

**Τμήμα Πληροφορικής & Τηλεπικοινωνιών Πανεπιστήμιο
Θεσσαλίας**



Πτυχιακή Εργασία

**Σχεδιασμός και υλοποίηση βιντεοπαιχνιδιού με χρήση της
μηχανής Unreal Engine**

Ονοματεπώνυμο Φοιτητή: Μυλωθρός Χαράλαμπος

ΑΜ:2113125

Επιβλέπων Καθηγητής: Σταμούλης Γεώργιος

Συν-επιβλέπων: Λέκτορας Π.Δ. 407/80 Λουκάκης Γεώργιος



«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 επεξεργάστηκαν, 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 αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσιάσή 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.

(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.»

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή Σταμούλη Γεώργιο καθώς και τον συν-επιβλέποντα λέκτορα Π.Δ. 407/80 Λουκάκη Γεώργιο για την υποστήριξη και την βοήθεια που μου παρείχε κατά την εκπόνηση αυτής της εργασίας στα εμπόδια που συνάντησα.

Περίληψη

Αντικείμενο της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση ενός τρισδιάστατου παιχνιδιού πρώτου προσώπου. Το παιχνίδι θεματολογικά, τοποθετείται ανάμεσα στις κατηγορίες τρόμου και μυστηρίου και αποτελείται από δυο επιμέρους επίπεδα. Σκοπός του παιχνιδιού, είναι η αναζήτηση και η εύρεση συγκεκριμένων κομματιών που αποτελούν τμήματα της ψυχής του βασικού χαρακτήρα, προκειμένου στη συνέχεια να αποδράσει από τον ενδιάμεσο χώρο του καθαρτηρίου των ψυχών (Purgatory). Στην εργασία αυτή, γίνεται αναφορά στην υλοποίηση του βασικού gameplay (τρόπος παιχνιδιού) και της μηχανικής του παιχνιδιού, καθώς και στον σχεδιασμό και την ανάπτυξη της διεπαφής. Μέσα από τις διαδικασίες σχεδιασμού και υλοποίησης που ακολουθήθηκαν, έγιναν κατανοητές οι απαραίτητες διεργασίες υλοποίησης ενός βιντεοπαιχνιδιού, ενώ παράλληλα δόθηκε η δυνατότητα βελτίωσης αρκετών προγραμματιστικών δεξιοτήτων μέσω των αναγκών που προέκυψαν κατά τη διάρκεια των απαραίτητων ρών εργασίας. Για την υλοποίηση του παιχνιδιού χρησιμοποιήθηκε η μηχανή Unreal Engine 4 (UE4) ενώ παράλληλα έγινε χρήση του λογισμικού Audacity για την επεξεργασία ήχου και του Adobe Photoshop CS6 για την επεξεργασία των επιμέρους γραφικών στοιχείων.

Abstract

The subject of this thesis is the design and realisation of a three dimensional First person shooter (FPS). The game can be classified into the horror and mystery genres and consists of two separate levels. The aim of the game is to find specific pieces, which are part of the soul of the main character in order to escape from the intermediate space of the Purgatory. This thesis describes the creation and engineering of the basic gameplay, as well as the designing and development of the interface. The planning and creation of this game has given insight into the essential game development processes, while it has been possible to improve several programming skills through the needs that emerged during the necessary workflows.

The tools used for the production of this game are Unreal Engine 4 (UE4), Audacity Software for sound processing, as well as Adobe Photoshop CS6 for image processing and computer graphics.

Περιεχόμενα

Περίληψη.....	σελ.2
1.Εισαγωγή.....	σελ.5
1.1 Μηχανή παραγωγής γραφικών.....	σελ.5
1.2 Unreal engine 4.....	σελ.6
1.3 Ιστορική αναδρομή UE4.....	σελ.6
1.4 Περιβάλλον της UE4.....	σελ.7
2.Τίτλος.....	σελ.8
3.Επισκόπηση παιχνιδιού.....	σελ.9
3.1 Είδος παιχνιδιού.....	σελ.9
3.2 Στοχευμένο κοινό.....	σελ.9
3.3Διάγραμμα ροής παιχνιδιού.....	σελ.10
3.4 Βασική εμφάνιση και αίσθηση του παιχνιδιού.....	σελ.10
4.Διεπαφή του χρήστη (Interface).....	σελ.10
4.1 Game controls.....	σελ.16
4.2 Ανάλυση gameplay.....	σελ.16
5. Blueprints.....	σελ.19
5.1 Τρόπος υλοποίησης mechanics.....	σελ.22
5.2 Σημειώσεις.....	σελ.53
6. Μελλοντικές Προεκτάσεις.....	σελ.53
7. Πηγές για τα asset που χρησιμοποιήθηκαν.....	σελ.53
8. Βιβλιογραφία.....	σελ.54

1. Εισαγωγή

Παιχνίδι είναι μια δομημένη δραστηριότητα με σκοπό την ψυχαγωγία ή την εκπαίδευση. Η δομή ενός παιχνιδιού ορίζεται από κανόνες και επιτεύγματα. Τα επιτεύγματα επιτυγχάνονται μέσω αλληλεπιδράσεων του παίχτη με άλλους παίχτες ή με τον υπολογιστή. Γενικά, σκοπός του παιχνιδιού είναι να υποβάλουν τον παίκτη σε μια ψυχική ή σωματική διέγερση. Ένα παιχνίδι μπορεί να βοηθήσει στην ανάπτυξη πρακτικών δεξιοτήτων και να έχει τη μορφή άσκησης. Επιπλέον, μπορεί να έχει εκπαιδευτικό, προσομοιωτικό ή θεραπευτικό χαρακτήρα.

Τα πρώτα παιχνίδια εμφανίστηκαν το 2600 π.Χ. και αποτελούνταν κυρίως από επιτραπέζια και από αθλήματα. Μεγάλη εξέλιξη στον τομέα των παιχνιδιών έγινε το 1947, όπου εμφανίστηκε το πρώτο βιντεοπαιχνίδι. Όπως ένα κανονικό παιχνίδι, έτσι και ένα βιντεοπαιχνίδι δομείται από κανόνες και επιτεύγματα. Σε αντίθεση όμως με ένα κανονικό παιχνίδι η αλληλεπίδραση που διαθέτει, συμβαίνει μέσω μιας διεπαφής χρήστη (οπτική ανάδραση σε μια συσκευή προβολής βίντεο, π.χ. υπολογιστής).

Τα βιντεοπαιχνίδια δεν άλλαξαν μόνο τον τρόπο ψυχαγωγίας των χρηστών, αλλά επηρέασαν και την παγκόσμια οικονομία. Πρόσφατες έρευνες έδειξαν ότι πάνω από 164 εκατομμύρια ενήλικες στις Ηνωμένες Πολιτείες παίζουν βιντεοπαιχνίδια. Το 2018 καταγράφηκαν στον κλάδο της βιομηχανίας των βιντεοπαιχνιδιών πωλήσεις άνω των 43,4 δισεκατομμυρίων δολαρίων.

Η ανάπτυξη ενός βιντεοπαιχνιδιού απαιτεί την διεκπεραίωση μιας σειράς διαδικασιών. Η ακόλουθη πτυχιακή εργασία αναφέρεται στις διαδικασίες αυτές και συγκεκριμένα στην ανάπτυξη ενός πρώτου προσώπου βιντεοπαιχνιδιού επιβίωσης, τρόμου και μυστηρίου. Για την υλοποίηση του χρησιμοποιήθηκε μια μηχανή παραγωγής παιχνιδιών η Unreal Engine 4. Θα αναλυθεί τι είναι μια μηχανή γραφικών και στην συνέχεια θα δοθεί περισσότερη έμφαση στην UE4.

1.1 Μηχανή παραγωγής παιχνιδιών

Οι μηχανές παραγωγής βιντεοπαιχνιδιών είναι μια κατηγορία λογισμικών που χρησιμοποιούνται για την δημιουργία βιντεοπαιχνιδιών και είναι σχεδιασμένες να υποστηρίζονται σε διάφορα λειτουργικά συστήματα. Οι μηχανές αυτές διαθέτουν διάφορα εργαλεία όπως π.χ. μια μηχανή φωταπόδοσης δισδιάστατων και τρισδιάστατων γραφικών ή μια εφαρμογή εντοπισμού συγκρούσεων. Επιπρόσθετα, διαθέτουν εργαλεία διαχείρισης ήχου, γραφικών και animation (κινούμενων σχεδίων), καθώς και εργαλεία ανάπτυξης κώδικα visual scripting (γραφικός προγραμματισμός), ανάπτυξης τεχνητής νοημοσύνης, δικτύωσης και διαχείρισης μνήμης.

Τα παραπάνω εργαλεία που περιέχονται στις μηχανές επιτρέπουν μια ευκολότερη και ταχύτερη διαδικασία υλοποίησης σε σχέση με το παρελθόν όπου η αποκλειστική χρήση προγραμματιστικού κώδικα ήταν μονόδρομος.

1.2 Unreal engine 4

Η UE4 είναι η μηχανή παραγωγής παιχνιδιών που χρησιμοποιήθηκε για την υλοποίηση του παιχνιδιού. Αναπτύχθηκε από την εταιρεία Epic Games και χρησιμοποιεί την γλώσσα προγραμματισμού C++. Χρησιμοποιήθηκε για πρώτη φορά το 1998 στο παιχνίδι πυροβολισμού πρώτου προσώπου Unreal. Αρχικά προορίζονταν αποκλειστικά για την δημιουργία παιχνιδιών πυροβολισμού, αλλά στην πορεία χρησιμοποιήθηκε και σε άλλα είδη με επιτυχία. Η τελευταία έκδοση είναι η Unreal Engine 4, κυκλοφόρησε το 2014 και παρέχεται δωρεάν για ακαδημαϊκή χρήση.

1.3 Ιστορική αναδρομή UE4

Η πρώτη έκδοση Unreal Engine 1 (UE1) αναπτύχθηκε από τον Tim Sweeney τον ιδρυτή της Epic Games. Το 1998 ανακοινώθηκε η κυκλοφορία του πρώτου παιχνιδιού που αναπτύχθηκε στην UE1, το Unreal. Το Unreal ήταν το πρώτο παιχνίδι πυροβολισμού πρώτου προσώπου που κυκλοφόρησε. Σε αυτή την πρώτη έκδοση εισήγαγε το λογισμικό έγχρωμης φωτοαπόδοσης, μια εφαρμογή εντοπισμού συγκρούσεων και το texture filtering (προσδιορισμός του χρώματος ενός αντικειμένου). Επιπλέον, η μηχανή παρείχε ένα Level Editor (επεξεργαστή επιπέδων), τον UnrealEd, που έδινε την δυνατότητα διαρρύθμισης των αντικειμένων στον χώρο του επιπέδου, πράγμα που μείωσε δραματικά τον χρόνο κατασκευής του επιπέδου και βελτίωσε την απόδοση των σχεδιαστών.

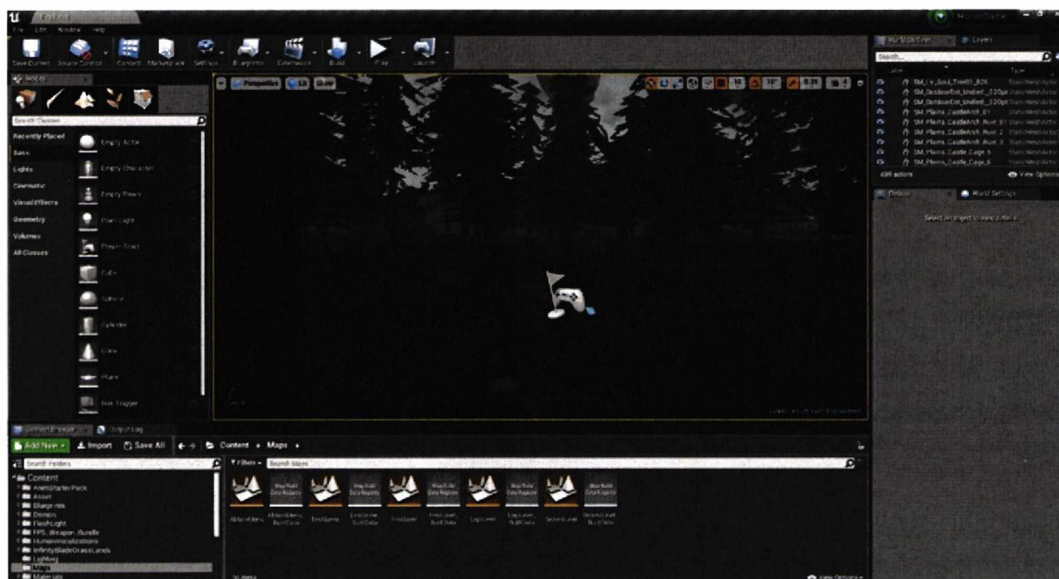
Η δεύτερη έκδοση της Unreal Engine, η Unreal Engine 2 (UE2), έκανε την πρώτη της εμφάνιση το 2002 με το παιχνίδι America's Army, ένα δωρεάν παιχνίδι πυροβολισμού για πολλούς παίκτες το οποίο αναπτύχθηκε από το στρατό των Η.Π.Α. ως εργαλείο επιλογής νεοσυλλέκτων. Σε αυτή την έκδοση το λογισμικό φωτοαπόδοσης γράφτηκε από την αρχή, καθώς παρουσιάστηκε και το UnrealEd 2 και λίγο αργότερα ακολούθησε το UnrealEd 3. Επιπλέον έγινε προσθήκη υποστήριξης για το Xbox.

Η ανάπτυξη της τρίτης έκδοσης της Unreal ανακοινώθηκε το 2004, όπου η μηχανή βρισκόταν υπό ανάπτυξη. Η Unreal Engine 3 (UE3) σχεδιάστηκε για να εκμεταλλευτεί πλήρως το προγραμματιζόμενο shader (παραγωγή κατάλληλων επιπέδων φωτεινότητας, σκοτεινότητας και χρώματος μέσα σε μια εικόνα).

Σημαντική ήταν η προσθήκη υποστήριξης του Android, του iOS και του Wii U. Τα δημοφιλέστερα παιχνίδια για UE3 ήταν το Batman: Arkham City και το Aliens: Colonial Marines.

Η τέταρτη έκδοση της Unreal Engine, η Unreal Engine 4 (UE4), έκανε την πρώτη της εμφάνιση το 2012. Η UE4 στοχεύει στην όγδοη γενιά κονσόλων, υπολογιστών και κινητών συσκευών. Η βασική προσθήκη της ήταν το global illumination (καθολικός φωτισμός) πραγματικού χρόνου. Μια άλλη σημαντική προσθήκη ήταν το νέο visual scripting system (γραφικό σύστημα προγραμματισμού), που ονομάζεται Blueprints, το οποίο επιτρέπει την ανάπτυξη ενός παιχνιδιού χωρίς να χρειάζεται να γίνει άμεση χρήση της γλώσσας προγραμματισμού C++. Με την χρήση των Blueprints δεν βελτιώνεται μόνο η απόδοση των προγραμματιστών, αλλά επιτυγχάνεται και η καλύτερη συνεργασία μεταξύ αυτών και των σχεδιαστών. Τα Blueprints αναλύονται στο κεφάλαιο 5.

1.4 Περιβάλλον της UE4



Εικόνα (Βασική διεπαφή χρήστη)

Η βασική διεπαφή χρήστη της UE4 φαίνεται στην παρακάτω Εικόνα.

Στο κέντρο της Εικόνας φαίνεται το viewport (παράθυρο προβολής), μέσα από το οποίο ο χρήστης μπορεί να επεξεργαστεί τα αντικείμενα του επιπέδου. Στο πάνω μέρος του viewport φαίνονται ορισμένα εργαλεία, τα οποία επιτρέπουν την ευκολότερη διαχείριση και επεξεργασία του επιπέδου.

Περισσότερα εργαλεία επεξεργασίας των στοιχείων του Level Editor υπάρχουν στον πίνακα Modes, ο οποίος βρίσκεται στα αριστερά της Εικόνας.



Διαθέτει μια συλλογή στοιχειωδών αντικειμένων που μπορούν να τοποθετηθούν στο επίπεδο. Για παράδειγμα έναν κύβο ή ένα στοιχείο φωτισμού.



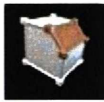
Διαθέτει διάφορα εργαλεία ζωγραφικής και την δυνατότητα χρωματισμού των τρισδιάστατων μοντέλων απευθείας μέσω του viewport.



Διαθέτει εργαλεία για την επεξεργασία του Landscape (τοπίων του εδάφους).



Διαθέτει εργαλεία που διευκολύνουν την μαζική τοποθέτηση συγκεκριμένων αντικειμένων μέσα στο επίπεδο. Για παράδειγμα μπορεί να χρησιμοποιηθεί για την δημιουργία δασών.



Διαθέτει κάποια βασικά εργαλεία επεξεργασίας τρισδιάστατων γεωμετρικών σχημάτων.

Κάτω αριστερά βρίσκεται ο content browser (περιηγητής περιεχομένου), μέσω του οποίου δημιουργούνται, εισάγονται, οργανώνονται, προβάλλονται και τροποποιούνται τα περιεχόμενα του project (έργο). Επιπλέον περιλαμβάνει βασικές επιλογές όπως, η μετονομασία, η μετακίνηση και η αντιγραφή φακέλων και αρχείων. Ένα ακόμα χρήσιμο εργαλείο που προσφέρει είναι η αναζήτηση, μέσω της οποίας μπορεί να επιτευχθεί η εύρεση οποιοδήποτε στοιχείου που περιέχεται στο παιχνίδι.

Πάνω δεξιά φαίνεται ο πίνακας World Outliner, ο οποίος εμφανίζει όλους τους Actor (αντικείμενα επιπέδου) μέσα στη σκηνή σε μορφή ιεραρχικού δέντρου. Οι Actors μπορούν μέσω του World Outliner να επιλεγθούν και να τροποποιηθούν απευθείας.

2. Τίτλος

Ο τίτλος του παιχνιδιού είναι **“Purgatory”**. Στα ελληνικά ο όρος Purgatory σημαίνει καθαρτήριο ψυχών, το οποίο είναι ο ενδιάμεσος χώρος ανάμεσα στην κόλαση και τον παράδεισο.

3. Επισκόπηση παιχνιδιού

Το παιχνίδι αφηγείται την ιστορία ενός χαρακτήρα, ο οποίος έχει χάσει την ψυχή του και έχει αποβιώσει. Το ταξίδι του ξεκινάει στην μεταθανάτιο ζωή, όπου ξυπνάει στο Purgatory (καθατήριο των ψυχών). Ο μόνος τρόπος για να σωθεί είναι να κερδίσει πίσω την ψυχή του. Αυτό δεν είναι μια εύκολη διαδικασία στο Purgatory. Ο χαρακτήρας καλείται να αντιμετωπίσει δαίμονες και να λύσει γρίφους, προκειμένου να πετύχει το σκοπό του. Η ιστορία ολοκληρώνεται μόνο όταν βρει και τα δύο κομμάτια της ψυχής του.

3.1 Είδος παιχνιδιού

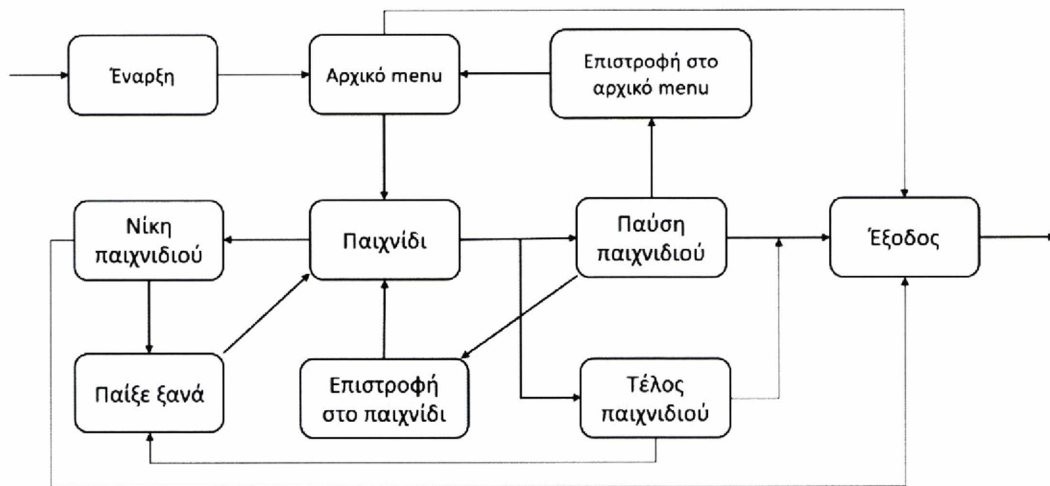
Το παιχνίδι κατατάσσεται στις ακόλουθες κατηγορίες:

- 1) Επιβίωσης (Survival)
- 2) Τρόμου (Horror)
- 3) Μυστηρίου (Mystery)

3.2 Στοχευμένο κοινό

Το ακόλουθο παιχνίδι αναφέρεται σε ηλικίες άνω των 16 ετών, γιατί έχει σκοπό να δημιουργήσει αισθήματα τρόμου και φόβου. Το περιβάλλον του παιχνιδιού και πολλά από τα mechanics (μηχανική του παιχνιδιού) έχουν σχεδιαστεί με τρόπο που εξυπηρετούν αυτόν τον σκοπό.

3.3 Διάγραμμα ροής παιχνιδιού



Διάγραμμα ροής παιχνιδιού

3.4 Βασική εμφάνιση και αίσθηση του Παιχνιδιού

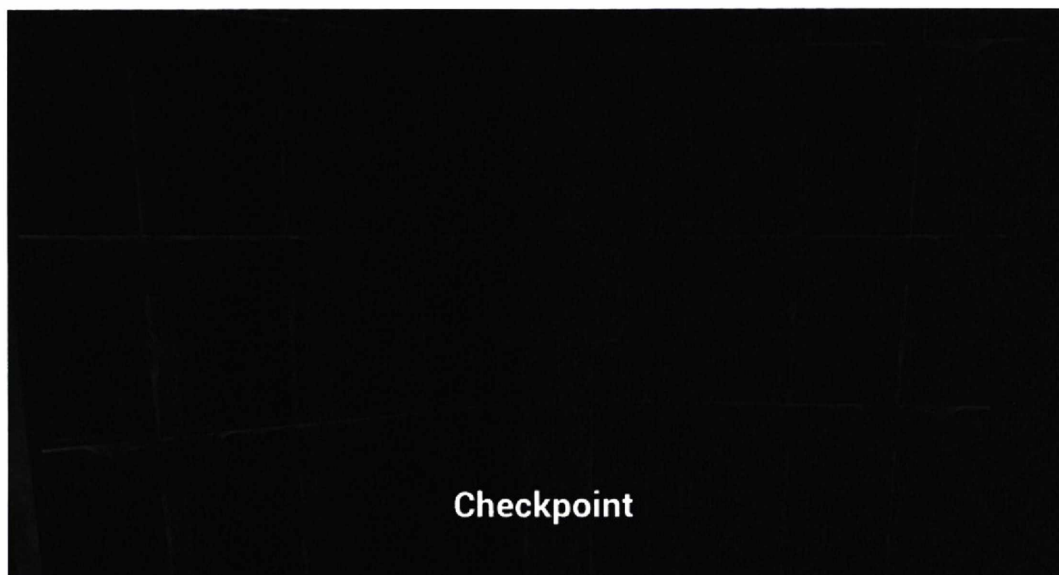
Στην έναρξη του παιχνιδιού παρουσιάζεται ένα σκοτεινό δάσος και ακούγεται μια μυστήρια μουσική, που σκοπό έχει την δημιουργία ανάμεικτων συναισθημάτων σύγχυσης και μυστηρίου. Το κλίμα μυστηρίου ενισχύεται λόγω της ύπαρξης διάφορων γρίφων. Η λύση των γρίφων απαιτεί την εξερεύνηση του περιβάλλοντος, η οποία ενδυναμώνει το μυστήριο κλίμα. Πολλές φορές κατά την εξερεύνηση του περιβάλλοντος ο χαρακτήρας αιφνιδιάζεται από την απότομη εμφάνιση δαιμόνων. Όχι μόνο οι απότομες εμφανίσεις των δαιμόνων, αλλά και η αποκρουστική τους εμφάνιση δημιουργούν επιπλέον συναισθήματα άγχους, τρόμου και φόβου. Η δυναμική αλλαγή της μουσικής του παιχνιδιού βοηθά στην κορύφωση των παραπάνω συναισθημάτων.

4. Διεπαφή του χρήστη (Interface)

Η γραφική διεπαφή ενημερώνει τον χρήστη σχετικά με την κατάσταση του παιχνιδιού. Για παράδειγμα τον ενημερώνει για την πρόοδο της λύσης των γρίφων και επιδεικνύει στον χρήστη συγκεκριμένες αλληλεπιδράσεις που μπορεί να έχει με τα αντικείμενα του περιβάλλοντος. Ακόμα, υπάρχουν menu (κατάλογος επιλογών) που επιτρέπουν στον χρήστη τη διαχείριση της εμφάνισης και της εξέλιξης του

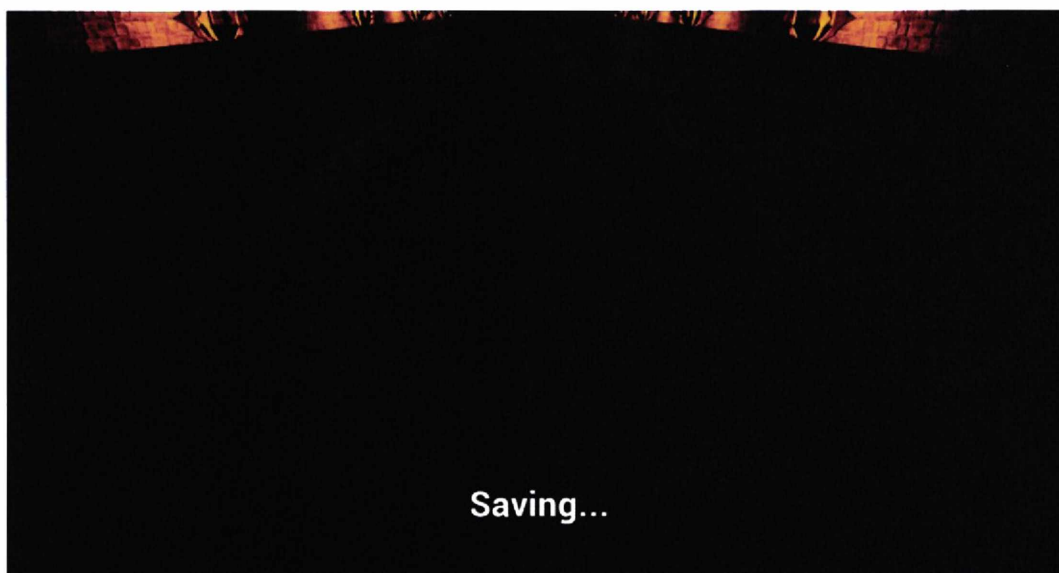
παιχνιδιού. Στην συνέχεια παρουσιάζονται στιγμιότυπα από την βασική διεπαφή του παιχνιδιού.

1) Όταν υπάρχει checkpoint.



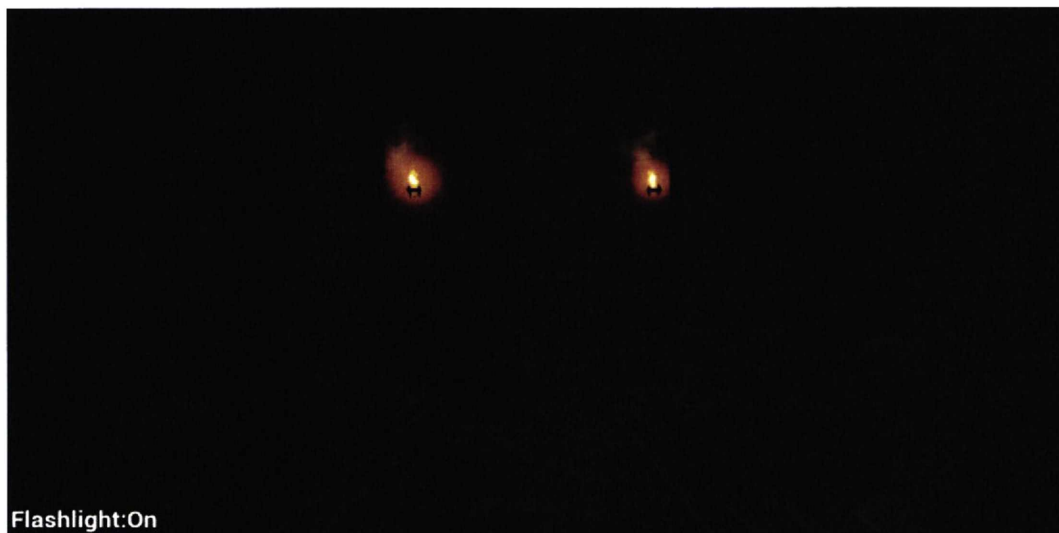
Checkpoint

2) Όταν αποθηκεύεται μια θέση του παιχνιδιού

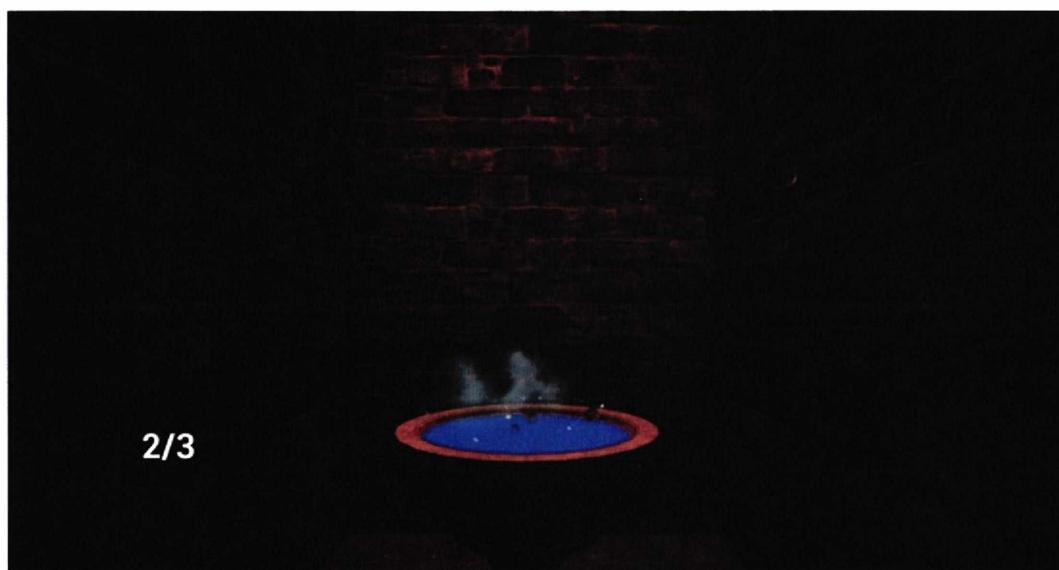


Save

3) Διάφορες πληροφορίες για τα αντικείμενα, όπως για παράδειγμα αν είναι ανοιχτός ο φακός ή πόσες ψυχές έχουν παραδοθεί (θα εξηγηθεί αναλυτικά το κομμάτι για τις ψυχές στο επόμενο υποκεφάλαιο).

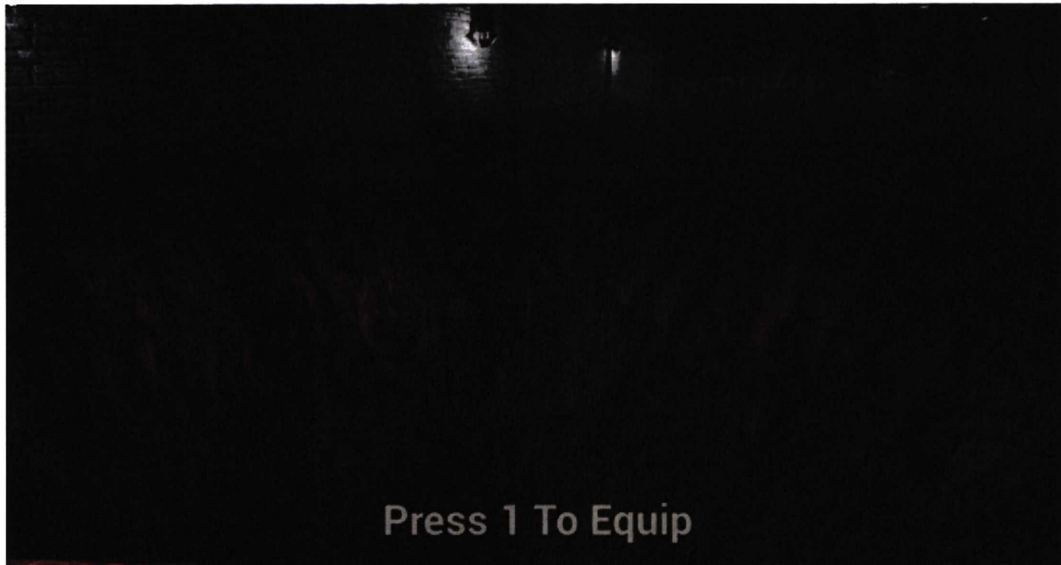


Λειτουργία φακού



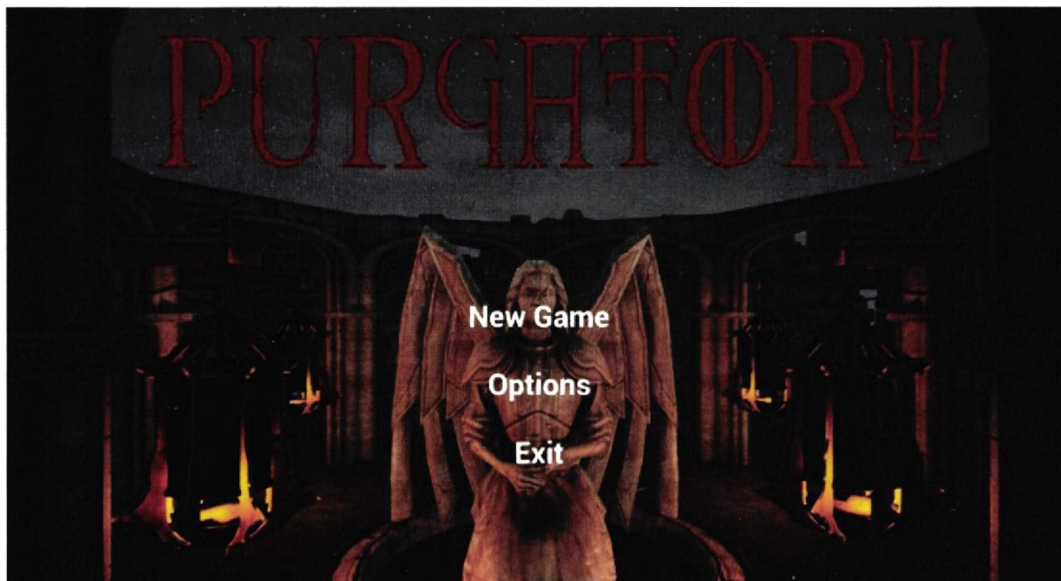
Ένδειξη τοποθετημένων ψυχών

5) Πληροφορίες για τον τρόπο χρήσης αντικειμένων.



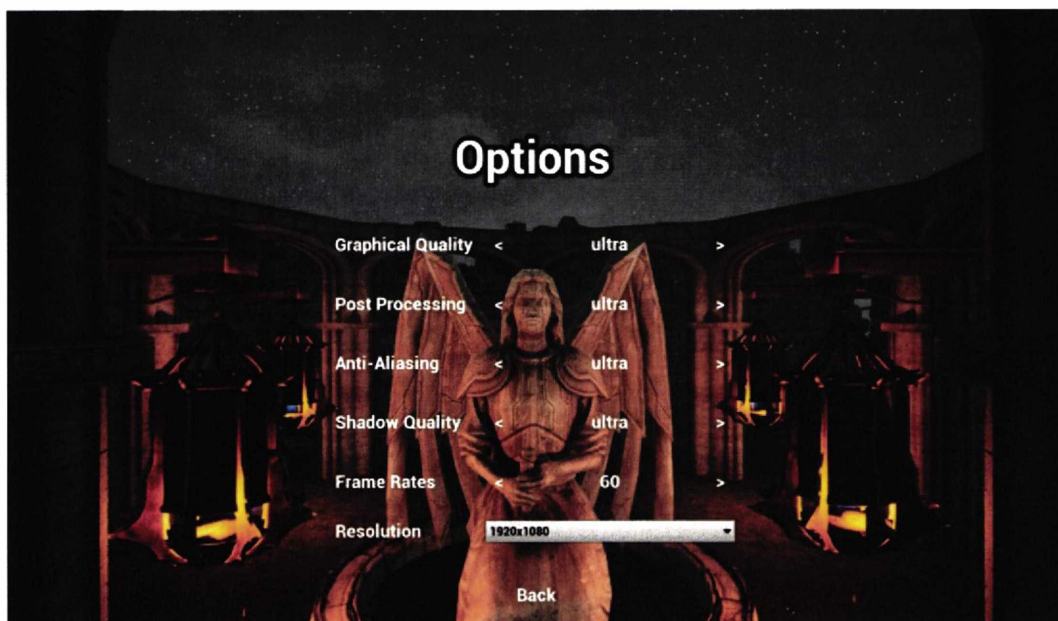
Χρήση Όπλου

6) Επιλογές στο αρχικό μενού.



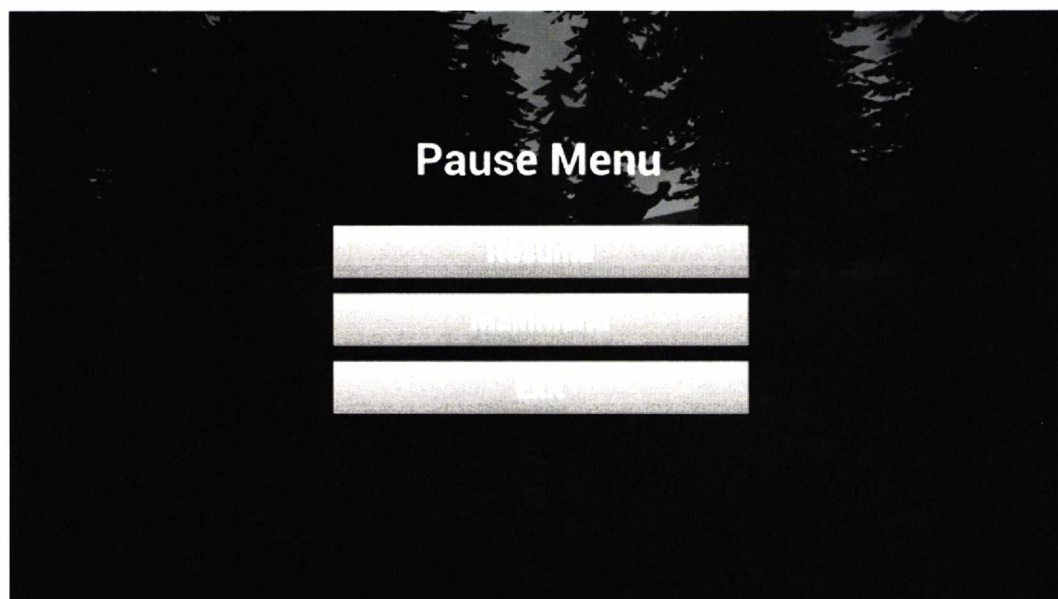
Αρχικό Menu

7) Επιλογές αλλαγής ρυθμίσεων του παιχνιδιού.



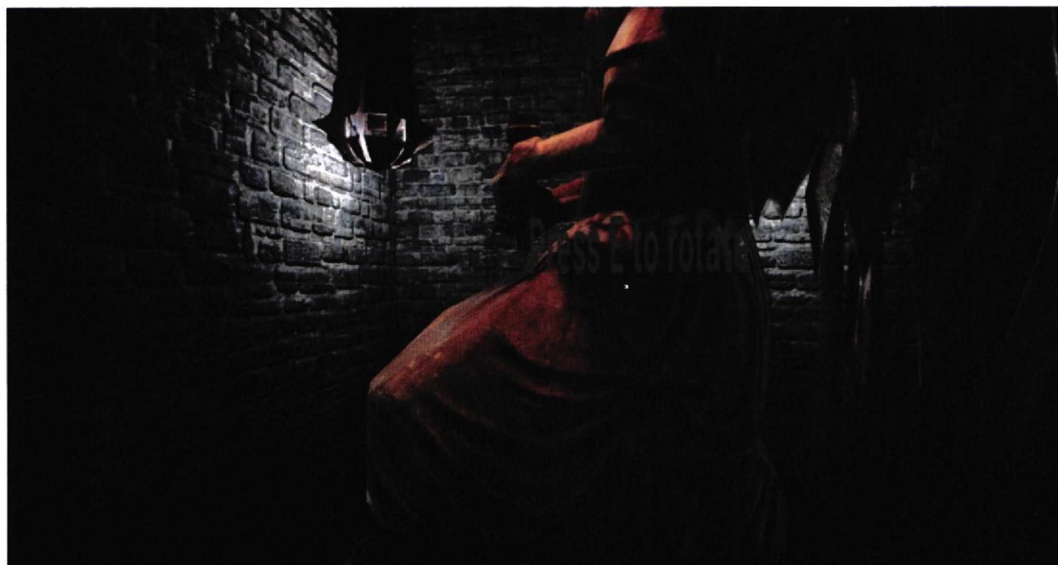
Ρυθμίσεις παιχνιδιού

8) Επιλογές κατά την παύση του παιχνιδιού.



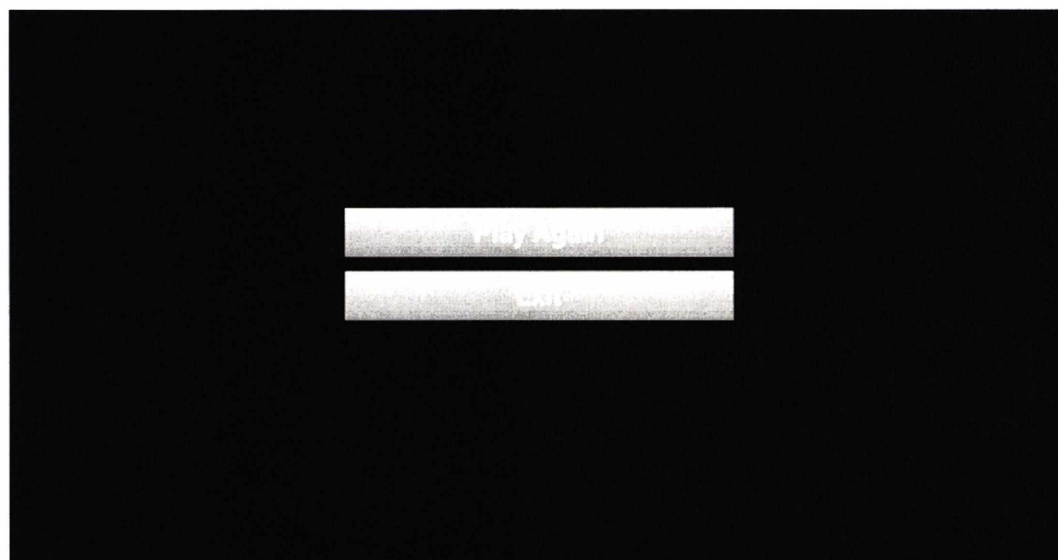
Μενυ παύσης

9) Πληροφορίες για αλληλεπίδραση με διάφορα αντικείμενα.



Περιστροφή αγάλματος

10) Όταν ο παίκτης πεθαίνει ή νικήσει το παιχνίδι



Μενου κατά το τέλος ή κατά την ολοκλήρωση του παιχνιδιού

4.1 Game controls

W--> Κίνηση Μπροστά

S--> Κίνηση Πίσω

A--> Κίνηση Αριστερά

D--> Κίνηση Δεξιά

E--> Αλληλεπίδραση με αντικείμενα

Q--> Αλληλεπίδραση με αντικείμενα

Space--> Αναπήδηση

Αριστερό Κλικ--> Αλληλεπίδραση με αντικείμενα και επίθεση

Κίνηση Ποντικιού--> Κίνηση Κάμερας

Esc--> Menu

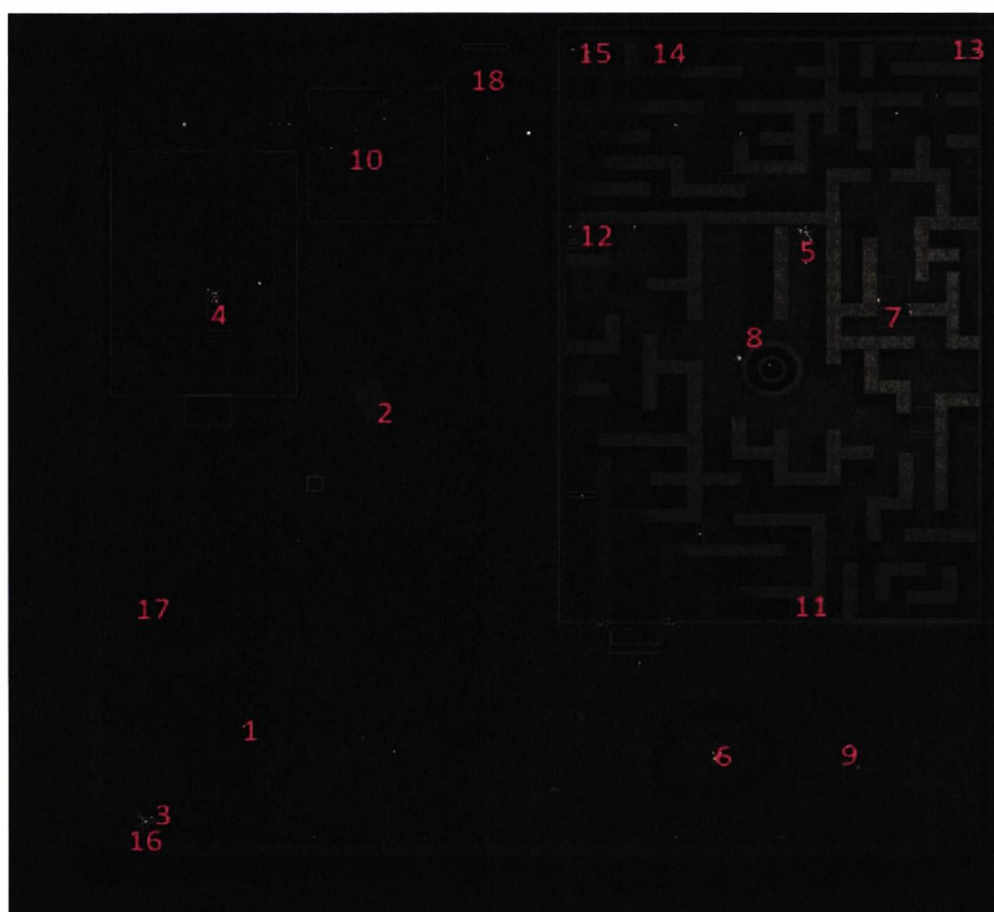
4.2 Ανάλυση gameplay

Επίπεδο 1:

Κατά την εκκίνηση του παιχνιδιού ο χαρακτήρας βρίσκεται στο νούμερο 1 της Εικόνας 1, όπου ξυπνάει σε ένα άλσος.

Χωρίς να γνωρίζει που βρίσκεται, αρχίζει να εξερευνεί τον χώρο και πρέπει να κατευθυνθεί στην θέση 2. Στην θέση 2 βρίσκει ένα γράμμα που περιέχει έναν γρίφο, η λύση του οποίου υποδεικνύει τις θέσεις που βρίσκονται τα αγάλματα (νούμερα 3, 4 και 5). Όταν λυθεί ο γρίφος αρχίζει να βρέχει, αυτό έχει ως αποτέλεσμα να δημιουργηθεί μια λίμνη στην θέση 6. Στην συνέχεια υψώνεται η γέφυρα που υπήρχε στον πάτο της λίμνης, η οποία επιτρέπει στον χαρακτήρα να πάει στο κέντρο της και να λάβει ένα όπλο. Σε περίπτωση που πάει να πάρει το όπλο χωρίς να βρέχει πέφτει μέσα στον άδειο λάκκο, με αποτέλεσμα ο χαρακτήρας να πεθάνει. Αν ο χαρακτήρας πεθάνει τότε το παιχνίδι θα ξεκινήσει από την αρχή. Δίπλα από το όπλο υπάρχει ένα γράμμα, το οποίο προειδοποιεί τον παίκτη ότι η επαναλαμβανόμενη χρήση του θα οδηγήσει στον θάνατο του χαρακτήρα. Με το όπλο αυτό πρέπει να πάει στο νούμερο 7 και να το χρησιμοποιήσει για να ρίξει την πόρτα, ώστε να μπορέσει να εξερευνήσει τον υπόλοιπο λαβύρινθο.

Μέσα στον λαβύρινθο που φαίνεται στην εικόνα, υπάρχουν βωμοί στα νούμερα 11, 12 και 13. Σε αυτούς τους βωμούς πρέπει να τοποθετήσει από μια ψυχή. Τις ψυχές τις βρίσκει στα σημεία 8, 9 και 10. Αφού παραδώσει τις ψυχές, θα κινηθεί ο τοίχος στο νούμερο 14 και θα μπορέσει να πάει στο νούμερο 15. Εκεί θα βρει ένα βάζο με λουλούδια και ένα γράμμα που θα γράφει την τοποθεσία στην οποία θα πρέπει να τα παραδώσει. Αυτή η τοποθεσία είναι στο νούμερο 16. Στο νούμερο 15 υπάρχει ένα checkpoint. Μέχρι να βγει από τον λαβύρινθο εμφανίζονται δαίμονες σε μερικά σημεία. Οι δαίμονες προσπαθούν να χτυπήσουν τον παίκτη, ο οποίος αν δεν τους αποφύγει θα πεθάνει. Στην περίπτωση που πεθάνει, εμφανίζεται πάλι στην θέση 15. Αν τα καταφέρει, πέφτει ένας κεραυνός και εμφανίζεται ένα κομμάτι από την ψυχή του παίκτη στο νούμερο 17. Για την ολοκλήρωση του επιπέδου 1 ο παίκτης πρέπει να περάσει από την πύλη που βρίσκεται στο νούμερο 18.



Εικόνα 1 (Πανοραμική απεικόνιση Level1)

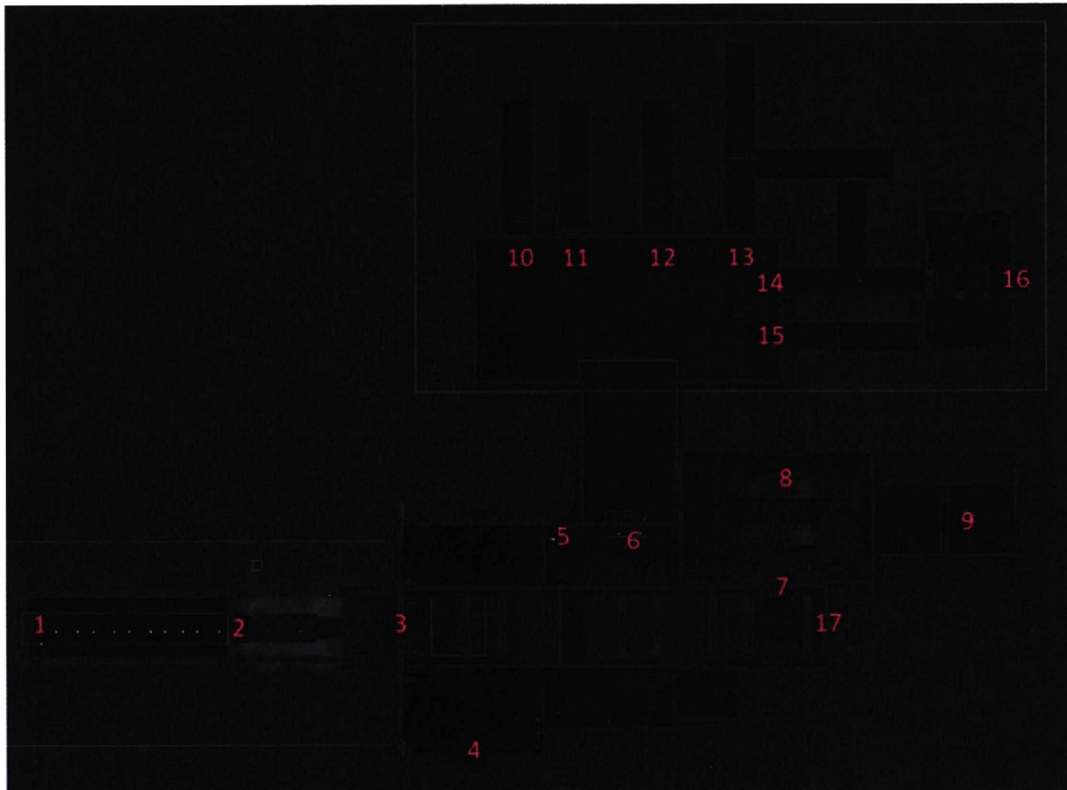
Επίπεδο 2:

Κατά την εκκίνηση του επιπέδου 2 ο χαρακτήρας βρίσκεται στο νούμερο 1 της Εικόνας 2.

Ο χρήστης καλείται να φτάσει στο νούμερο 2 πηδώντας από πλάκα σε πλάκα ενώ αυτές κινούνται. Αν ο χαρακτήρας πέσει στο κενό, τότε πεθαίνει και το επίπεδο 2 ξεκινάει από την αρχή. Στην συνέχεια ο χαρακτήρας πάει στο νούμερο 3 και εισέρχεται μέσα στο κάστρο. Στο νούμερο 4 βρίσκει έναν φακό. Επιπλέον, βρίσκει και ένα γράμμα που τον ενημερώνει ότι θα χρειαστεί αυτόν τον φακό στην συνέχεια του επιπέδου. Στο νούμερο 5 υπάρχει ένα γράμμα που αναφέρεται στο δωμάτιο στο νούμερο 6. Στο συγκεκριμένο δωμάτιο δεν μπορεί να εισέλθει, διότι χρειάζεται ένα κλειδί.

Όταν ο παίκτης φτάσει στο νούμερο 7, εισέρχεται σε μία βιβλιοθήκη. Στο νούμερο 8 βρίσκει ένα γράμμα, το οποίο αναφέρεται στην λύση ενός γρίφου. Ο γρίφος αυτός σχετίζεται με 3 βιβλία μέσα στην βιβλιοθήκη. Όταν λύσει τον γρίφο, θα κινηθεί ένας τοίχος κι έτσι θα μπορέσει να πάει στο δωμάτιο στο νούμερο 9. Εκεί θα βρει ένα γράμμα και το κλειδί που έψαχνε για να εισέλθει στο δωμάτιο με το νούμερο 6. Όταν πάρει το κλειδί και πάει να φύγει θα εμφανιστεί ένας δαίμονας που θα τρέχει καταπάνω του. Ο τρόπος για να αντιμετωπιστεί ο δαίμονας αναγράφεται στο γράμμα που υπήρχε δίπλα από το κλειδί. Αν δεν αντιμετωπιστεί σωστά ο δαίμονας, τότε ο χαρακτήρας πεθαίνει και ξεκινάει από την αρχή το επίπεδο 2. Αν τα καταφέρει, κατευθύνεται στο δωμάτιο στο νούμερο 6.

Όταν φτάσει στο νούμερο 6, ανοίγει την πόρτα με το κλειδί και γίνεται ένα checkpoint. Μέσα στο δωμάτιο υπάρχουν 6 διάδρομοι στα νούμερα 11, 12, 13, 14, 15 και 17, από τους οποίους μόνο ένας είναι ο σωστός. Σε κάθε διάδρομο θα αντιμετωπιστούν από έναν έως τρεις δαίμονες. Ο σωστός διάδρομος είναι αυτός στο νούμερο 15, ο οποίος καταλήγει σε ένα νέο δωμάτιο. Στο νέο δωμάτιο στο νούμερο 16 βρίσκεται το δεύτερο και τελευταίο κομμάτι της ψυχής του. Για την ολοκλήρωση του επιπέδου 2 ο παίκτης πρέπει να περάσει από την πύλη που βρίσκεται στο νούμερο 17. Η ολοκλήρωση του επιπέδου 2 σημαίνει και ολοκλήρωση του παιχνιδιού.

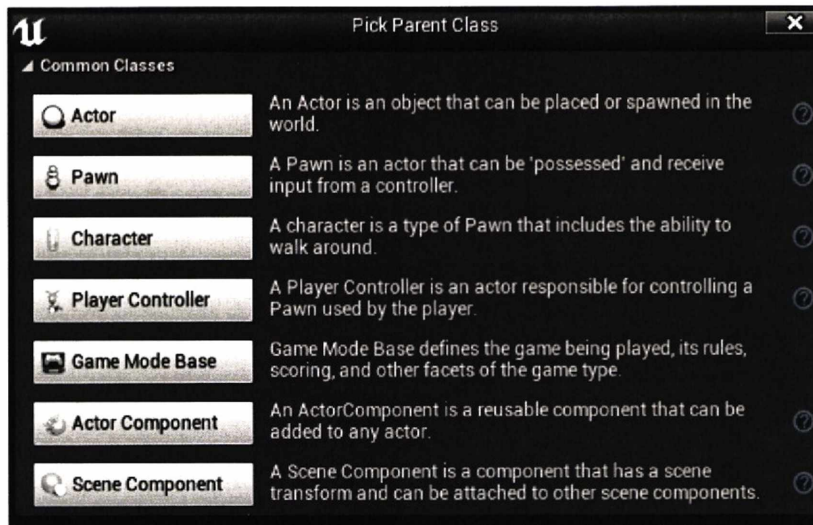


Εικόνα 2 (Πανοραμική απεικόνιση Level2)

5. Blueprints

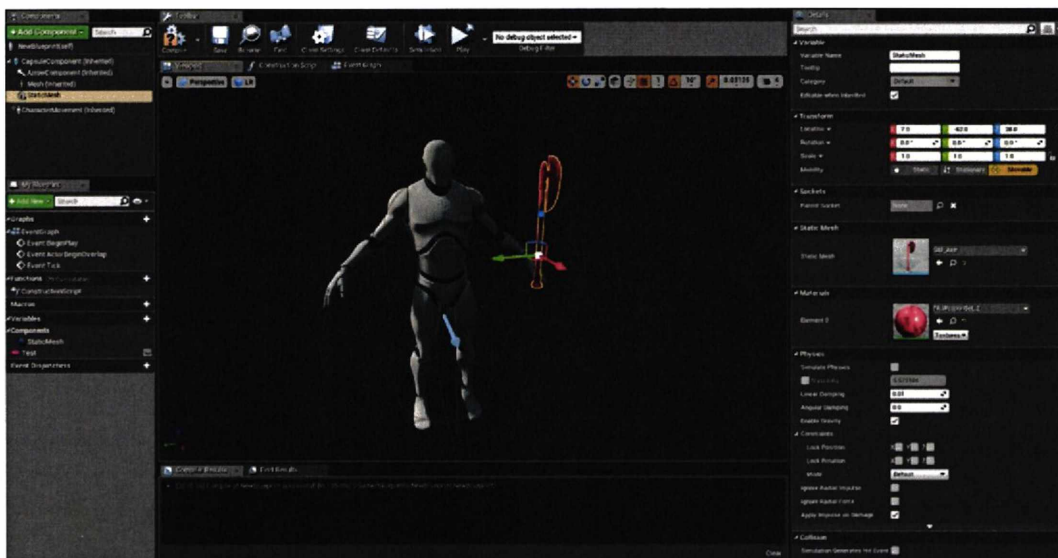
Όπως αναφέρθηκε κατά την εισαγωγή τα blueprint (BP) είναι ένα γραφικό σύστημα προγραμματισμού, το οποίο μπορεί να χρησιμοποιηθεί στην UE4. Στην παρακάτω εικόνα (Εικόνα 3) φαίνονται οι επιλογές που δίνονται για τις τάξεις (classes) των blueprints που μπορούν να δημιουργηθούν. Στο παιχνίδι που αναπτύχθηκε, χρησιμοποιήθηκαν οι ακόλουθες τάξεις.

- 1) Actor. Ένα αντικείμενο το οποίο μπορεί να τοποθετηθεί στο περιβάλλον του παιχνιδιού.
- 2) Character. Ο χαρακτήρας που χρησιμοποιείται στο παιχνίδι, ο οποίος έχει την δυνατότητα να μετακινείται μέσα στον χώρο.
- 3) Game Mode Base. Χρησιμοποιείται για να τεθούν οι κανόνες του παιχνιδιού, για παράδειγμα ποιος είναι ο χαρακτήρας που θα χρησιμοποιείται κατά την διάρκεια του παιχνιδιού.
- 4) Level Blueprint (δεν φαίνεται στην εικόνα). Είναι το Blueprint που ασχολείται με τα γεγονότα του επιπέδου, τα οποία είναι ανεξάρτητα από τις ενέργειες του χαρακτήρα.



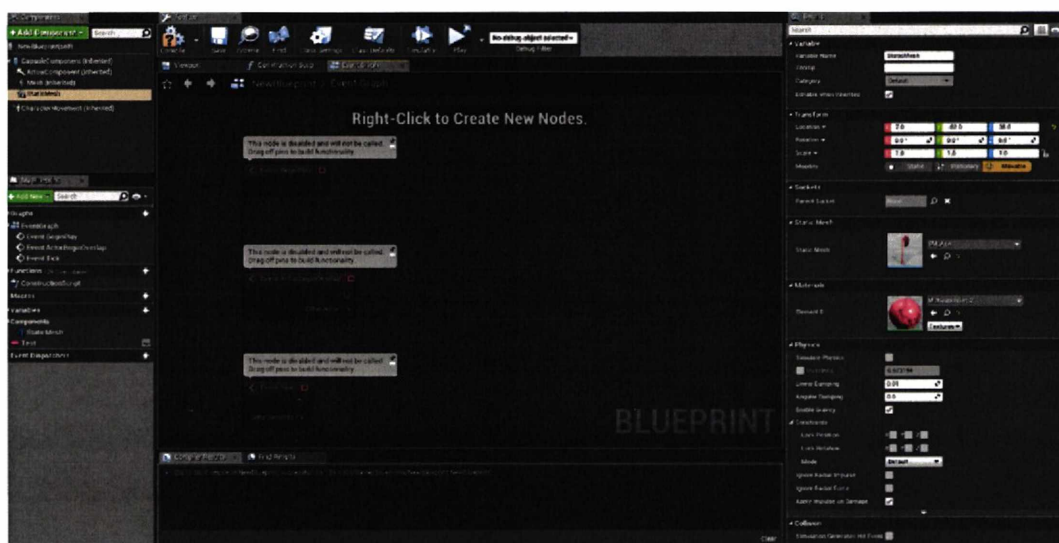
Εικόνα 3 (Τάξεις blueprints)

Κάθε blueprint διαθέτει ένα viewport (Εικόνα 4), μέσα από το οποίο επεξεργάζονται τα περιεχόμενα του. Για παράδειγμα, σε ένα character blueprint υπάρχει το κουμπί add component πάνω αριστερά, το οποίο δίνει την δυνατότητα προσθήκης ενός Static Mesh (τρισδιάστατο πλέγμα που αναπαριστά ένα αντικείμενο). Στην εικόνα φαίνεται ένα παράδειγμα Static Mesh (όπλο). Στον δεξί πίνακα αναγράφονται τα διάφορα χαρακτηριστικά που έχει κάθε component και δίνεται η δυνατότητα επεξεργασίας τους.



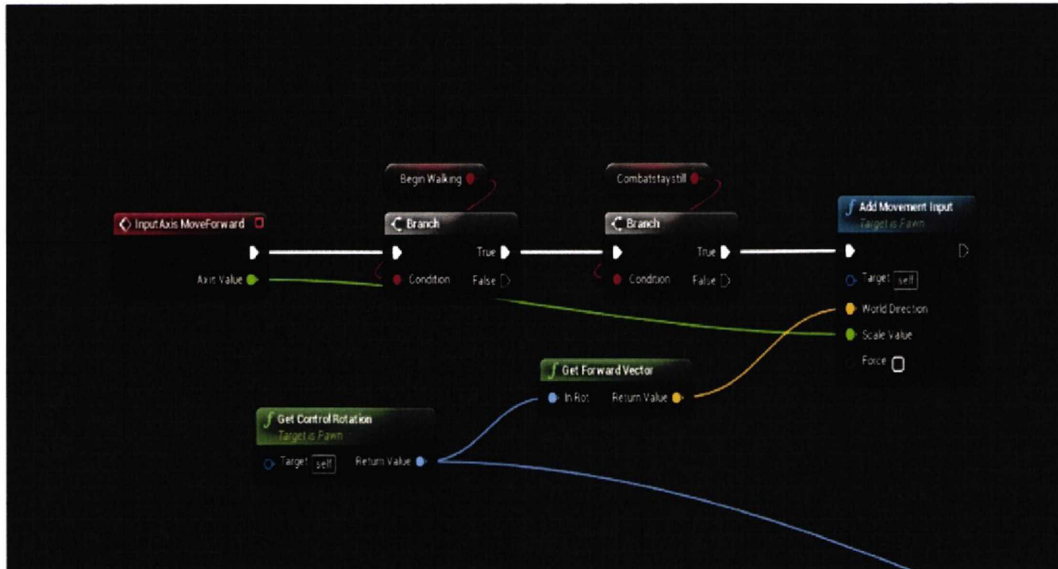
Εικόνα 4 (Viewport του Survivor BP)

Παρακάτω φαίνεται το event graph (γράφημα συμβάντων) ενός blueprint (Εικόνα 5). Τα τρία κουτάκια στο κέντρο ονομάζονται nodes και είναι ουσιαστικά τμήματα κώδικα C++. Τα nodes μπορεί να είναι μια μεταβλητή, μια μαθηματική πράξη, μια Δομή Αν ή μία συνάρτηση και γενικά οτιδήποτε χρησιμοποιείται σε μία γλώσσα προγραμματισμού. Νέα nodes μπορούν να δημιουργηθούν πατώντας δεξί κλικ και επιλέγοντας αυτό που χρειάζεται. Στην προκειμένη περίπτωση φαίνονται οι τρεις βασικές συναρτήσεις που χρησιμοποιούνται κατά την δημιουργία ενός παιχνιδιού. Η Event BeginPlay καλείται με το που ξεκινήσει το παιχνίδι. Η Event ActorBeginOverlap καλείται κάθε φορά που το είδος του blueprint που δημιουργήθηκε, έρθει σε επαφή με κάποιο αντικείμενο ή χαρακτήρα μέσα στο παιχνίδι. Η Event tick καλείται σε κάθε Frame του παιχνιδιού. Κάτω αριστερά βλέπουμε την λέξη variable, η οποία έχει να κάνει με τις μεταβλητές που θα χρειαστεί να χρησιμοποιηθούν. Επιλέγοντας το εικονίδιο variable και πατώντας πάνω στο σύμβολο + δημιουργείται μια μεταβλητή. Πάνω δεξιά αναγράφονται οι τύποι μεταβλητών που μπορούν να δημιουργηθούν. Παρακάτω θα αναλυθούν τα διάφορα blueprints που χρησιμοποιήθηκαν για την δημιουργία μερικών mechanics.



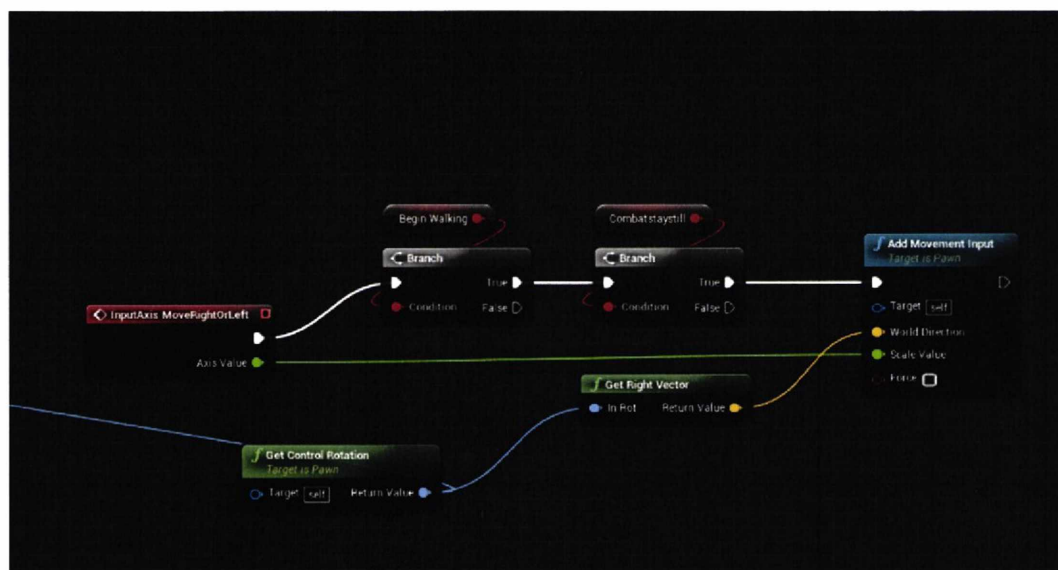
Εικόνα 5 (Editor του Survivor BP)

πλήκτρο **W** (πατημένο), ενώ η κίνηση προς τα πίσω ξεκινάει με το **S** (πατημένο). Ανάλογα με την τιμή του scale, το forward vector θα είναι προς τα πίσω (εφόσον είναι αρνητικό το scale) ή προς τα μπροστά (εφόσον είναι θετικό το scale).



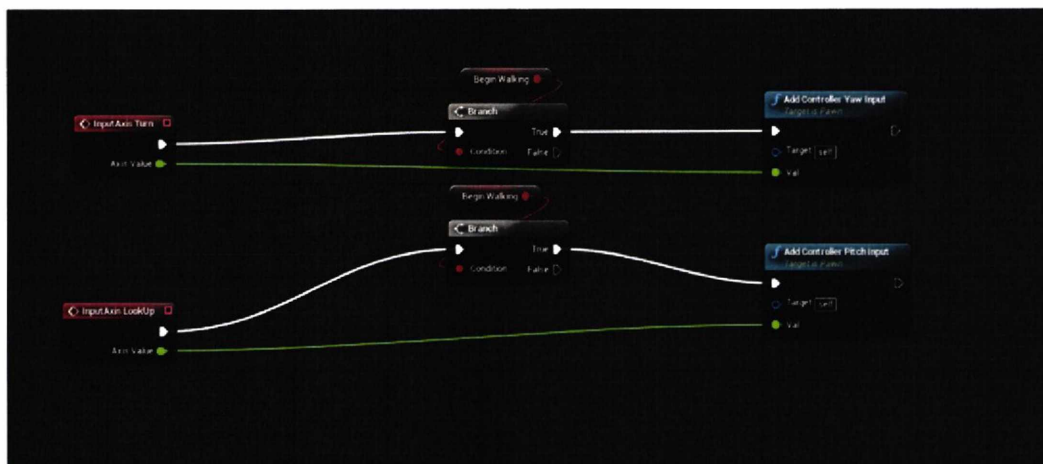
Εικόνα 7 (Κίνηση χαρακτήρα μπροστά και πίσω στο Survivor BP)

Στην Εικόνα 8 φαίνεται πως γίνεται η κίνηση του χαρακτήρα προς τα δεξιά και αριστερά. Ο τρόπος είναι ίδιος με τον προηγούμενο, με την διαφορά ότι εδώ, αντί να παρθεί το Forward Vector, λαμβάνεται το Right Vector που δίνει την κατεύθυνση του χαρακτήρα προς τα δεξιά. Η κατεύθυνση ανάλογα με το αν η τιμή του scale είναι θετική ή αρνητική θα είναι προς τα αριστερά ή προς τα δεξιά πατώντας τα κουμπιά **A** και **D** αντίστοιχα.



Εικόνα 8 (Κίνηση χαρακτήρα αριστερά και δεξιά στο Survivor BP)

Στην Εικόνα 9 φαίνεται η κίνηση της κάμερα, η οποία υπολογίζεται με βάση τις συναρτήσεις Add Controller Yaw Input και Add Controller Pitch Input. Η Add Controller Yaw Input παράγει την περιστροφή της κάμερας προς τα δεξιά ή προς τα αριστερά (yaw) και η Add Controller Pitch Input παράγει την περιστροφή της κάμερας προς τα πάνω ή προς τα κάτω (pitch), ανάλογα με τις τιμές που επιστρέφουν τα Axis Value (Scale).



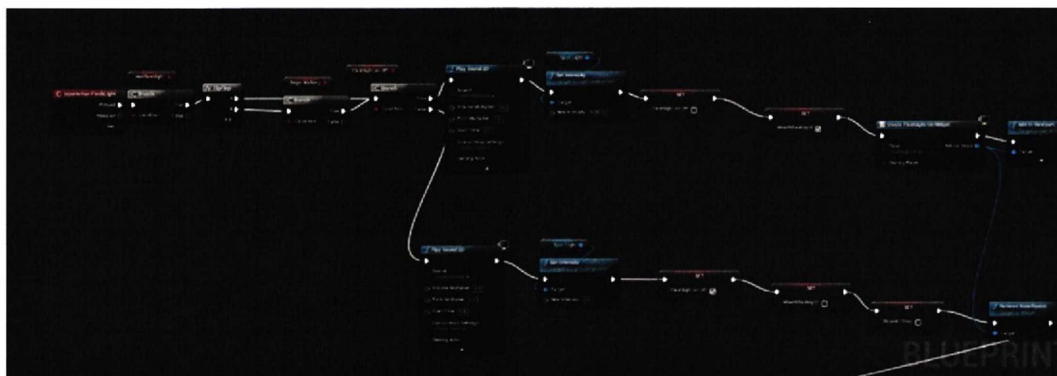
Εικόνα 9 (Κίνηση κάμερας του χαρακτήρα στο Survivor BP)

2) Άνοιγμα Φακού

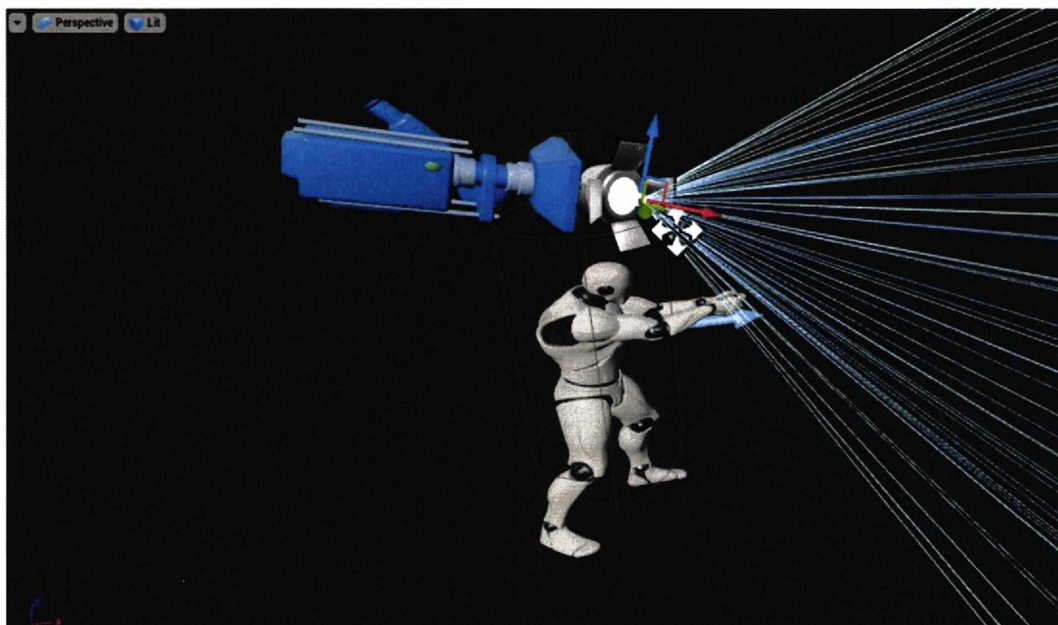
Στην Εικόνα 10 φαίνεται ο τρόπος με τον οποίο ανοιγοκλείνει ο φακός μέσα στο παιχνίδι.

Για αρχή η μεταβλητή spotlight είναι ένα component τύπου προβολέα (Spot Light) που έχει προστεθεί στον χαρακτήρα για να φωτίζει στην μπροστινή κατεύθυνση (Εικόνα 11). Το Event Input ActionFlashlight καλείται όταν πατιέται το πλήκτρο F. Στην αρχή Υπάρχουν 3 Δομές An, οι οποίες έχουν ως είσοδο τις μεταβλητές hasflashlight για το αν έχει βρεθεί ο φακός μέσα στο παιχνίδι, την begin walking για τον ίδιο λόγο που αναφέρθηκε στο mechanic 1 και την flashlight on off για το αν ο φακός είναι κλειστός ή ανοιχτός. Αν ο χρήστης έχει βρει τον φακό θα κληθεί η συνάρτηση Flip Flop που έχει εξόδους A και B ανάλογα με το πόσες φορές καλείται. Την πρώτη φορά που θα πατηθεί το F θα έχει έξοδο A και την δεύτερη φορά B, δηλαδή λειτουργεί εναλλάξ. Όταν θα έχει έξοδο A, καλείται η Δομή An, με την μεταβλητή flashlight on off (που είναι δηλωμένη ως αληθής εξαρχής) και η δομή θα επιστρέψει αληθής. Στην συνέχεια καλείται η συνάρτηση Play Sound 2D που παίζει έναν ήχο για να ακουστεί ότι άνοιξε ο φακός. Επιπλέον, τίθεται το Intensity της μεταβλητής Spot Light στην τιμή 5000 με την συνάρτηση Set Intensity. Η Set Intensity δίνει την δυνατότητα αυξομείωσης της φωτεινότητας του φακού. Αν η flashlight on off είναι ψευδής, τότε η Δομή An επιστρέφει ψευδής με αποτέλεσμα η

Play Sound2D να παίζει έναν αντίστοιχο ήχο και το Intensity της μεταβλητής Spot Light τίθεται ίσο με 0. Στην συνέχεια τίθεται η μεταβλητή flashlight on off αληθής στην πρώτη περίπτωση και ψευδής στην δεύτερη, ώστε να γίνεται εναλλαγή κάθε φορά που πατιέται το πλήκτρο F. Όταν ενεργοποιείται ο φακός καλείται η συνάρτηση Create flashlight on widget η οποία με την σειρά της καλεί την Add To Viewport με αποτέλεσμα να εμφανιστεί κάτω αριστερά στην οθόνη το γραφικό στοιχείο "Flashlight:ON" (το οποίο παρουσιάστηκε στο Κεφάλαιο 4). Όταν κλείσει ο φακός καλείται η συνάρτηση Remove From Parent που αφαιρεί από το viewport το γραφικό στοιχείο. Οι μεταβλητές hitwithflashlight και shoulddistor χρησιμοποιούνται για άλλους λόγους μέσα στο παιχνίδι και δεν επηρεάζουν τον φακό.



Εικόνα 10 (Άνοιγμα φακού στο Survivor BP)

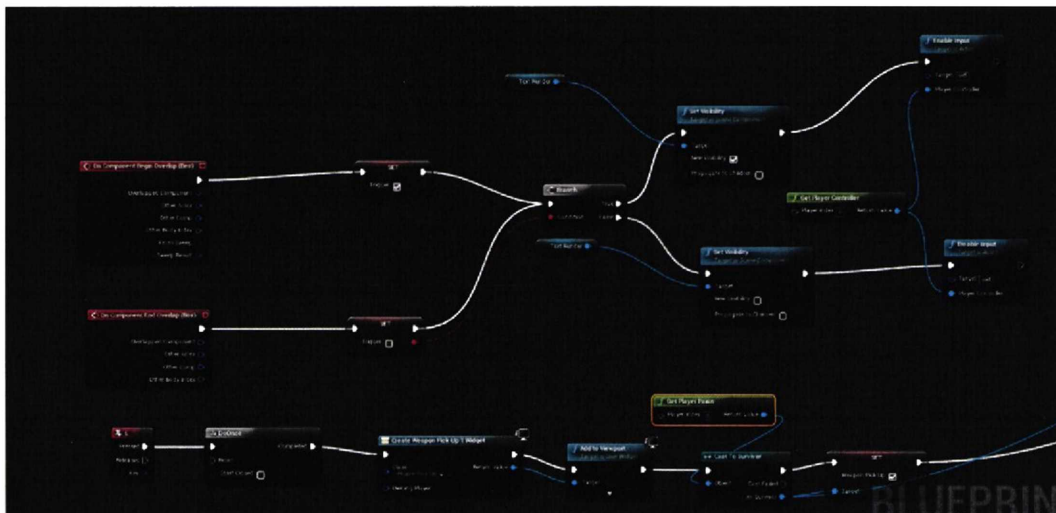


Εικόνα 11 (Viewport του Survivor BP και ένδειξη φακού)

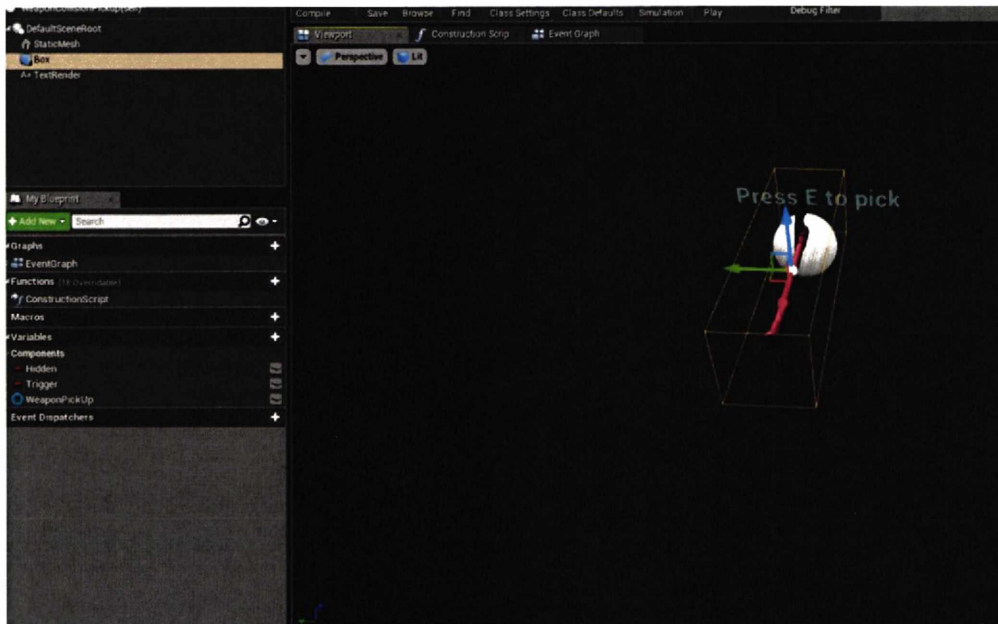
3) Pick Up αντικειμένων

Τα 2 Event στην Εικόνα 12 On Component Begin Overlap και On Component End Overlap καλούνται κάθε φορά που υπάρχει επαφή του χαρακτήρα με το component box (BP WeaponCollisionPickup Εικόνα 13) ή όταν χάνεται αντίστοιχα. Με το που έρθει σε επαφή, καλείτε το πρώτο Event και τίθεται η μεταβλητή trigger αληθής, ενώ γίνεται ψευδής όταν καλείται το δεύτερο event. Στην συνέχεια εφόσον η Δομή Αν είναι αληθής τίθεται το visibility της μεταβλητής text render αληθής. Σε περίπτωση που η Δομή Αν είναι ψευδής τίθεται το visibility της μεταβλητής text render σε ψευδής. Η text render είναι μια μεταβλητή τύπου text render component που εμφανίζεται στο σημείο που τοποθετείται ένα κείμενο (Press E to pick Εικόνα 13). Τέλος, καλούνται οι συναρτήσεις enable input και disable input όταν η Δομή Αν είναι αληθής ή ψευδής αντίστοιχα. Οι 2 αυτές συναρτήσεις έχουν ως είσοδο την συνάρτηση Get Player Controller που περιέχει τα input του χρήστη. Αυτό έχει ως αποτέλεσμα όταν ο χρήστης πατήσει κάποιο πλήκτρο να υπάρχει ανταπόκριση (Enable Input) ή όχι (Disable Input).

Εφόσον ο χρήστης μπορεί να πατήσει το πλήκτρο E, καλείται το Event E, το οποίο με την σειρά του θα καλέσει την συνάρτηση Do Once. Η Do Once σημαίνει πως όλα τα nodes μετά από αυτό θα τρέξουν μια φορά. Στη συνέχεια εμφανίζεται το γραφικό στοιχείο "Press 1 To Equip" στην οθόνη, καλώντας την create weapon pick up 1 widget και την συνάρτηση add to viewport. Τέλος, καλείται το node cast to Survivor (με αυτό τον τρόπο λαμβάνονται διάφορες πληροφορίες από το Blueprint του χαρακτήρα Survivor) για να τεθεί η μεταβλητή weapon pick up αληθής, η οποία βρίσκεται μέσα στο BP του Survivor (στο mechanic 4 θα εξηγηθεί γιατί γίνεται αληθής). Η cast to survivor έχει ως είσοδο την get player pawn η οποία με την σειρά της έχει ως είσοδο το νούμερο 0. Αυτό γίνεται για να γνωρίζει το BP ποιον χαρακτήρα να χειριστεί.



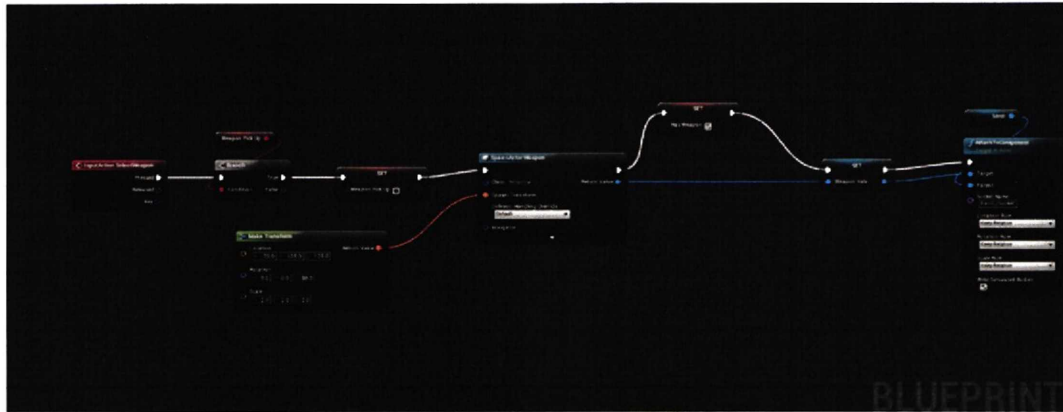
Εικόνα 12 (Pick Up όπλου στο WeaponCollisionPickup BP)



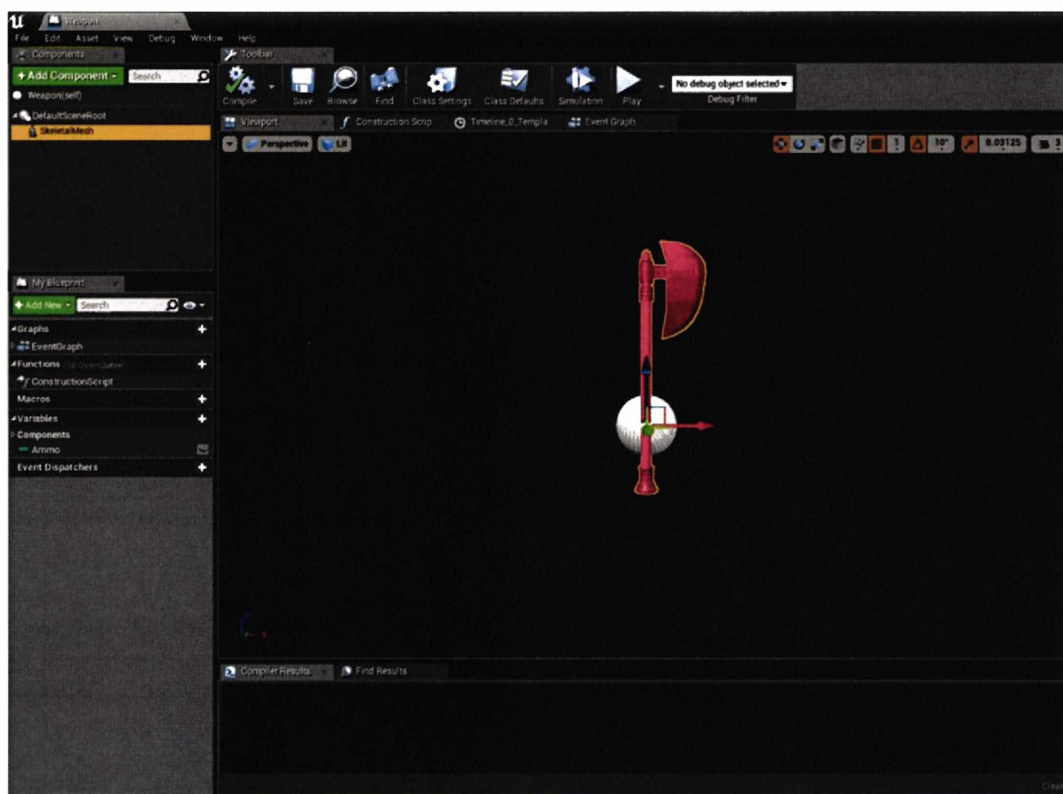
Εικόνα 13 (Viewport του WeaponCollisionPickup BP)

4) ΕQUIP ΟΠΛΟΥ

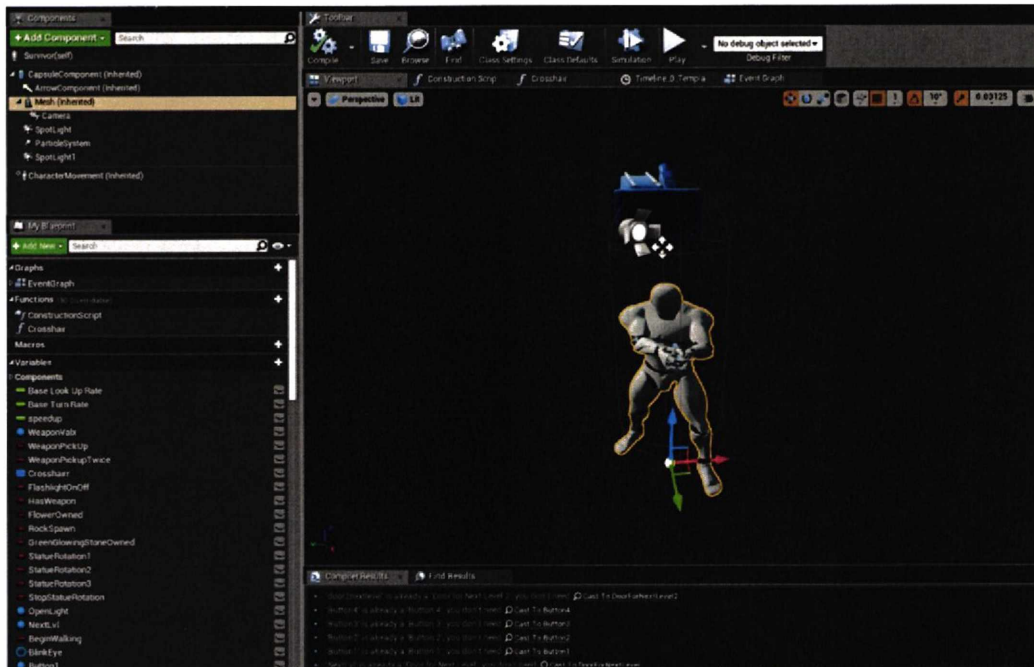
Το Event InputAction SelectWeapon καλείται όταν πατηθεί το πλήκτρο 1. Μετά φαίνεται ότι υπάρχει μια Δομή Αν, η οποία ελέγχει αν η μεταβλητή weapon pick up είναι αληθής. Αυτός είναι και ο λόγος που στο mechanic 3 τίθεται αληθής. Εφόσον η Δομή Αν είναι αληθής τίθεται η weapon pick up ψευδής, ώστε να μην εμφανίζεται πολλαπλές φορές το όπλο. Στην συνέχεια καλείται η Spawn Actor Weapon με την οποία θα εμφανιστεί το όπλο στο χαρακτήρα. Η Make Transform επιστρέφει το Location (θέση), το Rotation και το Scale (κλιμάκωση μεγέθους) του όπλου. Η έξοδος του Spawn Actor Weapon τίθεται στην μεταβλητή Weapon Valx (μεταβλητή τύπου Object που αναφέρεται στο BP weapon Εικόνα 15) και καλείται η συνάρτηση AttachToComponent. Η AttachToComponent παίρνει ως εισόδους το component Weapon Valx για να γνωρίζει ποιο θα είναι το αντικείμενο που θα εμφανίσει, την μεταβλητή mesh (component σκελετικού χαρακτήρα Εικόνα 16), ώστε να γνωρίζει σε ποιο component θα το εμφανίσει και το Socket Name hand_rSocket (δεξί χέρι χαρακτήρα Εικόνα 17) που είναι ουσιαστικά το ακριβές σημείο πάνω στο component του σκελετικού χαρακτήρα. Η μεταβλητή Has Weapon τίθεται αληθής και ο λόγος θα εξηγηθεί στο mechanic 5.



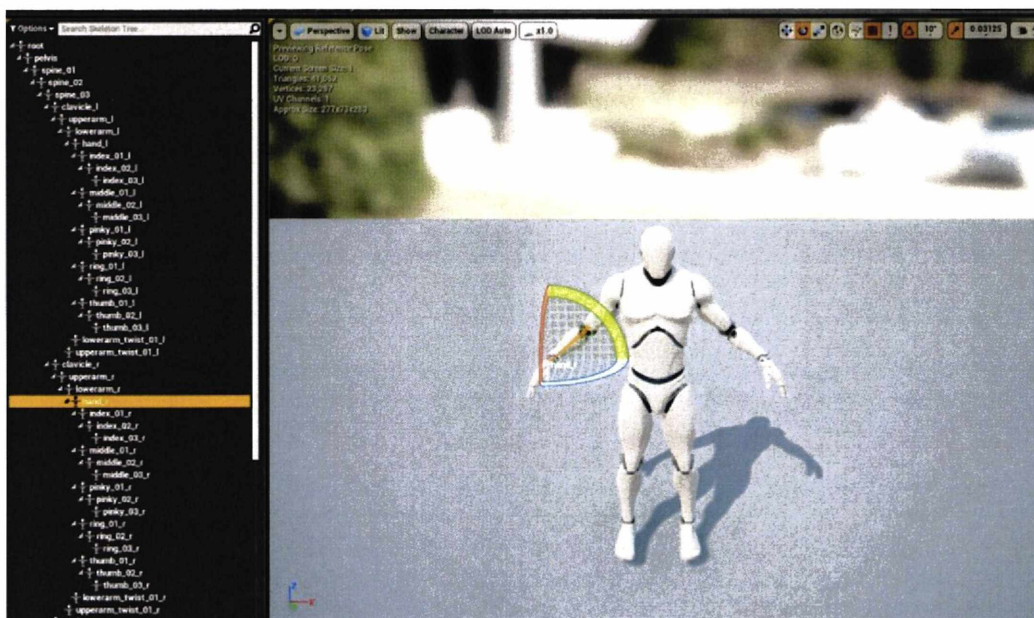
Εικόνα 14 (ΕQUIP όπλο στον χαρακτήρα στο Survivor BP)



Εικόνα 15 (Weapon BP)



Εικόνα 16 (Viewport του Survivor BP)

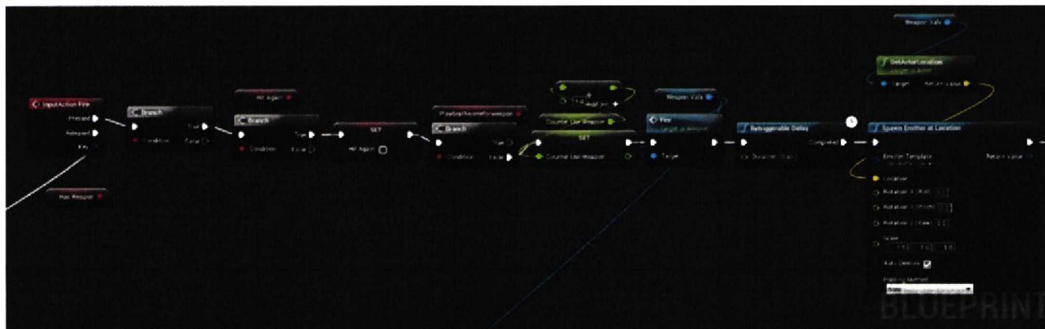


Εικόνα 17 (Skeletal mesh του χαρακτήρα)

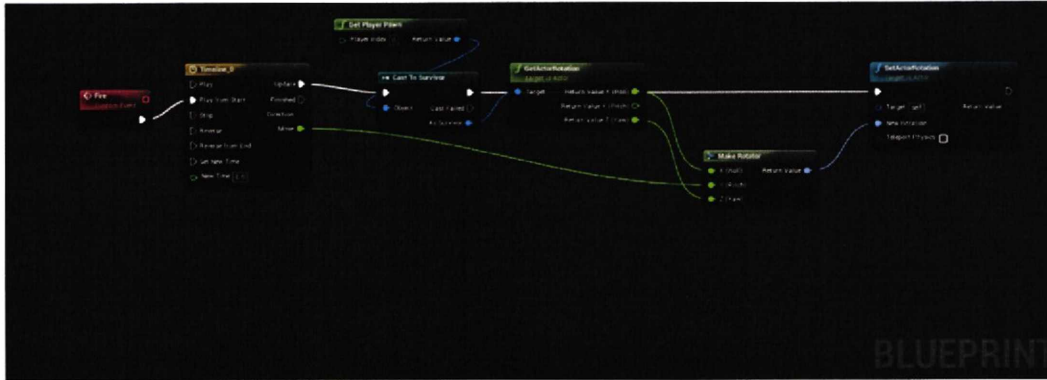
5) Επίθεση με το όπλο

Στην Εικόνα 18 φαίνεται το Event InputAction Fire που καλείται όταν ο χρήστης πιέζει το αριστερό κλικ του ποντικιού. Στη συνέχεια καλείται μία Δομή Αν, η

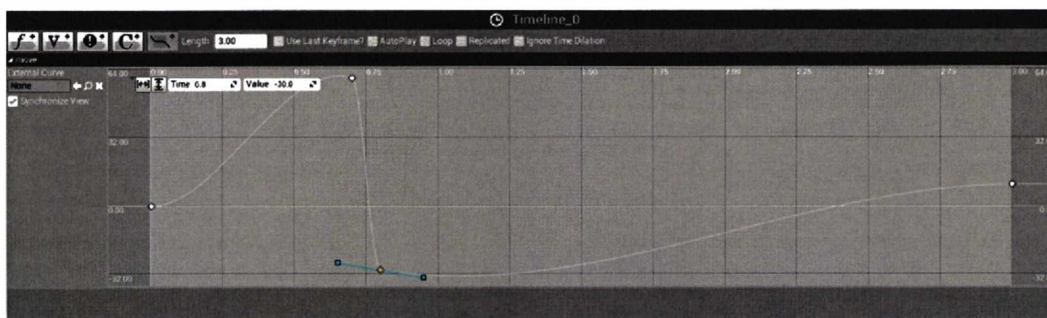
οποία είναι αληθής όταν η μεταβλητή Has Wearon είναι αληθής. Αυτός ήταν και ο λόγος που στο mechanic 4 τέθηκε η Has Wearon αληθής. Η επόμενη Δομή Αν έχει αληθής έξοδο όταν η μεταβλητή Hit Again είναι αληθής. Αμέσως μετά τίθεται η Hit Again ψευδής για να μην μπορεί να εκτελέσει ο χρήστης πολλαπλές επιθέσεις. Η τρίτη στην σειρά Δομή Αν είναι ψευδής καθώς η μεταβλητή Playdeathsceneforwearon είναι ψευδής. Στο επόμενο node προστίθεται η τιμή 1 στην μεταβλητή Counter Use Wearon, ώστε να μετρηθεί πόσες φορές έχει κάνει επίθεση ο χρήστης. Εν συνέχεια καλείται το Event Fire από το BP Wearon, αυτός είναι και ο λόγος που η συνάρτηση έχει ως είσοδο την μεταβλητή Wearon Valx, η οποία αναφέρεται στο Wearon BP. Η συνάρτηση φαίνεται αναλυτικά στην Εικόνα 19, όπου καλείται το Timeline_0, το οποίο έχει ως έξοδο την τιμή new. Η τιμή αυτή θα παίρνει διαδοχικά τιμές από 0 έως 60 και έως -30 για τα επόμενα 0.8 δευτερόλεπτα, όπως φαίνεται στο διάγραμμα στην Εικόνα 20. Το αποτέλεσμα της Timeline είναι να καλέσει στο τέλος την συνάρτηση SetActorRotation, ώστε να φανεί η κίνηση του όπλου όταν γίνει η επίθεση, περιστρέφοντάς το στον άξονα Y με την Make Rotator. Η Make Rotator χρησιμοποιείται για να επιλεγεί ο άξονας που θα γίνει η περιστροφή. Αυτό επιτυγχάνεται αλλάζοντας το Rotation του όπλου με τις διαδοχικές τιμές που δίνει το timeline. Επίσης, λαμβάνεται το Rotation του χαρακτήρα Survivor με την GetActorRotation, ώστε να γίνει η κίνηση στο σωστό σημείο στην οθόνη. Αφού τελειώσει η συνάρτηση Fire καλείται ένα Retriggerable Delay, το οποίο θα καθυστερήσει την εκτέλεση του επόμενου node για 0.85 δευτερόλεπτα, όσο δηλαδή θα χρειαστεί για να τελειώσει η συνάρτηση Fire. Τέλος, όταν ολοκληρωθεί η κίνηση του όπλου, καλείται η Spawn Emitter at Location η οποία παίρνει ως είσοδο την θέση (GetActorLocation) της μεταβλητής Wearon Valx και εμφανίζεται ένα Particle System (γραφικά που κινούνται μέσα στο παιχνίδι) καπνού.



Εικόνα 18 (Επίθεση με το όπλο του χαρακτήρα στο Survivor BP)



Εικόνα 19 (Επίθεση με το όπλο του χαρακτήρα στο Weapon BP)



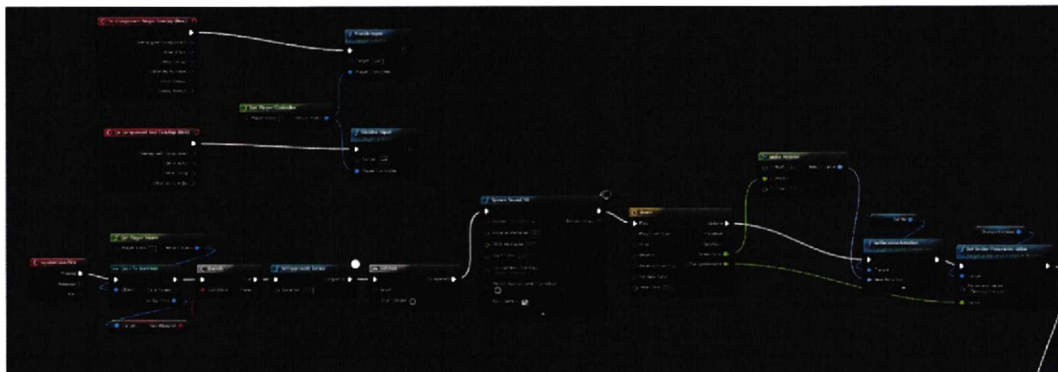
Εικόνα 20 (Timeline_0 στο Weapon BP)

6) Επίθεση στην πόρτα με το όπλο

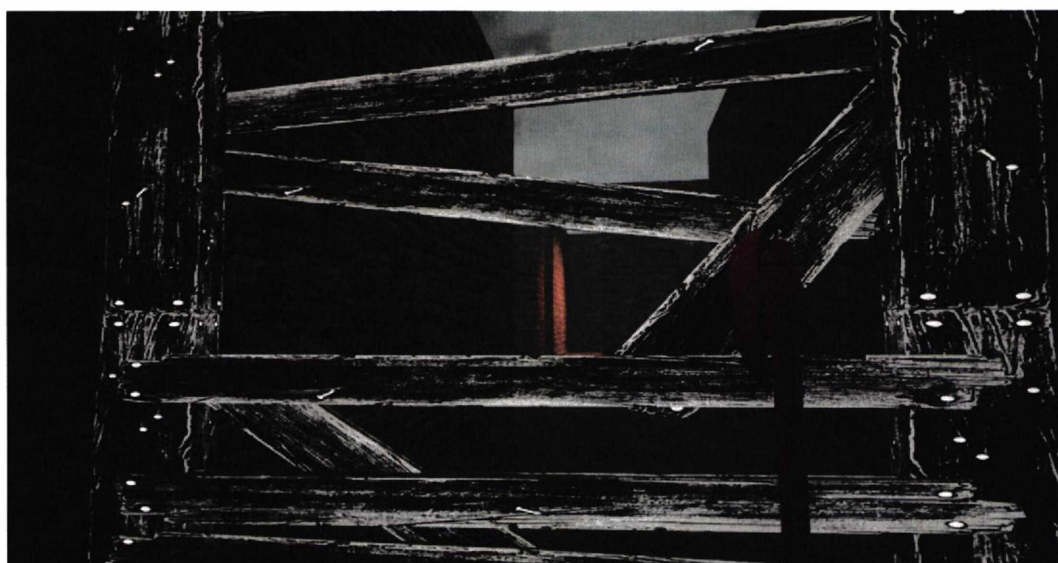
Στην Εικόνα 21 τα πρώτα 2 event overlap είναι τα ίδια με το mechanic 4. Το event InputAction Fire καλείται με το αριστερό κλικ. Στο επόμενο βήμα γίνεται προσβάσιμη η τιμή has weapon από το BP survivor καλώντας το node cast to Survivor (αυτό επιτυγχάνεται όπως είδαμε και στο mechanic 3) και έτσι τα υπόλοιπα nodes θα τρέξουν μόνο αν η Δομή Αν που έχει ως είσοδο την τιμή has weapon επιστρέψει αληθής. Στη συνέχεια υπάρχει ένα delay 0.8 δευτερολέπτων, ώστε να τελειώσει η επίθεση του χαρακτήρα και να ξεκινήσει το πέσιμο τις πόρτας. Η ύπαρξη της DoOnce επιτρέπει να τρέξουν τα υπόλοιπα nodes μια φορά, ειδάλλως με παρατεταμένα κλικ θα υπάρξει πρόβλημα. Εν συνέχεια καλείται η συνάρτηση Spawn Sound 2D και ακούγεται ένας ήχος μια πόρτας που πέφτει. Το timeline down που δίνει διαδοχικές τιμές στις μεταβλητές Down Door και Changeemissive, θα αλλάξει διαδοχικά το Rotation των μεταβλητών Ser Re και Change Emissive, καλώντας τις συναρτήσεις SetRelativeRotation και την Set Scalar Parameter Value αντίστοιχα για κάθε μεταβλητή. Η πρώτη μεταβλητή (Ser Re) αναφέρεται στο Static Mesh της πόρτας και η δεύτερη (Change Emissive) στο Scalar Parameter του material (εμφάνιση χρωμάτων πάνω σε ένα αντικείμενο) που χρησιμοποιεί το static mesh. Η

Scalar Parameter είναι ένας ακέραιος αριθμός που χρησιμοποιείται μέσα στο material για την αλλαγή του χρώματος. Αυτό θα έχει ως αποτέλεσμα η πόρτα να πέσει προς τα πίσω και να αλλάξει το άσπρο χρώμα που έχει πάνω της (Εικόνα 22 και 23).

Για να μπορέσουν να επεξεργαστούν μεταβλητές μέσα σε material έπρεπε να ακολουθηθεί μια συγκεκριμένη διαδικασία στην αρχή του παιχνιδιού (Εικόνα 24). Καλείται κατά το ξεκίνημα του παιχνιδιού, δηλαδή στο Event Begin Play η συνάρτηση Create Dynamic Material Instance που έχει ως είσοδο την μεταβλητή Ser Re (μεταβλητή που αναφέρετε στο static mesh πόρτα), το Source Material που είναι το material που χρησιμοποιήθηκε στο αντικείμενο και το Element Index, το οποίο γίνεται προσβάσιμο με την συνάρτηση Get Material Index. Η Get Material Index έχει ως είσοδο την μεταβλητή Ser Re και το Material Slot Name (το όνομα της θέσης που βρίσκεται το material Εικόνα 25). Τέλος η έξοδος της Create Dynamic Material Instance τίθεται στην μεταβλητή Change Emissive.



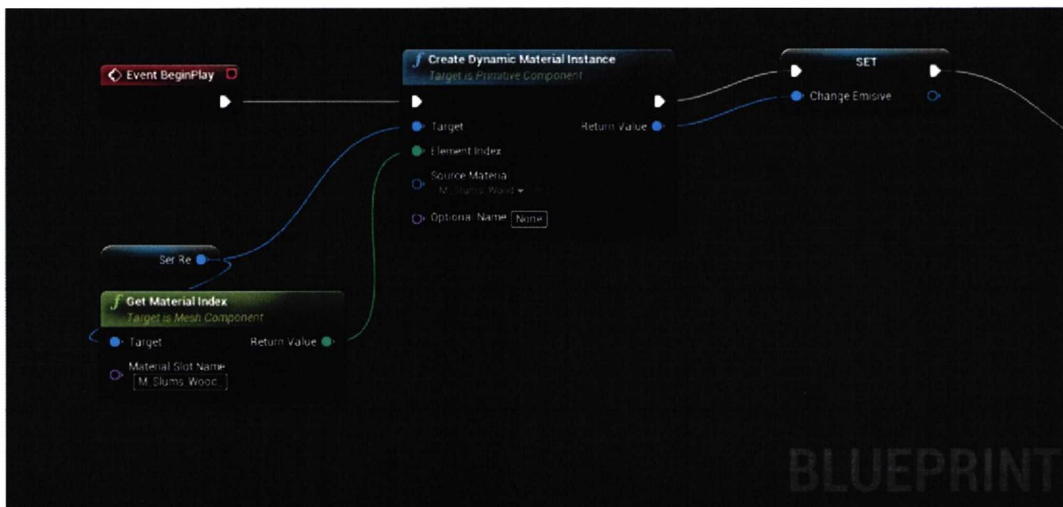
Εικόνα 21 (Πέσιμο πόρτας στο Destructive BP)



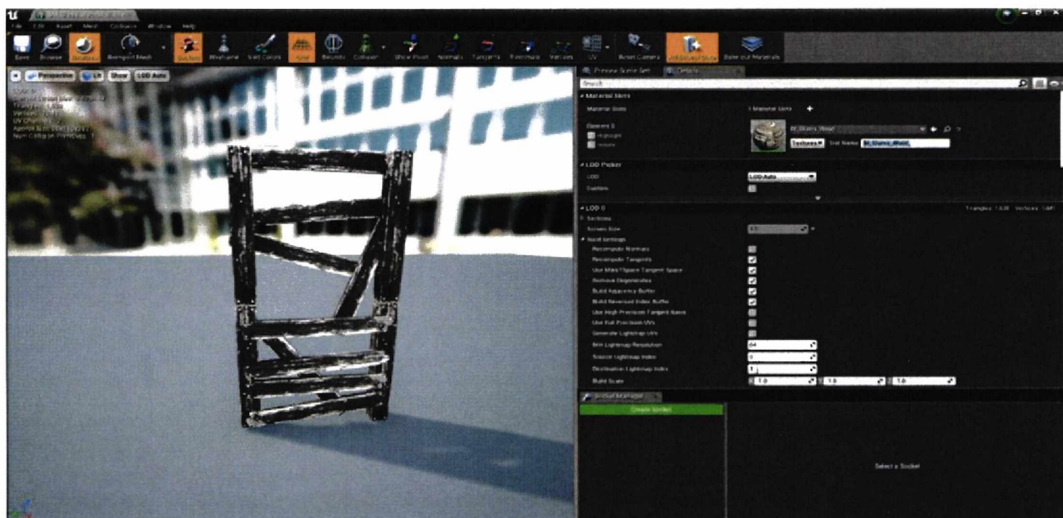
Εικόνα 22 (Χρώμα και θέση της πόρτας πριν την επίθεση)



Εικόνα 23 (Χρώμα και θέση της πόρτας μετά την επίθεση)



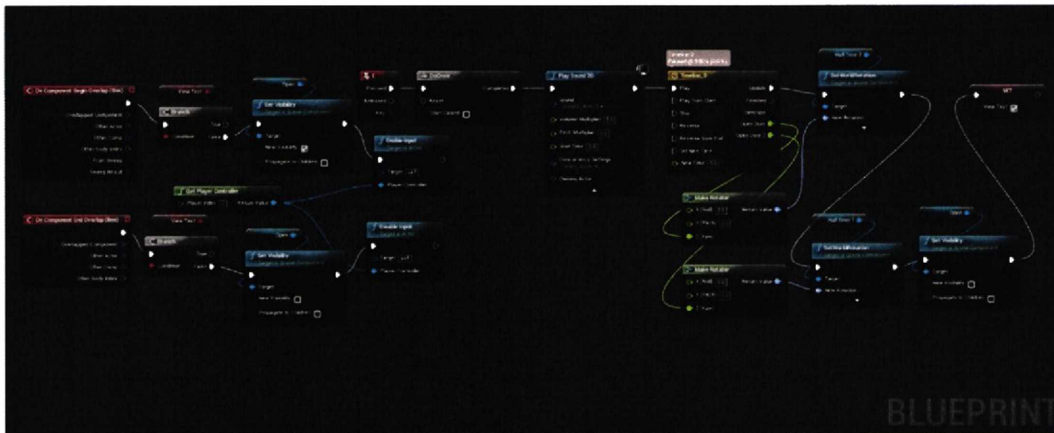
Εικόνα 24 (Διαδικασία επεξεργασίας μεταβλητών ενός material στο Destructive BP)



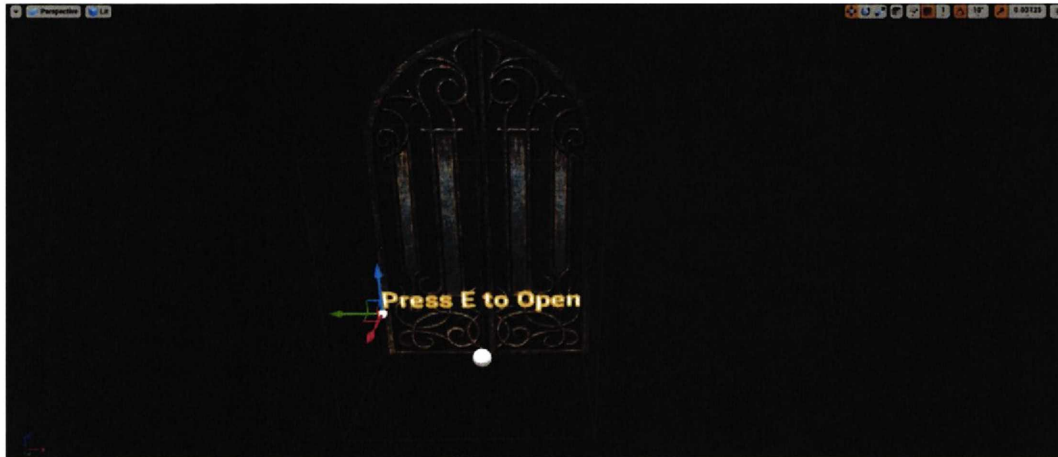
Εικόνα 25 (Ονομασία του Slot name του material που θα επεξεργασθεί)

7) Άνοιγμα Πόρτας

Στην Εικόνα 26 φαίνονται όπως και πριν τα δύο Event overlap. Η μόνη διαφορά με πριν, είναι ότι κάθε φορά που καλείται κάποιο από τα δύο αυτά Event, υπάρχει μια Δομή Αν που ελέγχει αν η τιμή View Text είναι ψευδής. Όπως θα φανεί στο τέλος του BP η View Text γίνεται αληθής και οι Δομές Αν θα επιστρέφουν αληθείς, με αποτέλεσμα να μην αλλάζει το visibility της μεταβλητής Open (μεταβλητή τύπου text render που εμφανίζει στην οθόνη “Press E to Open” Εικόνα 27). Όταν πατηθεί το πλήκτρο E και ο χαρακτήρας βρίσκεται κοντά στην πόρτα καλείται η συνάρτηση Do once, ώστε το άνοιγμα τις πόρτας να γίνει μία φορά. Η συνάρτηση Play Sound 2D παίζει έναν ήχο μιας πόρτας που ανοίγει. Στην συνέχεια καλείται το Timeline Timeline_0, στο οποίο όπως και στα προηγούμενα mechanic θα δοθούν διαδοχικές τιμές στις μεταβλητές Open Door και Open Door 2 για να αλλάξει το Rotation των μεταβλητών Half Door 1 και Half Door 2 (μεταβλητές που αναφέρονται στο static mesh της πόρτας) με τις συναρτήσεις SetWorldRotation. Τέλος τίθεται το visibility της Open ψευδής, ώστε να εξαφανιστεί το κείμενο που εμφανίζεται (Εικόνα 27).



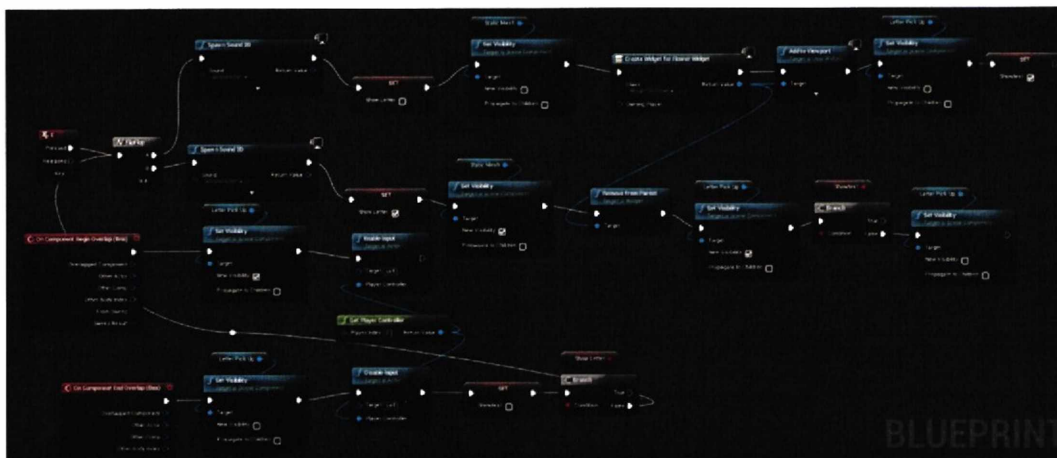
Εικόνα 26 (Άνοιγμα πόρτας στο Door BP)



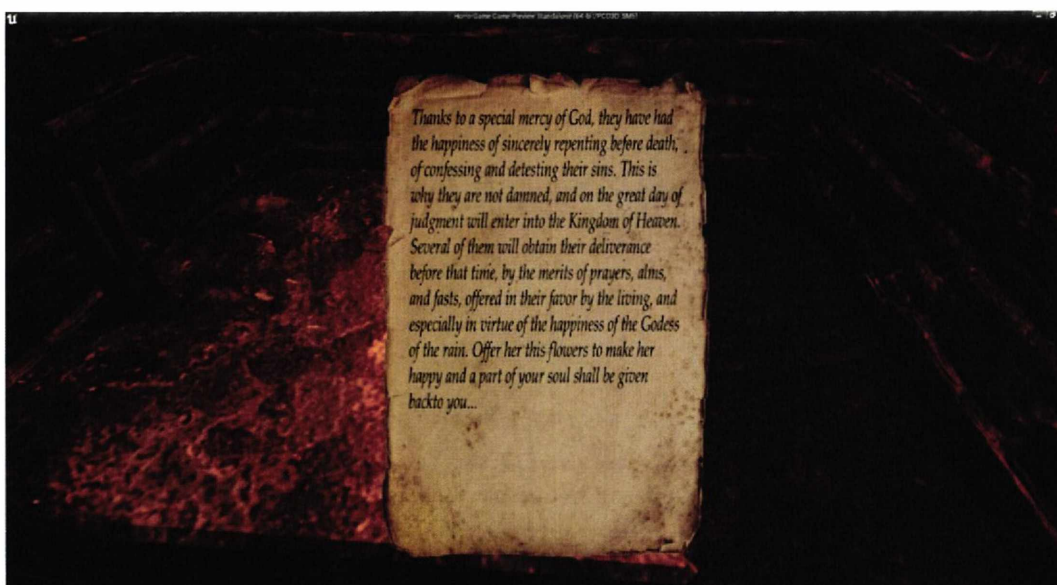
Εικόνα 27 (Viewport του Door BP)

7) Εμφάνιση γράμματος στην οθόνη

Στην Εικόνα 28 φαίνονται τα 2 Event overlap με τα οποία αλλάζει το visibility του Letter Pick Up (μεταβλητή τύπου text render που εμφανίζει στην οθόνη Press E to read). Η συνάρτηση Disable Input καλεί το node που θέτει την τιμή Showtext ψευδής και στην συνέχεια αν η τιμή Show Letter είναι ψευδής καλείται το Node Flip Flop. Το Flip Flop τρέχει εναλλάξ (mechanic 2). Και στις 2 περιπτώσεις της Flip Flop καλείται η συνάρτηση Spawn Sound 2D για να ακουστεί ο ήχος καθώς ένα γράμμα σηκώνεται ή όταν αφήνεται κάτω. Στην A περίπτωση τίθεται η μεταβλητή Show Letter ψευδής έτσι ώστε η Δομή Αν μετά το Disable Input να έχει ως έξοδο ψευδής, όταν δηλαδή καλεστεί το δεύτερο overlap event. Μετά τίθεται το visibility του Static Mesh (μεταβλητή που αναφέρετε στο Static Mesh του γράμματος) ψευδής με την συνάρτηση Set Visibility και εμφανίζεται στην οθόνη ένα Widget (γραφικό στοιχείο Εικόνα 29) με τις συναρτήσεις Create Widget for Flower Widget και Add to Viewport. Εφόσον εμφανιστεί το γραφικό στοιχείο του γράμματος στην οθόνη, τίθεται το visibility της μεταβλητής Letter Pick Up (μεταβλητή τύπου text render που εμφανίζει στην οθόνη "Press E to read") ψευδής με την συνάρτηση Set Visibility και έτσι η τιμή Showtext τίθεται αληθής. Στην B περίπτωση, όταν δηλαδή απομακρυνθεί ο χαρακτήρας από το γράμμα ή όταν ξαναπατηθεί το πλήκτρο E, τίθεται η μεταβλητή Show Letter, το visibility του Static Mesh και το visibility της Letter Pick Up αληθής. Επίσης εξαφανίζεται το Widget (γραφικό στοιχείο) με την Remove from Parent. Η μόνη διαφορά είναι η δομή Αν στο τέλος όπου αν η τιμή Show text είναι ψευδής, το visibility της Letter Pick Up θα γίνει ψευδής. Αυτή η δομή Αν, θα τρέξει μόνο σε περίπτωση που απομακρυνθεί ο χαρακτήρας από το γράμμα, όταν δηλαδή η μεταβλητή Showtext γίνεται ψευδής.



Εικόνα 28 (Εμφάνιση γράμματος στην οθόνη του χρήστη στο Letter BP)



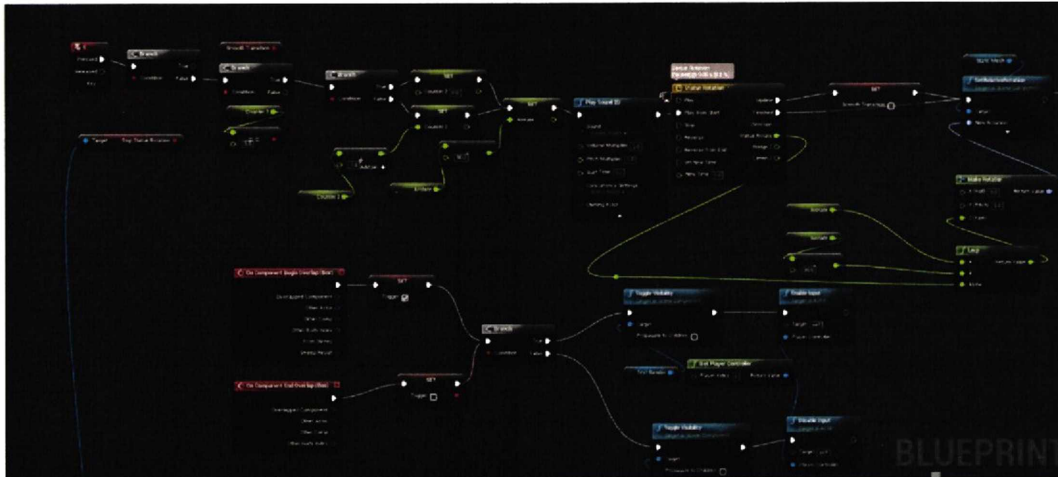
Εικόνα 29 (Εμφάνιση γράμματος στην οθόνη του χρήστη)

8) Γρίφος με την Περιστροφή των αγαλμάτων

Στην Εικόνα 30-31 όπως και στα προηγούμενα mechanics φαίνονται τα 2 event overlap. Η λειτουργία τους είναι ίδια με τα προηγούμενα. Όταν πατηθεί το πλήκτρο E θα τρέξει μια Δομή Αν, η οποία θα ελέγξει αν η μεταβλητή Stop Statue Rotation είναι ψευδής. Αυτή η μεταβλητή γίνεται αληθής όταν όλες οι θέσεις των αγαλμάτων είναι σωστές και έτσι ο χρήστης δεν μπορεί να περιστρέψει άλλο τα αγάλματα. Η επόμενη Δομή Αν, ελέγχει αν η μεταβλητή Smooth Transition είναι αληθής. Ο λόγος που γίνεται αυτό είναι για να μην μπορεί ο χαρακτήρας να

περιστρέψει το άγαλμα μέχρις ότου να τελειώσει το timeline που κάνει την κίνηση (μετά το timeline φαίνεται ότι τίθεται η τιμή Smooth Transition ψευδής). Στην τρίτη Δομή An ελέγχεται αν η μεταβλητή Counter 2 (που έχει ως περιεχόμενο το νούμερο 0), είναι ίση με 3, επειδή 4 είναι οι κινήσεις του αγάλματος 0,1,2 και 3. Αυτό γίνεται ώστε να τεθεί η Counter 2 ίση με 0 όταν γίνουν οι 4 περιστροφές, αλλιώς θα πρέπει να προστεθεί η μεταβλητή με τον αριθμό 1. Στην συνέχεια αφαιρείται από την μεταβλητή Arotate (που έχει ως αρχική τιμή το νούμερο 90) το νούμερο 90 και καλείται η συνάρτηση Play Sound 2D, έτσι ώστε να ακουστεί ο ήχος που γυρίζει το άγαλμα. Για να γίνει η περιστροφή πρέπει να κληθεί το timeline Statue Rotation που δίνει διαδοχικές τιμές από 0 έως 1 στην μεταβλητή Statue Rotate. Αυτό γίνεται ώστε η συνάρτηση Lerp που έχει ως A είσοδο την Arotate, ως B είσοδο την Arotate μειωμένη κατά 90 και ως είσοδο Alpha την Statue Rotate του timeline, να δώσει διαδοχικές τιμές από τις τιμές που παίρνει στις εισόδους A και B. Για την ακρίβεια ενώ η Alpha θα πλησιάζει να γίνει 1 οι τιμές την πρώτη φορά που θα γίνει η περιστροφή θα κυμαίνονται ανάμεσα από το 0 έως το -90. Η περιστροφή του αγάλματος γίνεται καλώντας την SetRelativeRotation με εισόδους την μεταβλητή Static Mesh (μεταβλητή που αναφέρετε στο static mesh άγαλμα) και την έξοδο της Make Rotator. Η Make Rotator έχει ως είσοδο την έξοδο της συνάρτησης Lerp στον άξονα z. Ο λόγος που αφαιρείται από την Arotate ο αριθμός -90, είναι για να αλλάζει το σημείο αναφοράς μετά από κάθε περιστροφή. Όπως φαίνεται στην Εικόνα 29 όταν τελειώσει το timeline μετά την έξοδο Finished θα τεθεί η μεταβλητή Smooth Transition αληθής, ώστε να μπορεί να γίνει ξανά περιστροφή του αγάλματος και θα ελεγχθεί αν η μεταβλητή Counter 2 είναι ίση με 0, που είναι και η θέση που πρέπει να βρίσκετε το άγαλμα για να λυθεί ο γρίφος. Η θέση 0 είναι η αρχική θέση στην περίπτωση αυτή, δηλαδή το άγαλμα δεν χρειάζεται να κινηθεί διότι βρίσκετε εξαρχής στην σωστή θέση. Εφόσον βρίσκεται στην σωστή θέση τίθεται αληθής η μεταβλητή Stature Rotation 2 του BP Survivor, ειδάλλως τίθεται ψευδής. Τέλος, ελέγχεται αν αυτή και οι μεταβλητές Stature Rotation 1 και Statue Rotation 3 είναι αληθής, έτσι ώστε να κληθεί το Event από το BP του Survivor Rain On (Εικόνα 32).

Σε αυτό το Event τίθεται το visibility της μεταβλητής Particle System (γραφικά που κινούνται μέσα στο παιχνίδι) αληθής και καλείται ένα delay για 1 δευτερόλεπτο για λόγους συγχρονισμού. Στην συνέχεια καλούνται οι συναρτήσεις Spawn Sound 2D και Fade In από 2 φορές ώστε να αρχίσουν να ακούγονται ήχοι βροχής αέρα. Η συνάρτηση Fade In έχει 4 εισόδους, το Target η οποία επιλέγει τον ήχο που θα ενεργοποιηθεί, την Fade In Duration που ορίζει τα δευτερόλεπτα που θα χρειαστεί να περάσουν για να ακουστεί ο ήχος στην μέγιστη ένταση, την Fade Volume Level που θέτει το ποσοστό της έντασης του ήχου και την Start Time για το πότε θα ξεκινήσει.



Εικόνα 30 (Περιστροφή αγάλματος στο Statue BP)



Εικόνα 31 (Περιστροφή αγάλματος στο Statue BP)

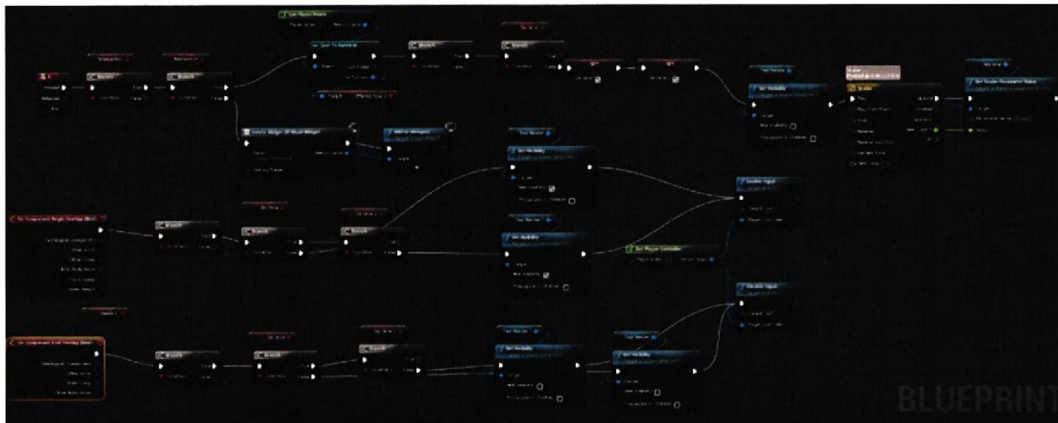


Εικόνα 32 (Περιστροφή αγάλματος στο Statue BP)

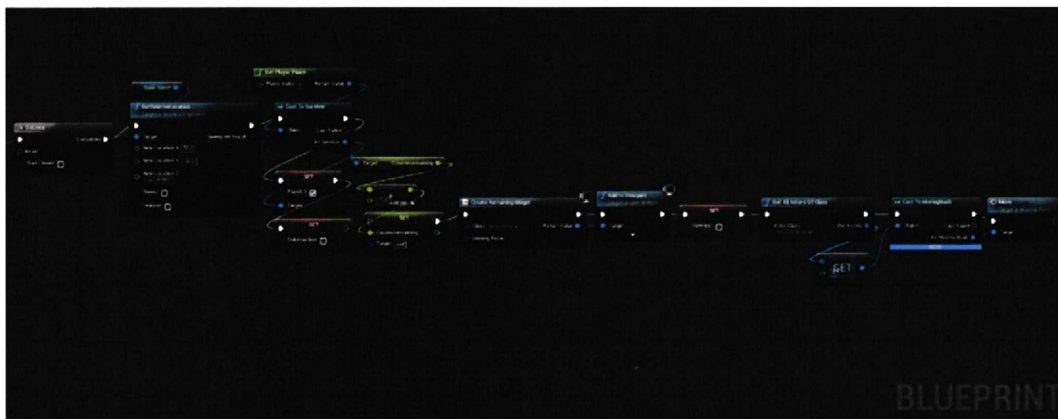
9) Τοποθέτηση των ψυχών στους βωμούς

Στις Εικόνες 33-34 φαίνονται τα 2 event overlap, όπου η λειτουργία τους είναι ίδια με τα προηγούμενα με την μόνη διαφορά ότι υπάρχουν πολλές Δομές An και 2 διαφορετικές Text Render μεταβλητές, οι οποίες αλλάζουν το κείμενο που εμφανίζεται στην οθόνη όταν πλησιάζει ο χαρακτήρας. Όταν πατηθεί το πλήκτρο E υπάρχουν 2 Δομές An με τις μεταβλητές Dokeyaction και Rainornot. Η πρώτη όταν όλα τα nodes του event E ολοκληρωθούν γίνεται ψευδής ώστε να μην μπορεί να καλεστεί ξανά. Η δεύτερη ελέγχει αν βρέχει (αν δηλαδή έχουν γίνει σωστά οι περιστροφές των αγαλμάτων). Στην περίπτωση αυτή η μεταβλητή Rainornot γίνεται αληθής, αλλιώς παραμένει ψευδής και δημιουργείται ένα widget με τις συναρτήσεις Create Widget Of Ritual Widget και Add to Viewport, εμφανίζοντας στην οθόνη το γραφικό στοιχείο "Ritual Requires Rain". Στην περίπτωση που είναι αληθής επιλέγεται από το BP του Survivor η μεταβλητή Offering Soul 2, η οποία τίθεται αληθής όταν έχει βρεθεί η ψυχή 2 μέσα στο επίπεδο 1. Στην συνέχεια μια Δομή An ελέγχει την μεταβλητή Do Once, η οποία γίνεται ψευδής όταν βρίσκει ο χαρακτήρας την ψυχή 2 όπως πριν. Μετά τίθενται οι μεταβλητές Do Once και Do Once 2 ψευδής καθώς τίθεται και το visibility της μεταβλητής Text Render ψευδής. Αυτό έχει ως αποτέλεσμα τα υπόλοιπα nodes να τρέξουν μόνο μια φορά και να μην ξαναεμφανιστεί κάποιο από τα 2 Text Render στην οθόνη αν πλησιάζει ο χαρακτήρας. Στην συνέχεια καλείται το timeline Scalar το οποίο δίνει διαδοχικές τιμές στην μεταβλητή New Track 0 που είναι η είσοδος της συνάρτησης Set Scalar Parameter Value καθώς η άλλη είσοδος είναι η μεταβλητή Material (μεταβλητή που αναφέρεται στην εμφάνιση χρωμάτων πάνω στο static mesh νερού Εικόνα 36). Καθώς αλλάζει το material καλείται η DoOnce για να τρέξουν τα υπόλοιπα nodes μια φορά και καλείται η συνάρτηση SetRelativeLocation μεταφέροντας στην σωστή θέση την μεταβλητή Static Mesh (το αντικείμενο νερού που περιέχει ο βωμός). Επιπροσθέτως καλείται το node Cast To survivor και τίθεται η μεταβλητή Psyche 3 του BP του Survivor αληθής, καθώς τίθεται και η μεταβλητή Dokeyaction ψευδής, η οποία ελέγχεται στην αρχή του Event. Στην συνέχεια αυξάνεται η μεταβλητή του Survivor Counterremaining κατά 1, ώστε να δημιουργηθεί ένα widget με τις συναρτήσεις Create Widget Of Ritual Widget και Add to Viewport, εμφανίζοντας στην οθόνη το γραφικό στοιχείο "Remaining:1/3" ή "Remaining:2/3" ή "Remaining 3/3" (Εικόνα 36), το οποίο δείχνει πόσες ψυχές απομένουν να βάλει ο χρήστης στους βωμούς. Ανάλογα με την τιμή του Counterremaining θα εμφανιστεί το αντίστοιχο widget. Τέλος, τίθεται η μεταβλητή Viewtext ψευδής και καλείται η συνάρτηση Get All Actors Of Class, η οποία έχει ως έξοδο μια get που δίνει τον πρώτο Actor από το Actor Class που επιλέχθηκε. Αυτό δίνει την δυνατότητα να μπορεί να κληθεί το node Cast To MovingBush όπως καλείται και το Cast To Survivor, με την μόνη διαφορά ότι τώρα επικοινωνεί το BP Actor με ένα άλλο BP Actor και όχι με Character BP. Εφόσον είναι επιτυχές το Cast καλείται το Event Move.

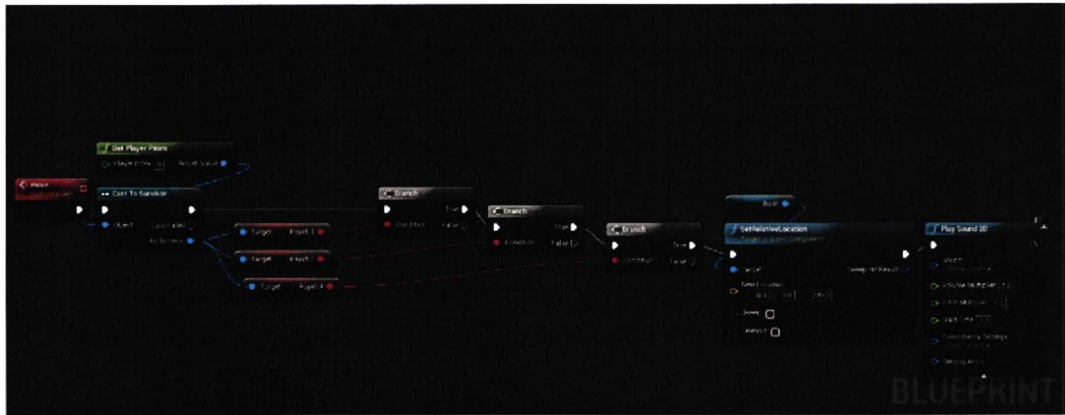
Στην Εικόνα 35 φαίνεται το Event Move, το οποίο έχει 3 Δομές Αν. Αυτές ελέγχουν τις μεταβλητές Psyχη 1, Psyχη 2, Psyχη 3 του BP του Survivor. Αν όλες είναι αληθείς αλλάζει η θέση της μεταβλητής Bush (μεταβλητή που αναφέρεται στο static mesh τοίχου) με την συνάρτηση SetRelativeLocation και παίζει έναν ήχο με την συνάρτηση Play Sound 2D για να ακουστεί ότι κάποιος τοίχος μέσα στον λαβύρινθο μετακινείται (Εικόνα 37-38).



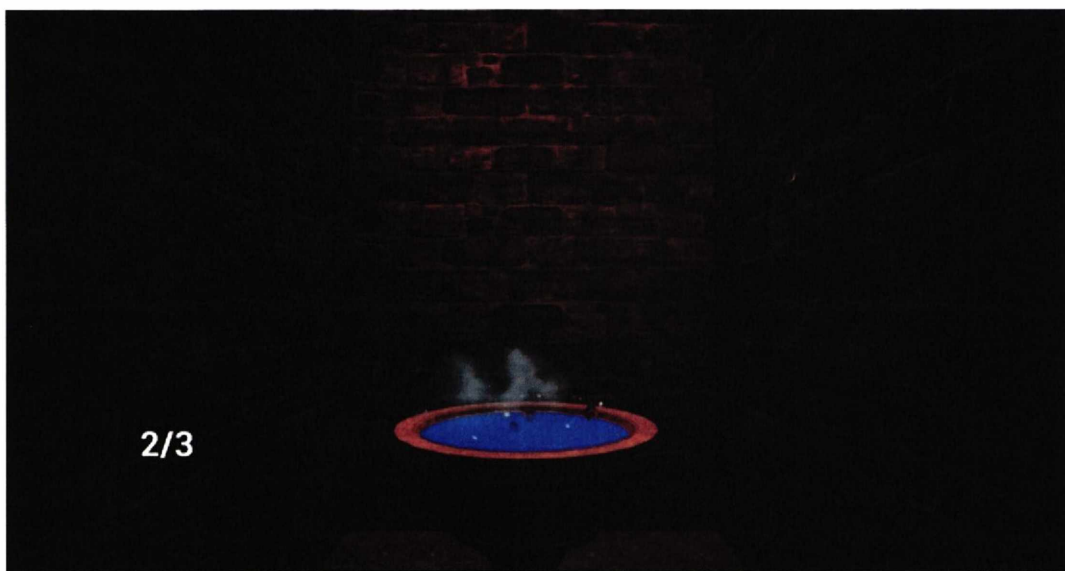
Εικόνα 33 (Τοποθέτηση ψυχής στον βωμό στο Button3 BP)



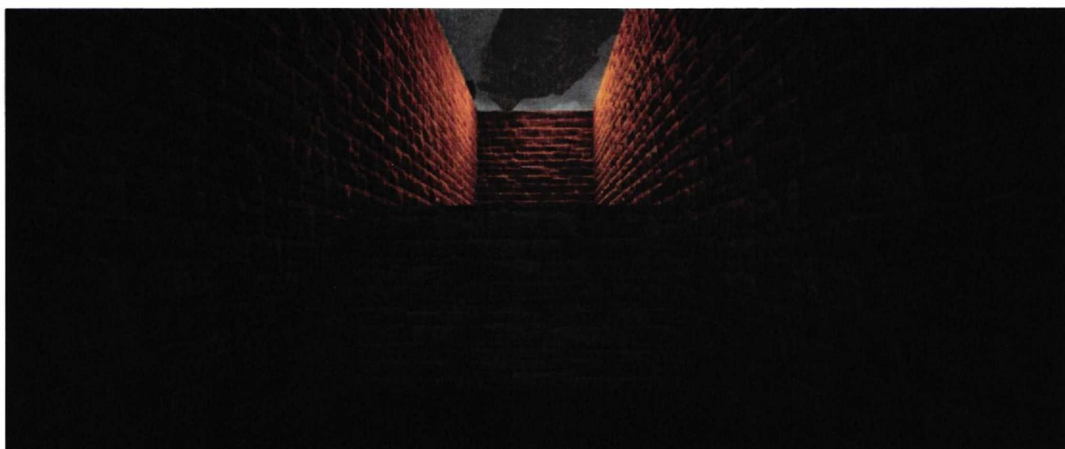
Εικόνα 34 (Τοποθέτηση ψυχής στον βωμό στο Button3 BP)



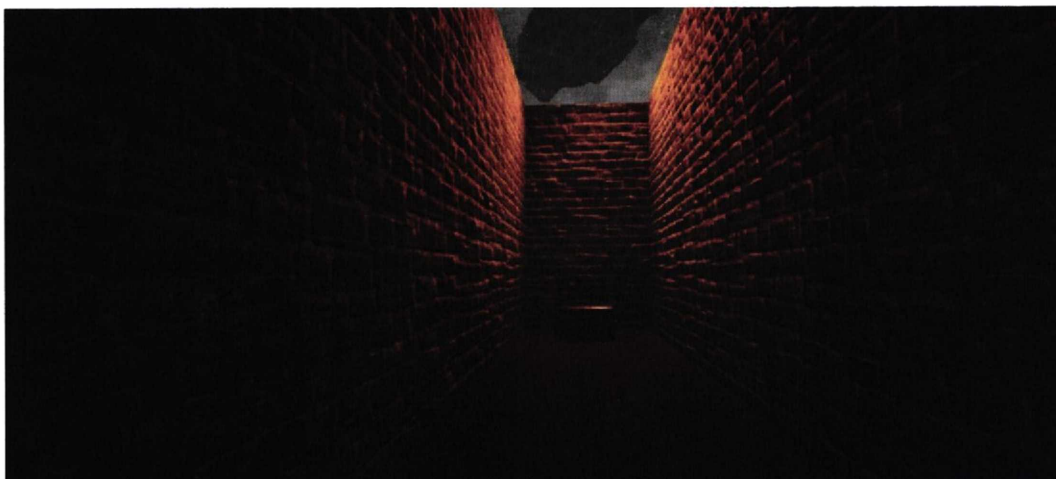
Εικόνα 35 (Κίνηση τοίχου στο MovingBush BP)



Εικόνα 36 (Τοποθέτηση ψυχής στον βωμό)



Εικόνα 37 (Τοίχος πριν την τοποθέτηση και των τριών ψυχών)



Εικόνα 38 (Τοίχος μετά την τοποθέτηση και των τριών ψυχών)

10) Γρίφος με τα βιβλία

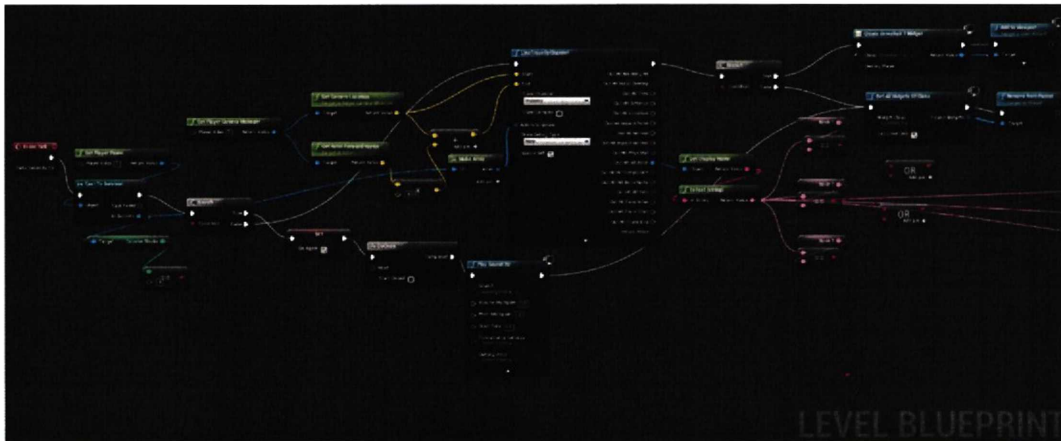
Στην Εικόνα 39 χρησιμοποιείται το Event Tick που καλείται κάθε Frame του παιχνιδιού. Στην αρχή ελέγχεται η μεταβλητή Counter Books του BP του Survivor αν ισούται με 3 και αν είναι ψευδής καλείται η συνάρτηση LineTraceByChannel. Αυτή η συνάρτηση εμφανίζει μια αόρατη κόκκινη γραμμή (στο Draw Debug Type μπορεί να επιλεγθεί να εμφανιστεί αν χρειάζεται για λόγους αποσφαλμάτωσης) μπροστά από τον παίκτη που αναγνωρίζει την επαφή με τα αντικείμενα. Μέσω της συνάρτησης Get Player Camera Manager, λαμβάνεται η θέση της κάμερας (Get Camera Location) και την μπροστινή κατεύθυνση (Get Actor Forward Vector) του χαρακτήρα. Η θέση της κάμερας θα είναι η είσοδο Start της LineTraceByChannel και η είσοδος End θα είναι η μπροστινή κατεύθυνση πολλαπλασιασμένη με το νούμερο 50. Το αποτέλεσμα αυτό προστίθεται στην θέση της κάμερας. Ουσιαστικά με την είσοδο End ορίζεται η απόσταση που θα έχει η κόκκινη γραμμή, η οποία θα τοποθετηθεί στο κέντρο της οθόνης. Επίσης έχει προστεθεί ο χαρακτήρας Survivor ως είσοδος στην Actors to ignore στην συνάρτηση LineTraceByChannel, ώστε να αγνοείται από το LineTrace (αόρατη κόκκινη γραμμή). Οι μεταβλητές Book1, Book2 και Book3 (μεταβλητές τύπου text) έχουν τα ονόματα των 3 βιβλίων του γρίφου. Στη συνέχεια ελέγχεται αν κάποια από τις τρεις μεταβλητές είναι ίδια με την έξοδο Out Hit Hit Actor (επιστρέφει το αντικείμενο που έρχεται σε επαφή το LineTrace) της LineTraceByChannel, μέσω μιας δομής Αν. Για να ληφθεί το όνομα του Hit Actor καλούνται οι συναρτήσεις Get Display Name που επιστρέφουν το όνομα σε τύπο string και την συνάρτηση ToText που μετατρέπει το string σε text. Εφόσον η προηγούμενη Δομή Αν είναι αληθής τότε ο χρήστης βρήκε το σωστό βιβλίο. Αυτό έχει ως αποτέλεσμα να δημιουργηθεί ένα widget (με τις συναρτήσεις Create Crosshair 1 Widget και Add to Viewport) Crosshair 1 και να εμφανιστεί ένα γραφικό

στοιχείο στο κέντρο της οθόνης. Αν δεν είναι αληθής, τότε καλείται η συνάρτηση Get All Widgets Of Class και επιλέγοντας το Crosshair 1 αφαιρείται από την οθόνη το γραφικό στοιχείο με την Remove from Parent. Τέλος στην περίπτωση που η μεταβλητή Counter Books είναι ίση με 3 τίθεται η μεταβλητή Do Again αληθής (θα αναφερθεί στην επόμενη παράγραφο) και καλούνται μετά το node DoOnce οι συναρτήσεις Play Sound 2D Get All Widgets Of Class και Remove from Parent, ώστε να ακουστεί ένας ήχος ότι λύθηκε ο γρίφος, ώστε να εξαφανιστεί από την οθόνη το γραφικό στοιχείο που αναφέρθηκε παραπάνω.

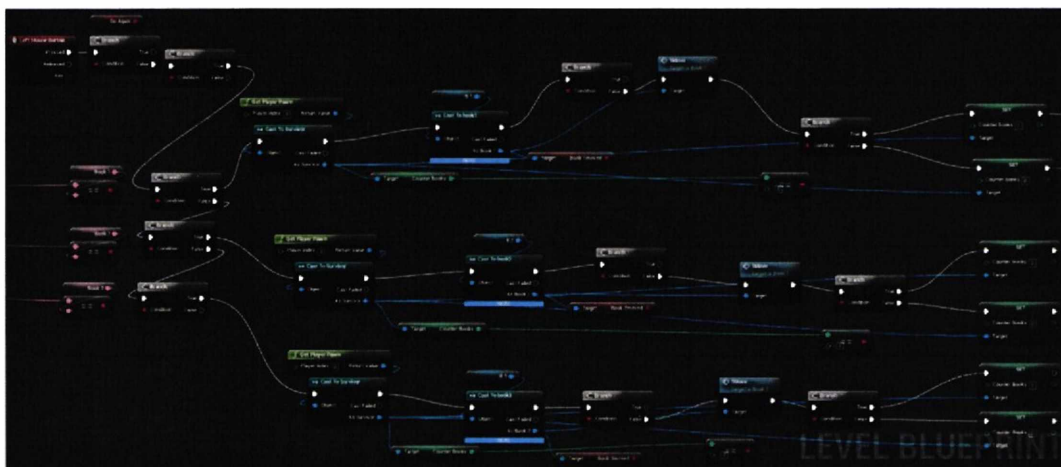
Στην εικόνα 40-41 φαίνεται το Event Left Mouse Button που ενεργοποιείται με κάθε αριστερό κλικ. Εδώ φαίνεται και ο λόγος ύπαρξης της μεταβλητής Do again, όπου αν είναι ψευδής τρέχει 3 Δομές Αν για τα τις μεταβλητές Book 1, Book 2, Book 3 ώστε να φανεί εάν είναι ίσες με την έξοδο Out Hit Hit Actor (στην περίπτωση που όλες είναι ψευδής δεν συμβαίνει τίποτα). Αν η πρώτη από αυτές τις δομές είναι αληθής γίνεται Cast To Survivor και Cast To book1, ώστε να παρθούν τα δεδομένα από αυτά τα BP και ελέγχεται η μεταβλητή Book 1moved του BP book1 και η μεταβλητή Counter Books του BP Survivor. Αν η Book 1moved είναι ψευδής καλείται το Event 1Move του BP book1 (κινείται το βιβλίο προς τα έξω, θα αναφερθεί στην επόμενη παράγραφο Εικόνα 42). Αν η Counter Books ισούται με 0 τίθεται η Counter Books να ισούται με 1, αλλιώς τίθεται με 0. Στις επόμενες 2 Δομές Αν που αναφέρονται στις ισότητες των μεταβλητών Book 2 και Book 3 γίνεται το ίδιο, με μόνες διαφορές στο δεύτερο Cast, όπου καλούνται τα nodes Cast To book2 και Cast To book3, καθώς και στην Δομή Αν για την ισότητα της μεταβλητής Counter Books του BP του Survivor, όπου τίθεται η Counter Books ίσων με 2 αν ισούται με 1 μετά το Cast To book2 ή τίθεται με 3 αν ισούται με 2 μετά το Cast To book3. Αυτό γίνεται ώστε ο χρήστης να πρέπει να κουνήσει (κουνιούνται με το Event 1Move Εικόνα 42) τα βιβλία με την σωστή σειρά για να λύσει τον γρίφο και η μεταβλητή Counter Books να γίνει 3. Αν η μεταβλητή Counter Books γίνει 3 στην αρχή του EventTick θα βγει η πρώτη Δομή Αν αληθής και ο γρίφος ουσιαστικά θα έχειλυθεί. Στην συνέχεια καλούνται τα nodes Cast To book1, Cast To book2 και Cast To book3, για να παρθούν οι μεταβλητές Book 1moved, Book 2moved και Book 3moved αντίστοιχα. Με 3 Δομές Αν ελέγχεται αν όλες είναι αληθής, το οποίο σημαίνει ότι έχει καλεστεί το Event 1Move (κινείται το βιβλίο προς τα έξω, θα αναφερθεί στην επόμενη παράγραφο Εικόνα 42) και με μια τέταρτη Δομή Αν ελέγχεται και η μεταβλητή Do again αν είναι ψευδής, έτσι ώστε να είναι γνωστό ότι ο γρίφος δεν λύθηκε ακόμα. Στην συνέχεια καλείται η συνάρτηση Delay για 2.1 δευτερόλεπτα για λόγους συγχρονισμού. Τέλος καλείται το Event 2Move (κινείται το βιβλίο προς τα μέσα, θα αναφερθεί στην επόμενη παράγραφο Εικόνα 42) για τα BP book1, book2 και book3 και τίθενται οι μεταβλητές Book 1move, Book 2moved και Book 3moved ψευδής αντίστοιχα, εφόσον επιστρέψουν στην κανονική τους θέση.

Στην Εικόνα 42 φαίνονται τα Event 1Move και 2Move που αναφέρθηκαν παραπάνω. Στο Event 1Move ελέγχεται με μια Δομή Αν η μεταβλητή Runmoveagain. Αν είναι αληθής καλείται η συνάρτηση Play Sound 2D για να ακουστεί ένας ήχος ότι κινείται το βιβλίο και τίθεται η μεταβλητή Book 1moved αληθής ώστε να μπορεί να

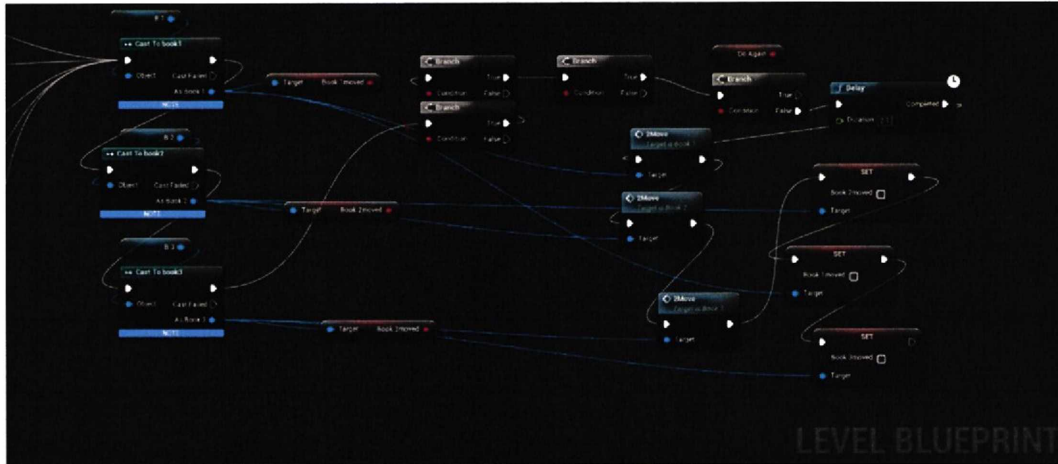
καλέσει το Event 2Move από το Event Tick που αναφέρθηκε παραπάνω. Στην συνέχεια καλείται ένα Timeline το Timeline_0 όπου δίνονται διαδοχικές τιμές στην μεταβλητή New Track 0, ώστε να αλλάξει η θέση της μεταβλητής Static Mesh (το αντικείμενο βιβλίο) με την συνάρτηση SetRelativeLocation και να μετακινηθεί το βιβλίο προς τα έξω. Στο Event 2Move το οποίο καλείτε στο Event Tick μόνο αν έχει καλεστεί το Event 1Move, τίθεται η μεταβλητή Runmoveagain ψευδής ώστε η πρώτη Δομή Αν του Event 1Move να έχει ως έξοδο ψευδής. Το Timeline_1 είναι το ίδιο με το Timeline_0 απλά κάνει την αντίθετη κίνηση στο Static Mesh, εφόσον έχει ως είσοδο το Reverse From End. Τέλος όταν τελειώσει το Timeline_1 τίθεται η μεταβλητή Runmoveagain αληθής και καλείται το node Cast To Survivor ώστε να τεθεί η τιμή της Counter Books του BP Survivor σε 0, εφόσον τα βιβλία δεν θα έχουν μετακινηθεί με την σωστή σειρά.



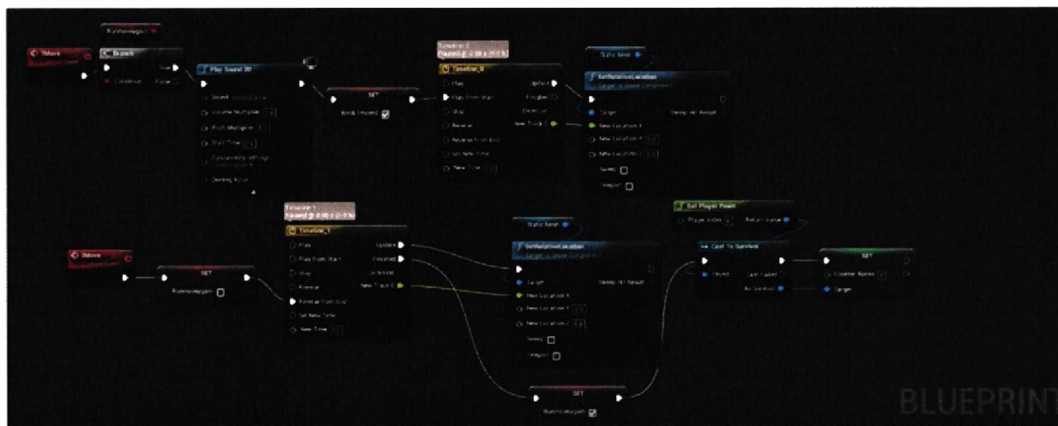
Εικόνα 39 (Αναγνώριση και μετακίνηση βιβλίων στο Level2 BP)



Εικόνα 40 (Αναγνώριση και μετακίνηση βιβλίων στο Level2 BP Level2 BP)



Εικόνα 41 (Αναγνώριση και μετακίνηση βιβλίων στο Level2 BP Level2 BP)



Εικόνα 42 (Κίνηση βιβλίου στο BP book1)

11) Αντιμέτωπιση εχθρού, θάνατος και αναγέννηση χαρακτήρα

Στην Εικόνα 43 φαίνεται ο τρόπος που εμφανίζεται ο εχθρός μέσα στο παιχνίδι. Το Event OnActorBeginOverlap καλείται όταν ο χαρακτήρας έρθει σε επαφή με το Trigger Box5 (ένα αόρατο αντικείμενο μέσα στον χάρτη). Στην συνέχεια καλείται η DoOnce για να εμφανιστεί ο εχθρός μόνο μια φορά και καλείται η συνάρτηση Spawn AI From Class που θα εμφανίσει τον εχθρό. Χρησιμοποιείται ως είσοδος της Spawn AI From Class, το Location του χαρακτήρα, το οποίο γίνεται προσβάσιμο με τις συναρτήσεις Get Player Character (επιστρέφει τον χαρακτήρα που χρησιμοποιείται) και GetWorldLocation (επιστρέφει την θέση του χαρακτήρα μέσα στο επίπεδο). Για να εμφανιστεί πιο μπροστά ο εχθρός προστίθεται στο Location το νούμερο -600 στον άξονα Y και το νούμερο 100 στον άξονα Z. Μετά τίθεται η μεταβλητή Combatstaystill ψευδής, ώστε να μην μπορεί να κουνηθεί ο

χαρακτήρας και να αντιμετωπίσει τον εχθρό ακίνητος. Στην συνέχεια καλούνται οι συναρτήσεις Play Sound 2D, Spawn Sound 2D και Delay. Η τελευταία Delay θα καλέσει μετά από 6 δευτερόλεπτα την συνάρτηση Fade Out (κάνει το αντίθετο με την Fade In), με είσοδο την έξοδο της συνάρτησης Spawn Sound 2D.

Στην Εικόνα 44-45 φαίνεται το Event Tick και το Event BeginPlay. Στο Event BeginPlay καλείται το Timeline Timeline_0 που θα τρέξει για 5 δευτερόλεπτα έτσι ώστε αν η μεταβλητή lthits Stop Running είναι αληθής τότε η Δομή An να καλεί διαδοχικά (λόγο της Timeline) την συνάρτηση Add Movement Input, η οποία θα δίνει κίνηση στον χαρακτήρα (εχθρό). Για να φαίνεται η κίνηση του χαρακτήρα καλείται η DoOnce που με την σειρά της θα καλέσει για 1 φορά μόνο την συνάρτηση Play Animation (κινήσεις χαρακτήρα). Ως είσοδο στην Play Animation δίνονται οι μεταβλητές Mesh (Skeletal Mesh του χαρακτήρα-εχθρού) και το Animation με το οποίο τρέχει ο εχθρός. Επίσης τίθεται αληθής το looping για να επαναλαμβάνεται το Animation. Στο Event Tick ελέγχεται η μεταβλητή Shouldistorspawn01 (θα αναφερθεί στην επόμενη παράγραφο Εικόνα 47) του BP Survivor και αν είναι αληθής η Δομή An θα καλέσει το Timeline Timeline_2. Η Timeline_2 δίνει διαδοχικές τιμές στην μεταβλητή New Track 1 από 600 έως 181, η οποία τίθεται ως είσοδο στο Node SET της μεταβλητής Max Walk Speed. Η μεταβλητή Max Walk Speed αναφέρεται στην ταχύτητα του εχθρού. Για να ληφθεί ή να τεθεί η Max Walk Speed χρησιμοποιείται η μεταβλητή Character Movement, η οποία έχει πληροφορίες για την κίνηση του χαρακτήρα. Όταν τελειώσει το Timeline_2 υπάρχει μια Δομή An που ελέγχει αν η συνάρτηση Get Max Speed (επιστρέφει την μεταβλητή Max Walk Speed) ισούται με 181. Αν η Δομή An είναι αληθής τότε θα ελέγξει ξανά την μεταβλητή Shouldistorspawn01 αν είναι αληθής και θα καλέσει την DoOnce ώστε να τρέξουν τα υπόλοιπα nodes 1 φορά, εφόσον τρέχουν κάθε Frame λόγω της Event Tick. Σε περίπτωση που η Δομή An πριν το Timeline_2 είναι ψευδής, γίνεται Reverse (αντιστροφή) διαδοχικά της τιμής New Track 1 και το Max Walk Speed του εχθρού θα μικραίνει. Στο τέλος και των 2 Event υπάρχει μια Δομή An που ελέγχει την μεταβλητή Runparticle αν είναι αληθής, έτσι ώστε να καλεστεί η συνάρτηση Spawn Emitter at Location (Εικόνα 45), η οποία θα εμφανίσει ένα particle system στην οθόνη. Χρησιμοποιείται ως είσοδο της συνάρτησης Spawn Emitter at Location, το Location του εχθρού, το οποίο γίνεται προσβάσιμο με την συνάρτηση GetWorldLocation που έχει ως είσοδο την μεταβλητή Mesh (αναφέρεται στον εχθρό). Επίσης προστίθεται το νούμερο 130 στον άξονα Y για λόγους αισθητικούς. Στην συνέχεια καλείται η συνάρτηση Delay για 2 δευτερόλεπτα και καταστρέφει με την συνάρτηση Destroy Actor τον εχθρό, ώστε να εξαφανιστεί από το παιχνίδι. Η Delay προστέθηκε για λόγους debugging (αποσφαλμάτωσης). Τέλος καλείται το Event Move Again Now (Εικόνα 46) του BP Survivor και τίθεται η μεταβλητή Hit it (θα αναφερθεί στην επόμενη παράγραφο Εικόνα 47) του BP Survivor σε "DemonCharacter_C_1.StaticMesh Sphere". Αυτό γίνεται γιατί η Unreal κάθε φορά που καταστρέφεται ένας εχθρός αλλάζει τα ονόματα των Component που περιείχε σε περίπτωση που ξαναεμφανιζόταν και στην συγκεκριμένη περίπτωση άλλαζε τον χαρακτήρα "1" της μεταβλητής Hit it, ο

οποίος ήταν "0" πριν καταστραφεί. Όσον αφορά το Event Move Again Now (Εικόνα 46), απλά τίθεται η μεταβλητή Combatstaystill αληθής, εφόσον έχει κληθεί η συνάρτηση delay για 3 δευτερόλεπτα για λόγους συγχρονισμού.

Στην Εικόνα 47 έχει δημιουργηθεί ένα LineTrace όπως και στο mechanic 10, με μόνη διαφορά στον πολλαπλασιασμό, όπου αντί για 500 έχει μπει 1000 για να επιμηκυνθεί το LineTrace. Επίσης έχει προστεθεί και η μεταβλητή New Var 2 (αναφέρεται στον χαρακτήρα Survivor) ως είσοδος στην Actors to ignore στην συνάρτηση LineTraceByChannel, ώστε ο χαρακτήρας να αγνοείται από το LineTrace. Η συνάρτηση αυτή τρέχει μόνο όταν η αρχική Δομή Av είναι αληθής όταν, δηλαδή η μεταβλητή HitwithFlashlight είναι αληθής, για την οποία έγινε αναφορά στο mechanic 2. Στην συνέχεια φαίνεται στην εικόνα το άσπρο βελάκι να πηγαίνει προς τα πάνω, διότι υπάρχουν τα ίδια nodes με διαφορετικές μεταβλητές, λόγω του ότι εμφανίζονται πολλοί εχθροί μέσα στο παιχνίδι. Με το LineTrace ελέγχεται αν η έξοδος Out Hit Hit Component ισούται με την μεταβλητή Sprawn 1 (αντίστοιχη με την Hit It απλά ενός άλλου εχθρού) που έχει ως τιμή το όνομα ενός αόρατου component static mesh στο κεφάλι του εχθρού (το οποίο δεν φαίνεται μέσα στο παιχνίδι Εικόνα 48). Έτσι αν ο χαρακτήρας σημαδεύει με τον φακό ανοιχτό (γιατί τότε η Hitwithflashlight είναι αληθής στο mechanic 2) το κεφάλι του εχθρού η Δομή Av, μετά την συνάρτηση της LineTrace θα έχει έξοδο αληθής. Στην συνέχεια καλείται η Δομή Av που ελέγχει την μεταβλητή Shouldsprawn 01, η οποία στην αρχή είναι ψευδής, με αποτέλεσμα να καλείται η συνάρτηση Sprawn Sound 2D, ώστε να ακουστεί ένας συγκεκριμένος ήχος όταν ο χρήστης χτυπήσει τον εχθρό. Μετά τίθεται η μεταβλητή Shouldsprawn 01 αληθής. Αν ο χρήστης χάσει τον στόχο του η δομή Av που ελέγχει την ισότητα της Sprawn 1 θα γίνει ψευδής, η δομή Av της Shouldsprawn 01 θα έχει ως έξοδο αληθής και θα καλεστεί η συνάρτηση Fade Out για να σταματήσει ο ήχος που εμφανίστηκε πριν με την Sprawn Sound 2D. Στην συνέχεια τίθεται η μεταβλητή Shouldsprawn 01 ψευδής. Όλο αυτό έχει ως αποτέλεσμα, αν καταφέρει ο χρήστης να σημαδεύει τον εχθρό στο κεφάλι για ένα συγκεκριμένο χρονικό διάστημα, να εμφανιστεί το Particle System που αναφέρθηκε παραπάνω, εφόσον η μεταβλητή Shouldsprawn 01 θα είναι αληθής και ο εχθρός θα έχει εξαφανιστεί.

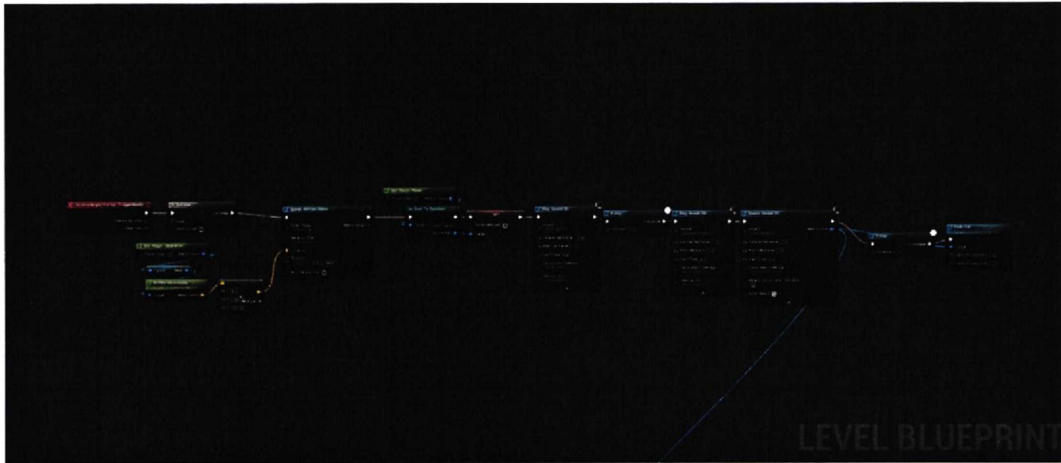
Στην Εικόνα 49 φαίνεται το Event On Component Begin Overlap (Box) το οποίο καλείται όταν το Box (Εικόνα 50) έρθει σε επαφή με τον χαρακτήρα Survivor. Αφού έρθει σε επαφή ελέγχεται με την Δομή Av, αν το όνομα του Other Actor ισούται με το όνομα του χαρακτήρα Survivor. Εφόσον επιστρέψει αληθής τίθεται η μεταβλητή Ithits Stop Running ψευδής, ώστε να σταματήσει να τρέχει η συνάρτηση Add Movement Input (Εικόνα 44) και καλείται η συνάρτηση Play Animation. Ως είσοδο στην Play Animation δίνονται οι μεταβλητές Mesh (αναφέρεται στο Skeletal Mesh του χαρακτήρα-εχθρού) και το Animation με το οποίο κάνει επίθεση ο εχθρός. Στην συνέχεια τίθεται η μεταβλητή RunParticle ψευδής, ώστε να μην εμφανίζεται το Particle System που εμφανίζεται μόνο όταν νικηθεί ο εχθρός. Τέλος, καλείται το Event Deathsecondlevel (Εικόνα 51) του BP Survivor και η συνάρτηση Delay, έτσι

ώστε μετά από 3 δευτερόλεπτα να εξαφανιστεί ο εχθρός από το παιχνίδι με την Destroy Actor.

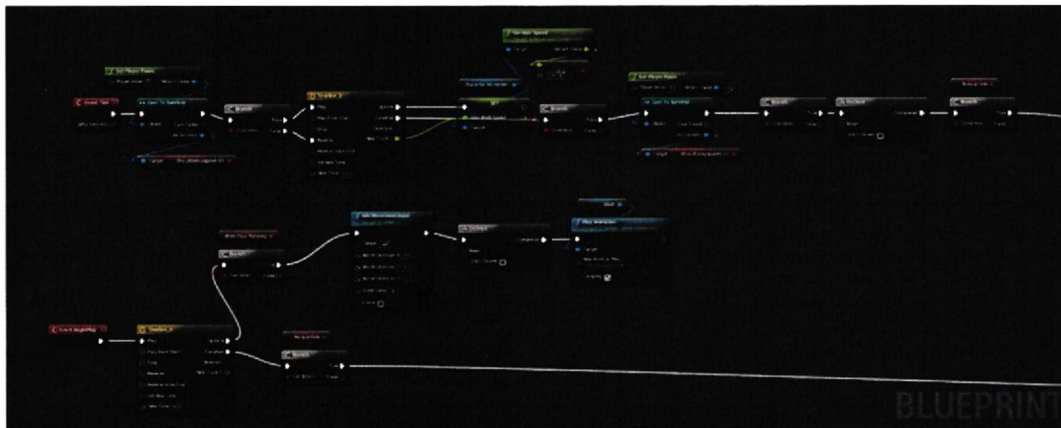
Στην Εικόνα 51 φαίνεται το Event Deathsecondlevel, όπου η μεταβλητή Hedied είναι εξαρχής ψευδής και έτσι η Δομή Αν επιστρέφει ψευδής. Στην συνέχεια τίθεται η τιμή Hedied αληθής, διότι η συνάρτηση πρέπει να τρέξει μόνο 1 φορά, εφόσον είναι δυνατόν να επιτεθεί στον χαρακτήρα από ένας έως τρεις εχθροί και ουσιαστικά να καλεστεί το Event δύο φορές. Η συνάρτηση Delay τρέχει για 0.5 δευτερόλεπτα για λόγους συγχρονισμού. Μετά την Delay καλείται το Event Call stopbreathing (θα αναφερθεί στην επόμενη παράγραφο Εικόνα 52) και παίζει ένας ήχος θανάτου με την συνάρτηση Play Sound2D. Ο χαρακτήρας (μεταβλητή Mesh) θα εκτελέσει το Animation θανάτου με την συνάρτηση Play Animation. Στην συνέχεια καλείται μια συνάρτηση Delay για 1.2 δευτερόλεπτα για λόγους συγχρονισμού και απενεργοποιούνται τα input του χρήστη με την συνάρτηση Disable Input η οποία έχει ως είσοδο την συνάρτηση Get Player Controller που περιέχει τα input του χρήστη. Αυτό έχει ως αποτέλεσμα ο χαρακτήρας να μην μπορεί να κινηθεί. Μετά δημιουργείται ένα Widget με την συνάρτηση Create Endgame Widget και εμφανίζεται στην οθόνη με την συνάρτηση Add to Viewport. Επίσης τίθεται η έξοδος της συνάρτησης Create Widget στην μεταβλητή End Game UI. Εφόσον δημιουργήθηκε το widget, καλείται μια συνάρτηση Delay για 1 δευτερόλεπτο για λόγους συγχρονισμού και τίθεται η μεταβλητή Combatstaystill αληθής. Τέλος καλείται ένα Event Call runteleport3 (θα αναφερθεί στην επόμενη παράγραφο Εικόνα 52) και τίθεται η μεταβλητή Hedied ψευδής, ώστε να μπορεί να ξανακαλεστεί το Event Deathsecondlevel.

Στην Εικόνα 52 φαίνεται ο τρόπος με τον οποίο χρησιμοποιείται το Event runteleport3. Το άσπρο βελάκι που έχει ως είσοδο το node Bind Event to runteleport3 συνδέεται με το Event BeginPlay. Αυτός ο τρόπος χρησιμοποιείται έτσι ώστε να μπορεί να καλεστεί ένα Event του Level Blueprint από άλλα Blueprint. Το Event teleport3 καλείται κατά το Event Deathsecondlevel που αναφέρθηκε στην προηγούμενη παράγραφο από το BP Survivor. Εφόσον ο χαρακτήρας πεθάνει θα γίνει μεταφορά του χαρακτήρα μέσω της συνάρτησης Teleport στο Location της μεταβλητής TriggerBox6 (αναφέρεται στο αντικείμενο Εικόνα 53). Η συνάρτηση Teleport έχει ως είσοδο τον χαρακτήρα Survivor και το Location της μεταβλητής TriggerBox6 το οποίο παίρνεται με την συνάρτηση GetActorLocation. Στην συνέχεια καλείται η συνάρτηση Delay για 1.5 δευτερόλεπτα για λόγους συγχρονισμού, καθώς καλείται και η συνάρτηση Enable Input που ενεργοποιεί τα input του χρήστη, ώστε ο χαρακτήρας να μπορεί να κινηθεί. Μετά καλώντας την συνάρτηση Set Animation Mode, επιλέγεται το Animation Mode του χαρακτήρα, δηλαδή επιλέγονται οι κινήσεις του χαρακτήρα όταν περπατάει, όταν τρέχει ή όταν μένει ακίνητος. Στη συνέχεια καλείται η συνάρτηση Delay για 2 δευτερόλεπτα για λόγους συγχρονισμού. Εξαφανίζεται από την οθόνη το widget που δημιουργήθηκε πριν, δηλαδή το End Game UI καλώντας την συνάρτηση Remove from Parent. Μετά τίθεται η μεταβλητή Sprawnagain ψευδής, η οποία χρησιμοποιείται για την επανεμφάνιση κάποιων εχθρών σε συγκεκριμένα σημεία. Τέλος τίθεται το intensity

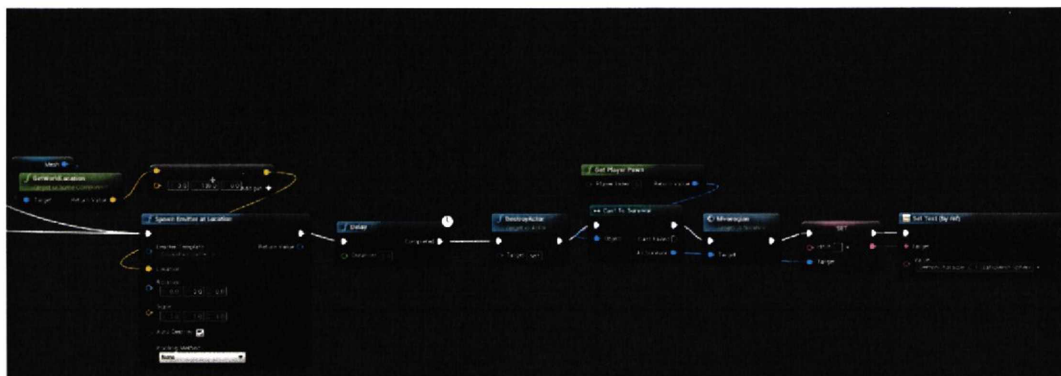
του Spot Light 1 σε 0, διότι ο χαρακτήρας θα κάνει teleport σε σημείο έξω από το σκοτεινό δωμάτιο, στο οποίο το intensity του spotlight που έχει τοποθετηθεί πάνω από τον χαρακτήρα (Εικόνα 54) είναι μεγαλύτερο του 0.



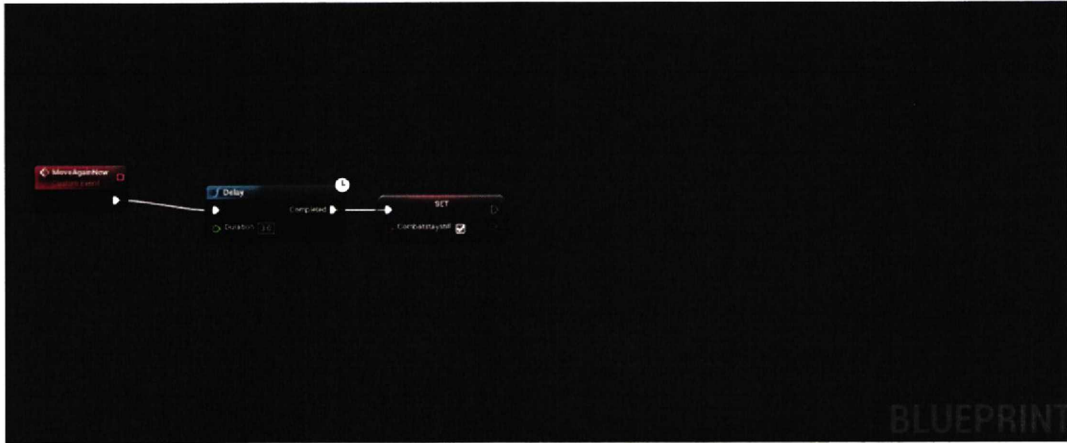
Εικόνα 43 (Εμφάνιση εχθρού στο Level2 BP)



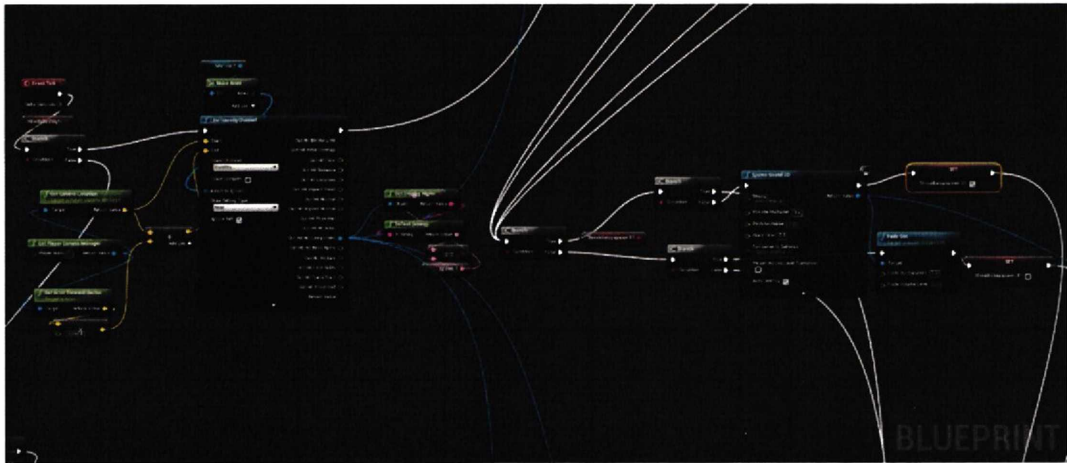
Εικόνα 44 (Κίνηση εχθρού προς χαρακτήρα στο spawn01 BP)



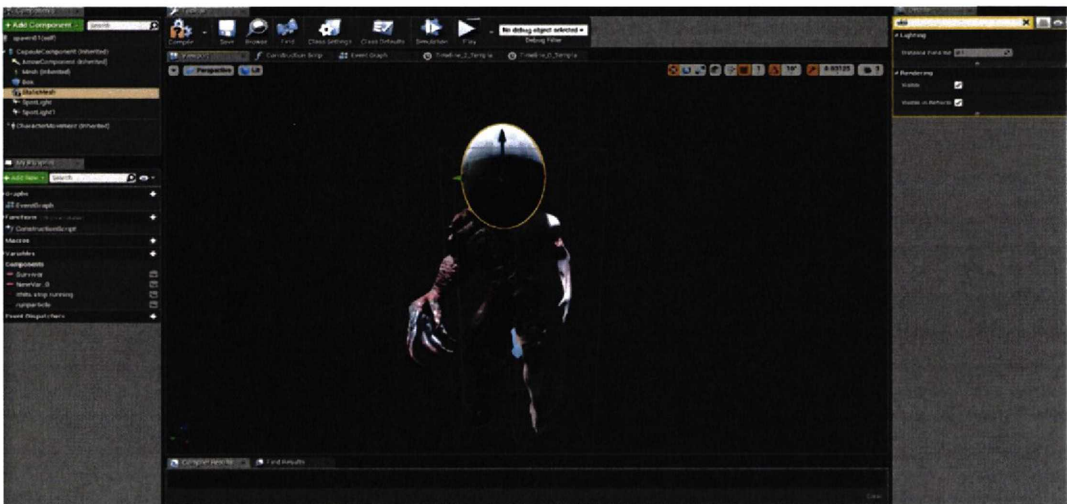
Εικόνα 45 (Κίνηση εχθρού προς χαρακτήρα στο spawn01 BP)



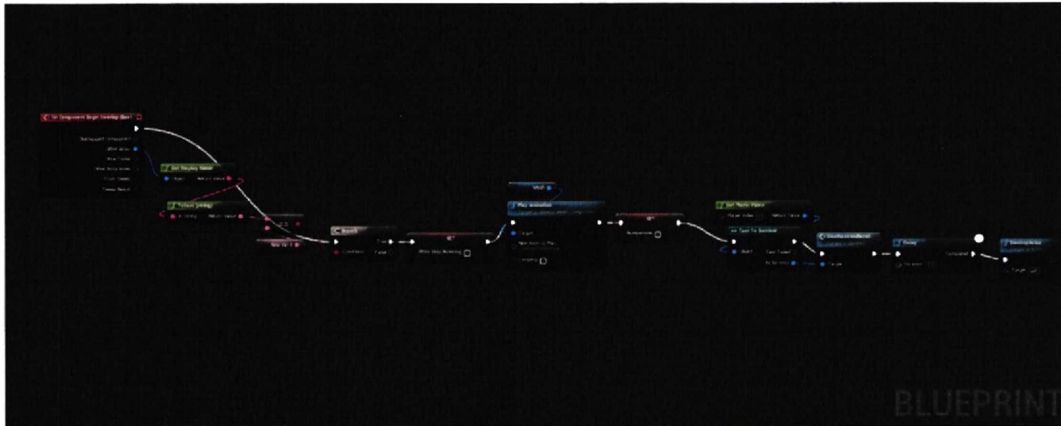
Εικόνα 46 (Ενεργοποίηση κίνησης χαρακτήρα στο Survivor BP)



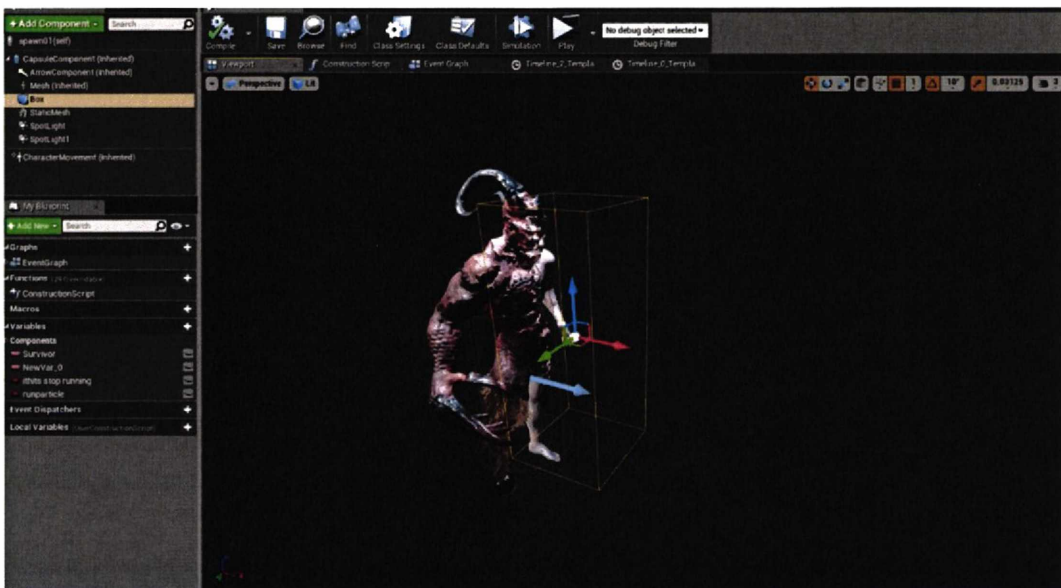
Εικόνα 47 (Αναγνώριση του κεφαλιού του εχθρού στο Survivor BP)



Εικόνα 48 (Static Mesh που χρησιμοποιήθηκε στην αναγνώριση στο spawn01 BP)



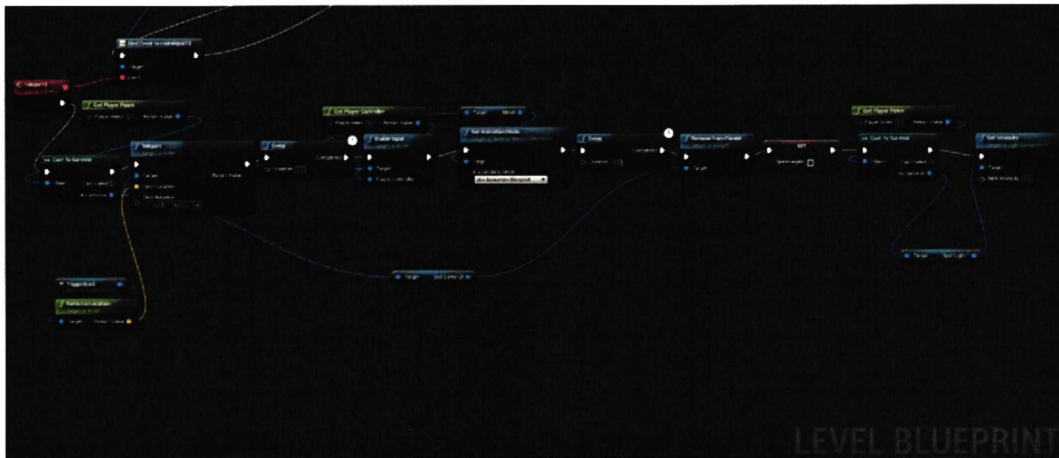
Εικόνα 49 (Επίθεση εχθρού στον χαρακτήρα στο sprawn01 BP)



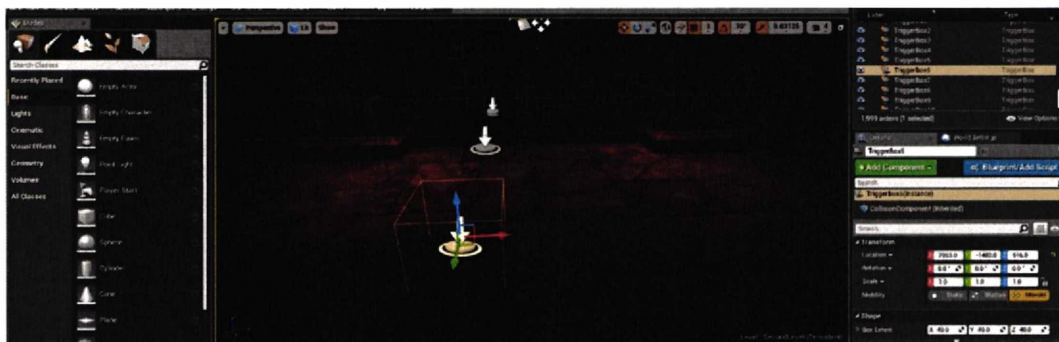
Εικόνα 50 (Box που χρησιμοποιήθηκε στην αναγνώριση του χαρακτήρα από τον εχθρό στο sprawn01 BP)



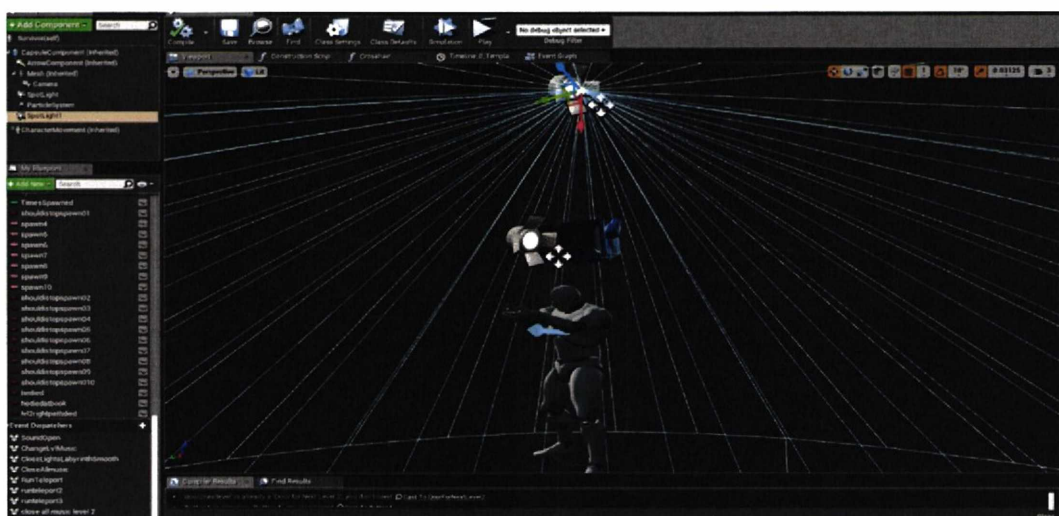
Εικόνα 51 (Θάνατος χαρακτήρα στο Survivor BP)



Εικόνα 52 (Μεταφορά χαρακτήρα μετα τον θάνατο στο Level2 BP)



Εικόνα 53 (Σημείο στο οποίο γίνεται η μεταφορά)



Εικόνα 54 (Spotlight του χαρακτήρα που χρησιμοποιείται μέσα στο σκοτεινό δωμάτιο στο Survivor BP)

5.2 Σημειώσεις

Όλα τα αναφερόμενα Blueprints επιδέχονται βελτίωση όσον αφορά την απόδοση τους (optimization).

6. Μελλοντικές Προεκτάσεις

Στο μέλλον προβλέπεται να προστεθούν νέα επίπεδα και mechanics. Επίσης προβλέπεται η βελτιστοποίηση των blueprints και η βελτίωση των γραφικών του παιχνιδιού.

7. Πηγές για τα asset που χρησιμοποιήθηκαν

1. https://www.solarsystemscope.com/textures/download/2k_moon.jpg
from <https://www.solarsystemscope.com/textures/>
2. <https://www.unrealengine.com/en-US/branding>
3. free background music from <https://www.fesliyanstudios.com>
4. <http://www.music-note.jp/bgm/mp3/0801/darkshadow.mp3> Music materials providing MusicNoteWorld <http://en.music-note.jp/>
5. Music materials providing <https://www.zapsplat.com>
6. Music materials providing https://www.freesoundeffects.com/free-sounds/doors-10030/0/low_high/20/1/
7. <https://www.turbosquid.com/FullPreview/Index.cfm/ID/306184>
8. <https://forums.unrealengine.com/community/community-content-tools-and-tutorials/16041-free-flashlight-for-you-all>
9. <https://forums.unrealengine.com/community/community-content-tools-and-tutorials/92563-free-aaa-quality-horror-game-monster-animations-and-ai-included>
10. <https://www.turbosquid.com/FullPreview/Index.cfm/ID/1294095>
11. <https://pngimage.net/old-letter-png/>

8. Βιβλιογραφία

1. What is a Game Engine?

(http://www.gamecareerguide.com/features/529/what_is_a_game_.php)

2. Sweeney, Tim (2005). "GPU Gems 2 – Foreword". Nvidia Developer. Archived from the original on October 3, 2017. (https://developer.nvidia.com/gpugems/GPUGems2/gpugems2_frontmatter.html)

3. Nutt, Christian (March 21, 2014). "Epic's Tim Sweeney lays out the case for Unreal Engine 4"

(https://www.gamasutra.com/view/news/213647/Epics_Tim_Sweeney_lays_out_the_case_for_Unreal_Engine_4.php)

4. Keighley, Geoffrey. "Blinded By Reality: The True Story Behind the Creation of Unreal"

(<https://web.archive.org/web/20010519154729/http://www.gamespot.com/features/makeunreal/>)

5. Horvath, Stu (May 17, 2012). "The Imagination Engine: Why Next-Gen Videogames Will Rock Your World" (<https://www.wired.com/2012/05/ff-unreal4/>)

6. Lily, Paul (July 21, 2009). "Doom to Dunia: A Visual History of 3D Game Engines" (https://web.archive.org/web/20090724065520/http://www.maximumpc.com/article/features/3d_game_engines?page=0%2C3)

7. "Unreal". *boot*. No. Aug/Sep 96 (Special Premiere Collector's ed.). p. 87

(<https://archive.org/stream/boot-magazine-vol01-issue01-sept-1996/Boot%20Magazine%20-%20Vol%201%20Issue%201%20August%20September%201996#page/n83/mode/2up>)

8. Smith, Tom (July 16, 2013). "Unreal: Epic's would-be Doom... er... Quake killer"

(https://www.theregister.co.uk/2013/07/16/antique_code_show_unreal/)

9. Wilson, Billy; O'Brien, Chris; Heir, Manveer; Howarth, Robert (May 25, 2000). "Talkin' nasty with Epic's Tim Sweeney"

(<https://web.archive.org/web/20000620093601/http://www.voodooextreme.com/games/interviews/timsweeney.html>)

10. Thomsen, Mike (February 23, 2010). "History of the Unreal Engine"

(<https://www.ign.com/articles/2010/02/23/history-of-the-unreal-engine>)

11. Burnes, Andre (June 8, 2012). "Epic Reveals Stunning Elemental Demo, & Tim Sweeney On Unreal Engine 4"

(<https://web.archive.org/web/20120610175656/http://www.geforce.com/whats->

new/articles/stunning-videos-show-unreal-engine-4s-next-gen-gtx-680-powered-real-time-graphics/)

12. Kosak, Dave. "Talking with Tim Sweeney about the evolution of a game engine" (https://web.archive.org/web/20010621202712/http://www.gamespy.com/legacy/interviews/sweeney_a.shtml)

13. McLeroy, Carrie (August 27, 2008). "Improving "America's Army"" (https://www.army.mil/article/11935/improving_americas_army)

14. Kennedy, Brian (July 11, 2002). "Uncle Sam Wants You (To Play This Game)" (<https://www.nytimes.com/2002/07/11/technology/uncle-sam-wants-you-to-play-this-game.html>)

15. McLean-Foreman, John (April 6, 2001). "An Interview with Epic Games' Tim Sweeney" (https://www.gamasutra.com/view/feature/131502/an_interview_with_epic_games_tim.php)

16. Reed, Kristan (July 1, 2004). "Unreal Engine 3 Interview" (https://www.eurogamer.net/articles/i_epicgames_june04)

17. Maximum PC. "Game Engines – Exposed!" (https://archive.org/details/Maximum_PC_Autumn_2004)

18. Caron, Frank (March 13, 2008). "Unreal Engine 4 to "exclusively target" next-gen consoles" (<https://arstechnica.com/gaming/2008/03/unreal-engine-4-to-exclusively-target-next-gen-consoles/>)

19. Houlihan, John (August 18, 2005). "Rein: "We've been working on Unreal Engine 4 for two years"" (<https://web.archive.org/web/20140110101539/http://www.computerandvideogames.com/123639/rein-weve-been-working-on-unreal-engine-4-for-two-years/>)

20. Valich, Theo (March 12, 2008). "Tim Sweeney, Part 3: Unreal Engine 4.0 aims at next-gen console war" (<http://www.tgdaily.com/business-and-law-features/36436-tim-sweeney-part-3-unreal-engine-40-aims-at-next-gen-console-war>)

21. Papadopoulos, John (April 20, 2013). "Epic Games' Tim Sweeney Explains Lack Of 'SVOGI' In Unreal Engine 4" (<https://www.dsogaming.com/news/epic-games-tim-sweeney-explains-lack-of-svogi-in-unreal-engine-4/>)

22. Grayson, Nathan (July 20, 2012). "Fortnite's Jessen Talks Minecraft, PC Gaming, UE4" (<https://www.rockpapershotgun.com/2012/07/20/fortnites-jessen-talks-minecraft-pc-gaming-ue4/>)

23. Evangelho, Jason (May 22, 2014). "Why Is Epic Games Promoting Unreal Engine 4 With A 'Flappy Bird' Clone?" (<https://www.forbes.com/sites/jasonevangelho/2014/05/22/why-is-epic-games-promoting-unreal-engine-4-with-a-flappy-bird-clone/>)

24. Thier, Dave (June 29, 2012). "Epic's Tim Sweeney on How Unreal Engine 4 Will Change The Way Games Are Made, and Why You Care" (<https://www.forbes.com/sites/davidthier/2012/06/29/epics-tim-sweeney-on-how-unreal-engine-4-will-change-the-way-games-are-made-and-why-you-care/>)
25. Totilo, Stephen (June 8, 2012). "How Unreal Engine 4 Will Change The Next Games You Play" (<https://kotaku.com/how-unreal-engine-4-will-change-the-next-games-you-play-5916859>)
26. Soubeyrand, Catherine (2000). «The Royal Game of Ur» (<http://www.gamecabinet.com/history/Ur.html>)
27. Green, William (2008-06-19). «Big Game Hunter» (http://content.time.com/time/specials/2007/article/0,28804,1815747_1815707_1815665,00.html)
28. U.S. Patent 2,455,992 (<https://patents.google.com/patent/US2455992>)
29. <https://www.theesa.com/wp-content/uploads/2019/05/2019-Essential-Facts-About-the-Computer-and-Video-Game-Industry.pdf>
30. <https://docs.unrealengine.com/en-US/Engine/UI/LevelEditor/Modes/index.html>
31. <https://docs.unrealengine.com/en-US/Engine/UI/LevelEditor/Viewports/Basics/index.html>
32. <https://docs.unrealengine.com/en-US/Engine/Content/Browser/index.html>
33. <https://docs.unrealengine.com/en-US/Engine/UI/LevelEditor/SceneOutliner/index.html>

