

Τμήμα Μηχανικών Μουσικής Τεχνολογίας & Ακουστικής

ΤΕΙ Ρεθύμνου



«Σχεδιασμός και δημιουργία ηχοτοπίου τρισδιάστατου αλληλεπιδραστικού εικονικού περιβάλλοντος»

Ανάπτυξη τρισδιάστατου εικονικού περιβάλλοντος με χρήση μηχανής γραφικών. Μελέτη, σχεδιασμός και υλοποίηση ηχοτοπίου του εικονικού περιβάλλοντος.

ΦΟΙΤΗΤΗΣ: Γιάννης Πλέσσας

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Παναγιώτης Ζέρβας

Πρόλογος

Αφορμή για την παρακάτω πτυχιακή εργασία υπήρξε η ενασχόληση και το προσωπικό μου ενδιαφέρον για τον ηχητικό σχεδιασμό επάνω στα τρισδιάστατα αλληλεπιδραστικά περιβάλλοντα και τα ηχοτοπία που μεταβάλλονται σε πραγματικό χρόνο, όπως αυτά των σημερινών 3D βιντεοπαιχνιδιών.

Η εργασία έχει ως σκοπό να ασχοληθεί με όλα τα στάδια της παραγωγής μιας τρισδιάστατης εφαρμογής, με την βαρύτητα να δίνεται κυρίως στο ηχητικό μέρος. Σκοπός είναι, η παρακάτω εργασία να αναδείξει όλες τις δυσκολίες της παραγωγής και την ποικιλία γνώσεων που απαιτούνται από τις σημερινές ομάδες ανάπτυξης βιντεοπαιχνιδιών, πριν καταλήξει μια τέτοια εφαρμογή σε τελικό στάδιο.

Απαραίτητα εργαλεία για την ολοκλήρωση αυτού του εγχειρήματος ήταν τα δύο βασικά λογισμικά με τα οποία επέλεξα να πειραματιστώ, όπως το FMOD Designer 2010 για τον ηχητικό σχεδιασμό και η μηχανή παιχνιδιών Cry Engine 3 SDK (ή «μηχανή γραφικών», όπως αναφέρεται πολλές φορές) για την δημιουργία του τρισδιάστατου περιβάλλοντος. Αξίζει να σημειωθεί ότι, τόσο τα δύο αυτά εργαλεία όσο και άλλα ανταγωνιστικά, όπως η Unreal Engine 4 ή η Unity Engine, διατίθενται δωρεάν από τις ιστοσελίδες των εταιρειών τους και μπορούν να χρησιμοποιηθούν ελεύθερα από τον καθένα για εκπαιδευτικούς σκοπούς και πειραματισμό.

Σκοπός επίσης της εργασίας αυτής ήταν να μπορέσω να εξοικειωθώ με τα περιβάλλοντα εργασίας των δύο συγκριμένων λογισμικών, καθώς αποτελούν εμπορικά προγράμματα που χρησιμοποιούνται από πολλές μεγάλες εταιρείες, κάτι που αποτελεί για εμένα σημαντική εμπειρία γνώσεων για το μέλλον. Το περιβάλλον εργασίας τους είναι συνήθως παρόμοιο με άλλα ανταγωνιστικά λογισμικά, κάτι που αποτέλεσε άλλο ένα κίνητρο για την εκμάθηση των δυνατοτήτων τους και την εξοικείωση με το user interface τους.

Ευρετήριο Περιεχομένων

Πρόλογος	1
Περίληψη Περιεχομένου	6
Ευρετήριο εικόνων	4

Κεφάλαιο 1: Η χρήση της μουσικής και του sound design στα βιντεοπαιχνίδια και σημασία του ήχου στα ηχοτοπία

Γιατί είναι σημαντικός ο ήχος στα βιντεοπαιχνίδια;	7
Το ξεκίνημα για το sound design στις κονσόλες	8
NES (Nintendo Entertainment System).....	8
Τα 16-bit συστήματα και ο θάνατος των arcade	9
SNES (Super Nintendo Entertainment System).....	10
Οι προσωπικοί υπολογιστές και η έλευση των MIDI.....	10
Η επανάσταση του CD-ROM στον ηχητικό σχεδιασμό	11
Στυλ μουσικής και σημαντικές συνθέσεις	12
Τα ηχοτοπία στον πραγματικό κόσμο.....	14
Παρουσίαση των ηχοτοπιών του χάρτη	16

Κεφάλαιο 2: Εισαγωγή στην Cry Engine 3. Δημιουργώντας ένα τρισδιάστατο περιβάλλον με μια μηχανή γραφικών.....

Τι είναι μια Game Engine;	21
Γιατί είναι σημαντικές οι μηχανές παιχνιδιών;	22
Ορισμένες δημοφιλείς μηχανές παιχνιδιών.....	23
Η επιλογή της μηχανής γραφικών Cry Engine 3	25
Προετοιμασία πριν τη δημιουργία και παρατηρήσεις	25
Δημιουργία του terrain.....	27
Επεξήγηση των εργαλείων του Terrain Editor:.....	28
Η Επιλογή των textures	30
Ποικιλία στο χρωματισμό των Texture εδάφους	32
Δημιουργία δρόμων και μονοπατιών	33
Το πρόβλημα της περιπλάνησης: Σχεδιασμός επιπλέον δρόμων	34
Δημιουργία ποταμών.....	34
Τροποποίηση του terrain για τη δημιουργία των καταρραχτών.....	35
Δημιουργία της σπηλιάς	36
Σχεδιασμός της βλάστησης.....	37
Μικρή βλάστηση: γρασίδι και χόρτα	39
Δημιουργία της λίμνης.....	40
Τοποθέτηση των αντικειμένων.....	41
Το εργαλείο των Flow Graph	42

Spawn point: Το σημείο εκκίνησης του χάρτη.....	43
Area Triggers Objects.....	45
Προσθήκη αντικειμένου μεταφοράς του χρήστη.....	46
Εισαγωγή των οπτικών εφέ.....	48
Κεφάλαιο 3: Ανάπτυξη του Ηχητικού Σχεδιασμού μέσω του FMOD Designer.....	56
Εισαγωγή στο FMOD.....	56
Προβλήματα και λύσεις με το μέγεθος των ήχων.....	57
Το περιβάλλον εργασίας του FMOD Designer.....	58
Δημιουργία ήχων καταρράκτη.....	61
Δημιουργία ήχων εστιών φωτιάς.....	64
Ηχητικό εφέ πτώσης βράχων.....	66
Ambient ήχοι περιβάλλοντος (ζούγκλα, σπηλιά, παραλία, κ.α.).....	66
Sound Defs: Ένα σημαντικό εργαλείο του FMOD.....	67
Ambient ήχοι περιβάλλοντος (συνέχεια...).....	68
Δημιουργία ήχων για τα ποτάμια.....	69
Δημιουργία του ηχοτοπίου της σπηλιάς.....	71
Δημιουργία ηχοτοπίου θάλασσας και παραλίας.....	72
Δημιουργία ηχοτοπίου ομίχλης.....	74
Δημιουργία ήχων μιας τροπικής καταιγίδας.....	75
Δημιουργία ηχοτοπίου σεισμικής δόνησης.....	76
Προσθήκη ηχητικών εφέ για κλειστό τούνελ.....	77
Επιπλέον ήχοι ζώων.....	78
Κεφάλαιο 4: Τοποθέτηση των ήχων μέσα στη Cry Engine 3.....	80
Τοποθέτηση των Ambient ήχων.....	81
Τοποθέτηση ήχων καταρρακτών.....	84
Τοποθέτηση μεμονωμένων ήχων.....	85
Παρατηρήσεις και συμπεράσματα για την εφαρμογή.....	87
Η Export διαδικασία του χάρτη στη μηχανή.....	87
Βιβλιογραφία και πηγές της εργασίας.....	89
Βιβλιοθήκη ηχητικών αρχείων που χρησιμοποιήθηκε.....	90

Ευρετήριο εικόνων

Εικόνες 1^{ου} Κεφαλαίου:

Εικόνα 1.1: “Super Mario Bros.” – Nintendo NES, 1985

Εικόνα 1.2: Η κονσόλα NES – “Nintendo Entertainment System”, 1983

Εικόνα 1.3: Η κονσόλα “SEGA Genesis”, 1988

Εικόνα 1.4: Η κονσόλα SNES της Nintendo – “Super Nintendo Entertainment System”, 1990

Εικόνα 1.5: MIDI Πληκτρολόγια

Εικόνα 1.6: Η κονσόλα “PlayStation” της SONY, 1994

Εικόνα 1.7: Οπτικός δίσκος CD-ROM

Εικόνα 1.8: Η κονσόλα “Nintendo 64”, 1996

Εικόνα 1.9: Ο συνθέτης των παιχνιδιών Final Fantasy, “Nobuo Uematsu”

Εικόνα 1.10: Σχέδιο επεξήγησης των sonotopes

Εικόνα 1.11: Σχέδιο επεξήγησης των ηχοτοπίων

Εικόνα 1.12: Πανοραμική παρουσίαση των ηχοτοπίων της εργασίας

Εικόνες 2^{ου} Κεφαλαίου:

Εικόνα 2.1: Το logo (λογότυπο) της Cry Engine, ιδιοκτησία της Crytek

Εικόνα 2.2: Ο ορισμός του μεγέθους του terrain (χάρτη)

Εικόνα 2.3: Καθορισμός του ύψους του ωκεανού στον χάρτη

Εικόνα 2.4: Σχεδιασμός της επιφάνειας της στεριάς, με τη χρήση εργαλείου θούρτσας

Εικόνα 2.5: Συνολικά η διαμόρφωση του εδάφους από ψηλά

Εικόνα 2.6: Ορισμός εφαρμογής επιφανειών (texture) στη στεριά, ανάλογα με το ύψος του εδάφους

Εικόνα 2.7: Συνολικά οι επιφάνειες (textures) που έχουν χρησιμοποιηθεί στο τοπίο

Εικόνα 2.8: Επιπλέον χρωματισμός των επιφανειών για περισσότερη αληθοφάνεια

Εικόνα 2.9: Επίπεδη διαμόρφωση του εδάφους, για την κατασκευή δρόμων και μονοπατιών

Εικόνα 2.10: Ορισμός του σχήματος του μονοπατιού, με χρήση ειδικού εργαλείου

Εικόνα 2.11: Κατασκευή ποταμίων, με ειδικό εργαλείο της Cry Engine

Εικόνα 2.12: Τοποθέτηση βράχων για τη δημιουργία των σημείων των καταρρακτών

Εικόνα 2.13: Τοποθέτηση βράχων για τη δημιουργία της σπηλιάς

Εικόνα 2.14: Εισαγωγή των assets της βλάστησης για τη δημιουργία του ειδικού εργαλείου «θούρτσας»

Εικόνα 2.15: Οι ρυθμίσεις των asset της βλάστησης για τον χώρο της ζούγκλας

Εικόνα 2.16: Προσθήκη γρασιδιού και οι βασικές ρυθμίσεις για το εργαλείο της «θούρτσας»

Εικόνα 2.17: Τα assets (αρχεία) που χρησιμοποιήθηκαν συνολικά για το γρασίδι

Εικόνα 2.18: Ορισμός του χώρου της λίμνης και τοποθέτηση της επιφάνειας του νερού

Εικόνα 2.19: Τοποθέτηση διαφόρων αντικειμένων στον χάρτη, με το «χέρι»

Εικόνα 2.20: Τοποθέτηση αντικειμένου Spawn Point και επιλογή εισαγωγής Flow Graph σε αυτό

Εικόνα 2.21: Δημιουργία Flow Graph για το αρχικό Spawn Point του τοπίου

Εικόνα 2.22: Δημιουργία Flow Graph για τη συμπεριφορά του εφέ της περιοχής της βροχής

Εικόνα 2.23: Δημιουργία Flow Graph για τον προγραμματισμό της συσκευής μεταφοράς του χρήστη

Εικόνα 2.24: Οι ρυθμίσεις του OneShot ήχου, με τη χρήση του FMOD, για τη συσκευή μεταφοράς

Εικόνα 2.25: Τοποθέτηση εφέ ομίχλης και οι ρυθμίσεις του αντίστοιχου object

Εικόνα 2.26: Ορισμός της περιοχής της βροχής με τη χρήση των Shapes

Εικόνα 2.27: Τοποθέτηση των objects για το εφέ της πτώσης των βράχων και οι ρυθμίσεις του αντίστοιχου AreaTrigger Object

Εικόνα 2.28: Η δημιουργία του Flow Graph για τον προγραμματισμό του εφέ της πτώσης βράχων στη περιοχή της σπηλιάς

Εικόνα 2.29: Ορισμός της περιοχής του σεισμού και οι ρυθμίσεις του «Shake» εφέ

Εικόνα 2.30: Δημιουργία Flow Graph για τη σύνδεση του εφέ Shake και του ήχου, για την περιοχή του σεισμού

Εικόνα 2.31: Τοποθέτηση εφέ νερού για δημιουργία καταρράκτη

Εικόνες 3^{ου} Κεφαλαίου:

Εικόνα 3.1: Ορισμός των φακέλων εισαγωγής και δημιουργίας (εξαγωγής) αρχείων στο FMOD

Εικόνα 3.2: Παράδειγμα της διάταξης του interface του FMOD, για τα Event Groups και των Multi και Simple Track Events

Εικόνα 3.3: Το interface ενός SimpleTrack Event Editor στο FMOD Designer 2010

Εικόνα 3.4: Το interface ενός MultiTrack Event Editor στο FMOD

Εικόνα 3.5: Παράδειγμα ενός Low-Pass φίλτρου, με τη βοήθεια του προγράμματος Audacity

Εικόνα 3.6: Τοποθέτηση των Volume Effect στα layers των καταρρακτών

Εικόνα 3.7: Οι επιλογές για τον ορισμό των παραμέτρων, όπως αυτής της απόστασης

Εικόνα 3.8: Τοποθέτηση SoundEventSpot αντικειμένου για την πηγή εστίας φωτιάς

Εικόνα 3.9: Αλλαγή του pitch για τη μία εστία φωτιάς, για την απόδοση πιο χαμηλού ηχοχρώματος

Εικόνα 3.10: Ηχητικά αρχεία και παράμετροι για το ηχητικό αποτέλεσμα πτώσης βράχων

Εικόνα 3.11: Ορισμός των Templates για τα 3D Ambiences

Εικόνα 3.12: Το interface των Sound Def στο FMOD και οι ρυθμίσεις για το παράδειγμα των πουλιών της ορεινής περιοχής

Εικόνα 3.13: Συνολικά τα 4 layers για το 3D Ambient περιβάλλον της ζούγκλας

Εικόνα 3.14: Δημιουργία ενός Low-Pass φίλτρου για ένα ηχητικό αρχείο ενός ποταμιού

Εικόνα 3.15: Η επιλογή των ρυθμίσεων για ένα από τα 3 ποτάμια του χάρτη

Εικόνα 3.16: Δημιουργία των layers και ρυθμίσεις για το ηχητικό περιβάλλον της σπηλιάς

Εικόνα 3.17: Επιλογή ηχητικού αποσπάσματος 25 δευτερολέπτων για τη δημιουργία αρχείου ήχου για την περιοχή της θάλασσας

Εικόνα 3.18: Τοποθέτηση 4 αρχείων με «φράσεις» γλάρων, για τη δημιουργία Sound Def για την περιοχή της θάλασσας

Εικόνα 3.19: Οι ρυθμίσεις των 2 layer στον Multitrack Editor, για τη δημιουργία του ηχητικού περιβάλλοντος της θάλασσας

Εικόνα 3.20: Οι ρυθμίσεις των τεσσάρων layer στον Multitrack Editor, για τη δημιουργία του ηχοτοπίου της ομίχλης

Εικόνα 3.21: Δημιουργία ήχων βροχής, σε συνδυασμό με κεραυνούς, σε 2 ξεχωριστά layers

Εικόνα 3.22: Δημιουργία ήχων ζώων σε αναστάτωση, από τα ίδια αρχεία ήχου, προκειμένου να εξομοιωθεί το ηχητικό αποτέλεσμα ενός σεισμού

Εικόνα 3.23: Χρήση Occlusion εφέ για το ηχοτοπίο του πέτρινου τούνελ

Εικόνα 3.24: Δημιουργία ενός Simpletrack Event διάφορους ήχους εντόμων

Εικόνα 3.25: Δημιουργία Simpletrack Event για τη δημιουργία randomize ήχων βατράχων

Εικόνες 4^{ου} Κεφαλαίου:

Εικόνα 4.1: Χρήση του εργαλείου “Shape”, για τον ορισμό των διαφορετικών ηχητικών τοπίων

Εικόνα 4.2: Οι παράμετροι του εργαλείου “Shape”

Εικόνα 4.3: Σύνδεση ενός AmbientSound Object με μια περιοχή (Shape)

Εικόνα 4.4: Εισαγωγή των αρχείων ήχων του FMOD στο Sound Object της Cry Engine 3

Εικόνα 4.5: Δημιουργία των σχημάτων (shape) για τα ηχητικά τοπία των 3 καταρρακτών

Εικόνα 4.6: Τοποθέτηση ενός SoundEventSpot Object στο κέντρο, για τον ορισμό της ηχητικής προέλευσης ενός καταρράκτη

Εικόνα 4.7: Τοποθέτηση ενός SoundEventSpot, για τον ορισμό του σημείο ηχητικής προέλευσης των ήχων των βατράχων, από την επιφάνεια της λίμνης

Περιληπτική επεξήγηση περιεχομένου

Η παρακάτω εργασία χωρίζεται σε τέσσερα κεφάλαια. Στο πρώτο, αναλύεται πρώτα το ιστορικό κομμάτι και γιατί το sound design είναι σημαντικό στα βιντεοπαιχνίδια, αλλά και τι συναντάμε σε ένα φυσικό ηχοτοπίο σε θεωρητική βάση, κάτι που θα μας βοηθήσει στη συνέχεια να συνθέσουμε τα δικά μας, τεχνητά ηχοτοπία. Τα υπόλοιπα τρία κεφάλαια αφορούν στο πρακτικό μέρος.

Στο πρακτικό μέρος σκοπός είναι να αναδειχθούν τα στάδια εξέλιξης και δημιουργίας μιας ολοκληρωμένης τρισδιάστατης αλληλεπιδραστικής εφαρμογής (ενός videogame δηλαδή). Η διαδικασία ξεκινάει από τη δημιουργία του τρισδιάστατου περιβάλλοντος και της τοποθέτησης των εφέ και στη συνέχεια παρουσιάζεται η διαδικασία δημιουργίας των ήχων στο FMOD Designer. Τέλος, στο τελευταίο μέρος οι ήχοι τοποθετούνται στο αλληλεπιδραστικό περιβάλλον, ώστε να δημιουργηθεί το τελικό προϊόν.

Πιο αναλυτικά, στο δεύτερο κεφάλαιο θα παρουσιαστεί κυρίως η διαδικασία δημιουργίας του terrain με την Cry Engine 3. Συγκεκριμένα θα δούμε πως μπορεί να δημιουργηθεί ένα ανοικτό περιβάλλον με λόφους και βουνά, ενώ στην συνέχεια θα τοποθετηθεί η βλάστηση, οι λίμνες και οι καταρράκτες, πολλά άλλα αντικείμενα, αλλά και εφέ που θα ενεργοποιούνται με τη κίνηση του χρήστη.

Το τρίτο κεφάλαιο είναι και το πιο ουσιαστικό για το Sound Design, καθώς εκεί ορίζουμε τη συμπεριφορά των ήχων μέσα από το FMOD Designer, πριν τους συνδέσουμε με τη μηχανή γραφικών, έτσι ώστε να δημιουργηθεί αυτή η ζωντανή αλληλεπίδραση. Τους ήχους τους έχουμε συγκεντρώσει κυρίως από δωρεάν Online βιβλιοθήκες, ενώ όσοι δεν ήταν σε WAV αρχείο ή ήταν πολύ μεγάλοι σε μέγεθος, έχουν επεξεργαστεί κατάλληλα με τη βοήθεια του δωρεάν προγράμματος Audacity.

Στο τελευταίο κεφάλαιο τοποθετούμε τα αρχεία του FMOD μέσα στο τρισδιάστατο terrain, αφού πρώτα ορίσουμε τις περιοχές που θέλουμε να ακούγονται οι διάφοροι ήχοι. Χρησιμοποιούμε τόσο ambient ήχους σε μεγάλες περιοχές του τοπίου, όσο και πηγές ήχου με συγκεκριμένη ηχητική πηγή (σημείο) στο περιβάλλον. Τέλος, εξάγουμε την εφαρμογή από τον Editor ώστε να είναι έτοιμη για χρήση μέσα από το Game SDK application που περιλαμβάνεται στα αρχεία της Cry Engine 3.

Κεφάλαιο 1: Η χρήση της μουσικής και του sound design στα βιντεοπαιχνίδια και σημασία του ήχου στα ηχοτοπία

Γιατί είναι σημαντικός ο ήχος στα βιντεοπαιχνίδια;

Οι παίκτες έχουν ενεργό ρόλο στον τρόπο με τον οποίο ενεργοποιείται ένα ηχητικό event μέσα στο παιχνίδι, το οποίο μπορεί να συμπεριλαμβάνει διαλόγους, ambient ήχους, ηχητικά εφέ και μουσικά events. Παρ' όλο που στα βιντεοπαιχνίδια ο παίκτης είναι ο υποδοχέας του τελικού ακουστικού αποτελέσματος, έχει ταυτόχρονα και τον ρόλο του «δημιουργού» αυτών των σημάτων, κάτι που προκύπτει από τις ίδιες τους τις πράξεις μέσα στο τρισδιάστατο περιβάλλον.

Υπάρχουν αρκετές ομοιότητες ανάμεσα στις ταινίες και τα βιντεοπαιχνίδια. Πολλές φορές τα παιχνίδια περιέχουν τα λεγόμενα Cinematics, δηλαδή full motion videos (FMV) ή σκηνές που δεν επιτρέπεται κανενός είδους αλληλεπίδραση από τον παίκτη. Η δημιουργία του ήχου για αυτές τις σκηνές παρουσιάζει πολλές ομοιότητες με το production του ήχου για τις ταινίες.

Ποιοι εργάζονται για το ηχητικό αποτέλεσμα:

Συνθέτες που έχουν την ευθύνη για τη δημιουργία των μουσικών κομματιών. Sound Designers που δημιουργούν και συνθέτουν μη-μουσικούς ήχους. Ηθοποιοί που δανείζουν τις φωνές τους, για τις ανάγκες των διαλόγων του παιχνιδιού. Τέλος, audio programmers, που προγραμματίζουν το πώς όλα αυτά τα στοιχεία θα λειτουργήσουν μαζί μέσα στο παιχνίδι.

Ο ήχος είναι σημαντικός επειδή:

Ο ήχος συμπληρώνει την γραφική απεικόνιση στο προσδιορισμό του κλίματος και του θέματος του:

- Μπορεί να υποστηρίξει διάφορα συμβάντα στο παιχνίδι
 - μπορεί να δημιουργήσει ατμόσφαιρα
- Τα ηχητικά εφέ κάνουν τον κόσμο να μοιάζει περισσότερο ρεαλιστικός
 - συμπληρώνει την εικόνα ως μέσο αντίληψης του εικονικού κόσμου
- Δρουν σαν ανάδραση στις πράξεις του παίκτη συνδέοντας τον με το περιβάλλον στο οποίο βρίσκεται.
 - η εικόνα μεταφέρει το τι γίνεται στο κόσμο, ο ήχος μεταφέρει το τι σημαίνει αυτό
- Ο ήχος αποτελεί και μηχανισμό (στοιχείο) του παιχνιδιού
- Μεγάλης σημασίας είναι επιπλέον η μουσική υπόκρουση σε ένα παιχνίδι

Το ξεκίνημα για το sound design στις κονσόλες

Ο προγραμματισμός και η δημιουργία του ήχου στα πρώτα μηχανήματα ήταν δύσκολη υπόθεση, λόγω της συνεχής «μάχης» για την μείωση του μεγέθους των sound files και της λίγης χωρητικότητας της εποχής. Η συνεχόμενη μουσική (loops) παρουσιάστηκε για πρώτη φορά το 1978 και θεωρήθηκε ως μια αρκετά υποσχόμενη τεχνική, που θα βελτίωνε τα βιντεοπαιχνίδια. Το Space Invaders (Midway, 1978) ήταν το πρώτο παιχνίδι που χρησιμοποίησε συνεχόμενη μουσική, έχοντας ένα μουσικό θέμα τεσσάρων τόνων να επαναλαμβάνεται και μάλιστα να αλλάζει σε ρυθμό, ανάλογα με τα τεκταινόμενα της οθόνης. Ωστόσο, τα επόμενα χρόνια η μουσική στα βιντεοπαιχνίδια δεν εξελίχθηκε ιδιαίτερα, αφού τα πρώτα μηχανήματα ήταν δύσκολα στον προγραμματισμό και η μουσική καταναλώνει πολύ χρόνο από την ανάπτυξη ενός τίτλου.



Εικόνα 1.1: "Super Mario Bros." – Nintendo NES, 1985

Όπως αναφέρει και ο Hirokazu Tanaka, συνθέτης τότε στην Nintendo, η μουσική των arcade της εποχής σχεδιαζόταν κομμάτι-κομμάτι και πολλές φορές δημιουργούνταν κατευθείαν στην CPU port, χρησιμοποιώντας «0» και «1» και στέλνοντας το σήμα του ήχου στην έξοδο. Η μνήμη των μηχανημάτων της εποχής ήταν περίπου 1K με 2K, κάτι που σήμαινε ότι οι δημιουργοί έπρεπε να φτιάχνουν μόνοι τους τα εργαλεία. Το 1980 οι κατασκευαστές των arcade μηχανημάτων συμπεριέλαβαν dedicated sound chips ή programmable sound generators (PSGs) μέσα στα μηχανήματα, με αποτέλεσμα η μουσική στο background να γίνει πιο τονική και τα ηχητικά εφέ πιο περίτεχνα. Ένα από τα πρώτα δείγματα αυτής της τεχνολογίας, που έκανε χρήση των digital-to-analog converters, ήταν το Rally X (Namco/ Midway 1980). Αργότερα, προστέθηκαν περισσότερα chip, ώστε να διαχειρίζονται ξεχωριστά τα ηχητικά εφέ και την φωνή. Αυτό είχε ως αποτέλεσμα το κεντρικό sound chip να «ελευθερωθεί» και να μπορέσει να παράγει πιο σύνθετο ήχο, όπως το Discs of Tron (Atari, 1983), το οποίο ήταν ένα από τα πρώτα βιντεοπαιχνίδια που χρησιμοποίησε στερεοφωνικό ήχο.

NES (Nintendo Entertainment System)

Το sound chip του NES σχεδιάστηκε από τον συνθέτη Yukio Kaneoka και χρησιμοποιούσε ένα custom PSG chip 5-καναλιών. Δύο κανάλια χρησιμοποιούσαν αποκλειστικά pulse-wave ήχους, συνολικής έκτασης 8 οκτάβων, ένα ακόμη pulse-wave κανάλι ικανό για frequency sweep λειτουργία (ιδανικό για «γλίστρημα» τόνων, όπως σε UFOs ή laser εφέ), ένα κανάλι για τριγωνικά κύματα, το οποίο ήταν μια οκτάβα χαμηλότερα από τα παλμικά κανάλια. Το τέταρτο κανάλι ήταν αφιερωμένο στον λευκό θόρυβο και, τέλος, το πέμπτο κανάλι αποτελούσε τον sampler ή αλλιώς, το Delta Modulation Channel (DMC).



Εικόνα 1.2: Η κονσόλα NES – "Nintendo Entertainment System", 1983

Παρόλο που αυτό το sound design σύστημα ήταν σαφώς το πιο ανεπτυγμένο για την εποχή, η μίξη αυτή καθ' αυτή δεν υφίστανται ακόμη σαν διαδικασία στα παιχνίδια, με αποτέλεσμα πολλοί ήχοι, εφέ και μουσική να αλληλεπικαλύπτονται συχνά. Η αρχή, λοιπόν, για το sound design όπως το ξέρουμε σήμερα έγινε στην εποχή του NES. Οι πρώτοι ηχολήπτες έπρεπε πολλές φορές να βρίσκουν ευφάνταστους τρόπους για να καλύπτουν τις μειωμένες δυνατότητες της κονσόλες. Για παράδειγμα, όταν χρησιμοποιούνται 3 κανάλια για την μουσική θα ήταν αδύνατον να μπορούσε να χρησιμοποιηθεί ένα κομμάτι με 4 φωνές. Οι συνθέτες, όμως, βρήκαν έναν τρόπο για να κάνουν την μία φωνή να παίζει «αρπάζ» τόσο γρήγορα, ώστε το αυτί να νομίζει ότι ακούει μια σταθερή φωνή.

Αυτή η τεχνική, για παράδειγμα, είχε ως αποτέλεσμα όχι μόνο να κάνει εφικτή την χρήση συγχοριών, αλλά «ελευθέρωσε» και δύο ακόμη κανάλια που μπορούσαν να χρησιμοποιηθούν κάπου αλλού. Ενδιαφέρουσα επίσης τεχνική για την εποχή ήταν αυτή του Delay ή, όπως το γνωρίζουμε σήμερα, της «ηχού». Οι σχεδιαστές, αν ήθελαν να δημιουργήσουν ηχώ σε έναν ήχο, έπρεπε να προσθέσουν επιπλέον επαναλήψεις του ήχου, οι οποίες θα ακουγόntonταν σε ελάχιστους χρόνους millisecond πιο μετά από τον αρχικό ήχο. Μπορεί οι δυνατότητες να ήταν περιορισμένες για την εποχή, όμως ήταν αυτός ο περιορισμός που ώθησε, τελικώς, σε υψηλά επίπεδα την φαντασία και την δημιουργικότητα των sound designers.

Στην εποχή των πρώτων συστημάτων βιντεοπαιχνιδιών (consoles) οι συνθέτες έπρεπε να είχαν και μεγάλη γνώση προγραμματισμού, ενώ η συνεργασία ανάμεσα στους προγραμματιστές, τους game designers και τους sound designers, έπρεπε να ήταν στενή. Ακριβώς επειδή οι sound designers έπρεπε να συνεργάζονται στενά με όλα τα τμήματα του παιχνιδιού, αυτό σήμαινε ότι οι ίδιοι θα έπρεπε να δουλεύουν μέσα σε αυτές τις εταιρίες. Στη σημερινή εποχή, είναι πολύ συνηθισμένο το sound design και η μουσική σύνθεση των κομματιών ενός παιχνιδιού να γίνεται εξ ολοκλήρου εκτός του Game studio (δηλαδή της ομάδας προγραμματισμού), ακόμα και σε συνεργασία με ανεξάρτητους συνθέτες. Επομένως, στα πρώτα βήματα των βιντεοπαιχνιδιών οι συνθέτες και οι sound designers ήταν καθιερωμένα μέλη της προγραμματιστικής ομάδας ανάπτυξης.

Τα 16-bit συστήματα και ο θάνατος των arcade

Μια από τις σημαντικότερες εξελίξεις στην 16-bit εποχή, ήταν η χρήση των Frequency Modulation (FM) στη σύνθεση. Η FM σύνθεση αναπτύχθηκε από τον John Chowning, στο Stanford University το 1967-68 και κυκλοφόρησε και βελτιώθηκε από την Yamaha, η οποία χρησιμοποίησε την μέθοδο αυτή σε διάφορα sound chips υπολογιστών, αλλά και στην σειρά DX των keyboard της. Οι πρώτες 16-bit κονσόλες άρχισαν να εισρέουν στις αγορές το 1987 και μια από τις δημοφιλέστερες, η οποία έκανε και χρήση FM τεχνικής, ήταν το Sega Genesis (Sega 1988). Το Genesis είχε ένα κατά πολύ ανώτερο hardware ήχου σε σχέση με την 8-bit κονσόλα της Nintendo (NES). Στο εσωτερικό του είχε ένα 3+1 PSG chip, ικανό να ελέγχει εφέ και μουσικά μέρη, μαζί με ένα Yamaha GM synthesis chip (YM



Εικόνα 1.3: Η κονσόλα "SEGA Genesis", 1988

2612), το οποία προσέφερε 6 κανάλια ψηφιακού στερεοφωνικού ήχου και ένα PCM 8-bit κανάλι (παρόμοιο με αυτό των DX27 και DX100 keyboard).

Ωστόσο, ο προγραμματισμός των δύο chip του Genesis δεν ήταν εύκολη υπόθεση, καθώς απαιτούσε χρήση γλώσσας assembly. Κάθε FM κανάλι είχε 4 operators (waveforms), κάθε ένα με συχνότητα και envelope για να τροποποιείται το αποτέλεσμα. Υπήρχαν 8 διαφορετικοί τύποι αλγόριθμου για αυτούς τους 4 operators. Το αποτέλεσμα του προγραμματισμού σε assembly μαζί με τους 8 αλγορίθμους σήμαινε ότι, πολλοί ήχοι του ίδιου τύπου χρησιμοποιούντουσαν ξανά και ξανά στα παιχνίδια.

SNES (Super Nintendo Entertainment System)

Με το Sega Genesis να βρίσκεται τεχνολογικά μπροστά από το NES, η Nintendo αποφασίζει να λανσάρει την δική της 16-bit κονσόλα. Το 1990 η Nintendo κυκλοφορεί το Super Famicom στην Ιαπωνία, το οποίο έγινε γνωστό στην Δύση με την ονομασία «SNES». Το SNES ήταν ανώτερο σε γραφικά και ήχο από το Genesis και παράλληλα εξελίχθηκε και σε εμπορική επιτυχία, σημειώνοντας 46εκ. πωλήσεις. Το sound module του SNES διέθετε διάφορα μέρη, με το πιο σημαντικό να είναι το Sony SPC-700 –ένας 8-bit co-processor, με έναν προσαρτημένο 16-bit (sony) digital signal επεξεργαστή, αλλά και έναν 16-bit stereo DAC.



Εικόνα 1.4: Η κονσόλα SNES της Nintendo – “Super Nintendo Entertainment System”, 1990

Το DSP ήταν ένα wavetable synthesizer που μπορούσε να υποστηρίξει μέχρι 8 στερεοφωνικά κανάλια, με προγραμματιζόμενη συχνότητα, ένταση, αλλά και επιλογές για εφέ όπως reverb, φίλτρα, rapping, envelope generators και MIDI όργανα. Αντίθετα με προηγούμενες τεχνικές σύνθεσης που χρησιμοποιούσαν στους ηλεκτρονικούς ήχους, το wavetable synthesis χρησιμοποιούσε έτοιμα ψηφιακά δείγματα οργάνων, σε συνδυασμό με βασικές κυματομορφές αναλογικών synth, με αποτέλεσμα το τελικό άκουσμα να είναι πολύ πιο ρεαλιστικό από αυτό της FM synthesis.

Οι προσωπικοί υπολογιστές και η έλευση των MIDI

Λόγω της ανάγκης των παιχνιδιών και των μουσικών για καλύτερο ήχο στα PC, στα μέσα του 1980 ξεκίνησαν να κυκλοφορούν στην αγορά οι πρώτες third-party FM κάρτες ήχου. Οι κάρτες αυτές απευθυνόντουσαν τόσο στον gamer όσο και στον composer: Διέθεταν joystick game port, που μπορούσε επίσης να χρησιμοποιηθεί και ως MIDI port, μέσα από ειδικούς αντάπτορες. Το πρωτόκολλο MIDI ορίστηκε το 1983, έτσι ώστε μουσικές συσκευές (synthesizers, keyboards, mixing desks, computers, κτλ) να μπορούν να χρησιμοποιούν ένα κοινό format. Στα MIDI μόνο το σήμα ήταν αυτό που μεταδιδόταν και όχι ο ήχος αυτός καθαυτός, με αποτέλεσμα τα αρχεία να είναι αρκετά μικρά σε μέγεθος.

Τα πλεονεκτήματα του MIDI ήταν πολύ σημαντικά για τους συνθέτες των βιντεοπαιχνιδιών: Διευκόλυνε τις συνθέσεις, αφού δεν χρειαζόντουσαν γνώσεις γλωσσών προγραμματισμού για κάτι τέτοιο και οι συνθέτες μπορούσαν να γράψουν απευθείας τη μουσική τους, χρησιμοποιώντας μουσικά keyboards. Ο μικρός όγκος των αρχείων ήταν επίσης ένα σημαντικό πλεονέκτημα των MIDI για τις gaming κονσόλες της εποχής, αφού δεν επιβάρυνε ιδιαίτερα την περιορισμένη RAM αυτών των συστημάτων. Ωστόσο τα MIDI διέθεταν και ορισμένα μειονεκτήματα.



Εικόνα 1.5: MIDI Πληκτρολόγια

Το πιο σημαντικό ήταν η περιορισμένη ποικιλία οργάνων, αφού υπήρχαν μόνο 128 προ-αποφασισμένα «όργανα», ενώ πολλές φορές ορισμένα από αυτά τα «όργανα» ήταν απλώς ήχοι πουλιών ή ελικοπτέρου, που δεν ήταν απαραίτητα στους μουσικούς, αλλά περισσότερο τους sound designers. Ένα ακόμη μειονέκτημα ήταν, το πόσο διαφορετικά ακουγόντουσαν οι ίδιοι ήχοι ανάμεσα σε διαφορετικές κάρτες ήχου. Ένας ήχος μπάσου ήταν εντελώς διαφορετικός από υπολογιστή σε υπολογιστή, με αποτέλεσμα οι συνθέσεις να μην ακούγονται πάντα όπως τις είχε φανταστεί ο δημιουργός. Ωστόσο, αυτό το τελευταίο μειονέκτημα εξαλείφθηκε στις gaming κονσόλες, αφού εκεί το hardware ήταν πάντα σταθερό και ο δημιουργός μπορούσε να ξέρει ακριβώς το πώς θα ακουστεί η σύνθεσή του.

Η επανάσταση του CD-ROM στον ηχητικό σχεδιασμό

PlayStation (Sony)

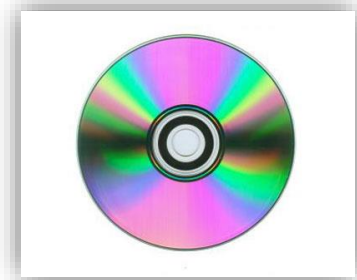
Η πραγματική επανάσταση για την βιομηχανία των βιντεοπαιχνιδιών και της χρήσης του ήχου, ήρθε το 1994 με το PlayStation της Sony. Αυτή η κονσόλα μπορούσε να παίξει 24 ηχογραφημένα κανάλια, κάτι που ήταν αρκετό ώστε να μπορέσουν για πρώτη φορά οι συνθέτες να αναπαράγουν όλα τα όργανα μιας συμφωνικής ορχήστρας. Δεν ήταν όμως μόνο αυτή η πραγματική βελτίωση. Η χρήση του CD-ROM ως αποθηκευτικό μέσον για τα παιχνίδια, είχε ως αποτέλεσμα οι δημιουργοί να μπορούν να αποθηκεύσουν πιο πιστούς ήχους, μεγαλύτερης χωρητικότητας, ενώ υπήρχε ακόμη και η δυνατότητα για Stereo σήματα. Εκεί όμως που διέφερε το PlayStation, ήταν στην αρχιτεκτονική, αφού πλέον οι σχεδιαστές είχαν τρεις επιλογές για την σύνθεση της μουσικής: Τα MIDI, τα MODS και το Red Book Audio.



Εικόνα 1.6: Η κονσόλα "PlayStation" της SONY, 1994

CD-ROM

Η τρίτη επιλογή που ήταν διαθέσιμη για το PlayStation είναι η λεγόμενη Red Book Audio, δηλαδή ψηφιακή μουσική ποιότητας CD. Σε αυτήν την μέθοδο γινόταν χρήση ακόμη μεγαλύτερου ποσοστού μνήμης και δύναμης του επεξεργαστή, αφού έπρεπε όλα τα αρχεία να είναι προ-ηχογραφημένα, ενώ σπαταλούσε και μεγαλύτερο χρόνο από το laser του driver. Επίσης αυτή η μέθοδος δεν επέτρεπε την αλληλεπίδραση μεταξύ του soundtrack και του παίκτη. Παρ' όλα αυτά, η ποιότητα ήταν η καλύτερη δυνατή και τα κομμάτια μπορούσαν να ηχογραφηθούν σε κανονικά studio, ακόμη και από συμφωνικές ορχήστρες και χορωδίες, για τις ανάγκες του παιχνιδιού.

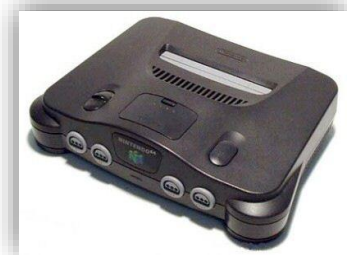


Εικόνα 1.7: Οπτικός δίσκος CD-ROM

Σημαντικό γεγονός ήταν ότι οι σχεδιαστές μπορούσαν να δημιουργούν συνθέσεις με παραπάνω από έναν τρόπους. Για παράδειγμα, το «One winged Angel», το μουσικό θέμα από το παιχνίδι Final Fantasy VII, χρησιμοποιούσε τόσο συνθετικούς ήχους, όσο και προ-ηχογραφημένη τεχνική (χορωδία).

Nintendo 64 (Nintendo)

Ο αντίπαλος του PlayStation, το Nintendo 64 της Nintendo, ήταν επίσης ικανό για αναπαραγωγή 24 καναλιών ήχου, όμως όχι με δυναμικό τρόπο. Αυτό σήμαινε ότι, όταν ένα patch ήχου διανεμόταν σε ένα κανάλι του συστήματος, μετά αυτό ήταν πολύ δύσκολο να αλλάξει σε ένα διαφορετικό patch. Επιπλέον, το Nintendo 64 δεν επέτρεπε τον τρίτο τρόπο σύνθεσης (τον Redbook audio), καθώς δεν διέθετε CD-ROM για αποθηκευτικό μέσον.



Εικόνα 1.8: Η κονσόλα "Nintendo 64", 1996

Στυλ μουσικής και σημαντικές συνθέσεις

Στυλ μουσικής

Το στυλ της μουσικής στα video games δεν μπορεί να προσδιοριστεί με ακρίβεια, επειδή φέρει πολλαπλές επιρροές. Στυλ όπως: Gothic, classical, rock, baroque, Caribbean, Celtic, hip hop, heavy metal, new age, techno, trance, dance, jazz, ακόμη και πεντατονική Ιαπωνική μουσική, είναι μερικά style που χρησιμοποιούνται στα σημερινά βιντεοπαιχνίδια. Ακριβώς λόγω αυτής την μεγάλης γκάμας και την απόλυτης ελευθερίας ως προς την φαντασία της σύνθεσης, πολλοί συνθέτες θεωρούν αρκετά ελκυστικό το γράψιμο μουσικής για βιντεοπαιχνίδια. Στη βιομηχανία του κινηματογράφου πολλές συνθέσεις βρίσκονται συνήθως μέσα σε στενά πλαίσια, που πολλές φορές ορίζονται και

από την ίδια την ταινία. Η βιομηχανία των βιντεοπαιχνιδιών επιτρέπει σαφώς μεγαλύτερο πειραματισμό.

Διαφορές στο στυλ μουσικής στα βιντεοπαιχνίδια συναντάμε και μεταξύ των δύο διαφορετικών αγορών, δηλαδή της δυτικής και της ιαπωνικής αγοράς. Οι πρώτοι συνθέτες για βιντεοπαιχνίδια ήταν, αρχικά, από την Ιαπωνία, ακριβώς επειδή από εκεί ξεκίνησαν και όλες οι κονσόλες παιχνιδιών (όπως αναφέρθηκε προηγουμένως, οι πρώτοι συνθέτες πάντα αποτελούσαν μέρος της προγραμματιστικής ομάδας και δούλευαν μέσα στις εταιρίες), κάτι που σημαίνει ότι για ένα μεγάλο διάστημα οι Ιάπωνες συνθέτες είχαν το «μονοπώλιο». Με την εξέλιξη των κονσολών και με την ολοένα και μεγαλύτερη ευκολία με την οποία μπορούσε κανείς να συνθέσει μουσική για ένα βιντεοπαιχνίδι, το προφίλ του ανεξάρτητου sound designer ήταν τελικά αυτό που επικράτησε.

Το ενδιαφέρον ολοένα και περισσότερων δυτικών εταιριών για τη βιομηχανία των βιντεοπαιχνιδιών τα τελευταία 20 χρόνια, αλλά και η ενασχόληση πολλών δυτικών εταιρειών hardware με τον χώρο (Microsoft, apple κτλ), έχουν πλέον φέρει μια ισορροπία ανάμεσα στις συνθέσεις. Παρ' όλα αυτά, όταν πρόκειται για περιγραφική μουσική, θεωρείται ότι οι ιαπωνικές συνθέσεις είναι ανώτερες από τις δυτικές.

Σημαντικές συνθέσεις

Το 1985, το Super Mario Bros, ήταν το πρώτο βιντεοπαιχνίδι που χρησιμοποίησε «περιγραφική μουσική», με τον συνθέτη (Koji Kondo) να δημιουργεί διαφορετικά κομμάτια που λειτουργούσαν ως «υπόβαθρό» για κάθε περιοχή που επισκεπτόταν ο παίκτης. Κομμάτια όπως το κεντρικό θέμα του Super Mario Bros, είναι δύσκολο να τοποθετηθούν σε κάποιο στυλ/ κατηγορία μουσικής. Σαφώς μέσα σε αυτό υπάρχουν επιρροές από Jazz μουσική, όμως ο τονισμός του ρυθμού είναι πιο folkly και η μελωδία χαρούμενη και με σύντομες φράσεις. Το θέμα δεν θυμίζει ούτε Ιαπωνική μουσική, αλλά ούτε και καθαρά δυτικό κομμάτι –πρόκειται για ένα εντελώς καινούργιο είδος μουσικής.

Θεωρείται ότι αυτός ο τρόπος είναι κάτι συνηθισμένο για την κουλτούρα των Ιαπώνων συνθετών, οι οποίοι πολλές φορές επηρεάζονται από ξένα είδη και, μέσα από μια δική τους οπτική, τα μετατρέπουν σε κάτι εντελώς ξεχωριστό. Παράδειγμα σε αυτό αποτελούν τόσο το ίδιο το θέμα του Super Mario Bros, όσο και ο συνθέτης των παιχνιδιών Final Fantasy, Nobuo Uematsu. Ο ίδιος δεν είχε κάποια ιδιαίτερη μουσική εκπαίδευση τα πρώτα χρόνια της ζωής του, παρ' όλο που του άρεσε η μουσική από μικρή ηλικία και ουσιαστικά ήταν αυτοδίδακτος μουσικός. Ο Nabuo Uematsu έγραψε το πρώτο ολοκληρωμένο soundtrack για το πρώτο παιχνίδι της σειράς Final Fantasy της Square, το 1987 και η μουσική του είχε επίσης έντονα περιγραφικά στοιχεία.



Εικόνα 1.9: Ο συνθέτης των παιχνιδιών Final Fantasy, "Nobuo Uematsu"

Ο Uematsu ποτέ δεν έκρυψε ότι οι επιρροές του ήταν πάντα δυτικές. Στα 12 του ξεκίνησε να μαθαίνει πιάνο θέλοντας να μοιάσει στον Elton John, ενώ του δημιουργούσαν έμπνευση οι The Beatles, Emerson, Lake & Palmer, Simon & Garfunkel και οι progressive rock μπάντες. Το πιο έντονο παράδειγμα αυτής της έντονης επιρροής, η οποία «διαστρεβλώνεται» με έναν μοναδικό τρόπο από την ιαπωνική κουλτούρα, αποτέλεσε το θέμα του για το final Fantasy VII, «One winged Angel», για το οποίο έχει αναφέρει ότι εμπνεύστηκε από το «Purple Haze» του Jimi Hendrix.

Τα ηχοτοπία στον πραγματικό κόσμο

Τι είναι ένα ηχοτοπίο;

Ένα ηχοτοπίο μπορεί να ορισθεί με πολλούς τρόπους. Για πολλούς ηχοτοπίο σημαίνει ένα σύνολο ήχων που βρίσκονται τοποθετημένοι σε ένα συγκεκριμένο περιβάλλον. Τέτοιο θα μπορούσε να είναι από το ηχοτοπίο ενός πολυσύχναστου πεζοδρομίου, μέχρι ένα δωμάτιο ή ένα τροπικό δάσος. Ουσιαστικά όμως, ο ορισμός των ηχοτοπίων, προερχόμενος από την αγγλική λέξη «soundscapes» (“sound” + “scapes”), αποτελεί ένα παράδοξο, καθώς ένας ήχος μπορεί να ακροασθεί, αλλά όχι να εμφανισθεί οπτικά. Ένας ακόμη βάσιμος ορισμός των ηχοτοπίων είναι ότι πρόκειται για περιβάλλοντα στα οποία οι γεωφωνικοί, βιοφωνικοί και ανθρωποφωνικοί ήχοι αλληλεπικαλύπτονται. Όλα τα προηγούμενα έχουν μεγάλη εφαρμογή και στα τεχνητά ηχοτοπία, όπως αυτά των τρισδιάστατων εφαρμογών, στα οποία συνήθως προσπαθούμε να αναπαραστήσουμε –με τεχνητά μέσα αλλά με ρεαλιστικό τρόπο- ένα πραγματικό ηχοτοπίο.

Είναι πολύ συνηθισμένο σε σύγχρονες τρισδιάστατες εφαρμογές, όπως αυτές των video games, να συναντάμε ήχους και των τριών κατηγοριών (γεωφωνικοί, βιοφωνικοί, ανθρωποφωνικοί), με τη μόνη διαφορά ότι είναι στην κρίση του sound designer ποια κατηγορία επιθυμεί κάθε φορά να αναπτύξει περισσότερο ή να προβάλλει περισσότερο στο θεατή. Πρακτικά αυτό σημαίνει ότι, πολλές φορές σε μια τρισδιάστατη εφαρμογή μπορούν οι υπόλοιποι ήχοι να μειωθούν σε ένταση, όπως για παράδειγμα όταν θέλουμε να δώσουμε έμφαση σε έναν διάλογο ή να μεταφέρουμε κάποια αίσθηση στο θεατή (παίκτη), όπως αυτή του τρόμου ή της έκπληξης. Στα τεχνητά ηχοτοπία είναι επομένως στο χέρι του σχεδιαστή το πώς θα χρησιμοποιήσει αυτά τα εργαλεία, προκειμένου να πετύχει σκηνοθετικά το αισθητικό αποτέλεσμα που επιθυμεί.

Ένα ακόμη σημαντικό κοινό χαρακτηριστικό των ηχοτοπίων του πραγματικού κόσμου και αυτών μιας τρισδιάστατης εφαρμογής, έχει να κάνει με την οπτική αντίληψη του ακροατή σε σχέση με την αίσθηση της ακοής του. Πρόκειται για δύο αισθήσεις οι οποίες βρίσκονται συνέχεια σε σύγκρουση και πολύ συχνά η μία αναζητά από την άλλη την επιβεβαίωση μιας πληροφορίας. Αυτό μπορεί να χρησιμοποιηθεί πολλές φορές ως πλεονέκτημα από τους sound designers αυτών των ηχοτοπίων, καθώς δεν είναι πάντοτε απαραίτητο ένα οπτικοακουστικό αποτέλεσμα να επιβεβαιώνεται και στις δύο αισθήσεις (ακοή, όραση). Όπως χαρακτηριστικά είχε εξηγήσει και ο Stephan Schutze σε ομιλία του στο Freerplay 2011: «αν ακούσουμε έναν ήχο ελικοπτέρου να διασχίζει τον ουρανό, αλλά δεν καταφέρουμε εκείνη τη στιγμή να το εντοπίσουμε με το μάτι, αυτό δεν σημαίνει ότι το ελικόπτερο αυτό δεν είναι υπαρκτό». Αυτό είναι κάτι που ισχύει τόσο για τον πραγματικό

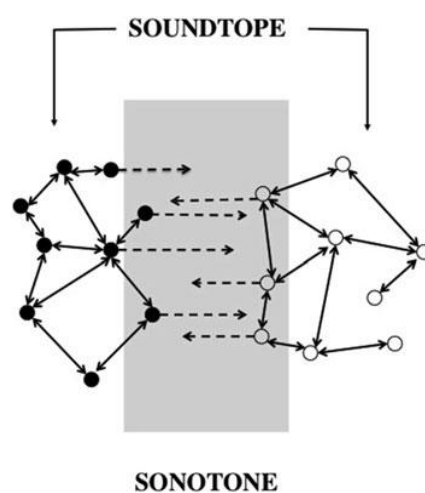
κόσμο όσο και για τα τρισδιάστατα ηχοτοπία και μάλιστα πολλές φορές μπορεί η χρήση τέτοιων παραδειγμάτων να είναι επιθυμητή. Το προηγούμενο το συναντάμε κυρίως σε περιπτώσεις όπου χρειάζεται να γίνει οικονομία των 3D assets και του μεγέθους μιας τρισδιάστατης εφαρμογής, καθώς το μόνο σίγουρο είναι ότι ένας ήχος «κοστίζει» υπολογιστικά για μια συσκευή πολύ λιγότερο από ένα τρισδιάστατο μοντέλο, όπως αυτό ενός ελικοπτέρου.

Είδη ήχων στα φυσικά -και όχι μόνο- περιβάλλοντα

Οι ήχοι που συναντάμε συνήθως στα ηχοτοπία κάθε είδους κατηγοριοποιούνται κυρίως σε τρεις κατηγορίες. Αυτοί είναι οι γεωφωνικοί ήχοι, οι βιοφωνικοί ήχοι και ανθρωποφωνικοί ήχοι. Στην πρώτη κατηγορία συναντάμε ήχους όπως από φυσικές πηγές, όπως τον ήχο του ανέμου, της θάλασσας, πιθανώς ηφαιστειών, τρεχούμενων ποταμιών, της βροχής, των κεραυνών, κατολισθήσεις κ.α.. Στους βιοφωνικούς ήχους συναντάμε κυρίως ήχους ζώων, όπως καλέσματα πουλιών, κραυγές ή φωνές κινδύνου θηλαστικών και γενικότερα όλους τους ήχους της πανίδας. Στους ανθρωποφωνικούς ήχους συναντάμε ήχους των οποίων η προέλευση είναι κάποια τεχνολογική πηγή, όπως κάποια συσκευή, κάποια μηχανή ή γενικότερα κάποια -αυτοματοποιημένη ή μη- πηγή δημιουργημένη από τον άνθρωπο. Σε αυτή την εργασία θα συναντήσουμε κυρίως ηχοτοπία από τις δύο πρώτες κατηγορίες, δηλαδή κυρίως ήχους περιβάλλοντος και ήχους ζώων.

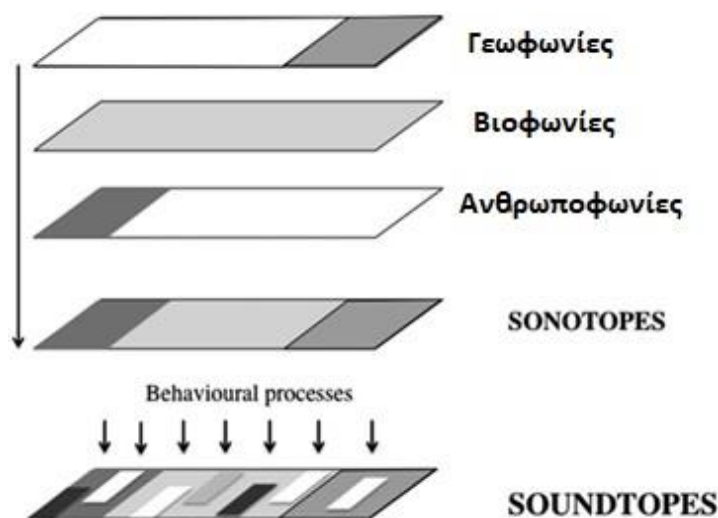
Αλληλεπίδραση ηχοτοπίων

Είναι πολλές φορές που σε ένα πραγματικό ηχοτοπία παρατηρούμε ένα σταθερό μοτίβο στους ήχους αλλά και έντονες αλλαγές. Οι αλλαγές μπορεί επίσης να είναι αναμενόμενες, τοποθετημένες σε πιο αργά μοτίβα, όπως η αλλαγή της ημέρας και της νύκτας ή ακόμα και των εποχών του έτους, αλλά και σε πιο άμεσα -απρόβλεπτα- γεγονότα, όπως κάποια ξαφνική νεροποντή ή ένας σεισμός. Το σύνολο των γεωφωνικών, βιοφωνικών και ανθρωποφωνικών ήχων σε μια περιοχή αποτελεί τελικά αυτό που ονομάζουμε «ηχοτοπία». Τα «sonotopes» αποτελούν τις περιοχές εκείνες που δύο ή περισσότερα διαφορετικά ηχοτοπία συνδέονται και αλληλεπικαλύπτονται. Επιπλέον, θα συναντήσουμε και τα soundtopes, τα οποία αποτελούν τις αλλαγές που παρατηρούνται στο σύνολο ενός ηχοτοπίου, όπως η αλλαγή του κελαϊδισμού των πουλιών κατά τη δύση του ηλίου ή οι κραυγές ζώων της ίδιας κοινότητας κατά την προειδοποίηση για κάποιον κίνδυνο.



Εικόνα 1.10: Σχέδιο επεξήγησης των sonotopes

Τέτοιες προγραμματισμένες ή μη αλλαγές, αποτέλεσαν σημαντικό παράγοντα στη δημιουργία της συγκεκριμένης εργασίας και έγινε προσπάθεια να τις αναπαραστήσουμε όσο πιο πιστά (ρεαλιστικά) γίνεται, σε πραγματικό χρόνο, στο τρισδιάστατο περιβάλλον κατά τη περιπλάνηση του ακροατή στο ηχοτοπία.



Εικόνα 1.11: Σχέδιο επεξήγησης των ηχοτοπίων

Παρουσίαση των ηχοτοπίων του χάρτη

Το τρισδιάστατο επίπεδο που έχουμε σχεδιάσει χωρίζεται σε πολλαπλά ηχοτοπία. Κάποια είναι χωρισμένα ανάλογα με το υψόμετρο, ενώ άλλα βρίσκονται στην πορεία του βασικού δρόμου που καθοδηγεί τον χρήστη, ενώ άλλα ηχοτοπία διαπερνούν άλλα μεγαλύτερα ηχοτοπία, αλλάζοντας ή προσθέτοντας στο συνολικό άκουσμα τους.

Το πρώτο ηχοτοπίο που θα συναντήσει ο χρήστης στην πορεία του είναι αυτό της θάλασσας και τη παραλία. Πρόκειται ουσιαστικά για δύο ηχοτοπία, που βρίσκονται στο ίδιο σημείο, με τη διαφορά ότι στο ένα ο ήχος των κυμάτων ακούγεται πιο δυνατά και από τη κατεύθυνση της θάλασσας, ενώ στο άλλο πιο απομακρυσμένα, όσο ο χρήστης απομακρύνεται από το νερό. Προκειμένου να διαχωρίσουμε τα ηχοτοπία το ένα από το άλλο έχουμε χρησιμοποιήσει το εργαλείο με τα *Shapes* (βλ. το αντίστοιχο κομμάτι στην Τοποθέτηση των ήχων), κάτι που τα κάνει να ξεχωρίζουν το ένα από το άλλο, επιτρέποντάς μας να παρακολουθήσουμε καλύτερα την έκταση του καθενός από την ισομετρική κάμερα.

Μια παρατήρηση που ισχύει για κάθε ηχοτοπίο, έχει να κάνει με την *fade-in, fade-out* απόσταση στην οποία ο ήχος χαμηλώνει ή αυξάνει όταν κινούμαστε μέσα ή έξω αντίστοιχα από τα σύνορα του κάθε ενός. Ανάλογα με τη φύση του κάθε ηχοτοπίου, η ρύθμιση αυτή μπορεί να έχει μια τιμή από 10 έως 20. Η ρύθμιση αυτή ονομάζεται «*OuterRadius*» και βρίσκεται στο *AmbientVolumeObject* της κάθε περιοχής. Για παράδειγμα, το περιβάλλον της ζούγκλας διαθέτει τιμή *OuterRadius=20*, καθώς επιθυμούμε οι ήχοι της ζούγκλας, ως πιο δυνατοί, να εμφανίζονται ακουστικά νωρίτερα από άλλους, καθώς αυτό θα γινόταν και στην πραγματικότητα. Αντίθετα, το περιβάλλον της θάλασσας διαθέτει *OuterRadius=15*, ενώ αυτό της σπηλιάς διαθέτει μικρότερο *OuterRadius=10*, καθώς επιθυμούμε η είσοδος

στο ηχοτοπίο της σπηλιάς να είναι πιο άμεση και να ταυτίζεται με το πραγματικό όριο που ορίζεται από την πέτρινη είσοδο της.

Το επόμενο ηχοτοπίο είναι αυτό της ζούγκλας κατά την ημερήσια εκδοχή της και χωρίς να υπάρχει κάποιο event που να αναστατώνει το τοπίο. Εδώ θα συναντήσουμε πλούσιο ηχόχρωμα από ζωηρά καλέσματα πουλιών (διαφόρων ειδών), κραυγές από μαϊμούδες πιο αραιά και τον ήχο του ανέμου να συμπληρώνει ηχητικά. Το επόμενο ηχοτοπίο που θα συναντήσει ο χρήστης είναι αυτό μιας πιο σκοτεινής έκδοσης της ίδια ζούγκλας, με πυκνή ομίχλη. Εδώ το ηχόχρωμα αλλάζει και χάνει την ζωηρή ημερήσια διάθεση του προηγούμενου ηχοτοπίου. Θα ακούσουμε να ξεχωρίζουν περισσότερο οι ήχοι των τριζονιών και των βατράχων, ενώ για πρώτη φορά θα ακούσει και ambient μουσική («*hollow_drone_001*», από την βιβλιοθήκη ήχων Sonniss), η οποία έχει ως σκοπό να δημιουργήσει μια αγωνία στο χρήστη και να τον προετοιμάσει για κάτι που πιθανώς θα ακολουθήσει στη πορεία. Ο ήχος του ανέμου βρίσκεται επίσης σε χαμηλή ένταση, ενώ το ίδιο ισχύει και για τον ήχο κάποιων μαϊμούδων, που ακούγονται σε μακρινή απόσταση.

Σε τέτοιου είδους περιπτώσεις, όπου ο σκηνοθέτης θέλει να μεταφέρει κάποια αίσθηση τρόμου στον τελικό χρήστη, ο ήχος κρίνεται ως ένα από τα πιο βασικά εργαλεία που έχει κανείς στη διάθεσή του, καθώς δημιουργεί μια ανησυχία στο χρήστη, «λέγοντάς» του νωρίτερα ότι πιθανώς να βρίσκεται σε κίνδυνο, αλλά ταυτόχρονα αφήνοντάς τον να κάνει ο ίδιος υποθέσεις στο μυαλό του, μέχρι η εικόνα να του επιβεβαιώσει ή όχι τελικά το γεγονός. Αυτή η τεχνική του λεγόμενου «build up» χρησιμοποιείται επιτυχημένα από πολλούς σκηνοθέτες σε πολλά βιντεοπαιχνίδια τρόμου, τα οποία ουσιαστικά καταφέρνουν να είναι επιτυχημένα ακριβώς επειδή πετυχαίνουν να «κτίσουν» αυτού του είδους την ανησυχία στο χρήστη, με αποτέλεσμα το όποιο γεγονός κινδύνου επακολουθήσει, να έχει ενισχυθεί προηγουμένως σημαντικά από το διάστημα «ησυχίας» στο sound design. Το επόμενο ηχοτοπίο που έχουμε επιλέξει έχει ως σκοπό να δημιουργήσει ένα τέτοιο event, χωρίς όμως να υπάρχει κάποιος πραγματικός κίνδυνος για τον χρήστη και δεν είναι άλλο από την απότομη τροπική βροχή υπό τη συνοδεία κεραυνών, η οποία ακολουθεί στο επόμενο μισό του ηχοτοπίου της ομίχλης.

Σε αυτό το σημείο θα ακούσουμε όλους τους προηγούμενους ήχους της ομίχλης μαζί με ένα επιπλέον ηχητικό event, το οποίο έχουμε προγραμματίσει να συμβαίνει μία φορά μόνο. Το event της βροχής ξεκινάει με την είσοδο του παίκτη στο συγκεκριμένο Share και αποτελείται από δύο ηχητικά layers: ένα της βροχής και ένα των κεραυνών. Μπαίνοντας ο χρήστης στη σπηλιά, η βροχή θα σταματήσει, όπως είναι φυσικό εξάλλου και δεν θα επαναληφθεί, για να ακολουθήσει ένα άλλο ηχοτοπίο.

Το ηχοτοπίο της σπηλιάς διαθέτει επίσης ένα ηχητικό event που ενεργοποιείται κατά την είσοδο του χρήστη σε αυτήν, το οποίο είναι η πτώση βράχων από την οροφή. Το συγκεκριμένο εφέ ήχου διαθέτει πολλαπλούς ήχους βράχων, έτσι ώστε να αποδοθεί καλύτερα η τυχαιότητα με την οποία πέφτουν και δημιουργούν το αντίστοιχο ηχητικό αποτέλεσμα. Το ηχοτοπίο της σπηλιάς, αν και το λιγότερο πλούσιο, διαθέτει τέσσερα

layers ήχων, με το πρώτο να είναι μια βοή χαμηλής συχνότητας, το δεύτερο να είναι ο άνεμος που πιθανώς να μπαίνει από την είσοδο και να αντανακλά στα τοιχώματά της, το τρίτο να είναι οι ήχοι της ζούγκλας (της περιοχής της ομίχλης) σε πολύ απομακρυσμένη απόσταση, ενώ το τέταρτο layer είναι σταγόνες νερού που πέφτουν στο έδαφος σε τυχαία σημεία από τα τοιχώματα της σπηλιάς. Ακόμα, η σπηλιά διαθέτει και δύο *SoundEventSpots* που έχουν να κάνουν με εστίες φωτιάς, με την μία να διαθέτει λίγο πιο βαθύ ηχώ από την άλλη. Επιπλέον, προκειμένου να αποδοθεί καλύτερα η κλειστή περιοχή του συγκεκριμένου ηχοτοπίου, έχουμε τοποθετήσει μέσα από την CryEngine 3 ένα Reverb εφέ για τον ήχο, το οποίο επηρεάζει συνολικά όλους τους ήχους που βρίσκονται μέσα στο shape της σπηλιάς.

Βγαίνοντας από τη σπηλιά ο χρήστης θα παρατηρήσει ότι η βροχή έχει σταματήσει και όσο προχωράει παρατηρεί ότι και ο καιρός βελτιώνεται, επαναφέροντας το περιβάλλον του ηχοτοπίου της ζούγκλας -τόσο ηχητικά αλλά και οπτικά- στο φυσιολογικό της ζωηρό ηχώχρωμα. Ακλουθώντας τον δρόμο και πηγαίνοντας αριστερά ο παίκτης μπορεί να κατευθυνθεί προς το υδάτινο ηχοτοπίο του χάρτη.

Στο υδάτινο ηχοτοπίο θα συναντήσει 3 ποτάμια με διαφορετικά ηχοχρώματα (ανάλογα με το βάθος τους), 3 καταρράκτες επίσης με διαφορετικά ηχοχρώματα (ανάλογα με το μέγεθός τους), μία λίμνη και ήχους από βατράχια, τόσο κοντά στην επιφάνειά της όσο και σε διάφορα σημεία κατά μήκος των ποταμών. Στο σημείο στο οποίο περίπου ξεκινάει το υδάτινο τοπίο, ο χρήστης θα έχει εισέλθει ήδη στο επόμενο ηχοτοπίο, το οποίο είναι ορεινό φθινοπωρινό περιβάλλον. Πρόκειται ουσιαστικά για το ενδιάμεσο μέρος που βρίσκεται μεταξύ της ζούγκλας και των χιονισμένων βουνών. Σε αυτό το ηχοτοπίο θα συναντήσουμε συνολικά τρία layers, ένα με πουλιά διαφορετικού είδους από αυτά της ζούγκλας, ένα ακόμα layer όπου ο άνεμος θα είναι πιο δυνατός και ένα ακόμα layer με μακρινούς ήχους εντόμων. Το συνολικό άκουσμα του ηχοτοπίου έχει ως σκοπό να είναι πιο ήρεμο από αυτό της ζούγκλας και με διαφορετική ποικιλία βιοφωνίας.

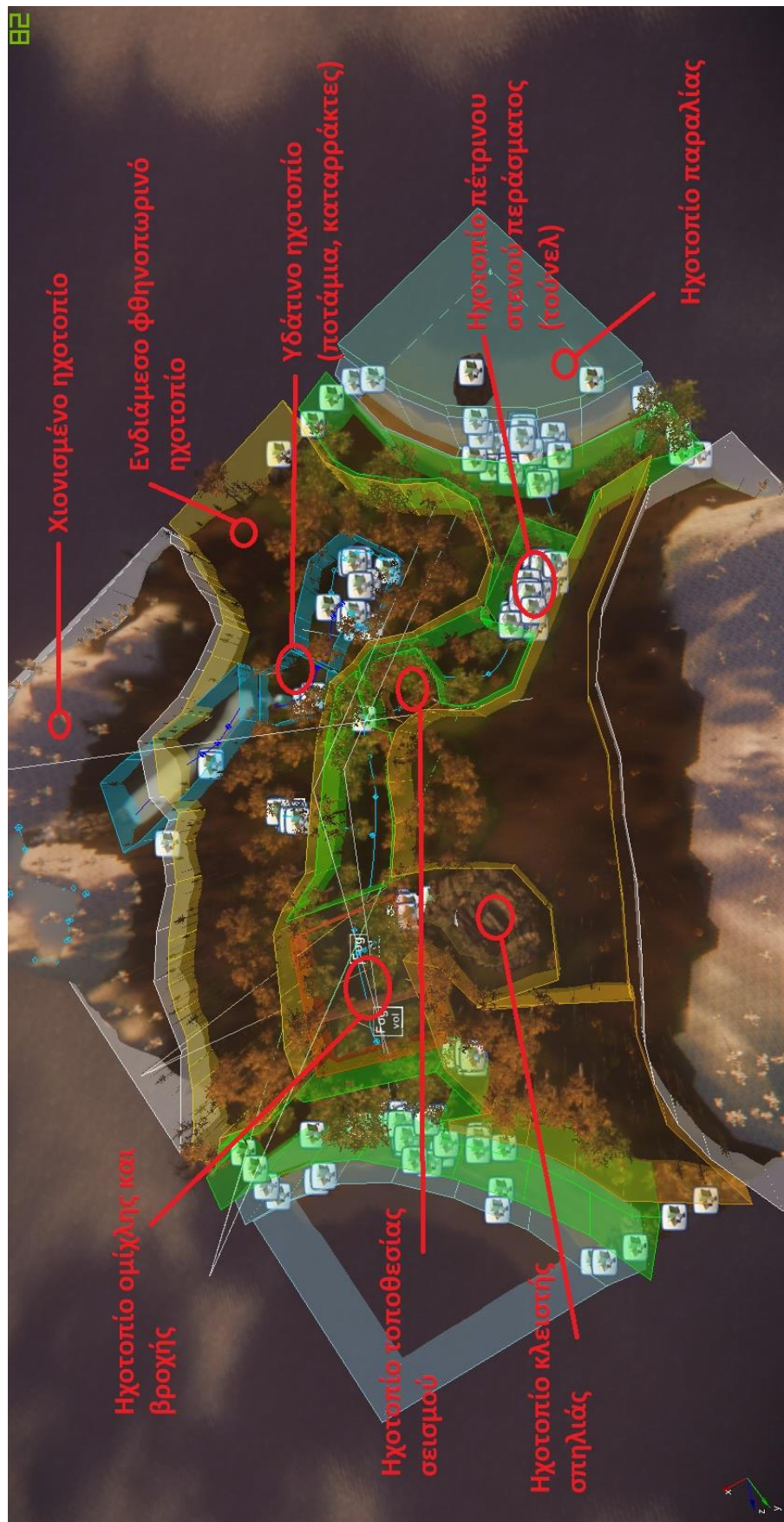
Αν ο χρήστης κατευθυνθεί βόρεια προς το μεγάλο καταρράκτη, θα εισέλθει αυτόματα στο πιο υπερυψωμένο χιονισμένο ηχοτοπίο. Στο χιονισμένο μέρος του χάρτη ο άνεμος αποτελεί το κυρίως άκουσμα, ενώ πολύ αραιά θα ακουστούν και φωνές από αετούς. Κατευθυνόμενος αριστερά από τον μεγάλο καταρράκτη, ο χρήστης θα βρει μια συσκευή μεταφοράς, η οποία θα τον μεταφέρει άμεσα στο περιβάλλον της ζούγκλας, προκειμένου να συνεχίσει την πορεία του στον κεντρικό δρόμο.

Το επόμενο ηχητοπίο είναι αυτό της περιοχής όπου γίνεται ένας σεισμός. Σκοπός εδώ είναι να προσδώσουμε στους υπάρχοντες ήχους του ηχοτοπίου της ζούγκλας μια κατάσταση πανικού. Θα ακούσουμε τις μαϊμούδες και τα πουλιά να φωνάζουν πολύ πιο πυκνά και δυνατά, ενώ υπάρχει επιπλέον και μια έντονη βοή που δημιουργείται από τον σεισμό. Βγαίνοντας από την κυκλική περιοχή του event του σεισμού κατευθυνόμαστε στο επόμενο και τελευταίο ηχοτοπίο του χάρτη, δηλαδή στο πέτρινο τούνελ. Σκοπός εδώ είναι να ακούγεται τόσο το ηχοτοπίο της ζούγκλας, αλλά με τη χρήση Occlusion εφέ μέσα από το FMOD, μαζί με Reverb εφέ από την Cry Engine, εξομοιώνοντας έτσι το πέρασμα των ήχων

της ζούγκλας μέσα από τα χοντρά πετρώματα και τις εισόδους του τούνελ. Όπως και στη πραγματικότητα, οι πιο χαμηλές συχνότητες καταφέρνουν να διασχίσουν ορισμένα υλικά, κάτι που στην ηχητική επεξεργασία επιτυγχάνεται είτε με Occlusion εφέ (αν υπάρχει διαθέσιμο), αλλιώς με low-pass φίλτρα, το οποίο μπορεί να δημιουργηθεί ευκολότερα σε οποιοδήποτε audio editing πρόγραμμα, όπως το Audacity. Στα βιντεοπαιχνίδια είναι πιο συνηθισμένο να συναντάμε αυτή τη τεχνική σε υποθαλάσσια σημεία, όπου επιθυμούμε να δώσουμε την εντύπωση ότι οι ήχοι διαπερνούν το νερό και αλλοιώνονται.

Τέλος, εξερχόμενος από το τούνελ ο χρήστης θα βρεθεί για μια ακόμη φορά στο ηχοτοπίο της ζούγκλας και στη συνέχεια θα εξέλθει στο τελικό περιβάλλον της παραλίας, για να ολοκληρώσει την διαδρομή του στο σημείο της βάρκας. Στην παρακάτω φωτογραφία μπορούμε να δούμε το πώς δείχνει συνολικά όλη η προηγούμενη διαδρομή, μέσα από την ισομετρική κάμερα του Editor.

Εικόνα 1.12: Πανοραμική παρουσίαση των ηχοτοπίων της εργασίας



Κεφάλαιο 2: Εισαγωγή στην Cry Engine 3. Δημιουργώντας ένα τρισδιάστατο περιβάλλον με μια μηχανή γραφικών

Τί είναι μια Game Engine;

Μια σύντομη περιγραφή μιας μηχανής βιντεοπαιχνιδιών (ή και «μηχανής γραφικών»), θα μπορούσε να ήταν ότι αποτελεί το περιβάλλον εργασίας που μπορεί να εκτελέσει ένα βιντεοπαιχνίδι. Τι όμως είναι ένα βιντεοπαιχνίδι; Πρακτικά θα μπορούσαμε να θεωρήσουμε ότι είναι οτιδήποτε μπορεί να λειτουργήσει μέσω βίντεο σήματος σε μια οθόνη και να εκτελείται σε πραγματικό χρόνο από έναν ή περισσότερους χρήστες. Τεχνικά όμως, ένα βιντεοπαιχνίδι αποτελείται από ένα σύνολο τεχνολογιών, οι οποίες μπορούν να προστεθούν μέσα στο κοινό περιβάλλον εργασίας μιας μηχανής γραφικών, ώστε, μέσω αυτής, να υπάρξει τελικά το αποτέλεσμα που βλέπουμε σήμερα στα βιντεοπαιχνίδια.

Οι τομείς που μπορεί να περιλαμβάνει ένα βιντεοπαιχνίδι και μπορούν να λειτουργήσουν σχεδόν σε όλα τα περιβάλλοντα των σημερινών μηχανών γραφικών είναι:

Physics – Εξομοίωση ρεαλιστικών ενεργειών που υπακούουν στους κανόνες της φυσικής του πραγματικού κόσμου, ανεπτυγμένες σε τρισδιάστατο περιβάλλον. Τέτοια παραδείγματα μπορεί να είναι από την ρεαλιστική κίνηση κάποιου υφάσματος, μέχρι την εξομοίωση του αναπηδήματος μια μπάλας ή της κίνησης των κυμάτων.

Input – Είναι το μέσο χειρισμού του τελικού χρήστη. Αυτό μπορεί να αποτελείται από κάποιο πληκτρολόγιο και ποντίκι, μέχρι ένα game χειριστήριο ή ακόμα και κάποια VR συσκευή με motion sensor αντάπτορες.

Rendering – Πρόκειται για την εικονική απεικόνιση, τόσο σε 3D όσο και σε 2D περιβάλλον. Η απεικόνιση μπορεί να στοχεύει είτε σε ένα ρεαλιστικό αποτέλεσμα είτε σε κάποιο πιο καλλιτεχνικό αποτέλεσμα. Στην περίπτωση του πρώτου, μια 3D εικόνα συνήθως αποτελείται από πολύγωνα, υφές (textures), φωτισμό και σκιές, αντανakλάσεις και άλλες τεχνικές ρεαλιστικής απεικόνισης.

Scripting – Αποτελεί τον κώδικα που μπορεί να δημιουργηθεί εκ νέου ή να τροποποιηθεί μέσα από τα ήδη υπάρχον scripts μιας μηχανής γραφικών. Συνήθως οι γλώσσες προγραμματισμού που χρησιμοποιούνται είναι: C# και C++, python, Java, Lua, μεταξύ άλλων.

Collision Detection – Είναι το σύστημα που ευθύνεται ώστε τα τρισδιάστατα αντικείμενα να έχουν υπόσταση στο τρισδιάστατο περιβάλλον, που να μην διαφοροποιείται από αυτή τη πραγματικού κόσμου. Τέτοια παραδείγματα θα μπορούσε να ήταν η επιβράδυνση ή ακόμα και η ζημιά που μπορεί να δημιουργηθεί σε ένα 3D αυτοκίνητο, αν τρακάρει σε μια

επιφάνεια και γενικότερα οτιδήποτε έχει να κάνει με επαφή δύο ή περισσότερων τρισδιάστατων μοντέλων.

Artificial Intelligence – Πρόκειται για το σύστημα δημιουργίας νοημοσύνης χαρακτήρων που δεν ελέγχονται από τον παίκτη. Τέτοια παραδείγματα θα μπορούσε να είναι κάποιο εχθρικός χαρακτήρας που έχει προγραμματιστεί να κυνηγά τον παίκτη ή αντίθετα κάποιος χαρακτήρας που έχει προγραμματιστεί να τον συνοδεύει.

Ήχος – Ο ήχος στα βιντεοπαιχνίδια μπορεί να διαθέτει συμπεριφορές που ορίζονται μέσα από properties τα οποία διαβάζει η μηχανή γραφικών, από κάποιο middleware πρόγραμμα (όπως FMOD ή Wwise). Ωστόσο όλες οι μηχανές γραφικών διαθέτουν λειτουργίες για πολύ πειστικά ηχητικά αποτελέσματα σε τρισδιάστατα περιβάλλοντα (ηχός, occlusion, profiling κ.α.), αλλά και λειτουργίες για δυναμική αλλαγή της μουσικής.

Φυσικά οι δυνατότητες των μηχανών γραφικών δεν σταματάνε μόνο σε αυτά, καθώς περιέχουν λειτουργίες ακόμα και για σκηνοθεσία, όπως εισαγωγή cut scenes, διαλόγων, FMV (full motion video) και CG γραφικών (pre-rendered γραφικά επιπέδου ταινίας), online λειτουργίες για συμμετοχή πολλών χρηστών και άλλα.

Γιατί είναι σημαντικές οι μηχανές παιχνιδιών;

Παλαιότερα, στην εποχή των 2D γραφικών, ήταν πολύ συνηθισμένο η κάθε εταιρεία να απέπτυε την δική της μηχανή γραφικών, με τα δικά της εργαλεία. Αυτό είναι κάτι που άλλαξε ριζικά ύστερα από την έλευση των 3D γραφικών στη βιομηχανία, με τις μηχανές γραφικών να αναπτύσσονται από ξεχωριστές εταιρείες, εμπεριέχοντας εργαλεία που χρησιμεύουν σε πολλά είδη βιντεοπαιχνιδιών. Αυτό ήταν απαραίτητο προκειμένου να μην χάνεται κάθε φορά χρόνος στο να αναπτύσσεται εξολοκλήρου από την αρχή μια μηχανή γραφικών για κάθε βιντεοπαιχνίδι, από την κάθε εταιρεία. Αυτό με τη σειρά του σημαίνει ότι, τα εργαλεία και ο χώρος εργασίας αυτών των μηχανών άρχισε να γίνεται πιο σταθερός και δεδομένος για όλους τους developers, με αποτέλεσμα να είναι πιο εύκολο κανείς να εκπαιδευτεί σε ένα τέτοιο περιβάλλον.

Τα προηγούμενα είχαν ως συνέπεια να μειωθεί κατά πολύ ο χρόνος και το κόστος ανάπτυξης ενός παιχνιδιού, το οποίο όμως αντισταθμίστηκε σημαντικά από το γεγονός ότι τα βιντεοπαιχνίδια έγιναν μεγαλύτερα, πιο πολύπλοκα και πιο φιλόδοξα, κάτι που όμως είναι θετικό για τη βιομηχανία. Με το τεχνικό μέρος να είναι πιο συγκεκριμένο, οι σχεδιαστές είχαν πλέον την ευκαιρία να επικεντρωθούν περισσότερο στο καλλιτεχνικό μέρος της εμπειρίας του χρήστη, όπως στην ιστορία, στους χαρακτήρες, στην ορχηστρική μουσική που συχνά γράφεται με τη διαδικασία του outsourcing για πολλά βιντεοπαιχνίδια, στις online λειτουργίες και γενικά στο περιεχόμενο. Ωστόσο υπάρχουν αρκετές εταιρείες που αναπτύσσουν ακόμα και σήμερα τις δικές τους μηχανές γραφικών, κάτι που όμως γίνεται ακριβώς επειδή μπορούν να απορροφήσουν ένα τέτοιο κόστος.

Ορισμένες δημοφιλής μηχανές παιχνιδιών

Unreal Engine: Είναι μια από τις δημοφιλέστερες μηχανές παιχνιδιών, αλλά και από τις παλαιότερες σε 3D περιβάλλον και αναπτύχθηκε από την Epic Games. Η πρώτη έκδοσή της κυκλοφόρησε το 1998 και συνεχίζει να χρησιμοποιείται μέχρι και σήμερα με την 4^η έκδοσή της. Ορισμένοι γνωστοί τίτλοι που αναπτύχθηκαν με τη συγκεκριμένη μηχανή, είναι τα Gears of War, Mass Effect, Bioshock, Batman: Arkham και άλλα. Πλεονέκτημα της συγκεκριμένης μηχανής είναι η ευκολία της στην παραμετροποίηση, κάτι που σημαίνει ότι μπορούσε να χρησιμοποιηθεί σε πολλά και διαφορετικά είδη παιχνιδιών.

Unity Engine: Η Unity αποτελεί επίσης μια εύκολη στη χρήση μηχανή παιχνιδιών και χρησιμοποιείται από πολλούς μικρούς και ανεξάρτητους δημιουργούς για interactive 3D παιχνίδια. Μπορεί επίσης να χρησιμοποιηθεί με τέτοιο τρόπο ώστε να δημιουργηθούν πολλά είδη παιχνιδιών και σε ένα μεγάλο εύρος συσκευών. Στα παραδείγματα ορισμένων τίτλων που αναπτύχθηκαν με την Unity Engine θα βρούμε τα: Lara Croft Go, Her Story, Pillars of Eternity, Kerbal Space Program και άλλα. Το ισχυρότερο χαρακτηριστικό της σημερινής έκδοσης της Unity (Unity 5) είναι ίσως το γεγονός ότι διατίθεται δωρεάν προς όλους, με πολλά χαρακτηριστικά και λειτουργίες και μπορεί να χρησιμοποιηθεί για ανάπτυξη σε όλες τις ενεργές πλατφόρμες της αγοράς. Ωστόσο στην Professional Edition, η οποία διαθέτει πιο προχωρημένες λειτουργίες, όπως beta access, game performance reporting, customizable splash screens, team license και άλλα, υπάρχει μηνιαίο κόστος απόκτησης.

GameMaker: Σε αντίθεση με άλλες μηχανές παιχνιδιού, το μεγαλύτερο πλεονέκτημα της GameMaker είναι ότι δεν απαιτεί γνώσεις προγραμματισμού και coding για να χρησιμοποιηθεί. Ο χρήστης μπορεί να δημιουργήσει παιχνίδια μέσω «point-and-click» λειτουργιών, με μεγαλύτερη ταχύτητα. Ορισμένα παραδείγματα τίτλων με την GameMaker είναι τα: Spelunky, Hotline Miami, Super Crate Box και Hyper Light Drifter. Παρόλο που η συγκεκριμένη μηχανή έχει σχεδιαστεί ώστε να είναι απλή στη χρήση, ωστόσο αυτό σημαίνει ότι έχει περιορισμούς ως προς την ποικιλία των τίτλων που μπορεί κανείς να δημιουργήσει. Ωστόσο, αν και διατίθεται δωρεάν σε μια βασική μορφή, η επαγγελματική της έκδοση πρέπει να αποκτηθεί επί πληρωμή.

Godot Engine: Η συγκεκριμένη μηχανή μπορεί να χρησιμοποιηθεί κυρίως για 2D αλλά και 3D βιντεοπαιχνίδια. Χρησιμοποιεί γνωστά εργαλεία και είναι δωρεάν αλλά και open source. Δεν υπάρχει κάποιο κόστος αγοράς, ούτε και royalties ή κάποιος άλλος περιορισμός για τον δημιουργό. Ακριβώς επειδή είναι open source, αυτό σημαίνει ότι διαθέτει ενεργή κοινότητα που διορθώνει συνεχώς τις λειτουργίες της και απαντάει σε απορίες των χρηστών της.

AppGameKit: Η συγκεκριμένη μηχανή είναι ιδανική κυρίως για ανεξάρτητους δημιουργούς και αρχάριους developers. Χρησιμοποιείται κυρίως για ανάπτυξη εφαρμογών σε φορητές συσκευές (Iphones, Android, κ.α.), αλλά και για Mac και Windows εφαρμογές. Η

AppGameKit κοστίζει περίπου 80 δολάρια και μπορεί να αποκτηθεί μαζί με διάφορα add-ons, όπως τον Visual Editor, ώστε να υπάρξει καλύτερη τελική τιμή για το χρήστη.

Cry Engine: Περισσότερα για την μηχανή παιχνιδιού που επιλέχθηκε στη συγκεκριμένη εργασία θα μάθουμε στο επόμενο κεφάλαιο. Η CryEngine διατίθεται πλέον δωρεάν για χρήση, με όλες τις λειτουργίες της και μπορεί να χρησιμοποιηθεί για ανάπτυξη σε όλες τις σημερινές πλατφόρμες (Xbox One, PlayStation 4, PC, Oculus Rift, κ.α.). Η Cry Engine έχει σχετικά καλή υποστήριξη από tutorials και documentation, ωστόσο δεν θεωρείται ως μια από τις ευκολότερες για αρχάριους δημιουργούς. Υπάρχει επίσης υποστήριξη για αγορά διαφόρων assets μέσα από το Cryengine Marketplace, κάτι που μπορεί να βοηθήσει μικρότερες ομάδες ανάπτυξης, ενώ υπάρχουν ακόμα και δωρεάν assets που διατίθενται μέσα σε «πακέτα».

Amazon Lumberyard: Η συγκεκριμένη μηχανή αποτελεί την πιο πρόσφατη προσπάθεια της Amazon ώστε να εισέλθει στην βιομηχανία ανάπτυξης βιντεοπαιχνιδιών. Η μηχανή διατίθεται δωρεάν, χωρίς royalties ή άλλα κόστη και επιτρέπει άμεση αλληλεπίδραση με online πλατφόρμες όπως τα Twitch (streaming πλατφόρμα) και AWS (Amazon Web Services). Η μηχανή της Amazon συγκαταλέγεται σε αυτές που είναι απλές στη χρήση και έχει και λειτουργίες για ανάπτυξη εφαρμογών πολλαπλών παικτών (multiplayer games).

RPG Maker Engine: Η συγκεκριμένη μηχανή είναι από τις πιο απλές στη χρήση και διαθέτει έτοιμα εργαλεία με τα οποία μπορεί ακόμα και ένας αρχάριος να δημιουργήσει ένα έτοιμο βιντεοπαιχνίδι, χωρίς απολύτως καμία γνώση προγραμματισμού ή κώδικα. Η μηχανή κοστίζει 80 δολάρια, ωστόσο υπάρχει και μια δωρεάν έκδοση για τα Windows. Αν και η συγκεκριμένη μηχανή απευθύνεται κυρίως σε αρχάριους χρήστες, πιο προχωρημένοι χρήστες μπορούν να κάνουν χρήση της Javascript γλώσσας προκειμένου να διαφοροποιήσουν το παιχνίδι τους από τις έτοιμες λειτουργίες που προσφέρονται.

LibGDX: Η συγκεκριμένη μηχανή χρησιμοποιεί Java περιβάλλον και API (Application Programming Interface) το οποίο υποστηρίζει όλες τις πλατφόρμες. Αυτό σημαίνει ότι είναι ιδανική για cross-platform δημιουργίες, δηλαδή όταν επιθυμούμε ο τίτλος μας να αναπτυχθεί παράλληλα σε πολλαπλές πλατφόρμες. Σημαντικό πλεονέκτημα της μηχανής είναι ότι επιτρέπει τον άμεσο έλεγχο και debugging μέσα από το desktop του χρήστη, κάτι που εξοικονομεί χρόνο στην όλη διαδικασία ελέγχου.

Urho 3D: Πρόκειται για μια δωρεάν και cross-platform μηχανή που χρησιμοποιεί την γλώσσα C++. Διαθέτει wiki σχετικά με τις λειτουργίες χρήσης της για τα Windows Visual Studio και MinGW, Android και Linux. Επειδή είναι δωρεάν διαθέτει ενεργή κοινότητα, ωστόσο απαιτεί από τον χρήστη να έχει καλές γνώσεις της γλώσσας C++.

Η επιλογή της μηχανής παιχνιδιών Cry Engine 3

Στην αγορά του 3D video game design υπάρχουν όπως αναφέραμε δεκάδες μηχανές παιχνιδιών, ωστόσο δεν είναι όλες ελεύθερες για χρήση από το ευρύ κοινό, καθώς αποτελούν εργαλεία των εταιρειών οι οποίες τις έχουν αναπτύξει αποκλειστικά για εσωτερική τους χρήση. Σε τέτοια παραδείγματα συγκαταλέγονται μηχανές παιχνιδιών όπως η Fox Engine της Konami ή η Frostbite Engine της Electronic Arts. Ωστόσο, πολλές από τις μεγαλύτερες μηχανές παιχνιδιών (όπως Unreal Engine, Unity, Cry Engine) διατίθενται και δωρεάν στο ευρύ κοινό, μέσω ειδικών -πλήρως λειτουργικών εκδόσεων- που υπάρχουν αποκλειστικά για εκπαιδευτική και ιδιωτική χρήση. Η Cry Engine 3 διαθέτει το CryEngine 3 Free SDK, το οποίο και χρησιμοποιούμε για την ανάπτυξη του περιβάλλοντος της συγκεκριμένης εργασίας.



Εικόνα 2.1: Το logo (λογότυπο) της Cry Engine, ιδιοκτησία της Crytek

Η Cry Engine δημιουργήθηκε από την Γερμανική game developer εταιρεία Crytek, τον Μάιο του 2002 και συνεχίζεται να χρησιμοποιείται έως και σήμερα, έχοντας φτάσει την έκδοση «Cry Engine V». Έχει χρησιμοποιηθεί σε όλους τους τίτλους της εταιρείας, με τον πρώτο να είναι το Far Cry (2004) και συνεχίζει να υποστηρίζει όλες τις ενεργές πλατφόρμες της αγοράς. Η τρίτη της έκδοση, η Cry Engine 3 η οποία και χρησιμοποιούμε στην παρακάτω εργασία, ανακοινώθηκε το 2009 και αναπτύχθηκε αρχικά για τις πλατφόρμες των Microsoft Windows, PlayStation 3, Xbox 360 και Wii U. Η μηχανή υποστηρίζει τις γλώσσες C++ και Lua Scripts και τις εκδόσεις των DirectX 9, 10 και 11.

Ορισμένα από τα πλεονεκτήματα της Cry Engine 3 είναι η ευκολία με την οποία μπορεί κανείς να δημιουργήσει ένα μεγάλο φυσικό περιβάλλον, χωρίς να έχει γνώσεις προγραμματισμού. Στο ηχητικό μέρος η μηχανή συνεργάζεται άμεσα με το middleware πρόγραμμα FMOD, ενώ μας δίνει τη δυνατότητα να διαχωρίσουμε εύκολα το περιβάλλον σε περιοχές-ηχοτοπία, κάτι που μας διευκολύνει ιδιαίτερα στο σκοπό αυτή της εργασίας. Άλλος ένας σημαντικός παράγοντας για την επιλογή της, αποτέλεσε το αναλυτικό documentation αλλά και τα επεξηγητικά Tutorial βίντεο που μπορεί να βρει κανείς από ανεξάρτητους developer, κάτι που βοήθησε σημαντικά στην ολοκλήρωση του 3D περιβάλλοντος που θέλαμε να αναπτύξουμε.

Προετοιμασία πριν τη δημιουργία και παρατηρήσεις

Πριν ξεκινήσουμε τη δημιουργία του terrain στη Cry Engine, είναι καλό να έχουμε στο μυαλό μας μια γενική εικόνα για το περιβάλλον το οποίο θέλουμε να δημιουργήσουμε. Συνήθως υπάρχουν επιπλέον βήματα πριν κάποιος ξεκινήσει να δημιουργεί ένα τρισδιάστατο περιβάλλον, καθώς στις μεγάλες παραγωγές συνηθίζεται να σχεδιάζονται

πρώτα δισδιάστατα σχέδια σε χαρτί (ή concept arts), τα οποία αποτελούν την αρχική έμπνευση για τους 3D designers, έτσι ώστε η 3D δημιουργία να βασιστεί σε αυτά.

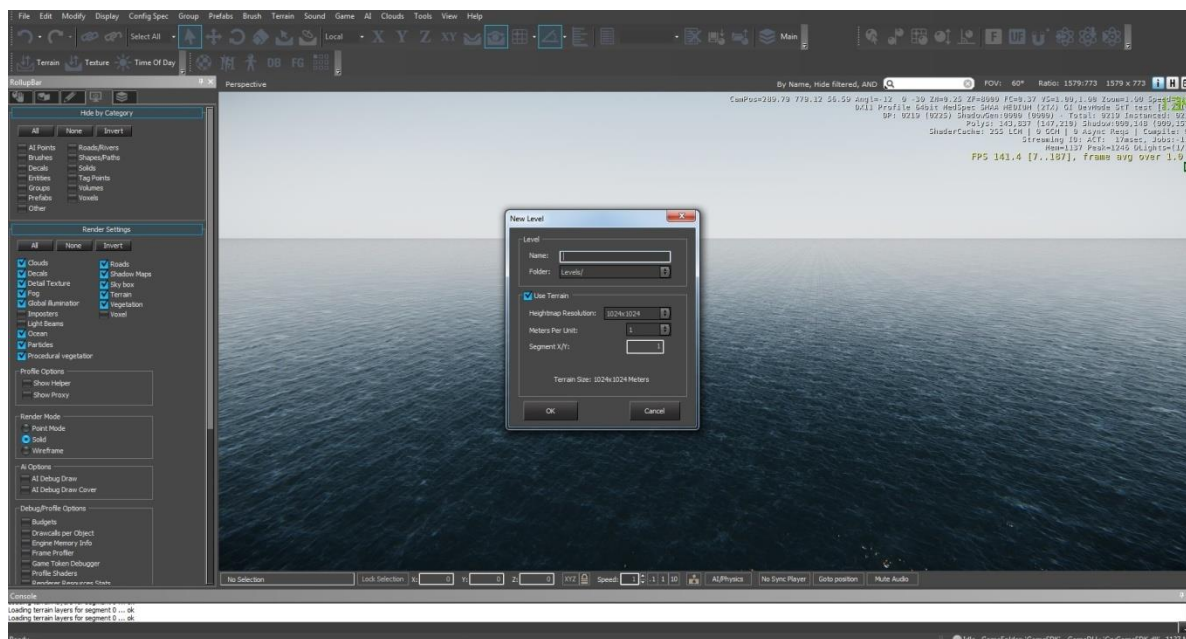
Ανάλογα με το μέγεθος του κάθε παιχνιδιού, το preproduction μπορεί να είναι αρκετά πολύπλοκη διαδικασία και μπορεί να χωρίζεται μέχρι και σε διαφορετικές ομάδες μέσα στην κάθε εταιρεία. Ένα παιχνίδι μπορεί να διαθέτει εκατοντάδες διαφορετικές τοποθεσίες και χιλιάδες assets (χαρακτήρες, φυτά, αντικείμενα κ.α.), κάθε ένα σχεδιασμένο τόσο σε 2D concept art, αλλά και σε 3D μοντέλο στη συνέχεια.

Στη συγκεκριμένη εργασία έχει προτιμηθεί ένα ανοικτό περιβάλλον το οποίο θυμίζει τροπικό νησί, όμως έχουν δημιουργηθεί επιπλέον τοποθεσίες που ξεφεύγουν αρκετά από το βασικό στυλ μιας πράσινης ζούγκλας, κάτι που έγινε προκειμένου να υπάρξουν διαφορετικά ηχητικά αποτελέσματα στην εργασία. Ωστόσο, ποτέ μια δημιουργία δεν είναι «τελική» από την αρχή και, αν παρατηρήθηκε κάτι κατά τη δημιουργία αυτής της εργασίας είναι, ότι μια δημιουργία ορίζεται μόνο σε ένα μικρό ποσοστό από την αρχική ιδέα, καθώς στην πορεία η ιδέα αυτή μπορεί να εξελιχθεί και να αναπτυχθεί πολύ διαφορετικά, έχοντας κυρίως δύο περιορισμούς: **(α)** τις ικανότητες του δημιουργού με τα εργαλεία που έχει στη διάθεσή του και **(β)** το μέγεθος της ομάδας.

Συνεπώς παρατηρήσαμε ότι, η αρχική ιδέα για το περιβάλλον που θέλαμε να δημιουργήσουμε αναπτύχθηκε περισσότερο σε κάποια σημεία σε σύγκριση με την αρχική ιδέα, ενώ αντίστοιχα έπρεπε να μην αναπτυχθούν άλλα σημεία που κρίθηκαν περιττά. Αυτό δείχνει και τα προβλήματα που συχνά αντιμετωπίζουν μικρά και μεγάλα studio κατά την ανάπτυξη τίτλων και θα μπορούσαμε να πούμε ότι, αυτά είναι συνήθως επίσης δύο: **(α)** να υπάρχει είτε μια υπερφιλόδοξη ιδέα, η οποία στην πορεία πρέπει να «υποβαθμιστεί» σε κάτι πιο ρεαλιστικό και πιο πρακτικό για το μέγεθος και τις ικανότητες του studio ή **(β)** να υπάρχει μια μικρή ιδέα, η οποία να εξελιχθεί ακόμη περισσότερο κατά την ανάπτυξη, χάρη στο ταλέντο της ομάδας.

Δημιουργία του terrain

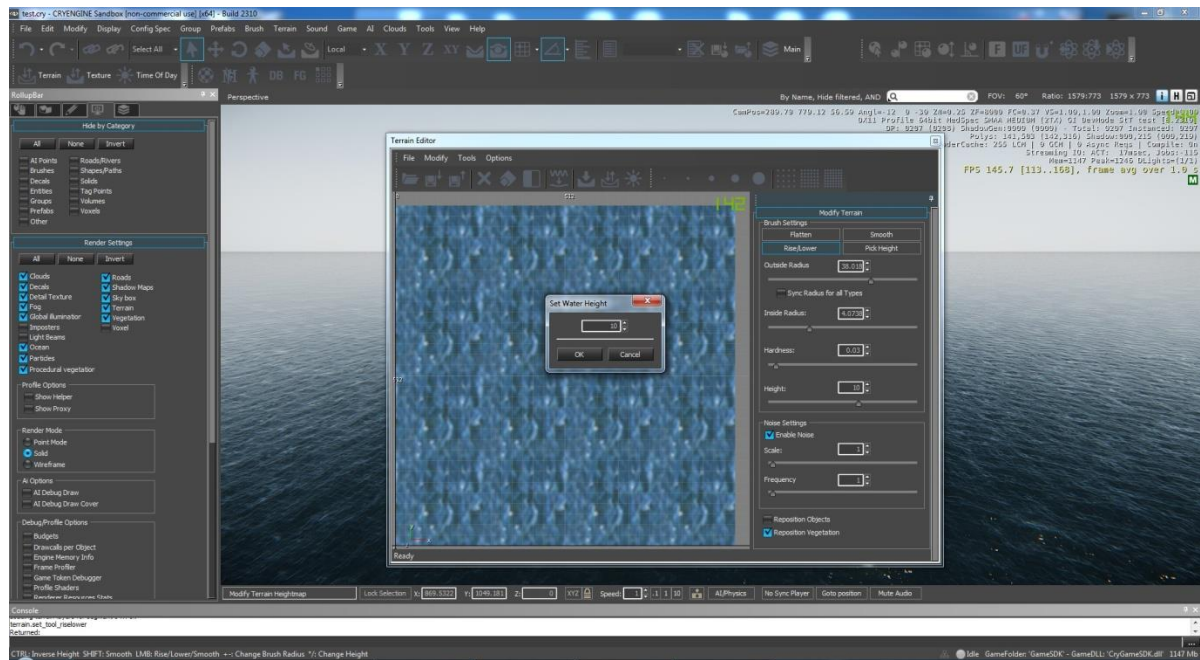
Το πρώτο πράγμα που πρέπει να κάνουμε ώστε να ξεκινήσουμε να σχεδιάζουμε το περιβάλλον είναι να δημιουργήσουμε ένα νέο αρχείο, στο οποίο θα αποθηκεύονται όλες οι αλλαγές και τα assets που προσθέτουμε στο συγκεκριμένο project. Πατώντας *File > New* θα εμφανιστεί ένα παράθυρο «*New Level*» στο οποίο θα κλιθούμε να ορίσουμε το μέγεθος του περιβάλλοντος και το όνομα. Οι μονάδες είναι σε μέτρα, οπότε για ένα terrain 1024x1024 και με βήμα το 1μέτρο, θα έχουμε μια συνολική έκταση του ενός τετραγωνικού χιλιομέτρου.



Εικόνα 2.2: Ο ορισμός του μεγέθους του terrain (χάρτη)

Στη συνέχεια πρέπει να ορίσουμε το έδαφος και τη θάλασσα, όπως επίσης και να ξεκινήσουμε να σχεδιάζουμε υψομετρικά το τοπίο μας. Αυτά μπορούν να γίνουν μέσα από τον Terrain Editor (*Terrain > Edit Terrain*). Το πρώτο που χρειάζεται να ορίσουμε σε αυτόν τον editor είναι το ύψος του ωκεανού. Είναι περισσότερο βολικό να ορισθεί ένα νούμερο που θα θυμόμαστε, καθώς στην πορεία θα κλιθούμε να ορίσουμε διάφορα άλλα settings του editor πάνω από το συγκεκριμένο νούμερο, οπότε στιδήποτε μεγαλύτερο από αυτό θα αποτελεί μέρος της στεριάς του χάρτη.

Πατώντας μέσα από το: *Terrain Editor > Modify > Set Ocean Height*, θα επιλέξουμε τελικά τα **10 μέτρα** ως το ανώτερο ύψος για την θάλασσα.



Εικόνα 2.3: Καθορισμός του ύψους του ωκεανού στον χάρτη

Στη συνέχεια πρέπει να ορίσουμε τη στεριά, κάτι το οποίο μπορεί να γίνει με πολλούς τρόπους και με διαφορετικές «βούρτσες» του editor. Επειδή στην προκειμένη περίπτωση χρειαζόμαστε μια πολύ μεγάλη επιφάνεια με ένα σταθερό ύψος πάνω από το ύψος της θάλασσας, για αυτό θα επιλέξουμε από τα Brush Settings την *Flatten* διαδικασία.

Επεξήγηση των εργαλείων του Terrain Editor:

Για τον καμβά: Το μπλε χρώμα απεικονίζει την θάλασσα που έχουμε ορίσει. Οτιδήποτε μαύρο απεικονίζει τη στεριά. Το μαύρο χρώμα για τη στεριά αλλάζει σταδιακά σε γκρι και άσπρο ανάλογα με την υψομετρική διαφορά. Για παράδειγμα, κάτι που είναι πολύ χαμηλά στην θάλασσα θα είναι με σκούρο μαύρο ενώ ένα βουνό θα απεικονίζεται με άσπρο χρώμα, όπως περίπου και σε ένα χάρτη από δορυφόρο.

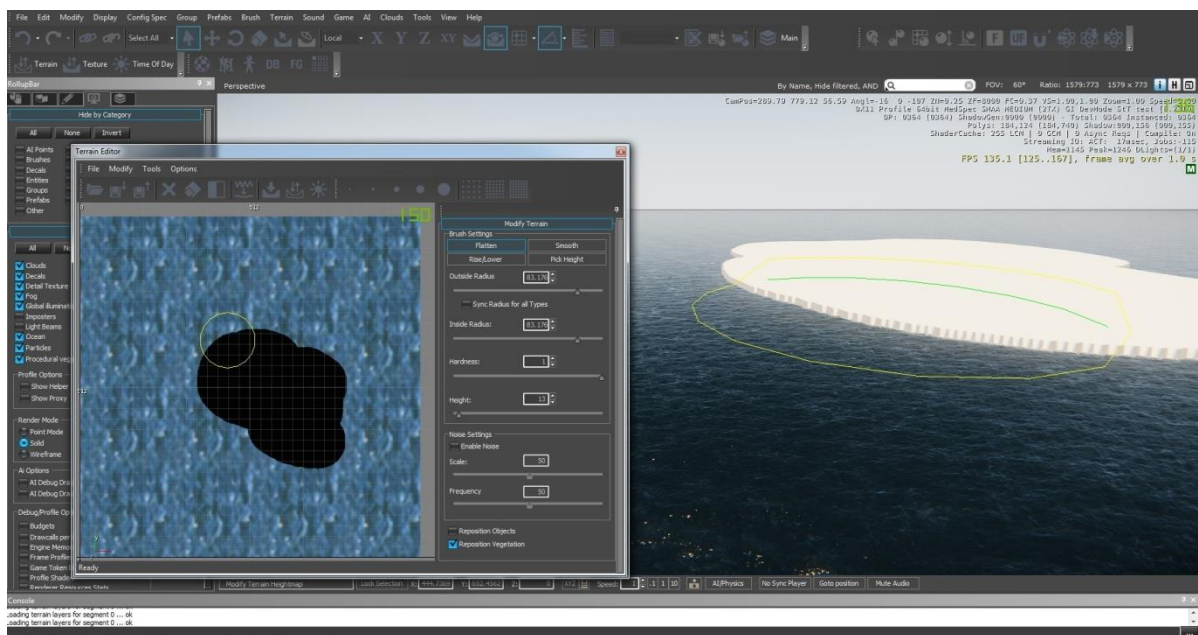
Outside Radius: Ορίζει το εξωτερικό μέγεθος της βούρτσας. Αυτό μπορεί να φανεί με δύο τρόπους: είτε από τον 2D μπλε καμβά που απεικονίζει τη θάλασσα στον editor, είτε κατευθείαν μέσα από τη 3D απεικόνιση.

Inside Radius: Ορίζει τον εσωτερικό κύκλο της βούρτσας, δηλαδή το πόσο κοντά στο εξωτερικό όριο θα είναι η αλλαγή που θέλουμε να επιφέρουμε. Στην προκειμένη περίπτωση οι δύο επιλογές μπορούν να είναι μαζί, καθώς θέλουμε να ορίσουμε μια μεγάλη flat επιφάνεια.

Hardness: Ορίζει το πόσο απότομα θα ανυψωθεί το έδαφος μέσα στο πλαίσιο της βούρτσας. Πάλι εδώ θα χρειαστούμε μια μεγάλη ρύθμιση, έτσι ώστε να μην υπάρξουν κενά στην επιφάνεια που θέλουμε αρχικά να ορίσουμε ως στεριά

Height: Ορίζει το ύψος της στεριάς. Αυτή είναι και η πιο σημαντική επιλογή που θα χρειαστεί να ορίσουμε εδώ, καθώς το νούμερο που θα τοποθετήσουμε εδώ θα ορίσει το αρχικό ύψος της στεριάς. Προηγούμενος είχαμε επιλέξει τα 10 μέτρα για το ύψος του ωκεανού, επομένως οτιδήποτε επιλέξουμε ως «height» πάνω από «10» θα αποτελέσει μέρος της στεριάς.

Παρακάτω μπορούμε να δούμε τις αλλαγές που επιφέρουν οι προηγούμενες ρυθμίσεις, τόσο στο δυσδιάστατο καμβά, όσο και στην τρισδιάστατη απεικόνιση από πίσω. Παρατηρούμε ότι, οπουδήποτε και αν μεταφέρουμε τη βούρτσα, θα δημιουργηθεί μια επιφάνεια με 3 μέτρα ύψος πάνω από τη θάλασσα:

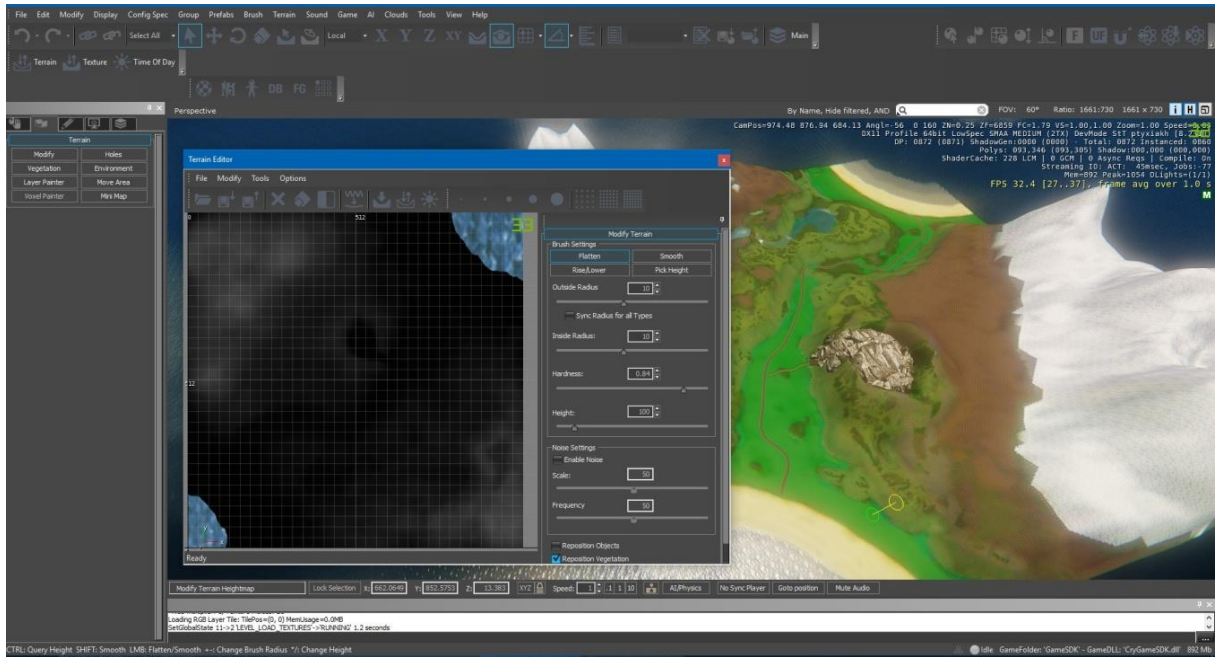


Εικόνα 2.4: Σχεδιασμός της επιφάνειας της στεριάς, με τη χρήση εργαλείου βούρτσας

Αυτός είναι ένας από τους πολλούς τρόπους με τους οποίους μπορεί κανείς να δημιουργήσει ένα playable terrain για το χρήστη. Το σημαντικό με την CryEngine 3 είναι ότι μας επιτρέπει να δοκιμάσουμε σε πραγματικό χρόνο το terrain που δημιουργήσαμε, μέσα από in-game κάμερα πρώτου προσώπου (Ctrl + G).

Είναι επίσης συνηθισμένο, σε μεγάλα περιβάλλοντα, οι level designers να μην δημιουργούν το terrain μέσα από τη μηχανή γραφικών, αλλά από τρίτα προγράμματα που συνεργάζονται με την Cry Engine και υπάρχουν αποκλειστικά γι' αυτόν το σκοπό. Η λογική τέτοιου λογισμικού είναι ουσιαστικά η ίδια με αυτή του λογισμικού FMOD που χρησιμοποιούμε σε αυτή την εργασία για τον ορισμό της συμπεριφοράς των ήχων, ενώ αντίστοιχα εργαλεία υπάρχουν και για άλλα τμήματα ενός τίτλου, όπως ειδικά λογισμικά για τους animators.

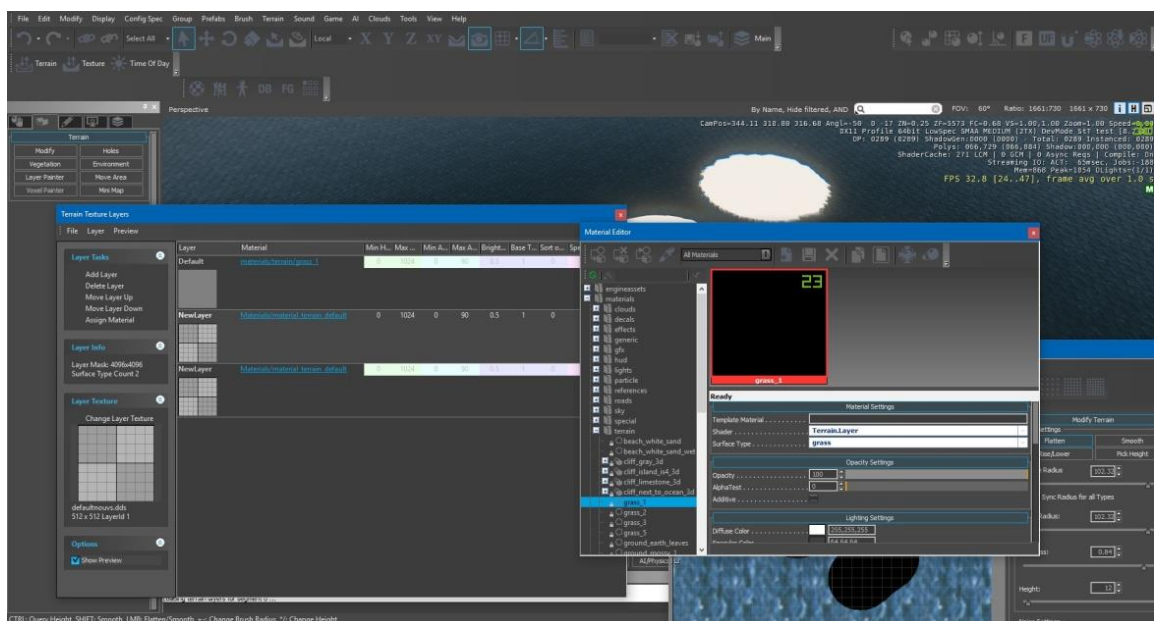
Παρακάτω στην **Εικόνα 2.5** μπορούμε να δούμε το τελικό σχήμα του terrain που έχουμε δημιουργήσει στον καμβά.



Εικόνα 2.5: Συνολικά η διαμόρφωση του εδάφους από ψηλά

Η Επιλογή των textures

Προκειμένου να δείχνει το περιβάλλον πιο αληθινό, είναι απαραίτητο να επιλέξουμε την κατάλληλη υφή, ή texture. Στην προκειμένη περίπτωση τα texture μας αποτελούν το έδαφος του τοπίου και τα έχουμε διαχωρίσει ανάλογα με το ύψος της κάθε περιοχής, έτσι ώστε να πετύχουμε και το αντίστοιχο αποτέλεσμα. Ο ορισμός αυτών των επιφανειών γίνεται μέσα από τον Texture Editor, ο οποίος μας δίνει πολλές δυνατότητες. Οι πιο σημαντικές που θα χρησιμοποιήσουμε στο δικό μας περιβάλλον είναι ο σχεδιασμός των texture ανά υψομετρική διαφορά.

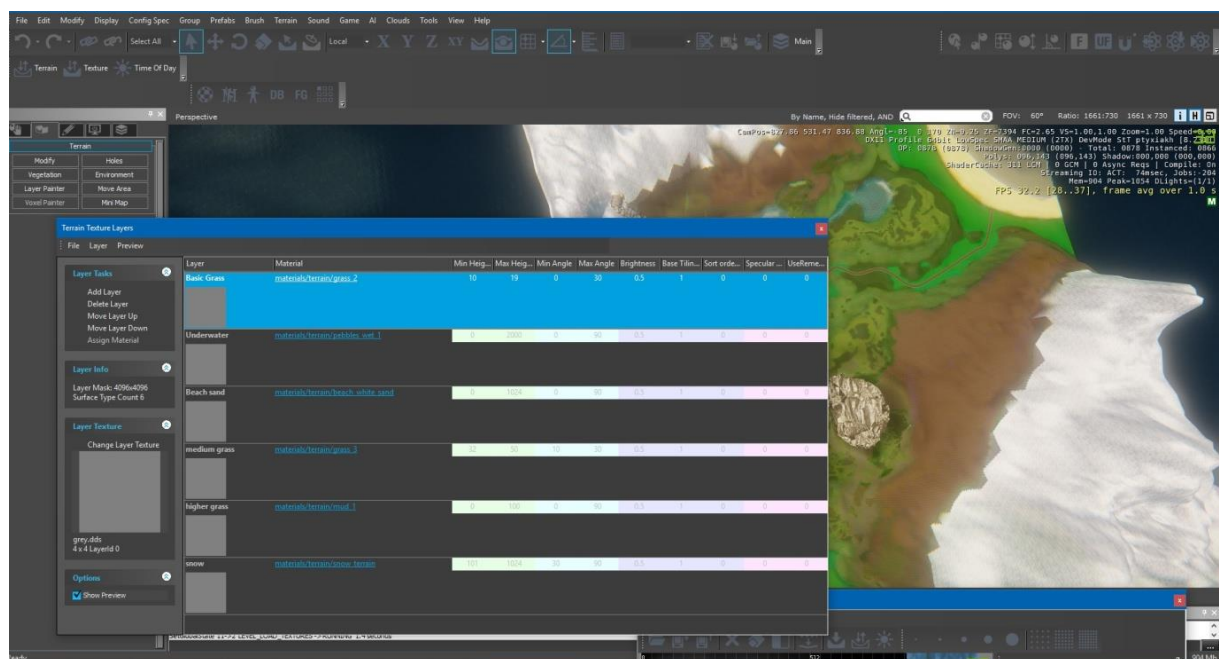


Εικόνα 2.6: Ορισμός εφαρμογής επιφανειών (texture) στη στεριά, ανάλογα με το ύψος του εδάφους

Όπως και με τα brushes του terrain, έτσι και εδώ ο editor μας αφήνει να ζωγραφίσουμε την επιφάνεια μιας περιοχής σε πραγματικό χρόνο, ωστόσο μας δίνει και ορισμένες επιλογές έτσι ώστε να μην υπάρχουν ανθρώπινα λάθη. Πριν ορίσουμε όμως τις επιφάνειες και τα settings για την κάθε μια, στην προηγούμενη φωτογραφία (**Εικόνα 2.6**) μπορούμε να δούμε το interface του texture editor. Στο αριστερό παράθυρο (Terrain Texture Layers) βλέπουμε συνολικά πόσες επιφάνειες έχει το terrain μας, ενώ η επιλογή ενός texture γίνεται από το νέο παράθυρο που μας εμφανίζεται, τον Material Editor.

Ανάλογα με το project, μια ομάδα μπορεί να επιλέξει μέσα από δεκάδες έτοιμα texture που υπάρχουν στη CryEngine, όμως συνηθίζεται η κάθε εταιρεία να έχει άτομα που να δημιουργούν καινούργια textures, σχεδιασμένα αποκλειστικά για την ανάγκη του κάθε παιχνιδιού. Αυτή η διαδικασία γίνεται συνήθως μέσω Photoshop.

Στην παρακάτω φωτογραφία (**Εικόνα 2.7**) βλέπουμε συνολικά τις επιφάνειες που έχουμε ορίσει για τις διαφορετικές περιοχές του χάρτη, αλλά και τα υψομετρικά settings της κάθε μιας, έτσι ώστε να μην αλληλεπικαλύπτει ποτέ η μία την άλλη με αφύσικό τρόπο.



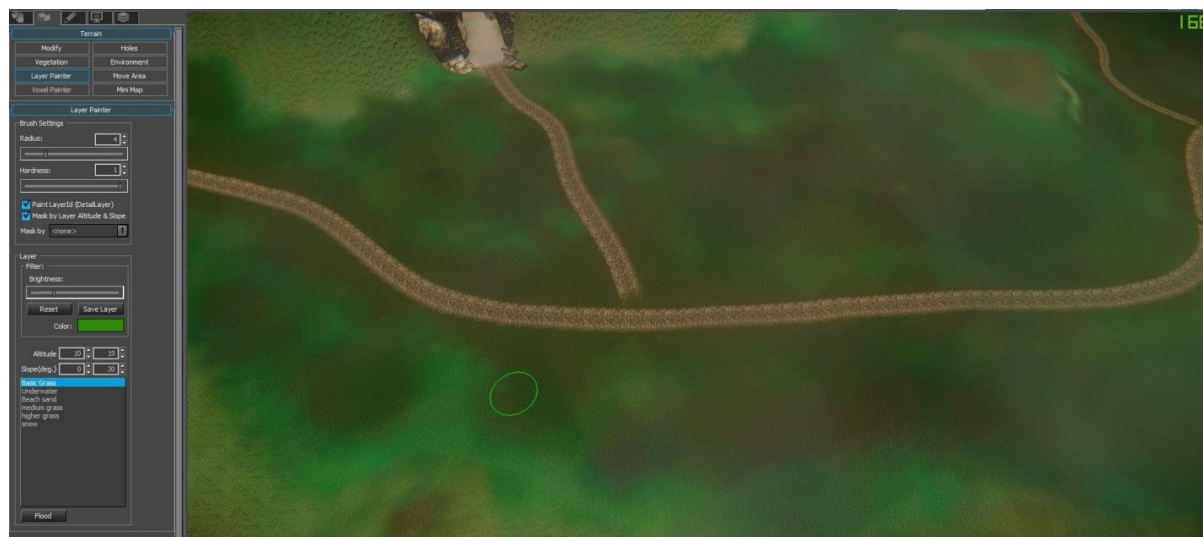
Εικόνα 2.7: Συνολικά οι επιφάνειες (textures) που έχουν χρησιμοποιηθεί στο τοπίο

Σκοπός ήταν να δημιουργήσουμε διαφορετικά επίπεδα στο συγκεκριμένο terrain, τα οποία ξεχωρίζουν τόσο από το texture του εδάφους, όσο και ηχητικά και φυσικά και από το γεωγραφικό ύψος. Έτσι, υπάρχει το πρώτο επίπεδο που αποτελεί το πιο πράσινο μέρος της ζούγκλας, το δεύτερο επίπεδο που αποτελεί ένα πιο ξερό «φθινοπωρινό» περιβάλλον με λιγότερη βλάστηση και το τρίτο επίπεδο που αποτελεί το χιονισμένο τοπίο. Είναι σημαντικό επίσης να επιλεγεί και ένα επιπλέον texture για την παραλία, αλλά και για τον βυθό της θάλασσας, των ποταμιών και της λίμνης.

Τέλος, για να τοποθετήσουμε τα textures θα πρέπει να πάμε στο *Terrain > Layer Painter*, ένα ακόμα εργαλείο βούρτσας, το οποίο παρατηρούμε ότι διαθέτει πλέον όλες τις πληροφορίες για τις επιφάνειες που ορίσαμε προηγουμένως. Συνεπώς, επιλέγουμε κάθε φορά ποιο texture θέλουμε να τοποθετήσουμε και στη συνέχεια περνάμε τη βούρτσα στο χάρτη, δίχως να απαιτείται μεγάλη ακρίβεια στο χειρισμό της, καθώς οι επιλογές που ορίσαμε προηγουμένως αποτρέπουν το ανθρώπινο λάθος και μας απαγορεύουν να τοποθετήσουμε άλλη επιφάνεια σε σημείο που δεν έχει ορισθεί.

Ποικιλία στο χρωματισμό των Texture εδάφους

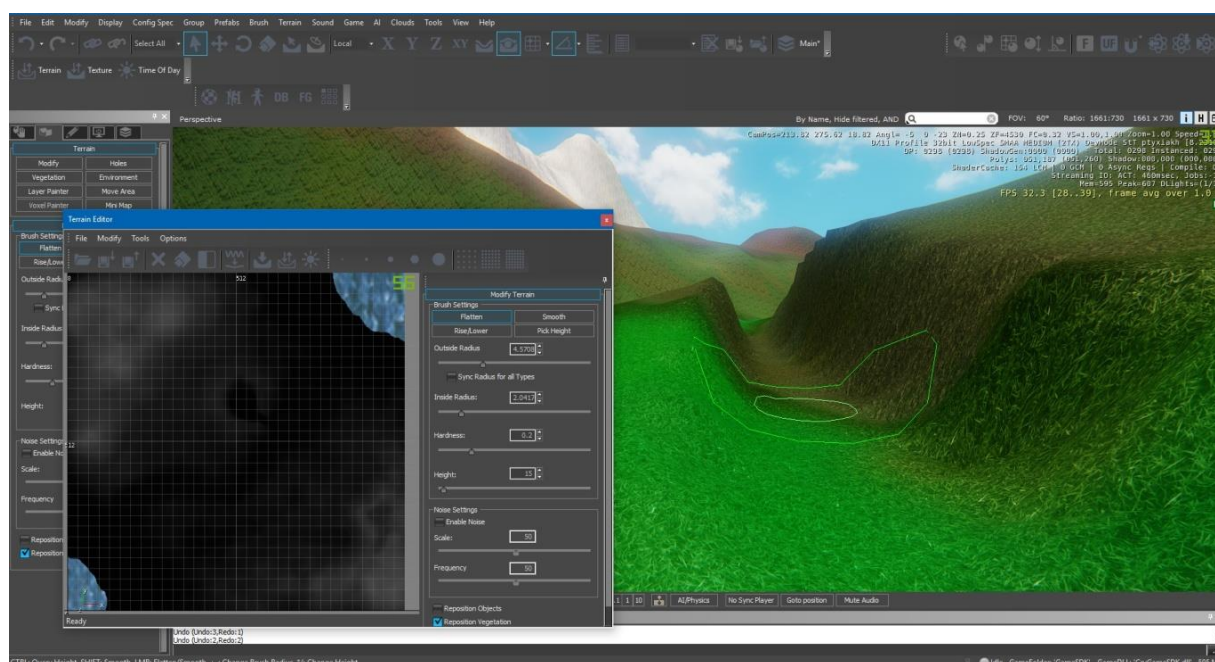
Μια σημαντική τεχνική που βοηθάει στο να γίνει πιο πειστικό το έδαφος του περιβάλλοντος και ιδιαίτερα της πράσινης επιφάνειας της ζούγκλας, είναι η χρήση πολλαπλών χρωμάτων της ίδια απόχρωσης (για παράδειγμα, πράσινου) στην επιφάνεια (texture) που καλύπτει ολόκληρη την περιοχή της ζούγκλας. Η συγκεκριμένη διαδικασία μπορεί να γίνει εύκολα μέσα από την επιλογή *Terrain > Layer Painter* και από το layer που είχαμε δημιουργήσει προηγουμένως για το texture της ζούγκλας, το οποίο είχαμε ονομάσει ως «*Basic Grass*». Επιλέγοντας το συγκεκριμένο layer και κάποιο διαφορετικό πράσινο χρώμα κάθε φορά, μπορούμε να δώσουμε περισσότερη ποικιλία στο έδαφος. Συγκεκριμένα επιλέξαμε τρεις διαφορετικές αποχρώσεις του πράσινου, οι οποίες τοποθετήθηκαν με τη βούρτσα του *Layer Painter editor*, με την πιο σκούρα πράσινη απόχρωση να τοποθετείται γύρω από τους δρόμους του χάρτη, όπως φαίνεται και από την παρακάτω **Εικόνα 2.8**.



Εικόνα 2.8: Επιπλέον χρωματισμός των επιφανειών για περισσότερη αληθοφάνεια

Δημιουργία δρόμων και μονοπατιών

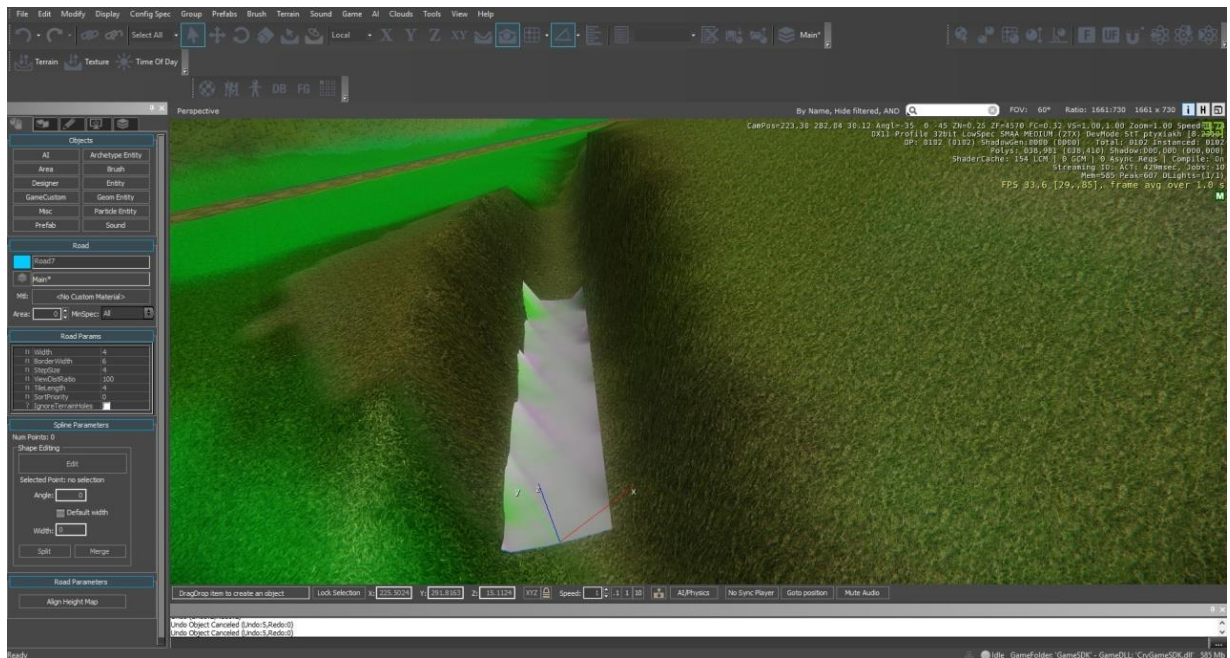
Πριν τοποθετήσουμε περισσότερες λεπτομέρειες στο terrain, καλό είναι να πάρουμε από τώρα ορισμένες αποφάσεις για τη γενικότερη λειτουργία του. Έτσι, είναι προτιμότερο αντικείμενα όπως δρόμοι να τοποθετηθούν κατά αυτή τη φάση της ανάπτυξης. Ένας από τους τρόπους για να δημιουργήσουμε ένα δρόμο μέσα στο τοπίο είναι να ανοίξουμε ένα επίπεδο πέρασμα πάνω στο οποίο θα τοποθετηθεί το layer του δρόμου. Από τα εργαλεία που χρησιμοποιήσαμε προηγουμένως από τον terrain editor, θα χρειαστούμε την Flatten βούρτσα. Γνωρίζοντας το ύψος του εδάφους, μπορούμε να ορίσουμε το ύψος από το οποίο θα περάσει ο δρόμος, όπως φαίνεται και στην παρακάτω φωτογραφία (**Εικόνα 2.9**).



Εικόνα 2.9: Επίπεδη διαμόρφωση του εδάφους, για την κατασκευή δρόμων και μονοπατιών

Στη συνέχεια, πηγαίνοντας αριστερά στην επιλογή Misc, όπως φαίνεται στην επόμενη φωτογραφία (**Εικόνα 2.10**), θα παρατηρήσουμε ότι η Cry Engine μας δίνει ένα ξεχωριστό εργαλείο για τη δημιουργία των δρόμων. Με αυτό, μπορούμε να δημιουργήσουμε μια επιφάνεια δρόμου με όποιο σχήμα θέλουμε, να τοποθετήσουμε ειδικά textures επάνω στην επιφάνεια του δρόμου, ενώ μπορούμε, αν θελήσουμε, να αλλάξουμε αργότερα ένα μέρος του σχήματος του δρόμου χωρίς να επηρεάσουμε ολόκληρο το τοπίο.

Με το που πατήσουμε το εργαλείο δημιουργίας δρόμων, ο κέρσορας αλλάζει και μπορούμε να ξεκινήσουμε να του δίνουμε σχήμα, προσθέτοντας διάφορες «ενώσεις» ή roints (με απλό κλικ) που ορίζουν τις καμπύλες του δρόμου. Όταν θελήσουμε να τον ολοκληρώσουμε πατάμε διπλό κλικ, για να ορισθεί το τελικό του σημείο. Αφού αποφασίσουμε για το σχήμα του δρόμου, στη συνέχεια απομένει να ορίσουμε από αριστερά το texture (Material) που θα χρησιμοποιήσουμε καθώς και κάποιες άλλες επιλογές εμφάνισης, αν το επιθυμούμε, όπως το πλάτος του.



Εικόνα 2.10: Ορισμός του σχήματος του μονοπατιού, με χρήση ειδικού εργαλείου

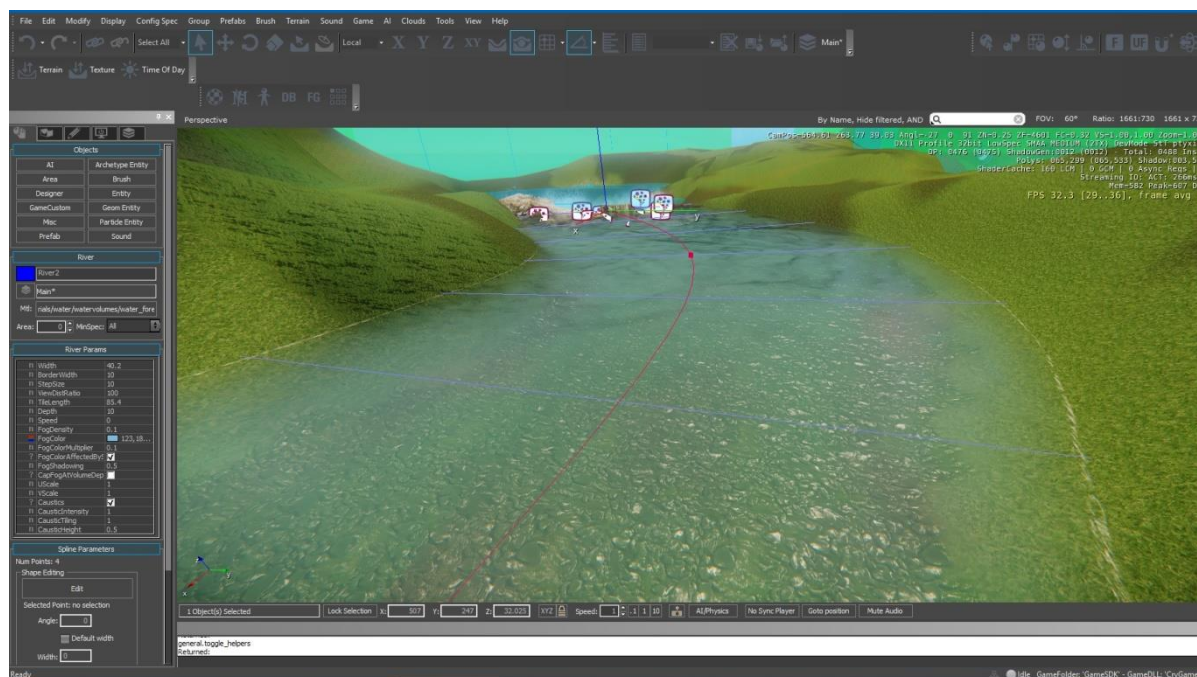
Το πρόβλημα της περιπλάνησης: Σχεδιασμός επιπλέον δρόμων

Μια προσθήκη η οποία κρίθηκε απαραίτητη κατά τη πορεία ανάπτυξης του τοπίου μας, είναι αυτή των επιπλέον δρόμων και περασμάτων, ως σημεία καθοδήγησης του ακροατή στο terrain. Παρ όλο που ολόκληρος ο χάρτης είναι ελεύθερος προς εξερεύνηση, ωστόσο η ιδέα των πολλαπλών δρόμων να καθοδηγούν τον θεατή σε συγκεκριμένα σημεία που θέλουμε να τον στείλουμε, λειτουργεί αποτελεσματικά για δύο λόγους. Πρώτον, είναι πιο εύκολο για τον σχεδιαστή να προσφέρει μια πιο πλούσια ηχητική εμπειρία στον ακροατή γύρω από αυτά τα μονοπάτια, καθώς υπάρχουν περισσότερες πιθανότητες να περάσει από εκεί ο ακροατής και άρα είναι πιο λογικό να τοποθετήσουμε εκεί περισσότερα ηχητικά events. Συγκεκριμένα, πολλοί ήχοι ζώων αλλά και κάποια από τα ηχοτοπία (όπως αυτό του σεισμού ή του τούνελ) έχουν τοποθετηθεί επάνω στην πορεία του δρόμου. Τέλος, η καθοδήγηση του παίκτη κρίνεται απαραίτητη προκειμένου να μπορέσει να περιηγηθεί πλήρως στο περιεχόμενο που έχει να του προσφέρει συνολικά ο συγκεκριμένος χάρτης, καθώς διαφορετικά θα χρειαζόταν πολύ χρόνος προκειμένου να ανακαλύψει από μόνος του κάποια ηχητικά events που βρίσκονται σε πιο μακρινές περιοχές.

Δημιουργία ποταμιών

Με περίπου αντίστοιχο τρόπο με αυτό για το δρόμο μπορούμε να δημιουργήσουμε και τα ποτάμια στην CryEngine. Η μόνη διαφορά εδώ είναι ότι ένα ποτάμι μπορεί να κινείται μόνο προς τα κάτω, επομένως πρέπει να τροποποιήσουμε αναλόγως συνολικά το terrain μας, ώστε να υπάρξει ένα φυσικό αποτέλεσμα. Στο συγκεκριμένο χάρτη αποφασίσαμε να δημιουργήσουμε 3 μικρά ποτάμια που καταλήγουν σε μια λίμνη.

Το εργαλείο «River» της Cry Engine μας δίνει πρακτικά τη δυνατότητα να ορίσουμε την επιφάνεια του νερού για το ποτάμι, καθώς και τη φορά στην οποία θα κινείται, μαζί με άλλες επιλογές, οι οποίες φαίνονται στα αριστερά στην επόμενη φωτογραφία (**Εικόνα 2.11**). Όπως και στη περίπτωση του δρόμου, για τη δημιουργία του ποταμιού ορίζουμε ένα σχήμα του με μια διαδρομή που αποτελείται από διαφορετικές «ενώσεις» (points) και εισάγονται με ένα κλικ, ενώ με διπλό κλικ ορίζουμε το σημείο που καταλήγει το ποτάμι.



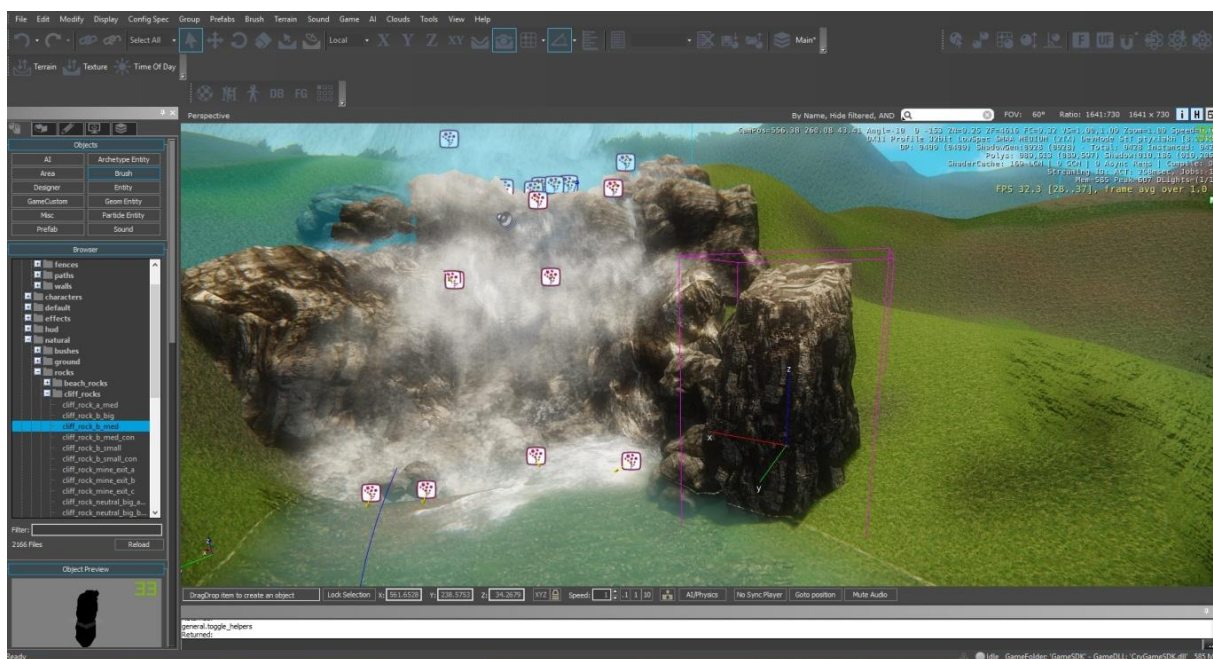
Εικόνα 2.11: Κατασκευή ποταμών, με ειδικό εργαλείο της Cry Engine

Τροποποίηση του terrain για τη δημιουργία των καταρραχτών

Προκειμένου να δημιουργηθεί το περιβάλλον ενός καταρράκτη, δεν μας δίνεται κάποιο συγκεκριμένο εργαλείο, όπως με τη δημιουργία των δρόμων και των ποταμών, αλλά μπορούμε να διαμορφώσουμε με χειροκίνητο τρόπο έτσι το τοπίο μας, ώστε το αποτέλεσμα να δείχνει αρκετά φυσικό. Στο χάρτη έχουμε επιλέξει να τοποθετήσουμε τρεις καταρράκτες διαφορετικών μεγεθών (έναν μεγάλο, έναν μεσαίο και έναν μικρότερο), οι οποίοι στην πορεία θα συνοδεύονται από διαφορετικά ακούσματα.

Στο μέρος που συνδέονται τα 3 ποτάμια μεταξύ τους, τα οποία έχουν έτσι και αλλιώς διαφορετικό ύψος, αφήσαμε ένα κενό το οποίο το καλύψαμε με αντικείμενα όπως πέτρες, δημιουργώντας έτσι τον «τοίχο» για τον καταρράκτη. Αντικείμενα όπως πέτρες μπορούν να βρεθούν πηγαίνοντας στην επιλογή *Brush > Browser* και επιλέγοντας στη συνέχεια το 3D αντικείμενο που μας ενδιαφέρει. Η τοποθέτησή του γίνεται άμεσα στο τοπίο, ενώ μπορούμε να το κάνουμε clipping μέσα στο έδαφος ή μέσα σε άλλα αντικείμενα (όπως άλλα βράχια), προκειμένου να φαίνεται το κάθε αντικείμενο λίγο πιο διαφορετικό από το άλλο (περισσότερα για τη τοποθέτηση αντικειμένων σε επόμενη ενότητα).

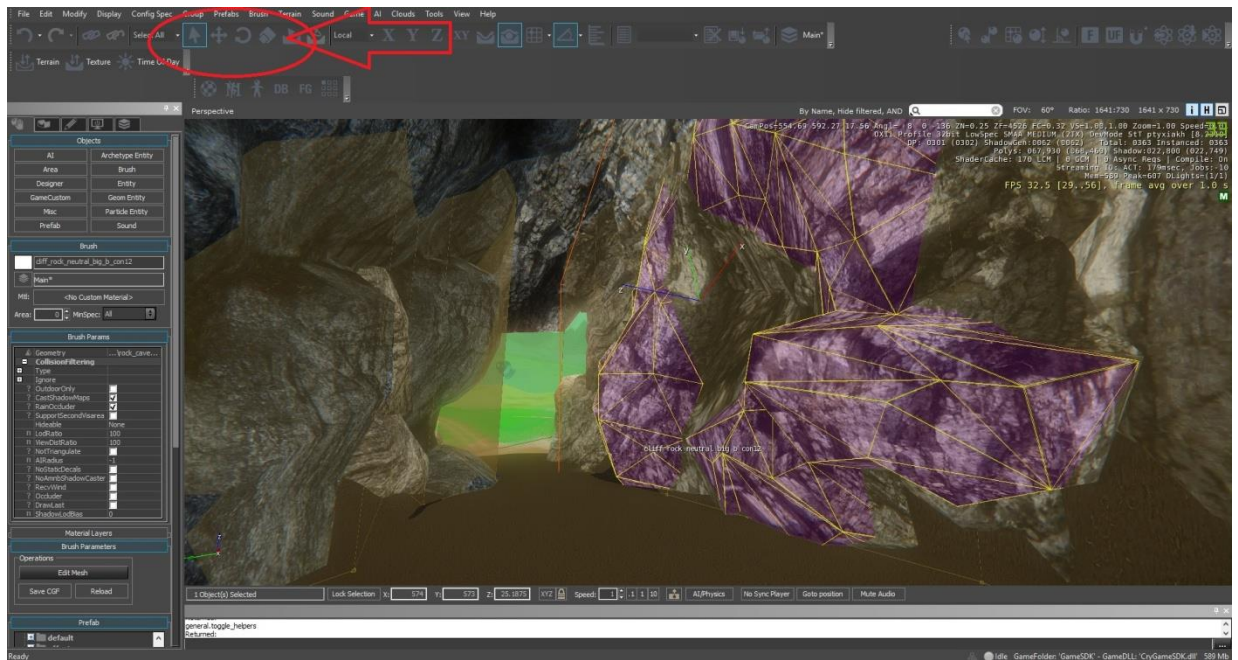
Επιπλέον μπορούμε να αλλάξουμε το μέγεθος των βράχων και τη γωνία τους, με συνέπεια να μη χρειάζεται να χρησιμοποιήσουμε περισσότερα από 3-4 είδη βράχων για ένα καλό αποτέλεσμα, καθώς είναι σκοπός να μην φανερώνεται η επανάληψη των assets. Αυτό είναι ιδιαίτερα χρήσιμο σε μεγάλες παραγωγές, έτσι ώστε να υπάρχει οικονομία στη χρήση των διαφορετικών assets, άρα και της συνολικής χωρητικότητας του τίτλου στο δίσκο.



Εικόνα 2.12: Τοποθέτηση βράχων για τη δημιουργία των σημείων των καταρρακτών

Δημιουργία της σπηλιάς

Για τη δημιουργία της σπηλιάς χρησιμοποιήσαμε μια αντίστοιχη τεχνική με αυτή για τους καταρράκτες, καθώς σε αυτή την έκδοση της CryEngine, το αποκλειστικό εργαλείο που υπάρχει για τη δημιουργία σπηλαίων δεν ήταν διαθέσιμο. Έτσι, ορίσαμε μια «λεκάνη» στο terrain την οποία στη συνέχεια καλύψαμε με βράχια από πάνω αλλά και περιμετρικά στην είσοδο και τους τοίχους. Στην παρακάτω φωτογραφία (**Εικόνα 2.13**) βλέπουμε το εσωτερικό της σπηλιάς, κοιτώντας προς τα έξω και την διαδικασία με την οποία τοποθετήθηκαν τα βράχια για τη διαμόρφωση των τοιχωμάτων. Σημαντικά εργαλεία προκειμένου να πετύχουμε ένα σωστό αποτέλεσμα, αποτέλεσαν τα εργαλεία περιστροφής, μεγέθυνσης και μετακίνησης προς όλους τους άξονες (X, Y, Z), τα οποία φαίνονται στο πάνω αριστερά μέρος της toolbar του Editor (βλ. κόκκινη επισήμανση).



Εικόνα 2.13: Τοποθέτηση βράχων για τη δημιουργία της σπηλιάς

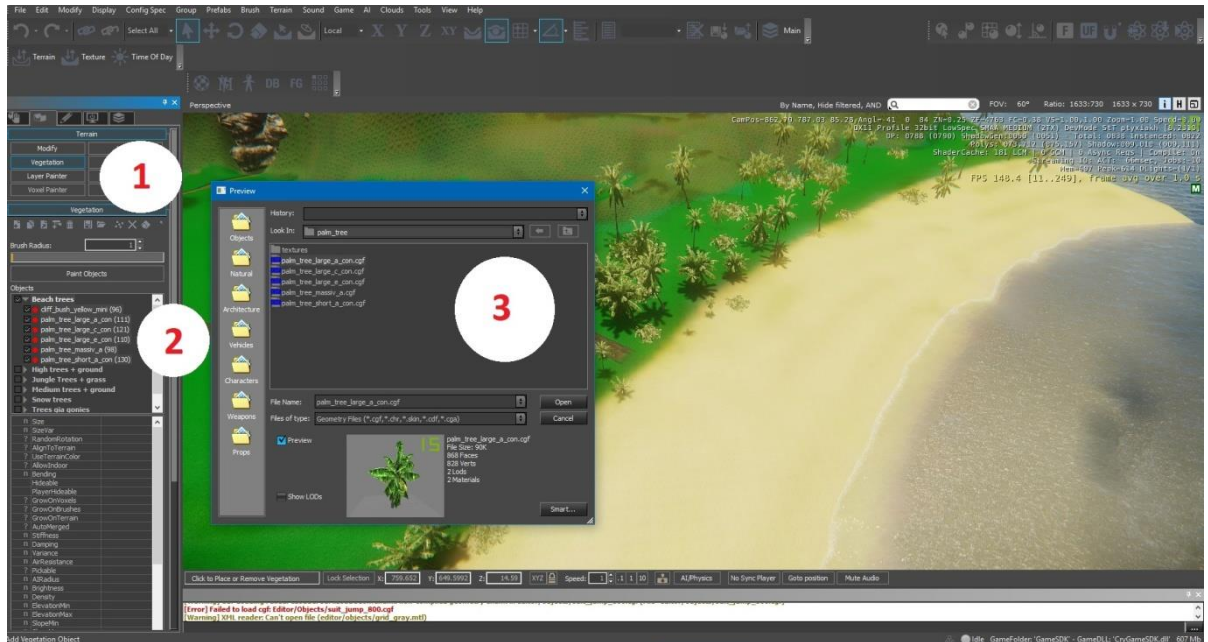
Σχεδιασμός της βλάστησης

Ένα από τα μεγαλύτερα πλεονεκτήματα της Cry Engine 3 είναι η ευκολία με την οποία μπορούμε να διαμορφώσουμε φυσικά τοπία, τεραστίων διαστάσεων, χωρίς να χρειάζεται να τοποθετούμε ένα-ένα τα assets (όπως δέντρα, θάμνους, πέτρες, κτλ). Όσο μεγάλο και αν είναι το terrain, μπορούμε μέσα σε μερικά δευτερόλεπτα να το εμπλουτίσουμε με διαφορετικά φυτά, αρκεί να ορίσουμε από πριν κάποιες σταθερές παραμέτρους. Φυσικά μπορούμε και να τοποθετήσουμε ένα-ένα ορισμένα από αυτά, σε περιπτώσεις που κάποιο σημείο χρειάζεται περισσότερη προσοχή, ενώ μπορούμε ακόμα και να μετακινήσουμε αυτά που έχουν ήδη τοποθετηθεί μαζικά, αν έχουν τοποθετηθεί σε σημεία που δεν επιθυμούμε.

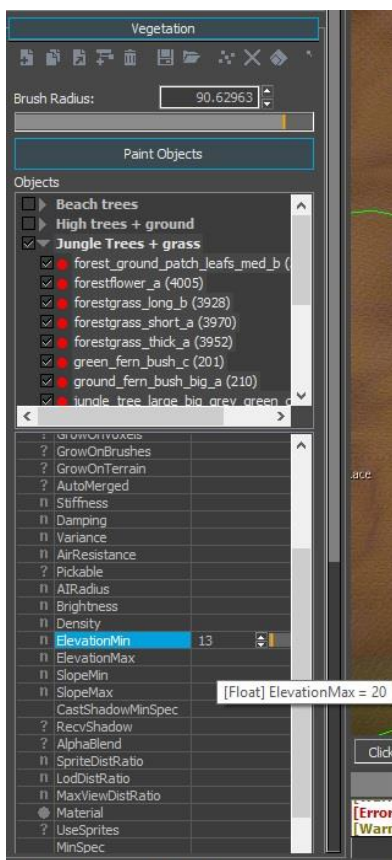
Στο δικό μας χάρτη γνωρίζουμε ότι έχουμε 3 διαφορετικά ηχητικά τοπία: αυτό μιας ζούγκλας, ένα πιο ορεινό φθινοπωρινό τοπίο στο ενδιαμέσο και ένα χιονισμένο τοπίο στο ψηλό μέρος του χάρτη. Από τη στιγμή που τα τρία αυτά τοπία έχουν διαφορετικό υψόμετρο, αυτό μας βοηθάει σημαντικά και στο σχεδιασμό των δέντρων. Το εργαλείο που θα χρησιμοποιήσουμε εδώ βρίσκεται στη κατηγορία *Terrain > Vegetation* και πρόκειται για μια «βούρτσα» η οποία μπορεί να ρυθμιστεί από πριν ώστε να «σχεδιάζει» δέντρα σε συγκεκριμένα σημεία μόνο του χάρτη, αποτρέποντας το ανθρώπινο λάθος.

Όπως φαίνεται και στη **Εικόνα 2.14** και στο σημείο «1», πρώτα επιλέγουμε το εργαλείο της βούρτσας και στη συνέχεια πρέπει να δημιουργήσουμε ορισμένες κατηγορίες για τα assets (στην περίπτωση μας, για τη βλάστηση). Η δημιουργία των κατηγοριών είναι σημαντική διότι ομαδοποιεί όσα assets βρίσκονται μέσα σε αυτή, με συνέπεια να μπορούμε και εκ

των υστέρων να αλλάξουμε ρυθμίσεις που να επηρεάζουν μαζικά όλα τα αντικείμενα της κατηγορίας ή να τη χρησιμοποιήσουμε ως έχει για να σχεδιάσουμε αργότερα, με τα ίδια settings, κάποιο άλλο σημείο του χάρτη. Στο σημείο «2» βλέπουμε ορισμένες από τις κατηγορίες που έχουμε δημιουργήσει για την ομαδοποίηση της βλάστησης.



Εικόνα 2.14: Εισαγωγή των assets της βλάστησης για τη δημιουργία του ειδικού εργαλείου «βούρτσας»



Εικόνα 2.15: Οι ρυθμίσεις των asset της βλάστησης για τον χώρο της ζούγκλας

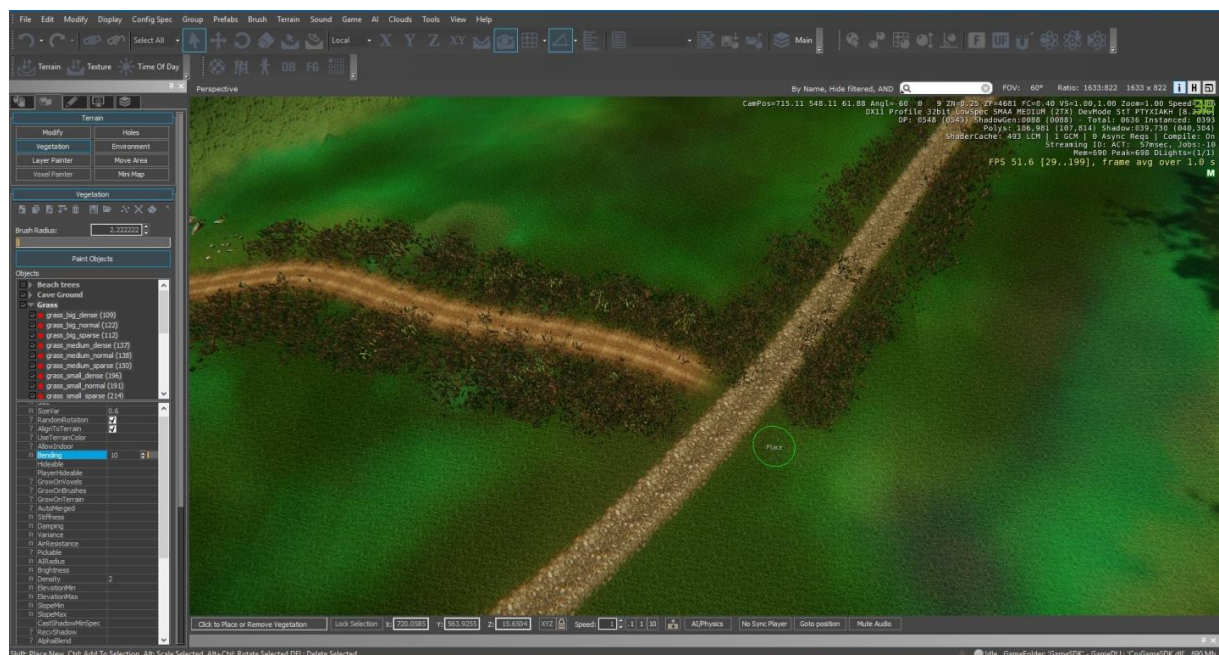
Μπορούμε να προσθέσουμε ή να αφαιρέσουμε assets οποιαδήποτε στιγμή από μια κατηγορία, πατώντας δεξί κλικ σε μια από αυτές και στη συνέχεια βρίσκοντας τα assets που μας δίνονται μέσα στη μηχανή γραφικών, όπως φαίνονται στο σημείο «3». Είναι επίσης δυνατό, να δημιουργήσει κάποιος τα δικά του αντικείμενα, όπως τα δέντρα και να τα κάνει import στη μηχανή, έτσι ώστε το αποτέλεσμα να είναι πιο αυθεντικό. Στο επόμενο παράδειγμα της **Εικόνας 2.15** βλέπουμε ορισμένα από τα settings που μπορούμε να ορίσουμε για να πετύχουμε ένα φυσικό αποτέλεσμα. Μεταξύ άλλων, μπορούμε να ορίσουμε το πόσο κοντά θα είναι τα δέντρα μεταξύ τους, να ορίσουμε το randomization της φοράς τους, αλλά ακόμα και το πόσο θα μεταβάλλονται από τον άνεμο.

Τα πιο σημαντικό setting είναι ίσως αυτά των «ElevationMin» και «ElevationMax». Για παράδειγμα, στην περίπτωση της βλάστησης για τη ζούγκλα, έχουμε ορίσει η συγκεκριμένη κατηγορία να σχεδιάζεται πάνω από τα 13

μέτρα (min) και κάτω από τα 20 μέτρα (max). Αυτό σημαίνει ότι η βούρτσα θα γεμίσει με δέντρα μόνο τα σημεία του χάρτη με το συγκεκριμένο υψομετρικό εύρος. Με τον ίδιο τρόπο συνεχίζουμε και για τις υπόλοιπες περιοχές του χάρτη, επιλέγοντας πάντα τα κατάλληλα objects που θεωρούμε ότι ταιριάζουν αισθητικά στη κάθε περιοχή, ενώ στο τέλος τροποποιούμε μεμονωμένα κάποια δέντρα ώστε να ξεχωρίζουν περισσότερο, κάνοντας χρήση των εργαλείων μετακίνησης, αύξησης και περιστροφής.

Μικρότερη βλάστηση: γρασίδι και χόρτα

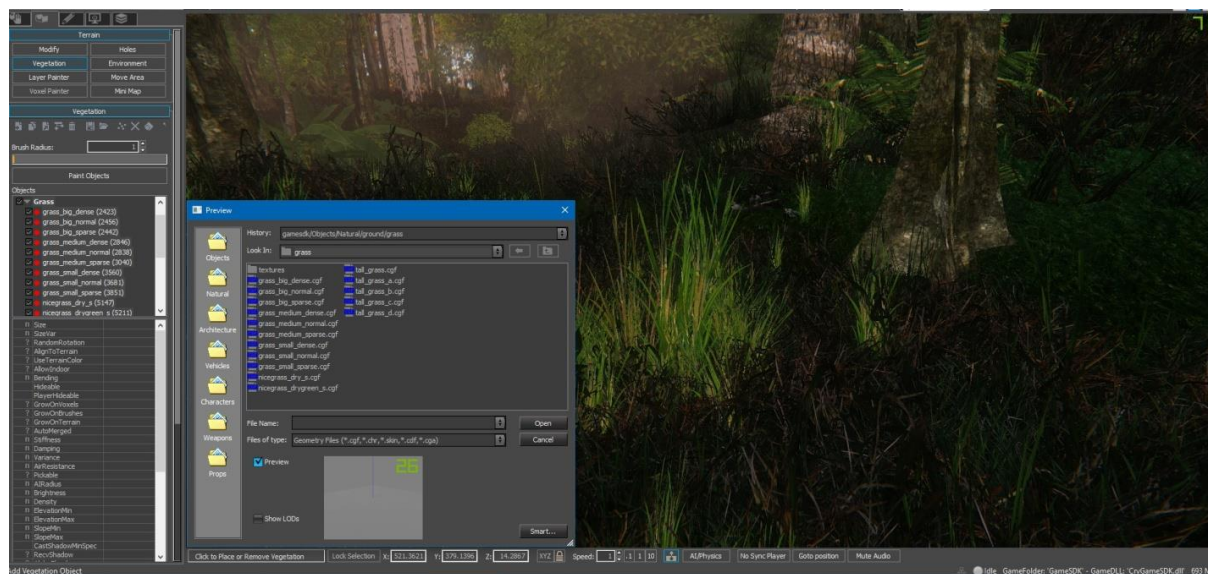
Μια σημαντική εισαγωγή που βοηθάει σημαντικά στην απόδοση ρεαλισμού συνολικά στο τοπίο, είναι αυτή της κοντής βλάστησης, όπως χόρτων και γρασιδιού. Η Cry Engine 3 είναι από τις μηχανές που επιτρέπουν να προστεθούν εύκολα και γρήγορα τα συγκεκριμένα assets της βλάστησης σε μεγάλα σημεία του χάρτη. Αυτό που πρέπει αρχικά να κάνουμε είναι να δημιουργήσουμε μια ακόμη κατηγορία στα Objects του Editor της βλάστησης (*Terrain > Vegetation*). Πατώντας δεξί κλικ στο παράθυρο και «Add Category», δημιουργούμε την κατηγορία «Grass», μέσα στην οποία θα τοποθετήσουμε ένα σύνολο από assets βλάστησης για το γρασίδι, προκειμένου να υπάρχει αρκετή ποικιλία ώστε να μην γίνεται εύκολα κατανοητή η επανάληψη.



Εικόνα 2.16: Προσθήκη γρασιδιού και οι βασικές ρυθμίσεις για το εργαλείο της «βούρτσας»

Με δεξί κλικ επάνω στη κατηγορία και επιλέγοντας «Add Object», μπορούμε να εισάγουμε τα αρχεία της βλάστησης στον editor. Τα συγκεκριμένα assets που επιλέξαμε βρίσκονται στη τοποθεσία «maesdk/Objects/Natural/ground/grass» και είναι τα παρακάτω που φαίνονται στην **Εικόνα 2.17**. Στο δεξί μέρος βλέπουμε το αποτέλεσμα στο τρισδιάστατο περιβάλλον. Όπως και με τα προηγούμενα παραδείγματα για την βλάστηση, το γρασίδι

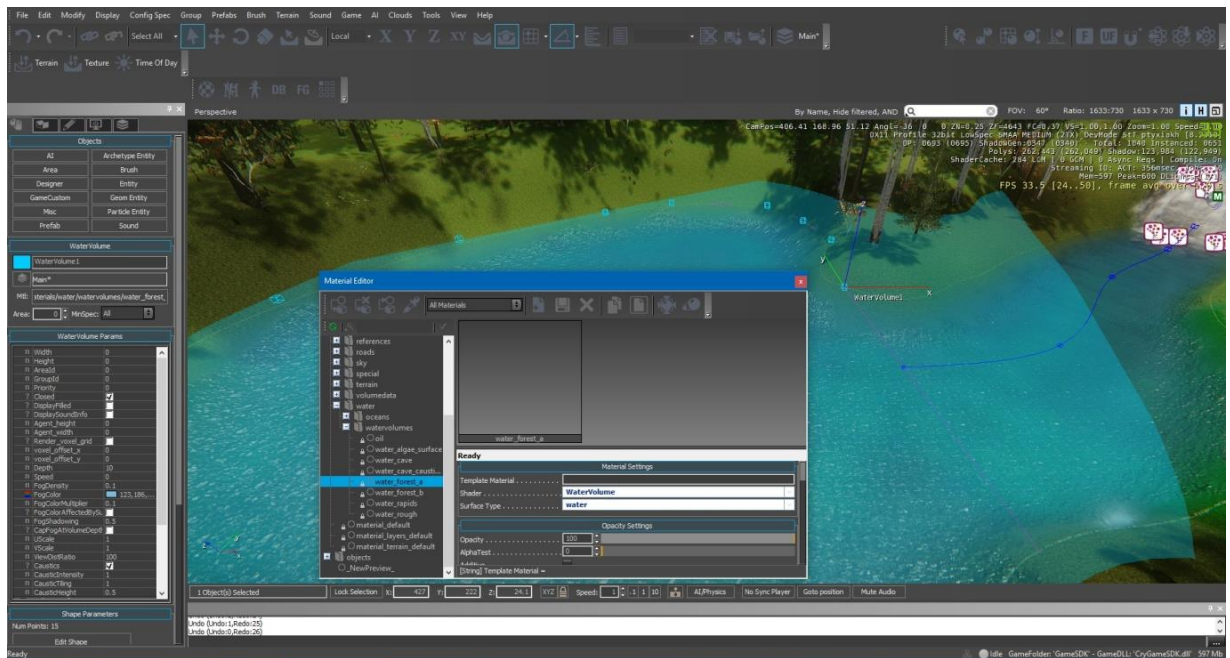
τοποθετείται με την επιλογή «*paint objects*», ενώ είναι απαραίτητο πρώτα να τροποποιήσουμε ορισμένες παραμέτρους, ώστε να πετύχουμε μια ισορροπία ανάμεσα στο οπτικά ρεαλιστικό και στο να μην είναι αρκετά «βαριά» η εφαρμογή που αναπτύσσουμε. Οι πιο σημαντικές παράμετροι που πρέπει να ενεργοποιηθούν είναι τα *RandomRotation*, *AlignToTerrain*, το *Density=20* και το *SizeVar=0,6*. Ανάλογα με το πόσο μεγάλη είναι η περιοχή που θέλουμε να καλύψουμε, μπορούμε να αυξήσουμε το μέγεθος της βούρτσας από το slider που βρίσκεται πάνω από το «*paint objects*». Για την περιοχή της ζούγκλας επιλέξαμε η βλάστηση να είναι πιο πυκνή στο κέντρο και γύρω από τον δρόμο, καθώς είναι πιο συνηθισμένο να κινηθεί εκεί ο παίκτης, ενώ για το υπόλοιπο τοπίο επιλέξαμε πιο αραιά τοποθέτηση βλάστησης.



Εικόνα 2.17: Τα assets (αρχεία) που χρησιμοποιήθηκαν συνολικά για το γρασίδι

Δημιουργία της λίμνης

Στην Cry Engine 3, η θάλασσα, τα ποτάμια και οι λίμνες αποτελούν ξεχωριστά στοιχεία και γ'αυτό το κάθε ένα έχει και τα δικά του εργαλεία και ρυθμίσεις. Το περιβάλλον μας περιέχει και δύο λίμνες, οι οποίες είναι τοποθετημένες στην αρχή και στο τέλος των τριών ποταμών. Το εργαλείο που θα χρησιμοποιήσουμε για τη δημιουργία της λίμνης βρίσκεται στα *Objects > Area > WaterVolume*. Το εργαλείο αυτό δίνει τη δυνατότητα να δημιουργήσουμε ένα οποιοδήποτε σχήμα στο περιβάλλον, οριοθετώντας κυκλικά τα σημεία του με τον κέρσορα επάνω στο τοπίο, δημιουργώντας έτσι την επιφάνεια μιας λίμνης. Στην παρακάτω φωτογραφία (Εικόνα 2.18) βλέπουμε με ανοιχτό μπλε χρώμα τα όρια της λίμνης και τις επιλογές του συγκεκριμένου εργαλείου.

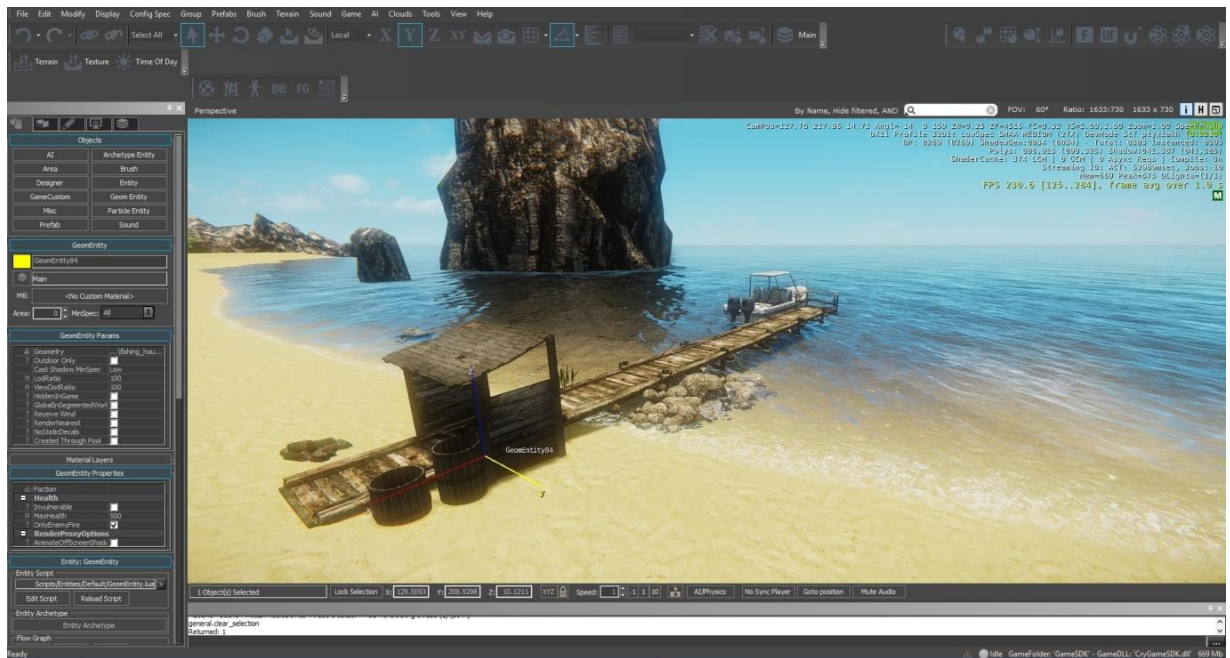


Εικόνα 2.18: Ορισμός του χώρου της λίμνης και τοποθέτηση της επιφάνειας του νερού

Μια σημαντική λεπτομέρεια, προκειμένου να καταλάβει ο editor ότι αυτά είναι τα όρια της λίμνης είναι, το πρώτο με το τελευταίο σημείο του σχήματος να εφάπτονται, έτσι ώστε να ολοκληρωθεί ο κύκλος ή το οποιοδήποτε σχήμα έχουμε επιλέξει. Προφανώς είχαμε διαμορφώσει από πριν έτσι το τοπίο μας ώστε να γνωρίζουμε ακριβώς το σημείο το οποίος θέλουμε να καλύψει η λίμνη, ενώ το μόνο που έπρεπε να προσέξουμε ήταν τα όρια στα οποία τελειώνει το object του ποταμιού με αυτά της λίμνης, να εφάπτονται ομαλά. Τέλος, πρέπει να επιλέξουμε ένα material (Mtl) για την επιφάνεια της λίμνης, ανάλογα με το πώς επιθυμούμε να δείχνει. Αυτό μπορούμε να το κάνουμε αριστερά από τις ιδιότητες που βλέπουμε στη φωτογραφία και στη συνέχεια επιλέγοντας το στοιχείο εμφάνισης από τον Material Editor και πατώντας «Assign Item to Selected Object».

Τοποθέτηση των αντικειμένων

Ένα σημαντικό στοιχείο που πρέπει να έχει κάθε περιβάλλον που προσπαθεί να δείχνει ρεαλιστικό είναι να προσπαθεί να χρησιμοποιεί τις επιλογές για μαζική τοποθέτηση αντικειμένων σε σημεία που θα είχαν έτσι και αλλιώς μια τυχαιότητα και στον πραγματικό κόσμο (όπως δέντρα, φυτά, πέτρες, κτλ), έτσι ώστε το μάτι να μην καταλαβαίνει εύκολα την επανάληψη. Η Cry Engine, μας δίνει τη δυνατότητα, ακόμα και στην μαζική τοποθέτηση αντικειμένων, να ορίσουμε το randomization τους, όπως είδαμε προηγουμένως με τη δημιουργία της ζούγκλας και των φυτών. Ωστόσο, για περιπτώσεις που θέλουμε να σχεδιάσουμε κάτι συγκεκριμένο, πρέπει να τοποθετήσουμε τα αντικείμενα ένα-ένα με το «χέρι». Αυτό μπορούμε να το κάνουμε από την επιλογή αριστερά «Geom Entity» (**Εικόνα 2.19**) και στη συνέχεια επιλέγοντας από τις κατηγορίες που εμφανίζονται τα αντικείμενα που μας ενδιαφέρουν.



Εικόνα 2.19: Τοποθέτηση διαφόρων αντικειμένων στον χάρτη, με το «χέρι»

Ορισμένα από τα αντικείμενα που έχουμε τοποθετήσει, είναι: κορμοί δέντρων στη παραλία, μια προβλήτα με ένα σκάφος, βράχια στα όρια της παραλίας, ορισμένα φυτά στη λίμνη και αρκετά αντικείμενα μέσα στη σπηλιά. Σημαντικά εργαλεία ώστε να μπορέσουμε να ταιριάξουμε το κάθε αντικείμενο με το περιβάλλον είναι η μεγέθυνση του αντικειμένου, η περιστροφή του και η μετακίνησή του στον τρισδιάστατο άξονα.

Το εργαλείο των Flow Graph

Πριν προχωρήσουμε στην επεξήγηση της χρήσης των εφέ και την διασύνδεσή τους με τους ήχους και τα ηχοτοπία των περιοχών του χάρτη, είναι σημαντικό να εξηγηθεί προηγουμένως η χρήση των Flow Graphs στην Cry Engine 3, καθώς πρόκειται για απαραίτητο εργαλείο, δίχως το οποίο θα ήταν αδύνατο να δοθεί η οποιαδήποτε σύνθετη και άμεση συμπεριφορά στο ηχοτοπίο. Τα Flow Graphs πρόκειται για ένα οπτικό scripting περιβάλλον προγραμματισμού, στο οποίο ο χρήστης μπορεί να συνδέσει όλα τα objects της μηχανής μεταξύ τους και να τους αποδώσει αλληλεπιδραστικές συμπεριφορές. Το περιβάλλον δεν απαιτεί από τον χρήστη να έχει γνώσεις scripting ή κώδικα και μπορεί να δημιουργήσει σύνθετες συμπεριφορές, για παράδειγμα, ανάμεσα στα objects των περιοχών (shapes), στα Ambient Volumes και στα οπτικά εφέ.

Αν και θα αναφερθούμε αναλυτικά για το κάθε Flow Graph που δημιουργήσαμε στα σχετικά σημεία που αφορούν τη δημιουργία των αντίστοιχων ηχοτοπιών περιβάλλοντος, ωστόσο μπορούν να αναφερθούν ορισμένα βασικά της τερμινολογίας των Flow Graphs, για την καλύτερη κατανόησή τους:

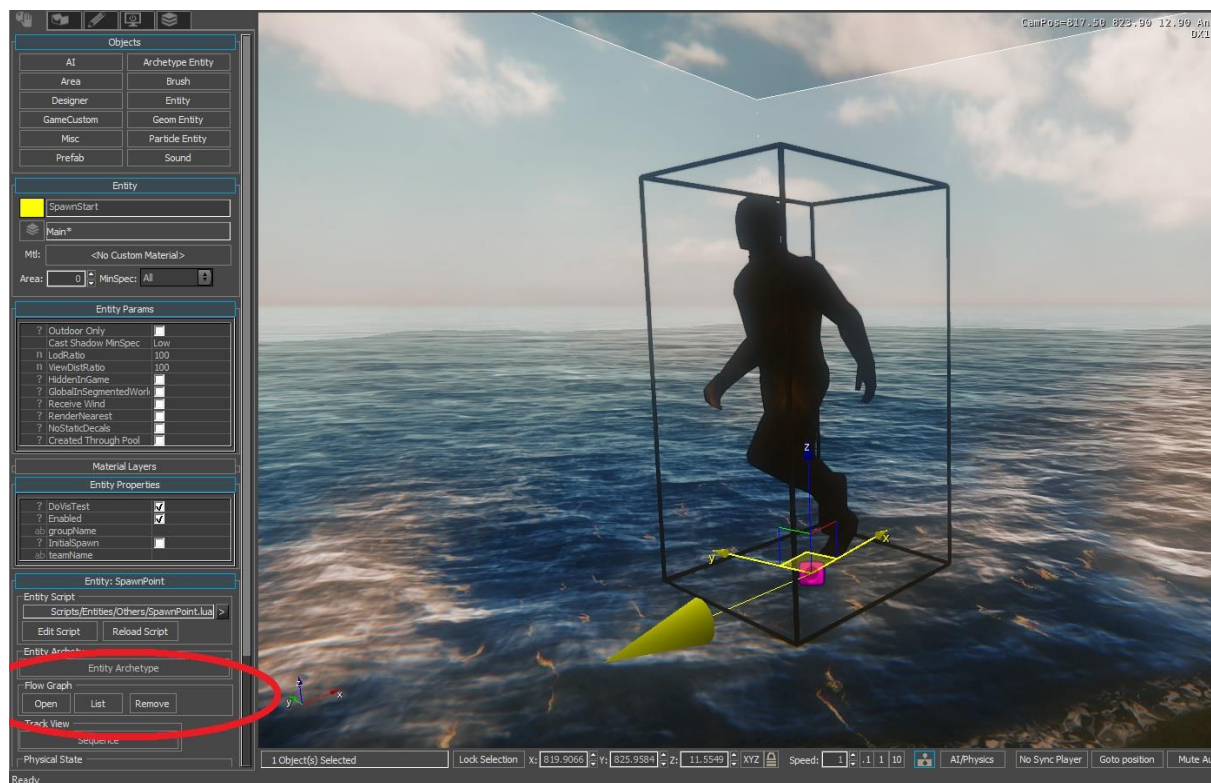
Terminology

Λέξη	Περιγραφή
Graph	Ένα από μόνο του Flow Graph αναφέρεται και ως <i>graph</i> .
Nodes	Τα Nodes αποτελούν την οπτική αναπαράσταση των entities (Entity node) ή των συνιστωσών (Component node) που διενεργούν συγκεκριμένες λειτουργίες.
Component Node	Τα Component nodes είναι ουσιαστικά nodes τα οποία δεν βρίσκονται ως entities μέσα στον χάρτη, αλλά προσφέρουν κάποια ιδιαίτερη λειτουργία στα flow graphs. Τα Component nodes χρειάζεται να καταλήγουν σε κάποιο entity set, στο οποίο και λειτουργούν.
Entity Node	Τα Entity Nodes αποτελούν entities μέσα στον χάρτη. Κάθε Entity έχει τα δικά του inputs και outputs, όπως έχουν ορισθεί στο καθένα.
Links	Τα Links είναι ο τρόπος με τον οποίον συνδέονται τα nodes. Τα Links εμφανίζονται ως «καλώδια» ή γραμμές και συνδέουν τις εισόδους και τις εξόδους των nodes.
Ports	Όλα τα Nodes έχουν εισόδους και εξόδους, οι οποίες ονομάζονται Ports. Τα Ports χρησιμοποιούνται για τη σύνδεση των Nodes με τα Links και είναι τα βελάκια χρώματος μπλε στις άκρες του πλαισίου ενός Node.
Graph Entity	Ένα Entity που περιέχει σωσμένο ένα ολόκληρο Graph, ονομάζεται και <i>graph entity</i> .

Spawn point: Το σημείο εκκίνησης του χάρτη

Προκειμένου να ορίσουμε το σημείο στο οποίο επιθυμούμε ο ακροατής/ χρήστης να ξεκινήσει την περιπλάνησή του στο τοπίο μας, πρέπει να ορίσουμε ένα spawn point. Σε περίπτωση που δεν ορισθεί κάτι τέτοιο, τότε αυτόματα η Cry Engine ξεκινάει τον παίκτη από τις συντεταγμένες 0,0,0 του χάρτη, κάτι που δεν είναι επιθυμητό. Ένας χάρτης μπορεί να έχει από ένα έως και άπειρα spawn points, ανάλογα με τις προϋποθέσεις κάθε φορά που έχει θέσει ο gameplay designer. Στο χάρτη της συγκεκριμένης εργασίας έχει ορισθεί ένα spawn point ως σημείο εκκίνησης και ένα ακόμα spawn point ως σημείο μεταφοράς του παίκτη σε άλλη περιοχή του χάρτη (portal).

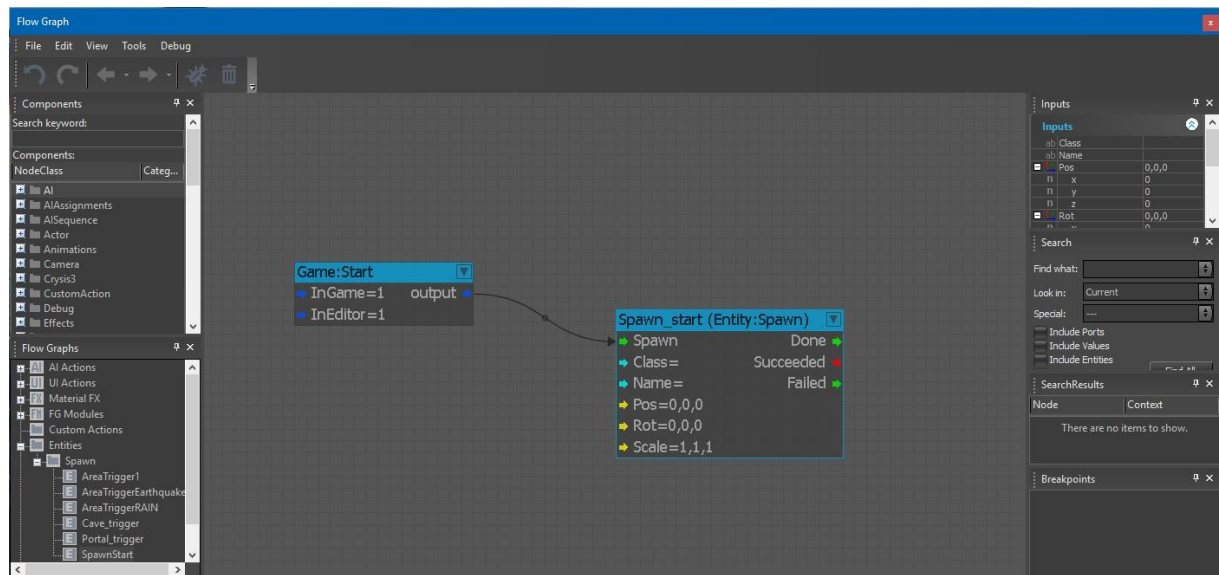
Τα spawn points αποτελούν ξεχωριστά objects μέσα στη μηχανή γραφικών και μπορούν να τοποθετηθούν ελεύθερα στο τρισδιάστατο περιβάλλον. Πηγαίνοντας στις επιλογές *Objects > Entity > Others > SpawnPoint* θα βρούμε το συγκεκριμένο object, το οποίο έχει τη συγκεκριμένη εμφάνιση και ελάχιστα properties, όπως φαίνεται και στην **Εικόνα 2.20**.



Εικόνα 2.20: Τοποθέτηση αντικείμενου *Spawn Point* και επιλογή εισαγωγής *Flow Graph* σε αυτό

Προκειμένου να ενεργοποιηθεί ένα *SpawnPoint* πρέπει να χρησιμοποιήσουμε το *Flow Graph* σύστημα της *Cry Engine*. Επιλέγοντας κάτω δεξιά στα properties του object την επιλογή «*Open*», στην ενότητα «*Flow Graph*», μπορούμε να προγραμματίσουμε μια συμπεριφορά για το συγκεκριμένο *Spawn Point*. Στη συγκεκριμένη περίπτωση επιθυμούμε επιλέξουμε να ενεργοποιείται κατά την έναρξη της διαδικασίας του παιχνιδιού, οπότε θα χρησιμοποιήσουμε την εντολή «*Game:Start*», το οποίο διαθέτει ένα output. Το επόμενο entity που θα χρειαστούμε είναι το ίδιο το *spawn point*. Έχοντάς το επιλεγμένο στο χάρτη, μπορούμε με δεξί κλικ επάνω στο *flow graph* να επιλέξουμε «*Add Selected Entity*», προκειμένου να εισάγουμε εύκολα το entity του στο *flow graph*. Στη συνέχεια συνδέουμε το output του *Game:Start* με το input «*Spawn*» του *Entity:Spawn*. Αυτό σημαίνει ότι το συγκεκριμένο entity θα ενεργοποιείται κάθε φορά που ξεκινάει ο χάρτης, μεταφέροντας έτσι τον παίκτη στο συγκεκριμένο σημείο που έχουμε ορίσει τοποθετώντας το στο περιβάλλον.

Στο συγκεκριμένο περιβάλλον επιθυμούμε ο παίκτης να ξεκινάει από τη περιοχή του ενός ηχοτοπίου της παραλίας, δίπλα στην κατεστραμμένη βάρκα. Παρακάτω στην **Εικόνα 2.21** βλέπουμε το *flow graph* που έχουμε ορίσει για το *spawn point* της εκκίνησης.



Εικόνα 2.21: Δημιουργία Flow Graph για το αρχικό Spawn Point του τοπίου

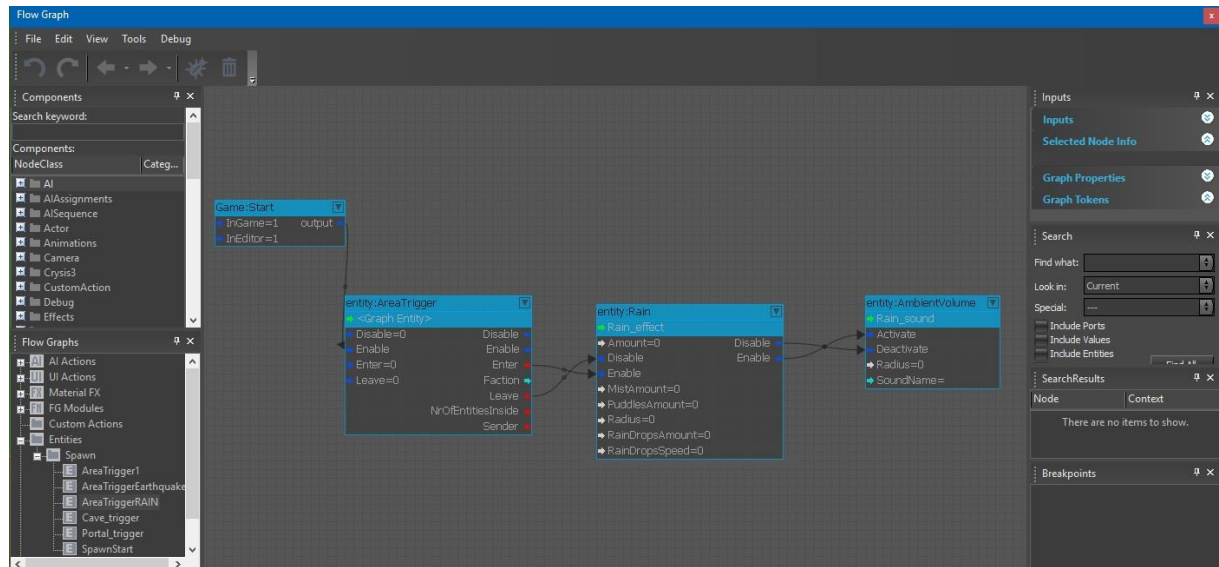
Area Trigger Objects

Πριν προχωρήσουμε στην επεξήγηση των εφέ που χρησιμοποιήθηκαν στο επίπεδο, είναι απαραίτητο να επεξηγηθεί η λειτουργία των AreaTrigger εργαλείων, τα οποία θα χρησιμοποιηθούν αρκετά στη συνέχεια στα Flow Graphs των οπτικών εφέ.

Τα AreaTriggers πρόκειται για Objects που βρίσκονται στην τοποθεσία *Objects > Entities > Triggers > AreaTrigger* και η βασική τους λειτουργία είναι να κάνουν enable το Port ενός Node του Flow Graph, τη στιγμή που ο χρήστης περάσει σε μια περιοχή με την οποία έχει συνδεθεί το συγκεκριμένο AreaTrigger Object. Η χρήση ενός AreaTrigger object μέσα στο FlowGraphs, έχει ως εξής: Αφότου τοποθετήσουμε το object στο χάρτη, μπορούμε να επιλέξουμε το Share που έχουμε δημιουργήσει, όπως για παράδειγμα αυτό για την βροχή και, πατώντας στα properties την επιλογή «pick» από το «target», μπορούμε να επιλέξουμε το AreaTrigger και να το συνδέσουμε στο Share της περιοχής. Το επόμενο βήμα είναι να επιλέξουμε το AreaTrigger και στα properties του Object να επιλέξουμε να δημιουργήσουμε ένα νέο Flow Graph (*create > new*). Στη συνέχεια θα μεταφερθούμε στον Flow Graph Editor όπου μπορούμε να ξεκινήσουμε να εισάγουμε nodes.

Το πρώτο που θα χρειαστούμε είναι το Game:Start. Το επόμενο είναι το ίδιο το AreaTrigger, το οποίο έχει την ονομασία entity:AreaTrigger. Προκειμένου να μην το αναζητάμε στην λίστα με όλα τα entities, μπορούμε εύκολα να το επιλέξουμε στο τοπίο και με δεξί κλικ να επιλέξουμε να το εισάγουμε απευθείας στο Flow Graph. Το επόμενο entity μπορεί να είναι το εφέ που θέλουμε να ενεργοποιηθεί με την είσοδό μας στην περιοχή, όπως στο παράδειγμα που φαίνεται στην παρακάτω φωτογραφία, για το εφέ της βροχής. Επιλέγοντας στο τοπίο το εφέ της βροχής, το εισάγουμε και αυτό με δεξί κλικ στο Flow Graph. Τέλος, προκειμένου να είναι ολοκληρωμένη η οπτικοακουστική διαδικασία, θα

χρειαστεί να εισάγουμε και ένα AmbientVolume object, με την ίδια διαδικασία, στο οποίο θα τοποθετηθεί το ηχοτοπίο της βροχής.



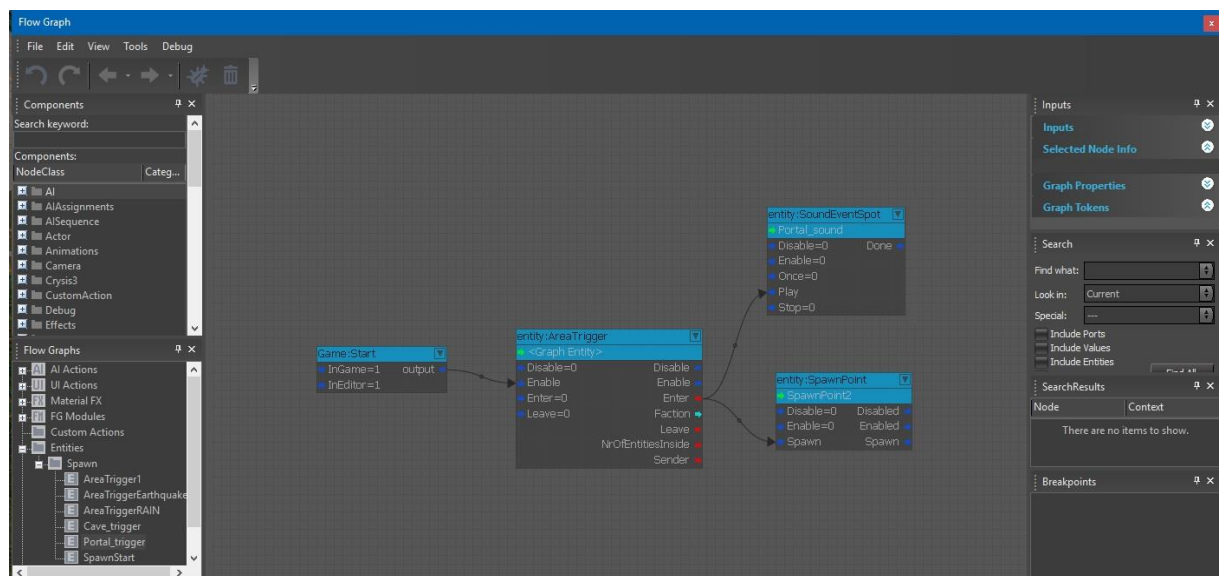
Εικόνα 2.22: Δημιουργία Flow Graph για τη συμπεριφορά του εφέ της περιοχής της βροχής

Η λογική του script που φαίνεται στην προηγούμενη φωτογραφία έχει ως εξής: Ο χάρτης ξεκινάει και το AreaTrigger ενεργοποιείται και περιμένει την είσοδο του χρήστη στη περιοχή → Ο χρήστης εισέρχεται στο σχήμα της περιοχής και → Το AreaTrigger με τη σειρά του ενεργοποιεί το επόμενο entity, όπως αυτό της βροχής → Το Entity της βροχής ενεργοποιείται και με τη σειρά του → Ενεργοποιεί τον Ambient ήχο της περιοχής. Όταν το AreaTrigger καταλάβει ότι ο χρήστης εξήλθε από την περιοχή → Απενεργοποιεί το εφέ της βροχής το οποίο με τη σειρά του → Απενεργοποιεί τον ήχο της βροχής. Σημαντικό προκειμένου όλο το παραπάνω script να ενεργοποιηθεί μία μόνο φορά είναι να έχουμε επιλεγμένη την επιλογή «Trigger Once», από τα properties του AreaTrigger. Αυτό σημαίνει ότι όσες φορές και να εισέλθει ξανά ο χρήστης στην περιοχή, η βροχή δεν θα ενεργοποιηθεί ξανά, δίνοντας έτσι την εντύπωση σκηνοθετικά, ότι ήταν μια τροπική ζαφνική μπόρα που πέρασε.

Προσθήκη αντικειμένου μεταφοράς του χρήστη

Μια ακόμη σημαντική προσθήκη που βοηθάει την γρήγορη περιήγηση του χρήστη στον χάρτη, είναι αυτή της συσκευής μεταφοράς την οποία έχουμε τοποθετήσει σε συγκεκριμένο σημείο του χάρτη. Ο λόγος ύπαρξης αυτής αυτού του αντικειμένου, το οποίο οπτικά το έχουμε ορίσει να θυμίζει μαγικό portal τηλεμεταφοράς, έχει τοποθετηθεί στο βόρειο τμήμα του χάρτη, αριστερά του μεγάλου καταρράκτη και έχει ως σκοπό να μειώσει στο ελάχιστο τον χρόνο που χρειάζεται κανείς για να μεταβεί από το βόρειο μέρος του χάρτη στο κέντρο της ζούγκλας.

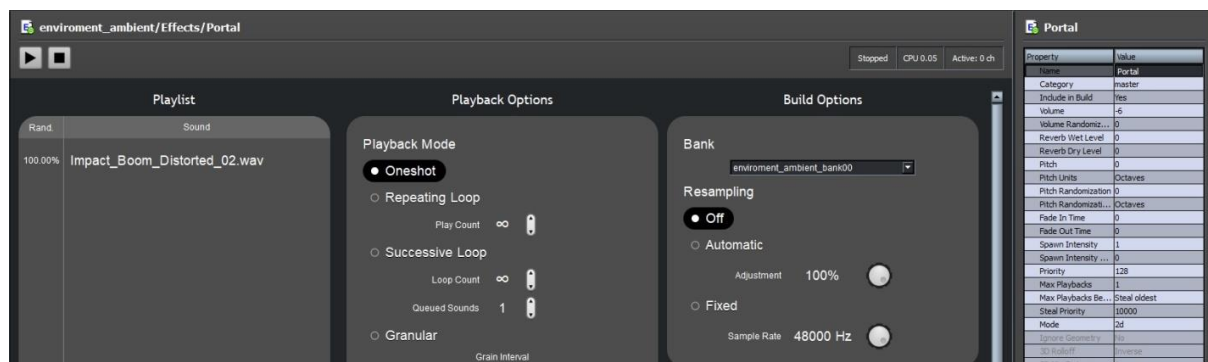
Η δημιουργία ενός τέτοιου σημείου μεταφοράς απαιτεί δύο σημεία: ένα σημείο εισόδου και ένα εξόδου. Για το πρώτο, απαιτείται η τοποθέτηση ενός επιπλέον Spawn Point στον χάρτη, το οποίο ορίζει το σημείο στο οποίο θα καταλήξει ο χρήστης. Για το σημείο εισόδου, ο τρόπος που έχει χρησιμοποιηθεί περιέχει ένα AreaTrigger Object και ένα Area Box Shape. Στο AreaTrigger είναι απαραίτητο να ορίσουμε ένα script συμπεριφοράς, το οποίο να εμπεριέχει στη λογική του και το Spawn Point που έχουμε τοποθετήσει, όπως θα δούμε στη συνέχεια. Το πρώτο βήμα είναι η τοποθέτηση του AreaBox Shape, το οποίο ορίζει το σημείο εισόδου. Στη συνέχεια τοποθετείται το AreaTrigger entity και το συνδέουμε με την επιλογή «Target Link» με το AreaBox shape. Αυτό σημαίνει ότι αυτά τα δύο entities πλέον αλληλεπιδρούν. Τέλος, το τελευταίο Object που πρέπει να προσθέσουμε έχει να κάνει με τον ήχο και είναι ένα απλό SoundEventSpot, καθώς επιθυμούμε, όταν ο χρήστης διασχίσει το σημείο μεταφοράς, να ακουστεί ένα ηχητικό εφέ.



Εικόνα 2.23: Δημιουργία Flow Graph για τον προγραμματισμό της συσκευής μεταφοράς του χρήστη

Επιλέγοντας το AreaTrigger θα δημιουργήσουμε ένα Flow Graph, προκειμένου να δημιουργήσουμε ένα Script. Το πρώτο Node που θα χρειαστούμε είναι το Game:Start και στη συνέχεια το entity:AreaTrigger. Τα επόμενα Nodes είναι τα entity:SoundEventSpot και entity:SpawnPoint και η συνολική συνδεσμολογία έχει ως εξής: Το Game:Start → Ενεργοποιεί το AreaTrigger με το που ξεκινήσει ο χάρτης, το οποίο με τη σειρά του → Αναμένει την είσοδο του παίκτη στο AreaBox με το οποίο έχει συνδεθεί. Όταν εισέλθει ο χρήστης στο AreaBox → Ενεργοποιούνται ταυτόχρονα τα: → (α) SpawnPoint και (β) SoundEventSpot. Το αποτέλεσμα θα είναι ο χρήστης να μεταφερθεί στο σημείο της περιοχής της ζούγκλας που έχουμε ορίσει, ακούγοντας ένα ηχητικό εφέ. Στην προηγούμενη **Εικόνα 2.23**, μπορούμε να δούμε συνολικά το Flow Graph που έχει χρησιμοποιηθεί

Για τη δημιουργία του ηχητικού εφέ χρησιμοποιήσαμε έναν ήχο 6 δευτερολέπτων από την βιβλιοθήκη Sonniss, με την ονομασία «Impact_Boom_Distorted_02», μεγέθους 3,3MB και στα 4608kbps. Το αρχείο WAV εισήχθη στο FMOD ως SimpleTrack Event και επιλέγοντας να ενεργοποιείται μία φορά με την επιλογή «OneShot», ενώ επιπλέον μειώσαμε και το Volume στα -6 dB (αναλυτική περιγραφή των εργαλείων του FMOD στο αντίστοιχο κεφάλαιο).



Εικόνα 2.24: Οι ρυθμίσεις του OneShot ήχου, με τη χρήση του FMOD, για τη συσκευή μεταφοράς

Εισαγωγή των οπτικών εφέ

Σε διάφορα σημεία του χάρτη έχουν χρησιμοποιηθεί τα λεγόμενα οπτικά εφέ, ή αλλιώς Particle Effects. Πρόκειται ουσιαστικά για Objects της Cry Engine τα οποία θα τα συναντήσουμε τόσο μέσα στις κατηγορίες Entities, όσο και Particle Entities. Τα αντικείμενα αυτά μπορούν να τοποθετηθούν οπουδήποτε στο τοπίο, όμως πολλά από αυτά, προκειμένου να λειτουργήσουν, χρειάζεται να συνδεθούν με κάποιο script που θα τους ορίζει μια συμπεριφορά, όπως αυτή που μπορούμε να επιτύχουμε με την χρήση των Flow Graphs.

Εφέ φωτιάς και καπνού: Προκειμένου να αποδοθεί πειστικότερα το εφέ της εστίας της φωτιάς, χρησιμοποιήσαμε συνολικά τρία Particle Effects.

Πρόκειται για τα:

```
«smoke_and_fire.ground_fire.burning_debris.small2.small4»,  
«smoke_and_fire.campfire.small.new_fire_burning1»  
«smoke_and_fire.ground_fire.burning_debris.small2.tank2».
```

Το πρώτο εφέ στη σειρά αποτελεί τον καπνό από τη φωτιά, το δεύτερο αποτελεί το φως που εκπέμπεται και το τρίτο αποτελεί την ίδια την φλόγα. Τα τρία αυτά εφέ έχουν τοποθετηθεί πολύ κοντά, αλλά με τρόπο που να παρομοιάζει μια πραγματική εστία φωτιάς, δηλαδή ο καπνός να βγαίνει πάνω από τη φωτιά, το φως να φαίνεται περιμετρικά, κ.ο.κ.. Και τα τρία εφέ που έχουν χρησιμοποιηθεί για τη φωτιά δεν είναι απαραίτητο να συνδέονται με κάτι άλλο ή να διαθέτουν κάποια συμπεριφορά, αρκεί να ενεργοποιηθούν

από τα properties τους ώστε να είναι active συνεχώς. Επιπλέον, κάθε εφέ μπορεί να ελεγχθεί περαιτέρω μέσα από το menu της database (επιλογή «Go To Database»), σε περίπτωση που θέλουμε να αλλάζουμε το χρώμα της φωτιάς τη διάρκεια, το μέγεθος, ή ακόμα και να εισάγουμε έναν ήχο στο εφέ (ωστόσο τη διαδικασία αυτή εμείς επιλέξαμε να την κάνουμε εξωτερικά, χρησιμοποιώντας ξεχωριστό SoundEventSpot Object).

Εφέ ομίχλης: Ένα από τα σημαντικά εφέ του χάρτη, το οποίο βοηθά σημαντικά στην εμφάνιση ενός από τα ηχοτοπία μας, είναι αυτό της ομίχλης. Πρόκειται για ένα Object το οποίο θα το βρούμε στα *Objects > Entity > Render > FogVolume* και μπορεί να τοποθετηθεί αυτούσιο στο χάρτη, χωρίς να χρειάζεται να συνδεθεί με κάτι άλλο. Είναι απαραίτητο να αλλάξουμε ορισμένα από τα properties του, προκειμένου το εφέ της ομίχλης να γίνει αρκετά πειστικό. Στην φωτογραφία (**Εικόνα 2.25**) μπορούμε να δούμε συνολικά τα settings που χρησιμοποιήσαμε προκειμένου να δημιουργηθεί μια μεγάλη έκταση στο χάρτη με ομίχλη. Στο χάρτη μας υπάρχουν συνολικά 2 εφέ ομίχλης, ένα με πιο πράσινη απόχρωση για το ηχοτοπίο πριν τη βροχή και ένα με μια πιο γκριζα απόχρωση για την περιοχή όπου υπάρχει και το εφέ της βροχής. Η αλλαγή αυτή στο χρώμα μπορεί να γίνει επίσης μέσα από τα properties του Object.



Εικόνα 2.25: Τοποθέτηση εφέ ομίχλης και οι ρυθμίσεις του αντίστοιχου object

Εφέ αέρα και πτώσης φύλλων: Δύο ακόμη εφέ που χρησιμοποιήθηκαν στην κεντρική περιοχή της ζούγκλας, είναι αυτά της πτώσης φύλλων και της σκόνης που μεταφέρεται με τον αέρα. Και τα δύο εφέ βρίσκονται μέσα στην Particle Entity κατηγορία. Τα συγκεκριμένα εφέ δεν συνδέονται άμεσα με κάποιο ηχητικό Object, όπως συνέβη με τις περιπτώσεις των εφέ της φωτιάς, όμως η προσθήκη τους είναι απαραίτητη καθώς βοηθούν στην αναπαράσταση του ανέμου ολόκληρου του ηχοτοπίου. Ο άνεμος στο ηχοτοπίο μας αλλάζει όσο ο ακροατής κατευθύνεται σε πιο βόρεια περιοχή, με την περιοχή της ζούγκλας

να διαθέτει την πιο ήρεμη ηχητική επένδυση του ανέμου και τις χιονισμένες περιοχές να έχουν πιο δυνατό άνεμο με τα πιο έντονα φυσήματα. Η συγκεκριμένη ηχητική διαφορά έχει αποτυπωθεί στο οπτικό μέρος με την πιο έντονη κίνηση των δέντρων, με τα δέντρα στις πιο βόρειες περιοχές να κινούνται εντονότερα, κάτι που ορίζεται εύκολα μέσα από την «*Bending*» επιλογή στον Vegetation Editor.

Το εφέ για τον αέρα που χρησιμοποιήθηκε είναι το «*environment_fx.ambience.bound_ground_fog_dark.density.swirl*» και μπορεί να τοποθετηθεί οπουδήποτε στο χάρτη, καλύπτοντάς τον ολόκληρο. Όπως και στον πραγματικό κόσμο, έτσι και εδώ ο αέρας είναι στην ουσία αόρατος στο ανθρώπινο μάτι και φαίνεται παρά μόνο αν παρασέρνει κάποιο αντικείμενο ή σκόνη. Το συγκεκριμένο εφέ οπότε πρόκειται ουσιαστικά για εφέ σκόνης, η οποία κινείται προς διάφορες κατευθύνσεις, δίνοντας έτσι την εντύπωση ότι υπάρχει άνεμος. Το εφέ της πτώσης των φύλλων που χρησιμοποιήθηκε είναι το «*weather.environment_dust.forest*» και σε αντίθεση με το εφέ του αέρα, διαθέτει μικρότερο αποτέλεσμα στο χάρτη. Συγκεκριμένα έχουμε τοποθετήσει πολλαπλά τέτοια Objects κατά μήκος του κεντρικού δρόμου και μόνο στην περιοχή της ζούγκλας, προκειμένου να καλυφθεί ολόκληρη η έκταση της.

Εφέ βροχής: Το εφέ της βροχής αποτελεί Entity Object, το οποίο βρίσκεται στην τοποθεσία: *Objects > Entity > Environmental > Rain*. Το συγκεκριμένο εφέ πρέπει να διαθέτει κάποια συμπεριφορά προκειμένου να εμφανίζεται στη περιοχή που επιθυμούμε, καθώς δεν θέλουμε η βροχή να εμφανίζεται σε ολόκληρο το χάρτη. Σκοπός είναι το εφέ της βροχής είναι να διαθέτει το δικό του ξεχωριστό sound design, το οποίο θα συνυπάρχει με το Ambient object της περιοχής της ομίχλης. Επιπλέον, επιθυμούμε η βροχή να αποτελεί ένα event που θα συμβεί για μια μόνο φορά και δεν θα επαναληφθεί. Επομένως όλες αυτές οι συμπεριφορές πρέπει να οριστούν με script μέσα από τα Flow Graphs.

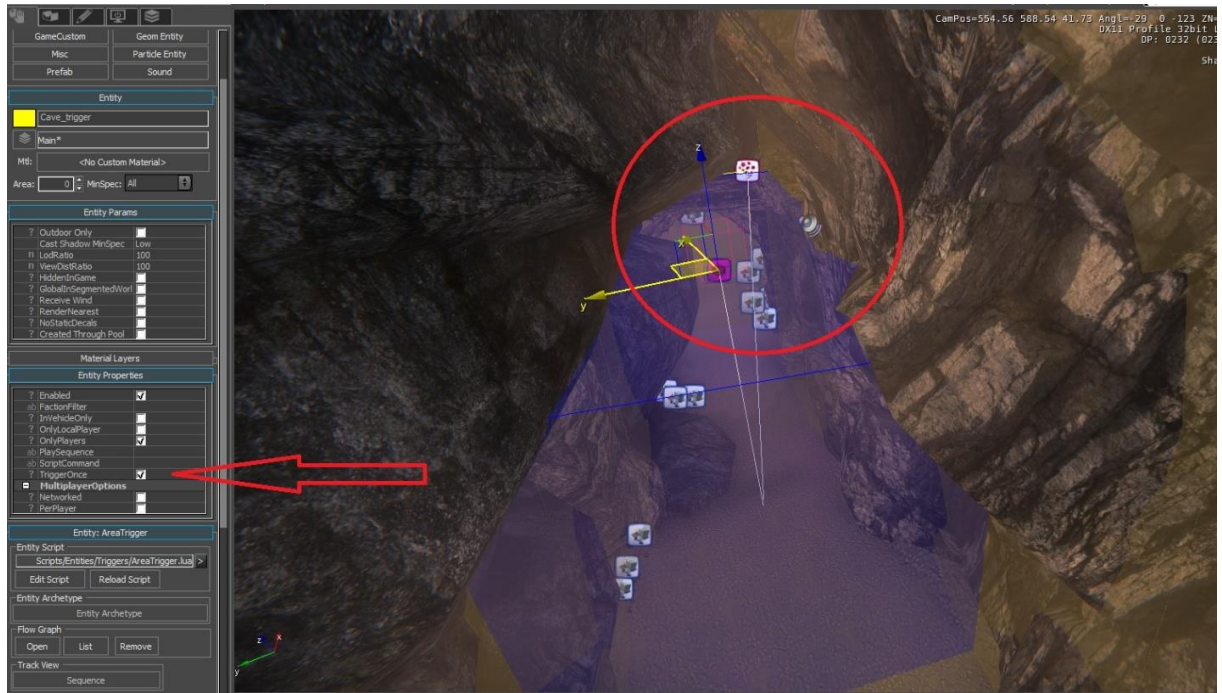
Το πρώτο που πρέπει να προσθέσουμε στο τοπίο είναι να ορίσουμε την περιοχή στην οποία θέλουμε να ενεργοποιείται το εφέ της βροχής. Αυτό μπορούμε να το κάνουμε με την χρήση των Shapes. Στη παρακάτω φωτογραφία (**Εικόνα 2.26**) με το ροζ χρώμα φαίνεται ο χώρος τον οποίο έχουμε δημιουργήσει για την βροχή.



Εικόνα 2.26: Ορισμός της περιοχής της βροχής με τη χρήση των Shapes

Το επόμενο βήμα είναι να τοποθετήσουμε το AreaTrigger object, αλλά και ένα AmbientVolume object και να ακολουθήσουμε τη διαδικασία δημιουργίας ενός script συμπεριφοράς, με τη χρήση των Flow Graphs, όπως εξηγήσαμε προηγουμένως στην ενότητα των Area Trigger Objects. Το αποτέλεσμα θα είναι η βροχή να ξεκινάει, μαζί με τον ήχο της, με το που εισέλθει ο χρήστης στη συγκεκριμένη περιοχή και να σταματάει όταν εξέλθει από αυτήν, χωρίς να επαναλαμβάνεται, δίνοντας έτσι την εντύπωση ότι πρόκειται για μια παροδική τροπική μπόρα. Η μεγαλύτερη κόκκινη περιοχή στην οποία βρίσκεται μέσα η περιοχή της βροχής αποτελεί την περιοχή της ομίχλης είναι ουσιαστικά η περιοχή στο οποίο τα δύο ηχοτοπία συνυπάρχουν με το ίδιο Ambient ηχητικό αποτέλεσμα.

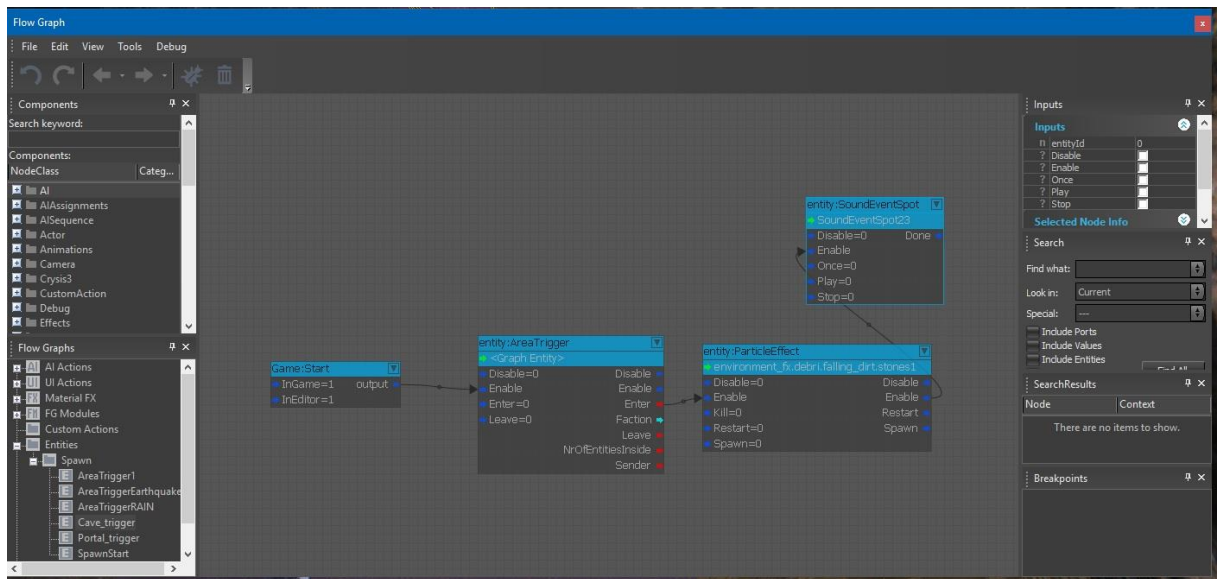
Εφέ πτώσης βράχων: Με την είσοδο του χρήστη στην σπηλιά θα ενεργοποιηθεί άλλο ένα μοναδικό event, του οποίου η λογική μοιάζει αρκετά με την λογική του event της βροχής. Λίγα μέτρα μετά την είσοδο στην σπηλιά έχουμε δημιουργήσει ένα κουτί (AreaBox), το οποίο θα αποτελέσει το σημείο ενεργοποίησης για το AreaTrigger entity. Το εφέ που θα χρησιμοποιήσουμε εδώ έχει να κάνει με την πρώτη βράχων και είναι το: «*environment_fx.debri.falling_dirt.stones*» και θα το βρει κανείς στα Particle Entities, ενώ επιπλέον θα χρειαστούμε και ένα Object για το ηχητικό εφέ της πτώσης των βράχων. Στην επόμενη φωτογραφία (**Εικόνα 2.27**) μπορούμε να δούμε το πώς έχουν τοποθετηθεί τα τρία objects, στο επάνω μέρος του κουτιού.



Εικόνα 2.27: Τοποθέτηση των objects για το εφέ της πτώσης των βράχων και οι ρυθμίσεις του αντίστοιχου AreaTrigger Object

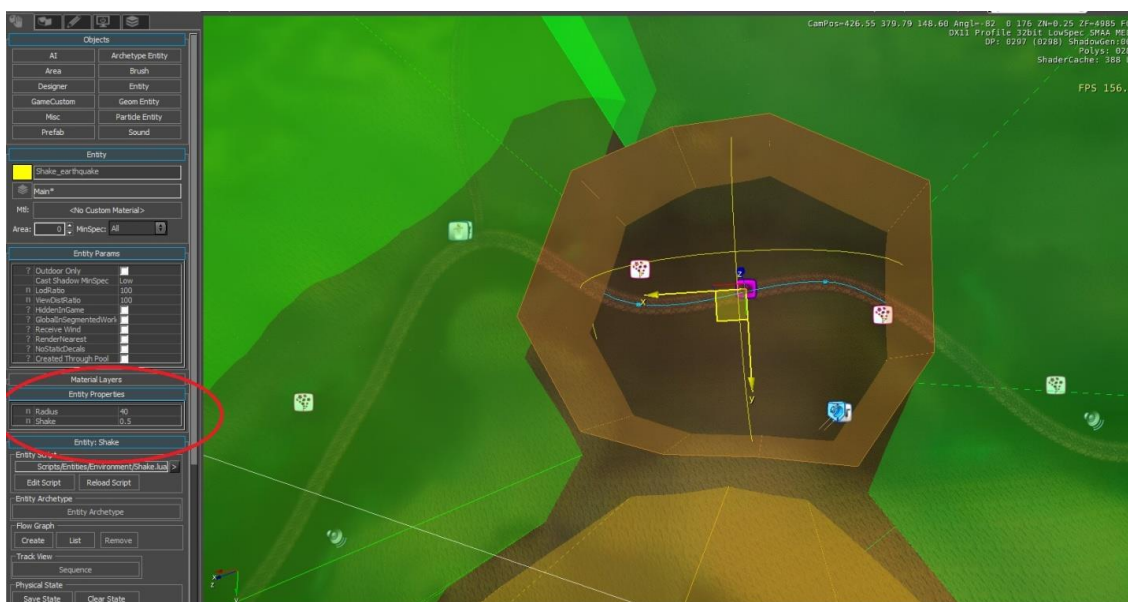
Το επόμενο βήμα είναι να συνδέσουμε με Link το AreaTrigger με το μπλε AreaBox και να ξεκινήσουμε τη διαδικασία δημιουργίας ενός ακόμα Flow Graph για το συγκεκριμένο AreaTrigger. Μέσα στο Flow Graph θα χρειαστούμε και πάλι το Node Game:Start, στη συνέχεια το ίδιο το AreaTrigger entity και στη συνέχεια το εφέ της πτώσης των βράχων. Τέλος, όπως και με το παράδειγμα της βροχής θα χρειαστεί να εισάγουμε τελευταίο το Object του ήχου.

Η συνδεσμολογία φαίνεται συνολικά στην παρακάτω φωτογραφία (**Εικόνα 2.28**) και η λογική του script είναι η εξής: Ο χάρτη ξεκινάει και το Game:Start → Ενεργοποιεί το AreaTrigger, το οποίο περιμένει την είσοδο του χρήστη. Όταν εισέλθει στη μπλε περιοχή το AreaTrigger → Ενεργοποιεί το εφέ της πτώσης βράχων, το οποίο με τη σειρά του → Ενεργοποιεί τον ήχο. Το όλο event συμβαίνει και σε αυτή την περίπτωση μία φορά και για αυτόν τον λόγο έχουμε ενεργοποιήσει την επιλογή «TriggerOnce» από τα properties του AreaTrigger entity.



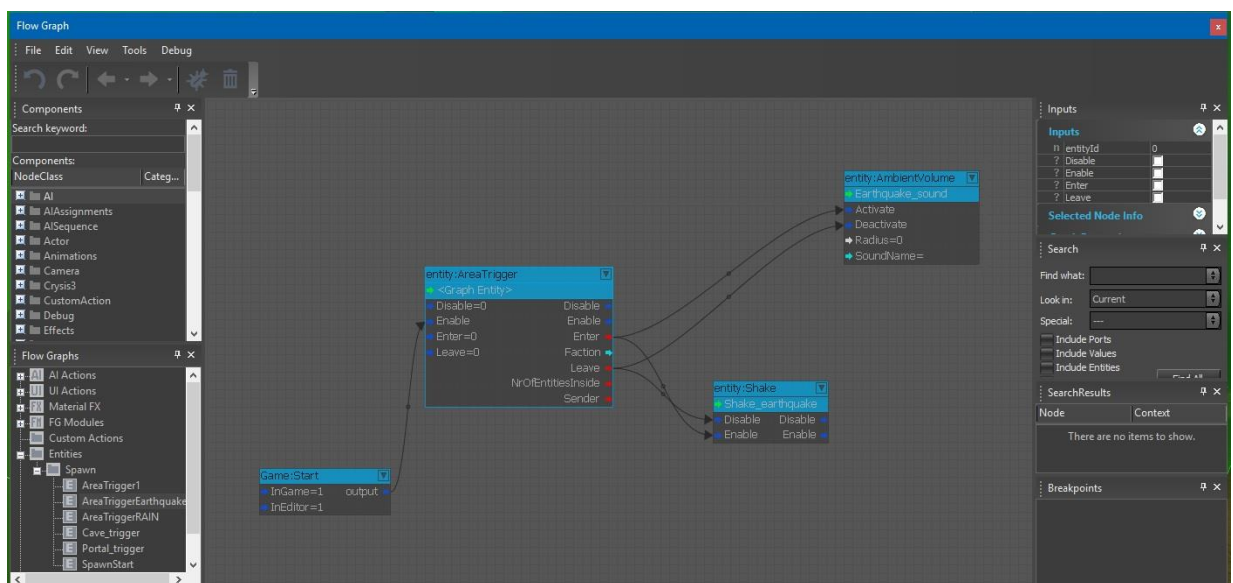
Εικόνα 2.28: Η δημιουργία του Flow Graph για τον προγραμματισμό του εφέ της πτώσης βράχων στη περιοχή της σπηλιάς

Εφέ σεισμού: Για τη δημιουργία του εφέ του σεισμού έχουμε χρησιμοποιήσει το Shake Object που βρίσκεται στην εξής τοποθεσία: *Objects > Entities > Enviroment > Shake*. Ουσιαστικά δεν πρόκειται για πραγματικό εφέ σεισμού που ενεργοποιεί με κάποιον τρόπο κίνηση στο τοπίο, αλλά για εφέ που κουνάει την κάμερα του χρήστη, πλησιάζοντας πολύ κοντά το αντίστοιχο αποτέλεσμα ενός σεισμού. Το συγκεκριμένο Object δεν το συνδέσαμε με την περιοχή την οποία έχουμε δημιουργήσει επάνω στην πορεία του κεντρικού μονοπατιού, καθώς διαθέτει από μόνο του ένα κυκλικό εύρος δράσης. Συγκεκριμένα έχουμε τροποποιήσει στα properties του Object την ακτίνα δράσης του στα 40 μέτρα και το εφέ του κουνήματος στο 0,5. Το εφέ του σεισμού τοποθετήθηκε στο κέντρο της περιοχής που έχουμε δημιουργήσει, όπως φαίνεται στην παρακάτω φωτογραφία (**Εικόνα 2.29**).



Εικόνα 2.29: Ορισμός της περιοχής του σεισμού και οι ρυθμίσεις του «Shake» εφέ

Το επόμενο βήμα είναι να ορίσουμε μια συμπεριφορά με την χρήση των Flow Graphs, όπως εξηγήσαμε στις προηγούμενες περιπτώσεις για την βροχή και την πτώση βράχων στη σπηλιά. Σκοπός είναι, όταν εισέλθει ο χρήστης στην περιοχή, η κάμερα να αρχίζει να τρέμει και το ηχοτοπίο να αλλάζει, εκφράζοντας τον πανικό που δημιουργείται. Αυτό θα αναπαρασταθεί τόσο με τους εντονότερους ήχους ζώων όσο και με το βουητό που θα δημιουργείται από το γεγονός του σεισμού. Όταν εξέλθει ο χρήστης από την εν λόγω περιοχή και σταματήσει ο σεισμός, το ηχοτοπίο θα επανέλθει στην φυσιολογική του κατάσταση. Η παραπάνω συμπεριφορά δημιουργήθηκε με το εξής script στα Flow Graphs, όπως φαίνεται στην παρακάτω φωτογραφία (**Εικόνα 2.30**).



Εικόνα 2.30: Δημιουργία Flow Graph για τη σύνδεση του εφέ Shake και του ήχου, για την περιοχή του σεισμού

Εφέ καταρρακτών: Τα εφέ που χρησιμοποιήθηκαν για τους καταρράκτες είναι πολλαπλά και τοποθετήθηκαν επάνω στα βράχια που έχουμε προηγουμένως εισάγει στο τοπίο μας, προκειμένου να δώσουν την εντύπωση του νερού που πέφτει από ύψος. Η χρήση διαφορετικών εφέ του ίδιο τύπου ήταν απαραίτητη έτσι ώστε να δημιουργηθεί μια τυχαιότητα στην εικόνα και να μην υπάρχει έντονη επανάληψη των assets. Τα εφέ αυτά μπορούν να βρεθούν στην κατηγορία *Objects > Particle Entity > Water > Waterfalls* και έχουν τοποθετηθεί στο περιβάλλον με τον τρόπο που φαίνεται στην παρακάτω φωτογραφία (**Εικόνα 2.31**). Παρόλο που μπορούμε να εισάγουμε ήχο απευθείας στο κάθε εφέ μέσα από την database τους, έχουμε επιλέξει και εδώ να αναπαραστήσουμε τον ήχο από εξωτερικό *SoundEventSpot*, για μεγαλύτερη ευκολία σχετικά με τον ορισμό της κατεύθυνσης του ήχου.



Εικόνα 2.31: Τοποθέτηση εφέ νερού για δημιουργία καταρράκτη

Κεφάλαιο 3: Ανάπτυξη του Ηχητικού Σχεδιασμού μέσω του FMOD Designer

Εισαγωγή στο FMOD

Η εισαγωγή ενός ήχου στο τρισδιάστατο περιβάλλον μπορεί να γίνει με δύο τρόπους. Ο πρώτος είναι περισσότερο πρόχειρος και δεν συνιστάται, καθώς προσφέρει περιορισμένες λειτουργίες γύρω από τη συμπεριφορά του ήχου –η εισαγωγή του αρχείου ήχου κατευθείαν μέσα στην Cry Engine 3. Ο δεύτερος τρόπος είναι κάνοντας χρήση ορισμένων «middleware» προγραμμάτων, τα οποία εξειδικεύονται στην επικοινωνία με τις γνωστότερες μηχανές παιχνιδιών, όπως τις Unreal Engine, Unity και την Cry Engine που χρησιμοποιούμε εμείς. Τα πιο δημοφιλή προγράμματα που είναι σχεδιασμένα για sound designers είναι αυτή τη στιγμή δύο: τα Wwise και FMOD Designer, το οποίο πλέον θα αντικατασταθεί από το FMOD Studio.

Το FMOD Designer 2010 που χρησιμοποιήσαμε σε αυτή την εργασία, διατίθεται ως δωρεάν download από το site της εταιρείας (<http://www.fmod.com/>) και αποτελεί το εργαλείο για τη δημιουργία των δυναμικών ηχητικών events που απαιτούνται στα videogames. Το περιβάλλον εργασίας του FMOD είναι σχεδιασμένο με τέτοιο τρόπο ώστε να μην απαιτούνται γνώσεις προγραμματισμού από το χρήστη, ο οποίος μπορεί να έχει τον απόλυτο έλεγχο της συμπεριφοράς των ήχων μέσα στο τρισδιάστατο ή το δισδιάστατο περιβάλλον.

Ορισμένες από τις σημαντικότερες λειτουργίες του FMOD είναι αυτές των layers, των παραμέτρων, των αυτοματισμών, του crossfading, των DSP (Digital Sound Processing) εφέ που μπορούμε να εισάγουμε σε κάθε layer, ενώ πολύ σημαντική λειτουργία είναι αυτή της «τυχαίας συμπεριφοράς», ή αλλιώς, randomization. Εκεί που διαφοροποιούνται σημαντικά τα βιντεοπαιχνίδια σε σχέση με άλλα είδη τέχνης γραμμικού μοντάζ (όπως ταινίες, σειρές κτλ), είναι ακριβώς ότι, όλα τα παραπάνω συμβαίνουν σε πραγματικό χρόνο μέσα στο παιχνίδι. Η επικοινωνία του FMOD με τη μηχανή παιχνιδιού (Cry Engine) γίνεται σε επίπεδο παραμέτρων και όχι κατευθείαν σε επίπεδο αρχείων ήχων, κάτι που είναι πολύ σημαντικό.

Σκοπός του FMOD είναι να δημιουργήσει ένα output αρχείο παραμέτρων (FDP), το οποίο περιέχει όλες τις συμπεριφορές ενός ήχου ή ενός ολόκληρου project άπειρων ήχων και μπορεί να «διαβαστεί» σε πραγματικό χρόνο από τη μηχανή παιχνιδιού. Προγράμματα όπως το FMOD υποστηρίζουν επίσης outputs για όλες τις διαθέσιμες πλατφόρμες και συστήματα, όπως είναι τα PC, Xbox, PlayStation, Nintendo, android και άλλα. Στη συγκεκριμένη εργασία χρησιμοποιούμε ως output την επιλογή για τα PC.

Προβλήματα και λύσεις με το μέγεθος των ήχων

Ο τομέας του ήχου των βιντεοπαιχνιδιών έχει αυξήσει σημαντικά τα τελευταία χρόνια το μέγεθος των τίτλων, με πολλά projects να διαθέτουν πλέον μέχρι και πάνω από 10GB ηχητικού υλικού. Συνεπώς, η χωρητικότητα αποτελούσε πάντοτε πρόβλημα για τους sound designers και πρέπει να υπάρχει συνεχώς προσπάθεια εξοικονόμησης πόρων, έτσι ώστε το μέγεθος του παιχνιδιού να μην είναι ασύμφορο για τον τελικό καταναλωτή, αλλά και να μην δημιουργείται πρόβλημα με την υπόλοιπη ομάδα που εργάζεται στον ίδιο τίτλο, καθώς όλοι «ανταγωνίζονται» πάνω στον ίδιο χώρο. Υπάρχουν διάφορες τεχνικές συμπίεσης, προκειμένου να μειωθεί και άλλο το μέγεθος του ηχητικού υλικού, όμως είναι επίσης συνηθισμένο πολλά παιχνίδια να έχουν uncompressed ηχητικά assets, έτσι ώστε να υπάρχει μεγαλύτερη διαύγεια στο τελικό προϊόν.

Προκειμένου να καταλάβουμε τις μεταβολές που υπήρξαν τα τελευταία χρόνια στον τομέα του ήχου, θα μπορούσαμε να χωρίσουμε τα ηχητικά assets σε 3 βασικές κατηγορίες. Η πρώτη κατηγορία είναι αυτή της μουσικής. Όπως αναφέρθηκε και στο ιστορικό μέρος, από τότε που εισήχθησαν τα CD ROM στην αγορά, τα videogames ξεκίνησαν να χρησιμοποιούν πλήρη soundtracks, με το μέγεθος του μουσικού υλικού να αυξάνεται και άλλο στην εποχή του DVD format αλλά και επιπλέον με το BluRay και τις μεγάλες χωρητικότητες των σκληρών δίσκων. Σε πολλά σημεία ενός τίτλου ο sound designer μπορεί να θελήσει να χρησιμοποιήσει αυτούσιο ένα μουσικό θέμα και, είτε να σταματήσει να παίζει με το που τελειώσει το κομμάτι, είτε να εισάγει μια απλή συμπεριφορά επανάληψης.

Ωστόσο ο τομέας της μουσικής των παιχνιδιών έχει βοηθηθεί σημαντικά τις τελευταίες δεκαετίες με μια πιο έξυπνη χρήση του μουσικού υλικού, όπως είναι αυτή της δυναμικής μουσικής. Ο designer μπορεί να γράψει ένα μικρό μουσικό θέμα μερικών δευτερολέπτων, το οποίο θα επαναλαμβάνεται υπό κάποιους όρους, οι οποίοι ορίζονται από middleware προγράμματα όπως το FMOD. Αυτό το μουσικό θέμα μπορεί να εναλλάσσεται συνέχεια με άλλα, ανάλογα με τις πράξεις του παίκτη και ανάλογα με τη σκηνοθεσία που θέλει να πετύχει ο δημιουργός. Για παράδειγμα, μπορεί να υπάρξει ένα πιο αργό μουσικό θέμα μερικών δευτερολέπτων, για σημεία στα οποία δεν υπάρχει κάποιος εχθρός για τον παίκτη στην οθόνη του, το οποίο να εναλλάσσεται με κάποιο πιο γρήγορο θέμα με το που καταλάβει το FMOD ότι παρουσιάστηκε έστω και ένας αντίπαλος στην οθόνη.

Με τη δυναμική μουσική καταφέρνουμε να έχουμε ένα αλληλεπιδραστικό ηχητικό αποτέλεσμα άπειρης θεωρητικά διάρκειας, κάτι που ταιριάζει απόλυτα με την απρόβλεπτη φύση ενός videogame, ενώ ταυτόχρονα καταφέρνουμε να μειώσουμε σημαντικά το μέγεθος των αρχείων που χρησιμοποιούνται. Μια ακόμη κατηγορία ήχων είναι φυσικά οι ήχοι των εφέ και των ambient ήχων. Αυτός ο τομέας, όπως θα δούμε και στη πράξη, είναι ο μοναδικός που μπορούμε να εξοικονομήσουμε το περισσότερο μέγεθος, πάλι κάνοντας χρήση δυνατοτήτων και τεχνικών που δίνονται μέσα στο FMOD.

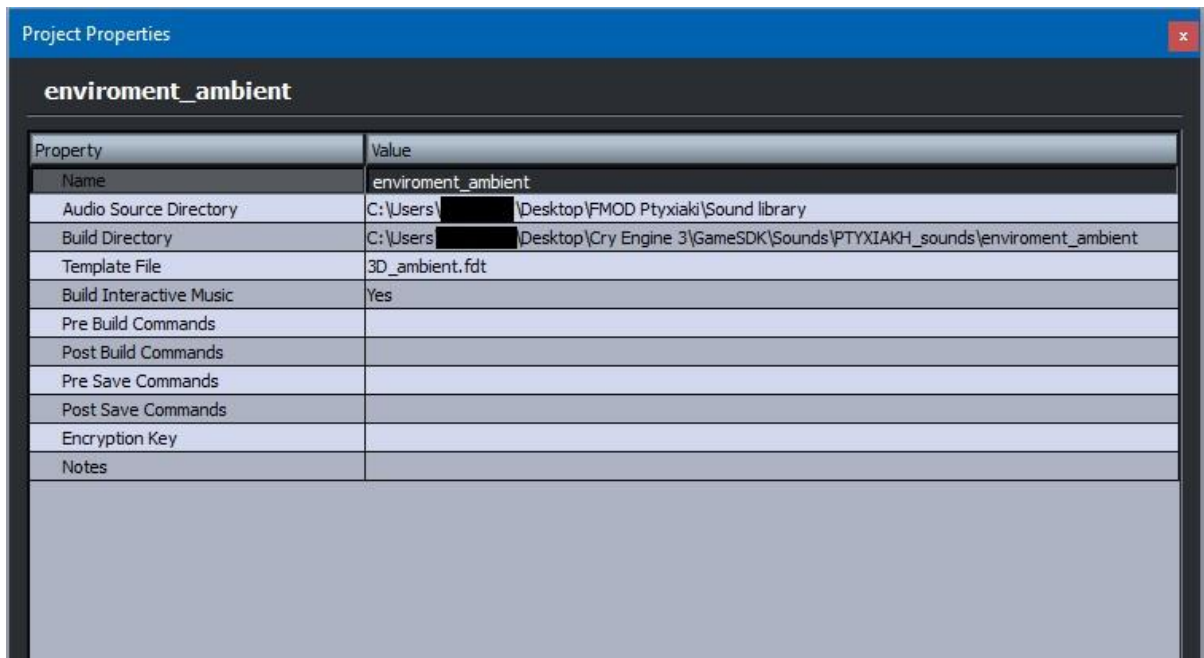
Τέλος, αν κάτι έχει αυξηθεί δραματικά τα τελευταία χρόνια στο συνολικό μέγεθος ενός videogame τίτλου, είναι το ηχητικό υλικό από τους διαλόγους και τις ομιλίες των ηθοποιών, οι οποίοι καλούνται σε ολοένα και περισσότερες παραγωγές για να ηχογραφηθούν. Ένα μουσικό θέμα μπορεί να είναι μόλις μερικά δευτερόλεπτα και να πραγματοποιεί δυναμική επανάληψη για πολλές ώρες, ενώ ένας διάλογος δεν μπορεί να επαναχρησιμοποιηθεί, καθώς αποτελεί ένα μοναδικό γεγονός το οποίο έχει ηχογραφηθεί για κάποιο συγκεκριμένο κομμάτι της ιστορίας ή της αφήγησης. Συνεπώς, πρέπει να χρησιμοποιηθεί αυτούσιο. Αν πραγματοποιηθεί μεγάλη συμπίεση αυτό θα έχει αντίκτυπο στη ποιότητα του ήχου της ομιλίας, ενώ δεν είναι και λογικό να δοθεί η φωνή κάποιου γνωστού ηθοποιού, προκειμένου να υπάρξει ένα ρεαλιστικό και ποιοτικό αποτέλεσμα, για να καταστραφεί τελικά αυτό στη συμπίεση. Προφανώς τέτοιες περιπτώσεις εξαρτώνται από το είδος του παιχνιδιού, με κάποια να μην έχουν μεγάλες απαιτήσεις σε διαλόγους (όπως κάποιο racing παιχνίδι), ενώ άλλα να βασίζονται σε αυτούς (όπως RPG παιχνίδια).

Στη συγκεκριμένη εργασία κάνουμε χρήση κυρίως ambient ήχων και για αυτόν τον λόγο θα παρατηρήσουμε καλύτερα τις τεχνικές που μπορούμε να χρησιμοποιήσουμε στα αρχεία ήχου, έτσι ώστε να υπάρξει καλύτερη διαχείριση των assets.

Το περιβάλλον εργασίας του FMOD Designer

Πριν ξεκινήσουμε ένα project στο FMOD είναι καλό να ορίσουμε από πριν ένα Audio Source Directory και ένα Build Directory. Ο πρώτος φάκελος είναι αυτός από τον οποίο θα διαβάσει το πρόγραμμα τα αρχικά αρχεία ήχου, όπως κάποια βιβλιοθήκη με assets ήχων που έχουμε ήδη βρει ή ηχογραφήσει, ενώ το δεύτερο αποτελεί τον φάκελο-έξοδο μέσα στη μηχανή παιχνιδιού, στον οποίο το FMOD θα δημιουργήσει όλα τα αρχεία που χρειάζονται ώστε να διαβαστούν οι παράμετροι από τη μηχανή. Αν και το FMOD δεν αλλάζει τα αρχικά αρχεία ήχου (όπως τα WAV) αλλά απλώς τα διαβάσει και δημιουργεί μια σειρά από δικά του build αρχεία (FDP, FSV, FAV, BAK και άλλα) που περιέχουν όλες τις παραμέτρους μας, καλό είναι ωστόσο να υπάρχει πάντα back up των αρχικών ήχων, σε περίπτωση corruption ή κατά λάθος σβησιμάτων από copy-paste. Μια σημαντική παράμετρος πριν το τελικό build είναι να επιλέξουμε στο tab «Banks» οι ήχοι να γίνονται «Streaming From Disk», έτσι ώστε να μην υπάρχει καθυστέρηση στο φόρτωμα των αρχείων μέσα στη Cry Engine.

Για τον φάκελο των sound libraries επιλέξαμε να δημιουργήσουμε έναν καινούργιο φάκελο στον υπολογιστή και μέσα να τοποθετήσουμε όλα τα ηχητικά assets που διαθέτουμε, έτσι ώστε να έχουμε άμεση πρόσβαση σε όλα. Για το build directory, κάτι που είναι πολύ σημαντικό να γνωρίζουμε, διαφορετικά η μηχανή γραφικών δεν μπορεί να διαβάσει τα αρχεία του FMOD είναι, η δημιουργία του build αρχείου **να έχει την ίδια ονομασία** με αυτή του build φακέλου. Για παράδειγμα, δημιουργήσαμε έναν φάκελο «environmental_ambient» μέσα στον οποίο το τελικό build αρχείο του FMOD καταλήγει σε «environmental_ambient.fdp».



Εικόνα 3.1: Ορισμός των φακέλων εισαγωγής και δημιουργίας (εξαγωγής) αρχείων στο FMOD

Καλό είναι πριν ξεκινήσουμε κάποιο project να δημιουργήσουμε μια πρώτη δομή οργάνωσης με τα εργαλεία που μας δίνει το FMOD. Τα πιο σημαντικά που θα χρησιμοποιήσουμε για την οργάνωση του project είναι τα Templates, τα Event Groups και τα Multitrack και Simple Events.

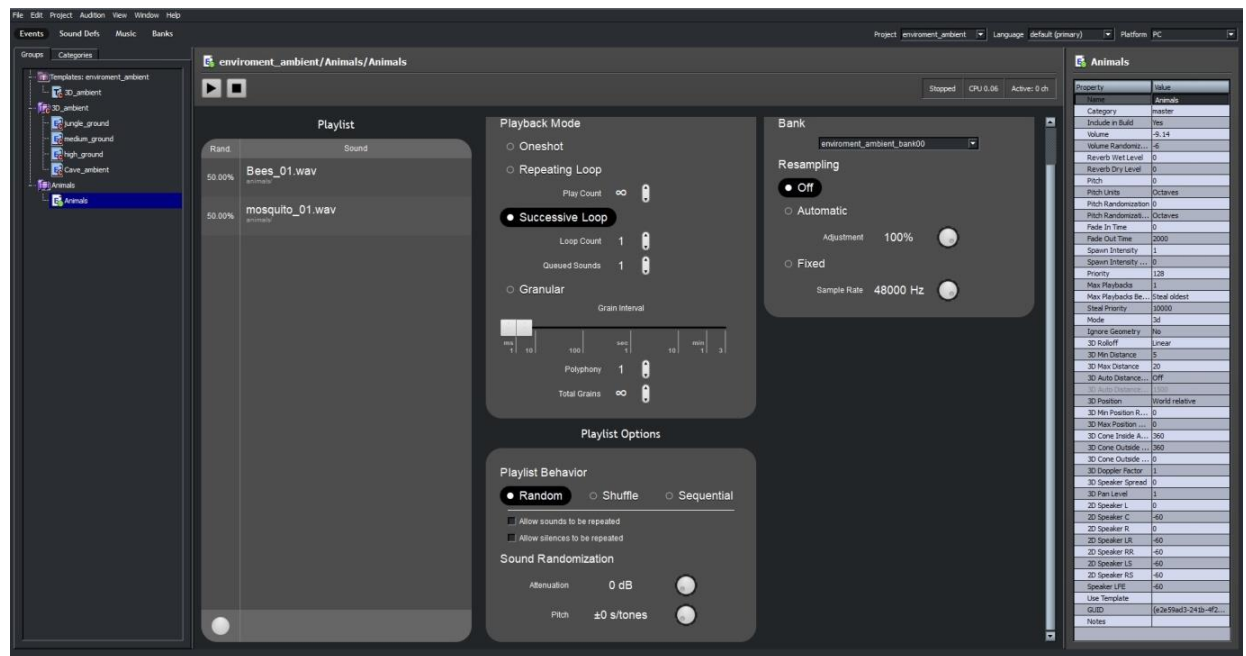
Templates: Πρόκειται για την επιλογή που μας δίνει τη δυνατότητα να σώζουμε κάποιες παραμέτρους και ρυθμίσεις, έτσι ώστε να μπορούμε εύκολα να τις εφαρμόσουμε στο μέλλον σε κάποιο παρόμοιο ηχητικό event. Στη συγκεκριμένη εργασία τα Templates μας βοηθούν ώστε να χρησιμοποιήσουμε τα ίδια settings σε όλους τους ambient ήχους του περιβάλλοντος, κάτι που έχει να κάνει τόσο με εξοικονόμηση χρόνου, αλλά βοηθάει επίσης σημαντικά σε περίπτωση που θέλουμε να υπάρχει ομοιομορφία στο ηχητικό αποτέλεσμα.

Event Groups: Ένα project θα περιέχει πολλαπλά Event Groups μέσα στο οποίο θα τοποθετήσουμε τα ηχητικά events του παιχνιδιού. Τα events είναι ανεξάρτητα μεταξύ τους και γι'αυτό μπορούμε στο ίδιο project να έχουμε ένα event group με τους ambient ήχους και άλλο event group με ήχους εκρήξεων. Μέσα σε αυτά τα group θα τοποθετήσουμε στη συνέχεια τα Multitrack και Simpletrack events.



Εικόνα 3.2: Παράδειγμα της διάταξης του interface του FMOD, για τα Event Groups και των Multi και Simple Track Events

Simpletrack Event: Η συγκεκριμένη κατηγορία είναι και η πιο σημαντική που θα κλιθεί να χρησιμοποιήσει ο sound designer στο FMOD, καθώς κάθε ένα από αυτά τα δύο events έχει τη δική του χρησιμότητα. Αρχικά, το Simpletrack event μας δίνει τη δυνατότητα να τοποθετήσουμε πολλαπλά αρχεία ήχου και να ορίσουμε να έχουν κάποια συμπεριφορά μεταξύ τους. Αυτή μπορεί να είναι από το να ακουστούν μια φορά σε σειρά, μέχρι να γίνεται άπειρο randomization της σειράς ακρόασής τους, ενώ μπορούμε να κάνουμε το αποτέλεσμα ακόμη πιο «τυχαίο», προσθέτοντας pitch και volume randomization παράγοντες. Μπορούμε επίσης να χρησιμοποιήσουμε Granual randomization, δηλαδή οι ήχοι να παίζουν τυχαία μέσα σε ένα χρονικό «παράθυρο». Τέτοιες ρυθμίσεις χρησιμοποιούνται συχνά σε περιπτώσεις όπως οι ήχοι των πουλιών σε ένα περιβάλλον, καθώς το πλεονέκτημα είναι, ότι με ελάχιστο χώρο, μπορούμε να έχουμε ένα πραγματικά ρεαλιστικό αποτέλεσμα, το οποίο μάλιστα γίνεται δημιουργείται σε πραγματικό χρόνο.

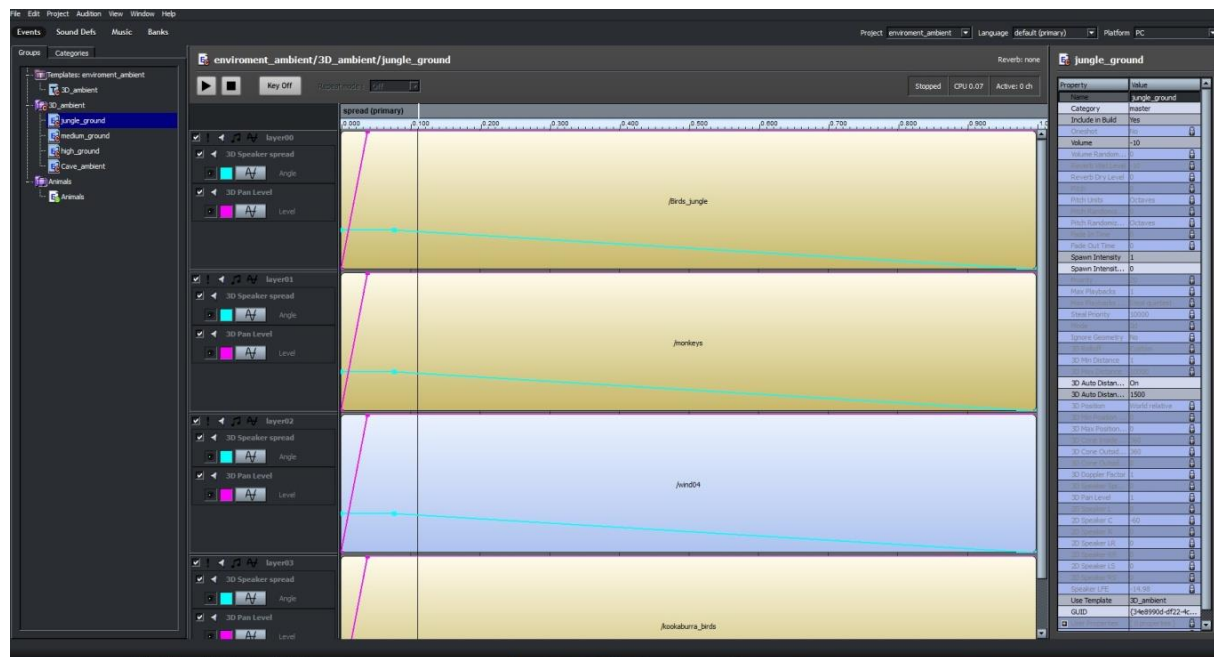


Εικόνα 3.3: Το interface ενός SimpleTrack Event Editor στο FMOD Designer 2010

Multitrack Events: Τα Multitrack Events μας αφήνουν να δημιουργήσουμε ακόμα πιο σύνθετα αποτελέσματα σε σύγκριση με τα Simpletrack Events. Ουσιαστικά ένα Multitrack Event είναι ένα σύνολο από Simpletrack Events τα οποία συνηχούν σε διαφορετικά layers. Μια σημαντική διαφορά σε σχέση με τα Simpletrack events είναι οι αυτοματισμοί και τα διάφορα εφέ που μπορούν να προστεθούν σε ένα Multitrack Event, επιπλέον με όλες τις δυνατότητες ενός Simpletrack Event που είδαμε προηγουμένως. Προκειμένου να δημιουργήσουμε ένα Multitrack Event πρέπει πρώτα να δημιουργήσουμε ένα «Sound Def», από την αντίστοιχη επιλογή επάνω αριστερά. Στη συγκεκριμένη επιλογή θα δημιουργήσουμε διαφορετικές κατηγορίες με Simpletrack events, τα οποία θα περιέχουν πολλαπλούς ήχους και τα οποία θα χρησιμοποιηθούν στην πορεία ως «πακέτα» στα layers του Multitrack Editor. Στη συγκεκριμένη εργασία παρατηρούμε ότι έχουμε χρησιμοποιήσει

ξεχωριστό Multitrack Event για κάθε είδος ζώου (όπως πουλιά, πίθηκοι, κ.α.), ενώ ακόμα και ο ήχος του αέρα αποτελείται από το δικό του ξεχωριστό Multitrack Event.

Με αυτό καταφέρνουμε να ελέγξουμε ξεχωριστά το κάθε ηχητικό αποτέλεσμα, ανάλογα με το τι θέλουμε να καταφέρουμε στο ηχητικό περιβάλλον. Για παράδειγμα, επιθυμούμε οι ήχοι των πουλιών να ακούγονται πιο συχνά με randomization τόσο στο pitch όσο και στο volume, ενώ αντίστοιχα επιθυμούμε οι ήχοι των πιθήκων να ακούγονται σε ένα πιο σπάνιο χρονικό παράθυρο και σε διαφορετική ένταση. Παρακάτω (**Εικόνα 3.4**) μπορούμε να παρατηρήσουμε καλύτερα το περιβάλλον εργασίας του Multitrack Event Editor.



Εικόνα 3.4: Το interface ενός MultiTrack Event Editor στο FMOD

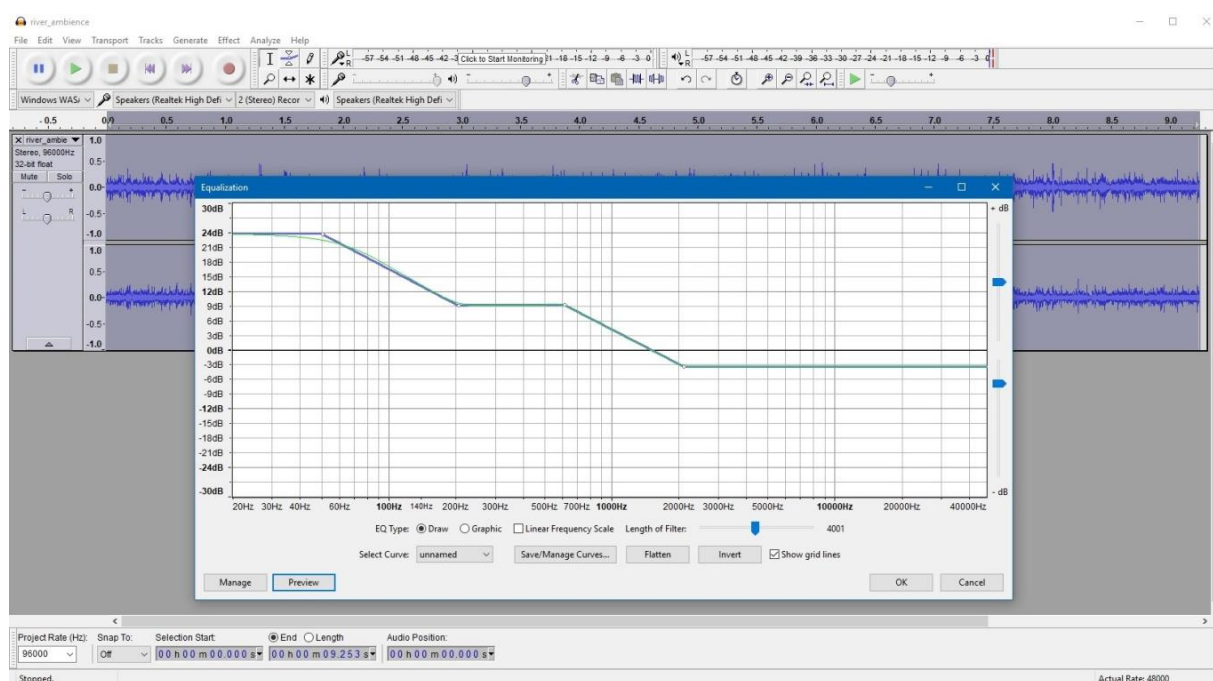
Δημιουργία ήχων καταρράκτη

Όπως αναφέραμε και προηγουμένως, πρέπει να υπάρχει συνεχώς προσπάθεια τα αρχεία ήχου σε ένα videogame να είναι όσο το δυνατόν μικρότερα σε μέγεθος και να χρησιμοποιούνται όσο το δυνατόν περισσότερες φορές, καθώς αυτό προσδίδει αποδοτικότητα σε όλο το project. Για τη δημιουργία ήχων νερού, όπως ενός καταρράκτη ή ενός ποταμιού, μπορούμε να χρησιμοποιήσουμε τρεις προσεγγίσεις, από την πιο «ελαφριά» αλλά μη ρεαλιστική, μέχρι την πιο πειστική αλλά αυτή με το μεγαλύτερο μέγεθος αρχείων.

Η πιο «ελαφριά» θα ήταν να μην χρησιμοποιήσουμε καθόλου αρχείο ήχου και να επιλέξουμε από το FMOD τη χρήση ενός white noise σήματος, το οποίο, μέσα από διάφορα εφέ όπως το pitch, μπορούμε να το κάνουμε να θυμίζει τον ήχο τρεχούμενου νερού. Μια τέτοια προσέγγιση θα ήταν ιδανική για κάποιο project που χρειάζεται να κάνει εξαρχής οικονομία στο μέγεθος των assets, όπως κάποιο παιχνίδι για φορητές συσκευές ή

android. Η δεύτερη προσέγγιση θα ήταν να χρησιμοποιήσουμε μόνο ένα αρχείο ήχου και, πάλι μέσα από το pitch effect, να πετύχουμε τρία διαφορετικά αποτελέσματα για τον κάθε καταρράκτη. Τέλος, η πιο ρεαλιστική μέθοδος αλλά αυτή με το μεγαλύτερο μέγεθος αρχείου ήχου είναι να χρησιμοποιήσουμε διαφορετικό αρχείο ήχου για κάθε ηχητική πηγή (καταρράκτη).

Αν κάποιος έχει έναν μόνο ηχογραφημένο ήχο και θέλει να δημιουργήσει περισσότερα αρχεία χωρίς τη χρήση του FMOD, θα μπορούσε να χρησιμοποιήσει κάποιο equalizer για να αλλάξει το ύψος τους. Κάτι τέτοιο θα μπορούσε να γίνει με προγράμματα όπως το Audacity και, με τον τρόπο που φαίνεται στην παρακάτω φωτογραφία (**Εικόνα 3.5**), μπορούμε να δημιουργήσουμε ένα low-pass φίλτρο για έναν ήχο, το οποίο θα μπορούσε στη συνέχεια να χρησιμοποιηθεί στο FMOD ως «όγκος» για το background κάποιου καταρράκτη, ταυτόχρονα με ένα διαφορετικό αρχείο ήχου με πιο «λαμπρό» ηχόχρωμα.

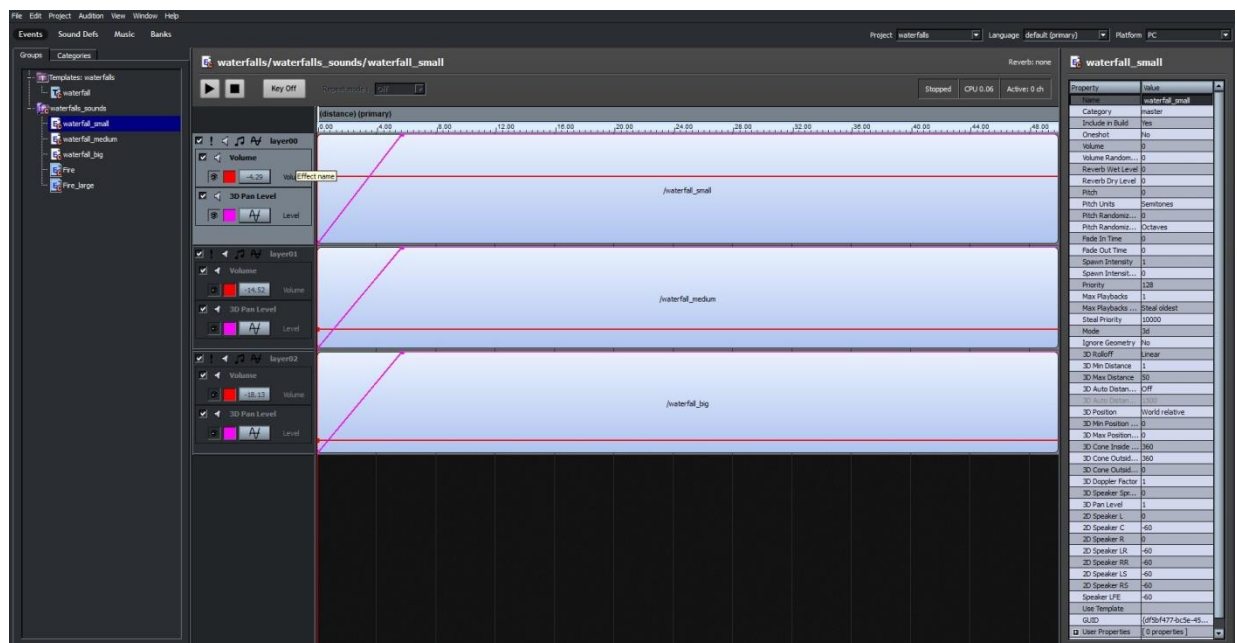


Εικόνα 3.5: Παράδειγμα ενός Low-Pass φίλτρου, με τη βοήθεια του προγράμματος Audacity

Σε αυτή την εργασία επιλέξαμε για τους τρεις καταρράκτες την τρίτη μέθοδο, καθώς χρησιμοποιήσαμε τρία διαφορετικά αρχεία ήχου για το ηχητικό αποτέλεσμα των τριών καταρράκτων. Ωστόσο αυτό δεν σημαίνει ότι κάθε καταρράκτης έχει από ένα μόνο αρχείο ήχου το οποίο παίζει απλά σε επανάληψη. Προκειμένου να δώσουμε περισσότερη λεπτομέρεια και πειστικότητα στον ήχο του καθενός, χρησιμοποιήσαμε για κάθε έναν καταρράκτη και τα τρία αρχεία ήχου, καθώς έτσι και αλλιώς το τελικό μέγεθος του build φακέλου δεν επηρεάζεται από το πόσες φορές θα χρησιμοποιηθεί το ίδιο υλικό μέσα σε ένα project. Πράγματι, το τελικό αρχείο του build που χρησιμοποιείται μόνο για τους καταρράκτες, φτάνει τα 6,60MB, κάτι που είναι αρκετά κοντά στο προστιθέμενο μέγεθος των τριών .WAV αρχείων.

Τους τρεις καταρράκτες τους ομαδοποιήσαμε σε ένα Event Group με το όνομα «Waterfall_sounds» με τον κάθε ένα να χαρακτηρίζεται ανάλογα με το μέγεθός του (Waterfall_small, Waterfall_medium, Waterfall_big). Για κάθε ένα καταρράκτη δημιουργήσαμε τρία layers, στα οποία τοποθετήθηκαν τα αντίστοιχα αρχεία ήχου, τα οποία για ευκολία έχουν το ίδιο όνομα με το όνομα των καταρρακτών: Waterfall_small, Waterfall_medium, Waterfall_big.

Πολύ σημαντική λειτουργία για έναν ήχο όπως αυτός ενός καταρράκτη είναι, το να πετύχουμε την ελάττωση της έντασης ανάλογα με την απόσταση στην οποία βρίσκεται κάθε φορά ο ακροατής/ παίκτης. Υπάρχουν δύο τρόποι να καθορίσουμε την ένταση της απόστασης του ήχου. Ο ένας είναι να ορίσουμε εμείς οι ίδιοι τον αυτοματισμό της ελάττωσης να γίνεται μέσω μιας φθίνουσας γραμμής στο Volume Effect του κάθε Layer, και ο άλλος τρόπος είναι να ορίσουμε στο FMOD την απόσταση στα properties δεξιά και στα 3D settings. Και οι δύο τεχνικές απαιτούν την προϋπόθεση να έχουμε φτιάξει και την αντίστοιχη παράμετρο σε μέτρα στο FMOD, μια βασική λειτουργία που θα εξηγηθεί παρακάτω.



Εικόνα 3.6: Τοποθέτηση των Volume Effect στα layers των καταρρακτών

Στο συγκεκριμένο παράδειγμα έχουμε χρησιμοποιήσει τη δεύτερη μέθοδο. Έτσι λοιπόν πετώντας δεξιά κλικ πάνω από το πρώτο layer μπορούμε να χρησιμοποιήσουμε τη δυνατότητα του FMOD με τις παραμέτρους –μια άκρως βασική λειτουργία για το sound design πολλών project. Ενώ είναι συνηθισμένο σε προγράμματα ήχου η γραμμή μέτρησης να αποτελεί πάντοτε το χρόνο, σε προγράμματα όπως το FMOD η γραμμή αυτή μπορεί να παραμετροποιηθεί και να ορισθεί ελεύθερα από τον sound designer με οποιαδήποτε μονάδα μέτρησης απαιτεί το project. Έτσι λοιπόν, η μονάδα μέτρησης μπορεί να περιέχει

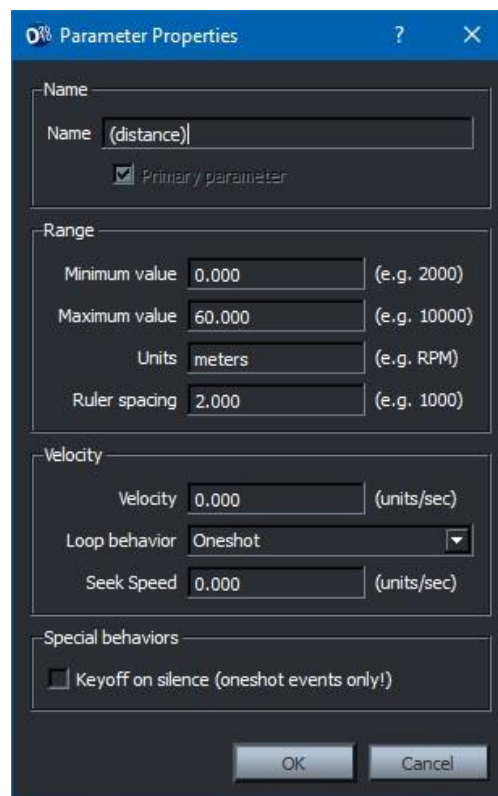
από μέτρα και απόσταση, μέχρι ταχύτητα, ύψος ή ακόμα και RPM μηχανής, σε περίπτωση που θέλουμε να δημιουργήσουμε ήχους αυτοκινήτων.

Στο συγκεκριμένο παράδειγμα ορίσαμε ως μονάδα μέτρησης τα μέτρα. Για τους καταρράκτες «small» και «medium» έχουμε ορίσει την απόσταση των 50 και 60 μέτρων αντίστοιχα ως μέγιστη, ενώ για τον μεγαλύτερο καταρράκτη ορίσαμε το ίδιο νούμερο στα 100m. Αυτή η παράμετρος θα ενσωματωθεί αργότερα στο τελικό build και θα αναγνωριστεί από τη Cry Engine. Επιπλέον, δεξιά στις παραμέτρους πρέπει να ενεργοποιήσουμε τις επιλογές για να ακούγεται ο ήχος σε 3D περιβάλλον (2D ήχοι είναι συνήθως κάποιο μουσικό θέμα), να αλλάξουμε το Max 3D Distance με το ίδιο νούμερο που έχουμε ορίσει για τον κάθε καταρράκτη και, επιπλέον, στην επιλογή «Rolloff» να επιλέξουμε το «Linear».

Κάθε καταρράκτης αποτελείται από 3 ήχους. Προκειμένου να πετύχουμε το αποτέλεσμα που θέλουμε για τον κάθε ένα, χρησιμοποιήσαμε το Volume Effect και ορίσαμε κάθε φορά ποιο αρχείου ήχου θα ακούγεται πιο δυνατά στον κάθε ένα. Επιπλέον, δεξιά στις παραμέτρους αλλάξαμε το Pitch για το κάθε multitrack event, ξεκινώντας με 0 ημιτόνια (Pitch) για τον μικρό καταρράκτη, μετά με -3 ημιτόνια για τον μεσαίο και -5 ημιτόνια για τον μεγάλο. Η επιλογή αυτή, σε αντίθεση με το Volume effect που τοποθετείται σε κάθε layer ξεχωριστά, επηρεάζει και τα 3 layers του κάθε καταρράκτη, επομένως είναι καλό να μην ξεπεράσουμε διαφοροποιήσεις άνω των 6dB, καθώς μετά οι ήχοι αλλοιώνονται σημαντικά. Τέλος, κάνοντας Build το project, δημιουργήσαμε ένα αρχείο το οποίο μπορεί να διαβαστεί από τη Cry Engine και να τοποθετηθεί μέσα στο 3D περιβάλλον με τη χρήση των «Sound Event Spot» (βλ. Κεφάλαιο 4).

Δημιουργία ήχων εστιών φωτιάς

Όπως είπαμε και σε προηγούμενη ενότητα, σε διάφορα σημεία του χάρτη έχουμε τοποθετήσει ορισμένα εφέ φωτιάς, τα οποία θα συνοδεύουν και από τους αντίστοιχους ήχους. Οι ήχοι αυτοί θα λειτουργούν ανεξάρτητα από τα υπόλοιπα ηχοτοπία και θα αποτελούν sound spots που θα είναι συνεχώς ενεργοποιημένα, εκπέμποντας ένα ηχητικό αποτέλεσμα, από συγκεκριμένη ηχητική πηγή. Το πρώτο σημείο είναι η κατεστραμμένη βάρκα στην αρχή του χάρτη, στην οποία τοποθετήσαμε 3 τέτοια εφέ, ενώ το επόμενο είναι δύο διαφορετικές εστίες φωτιάς στο εσωτερικό της σπηλιάς.

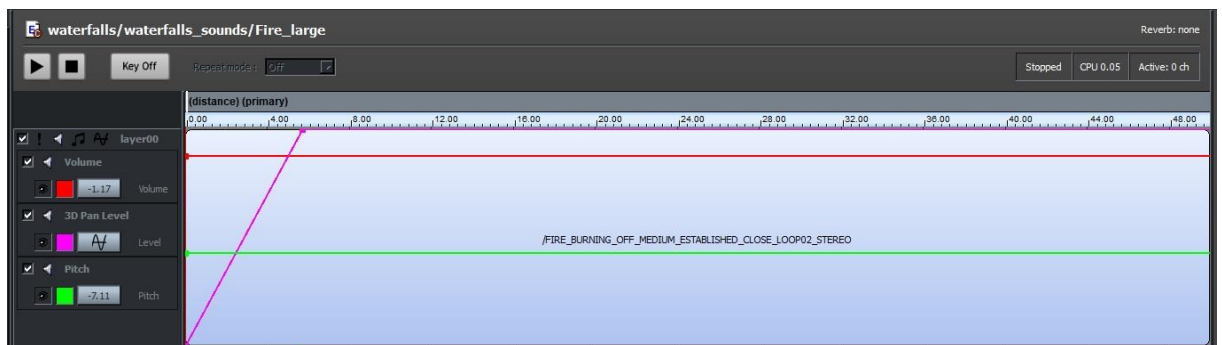


Εικόνα 3.7: Οι επιλογές για τον ορισμό των παραμέτρων, όπως αυτής της απόστασης



Εικόνα 3.8: Τοποθέτηση *SoundEventSpot* αντικειμένου για την πηγή εστίας φωτιάς

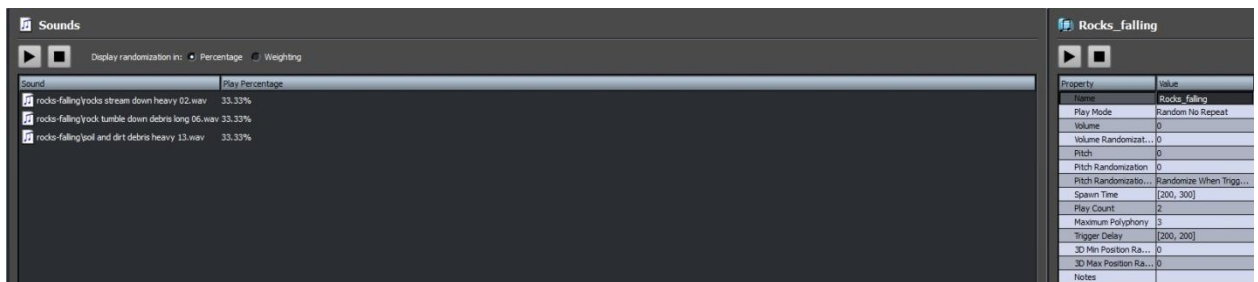
Οι ήχοι που προστέθηκαν για να ολοκληρωθεί το οπτικοακουστικό αποτέλεσμα σε αυτές τις εστίες, είναι συνολικά 4 *SoundEventSpot* Objects, τα οποία βρίσκονται στις επιλογές *Objects > Sound > SoundEventSpot*, και έχουν τοποθετηθεί πολύ κοντά στις εστίες. Το ηχητικό αρχείο που επιλέχθηκε για να χρησιμοποιηθεί στο FMOD βρέθηκε από τη βιβλιοθήκη ήχων *Foundation Library* και επεξεργάστηκε με τρεις τρόπους: Πρώτον, από το αρχικό αρχείο των 56 δευτερολέπτων χρησιμοποιήσαμε ένα απόσπασμα 14 δευτερολέπτων, το οποίο στη συνέχεια τοποθετήθηκε στο ίδιο project του FMOD στο οποίο έχουμε αποθηκεύσει τα *Sound Event* των καταρρακτών. Το προηγούμενο το πράξαμε προκειμένου να χρησιμοποιήσουμε για ευκολία τα ίδια *Templates* με αυτά των καταρρακτών. Η δεύτερη επεξεργασία που διαθέτει ο παραπάνω ήχος έχει να κάνει με το *Volume*, το οποίο μειώθηκε κατά 6 dB μέσω του *Volume Effect* του FMOD, ενώ η τρίτη επεξεργασία έχει να κάνει με το *pitch* του ήχου, αλλά μόνο για την περίπτωση της εστίας των χημικών στη σπηλιά, όπου και προσθέσαμε το παρακάτω εφέ στο FMOD (**Εικόνα 3.9**), προκειμένου να δώσουμε ένα πιο βαθύ ηχόχρωμα στο συγκεκριμένο ηχητικό αποτέλεσμα.



Εικόνα 3.9: Αλλαγή του *pitch* για τη μία εστία φωτιάς, για την απόδοση πιο χαμηλού ηχοχρώματος

Ηχητικό εφέ πτώσης βράχων

Όπως ειπώθηκε προηγουμένως στην ενότητα για τα εφέ, κατά την είσοδο του χρήστη στην σπηλιά, έχει προστεθεί το εφέ πτώσης βράχων από την οροφή. Για να συνοδέψουμε το συγκεκριμένο event θα πρέπει να δημιουργήσουμε έναν ήχο στο FMOD που να διαθέτει μια τυχαιότητα στο ηχητικό αποτέλεσμα. Για αυτόν τον σκοπό μπορούμε να χρησιμοποιήσουμε πολλαπλά αρχεία με ήχους από πτώσεις βράχων και να ορίσουμε μια συμπεριφορά. Συγκεκριμένα έχουν χρησιμοποιηθεί 3 διαφορετικά αρχεία WAV σε ένα Sound Def με την ονομασία Rocks_falling, όπως φαίνεται και στην παρακάτω **Εικόνα 3.10**, ενώ στα properties του Sound Def έχουμε ορίσει τις εξής παραμέτρους: Spawn Time = 200-300ms, Play Count = 2, Trigger Delay = 200ms. Το συγκεκριμένο sound def έχει τοποθετηθεί στη συνέχεια σε ένα Multitrack event του FMOD, με ένα μόνο layer.

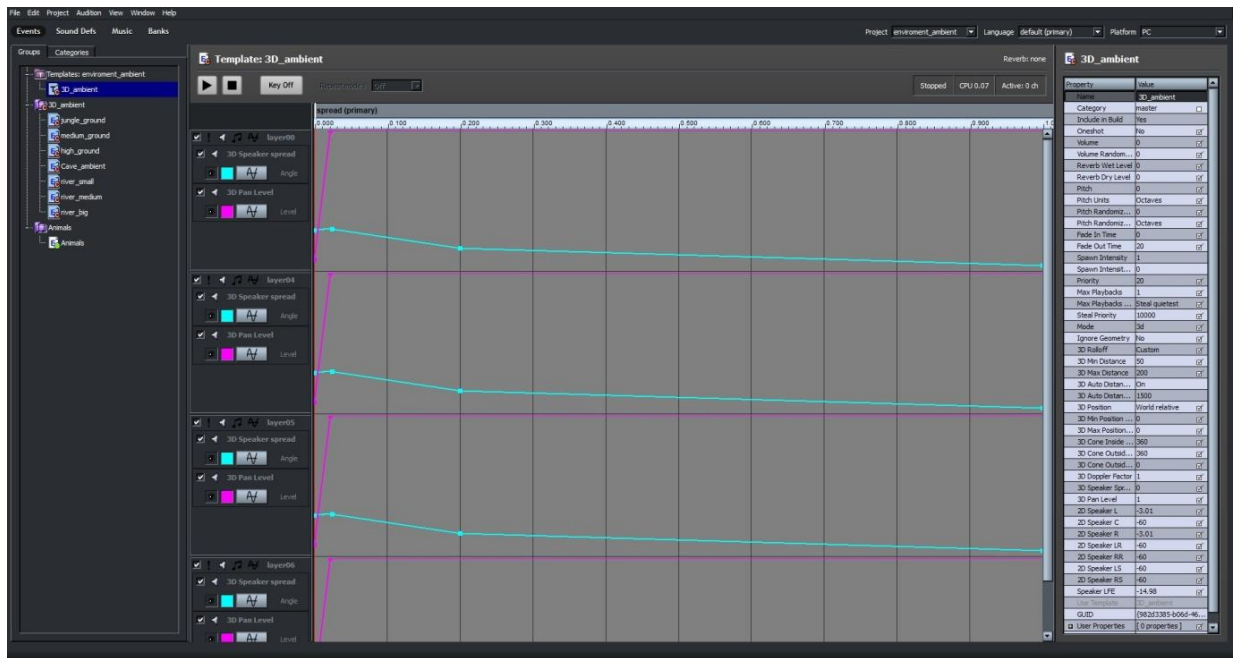


Εικόνα 3.10: Ηχητικά αρχεία και παράμετροι για το ηχητικό αποτέλεσμα πτώσης βράχων

Ambient ήχοι περιβάλλοντος (ζούγκλα, σπηλιά, παραλία, κ.α.)

Μια πολύ σημαντική κατηγορία ήχων για κάθε βιντεοπαιχνίδι είναι αυτή των ambient ήχων. Πρόκειται για περιοχές, οι οποίες μπορεί να είναι από ένα δωμάτιο μέχρι μια τεράστια έκταση, στις οποίες θέλουμε να τοποθετήσουμε ένα διαρκές ηχητικό αποτέλεσμα, του οποίου η προέλευση δεν θα είναι συγκεκριμένη. Για αυτούς τους ambient ήχους χρησιμοποιούνται τα ίδια 3D settings στο FMOD όπως και με τα υπόλοιπα sound event spots, ενώ μόνο τα μουσικά κομμάτια συνήθως αποτελούν 2D ήχους.

Στη συγκεκριμένη εργασία θα χρησιμοποιήσουμε σε αρκετά σημεία ambient ήχους και σκοπός είναι κάθε περιοχή του χάρτη να έχει και το ανάλογο άκουσμα που ταιριάζει με το οπτικό της στυλ (textures, βλάστηση κτλ). Οι ρυθμίσεις στο FMOD για όλες τις ambient περιοχές που θα εισάγουμε είναι κοινές, επομένως μπορούμε να χρησιμοποιήσουμε αρκετά αποτελεσματικά το εργαλείο των Templates. Γι αυτόν τον λόγο έχουμε δημιουργήσει ένα template με την ονομασία «3D_Ambient», το οποίο θα χρησιμοποιηθεί, με μικρές διαφοροποιήσεις, για τα παρακάτω περιβάλλοντα: της ζούγκλας, του φθινοπωρινού τοπίου, του χιονισμένου τοπίου, στη σπηλιά, στη παραλία και στα ποτάμια. Στην **Εικόνα 3.11** μπορούμε να δούμε τις ρυθμίσεις που χρησιμοποιήσαμε για το ambient template:

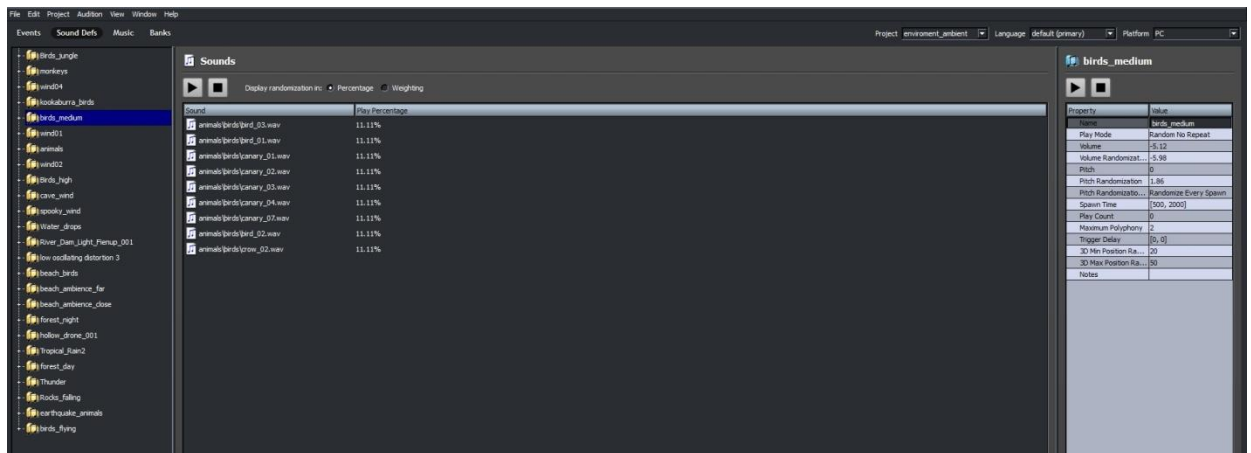


Εικόνα 3.11: Ορισμός των Templates για τα 3D Ambiences

Sound Defs: Ένα σημαντικό εργαλείο του FMOD

Πριν προχωρήσουμε στη δημιουργία των ήχων του περιβάλλοντος, μια σημαντική λειτουργία του FMOD την οποία θα χρησιμοποιήσουμε συχνά στα Multitrack Events των ambient ήχων, είναι αυτή των Sound Defs. Τα Sound Defs θα μπορούσαμε να τα εξηγήσουμε με απλό τρόπο ως Simpletrack Events που τοποθετούνται ως ένας ενιαίος ήχος, μέσα στα layer του Multitrack Editor. Αυτό σημαίνει, ότι μπορούμε να εισάγουμε στον Sound Def editor ένα σύνολο ήχων, στο οποίο θα ορίσουμε παραμέτρους, randomizations και οτιδήποτε άλλο θα μας επέτρεπε ένα Simpletrack Event και, στη συνέχεια, να χρησιμοποιήσουμε ολόκληρο αυτό το Simpletrack Event ως ένα ηχητικό αρχείο που, μαζί με άλλα, θα αποτελούν ένα μέρος ενός ηχητικού συνόλου (ή αλλιώς, ενός Layer του Multitrack Editor).

Στην παρακάτω φωτογραφία (**Εικόνα 3.12**) βλέπουμε τις ρυθμίσεις ενός Sound Def για τους ήχους των πουλιών της ορεινής περιοχής. Παρατηρούμε ότι έχουν εισαχθεί πολλαπλά αρχεία .WAV, τα οποία, όπως φαίνεται δεξιά στα properties, διαθέτουν randomization settings στην ένταση, στο pitch και στο κάθε πότε θα ακούγονται (Spawn Time). Συγκεκριμένα, θέλουμε να υπάρχει ένα σχετικά μέτρια-πυκνό ηχητικό αποτέλεσμα, οπότε επιλέξαμε οι ήχοι των πουλιών να ενεργοποιούνται μέσα σε ένα μικρό ηχητικό «παράθυρο» των περίπου 500-2000ms, ενώ επιπλέον επιλέξαμε να μπορούν να ακουστούν έως και 2 ήχοι πουλιών ταυτόχρονα, σε περίπτωση που ενεργοποιηθούν τυχαία 2 ήχοι μαζί (Maximum polyphony).

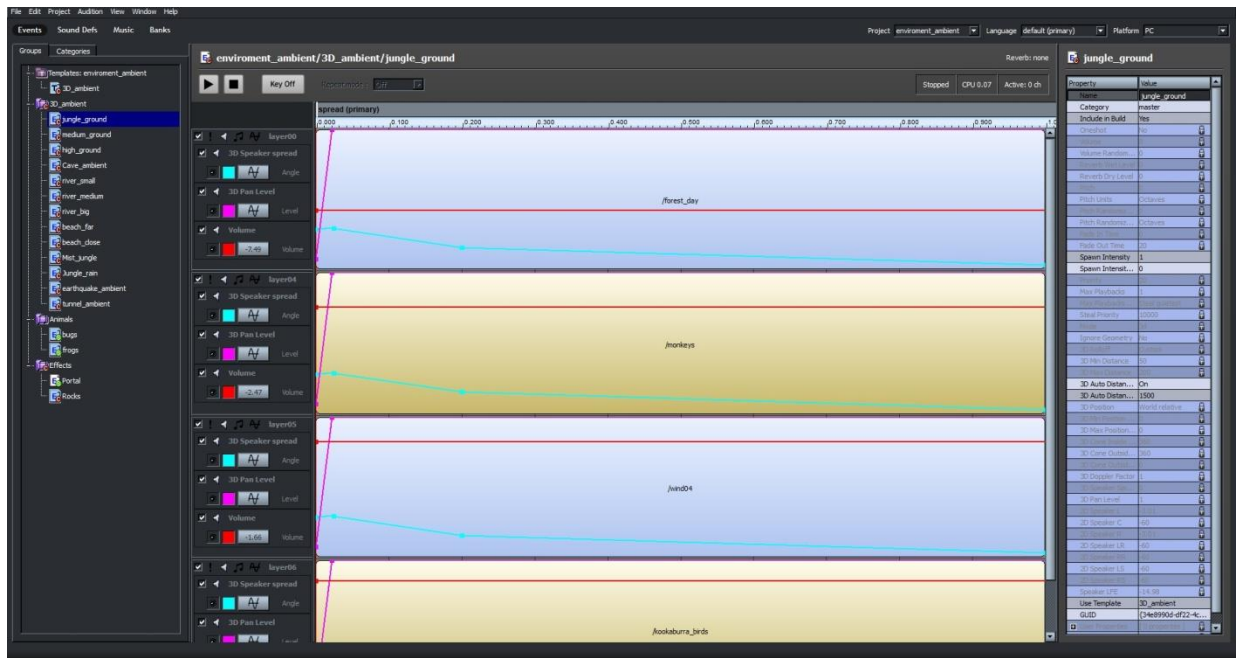


Εικόνα 3.12: Το interface των Sound Defs στο FMOD και οι ρυθμίσεις για το παράδειγμα των πουλιών της ορεινής περιοχής

Ambient ήχοι περιβάλλοντος (συνέχεια...)

Έχοντας δημιουργήσει όλα τα Sound Defs με τα ηχητικά ακούσματα μιας ζούγκλας, μπορούμε πλέον να τα τοποθετήσουμε στον Multitrack Event Editor προκειμένου να ορίσουμε συνολικά την συμπεριφορά τους. Για τους ήχους της ζούγκλας θα χρησιμοποιήσουμε συνολικά 4 layers, τρία από τα οποία θα χρησιμοποιηθούν για διαφορετικούς ήχους ζώων και ένα για τον ήχο του ανέμου. Τα ηχητικά αρχεία, ειδικά στους ήχους των ζώων, δεν χρειάζεται να είναι μεγάλα σε διάρκεια καθώς, λόγω του randomization των Sound Defs, ο χρήστης δεν θα καταλαβαίνει εύκολα ότι πρόκειται για λιγοστά ηχητικά assets τα οποία επαναλαμβάνονται.

Το μεγαλύτερο WAV αρχείο σε αυτό το layer είναι το Forest_day και πρόκειται για 39 δευτερόλεπτα ηχητικού ambience τροπικού δάσους, το οποίο χρησιμοποιήσαμε ως «βάση» για τα σημεία που το περιβάλλον της ζούγκλας θα βρίσκεται σε ημερήσια, φυσιολογική κατάσταση. Για τον άνεμο, το αρχείο έχει διάρκεια 22 δευτερολέπτων, ενώ για τους υπόλοιπους ήχους των ζώων τα αρχεία είναι πολύ μικρότερα, όπως του ενός ή δύο δευτερολέπτων, καθώς πρόκειται για μεμονωμένα τιτιβίσματα ή κραυγές ζώων. Σε πολλές περιπτώσεις που δεν μπορούσαμε να βρούμε πολλαπλά αρχεία με διαφορετικά ζώα, προτιμήσαμε να πάρουμε μια συνεχόμενη φράση ενός ζώου και να απομονώσουμε κάθε «λέξη» του, κόβοντάς την με το Audacity, δημιουργώντας έτσι πολλαπλά .WAV αρχεία, ώστε να δουλέψει καλύτερα το randomization των Sound Defs.



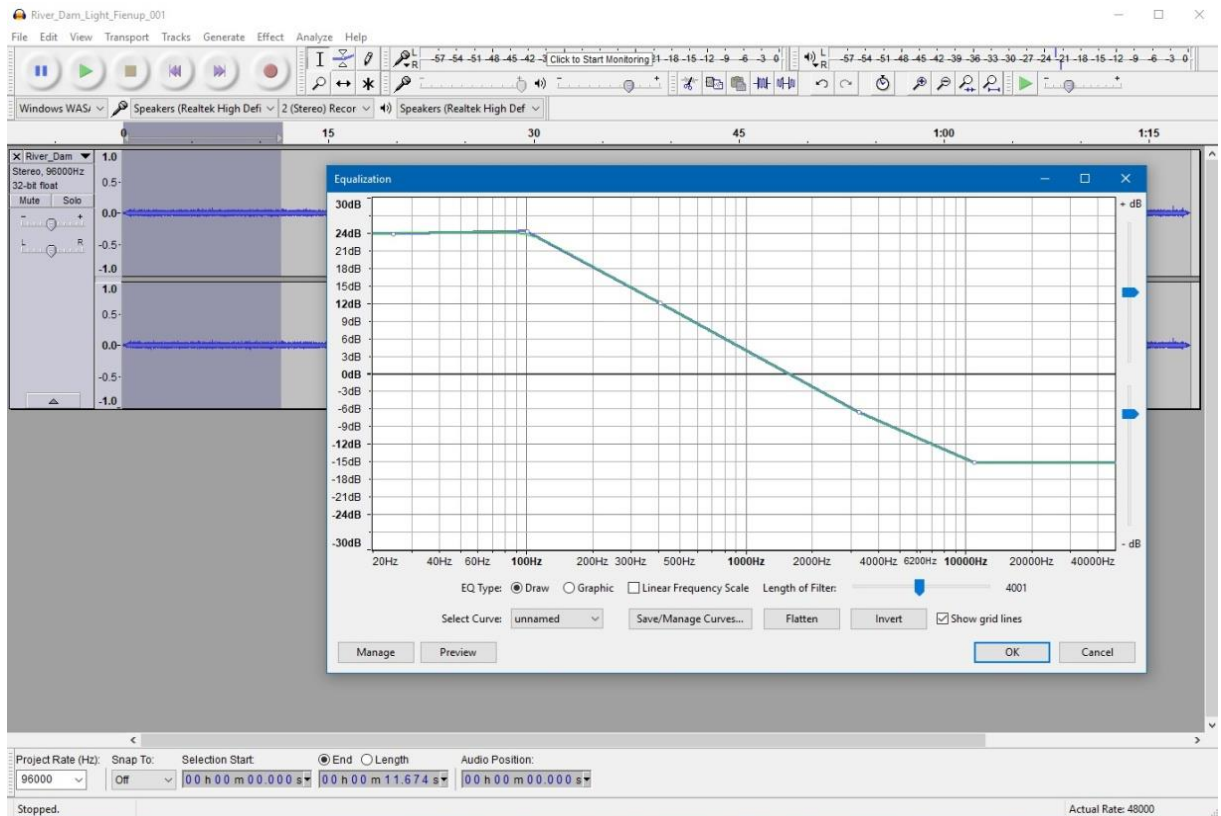
Εικόνα 3.13: Συνολικά τα 4 layers για το 3D Ambient περιβάλλον της ζούγκλας

Στην προηγούμενη φωτογραφία (**Εικόνα 3.13**) παρατηρούμε το πώς δείχνουν τα 4 layers με το συνολικό ηχητικό αποτέλεσμα για το περιβάλλον της ζούγκλας. Τέλος, είναι απαραίτητο δεξιά στα properties να είναι ενεργοποιημένο το «Mode: 3D».

Δημιουργία ήχων για τα ποτάμια

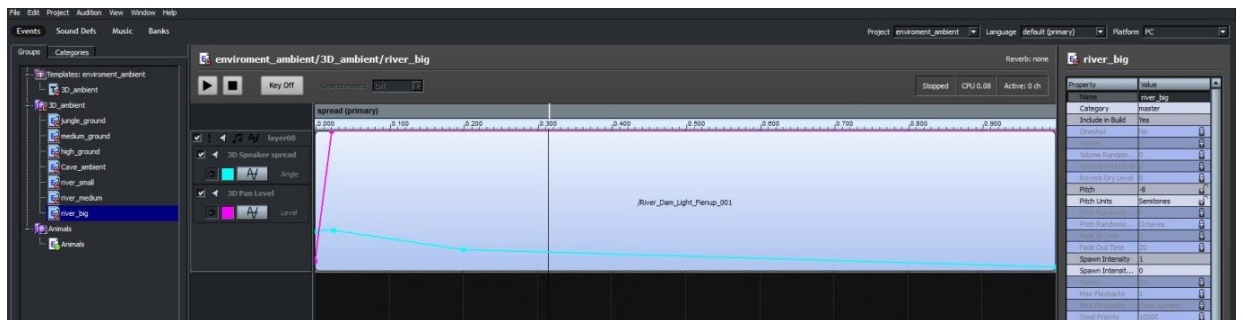
Με αντίστοιχο τρόπο θα εργαστούμε και για τον ήχο του ποταμιού, με τη μόνη διαφορά ότι αυτός θα αποτελέσει ambient ήχο και όχι ηχητικό γεγονός που θα έχει συγκεκριμένο σημείο στο περιβάλλον ως πηγή, όπως στη περίπτωση με τους καταρράκτες. Η διαφορά ανάμεσα στις δύο περιπτώσεις είναι, ότι στους ambient ήχους ορίζουμε ένα «κουτί» ή μια περιοχή (το ποτάμι) μέσα στην οποία ακούγεται το συγκεκριμένο ηχητικό αποτέλεσμα, ενώ αντίθετα όταν ο ήχος παράγεται από ένα συγκεκριμένο σημείο (καταρράκτης), επηρεάζεται κυρίως από την απόσταση.

Για τον ήχο του ποταμιού χρησιμοποιήσαμε ένα ηχογραφημένο ambience από τη δωρεάν βιβλιοθήκη ήχων Soniss.com, τον οποίο επεξεργαστήκαμε πρώτα στο Audacity με τους εξής τρόπους. Επειδή το αρχείο ήταν πάνω από 1 λεπτό, χρησιμοποιήσαμε μόνο τα πρώτα 10 δευτερόλεπτα και επιπλέον σχεδιάσαμε και χρησιμοποιήσαμε το παρακάτω equalizer (**Εικόνα 3.14**), προκειμένου να δώσουμε πιο βαθιά υπόσταση στο άκουσμα του νερού.



Εικόνα 3.14: Δημιουργία ενός Low-Pass φίλτρου για ένα ηχητικό αρχείο ενός ποταμού

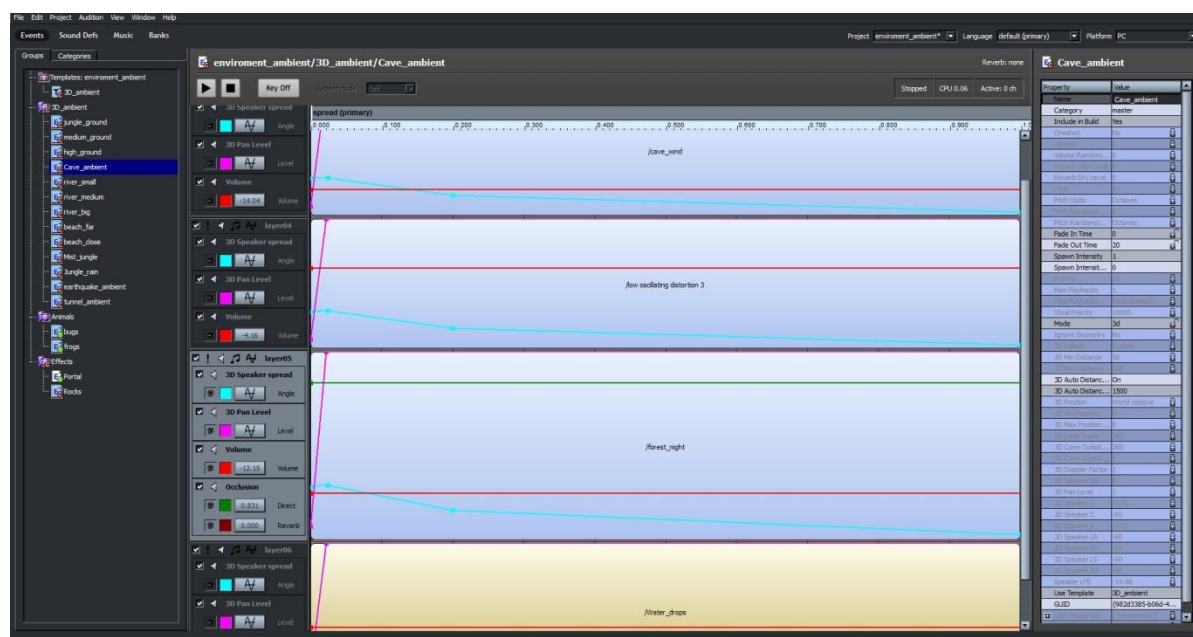
Στη συνέχεια εξάγαμε το συγκεκριμένο απόσπασμα ως αρχείο .WAV και σε πολύ υψηλό bit rate και το τοποθετήσαμε στη βιβλιοθήκη με τους υπόλοιπους ήχους του FMOD. Στη συνέχεια στο FMOD χρησιμοποίησαμε τα ίδια templates με τα οποία φτιάξαμε και τους υπόλοιπους ήχους περιβάλλοντος. Για περισσότερο ρεαλισμό στο τελικό αποτέλεσμα χρησιμοποίησαμε την ίδια τεχνική με αυτή που αναφέραμε προηγουμένως με τους τρεις καταρράκτες. Έτσι λοιπόν δημιουργήσαμε τρία διαφορετικά Multitrack events, ένα για κάθε ποτάμι, τα οποία διαφοροποιούνται ανάλογα με το pitch: -3 ημιτόνια για το μικρό, -5 ημιτόνια για το μεσαίο και -8 ημιτόνια για το πιο βαθύ και μεγάλο ποτάμι. Στην παρακάτω φωτογραφία (**Εικόνα 3.15**) φαίνεται η δημιουργία του multitrack event για το ποτάμι.



Εικόνα 3.15: Η επιλογή των ρυθμίσεων για ένα από τα 3 ποτάμια του χάρτη

Δημιουργία του ηχοτοπίου της σπηλιάς

Για τη δημιουργία του ηχητικού περιβάλλοντος της σπηλιάς δεν θα χρειαστούμε πλούσιο ηχητικό αποτέλεσμα, ωστόσο μπορούμε να χρησιμοποιήσουμε διάφορες τεχνικές ώστε να μεταφέρουμε στον παίκτη την αίσθηση του όγκου της σπηλιάς. Όπως είδαμε επίσης σε προηγούμενη ενότητα, το μέρος της σπηλιάς περιλαμβάνει και ένα ηχητικό event, όπως αυτό της πτώσης των βράχων από την οροφή, κάτι που όμως αποτελεί one-shot ήχο και δεν περιλαμβάνεται στο συνολικό ambient του τοπίου της σπηλιάς. Υπενθυμίζεται επίσης ότι το περιβάλλον της σπηλιάς διαθέτει μέσα στην CryEngine ένα ολικό Reverb εφέ, το οποίο επηρεάζει συνολικά όλους τους ήχους σε αυτή την περιοχή.



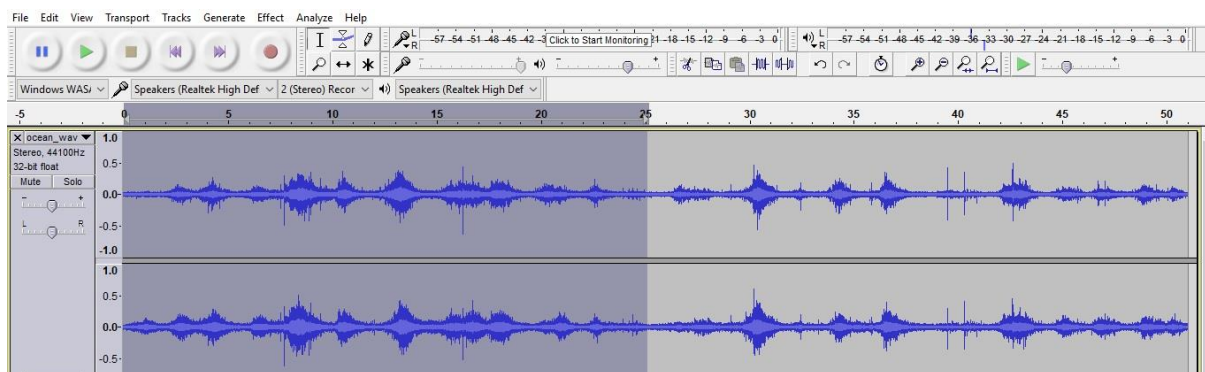
Εικόνα 3.16: Δημιουργία των layers και ρυθμίσεις για το ηχητικό περιβάλλον της σπηλιάς

Στη σπηλιά χρησιμοποιήσαμε συνολικά 4 layers. Τα δύο πρώτα αποτελούν το ηχητικό background του χώρου, με μια χαμηλή βοή ανέμου και ένα ακόμα Layer με μια χαμηλή συχνότητα για να καλύπτει το κενό. Στο τρίτο layer έχουμε χρησιμοποιήσει το ambience από τη νυχτερινή περιοχή της ζούγκλας, το οποίο θέλουμε να ακούγεται παραμορφωμένο, σαν να ακούγεται έξω από τη σπηλιά. Για να το πετύχουμε αυτό έχουμε χρησιμοποιήσει το εφέ Occlusion από το FMOD σε συνδυασμό με το εφέ έντασης. Στο τελευταίο Layer έχουμε χρησιμοποιήσει σε ένα Sound Def πέντε ήχους από σταγόνες νερού, οι οποίες διαθέτουν randomization ρυθμίσεις για το pitch, το volume και το spawn time. Όπως και με τους υπόλοιπους ήχους από τις άλλες περιοχές του χάρτη, χρησιμοποιήσαμε και στη σπηλιά τις ρυθμίσεις του template που είχαμε δημιουργήσει προηγουμένως. Το volume εφέ το χρησιμοποιήσαμε και για τα υπόλοιπα layers, ανάλογα με το τι θέλουμε να ξεχωρίζει στο τοπίο της σπηλιάς, όπως φαίνεται στην **Εικόνα 3.16**.

Δημιουργία ηχοτοπίου Θάλασσας και παραλίας

Προκειμένου να έχουμε ένα φυσικό αποτέλεσμα για την περιοχή της παραλίας, χρειάζεται ο ήχος της θάλασσας να ακούγεται πιο δυνατά όσο πλησιάζουμε προς τα κύματα και λιγότερο όσο βρισκόμαστε πιο μακριά. Για αυτόν τον λόγο θα δημιουργήσουμε δύο διαφορετικά multitrack events, ένα για την πιο απομακρυσμένη περιοχή της παραλίας και ένα για τους ήχους των κυμάτων.

Χρησιμοποιήσαμε δύο ηχογραφήσεις από ambience ωκεανού από τη δωρεάν βιβλιοθήκη Sonniss, μια για τη κάθε περίπτωση. Επειδή τα αρχεία θα παίζουν σε επανάληψη, δεν είναι απαραίτητο να είναι μεγάλα σε διάρκεια και για αυτόν τον λόγο «κόψαμε» από την κάθε ηχογράφηση ένα απόσπασμα των 25 με 30 δευτερολέπτων, δημιουργώντας έτσι, με το Audacity, δύο διαφορετικά αρχεία ήχου. Για ευκολία, τα αρχεία αυτά τα ονομάσαμε «*beach_ambience_far*» και «*beach_ambience_close*».



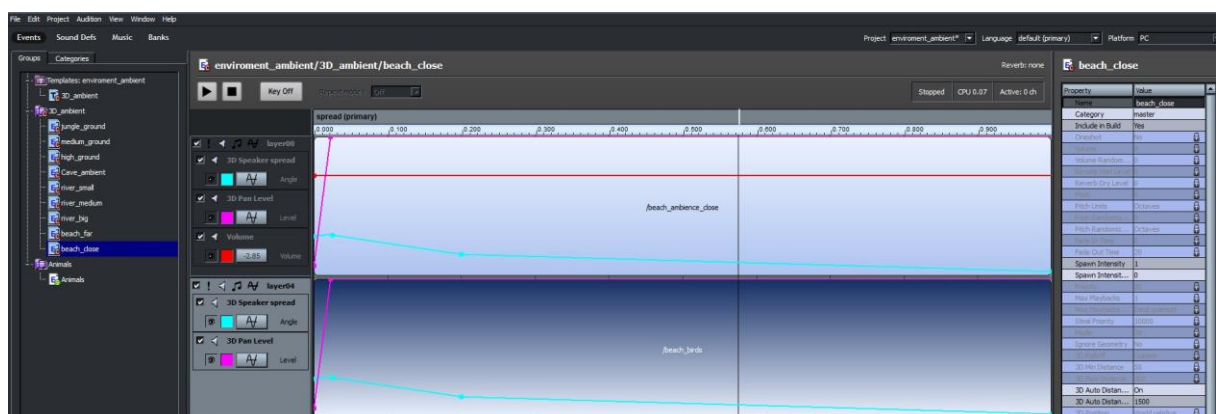
Εικόνα 3.17: Επιλογή ηχητικού αποσπάσματος 25 δευτερολέπτων για τη δημιουργία αρχείου ήχου για την περιοχή της θάλασσας

Στη συνέχεια δημιουργήσαμε δυο νέα Multitrack Events στο FMOD, με τις ονομασίες «*beach_close*» και «*beach_far*». Τα events αυτά θα έχουν δύο μόνο layers, ένα για τα δύο WAV αρχεία που δημιουργήσαμε από το Audacity και ένα δεύτερο layer με ήχους γλάρων που έχουν τοποθετηθεί προηγουμένως σε ένα Sound Def. Για τους ήχους των γλάρων θα ακολουθήσουμε την ίδια τεχνική και με τα προηγούμενα παραδείγματα. Θα χρησιμοποιήσουμε συνολικά 4 μικρές εκφράσεις γλάρων τα οποία θα εισαχθούν σε ένα Sound Def, έτσι ώστε να χρησιμοποιήσουμε ορισμένα randomizations, πριν εισαχθούν στο δεύτερο layer του Multitrack Event. Στην παρακάτω φωτογραφία (**Εικόνα 3.18**) βλέπουμε τις ρυθμίσεις που έχουμε ορίσει στο Sound Def των γλάρων (*beach_birds*).



Εικόνα 3.18: Τοποθέτηση 4 αρχείων με «φράσεις» γλάρων, για τη δημιουργία Sound Def για την περιοχή της θάλασσας

Οι γλάροι δεν είναι απαραίτητο να ακούγονται πολύ συχνά και για αυτό έχουμε ορίσει να γίνονται spawn έως και δύο φράσεις, μέσα σε ένα παράθυρο 5 έως 10 δευτερολέπτων. Επίσης ορίσαμε randomizations για pitch και volume, αλλά και επιπλέον 3D randomization για αποστάσεις ανάμεσα στα 50 και 100 μέτρα. Το τελευταίο θα βοηθήσει ιδιαίτερα ώστε οι γλάροι να ακούγονται σαν να βρίσκονται μακριά. Στην παρακάτω φωτογραφία (**Εικόνα 3.19**) βλέπουμε πως εισάγουμε το sound def και τον ήχο των κυμάτων στα δύο Layers του Multitrack Event. Οι ρυθμίσεις των εφέ παραμένουν οι ίδιες με αυτές των υπολοίπων templates, ενώ μόνο για το event «beach_close» χαμηλώσαμε λίγο την ένταση των κυμάτων με ένα volume effect.



Εικόνα 3.19: Οι ρυθμίσεις των 2 layer στον Multitrack Editor, για τη δημιουργία του ηχητικού περιβάλλοντος της θάλασσας

Τέλος, επειδή στο ηχοτοπίο της παραλίας έχουμε δύο διαφορετικά ακούσματα, αυτό σημαίνει ότι αργότερα, κατά την τοποθέτηση των ήχων, θα πρέπει να ορίσουμε δύο διαφορετικές περιοχές (shapes) στις οποίες θα ακούγεται το κάθε ένα, κάτι που θα δούμε πιο αναλυτικά στο Κεφάλαιο 4.

Δημιουργία ηχοτοπίου ομίχλης

Στο ηχοτοπίο της ομίχλης επιθυμούμε να πετύχουμε την ηχητική αποτύπωση ενός πιο ήρεμου περιβάλλοντος ζούγκλας, θυμίζοντας αρκετά νυχτερινό περιβάλλον. Το συγκεκριμένο ηχοτοπίο εκτείνεται στην ίδια έκταση με το εφέ της ομίχλης (βλ. αντίστοιχη ενότητα) και περιλαμβάνει και το σημείο στο οποίο θα ακούγεται η τροπική καταιγίδα. Στο FMOD χρησιμοποιήσαμε το 3D Template που έχουμε δημιουργήσει από πριν για όλους τους 3D Ambient ήχους και επιπλέον δημιουργήσαμε ένα νέο MultiTrack Event με την ονομασία Mist_jungle στο οποίο θα εισαχθούν οι ήχοι. Θα χρησιμοποιηθούν συνολικά 4 layers με ηχητικά events: Το πρώτο layer αποτελείται από το ηχητικό αρχείο forest_night και πρόκειται για ambient ήχους ζώων βραδινής ώρας (όπως τριζόνια). Το συγκεκριμένο Layer θα είναι πιο δυνατά από τα υπόλοιπα, καθώς επιθυμούμε να ξεχωρίζει στο ηχοτοπίο και για αυτό έχουμε θέσει ένα Volume Effect περίπου στο 80% της έντασης. Το δεύτερο layer αποτελείται από ambient μουσική (Hollow_drone_001) και έχει εισαχθεί προκειμένου να δημιουργήσει ατμόσφαιρα στον ακροατή, σε συνδυασμό με το σκοτεινό περιβάλλον που δημιουργεί το εφέ της ομίχλης.



Εικόνα 3.20: Οι ρυθμίσεις των τεσσάρων layer στον Multitrack Editor, για τη δημιουργία του ηχοτοπίου της ομίχλης

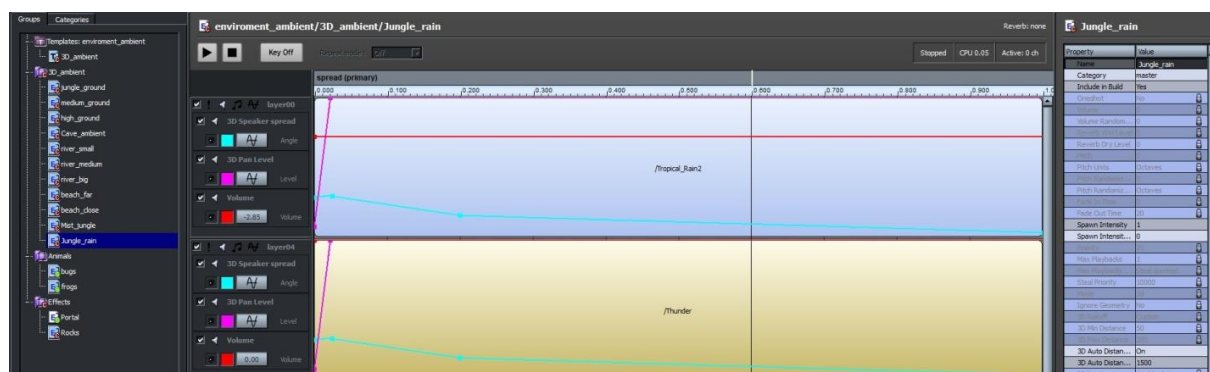
Το τρίτο layer περιλαμβάνει τον άνεμο (Cave_wind) μαζί με ένα εφέ έντασης που το μειώνει σχεδόν στο 20% της έντασης του. Το τελευταίο layer περιλαμβάνει ένα σύνολο από κραυγές μαϊμούδων, οι οποίες έχουν αποθηκευτεί νωρίτερα ως Sound Def και έχουν χρησιμοποιηθεί επίσης και στην περίπτωση του ηχοτοπίου της υπόλοιπης ζούγκλας. Η διαφορά εδώ είναι, ότι χρησιμοποιούμε το ίδιο Sound Def αλλά σε πολύ μικρότερη ένταση, περίπου στο 40%, προκειμένου να δώσουμε την εντύπωση της απόστασης. Ο σκοπός του ηχοτοπίου της νυχτερινής ομίχλης είναι να αποτελέσει τον προάγγελο ενός άλλου μεγαλύτερου event, όπως της τροπικής καταιγίδας που θα ακολουθήσει στην πορεία. Τέλος, μέσα στην Cry Engine έχουμε τοποθετήσει στη διαδρομή της ομίχλης και

ορισμένους ξεχωριστούς ήχους ζώων, ως Sound event spots, όπως τους ήχους των βατράχων που έχουμε χρησιμοποιήσει και στο παράδειγμα του υδάτινου ηχοτοπίου (βλ. αντίστοιχη ενότητα).

Δημιουργία ήχων μιας τροπικής καταιγίδας

Στο μισό περίπου μέρος από το συνολικό ηχοτοπίο της ομίχλης, έχουμε τοποθετήσει το εφέ μιας τροπικής καταιγίδας. Σκοπός εδώ είναι το προηγούμενο ηχοτοπίο της νυχτερινής ομίχλης να συνυπάρχει με την καταιγίδα και για αυτόν τον λόγο τα δύο ambiences συνηθούν. Όπως είναι φυσικό η καταιγίδα πρέπει να επικαλύπτει σε μεγάλο βαθμό τους υπόλοιπους ήχους, και για αυτό έχει ορισθεί να ακούγεται πιο δυνατά. Η καταιγίδα διαθέτει το δικό της MultiTrack Event στο FMOD με την ονομασία Jungle_rain και περιλαμβάνει 2 layers. Στο πρώτο περιλαμβάνεται το Ambience από την ηχογράφιση μια τροπικής βροχής, με την ονομασία Tropical_rain2 και στο δεύτερο έχουμε δημιουργήσει ένα Sound Def με τους ήχους κεραυνών. Το συγκεκριμένο Sound Def περιλαμβάνει τρία ηχητικά αρχεία WAV με κεραυνούς, τα οποία έχουν δημιουργηθεί από ένα μεγαλύτερο αρχείο, κόβοντας τρεις κεραυνούς συνολικά με τη χρήση του Audacity. Τα τρία αυτά WAV αρχεία διαθέτουν τις εξής ρυθμίσεις στο Sound Def του FMOD: Pitch Randomization=6, Spawn Time=8000-15000, Maximum Polyphony=1, Min-Max 3D Distance=100-500.

Παρόλο που έχουμε λίγα αρχεία για τους κεραυνούς, ορίζοντας ένα pitch randomization στα 6dB από τα properties του FMOD, καταφέρνουμε να τα κάνουμε να ακούγονται κάθε φορά αρκετά διαφορετικά. Επιπλέον, το μεγάλο χρονικό παράθυρο στο οποίο ενεργοποιούνται οι κερανοί, μειώνει αρκετά την επανάληψη κατά την περιπλάνηση του χρήστη στη συγκεκριμένη περιοχή. Στη συνέχεια το Sound Def των κεραυνών εισάγεται στο δεύτερο Layer του Multitrack Event προκειμένου να δημιουργηθεί συνολικά ο ήχος μιας τροπικής καταιγίδας, όπως φαίνεται και στην παρακάτω εικόνα.

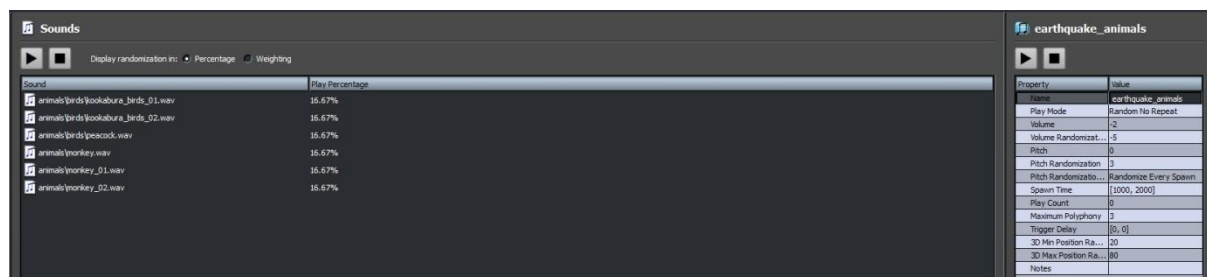


Εικόνα 3.21: Δημιουργία ήχων βροχής, σε συνδυασμό με κεραυνούς, σε 2 ξεχωριστά layers

Δημιουργία ηχοτοπίου σεισμικής δόνησης

Για το ηχοτοπίο στο οποίο θα λάβει χώρα το γεγονός του σεισμού, σκοπός είναι να δώσουμε την εντύπωση της αναστάτωσης στο ηχητικό αποτέλεσμα. Επειδή το γεγονός συμβαίνει στο επίπεδο της ζούγκλας, αυτό σημαίνει ότι οι ήχοι των πουλιών θα παραμείνουν σε μικρότερη ένταση, ενώ επιπλέον έχουμε προσθέσει κάποια επιπλέον layers στο FMOD. Συνολικά θα χρησιμοποιηθούν 5 layers στο FMOD. Το πρώτο αποτελεί τους ήχους των πουλιών της ζούγκλας (forest_day) που έχει χρησιμοποιηθεί και στα προηγούμενα παραδείγματα. Το δεύτερο layer αποτελεί ένα καινούργιο Sound Def με ήχους διαφόρων ζώων, όπως μαϊμούδων και τροπικών πουλιών. Τα properties σε αυτό το sound def έχουν ρυθμιστεί με τέτοιο τρόπο ώστε τα ζώα να ακούγονται πιο συχνά (Spawn time = 1000 με 2000ms), το ένα πάνω στο άλλο (Polyphony = 3), με επιπλέον randomizations στο Pitch (3dB) και στο Volume (-5 dB). Το επόμενο layer αποτελεί την αναπαράσταση ενός ταρακουνήματος χαμηλής συχνότητας που θέλουμε να ακούγεται σε αόριστη κατεύθυνση σε όλη την περιοχή. Για αυτόν τον λόγο μπορούμε να χρησιμοποιήσουμε από τα αρχεία των Sound Defs που έχουμε ήδη, όπως είναι οι ήχοι των βράχων που πέφτουν στην είσοδο της σπηλιάς.

Η μόνη επεξεργασία που θα χρειαστεί να προσθέσουμε από το FMOD, είναι ένα Pitch εφέ, το οποίο το ρυθμίζουμε στα -34 ημιτόνια. Αυτό κάνει αυτομάτως τον ήχο πολύ πιο βαθύ αλλά και πιο αργό, δίνοντας την εντύπωση ότι προέρχεται από μεγάλη απόσταση, από κάτι μεγάλο που κινείται.



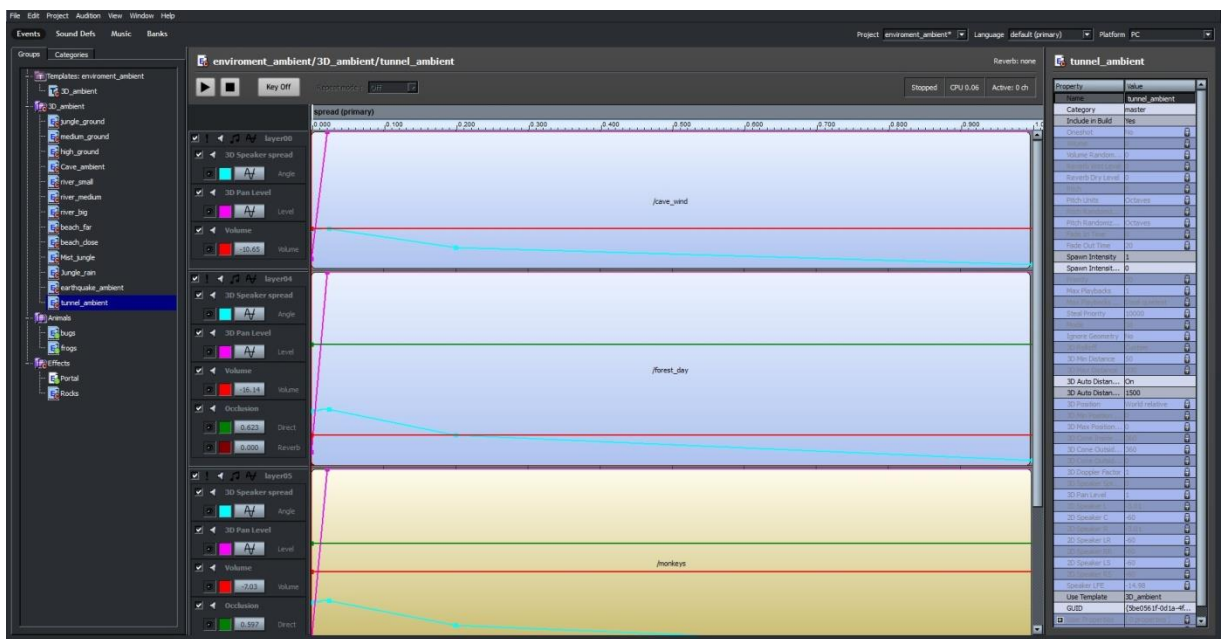
Εικόνα 3.22: Δημιουργία ήχων ζώων σε αναστάτωση, από τα ίδια αρχεία ήχου, προκειμένου να εξομοιωθεί το ηχητικό αποτέλεσμα ενός σεισμού

Τέλος, το επόμενο Layer χρησιμοποιείται από τον άνεμο (το ίδιο αρχείο ήχου με το παράδειγμα της υπόλοιπης ζούγκλας), ενώ ως τελευταίο layer έχουμε προσθέσει ήχους πουλιών που πετάνε. Για το τελευταίο χρησιμοποιήθηκε ένα αρχείο ήχου μόνο (birds_flying), το οποίο εισήχθη ως sound def, με Spawn Time τα 4 sec, Trigger Delay τα 10 sec και pitch randomization στα 3 dB.

Προσθήκη ηχητικών εφέ για κλειστό τούνελ

Ένα ακόμη ηχοτοπίο που βρίσκεται στον κεντρικό δρόμο, είναι αυτό ενός πέτρινου τούνελ. Στο συγκεκριμένο επιθυμούμε να ακούγονται οι ήχοι της ζούγκλας κατά την ημέρα, όμως με έντονο reverb εφέ, το οποίο, όπως και στη περίπτωση της σπηλιάς, θα το προσθέσουμε μέσα από την μηχανή γραφικών. Πηγαίνοντας στα *Objects > Sound > ReverbVolume*, μπορούμε να τοποθετήσουμε το συγκεκριμένο Object στο περιβάλλον και να το συνδέσουμε με το Share του περάσματος. Αυτό σημαίνει ότι, όλοι οι ήχοι που θα ακούγονται μέσα αυτή τη περιοχή, θα επηρεάζονται από το Reverb. Το συγκεκριμένο Reverb διαθέτει λίστα με διάφορα presets και έχει επιλεγθεί το *Corridor_Narrow*.

Επιπλέον, έχουμε δημιουργήσει ένα ξεχωριστό MultiTrack event στο FMOD, με 3 layers, προκειμένου να μπορέσουμε να προσθέσουμε ορισμένα εφέ στους ήδη υπάρχον ήχους της ζούγκλας. Σκοπός εδώ είναι να δώσουμε την εντύπωση ότι οι ήχοι διαπερνούν ένα υλικό, οπότε θα χρησιμοποιήσουμε το εφέ Occlusion, για τα Layers των ήχων των πουλιών και των μαϊμούδων. Επιπλέον θα χρησιμοποιήσουμε και ένα εφέ έντασης για να ορίσουμε ποιο ήχοι θα ακούγονται πιο δυνατά από, όπως φαίνεται στην παρακάτω **Εικόνα 3.23**.

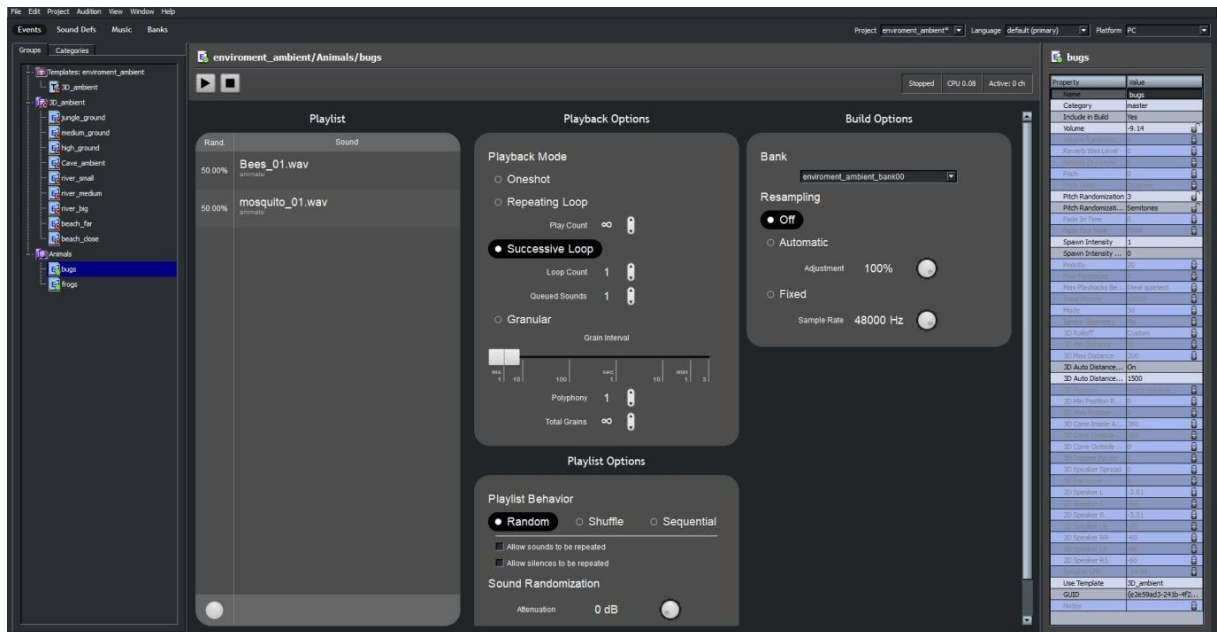


Εικόνα 3.23: Χρήση Occlusion εφέ για το ηχοτοπίο του πέτρινου τούνελ

Επιπλέον ήχοι ζώων

Μέσα στο περιβάλλον έχουν προστεθεί και ορισμένοι άλλοι οι ήχοι οι οποίοι δεν έχουν να κάνουν με ενιαία ηχοτοπία, αλλά με συγκεκριμένες ηχητικές πηγές. Τέτοιοι ήχοι είναι ήχοι κουνουπιών και εντόμων που πετάνε (κυρίως για το περιβάλλον της ζούγκλας), αλλά και ήχοι βατράχων για το περιβάλλον όπου υπάρχει νερό. Τα σημεία αυτά θα τοποθετηθούν συγκεκριμένα σε μέρη από τα οποία ενδέχεται να περάσει ο χρήστης.

Για τη δημιουργία τους φτιάξαμε δύο SimpleTrack Events, ένα με την ονομασία «bugs» και ένα με την ονομασία «frogs». Για το πρώτο χρησιμοποιήσαμε δύο αρχεία με ήχους από μέλισσες και κουνούπια, ενώ προσθέσαμε και randomization +/- 3 ημιτονίων στο pitch. Σε αντίθεση με το επόμενο παράδειγμα, για τους δύο αυτούς ήχους έχουμε χρησιμοποιήσει την επιλογή «Successive Loop», κάτι που σημαίνει ότι, όταν περάσει ο παίκτης από το συγκεκριμένο σημείο, θα παίξει τυχαία ένας από τους δύο ήχους, χωρίς να υπάρξει επανάληψη (loop). Στην παρακάτω φωτογραφία (**Εικόνα 3.24**) βλέπουμε το πώς φαίνεται το συγκεκριμένο SimpleTrack Event για τους ήχους των εντόμων.



Εικόνα 3.24: Δημιουργία ενός Simpletrack Event διάφορους ήχους εντόμων

Για τους ήχους των βατράχων χρησιμοποιήσαμε περισσότερα αρχεία ήχου, έτσι ώστε να είναι πιο τυχαίο το αποτέλεσμα, καθώς θα χρειαστεί να ορίσουμε να παίζουν για άπειρο χρονικό διάστημα. Αυτή τη φορά χρησιμοποιήσαμε την Granular ρύθμιση για επανάληψη και ορίσαμε ένα παράθυρο από περίπου 2 δευτερόλεπτα μέχρι 10 δευτερόλεπτα (Grain Interval), μέσα στα οποία μπορούν να ηχήσουν ταυτόχρονα έως και 2 βατράχια (polyphony = 2). Τέλος χρησιμοποιήσαμε randomization στο pitch των +/- 6 ημιτονίων. Στην παρακάτω φωτογραφία (**Εικόνα 3.25**) μπορούμε να δούμε πώς δείχνουν συνολικά τα αρχεία και οι ρυθμίσεις για τα βατράχια.

Κεφάλαιο 4: Τοποθέτηση των ήχων μέσα στη Cry Engine 3

Σε πολλά μεγάλα studio και projects, προκειμένου να τοποθετηθούν σε ένα περιβάλλον οι ήχοι που έχουν δημιουργηθεί μέσα από προγράμματα όπως το FMOD, απαιτείται μια επιπλέον ειδικότητα, όπως αυτή του Audio Programmer. Συνήθως χρειάζεται καλή επικοινωνία ανάμεσα στους Sound Designers και Audio Programmers ενός project, καθώς και οι δύο είναι υπεύθυνοι για το 100% του ηχητικού αποτελέσματος ενός τίτλου. Μια διαφορά ανάμεσα στους δύο είναι, ότι ο audio programmer πρέπει να έχει γνώσεις της μηχανής γραφικών στην οποία αναπτύσσεται το project, ενώ ο sound designer όχι απαραίτητα. Προκειμένου να υπάρχει καλύτερη επικοινωνία ανάμεσα στον τεράστιο όγκο των ηχητικών event και των παραμέτρων που μπορεί να υπάρχει σε ένα μόνο αρχείο του FMOD, υπάρχει η δυνατότητα ο sound designer να γράψει ορισμένες σημειώσεις (Notes) τόσο στο Event Group στο FMOD, όσο και στο κάθε Multitrack ή Simpletrack Event ξεχωριστά.

Προκειμένου να διαβάσει η Cry Engine το αρχείο που αναπτύξαμε προηγουμένως στο FMOD, πρέπει να το τοποθετήσουμε μέσα στον φάκελο με τα υπόλοιπα sound assets της μηχανής. Στη συγκεκριμένη εργασία έχουμε δημιουργήσει δύο φακέλους, έναν για τους ambient ήχους και άλλον έναν για τα υπόλοιπα ηχητικά events (όπως καταρράκτες κ.α.). Οι φάκελοι αυτοί βρίσκονται στις παρακάτω τοποθεσίες:

Ambient ήχοι:

```
« Cry Engine 3\GameSDK\Sounds\PTYXIAKH_sounds\enviroment_ambient»
```

Υπόλοιποι ήχοι:

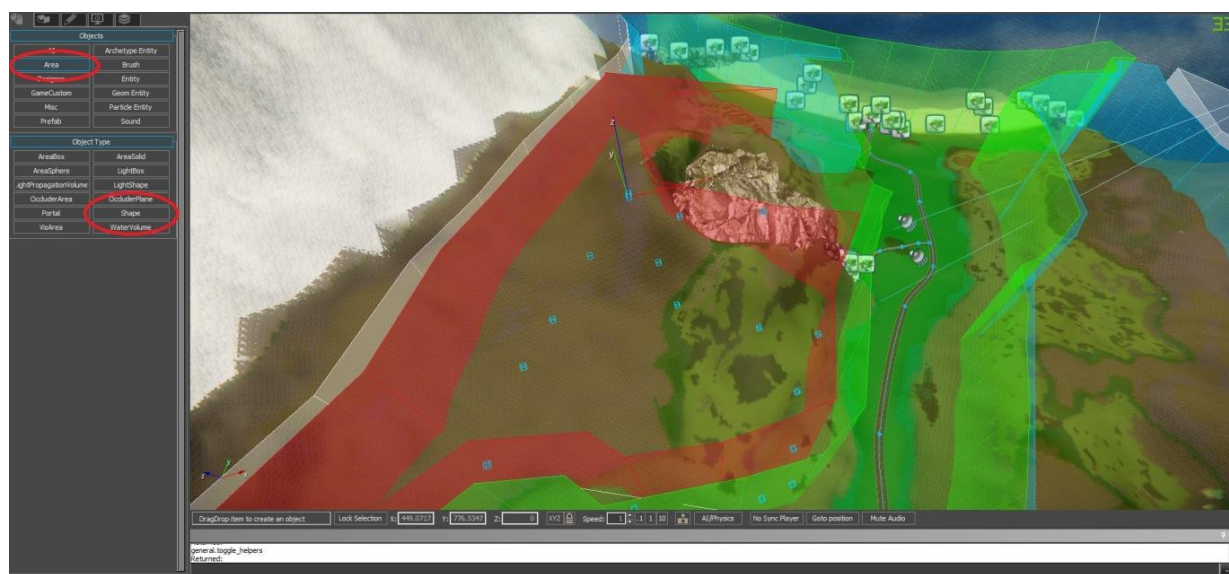
```
« Cry Engine 3\GameSDK\Sounds\PTYXIAKH_sounds\waterfalls»
```

Όπως είπαμε και προηγουμένως στην αρχή της ενότητας για το FMOD, πρέπει μέσα στον τελευταίο φάκελο, το build αρχείο να έχει **το ίδιο όνομα** με τον φάκελο. Έτσι, θα βρούμε μέσα τα αρχεία «*enviroment_ambient.FEV*» και «*waterfalls.FEV*» αντίστοιχα, τα οποία είναι αυτά που θα «διαβάσει» η μηχανή γραφικών. Προκειμένου να ορίσουμε τον χώρο που θα ακούγονται αυτοί οι ήχοι, υπάρχουν διάφορες τεχνικές, ανάλογα και με τη φύση των ήχων. Για τους ambient ήχους (ζούγκλα, ποτάμια κ.τ.λ.) μπορούμε να ορίσουμε μια περιοχή στην οποία θα ακούγεται μονίμως το κάθε ηχητικό background, ενώ για άλλες ηχητικές πηγές θα ορίσουμε ένα σημείο στο περιβάλλον.

Τοποθέτηση των Ambient ήχων

Για την τοποθέτηση των ambient ήχων πρέπει να δημιουργήσουμε τον χώρο που θα ακούγεται ο κάθε ήχος, χρησιμοποιώντας από τα Area Objects το «Shape» (*Objects > Area > Shape*). Αυτό το εργαλείο μας δίνει την δυνατότητα να ορίσουμε μια περιοχή στο περιβάλλον, οποιουδήποτε σχήματος και, στη συνέχεια, να συνδέσουμε σε αυτό το σχήμα κάποια ιδιότητα. Στην περίπτωση του ήχου, θα επιλέξουμε να το συνδέσουμε ένα άλλο object για τον ήχο, το «Ambient Volume» (*Objects > Sound > AmbientVolume*). Τα σχήματα στην περίπτωση ενός εξωτερικού περιβάλλοντος, μπορεί να έχουν ακανόνιστο σχήμα, το οποίο έχουμε δημιουργήσει σημείο-σημείο (δίνεται επίσης η δυνατότητα να μεταβάλλουμε τα σημεία αυτά στη πορεία, σε περίπτωση που το τοπίο μας αλλάξει σχήμα), ενώ υπάρχει ακόμα η δυνατότητα να χρησιμοποιήσουμε έτοιμα σχήματα, όπως τετράγωνα ή παραλληλόγραμμα, στη περίπτωση που έχουμε ambient ήχους δωματίου.

Μια επιπλέον λεπτομέρεια που πρέπει να προσεχθεί είναι τα σχήματα αυτά να καλύπτουν ολόκληρη τη περιοχή στην οποία θα μπορούσε να βρεθεί ο χρήστης στο περιβάλλον, έτσι ώστε να μην υπάρχουν ηχητικά κενά. Στην παρακάτω φωτογραφία (**Εικόνα 4.1**) βλέπουμε το πώς μπορούμε να ορίσουμε τέτοιου είδους περιοχές σε ένα τοπίο.



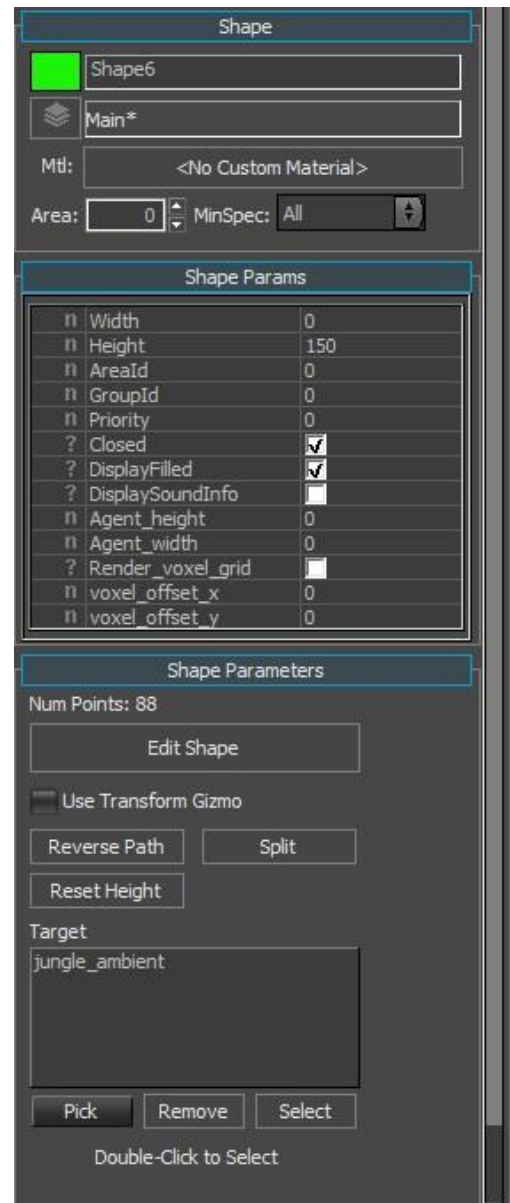
Εικόνα 4.1: Χρήση του εργαλείου “Shape”, για τον ορισμό των διαφορετικών ηχητικών τοπίων

Αφότου δημιουργήσαμε τους χώρους για τους ambient ήχους, το επόμενο βήμα είναι να τους συνδέσουμε με τα sound objects. Τα συγκεκριμένα αντικείμενα μπορούν να τοποθετηθούν οπουδήποτε στο περιβάλλον, αρκεί να γίνονται link με κάποιο shape. Προκειμένου να το κάνουμε αυτό, πρέπει να πάμε στα «Shape Parameters» και να ενεργοποιήσουμε την επιλογή «rick». Στη συνέχεια πρέπει να επιλέξουμε το sound object το οποίο θέλουμε να συνδέσουμε στο σχήμα. Στο επόμενο παράδειγμα (**Εικόνα 4.2**) βλέπουμε πως δείχνει ο συγκεκριμένη διαδικασία για τον ambient ήχο της ζούγκλας.

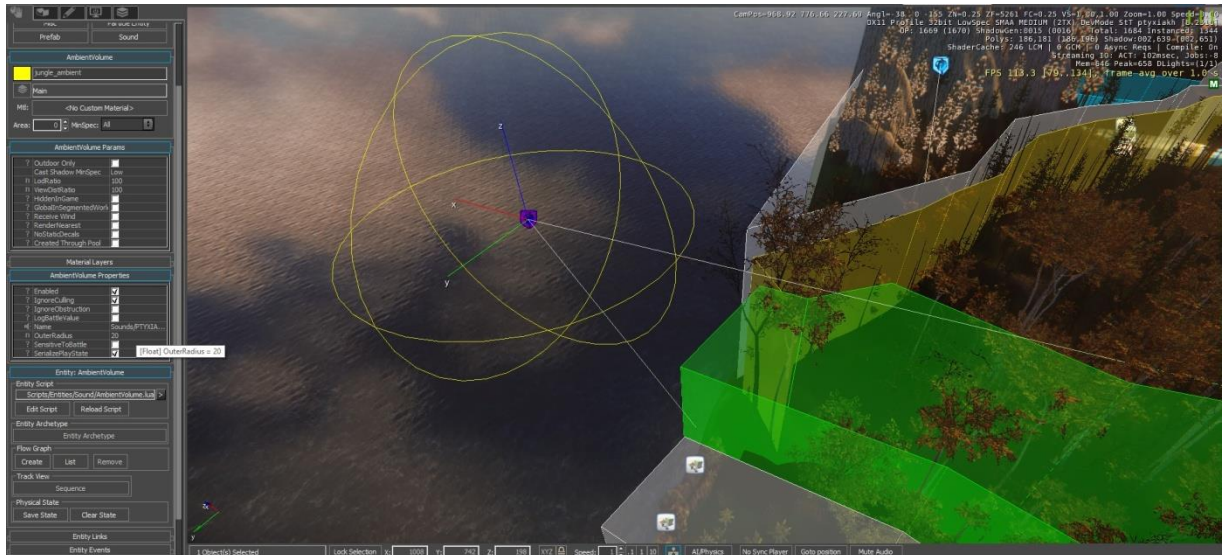
Για να είναι πιο εύκολη η εμφάνιση των σχημάτων σε ένα περιβάλλον, επιλέξαμε επίσης την επιλογή «*DisplayFilled*», το οποίο γεμίζει με χρώμα ολόκληρο το σχήμα, κάνοντάς το να μοιάζει με τοίχο. Επιλέξαμε επίσης και από ένα διαφορετικό χρώμα για κάθε σχήμα, έτσι ώστε να ξεχωρίζει η μια περιοχή με την άλλη, όπως για παράδειγμα το πράσινο χρώμα για το shape της ζούγκλας. Επίσης, υψώσαμε αρκετά όλα τα σχήματα, έτσι ώστε να μπορούμε να εργαστούμε από μεγάλη απόσταση στο περιβάλλον, αλλά και για να καλυφθούν τα οποία κενά που πιθανώς να υπάρχουν. Αυτό σημαίνει ότι τα σχήματα δεν εκτείνονται μόνο αρκετά πάνω από το έδαφος, αλλά και κάτω από αυτό.

Τα «*AmbientSound*» objects τα οποία συνδέονται με το κάθε σχήμα είναι και αυτά που μας επιτρέπουν να φορτώσουμε τα αρχεία ήχου του FMOD. Ωστόσο μας αφήνουν να κάνουμε και ορισμένες ρυθμίσεις που επηρεάζουν το άκουσμα των ήχων, όπως να ορίσουμε την απόσταση του fade-in και fade-out που θα ακούγονται πριν εισέλθουμε και εξέλθουμε από έναν τέτοιο χώρο αντίστοιχα. Στην παρακάτω φωτογραφία (**Εικόνα 4.3**) βλέπουμε το πώς δείχνει ένα Sound AmbientVolume Object όταν είναι συνδεδεμένο με μια περιοχή, ενώ αριστερά μπορούμε να ορίσουμε τις παραμέτρους του.

Η παράμετρος «*OuterRadius*» έχει οριστεί στα 20 μέτρα, κάτι που σημαίνει ότι το ηχητικό τοπίο θα εκτείνεται για 20 μέτρα έξω από τα όρια του σχήματος που έχουμε ορίσει, δημιουργώντας έτσι ένα fade-in και fade-out εφέ, που κάνει το ξεκίνημα του ήχου λιγότερο απότομο. Σε άλλα σχήματα έχουμε ορίσει διαφορετικό νούμερο στην αντίστοιχη ρύθμιση, όπως στη περίπτωση της σπηλιάς (*OuterRadius* = 5). Επιπλέον πρέπει ο ήχος να είναι ενεργοποιημένος (επιλογή «*Enabled*») και τέλος να επιλέξουμε το ηχητικό asset που έχουμε δημιουργήσει από την παράμετρο «*Name*».



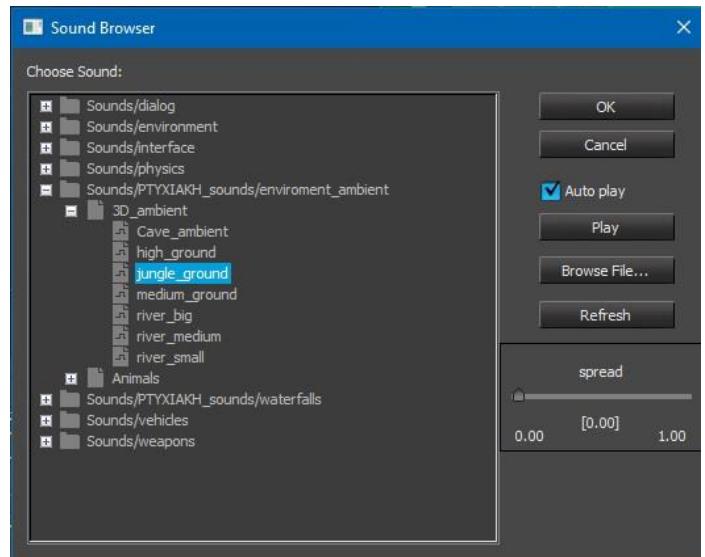
Εικόνα 4.2: Οι παράμετροι του εργαλείου «*Shape*»



Εικόνα 4.3: Σύνδεση ενός AmbientSound Object με μια περιοχή (Shape)

Πατώντας επάνω στην παράμετρο Name μπορούμε να ανοίξουμε τον Sound Browser μέσα στον οποίο θα βρούμε όλα τα ηχητικά αρχεία που έχουμε δημιουργήσει με το FMOD. Παρατηρούμε επίσης, όπως φαίνεται και στην **Εικόνα 4.4**, ότι η CryEngine αναγνωρίζει άμεσα και τις παραμέτρους που έχουμε ορίσει για το κάθε αρχείο, όπως στο συγκεκριμένο παράδειγμα με τον ήχο της ζούγκλας και την παράμετρο «spread». Μας δίνει επίσης τη δυνατότητα να κάνουμε preview το ηχητικό απόσπασμα, μεταβάλλοντας σε πραγματικό χρόνο τις παραμέτρους, πριν τις τοποθετήσουμε στο sound object.

Πατώντας «OK» μπορούμε να τοποθετήσουμε το αρχείο πλέον στο περιβάλλον, κάτι που σημαίνει ότι, κάθε φορά που θα εισέλθουμε στο πράσινο σχήμα θα ενεργοποιείται ο ambient ήχος «jungle_ground», έχοντας ένα fade-in/ fade-out 20 μέτρων από τα όρια του σχήματος. Αυτό επίσης σημαίνει ότι στα όρια των σχημάτων μπορεί να ακούσουμε ήχους από δύο διαφορετικά ηχητικά τοπία, κάτι όμως που είναι πιο ρεαλιστικό στο άκουσμα από το να υπάρχει άμεση διακοπή ενός ήχου.



Εικόνα 4.4: Εισαγωγή των αρχείων ήχων του FMOD στο Sound Object της Cry Engine 3

Είναι επίσης εφικτό μέσα σε μια περιοχή να υπάρχουν και επιπλέον σχήματα με άλλους ambient ήχους, όπως συμβαίνει στο περιβάλλον μας με τους ήχους των τριών ποταμών. Σε αυτή την περίπτωση έχουμε δημιουργήσει τρία διαφορετικά σχήματα τα οποία

συνδέονται με τρία διαφορετικά sound objects, ένα για κάθε ποτάμι. Στο συγκεκριμένο περιβάλλον, όταν ο παίκτης πλησιάσει κοντά στην περιοχή του ποταμιού, θα ακούσει τόσο τον ήχο των πουλιών από το γενικότερο περιβάλλον, αλλά και τον ήχο του ποταμιού. Η ρύθμιση του OuterRadius έχει ορισθεί εδώ στα 20 μέτρα, έτσι ώστε ο ήχος του νερού να αποκαλύπτει νωρίτερα στον παίκτη την τοποθεσία του ποταμιού. Παρακάτω (**Εικόνα 4.5**) βλέπουμε πώς δείχνουν στο περιβάλλον τα σχήματα που έχουμε επιλέξει για τα τρία ποτάμια:



Εικόνα 4.5: Δημιουργία των σχημάτων (shape) για τα ηχητικά τοπία των 3 καταρρακτών

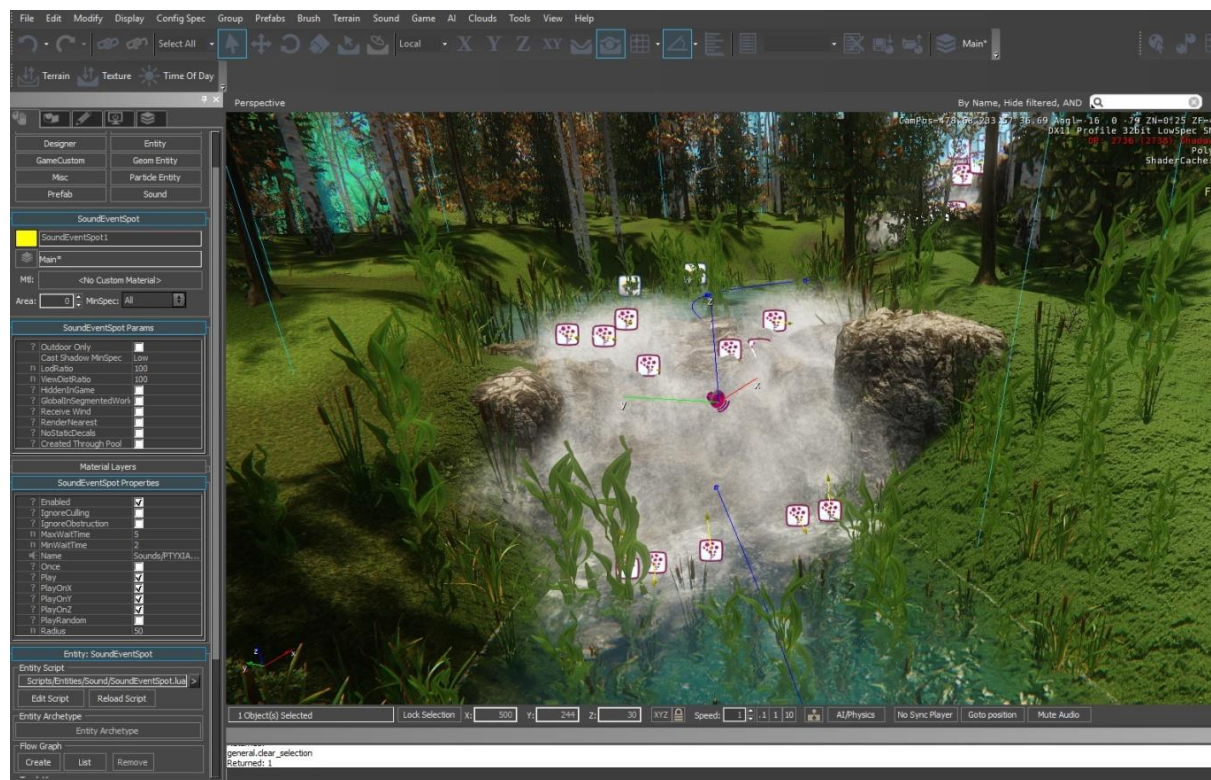
Τέλος, με τον ίδιο τρόπο εργαστήκαμε και για τα σχήματα των υπόλοιπων ambient τοπίων του χάρτη, όπως αυτά της σπηλιάς, της παραλίας, του φθινοπωρινού τοπίου και του βουνού.

Τοποθέτηση ήχων καταρρακτών

Για να τοποθετήσουμε συγκεκριμένες ηχητικές πηγές δεν χρειάζεται να ορίσουμε κάποιο περιβάλλον ή σχήμα, αλλά να ορίσουμε την πηγή του ήχου κάνοντας χρήση ενός άλλου sound object, του «SoundEventSpot». Τα συγκεκριμένα σημεία μπορούμε να τα τοποθετήσουμε οπουδήποτε, με τον ήχο να επηρεάζεται πλέον από την απόσταση που του έχουμε ορίσει. Η επιλογή του ηχητικού αρχείου γίνεται με παρόμοιο τρόπο, όπως και στην ενότητα με τους ambient ήχους. Παρακάτω (**Εικόνα 4.6**) βλέπουμε το πώς μπορούμε να ορίσουμε την πηγή ήχου για τους καταρράκτες, τοποθετώντας το SoundEventSpot αντικείμενο περίπου στο κέντρο του κάθε καταρράκτη.

Η μοναδική τροποποίηση που έχουμε κάνει στα properties του object είναι στο Radius, το οποίο το έχουμε βάλει να ταιριάζει με το radius του αρχείου που έχουμε δημιουργήσει στο

FMOD. Για τον πιο μικρό καταρράκτη, βλέπουμε ότι υπάρχει ένα μέγιστο απόστασης των 50 μέτρων, μέσα στα οποία ξεκινάει να ακούγεται ο ήχος, ο οποίος δυναμώνει όσο ο παίκτης πλησιάζει στη πηγή. Με τον ίδιο τρόπο θα εργασθούμε για να τοποθετήσαμε στο περιβάλλον τους ήχους και των υπόλοιπων δύο καταρρακτών.

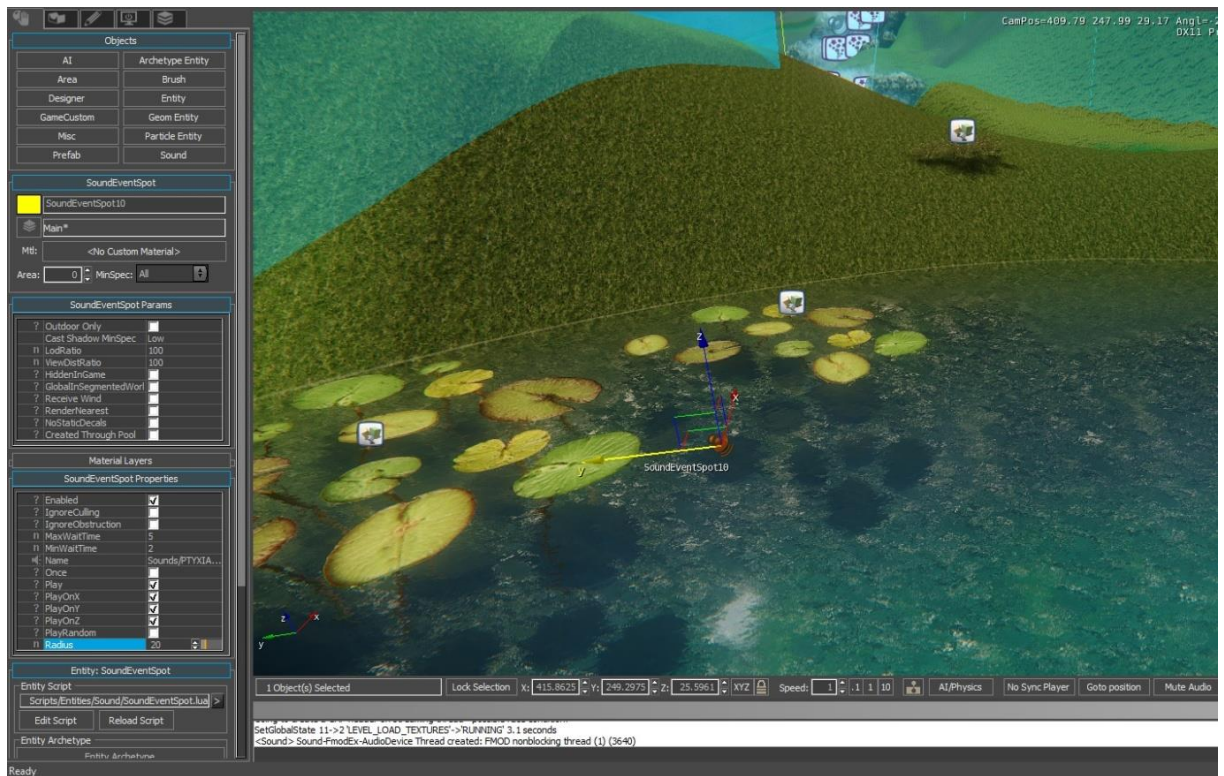


Εικόνα 4.6: Τοποθέτηση ενός SoundEventSpot Object στο κέντρο, για τον ορισμό της ηχητικής προέλευσης ενός καταρράκτη

Τοποθέτηση μεμονωμένων ήχων

Για τους ήχους που θέλουμε να έχουν συγκεκριμένη πηγή προέλευσης, όπως τα δύο SimpleTrack Events που δημιουργήσαμε προηγουμένως για τους ήχους των εντόμων και των βατράχων, θα πρέπει να τοποθετήσουμε με το χέρι τις ηχητικές πηγές αυτές στο περιβάλλον, κάνοντας χρήση των SoundEventSpots. Αυτό σημαίνει ότι πρέπει να τοποθετήσουμε τους ήχους σε σημεία που θα είχαν σημασία, όπως τους ήχους των βατράχων στην λίμνη και τα κουνούπια σε σημεία με πυκνή βλάστηση.

Στην παρακάτω φωτογραφία (**Εικόνα 4.7**) βλέπουμε την τοποθέτηση ενός SoundEventSpot στην επιφάνεια της λίμνης, από το οποίο έχουμε επιλέξει να ακούγεται το SimpleTrack Event με τα βατράχια. Προκειμένου το ηχητικό αποτέλεσμα να ακούγεται σε ένα μεγάλο εύρος, έχουμε ορίσει το Radius της ηχητικής πηγής στα 20 μέτρα. Ανάλογα με το σημείο του περιβάλλοντος ενδέχεται να ορίσουμε διαφορετικά και ανά περίπτωση τις αποστάσεις των υπολοίπων SoundEventSpots.



Εικόνα 4.7: Τοποθέτηση ενός *SoundEventSpot*, για τον ορισμό του σημείο ηχητικής προέλευσης των ήχων των βατράχων, από την επιφάνεια της λίμνης

Παρατηρήσεις και συμπεράσματα για την εφαρμογή

Η Export διαδικασία του χάρτη στη μηχανή

Το τελευταίο στάδιο για την ολοκλήρωση του χάρτη δεν είναι άλλο από το export του στην GameSDK μηχανή της Cry Engine 3. Οι τρεις σημαντικές διαδικασίες που πρέπει να ολοκληρωθούν προκειμένου το export να είναι πλήρες είναι οι εξής: Από την επιλογή του menu πρέπει πρώτα να σώσουμε τον χάρτη, ώστε το τελευταίο save back up να είναι συμβαδίζει με το τελικό build, κάτι που είναι περισσότερο πρακτικό για εμάς. Δεύτερον, πρέπει να επιλέξουμε την επιλογή «*Generate Terrain Texture*», κάτι που ανανεώνει συνολικά στην μηχανή όλες τις αλλαγές που έχουμε κάνει στις επιφάνειες του εδάφους. Αυτές μπορεί να έχουν να κάνουν τόσο με την τοποθέτηση κάποια άλλης επιφάνειας σε μια ήδη υπάρχουσα, όσο και την αλλαγή κάποιου χρώματος που ενδεχομένως κάναμε προηγουμένως. Τέλος, πρέπει να επιλέξουμε το «*Export to Engine*», κάτι που μεταφέρει ολόκληρο το περιεχόμενο του χάρτη σε playable μορφή στο Game SDK.

Το GameSDK περιλαμβάνεται μέσα στον ίδιο φάκελο με τον editor και μπορούμε να δοκιμάσουμε τον χάρτη μας ακριβώς όπως θα φαινόταν και στο τελικό προϊόν και μέσα από ένα UI που θυμίζει έντονα αυτό σημερινών βιντεοπαιχνιδιών, με επιλογές για ρυθμίσεις των γραφικών (low, medium, high, κτλ), αλλά και άλλες ρυθμίσεις για τον ήχο, τον χειρισμό και την ανάλυση. Συγκεκριμένα προτείνουμε το περιβάλλον να εμφανισθεί σε ανάλυση 1920x1080 και σε Low Settings γραφικών ή διαφορετικά σε ανάλυση 1280x720 Low settings για συστήματα με λιγότερο δυνατή GPU. Το σύστημα στο οποίο δοκιμάστηκαν οι δύο προηγούμενες ρυθμίσεις διαθέτει 2GB μνήμης στην κάρτα γραφικών, με το αποτέλεσμα να εμφανίζεται στα 60 καρέ ανά δευτερόλεπτο.

Εκτέλεση της εφαρμογής μέσα από τον Editor

Εάν θέλει κάποιος να μελετήσει τον χάρτη και τη κατασκευή του, προτείνουμε να τον ανοίξει μέσα από την επιλογή «*Editor*» και όχι μέσα από το «*GameSDK*». Ο Editor της μηχανής βρίσκεται στο path: *Cry Engine 3 > Bin32 > Editor*. Μέσα στον Editor μπορούμε να φορτώσουμε τον φάκελο «PTYΧΙΑΚΗ», ο οποίος είναι ουσιαστικά ολόκληρος ο χάρτης που έχουμε δημιουργήσει. Με τη χρήση της κάμερας μπορούμε να περιπλανηθούμε ελεύθερα σε ολόκληρο το χάρτη, ενώ με τη χρήση του SHIFT μπορούμε να επιταχύνουμε την κίνηση. Πατώντας επίσης Control+G μπορούμε να δοκιμάσουμε το περιβάλλον από την οπτική γωνία του παίκτη.

Πώς επιβαρύνει ο ήχος μια εφαρμογή

Όπως συμβαίνει σε κάθε παιχνίδι, οποιαδήποτε αλλαγή ή προσθήκη έχει υπολογιστικό κόστος. Οτιδήποτε προσθέσαμε επάνω στο χάρτη, όπως δέντρα, αντικείμενα, νερό κ.α., επιβαρύνουν τόσο την GPU όσο και την CPU. Το ίδιο όμως συμβαίνει και με τον τομέα του ήχου. Συγκεκριμένα, λειτουργίες του FMOD οι οποίες εκτελούνται σε πραγματικό χρόνο από τη μηχανή, όπως randomizations, αυτοματισμοί, sprawn παράθυρα, εφέ κ.α., έχουν επίσης κόστος και καταναλώνεται κυρίως η δύναμη της CPU. Γενικότερα ο ήχος των παιχνιδιών είναι κυρίως CPU based περιβάλλον, ενώ ο οπτικός τομέας περισσότερο GPU περιβάλλον.

MP3 vs. WAV format. Ποιο είναι κατάλληλο;

Μια σημαντική παρατήρηση που ανακαλύψαμε κατά τη διάρκεια της επεξεργασίας των ήχων, είχε να κάνει με το MP3 format. Ενώ πολλοί ήχοι, οι οποίοι δεν χρειαζόταν να είναι σε καλή ποιότητα (όπως για παράδειγμα κάποιο χαμηλής συχνότητας βουητό) μπορούσαν να βρίσκονται σε μορφή MP3, αυτό τελικά δεν ήταν εφικτό, αφού τα MP3 αρχεία είναι ακατάλληλα για οποιαδήποτε συνεχόμενη επανάληψη (loop). Ο λόγος ήταν ότι το MP3 ως format έχει κάποιο πολύ μικρό χρονικό κενό στην αρχή του κάθε αρχείου με αποτέλεσμα, όταν γίνεται επανάληψη αυτού του αρχείου, να ακούγεται κάθε φορά ένα κενό, ως ένα «κλικ». Για αυτόν τον λόγο έπρεπε να μετατρέψουμε κάθε MP3 αρχείο, με τη βοήθεια του Audacity, σε WAV format, το οποίο ήταν ιδανικό για looped ήχους.

Τι είναι το FSB extension;

Το .FSB file extension είναι ένα από τα αρχεία που δημιουργεί στον Build φάκελο το FMOD, μαζί με τα .FDP (FMOD Project File) και .FEV (FMOD Audio Events) αρχεία και σημαίνει «FMOD Sample Bank Format» file. Ο φάκελος αυτός περιέχει πληροφορίες ήχου, όπως για μουσική και ομιλία, για βιντεοπαιχνίδια που σχεδιάστηκαν για κονσόλες όπως PlayStation και Xbox.

Βιβλιογραφία και πηγές της εργασίας

Βιβλίο: “Game Sound” – Karen Collins

“An introduction to the history, Theory, and practice of Video Game Music and sound Design”

Βιβλίο: “ Soundscape Ecology Principles” – Almo Farina

“Soundscape Ecology: Principles, Patterns, Methods and Applications”

Διαλέξεις του καθηγητή Κώστα Αναγνώστου

Ο Κώστας Αναγνώστου έχει διδακτορικό στα Γραφικά με Υπολογιστή και έχει εργαστεί για χρόνια στην Microsoft Game Studios, στην Αγγλία, αναπτύσσοντας παιχνίδια για το Xbox και Xbox360. Έχει λάβει μέρος στην ανάπτυξη των παιχνιδιών Conker: Live and Reloaded για το Xbox και Kameo: Elements of Power, Perfect Dark Zero και Forza 2 για το Xbox360.

Διδάσκει Τεχνολογίες Ψυχαγωγικού Λογισμικού (videogames), Γραφικά με Υπολογιστή και Εικονική Πραγματικότητα στο Τμήμα Πληροφορικής του Ιονίου Πανεπιστημίου.

Στα ερευνητικά του ενδιαφέροντα περιλαμβάνονται οι Τεχνολογίες Βιντεοπαιχνιδιών και Εικονικών Κόσμων, με ιδιαίτερη έμφαση στην εφαρμογή τους στη μάθηση και στην κοινωνικοποίηση, και οι Τεχνολογίες τριδιάστατων γραφικών με υπολογιστή.

Το site του βιβλίου του Κ. Αναγνώστου, «Βιντεοπαιχνίδια: Βιομηχανία και ανάπτυξη»:

<http://videogamesbook.wordpress.com/extra/>

Επιπλέον αρθρογραφία που χρησιμοποιήθηκε από Online σελίδες

<https://www.emusician.com/>

<https://www.gamedesigning.org/career/video-game-engines/>

<http://filmsound.org/game-audio/>

<https://en.wikipedia.org/wiki/CryEngine>

Πηγές ηχητικού υλικού

<https://sonniss.com/>

<https://www.soundsnap.com>

<http://www.grsites.com/archive/sounds/>

<https://freesound.org/>

FMOD Designer 2010

Download της εφαρμογής και του Manual “FMOD Designer User Manual”

<https://www.fmod.com/>

Cry Engine 3

Download του “Free SDK”, έκδοση 3.5.8

<https://www.cryengine.com/>

Cry Engine 3 Documentation

To Online Manual της Cry Engine 3

<http://docs.cryengine.com/display/SDKDOC2/Home>

Audacity

Η δωρεάν εφαρμογή “Audacity” που χρησιμοποιήθηκε για την επιπλέον επεξεργασία ήχου

<https://www.audacityteam.org/>

Βιβλιοθήκη ηχητικών αρχείων που χρησιμοποιήθηκε

Τα ηχητικά αρχεία που έχουν χρησιμοποιηθεί είναι στο σύνολό τους WAV αρχεία, υψηλού bitrate, τόσο μονοφωνικά αλλά και στερεοφωνικά και έχουν βρεθεί από τις παρακάτω online βιβλιοθήκες:

<https://www.soundsnap.com/tags/rainforest>

<https://sonniss.com/>

<http://www.grsites.com/archive/sounds/>

<https://freesound.org/>

Πολλά αρχεία τα οποία κρίθηκαν μεγάλα σε διάρκεια, επεξεργάστηκαν στη πορεία με τη βοήθεια του Audacity, προκειμένου να χρησιμοποιηθεί ένα μικρό απόσπασμα αυτών στο τελικό mixing μέσω του FMOD. Παρακάτω ακολουθεί η αναλυτική λίστα των αρχείων που έχουν χρησιμοποιηθεί σε όλα τα ηχητικά events και ambiences:

Filename	Quality	Resample Rate	Original Rate	Bit depth	Channels	Sound type	File size	Used
 FIRE_BURNING_OFF_MEDIUM_ESTABLISHED_CLOSE_LOOP02_STEREO.wav	bank	Automatic Rate	48000 Hz	floating point	Stereo	Microsoft WAV	5605 kb	SW 3D
 waterfall\waterfall_big.wav	bank	Automatic Rate	22050 Hz	16-bit	Mono	Microsoft WAV	359 kb	SW 3D
 waterfall\waterfall_medium.wav	bank	Automatic Rate	22050 Hz	16-bit	Mono	Microsoft WAV	316 kb	SW 3D
 waterfall\waterfall_small.wav	bank	Automatic Rate	96000 Hz	24-bit	Stereo	Microsoft WAV	5222 kb	SW 3D

Filename	Quality	Resample Rate	Original Rate	Bit depth	Channels	Sound type	File size	Used
animals\Bees_01.wav	bank	No Resampling	11025 Hz	16-bit	Stereo	Microsoft WAV	249 kb	SW 3D
animals\birds\bird_01.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	40 kb	SW 3D
animals\birds\bird_02.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	61 kb	SW 3D
animals\birds\bird_03.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	39 kb	SW 3D
animals\birds\birds_flying.wav	bank	Automatic Rate	22050 Hz	8-bit	Mono	Microsoft WAV	100 kb	SW 3D
animals\birds\canary_01.wav	bank	Automatic Rate	11025 Hz	8-bit	Mono	Microsoft WAV	10 kb	SW 3D
animals\birds\canary_02.wav	bank	Automatic Rate	11025 Hz	8-bit	Mono	Microsoft WAV	10 kb	SW 3D
animals\birds\canary_03.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	24 kb	SW 3D
animals\birds\canary_04.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	23 kb	SW 3D
animals\birds\canary_07.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	10 kb	SW 3D
animals\birds\crow.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	8 kb	SW 3D
animals\birds\crow_02.wav	bank	Automatic Rate	11025 Hz	8-bit	Mono	Microsoft WAV	22 kb	SW 3D
animals\birds\hawk.wav	bank	Automatic Rate	22050 Hz	16-bit	Mono	Microsoft WAV	138 kb	SW 3D
animals\birds\kookabura_birds_01.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	478 kb	SW 3D
animals\birds\kookabura_birds_02.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	409 kb	SW 3D
animals\birds\kookabura_birds_03.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	463 kb	SW 3D
animals\birds\peacock.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	61 kb	SW 3D
animals\birds\seagull_01.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	67 kb	SW 3D
animals\birds\seagull_02.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	225 kb	SW 3D
animals\birds\seagull_03.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	335 kb	SW 3D
animals\birds\seagull_04.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	325 kb	SW 3D
animals\birds\vulture_01.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	61 kb	SW 3D
animals\frog_01.wav	bank	No Resampling	44100 Hz	16-bit	Stereo	Microsoft WAV	183 kb	SW 3D
animals\frog_02.wav	bank	No Resampling	44100 Hz	16-bit	Stereo	Microsoft WAV	113 kb	SW 3D
animals\frog_03.wav	bank	No Resampling	44100 Hz	16-bit	Stereo	Microsoft WAV	223 kb	SW 3D
animals\frog_04.wav	bank	No Resampling	44100 Hz	16-bit	Stereo	Microsoft WAV	463 kb	SW 3D
animals\frog_05.wav	bank	No Resampling	44100 Hz	16-bit	Stereo	Microsoft WAV	847 kb	SW 3D
animals\frog_06.wav	bank	No Resampling	44100 Hz	16-bit	Stereo	Microsoft WAV	565 kb	SW 3D
animals\monkey.wav	bank	Automatic Rate	11025 Hz	8-bit	Mono	Microsoft WAV	85 kb	SW 3D
animals\monkey_01.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	100 kb	SW 3D
animals\monkey_02.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	69 kb	SW 3D
animals\mosquito_01.wav	bank	No Resampling	11025 Hz	16-bit	Mono	Microsoft WAV	104 kb	SW 3D
forest sounds\forest_day.wav	bank	Automatic Rate	48000 Hz	floating point	Stereo	Microsoft WAV	14847 kb	SW 3D
forest sounds\forest_night.wav	bank	Automatic Rate	48000 Hz	floating point	Stereo	Microsoft WAV	7486 kb	SW 3D
Impact_Boom_Distorted_02.wav	bank	No Resampling	96000 Hz	24-bit	Stereo	Microsoft WAV	3414 kb	SW 2D
low oscillating distortion 3.wav	bank	Automatic Rate	96000 Hz	floating point	Stereo	Microsoft WAV	7176 kb	SW 3D
music-ambiences\hollow_drone_001.wav	bank	Automatic Rate	96000 Hz	24-bit	Stereo	Microsoft WAV	33400 kb	SW 3D
ocean\beach_ambience_close.wav	bank	Automatic Rate	44100 Hz	floating point	Stereo	Microsoft WAV	8678 kb	SW 3D
ocean\beach_ambience_far.wav	bank	Automatic Rate	48000 Hz	floating point	Stereo	Microsoft WAV	11389 kb	SW 3D
rocks-falling\rock tumble down debris long 06.wav	bank	No Resampling	96000 Hz	24-bit	Stereo	Microsoft WAV	2963 kb	SW 3D
rocks-falling\rocks stream down heavy 02.wav	bank	No Resampling	96000 Hz	floating point	Stereo	Microsoft WAV	5411 kb	SW 3D
rocks-falling\soil and dirt debris heavy 13.wav	bank	No Resampling	96000 Hz	24-bit	Stereo	Microsoft WAV	1323 kb	SW 3D
waterfall\River_Dam_Light_Fienup_001.wav	bank	Automatic Rate	96000 Hz	floating point	Stereo	Microsoft WAV	7735 kb	SW 3D
waterfall\Tropical_Rain2.wav	bank	Automatic Rate	48000 Hz	floating point	Stereo	Microsoft WAV	7542 kb	SW 3D
waterfall\water drops(many)\water_drops1.wav	bank	Automatic Rate	22050 Hz	floating point	Mono	Microsoft WAV	57 kb	SW 3D
waterfall\water drops(many)\water_drops2.wav	bank	Automatic Rate	22050 Hz	floating point	Mono	Microsoft WAV	76 kb	SW 3D
waterfall\water drops(many)\water_drops3.wav	bank	Automatic Rate	22050 Hz	floating point	Mono	Microsoft WAV	25 kb	SW 3D
waterfall\water drops(many)\water_drops4.wav	bank	Automatic Rate	22050 Hz	floating point	Mono	Microsoft WAV	28 kb	SW 3D
waterfall\water drops(many)\water_drops5.wav	bank	Automatic Rate	22050 Hz	floating point	Mono	Microsoft WAV	12 kb	SW 3D
wind\cave_wind.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	5771 kb	SW 3D
wind\spooky_wind.wav	bank	Automatic Rate	11025 Hz	8-bit	Mono	Microsoft WAV	61 kb	SW 3D
wind\thunders1.wav	bank	Automatic Rate	44100 Hz	floating point	Stereo	Microsoft WAV	2191 kb	SW 3D
wind\thunders2.wav	bank	Automatic Rate	44100 Hz	floating point	Stereo	Microsoft WAV	3324 kb	SW 3D
wind\thunders3.wav	bank	Automatic Rate	44100 Hz	floating point	Stereo	Microsoft WAV	2245 kb	SW 3D
wind\wind01.wav	bank	Automatic Rate	11025 Hz	16-bit	Mono	Microsoft WAV	307 kb	SW 3D
wind\wind02.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	4558 kb	SW 3D
wind\wind04.wav	bank	Automatic Rate	44100 Hz	16-bit	Stereo	Microsoft WAV	3852 kb	SW 3D