



**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής**

Πτυχιακή εργασία

**Τίτλος: Παιχνίδι Capture Towers στην Unreal
Engine 4**

**Δούβρης Γρηγόριος (ΑΜ: 3252)
Προεστός Μάριος (ΑΜ: 3258)**

Επιβλέπων εκπαιδευτικός : Παχουλάκης Ιωάννης

ΗΡΑΚΛΕΙΟ

2019

Abstract

In the context of the thesis, a game will be developed in Unreal Engine 4, which will include a big level, there will be towers (outposts) scattered on the map and each tower will possess different enemies. The player will be able to choose freely in which order he wants to conquer the towers. The goal is to defeat the enemy of the last tower, in order to beat the game. The character, which the player will control, will level up based on the towers which he conquered, in order to reach the power of the enemies of the next towers.

Σύνοψη

Στο πλαίσιο της πτυχιακής εργασίας, θα αναπτυχθεί ένα παιχνίδι στην Unreal Engine 4, το οποίο θα περιλαμβάνει ένα μεγάλο επίπεδο, θα υπάρχουν πύργοι (φυλάκια) σκορπισμένοι στο χάρτη και κάθε πύργος θα έχει διαφορετικούς αντιπάλους. Ο παίκτης θα μπορεί να επιλέξει ελεύθερα με ποια σειρά θέλει να κατακτήσει τους πύργους. Στόχος είναι να νικήσει τον αντίπαλο του τελευταίου πύργου, ώστε να ολοκληρωθεί το παιχνίδι. Ο χαρακτήρας, που θα χειρίζεται ο παίκτης, θα ανεβαίνει επίπεδο με βάση τους πύργους που έχει κατακτήσει, έτσι ώστε να φτάσει σε δύναμη τους αντιπάλους των επόμενων πύργων.

Πίνακας Περιεχομένων

Abstract	ii
Σύνοψη	iii
Πίνακας Περιεχομένων	iv
Λίστα Εικόνων	v
1. Εισαγωγή	1
1.1. Περίληψη.....	1
1.2. Κίνητρο για την Διεξαγωγή της Εργασίας - Στόχοι.....	1
1.3. Δομή Εργασίας	1
2. Μεθοδολογία Υλοποίησης	2
3. Υλοποίηση Πτυχιακής Εργασίας	3
3.1. MakeHuman	3
3.2. Blender	5
3.2.1. Δημιουργία Αντικειμένων	5
3.2.2. Animations	11
3.3. Unreal Engine 4	15
3.3.1. Level Design.....	18
3.3.1.1. Δημιουργία Χάρτη.....	18
3.3.1.2. Τοποθέτηση και Επεξεργασία Αντικειμένων	22
3.3.1.3. Υλικά και Εφέ.....	27
3.3.2. Blueprints και Προγραμματισμός Παιχνιδιού.....	37
3.3.2.1. Character Blueprints.....	37
3.3.2.2. Animation Blueprints	53
3.3.2.3. User Interface and Widget Blueprints.....	67
3.3.2.4. Level Blueprints and Final Settings.....	75
4. Αποτελέσματα	79
4.1. Συμπεράσματα	79
4.2. Μελλοντική Εργασία και Επεκτάσεις.....	79
Βιβλιογραφία	80

Λίστα Εικόνων

Εικόνα 1: Ρυθμίσεις Χαρακτήρα στο MakeHuman	3
Εικόνα 2: Σκελετός Χαρακτήρα στο MakeHuman	3
Εικόνα 3: Εξαγωγή Χαρακτήρα από το MakeHuman.....	4
Εικόνα 4: Create Menu στο Blender.....	5
Εικόνα 5: Τρισδιάστατο Σχήμα του Blender	5
Εικόνα 6: Στοιχεία Σχήματος στο Blender	6
Εικόνα 7: Τροποποιημένο Σχήμα στο Blender	6
Εικόνα 8: Ρυθμίσεις Σχήματος στο Blender	7
Εικόνα 9: Modifier στο Blender.....	7
Εικόνα 10: Τροποποίηση Σχήματος με Difference Boolean Modifier στο Blender	7
Εικόνα 11: Αποτέλεσμα Τροποποίησης Σχήματος με Difference Boolean Modifier στο Blender.....	8
Εικόνα 12: Edit Mode στο Blender	8
Εικόνα 13: Tools Menu στο Blender.....	9
Εικόνα 14: Σχήματα Ενωμένα και Ομαδοποιημένα στο Blender	9
Εικόνα 15: Χρωματισμός Material των Αντικειμένων στο Blender	10
Εικόνα 16: Χρωματισμένα Αντικείμενα στο Blender	10
Εικόνα 17: Τελικό Αντικείμενο στο Blender.....	11
Εικόνα 18: Επιλογή Bone στο Blender	11
Εικόνα 19: Επιλογή Σκελετού στο Blender.....	12
Εικόνα 20: Dope Sheet στο Blender	12
Εικόνα 21: Επιλογή Animation στο Blender.....	13
Εικόνα 22: Timeline στο Blender	13
Εικόνα 23: Transform Ρυθμίσεις στο Blender.....	14
Εικόνα 24 Frame στο Blender	14
Εικόνα 25 Frame στο Blender	14
Εικόνα 26: Γενικές Ρυθμίσεις Εξαγωγής από το Blender	15
Εικόνα 27: Ρυθμίσεις Εξαγωγής Αντικειμένων από το Blender.....	16
Εικόνα 28: Ρυθμίσεις Εξαγωγής Χαρακτήρα με Σκελετό από το Blender.....	16
Εικόνα 29: Ρυθμίσεις Εξαγωγής Animations από το Blender	16
Εικόνα 30: Ρυθμίσεις Εισαγωγής Αντικειμένων στο Unreal Engine 4	17
Εικόνα 31: Επιλογές Menu στην Unreal Engine 4	18
Εικόνα 32: Landscape Menu στην Unreal Engine 4	18
Εικόνα 33: Έδαφος Επιπέδου στην Unreal Engine 4.....	19
Εικόνα 34: Έδαφος Επιπέδου στην Unreal Engine 4.....	19
Εικόνα 35: World Outliner στην Unreal Engine 4.....	20
Εικόνα 36: Επιλογή Sculpt στην Unreal Engine 4	20
Εικόνα 37 Sculpt Menu στην Unreal Engine 4.....	21
Εικόνα 38: Αλλαγή Σχήματος Εδάφους στην Unreal Engine 4.....	21
Εικόνα 39: Έτοιμα Αντικείμενα στην Unreal Engine 4	22
Εικόνα 40: Έτοιμα Αντικείμενα στην Unreal Engine 4	22
Εικόνα 41: Ρυθμίσεις του Details Tab στην Unreal Engine 4	23
Εικόνα 42: Edit Mode στην Unreal Engine 4	23
Εικόνα 43: Σχήμα σε Edit Mode στην Unreal Engine 4	23

Εικόνα 44: Επιλογές Αντικειμένου στην Unreal Engine 4	24
Εικόνα 45: Πυρσός στην Unreal Engine 4	24
Εικόνα 46: Πυρσός στην Unreal Engine 4	25
Εικόνα 47: Επιλογή Manage στην Unreal Engine 4.....	25
Εικόνα 48: Μονοπάτι από Splines στην Unreal Engine 4.....	26
Εικόνα 49: Ρυθμίσεις Mesh στην Unreal Engine 4.....	26
Εικόνα 50: Splines Μονοπάτια στην Unreal Engine 4.....	27
Εικόνα 51: Content Browser στην Unreal Engine 4.....	27
Εικόνα 52: Κτίριο που έχει εισαχθεί από το Blender στην Unreal Engine 4.....	28
Εικόνα 53: Materials Αντικειμένου στην Unreal Engine 4	28
Εικόνα 54: Ομαδική Επιλογή Material στην Unreal Engine 4.....	29
Εικόνα 55: Επιλογή Material στην Unreal Engine 4	29
Εικόνα 56: Δημιουργία Νέου Material στην Unreal Engine 4.....	29
Εικόνα 57: Material Blueprint στην Unreal Engine 4	30
Εικόνα 58: Material Preview στην Unreal Engine 4	30
Εικόνα 59: Material Blueprint στην Unreal Engine 4	31
Εικόνα 60: Ρυθμίσεις Texture στην Unreal Engine 4	31
Εικόνα 61: Material Preview στην Unreal Engine 4	32
Εικόνα 62: Landscape Material στην Unreal Engine 4	32
Εικόνα 63: Material Layers στην Unreal Engine 4.....	33
Εικόνα 64: Παράδειγμα χρήσης του Paint με το ground material στην Unreal Engine 4.....	33
Εικόνα 65: Material Blueprint στην Unreal Engine 4	34
Εικόνα 66: Material Color στην Unreal Engine 4.....	34
Εικόνα 67: Material Preview στην Unreal Engine 4	35
Εικόνα 68: Material Blueprint στην Unreal Engine 4	35
Εικόνα 69: Material Blueprint στην Unreal Engine 4	35
Εικόνα 70: Material Blueprint στην Unreal Engine 4	36
Εικόνα 71: Lightning Material Preview στην Unreal Engine 4.....	37
Εικόνα 72: Sockets του Σκελετού στην Unreal Engine 4	38
Εικόνα 73: Αντικείμενα πάνω στο Μοντέλο του Χαρακτήρα στην Unreal Engine 4	39
Εικόνα 74: Μεταβλητές Player Character στην Unreal Engine 4	40
Εικόνα 75: Player Character Blueprint στην Unreal Engine 4	41
Εικόνα 76: Player Character Blueprint στην Unreal Engine 4	41
Εικόνα 77: Player Character Blueprint στην Unreal Engine 4	42
Εικόνα 78: Player Character Blueprint στην Unreal Engine 4	43
Εικόνα 79: Player Character Blueprint στην Unreal Engine 4	44
Εικόνα 80: Player Character Blueprint στην Unreal Engine 4	45
Εικόνα 81: Player Character Blueprint στην Unreal Engine 4	45
Εικόνα 82: Player Character Blueprint στην Unreal Engine 4	46
Εικόνα 83: Player Character Blueprint στην Unreal Engine 4	46
Εικόνα 84: Player Character Blueprint στην Unreal Engine 4	47
Εικόνα 85: Player Character Blueprint στην Unreal Engine 4	47
Εικόνα 86: Μεταβλητές NPC στην Unreal Engine 4	48
Εικόνα 87: Non-Player Character Blueprint στην Unreal Engine 4	49
Εικόνα 88: Non-Player Character Blueprint στην Unreal Engine 4	49

Εικόνα 89: Non-Player Character Blueprint στην Unreal Engine 4	50
Εικόνα 90: Non-Player Character Blueprint στην Unreal Engine 4	51
Εικόνα 91: Non-Player Character Blueprint στην Unreal Engine 4	51
Εικόνα 92: Non-Player Character Blueprint στην Unreal Engine 4	52
Εικόνα 93: Non-Player Character Child Blueprints στην Unreal Engine 4.....	53
Εικόνα 94: Player Character Animation Blueprint στην Unreal Engine 4.....	54
Εικόνα 95: Non-Player Character Animation Blueprint στην Unreal Engine 4.....	54
Εικόνα 96: Player Character Animation Blueprint στην Unreal Engine 4.....	55
Εικόνα 97: Player Character Animation Blueprint στην Unreal Engine 4.....	55
Εικόνα 98: Non-Player Character Animation Blueprint στην Unreal Engine 4.....	56
Εικόνα 99: Player Character Animation Blueprint στην Unreal Engine 4.....	56
Εικόνα 100: Player Character Animation Blueprint στην Unreal Engine 4	57
Εικόνα 101: Player Character Animation Blueprint στην Unreal Engine 4	58
Εικόνα 102: Player Character Animation Blueprint στην Unreal Engine 4	59
Εικόνα 103: Player Character Animation Blueprint στην Unreal Engine 4	60
Εικόνα 104: Player Character Animation Blueprint στην Unreal Engine 4	61
Εικόνα 105: Player Character Animation Blueprint στην Unreal Engine 4	61
Εικόνα 106: Player Character Animation Blueprint στην Unreal Engine 4	62
Εικόνα 107: Player Character Animation Blueprint στην Unreal Engine 4	63
Εικόνα 108: Player Character Animation Blueprint στην Unreal Engine 4	64
Εικόνα 109: Player Character Animation Blueprint στην Unreal Engine 4	65
Εικόνα 110: Non-Player Character Animation Blueprint στην Unreal Engine 4.....	66
Εικόνα 111: HUD Widget Blueprint στην Unreal Engine 4.....	67
Εικόνα 112: HUD Widget Blueprint Συναρτήσεις στην Unreal Engine 4.....	68
Εικόνα 113: HUD Widget Blueprint Συνάρτηση στην Unreal Engine 4	69
Εικόνα 114: HUD Widget Blueprint Συναρτήσεις στην Unreal Engine 4.....	70
Εικόνα 115: Pause Menu Widget Blueprint στην Unreal Engine 4	70
Εικόνα 116: Pause Menu Widget Blueprint Συναρτήσεις στην Unreal Engine 4.....	71
Εικόνα 117: Pause Menu Widget Blueprint στην Unreal Engine 4	72
Εικόνα 118: Pause Menu Widget Blueprint στην Unreal Engine 4	72
Εικόνα 119: Pause Menu Widget Blueprint Συνάρτηση στην Unreal Engine 4	72
Εικόνα 120: Main Menu Widget Blueprint στην Unreal Engine 4.....	73
Εικόνα 121: Main Menu Widget Blueprint στην Unreal Engine 4.....	74
Εικόνα 122: Main Menu Widget Blueprint στην Unreal Engine 4.....	74
Εικόνα 123: Main Menu Level Blueprint στην Unreal Engine 4	75
Εικόνα 124: Level Blueprint του Επιπέδου του Παιχνιδιού στην Unreal Engine 4.....	76
Εικόνα 125: Level Blueprint του Επιπέδου του Παιχνιδιού στην Unreal Engine 4.....	76
Εικόνα 126: Level Blueprint του Επιπέδου του Παιχνιδιού στην Unreal Engine 4.....	77
Εικόνα 127: Level Blueprint του Επιπέδου του Παιχνιδιού στην Unreal Engine 4.....	78

1. Εισαγωγή

1.1. Περίληψη

Σε αυτή τη πτυχιακή θα ασχοληθούμε με την δημιουργία ενός παιχνιδιού στην Unreal Engine 4. Θα δούμε την κατασκευή του χαρακτήρα, των animations και των αντικειμένων, που θα χρησιμοποιηθούν ως όπλα και πανοπλία για τον χαρακτήρα ή ως κτίρια για το επίπεδο του παιχνιδιού. Επίσης, θα δούμε πως φτιάχνεται ένα επίπεδο και πως προγραμματίζεται το παιχνίδι.

Ο Δούβρης Γρηγόριος (AM: 3252) απασχολήθηκε με το προγραμματισμό του παιχνιδιού στην Unreal Engine 4 σε Blueprints ([Κεφάλαιο 3.3.2.](#)) και το σχεδιασμό του μοντέλου και του σκελετού του χαρακτήρα στο MakeHuman ([Κεφάλαιο 3.1.](#)).

Ο Προεστός Μάριος (AM: 3258) απασχολήθηκε με τη δημιουργία αντικειμένων και animations στο Blender ([Κεφάλαιο 3.2.](#)) και τη δημιουργία του επιπέδου στην Unreal Engine 4 ([Κεφάλαιο 3.3.1.](#)).

1.2. Κίνητρο για την Διεξαγωγή της Εργασίας - Στόχοι

Την πτυχιακή αυτή την επιλέξαμε γιατί θέλαμε να μάθουμε πως φτιάχνονται τα βιντεοπαιχνίδια, τα οποία είναι μια από τις αγαπημένες μας μορφές διασκέδασης. Ένας άλλος λόγος ήταν να μάθουμε πόσο εύκολο ή δύσκολο είναι κάποιος να ασχοληθεί με την κατασκευή βιντεοπαιχνιδιών, με τη χρήση της τεχνολογίας που είναι διαθέσιμη σήμερα. Επίσης, θέλαμε να χρησιμοποιήσουμε την Unreal Engine 4 γιατί είναι μια καλή και ενδιαφέρουσα μηχανή για την κατασκευή παιχνιδιών και για να χρησιμοποιήσουμε τα Blueprints. Τέλος, θεωρήσαμε ότι αυτή η πτυχιακή θα είναι ξεχωριστή, ως μία από τις λίγες που χρησιμοποιεί την Unreal Engine 4, γιατί μέχρι την ολοκλήρωση αυτής της πτυχιακής, υπάρχουν πάρα πολύ λίγες πτυχιακές που χρησιμοποιούν την Unreal Engine 4 ή προηγούμενη της έκδοση.

Στο τέλος αυτής της πτυχιακής θέλουμε να έχουμε ένα παιχνίδι που μας δείχνει τι μπορούμε να κάνουμε, σε αρχάριο επίπεδο, με την Unreal Engine 4 και να δούμε τι λειτουργίες μπορούμε να δώσουμε στο παιχνίδι μας. Σκοπός μας είναι η ψυχαγωγία και η κίνηση του ενδιαφέροντος των συναδέλφων μας για τη χρήση της Unreal Engine 4 και των Blueprints.

1.3. Δομή Εργασίας

Πρώτα θα δούμε μια γενική ανάλυση της πτυχιακής, των προγραμμάτων και τη ροή εργασίας μας. Έπειτα, θα δούμε την υλοποίηση της εργασίας σε κάθε πρόγραμμα και την ανάλυσή της. Τέλος, θα δούμε τα αποτελέσματα, τα συμπεράσματα και τις πιθανές προσθήκες, που θα μπορούσαν να γίνουν με βάση τις γνώσεις που αποκτήθηκαν κατά τη διάρκεια της εργασίας από τη χρήση των προγραμμάτων.

2. Μεθοδολογία Υλοποίησης

Σε αυτό το κεφάλαιο θα αναφερθούμε στα προγράμματα που θα χρησιμοποιήσουμε για την υλοποίηση της πτυχιακής. Τα προγράμματα που θα χρειαστούμε σε αυτή την πτυχιακή είναι τα MakeHuman, Blender και Unreal Engine 4.

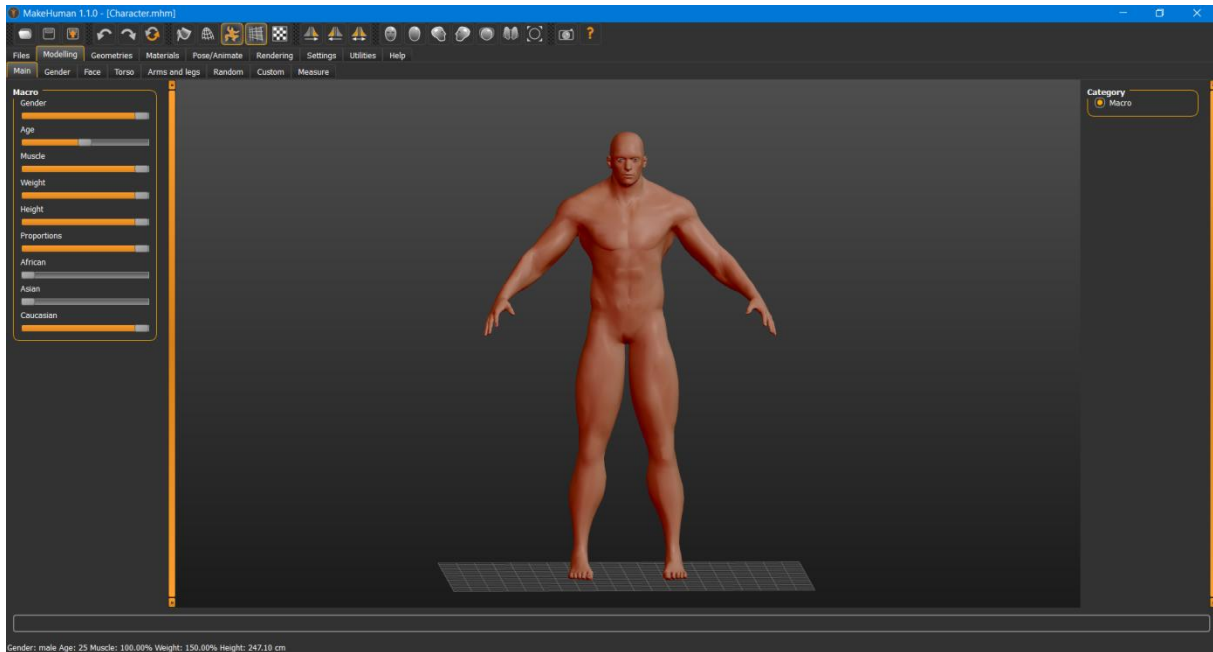
Το MakeHuman είναι ένα δωρεάν πρόγραμμα ανοιχτού κώδικα για εύκολο σχεδιασμό τρισδιάστατων μοντέλων με τη μορφή ανθρώπου. Έχει σχεδιαστεί από μια κοινότητα προγραμματιστών και σχεδιαστών. Το πρόγραμμα λειτουργεί παρόμοια με τα συστήματα δημιουργίας χαρακτήρων που χρησιμοποιούνται από πολλά παιχνίδια, και δίνουν τη δυνατότητα στους παίκτες να φτιάξουν τους χαρακτήρες, που θα χρησιμοποιήσουν στο παιχνίδι, με βάση τις δικές τους προτιμήσεις. Αυτό κάνει το MakeHuman πολύ εύκολο στη χρήση. Για το MakeHuman θα χρειαστούμε το mhx2 plugin, για να μπορέσουμε να εισάγουμε το χαρακτήρα μας στο Blender. Θα το χρησιμοποιήσουμε για το σχεδιασμό του μοντέλου και του σκελετού του χαρακτήρα μας.

Το Blender είναι ένα δωρεάν πρόγραμμα ανοιχτού κώδικα για το σχεδιασμό μοντέλων και τη δημιουργία animations. Έχει σχεδιαστεί από την Blender Foundation. Η έκδοση που χρησιμοποιήθηκε είναι η Blender 2.78c, με εγκατεστημένα τα Blender Tools από το MakeHuman, για να μπορέσουμε να εισάγουμε το χαρακτήρα μας από το MakeHuman. Θα το χρησιμοποιήσουμε για το σχεδιασμό των αντικείμενων που θα χρειαστούμε για το παιχνίδι, δηλαδή τα όπλα και την πανοπλία των χαρακτήρων και τα κτίρια. Επίσης, θα το χρησιμοποιήσουμε για το σχεδιασμό των animations των χαρακτήρων.

Το Unreal Engine 4 είναι η πιο πρόσφατη παιχνιδομηχανή που έχει σχεδιαστεί από την Epic Games και χρησιμοποιείται για την κατασκευή και λειτουργία βιντεοπαιχνιδιών. Η έκδοση που χρησιμοποιήθηκε είναι η Unreal Engine 4.16.3. Θα χρησιμοποιηθεί για το σχεδιασμό του επιπέδου του παιχνιδιού, για το στήσιμο των χαρακτήρων και για τον προγραμματισμό του παιχνιδιού. Το Unreal Engine 4 επιτρέπει τη χρήση C++ και Blueprints για προγραμματισμό. Θα χρησιμοποιήσουμε τα Blueprints που είναι ένα σύστημα οπτικής συγγραφής κώδικα με τη χρήση κόμβων για τη δημιουργία διαγραμμάτων.

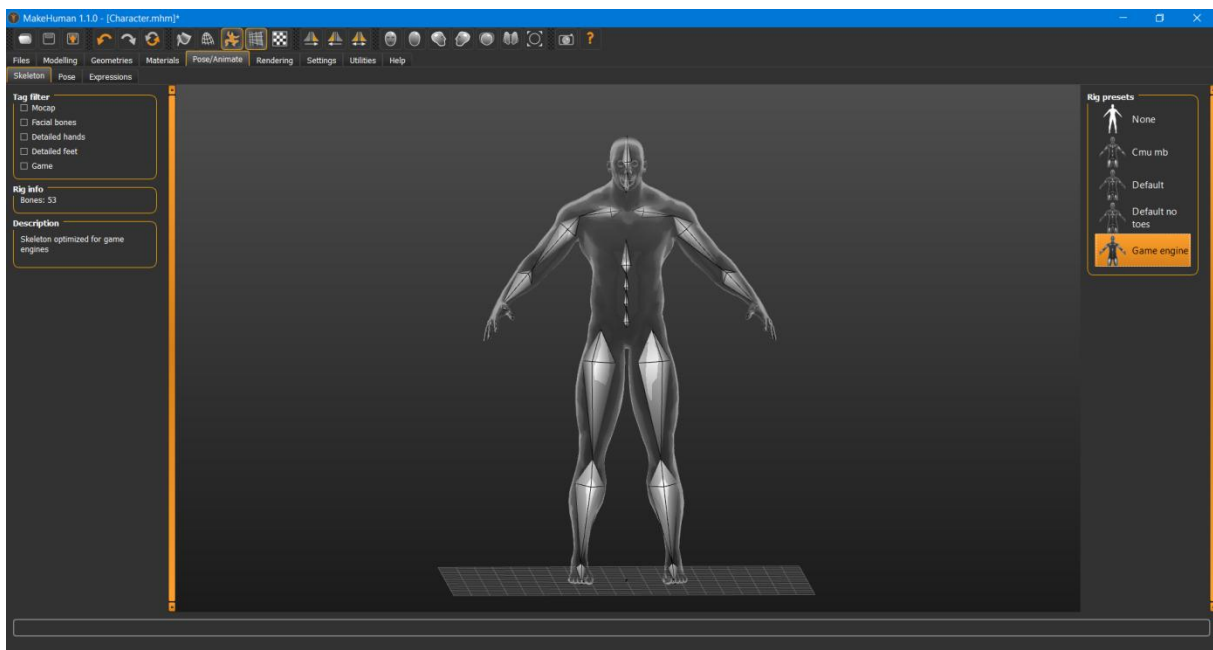
3. Υλοποίηση Πτυχιακής Εργασίας

3.1. MakeHuman



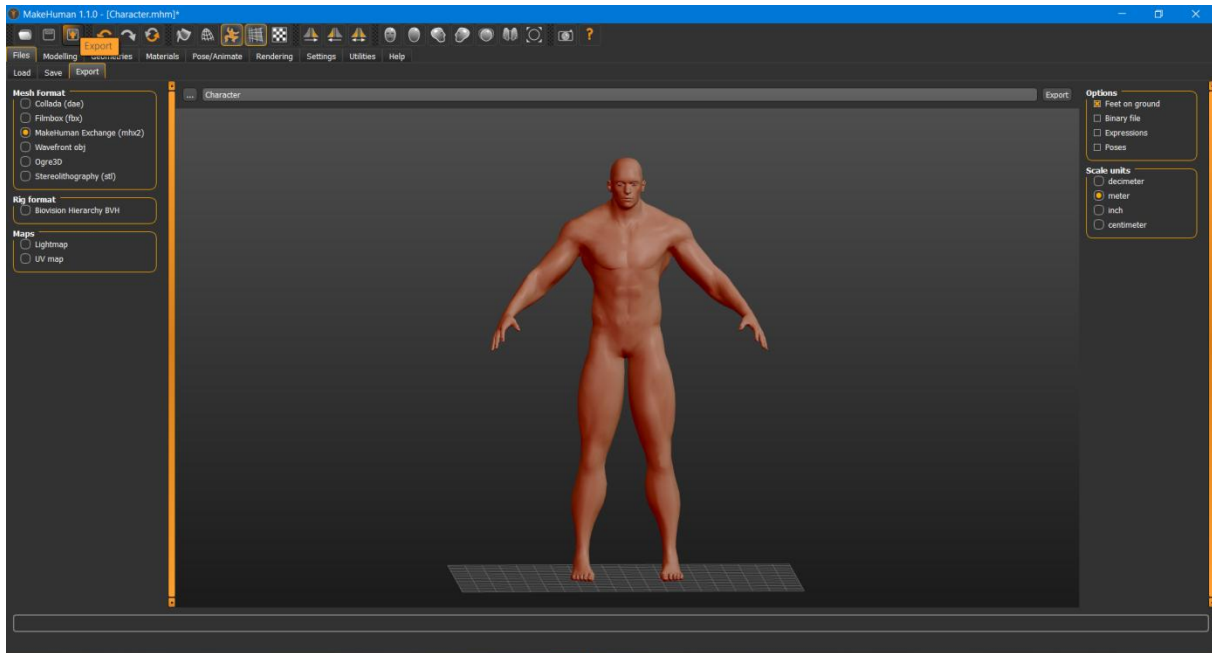
Εικόνα 1: Ρυθμίσεις Χαρακτήρα στο MakeHuman

Η Εικόνα 1 δείχνει τις ρυθμίσεις που θα χρησιμοποιηθούν για το μοντέλο που θα χρησιμοποιηθεί στο παιχνίδι. Τα sliders αντιπροσωπεύουν τις τιμές των μεταβλητών που επηρεάζουν την μορφή του μοντέλου. Για παράδειγμα, το πρώτο slider επηρεάζει το φύλο του χαρακτήρα, βάζοντάς το δεξιά το σώμα του χαρακτήρα έχει μόνο ανδρικά χαρακτηριστικά, ενώ στα αριστερά θα είχε μόνο θηλυκά, οι ενδιάμεσες τιμές δημιουργούν ένα σώμα με διαφορετικές αναλογίες αρσενικών και θηλυκών χαρακτηριστικών.



Εικόνα 2: Σκελετός Χαρακτήρα στο MakeHuman

Η Εικόνα 2 δείχνει την προσθήκη σκελετού στο μοντέλο, για να χρησιμοποιηθεί για τα animations σε επόμενο βήμα στο Blender. Στο Pose/Animate Tab στα Rig presets επιλέγουμε Game engine, αυτή η επιλογή δίνει ένα βασικό σκελετό που μπορεί να χρησιμοποιηθεί στα animations στο Blender.



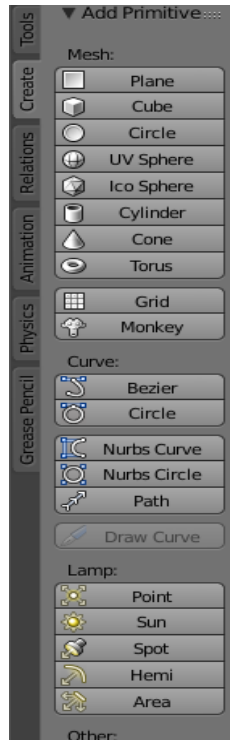
Εικόνα 3: Εξαγωγή Χαρακτήρα από το MakeHuman

Η Εικόνα 3 δείχνει την εξαγωγή του χαρακτήρα από το MakeHuman. Πατώντας το κουμπί Export (τρίτο κουμπί από τα αριστερά, στην πάνω αριστερή γωνία, στην πρώτη γραμμή) ανοίγει το Export Tab και επιλέγουμε: στο Mesh Format την επιλογή MakeHuman Exchange (.mhx2), στο Options την επιλογή Feet on ground και στο Scale units την επιλογή meter. Έπειτα γράφουμε το όνομα του αρχείου (Character) και πατάμε Export (δίπλα από το filename, είναι διαφορετικό κουμπί από το Export στην αρχή). Τέλος, στο Blender πρέπει να εισάγουμε το μοντέλο που εξάγαμε από το MakeHuman, για να γίνει αυτό πρέπει να κάνουμε τις επιλογές:

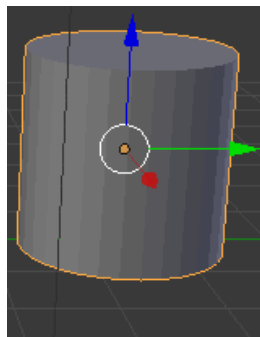
File → Import → MakeHuman (.mhx2) → επιλέγουμε το αρχείο (Character.mhx2) → Import MHX2.

3.2. Blender

3.2.1. Δημιουργία Αντικειμένων

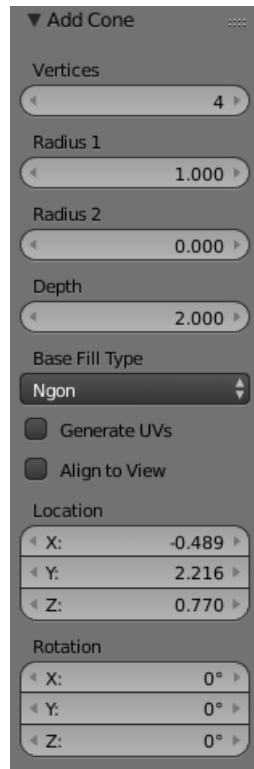


Εικόνα 4: Create Menu στο Blender

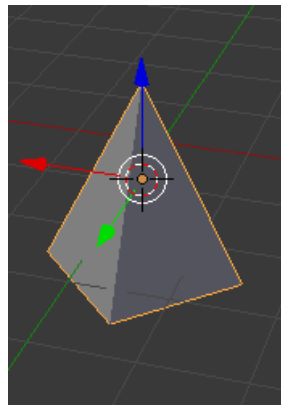


Εικόνα 5: Τρισδιάστατο Σχήμα του Blender

Με τη χρήση του create menu (Εικόνα 4) έχουμε τη δυνατότητα να χρησιμοποιήσουμε ένα από τα ήδη έτοιμα τρισδιάστατα σχήματα που υπάρχουν στο blender και να τα τροποποιήσουμε όπως εμείς επιθυμούμε. Στην Εικόνα 5 βλέπουμε έναν απλό κύλινδρο, που δημιουργήσαμε, παρακάτω θα δούμε πως μπορούμε να τροποποιήσουμε τα έτοιμα σχήματα.

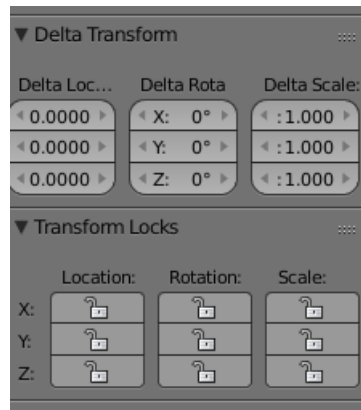


Εικόνα 6: Στοιχεία Σχήματος στο Blender



Εικόνα 7: Τροποποιημένο Σχήμα στο Blender

Πριν τοποθετήσουμε το αντικείμενο, έχουμε την επιλογή να τροποποιήσουμε κάποιες προεπιλεγμένες τιμές των παραμέτρων του (Εικόνα 6), όπως τον αριθμό των κορυφών του και την ακτίνα του, δίνοντας τις επιθυμητές τιμές για το σχήμα μας. Στην Εικόνα 7 βλέπουμε έναν κώνο, αποτελούμενο από 4 κορυφές, έχοντας ως αποτέλεσμα το τελικό σχήμα να είναι μία πυραμίδα.



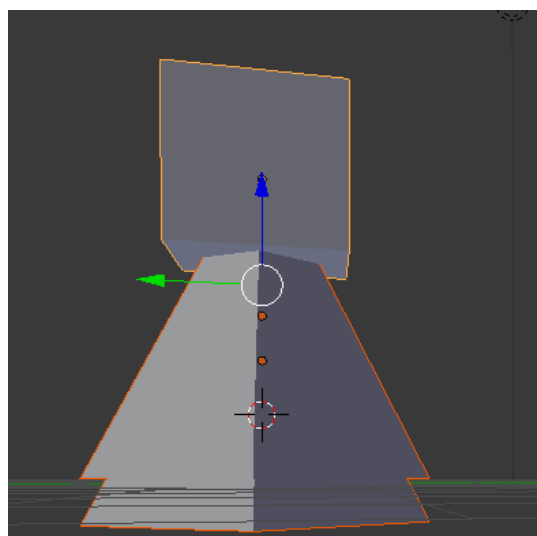
Εικόνα 8: Ρυθμίσεις Σχήματος στο Blender

Στην Εικόνα 8 φαίνεται η καρτέλα που μας επιτρέπει να επηρεάσουμε τη θέση, την περιστροφή και το μέγεθος του αντικειμένου ως προς τους X, Y, Z άξονες.



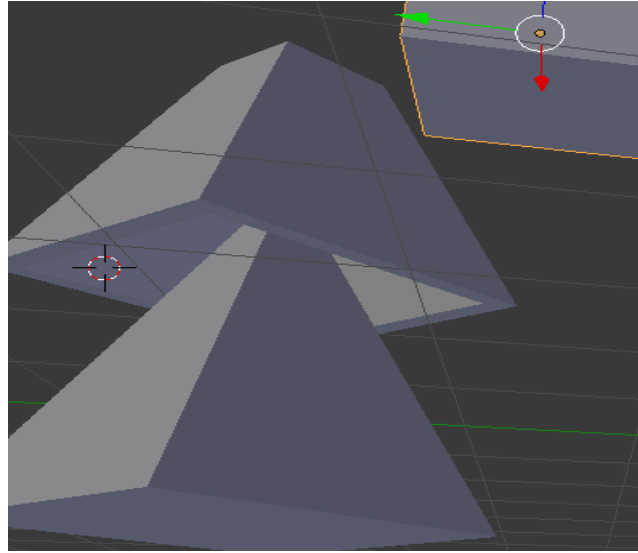
Εικόνα 9: Modifier στο Blender

Με τη χρήση του modifier (Εικόνα 9), έχουμε τη δυνατότητα να μεταποιήσουμε το σχήμα, αφαιρώντας του κομμάτια με τη χρήση του difference ή προσθέτοντάς του με το intersect από το Boolean modifier, δίνοντάς του έτσι την ακριβή μορφή που θέλουμε να έχει.

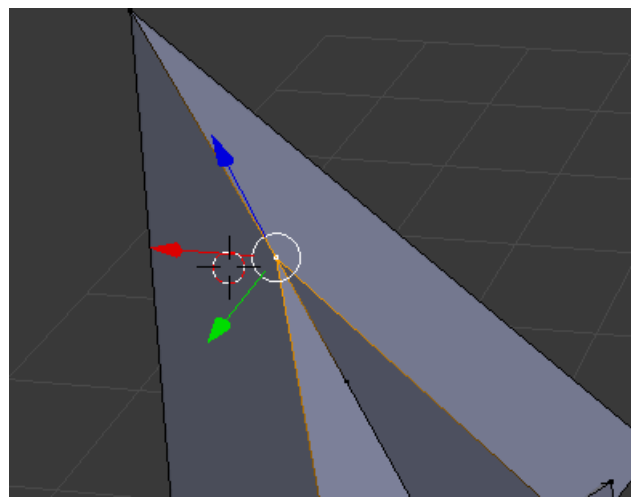


Εικόνα 10: Τροποποίηση Σχήματος με Difference Boolean Modifier στο Blender

Με τη χρήση ενός κύβου και ενός αντιγράφου της πυραμίδας (Εικόνα 10), που φτιάξαμε με τις εντολές Ctrl+C και Ctrl+V, αφαιρέσαμε την κορυφή της πυραμίδας και ανοίξαμε κούφωμα μέσα στην πυραμίδα με τη χρήση του difference από το Boolean modifier (Εικόνα 11).

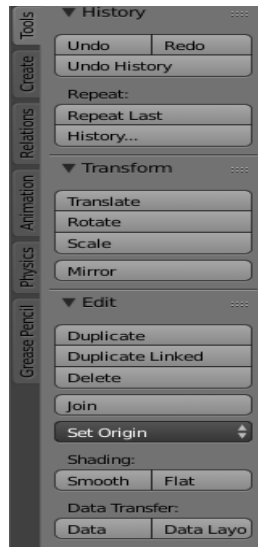


Εικόνα 11: Αποτέλεσμα Τροποποίησης Σχήματος με Difference Boolean Modifier στο Blender



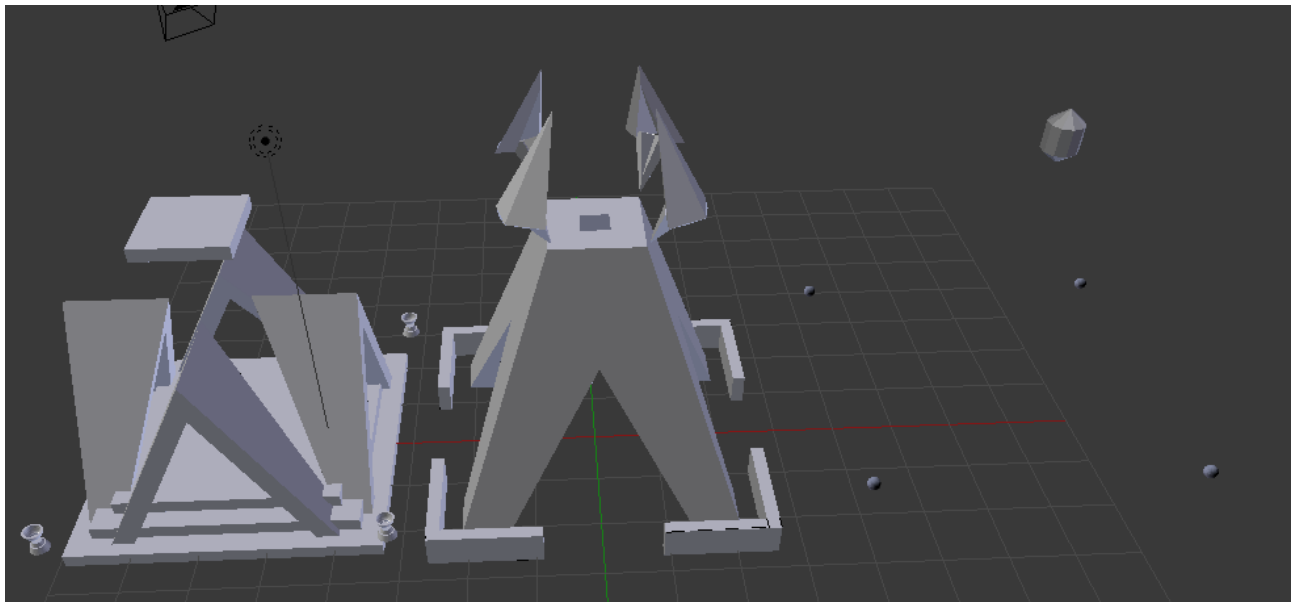
Εικόνα 12: Edit Mode στο Blender

Μια άλλη μέθοδος για να αλλάξουμε τη μορφή ενός αντικειμένου είναι με τη χρήση του Edit Mode. Το οποίο μας επιτρέπει να επηρεάσουμε μία-μία τις άκρες ενός αντικειμένου ή με το πάτημα του Shift+ επιλέγουμε πολλαπλές άκρες (Εικόνα 12).



Εικόνα 13: Tools Menu στο Blender

Με τη χρήση των παραπάνω μεθόδων φτιάχνουμε τα παραπάνω σχήματα και τα ομαδοποιούμε, χρησιμοποιώντας την επιλογή Join από τα Tools (Εικόνα 13), τα ενώνουμε και τα ομαδοποιούμε (Εικόνα 14).

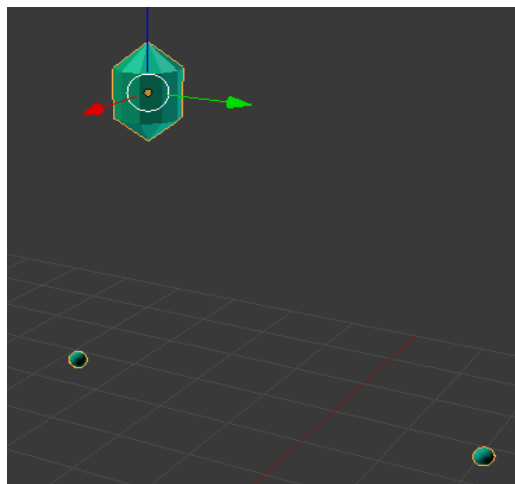


Εικόνα 14: Σχήματα Ενωμένα και Ομαδοποιημένα στο Blender



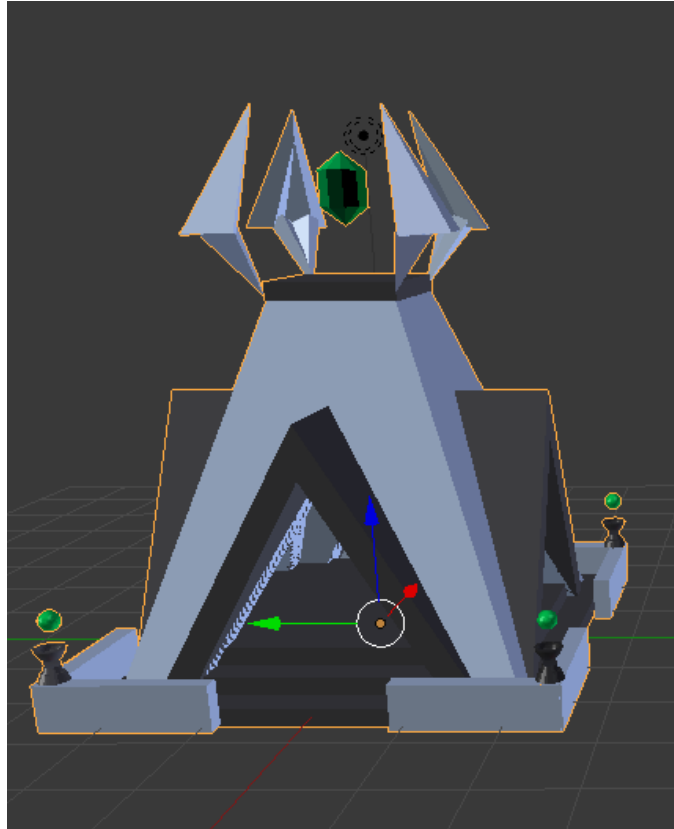
Εικόνα 15: Χρωματισμός Material των Αντικειμένων στο Blender

Τα αντικείμενα που επιλέξαμε για ομαδοποίηση είναι αυτά που θα έχουν το ίδιο χρώμα, για να μην τα χρωματίζουμε ξεχωριστά το καθένα. Έτσι αφού τα ομαδοποιήσουμε, τα χρωματίζουμε σαν ομάδα (Εικόνα 15), αλλάζοντας το χρώμα τους στη καρτέλα Material (Εικόνα 16).



Εικόνα 16: Χρωματισμένα Αντικείμενα στο Blender

Τέλος, για να δημιουργήσουμε ένα ολοκληρωμένο αντικείμενο, πρέπει να ομαδοποιήσουμε όλα τα αντικείμενα που το αποτελούν. Έτσι τα επιλέγουμε όλα μαζί σε μια ομάδα και με τη χρήση της επιλογής Join από τα Tools, δημιουργούμε το τελικό αντικείμενό μας (Εικόνα 17). Ένα αντικείμενο μπορεί να αποτελείται από πολλά απλά αντικείμενα, πολλά σύνθετα αντικείμενα, που δημιουργήθηκαν το καθένα με ομαδοποίηση, ή από πολλά απλά και σύνθετα αντικείμενα.



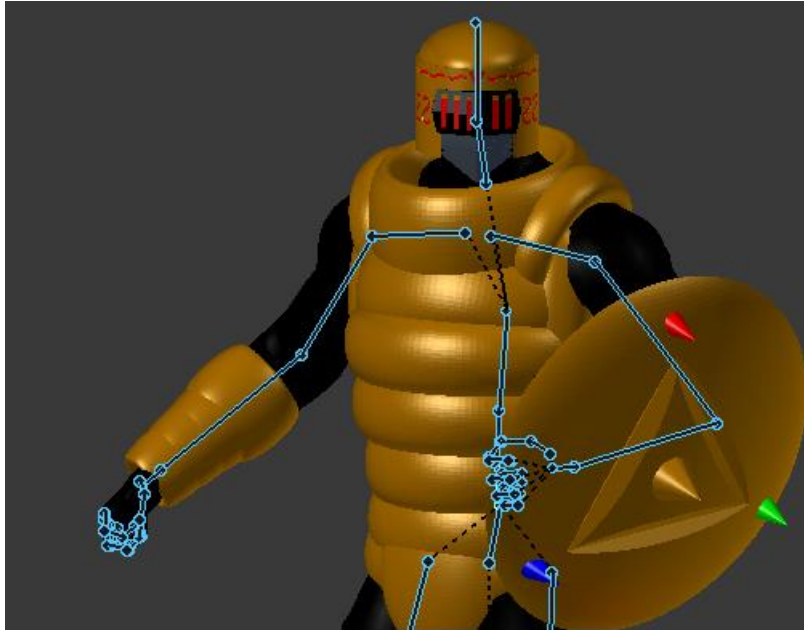
Εικόνα 17: Τελικό Αντικείμενο στο Blender

3.2.2. Animations

Για να φτιάξουμε ένα animation κάνουμε import το χαρακτήρα και τα κομμάτια εξοπλισμού που χρειάζεται να έχει πάνω του. Τα αντικείμενα, για λόγους ευκολίας φτιάχτηκαν σε ξεχωριστά αρχεία, για αυτό το λόγο πρέπει να τα κάνουμε import από τα αντίστοιχα αρχεία στο αρχείο του χαρακτήρα μας, όπου και θα κάνουμε τα animations.

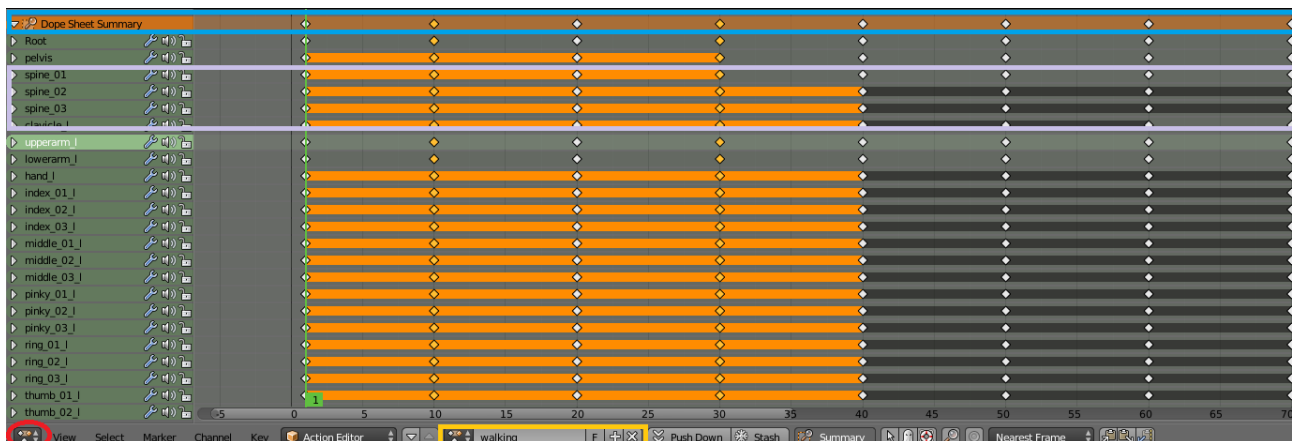


Εικόνα 18: Επιλογή Bone στο Blender



Εικόνα 19: Επιλογή Σκελετού στο Blender

Με δεξί κλικ μπορούμε να επιλέξουμε ένα bone (Εικόνα 18), ενώ με Shift + Δεξί Κλικ μπορούμε να επιλέξουμε πολλαπλά bones. Επίσης, επιλέγοντας με Δεξί Κλικ ένα bone και μετά πατώντας 2 φορές το A μπορούμε να επιλέξουμε όλο το σκελετό (Εικόνα 19).

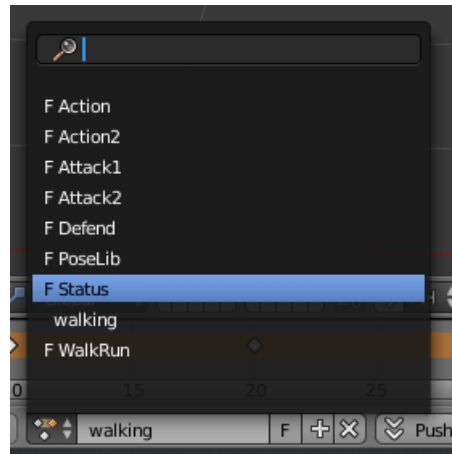


Εικόνα 20: Dope Sheet στο Blender

Στην Εικόνα 20 παρατηρούμε κάτω αριστερά, που είναι μαρκαρισμένο με έναν κόκκινο κύκλο, ότι έχουμε επιλέξει τη Dope Sheet παρουσίαση του Timeline, έτσι ώστε να φτιάξουμε το animation.

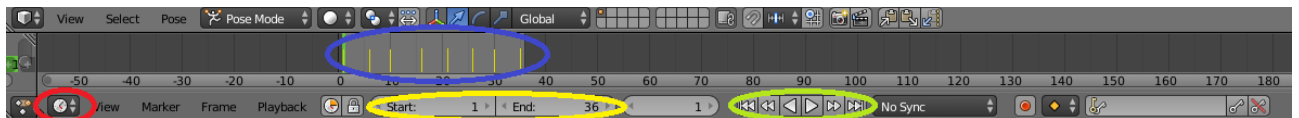
Στην περιοχή του Timeline, που είναι μαρκαρισμένη με μπλε και ονομάζεται Dope Sheet Summary, παρατηρούμε άσπρους κύκλους, οι οποίοι μας δείχνουν τα διαφορετικά frames που τοποθετούμε στο Timeline κάθε φορά που θέλουμε να κάνουμε αλλαγή στη στάση του σώματος του χαρακτήρα που φτιάχνουμε. Πατώντας πάνω τους με Δεξί Κλικ μπορούμε να τα επιλέξουμε ένα τη φορά, ή με Shift + Δεξί Κλικ μπορούμε να επιλέξουμε πολλαπλά και να τα σύρουμε στο Timeline. Με το Ctrl+C μπορούμε να τα αντιγράψουμε και με το Ctrl+V να τα τυπώσουμε σε άλλη θέση στο Timeline.

Όταν επεκτείνουμε το Dope Sheet Summary μπορούμε να δούμε τα διάφορα μέρη του σκελετού του χαρακτήρα και σε ποια frames του animation έχουν γίνει αλλαγές πάνω τους. Οι πορτοκαλί γραμμές δείχνουν τα frames πριν και μετά από αυτά που έχουμε επιλέξει. Ενώ ο πράσινος κέρσορας μας δείχνει ποιο είναι το παρόν frame στο οποίο θα ξεκινήσει το animation.



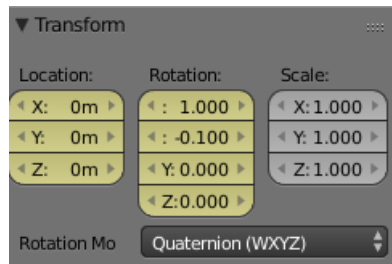
Εικόνα 21: Επιλογή Animation στο Blender

Από την περιοχή, που είναι μαρκαρισμένη με κίτρινο (Εικόνα 20), μπορούμε να φορτώσουμε animations (Εικόνα 21). Επίσης, μπορούμε να τα αποθηκεύσουμε δίνοντας ένα όνομα και πατώντας το + κουμπί (Εικόνα 21).



Εικόνα 22: Timeline στο Blender

Στην Εικόνα 22, το σημείο που είναι μαρκαρισμένο με το κόκκινο κύκλο, είναι για να ρυθμίζουμε τον Editor να είναι στο Timeline, το οποίο μας επιτρέπει να ελέγξουμε, αν το animation που φτιάξαμε φαίνεται όπως θέλουμε, κατά την εκτέλεση του σε διάρκεια συγκεκριμένων frames που εμείς επιλέγουμε. Στο σημείο που είναι μαρκαρισμένο με το μπλε κύκλο, φαίνεται το Timeline του animation, ενώ οι κίτρινες γραμμές δείχνουν τα frames του animation, η πράσινη γραμμή δείχνει το παρόν frame που προβάλλεται και το γκριζωπό φόντο δείχνει τη διάρκεια σε frames του animation. Στο σημείο που είναι μαρκαρισμένο με το κίτρινο κύκλο, μπορούμε να επιλέξουμε ποιο θα είναι το αρχικό και ποιο θα είναι το τελικό frame που θέλουμε να εκτελεστεί. Ενώ στο σημείο που είναι μαρκαρισμένο με τον πράσινο κύκλο, είναι τα κουμπιά που μας επιτρέπουν να χειριστούμε το Timeline, οι δυνατότητες των κουμπιών είναι play, go to first frame και go to last frame.

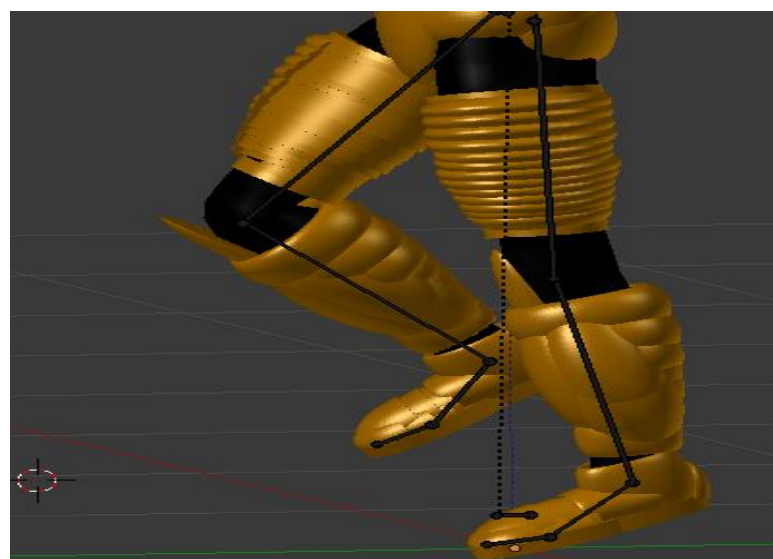


Εικόνα 23: Transform Ρυθμίσεις στο Blender

Με τη χρήση των επιλογών που μας δίνει το Transform έχουμε με τη δυνατότητα να μετατοπίσουμε, να περιστρέψουμε, ή να αλλάξουμε το μέγεθος ενός αντικειμένου ως προς τους άξονες X, Y, Z (Εικόνα 23).



Εικόνα 24 Frame στο Blender



Εικόνα 25 Frame στο Blender

Βλέπουμε από τις Εικόνες 24 και 25, δύο frames από το walking animation όπου χρησιμοποιήσαμε για να αντιστρέψουμε τις συντεταγμένες κάθε άκρου στο σκελετό των ποδιών με τη χρήση του Transform ώστε να πετύχουμε το αντίθετο βηματισμό.

3.3. Unreal Engine 4

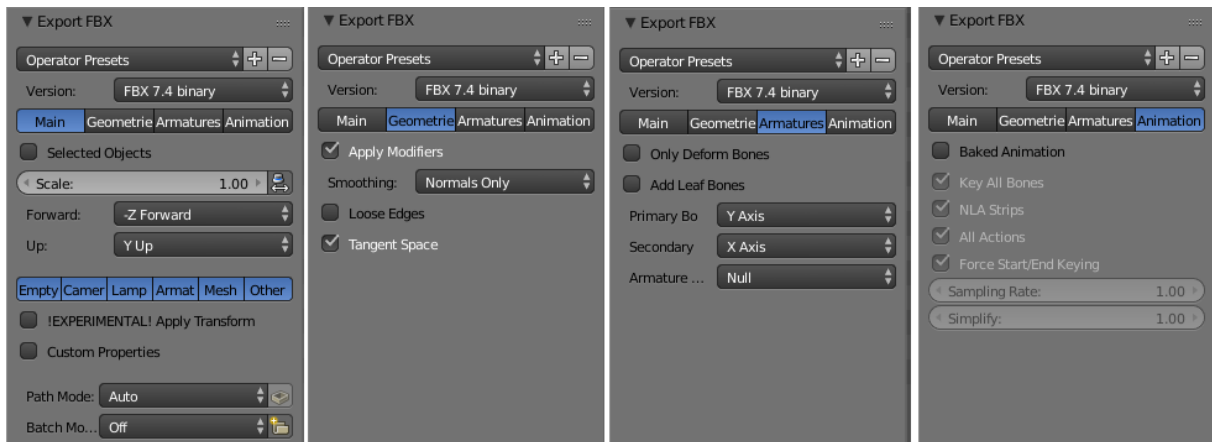
Για να συνεχίσουμε στην Unreal Engine 4 πρέπει πρώτα να αλλάξουμε κάποιες ρυθμίσεις στο Blender. Οι ρυθμίσεις είναι στην Εικόνα 26 παρακάτω.



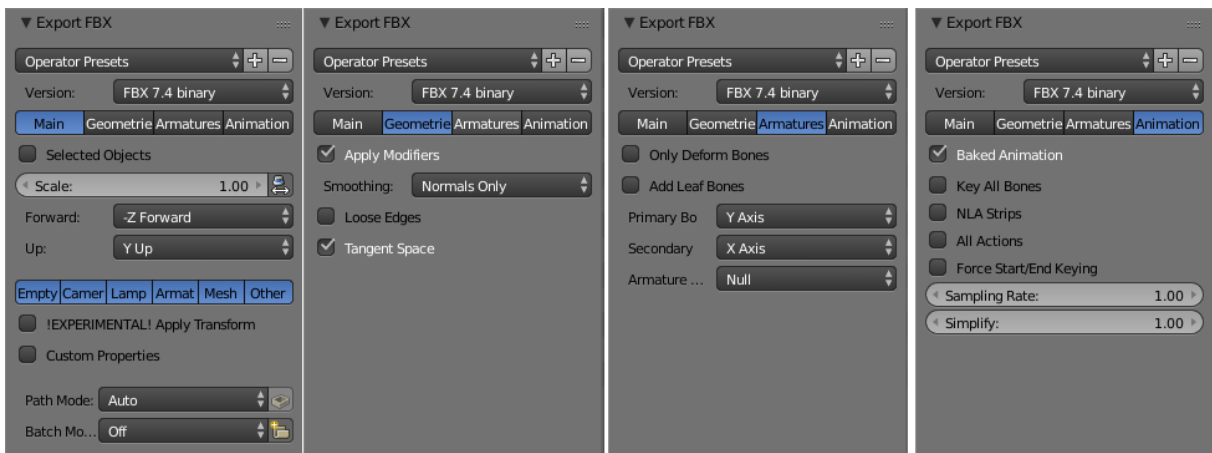
Εικόνα 26: Γενικές Ρυθμίσεις Εξαγωγής από το Blender

Τώρα πρέπει να εξάγουμε τα αντικείμενα (weapons, armors, buildings), τα animations και το χαρακτήρα με τον σκελετό ως fbx αρχεία. Όταν εξάγουμε αντικείμενα πρέπει να τα επιλέξουμε και να εξάγουμε ένα τη φορά. Για τα animations και το χαρακτήρα με το σκελετό πρέπει να επιλέξουμε όλα τα κόκκαλα, επιλέγοντας ένα από τα κόκκαλα και πατώντας A δύο

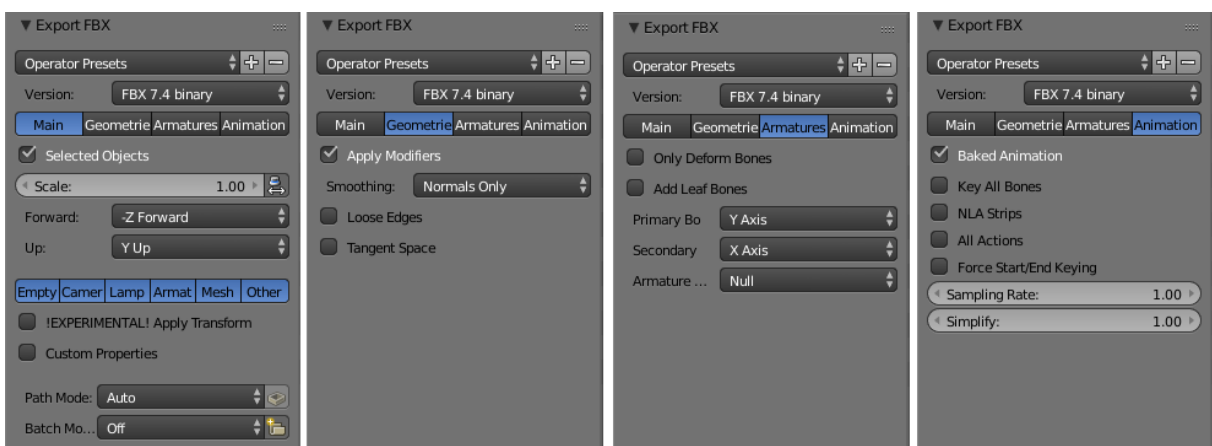
φορές τα επιλέγουμε όλα, και μετά τα εξαγωγή. Τα animations, πριν εξαχθούν, χρειάζεται να βάλουμε τα σωστά Start και End Frames στο Timeline, για να εξαχθούν ολόκληρα, αλλιώς κόβονται frames κατά την εξαγωγή τους. Οι ρυθμίσεις για την εξαγωγή είναι παρακάτω, στις Εικόνες 27, 28, 29 (για αντικείμενα, χαρακτήρα με σκελετό και animations, αντίστοιχα).



Εικόνα 27: Ρυθμίσεις Εξαγωγής Αντικειμένων από το Blender



Εικόνα 28: Ρυθμίσεις Εξαγωγής Χαρακτήρα με Σκελετό από το Blender

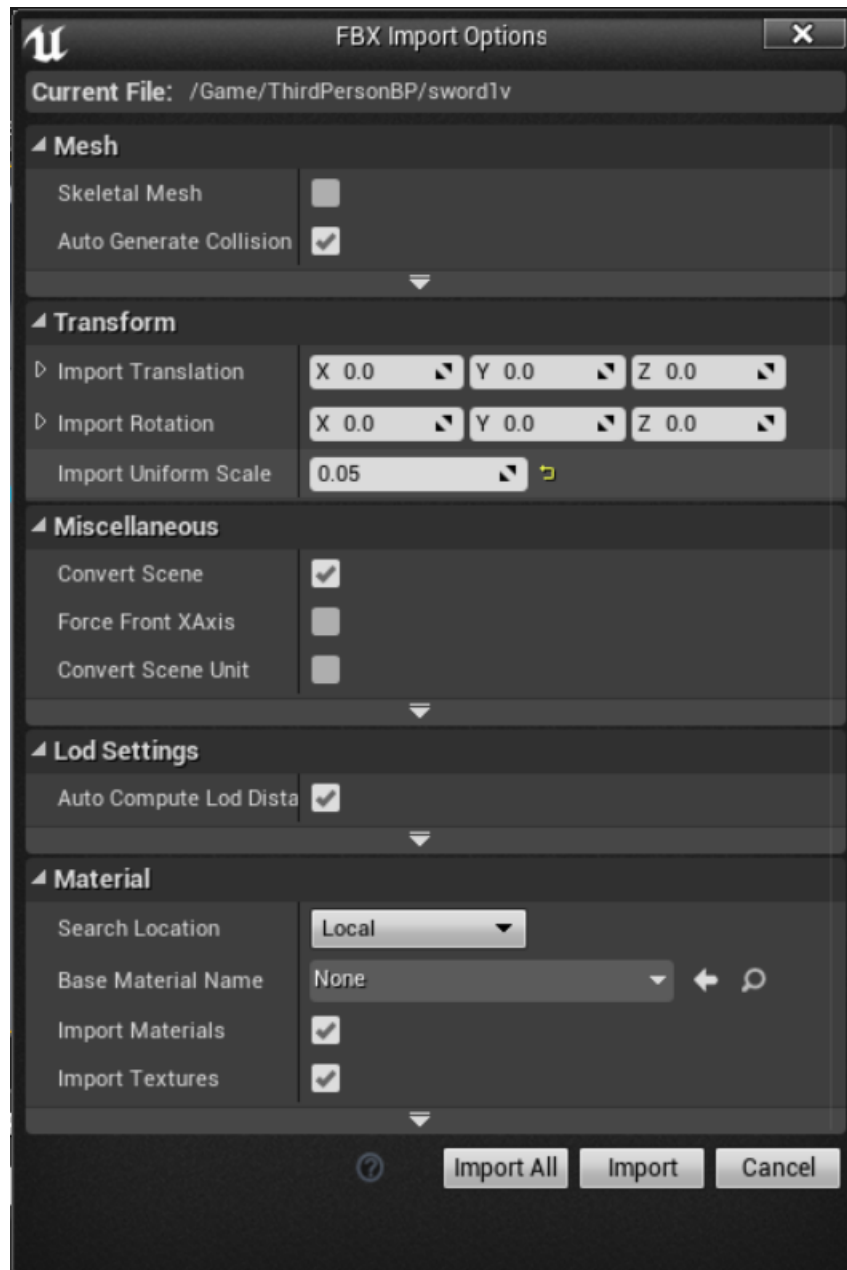


Εικόνα 29: Ρυθμίσεις Εξαγωγής Animations από το Blender

Ανοίγοντας το πρόγραμμα, επιλέγουμε New Project → Blueprint → Third Person, γράφουμε το όνομα του Project και πατάμε Create Project. Έπειτα εισάγουμε τα αντικείμενα, το χαρακτήρα με το σκελετό και τα animations, πατώντας το κουμπί Import και επιλέγουμε

ένα προς ένα τα αρχεία fbx που εξάγαμε από το Blender, αφού βάλουμε τις κατάλληλες ρυθμίσεις όπου χρειάζονται πατάμε Import All ή Import, για να εισάγουμε τα αρχεία και να ολοκληρωθεί η διαδικασία. Λόγω ότι τα animations χρειάζονται το σκελετό του χαρακτήρα μας, για την εισαγωγή τους στην Unreal Engine 4, πρέπει να τα εισάγουμε μετά από το χαρακτήρα μας και το σκελετό του.

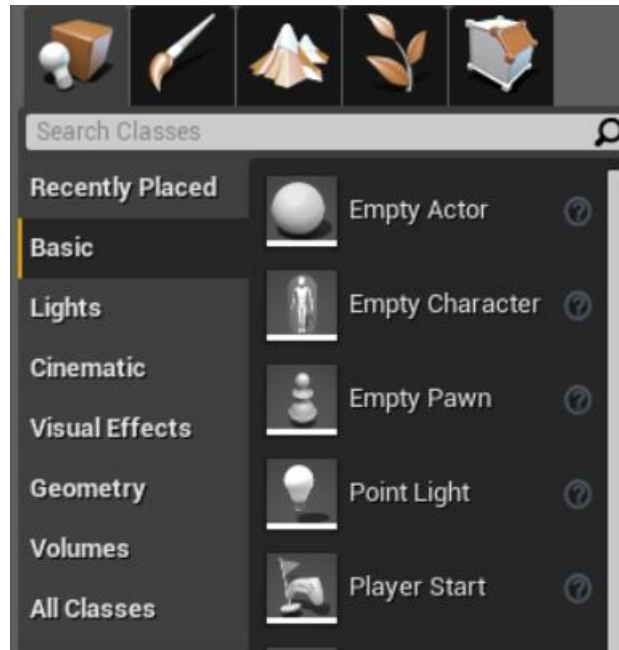
Όσον αφορά τις ρυθμίσεις, τα animations και ο χαρακτήρας με τον σκελετό χρησιμοποιούν τις αρχικές ρυθμίσεις της Unreal Engine 4. Ενώ οι ρυθμίσεις των αντικειμένων είναι στην Εικόνα 30, παρακάτω.



Εικόνα 30: Ρυθμίσεις Εισαγωγής Αντικειμένων στο Unreal Engine 4

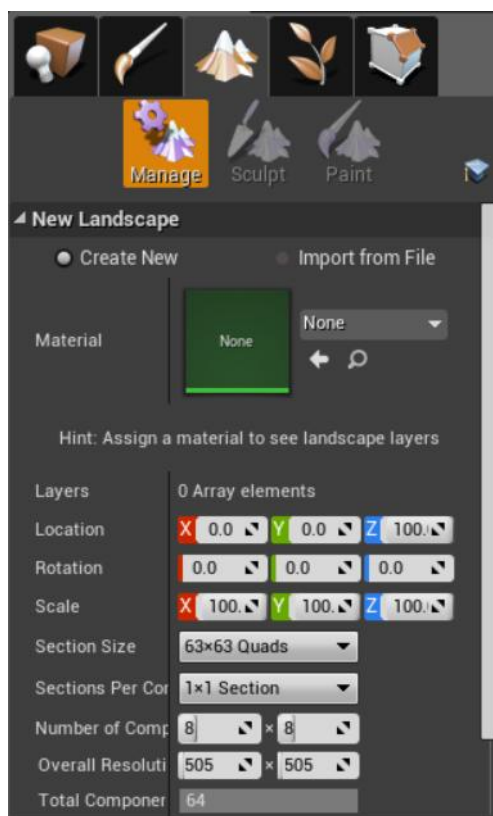
3.3.1. Level Design

3.3.1.1. Δημιουργία Χάρτη



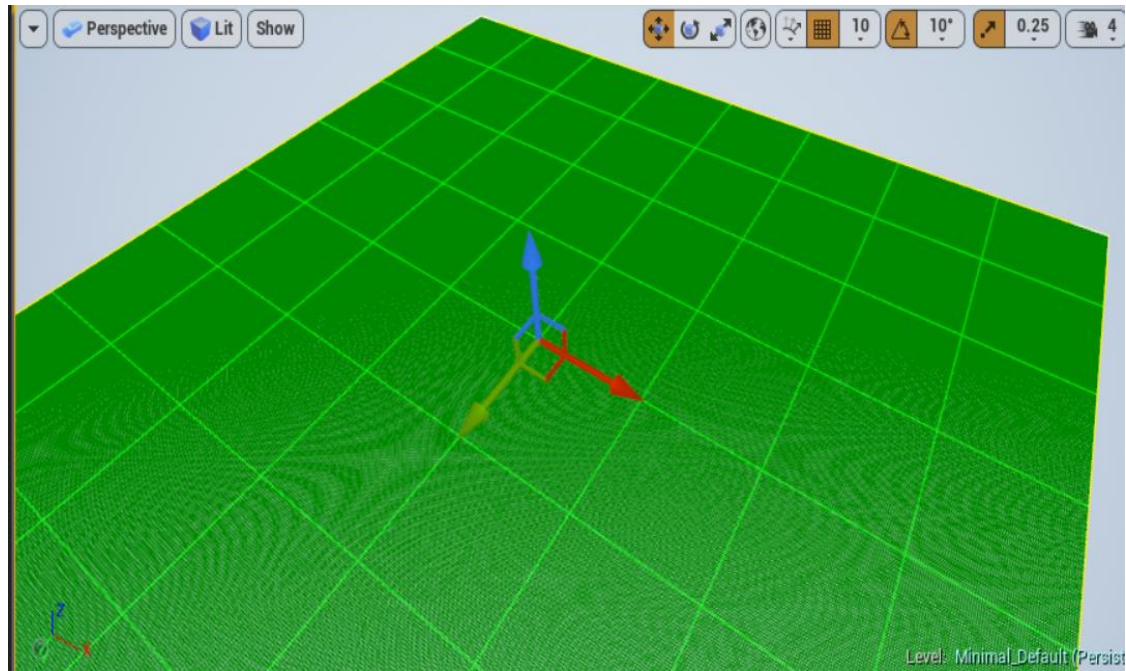
Εικόνα 31: Επιλογές Menu στην Unreal Engine 4

Το πρώτο βήμα που θα κάνουμε για τη δημιουργία ενός επιπέδου είναι με τη βοήθεια της Unreal Engine 4. Πάμε στις επιλογές της Εικόνας 31 και θα επιλέξουμε την επιλογή για το landscape (Εικόνα 32).

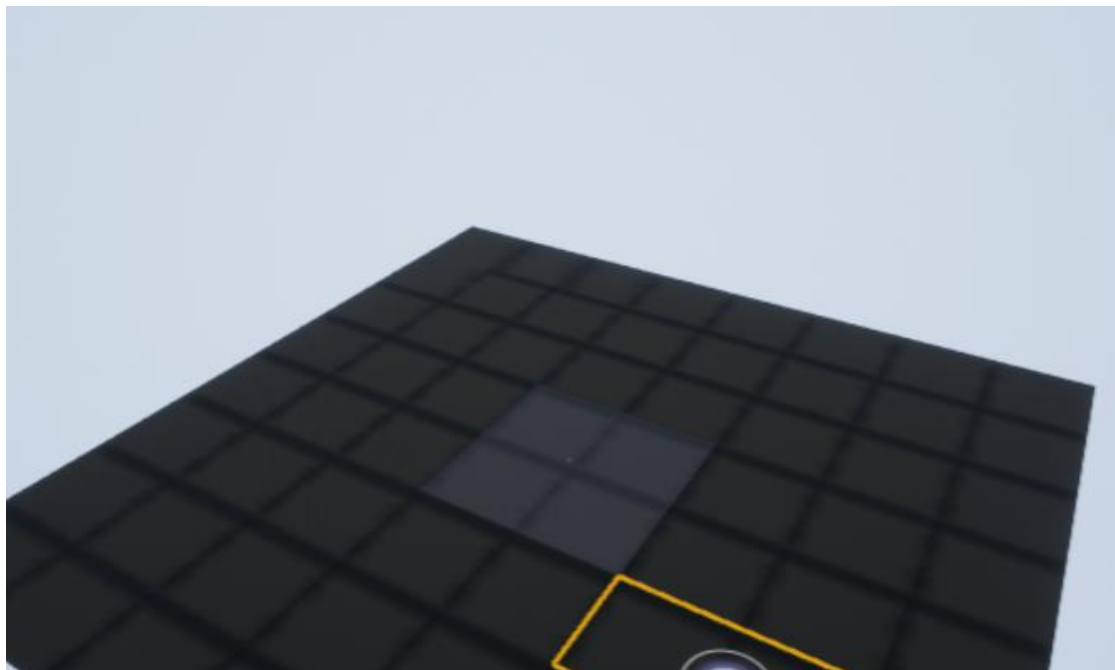


Εικόνα 32: Landscape Menu στην Unreal Engine 4

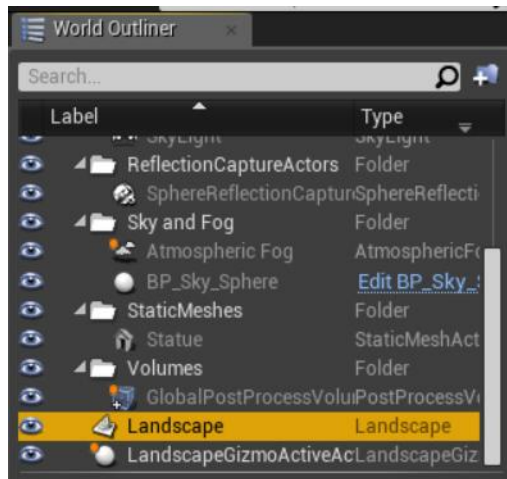
Από την Εικόνα 32 παρατηρούμε ότι κατά τη δημιουργία του εδάφους, μας δίνει επιλογές όπως το Material για την πίστα μας, τοποθεσία, περιστροφή και μέγεθος, τα οποία μπορούμε να αλλάξουμε με βάση τους άξονες X, Y, Z αλλά και επιλογές όπως η ανάλυσή του. Αφού δώσουμε τις κατάλληλες ρυθμίσεις, πατάμε Create, για να εμφανιστεί το αποτέλεσμα των επιλογών μας (Εικόνες 33, 34).



Εικόνα 33: Έδαφος Επιπέδου στην Unreal Engine 4



Εικόνα 34: Έδαφος Επιπέδου στην Unreal Engine 4



Εικόνα 35: World Outliner στην Unreal Engine 4

Έπειτα, το αντικείμενο εμφανίζεται στο World Outliner Menu (Εικόνα 35). Κάθε νέο αντικείμενο που δημιουργούμε θα εμφανίζεται μέσα στο World Outliner. Μπορούμε εύκολα να μετακινηθούμε στην πίστα μας, χρησιμοποιώντας τα ακόλουθα κουμπιά:

up : zoom-in camera

down : zoom-out camera

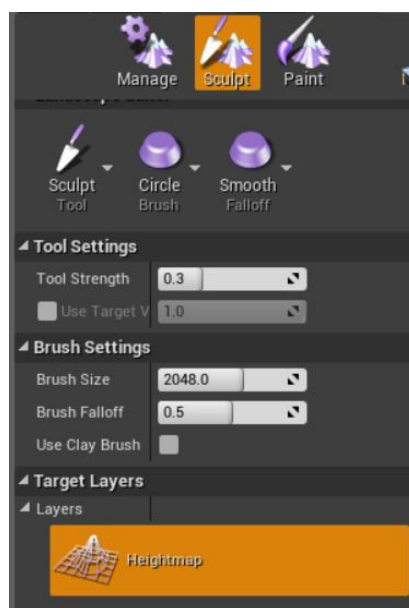
left : move camera left

right : move camera right

scroll mouse : zoom-in camera and zoom-out camera

press scroll of mouse and drag : move camera up and down

right click : rotate camera

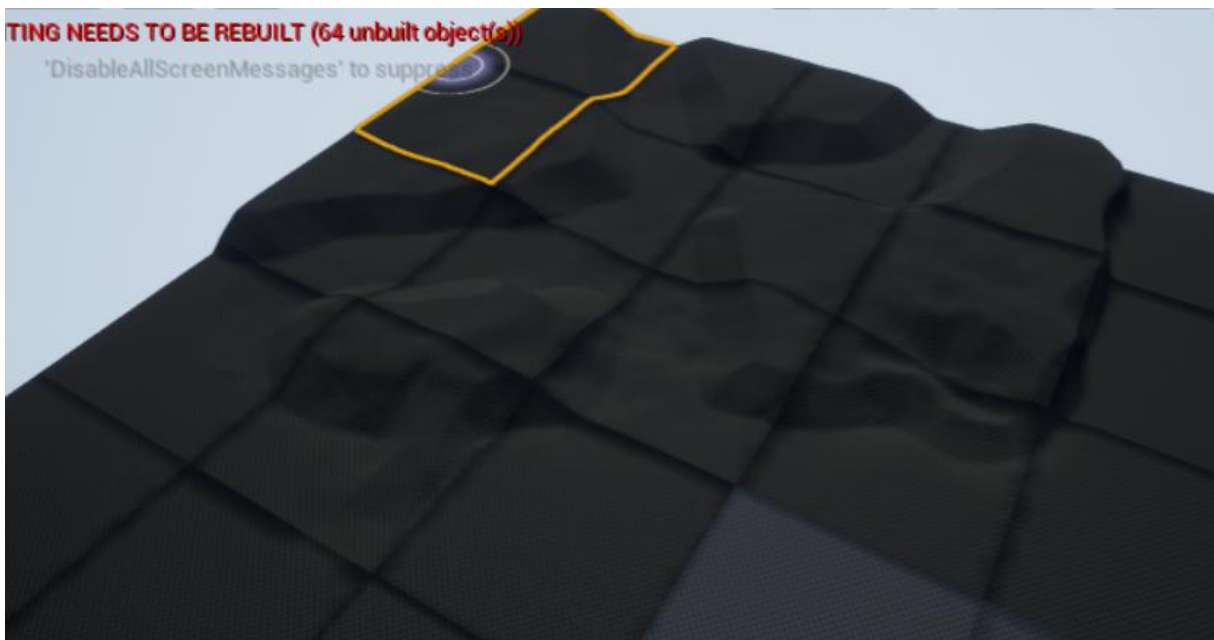


Εικόνα 36: Επιλογή Sculpt στην Unreal Engine 4



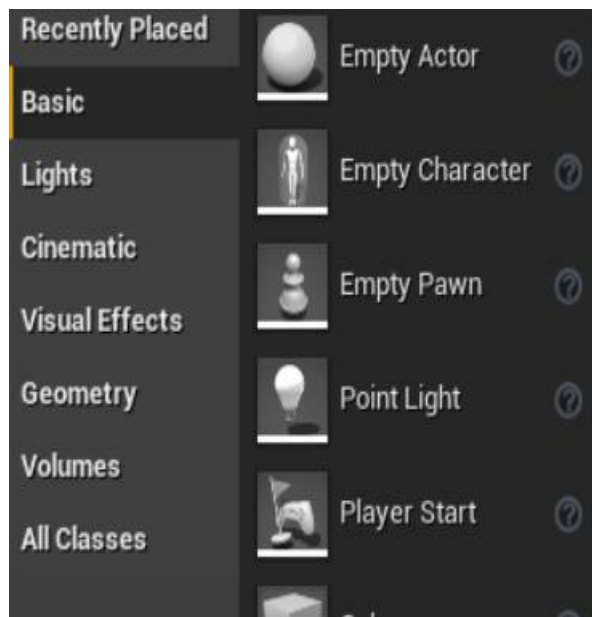
Εικόνα 37 Sculpt Menu στην Unreal Engine 4

Από την ετικέτα Landscape και χρησιμοποιώντας τις επιλογές όπως το sculpt (Εικόνες 36, 37) έχουμε τη δυνατότητα να δώσουμε το σχήμα που επιθυμούμε στο περιβάλλον μας (Εικόνα 38).

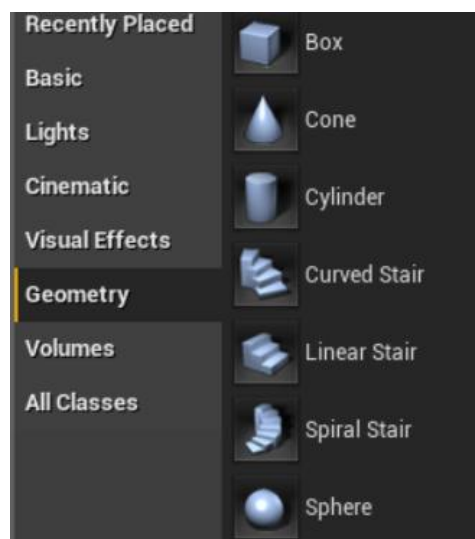


Εικόνα 38: Αλλαγή Σχήματος Εδάφους στην Unreal Engine 4

3.3.1.2. Τοποθέτηση και Επεξεργασία Αντικειμένων

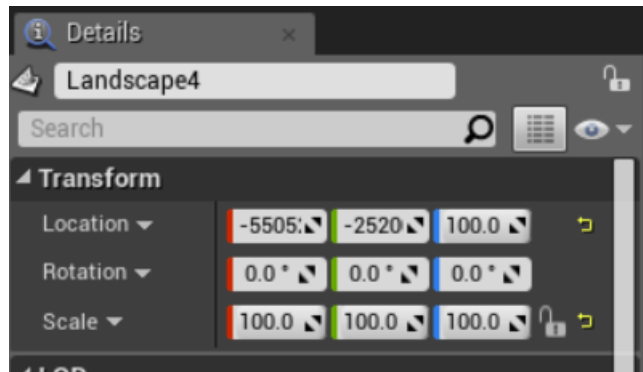


Εικόνα 39: Έτοιμα Αντικείμενα στην Unreal Engine 4

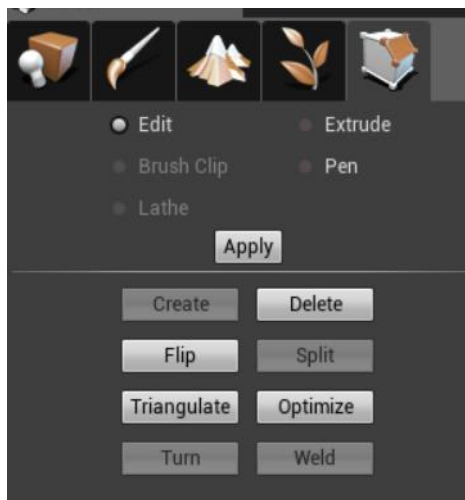


Εικόνα 40: Έτοιμα Αντικείμενα στην Unreal Engine 4

Από τις καρτέλες των Εικόνων 39 και 40 μπορούμε να επιλέξουμε έτοιμα αντικείμενα, όπως κύβους, σφαίρες, αλλά και πιο σύνθετα αντικείμενα, όπως σκάλες, για να τα τοποθετήσουμε στο επίπεδό μας και να τα τροποποιήσουμε όπως εμείς θέλουμε. Με τη χρήση της Details Tab μπορούμε να επηρεάσουμε τη τοποθεσία, περιστροφή και μέγεθος των αντικειμένων με βάση τους άξονες X, Y, Z (Εικόνα 41).

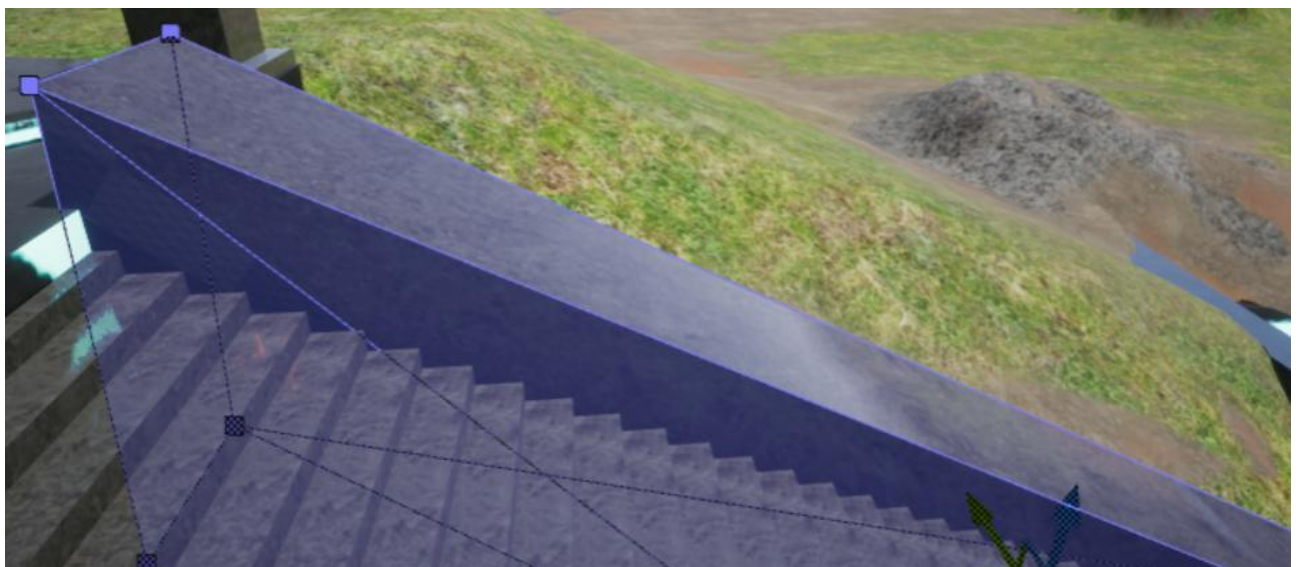


Εικόνα 41: Ρυθμίσεις του Details Tab στην Unreal Engine 4

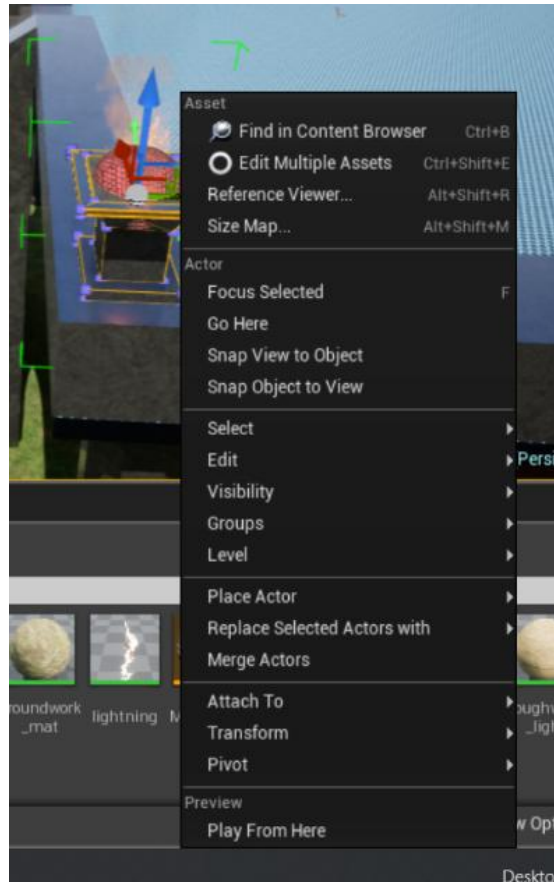


Εικόνα 42: Edit Mode στην Unreal Engine 4

Από το Edit Mode έχουμε τη δυνατότητα να παραμορφώσουμε το σχήμα ενός αντικειμένου (Εικόνα 42). Οι μπλε γραμμές στην Εικόνα 43 μας επιτρέπουν να παραμορφώσουμε το σχήμα όπως επιθυμούμε.



Εικόνα 43: Σχήμα σε Edit Mode στην Unreal Engine 4



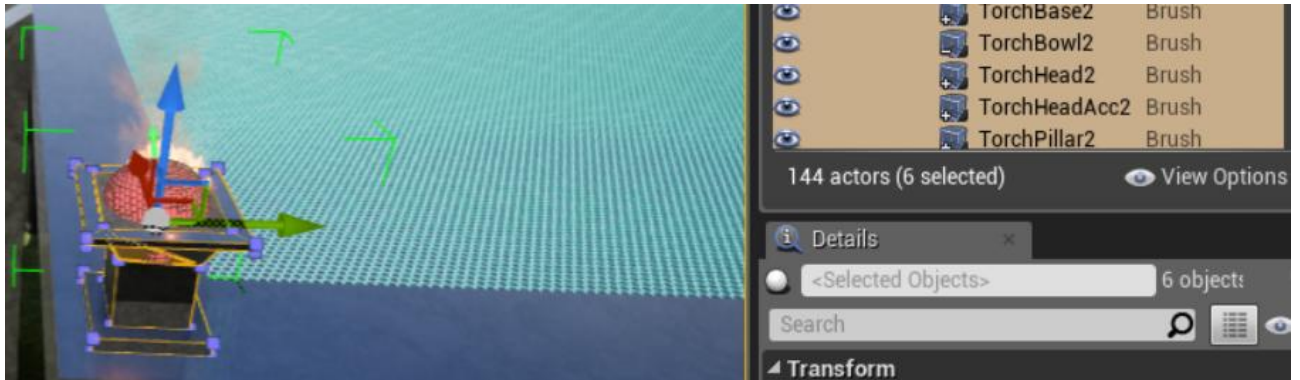
Εικόνα 44: Επιλογές Αντικειμένου στην Unreal Engine 4

Επιλέγοντας με Δεξί Κλικ ένα αντικείμενο μέσα στην πίστα μας, μπορούμε να το επεξεργαστούμε (Εικόνα 44). Για παράδειγμα, να επικεντρωθούμε πάνω του με το Go Here, να το επιλέξουμε με το Select, να το αντιγράψουμε ή να διαγράψουμε με το Edit, να το αφήσουμε ορατό με το Visibility.



Εικόνα 45: Πυρός στην Unreal Engine 4

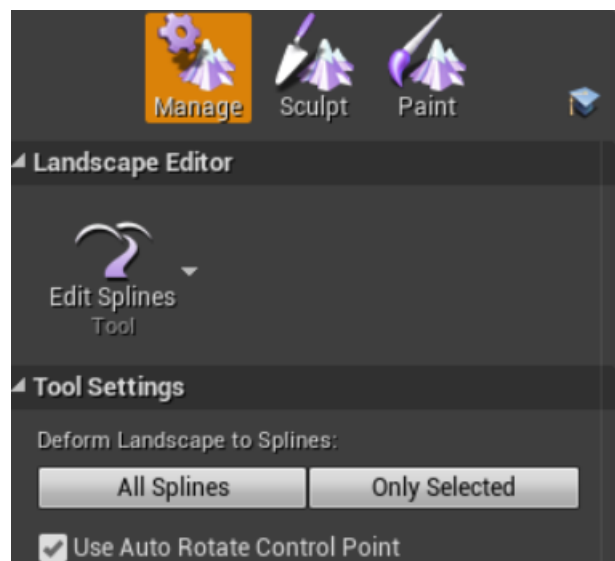
Στο παράδειγμα της Εικόνας 45, ο πυρσός δημιουργήθηκε από συνδυασμό 4 κύβων, επεξεργασμένων με το Edit Mode, μίας σφαίρας, που της αφαιρέσαμε το Visibility, και την προσθήκη εφέ φωτιάς (για χρώματα και εφέ θα αναφερθούμε παρακάτω).



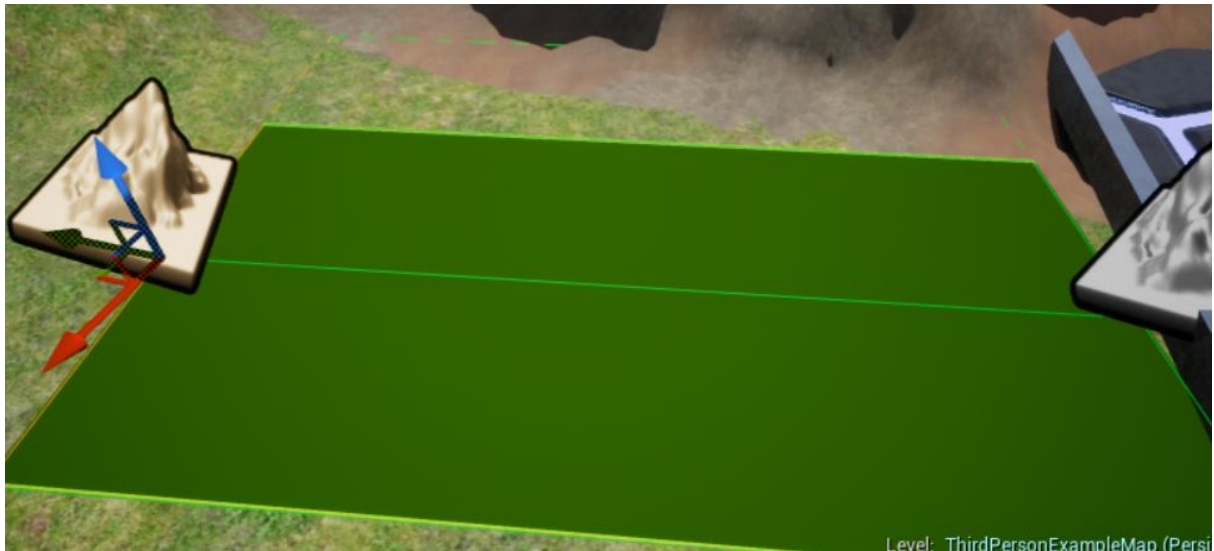
Εικόνα 46: Πυρσός στην Unreal Engine 4

Αφού επιλέξουμε όλα τα τμήματα του πυρσού τα ενώνουμε μαζί με την επιλογή Attach To (Εικόνα 46). Για να επιλέξουμε πολλαπλά αντικείμενα πατάμε Ctrl + Αριστερό Κλικ στα αντικείμενα που θέλουμε ή Shift + Αριστερό Κλικ πάνω στα αντικείμενα που επιθυμούμε στο World Outliner.

Τα Splines είναι ένα χρήσιμο εργαλείο για τη δημιουργία μονοπατιών, δρόμων ή και ποταμιών. Επιλέγουμε το Splines από το Manage πεδίο (Εικόνα 47).

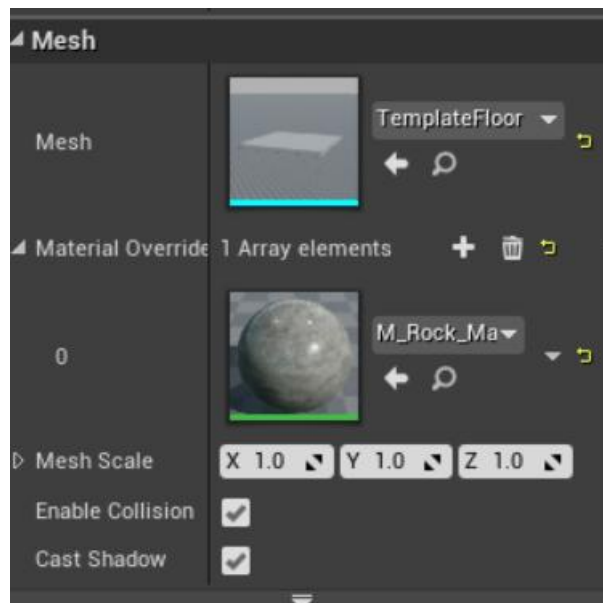


Εικόνα 47: Επιλογή Manage στην Unreal Engine 4

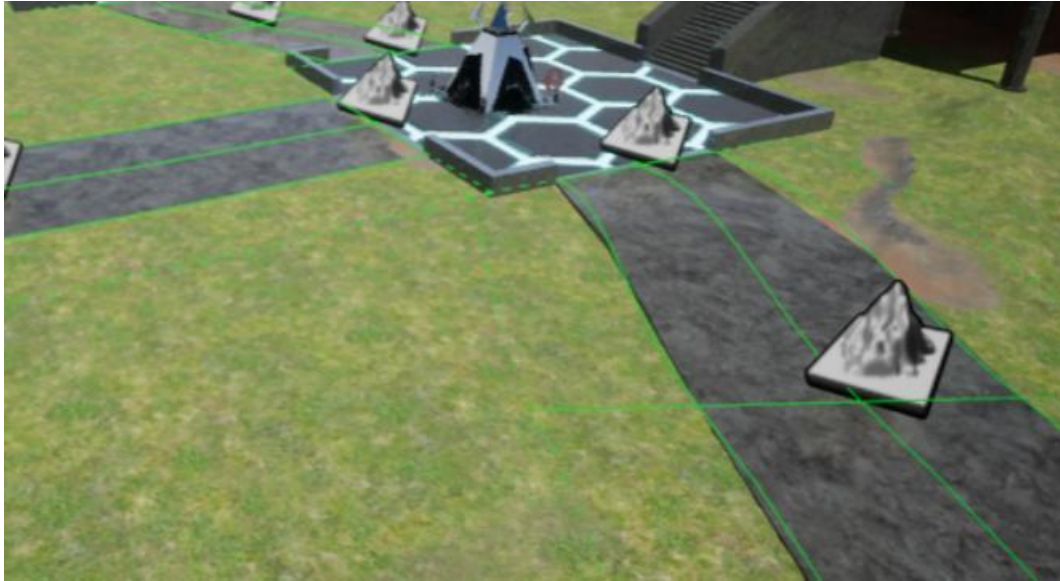


Εικόνα 48: Μονοπάτι από Splines στην Unreal Engine 4

Στην Εικόνα 48 βλέπουμε τη δημιουργία ενός μονοπατιού από Splines χρησιμοποιώντας Ctrl + Αριστερό Κλικ. Έπειτα, του δίνουμε Mesh και Material, ενώ ρυθμίζουμε το Scale του Mesh (Εικόνα 49). Στην Εικόνα 50 βλέπουμε ένα παράδειγμα των Splines μονοπατιών που έχουμε φτιάξει.



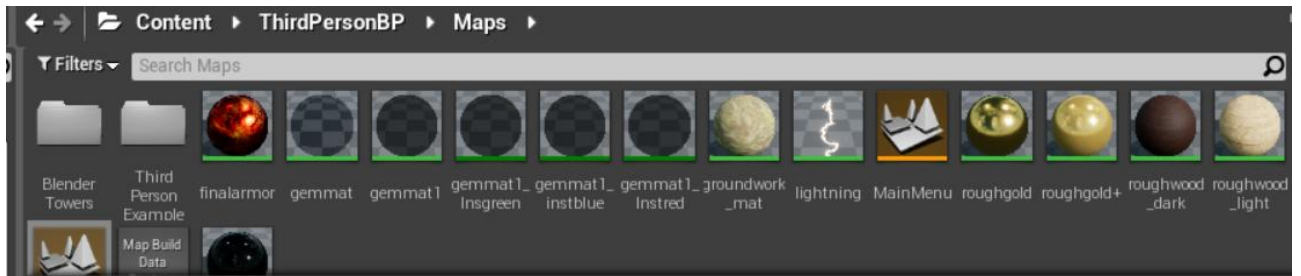
Εικόνα 49: Ρυθμίσεις Mesh στην Unreal Engine 4



Εικόνα 50: Splines Μονοπάτια στην Unreal Engine 4

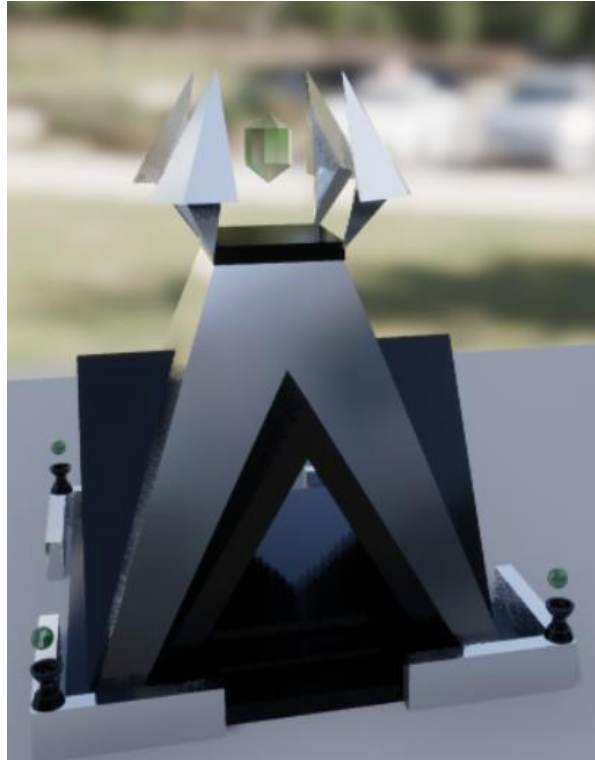
3.3.1.3. Υλικά και Εφέ

Στις πιο πάνω διαφάνειες αναφερθήκαμε στο πως δημιουργούμε ένα landscape και για το πως μπορούμε να δώσουμε το σχήμα που επιθυμούμε στα αντικείμενά μας. Σε αυτήν την ενότητα θα δούμε πως τα υλικά και τα εφέ μπορούν να χρησιμοποιηθούν για να κάνουν το παιχνίδι πιο ευπαρουσίαστο.

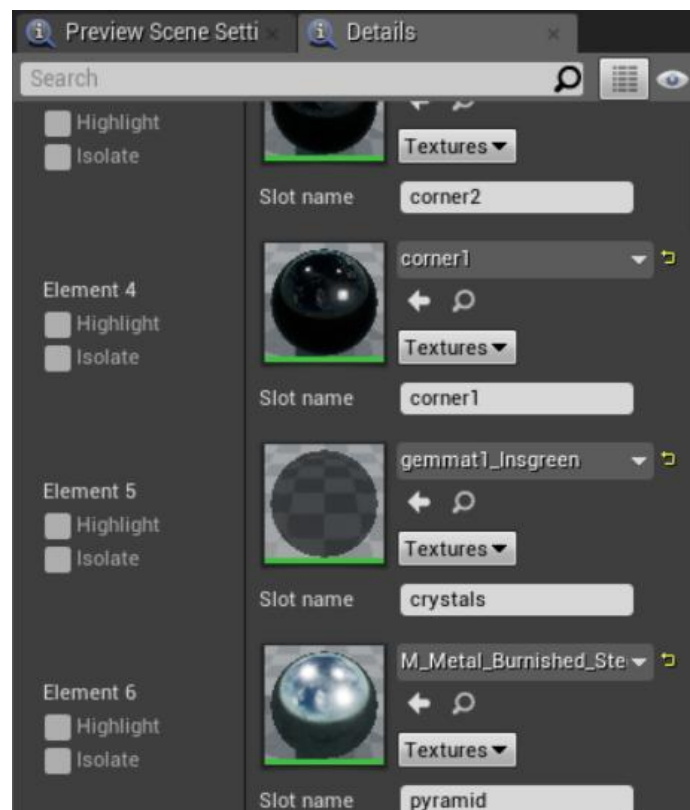


Εικόνα 51: Content Browser στην Unreal Engine 4

Στο Content Browser (Εικόνα 51) βρίσκονται αποθηκευμένα υλικά και εφέ, όπως η φωτιά από την Εικόνα 45, που μας προσφέρει έτοιμα η Unreal Engine 4, αλλά και αντικείμενα που έχουμε εισάγει από άλλα προγράμματα. Όπως το κτίριο της Εικόνας 52 που έχουμε κάνει import από το Blender.



Εικόνα 52: Κτίριο που έχει εισαχθεί από το Blender στην Unreal Engine 4



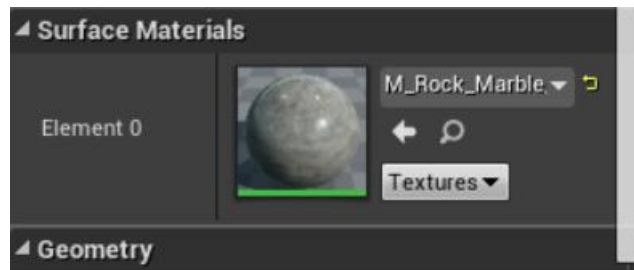
Εικόνα 53: Materials Αντικειμένου στην Unreal Engine 4

Στο κτίριο της Εικόνας 52 έχουμε ήδη τοποθετήσει materials, κάνοντας διπλό Αριστερό Κλικ πάνω του μπορούμε να τοποθετήσουμε τα materials που επιθυμούμε επιλέγοντας από μια λίστα από materials (Εικόνα 53).

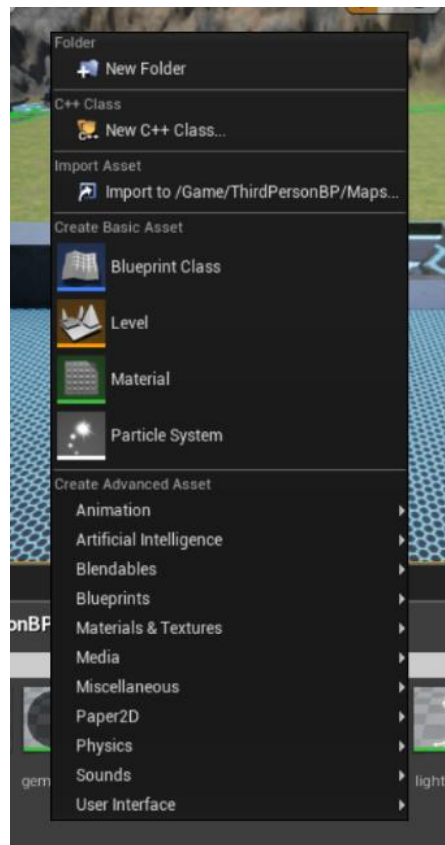


Εικόνα 54: Ομαδική Επιλογή Material στην Unreal Engine 4

Πατώντας Ctrl + Δεξί Κλικ (Εικόνα 54) και κάνοντας πολλαπλές επιλογές, μπορούμε να αναθέσουμε ομαδικά ένα συγκεκριμένο material (Εικόνα 55). Έτσι μπορούμε εύκολα να βάλουμε το ίδιο material σε πολλές επιφάνειες, ταυτόχρονα, αντί να το αλλάζουμε στην κάθε μία ξεχωριστά.

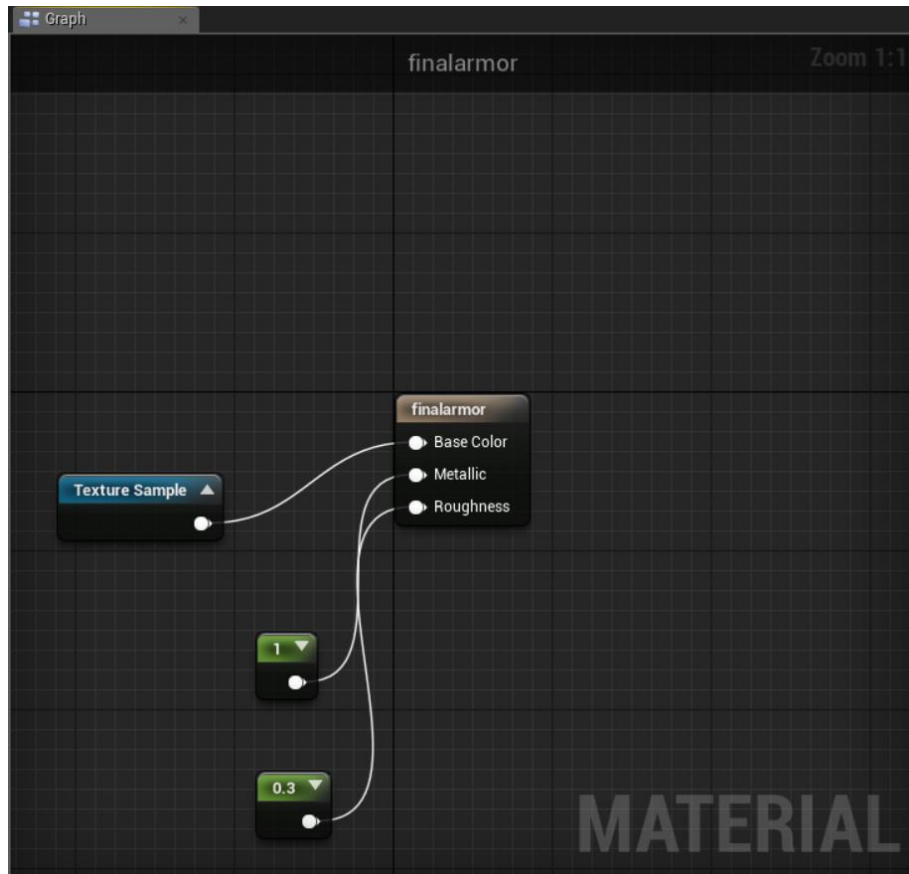


Εικόνα 55: Επιλογή Material στην Unreal Engine 4



Εικόνα 56: Δημιουργία Νέου Material στην Unreal Engine 4

Ο Content Browser (Εικόνα 51) μας δίνει τη δυνατότητα να δημιουργήσουμε δικά μας materials, όταν πατήσουμε Δεξί Κλικ μέσα του και επιλέγοντας από το Create Basic Asset την επιλογή Material (Εικόνα 56). Για να φτιάξουμε το χρώμα που επιθυμούμε φτιάχνουμε ένα material με διπλό Αριστερό Κλικ και το ανοίγουμε για επεξεργασία.

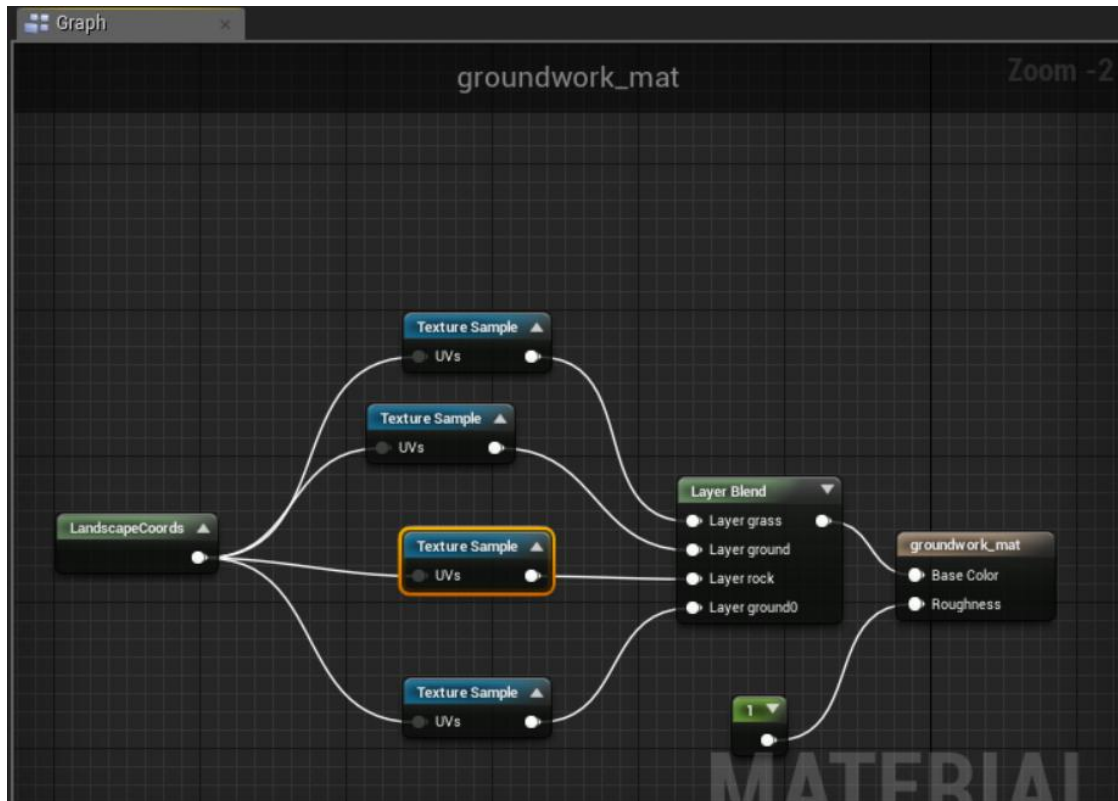


Εικόνα 57: Material Blueprint στην Unreal Engine 4

Από την Εικόνα 57 βλέπουμε ότι για χρώμα δώσαμε ένα ήδη υπάρχον texture στο Base Color, ενώ στα Metallic και Roughness ενώσαμε δύο σταθερές, ώστε να μειώσουμε το Roughness και να διατηρήσουμε το Metallic (Εικόνα 57). Η Εικόνα 58 παρουσιάζει μία προεπισκόπηση (preview) του νέου χρώματος, που έχουμε φτιάξει.

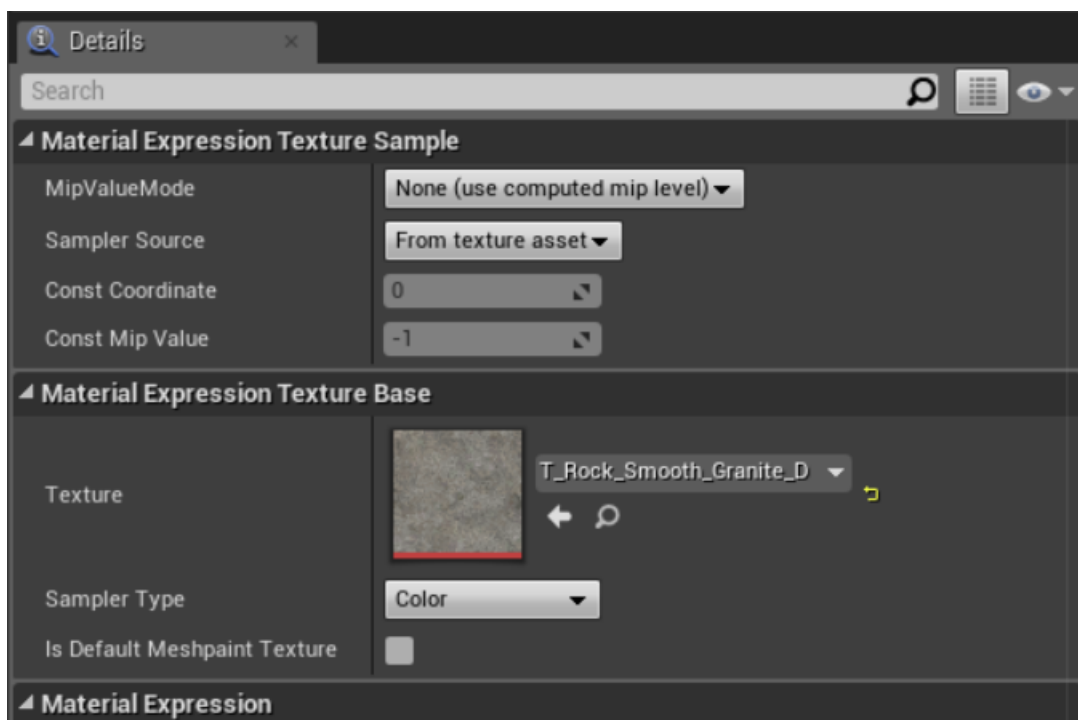


Εικόνα 58: Material Preview στην Unreal Engine 4



Εικόνα 59: Material Blueprint στην Unreal Engine 4

Για να φτιάξουμε το material του ground θα χρησιμοποιήσουμε 4 ήδη υπαρκτά materials, τα οποία καλούνται από Texture Samples μέσω ενός Layer Blend στο Base Color του material μας (Εικόνα 59). Τα Texture Samples είναι συνδεδεμένα σε ένα LandscapeCoords, έτσι ώστε να έχουν πιο ακριβή τοποθέτηση.

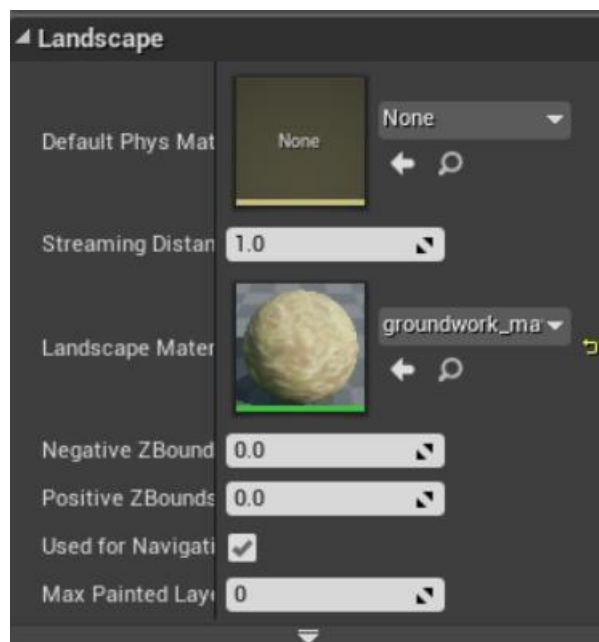


Εικόνα 60: Ρυθμίσεις Texture στην Unreal Engine 4

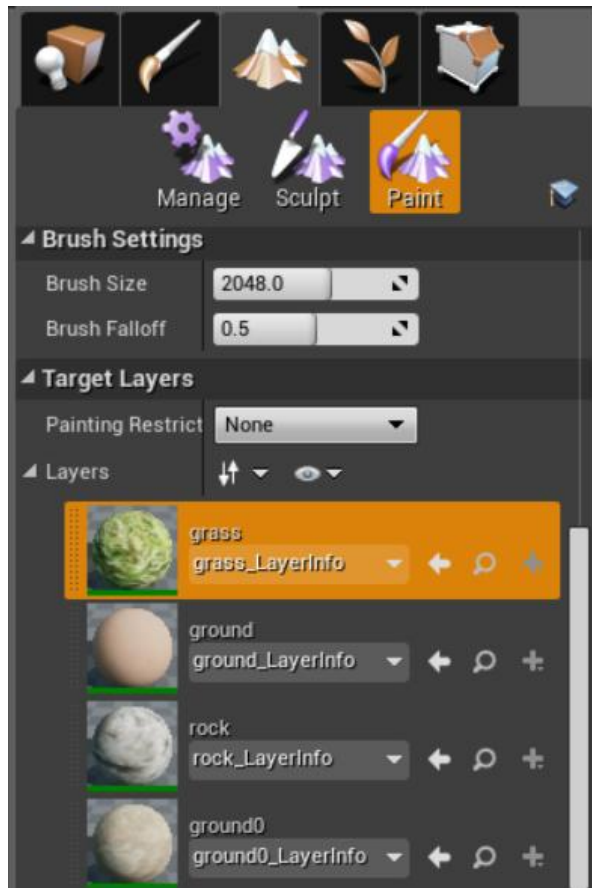
Στην Εικόνα 60 βλέπουμε το περιεχόμενο ενός Texture Sample, όπου του έχουμε δώσει texture, παρομοίως και στα άλλα τρία, έχουμε δώσει στο καθένα το δικό του texture. Η Εικόνα 61 μας παρουσιάζει μία προεπισκόπηση (preview) του νέου material, που έχουμε φτιάξει. Αφού τελειώσουμε κάνουμε Apply και Save. Έπειτα, τοποθετούμε το ground material, που φτιάξαμε, σαν Landscape Material (Εικόνα 62).



Εικόνα 61: Material Preview στην Unreal Engine 4



Εικόνα 62: Landscape Material στην Unreal Engine 4

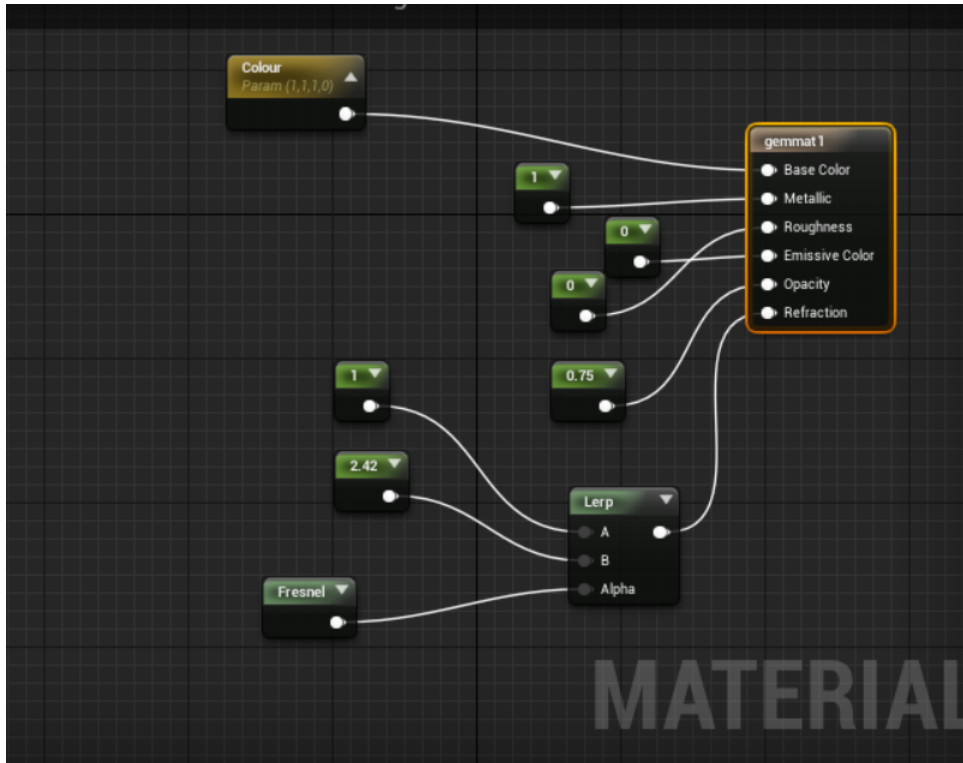


Εικόνα 63: Material Layers στην Unreal Engine 4

Από το πεδίο Landscape επιλέγουμε Paint και παρατηρούμε ότι μας έχει εμφανίσει τα Layers των materials του ground (Εικόνα 63). Όταν θα χρησιμοποιήσουμε το Brush για να δώσουμε χρώμα στην πίστα μας, ανάλογα με το πιο Layer επιλέξουμε, θα ζωγραφίζει με το αντίστοιχο material (Εικόνα 64).

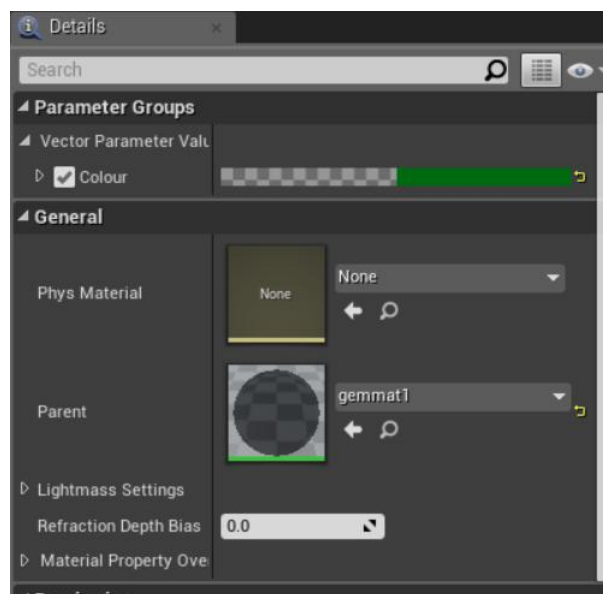


Εικόνα 64: Παράδειγμα χρήσης του Paint με το ground material στην Unreal Engine 4



Εικόνα 65: Material Blueprint στην Unreal Engine 4

Για να φτιάξουμε ένα material που θα έχει το texture ενός κρυστάλλου (Εικόνα 65), πρέπει στο Base Color να δώσουμε λευκό χρώμα, στο Metallic να δώσουμε τη τιμή 1, επειδή θέλουμε να διατηρήσουμε τη λάμψη του, να μηδενίσουμε το Roughness και Emissive Colour, ενώ να μειώσουμε το Opacity στο 0.75 χρησιμοποιώντας σταθερές. Για να πετύχουμε τις κατάλληλες τιμές στο Refraction χρησιμοποιήσαμε το LERP(Linear Interpolate), το οποίο μας επιτρέπει να επιλέξουμε μεταξύ δύο καναλιών A και B επιλέγοντας μέσω του ALPHA καναλιού. Με τη χρήση του Fresnel υπολογίζουμε την τιμή του φωτός στέλνοντας σήμα στο κανάλι ALPHA ώστε να επιλέξει μεταξύ της A και B τιμής.

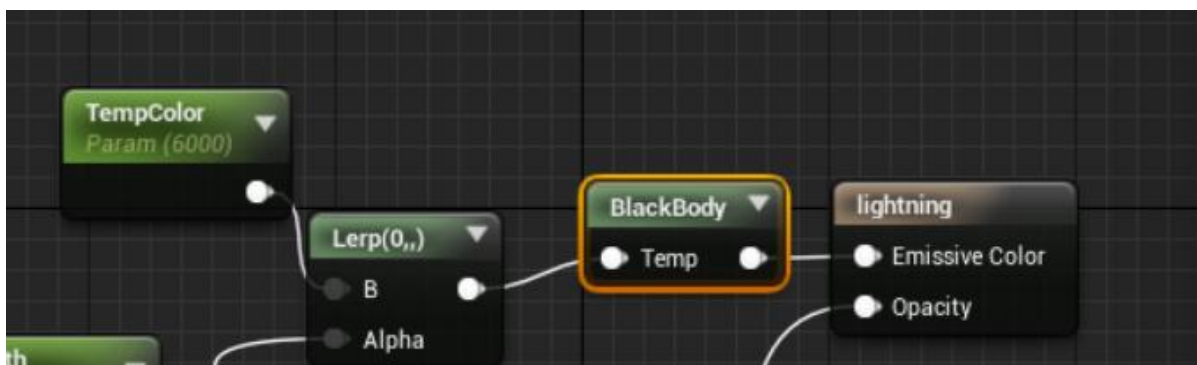


Εικόνα 66: Material Color στην Unreal Engine 4

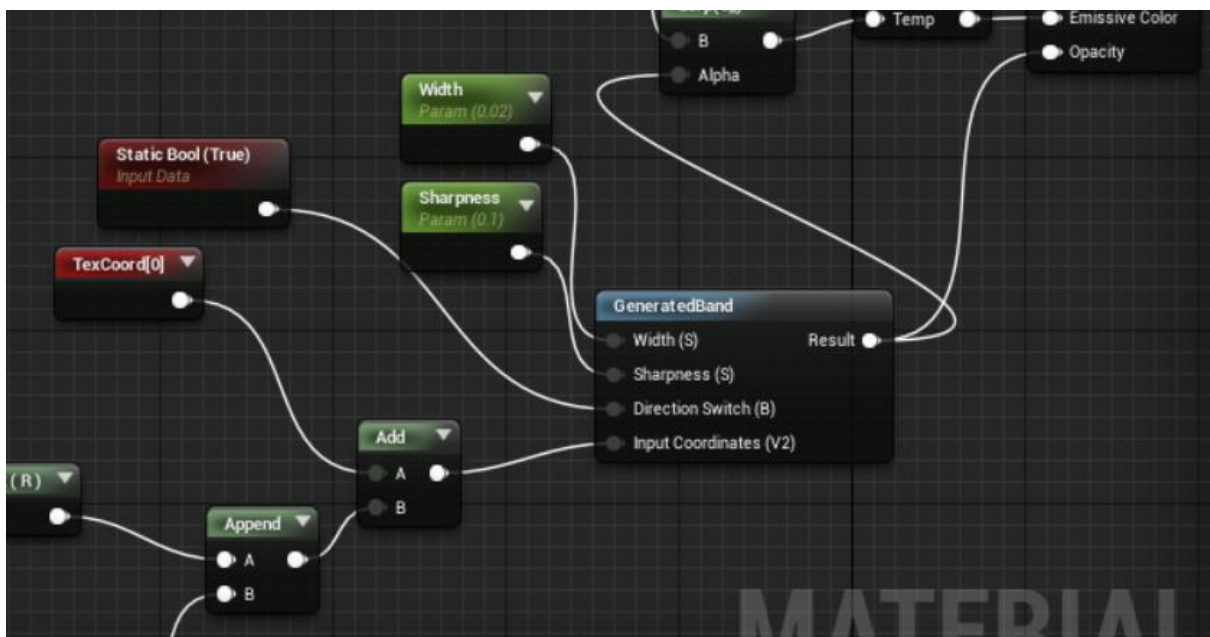
Πατώντας Δεξί Κλικ στο material του κρυστάλλου μας, φτιάχνουμε ένα instance, του οποίου του δίνουμε το χρώμα που χρειαζόμαστε στο πεδίο Colour (Εικόνα 66). Η Εικόνα 67 παρουσιάζει μία προεπισκόπηση (preview) του νέου material, που έχουμε φτιάξει. Όταν θέλουμε να χρησιμοποιήσουμε σαν material τον άχρωμο κρύσταλλο, τότε χρησιμοποιούμε το αρχικό material, σε περίπτωση που θέλουμε τον έγχρωμο κρύσταλλο, τότε χρησιμοποιούμε το instance σαν material.



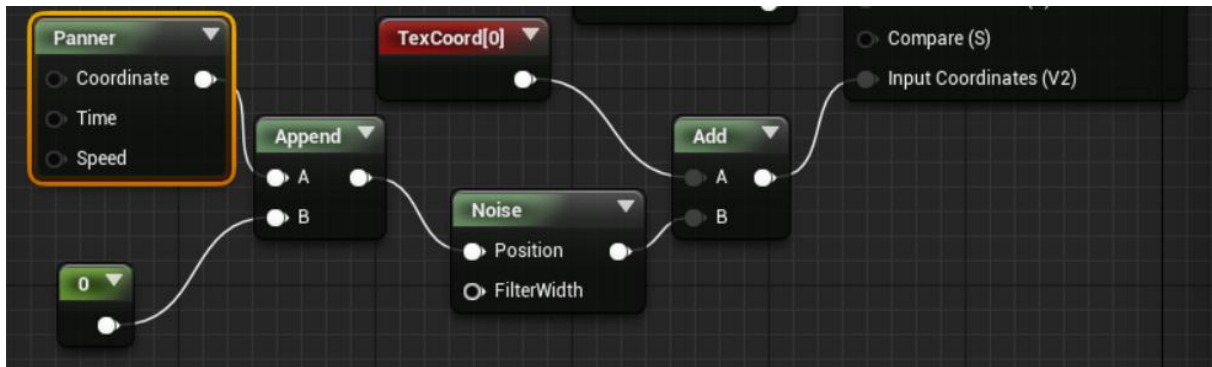
Εικόνα 67: Material Preview στην Unreal Engine 4



Εικόνα 68: Material Blueprint στην Unreal Engine 4



Εικόνα 69: Material Blueprint στην Unreal Engine 4



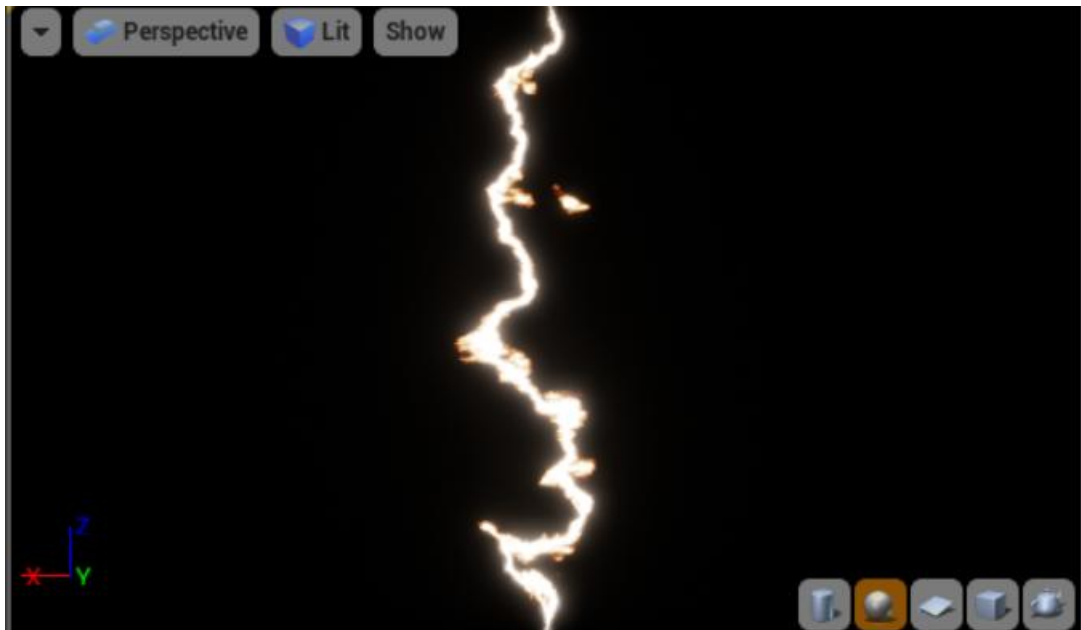
Εικόνα 70: Material Blueprint στην Unreal Engine 4

Για να φτιάξουμε το material της αστραπής (Εικόνα 68, Εικόνα 69, Εικόνα 70 και Εικόνα 71), χρειάζεται να επηρεάσουμε τα πεδία Emissive Colour και Opacity του material.

Το Emissive Colour επιτρέπει στο material να έχει χρώμα που αφήνει ακτινοβολία. Για να το πετύχουμε αυτό χρησιμοποιούμε μια μεταβλητή που θα κρατά προσωρινά μια τιμή, που παίρνει από ένα LERP, στο οποίο έχει στο B κανάλι του τη τιμή 6000 και δηλώνει το χρώμα του, ενώ στο ALPHA κανάλι του παίρνει τιμές από ένα Generated Band το οποίο τροφοδοτεί με τιμές και το Opacity του material.

Το Generated Band μας επιτρέπει να φτιάξουμε ένα Band από διάφορα μεγέθη μοτίβων. Χρησιμοποιούμε παραμέτρους για να ρυθμίσουμε το Sharpness και το Width, ενώ ορίζουμε True το Direction Switch, έτσι ώστε το μοτίβο μας να είναι κάθετο.

Για να πετύχουμε στην κατασκευή του υλικού, που έχει κίνηση, θα πρέπει να επηρεάσουμε την τιμή στο πεδίο Input Coordinates του Generated Band. Με τη χρήση του Panner, το οποίο χρησιμοποιείται για τη δημιουργία κινούμενων materials, ρυθμίσαμε την ταχύτητα στο άξονα του X ώστε να πετύχουμε οριζόντια κίνηση. Με τη χρήση του Append ενώνουμε το αποτέλεσμα του Panner με μια const, ώστε να δημιουργήσουμε ένα διάνυσμα. Έπειτα, περνάμε το διάνυσμα από ένα φίλτρο θορύβου το οποίο επηρεάζει το μέγεθός του. Εν τέλει με τη χρήση της Add προσθέτουμε τις τιμές του διανύσματος μας με τις τιμές του TexCoord, το οποίο έχει τη δυνατότητα να αλλάξει τις τιμές των tiles του material και το αποτέλεσμά του δίνεται σαν τιμή για το Input Coordinates.



Εικόνα 71: Lightning Material Preview στην Unreal Engine 4

3.3.2. Blueprints και Προγραμματισμός Παιχνιδιού

3.3.2.1. Character Blueprints

Πρώτα, πρέπει να φτιάξουμε τα αρχεία και τα blueprints που δεν έχει το project, για να τα χρησιμοποιήσουμε αργότερα. Στο Content Browser (το οποίο βρίσκεται στο κάτω μέρος του προγράμματος, αλλιώς το ανοίγουμε από το Window → Content Browser) πατάμε δεξί κλικ, για να εμφανιστεί το menu. Τα Assets που χρειαζόμαστε είναι Level, Animation Blueprint (από τη κατηγορία Animation) και Widget Blueprint (από τη κατηγορία User Interface). Παρακάτω θα δούμε πόσα από το καθένα θέλουμε και πως θα τα χρησιμοποιήσουμε στο Project.

Αρχικά, πρέπει να κάνουμε Copy(Ctrl+C) + Paste(Ctrl+V) τα ThirdPersonCharacter και ThirdPerson_AnimBP, έπειτα πρέπει να τα μετονομάσουμε σε PlayerCharacter και PlayerCharacter_Anim_BP και κάνουμε αλλαγές στις ρυθμίσεις όπως παρακάτω. Επίσης, στο ThirdPersonGameMode πρέπει να αλλάξουμε το Default Pawn Class σε PlayerCharacter, σε αυτή τη ρύθμιση πρέπει πάντα να βάζουμε το blueprint του χαρακτήρα του παίκτη.

Στο PlayerCharacter, πάμε Components → Mesh(inherited) και μετά Details → Animation → Animation Mode → Use Animation Blueprint → Anim Class → επιλέγουμε το Animation Blueprint μας → Mesh → Skeletal Mesh → επιλέγουμε το χαρακτήρα που εισάγαμε από το Blender. Έπειτα, πάμε Capsule Component → Shape → Capsule Half Height → 135 και Capsule Radius → 110. Επίσης, προσαρμόζουμε το CameraBoom και το FollowCamera.

Το τελευταίο βήμα, πριν ξεκινήσουμε να ασχολούμαστε με τον κώδικα, είναι να δημιουργήσουμε τα sockets στο σκελετό, που θα χρησιμοποιήσουμε για τα αντικείμενά μας, τα οποία πρέπει να τοποθετηθούν πάνω στο χαρακτήρα μας. Ανοίγοντας το αρχείο του σκελετού, θα δούμε όλα τα κόκκαλα στην αριστερή καρτέλα, η οποία λέγεται Skeleton Tree Tab. Στην αναζήτηση μπορούμε να βρούμε το κόκκαλο που θέλουμε πληκτρολογώντας το όνομα, αν δε μπορούμε να το βρούμε στην καρτέλα, κάνουμε δεξί κλικ πάνω στο κόκκαλο, επιλέγουμε Add Socket και το ονομάζουμε. Έπειτα, κάνουμε δεξί κλικ πάνω στο socket,

επιλέγουμε Add Preview Asset και βρίσκουμε το αντικείμενο από την αναζήτηση και το προσθέτουμε στο socket. Τέλος, πρέπει να κάνουμε αλλαγές στις τιμές Location, Rotation και Scale για να πάει το αντικείμενο στη σωστή θέση. Όμως, τα αντικείμενα δεν είναι τοποθετημένα στο μοντέλο του χαρακτήρα μας, είναι μόνο προεπισκόπηση για να ρυθμίσουμε τα sockets, η τοποθέτηση γίνεται αμέσως μετά, αφού φτιάξουμε και ρυθμίσουμε όλα τα sockets που χρειαζόμαστε (Εικόνα 72).



Εικόνα 72: Sockets του Σκελετού στην Unreal Engine 4

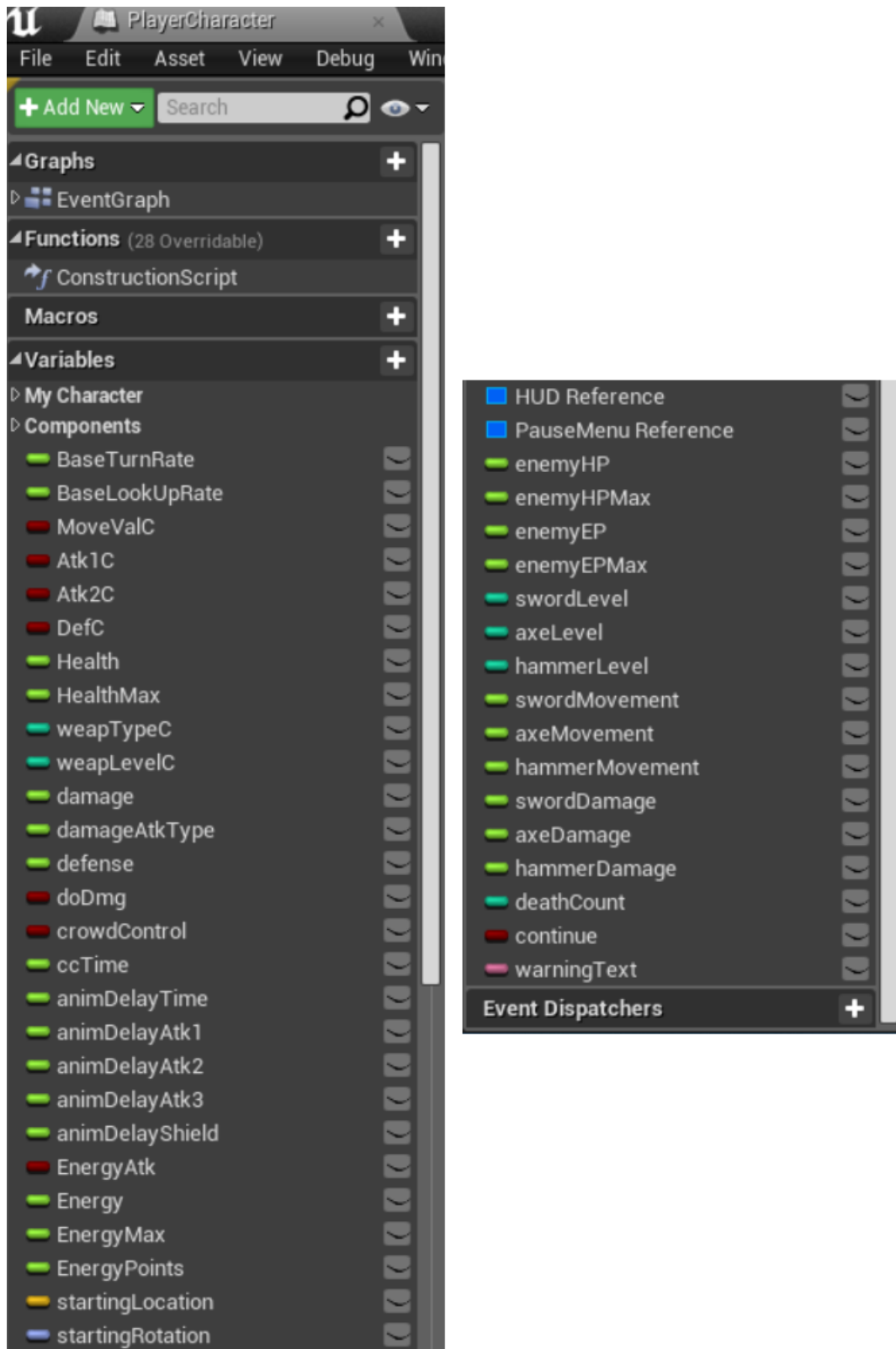
Κλείνοντας αυτό το βήμα, θα βάλουμε τα αντικείμενά μας πάνω στο μοντέλο του χαρακτήρα μας, χρησιμοποιώντας τα sockets που φτιάξαμε παραπάνω. Πρώτα, ανοίγουμε το

PlayerCharacter Blueprint και πάμε στο Viewport Tab, για να βλέπουμε το μοντέλο. Έπειτα, πάμε στο Components Tab και επιλέγουμε το Mesh(Inherited), μετά μικραίνουμε το παράθυρο για να μπορούμε να έχουμε εύκολη πρόσβαση στο Content Browser από το κύριο παράθυρο. Αναζητώντας τα αντικείμενα στο Content Browser, επιλέγοντάς τα και πατώντας Add Component στο Components Tab μπορούμε να βρούμε τα αντικείμενα στη λίστα που εμφανίζεται και να τα προσθέσουμε στο μοντέλο του χαρακτήρα μας (Σημείωση: πρέπει πάντα να έχουμε επιλεγμένο το Mesh(Inherited) και μετά να πατάμε το Add Component, γιατί αλλιώς το αντικείμενο δεν θα εφαρμόζει στο μοντέλο του χαρακτήρα μας, όταν του βάλουμε το socket, όπως το είχαμε ρυθμίσει παραπάνω). Αφού προσθέσουμε όλα τα αντικείμενα στο μοντέλο του χαρακτήρα μας (Εικόνα 73), επιλέγουμε το κάθε αντικείμενο, πάμε στο Details Tab και μετά πάμε Sockets → Parent Socket και επιλέγουμε το socket που θέλουμε για το κάθε αντικείμενο.



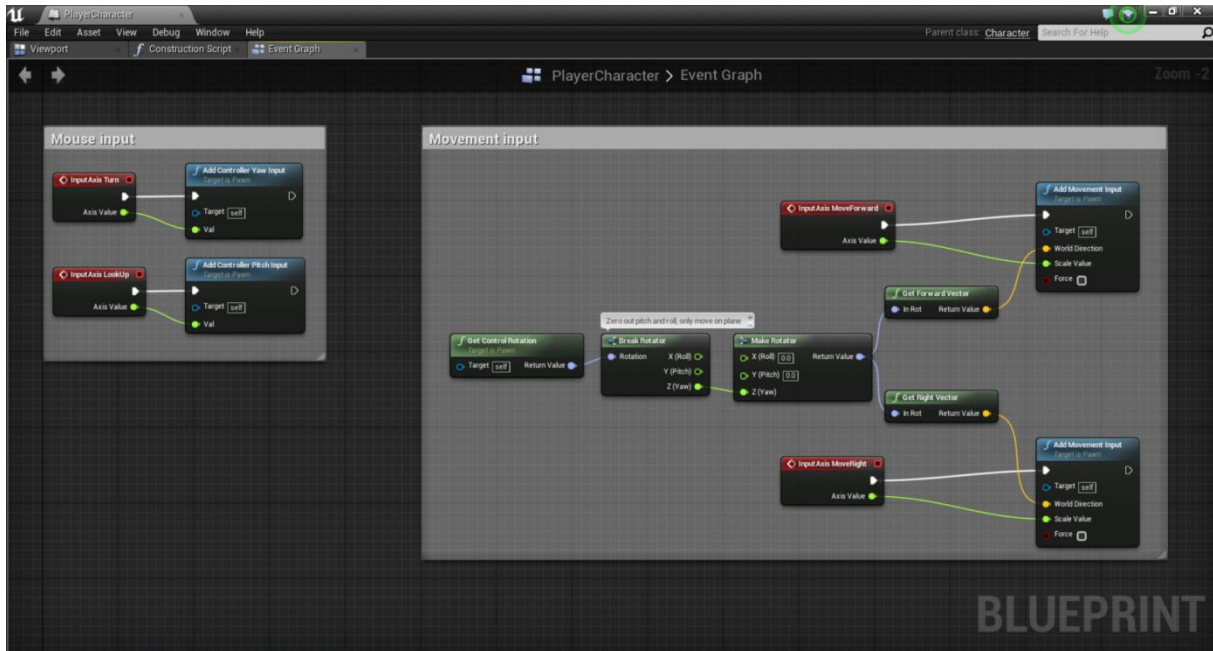
Εικόνα 73: Αντικείμενα πάνω στο Μοντέλο του Χαρακτήρα στην Unreal Engine 4

Στη συνέχεια, θα φτιάξουμε τα blueprints, που είναι ο τρόπος που στην Unreal Engine 4 γράφουμε κώδικα, θα δούμε πρώτα το blueprint του χαρακτήρα του παίκτη και θα εξηγήσουμε τη λειτουργία κάθε κομματιού του blueprint.

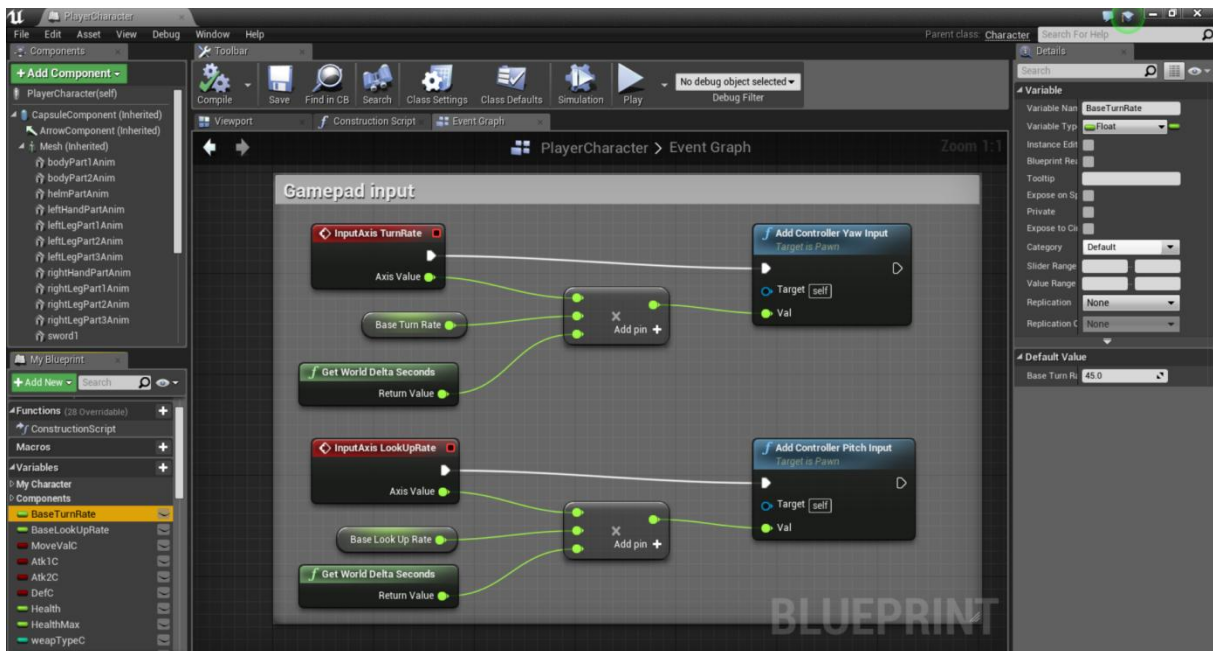


Εικόνα 74: Μεταβλητές Player Character στην Unreal Engine 4

Από το blueprint του χαρακτήρα, από το Third Person template, κρατάμε το Mouse input, το Movement input και το Gamepad input. Με αυτά ο χαρακτήρας μας μπορεί να στρίβει δεξιά-αριστερά με τα αντίστοιχα βελάκια και το ποντίκι ή να κοιτά πάνω-κάτω με το ποντίκι και να πηγαίνει μπρος-πίσω με τα αντίστοιχα βέλη ή τα κουμπιά W και S του πληκτρολογίου (Εικόνες 75 και 76). Επίσης, προσθέτουμε τις μεταβλητές που βλέπουμε στην Εικόνα 74, οι οποίες θα μας χρησιμεύσουν αργότερα.

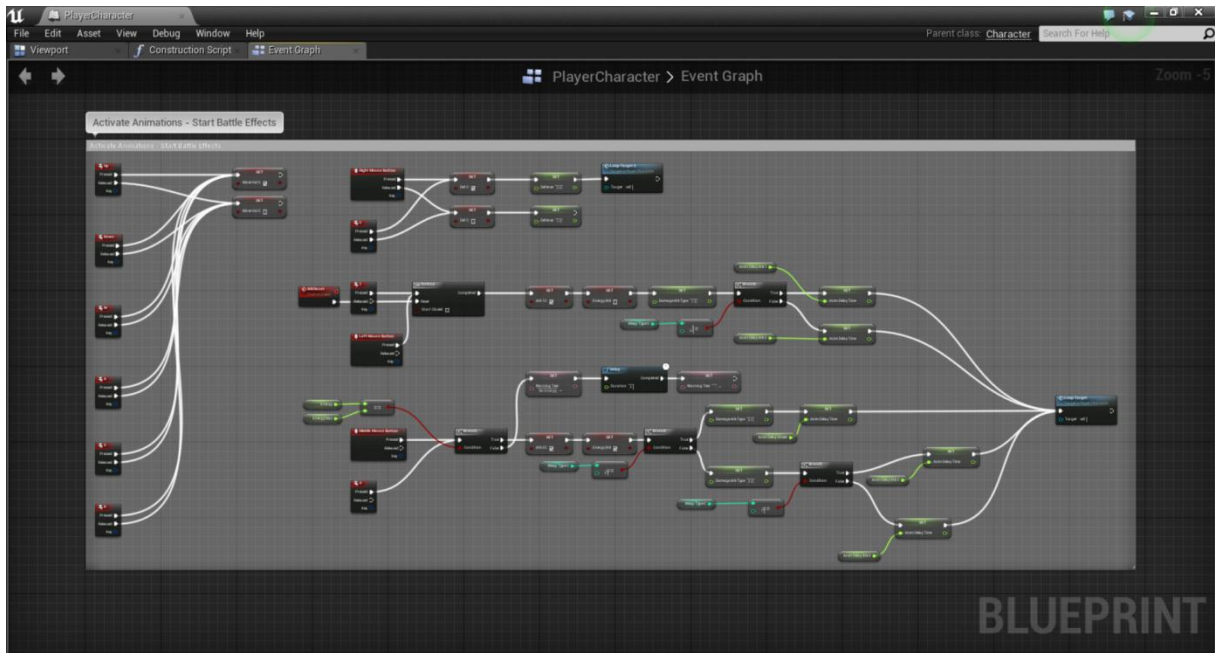


Εικόνα 75: Player Character Blueprint στην Unreal Engine 4



Εικόνα 76: Player Character Blueprint στην Unreal Engine 4

Έπειτα έχουμε την αρχικοποίηση μεταβλητών και την εγκατάσταση του HUD (Heads-Up Display) UI στο χαρακτήρα του παίκτη (Εικόνα 77). Το HUD θα το αναλύσουμε παρακάτω, κανονικά αυτό το κομμάτι του blueprint χρειάζεται αντικείμενα που δεν έχουμε αναλύσει ακόμα και πρέπει να γίνει αφού ολοκληρώσουμε τη βασική λειτουργία του HUD,

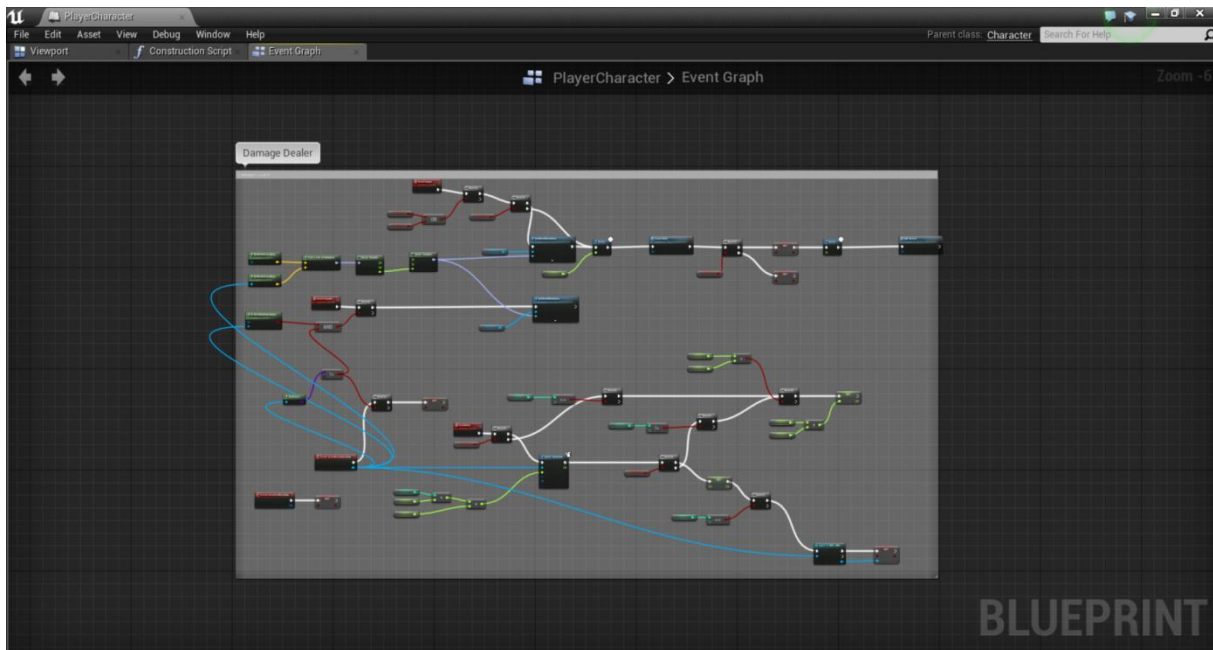


Εικόνα 78: Player Character Blueprint στην Unreal Engine 4

Σε αυτό το κομμάτι του blueprint του χαρακτήρα του παίκτη ελέγχουμε τις μεταβλητές που επηρεάζουν τα animations και τις λειτουργίες της μάχης (Εικόνα 78). Στα αριστερά βλέπουμε ότι όταν πατάμε τα κουμπιά W, A, S, D, Πάνω και Κάτω βελάκι γίνεται αληθής μια μεταβλητή και όταν τα αφήνουμε γίνεται ψευδής, αυτή η μεταβλητή όσο είναι αληθής ενεργοποιεί το animation του περπατήματος και όταν γίνεται ψευδής το animation σταματά. Το πάνω κομμάτι του διαγράμματος, πατώντας C ή το δεξί κουμπί του ποντικιού, επηρεάζει τις μεταβλητές του animation της άμυνας, τη μεταβλητή του παράγοντα της άμυνας, που χρησιμοποιείται για να αλλάξουμε τη ζημιά που δέχεται ο χαρακτήρας, και όταν πατάμε ένα από τα κουμπιά για άμυνα καλούμε και ένα Custom Event που καλεί ένα άλλο κομμάτι του blueprint που κάνει τον χαρακτήρα να γυρνά πάντα κατά το στόχο του, όταν έχει. Το αμέσως επόμενο κομμάτι, πατώντας Z ή το αριστερό κουμπί του ποντικιού, επηρεάζει τις μεταβλητές επίθεσης, έτσι ώστε να εκτελεστεί η απλή επίθεση στη μάχη αλλά και το αντίστοιχο animation, ανάλογα το όπλο που χρησιμοποιούμε. Αυτό το κομμάτι, όπως και το επόμενο καταλήγουν σε ένα Custom Event που αλλάζει την κατεύθυνση του χαρακτήρα, έτσι ώστε να βλέπει πάντα το στόχο του, αν έχει, κατά την εκτέλεση του animation της επίθεσης, και κάνει ζημιά στο στόχο. Στο τελευταίο κομμάτι γίνεται ακριβώς το ίδιο με το πάνω κομμάτι, αλλά αφορά την ειδική επίθεση του χαρακτήρα, που εκτελείται πατώντας X ή το μεσαίο κουμπί του ποντικιού, γι' αυτό το λόγο και κάνει έλεγχο της μεταβλητής του έχει την τιμή της Ενέργειας(Energy) του χαρακτήρα, που είναι ο πόρος(resource) που χρειαζόμαστε για να γίνει η ειδική επίθεση, αλλιώς εκτυπώνεται μήνυμα ότι δεν υπάρχει αρκετή Ενέργεια.

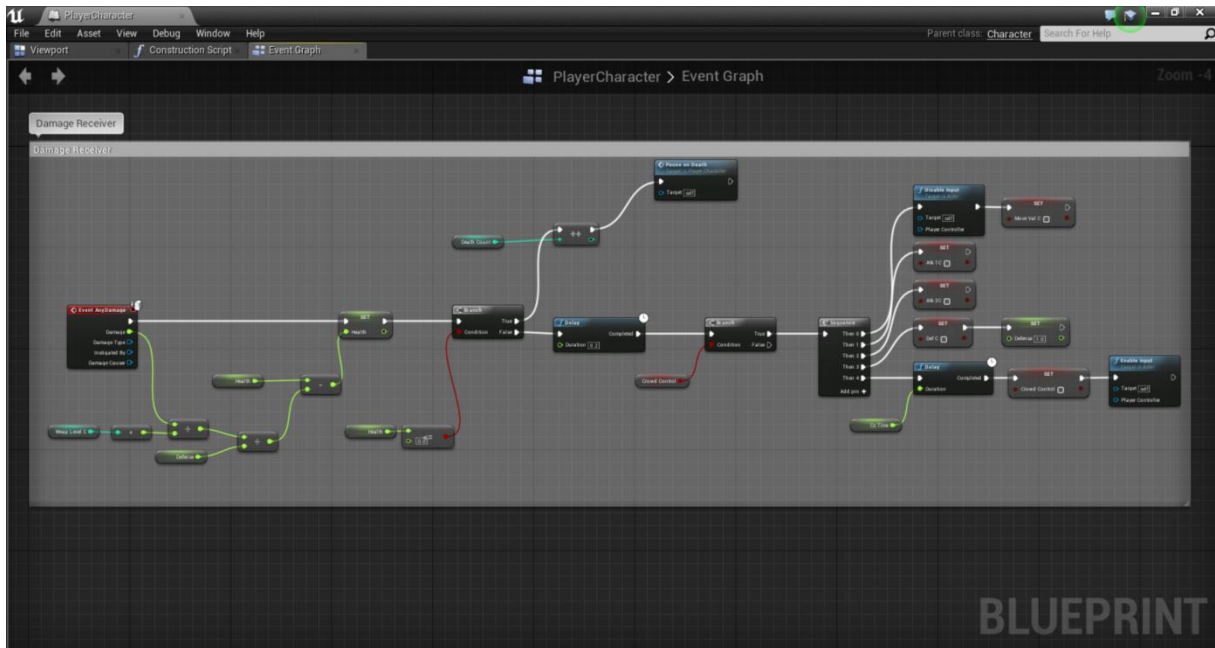
Τα Custom Events μας επιτρέπουν να συνδέσουμε διαφορετικά τμήματα ενός blueprint χωρίς να κάνουμε πολύπλοκες συνδέσεις ή να χρειάζεται να βάλουμε τα διαγράμματα κοντά μεταξύ τους, επίσης μπορούν να χρησιμοποιηθούν σαν functions, καλώντας διαγράμματα που κάνουν μια συγκεκριμένη λειτουργία και δεν χρησιμοποιούνται από μόνα τους. Αλλά και σαν την εντολή GoTo, ενεργοποιώντας άλλα κομμάτια του blueprint, τα οποία όμως εκτελούνται κανονικά και χωρίς να κληθούν από τα Custom Events, αλλά όπως και η εντολή GoTo αυτός ο τρόπος είναι επικίνδυνος για την σωστή ροή και λειτουργία του αλγόριθμού μας.

Τα animations ελέγχονται από το αντίστοιχο Animation Blueprint του κάθε χαρακτήρα, το οποίο θα το δούμε πιο αναλυτικά παρακάτω.



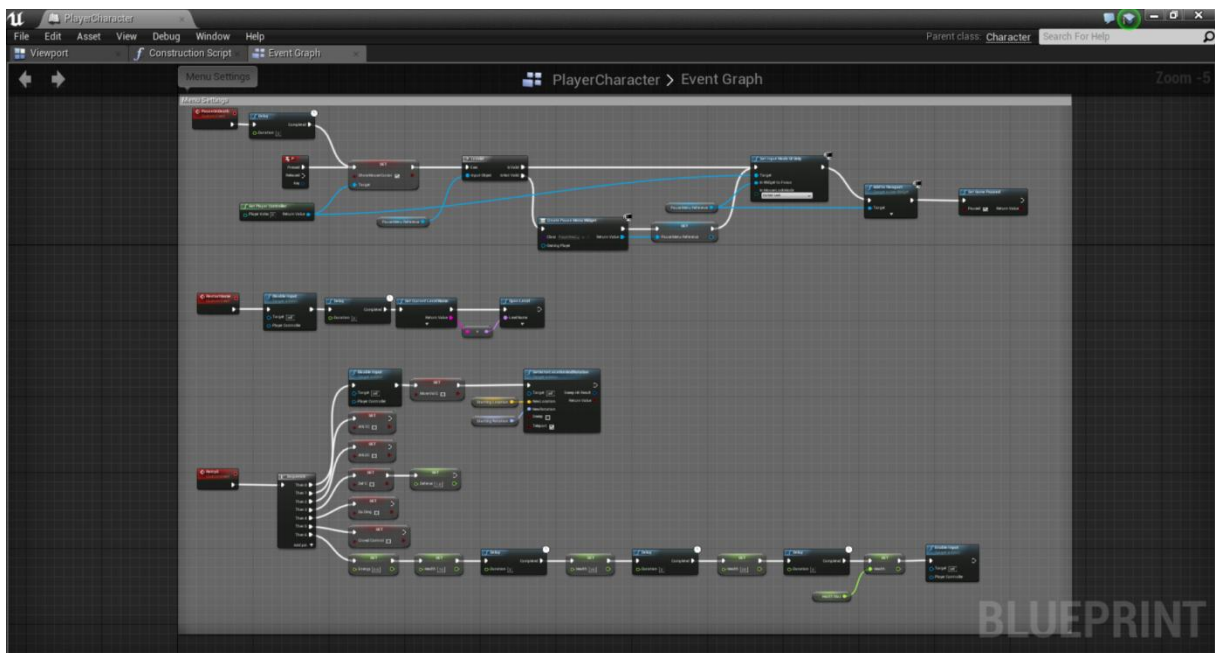
Εικόνα 79: Player Character Blueprint στην Unreal Engine 4

Στην Εικόνα 79 βλέπουμε το Damage Dealer τμήμα του blueprint, το οποίο έχει πέντε κόκκινους κόμβους από τους οποίους ξεκινούν τα πέντε διαφορετικά τμήματα του διαγράμματος. Το κάτω τμήμα ξεκινά με τον κόμβο Event ActorEndOverlap και κάνει τη μεταβλητή, που ελέγχει την ικανότητα του χαρακτήρα να κάνει ζημιά στον αντίπαλο, ψευδή, έτσι ώστε όταν ο χαρακτήρας σταματήσει να κάνει επαφή με έναν αντίπαλο, να μη μπορεί να κάνει ζημιά πλέον. Το αμέσως επόμενο τμήμα ξεκινά με τον κόμβο Event ActorBeginOverlap και κάνει τη μεταβλητή, που ελέγχει την ικανότητα του χαρακτήρα να κάνει ζημιά στον αντίπαλο, αληθή, όταν ο χαρακτήρας έρχεται σε επαφή με αντίπαλο, ενώ αγνοεί τα Trigger Box που θα δούμε την λειτουργία τους παρακάτω, έτσι ώστε να ξεκινήσει η μάχη και να μπορεί να κάνει ζημιά στον αντίπαλο. Το επόμενο τμήμα είναι υπεύθυνο για να κάνει ζημιά στον αντίπαλο και να μας δώσει Ενέργεια, αν χρησιμοποιούμε το Σπαθί ή το Τσεκούρι, ενώ σε περίπτωση που αστοχήσουμε, αν χρησιμοποιούμε το Σφυρί μας δίνει Ενέργεια, αλλιώς δεν κάνει τίποτα. Το διάγραμμα ξεκινά με το Custom Event loopDmg, το οποίο καλείται από άλλο τμήμα αυτού του διαγράμματος. Τα δύο τμήματα που απομένουν αφορούν το ίδιο θέμα, ξεκινούν με δύο Custom Events τα loopTarget και loopTarget2. Το loopTarget2 καλείται όταν πατάμε ένα από τα κουμπιά της άμυνας και στρέφει το χαρακτήρα προς το στόχο του, αν υπάρχει. Το loopTarget κάνει το ίδιο, αλλά καλείται όταν κάνει ο χαρακτήρας επίθεση, γι' αυτό το λόγο επηρεάζει μεταβλητές που έχουν να κάνουν με τα animations και επίσης καλεί το loopDmg για να κάνει ζημιά στο στόχο.



Εικόνα 80: Player Character Blueprint στην Unreal Engine 4

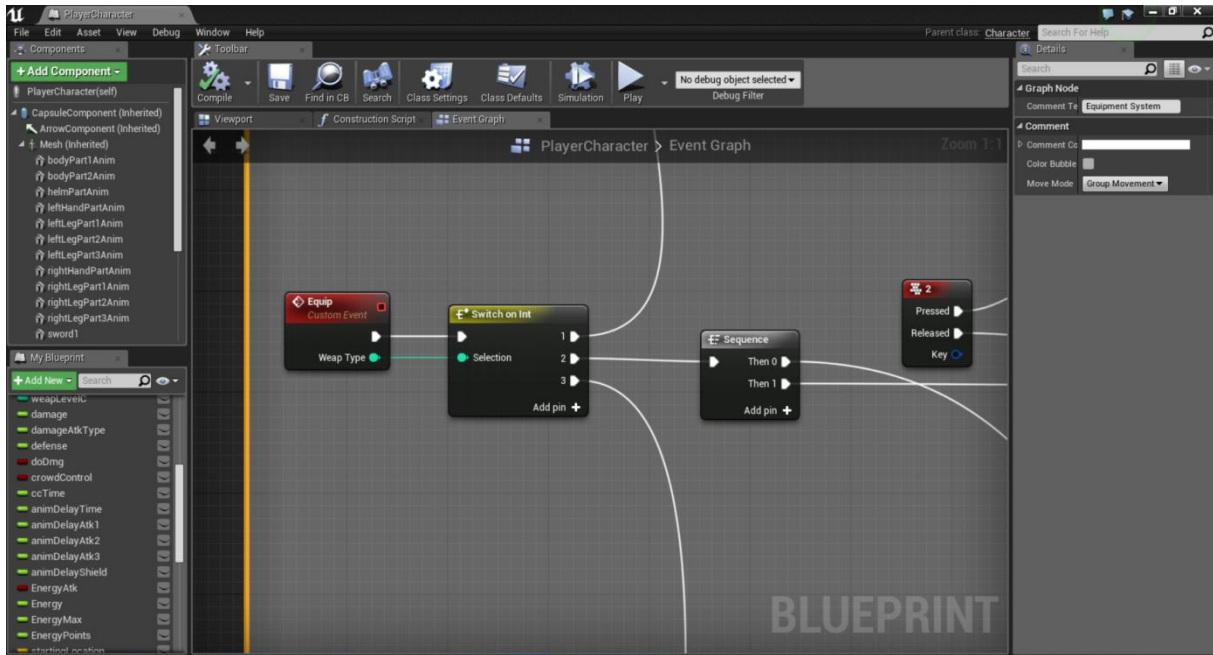
Το Damage Receiver τμήμα του blueprint (Εικόνα 80) είναι πολύ απλό σε σύγκριση με άλλα τμήματα. Αυτό που κάνει είναι να δέχεται τη ζημιά που προκαλεί ο αντίπαλος στο χαρακτήρα, υπολογίζει τη ζημιά που πρέπει να δεχτεί ο χαρακτήρας, η οποία επηρεάζεται αν πατήσουμε το κουμπί της άμυνας, και την αφαιρεί από τη ζωή του χαρακτήρα. Μετά γίνεται έλεγχος αν πέθανε ο χαρακτήρας, για να αυξήσει το Death Count και να ενεργοποιήσει το Pause Menu, ή αν δεν πέθανε ο χαρακτήρας ελέγχει αν ο αντίπαλος ακινητοποίησε το χαρακτήρα, με την ειδική επίθεση του Σφυριού, και αλλάζει τις τιμές των μεταβλητών κατάλληλα.



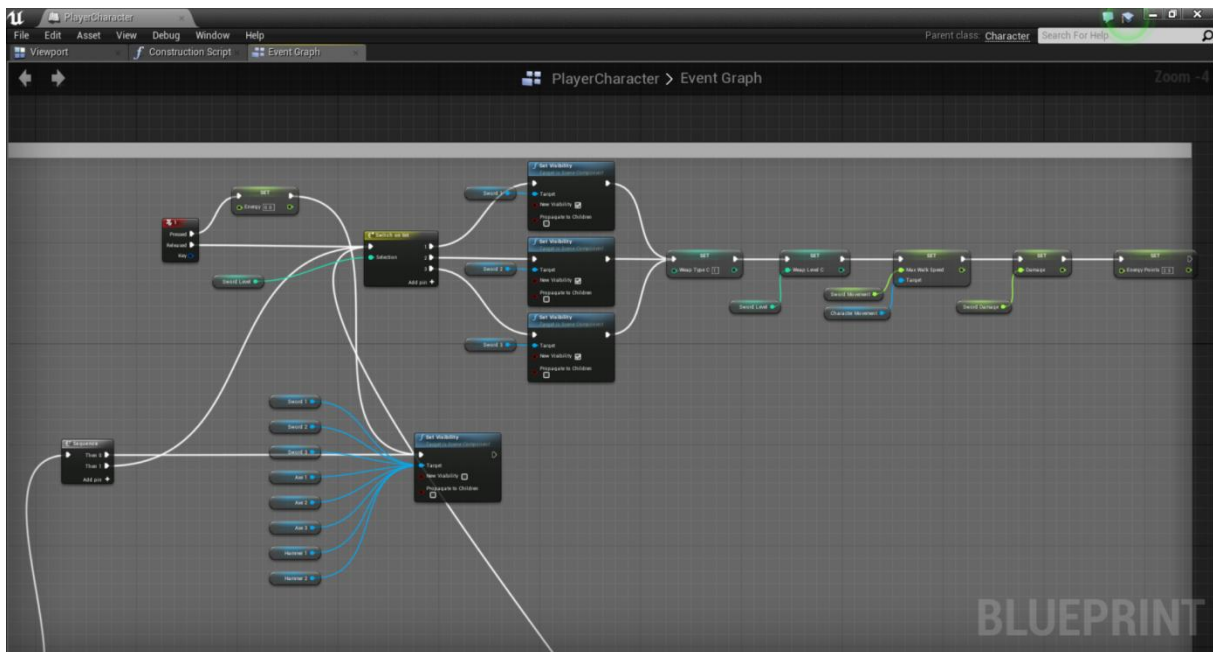
Εικόνα 81: Player Character Blueprint στην Unreal Engine 4

Το Menu Settings τμήμα του blueprint (Εικόνα 81) αποτελείται από τρία διαγράμματα, τα οποία χρησιμοποιούνται για τη χρήση των επιλογών του menu του παίκτη.

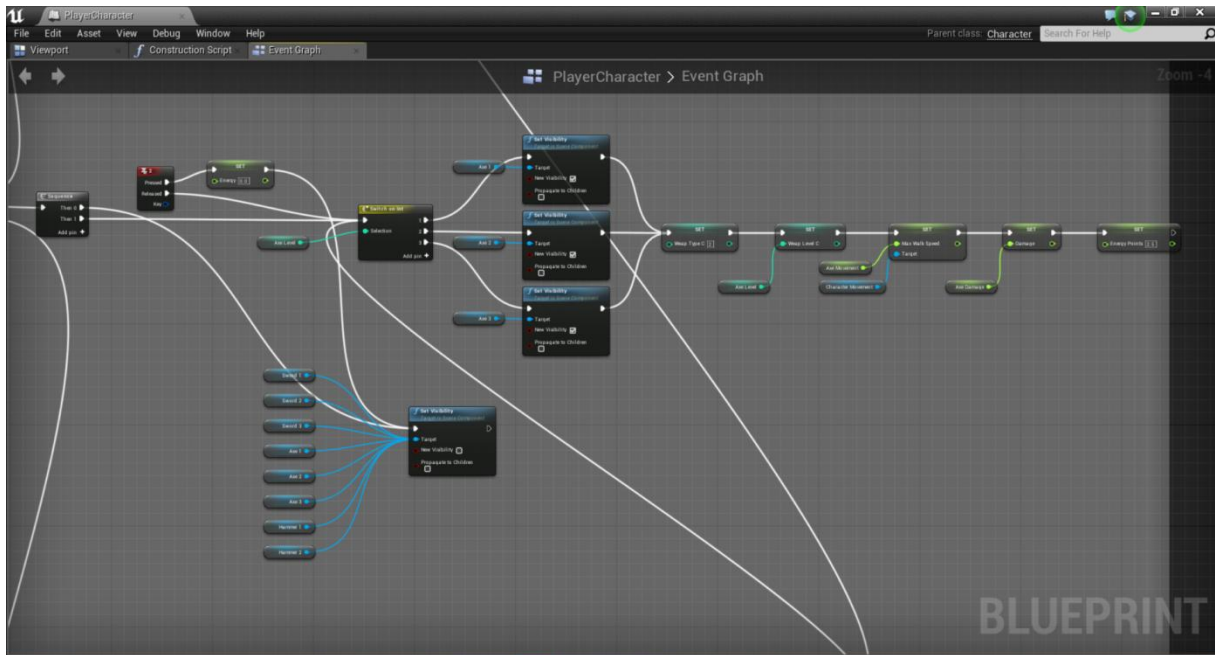
Το πρώτο διάγραμμα ενεργοποιείται αν πατήσουμε το κουμπί P ή καλέσουμε το Custom Event, όταν ο χαρακτήρας πεθάνει όπως αναφέραμε παραπάνω, στο Damage Receiver. Αυτό που κάνει είναι να εμφανίζει τον κέρσορα, να ελέγχει αν η μεταβλητή του Pause Menu είναι δεκτή, αν όχι τη δημιουργεί και της δίνει τη σωστή τιμή, και αμέσως μετά την προσθέτει στην εικόνα που βλέπει ο παίκτης και σταματά το παιχνίδι. Το δεύτερο διάγραμμα χρησιμοποιείται από το Pause Menu και η λειτουργία της είναι να φορτώνει από την αρχή το επίπεδο του παιχνιδιού όταν πατήσουμε το κουμπί Restart. Το τρίτο διάγραμμα, επίσης χρησιμοποιείται από το Pause Menu, χρησιμεύει για την περίπτωση που ο παίκτης θέλει να ξαναπροσπαθήσει, οπότε όλες οι σχετικές μεταβλητές παίρνουν την αρχική τους τιμή και ο χαρακτήρας εμφανίζεται στην αρχική τοποθεσία του, από όπου είχε ξεκινήσει.



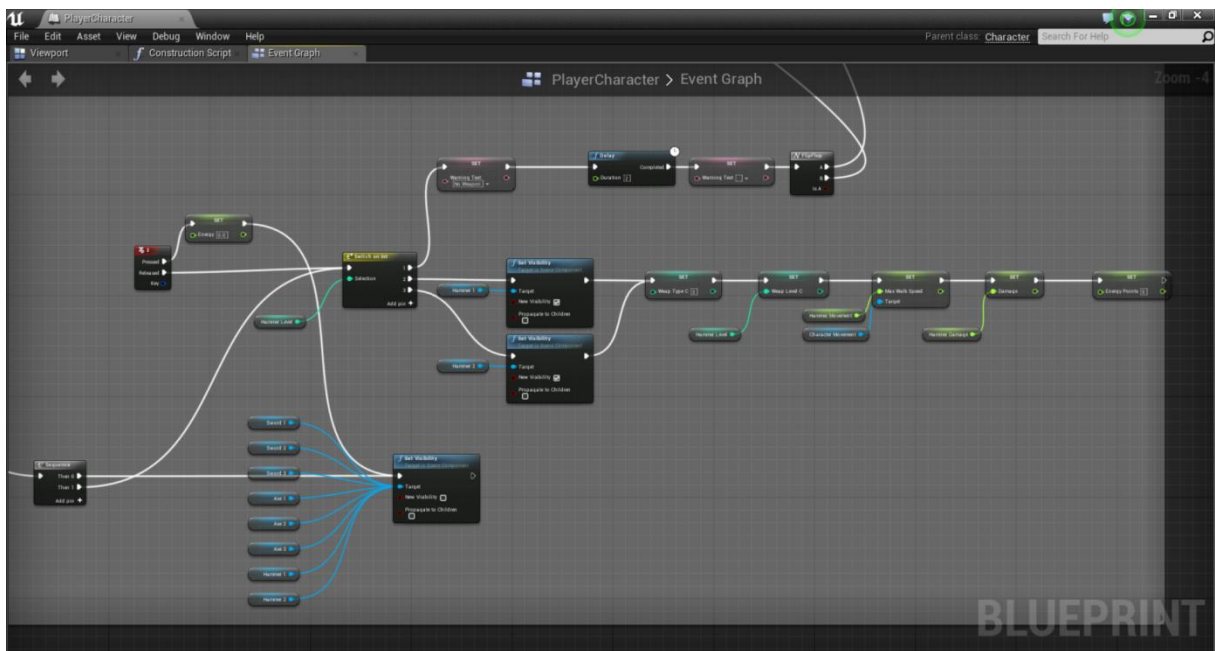
Εικόνα 82: Player Character Blueprint στην Unreal Engine 4



Εικόνα 83: Player Character Blueprint στην Unreal Engine 4



Εικόνα 84: Player Character Blueprint στην Unreal Engine 4

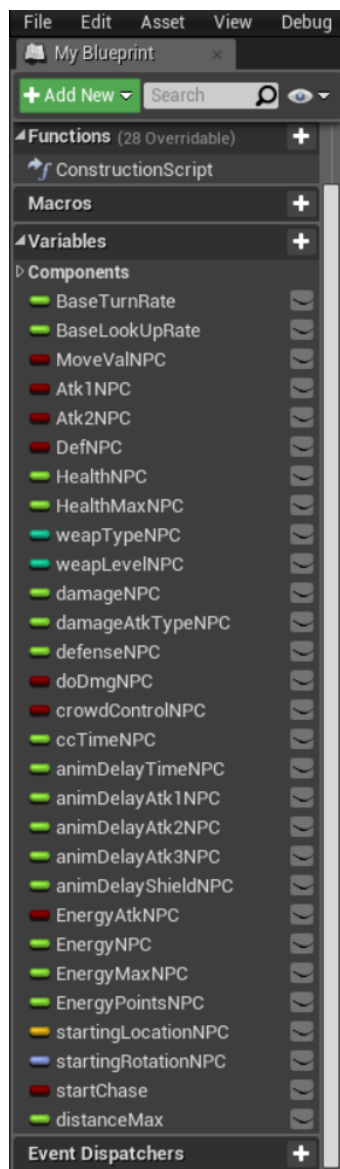


Εικόνα 85: Player Character Blueprint στην Unreal Engine 4

Στις Εικόνες 82, 83, 84, 85 βλέπουμε το Equipment System, το οποίο μας επιτρέπει να αλλάξουμε όπλα κατά τη διάρκεια του παιχνιδιού. Στην Εικόνα 82 βλέπουμε το Custom Event που χρησιμοποιεί το Level Blueprint, όταν δώσει στο χαρακτήρα ένα καινούριο όπλο, για να τον κάνει να το κρατήσει, θα δούμε περισσότερα για αυτή τη λειτουργία παρακάτω. Οι Εικόνες 83, 84, 85 έχουν τρία διαγράμματα τα οποία κάνουν το ίδιο πράγμα αλλά για διαφορετικά όπλα(83 για το Σπαθί, 84 για το Τσεκούρι και 85 για το Σφυρί). Τα διαγράμματα αυτά καλούνται όταν πατάμε τα κουμπιά 1, 2, 3 ή όταν το Level Blueprint καλέσει το Custom Event που αναφέραμε παραπάνω. Αυτά τα διαγράμματα χρησιμοποιούν τα όπλα που βάλουμε στο χαρακτήρα μας ως Components σε προηγούμενο βήμα. Κάθε φορά που ένα από αυτά τα διαγράμματα καλείται, όλα τα όπλα γίνονται αόρατα και μετά το σωστό όπλο εμφανίζεται

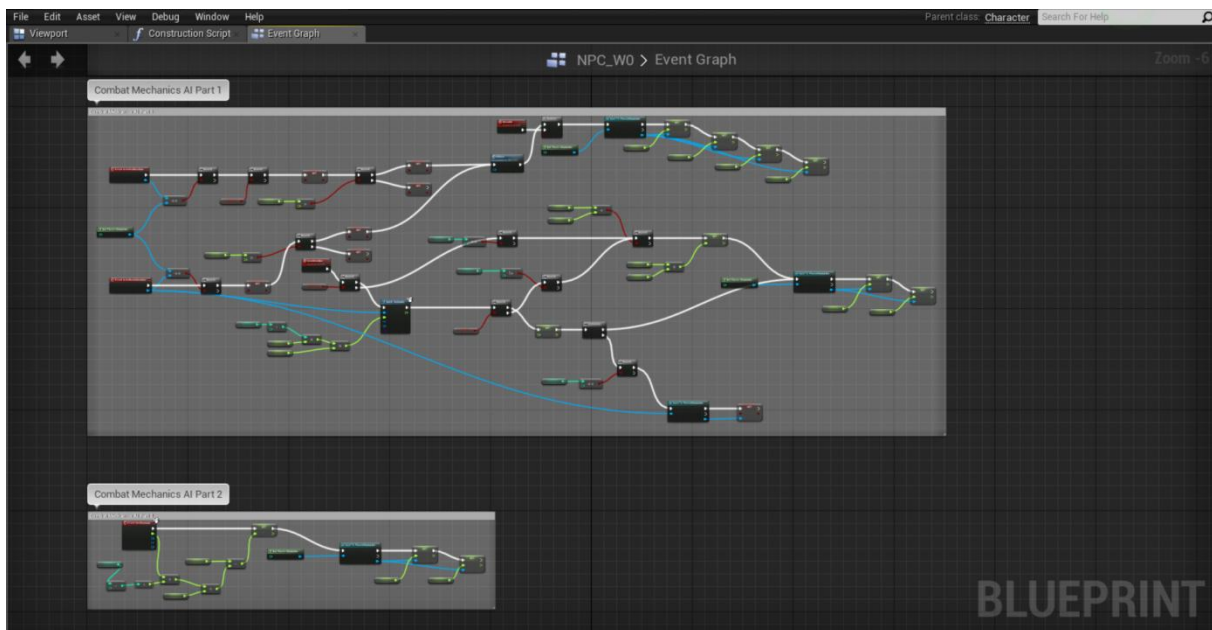
στο χέρι του χαρακτήρα, έπειτα όλες οι σχετικές μεταβλητές παίρνουν τις αντίστοιχες τιμές τους για αυτό το όπλο, επίσης ως πέναλτι για την αλλαγή ο χαρακτήρας χάνει την Ενέργεια που έχει μαζέψει(Σημείωση: λόγω ότι το Σφυρί δεν έχει πρώτο επίπεδο όπως τα άλλα δύο όπλα, πατώντας το κουμπί 3 πριν πάρουμε το Σφυρί, ο χαρακτήρας εξοπλίζεται με ένα από τα δύο άλλα όπλα και εκτυπώνεται κατάλληλο μήνυμα).

Στη συνέχεια θα αναλύσουμε το blueprint του NPC, αυτό το blueprint είναι ένα αντίγραφο του blueprint του χαρακτήρα του παίκτη αλλά με διαφορετικά διαγράμματα και μεταβλητές. Πρώτα, όμως, πρέπει να προσθέσουμε ένα Pawn Sensing Component, στο Components Tab → επιλέγουμε το NPC(self) → πατάμε Add Component → με την αναζήτηση βρίσκουμε το Pawn Sensing και το επιλέγουμε. Αυτό δίνει την ικανότητα στον αντίπαλο να αντιλαμβάνεται τι συμβαίνει γύρω του. Έπειτα πάμε στον χάρτη μας και προσθέτουμε ένα Nav Mesh Bounds Volume πηγαίνοντας Modes Tab → Volumes → Nav Mesh Bounds Volume, αυτό χρησιμεύει στο να φτιαχτούν μονοπάτια που το ΑΙ μπορεί να χρησιμοποιήσει έτσι ώστε να κάνει τους αντιπάλους να ακολουθήσουν τον παίκτη(Σημείωση: Το Nav Mesh Bounds Volume πρέπει να καλύπτει όλο το επίπεδο, οπότε πρέπει να προσαρμόσουμε τις διαστάσεις του ανάλογα το επίπεδό μας).



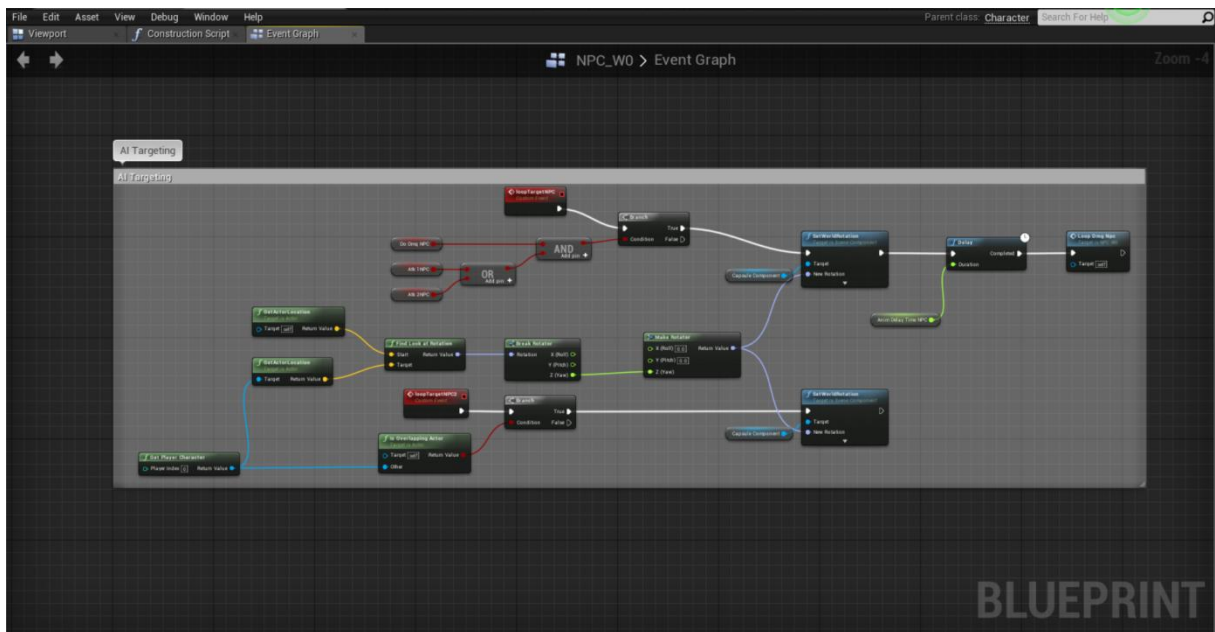
Εικόνα 86: Μεταβλητές NPC στην Unreal Engine 4

Το διάγραμμα Basic AI (Εικόνα 88) είναι το κομμάτι του αλγορίθμου του NPC που αναλαμβάνει όλες τις αποφάσεις του NPC, εκτός από τη μάχη. Αν ο παίκτης έχασε καλείται το διάγραμμα ResetAI, το οποίο επαναφέρει τον NPC στην αρχική του κατάσταση και θα το δούμε παρακάτω, ενώ αν ο παίκτης είναι ακόμα ενεργός τότε η διαδικασία συνεχίζεται. Στον επόμενο έλεγχο αν ο NPC έχει νικηθεί ξεκινά η διαδικασία καταστροφής του μοντέλου του, που θα τον διαγράψει από το παιχνίδι. Αν ο NPC δεν έχει νικηθεί, τότε πάμε στον επόμενο έλεγχο, βλέπει αν ο παίκτης τον έχει ακινητοποιήσει, με τη χρήση της ειδικής επίθεσης του Σφυριού, αν ναι τότε αλλάζει τις τιμές στις σχετικές μεταβλητές. Στη συνέχεια γίνεται έλεγχος αν ο παίκτης είναι σε απόσταση επίθεσης, αν ναι καλείται το διάγραμμα CombatAI που ασχολείται με τη μάχη και θα το δούμε παρακάτω, αλλιώς γίνεται έλεγχος για την απόσταση του παίκτη από την αρχική θέση του NPC. Αν ο παίκτης είναι πολύ μακριά, τότε καλείται το ResetAI, αλλιώς ο NPC προσπαθεί να φτάσει στη θέση του παίκτη, όταν φτάσει στην τοποθεσία, η διαδικασία ξεκινά από την αρχή, το οποίο έχει ως αποτέλεσμα ο NPC να ακολουθεί τον παίκτη μέχρι η τοποθεσία του να είναι εκτός ορίων ή να ξεκινήσει μάχη.



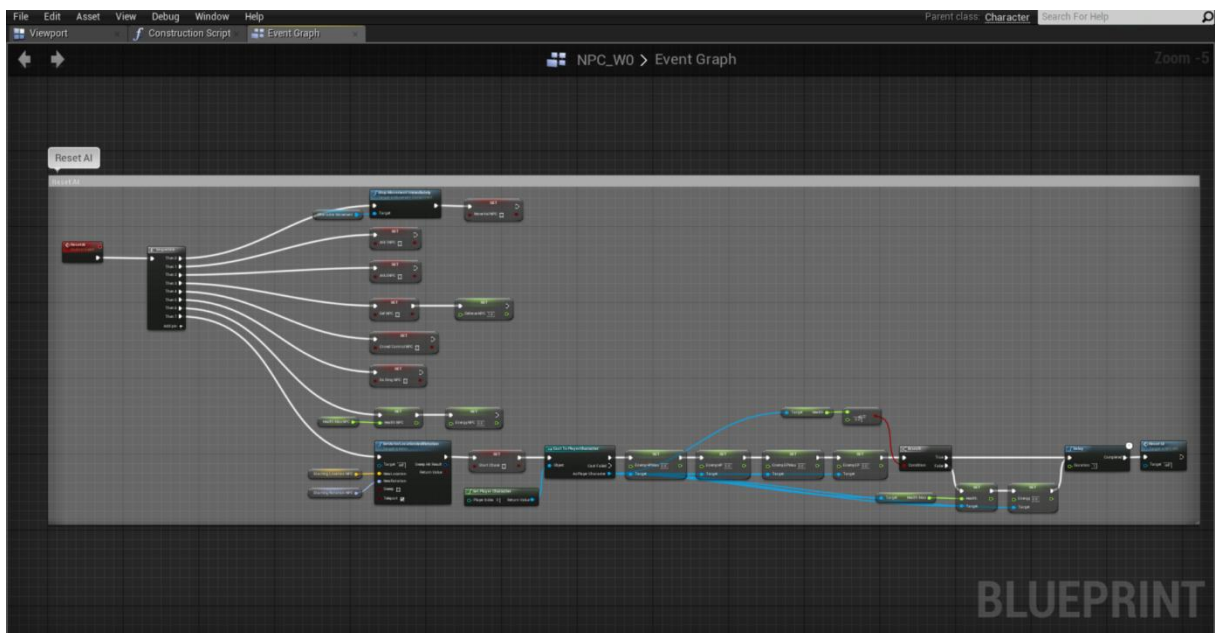
Εικόνα 89: Non-Player Character Blueprint στην Unreal Engine 4

Τα διαγράμματα Combat Mechanics (Εικόνα 89) είναι τροποποιημένες εκδόσεις των διαγραμμάτων Damage Dealer και Damage Receiver του παίκτη. Σε σύγκριση με το Damage Receiver το Part 2 δεν έχει το μηχανισμό ακινητοποίησης, ο οποίος είναι μέρος του Basic AI για πιο ομαλή διαχείριση. Σε σύγκριση με το Damage Dealer το Part 1 απλά ανανεώνει κάποιες μεταβλητές του παίκτη που χρησιμεύουν στο UI και την προβολή πληροφοριών στον παίκτη για την κατάσταση του NPC όπως Ζωή και Ενέργεια, ενώ το διάγραμμα στόχευσης είναι ξεχωριστό.



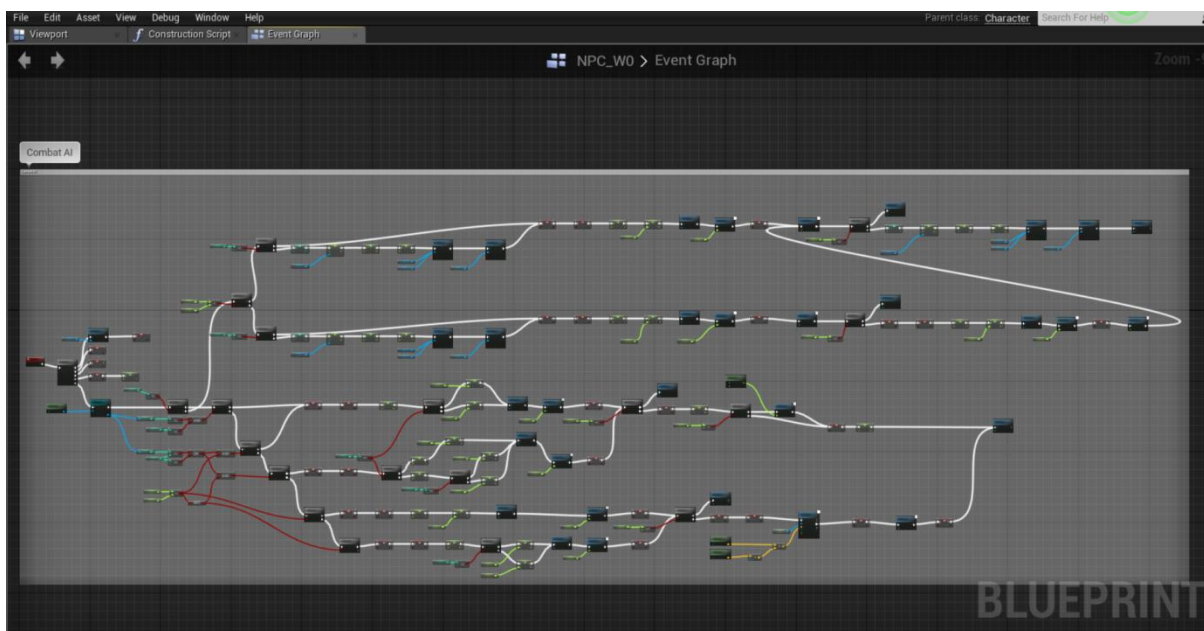
Εικόνα 90: Non-Player Character Blueprint στην Unreal Engine 4

Το AI Targeting (Εικόνα 90) είναι το ίδιο με το διάγραμμα στόχευσης του Damage Dealer, απλά είναι διαχωρισμένο από το Combat Mechanics για ευκολία, απλοποίηση του διαγράμματος και επίσης γιατί είναι πιο εύκολο να στοχοποιηθεί ο παίκτης σε σύγκριση με τους NPCs, οπότε η χρήση των αντίστοιχων δεδομένων είναι πιο εύκολη στο NPC Blueprint από ό,τι στο Blueprint του χαρακτήρα του παίκτη.



Εικόνα 91: Non-Player Character Blueprint στην Unreal Engine 4

Το διάγραμμα ResetAI (Εικόνα 91) είναι απλό, η δουλειά του είναι να δίνει στις μεταβλητές του NPC τις αρχικές τους τιμές και να τον τοποθετεί στην αρχική του θέση. Η χρήση είναι να καλείται, όπως είδαμε παραπάνω, από άλλα διαγράμματα σε άλλα σημεία του blueprint του NPC, όταν ο παίκτης χάσει ή απομακρυνθεί πολύ από την αρχική θέση του NPC.



Εικόνα 92: Non-Player Character Blueprint στην Unreal Engine 4

Το Combat AI (Εικόνα 92) είναι το κομμάτι του αλγορίθμου του NPC που αναλαμβάνει όλες τις αποφάσεις του NPC κατά τη διάρκεια της μάχης. Αυτό το διάγραμμα καλείται όταν ο NPC είναι σε απόσταση επίθεσης από τον παίκτη. Αν ο NPC είναι ο τελικός αντίπαλος, τότε ο αλγόριθμος είναι να ακολουθεί τον παίκτη χρησιμοποιώντας το Σπαθί για την ταχύτητά του, ενώ όταν είναι σε απόσταση επίθεσης του επιτίθεται με το Τσεκούρι, αν υπάρχει αρκετή Ενέργεια τότε επιτίθεται με την ειδική επίθεση του Σφυριού, για να ακινητοποιήσει τον παίκτη, και μετά κάνει μία κανονική επίθεση με το Σφυρί, αυτή η τακτική συνεχίζεται μέχρι να χάσει ο παίκτης ή να νικηθεί ο NPC. Σε οποιαδήποτε άλλη περίπτωση, επειδή ο NPC δε μπορεί να αλλάξει όπλο ακολουθεί μια τακτική βασισμένη στο συνδυασμό των όπλων που χρησιμοποιούνται στη μάχη και από τον NPC και από τον παίκτη. Όταν ο NPC έχει όπλο πιο αργό ή με ίση ταχύτητα με του παίκτη τότε η τακτική του NPC είναι να επιτίθεται και να αμύνεται μέχρι να έχει αρκετή Ενέργεια για να χρησιμοποιήσει την ειδική του επίθεση, εξαιρεση στον κανόνα είναι όταν ο NPC έχει το Τσεκούρι και ο παίκτης έχει το Σπαθί, ο NPC δεν χρησιμοποιεί την ειδική επίθεση σε αυτήν την περίπτωση. Όταν ο NPC έχει πιο γρήγορο όπλο από τον παίκτη τότε η τακτική είναι να επιτίθεται και να φεύγει από τον παίκτη για να αποφύγει την ζημιά, όταν έχει αρκετή Ενέργεια προσπαθεί να χρησιμοποιήσει την ειδική του επίθεση.

Το Blueprint του NPC χρησιμοποιείται ως Parent Blueprint για τα blueprints των αντιπάλων, αυτό γίνεται κάνοντας δεξί κλικ στο Blueprint και επιλέγοντας Create Child Blueprint Class, έτσι δημιουργούμε τα blueprints των αντιπάλων, τα οποία μοιράζονται τον κώδικα που φτιάξαμε, τώρα πρέπει απλώς να βάλουμε την αρχικοποίηση των μεταβλητών τους για να ολοκληρωθούν και να έχουν τις λειτουργίες που θέλουμε.

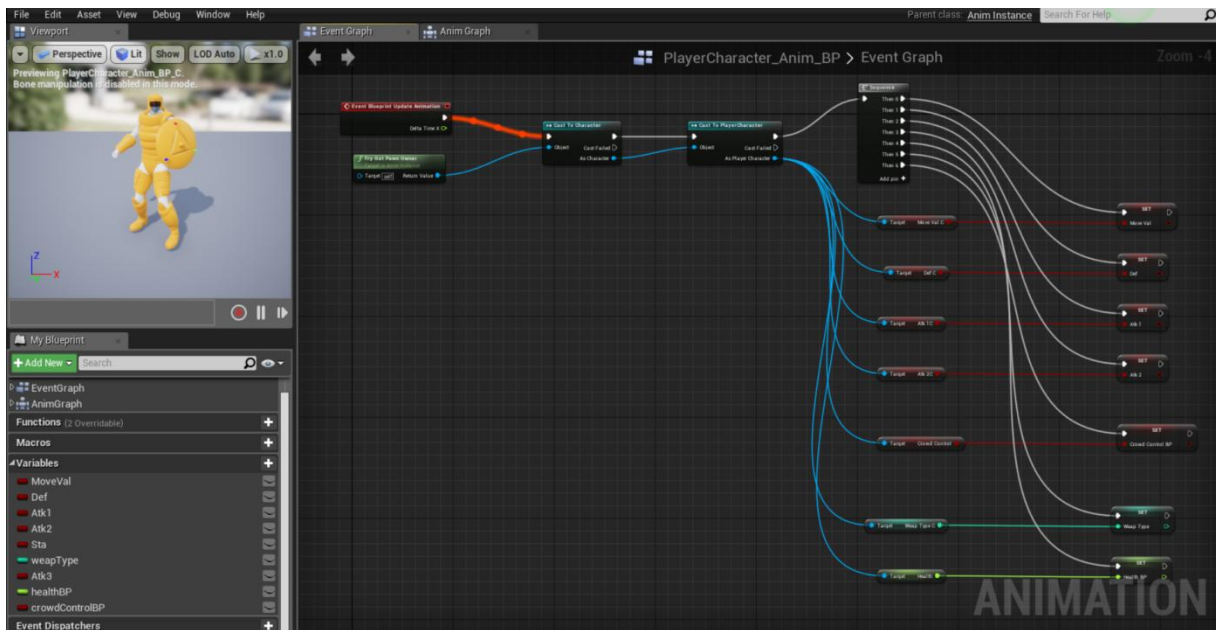


Εικόνα 93: Non-Player Character Child Blueprints στην Unreal Engine 4

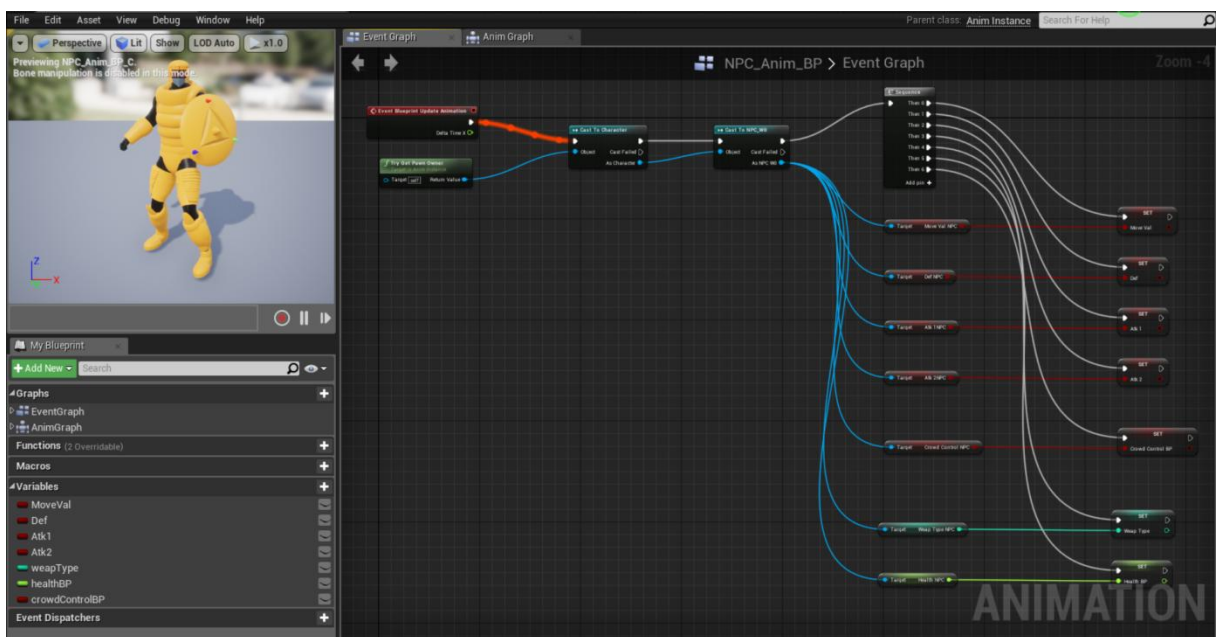
Στην Εικόνα 93 βλέπουμε τα επτά διαγράμματα των blueprints των αντιπάλων, όπως βλέπουμε είναι ακριβώς τα ίδια με τη μόνη διαφορά να είναι οι τιμές που αρχικοποιούν στις μεταβλητές. Λόγω των διαφορετικών τιμών ο κάθε NPC αντίπαλος λειτουργεί διαφορετικά από τους υπόλοιπους χρησιμοποιώντας τον ίδιο αλγόριθμο.

3.3.2.2. Animation Blueprints

Σε αυτό το βήμα θα δούμε πως συνδέουμε τα animations, που έχουμε φτιάξει για κάθε χαρακτήρα, με το χαρακτήρα και τις δραστηριότητές του στο παιχνίδι. Γι' αυτό χρειαζόμαστε δύο Animation Blueprints, ένα για το χαρακτήρα του παίκτη και ένα για τον NPC που θα το κληρονομήσει στα παιδιά του, που δημιουργήσαμε και είναι οι αντίπαλοί μας στο παιχνίδι.

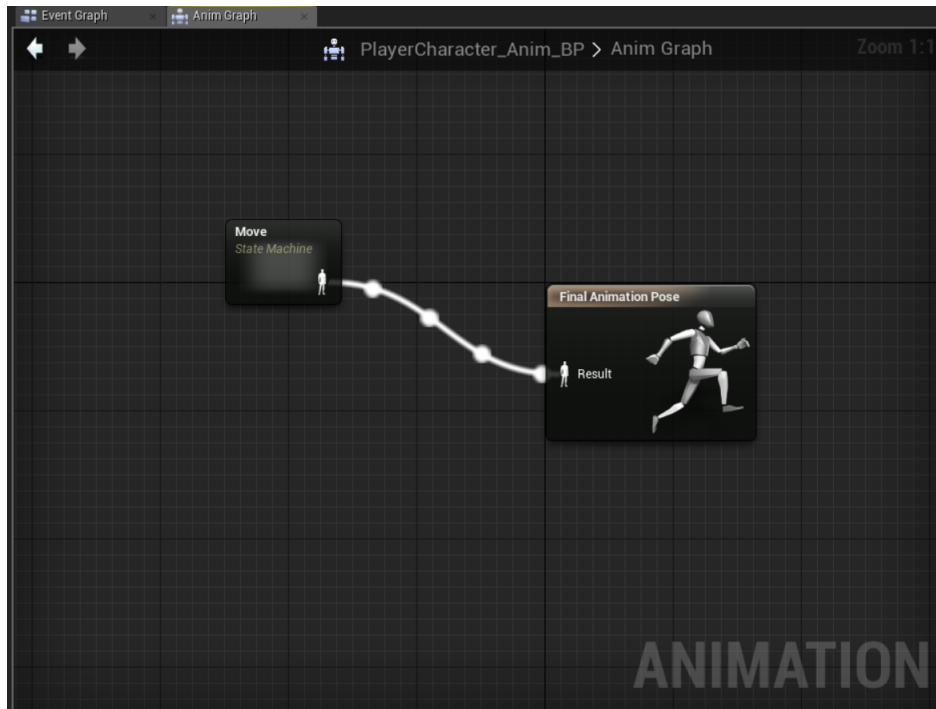


Εικόνα 94: Player Character Animation Blueprint στην Unreal Engine 4



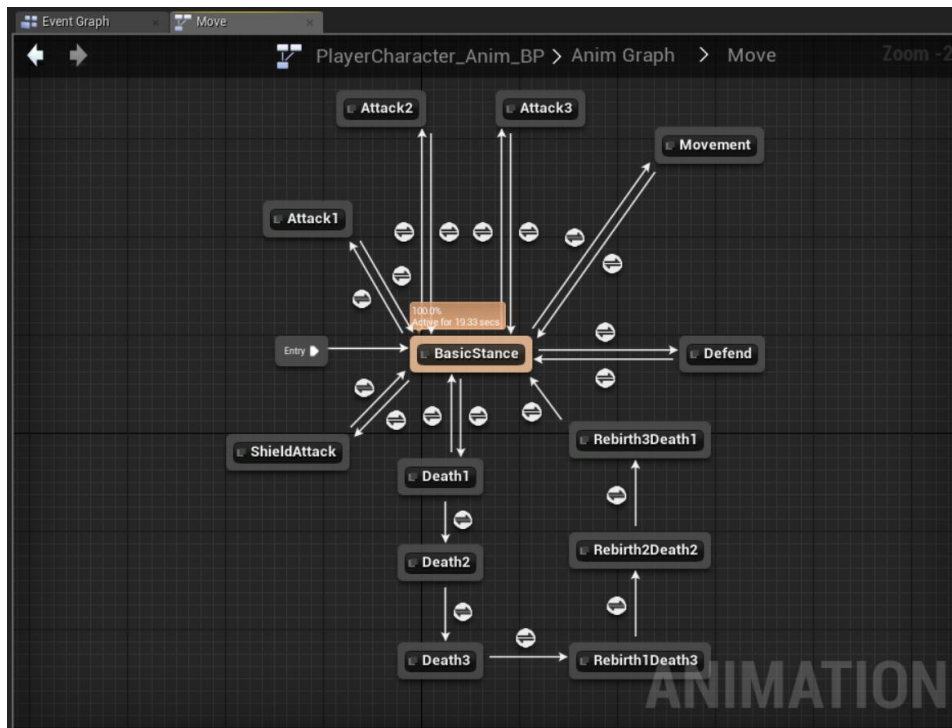
Εικόνα 95: Non-Player Character Animation Blueprint στην Unreal Engine 4

Στις Εικόνες 94, 95 βλέπουμε τις μεταβλητές και τα Event Graphs των Animation Blueprints του χαρακτήρα του παίκτη και του NPC. Όπως βλέπουμε τα διαγράμματα των Event Graphs είναι ίδια, γιατί αυτό που κάνουν είναι να παίρνουν τις μεταβλητές του χαρακτήρα του παίκτη και του NPC και να τις περνούν στις αντίστοιχες μεταβλητές των Animation Blueprints. Αυτές οι μεταβλητές θα χρησιμοποιηθούν παρακάτω για τις συνθήκες που θα μας επιτρέπουν να χρησιμοποιήσουμε τα animations όταν τα χρειαζόμαστε.

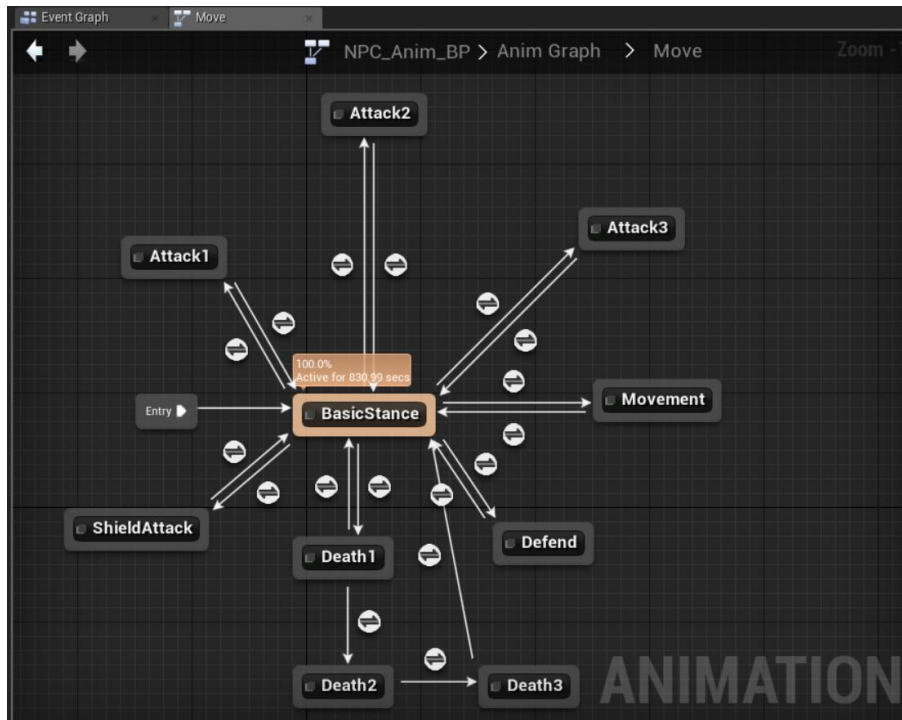


Εικόνα 96: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 96 βλέπουμε το Anim Graph του χαρακτήρα του παίκτη, του NPC είναι ακριβώς το ίδιο, το οποίο παράγει το τελικό animation που θα χρησιμοποιήσει ο χαρακτήρας. Όπως βλέπουμε έχουμε το State Machine με το όνομα Move ως είσοδο στο Final Animation Pose, το State Machine Move υπολογίζει ποιο animation θέλουμε να παίξει και το Final Animation Pose παίρνει αυτό το animation από το State Machine Move και βάζει το σκελετό, και κατ' επέκταση το μοντέλο του χαρακτήρα, να εκτελέσει αυτό το animation, έτσι εκτελεί τα animations ο κάθε χαρακτήρας.

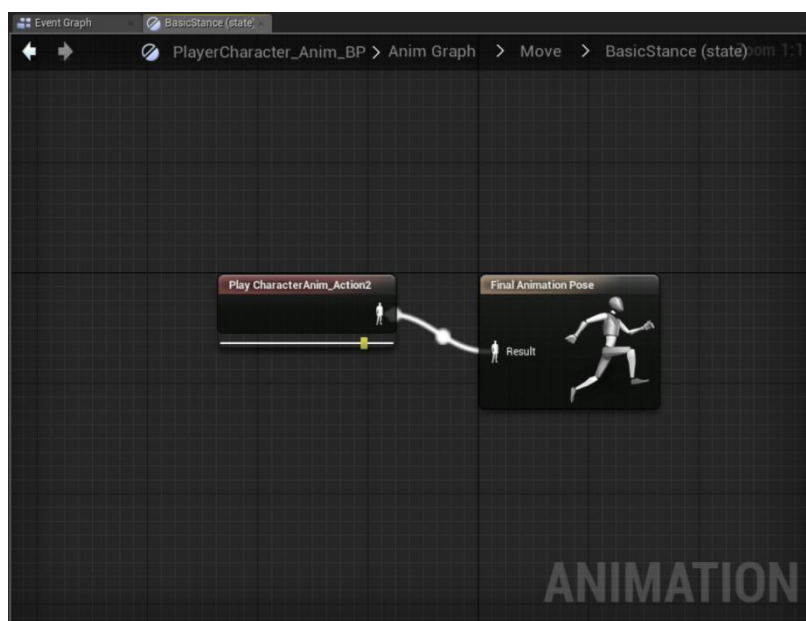


Εικόνα 97: Player Character Animation Blueprint στην Unreal Engine 4



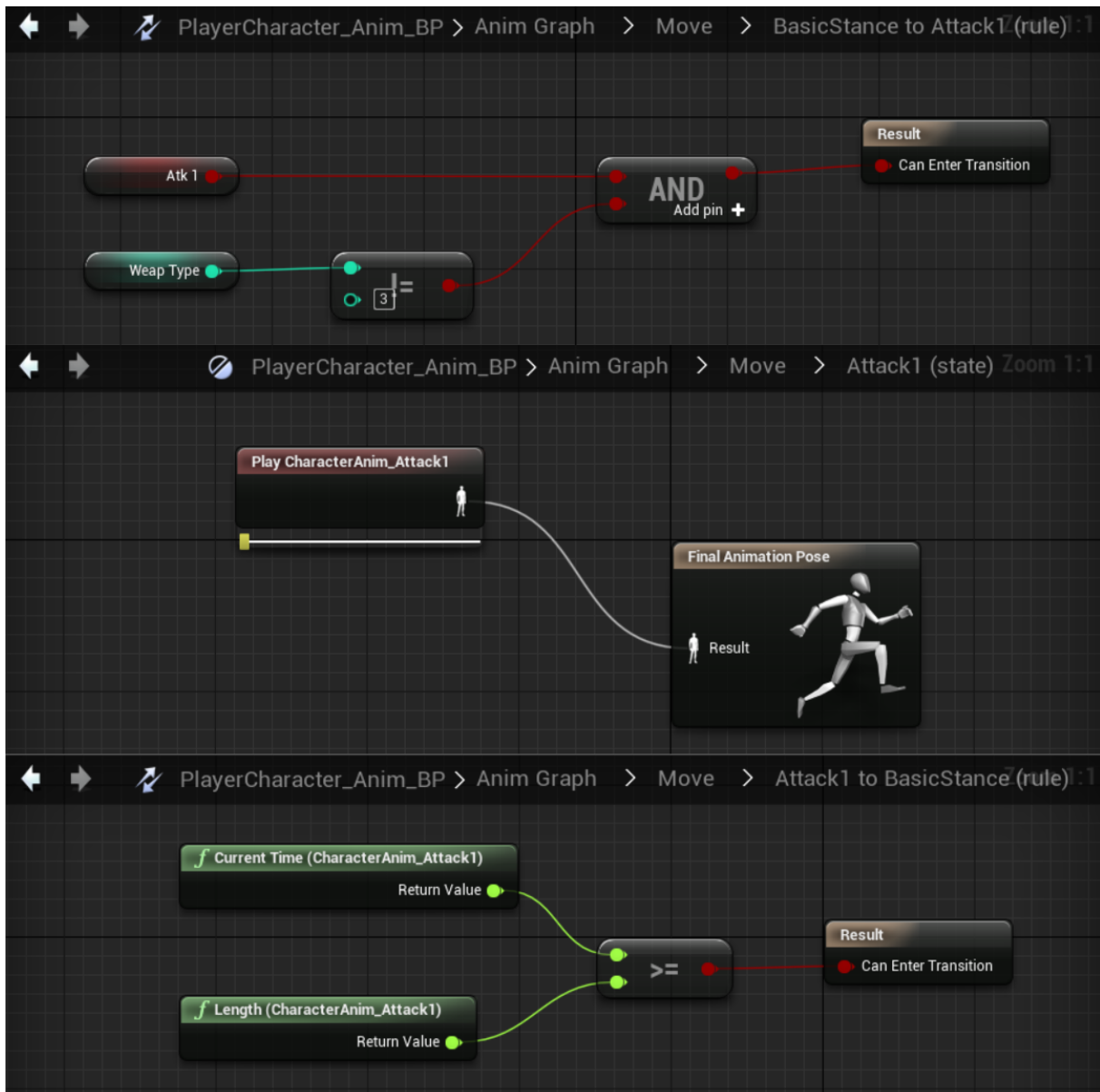
Εικόνα 98: Non-Player Character Animation Blueprint στην Unreal Engine 4

Στις Εικόνες 97, 98 βλέπουμε τα State Machines του χαρακτήρα του παίκτη και του NPC, αντίστοιχα, αυτά τα διαγράμματα χρησιμεύουν στον υπολογισμό του κατάλληλου animation για κάθε χαρακτήρα, ανάλογα με τις τιμές των αντίστοιχων μεταβλητών, που είδαμε παραπάνω, οι οποίες τροποποιούνται από τον παίκτη και από το AI του NPC. Το διάγραμμα ενός State Machine αποτελείται από δύο βασικά στοιχεία τα States(Καταστάσεις), που περιγράφουν την εκτέλεση των animations, και τα Transition Rules(Κανόνες Μετάβασης), που περιγράφουν τις προϋποθέσεις που χρειάζονται έτσι ώστε να μεταβούμε από το ένα State στο άλλο, δηλαδή από animation σε animation. Όπως μπορούμε να δούμε η αρχή του κάθε διαγράμματος είναι το Entry Point, το οποίο χρησιμεύει στην ενεργοποίηση του διαγράμματος όταν το παιχνίδι ξεκινά.



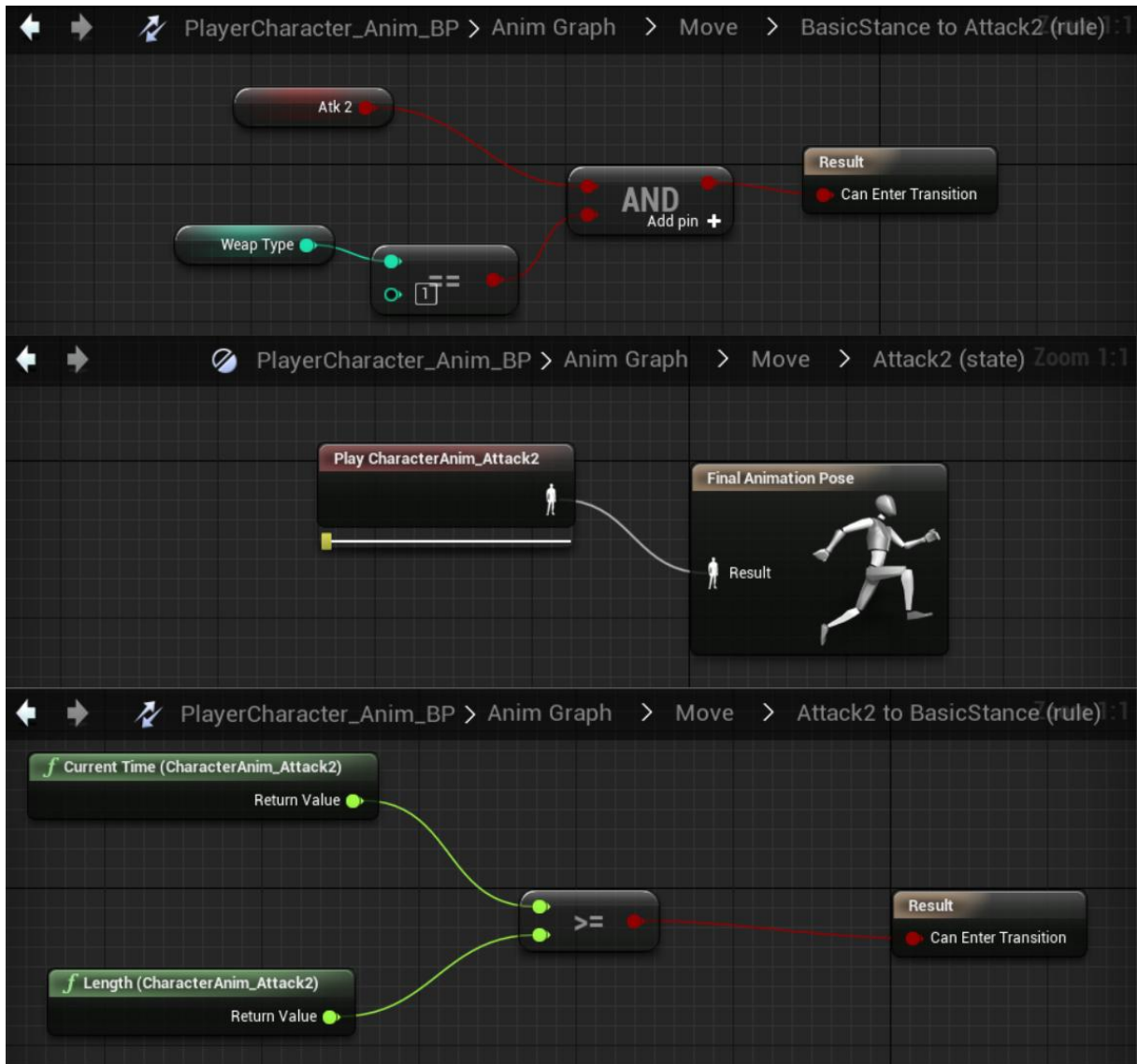
Εικόνα 99: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 99 βλέπουμε το BasicStance state το οποίο εκτελείται όταν ξεκινά το παιχνίδι και κάθε φορά που τελειώνει ένα άλλο animation και επιστρέφει το διάγραμμα σε αυτό το state. Με αυτόν τον τρόπο ο χαρακτήρας έχει τη στάση του σώματος που βλέπουμε στο παιχνίδι αντί του T-rose που έχει από την εισαγωγή του από το Blender.



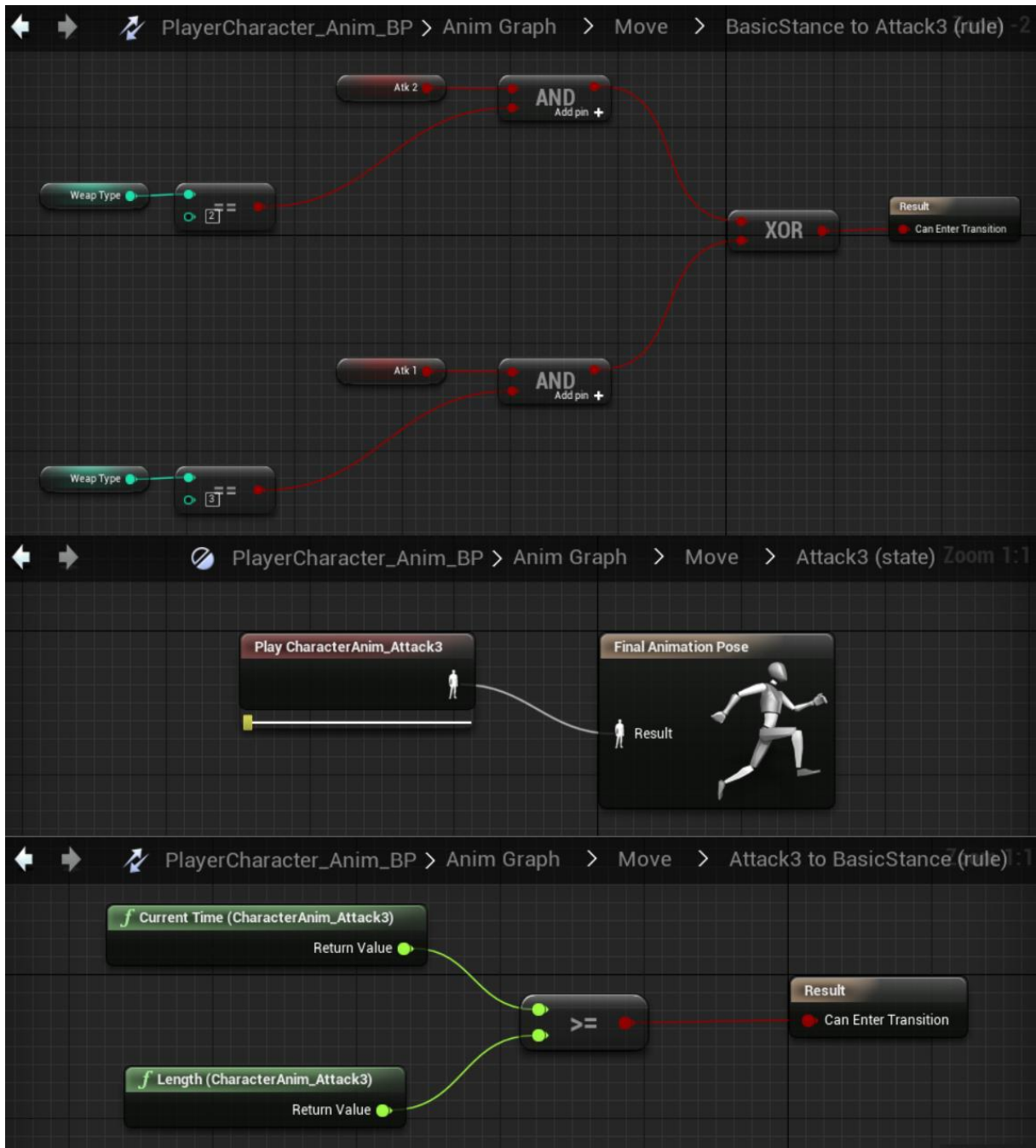
Εικόνα 100: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 100 βλέπουμε το Transition Rule από το Basic Stance στο Attack1, το Attack1 state και το Transition Rule από Attack1 το στο Basic Stance. Βλέπουμε ότι για να παίξει το animation πρέπει να επιλεγεί η απλή επίθεση και το όπλο να μην είναι το Σφυρί, αφού τελειώσει το animation, γίνεται έλεγχος στο χρόνο του και επιστρέφει πίσω.



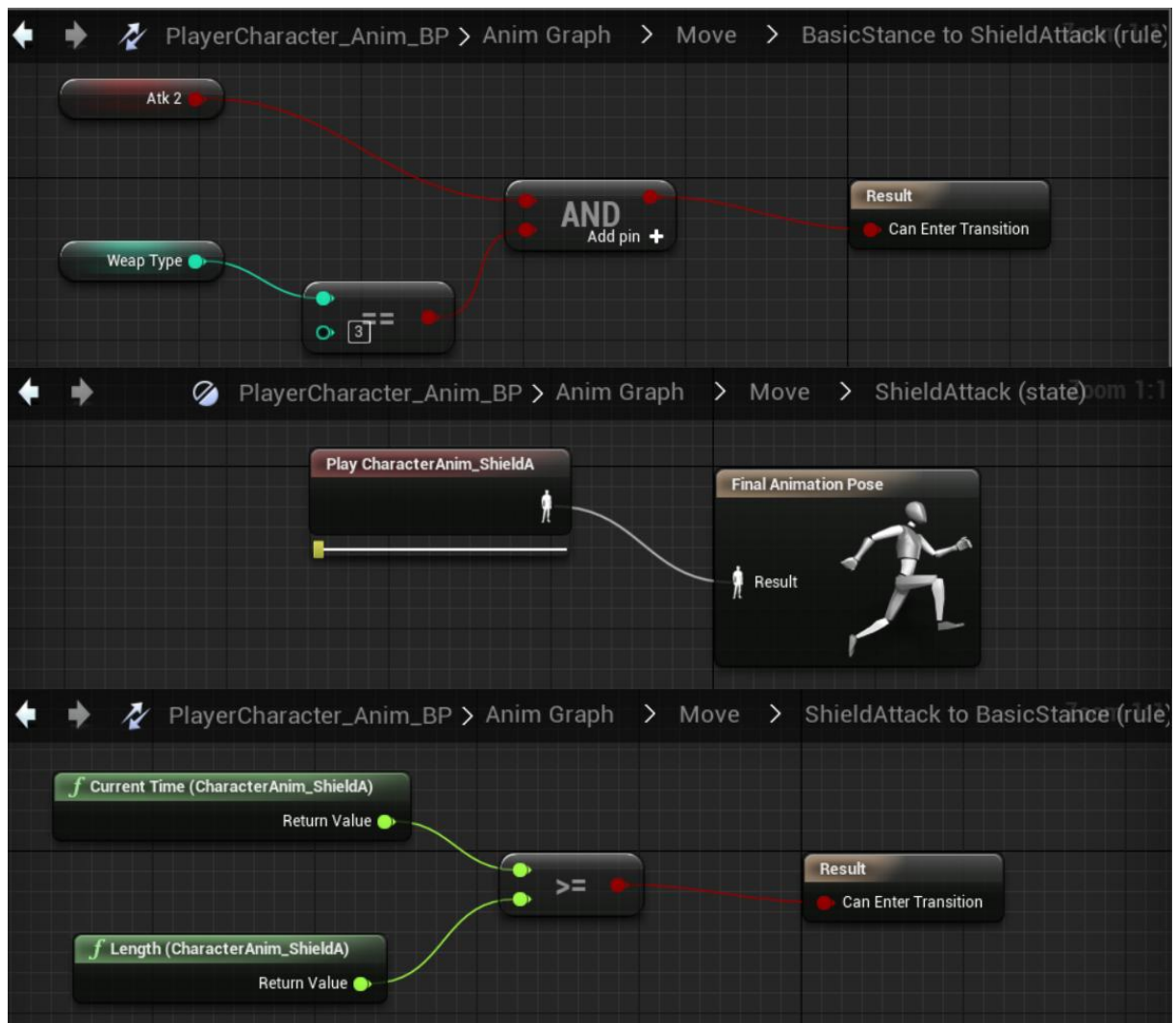
Εικόνα 101: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 101 βλέπουμε την ειδική επίθεση του Σπαθιού, όταν επιλεγθεί η ειδική επίθεση και το όπλο είναι το Σπαθί εκτελείται το animation, όταν ολοκληρωθεί επιστρέφει πίσω.



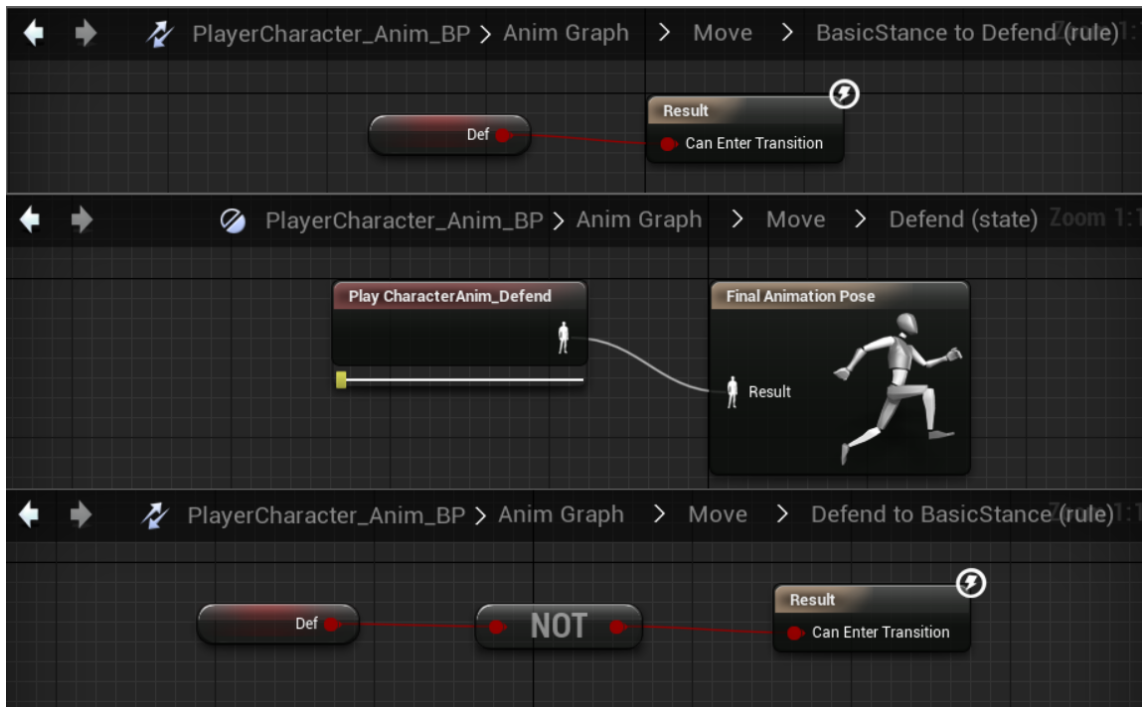
Εικόνα 102: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 102 βλέπουμε την ειδική επίθεση του Τσεκουριού και την απλή επίθεση του Σφυριού, αυτό το animation παίζει μόνο όταν επιλέξουμε την ειδική επίθεση και το όπλο είναι το Τσεκούρι ή όταν επιλέξουμε την απλή επίθεση και το όπλο είναι το Σφυρί, όταν τελειώσει επιστρέφει πίσω.



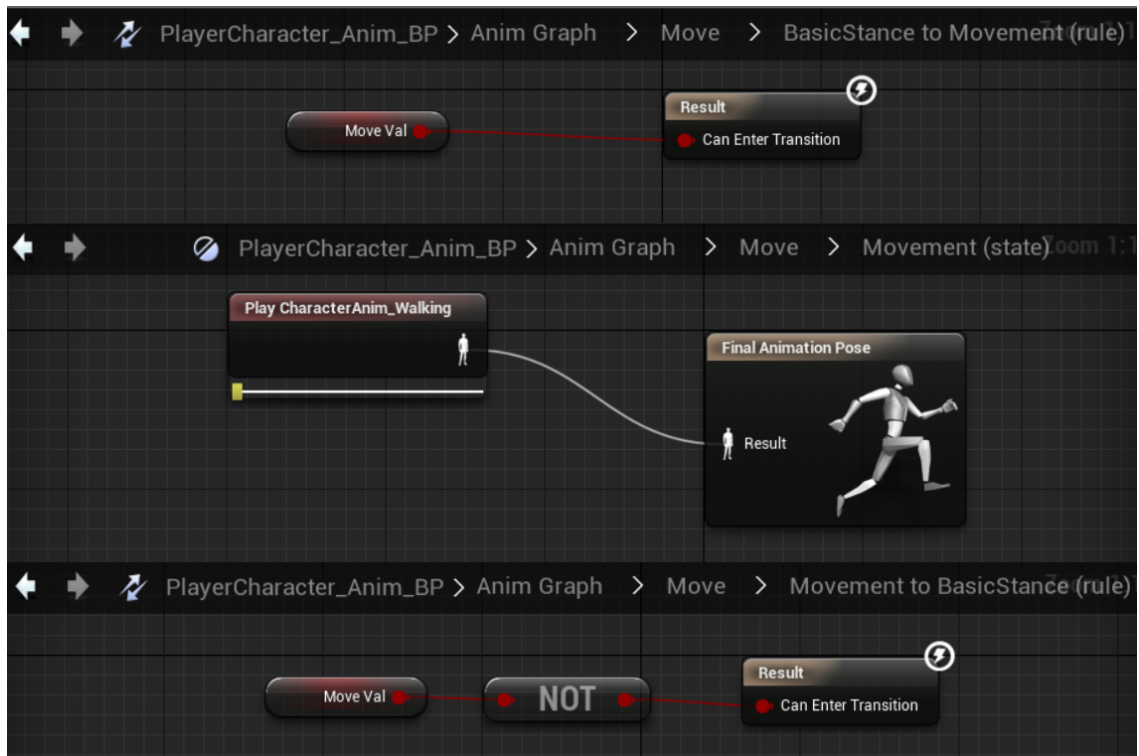
Εικόνα 103: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 103 βλέπουμε το animation της ειδικής επίθεσης του Σφυριού, όταν επιλέξουμε την ειδική επίθεση και το όπλο μας είναι το Σφυρί παίζει το animation και όταν ολοκληρωθεί επιστρέφει πίσω.



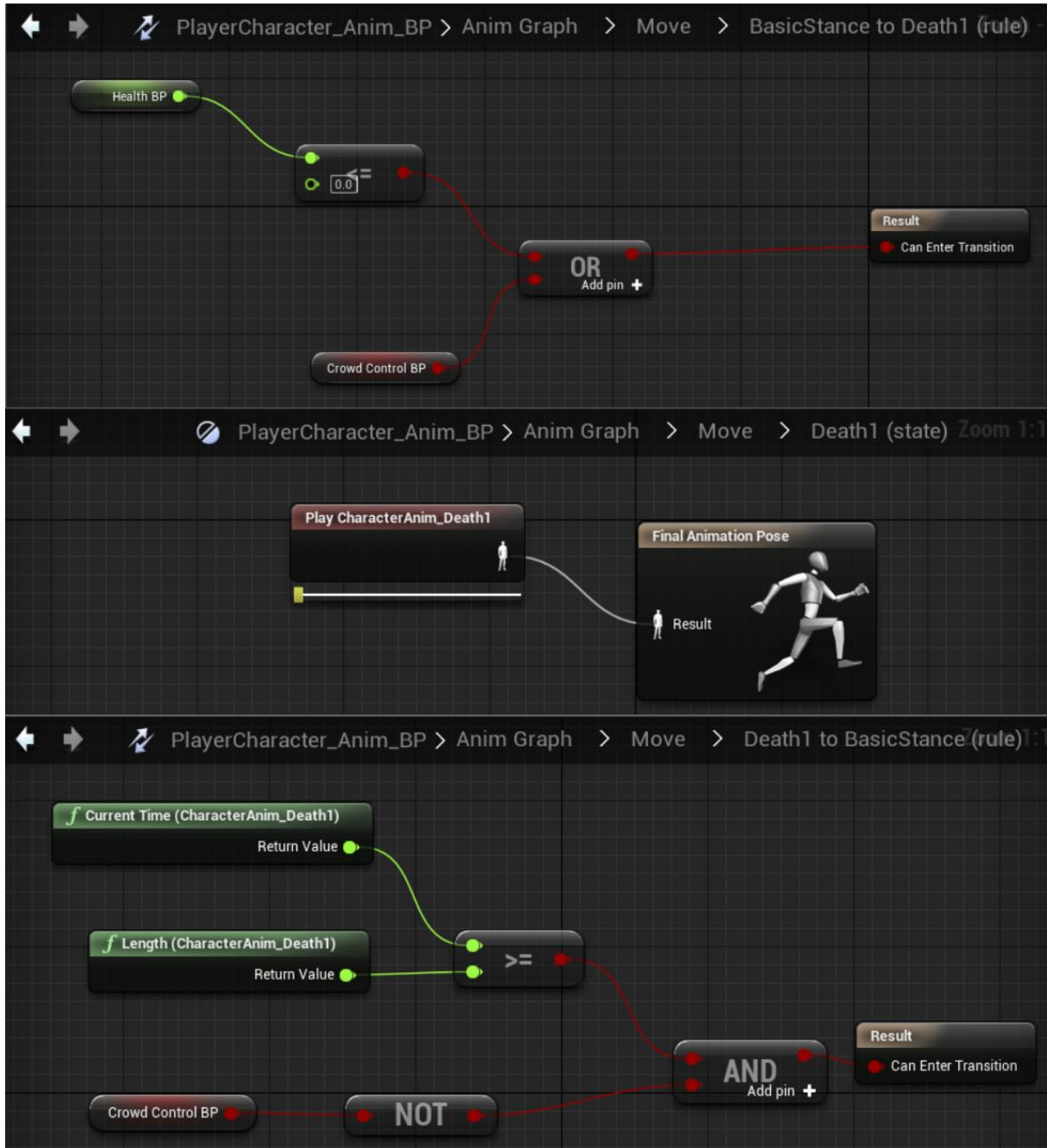
Εικόνα 104: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 104 βλέπουμε το animation της άμυνας, η μετάβαση γίνεται μόνο με βάση τη μεταβλητή, όσο η μεταβλητή είναι αληθής παίζει το animation της άμυνας, όταν γίνει ψευδής επιστρέφει πίσω (Σημείωση: Στο animation της άμυνας πρέπει να αλλάξουμε τη ρύθμιση Loop Animation και να την κάνουμε ψευδή, έτσι ώστε όταν παίζει το animation να μείνει στο τελευταίο frame με την ασπίδα ψηλά όσο κρατάμε την μεταβλητή αληθή, αλλιώς το animation θα επαναλαμβάνεται μέχρι να γίνει ψευδής η μεταβλητή).



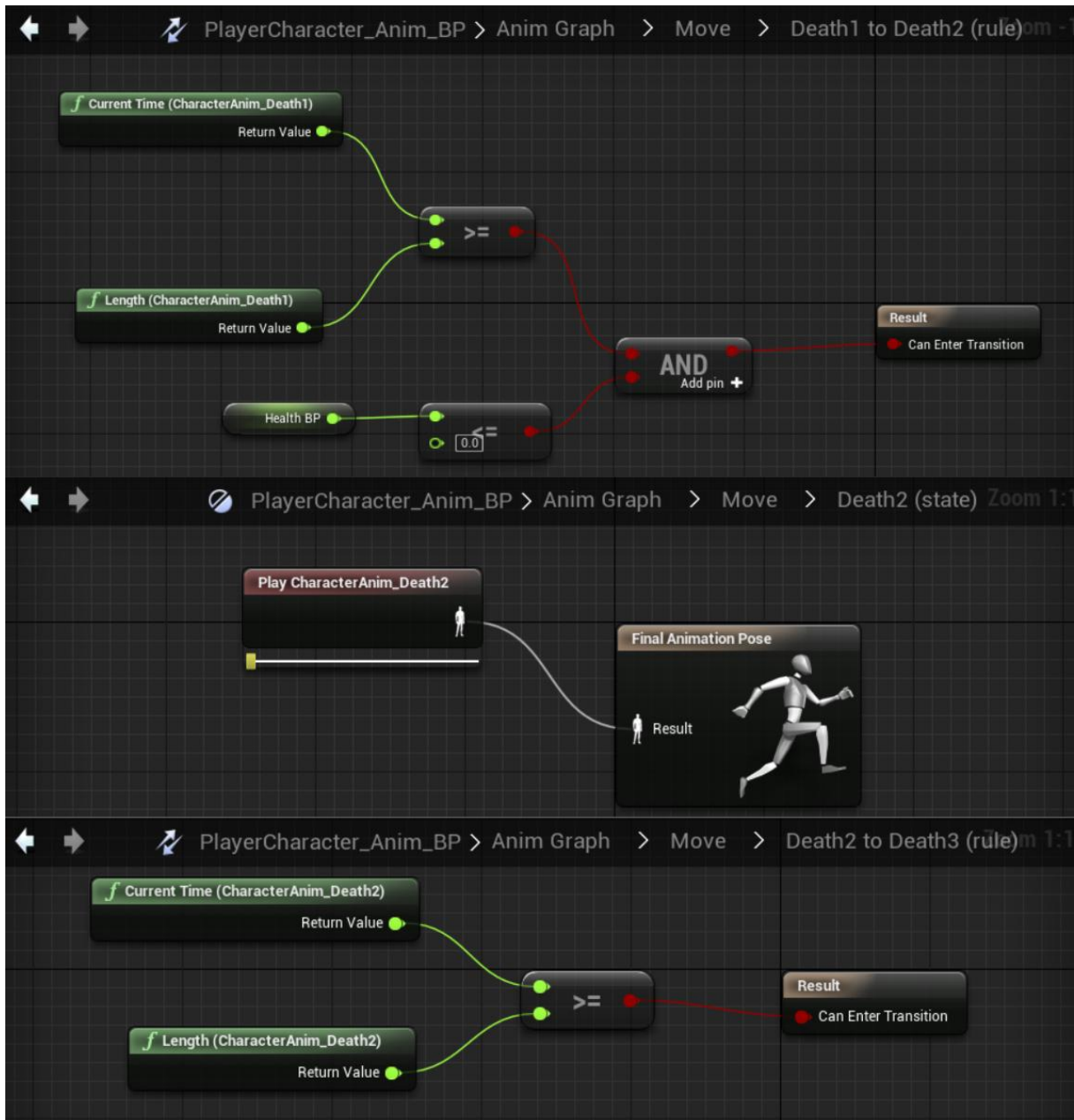
Εικόνα 105: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 105 βλέπουμε το animation της κίνησης του χαρακτήρα. Όπως βλέπουμε είναι η λειτουργία είναι ίδια με την άμυνα, όσο είναι αληθής η μεταβλητή παίζει το animation και όταν γίνεται ψευδής επιστρέφει πίσω, αλλά αντίθετα με την άμυνα θέλουμε το animation της κίνησης να έχει το Loop Animation αληθές, έτσι ώστε όσο παίζει να βλέπουμε το περπάτημα να επαναλαμβάνεται.



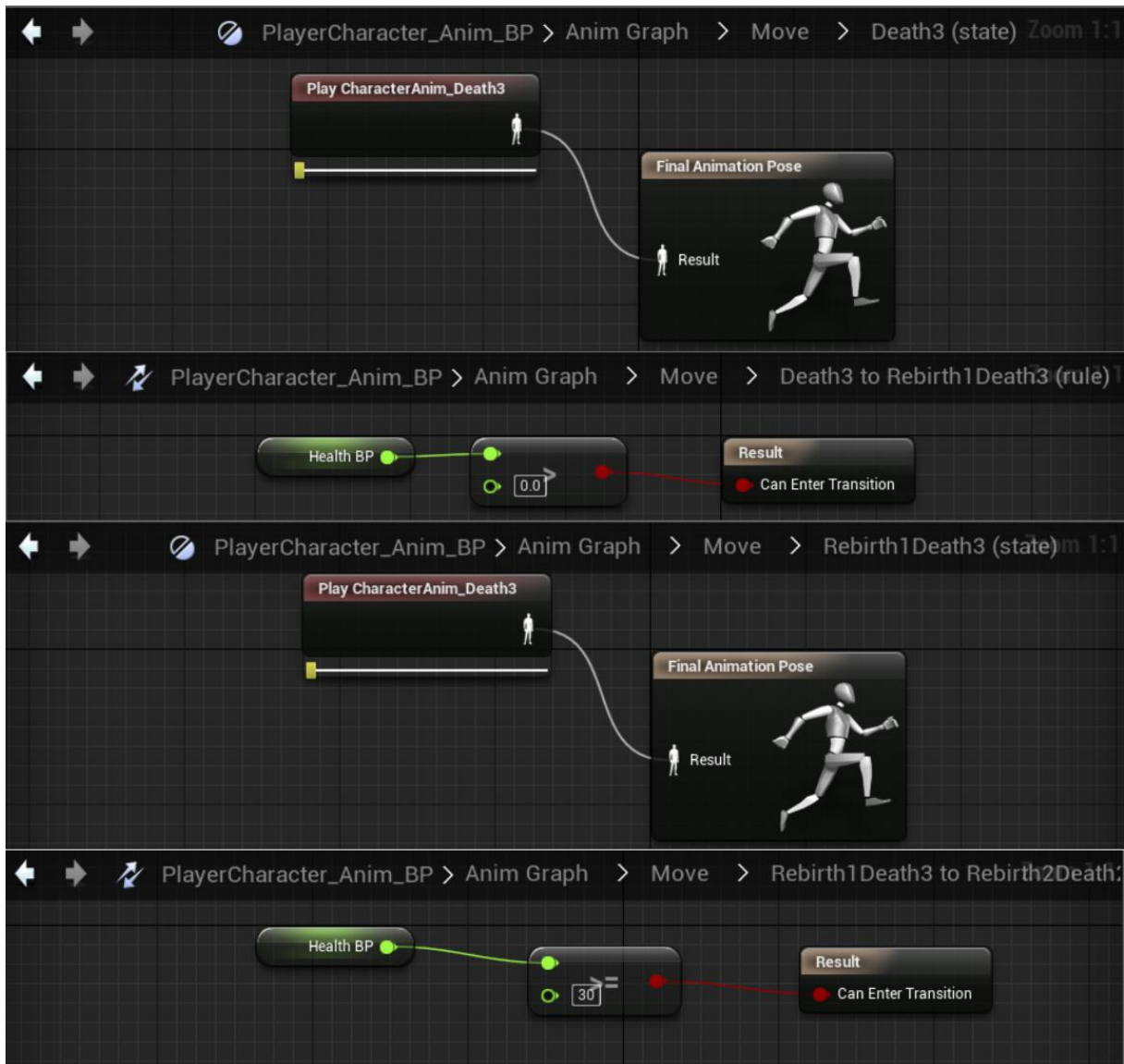
Εικόνα 106: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 106 βλέπουμε την αρχή του animation του θανάτου και το animation της ακινητοποίησης. Όταν η ζωή ενός χαρακτήρα είναι τελειώσει ή ο χαρακτήρας ακινητοποιηθεί παίζει το animation. Επιστροφή στην αρχική κατάσταση γίνεται μόνο όταν περάσει ο χρόνος ακινητοποίησης, αν ο χαρακτήρας νικήθηκε τότε πάει σε άλλη κατάσταση που θα δούμε παρακάτω.



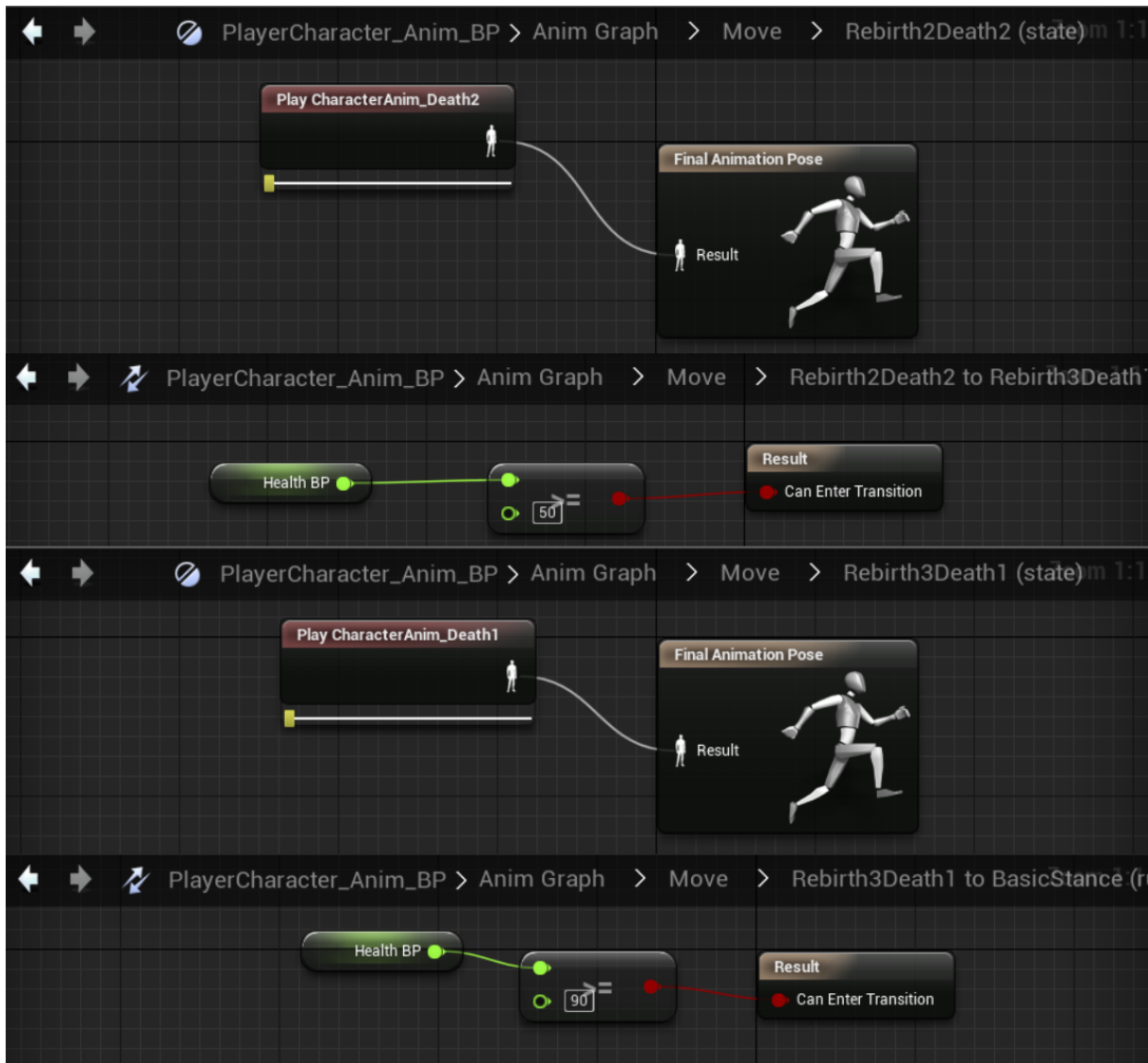
Εικόνα 107: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 107 βλέπουμε το επόμενο βήμα του animation του θανάτου. Το animation του θανάτου είναι σπασμένο σε τρία μέρη, για να είναι πιο εύκολο στη ρύθμιση, το πρώτο μέρος επίσης όπως είδαμε χρησιμοποιείται παραπάνω και ως το animation της ακινητοποίησης. Όταν τελειώσει το πρώτο μέρος και η ζωή του χαρακτήρα έχει τελειώσει τότε παίζει αυτό το animation και μετά πάει στο τρίτο και τελευταίο μέρος.



Εικόνα 108: Player Character Animation Blueprint στην Unreal Engine 4

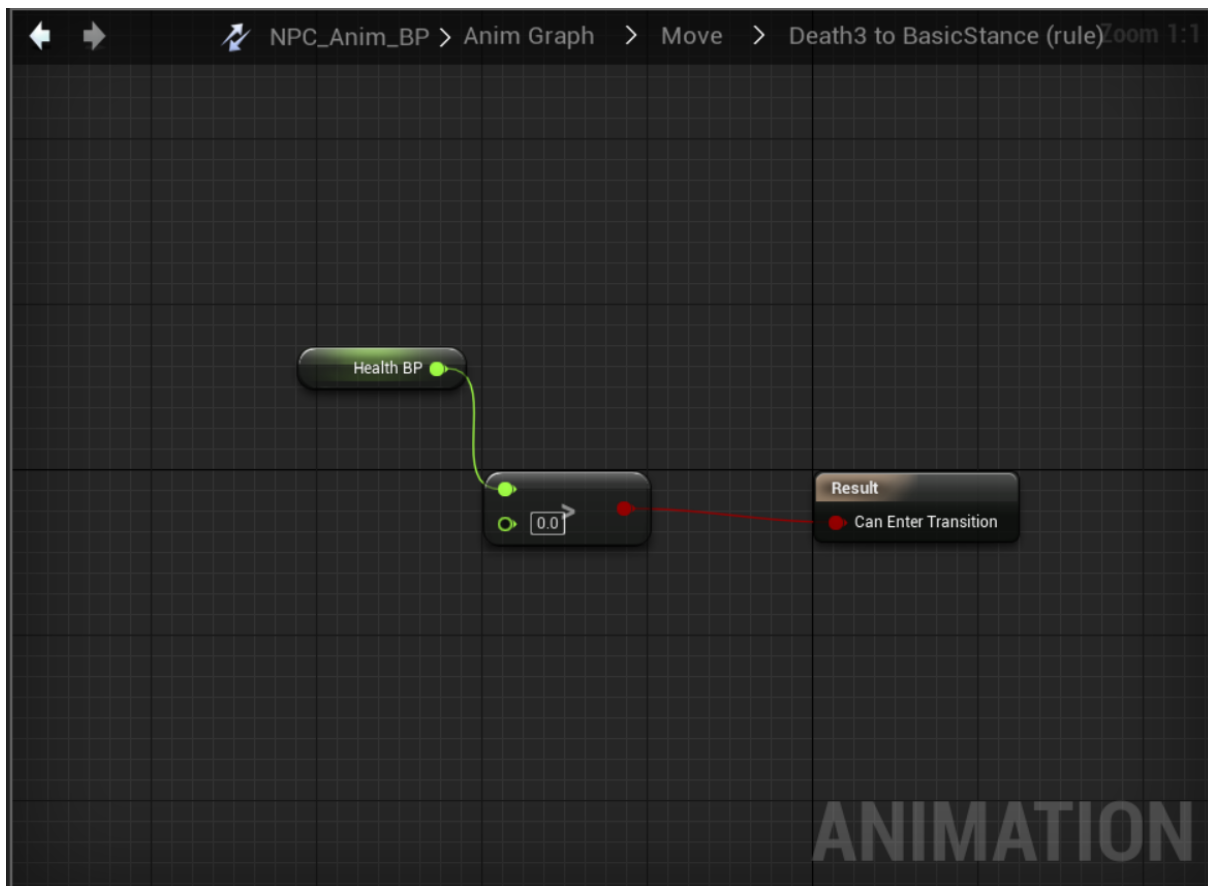
Στην Εικόνα 108 βλέπουμε το τελευταίο μέρος του animation του θανάτου και την αρχή του animation της επιστροφής του χαρακτήρα. Όταν ολοκληρωθεί το animation του θανάτου ο παίκτης έχει την επιλογή να ξαναπροσπαθήσει, όπως είδαμε παραπάνω. Όταν επιλέξει να ξαναπροσπαθήσει τότε η ζωή του χαρακτήρα του παίκτη αυξάνεται σιγά-σιγά όσο περνά ο χρόνο, σε συγκεκριμένες τιμές ένα διαφορετικό animation παίζει. Τα animations της επιστροφής είναι τα animations του θανάτου αλλά παίζουν αντίστροφα με αρνητικό Play Rate και φυσικά έχουμε βάλει τα τρία κομμάτια με την αντίστροφη σειρά, έτσι ο χαρακτήρας σηκώνεται στην αρχική του θέση και πόζα.



Εικόνα 109: Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 109 βλέπουμε τα δύο τελευταία κομμάτια της επιστροφής του παίκτη στο παιχνίδι, όπως βλέπουμε όσο ο χαρακτήρας παίρνει ζωή περνά σε άλλο state, μέχρι να φτάσει πίσω στο αρχικό BasicStance και να ξαναρχίσει να παίζει.

Ο NPC χρησιμοποιεί ακριβώς τα ίδια Animations, Transition Rules και States, όπως ο χαρακτήρας του παίκτη. Η μόνη διαφορά είναι ότι δεν έχει το κομμάτι της επιστροφής, γιατί ο NPC δεν μπορεί να επιστρέψει στο παιχνίδι μετά την ήττα του. Όμως, υπάρχει περίπτωση να νικηθούν ο παίκτης και ο αντίπαλός του ταυτόχρονα, σε αυτή την περίπτωση ο αντίπαλος δεν εξαφανίζεται, αλλά επαναφέρεται στην αρχική του κατάσταση, για να τον αντιμετωπίσει ξανά ο παίκτης. Σε αυτή την περίπτωση έχουμε ένα Transition Rule που θα δούμε παρακάτω.



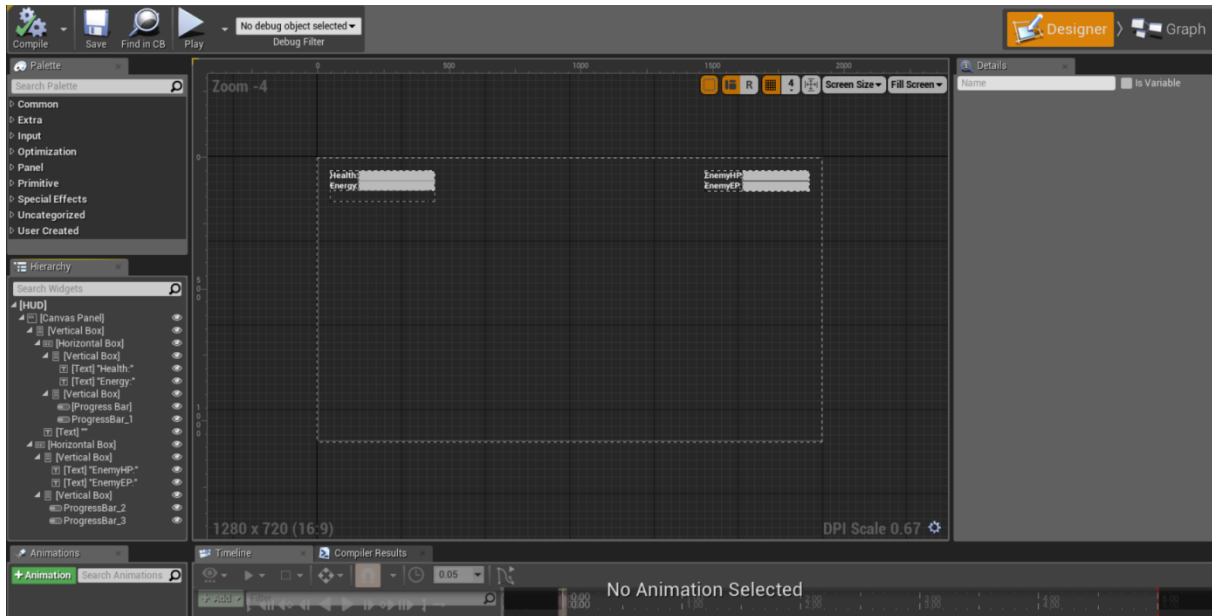
Εικόνα 110: Non-Player Character Animation Blueprint στην Unreal Engine 4

Στην Εικόνα 110 βλέπουμε το Transition Rule που επιστρέφει τον NPC στην αρχική του κατάσταση, στο BasicStance state, όταν η ζωή του επιστρέψει με την ήττα του παίκτη.

Τώρα που τελειώσαμε με τα Character και Animation Blueprints του χαρακτήρα του παίκτη και του NPC πρέπει να τα συνδέσουμε, έτσι ώστε να μπορούν τα Character Blueprints να χρησιμοποιήσουν τα Animation Blueprints και οι χαρακτήρες να μπορούν να εκτελούν τα animations. Για να γίνει αυτό πρέπει να αλλάξουμε μια ρύθμιση στα Character Blueprints, επιλέγουμε Mesh(Inherited) στο Components Tab και στο Details Tab στο Animation επιλέγουμε Animation Mode → Use Animation Blueprint και Anim Class → επιλέγουμε το Animation Blueprint που αντιστοιχεί στο κάθε Character Blueprint. Τέλος, πρέπει να κάνουμε drag and drop τα Child Blueprints του NPC, που φτιάξαμε παραπάνω, στο επίπεδο του παιχνιδιού και να τα τοποθετήσουμε στις θέσεις που θέλουμε.

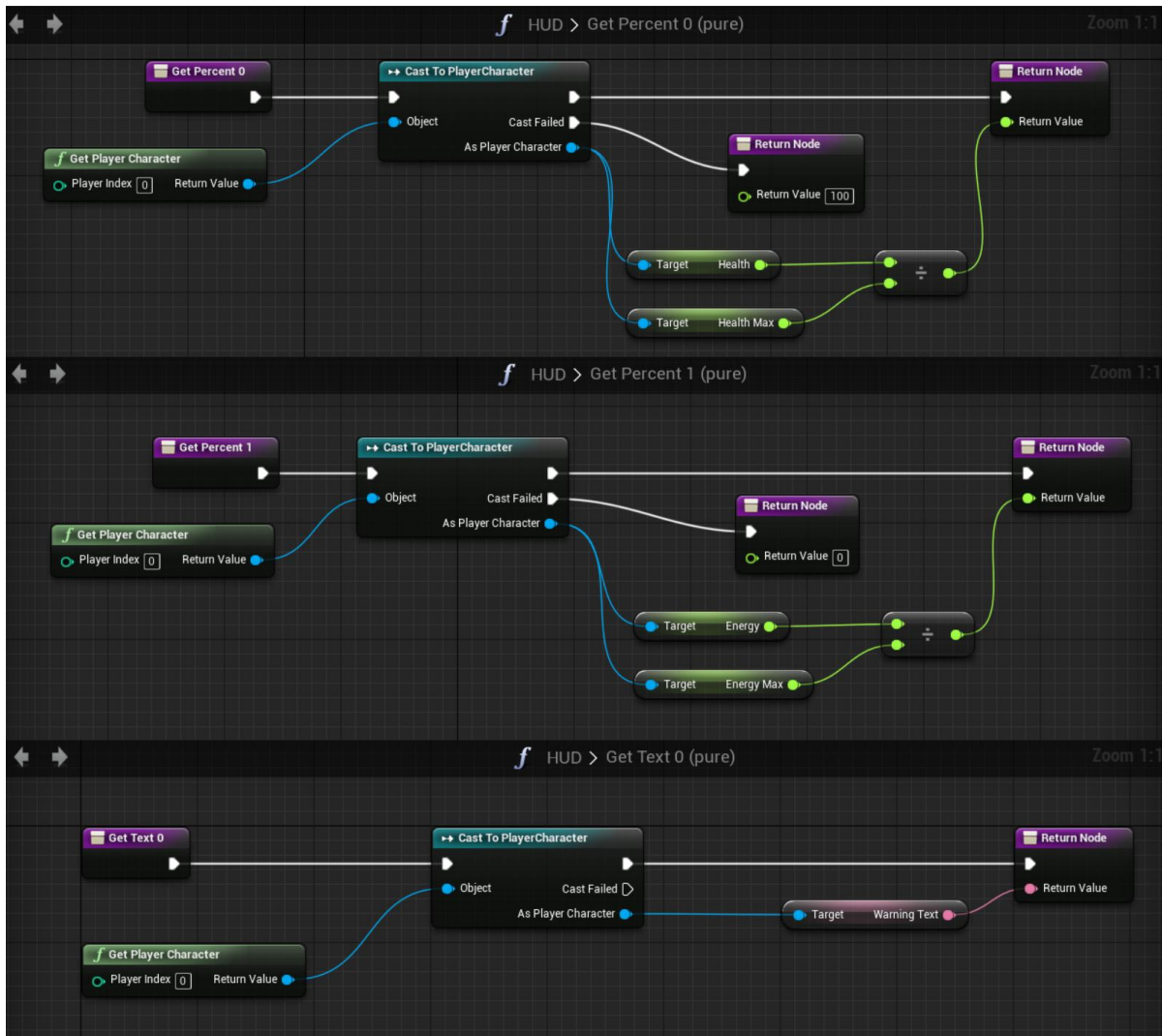
3.3.2.3. User Interface and Widget Blueprints

Σε αυτό το βήμα θα ασχοληθούμε με το User Interface και γι' αυτό θα χρειαστούμε τρία Widget Blueprints, τα οποία θα ονομάσουμε HUD, MainMenu και PauseMenu. Τα Widget Blueprints είναι ένας εύκολος τρόπος να σχεδιάσουμε και να προγραμματίσουμε το User Interface ενός παιχνιδιού στην Unreal Engine 4, παρακάτω θα δούμε πως σχεδιάζουμε το UI και πως μπορούμε να προγραμματίσουμε τις λειτουργίες του με τη χρήση διαγραμμάτων σε blueprints.



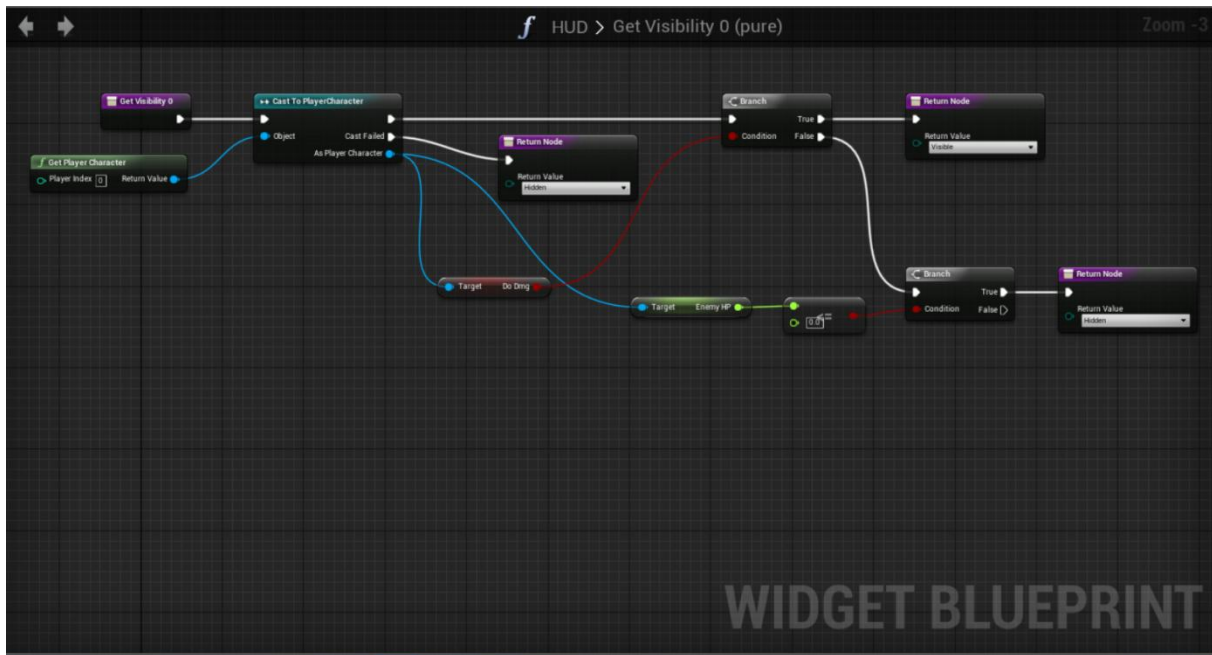
Εικόνα 111: HUD Widget Blueprint στην Unreal Engine 4

Στην Εικόνα 111 βλέπουμε το HUD, το οποίο είναι το UI που βλέπει ο παίκτης στην οθόνη του, όταν παίζει το παιχνίδι. Για να σχεδιάσουμε το UI σε ένα Widget Blueprint παίρνουμε τα στοιχεία του UI που χρειαζόμαστε από το Palette Tab και τα τοποθετούμε στο Hierarchy Tab. Όπως βλέπουμε το Hierarchy Tab, όπως λέει και το όνομά του, έχει όλα τα στοιχεία του UI μας σε μια συγκεκριμένη ιεραρχία, η οποία μας δείχνει ποιο στοιχείο περιέχεται σε ποιο άλλο στοιχείο του UI. Αυτή η ιεραρχία μας επιτρέπει να ενώσουμε διαφορετικά στοιχεία σε μία σύνθετη μορφή την οποία μετά μπορούμε να την τοποθετήσουμε όπου θέλουμε, χωρίς να χρειάζεται να μετακινούμαστε το κάθε στοιχείο ξεχωριστά ή το κάθε στοιχείο να θέλει ξεχωριστές ρυθμίσεις.



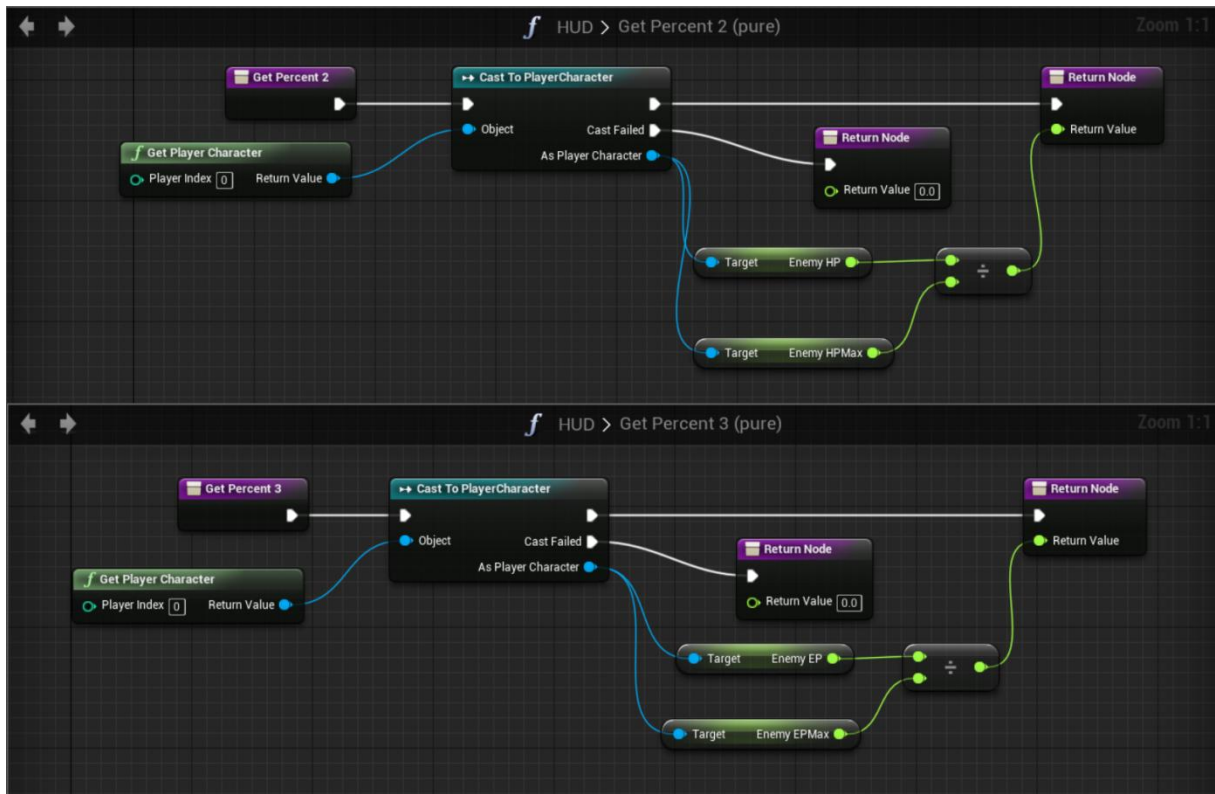
Εικόνα 112: HUD Widget Blueprint Συναρτήσεις στην Unreal Engine 4

Έπειτα, πρέπει να προσθέσουμε τη λειτουργικότητα του UI (Εικόνα 112). Πρώτα, θα φτιάξουμε τα Progress Bars που αντιπροσωπεύουν τη Ζωή και την Ενέργεια του χαρακτήρα του παίκτη, αυτό γίνεται επιλέγοντας το κάθε ένα από αυτά και στο Details Tab, αλλάζουμε το χρώμα στο Appearance → Fill Color and Opacity και στο Progress → Percent → Bind → Create Binding δημιουργούμε μία συνάρτηση. Οι συναρτήσεις των Progress Bars της Ζωής και της Ενέργειας είναι οι Get Percent 0 και Get Percent 1, αντίστοιχα, και μπορούμε να τις δούμε στην Εικόνα X. Όπως βλέπουμε οι συναρτήσεις είναι οι ίδιες και αυτό που κάνουν είναι να παίρνουν τις τιμές των αντίστοιχων μεταβλητών και να τις προσαρμόζουν έτσι ώστε να βλέπουμε τα Progress Bars να αλλάζουν με βάση τις τιμές (Σημείωση: Όπως βλέπουμε η προσαρμογή γίνεται με τις πράξεις Health/HealthMax και Energy/EnergyMax, αυτό γίνεται γιατί οι τιμές ενός Progress Bar είναι μεταξύ 0 και 1, έτσι διαιρώντας τις τιμές αυτές με τις μέγιστες τιμές τους παίρνουμε ένα αποδεκτό αποτέλεσμα). Αντίστοιχα για το Text, αλλάζουμε το χρώμα στο Appearance → Fill Color and Opacity και μετά στο Content → Bind → Create Binding δημιουργούμε τη συνάρτηση Get Text 0, η οποία παίρνει την αντίστοιχη τιμή από το Blueprint του χαρακτήρα του παίκτη.



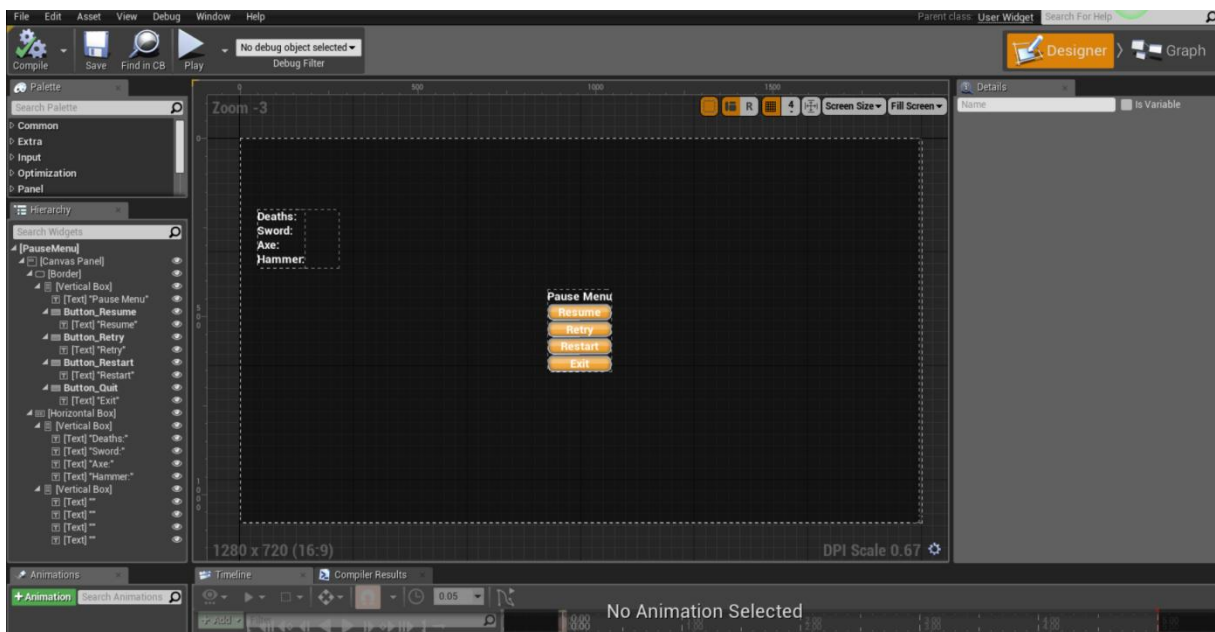
Εικόνα 113: HUD Widget Blueprint Συνάρτηση στην Unreal Engine 4

Στην Εικόνα 113 βλέπουμε τη συνάρτηση Get Visibility 0, η οποία αναφέρεται στο Horizontal Box που περιλαμβάνει τα στοιχεία του αντίπαλου NPC, κατά τη διάρκεια της μάχης. Για να τη δημιουργήσουμε πρέπει να πάμε Behavior → Visibility → Bind → Create Binding. Η συνάρτηση χρησιμοποιεί τη μεταβλητή από το blueprint του χαρακτήρα του παίκτη που παρακολουθεί την τιμή της Ζωής του αντιπάλου και τη μεταβλητή της μάχης. Ανάλογα με τα αποτελέσματα όλα τα στοιχεία που περιέχονται στο Horizontal Box γίνονται ορατά ή αόρατα στην οθόνη.



Εικόνα 114: HUD Widget Blueprint Συναρτήσεις στην Unreal Engine 4

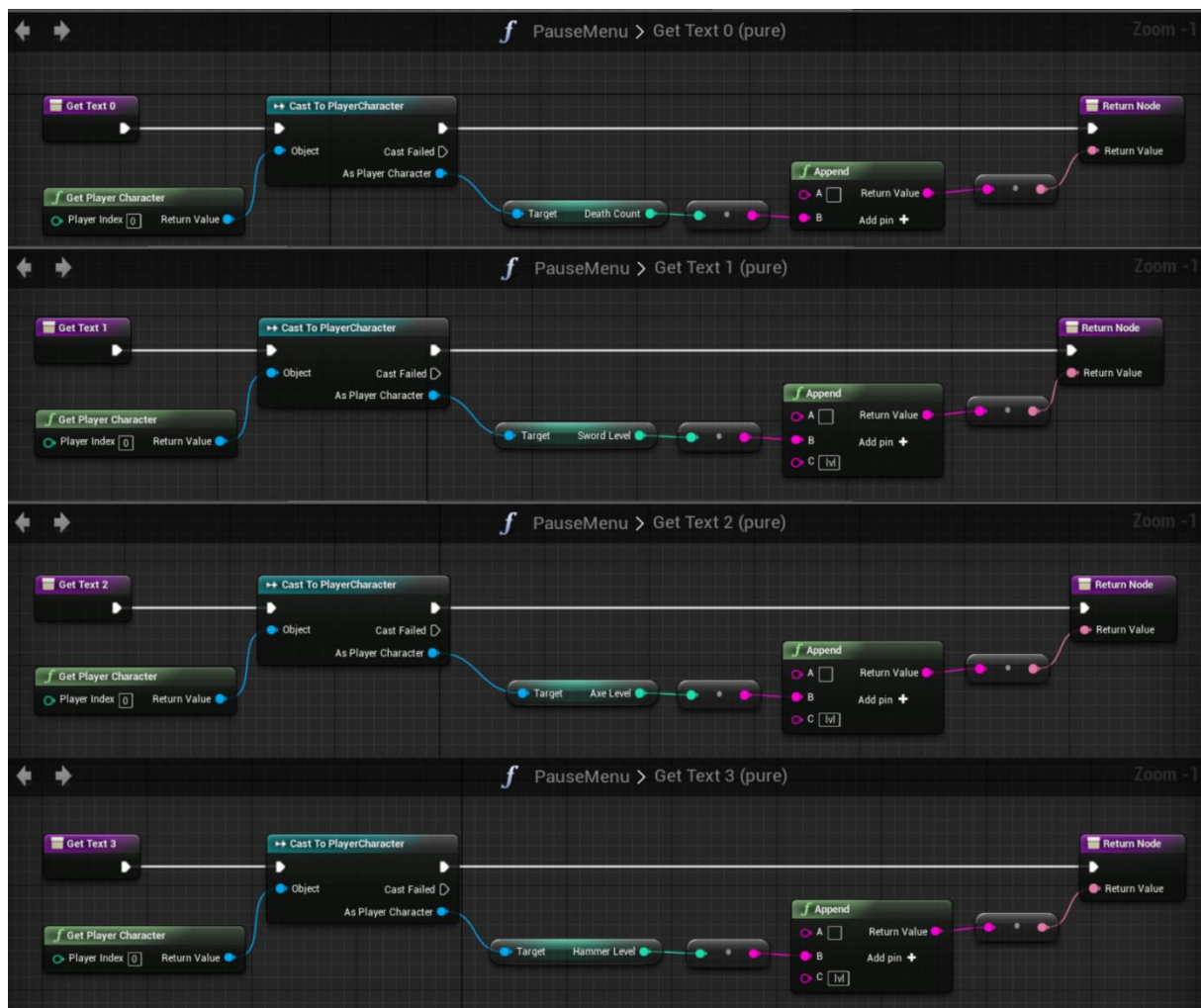
Στην Εικόνα 114 βλέπουμε τις συναρτήσεις που επηρεάζουν τα Progress Bars που αφορούν τα στοιχεία του αντιπάλου, οι συναρτήσεις και οι ρυθμίσεις των Progress Bars είναι ίδιες με τις αντίστοιχες των Progress Bars των στοιχείων του χαρακτήρα του παίκτη που είδαμε παραπάνω.



Εικόνα 115: Pause Menu Widget Blueprint στην Unreal Engine 4

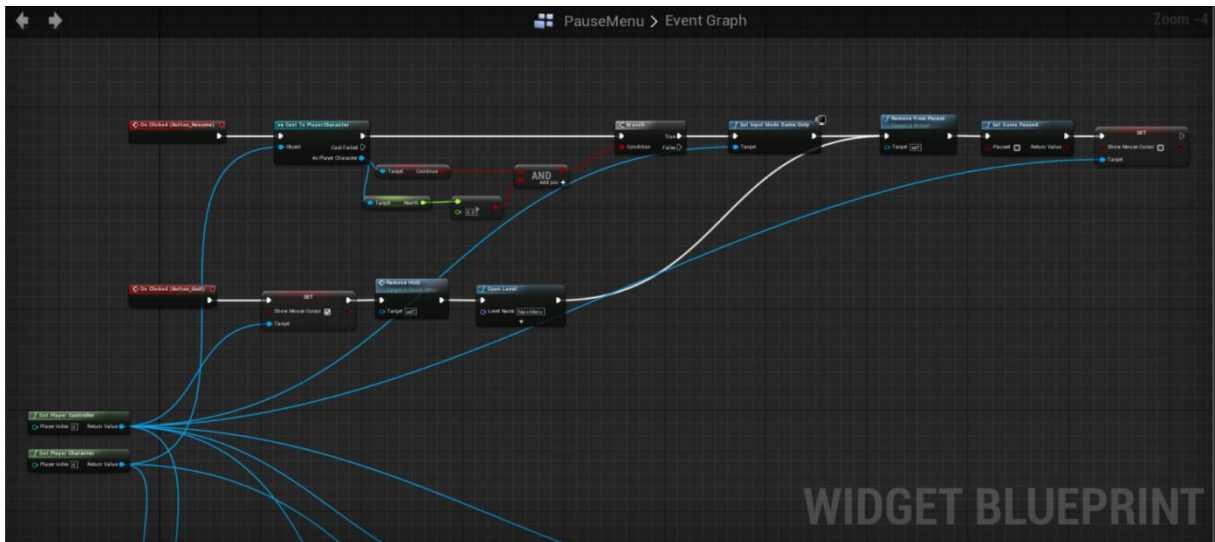
Στην Εικόνα 115 βλέπουμε το Pause Menu που εμφανίζεται όταν ο παίκτης πατήσει το κουμπί P ή χάσει. Για να φτιάξουμε αυτό το UI θα χρειαστούμε να εισάγουμε μία εικόνα

στην Unreal Engine 4 για τα κουμπιά του UI, αυτό γίνεται κάνοντας drag and drop την εικόνα στο Content Browser στο κεντρικό παράθυρο του προγράμματος. Η κατασκευή του UI γίνεται με τον ίδιο τρόπο που εξηγήσαμε παραπάνω. Όσον αφορά τα κουμπιά του menu πρέπει να τα επιλέξουμε ένα προς ένα και να αλλάξουμε τις ρυθμίσεις τους. Πηγαίνοντας στο Appearance → Style → Normal → Image επιλέγουμε την εικόνα που εισάγαμε, έπειτα στις ρυθμίσεις Style → Hovered και Style → Pressed αλλάζουμε το χρώμα στη ρύθμιση Tint σε κίτρινο και κόκκινο, αντίστοιχα. Τέλος, πρέπει να αλλάξουμε τη ρύθμιση Is Variable σε αληθή σε όλα τα κουμπιά, για να μπορούμε να χρησιμοποιήσουμε τα σχετικά Events στις συναρτήσεις μας, όπως το On Clicked Event.

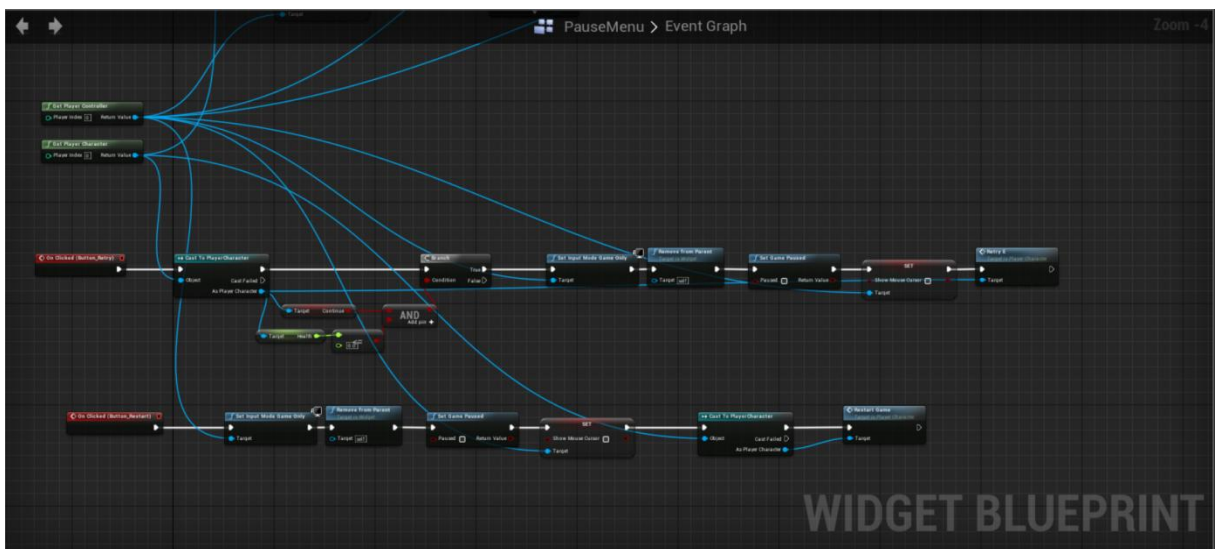


Εικόνα 116: Pause Menu Widget Blueprint Συναρτήσεις στην Unreal Engine 4

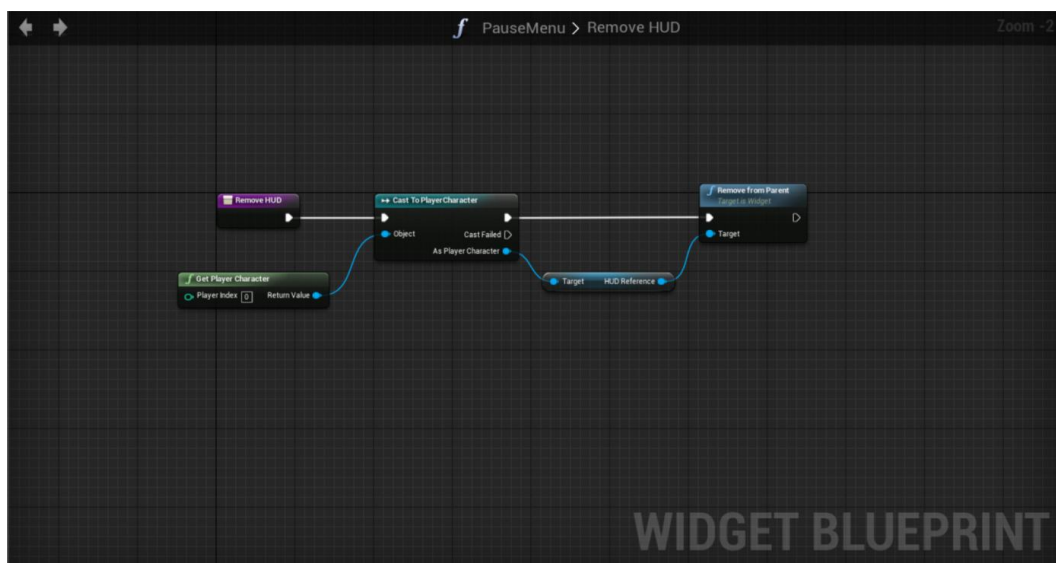
Στην Εικόνα 116 βλέπουμε τις συναρτήσεις που χρησιμοποιούν τα Texts, για να προβάλουν τις τιμές που θέλουμε, από το blueprint του χαρακτήρα του παίκτη, στην οθόνη όταν ανοίγουμε το Pause Menu. Η διαδικασία για τη δημιουργία τους και την συσχέτισή τους με τα Texts είναι ίδια όπως παραπάνω.



Εικόνα 117: Pause Menu Widget Blueprint στην Unreal Engine 4

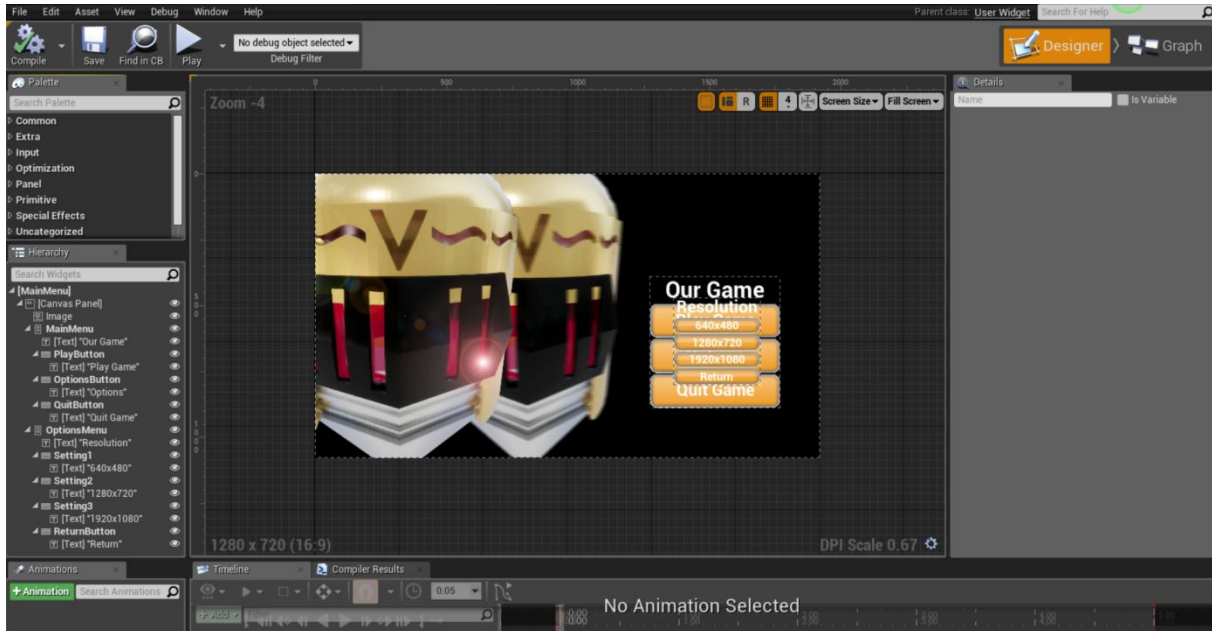


Εικόνα 118: Pause Menu Widget Blueprint στην Unreal Engine 4



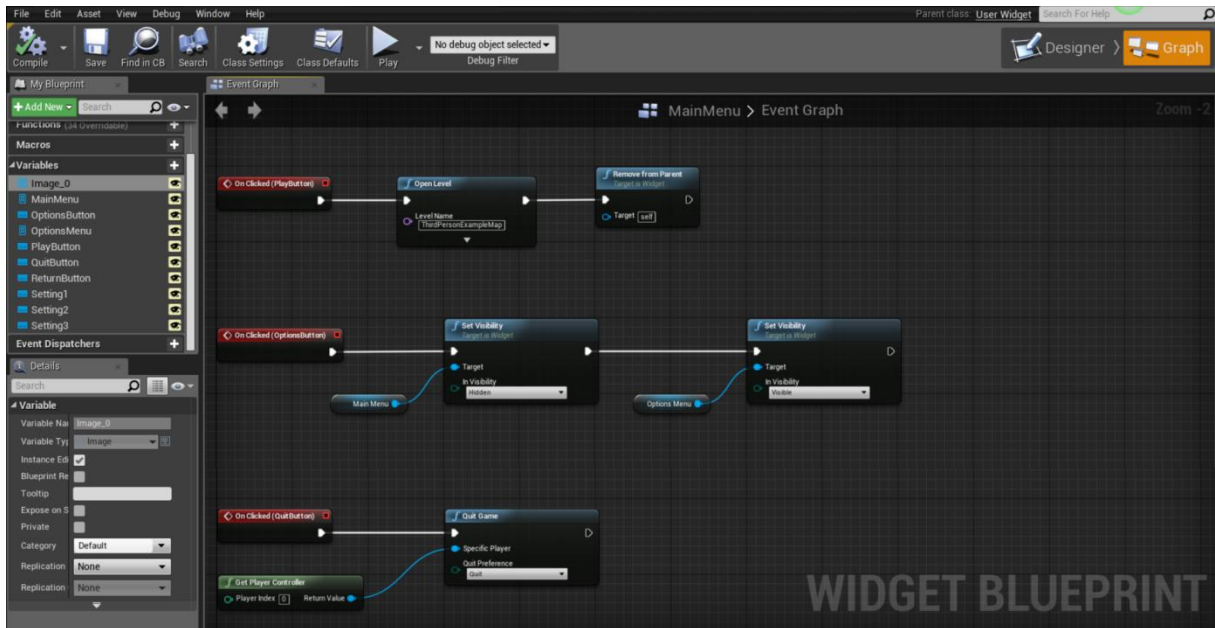
Εικόνα 119: Pause Menu Widget Blueprint Συνάρτηση στην Unreal Engine 4

Στις Εικόνες 117, 118 βλέπουμε το Event Graph που περιγράφει τις ρυθμίσεις που γίνονται όταν πατάμε κάποιο από τα κουμπιά του Pause Menu, ενώ η συνάρτηση της Εικόνα Z, είναι μια συνάρτηση που καλείται από το Event Graph. Το Resume κλείνει το Pause Menu και ξεκινά το παιχνίδι. Το Quit καλεί τη συνάρτηση για να αφαιρέσει το HUD (Εικόνα 119), ανοίγει το επίπεδο του Main Menu και αφαιρεί το Pause Menu. Το Retry αφαιρεί το Pause Menu και καλεί μια συνάρτηση από το blueprint χαρακτήρα του παίκτη, την οποία αναφέραμε παραπάνω, το ίδιο και στην περίπτωση του Restart.

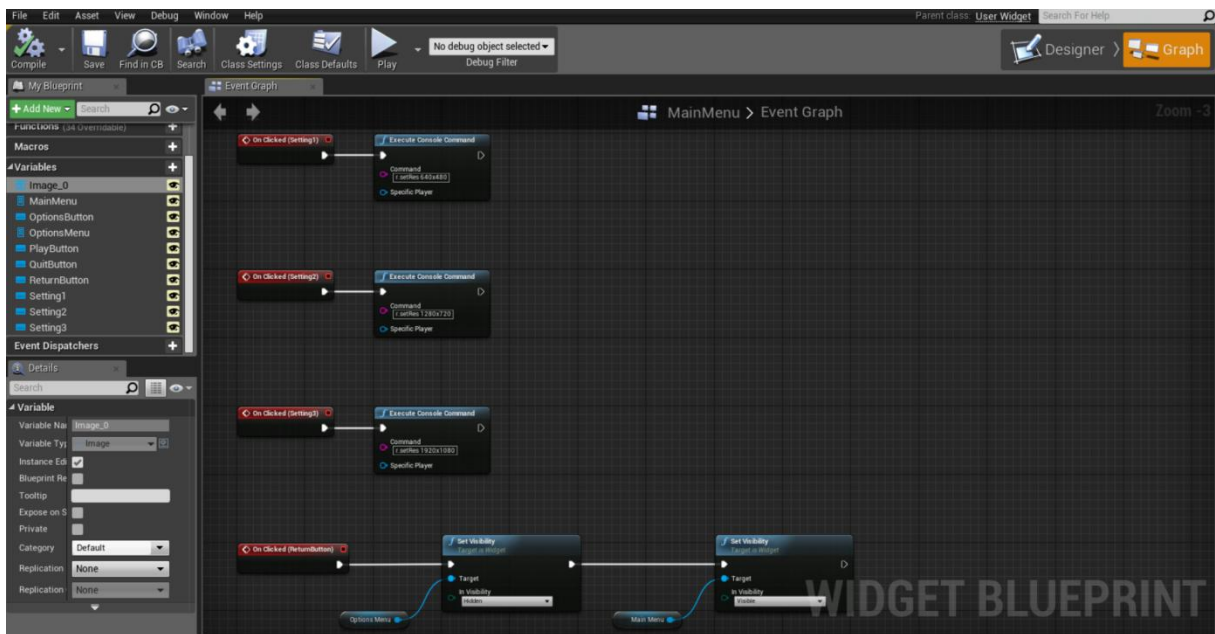


Εικόνα 120: Main Menu Widget Blueprint στην Unreal Engine 4

Στην Εικόνα 120 βλέπουμε το Main Menu, το οποίο είναι το menu που εμφανίζεται όταν ξεκινάμε το παιχνίδι. Για το Main Menu θα χρειαστούμε μία εικόνα για φόντο, την οποία την εισάγουμε με τον ίδιο τρόπο όπως την εικόνα για τα κουμπιά παραπάνω. Έπειτα πάμε Appearance → Brush → Image και επιλέγουμε την εικόνα και μετά στο Slot(Canvas Panel Slot) → Anchors επιλέγουμε το Anchor που υπάρχει στην κάτω δεξιά γωνία για να τοποθετήσουμε την εικόνα στο φόντο και να προσαρμόζεται πάντα σε όλη την οθόνη. Επίσης θέλουμε τα κουμπιά να έχουν τις ίδιες ρυθμίσεις με τα κουμπιά του Pause Menu παραπάνω. Τα MainMenu και OptionsMenu, τα οποία είναι Vertical Boxes, πρέπει να έχουν τη ρύθμιση Is Variable αληθή, γιατί έτσι μπορούμε να τα χρησιμοποιήσουμε στο Event Graph παρακάτω.



Εικόνα 121: Main Menu Widget Blueprint στην Unreal Engine 4

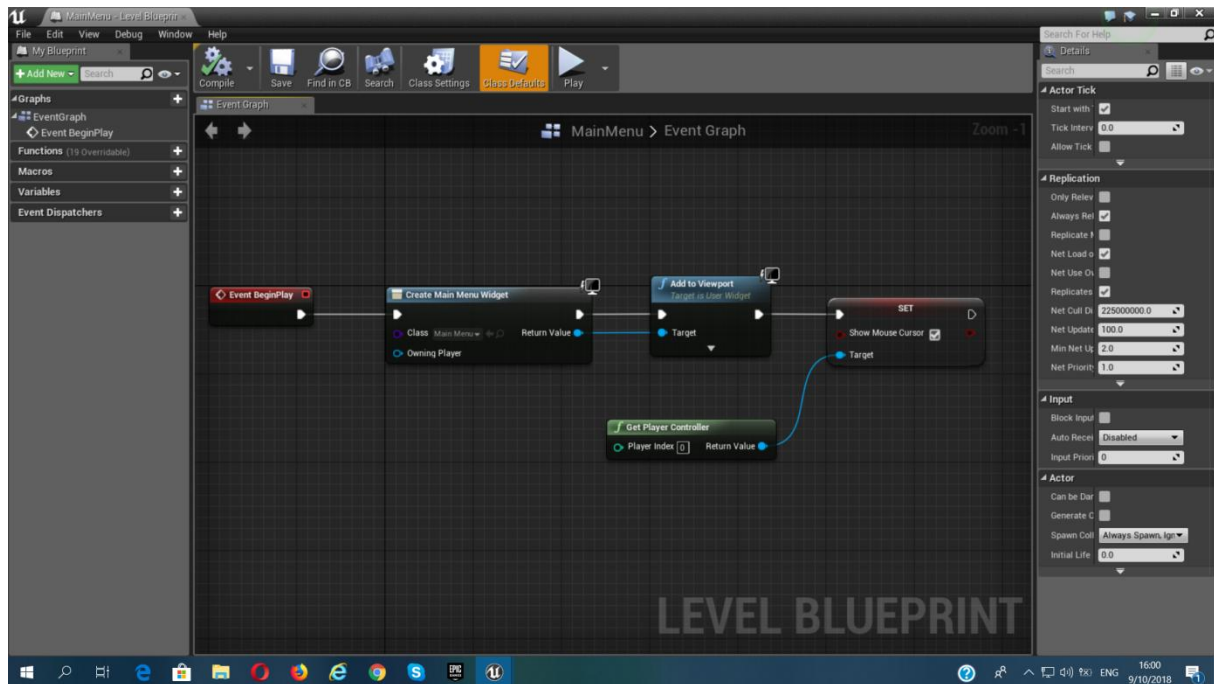


Εικόνα 122: Main Menu Widget Blueprint στην Unreal Engine 4

Στις Εικόνες 121, 122 βλέπουμε το Event Graph του Main Menu, το οποίο χρησιμοποιείται για τη λειτουργικότητα του menu και των κουμπιών του. Στην Εικόνα 121 έχουμε τα διαγράμματα των κουμπιών του MainMenu, το κουμπί Play ανοίγει το επίπεδο του παιχνιδιού και κλείνει το menu, το κουμπί Options κάνει άορατο το MainMenu και ορατό το OptionsMenu, ενώ το κουμπί Quit κλείνει το παιχνίδι. Στην Εικόνα 122 έχουμε τα διαγράμματα των κουμπιών του OptionsMenu, το κουμπί Return κάνει το ανάποδο από το κουμπί Options, δηλαδή κάνει άορατο το OptionsMenu και ορατό το MainMenu, ενώ τα άλλα κουμπιά αλλάζουν την ανάλυση του παιχνιδιού, εκτελώντας την αντίστοιχη εντολή.

3.3.2.4. Level Blueprints and Final Settings

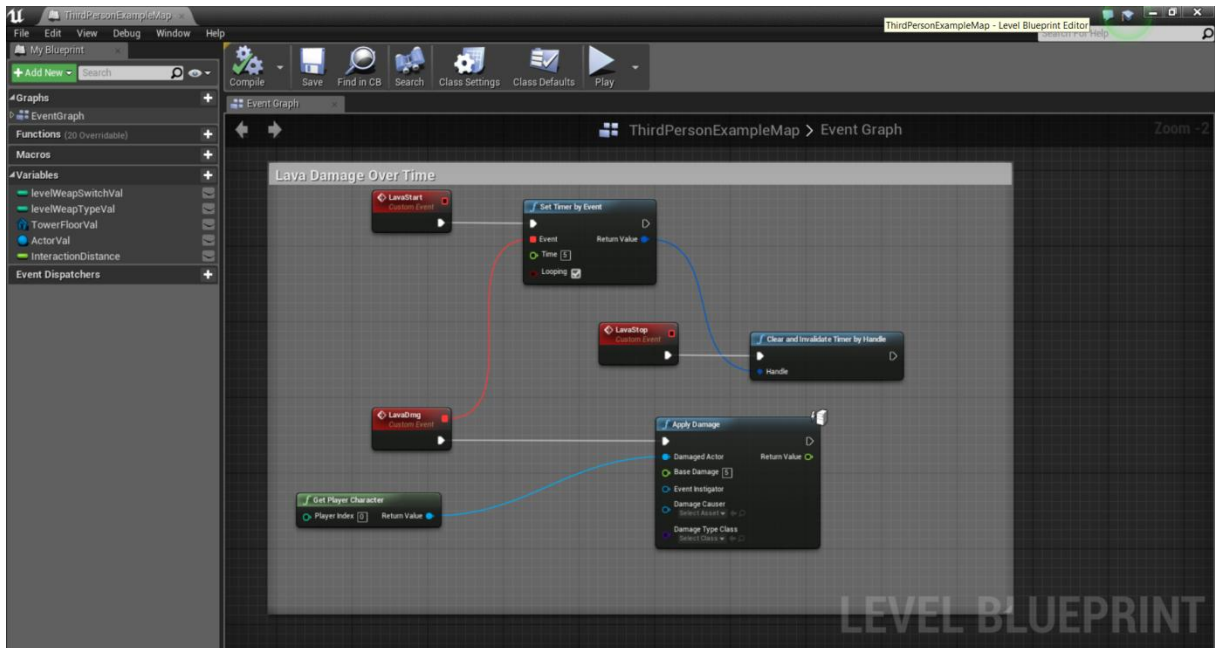
Στο τελευταίο βήμα θα ασχοληθούμε με τα Επίπεδα του παιχνιδιού, τα Level Blueprints και τα διαγράμματά τους, καθώς επίσης και με κάποιες ρυθμίσεις του παιχνιδιού και του UI.



Εικόνα 123: Main Menu Level Blueprint στην Unreal Engine 4

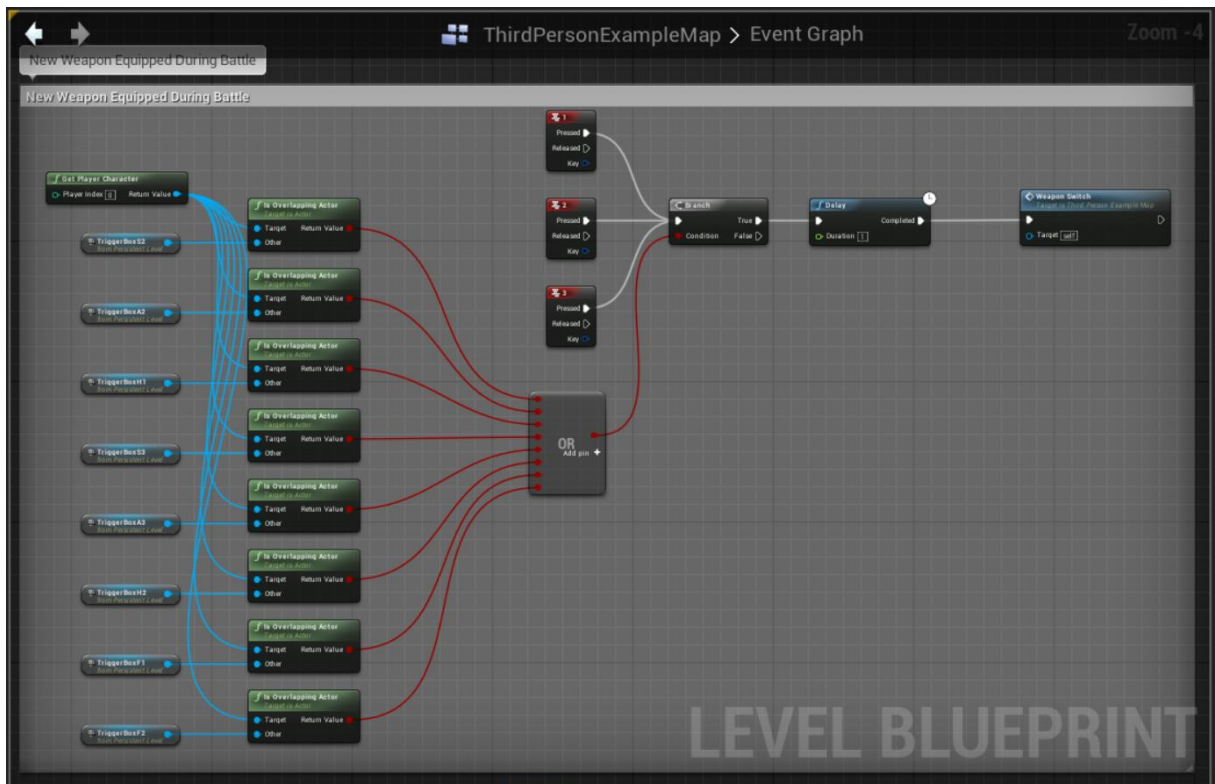
Στην Εικόνα 123 βλέπουμε το Level Blueprint του Main Menu Level, το οποίο έχει ένα διάγραμμα που δημιουργεί το Main Menu και ενεργοποιεί τον κέρσορα. Αυτό το Level πρέπει να το δημιουργήσουμε γιατί δεν υπάρχει στο project μας, οπότε πάμε δεξί κλικ στο Content Browser και επιλέγουμε Level, όπως είπαμε και παραπάνω. Αυτό το Level είναι κενό, γιατί η μόνη χρήση του είναι να περιέχει το UI του Main Menu. Το διάγραμμα ξεκινά όταν αρχίσει το παιχνίδι και ανοίξει το επίπεδο, δημιουργεί το UI του Main Menu και ενεργοποιεί τον κέρσορα.

Τέλος, θα ασχοληθούμε με το δεύτερο και τελευταίο επίπεδο του παιχνιδιού, που είναι το επίπεδο που παίζεται το παιχνίδι. Παρακάτω θα δούμε τα διαγράμματα του Level Blueprint και τη λειτουργία τους.



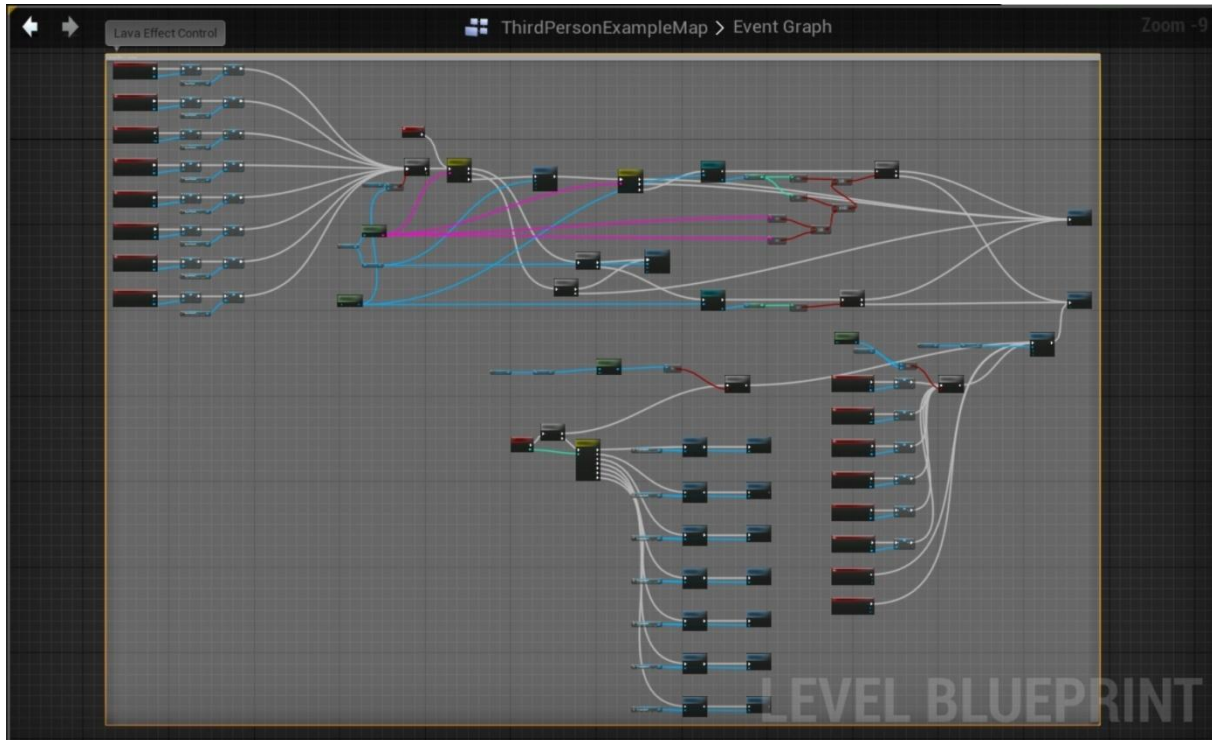
Εικόνα 124: Level Blueprint του Επιπέδου του Παιχνιδιού στην Unreal Engine 4

Στην Εικόνα 124 βλέπουμε το Lava Damage Over Time διάγραμμα του Event Graph του Level Blueprint. Το διάγραμμα LavaDmg κάνει ζημιά στον παίκτη όταν καλεστεί. Το διάγραμμα LavaStart ενεργοποιεί έναν Timer που καλεί το LavaDmg ανά τακτά χρονικά διαστήματα με βάση το χρόνο που διαλέξαμε, ενώ το LavaStop σταματά τον Timer, οπότε σταματά και η ζημιά προς τον παίκτη.



Εικόνα 125: Level Blueprint του Επιπέδου του Παιχνιδιού στην Unreal Engine 4

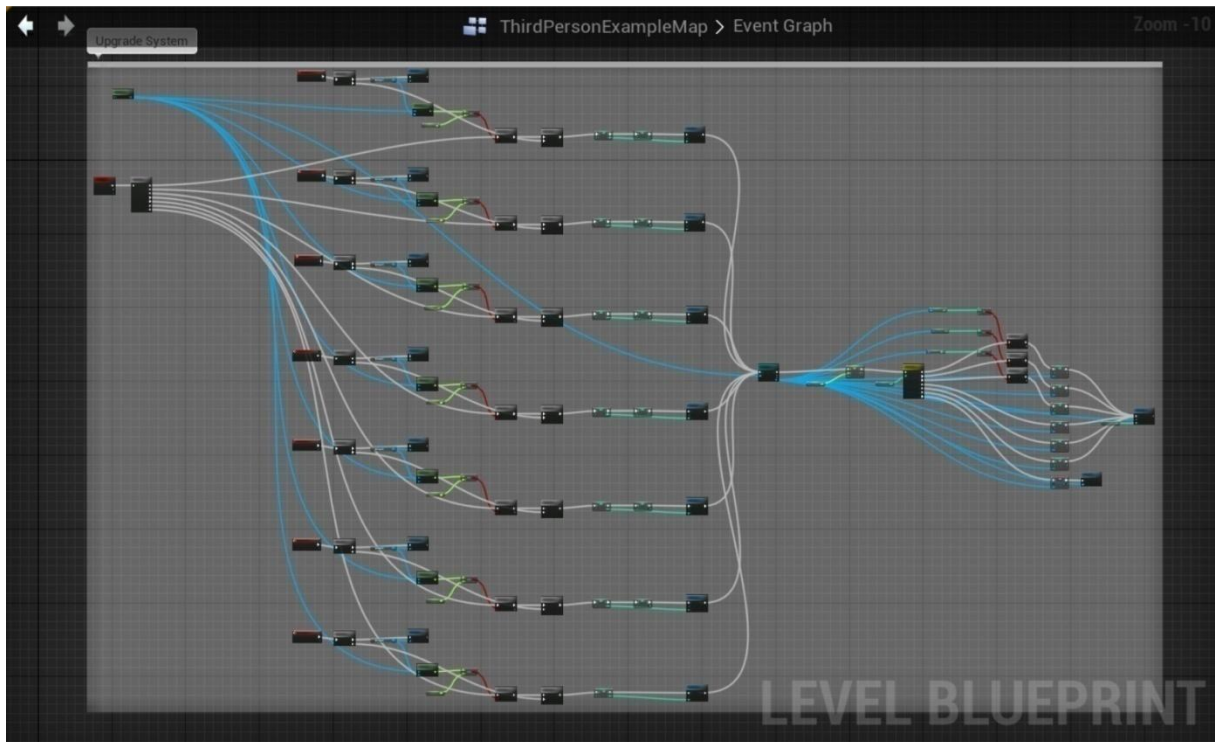
Στην Εικόνα 125 βλέπουμε το New Weapon Equipped During Battle διάγραμμα του Event Graph του Level Blueprint. Αυτό το διάγραμμα, όταν ο παίκτης πατήσει ένα από τα κουμπιά 1,2 ή 3, τα οποία αλλάζουν το όπλο του χαρακτήρα, ενώ ο χαρακτήρας βρίσκεται σε περιοχή με λάβα, καλεί το διάγραμμα Lava Effect Control, που θα δούμε παρακάτω, μέσω του Custom Event WeaponSwitch.



Εικόνα 126: Level Blueprint του Επιπέδου του Παιχνιδιού στην Unreal Engine 4

Στην Εικόνα 126 βλέπουμε το Lava Effect Control διάγραμμα του Event Graph του Level Blueprint. Αυτό το διάγραμμα αλλάζει το υλικό(material) του δαπέδου σε λάβα όταν ο παίκτης έρθει σε επαφή με το TriggerBox που αντιστοιχεί σε αυτό το δάπεδο ή το επαναφέρει στο αρχικό υλικό όταν ο παίκτης απομακρυνθεί. Επίσης με βάση τον αλγόριθμο του διαγράμματος καλεί το LavaStart ή το LavaStop, που είδαμε παραπάνω, με βάση το όπλο που έχει ο παίκτης ενεργό. Το Custom Event NoLava καταστρέφει τα TriggerBoxes και επαναφέρει τα δάπεδα στην αρχική τους κατάσταση όταν καλεστεί από το διάγραμμα Upgrade System, που θα δούμε παρακάτω, με βάση την αντίστοιχη μεταβλητή. (Σημείωση: Για να γίνει η αλλαγή του υλικού του δαπέδου, πρέπει το αντικείμενό μας να είναι Static Mesh Actor και όχι Brush, σε αυτή την περίπτωση επιλέγουμε το αντικείμενο Brush, στο World Outliner Tab, και μετά πάμε Details Tab → Brush Settings → Create Static Mesh, αν δεν υπάρχει η ρύθμιση πατάμε στο βελάκι για να δούμε όλες τις ρυθμίσεις.)

Για να τοποθετήσουμε τα Triggerboxes, πρέπει να πάμε στο κεντρικό παράθυρο του Unreal Engine 4 και από το Modes Tab, τα κάνουμε drag and drop στο χάρτη και τα προσαρμόζουμε στην κάθε περιοχή που θέλουμε, θα χρειαστούμε οκτώ συνολικά, ένα για κάθε περιοχή. Έπειτα στο World Outliner Tab σε κάθε Triggerbox κάνουμε δεξί κλικ και στο Add Event επιλέγουμε τα OnActorBeginOverlap και OnActorEndOverlap, για να προστεθούν στο Level Blueprint.



Εικόνα 127: Level Blueprint του Επιπέδου του Παιγνιδιού στην Unreal Engine 4

Το διάγραμμα Upgrade System (Εικόνα 127) ενεργοποιείται όταν πατά ο παίκτης το Spacebar, αν βρίσκεται σε μια προκαθορισμένη απόσταση από ένα από τα lightCage(εναλλακτικά μπορούμε να επιλέξουμε τον αντίστοιχο πύργο) και ο αντίστοιχος αντίπαλος έχει ηττηθεί. Το διάγραμμα καλεί το NoLava, που αναφέραμε παραπάνω, επαναφέρει τη Ζωή του χαρακτήρα του παίκτη στην αρχική της κατάσταση και ξεκλειδώνει το αντίστοιχο όπλο. Το όπλο εμφανίζεται αυτόματα πάνω στο χαρακτήρα λόγω της κλήσης του Equip από το blueprint του χαρακτήρα, εκτός και αν τελείωσε το παιχνίδι όπου καλείται το Pause Menu, χωρίς όμως να μπορούμε να επιλέξουμε Resume ή Retry, οι μόνες επιλογές είναι Restart ή Exit.

Τέλος, πρέπει να αλλάξουμε κάποιες ρυθμίσεις του project πηγαίνοντας στο Settings → Project Settings. Πρέπει να αλλάξουμε το Default Pawn Class σε PlayerCharacter(το όνομα του blueprint του χαρακτήρα του παίκτη), το Game Default Map σε MainMenu(το όνομα του Level του Main Menu) και να αλλάξουμε το splash στην εικόνα που θέλουμε.

4. Αποτελέσματα

Το παιχνίδι παίζεται χρησιμοποιώντας τα κουμπιά 1, 2, 3 (για να αλλάξουμε όπλα σε Σπαθί, Γσεκούρι και Σφυρί, αντίστοιχα), τα βελάκια ή τα κουμπιά W, A, S, D (για να κινείται), Z ή το αριστερό κουμπί του ποντικιού, X ή το μεσαίο κουμπί του ποντικιού, C ή το δεξί κουμπί του ποντικιού (για να χρησιμοποιήσει την επίθεση, ειδική επίθεση ή άμυνα, αντίστοιχα).

Ο παίκτης κερδίζει νέα όπλα νικώντας αντιπάλους και χρησιμοποιώντας τον αντίστοιχο πύργο, επίσης θεραπεύεται με κάθε χρήση. Το παιχνίδι τελειώνει όταν νικηθεί ο αντίπαλος του τελευταίου πύργου. Επίσης, ανάλογα το όπλο που χρησιμοποιεί ο παίκτης ενάντια σε κάποιον αντίπαλο μπορεί να λαμβάνει ζημιά από την αρένα.

4.1. Συμπεράσματα

Μέσα από αυτή τη πτυχιακή καταλάβαμε ότι η δημιουργία ενός παιχνιδιού είναι ένα δύσκολο εγχείρημα για αρχάριους προγραμματιστές. Η Unreal Engine 4 είναι εύκολη στην εκμάθηση και χρήση, αλλά είναι δύσκολο να βρεθεί υλικό για μελέτη. Το μεγαλύτερο κομμάτι του υλικού που μελετήσαμε ήταν απαρχαιωμένο και είχε να κάνει με προηγούμενες εκδόσεις του προγράμματος που δε λειτουργούσαν με τον ίδιο τρόπο, το υλικό προήρθε από βίντεο από το YouTube, το documentation της Unreal Engine 4 και σχετικά forums. Η χρήση των Blueprints είναι πολύ πιο εύκολη από την C++ και είναι φιλικά προς νέους χρήστες, ίσως σε τέτοιο βαθμό που και μη-προγραμματιστές θα μπορούσαν να τα χρησιμοποιήσουν.

Επίσης, ένα παιχνίδι αποτελείται από τμήματα, όπως μοντέλα και σκελετοί χαρακτήρων και μοντέλων, animations, level design, γραφικά, σχεδιασμός του User Interface, προγραμματισμός αλλά και ο σχεδιασμός του τρόπου με τον οποίο παίζεται το παιχνίδι, που χρειάζονται άτομα με εξειδικευμένες γνώσεις στο κάθε αντικείμενο, δηλαδή συμπεραίνουμε ότι για τον καλύτερο σχεδιασμό ενός εμπορικού παιχνιδιού χρειάζεται μια ομάδα ατόμων.

4.2. Μελλοντική Εργασία και Επεκτάσεις

Για να βελτιώσουμε την πτυχιακή θα μπορούσαμε να προσθέσουμε περισσότερα όπλα, επίπεδα, πανοπλίες και μοντέλα χαρακτήρων. Θα μπορούσαμε να βελτιώσουμε τα animations και τα γραφικά ή να κάνουμε τη μάχη πιο πολύπλοκη ή και το AI των αντιπάλων. Επίσης, θα μπορούσαμε να προσθέσουμε ήχο για διάφορες δραστηριότητες των χαρακτήρων ή για το επίπεδο.

Η πτυχιακή, στην τωρινή της κατάσταση, δεν είναι κατάλληλη για τη δημιουργία ενός εμπορικού παιχνιδιού, αλλά είναι καλή για την εκμάθηση της Unreal Engine 4 και τη δημιουργία παιχνιδιών σε αρχάριο επίπεδο στο πλαίσιο άλλων πτυχιακών εργασιών.

Βιβλιογραφία

- [1] Σελίδα του MakeHuman στην Wikipedia. [Online]. <https://en.wikipedia.org/wiki/MakeHuman>
- [2] Σελίδα του Unreal Engine 4 στην Wikipedia. [Online].
https://en.wikipedia.org/wiki/Unreal_Engine#Unreal_Engine_4
- [3] Σελίδα του Blender στην Wikipedia. [Online]. [https://en.wikipedia.org/wiki/Blender_\(software\)](https://en.wikipedia.org/wiki/Blender_(software))
- [4] Σελίδα του MakeHuman. [Online]. <http://www.makehumancommunity.org/>
- [5] Σελίδα του mhx2 plugin. [Online]. <https://thomasmakehuman.wordpress.com/mhx2-documentation/preparation/>
- [6] Σελίδα του mhx2 plugin. [Online]. https://thomasmakehuman.wordpress.com/mhx2-documentation/mh_export/
- [7] Σελίδα του mhx2 plugin. [Online]. <https://thomasmakehuman.wordpress.com/mhx2-documentation/import-into-blender/>
- [8] Σελίδα του mhx2 plugin. [Online]. <https://thomasmakehuman.wordpress.com/mhx2-documentation/mhx-setup-panel/>
- [9] Ροή Εργασίας από το MakeHuman στο Blender. [Online].
https://wiki.unrealengine.com/Workflow: MakeHuman_and Blender
- [10] Εγκατάσταση των Blender Tools. [Online].
] http://www.makehumancommunity.org/wiki/Documentation:Getting_and_installing_BlenderTools
- [11] Εξαγωγή από το Blender στο Wiki της Unreal Engine. [Online].
] https://wiki.unrealengine.com/Static_Mesh_from_Blender#Exporting
- [12] Ρυθμίσεις για Εισαγωγή/Εξαγωγή από το Blender στην Unreal Engine 4 από το Unreal Engine
] Forums. [Online]. <https://forums.unrealengine.com/development-discussion/content-creation/33915-blender-2-73a-and-2-74-working-export-to-ue4-settings-for-various-types>
- [13] Σελίδα της Unreal Engine. [Online]. <https://www.unrealengine.com/en-US/what-is-unreal-engine-4>
] [engine-4](https://www.unrealengine.com/en-US/what-is-unreal-engine-4)
- [14] Σελίδα του Unreal Engine 4 Documentation. [Online]. <https://docs.unrealengine.com/en-us/>
]
- [15] Retro Gamer. Unreal Engine 4- Lava Tutorial. Για Lava Material 1. [Online].
] <https://www.youtube.com/watch?v=H13BbNvKYjA>
- [16] Dean Ashford. UE4 - Tutorial - Lava! Για Lava Material 2. [Online].

-] <https://www.youtube.com/watch?v=blvjz3A3anQ>
- [17 Youtube Κανάλι: Virtus Learning Hub. [Online].
] <https://www.youtube.com/user/VirtusEdu/featured>
- [18 Youtube Κανάλι: Dean Ashford. [Online].
] <https://www.youtube.com/channel/UCAaWnOJ4iFSQluBVNS2d-Ew/featured>
- [19 Youtube Κανάλι: Leif Peterson. [Online].
] <https://www.youtube.com/user/TheFurryBothan/featured>
- [20 Youtube Κανάλι: Unreal Engine. [Online].
] <https://www.youtube.com/user/UnrealDevelopmentKit/featured>
- [21 Youtube Κανάλι: Tesla Dev. [Online]. <https://www.youtube.com/user/TeslaUE4/featured>
]
- [22 Youtube Κανάλι: Titanic Games. [Online].
] https://www.youtube.com/channel/UCdoWGpMQK_L29bWXoDelltw/featured
- [23 Youtube Κανάλι: CodeViper. [Online]. <https://www.youtube.com/channel/UC-nmsM7CVwn6WscuZHmN0TA/featured>
]
- [24 Youtube Κανάλι: Ben Ormstad. [Online]. <https://www.youtube.com/user/3Dstuffz/featured>
]
- [25 Youtube Κανάλι: Logos @ Critical Gameware. [Online].
] <https://www.youtube.com/channel/UCCcBLSfQ7zWR4Xyo8nOwPBw/featured>
- [26 Youtube Κανάλι: Quick and Easy. [Online].
] <https://www.youtube.com/channel/UCMeRC08nC4tsQ5mD9n8rZRg/featured>
- [27 Youtube Κανάλι: MrFalaranah. [Online]. <https://www.youtube.com/user/MrFalaranah/featured>
]
- [28 Youtube Κανάλι: Totally Unreal. [Online].
] <https://www.youtube.com/user/AdvancedGamers34/featured>
- [29 Youtube Κανάλι: Greg Mladucky. [Online].
] <https://www.youtube.com/user/GregMladucky/playlists>
- [30 Youtube Κανάλι: Aberforth Dimension. [Online].
] https://www.youtube.com/channel/UCKGWdrI_2X3CjQePf6ay_nQ/featured
- [31 Youtube Κανάλι: Isaac Threedart. [Online]. <https://www.youtube.com/user/isaac3dart/playlists>
]

- [32 Youtube Κανάλι: Mathew Wadstein. [Online].
] <https://www.youtube.com/channel/UCOVfF7PflbRdVEm0hONTrNQ/featured>
- [33 Youtube Κανάλι: Rakiz Farooq. [Online].
] <https://www.youtube.com/channel/UCM4kmLaBF10J2gsezFz-zxA/featured>
- [34 Youtube Κανάλι: PRDV Entertainment. [Online].
] <https://www.youtube.com/user/rebulerom3/featured>
- [35 Youtube Κανάλι: Joseph Whitworth. [Online].
] <https://www.youtube.com/user/NewChanTutorials/featured>
- [36 Youtube Κανάλι: Basement Games. [Online]. <https://www.youtube.com/user/DwithC/videos>
]
- [37 Youtube Κανάλι: Jason Harris. [Online]. <https://www.youtube.com/user/1337Reloaded/feed>
]
- [38 Youtube Κανάλι: Peter L. Newton. [Online].
] <https://www.youtube.com/user/PeterLNewton/featured>
- [39 Youtube Κανάλι: PyroDev. [Online].
] <https://www.youtube.com/channel/UCC8f6SxKJEIVvaRb7nF4Ayg/featured>
- [40 Youtube Κανάλι: MidnightInvader. [Online].
] <https://www.youtube.com/channel/UCI2hk17vpYE0yKC2f0PRNDA/featured>