή can't sprint όπου είχε οριστεί ως αληθής όταν σημαδεύσε. Αφού όρισα τις τιμές των μεταβλητών στις τιμές όπου ήθελα ώστε να λειτουργήσει σωστά ο τρόπος με τον οποίο ήθελα να τρέχει ο παίχτης κάνω ακόμα έναν έλεγχο για ασφάλεια αν η τιμή του can't sprint είναι αληθής. Αν ναι τότε ο χαρακτήρας δεν θα κάνει τίποτα. Αν όχι τότε ορίζω την τιμή του max walk speed σε 600 από 250 όπου ήταν προκαθορισμένη στην μεταβλητή. Στην συνέχεια υπάρχει ένα μικρό delay 0.2 δευτερολέπτων ώστε να κυλάει ομαλά το animation του τρεξιματος και αμέσως μετά καλείται το stamina system το οποίο θα περιγράψω αμέσως μετά. Τέλος καλείται η συνθήκη branch όπου ελέγχει αν η τιμή του running είναι αληθής δηλαδή ο παίχτης συνεχίζει να τρέχει ώστε να επαναλάβει ξανά την διαδικασία αλλιώς να σταματήσει να κινείται.

Στην περίπτωση όπου ο παίχτης σταματήσει να πατάει το left shift τότε η τιμή του is running γίνεται ψευδής και η τιμή του Max walk speed επιστρέφει στα 250. Ξανά υπάρχει ένα μικρό delay ώστε να λειτουργεί ομαλά ο κώδικας και αμέσως μετά καλείται η Increase stamina. Τέλος καλούμε την συνθήκη για το αν το stamina του παίχτη μπορεί να αυξηθεί ή

έχει φτάσει στο 100 όπου είναι η μέγιστη τιμή. Αν δεν έχει φτάσει στο 100 τότε υπάρχει άλλος ένας έλεγχος για το αν ο παίχτης τρέχει ακόμα και αν είναι ψευδής τότε η διαδικασία επαναλαμβάνεται.



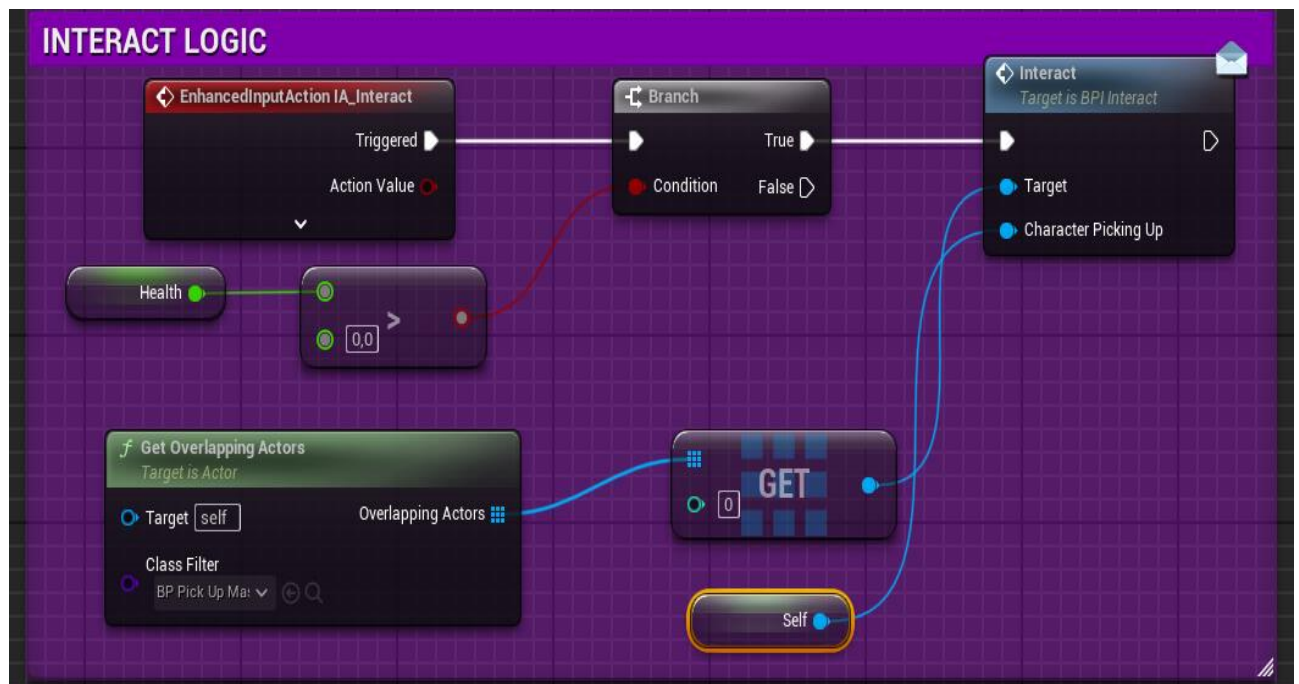
Εικόνα 3.4



Εικόνα 3.5 Κώδικες για το stamina system και το increase stamina.

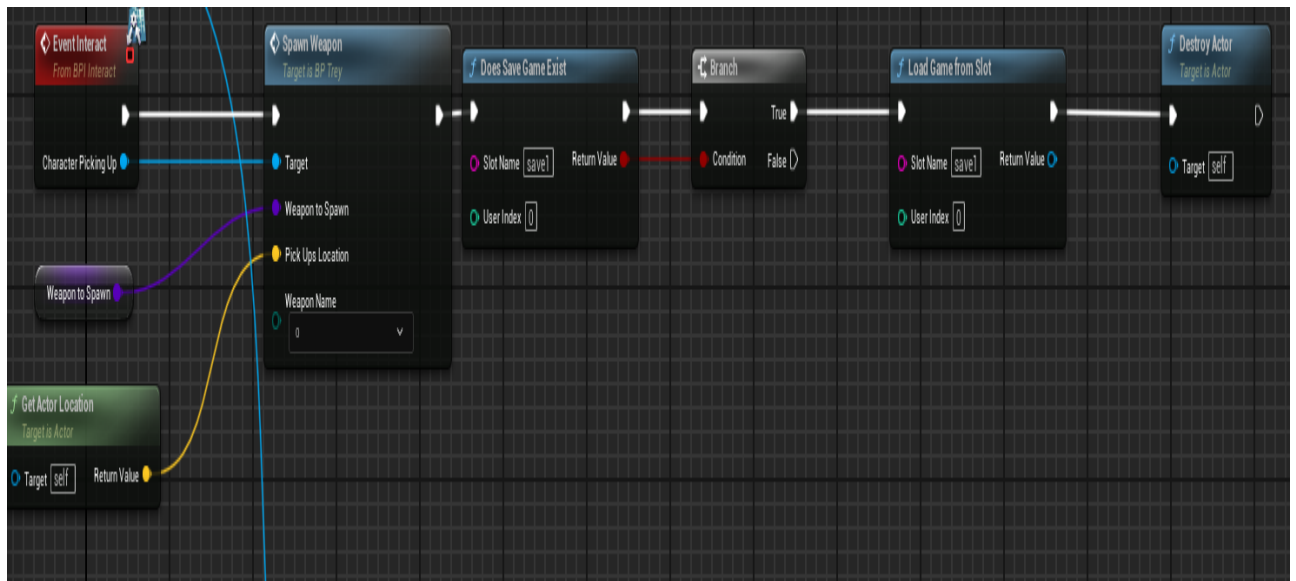
Όπως ανέφερα παραπάνω τα custom events Stamina system και το increase stamina καλούνται μέσω του κώδικα Sprint system. Το stamina system είναι ένα custom event το οποίο με το που καλείται μειώνει την τιμή όπου έχει το την συγκεκριμένη στιγμή που καλείται κατά 3. Η τιμή του stamina είναι ορισμένη ως 100 αλλά αλλάζει συνεχώς . Στην συνέχεια ελέγχει μέσω της συνθήκης branch αν το stamina είναι μικρότερο ή ίσο του 0. Αν είναι αληθής η συνθήκη τότε μέσω του Play sound 2D ακούγεται στο παιχνίδι ένας ήχος όπου δίνει την εντύπωση πως ο παίχτης κουράστηκε και δεν μπορεί να τρέξει άλλο. Στην συνέχεια αφού το stamina έχει την τιμή 0 ορίζουμε την μεταβλητή can't sprint σε αληθής ώστε να μην μπορεί ο παίχτης να πατήσει το κουμπί Left shift και να τρέξει και το is Running σε false αφού ο παίχτης μπορεί να στέκεται ακίνητος ή να περπατάει όταν το stamina είναι 0. Τέλος ορίζω το Max Walk speed σε 250 ώστε ο παίχτης από εκεί που έτρεχε να ξεκινήσει να περπατάει και βάζω ένα delay 6 δευτερόλεπτων πριν ορίσω το can't sprint ξανά σε false ώστε για αυτά τα δευτερόλεπτα να μην μπορεί ο παίχτης να ξανά τρέξει.

Όταν το stamina φτάσει 0 και γίνει η παραπάνω διαδικασία ή ο παίχτης σταματήσει να τρέχει από επιλογή του τότε καλείται το custom event increase stamina. Σε αυτή την περίπτωση αυξάνεται κατά 4 η τιμή που έχει το stamina την δεδομένη στιγμή και ελέγχεται αν η τιμή της είναι ίση ή μεγαλύτερη του 100. Αν ναι τότε το can increase γίνεται false ώστε να σταματήσει η επαναληπτική διαδικασία αύξησης του stamina αλλιώς γίνεται true και επαναλαμβάνεται μέχρι το stamina να γίνει 100 ή όχι παίχτης να ξανά πατήσει το Left shift.



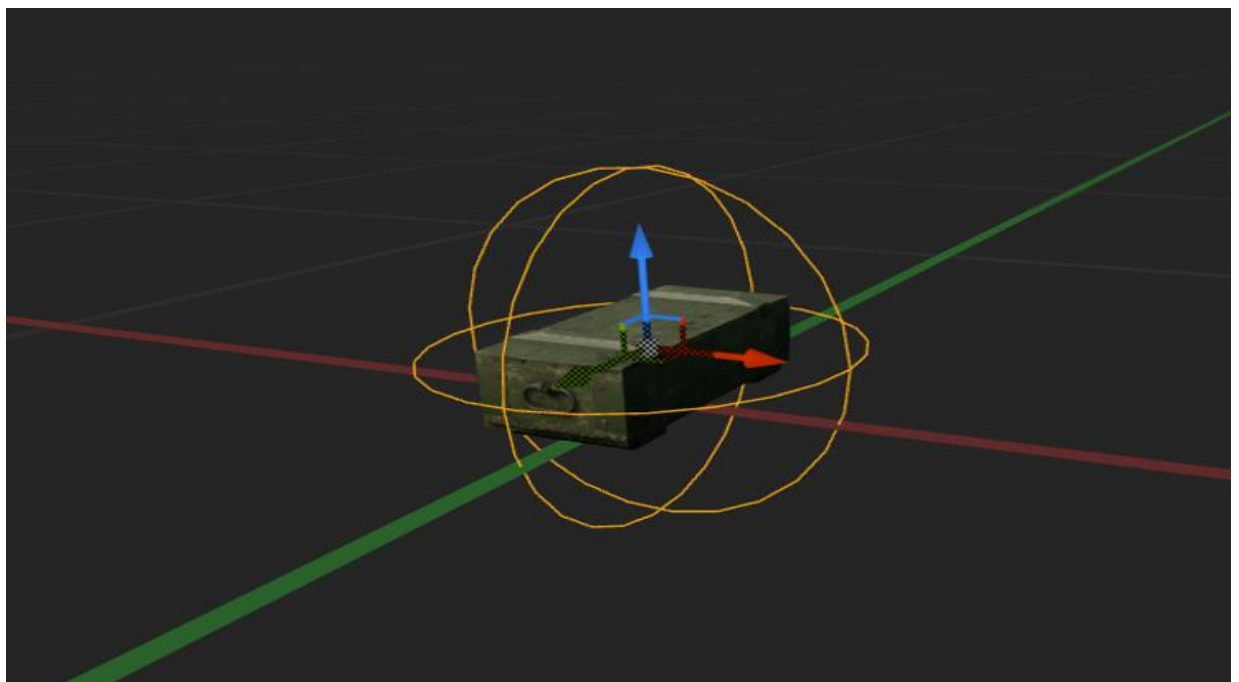
Εικόνα 3.6 Κώδικας για το πώς ο παίκτης κάνει interact με αντικείμενα.

Ο παίκτης μέσα στο παιχνίδι έχει την δυνατότητα να μπορεί να έχει αλληλεπίδραση με αντικείμενα. Αυτό επιτυγχάνεται με την χρήση του Input action. Όταν καλείται ελέγχει αν ο χαρακτήρας έχει ζωή πάνω από 0 δηλαδή να είναι ζωντανός και στην συνέχεια καλεί το interact εισάγοντας σε αυτό τις μεταβλητές self όπου είναι το actor του παίχτη και ως target για το interact το blueprint Pick Up master. Το self χρειάζεται ως μετά ληστή ώστε να μπορεί να εντοπίσει ο κώδικας το παίκτη και να εκτελεστεί ο κώδικας.



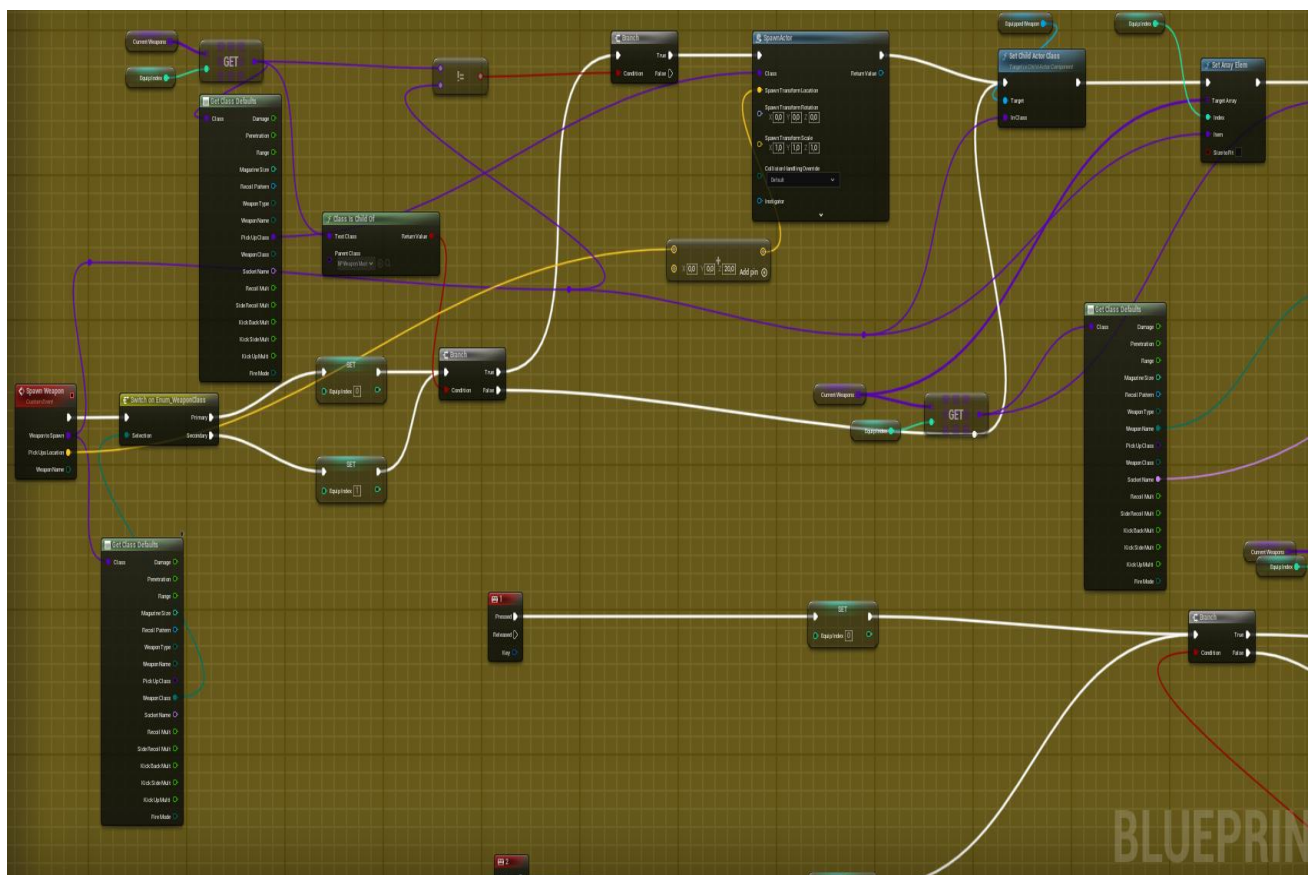
Εικόνα 3.7 Συνέχεια του κώδικα interact εντός του PickUpMaster.

Μέσα στο PickUpMaster καλείται το custom event του spawn weapon εισάγοντας του τις τιμές του όπλου του οποίου αλληλεπιδρά ο παίκτης και την θέση του. Στην συνέχεια αποθηκεύονται αυτά τα δεδομένα ώστε το όπλο να παραμένει στο χέρι του παίχτη και μόλις εκτελεστεί αυτό καταστρέφει το όπλο όπου βρίσκεται στο περιβάλλον αφού πλέον βρίσκεται στα χέρια του παίχτη.

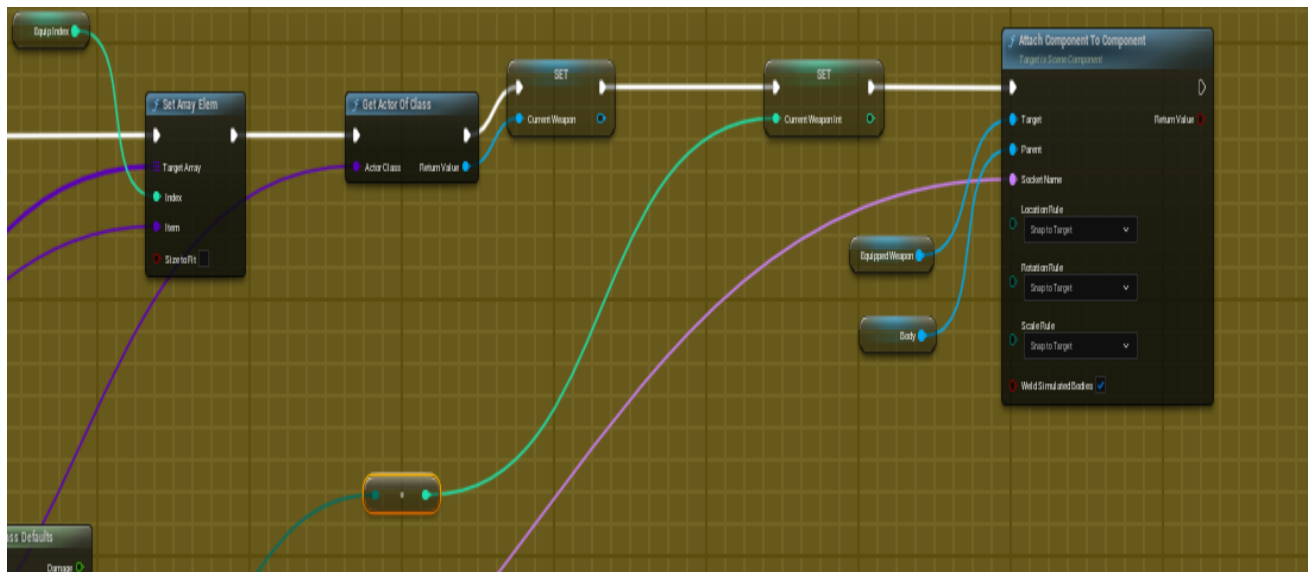


Εικόνα 3.7 Η σφαίρα με την οποία ο κώδικας εντοπίζει αν ο παίχτης βρίσκεται στην ακτίνα αλληλεπίδρασης.

Όταν ο παίχτης είναι μέσα σε αυτή την σφαίρα τότε καλείται το Interact. Αυτή η σφαίρα είναι απαραίτητη ώστε να υπάρξει περιορισμός στην εμβέλεια όπου ο παίχτης μπορεί να αλληλεπιδράσει με αντικείμενα και να μην μπορεί να το κάνει από κάθε σημείο στον χάρτη. Εντός της υπάρχει η συνθήκη για να γίνει κλήση του interact αν ο παίχτης είναι εντός εμβέλειας και κρατήσει πατημένο το κουμπί F που έχω ορίσει.



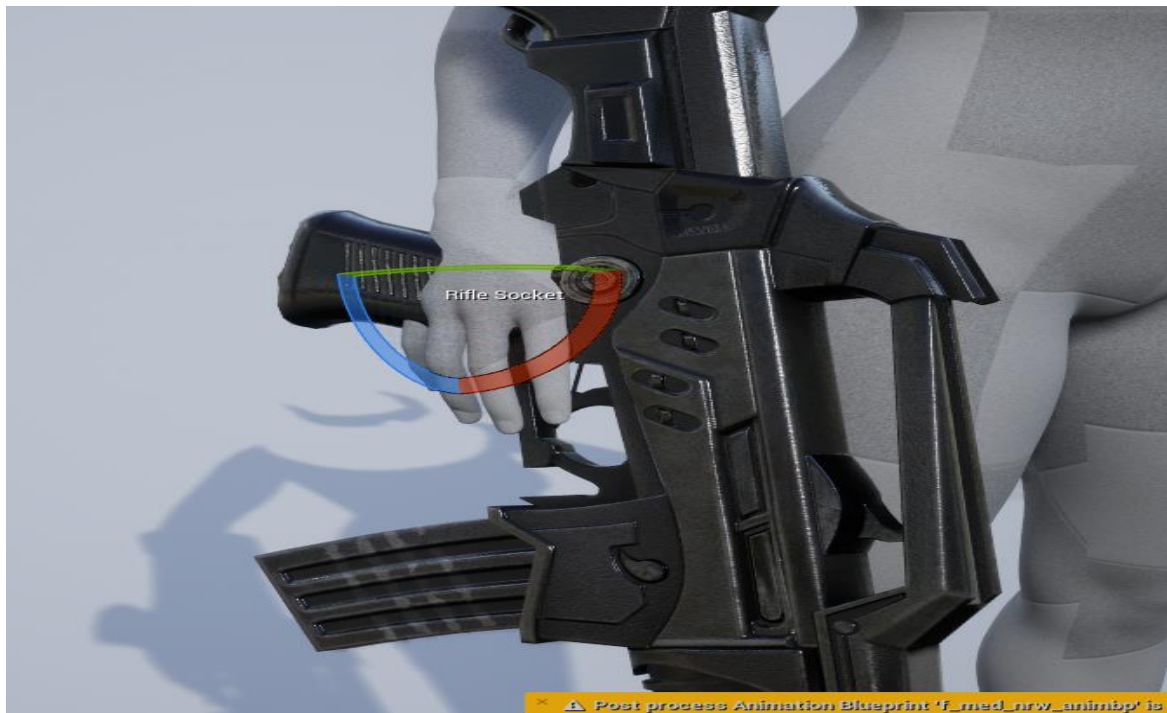
Εικόνα 3.8



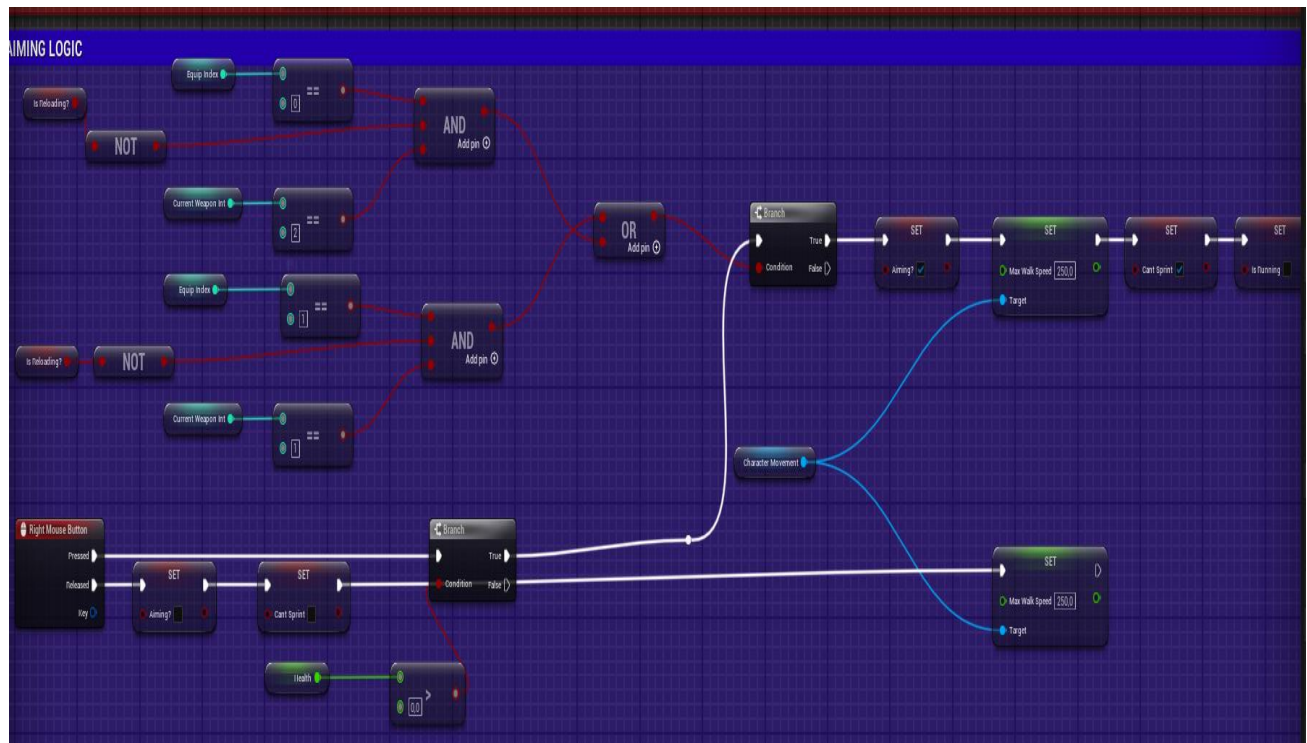
Εικόνα 3.9 Κώδικας Spawn Weapon.

Με την κλήση του custom event Spawn weapon εκτελείται μια εντολή switch ανάμεσα σε δυο κατηγορίες που έχω ορίσει σε ένα Enumerator weapon class. Αυτές οι δυο κατηγορίες ονομάζονται primary και secondary οι οποίες χρησιμοποιούνται για το πολυβόλο και το πιστόλι αντίστοιχα. Αυτό γίνεται ώστε να είναι πιο ξεκάθαρος ο κώδικας και να μην υπάρχουν επιπλοκές στις αλλαγές των όπλων. Το κάθε όπλο έχει το τίτλο primary αν είναι πολυβόλο και secondary αν είναι πιστόλι ώστε να περνάει τα δεδομένα στο spawn weapon. Στην συνέχεια ανάλογα σε ποιά από τις δυο κατηγορίες ανήκει το όπλο εκχωρούμε στο Equip index (το οποίο είναι η μεταβλητή για την οποία γνωρίζει ο κώδικας σε ποια θέση να κάνει equip το όπλο) την τιμή 0 αν είναι primary και 1 αν είναι secondary. Στην συνέχεια υπάρχει μια συνθήκη που ελέγχει αν το όπλο είναι παιδί της Weapon Master και αν ναι τότε περνάει σε μια ακόμα συνθήκη που ελέγχει αν το equip index και το όπλο το οποίο κρατάει ήδη ο παίχτης στα χέρια του είναι διαφορετικά από αυτό όπου αλληλεπιδρά και αν είναι αληθής τότε εμφανίζει το όπλο στα χέρια του μέσω του spawn actor. Αμέσως μετά ορίζει το equipped weapon που είναι το νούμερο του όπλου όπου σήκωσε ο παίχτης ως παιδί της κλάσης weapon to spawn ώστε να πάρει τα δεδομένα από το συγκεκριμένο όπλο. Αυτά τα δεδομένα χρησιμοποιούνται αμέσως και είναι το equip index του όπλου όπου αποθηκεύεται στον πίνακα το όπλων που έχει σηκώσει ο παίκτης, το current weapon που είναι το όπλο το οποίο κρατάει στην συγκεκριμένη περίπτωση αυτό που μόλις σήκωσε διότι το κάνει αμέσως equip στον παίκτη οπότε αλλάζει από το προηγούμενο που είχε στην κατοχή του και το current weapon int όπου είναι ο αριθμός του όπλου που κρατάει στα χέρια του την δεδομένη στιγμή. Τέλος καλεί το function attach

component to component ώστε να τοποθετήσει το equipped weapon στο body του παίκτη ,πιο συγκεκριμένα να τοποθετήσει το όπλο στα χέρια του στο σημείο του socket που έχω ορίσει με το όνομα του εκάστοτε όπλου. Για παράδειγμα αν το όπλο που προσπαθεί να σηκώσει ο παίκτης είναι πιστόλι το socket name είναι pistol socket και τα χέρια του παίχτη τα έχω προσαρμόσει ώστε να κρατάνε σωστά το κάθε όπλο.

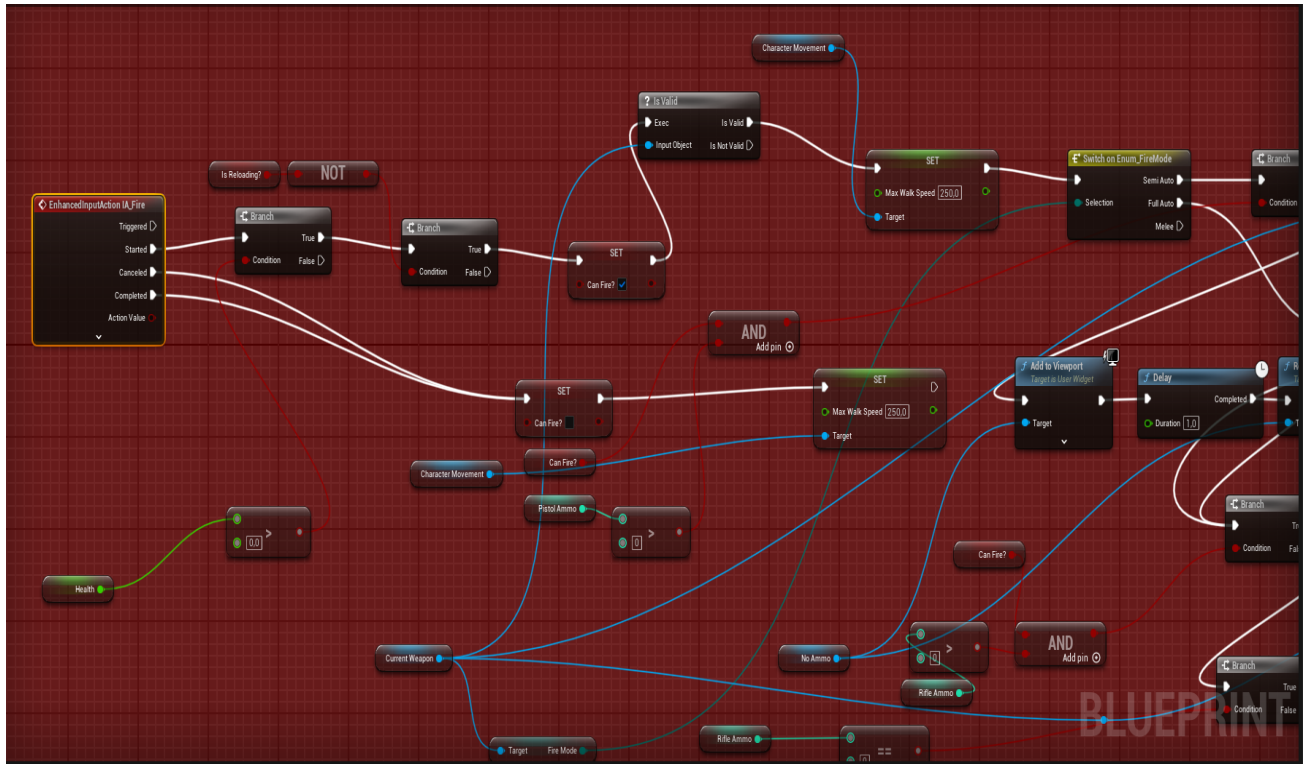


Εικόνα 3.10 Παράδειγμα του Socket.

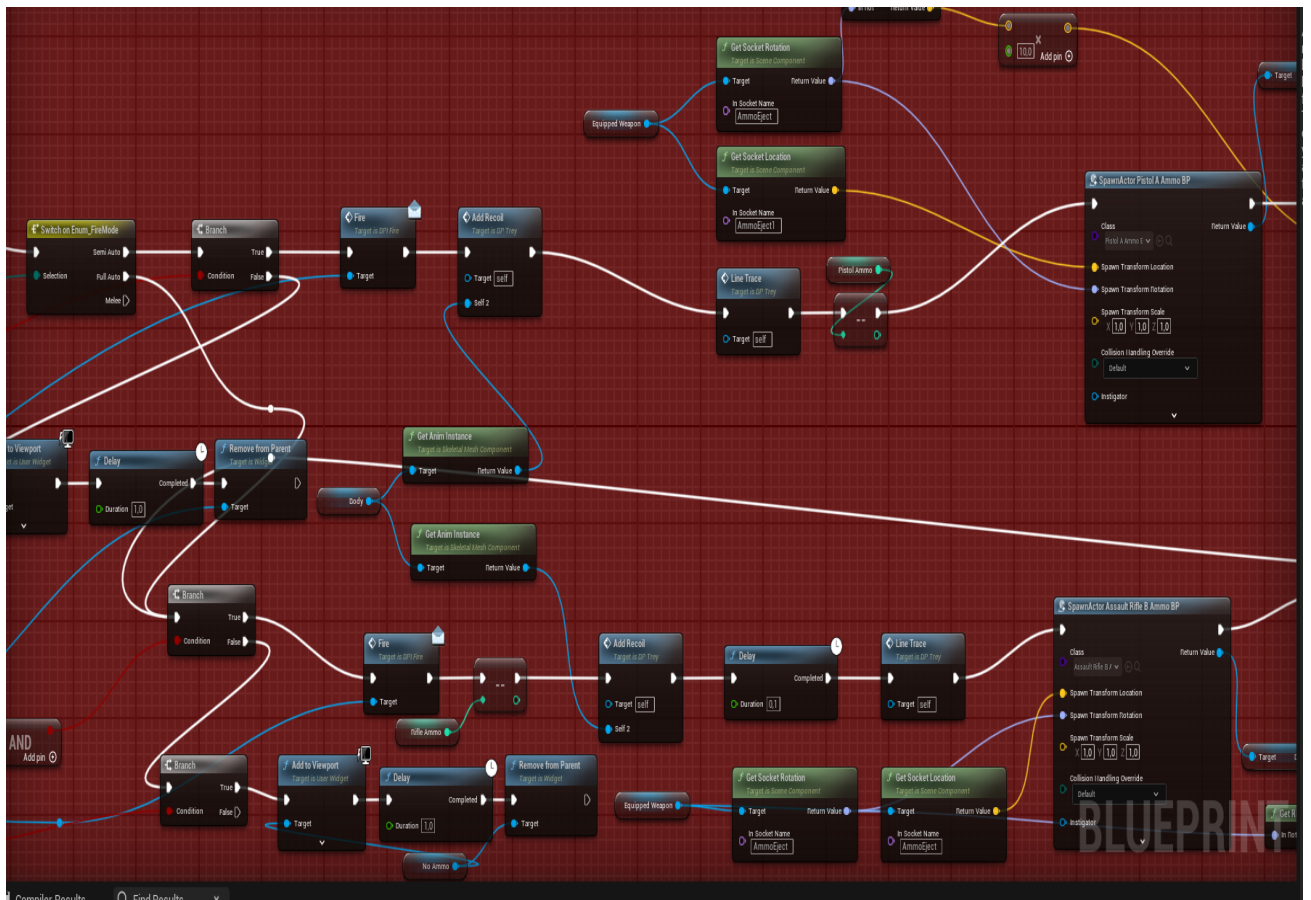


Εικόνα 3.11 Κώδικας για τον τρόπο που σημαδεύει ο παίχτης.

Όταν ο παίχτης πατήσει το Right Mouse Button τότε καλείται η συνθήκη που ελέγχει αν ο χαρακτήρας είναι ζωντανός και εφόσον είναι αληθής τότε μπαίνει σε άλλη μια πολλαπλή συνθήκη. Σε αυτή την συνθήκη ελέγχεται ταυτόχρονα αν το Equip index είναι 0 , ο παίχτης δεν γεμίζει το όπλο και αν το όπλο που κρατάει είναι πολυβόλο ή αν το equip index είναι 1, παίχτης δεν γεμίζει το όπλο και το οποίο που κρατάει είναι πιστόλι. Αν μια από τις δυο περιπτώσεις ισχύει τότε η συνθήκη είναι αληθής. Τότε η τιμή του Aiming? γίνεται αληθής , το Max Walk Speed γίνεται 250, το Can't sprint γίνεται αληθής και το is Running false έτσι ώστε αν ο χαρακτήρας τρέχει και πατήσει να σημαδέψει να τον κάνει να περπατάει. Τέλος αν ο παίχτης αφήσει το Right Mouse Button τότε το Aiming? γίνεται false όπως και το Can't sprint και το Max walk speed ορίζεται σε 250 ώστε ο χαρακτήρας να μπορεί να ξανά κινηθεί.



Εικόνα 3.12



Εικόνα 3.13



Εικόνα 3.14 Κώδικας με τον οποίο πυροβολεί ο παίχτης.

Για τον κώδικα με τον οποίο πυροβολεί το όπλο χρησιμοποιώ ένα Input Action που ονομάζεται IA_Fire. Αυτό το input action εκτελείται όταν ο παίχτης πατάει το left click. Με την εκτέλεση της αρχικά ελέγχει την συνθήκη ότι ο χαρακτήρας είναι ζωντανός. Αν είναι αληθής τότε ελέγχει αν γεμίζει το όπλο. Αν και πάλι είναι αληθής τότε εκχωρείται στην μεταβλητή can fire η τιμή αληθής. Στην συνέχεια χρησιμοποιώ την εντολή is valid όπου ελέγχει αν υπάρχει όπλο στα χέρια του παίχτη ώστε να μπορεί να πυροβολήσει. Εφόσον είναι αληθής τότε εκχωρείται στην μεταβλητή Max walk speed η τιμή 250 ώστε σε περίπτωση που ο χαρακτήρας τρέχει να σταματήσει μια να συνεχίσει περπατώντας. Επακολουθεί μια εντολή switch on όπου χρησιμοποιώ τον Enumerator που έχω φτιάξει ο οποίος περιέχει τις τιμές semi auto και auto για το πιστόλι και το πολυβόλο αντίστοιχα. Επειδή το κάθε όπλο έχει ορισμένο τον τρόπο με τον οποίο πυροβολεί αν είναι semi auto τότε σημαίνει πως ο χαρακτήρας χρησιμοποιεί πιστόλι και αν είναι auto πολυβόλο. Και στις δυο περιπτώσεις η συνέχεια είναι η ίδια όποτε θα εξετάσουμε την περίπτωση όπου ο χαρακτήρας κρατάει πιστόλι. Εκτελείται άλλη μια συνθήκη που ελέγχει αν το όλο μπορεί να ρίξει και αν έχει σφαίρες. Αν η συνθήκη είναι αληθής τότε καλείται η εντολή fire και η add recoil όπου η πρώτη εμφανίζει μια λάμψη στην άκρη του όπλου και αναπαράγει τον ήχο του πυροβολισμού και η δεύτερη μετακινεί το όπλο και τα χέρια του παίχτη όσο πυροβολεί ώστε να δίνεται η εντύπωση πως το όπλο « κλωτσάει ». Τέλος καλείται το Custom event Line trace και μειώνει της

Στην περίπτωση όπου η συνθήκη ελέγχου για σφαίρες είναι ψευδής τότε εμφανίζεται στην οθόνη το UI «No ammo left» και μετά από 1 δευτερόλεπτο εξαφανίζεται. Ακριβώς η ίδια διαδικασία χρειάζεται και για το πολυβόλο με την μόνη διαφορά να είναι ότι στο τέλος καλείται ξανά η συνθήκη ελέγχου για το αν το όπλο έχει σφαίρες και ο παίχτης ακόμα πυροβολεί ώστε να συνεχίσει η διαδικασία επαναληπτικά. Τέλος αν ο παίχτης αφήσει το left mouse button ή κάποια από τις συνθήκες ελέγχου γίνει ψευδής τότε η μεταβλητή can fire γίνεται ψευδής και η μεταβλητή max walk speed 250.

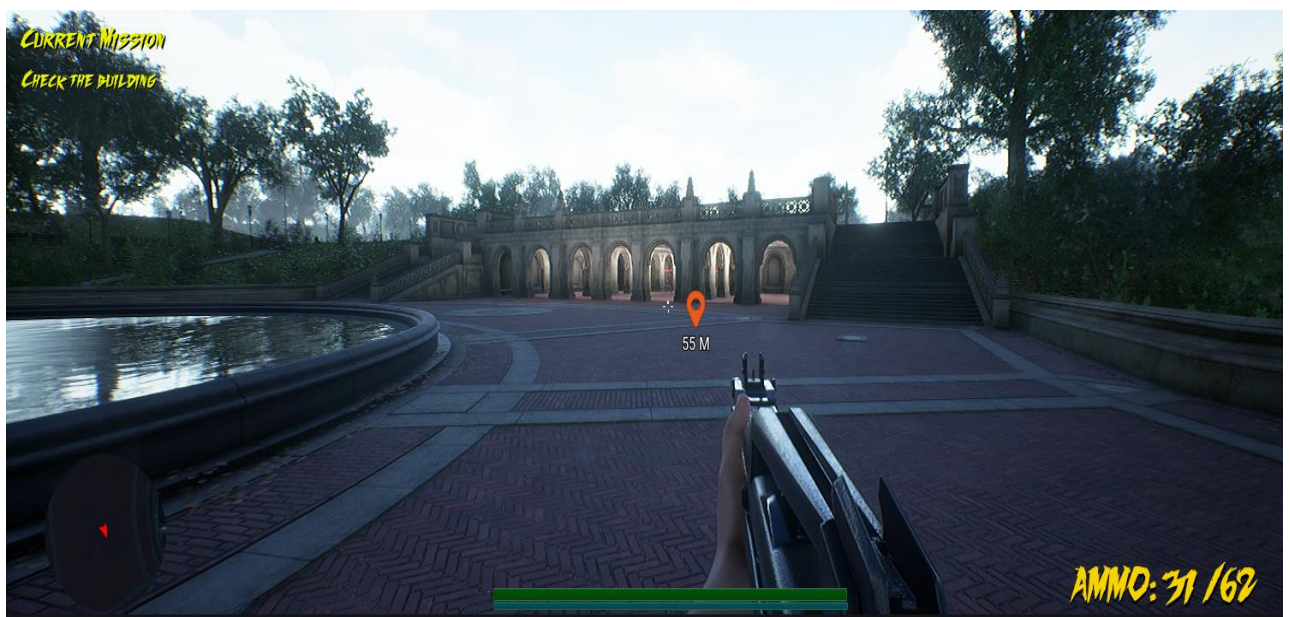




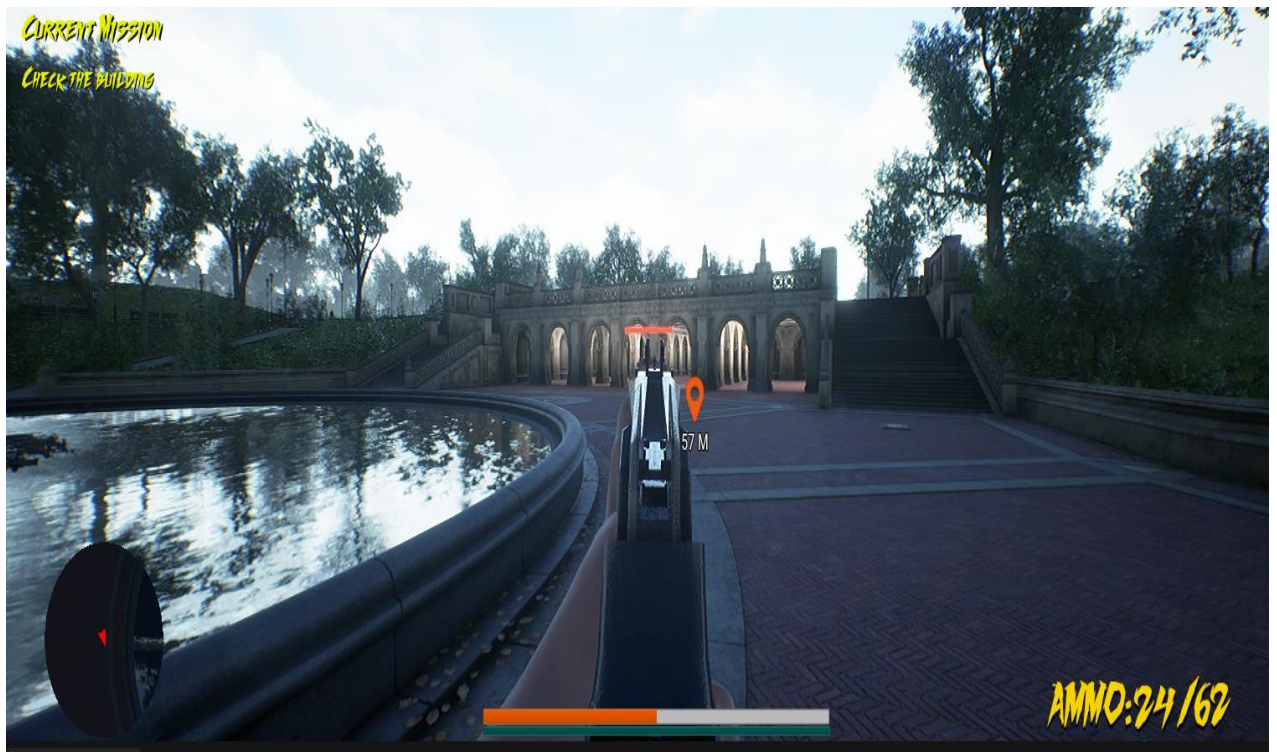
Εικόνα 3.16 Κώδικας Line Tracing.

Ο κώδικας Line tracing καλείται όπως ανέφερα πιο πάνω μέσω του κώδικα Fire. Το Line Trace είναι ένα custom event το οποίο όταν γίνει η κλήση του παράγει μια νοητή γραμμή μέσω της εντολής Line trace by channel με αρχή την κάμερα του παίχτη και τέλος την απόσταση που έχει οριστεί. Η παραγωγή της πρώτης γραμμής είναι ενδεικτική ώστε να γνωρίζουμε σε ποιο σημείο κοιτάζει ο χαρακτήρας. Όποτε ορίζουμε ως αρχή της γραμμής το World location της κάμερας δηλαδή το σημείο όπου βρίσκεται η κάμερα σε σχέση με τον χάρτη του παιχνιδιού και ως τέλος το σημείο το οποίο κοιτάζει ο παίχτης πολλαπλασιασμένο με το 15000 όπου είναι η τιμή που έχω ορίσει για την απόσταση όπου θέλω να φτάνουν οι σφαίρες. Στην συνέχεια υπάρχει μια συνθήκη branch όπου ελέγχει αν ο χαρακτήρας σημαδεύει ή όχι. Και στις δυο περιπτώσεις η διαδικασία είναι η ίδια με την μόνη διαφορά ότι αλλάζει το σημείο όπου ξεκινάει η νοητή γραμμή. Δηλαδή στην περίπτωση όπου ο παίχτης δεν σημαδεύει και έχει το όπλο χαμηλά ξεκινάει να παράγεται από το σημείο όπου βρίσκεται η κάνη του όπλου εκείνη την στιγμή και αντίστοιχα όταν το έχει ψηλά για να σημαδέψει η νοητή γραμμή παράγεται από την κάνη του όπλου στο σημείο όπου βρίσκεται τότε. Επιπλέον στην περίπτωση όπου ο παίχτης σημαδεύει η διασπορά του όπλου είναι μικρότερη από εκείνη όταν έχει το όπλο στην μέση του διότι πρέπει να υπάρχει μεγαλύτερη ακρίβεια. Αυτό επιτυγχάνεται ξανά με την χρήση της εντολής Line trace by channel αλλά αυτή την φορά έχοντας ως αρχή το socket όπου έχω βάλει στην κάνη του όπλου και ως τέλος την ευθεία της

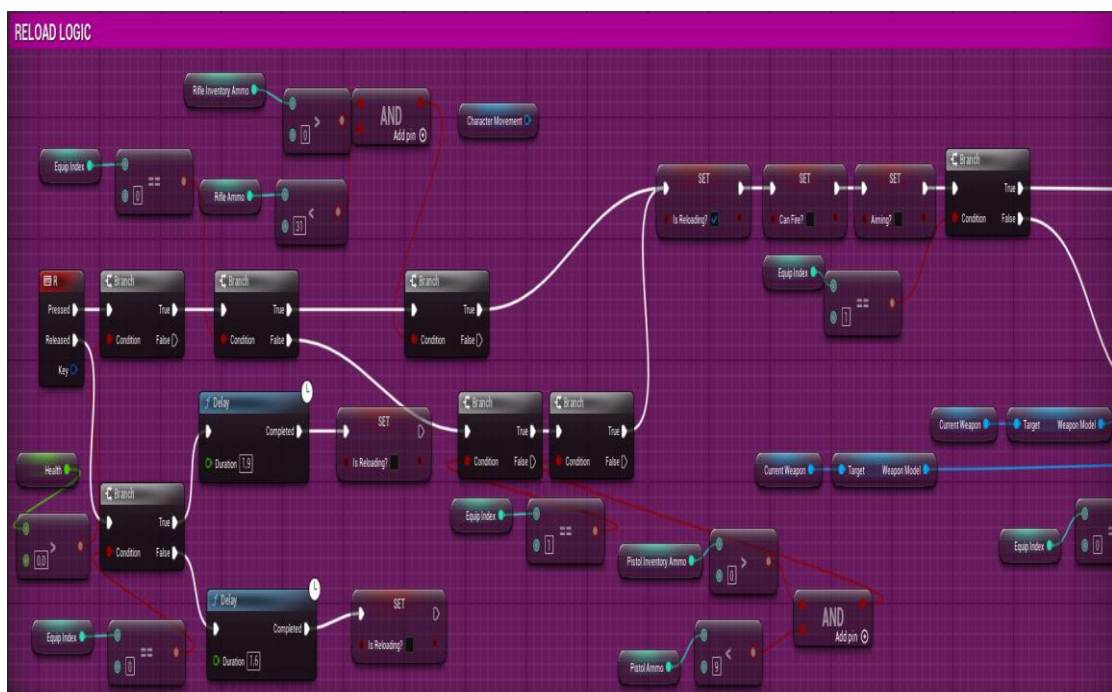
κάμερας πολλαπλασιασμένη με το 15000 αθροίζοντας μαζί και τρεις random τιμές για τους άξονες X Y Z. Αυτές οι τιμές αφορούν την διασπορά του όπλου δηλαδή όσο μεγαλύτερες οι τιμές τόσο απόσταση απέχει η κάθε νοητή γραμμή από την άλλη και αντίστοιχα και οι σφαίρες. Τέλος υπάρχει μια συνθήκη ελέγχου ώστε να γνωρίζουμε αν οι γραμμές ήρθαν σε επαφή με κάποιο αντικείμενο μέχρι να φτάσουν στο τέλος της απόστασης όπου έχουν οριστεί. Στην περίπτωση όπου είναι αληθής σημαίνει πως βρήκαν κάποιο εμπόδιο και καλείται μια εντολή switch on αλλά αυτή την φορά για physical surface. Ως physical face έχω ορίσει τα zombie όπου βρίσκονται στον χάρτη και έχω δώσει στο κεφάλι τους το όνομα headshot και στα υπόλοιπα μέρη του σώματος το όνομα body shot. Όποτε αν η γραμμή έρθει σε επαφή με αντικείμενο που έχει το όνομα headshot καλείται η εντολή apply damage όπου κάνει ζημιά στον αντίπαλο χαρακτήρα ανάλογα με την τιμή που του έχουμε ορίσει. Στην συγκεκριμένη περίπτωση το damage που θα κάνει στον αντίπαλο θα είναι τριπλάσιο αν τον πετύχει στο κεφάλι αλλιώς θα είναι η κανονική τιμή που έχει οριστεί δηλαδή η τιμή 20.



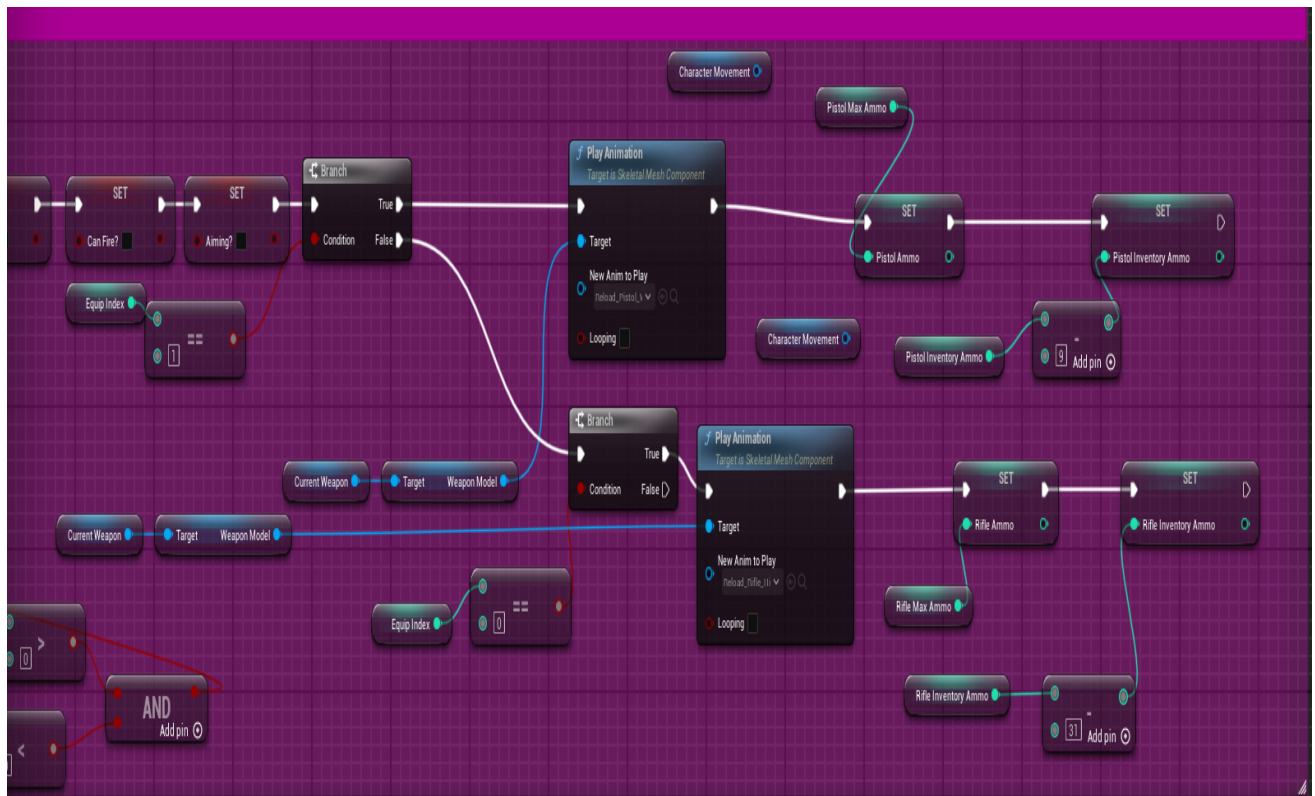
Εικόνα 3.17



Εικόνα 3.18 Εικόνες από το παιχνίδι με το όπλο να είναι στην μέση του παίχτη και ο παίχτης να σημαδεύει.



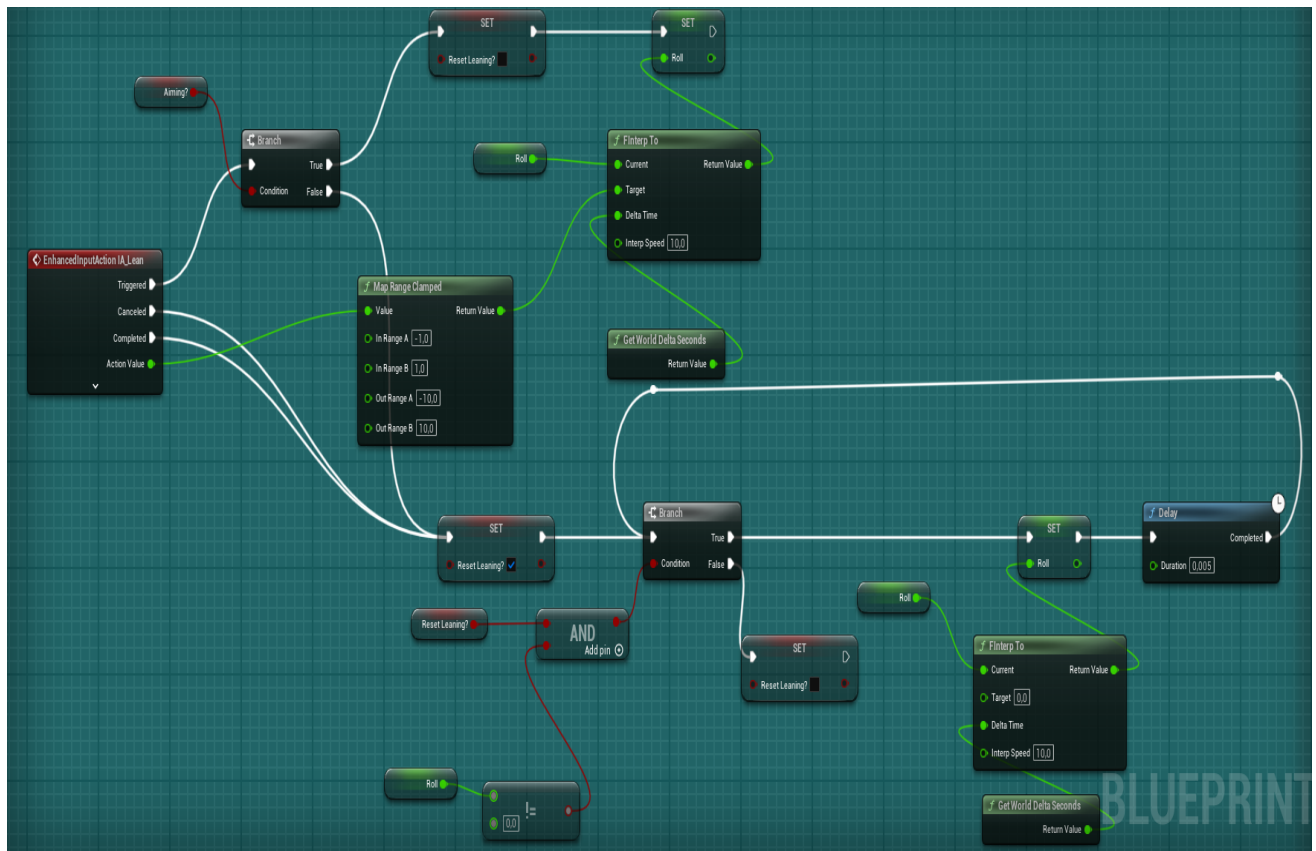
Εικόνα 3.19



Εικόνα 3.20 Κώδικας για τον τρόπο γεμίσματος του όπλου.

Ο κώδικας reload καλείται όταν ο παίχτης πατήσει το κουμπί R. Αρχικά υπάρχει μια συνθήκη όπου ελέγχει αν ο παίχτης είναι ζωντανός. Στην περίπτωση που είναι αληθής τότε υπάρχει μια δεύτερη συνθήκη αν το όπλο που κρατάει στο χέρι του είναι το πολυβόλο. Αν ναι τότε περνάει σε μια τρίτη συνθήκη που ελέγχει αν υπάρχουν σφαίρες στην κατοχή του παίχτη ώστε να μπορεί να γεμίσει το όπλο. Στην περίπτωση που είναι ψευδής ελέγχεται αν ο παίχτης κρατάει πιστόλι και αν είναι αληθής επαναλαμβάνεται η ίδια διαδικασία ελέγχου για το αν υπάρχουν σφαίρες στην κατοχή του. Και στις δυο περιπτώσεις όταν γίνουν οι συγκεκριμένοι έλεγχοι εκχωρείται στην μεταβλητή reloading η τιμή αληθής, στην can fire η τιμή ψευδής όπως και στην aiming. Αυτό σημαίνει ότι ο χαρακτήρας περνάει στο στάδιο γέμισμα τις του όπλου και δεν θα πρέπει να μπορεί να σημαδέψει η να πυροβολήσει όσο γίνεται αυτή η διαδικασία. Στην συνέχεια επειδή οι προηγούμενες συνθήκες συγχωνεύτηκαν ώστε να εκτελεστούν οι κοινές εντολές ξανά υπάρχει έλεγχος για το αν κρατάει πιστόλι ο παίχτης. Αν ναι τότε παίζει το animation γεμίσματος του πιστολιού με την εντολή play animation αλλιώς παίζει το animation για το πολυβόλο. Τέλος σε κάθε περίπτωση εκχωρείται στην μεταβλητή που περιέχει τον αριθμό των σφαιρών του γεμιστήρα η τιμή των σφαιρών που χωράει το όπλο και

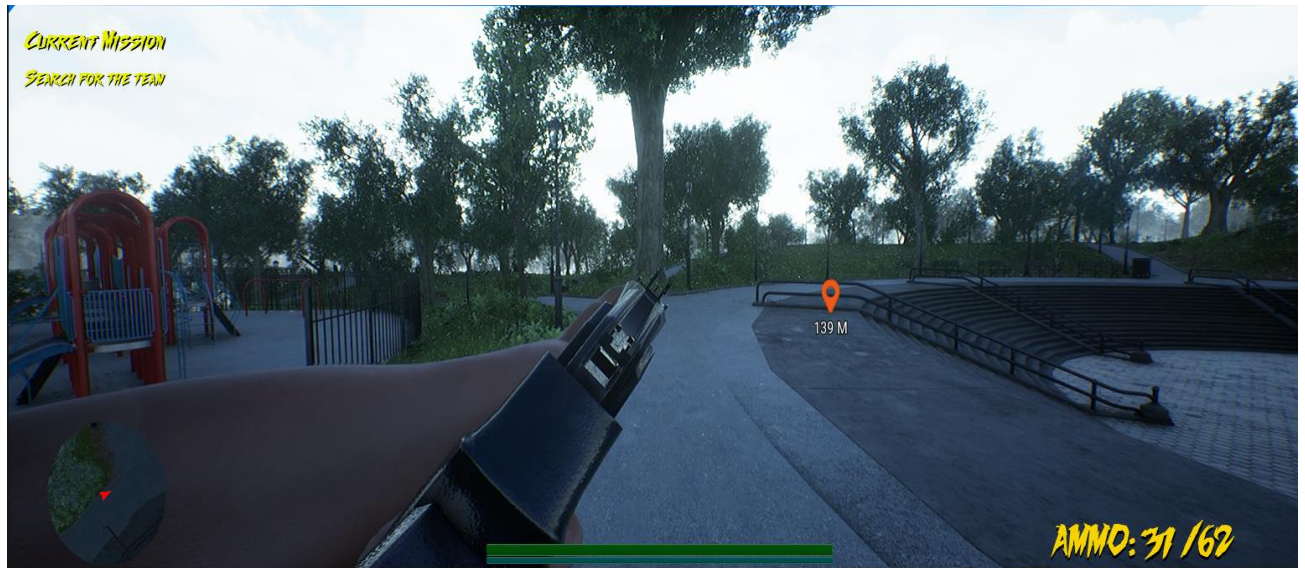
μειώνεται η τιμή της αποθήκης των σφαιρών του παίχτη κατά τον αριθμό των σφαιρών όπου μπηγαν στο όπλο.



Εικόνα 3.21 Κώδικας για την διαδικασία κλίσης του χαρακτήρα ενώ σημαδεύει.

Για τον κώδικα leaning χρησιμοποιώ ένα input action που ονομάζεται IA_lean. Αυτό το input action καλείται όταν ο παίχτης πατήσει τα κουμπιά Q ή E. Όταν γίνει trigger τότε ελέγχεται η συνθήκη για το αν ο παίχτης σημαδεύει. Αν ναι τότε η μεταβλητή Reset leaning γίνεται false ώστε ο παίχτης να μπορεί να πάρει κλίση στην μεταβλητή roll εισάγουμε το αποτέλεσμα της εντολής FInterp To. Αυτή η εντολή παίρνει τις τιμές των roll και Map range clamped όπου έχει τις τιμές για το πόσο μεγάλη κλίση θέλω να πάρει ο παίχτης με το όπλο και τα world data seconds ώστε να κάνει την μετατροπή σε πραγματικό χρόνο. Με την συγκεκριμένη εντολή δηλαδή ο παίχτης θα μπορεί να στρίψει 10 μοίρες δεξιά ή 10 μοίρες αριστερά αν σημαδεύει με το όπλο και πατηθούν ένα εκ των κουμπιών Q (για αριστερά) και E (για δεξιά). Στην περίπτωση όπου η συνθήκη για το αν ο παίχτης σημαδεύει είναι ψευδής ή αν ο παίχτης αφήσει το κουμπι τότε η

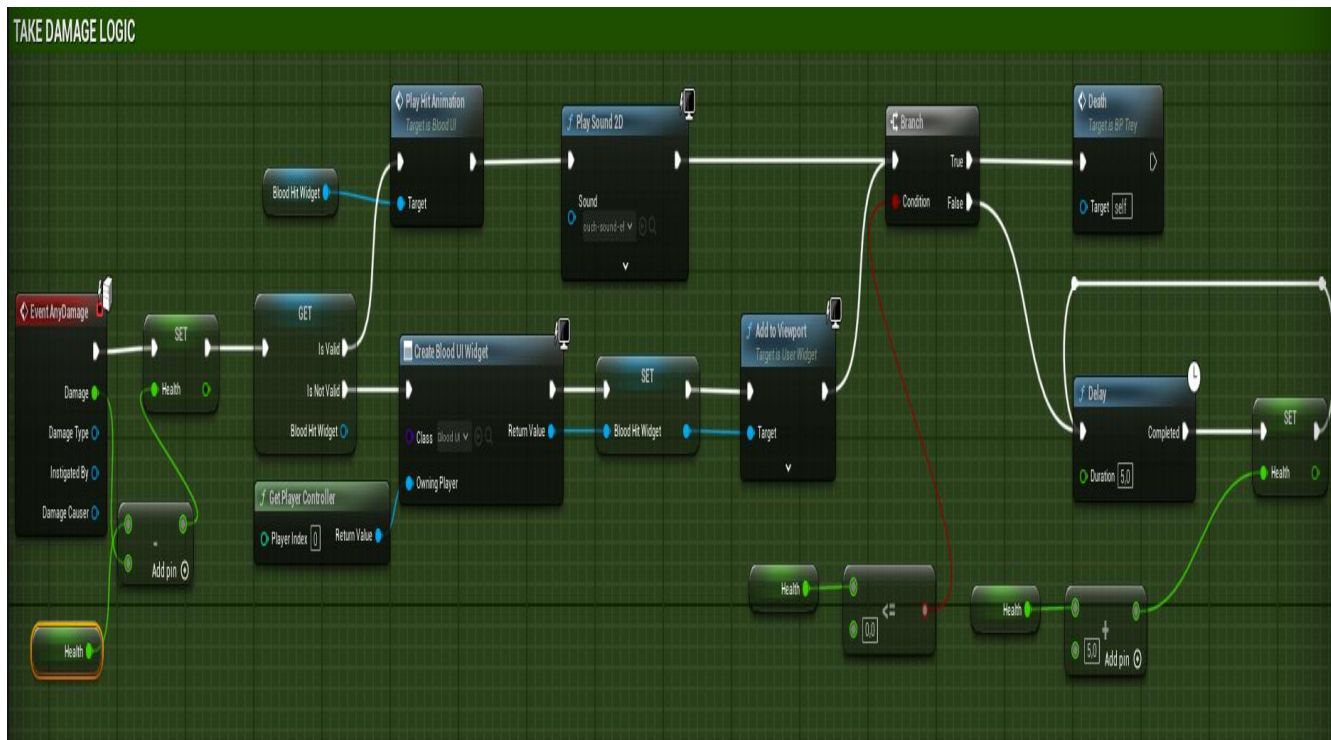
μεταβλητή reset leaning γίνεται true και ελέγχεται η συνθήκη για το αν το reset leaning είναι true και το roll είναι διάφορο του 0. Αν ισχύει τότε η τιμή του roll μέσω του Flbterp Το γίνεται 0 και μετά από ένα μικρό delay ώστε να γίνει ομαλά η μετατροπή ελέγχεται ξανά η συνθήκη ώστε να είναι σίγουρο πως ο παίχτης έχει επιστρέψει στην αρχική του θέση.



Εικόνα 3.22



Εικόνα 3.23 Εικόνες από το παιχνίδι με τον παίχτη να βρίσκεται σε κλίση.

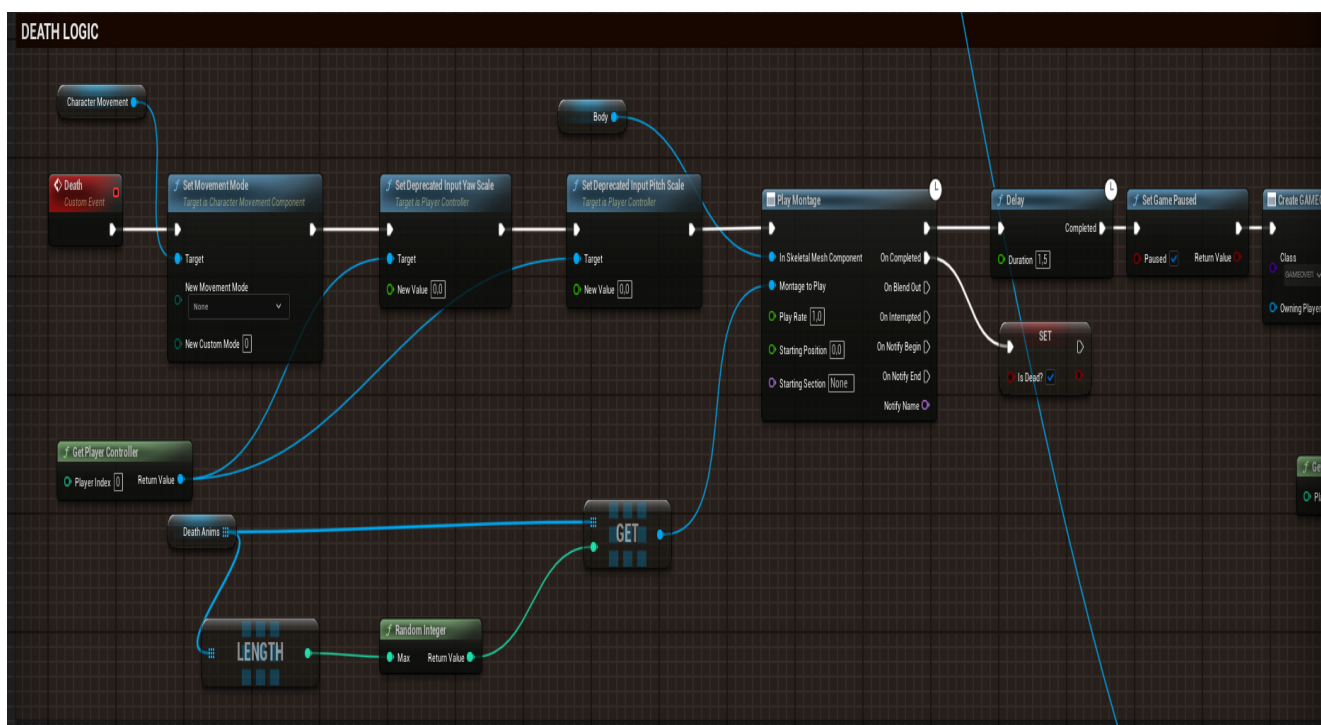


Εικόνα 3.24 Κώδικας για τον τρόπο όπου δέχεται ζημιά ο χαρακτήρας.

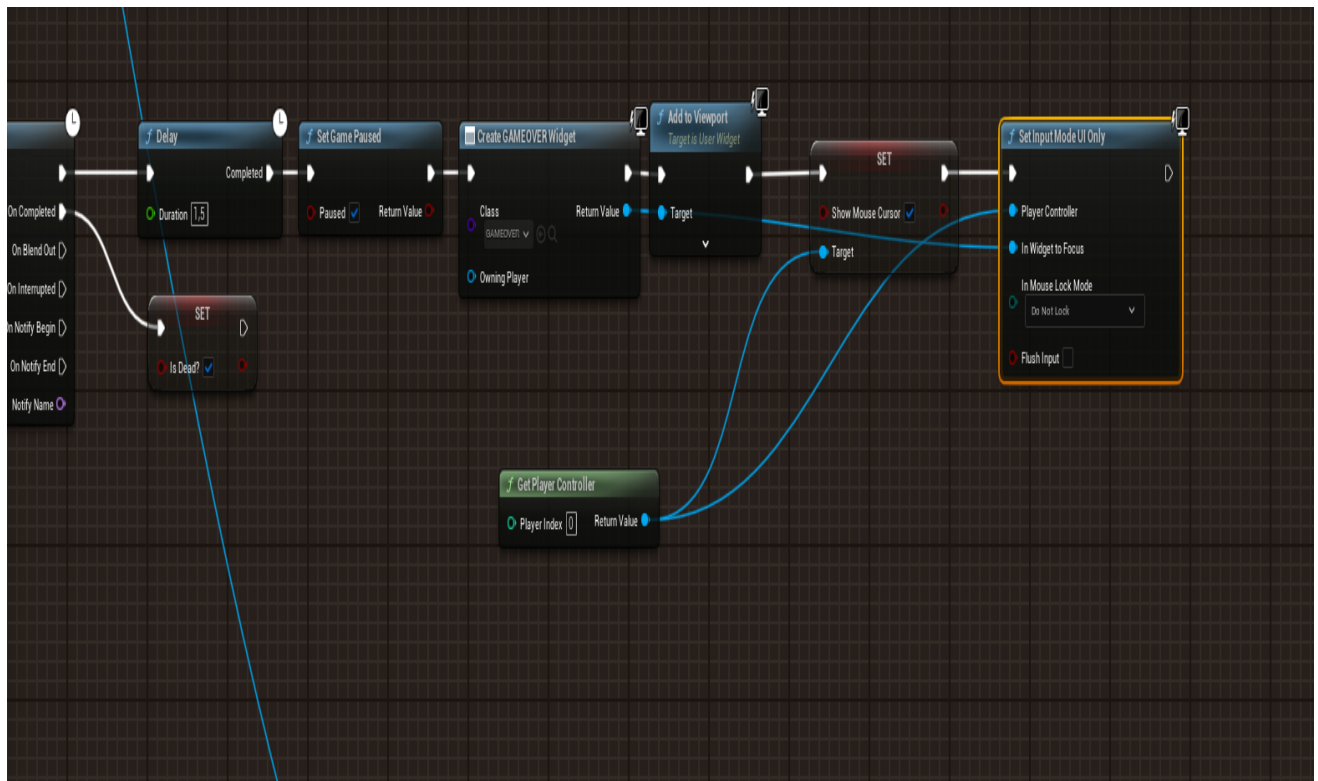
Για τον τρόπο με τον οποίο ο χαρακτήρας δέχεται ζημιά γίνεται η χρήση του custom event Any damage. Η κλήση του γίνεται μέσω μιας εντολής apply damage όπου βρίσκεται μέσα στο blueprint του εχθρού όπου θα δούμε στην συνέχεια. Όταν καλεστεί αρχικά αφαιρεί από την ζωή που έχει εκείνη την στιγμή ο χαρακτήρας το damage όπου ο αντίπαλος έκανε στον παίχτη και ανανεώνει την ζωή που έχει εκείνη την στιγμή ο παίχτης. Στην συνέχεια χρησιμοποιώ μια εντολή get όπου ελέγχει αν ο παίχτης έχει ξανά χτυπηθεί. Στην περίπτωση όπου δεν έχει χτυπηθεί ξανά τότε δημιουργεί ένα widget όπου είναι η εικόνα του αίματος που φαίνεται στην οθόνη όταν ο παίχτης δέχεται ζημιά. Στην συνέχεια αν ξανά δεχτεί ζημιά τότε εμφανίζεται η εικόνα όπου έφτιαξε και παίζει έναν ήχο χτυπήματος. Τέλος με την συνθήκη branch ελέγχει αν ο παίχτης είναι ζωντανός στην περίπτωση που είναι αρχίζει μια επαναλήψεις όπου ανά 5 δευτερόλεπτα αυξάνει την ζωή του παίχτη κατά 5, αν ο παίχτης δεν είναι ζωντανός τότε καλείται ο κώδικας death.



Εικόνα 3.25 Εμφάνιση χτυπήματος χαρακτήρα.

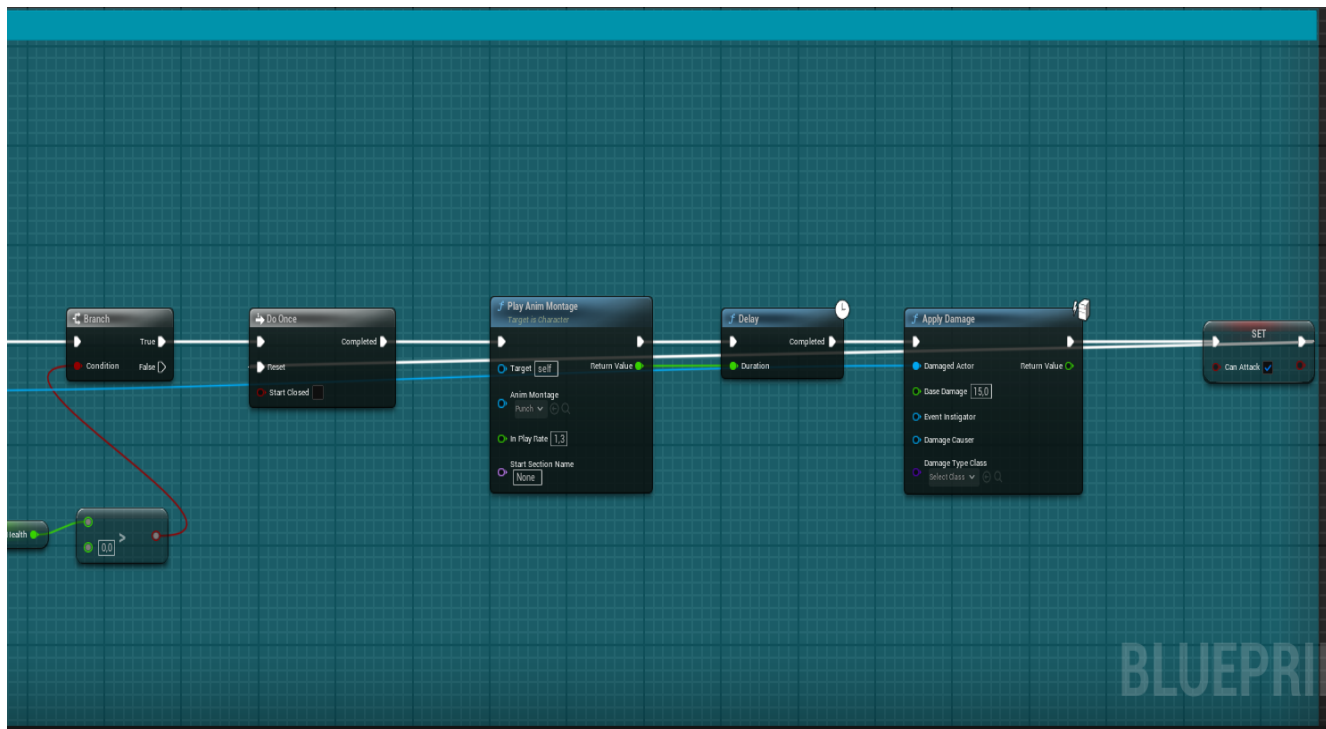


Εικόνα 3.26



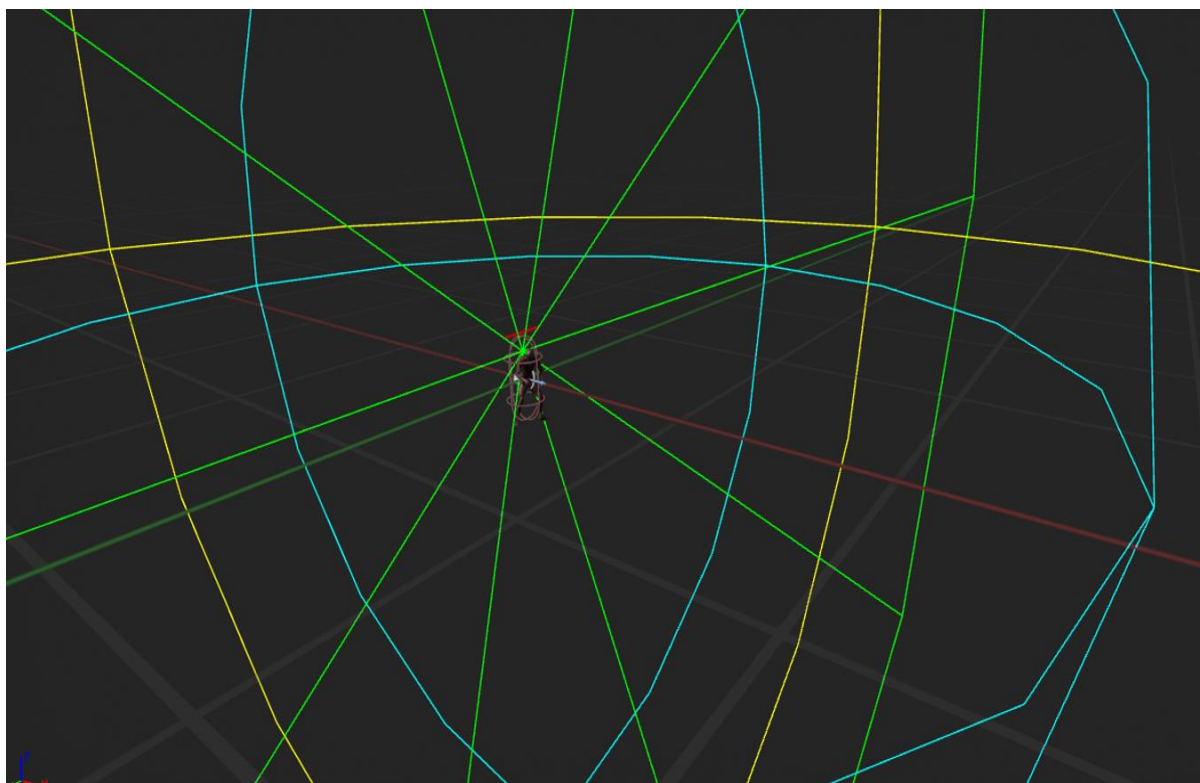
Εικόνα 3.27 Κώδικας για τον τρόπο θανάτου του χαρακτήρα.

Με την κλήση του Custom event death η πρώτη ενέργεια που γίνεται είναι να οριστεί το character movement με την τιμή none ώστε ο παίχτης να μην μπορεί να μετακινηθεί καθώς και το Input yaw και input scale με την τιμή 0 ώστε ο παίχτης να μην μπορεί να κινήσει την κάμερα με το ποντίκι. Στην συνέχεια γίνεται η κλήση της εντολής play montage όπου δέχεται έναν πίνακα με animation θανάτου και μέσω της εντολή random integer παράγει τυχαία ένα από αυτά μέσω τυχαίων ακέραιων. Όταν εκτελεστεί το animation τότε η μεταβλητή is dead γίνεται αληθής ώστε να γνωρίζει πως ο χαρακτήρας είναι νεκρός. Μετά από ένα μικρό delay 1,5 δευτερολέπτου το παιχνίδι μπαίνει σε κατάσταση παύσης και εμφανίζεται στην οθόνη του μενού Game Over με τις επιλογές όπου έχει ο χρήστης. Τέλος εμφανίζεται ο κέρσορας του ποντικιού ώστε να μπορεί ο χρήστης να επιλέξει το αν θέλει να συνεχίσει το παιχνίδι ή όχι.



Εικόνα 3.30 Κώδικας ζημιάς αντιπάλου.

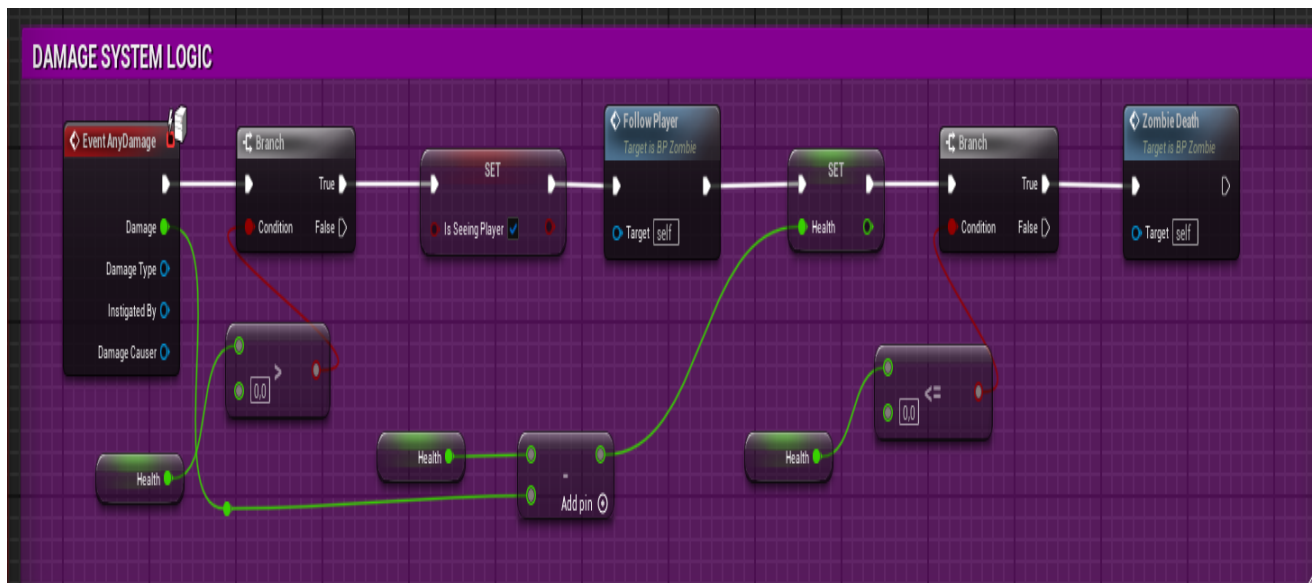
Για τον αντίπαλο χαρακτήρα χρησιμοποιώ μια σφαίρα όπου ονομάζεται pawn sensing. Αυτή η σφαίρα εντοπίζει χαρακτήρες όπου βρίσκονται μέσα σε εκείνη. Στην συγκεκριμένη περίπτωση στο σώμα του βασικού χαρακτήρα έχω δώσει το όνομα player body. Η σφαίρα αυτή ελέγχει αν στην ακτίνα της (ορισμένη ακτίνα 750 μέτρα παιχνιδιού) έχει βρεθεί σώμα με το όνομα player body (δίνει την εντύπωση ότι ο αντίπαλος έχει πραγματική όραση). Όταν εντοπίσει το player body τότε η μεταβλητή is seeing player γίνεται αληθής και καλεί το custom event follow player. Με την κλήση του follow player η μεταβλητή can attack γίνεται αληθής και ελέγχεται η συνθήκη για το αν ο αντίπαλος βλέπει τον χαρακτήρα. Αν είναι αληθής τότε μέσω της εντολή AI move to εισάγω ως pawn το σώμα του αντιπάλου και ως target το player body. Στην συνέχεια ελέγχεται η συνθήκη για το αν ο αντίπαλος μπορεί να επιτεθεί. Αν είναι ψευδής τότε το is seeing player γίνεται ψευδής γιατί σημαίνει πως ο χαρακτήρας χάθηκε από το οπτικό πεδίο του αντιπάλου. Αν είναι αληθής τότε το can attack γίνεται ψευδής, γίνεται έλεγχος για το αν ο παίχτης είναι ζωντανός και με μια εντολή do once παίζει το animation όπου ο αντίπαλος χτυπάει τον παίχτη και του κάνει ζημιά. Τέλος η μεταβλητή can attack ξανά γίνεται αληθής και καλείται επαναληπτικά η do once μέχρι ο αντίπαλος να μην βλέπει τον παίχτη πια ή ένας από τους δυο να είναι νεκρός.



Εικόνα 3.31 Σφαίρα pawn sensing.

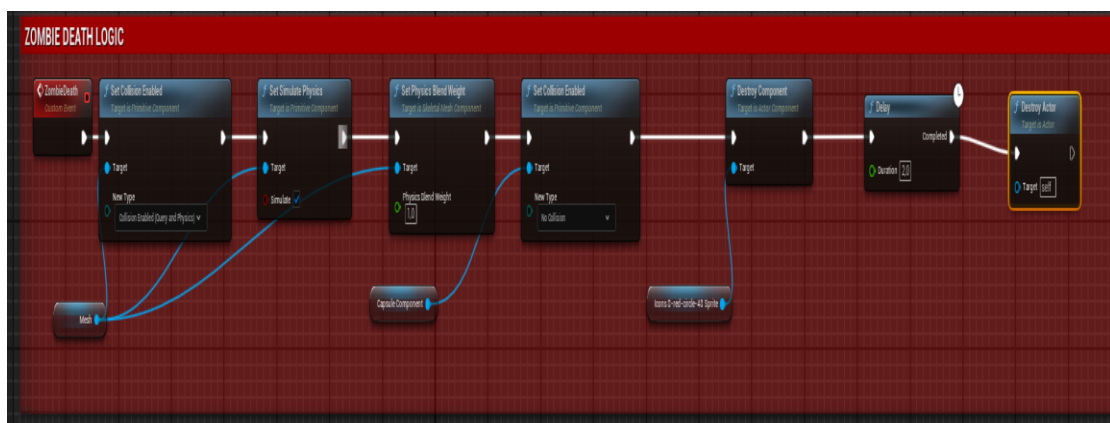


Εικόνα 3.32 Παράδειγμα χτυπήματος αντιπάλου.



Εικόνα 3.33 Κώδικας για τον τρόπο όπου δέχεται ζημιά ο αντίπαλος.

Η κλίση του Custom event Any damage καλείται όπως ανέφερα παραπάνω με την χρήση της εντολής apply damage στο όπλο του βασικού χαρακτήρα. Όταν καλείται ελέγχει αν ο αντίπαλος είναι ζωντανός, στην συνέχεια η μεταβλητή is seeing player γίνεται αληθής και καλείται το custom event follow player. Αυτό γίνεται έτσι ώστε αν ο αντίπαλος δεν έχει δει τον παίκτη για παράδειγμα αν του ρίχνει πισώπλατα τότε να γυρίσει και να τον κυνηγήσει. Επίσης μειώνεται η ζωή του αντιπάλου όσο είναι η ζημιά όπου δέχεται και ελέγχεται αν είναι νεκρός. Αν η συνθήκη είναι αληθής τότε καλείται το custom event zombie death.

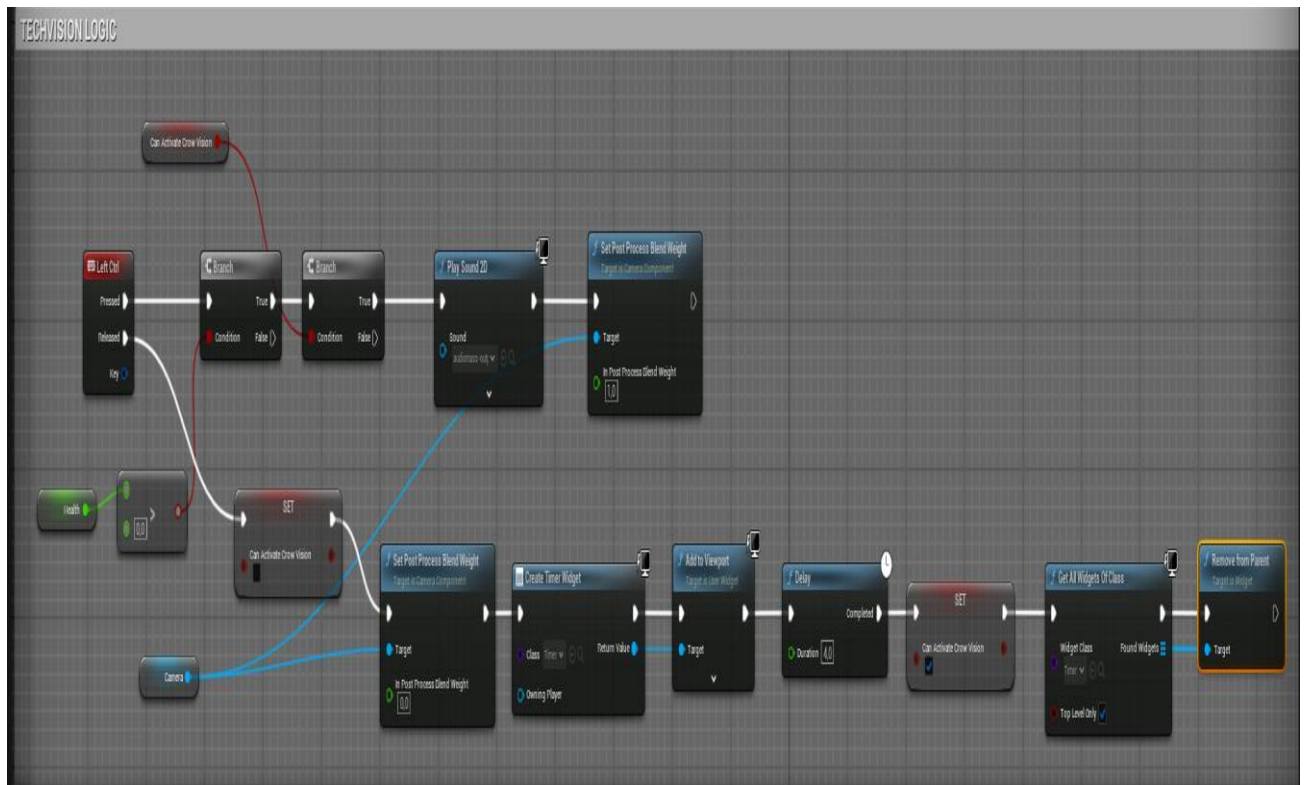


Εικόνα 3.34 Κώδικας θανάτου αντιπάλου.

Με την κλίση του custom event ZombieDeath αφαιρώ το collision και το βάρος από τον αντίπαλο δηλαδή τον κάνει να μοιάζει με κούκλα ώστε να δίνεται η εντύπωση πως δεν ελέγχει πια τους μυς του σώματος του. Στην συνέχεια καταστρέφει την εικόνα της κόκκινης σφαίρας όπου φαίνεται στο mini map για να φαίνεται πως ο εχθρός δεν υπάρχει πια και μετά από 2 δευτερόλεπτα καταστρέφει και το zombie.

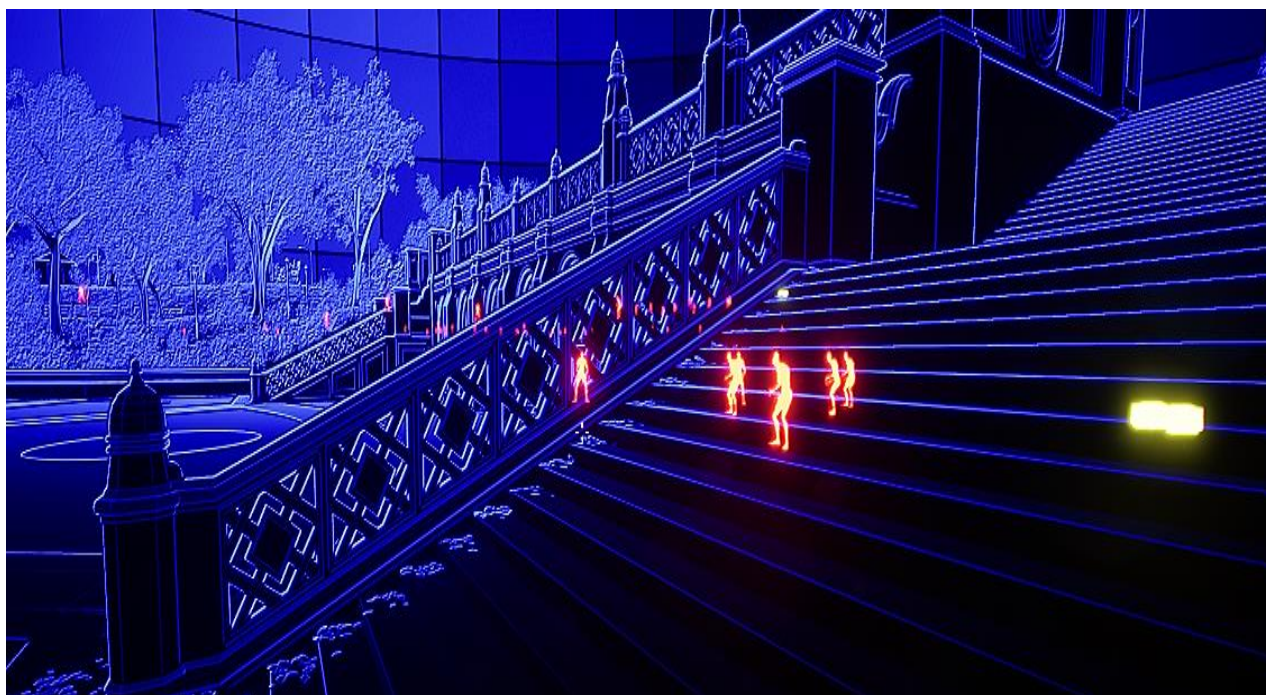


Εικόνα 3.35 Θάνατος αντιπάλου εντός παιχνιδιού.



Εικόνα 3.36 Κώδικας TechVision.

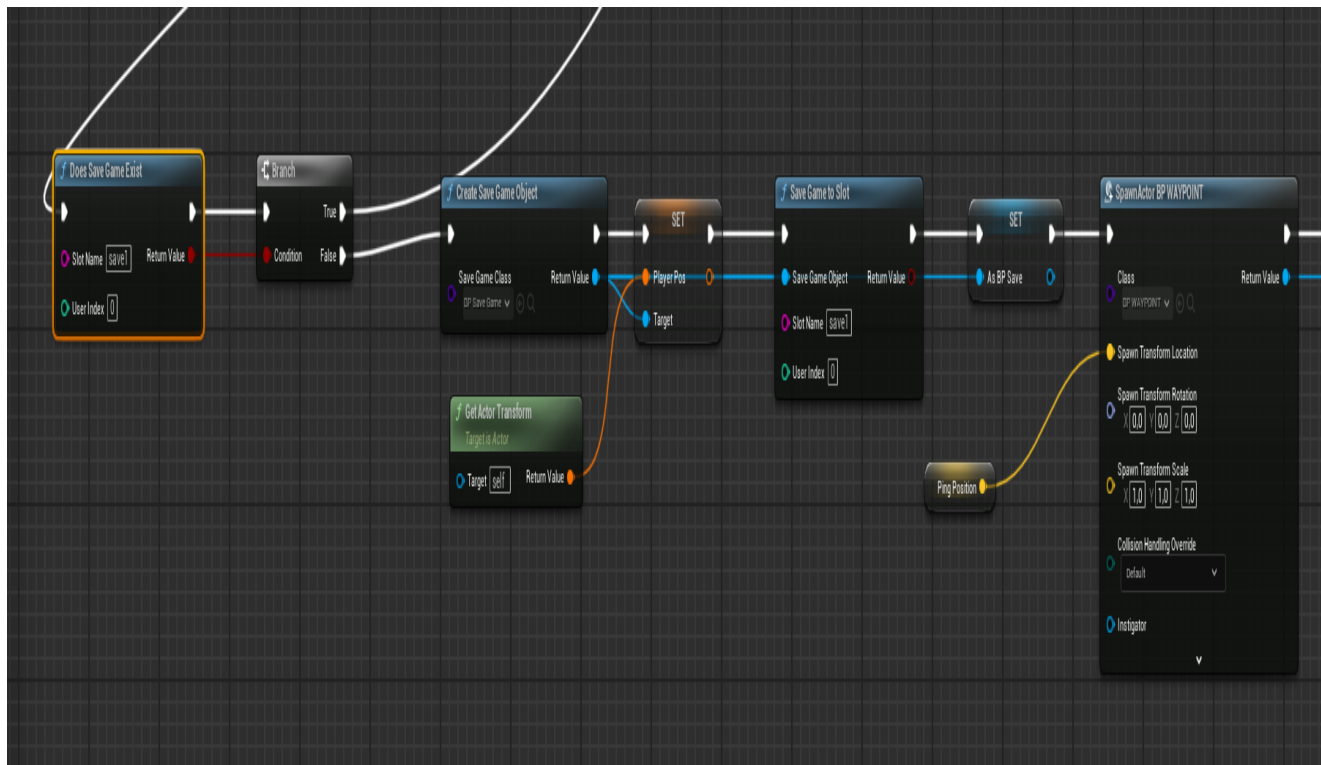
Το techvision είναι μια λειτουργία εντός παιχνιδιού όπου βοηθάει τον παίκτη να βλέπει αντιπάλους και κουτιά με σφαίρες μέσα από αντικείμενα. Αυτό επιτυγχάνεται με την αλλαγή του χρώματος του material στα αντικείμενα αυτά. Όταν ο χρήστης πατήσει το left ctrl τότε ελέγχεται η συνθήκη για το αν ο παίκτης είναι ζωντανός και στην συνέχεια αν μπορεί να ενεργοποιήσει το techvision. Αν και οι δυο έλεγχοι είναι αληθείς τότε ακούγεται ένας ήχος ενεργοποίησης και ενεργοποιείται το material αλλαγής χρωμάτων. Όταν αφήσει ο χρήστης το κουμπι τότε δεν μπορεί να ενεργοποιηθεί το techvision, το material απενεργοποιείται και εμφανίζεται στην οθόνη ένα χρονόμετρο αντιστροφής μέτρησης 4 δευτερολέπτων. Μετά από 4 δευτερόλεπτα μπορεί να ενεργοποιηθεί το techvision και το χρονόμετρο εξαφανίζεται.



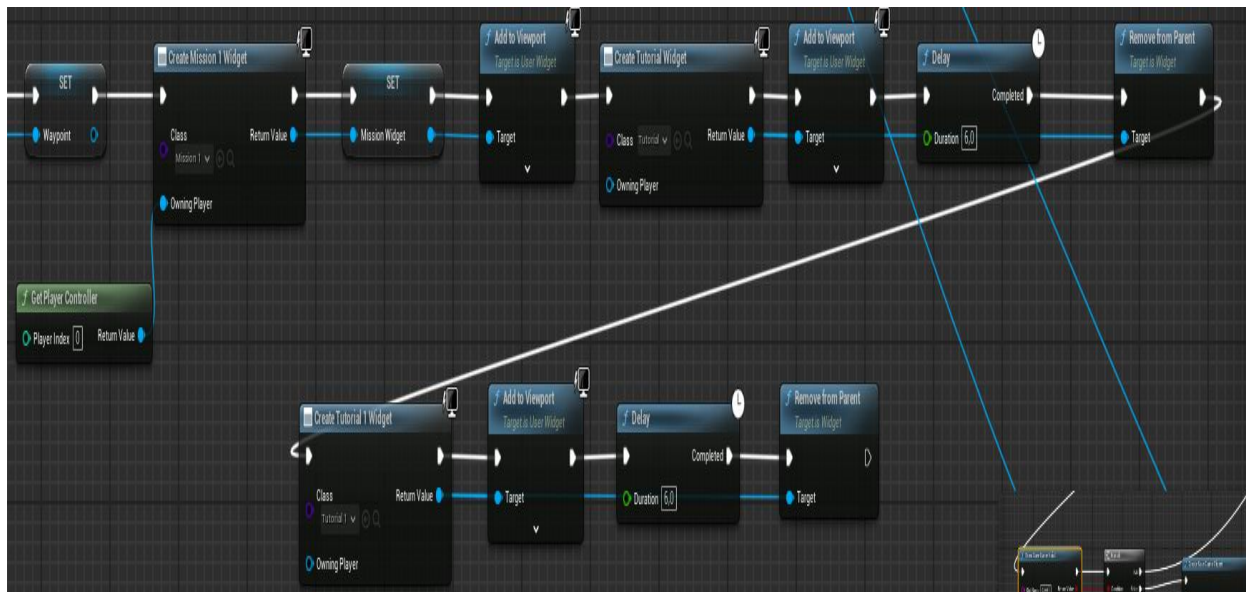
Εικόνα 3.37 Παράδειγμα με ενεργοποίηση του TechVision.



Εικόνα 3.38 Παράδειγμα με την εμφάνιση του timer.



Εικόνα 3.39



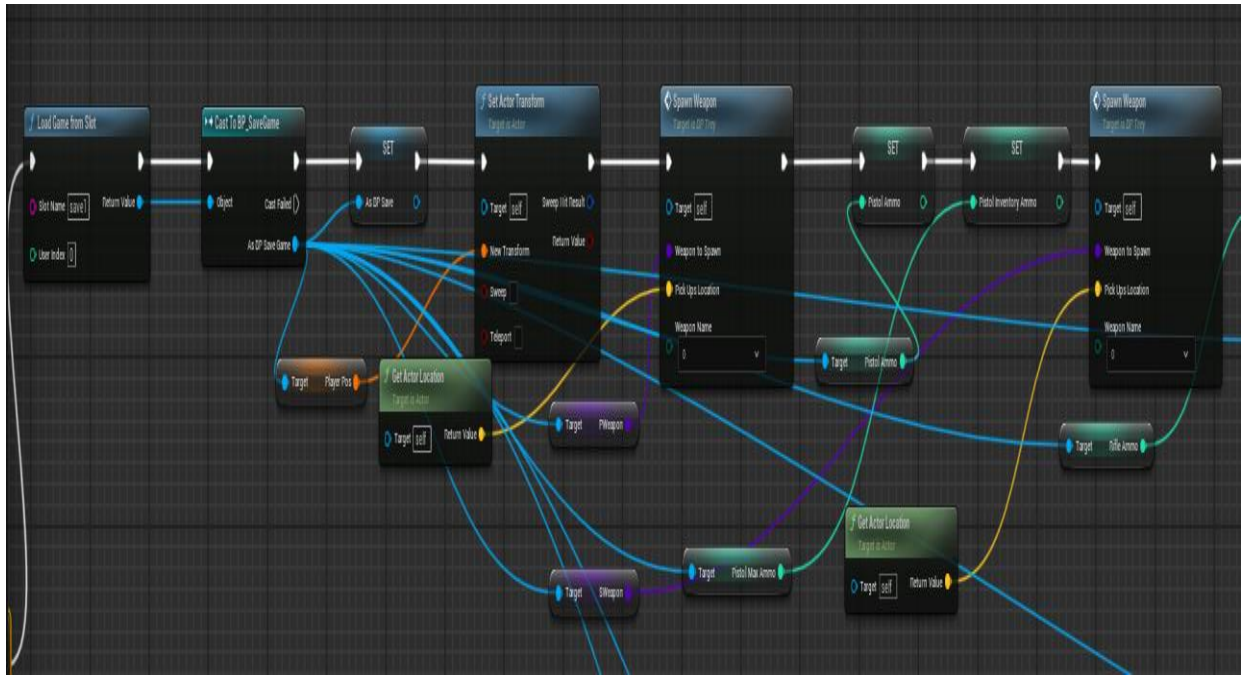
Εικόνα 3.40 Κώδικας για νέο παιχνίδι.

Με την έναρξη του παιχνιδιού γίνεται έλεγχος με την εντολή does save game exist αν υπάρχουν αποθηκευμένα δεδομένα με το όνομα save1. Αν δεν υπάρχουν τότε με το function create save game object δημιουργείται καινούργιο save game, στην μεταβλητή player pos αποθηκεύεται η τιμή της τοποθεσίας που βρίσκεται ο παίχτης και αποθηκεύουμε το save game στην

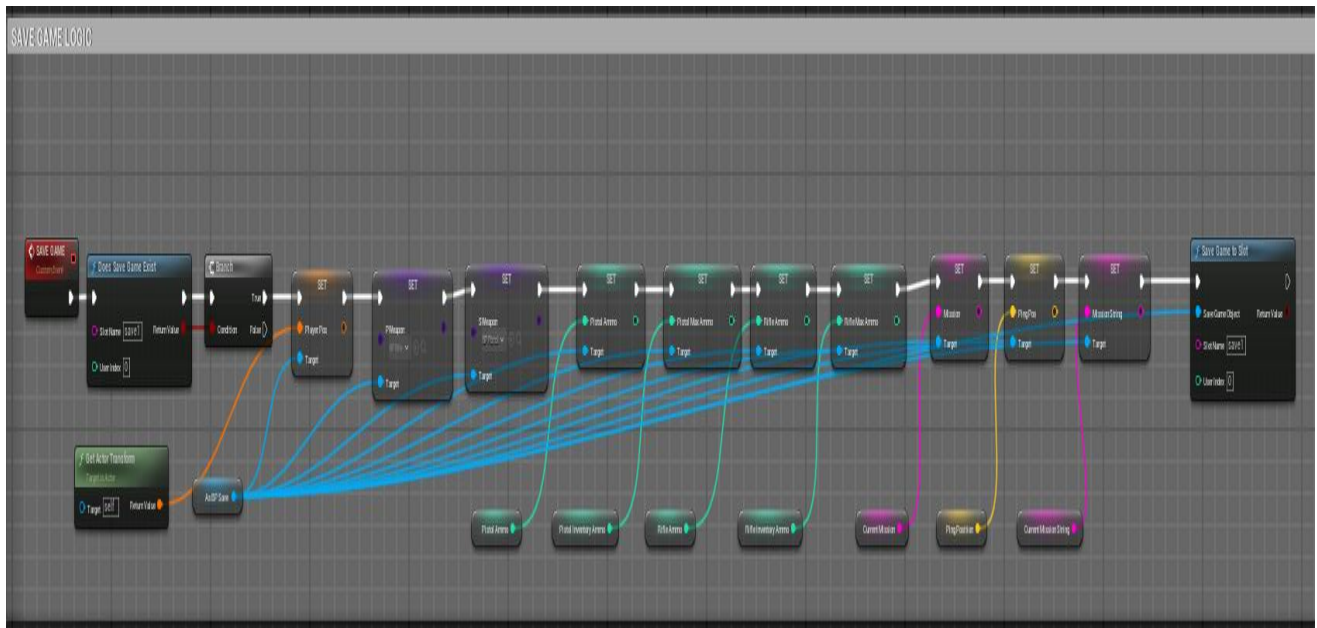
θέση save1. Στην συνέχεια αποθηκεύει το function σε μια μεταβλητή με όνομα BP save ώστε να υπάρχει να μελλοντική χρήση όταν χρειαστεί να καλεστεί κάποια μεταβλητή ή εντολή από το save game. Επιπλέον εμφανίζεται στην θέση ring position (η ring position έχει την θέση της πρώτης αποστολής του παιχνιδιού και είναι προκαθορισμένη τοποθεσία εντός της μεταβλητής) ένα ring όπου δείχνει στον παίχτη που να κατευθυνθεί. Τέλος εμφανίζεται το widget με την πρώτη αποστολή του παίχτη μέσω των εντολών create mission 1 widget και add to viewport και επίσης εμφανίζεται το tutorial widget με τον ίδιο τρόπο το οποίο δίνει πληροφορίες στον παίχτη για τα κουμπιά του παιχνιδιού.



Εικόνα 3.41 Εμφάνιση mission, ping και tutorial με την έναρξη του παιχνιδιού.



Εικόνα 3.42 Κώδικας για συνέχεια αποθηκευμένου παιχνιδιού.



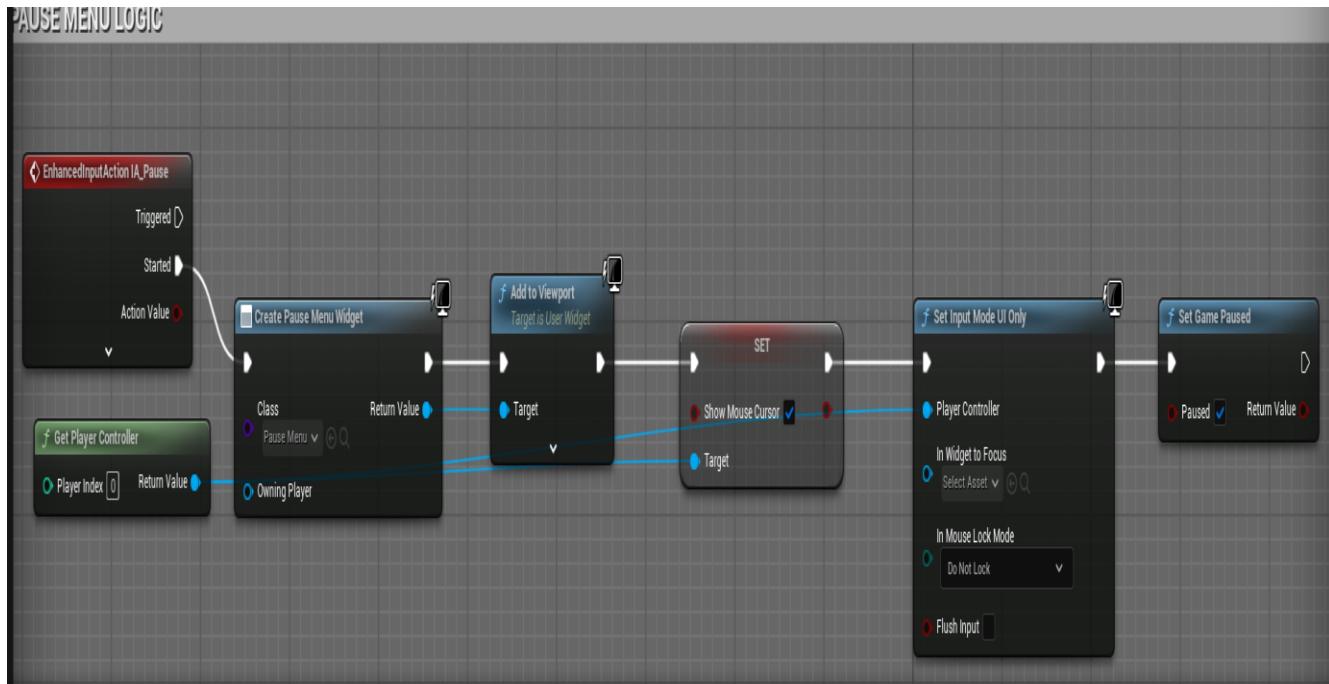
Εικόνα 3.43 Κώδικας για αποθήκευση παιχνιδιού.

Στην περίπτωση όμως όπου η συνθήκη save game exist είναι αληθής τότε εκτελείται η εντολή Load game from slot και συγκεκριμένα από το slot save 1 που έχει αποθηκευτεί. Αρχικά μέσω του Cast To BP save ο κώδικας παίρνει τα αποθηκευμένα δεδομένα από τις μεταβλητές όπου βρίσκονται στο blueprint του save game. Χρησιμοποιώντας τα εμφανίζει τον χαρακτήρα

στην τοποθεσία όπου έγινε το τελευταίο save, εμφανίζει τα όπλα στην χέρια του παίχτη και αλλάζει τις σφαίρες του γεμιστήρα και της αποθήκης σε αυτές όπου είχε όταν έκανε save. Το custom event save game καλείται όταν ο παίχτης πατήσει στο pause menu το κουμπί save game. Με την κλήση της γίνεται έλεγχος για το αν υπάρχει ήδη προηγούμενο save game και αν ναι τότε αποθηκεύει όλα τα δεδομένα των μεταβλητών player pos (θέση του παίχτη), pistol ammo, pistol inventory ammo, rifle ammo, rifle inventory ammo, current mission (η αποστολή όπου βρίσκεται) ,ping position, current mission string (το όνομα της αποστολής) καθώς και τα όπλα που κρατάει ο παίχτης. Τέλος αποθηκεύει το παιχνίδι στο slot save 1.

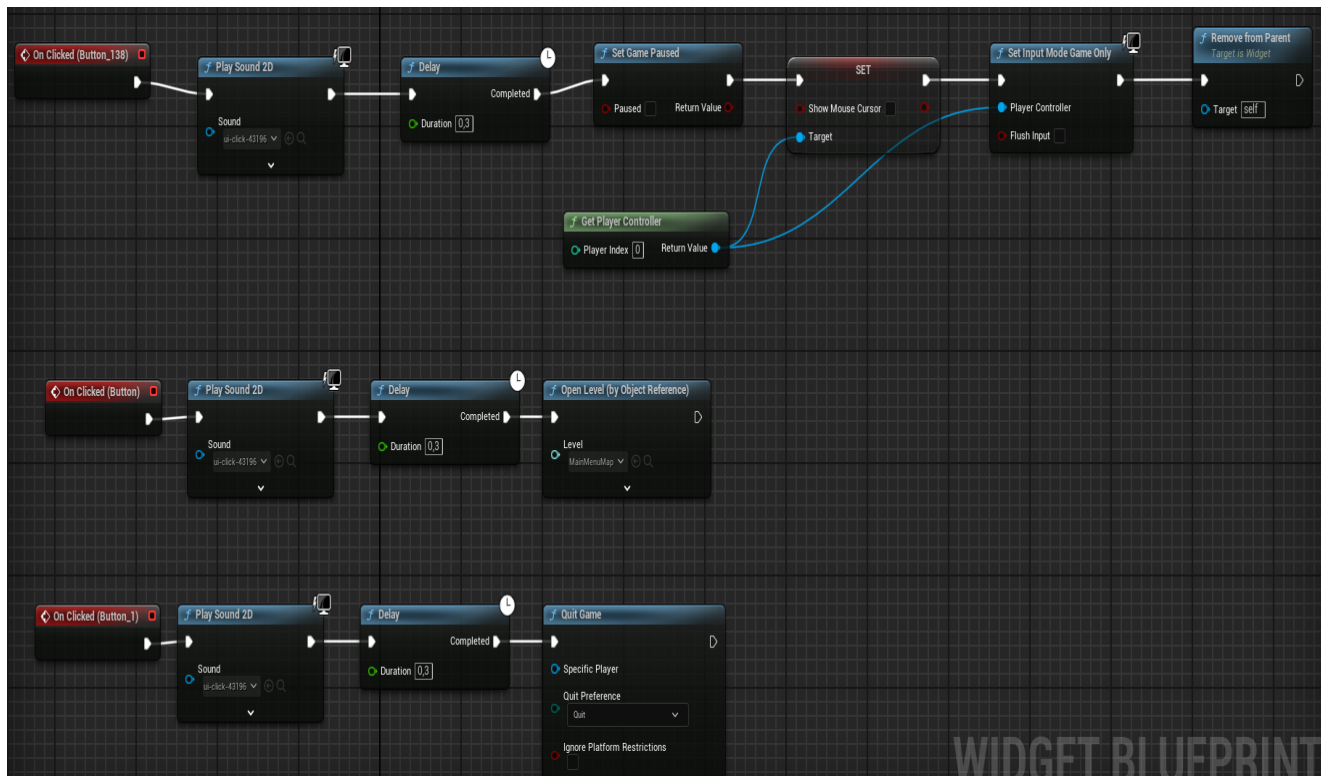


Εικόνα 3.44 Μενού παύσης.



Εικόνα 3.45 Κώδικας για παύση παιχνιδιού.

Το pause menu εμφανίζεται όταν ο χρήστης πατήσει το κουμπί P. Αυτό γίνεται με την χρήση του input action IA_Pause το οποίο όταν καλεστεί εμφανίζει το widget όπου φαίνεται στην εικόνα 3.44. Στην συνέχεια εμφανίζεται ο κέρσορας του ποντικού και εκτελείται η εντολή set input mode ui only ώστε ο χρήστης να μπορεί να κάνει κλικ μόνο σε UI δηλαδή στα κουμπιά του pause menu. Στην συνέχεια το παιχνίδι μπαίνει σε κατάσταση παύσης.

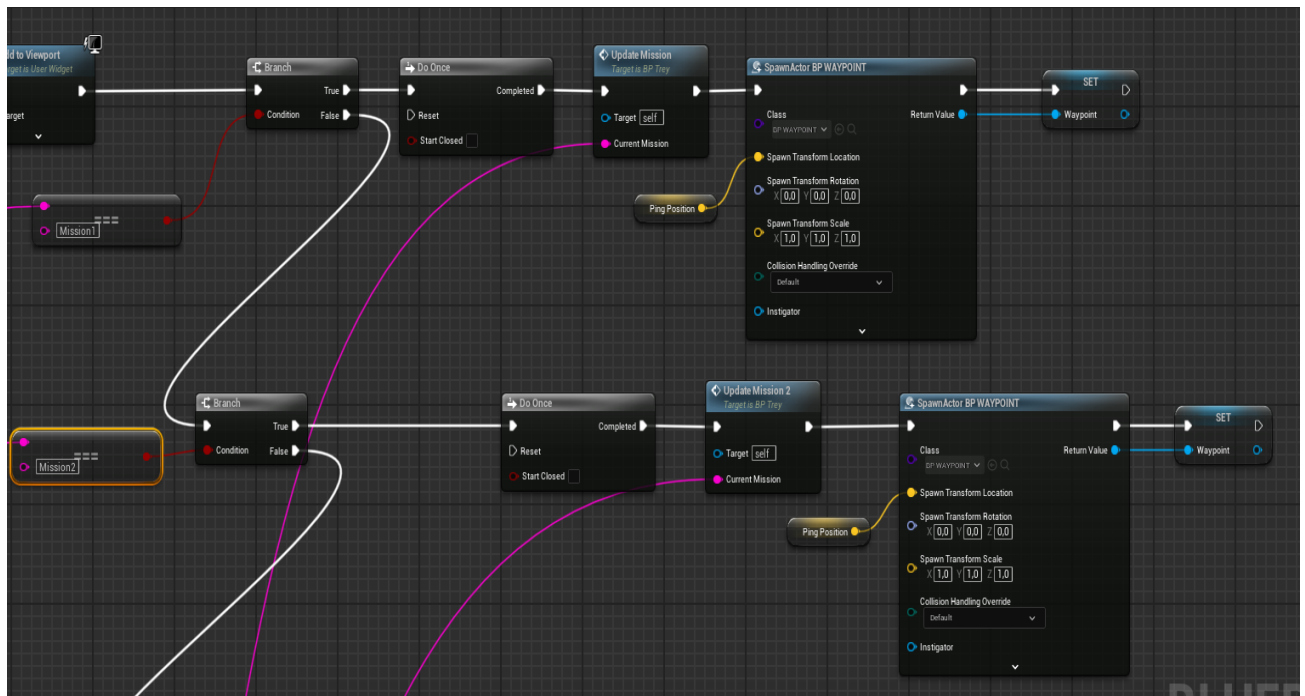


Εικόνα 3.46 Κώδικας για τα κουμπιά του μενού παύσης.

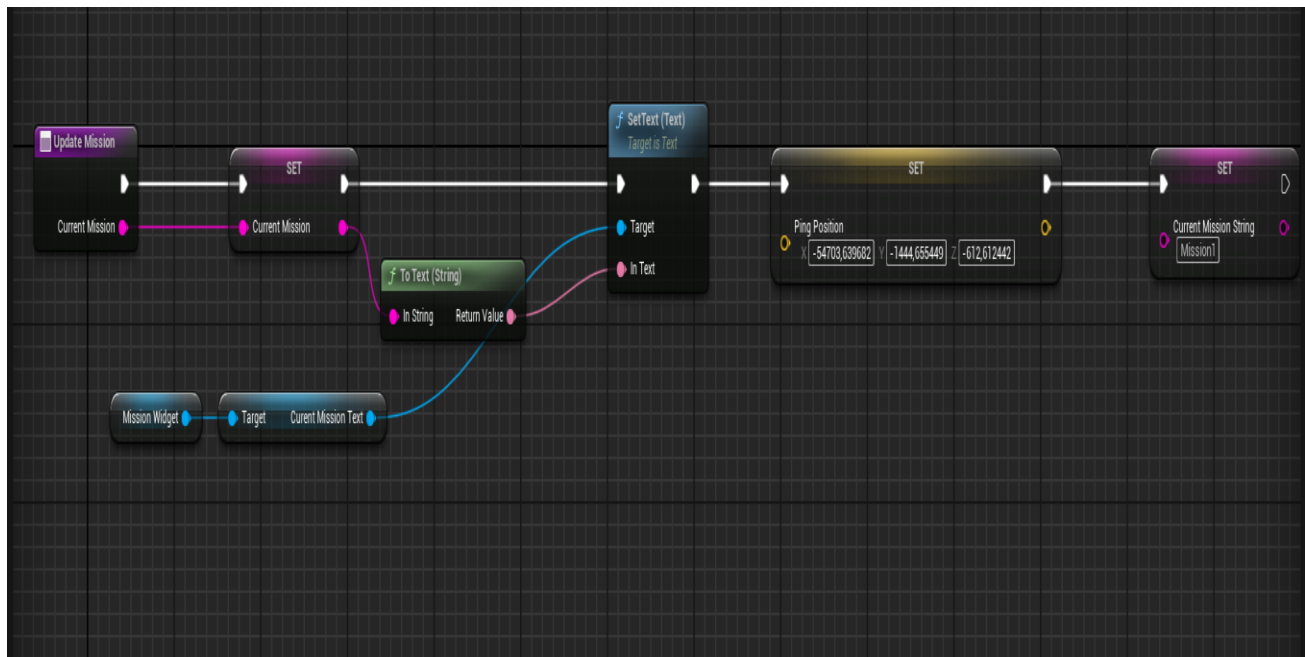
Αν ο χρήστης κάνει κλικ στο resume button τότε ακούγεται ένας ήχος (ο οποίος είναι κοινός για όλα τα κουμπιά) που δίνει την αίσθηση ότι ο χρήστης πάτησε κάποιο κουμπί. Στην συνέχεια μετά από ένα delay 0.3 δευτερολέπτων ώστε να προλάβει να εκτελεστεί ο ήχος το παιχνίδι βγαίνει από κατάσταση παύσης κάνοντας την τιμή της μεταβλητής set game paused false. Επιπλέον ο κέρσορας εξαφανίζεται, το input mode καθορίζεται σε κατάσταση παιχνιδιού και το menu εξαφανίζεται από την οθόνη.

Αν ο χρήστης πατήσει το κουμπί return to main menu τότε επαναλαμβάνεται η διαδικασία με τον ήχο όπως και στο κουμπί resume και καλείται το αρχικό μενού με την χρήση του function open level by object reference κάνοντας αναφορά στο όνομα του menu.

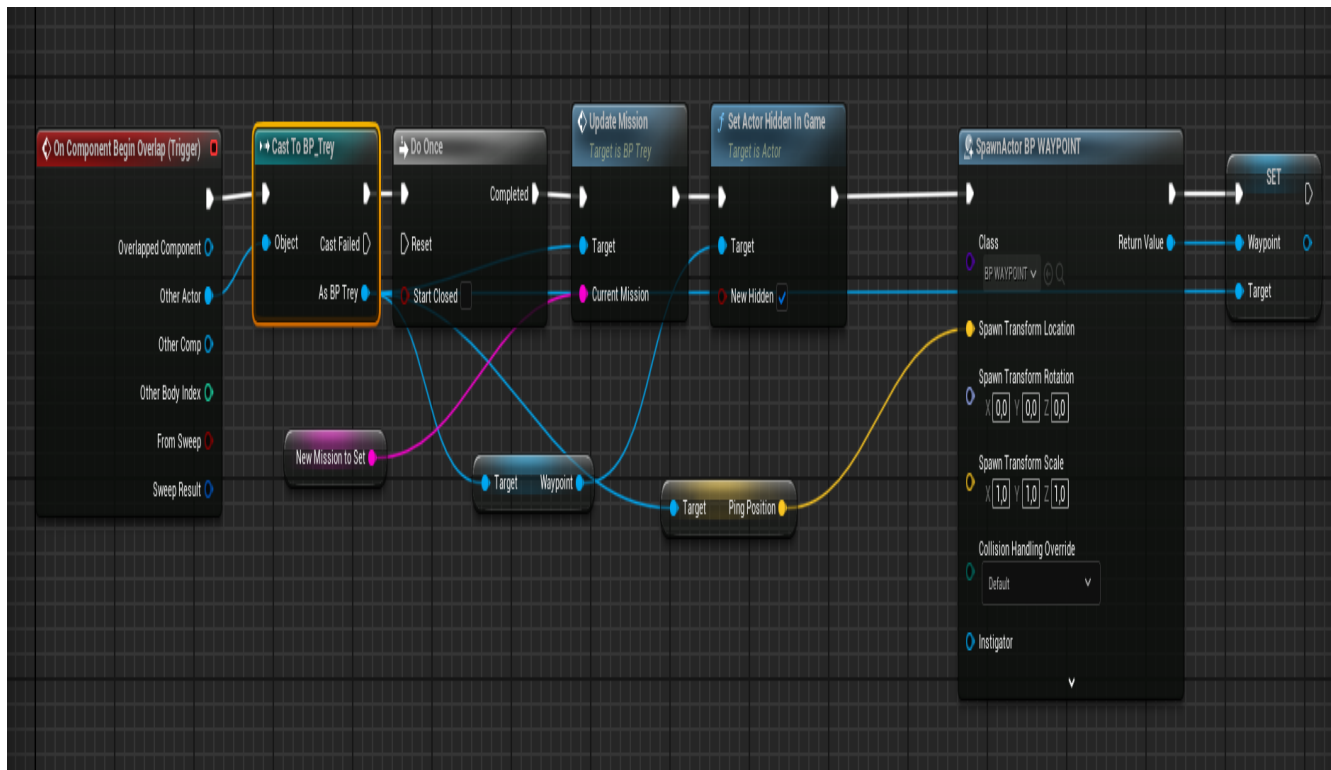
Αν ο χρήστης κάνει κλικ στο κουμπί quit game τότε για ακόμη μια φορά γίνεται η διαδικασία αναπαραγωγής ήχου και μέσω του function quit game το παιχνίδι τερματίζει και ο χρήστης βρίσκεται στην αρχική οθόνη του υπολογιστή του.



Εικόνα 3.47 Κώδικας για την ενεργοποίηση αποστολών.



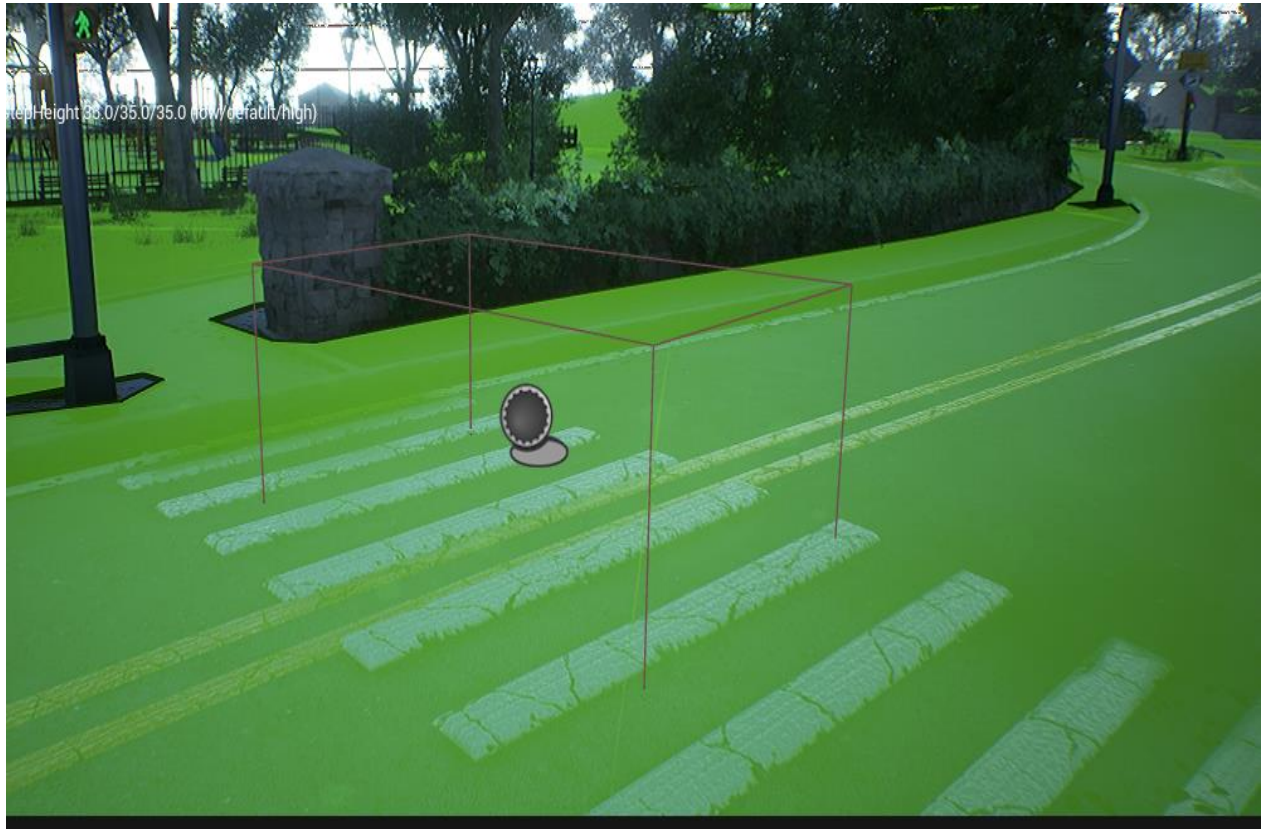
Εικόνα 3.48 Function Update Mission.



Εικόνα 3.49 Κώδικας για το trigger box.

Για να γίνει η κλήση της εκάστοτε αποστολής αρχικά γίνεται χρήση μια συνθήκης όπου ελέγχει σε ποια αποστολή βρίσκεται ο παίχτης. Αυτό γίνεται με την χρήση ενός string όπου αναγράφει το όνομα της αποστολής που βρίσκεται ο παίχτης. Αν η συνθήκη είναι αληθής τότε με την χρήση της εντολής do once εκτελείται η συνέχεια του κώδικα για μόνο μια φορά. Αν ο παίχτης μπει σε ένα trigger box (trigger box είναι αόρατα κουτιά όπου έχω τοποθετήσει στον χάρτη έτσι ώστε όταν ο παίχτης πατήσει μέσα σε αυτά να καλείται η εκάστοτε αποστολή) καλείται η function Update mission. Μέσα σε εκείνη μεταφράζουμε την τιμή του current mission (η τιμή του ορίζεται από το trigger box) σε κείμενο έτσι ώστε μέσω του function set text να το εισάγει ο κώδικας στην μεταβλητή current mission text μια να αλλάξει η αποστολή στην οθόνη. Στην συνέχεια γίνεται αλλαγή της τιμής του ping position εισάγοντας τις συντεταγμένες όπου θα εμφανιστεί το επόμενο ping στον χάρτη και γίνεται αλλαγή current mission string στο string όπου αντιστοιχεί στην αποστολή όπου μπαίνει ο παίχτης. Η διαδικασία αυτή είναι κοινή για όλες τις αποστολές. Όταν ο χρήστης μπει σε ένα trigger box τότε μέσω του custom event on component begin overlap ξανά για μια φορά μέσω τις do once καλείται το function update mission εισάγοντας στην μεταβλητή current mission την τιμή της μεταβλητής new mission to set που για την κάθε αποστολή έχει ένα συγκεκριμένο κείμενο οδηγιών όπου

του έχω δώσει από το πληκτρολόγιο. Όταν τελειώσει αυτή η διαδικασία μέσω του function set actor hidden in game και κάνοντας target το ring όπου έδειχνε την προηγούμενη αποστολή γίνεται αόρατο και στην συνέχεια μέσω της εντολής spawn actor γίνεται εμφάνιση του ring στην επόμενη τοποθεσία που έχει οριστεί.



Εικόνα 3.50 Παράδειγμα trigger box εντός παιχνιδιού.

ΚΕΦΑΛΑΙΟ 5 ΣΥΜΠΕΡΑΣΜΑΤΑ

Με βάση όλων όσων εξετάστηκαν σε αυτή την πτυχιακή εργασία συμπεραίνουμε πως τα παιχνίδια πέρα από έναν τρόπο ψυχαγωγίας είναι και ένας κλάδος όπου απαιτεί ικανότητες, χρόνο και γνώσεις ώστε να γίνει η παραγωγή τους. Επιπλέον πρέπει να υπάρχει συνεχής ενημέρωση για τους προγραμματιστές ειδικά για τα νέα ΑΙ εργαλεία που συνεχώς αναπτύσσονται ώστε να φτάσουν οι δυνατότητες τους στον μέγιστο βαθμό. Το ΑΙ βρίσκεται ακόμα στα αρχικά στάδια παραγωγής του και γνωρίζουμε μόνο τα ελάχιστα όπου μπορεί να προσφέρει στα παιχνίδια καθώς οι δυνατότητες του συνεχώς μας εκπλήσσουν. Ήδη τα παιχνίδια με την χρήση του είναι πολύ πιο απαιτητικά στην παραγωγή τους σε σχέση με το πως ήταν πριν κάποια χρόνια και για αυτό υπάρχουν χιλιάδες προγραμματιστές όπου συνεργάζονται ώστε να υπάρξει το επιθυμητό αποτέλεσμα. Τέλος συμπεραίνουμε πως παρόλο που τα online multiplayer παιχνίδια είναι πιο δημοφιλή λόγω της δυσκολία λόγω του ανταγωνισμού με άλλους παίχτες, τα single player παιχνίδια με την χρήση του ΑΙ μπορούν να φτάσουν σε επίπεδα όπου η δυσκολία να ξεπερνά τις ικανότητες ενός ανθρώπου κάνοντας τα ξανά δημοφιλή για τους χρήστες όπου ψάχνουν ένα πιο ανταγωνιστικό τρόπο παιχνιδιού.