

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Πτυχιακή εργασία

ΜΕΘΟΔΟΙ ΚΑΙ ΤΕΧΝΙΚΕΣ ΑΝΑΠΤΥΞΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΑΙΧΝΙΔΙΩΝ



Πανεπιστήμιο Ιωαννίνων

**Βάκος Μαρίνος
Α.Μ:1342**

**Επιβλέπον
Καρβέλης Πέτρος**

ΜΕΘΟΔΟΙ ΚΑΙ ΤΕΧΝΙΚΕΣ ΑΝΑΠΤΥΞΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΠΑΙΧΝΙΔΙΩΝ

Στην παρούσα πτυχιακή εργασία αναφέρονται οι μέθοδοι και οι τεχνικές που εφαρμόζονται για την παραγωγή ηλεκτρονικών παιχνιδιών αλλά και τα απαραίτητα εργαλεία για την μοντελοποίηση και ανάπτυξη τέτοιων έργων. Παρόλο τη δυσκολία και την τεχνογνωσία που ήταν απαιτητή από κάποιον για να ασχοληθεί με το συγκεκριμένο αντικείμενο, πλέον δίνεται η δυνατότητα για ανάπτυξη ηλεκτρονικών παιχνιδιών με πληθώρα βοηθημάτων από το διαδίκτυο. Η μελέτη και υλοποίηση των τεχνικών που μελετούνται στην παρούσα πτυχιακή είναι πάνω στο λογισμικό της Unreal Engine 4, που παρέχεται δωρεάν από την Epic Games. Παρόλα αυτά αρκετές από τις τεχνολογίες της Unreal, τις συναντάει κάποιος και σε άλλες μηχανές γραφικών όπως στη unity με αποτέλεσμα να μπορέσει να εφαρμόσει αρκετά από τα παραδείγματα και σε αυτές.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή στην Unreal Engine 4

- 1.1 Unreal Engine 4
- 1.2 Τεχνικές την Unreal Engine 4

Κεφάλαιο 2

Φωτισμός

- 2.1 Κοινά χαρακτηριστικά των ειδών φωτισμού
- 2.2 Είδοι φωτισμού

Κεφάλαιο 3

Ατμόσφαιρα

- 3.1 BP_Sky_Sphere και cubemaps
- 3.2 Sky atmosphere

Κεφάλαιο 4

Εμπλουτισμός επιπέδου

- 4.1 Δημιουργία και επεξεργασία εδάφους
 - 4.1.1 Έδαφος-Landscape
 - 4.1.2 Γλυπτική Εδάφους-Sculpting
- 4.2 3D assets
 - 4.2.1 Κατασκευή μοντέλων
 - 4.2.2 Γλυπτική μοντέλων

- 4.2.3 UV χαρτογράφηση
- 4.3 Βελτιστοποίησης και εξοικονόμησης πόρων
 - 4.3.1 Επίπεδο λεπτομέρειας μοντέλου -LOD
 - 4.3.2 Σύγκρουση - Collision
 - 4.3.3 Εξαφάνιση μοντέλου - Culling

Κεφάλαιο 5

Υφές και υλικά

- 5.1 Υφές
- 5.2 Υλικά εδάφους
- 5.3 Υλικά μοντέλων
 - 5.3.1 Ζωγραφιστές υφές
 - 5.3.2 Διαδικαστικά δημιουργημένες υφές
 - 5.3.3 Virtual Textures

Κεφάλαιο 6

Δράστες

- 6.1 Blueprint Actors και User Interface
- 6.2 Τεχνητή νοημοσύνη
 - 6.2.1 Blackboard
 - 6.2.2 Δέντρα Συμπεριφοράς
 - 6.2.3 Μηχανές πεπερασμένων καταστάσεων
 - 6.2.4 Λόγοι μεταφοράς από μηχανές πεπερασμένων καταστάσεων σε δέντρα συμπεριφοράς

Κεφάλαιο 1 Εισαγωγή στην Unreal Engine 4

1.1 Unreal Engine 4

Η τέταρτη έκδοση της Unreal έγινε διαθέσιμη στο κοινό στις 29 Απριλίου, 2014. Αρχικά με ένα ποσό 19 δολαρίων μηνιαίος ο οποιοσδήποτε έχει πρόσβαση στην unreal engine 4 με την εταιρία να έχει μερίδιο 5% των εσόδων πώλησης από οποιοδήποτε παιχνίδι που παραγόντουσαν από ανεξάρτητους προγραμματιστές. Στις 4 Σεπτεμβρίου του ίδιου έτους, η Epic Games έδωσε δωρεάν εκδόσεις του λογισμικού σε σχολεία της δευτεροβάθμιας και τριτοβάθμιας εκπαίδευσης, μαζί με ένα ποσό των 5 εκατομμυρίων στον οργανισμό Unreal Dev Grants που παρείχε επιχορηγήσεις, σε δημιουργικά έργα, που χρησιμοποιούσαν την Unreal. Τον Μάρτιο του 2015 η Epic Games έδωσε χωρίς συνδρομή την μηχανή γραφικών στο κοινό, με συχνές ανανεώσεις και καινούργιες τεχνολογίες, ενώ ταυτόχρονα κρατάει ένα ευδιάκριτο και διαυγές περιβάλλον για τον προγραμματιστή. Μετά από έξι χρόνια υποστήριξης στην ue4 η Epic Games ανακοίνωσε πως στα τέλη του 2021 θα είναι έτοιμη για την προώθηση της unreal engine 5 με ολοκαίνουργιες τεχνολογίες όπως την Nanite και ανανεωμένη έκδοση του Niagara Particle System.

1.2 Τεχνολογίες της Unreal Engine 4

Κάθε κύκλος ανάπτυξης ενός παιχνιδιού, περνάει από τρία βασικά στάδια ξεκινώντας πάντα από την ίδια την ιδέα που στις περισσότερες περιπτώσεις είναι η συλλογική προσπάθεια των προγραμματιστών, αλλά και τον στελεχών, αν η ανάπτυξη γίνεται από ένα μεγάλο studio. Σε ειδικές περιπτώσεις είναι η μετενσάρκωση της ιδέας ενός έμπειρου επικεφαλής ομάδας, που έχει εμπειρία με το αντικείμενο. Παρόλα αυτά ένα παιχνίδι δεν θα μπει ποτέ σε παραγωγή αν πρώτα δεν ελεγχθεί έστω και στην εμβρυϊκή του μορφή για το αν πραγματικά αξίζει την δαπάνη υλών και ωρών εργασίας. Αυτό επιτυγχάνεται από την υλοποίηση ενός αρχικού πρωτότυπου, που στην συνέχεια εμπλουτίζεται αν είναι διακριτή η επιτυχία του. Η μεταβολή του πρωτοτύπου σε ένα ατόφιο παιχνίδι, μπορεί να κρατήσει από εβδομάδες μέχρι και χρόνια, ανάλογα με το σκέλος και τις λειτουργίες του παιχνιδιού. Για αυτόν τον λόγο η κάθε εταιρία ή η κάθε ανεξάρτητη ομάδα προσπαθεί να μειώσει τον χρόνο ανάπτυξης παιχνιδιών, υιοθετώντας κάποια στάνταρ. Το πιο σημαντικό από αυτά είναι η απόφαση για το ποιό λογισμικό θα χρησιμοποιηθεί. Το μεγαλύτερο ποσοστό των εταιρειών έχει αναπτύξει το δικό της λογισμικό, ανάλογα με τις γνώσεις και τις προγραμματιστικές ικανότητες των ομάδων ανάπτυξής της. Μια μηχανής γραφικών απαιτεί εμπειρία και χρόνο για να υλοποιηθεί, που σημαίνει ότι δεν είναι μία προσιτή ιδέα για μία μικρή ομάδα που δεν έχει την ικανότητα ανάπτυξης μιας τέτοιας τεχνολογίας. Παρόλα αυτά λογισμικά όπως η Unreal Engine και η Unity λύνουν αυτό το πρόβλημα από την στιγμή που υπάρχει δωρεάν πρόσβαση σε αυτές.

Έχοντας επιλέξει το αρμόδιο λογισμικό ο κύκλος ανάπτυξης περνάει το επόμενο στάδιο που είναι η υλοποίηση. Με διαφορά το σημαντικότερο κομμάτι για την ανάπτυξη ενός παιχνιδιού που πρέπει να γίνει στενή συνεργασία από όλες τις ομάδες με πλήρη κατανόηση του στόχου για την βελτιστοποίηση του χρόνου παραγωγής και την ελαχιστοποίηση σφαλμάτων και προβλημάτων που μπορεί να υποστεί το παιχνίδι. Αυτό απαιτεί ένα οργανωμένο περιβάλλον που επιτρέπει μια τέτοια συνεργασία, δίνοντας την αμεσότητα που απαιτούν οι προγραμματιστές και άλλες ομάδες όπως οι 3d Artists, animators και ηχολήπτες, παρέχοντας πρόσβαση σε πραγματικό χρόνο στο παιχνίδι για προσθήκη, επεξεργασία και πιθανόν αλλαγές που θα υποστεί κατά την ανάπτυξη. Ένα τέτοιο περιβάλλον βρίσκεται στην Unreal Engine 4 που επιτρέπει μία ομαλή εισαγωγή στον κόσμο των μηχανών γραφικών με την τεχνολογία του Blueprint. Το πρόβλημα που προκύπτει αρκετές φορές σε άτομα που θέλουν να περάσουν στον κλάδο της ανάπτυξης ηλεκτρονικών παιχνιδιών είναι η εμπειρία που απαιτείται από πλευράς προγραμματισμού. Μέσω της τεχνολογίας του Blueprint αντικαθίστανται κομμάτια κώδικα με κόμβους που λειτουργούν αλληλένδετα μεταξύ τους. Αυτό το σύστημα είναι αρκετά ευέλικτο καθώς παρέχει τη δυνατότητα στους σχεδιαστές να χρησιμοποιούν σχεδόν όλο το φάσμα των εργαλείων που ήταν διαθέσιμο μόνο στους προγραμματιστές και τους επιτρέπουν να αναπτύξουν το πλήθος των λειτουργιών του παιχνιδιού, χωρίς την απαίτηση γνώσης προγραμματισμού. Επίσης μέσω C++ οι προγραμματιστές μπορούν να

δημιουργήσουν βασικά συστήματα που μπορούν να επεκταθούν από τους σχεδιαστές μέσω του Blueprint.

Η Unreal παρέχει μία πληθώρα διαφόρων ειδών Blueprint με τρία από αυτά να είναι η βάση για την ανάπτυξη των λειτουργιών ενός παιχνιδιού. Το πρώτο και πιο σημαντικό είναι το Class Blueprint που επιτρέπει στους δημιουργούς περιεχομένου οι σχεδιαστές να προσθέσουν εύκολα λειτουργίες πάνω από τις υπάρχουσες κλάσεις παιχνιδιού. Με αυτόν τον τρόπο, γίνεται να οριστεί μία καινούρια κλάση ή ένα νέου είδους δράστη, που μετέπειτα μπορεί να τοποθετηθεί στο χάρτη ή στο level του παιχνιδιού. Αρκετά δυνατό εργαλείο του Class Blueprint είναι η κληρονομικότητα που μπορεί να περάσει πληροφορία από τον γονέα στο παιδί. Πριν δοθεί αυτή η ικανότητα πρέπει πρώτα να αποφασιστεί το είδος του γονέα ή ο τύπος κλάσης. Τα βασικότερα είδη κλάσης είναι ο δράστης, που είναι ένα αντικείμενο που μπορεί να τοποθετηθεί στον κόσμο, το πiónι που μπορεί να ελεγχθεί από έναν χειριστή, ένας χαρακτήρας με δυνατότητες που μπορεί να προσθέσει ο σχεδιαστής του παιχνιδιού όπως τρέξιμο και περπάτημα και ο χειριστής που είναι υπεύθυνος για τον χειρισμό του πιονιού. Αν για παράδειγμα υπάρχει ένα τοπίο όπως ένα δάσος και χρειαστεί να κατοικηθεί από διάφορα ζώα, τότε ο σχεδιαστής δεν θα χρειαστεί να φτιάξει εξ ολοκλήρου το κάθε είδος ζώου. Έχοντας σχεδιάσει ένα αρχικό πiónι με χαρακτηριστικά όπως το σκελετό και την κίνηση, θα μπορέσει να φτιάξει αντίγραφα που κληρονομούν αυτά τα χαρακτηριστικά και μετέπειτα απλά θα ορίσει το κάθε ζώο με τον κατάλληλο σκελετό και την κίνηση που του αρμόζει.

Εξίσου σημαντικά είναι τα Blueprint Interfaces, που αποθηκεύουν λειτουργίες που μετέπειτα μπορούν να χρησιμοποιηθούν από άλλα Blueprints. Αυτό ισχύει για τα Blueprints που τους έχει προστεθεί η συγκεκριμένη διασύνδεση, επιτρέποντας την χρήση των λειτουργιών του Blueprint Interface. Αυτό ουσιαστικά μοιάζει με την έννοια μιας διεπαφής στο γενικό προγραμματισμό, η οποία επιτρέπει την κοινή χρήση πολλών διαφορετικών τύπων αντικειμένων και την πρόσβαση τους μέσω μιας κοινής διεπαφής. Με την χρήση των Blueprint Interfaces επιτρέπεται να χρησιμοποιηθεί μία κοινή μέθοδο αλληλεπίδρασης σε πολλούς διαφορετικούς τύπους αντικειμένων. Αυτό σημαίνει, ότι είναι δυνατό να υπάρχουν εντελώς διαφορετικοί τύποι αντικειμένων, όπως ένα φυτό και ένα όπλο, που μοιράζονται μία συγκεκριμένη λειτουργία όπως και τα δύο μπορεί να τα μαζέψει ο παίκτης από το έδαφος. Δημιουργώντας μια διεπαφή Blueprint που περιέχει μια συνάρτηση Interact και έχοντας τόσο το όπλο όσο και το φυτό εφαρμόσει αυτήν τη διασύνδεση (Blueprint Interface), μπορούν και τα δύο να καλέσουν την συγκεκριμένη λειτουργία. Παρόλο που σε ένα Blueprint Interface μπορούν να δοθούν είσοδοι και έξοδοι υπάρχουν μερικοί περιορισμοί. Μία λειτουργία του Interface δεν μπορεί να προσθέσεις νέες μεταβλητές, στο τρέχον Blueprint ή να τροποποιήσει γραφήματα που συνδέονται με αυτό.

Με κάθε νέο επίπεδο που ο σχεδιαστής ή ο προγραμματιστής που δημιουργεί ταυτόχρονα η μηχανή γραφικών θα ορίσει και ένα Blueprint επιπέδου που θα λειτουργεί σαν ένα "παγκόσμιο" γράφημα συμβάντων που θα είναι υπεύθυνο για την

εκτέλεση ακολουθιών από ενέργειες. Παρόλα αυτά, δεν μπορούν να δημιουργηθούν νέα Level Blueprint μέσω της διεπαφής της μηχανής γραφικών. Αυτή η μοναδικότητα ισχύει για κάθε επίπεδο που θα φτιαχτεί δίνοντας την επιλογή να εφαρμοστούν διαφορετικό είδους λειτουργίες για το καθένα με την κάθε ιδιότητα του επιπέδου να μην επηρεάζει τα άλλα. Μέσω των Level Blueprint μπορούν να τεθούν λειτουργίες που θα επηρεάζουν των παίκτη αλλά τους υπόλοιπους δράστες όπως αλλαγές στο περιβάλλον αλλά και γεγονότα που θα έχουν μόνο έναν και μοναδικό σκοπό όπως θάνατος του χαρακτήρα ή αλλαγή ώρας και κύκλος μέρας-νύχτας.

Κεφάλαιο 2 Φωτισμός

2.1 Κοινά Χαρακτηριστικά Των Ειδών Φωτισμού

Ανάλογα με τον στόχο ενός παιχνιδιού, το θεματικό του και την ατμόσφαιρα που προσπαθεί να ενσαρκώσει λαμβάνονται υπόψη αρκετές αποφάσεις για το στήσιμο βασικών παραμέτρων όπως ο φωτισμός και η ατμόσφαιρα. Παρόλο που αρκετά παιχνίδια αποφεύγουν την πλήρη ρεαλιστικότητα στο περιβάλλον κρατώντας ένα παλαιακό στυλ, που φέρει πιο κοντά την κυριολεκτική έννοια του όρου “παιχνίδι”, κυρίως για δημιουργικούς λόγους, δεν παρατηρείται η ανάγκη για ένα αξιοπρεπή και καλοσχεδιασμένο στήσιμο φωτισμού. Άσχετα με την λεπτομερειακή υλοποίηση που μπορεί να έχει λάβει χώρα σε ένα επίπεδο, από μεριάς κατοίκησης και εμπλουτισμού του περιβάλλοντος από αντικείμενα και δράστες, θα πρέπει να είναι διακριτό το αποτέλεσμα, αλλά και ο λόγος, που ο σχεδιαστής πήρε τις ανάλογες δημιουργικές αποφάσεις.

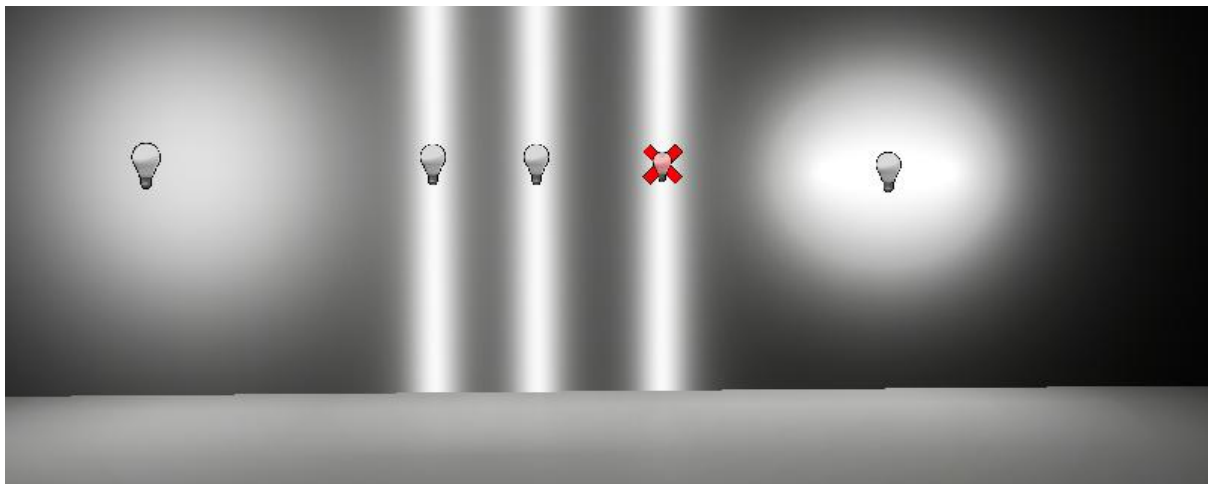
Ο φωτισμός στην Unreal χωρίζεται σε τέσσερις τύπους, Directional light, Sky light, Point light και Spot light. Κάθε τύπου φωτισμός, έχει σχεδόν ίδιες ρυθμίσεις αλλά με τον καθένα να έχει τελείως διαφορετικές ιδιότητες, με αποτέλεσμα συγκεκριμένου είδους φωτισμοί να είναι περισσότερο επιθυμητοί σε συγκεκριμένες περιπτώσεις από κάποιους άλλους. Καλή πρακτική είναι να γίνει μελέτη των χαρακτηριστικών που είναι κοινά ανάμεσα τους και η διαχείρισή τους ανάλογα με τις δυνατότητές τους. Ένα από τα κοινά και πιο σημαντικά χαρακτηριστικά που έχουν όλοι οι φωτισμοί είναι η κινητικότητα που μπορεί να τεθεί σε τρεις τροπολογίες

- στατική
- ακίνητη
- κινητή

Τα στατικά φώτα δεν έχουν την δυνατότητα αλλαγής ή μετακίνησης από την στιγμή που το παιχνίδι έχει τεθεί σε εκτέλεση. Ο υπολογισμός του φωτισμού γίνεται μόνο μέσω των Lightmaps, και μόλις υποβληθεί σε επεξεργασία, δεν θα έχει περαιτέρω επιπτώσεις στην απόδοση. Σαν αποτέλεσμα αν ένα κινούμενο αντικείμενο έρθει σε επαφή με μία περιοχή που είναι προκατασκευασμένη με στατικό φωτισμό, τότε το αντικείμενο δεν θα επηρεάσει τον φωτισμό, επομένως η χρησιμότητα του στατικού

φωτισμού περιορίζεται στα σημεία που δεν περνάνε κινούμενα αντικείμενα. Παράλληλα λόγω της αμεσότητας που οι σκιάς έχουν με την ανάλυση του Lightmap του κάθε αντικειμένου θα πρέπει να γίνουν και οι απαραίτητες ρυθμίσεις, συνήθως με την αύξηση της Lightmap Resolution μεταβλητής στο κάθε αντικείμενο. Μεγαλύτερες τιμές, σημαίνει υψηλότερη ανάλυση, αλλά και μεγαλύτερους χρόνους κατασκευής ταυτόχρονα με μεγαλύτερη κατανάλωση μνήμης. Από τις τρεις τροπολογίες, τα στατικά φώτα τείνουν να έχουν μέση ποιότητα, χαμηλότερη μεταβλητότητα και το χαμηλότερο κόστος απόδοσης

Σε αντίθεση με τα στατικά φώτα τα ακίνητα έχουν την δυνατότητα να αλλάξουν τα χαρακτηριστικά τους, όπως η ένταση του φωτισμού και το χρώμα αλλά παραμένουν σε μία σταθερή θέση κατά την εκτέλεση του παιχνιδιού. Ωστόσο οι αλλαγές που μπορεί να υποστεί το ακίνητο φως κατά την εκτέλεση του παιχνιδιού επηρεάζουν μόνο τον άμεσο φωτισμό αλλά όχι τον έμμεσο, όπως οι αντανάκλασεις, από την στιγμή που είναι υπολογισμένο από το Lightmass του κάθε ενός αντικειμένου στο επίπεδο. Από την στιγμή που ο άμεσος φωτισμός είναι δυναμικά τοποθετημένος στην σκηνή μεταβλητές όπως η ορατότητα και η ισχύς μπορούν να τροποποιηθούν ανάλογα με τον τρόπο που το ίδιο το φως θα χρησιμοποιηθεί στο παιχνίδι, π.χ. φακός που θα ελέγχεται από τον παίκτη ή μία λάμπα που θα τρεμοπαίζει. Αντίθετα ο έμμεσος φωτισμός από την στιγμή που είναι προκατασκευασμένος από το παιχνίδι δεν θα υποστεί τις ίδιες αλλαγές προκαλώντας αρκετά προβλήματα στο τριγύρω περιβάλλον με ανεπιθύμητα αποτελέσματα. Για την καταπολέμηση του έμμεσου φωτισμού σε μη επιθυμητά σημεία στον χάρτη γίνεται μέσω του post process volume με την επιλογή της ιδιότητας που ονομάζεται Indirect Lighting Intensity, η οποία επιτρέπει την κλιμάκωση της συμβολής του lightmap για όλα τα φώτα που είναι τοποθετημένα στο επίπεδο, το οποίο μπορεί να αλλαχθεί κατά το χρόνο εκτέλεσης μέσω Blueprint. Επίσης ένα πρόβλημα που μπορεί να δημιουργηθεί με την χρήση πολλαπλών ακινητών φωτισμών είναι ο αριθμός που μπορεί να υπάρχει, αν επικαλύπτει το ένα το άλλο. Το μεγαλύτερο πλήθος από επικαλυπτόμενα στατικά φώτα που μπορούν να υπάρχουν στην σκηνή είναι 4, επειδή τα φώτα πρέπει να εκχωρηθούν σε διαφορετικά κανάλια lightmap. Από το τέταρτο stationary φως και μετά, αντί να δημιουργηθούν στατικοί φωτισμοί θα μετατραπούν σε δυναμικούς, που μπορεί να έχει επίπτωση στην ταχύτητα εκτέλεσης του παιχνιδιού. Η unreal engine σε περίπτωση που ο αριθμός των επικαλυπτόμενων στατικών φωτισμών υπερβεί το όριο, προσθέτει ένα κόκκινο Χ στον δρόστη για να ενημερώσει τον σχεδιαστή.



Εικόνα 1.1 Επικαλυπτόμενα Static Lights

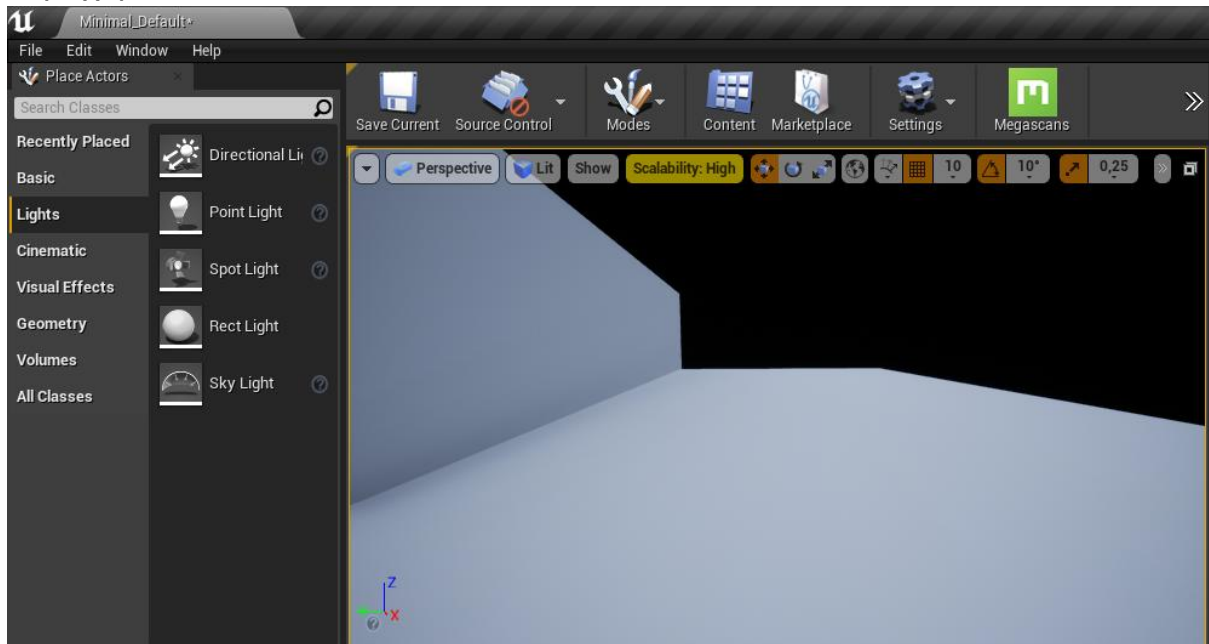
Η τελευταία επιλογή που δίνεται στον σχεδιαστή για τον τρόπο που θα κινείται το κάθε φως, στο παιχνίδι, είναι η κινητή. Το κινητό φως μπορεί να αλλάξει σε πραγματικό χρόνο σχεδόν κάθε ιδιότητα που έχει. Η χρήση του γίνεται συνήθως όταν απαιτείται άμεση αλλαγή φωτισμού, ενώ έχει στηθεί το παιχνίδι. Αυτό έχει σαν αποτέλεσμα κάθε αντικείμενο που έρχεται σε επαφή με αυτού του είδους φωτισμό να προκαλεί σκιές, που πρέπει να υπολογιστούν την ίδια χρονική στιγμή. Αυτή η διαδικασία επιβαρύνεται ανάλογα και με το πλήθος των τριγώνων που έχει το κάθε αντικείμενο. Για να είναι λιγότερο κοστοβόρα η διαδικασία κινητού φωτισμού ο σχεδιαστής μπορεί να επιλέξει σημεία που ο παίκτης δεν θα έρθει σε επαφή, ή θα αντικρίζει από μακριά ώστε να αντικαταστήσει τον κινητό με στατικό φωτισμό.

Δύο χρήσιμες ρυθμίσεις που βοηθούν στην ομαλή προβολή των σκιών είναι το Slope Bias και το Shadow Slope Bias, που σε συνεργασία αφαιρούν αντικείμενα χαρτογράφησης σκιάς (artifacts) που μπορεί να προκύψουν. Η επιλογή shadow amount, ρυθμίζει την ένταση της σκιάς και παίρνει τιμές από 0.0-1.0. Αυτή η επιλογή δεν συμβάλλει στην πιο γρήγορη επεξεργασία του φωτισμού αν η τιμή βρίσκεται στο 0 και κατά συνέπεια δεν επηρεάζει την εκτέλεση του παιχνιδιού με θετικό ή αρνητικό τρόπο, από την στιγμή που ο υπολογισμός για την σκίαση του κάθε αντικειμένου θα συνεχίσει να υπάρχει. Ακόμα δύο βασικές ιδιότητες είναι, η ένταση και το χρώμα που ανάλογα με το είδος φωτισμού που έχει επιλεγεί προσαρμόζεται ανάλογα από τον σχεδιαστή, μέσω του παραθύρου λεπτομερειών όπως και οι προηγουμένως αναφερόμενες ιδιότητες αλλά και μέσω Blueprint.

2.2 Είδοι Φωτισμού και υπολογισμός ακτίνων

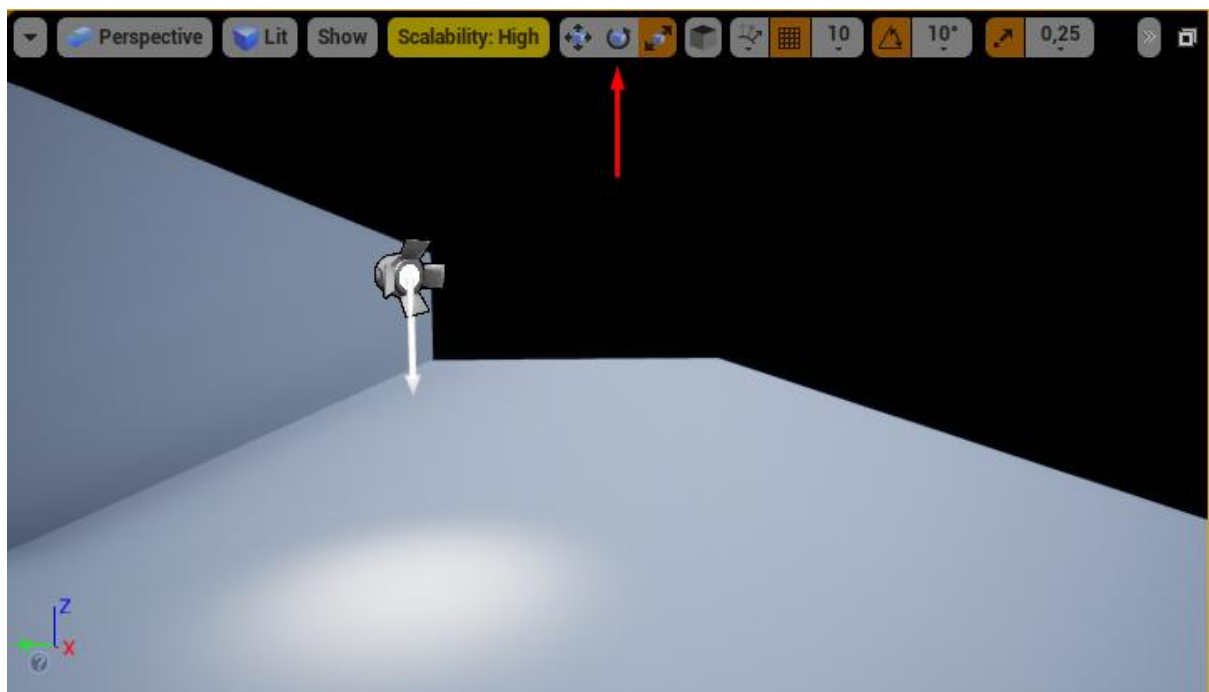
Στην Unreal Engine 4 μπορεί να υπάρξει μία κατηγοριοποίηση όσον αφορά τα είδη φωτισμών και των λειτουργιών τους. Το spot light και point light χρησιμοποιούνται κυρίως για τον φωτισμό σε μικρού μήκους αποστάσεις όπως λάμπες ή φακοί που προβάλλουν φως από ένα μόνο σημείο. Αντίθετα ο σκοπός του Directional light και

Sky light είναι η απεικόνιση του φωτισμού του ήλιου καλύπτοντας όλο το περιβάλλον με φως σε άπειρη απόσταση. Το Sky light εφαρμόζει φωτισμό με βάση το φόντο και περιεχόμενο του κάθε επιπέδου.



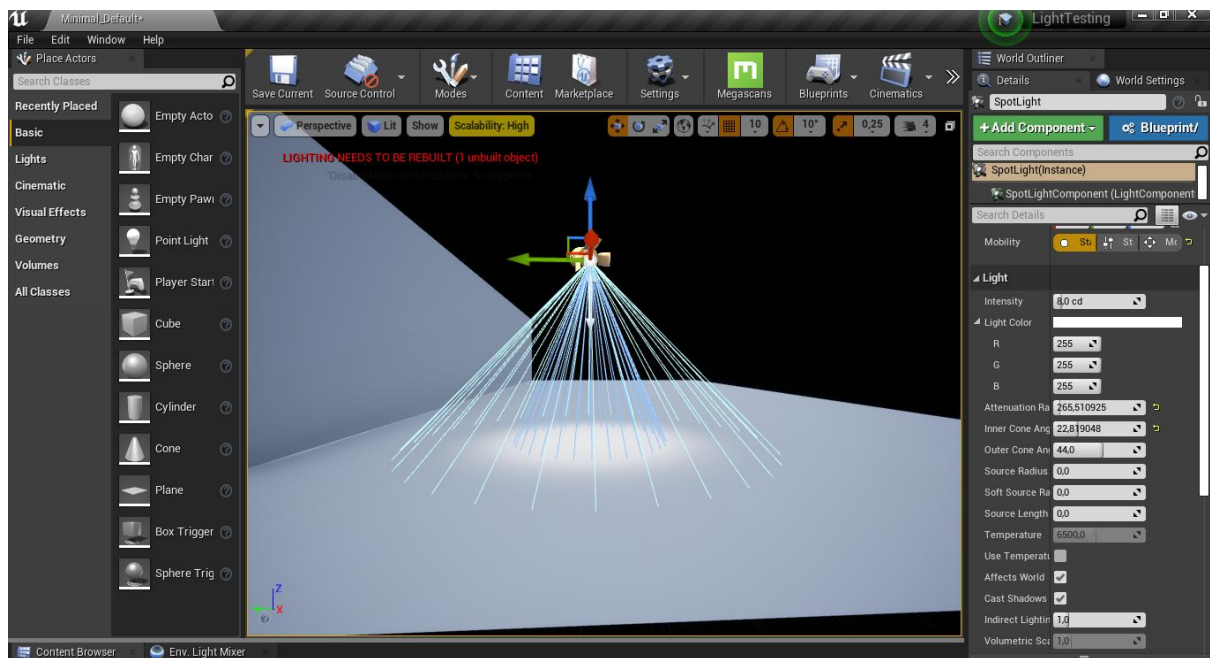
Εικόνα 2.1 Place Actors:Lights

Spot Light : Το Spot light έχει το εικονίδιο ενός προβολέα. Αν τοποθετηθεί στο επίπεδο εμφανίζεται ο δράστης (actor) του Spot light που δηλώνει την θέση του στον κόσμο. Αφού επιλεγθεί, ο δράστης, στο πάνω κεντρικό μέρος του viewport δίνονται επιλογές για την μεταφορά, περιστροφή και αλλαγή σκέλους του φωτισμού και πατώντας τα πλήκτρα W, R, E αντίστοιχα μπορεί να γίνει αλλαγή ανάμεσα σε αυτές τις επιλογές.



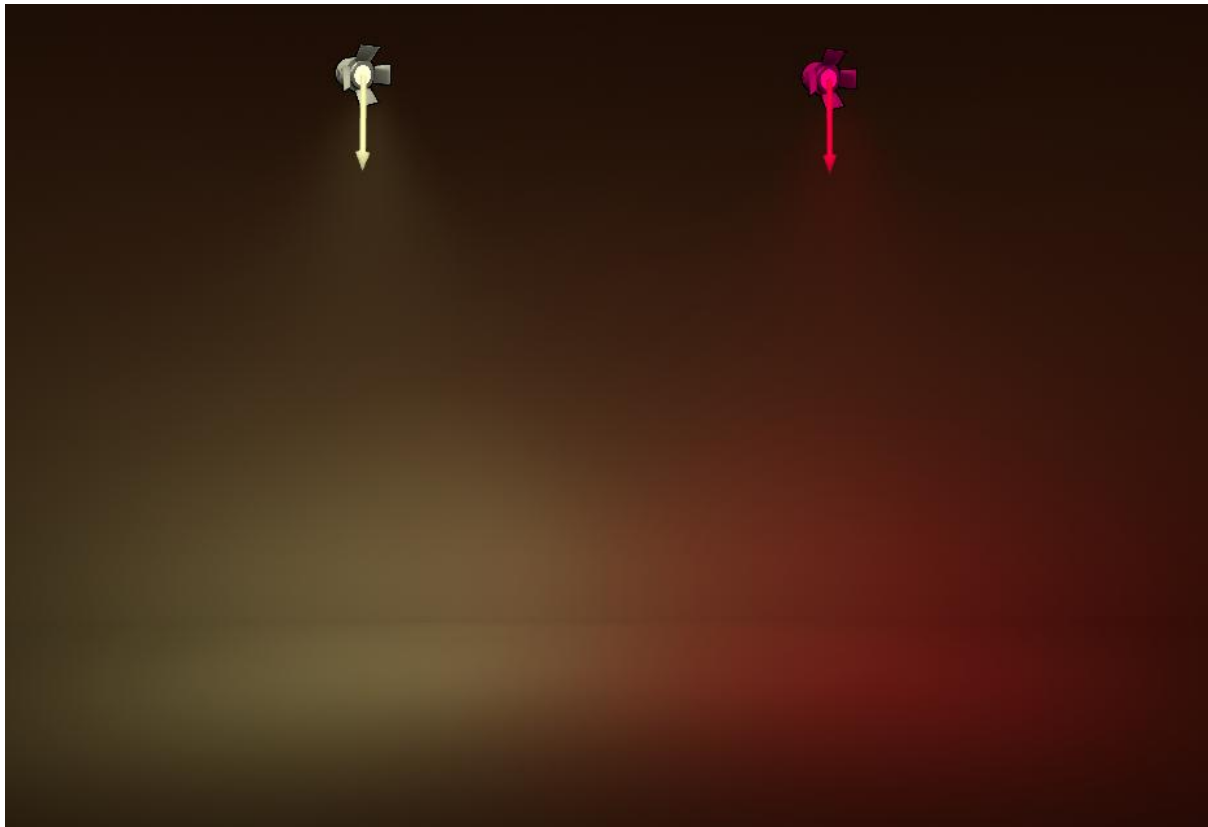
Εικόνα 2.2 Move,Rotate και Scale Options

Ενώ είναι επιλεγμένο το Spot Light, στην δεξιά μεριά της οθόνης υπάρχει το παράθυρο λεπτομερειών που επιτρέπει στον σχεδιαστή να επεξεργαστεί και να τροποποιήσει τις ιδιότητες του φωτισμού. Σε περίπτωση που το παράθυρο λεπτομερειών δεν είναι αυτόματα ανοιχτό, μέσω του μενού πατώντας το Window και επιλέγοντας την δεύτερη επιλογή Details μπορεί να εμφανιστεί το αντίστοιχο παράθυρο. Οι πρώτες τρεις ιδιότητες είναι ίδιες με τις επιλογές που είναι και στο Viewport, σχετικά με την μεταφορά, περιστροφή και αλλαγή σκέλους. Η τέταρτη ιδιότητα, είναι η κινητικότητα του αντικειμένου που αναφέρθηκε προηγουμένως. Η μεγαλύτερη βάση δίνεται στις ιδιότητες που είναι ομαδοποιημένες, κάτω από την λεζάντα Light. Εκεί φυλάσσονται οι πληροφορίες σχετικά με την ένταση, το χρώμα αλλά και τις αποκλειστικές ιδιότητες του φωτισμού. Στην περίπτωση του Spot light αυτές οι αποκλειστικές ιδιότητες, έρχονται να μεταβάλλουν τα χαρακτηριστικά του κώνου που είναι υπεύθυνος για την προβολή του φωτισμού.



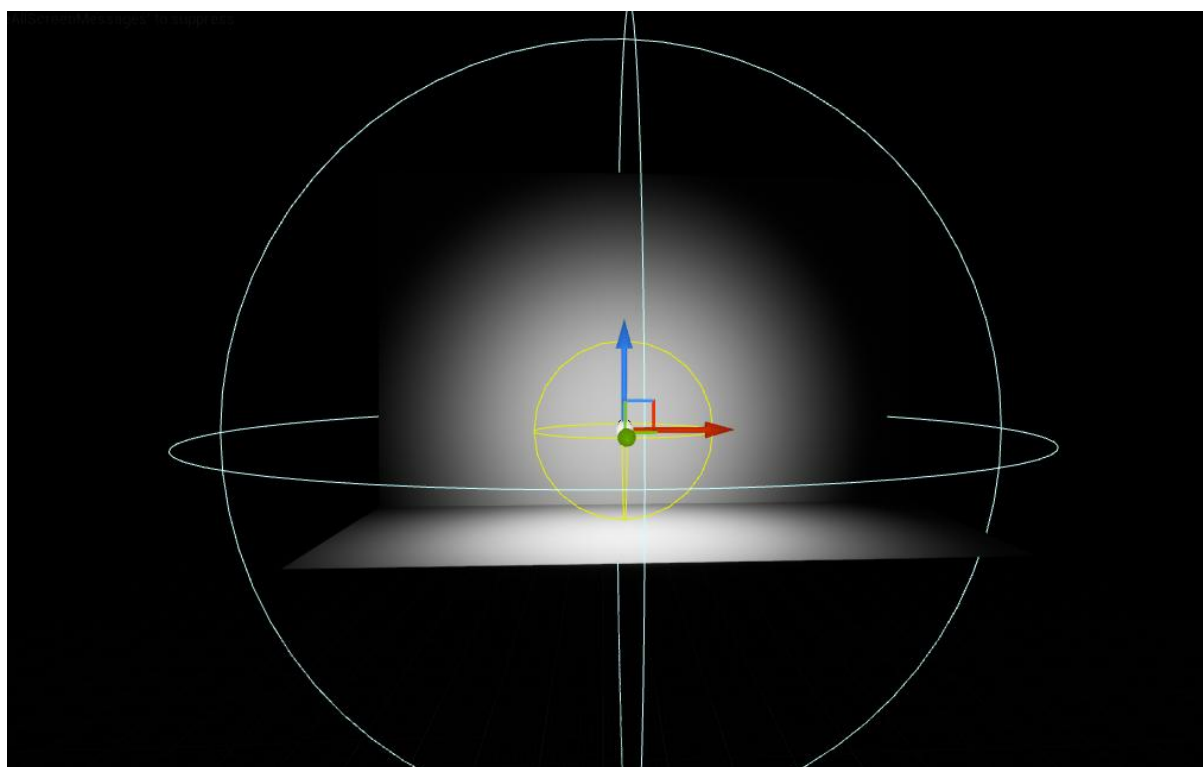
Εικόνα 2.3 Details Window

Μετά το χρώμα του φωτισμού η αμέσως επόμενη ιδιότητα είναι το Attenuation Radius που αυξάνει ή μειώνει την απόσταση από την πηγή ως το σημείο πλήρης εξασθένισης του φωτισμού. Το Inner Cone Angle και το Outer Cone Angle είναι υπεύθυνα για τις διαστάσεις των κώνων, που εμφανίζονται με μπλε και γαλάζιες γραμμές αντίστοιχα. Σε συνεργασία μπορούν να ορίσουν την διάχυση που θα έχει το φως στις επιφάνειες που προσπίπτει. Αν το φως χρειάζεται να έχει μεγάλη διάχυση τότε το Inner Cone Angle θα πρέπει να μειωθεί ή να αυξηθεί η διάμετρος του Outer Cone Angle. Όσο μικρότερη η διαφορά μεταξύ τους τόσο πιο έντονη θα είναι η διαφορά μεταξύ φωτιζόμενης και μη φωτιζόμενης επιφάνειας. Οι τελευταίες δύο επιλογές της ομάδας Light, Indirect Light Intensity και Volumetric Scattering Intensity, έχουν τεράστια επιρροή στο ρεαλισμό του φωτισμού. Το Indirect Light Intensity επηρεάζει τις τριγύρω επιφάνειες προσθέτοντας έμμεσο φωτισμό. Αν υπάρχουν και άλλες πηγές φωτισμού γίνεται εξομάλυνση στα σημεία, που τα φώτα εφάπτονται και δεν υπάρχει απότομη αλλαγή της έντασης φωτισμού. Η επιλογή Volumetric Scattering Intensity επιτρέπει τον φωτισμισμό των volume που διαπερνά το φως. Αν στο επίπεδο υπάρχει ομίχλη μπορεί να χρησιμοποιηθεί το Volumetric Scattering Intensity για να αναπαραστήσει τη διάθλαση του φωτισμού που διαπερνά την ομίχλη.

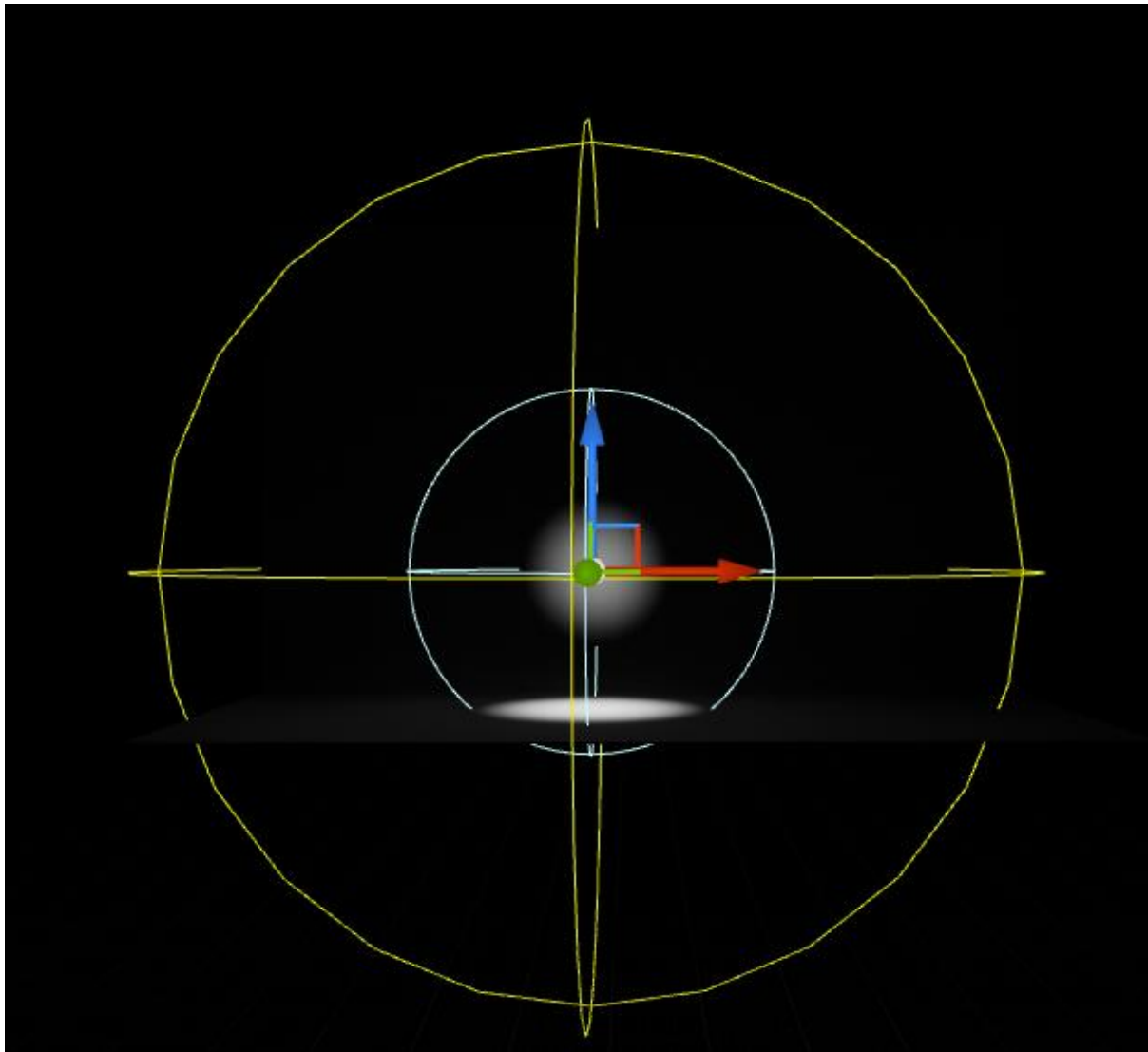


Εικόνα 2.4 Spot Light

Point Light: Το Point Light έχει αρκετά κοινά χαρακτηριστικά με το Spot Light με τις διαφορές τους να είναι κυρίως στον τρόπο προβολής φωτισμού. Με το Point Light δεν υπάρχει η έννοια των κώνων αλλά της σφαίρας. Η επιλογή της έντασης και του χρώματος μένουν ίδιες. Κάτω από το Attenuation Radius υπάρχει η ιδιότητα Source Radius, που αυξάνει την διάμετρο που ο φωτισμός θα επηρεάσει τις τριγύρω επιφάνειες. Αυτή η επιλογή είναι που αντικαθιστά το Inner Cone Angle και το Outer Cone Angle. Το Source Radius ορίζει την διάμετρο της πηγής, αλλά δεν επηρεάζει την ένταση και την απόσταση του φωτισμού. Άρα αν αυξηθεί το Source Radius και η διάμετρός του είναι μεγαλύτερη από το Attenuation Radius, δεν θα υπάρξει αύξηση του μεγέθους του φωτισμού.

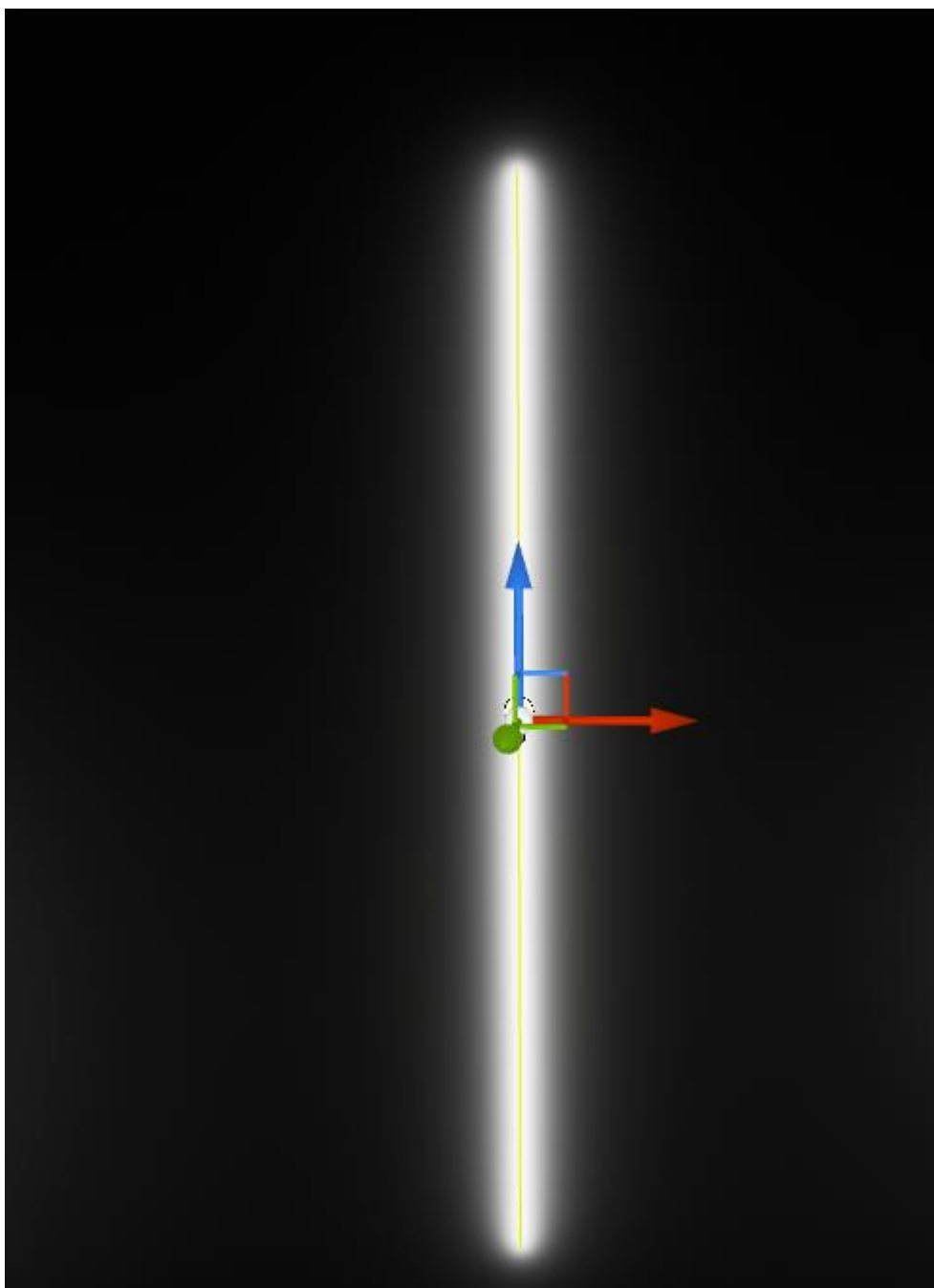


Εικόνα 2.5 Attenuation Radius > Source Radius



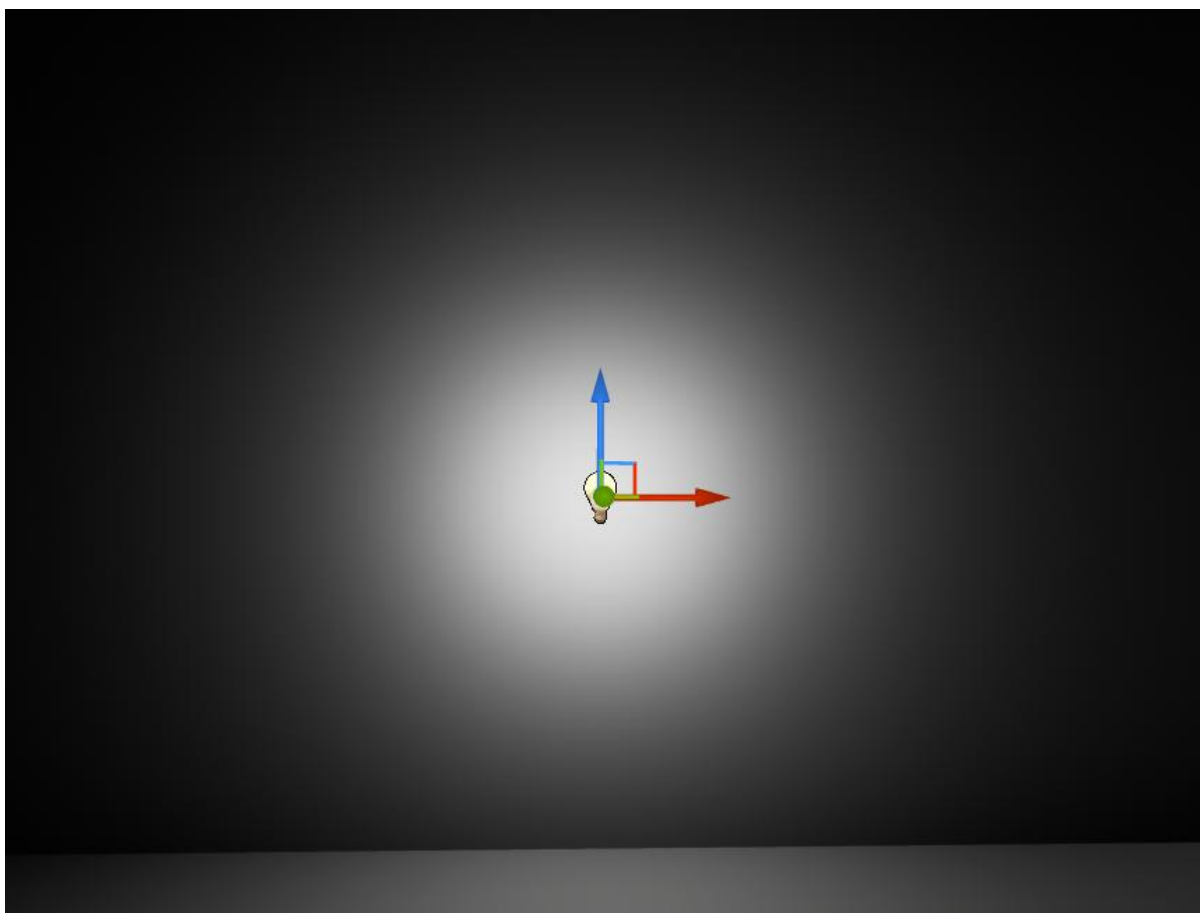
Εικόνα 2.6 Attenuation Radius < Source Radius

Η δεύτερη ιδιότητα που επίσης επηρεάζει την πηγή του φωτισμού είναι το Source Length που μεταβάλλει το ύψος ή το πλάτος της πηγής φωτισμού ανάλογα με την φορά που του έχει ανατεθεί στο επίπεδο. Με τον συνδυασμό του Source Length και Source Radius μπορεί να γίνει προσαρμογή ανάλογα με το τι είδους πηγής είναι απαραίτητη για την συγκεκριμένη σκηνή. Για παράδειγμα, αν χρειάζεται να απεικονιστεί κάποιο είδους led, που συνήθως έχει μικρή σε όγκο πηγή αλλά μακρόστενη διάμετρο, θα πρέπει να μειωθεί το Source Radius και να αυξηθεί το Source Length.



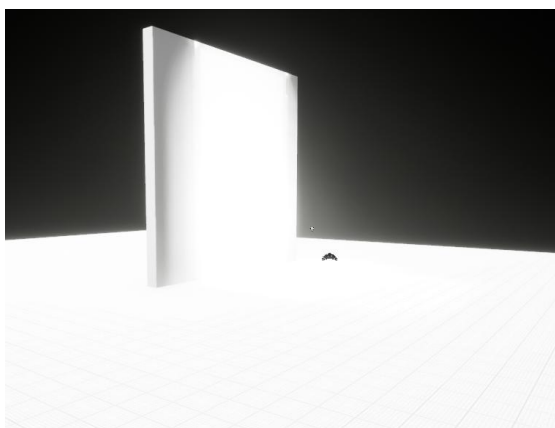
Εικόνα 2.7 Source Radius

Παρόλο που το Source Radius στη συγκεκριμένο παράδειγμα είναι ίσο με το μηδέν ($\text{Source Radius} = 0,0$) συνεχίζει και υπάρχει φως. Αν το Source Radius και Source Length είναι μηδενικά δεν σημαίνει ότι δεν υπάρχει πηγή αλλά ότι το φως θα αρχίσει να προβάλλεται από το κεντρικό σημείο του δράστη.

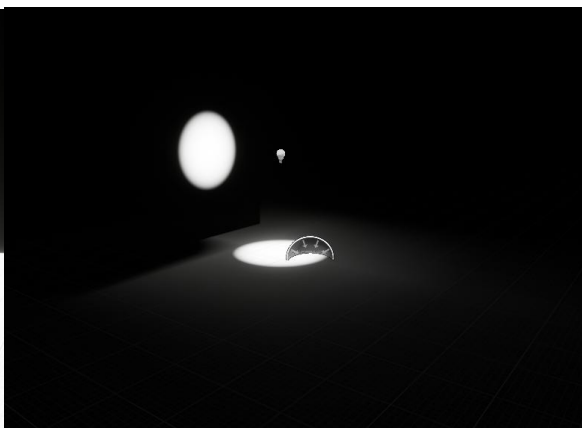


Εικόνα 2.8 Source Radius & Source Length = 0.0

Sky Light: Το Sky Light λαμβάνει πληροφορίες από αντανακλώμενο φωτισμό από το τριγύρω περιβάλλον, όπως από άλλους φωτισμούς που μπορεί να υπάρχουν, π.χ. ένα point light ή directional light αλλά και από την ατμόσφαιρα και τις μετατρέπει σε φως. Αυτή η ιδιότητα μπορεί να ρυθμιστεί μέσω του Source type. Όταν προστεθεί το Sky Light στο επίπεδο λόγω της προρυθμισμένης τιμής που έχει το Sky Distance Threshold (150000,0) μπορεί να μην παρατηρηθεί καμία αλλαγή στον φωτισμό. Το Sky Distance Threshold συμβολίζει το σημείο από το οποίο το Sky Light θα θεωρεί σαν ουρανό και θα το προβάλλει σαν φως. Αν το Sky Distance Threshold έχει πολύ μεγάλη τιμή και δεν υπάρχει σε εκείνο το σημείο πληροφορία, τότε δεν θα υπάρξει φως, που σημαίνει ότι η τιμή θα πρέπει να μειωθεί. Ταυτόχρονα αν υπάρχουν πολλές πηγές φωτισμού με πολλές αντανакλάσεις, η ένταση του φωτισμού θα αυξηθεί δραματικά. Σε αντίθεση με τα άλλα φώτα που αναφέρθηκαν δεν υπάρχει κάποια ένδειξη για να ενημερώνει τον σχεδιαστή για την απόσταση που βρίσκεται το Sky Distance Thresholdm παρα μόνο με αριθμούς, οπότε θα πρέπει ο καθένας να πειραματιστεί την κατάλληλη τιμή που θα χρειαστεί, για να υπάρξει ένα επιθυμητό αποτέλεσμα.

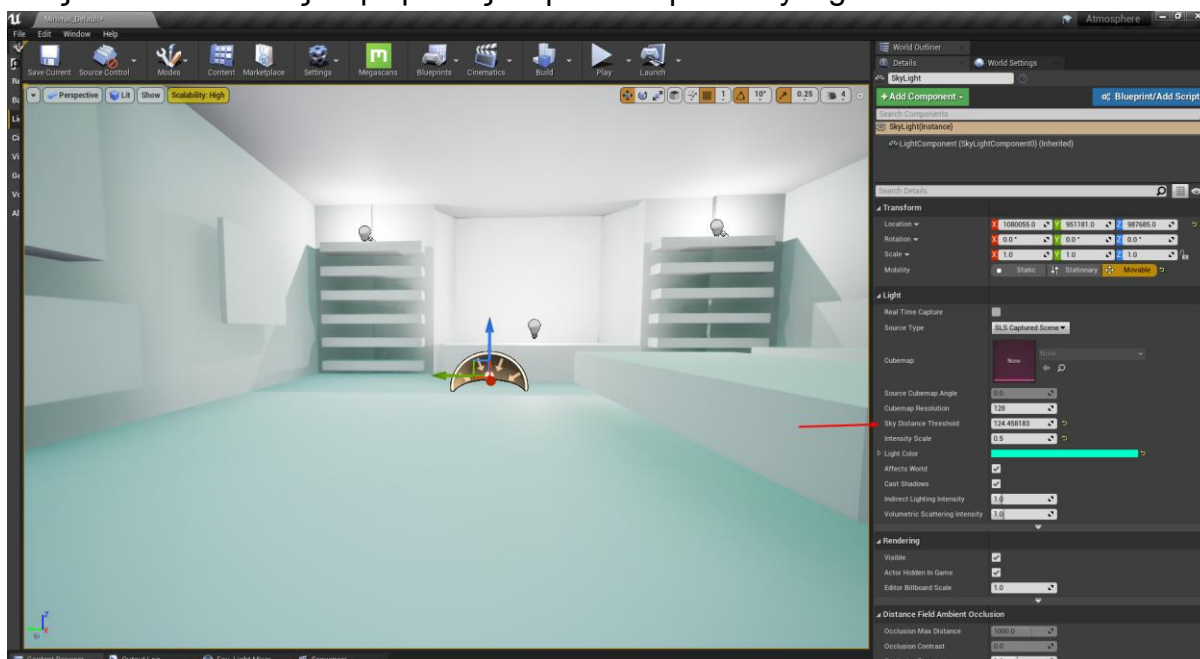


Εικόνα 2.9 Sky Distance Threshold 70



Εικόνα 2.10 Sky Distance Default Threshold 15000

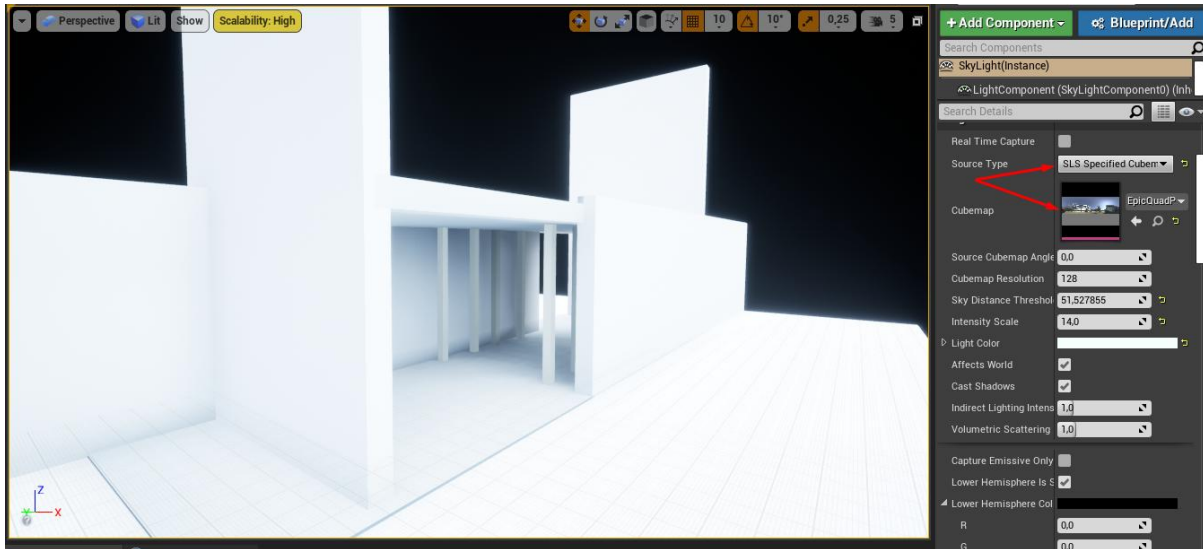
Στην πρώτη εικόνα η πληροφορία φωτισμού για το Sky Light έρχεται από το point light αλλά στην δεύτερη εικόνα το Sky Distance Threshold είναι αρκετά μεγαλύτερο από την απόσταση Sky Light με point light και δεν υπάρχει φωτισμός. Αρά άσχετα με την ονομασία του το Sky Light μπορεί λειτουργήσει και σαν φωτισμός εσωτερικών χώρων, ρυθμίζοντας την ιδιότητα Sky Distance Threshold. Το γεγονός ότι το Sky Light κάνει χρήση των φωτισμών, που το περιβάλλουν, σημαίνει ότι η ένταση τους θα είναι και ένας παράγοντας στην ένταση του Sky Light.



Εικόνα 2.11 Πολλαπλές πηγές φωτισμού

Η δεύτερη επιλογή του Source type είναι το SLS Specified Cubemap που χρησιμοποιεί Cubemaps για πηγή φωτισμού. Τα Cubemaps είναι μια μέθοδος χαρτογράφησης περιβάλλοντος, στην οποία απομακρυσμένα τοπία χαρτογραφούνται σε μια πανοραμική υφή. Το Sky Light λαμβάνει πληροφορία από τον φωτισμό του Cubemap και δημιουργεί το αντίστοιχο φως. Από την στιγμή που το Cubemap έχει

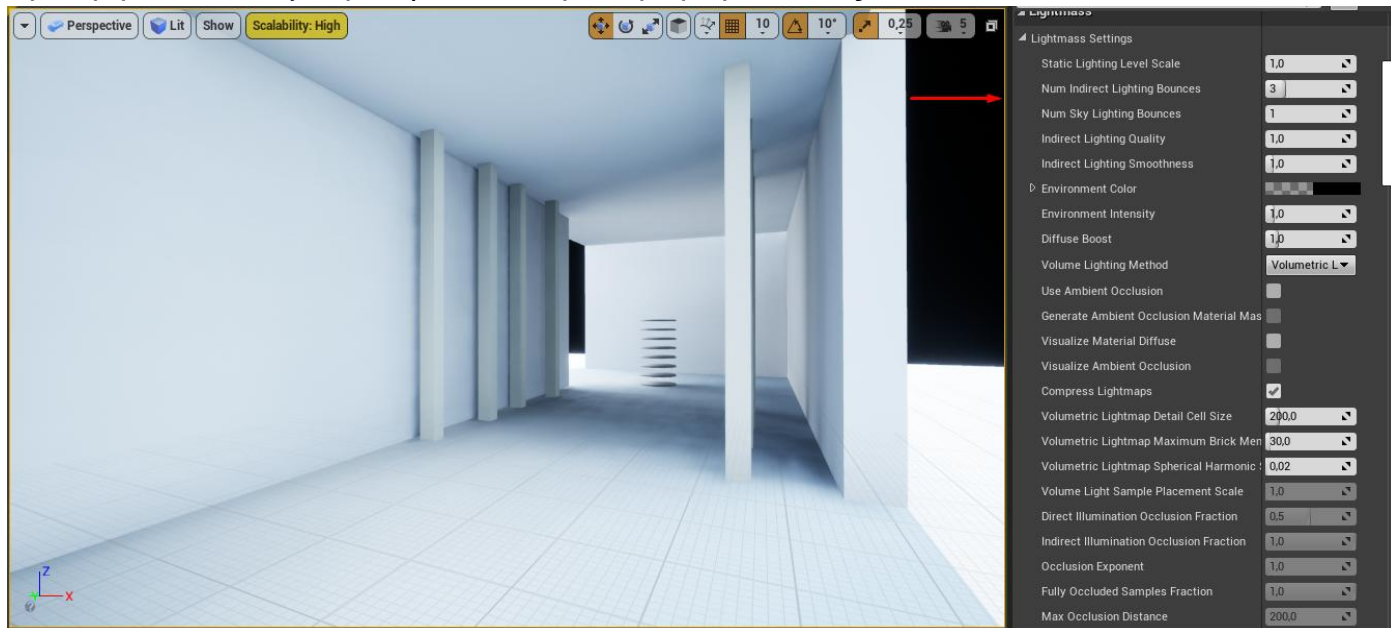
τεθεί σαν την πηγή πληροφορίας για το φωτισμό, η ιδιότητα Sky Distance Threshold δεν έχει καμία επιρροή. Η επιλογή Cubemap Resolution, παρέχει μεγαλύτερη και πιο ακριβής πληροφορία φωτισμού και χρώματος ανάλογα με την τιμή που έχει τεθεί. Αυτή η ιδιότητα είναι ενεργή είτε υπάρχει Cubemap είτε χρησιμοποιείται η επιλογή SLS Captured Scene από την στιγμή που το τριγύρω περιβάλλον, θεωρείται σαν το επιλεγόμενο Cubemap. Πρέπει να τονιστεί πως με μεγαλύτερες τιμές το κόστος επεξεργασίας ανεβαίνει.



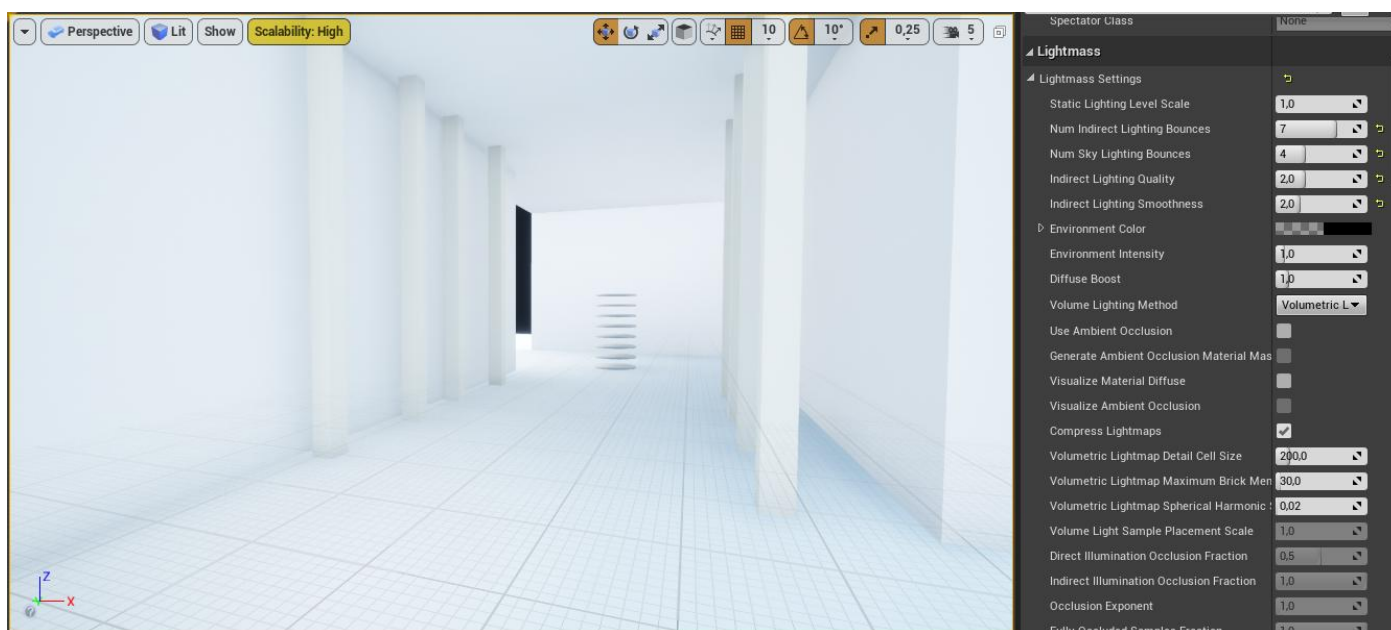
Εικόνα 2.12 Χρήση Cubemap για φωτισμό

Στο παράθυρο World Settings στην ομάδα Lightmass δίνονται επιλογές για την ανάλυση και την ποσότητα των έμμεσων μεταπιδίσεων φωτισμών που θα έχει η σκηνή. Το Lightmass είναι τεχνολογία που χρησιμοποιεί σωματίδια για τον υπολογισμό του έμμεσου φωτισμού και συμβάλλει στον προϋπολογισμό τμημάτων φωτισμού, με στατική ή ακίνητη κινητικότητα. Η χρήση του Lightmass επιτρέπει καλύτερους υπολογισμούς φωτισμού κατά την δημιουργία του παιχνιδιού και δεν είναι ενεργό κατά την εκτέλεση. Μία ιδιότητα που προσφέρει είναι η μεταφορά χρωματικής πληροφορίας των τριγύρω μοντέλων σε φως. Κλειστοί χώροι που κατοικούνται από 3D μοντέλα έχουν πιο καλοσχεδιασμένο και ρεαλιστικό φωτισμό λόγω των ανακλώμενων ακτίνων που υπολογίσει το Lightmass. Από την στιγμή που ένα παιχνίδι περιέχει διαφορετικά είδη φωτισμού, που το καθένα μπορεί να βρίσκεται σε διαφορετική κινητικότητα, θα πρέπει να υπολογιστούν μόνο τα σημεία με ακίνητο ή στατικό φωτισμό και όχι αυτά με κινητό. Αυτό γίνεται για εξοικονόμηση χρόνου κατασκευής, αλλά και πόρων ειδικά όταν πρόκειται για επίπεδα που περιέχουν πληθώρα από 3D μοντέλα. Το Lightmass Importance Volume είναι ένας δρόσσης που λειτουργεί σαν ένα κλουβί, δίνοντας μία ένδειξη στην μηχανή για το ποιά σημεία του χάρτη θα πρέπει να υπολογίσει το Lightmass και ποιά όχι. Η Unreal Engine προσφέρει πολλές ρυθμίσεις για την παραμετροποίηση του Lightmass, αλλά οι βασικότερες που κάποιος μπορεί να χρειαστεί να αλλάξει, βρίσκονται στην ομάδα ρυθμίσεων Lightmass Settings. Λόγω της επιρροής που έχουν στον φωτισμό καθώς φτιάχνεται το παιχνίδι, μεγάλες τιμές στις ρυθμίσεις των ανακλάσεων και της ποιότητας μπορεί προκαλέσει πρόβλημα και εν

τέλη να μην ολοκληρωθεί η διαδικασία. Η αλλαγή αυτών των ρυθμίσεων πρέπει να γίνεται με μικρές παραλλαγές στις τιμές, αλλά η αύξηση τους κατά μία μονάδα προσφέρει πάντα αξιοπρεπή αποτελέσματα με μικρό κόστος.



Εικόνα 2.13 Αρχικές ρυθμίσεις Lightmass

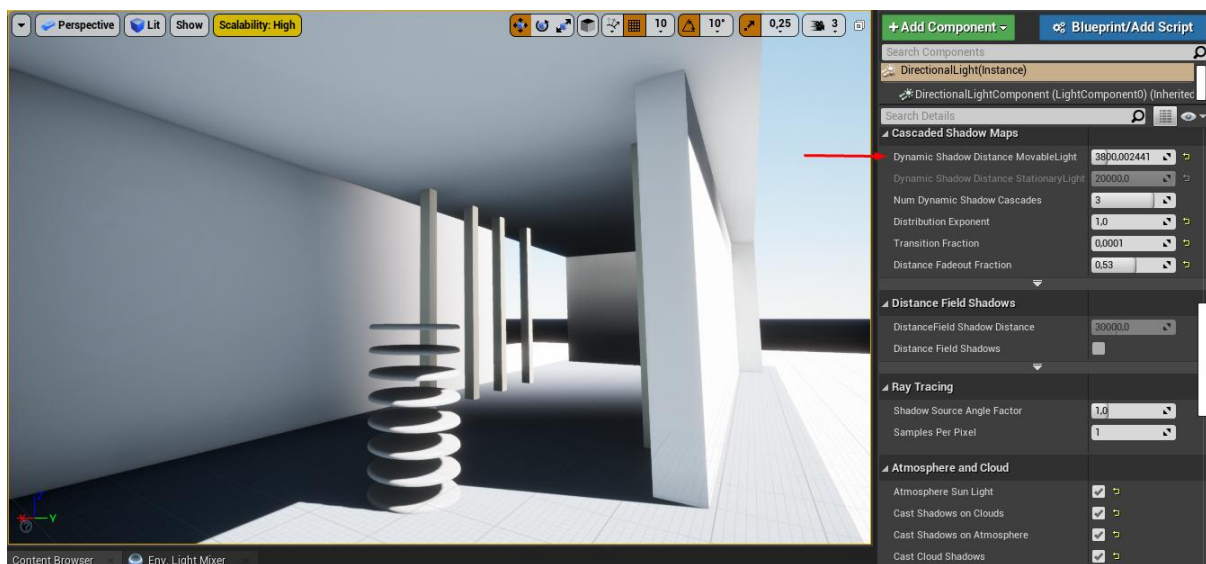


Εικόνα 2.14 Τροποποιημένες ρυθμίσεις Lightmass

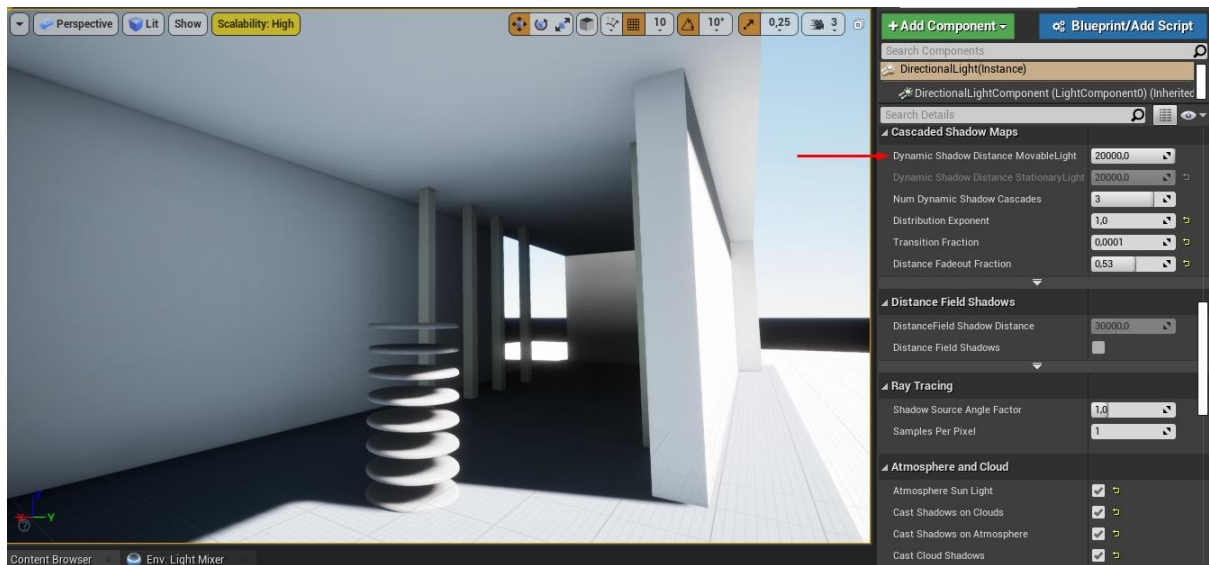
Directional Light: Το Directional Light έχει σαν πηγή ένα απομακρυσμένο σώμα σε άπειρη απόσταση και είναι ο πιο αποτελεσματικός τρόπος για την απεικόνιση του ήλιου. Αυτό σημαίνει πως όλες οι σκιές είναι παράλληλες που βοηθάει με τον ρεαλισμό του παιχνιδιού. Οι ιδιότητες σχετικά με τον φωτισμό είναι σχεδόν ίδιες με το sky light με πρόσθετες επιλογές για την απεικόνιση του ήλιου αν θεωρηθεί πως υπάρχει μία ατμόσφαιρα. Η ιδιότητα Source Angle είναι υπεύθυνη για την διάμετρο

που θα έχει το φωτεινό σώμα. Η αρχική τιμή είναι 0,527 και είναι ίδια με την διάσταση του ήλιου που είναι ορατή από την επιφάνεια της γής. Η αύξηση ή μείωση της τιμής αυτής δεν επηρεάζει την ένταση του φωτισμού. Παρόλο που η αρχική τιμή είναι βασισμένη σε πραγματικά δεδομένα η αύξηση της κατά 0.5-1 μονάδες παρέχει πιο ρεαλιστικό αποτέλεσμα στο περιβάλλον την Unreal. Παράλληλα λόγω της δυνατότητας που δίνει ο δυναμικός φωτισμός μπορούν να εμπλακούν και παραπάνω παράγοντες σχετικά με τις σκιές.

Αρχικά για την τροποποίηση της ποιότητας και ποσότητας των σκιών στο επίπεδο θα γίνει μέσω των ιδιοτήτων που βρίσκονται στην ομάδα Cascaded Shadow Maps αν το Directional Light είναι stationary ή movable. Η πρώτη ιδιότητα dynamic shadow distance αναφέρεται στην απόσταση που θα είναι ορατές οι σκιές. Ο σωστός χειρισμός αυτής της ιδιότητας, μπορεί να αποτελέσει έναν από του κύριους λόγους που ένα παιχνίδι είναι αρκετά βέλτιστο σε θέμα επεξεργασίας. Οι μεγάλες τιμές σε επίπεδα με πληθώρα από γεωμετρία σημαίνει και μεγάλο κόστος επεξεργασίας, αλλά αν ένα επίπεδο είναι σχετικά άδειο ή τα 3D assets έχουν μικρό πλήθος από τρίγωνα, τότε είναι λιγότερο κοστοβόρο για τον υπολογιστή.

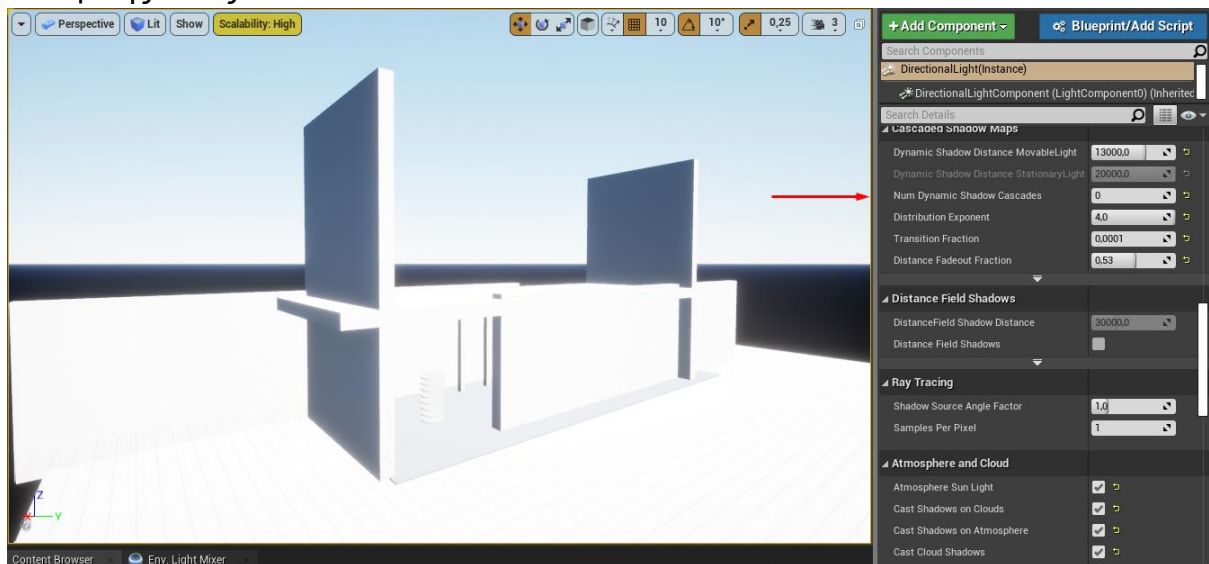


Εικόνα 2.15 dynamic shadow distance < 4000 (ορατή αλλαγή = κακό αποτέλεσμα)

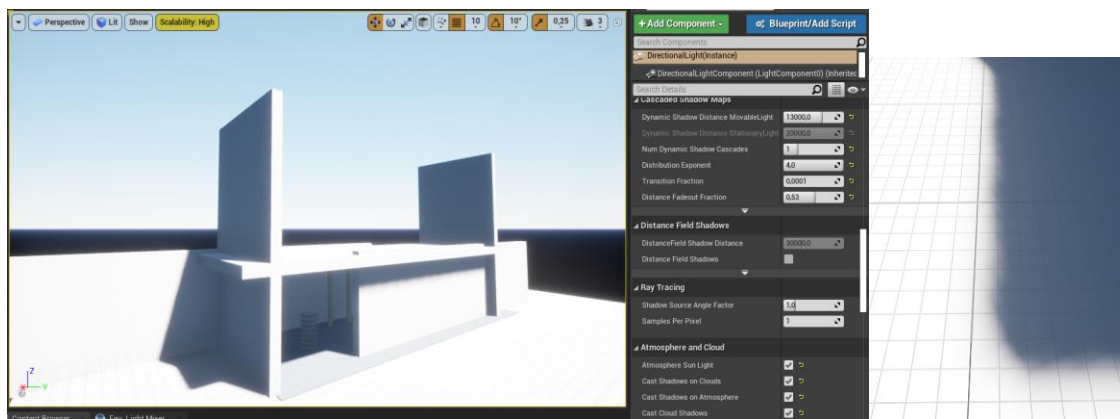


Εικόνα 2.16 dynamic shadow distance μεγάλη τιμή (μεγάλο κόστος)

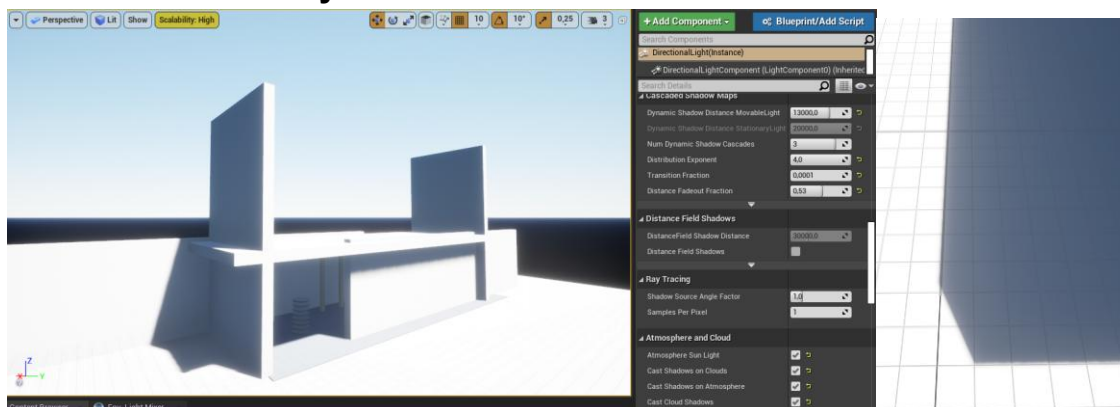
Η δεύτερη ιδιότητα, Num Dynamic Shadow Cascades είναι υπεύθυνη για την ποιότητα των σκιών από τις τριγύρο επιφάνειες. Αν το Num Dynamic Shadow Cascades έχει μηδενική τιμή αυτό σημαίνει πως οι επιφάνειες που έρχονται σε επαφή με το φως θα είναι φωτιζόμενες, ενώ όποια επιφάνεια δεν φωτίζεται, θα είναι σκούρα χωρίς να υπάρχουν ενδιάμεσες σκιές. Όσο η τιμή είναι πάνω από το μηδέν θα υπάρχει σκιά από άλλες επιφάνειες και ταυτόχρονα θα αυξάνεται και η ποιότητα των σκιών. Οι επόμενες τρεις επιλογές στην ομάδα Cascaded Shadow Maps αφορούν την ομαλή προβολή των Cascaded σκιών. Συνοπτικά το Distribution exponent μεταβάλλει την απόσταση που θα είναι ορατά τα cascades σε σχέση με την κάμερα, το Transition Fraction που ορίζει την περιοχή εξασθένησης των cascade και τέλος το Distance Fadeout Fraction που ελέγχει το σημείο εξασθένησης στην έκταση της επιρροής της δυναμικής σκιάς.



Εικόνα 2.17 Num Dynamic Shadow Cascades = 0



Εικόνα 2.18 Num Dynamic Shadow Cascades =1



Εικόνα 2.19 Num Dynamic Shadow Cascades =3

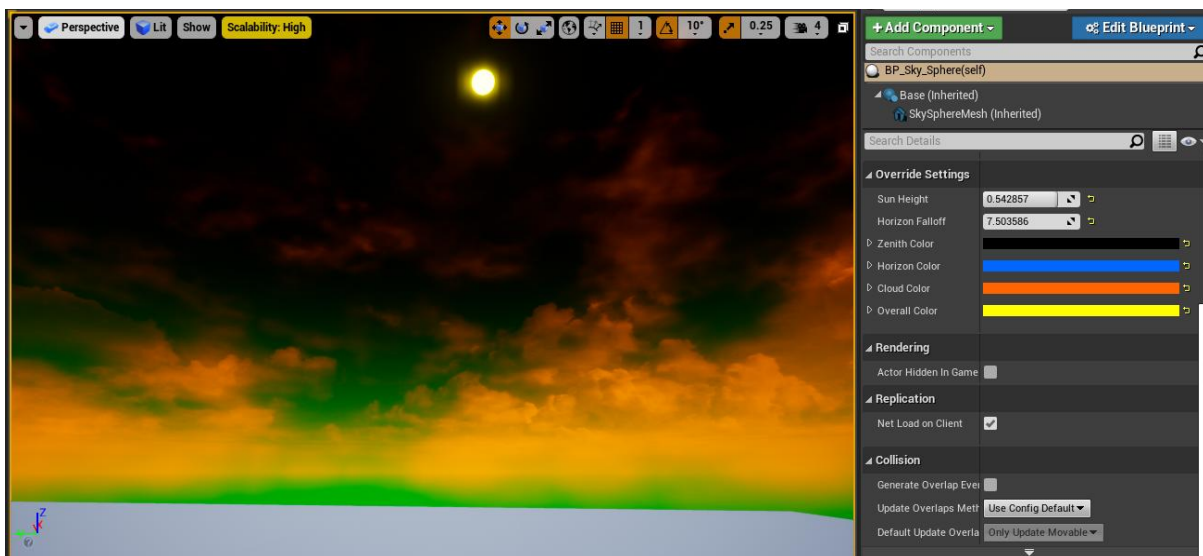
Κεφάλαιο 3 Ατμόσφαιρα

3.1 BP_Sky_Sphere

Ανάλογα με το είδος του παιχνιδιού ο σχεδιαστής ή ο προγραμματιστής δεν θα αρκεστεί μόνο στον φωτισμό για να κάνει πιο αληθοφανές το κάθε επίπεδο. Για παράδειγμα αρκετά από τα παιχνίδια του είδους Horror πολλές φορές διαδραματίζονται μέσα σε κλειστούς χώρους χωρίς να χρειάζεται να σχεδιαστεί η ατμόσφαιρα που θα τα φιλοξενεί. Αντίθετα σε Open World παιχνίδια ή οποιοδήποτε παιχνίδι που ο παίκτης θα χρειαστεί να έρθει σε επαφή με το εξωτερικό περιβάλλον θα πρέπει να υπάρχει και μία ατμόσφαιρα που θα αποτελείται τουλάχιστον από έναν ουρανό και έναν δρόστη για τον ήλιο ή για το φεγγάρι. Στην Unreal Engine υπάρχουν δύο βασικοί τρόποι για να απεικονιστεί η ατμόσφαιρα σε ένα επίπεδο. Η πρώτη επιλογή έρχεται με την χρήση του δράση BP_Sky_Sphere που είναι και η προκατασκευασμένη επιλογή με την δημιουργία ενός project. Αν για κάποιο λόγο έχει διαγραφεί το BP_Sky_Sphere μπορεί να βρεθεί μέσα στον φάκελο Engine Content > Engine Sky > BP_Sky_Sphere. Από την στιγμή που το BP_Sky_Sphere ορίζεται ως

δράστης γίνεται να τροποποιηθούν τα χαρακτηριστικά του μέσω του Blueprint Editor ενώ ταυτόχρονα μπορεί να αλλάξει σκέλος (σαν να είναι ένα αντικείμενο με γεωμετρία). Αυτό παίζει ρόλο ανάλογα με το πόσο μεγάλο είναι το επίπεδο που θα χρειαστεί να καλυφθεί από την ατμόσφαιρα. Επίσης η αλλαγή της κλίμακας θα πρέπει να είναι ομοιόμορφη στους x,y,z άξονες για να μην υπάρχουν παραμορφώσεις. Οι βασικές αλλαγές σχετικά με τα χαρακτηριστικά του δράστη μπορούν να γίνουν μέσω των Default και Override Settings.

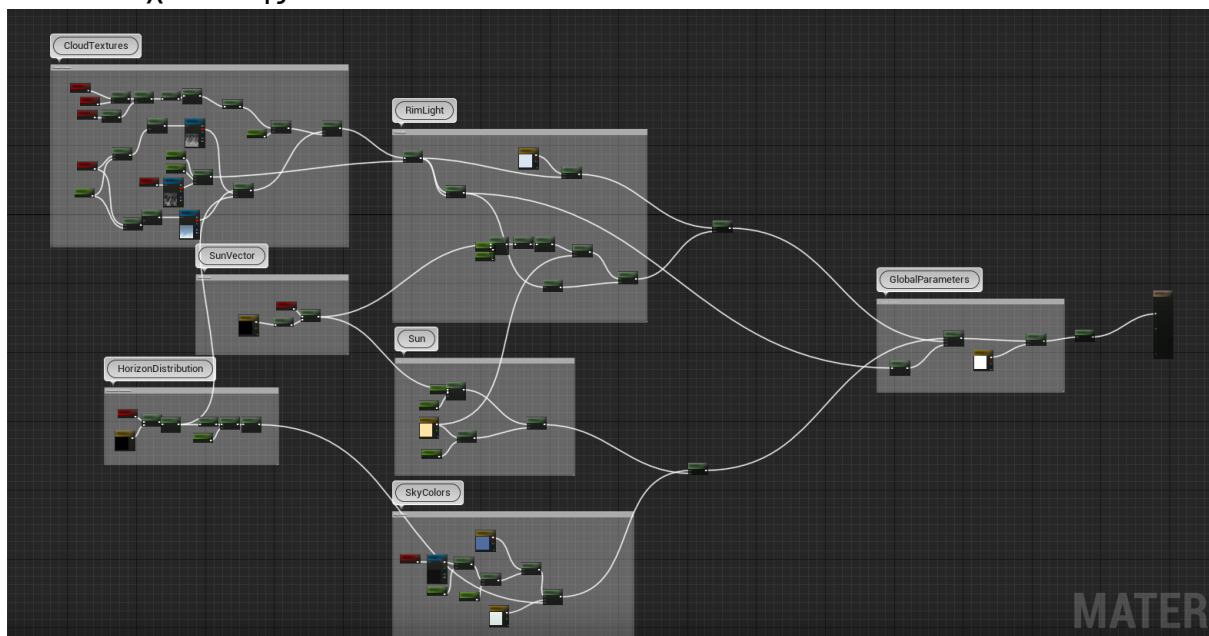
Στην Override Settings υπάρχουν επιλογές για την αλλαγή των χρωματικών τιμών του ουρανού, του ορίζοντα, των συννέφων, αλλά και του γενικού τόνου χρώματος που θα έχει η ατμόσφαιρα. Πάνω από τις χρωματικές ιδιότητες υπάρχει η επιλογή για την ελάττωση της διαφοράς του ορίζοντα με τον ουρανό (Horizon Falloff) και της θέσης του ήλιου, που ανάλογα με την θέση του θα υπάρχει και αλλαγή στον ουρανό, αν η επιλογή Colors Determined By Sun Position είναι αληθής.



Εικόνα 3.1 Προσαρμογή Override Settings.

Στην Default ομάδα υπάρχουν επιλογές για τα φωτεινά σώματα που έχει η ατμόσφαιρα όπως ο ήλιος και τα αστέρια. Τα σύννεφα μπορούν να πάρουν τιμές για την ταχύτητα αλλά και την πυκνότητα μέσω του Cloud speed και Opacity ενώ η ιδιότητα Sun Brightness μεταβάλλει το μέγεθος του φωτεινού σώματος που απεικονίζει τον ήλιο. Αν τοποθετηθούν αντικείμενα στο επίπεδο άσχετα με την θέση και την ένταση του ήλιου (Sun Brightness, Sun Height) δεν θα υπάρχει φωτισμός. Ο ήλιος που είναι ορατός, δεν είναι στην πραγματικότητα παρα μια απεικόνιση ενός φωτεινού σώματος. Αν τοποθετηθεί ένα directional light λύνεται αυτό το πρόβλημα αλλά υπάρχουν συγκεκριμένοι περιορισμοί. Αν η ένταση από το directional light έχει σταθερή τιμή, σημαίνει πως μέρα ή νυχτα, η ένταση στο φως θα παραμείνει ίδια. Αυτό μπορεί να ρυθμιστεί ανάλογα από τον σχεδιαστή και δεν θα τον επηρεάσει αν δεν είναι απαραίτητη αυτή η αλλαγή. Από την επιλογή Direction Light Actor μπορεί να οριστεί η πηγή του φωτισμού.

Η τεχνική που χρησιμοποιείται για την δημιουργία ατμόσφαιρας με το BP_Sky_Sphere λέγεται skybox και είναι η πιο πολυχρησιμοποιημένη λόγω του χαμηλού κόστους που έχει. Στην πραγματικότητα αυτό που προβάλλεται σαν ατμόσφαιρα είναι μία στατική εικόνα ή ένα υλικό, που παίρνει το σκέλος του δράστη ο οποίος έχει οριστεί σαν ατμόσφαιρα. Συνήθως αυτός ο δράστης είναι ένας κύβος ή μία σφαίρα. Αν ο ουρανός πρέπει να αλλάζει από μέρα σε νύχτα ή πρέπει να υποστεί χρωματική αλλαγή, με βάση κάποιου γεγονότος στο παιχνίδι, δεν μπορεί να γίνει χρήση στατικής εικόνας. Θα πρέπει να δημιουργηθεί ένα υλικό μέσα στην Unreal Engine, που θα αλλάζει παραμέτρους ανάλογα με τις προδιαγραφές που του έχει δώσει ο σχεδιαστής.

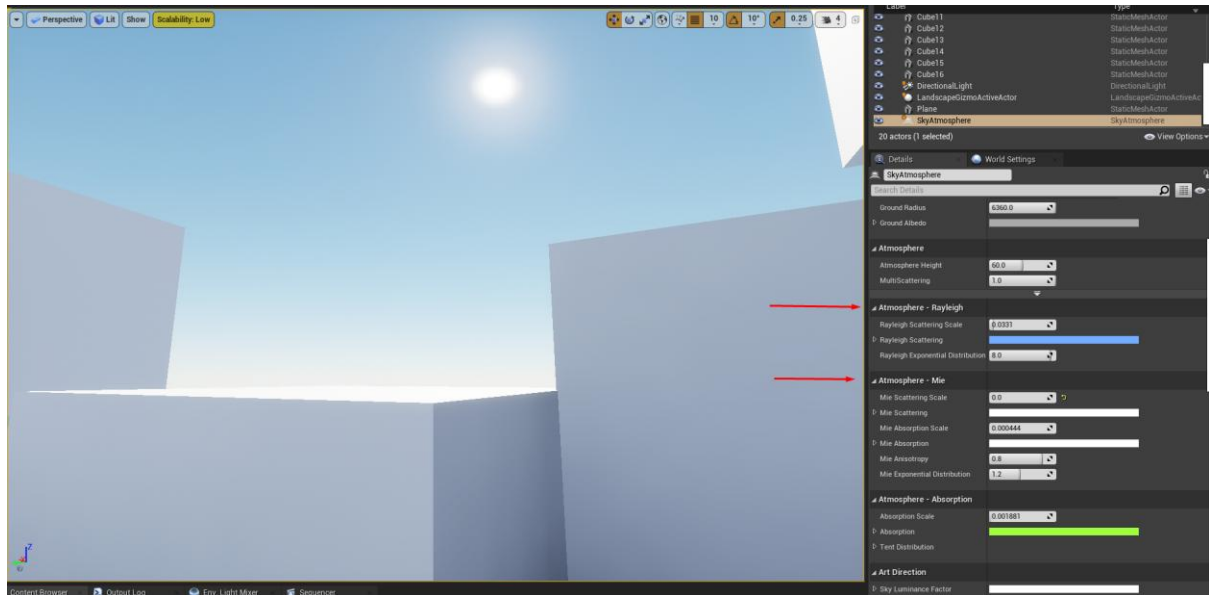


Εικόνα 3.2 Υλικό ατμόσφαιρας BP_Sky_Sphere.

3.2 Sky atmosphere

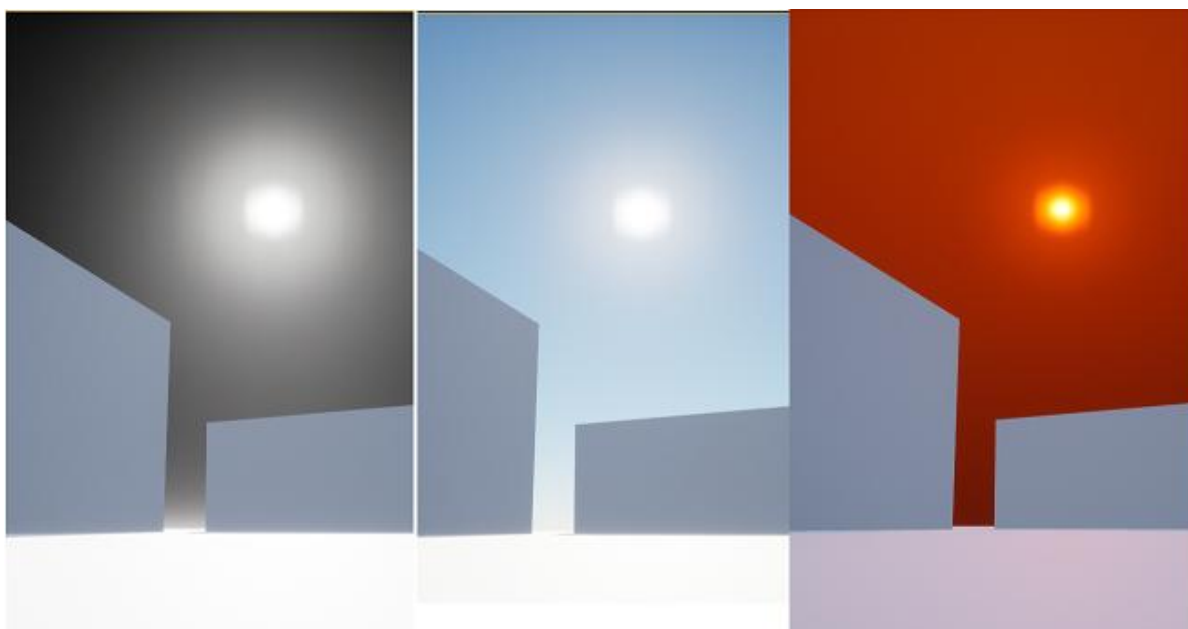
Στο παράθυρο place actors στην ομάδα Visual Effects υπάρχει ο δράστης Sky Atmosphere. Αν τοποθετηθεί στο επίπεδο δεν θα παρατηρηθεί απευθείας κάποια αλλαγή. Ο λόγος είναι πως το sky atmosphere λειτουργεί σαν μία πραγματική ατμόσφαιρα που πρέπει να έχει αντανάκλασεις ήλιου για να είναι ορατή. Όπως και στο BP_Sky_Sphere η πηγή θα είναι το directional light. Αφού προστεθεί το επίπεδο στις ιδιότητες του directional light στην ομάδα Atmosphere and Cloud πρέπει η επιλογή Atmosphere Sun Light να τεθεί σαν True. Έπειτα η ατμόσφαιρα θα γίνει ορατή. Πατώντας control-L και με την χρήση του ποντικιού μπορεί να αλλαχθεί η θέση του ήλιου, επιτρέποντας άμεση αλλαγή από μέρα σε νύχτα. Σε πλανήτες με ατμόσφαιρα σαν της Γης υπάρχουν πολλαπλά στρώματα αερίων που τα ίδια αποτελούνται από σωματίδια. Ανάλογα με την πυκνότητα των σωματιδίων οι ακτίνες του ήλιου (directional light) θα απορροφούνται ή θα αντανakλώνται. Στην unreal όπως και στην ατμόσφαιρα της Γης υπάρχουν δύο είδη σκέδασης, rayleigh και mie. Ανάλογα με τις

τιμές που θα έχουν τα είδη σκέδασης θα υπάρχουν και χρωματικές παραμορφώσεις στην ατμόσφαιρα ανάλογα με ποια μήκη κύματος θα ανακληθούν.



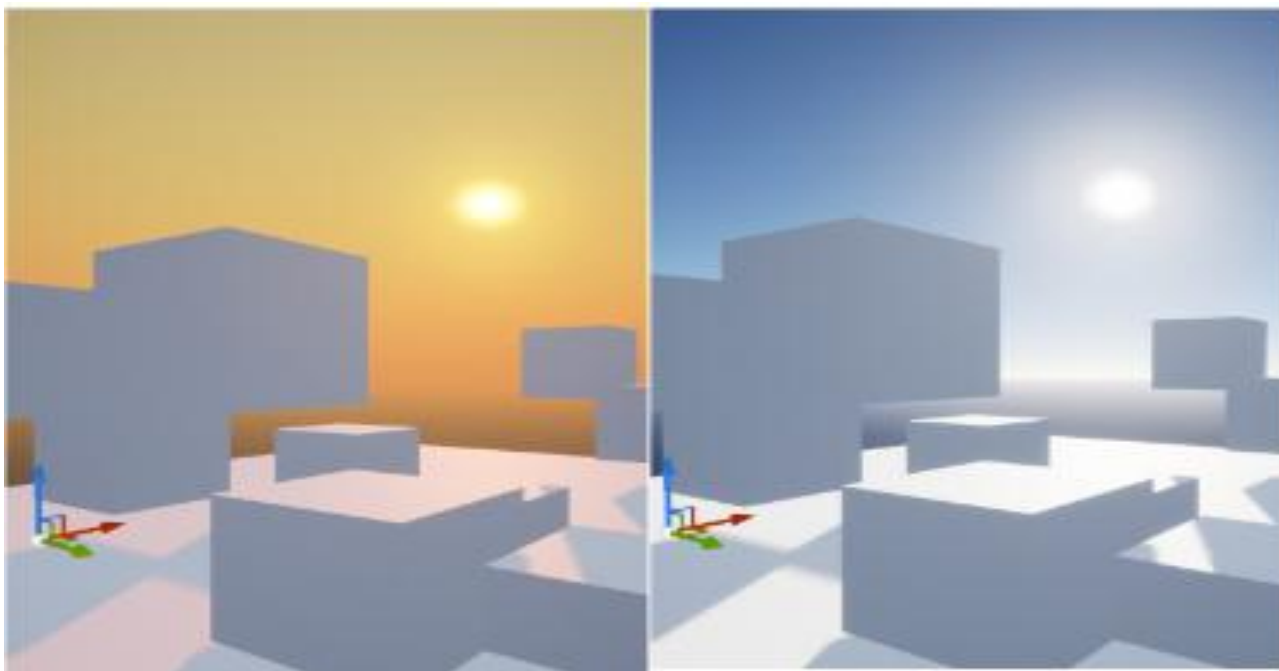
Εικόνα 3.3 Είδη Σκέδασης

Rayleigh: Η Rayleigh σκέδαση είναι αποτέλεσμα ανάκλασης ακτίνων με σωματίδια που είναι μικρότερα από το μήκος κύματος της ακτίνας. Τα μικρότερα μήκη κύματος (μπλε) είναι διασκορπισμένα πιο έντονα από τα μεγαλύτερα μήκη κύματος (κόκκινα). Αυτό έχει ως αποτέλεσμα το έμμεσο μπλε φως που προέρχεται από τον ουρανό κατά την διάρκεια της ημέρας και κίτρινο και κόκκινο το απόγευμα που το φως του ήλιου πρέπει να διέρχεται μεγαλύτερο ποσοστό της ατμόσφαιρας. Με την μείωση του Rayleigh Scattering Scale θα υπάρχει μικρότερο ποσοστό ανάκλασης. Αν η τιμή του Rayleigh Scattering Scale είναι 0 ο ουρανός θα γίνει μαύρος, από την στιγμή που καμία ακτίνα δεν θα ανακλαστεί και στην πραγματικότητα αυτό που είναι ορατό είναι το διάστημα. Το ακριβώς αντίθετο θα προκύψει με την αύξηση αυτής της τιμής, μεγαλώνοντας την πυκνότητα της ατμόσφαιρας, προκαλώντας ανάκλαση και στα μεγαλύτερα μήκη κύματος.



Εικόνα 3.4 Rayleigh Scattering Scale 0.0 ,0.034, 2.0

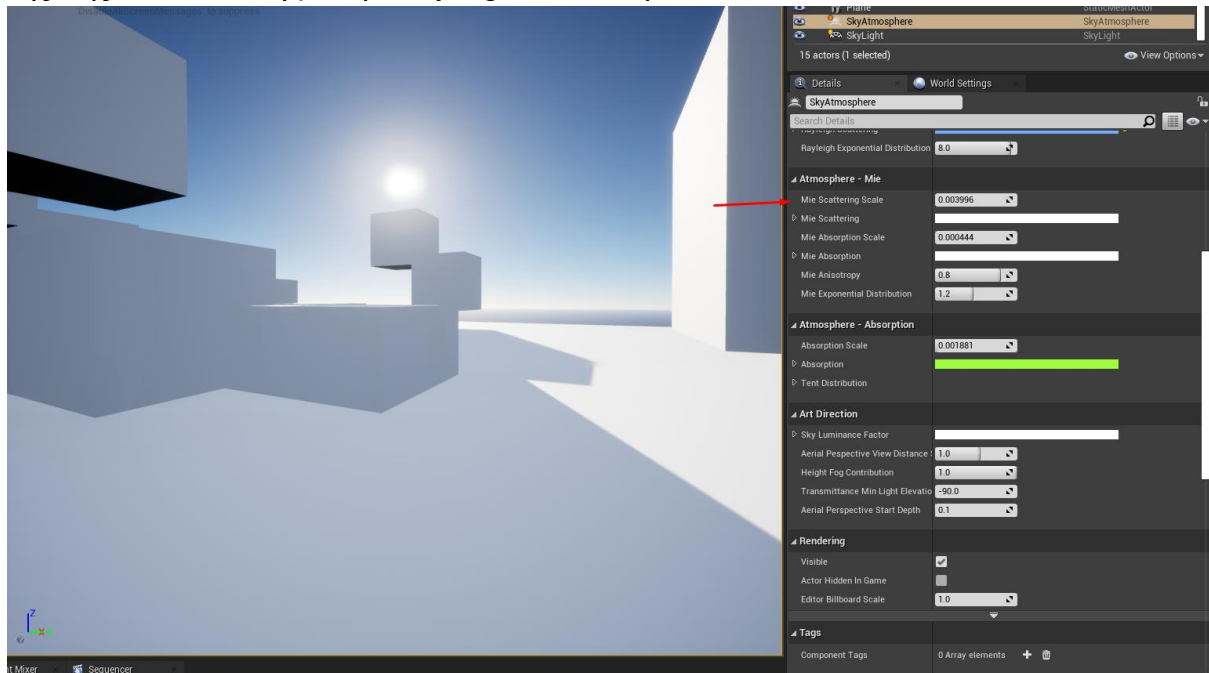
Με την ελάττωση ή την αύξηση του Rayleigh Scattering Scale μπορεί να εμπλουτιστεί ο φωτισμοί και η αίσθηση του επιπέδου. Αν το παιχνίδι ή το συγκεκριμένο επίπεδο λαμβάνει χώρα σε περιοχές όπως έρημο ή πλανήτες όπως ο Άρης οι τιμές κοντά στο 0.19-0.31 μπορούν να δώσουν την αίσθηση ουρανού, με πυκνά σωματίδια σκόνης και άμμου. Αντίθετα τιμές μικρότερες της αρχική (0.0331) δίνουν μία πιο ψυχρή όψη στην ατμόσφαιρα.



Εικόνα 3.5 Rayleigh Scattering Scale 0.31 , 0.02

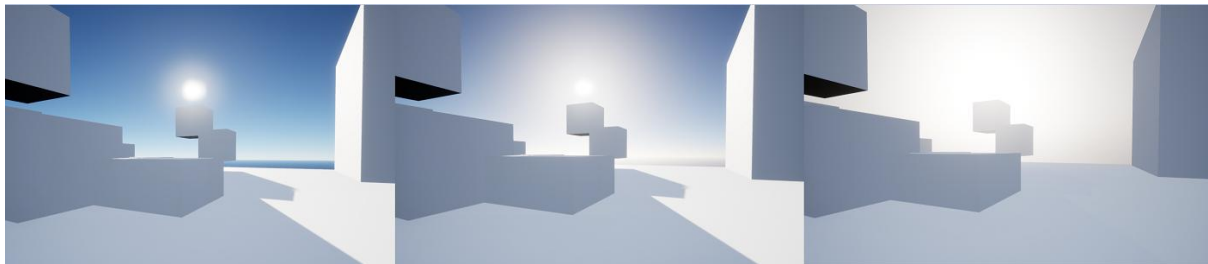
Mie: Η Mie σκέδαση προέρχεται από την επαφή των ακτίνων του ήλιου με σωματίδια που έχουν παρόμοια ή μεγαλύτερη διάμετρο από το μήκος κύματος των

ακτίνων. Αυτά τα σωματίδια αποτελούνται από σκόνη, γύρη και καυσαέριο που σημαίνει ότι η Mie σκέδαση προκαλείται σε μικρότερη απόσταση από την επιφάνεια της Γης σε αντίθεση με την Rayleigh σκέδαση.



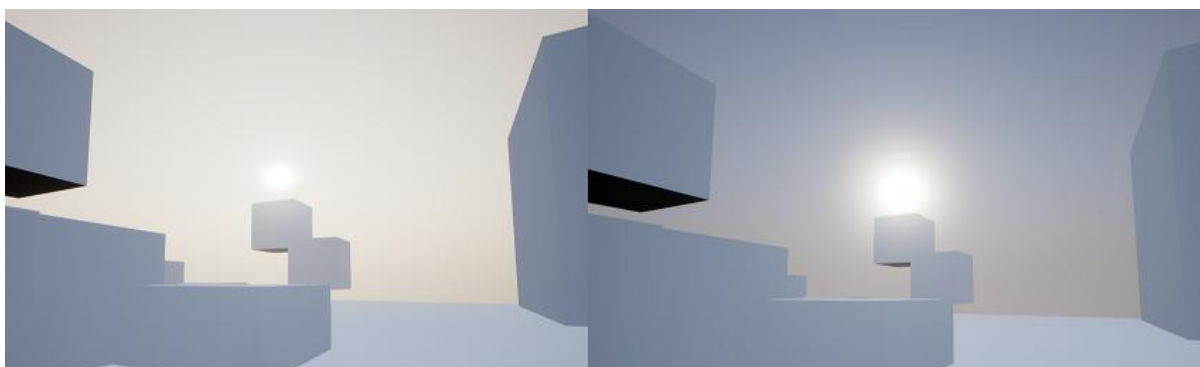
Εικόνα 3.6 Mie Scattering

Ανάλογα με το ποσοστό του Mie Scattering Scale και από την στιγμή που οι ακτίνες απορροφώνται άσχετα με το μήκος κύματος τους το αποτέλεσμα θα κυμαίνεται ανάμεσα από μία καθαρή ατμόσφαιρα ή ατμόσφαιρα με μεγάλο ποσοστό νέφους ή ομίχλης (υγρασίας στην ατμόσφαιρα).



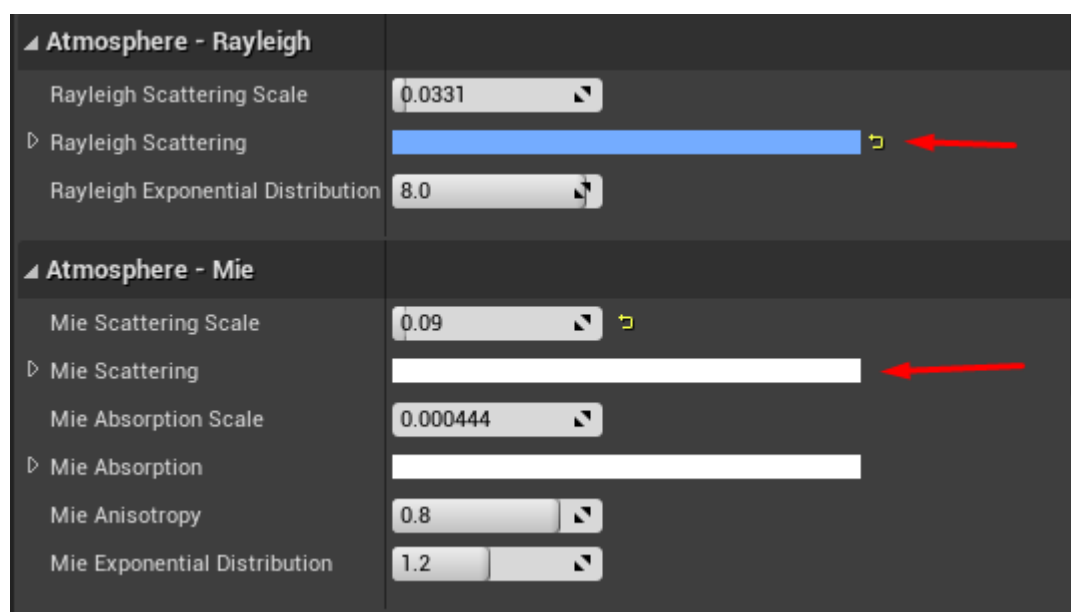
Εικόνα 3.7 Mie Scattering Scale 0.0,0.1,0.9

Στις παραπάνω εικόνες παρατηρείται ότι αν το ποσοστό της Mie σκέδαση είναι αυξημένο κατά μεγάλο ποσοστό παύουν να υπάρχουν σκιές στα αντικείμενα του επιπέδου, κάτι που μπορεί να επηρεάσει αρνητικά, την αίσθηση του παιχνιδιού με την παρατεταμένη ή έντονη χρήση του. Επίσης η φορά της σκέδασης μπορεί να ρυθμιστεί από την ιδιότητα Mie Anisotropy κάνοντας τις ακτίνες να διαχέονται με πιο απότομη εμπρόσθια φορά.



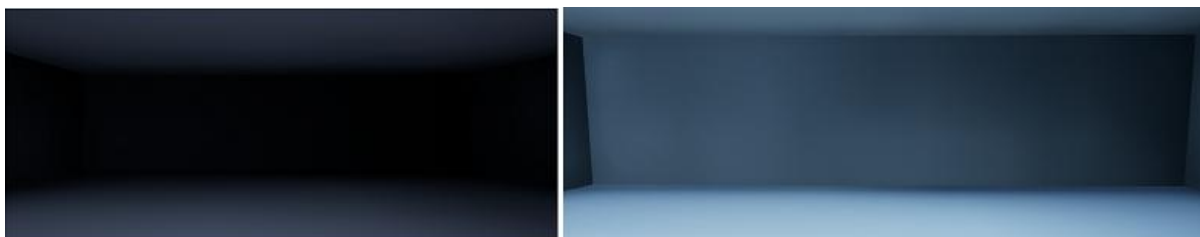
Εικόνα 3.8 Mie Anisotropy 0.3 και 0.99

Μία ιδιότητα που έχουν και τα δύο είδη σκέδασης είναι η χρωματική αλλαγή που μπορεί να ρυθμιστεί, ανάλογα με το περιβάλλον που ο σχεδιαστής θέλει να δημιουργήσει.



Εικόνα 3.9 Χρωματικές ιδιότητες Σκέδασης

Σε προηγούμενο υποκεφάλαιο αναφέρθηκε η ιδιότητας Real Time capture του Sky Light. Τώρα με την ύπαρξη ατμόσφαιρας το Sky Light μπορεί να έχει σαν παραμέτρους για την δημιουργία φωτισμού, τα περισσότερα από τα αντικείμενα που αποτελούν την ατμόσφαιρα όπως ο ήλιος, ο ουρανός, τα σύννεφα αν έχει προστεθεί ο δρόμησης Volumetric Clouds, και αντανakλάσεις από 3D assets. Άλλο ένα έναυσμα που παρέχει η πρόσθεση ενός Sky Light στο επίπεδο ή στο παιχνίδι είναι η συνεργασία που έχει με το Directional Light επιτρέποντας γρήγορες αλλαγές, σε τιμές όπως το Intensity Scale, για τυχόν εξομάλυνση σκιών σε πολύ σκοτεινά μέρη αλλά και χρωματικές αλλαγές.



Εικόνα 3.10 Intensity Scale του Sky Light 0 και 0.14 με πρωτεύον πηγή το Directional Light

Μία ιδιότητα του Directional Light που μπορεί να ρυθμιστή μόνο αν υπάρχει ατμόσφαιρα είναι το Light Shafts Occlusion και Bloom, που σαν στόχο έχουν την αναπαράσταση των Crepuscular ακτίνων ή God rays που λάμπουν μέσα από ανοίγματα σε σύννεφα ή μεταξύ άλλων αντικειμένων, όπως βουνά και κτίρια. Η μέθοδος Light Shaft Occlusion δημιουργεί μια μάσκα στα βαθύτερα αντικείμενα της οθόνης και με την χρήση της ατμοσφαιρικής ομίχλης, το σημείο επαφής ακτίνας και αντικειμένου γίνεται πιο θολό. Η συγκεκριμένη ιδιότητα προορίζεται για παιχνίδια που βασίζονται στον ρεαλισμό αλλά αυτό σημαίνει επίσης, ότι οι άξονες φωτός μπορούν να είναι τόσο ισχυροί όσο η ομίχλη της ατμόσφαιρα. Η μέθοδος Bloom αποτυπώνει το χρώμα της σκηνής, γύρω από την πηγή φωτός, στον χώρο και την θολώνει ακτινικά, μακριά από το φως. Αυτή η μέθοδος δεν μιμείται πραγματικά κάτι που συμβαίνει στον πραγματικό κόσμο, αλλά είναι πολύ ελεγχόμενη και δεν περιορίζεται από την πυκνότητα της ομίχλης.



Εικόνα 3.11 Occlusion

Εικόνα 3.12 Bloom

Κεφάλαιο 4 Εμπλουτισμός επιπέδου

Πριν ξεκινήσει η ανάπτυξη ενός παιχνιδιού και όταν η ομάδα έχει μόνο την αρχική ιδέα σαν οδηγό δεν συνίσταται και σχεδόν ποτέ δεν εφαρμόζεται η απευθείας υλοποίησή του. Μία σωστή μελέτη και κατανόηση του περιβάλλοντος και των χαρακτήρων, που θα αποτελούν το σύνολο του παιχνιδιού συντελεί και ένα καλά ολοκληρωμένο παιχνίδι, με την κατάλληλη συνοχή, για να μην κάνει τον παίκτη να αισθάνεται αφιλόξενος. Σε αυτό το στάδιο είναι πολύ σημαντικός ο ρόλος των concept artists, που είναι υπεύθυνοι για την εύρεση του στιλ και της εμφάνισης του παιχνιδιού. Στόχος τους είναι να δώσουν την πρώτη και βασική απεικόνιση του περιβάλλοντος, των εχθρών και των χαρακτήρων, επιτρέποντας στους 3d artists και στους προγραμματιστές να έχουν ένα γενικό πλάνο για την κατεύθυνση που θα πάρει το παιχνίδι. Προβλήματα σε αυτόν τον τομέα αντιμετώπιζαν μικρές ομάδες που δεν είχαν το απαραίτητο προσωπικό ή προϋπολογισμό. Πλέον μέσω του διαδικτύου είναι εύκολο για κάποιον, να συλλέξει ιδέες και υλικό για να εμπλουτίσει το περιεχόμενο του αναπτυσσόμενου παιχνιδιού. Ιδιαίτερη προσοχή πρέπει να δοθεί στο περιβάλλον, μιας και είναι ο σημαντικότερος παράγοντας που θα ελκύσει τον παίκτη από αισθητική μεριά. Ένα ενδιαφέρον περιβάλλον παρακινεί τον παίκτη να το εξερευνήσει και να δεθεί με αυτό, προσθέτοντας στην τελική εμπειρία. Η κάθε ομάδα ανάλογα με το παιχνίδι αποφασίζει το σκέλος και το στυλ που θα ακολουθήσει, με αποτέλεσμα να μην υπάρχει μια συγκεκριμένη συνταγή για την δημιουργία περιβάλλοντος στα παιχνίδια.

4.1 Δημιουργία και επεξεργασία εδάφους

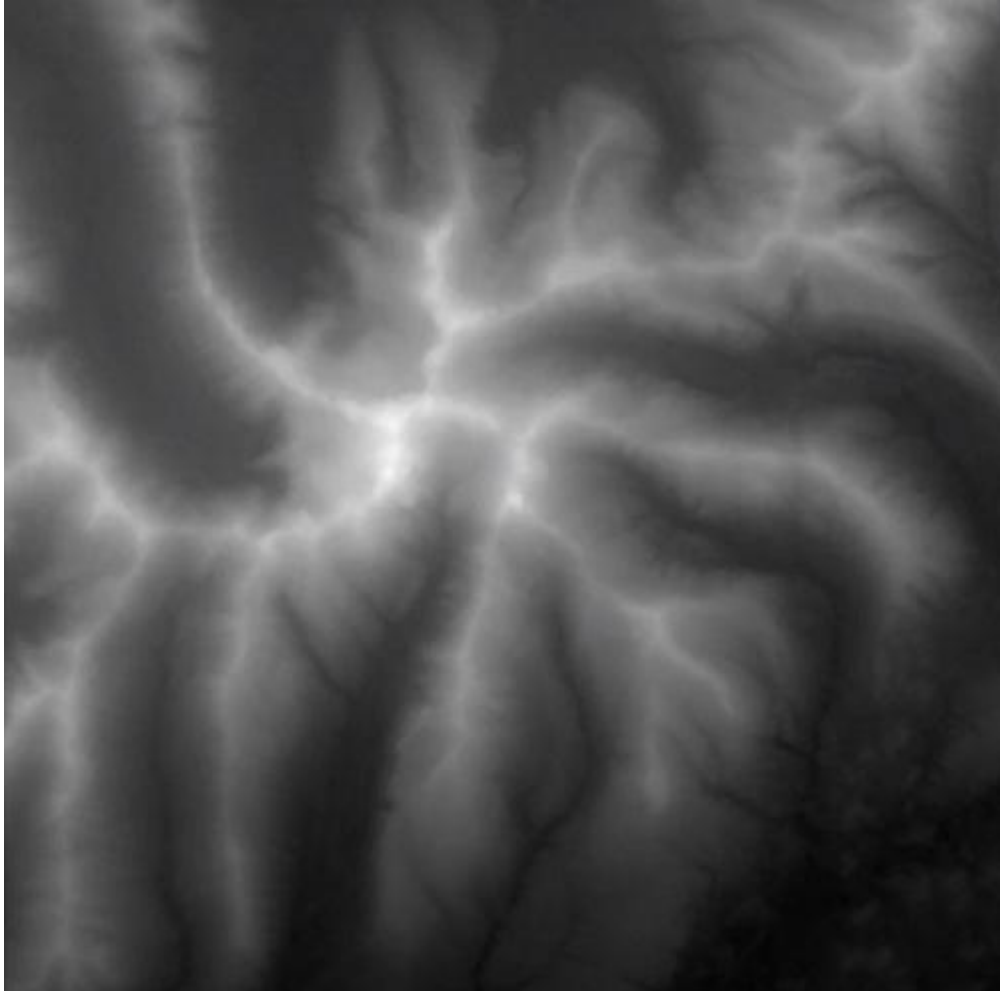
4.1.1 Έδαφος-Landscape

Η ύπαρξη εδάφους ή landscape στην Unreal μπορεί να επιτευχθεί με δύο βασικούς τρόπους. Ο πρώτος και πιο εύκολος είναι με την χρήση του εργαλείου New Landscape, που βρίσκεται στο Modes>Landscape>Manage>New και ο δεύτερος είναι, με την χρήση ενός εισαγόμενου επιπέδου (plane) από κάποιο δευτερεύον πρόγραμμα όπως Blender ή Zbrush, που θα έχουν έτοιμη σκαλισμένη επιφάνεια. Η διαφορά μεταξύ δημιουργίας εδάφους στην μηχανή και εισαγόμενου επιπέδου από άλλο πρόγραμμα, είναι ότι η μηχανή θα αναγνωρίσει το έδαφος που δημιουργήθηκε σε αυτήν σαν landscape, αλλά το εισαγόμενο σαν static mesh. Αυτό καθιστά αδύνατη την περαιτέρω τροποποίηση της γεωμετρίας του εισαγόμενου επιπέδου από τα εργαλεία γλυπτικής εδάφους. Στο παράθυρο Landscape οι δύο πρώτες επιλογές επιτρέπουν στον σχεδιαστή, να δημιουργήσει ένα νέο έδαφος με επίπεδη γεωμετρία ή να χρησιμοποιήσει αρχείο Heightmap, που περιέχει πληροφορίες για το ύψος του εδάφους.

Κάθε φορά που προστίθεται ένα καινούργιο αντικείμενο σε ένα επίπεδο, πρέπει πάντα να λαμβάνεται υπόψη το αντίκτυπο που θα έχει το υλικό στον χρόνο επεξεργασίας. Αυτός ο κανόνας ισχύει και για το έδαφος από την στιγμή που είναι ένα σύνολο από κορυφές. Η λεπτομέρεια που μπορεί να έχει το έδαφος μπορεί να οριστεί μέσω των παραμέτρων Section Size, Section Per Component, Number of Components και Overall Resolution.

- **Section Size:** Ένα section αποτελείται από έναν αριθμό κορυφών και τετραγώνων με κύριο σκοπό την αποθήκευση της πληροφορίας ύψους του εδάφους. Κάθε τετράγωνο αποτελείται από τέσσερις κορυφές, μία σε κάθε γωνιά του τετραγώνου, που σημαίνει ότι ο συνολικός αριθμός των κορυφών είναι πάντα δύναμη του δύο. Αν ένα section αποτελείται από από 2x2 τετράγωνα ο συνολικός αριθμός κορυφών θα είναι 3x3 που δεν είναι δύναμη του δύο αλλά ένα section με 3x3 τετράγωνα έχει 4x4 κορυφές. Συμπερασματικά παρατηρούμε ότι για ζυγό αριθμό κορυφών, πάντα της δύναμης του δύο, θα πρέπει να υπάρχει μονός αριθμός τετραγώνων. Ο μέγιστος αριθμός κορυφών που μπορεί να έχει ένα section είναι 256x256 άρα 255x255 τετράγωνα.
- **Number of Components:** Κάθε έδαφος χωρίζεται σε Components τα οποία είναι η βάση μέτρησης της Unreal για τον υπολογισμό ορατότητας, σύγκρουσης και σύνθεσης εικόνας με κάθε component να αποτελείται από έναν αριθμό από sections. Αυτή η παράμετρος ορίζεται τον αριθμό των component με τον μέγιστο να είναι 32x32 για λόγους βελτιστοποίησης. Ένα έδαφος με μεγάλο αριθμό component, έχει μεγαλύτερη επιβάρυνση στον κάθε υπολογιστή που θα τρέχει το παιχνίδι. Για αυτόν τον λόγο είναι προτιμότερη η χρήση component, με μεγάλο αριθμό sections, παρά η αύξηση κλίμακας του εδάφους, με την προσθήκη component.
- **Section Per Component:** Για κάθε component υπάρχει η δυνατότητα διαίρεσης του σε επιμέρους τμήματα των 1x1 ή 2x2 sections. Με αυτόν τον τρόπο γίνεται να αυξηθεί το μέγεθος του εδάφους με μικρότερο κόστος. Για παράδειγμα αν υπάρχει ένα έδαφος με 4x4 components και με 1x1 sections και ένα έδαφος με 2x2 components και με 2x2 sections, το αποτέλεσμα και στις δύο περιπτώσεις θα είναι ένα έδαφος 4x4. Παρόλα αυτά το έδαφος με λιγότερα components, θα έχει καλύτερη απόδοση λόγω του υπολογισμού λεπτομέρειας (LOD) της μηχανής.
- **Overall Resolution:** Η συνολική ανάλυση ή Overall Resolution του εδάφους είναι ο πολλαπλασιασμός όλων των παραπάνω μεταβλητών. Αυτή η πληροφορία είναι αρκετά σημαντική σε περίπτωση που χρειαστεί να εισαχθεί ένα heightmap. Το overall resolution του εδάφους και του heightmap, πρέπει να έχουν ίδιο μέγεθος. Ο υπολογισμός είναι ίσος με το $[(\text{section size}) * (\text{section per component}) * (\text{number of components}) + 1]$ με +1 να είναι για την επιπλέον κορυφή.

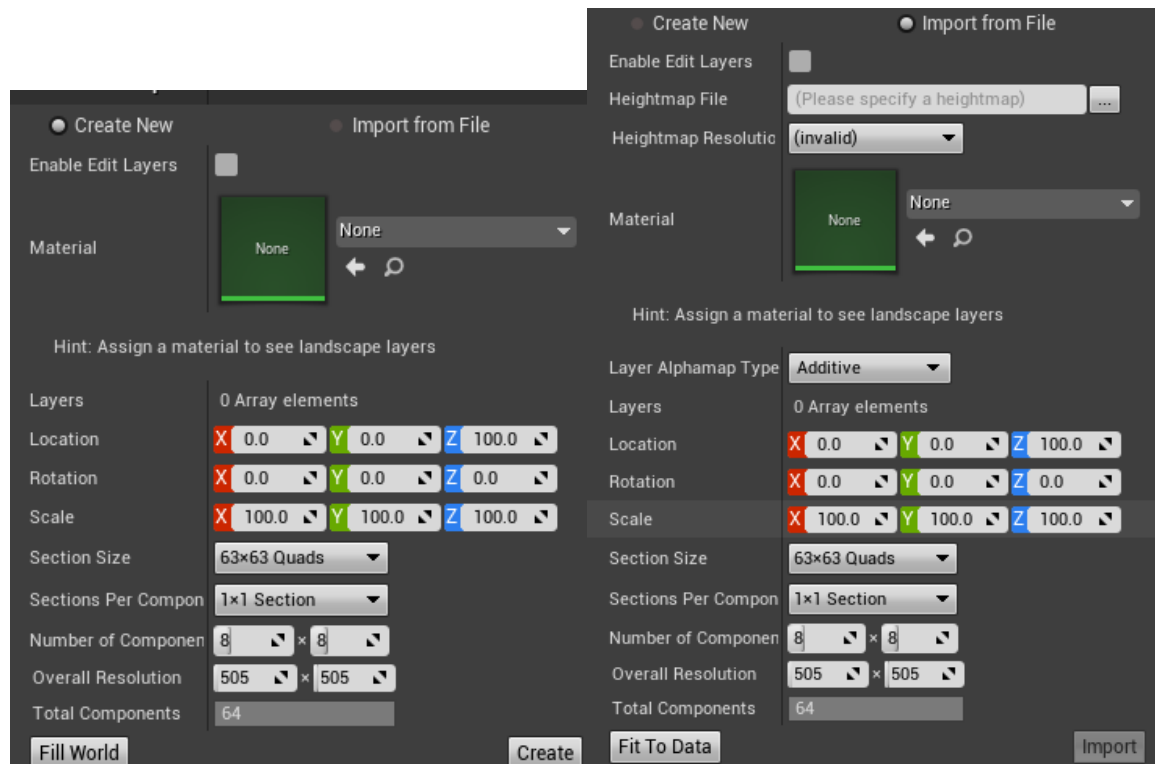
Ένα heightmap χρησιμοποιεί τρία χρώματα για την αποθήκευση πληροφορίας ύψους. Με μαύρο ορίζεται η χαμηλή ανύψωση, με άσπρο η υψηλή και με τις ενδιάμεσες αποχρώσεις του γκρι, να είναι η μέση του heightmap. Αυτή η τεχνοτροπία είναι η ίδια που χρησιμοποιείται και στους χάρτες, για την απεικόνιση βουνών και πεδιάδων. Με την χρήση προγραμμάτων όπως το 3d-mapper ή 3d map generator μπορούν να επιτευχθούν αρκετά ρεαλιστικά αποτελέσματα. Σημαντικό είναι κάθε φορά που γίνεται import ενός heightmap, να είναι grayscale και η ανάλυση να είναι ίδια με την συνολική ανάλυση του εδάφους (Overall Resolution).



Εικόνα 4.1 Heightmap

Σε αυτό το στάδιο αν η παράμετρος enable edit layer γίνει αληθής, θα επιτρέψει την αποθήκευση διαφορετικών πληροφοριών μετά την δημιουργία του εδάφους, σχετικά με τις αλλαγές, που θα υποστεί από τον σχεδιαστή με τα εργαλεία γλυπτικής. Αυτές οι πληροφορίες αποθηκεύονται σε επίπεδα (layers), με το κάθε ένα να είναι ορατό και προσβάσιμο όποτε και αν το επιθυμεί ο σχεδιαστής. Αυτό είναι ένα αρκετά δυνατό εργαλείο που επιτρέπει την άμεση παρέμβαση και αλλαγή του εδάφους, με μη καταστροφικά αποτελέσματα. Τέλος ο ορισμός του υλικού ή material ευθύνεται για την εμφάνιση του εδάφους. Ένα material μπορεί να είναι μία απλή απόχρωση αλλά

συνήθως είναι ένα σύνολο από υφές, που η κάθε μία έχει διαφορετική πληροφορία σχετιζόμενη με τις ιδιότητες του material.



Εικόνα 4.2 Παράθυρο δημιουργίας εδάφους και εισαγωγής heightmap

4.1.2 Γλυπτική Εδάφους-Sculpting

Στο στάδιο γλυπτικής εδάφους υπάρχουν πάντα κάποιοι παράγοντες που θα επηρεάσουν τις αποφάσεις του κάθε σχεδιαστή. Για παράδειγμα, αν σε αυτό το επίπεδο θα χρειαστεί μετέπειτα να προστεθούν εχθροί ή αν η περιοχή θα είναι προσβάσιμη από τον παίκτη, αν το επίπεδο θα διαδραματίζεται σε ένα περιβάλλον με συγκεκριμένα χαρακτηριστικά και κατά πόσο ρεαλιστικό θα πρέπει να είναι το συγκεκριμένο τοπίο. Η Unreal επιτρέπει την τροποποίηση του εδάφους με τα εργαλεία γλυπτικής, αφήνοντας στον σχεδιαστή να πάρει αποφάσεις και να τις εφαρμόσει άμεσα βλέποντας τι ταιριάζει και τι όχι.

Στο Landscape Mode υπάρχουν τρία διαφορετικά παράθυρα. Το πρώτο (Manage) είναι αυτό που επιτρέπει την δημιουργία εδάφους. Στο δεύτερο (Sculpt) είναι που ο σχεδιαστής θα περάσει αρκετό χρόνο μιας και είναι αυτό που του δίνει τα εργαλεία γλυπτικής. Η αλλαγή εδάφους αποθηκεύεται στο δικό του heightmap. Heightmap και μορφή εδάφους είναι όροι που αλληλοσχετίζονται και μπορούν να χρησιμοποιηθούν με αλληλουχία. Για λόγους συντομίας θα χρησιμοποιείται ο όρος heightmap. Στο πρώτο βήμα ο σχεδιαστής θα τροποποιήσει το heightmap και μετέπειτα θα προσθέσει παραπάνω λεπτομέρεια με τα υπόλοιπα εργαλεία. Το

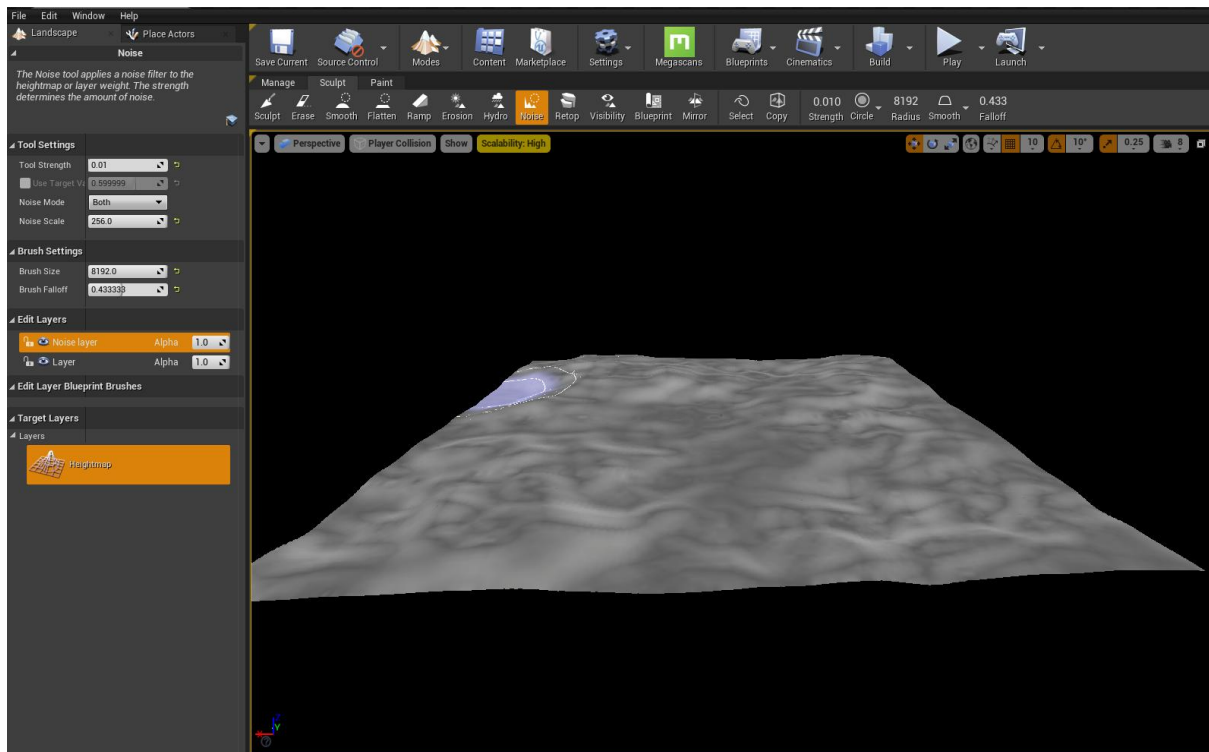
καλύτερο εργαλείο για το συγκεκριμένο βήμα είναι το Noise Tool που προκαλεί τυχαίες αλλαγές στο έδαφος. Αρχικά, θα πρέπει να γίνουν κατανοητές οι ιδιότητες του Noise Tool, ξεκινώντας από αυτές που είναι κοινές και με τα υπόλοιπα εργαλεία γλυπτικής.

- **Tool Strength:** Το ποσοστό με το οποίο το εργαλείο γλυπτικής θα επηρεάζει το έδαφος ή η δύναμη με την οποία θα τροποποιεί το έδαφος.
- **Brush Size:** Το μέγεθος που θα έχει η βούρτσα ένδειξης. Ο μέγιστος αριθμός με την χρήση του slider είναι 8192,0 αλλά μπορεί να πάρει τιμές μέχρι και το όριο 65536.0, αν η αλλαγή γίνει γράφοντας στο πλαίσιο μέσω πληκτρολογίου.
- **Brush Falloff:** Καθορίζει πόσο έντονα είναι τα άκρα της βούρτσας. Στο όριο 0,0 σημαίνει ότι θα έχει πλήρες αποτέλεσμα σε όλη την επιφάνεια. Στο όριο 1,0 θα έχει πλήρες αποτέλεσμα μόνο στο κέντρο και το αποτέλεσμα θα μειωθεί, σε ολόκληρη την περιοχή του μέχρι και τις άκρες.

Οι ιδιότητες που είναι μοναδικές στο Noise Tool είναι:

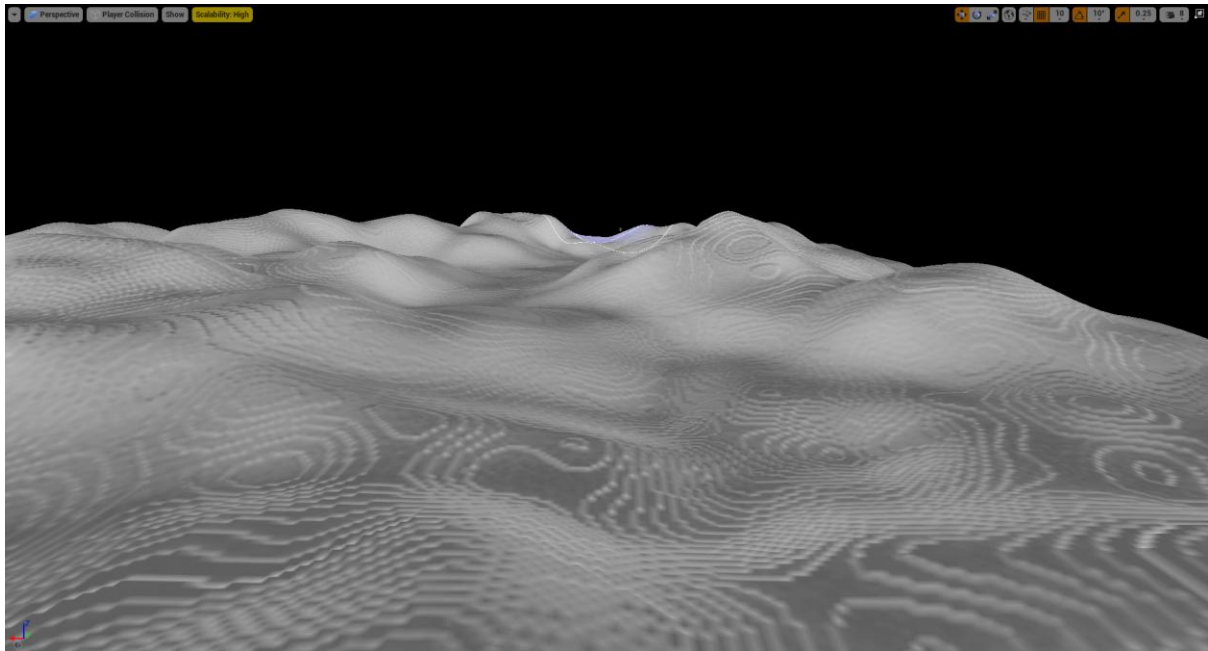
- **Noise Mode:** Noise ή θόρυβος είναι ο τυχαίος τρόπος με τον οποίο θα γίνει η αλλαγή του heightmap με την προσθήκη λοφίσκων ή υπόσκαφων επιφανειών.
 - ❖ **Add:** Ο θόρυβος θα μπορεί μόνο να αυξήσει το heightmap
 - ❖ **Sub:** Ο θόρυβος θα μπορεί μόνο να μειώσει το heightmap
 - ❖ **Both:** Ο θόρυβος θα μπορεί να μειώσει και να αυξήσει το heightmap
- **Noise Scale:** Το μέγεθος του φίλτρου θορύβου που χρησιμοποιείται. Τιμές κοντά στο 0,0 θα προκαλέσουν αυξημένο ποσοστό θορύβου και τιμές κοντά στο 256 θα προκαλέσουν μικρότερη αλλαγή στο heightmap.

Το Noise Tool θα δώσει την πρώτη ένδειξη του, πως θα μοιάζει το έδαφος και για αυτόν τον λόγο γίνεται να δοθεί μία μεγάλη τιμή στο brush size, μιας και η ακρίβεια και η λεπτομέρεια, θα προέλθει από την χρήση των άλλων εργαλείων στα επόμενα βήματα. Το tool strength χρειάζεται να έχει μία αρκετά μικρή τιμή, αλλιώς θα υπάρχουν απότομες αλλαγές στο heightmap. Οι τιμές στο noise mode και noise scale μπορούν να αλλάξουν ανάλογα με το αποτέλεσμα που θέλει ο κάθε σχεδιαστής. Στο edit layer, με δεξί κλικ>create γίνεται να προστεθεί ένα καινούριο layer που θα είναι αποκλειστικά για την σχεδίαση με το Noise Tool.



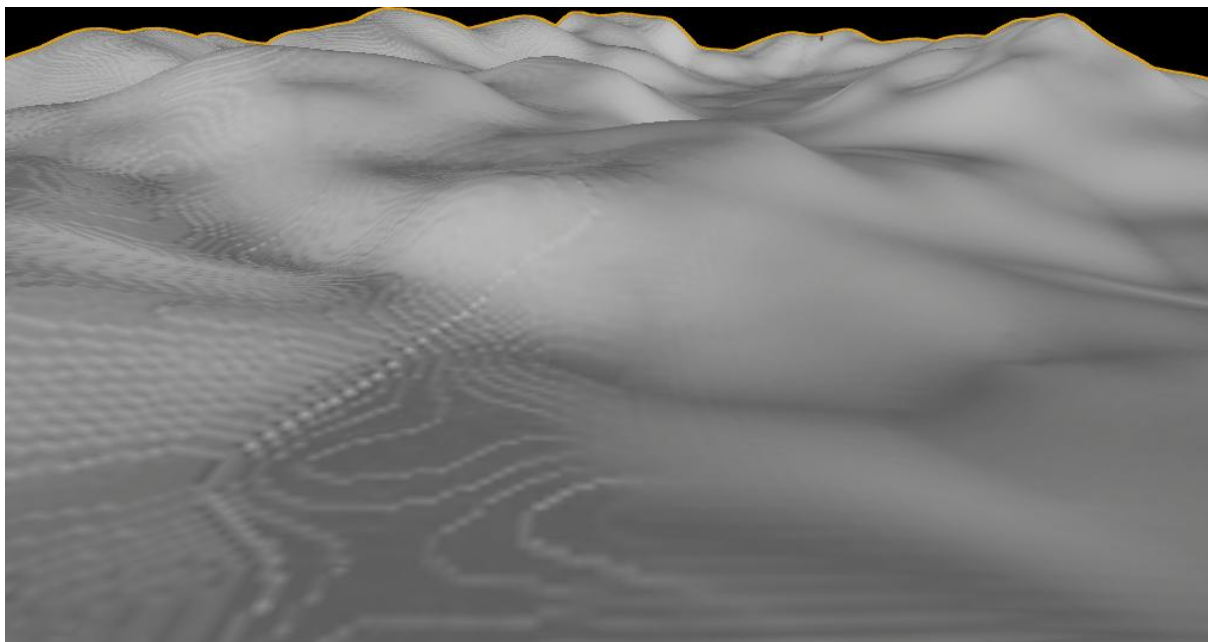
Εικόνα 4.3 Noise tool Settings και Edit layer

Για να διακρίνεται πιο εύκολα η αλλαγή του εδάφους στον σχεδιαστή, στο view mode μπορούν να επιλεγθούν οι επιλογές player collision ή reflection. Σε περίπτωση που χρειάζονται να προστεθούν λοφίσκοι μεγαλύτερης κλίμακας ο σχεδιαστής έχει δύο επιλογές. Η πρώτη είναι, να χρησιμοποιήσει το εργαλείο Sculpt, που προσθέτει ή αφαιρεί ύψος στο έδαφος με ομοιόμορφο τρόπο ή να συνεχίσει με το Noise Tool, αλλά με το Noise Mode να είναι Add. Η ανομοιομορφία που προσφέρει το Noise Tool θα δώσει ένα πιο ρεαλιστικό αποτέλεσμα, αλλά οι προτιμήσεις πάντα βασίζονται και στο αποτέλεσμα που θέλει ο καθένας. Κατά την αλλαγής του heightmap συνίσταται η κάμερα του σχεδιαστή να είναι κοντά στο έδαφος ή στο σημείο που θα είναι κάμερα του παίκτη. Αυτό δίνει στον σχεδιαστή μία καθαρή ματιά, στο τι θα βλέπει ο παίκτης και μπορεί να κρίνει άμεσα, αν ένα σημείο του εδάφους χρειάζεται κάποια αλλαγή. Με την ίδια λογική ο σχεδιαστής δεν χρειάζεται να δώσει μεγάλη έμφαση στα σημεία τα οποία δεν θα είναι ορατά. Όταν το έδαφος είναι σε ένα επιθυμητό στάδιο και πριν γίνουν περαιτέρω αλλαγές στο heightmap, μπορούν να παρατηρηθούν περιοχές στο έδαφος, με περίεργη γεωμετρία που θα πρέπει να διορθωθεί.



Εικόνα 4.4 Κακή γεωμετρία εδάφους

Το κατάλληλο εργαλείο για την αφαίρεση κακής γεωμετρίας και εξομάλυνσης της επιφάνειας του εδάφους είναι το Smooth Tool, το οποίο προσπαθεί να ισορροπήσει το heightmap, με βάση του μέσου όρου ύψους των τριγώνων που απαρτίζουν το έδαφος. Αν το Tool Strength έχει μεγάλη τιμή, μεγάλο μέρος του heightmap θα χαθεί, οπότε για σημεία που απαιτούν ραγδαίες αλλαγές από το Smooth Tool, θα πρέπει να μειωθεί το μέγεθος της βούρτσας για να είναι πιο ακριβής και να μην χαθούν πιθανόν λεπτομέρειες. Όπως με Noise Tool το ίδιο και με το Smooth Tool καλό είναι να αποθηκεύεται η πληροφορία στο δικό του layer.



Εικόνα 4.5 Εξομάλυνση heightmap με το Smooth Tool

Έχοντας ολοκληρώσει και το κομμάτι της εξομάλυνσης πλέον μπορεί να προστεθούν παραπάνω λεπτομέρειες στο έδαφος. Η πρόσθεση λεπτομέρειας είναι το τμήμα της διαδικασίας τροποποίησης του heightmap, που απαιτεί αρκετό πειραματισμό για ένα επιθυμητό αποτέλεσμα. Ταυτόχρονα ο σχεδιαστής θα χρειάζεται να εναλλάσσει τα εργαλεία τα οποία χρησιμοποιεί, ανάλογα με την επιφάνεια του εδάφους και για αυτό είναι χρήσιμο να αξιοποιήσει την λειτουργία του edit layer. Μία πιθανή ενέργεια που μπορεί να βοηθήσει στην καλύτερη διαμόρφωση του εδάφους είναι να τονίσει τις κορυφές και τις σιλουέτες των βουνών ή των λοφίσκων με το Sculpt Tool. Το σημαντικό βήμα που δεν πρέπει να παραλειφθεί αν το επιθυμητό αποτέλεσμα είναι ένα σχετικά ρεαλιστικό έδαφος, είναι η χρήση του Erosion Tool που μιμείται την διάβρωση του εδάφους από τον αέρα και τον ήλιο.

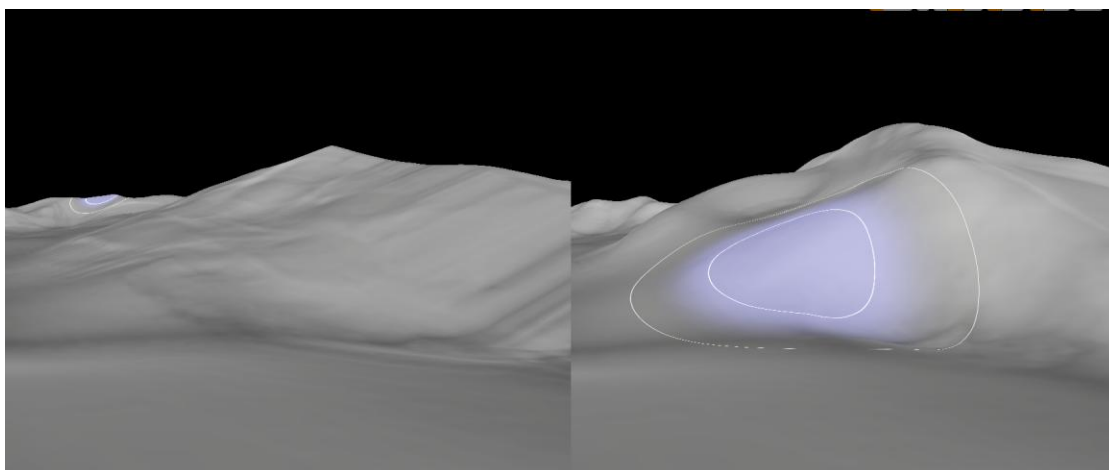
Erosion Tool Settings:

Threshold-Κατώφλι: Η ελάχιστη διαφορά ύψους που απαιτείται για την εφαρμογή της διάβρωσης. Μικρές τιμές θα οδηγήσουν σε εντονότερη διάβρωση.

Surface Thickness-Πάχος επιφάνειας: Το πάχος της επιφάνειας που πρέπει να έχει το έδαφος για να λειτουργήσει το φαινόμενο διάβρωσης.

Iterations-Επαναλήψεις: Ο αριθμός των επαναλήψεων που πραγματοποιούνται για την εφαρμογή της διάβρωσης. Μεγαλύτερες τιμές έχουν ως αποτέλεσμα περισσότερα επίπεδα διάβρωσης.

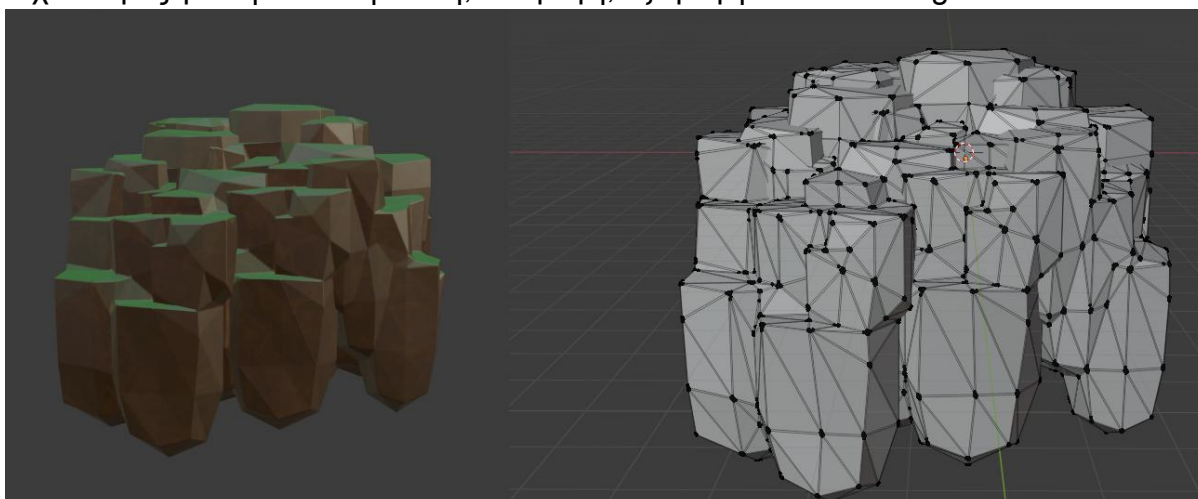
Το Erosion Tool χρησιμοποιεί την κλίση του εδάφους για να προσθέσει λεπτομέρεια. Σε απότομες κλίσεις το εφέ θα είναι πιο έντονο. Ταυτόχρονα προσθέτει και θόρυβο, που μπορεί επίσης να ρυθμιστεί, έτσι ώστε το αποτέλεσμα να είναι πιο αληθοφανές. Περνώντας τις επιφάνειες με τα παραπάνω εργαλεία και αλλάζοντας συνεχώς τις ρυθμίσεις, μπορούν να προκύψουν ποικίλα είδη αποτελεσμάτων. Αν ο σχεδιαστής έχει παράλληλα ως οδηγό κάποιες αναφορές που μπορούν να τον καθοδηγήσουν, αυτή η διαδικασία βελτιστοποιείται από θέμα χρόνου, ενώ ο σχεδιαστής εξοικειώνεται με το περιβάλλον που προσπαθεί να χτίσει. Τέλος κατά την διάρκεια των αλλαγών, στο edit layer δίπλα από κάθε επίπεδο υπάρχει μία ένδειξη τιμής Alpha που μπορεί να πάρει τιμές από το -1 έως το 1. Αυτή είναι μία δυναμική τιμή που αλλάζοντας την το αντίστοιχο layer θα έχει μεγαλύτερο ή μικρότερο αντίκτυπο στο έδαφος.



Εικόνα 4.6 Έδαφος με και χωρίς Διάβρωση.

4.2 3D assets

Αρκετές από τις τεχνολογίες που χρησιμοποιούνται στην κατασκευή ηλεκτρονικών παιχνιδιών είναι ευρέως διαδεδομένες και σε άλλους τομείς, που είναι εξίσου εξαρτημένοι από την δημιουργία 3D γραφικών όπως κινηματογράφος και ψηφιακή τέχνη. Αυτό που είναι κοινό μεταξύ τους είναι η έννοια των 3D asset ή τρισδιάστατων μοντέλων. Σκοπός τους είναι ο εμπλουτισμός της σκηνής ή του κόσμου που θα εμπεριέχονται, δίνοντας έναν καινούριο χαρακτήρα, αφαιρώντας την μονοτονία που προϋπήρχε. Η ενασχόληση κάποιου με την δημιουργία 3d asset είναι ένα βασικό κομμάτι ανάπτυξης ηλεκτρονικών παιχνιδιών και ψηφιακής τέχνης, το οποίο απαιτεί την εκμάθηση τρίτων λογισμικών όπως Blender, Zbrush, Maya, Houdini και Cinema 4D. Οι τεχνικές πίσω από όλα τα 3D modeling λογισμικά ποικίλουν, αλλά σαν βασική δομή έχουν την αλλαγή τοπολογίας κορυφών, για την αναπαράσταση ενός ψηφιακού μοντέλου. Το κάθε λογισμικό έρχεται και με τα δικά του εργαλεία προσφέροντας ένα χέρι βοήθειας στον σχεδιαστή, έχοντας παράλληλα βέλτιστες τεχνολογίες για την αποθήκευση,εισαγωγή, εξαγωγή και rendering.

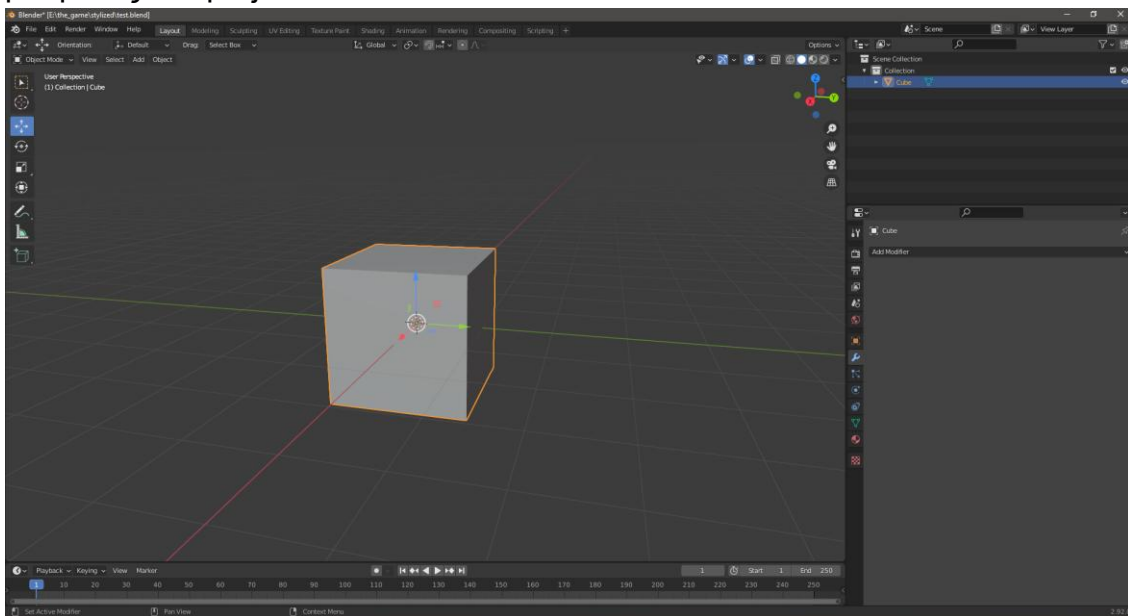


Εικόνα 4.7 3D μοντέλο

4.2.1 Κατασκευή μοντέλων

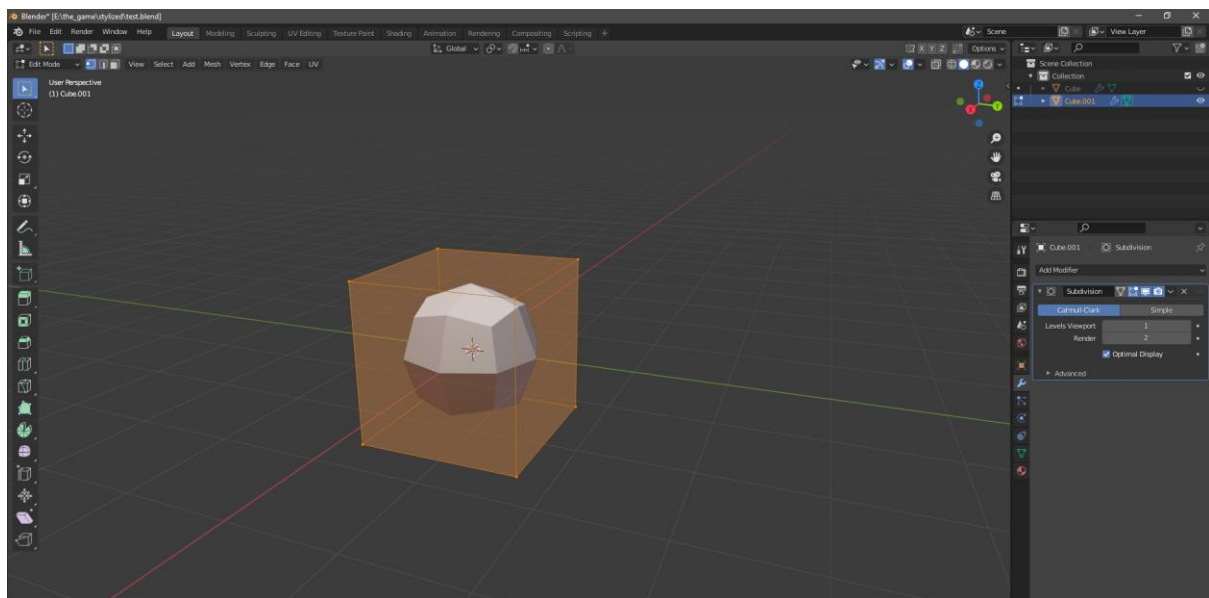
Ένα 3D μοντέλο είναι αποτέλεσμα μελετημένης και συχνά επιβλεπόμενης προσπάθειας από τους σχεδιαστές, να μεταφέρουν τις ιδέες των concept artist σε τρισδιάστατη μορφή. Θα πρέπει να μην υπάρχει σχεδιαστικό χάσμα ανάμεσα στα 3D assets έτσι ώστε το τελικό αποτέλεσμα να είναι όσο πιο οργανικό γίνεται. Στο στάδιο αυτό το παιχνίδι αρχίζει και αποκτά βάθος, μιας και ο κόσμος αρχίζει να κατοικείται από τα 3D assets. Η διαδικασία του εμπλουτισμού του παιχνιδιού μπορεί να δώσει μία αρκετά λεπτομερειακή ματιά στους σχεδιαστές, επιτρέποντάς τους να πάρουν άμεσα δημιουργικές αποφάσεις. Αρκετές φορές ένα επίπεδο ή ένα συγκεκριμένο σημείο στο παιχνίδι, μπορεί να περάσει από πολλές φάσεις τροποποίησης και αλλαγής ώστε να βρεθούν τα απαραίτητα assets για ένα επιθυμητό αποτέλεσμα.

Στο πρόγραμμα Blender ο σχεδιαστής έχει την δυνατότητα να μετατρέψει απλά γεωμετρικά σχήματα σε πολύπλοκα τρισδιάστατα μοντέλα, με την βοήθεια των εργαλείων που του προσφέρει το λογισμικό. Στο πάνω μέρος του menu υπάρχουν δέκα σελίδες που αντιστοιχούν σε ενέργειες που μπορεί να χρειαστεί να εκτελέσει ο σχεδιαστής. Στην περίπτωση των 3D μοντέλων τα βασικότερα παράθυρα είναι: α) το Layout που είναι υπεύθυνο για την προβολή του μοντέλου αλλά και της πρόσθεσης τροποποιητών, β) το Modeling, που επιτρέπει στον σχεδιαστή να αλλάξει τις κορυφές, τις άκρες και τις πλευρές του μοντέλου και γ) το παράθυρο Sculpting, που προσφέρει μία πιο δημιουργική πλευρά στην όλη διαδικασία αφήνοντας τον 3D artist να πλάσει το μοντέλο σε οποιαδήποτε μορφή επιθυμεί. Στην δεξιά μεριά της οθόνης διεπαφής υπάρχουν δύο παράθυρα. Το ένα δίνει πληροφορίες για το ποιά αντικείμενα κατοικούν την σκηνή και το δεύτερο παράθυρο που βρίσκεται ακριβώς από κάτω αφορά τις ρυθμίσεις του project.



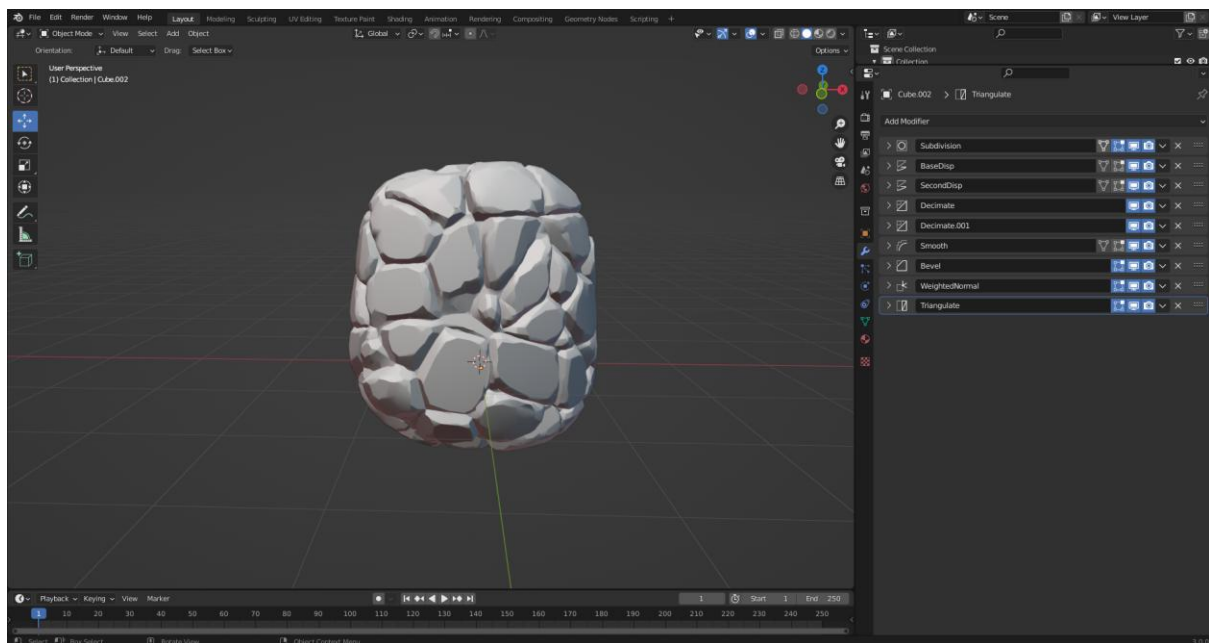
Εικόνα 4.8 Blender interface

Το Blender έχει πληθώρα από εργαλεία που μπορούν να βοηθήσουν τον 3D artist στην δημιουργία 3D μοντέλων. Δίνει την δυνατότητα κατασκευής άλλων θεμελιωδών συστατικών, που μπορούν να εκμεταλλευτούν οι σχεδιαστές στην πορεία ενός παιχνιδιού, όπως υφές. Μέσω του Blender είναι εφικτή η παραγωγή animation, για τους πιθανούς χαρακτήρες που θα έχει το παιχνίδι. Ένα από τα βασικότερα εργαλεία του Blender και άλλων 3D λογισμικών είναι οι τροποποιητές, που μπορούν να προστεθούν στα μοντέλα, με σκοπό να μεταβληθεί η γεωμετρία τους. Η πρόσθεση τους σε ένα αντικείμενο είναι μη καταστροφική, με αποτέλεσμα ανά πάσα στιγμή να μπορούν να προστεθούν ή να αφαιρεθούν, χωρίς να τροποποιηθεί η γεωμετρία του μοντέλου, πριν την ολοκληρωτική τους οριστικοποίηση. Με αυτόν τον τρόπο ο σχεδιαστής έχει την ικανότητα να αλλάξει τις κορυφές, τις πλευρές και τις άκρες, του μοντέλου και να δει άμεσα το τελικό αποτέλεσμα.



Εικόνα 4.9 Εφαρμογή modifier

Στην παραπάνω εικόνα φαίνεται ένας κύβος που του έχει προστεθεί ένας subdivision surface modifier για την υποδιαίρεση των κορυφών. Στο εσωτερικό του κύβου φαίνεται το τελικό αποτέλεσμα, αλλά ο κύβος παραμένει στην αρχική του τοπολογία. Με αυτόν τον τρόπο αλλάζοντας τις συντεταγμένες στις τέσσερις, μπορεί να παραχθεί μία γκάμα από ποικίλα αποτελέσματα. Έχοντας αυτό το μοντέλο σαν γονέα χωρίς να έχουν οριστικοποιηθεί οι modifiers, μπορεί να παραχθούν αντίγραφα του που θα κληρονομήσουν του modifiers του αρχικού μοντέλου. Επίσης οι τροποποιητές χρησιμοποιούνται για την μείωση χρόνου σε διαδικασίες που ο σχεδιαστής θα έπρεπε να κάνει χειρωνακτικά, όπως η δημιουργία αντιγράφου του μοντέλου με μικρότερο πλήθος κορυφών ή το low poly του αρχικού μοντέλου.



Εικόνα 2.10 Συνδυασμός modifier.

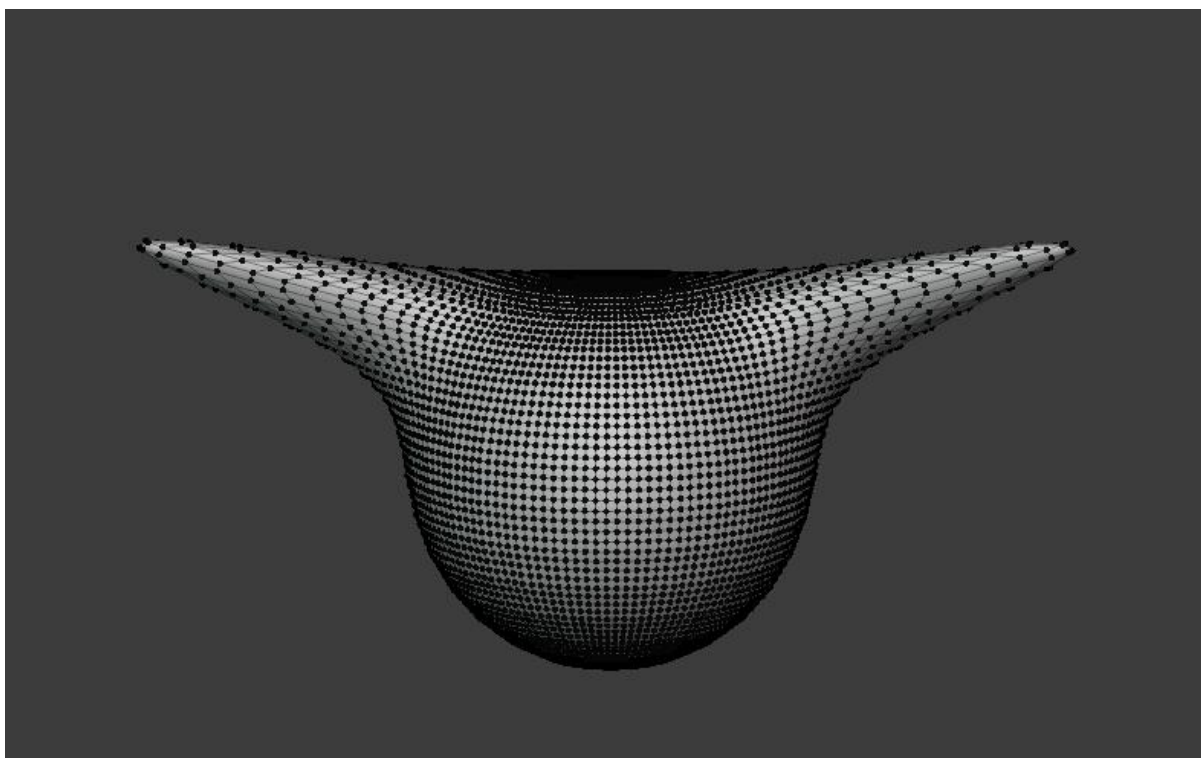
Η μεθοδολογία πίσω από κατασκευές 3D μοντέλων σε λογισμικά είναι αμεσα βασισμένη στον τύπο και στυλ του παιχνιδιού και απαιτεί διαφορετική προσέγγιση. Στην παρούσα περίοδο τα δυο διασημότερα στυλ παιχνιδιών από θέμα εμφάνισης και γραφικών είναι το ρεαλιστικό και τα στυλιζαρισμένο. Το ρεαλιστικό προσπαθεί να κάνει τον κόσμο να μοιάζει όσο το δυνατόν πιο κοντά στην πραγματικότητα και το stylized ή στυλιζαρισμένο, δίνει περισσότερη έμφαση στα χρώματα και στην απλούστερη γεωμετρική σύσταση των μοντέλων. Για κάποιον που ασχολείται για πρώτη φορά με γραφικά και 3D modeling, θα του φανεί πιο εύκολο να ξεκινήσει φτιάχνοντας απλά μοντέλα, κοντά στο στυλ των στυλιζαρισμένων παιχνιδιών, μιας και το κεντρικό χαρακτηριστικό τους είναι το σχετικά μικρό πλήθος από κορυφές, που έχουν, με αποτέλεσμα να μην χρειάζεται ο σχεδιαστής να δώσει ιδιαίτερη βάση στην βελτιστοποίησή τους, για γρηγορότερη επεξεργασία.



Εικόνα 4.11 Στυλιζαρισμένα γραφικά

4.2.2 Γλυπτική μοντέλων

Σε περίπτωση που ο σχεδιαστής κρίνει πως το μοντέλο χρειάζεται παραπάνω λεπτομέρεια που δεν είναι δυνατό να την παράγουν οι τροποποιητές θα περάσει στην γλυπτική και θα προσθεσει ο ίδιος τις μικρές λεπτομέρειες εκεί που θέλει. Καθώς η 3D γλυπτική είναι ένα είδος ψηφιακής τέχνης, είναι αρκετά δύσκολο για κάποιον να εξοικειωθεί άμεσα με αυτήν. Το σημαντικό που πρέπει να σημειωθεί είναι ότι στην γλυπτική τα εργαλεία χρειάζονται επαρκή αριθμό από κορυφές, για να υπάρχει ένα καλό και λεπτομερικό αποτέλεσμα. Υπάρχουν αρκετοί τρόποι για να προστεθεί γεωμετρία σε ένα μοντέλο, εκ των οποίων ο ένας είναι με τον τροποποιητή Subdivision surface στο Blender, ή το divide στο ZBrush. Σε μικρό βαθμό γλυπτικής αυτός ο τρόπος αυτός εξυπηρετεί αλλά δεν έχει την δυνατότητα αύξησης των κορυφών κατα την ίδια την διαδικασία της γλυπτικής, που σημαίνει ότι αν το μοντέλο τροποποιηθεί σε ακραίο βαθμό, θα γίνουν ορατά τα σημεία που οι κορυφές έχουν τραβηχτεί.



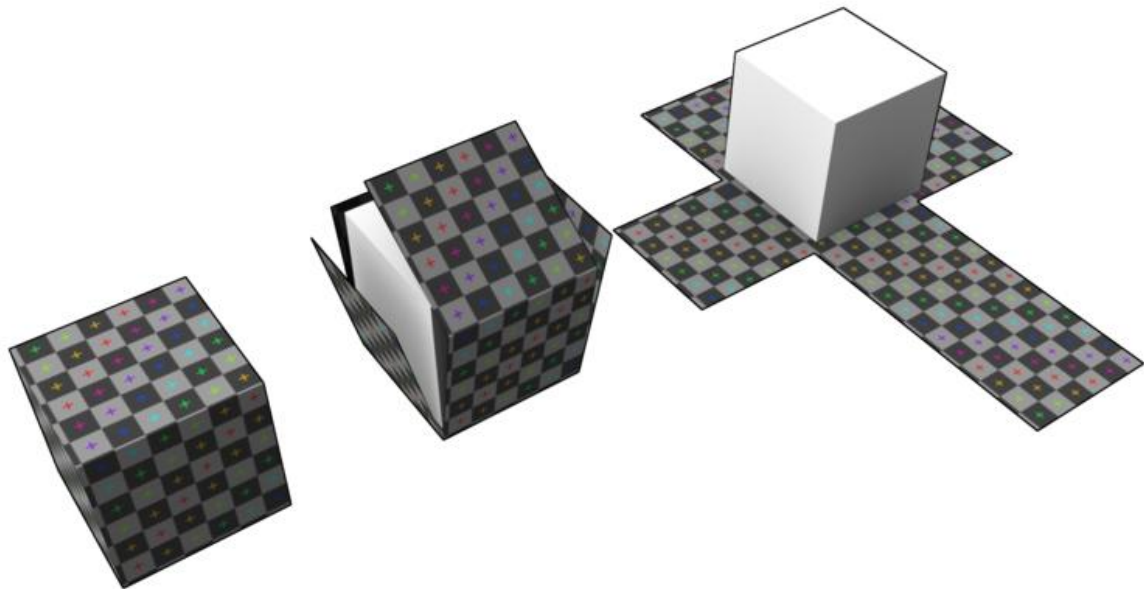
Εικόνα 4.12 Κακή τοπολογία

Για να αποφευχθεί αυτό το αποτέλεσμα το Blender και το ZBrush έχουν εφαρμόσει μεθόδους αύξησης τοπολογίας μοντέλων κατά την διαδικασία γλυπτικής, ανάλογα με την περιοχή που περιεργάζεται ο σχεδιαστής. Στο Blender αυτή η λειτουργία ονομάζεται *dyntopo* και με βάση την περιοχή που γίνεται η γλυπτική προστίθενται επιπλέον κορυφές για μία ομαλότερη επιφάνεια. Στο ZBrush η πρόσθεση λεπτομέρειας γίνεται με έναν προσαρμοστικό τρόπο μέσω του *sculptris pro mode* ή με την λειτουργία *dynamesh*. Η διαφορά μεταξύ των δύο είναι ότι το *sculptris pro* θα προσθέσει λεπτομέρεια μόνο στο σημείο που τροποποιείται όπως το *Dyntopo* **ενώ** το *dynamesh* θα επανεκτιμηθεί ολόκληρο το μοντέλο και μετέπειτα θα προστεθεί παραπάνω τοπολογία. Με παρόμοιο τρόπο λειτουργεί στο Blender το *voxel remesh*, τεχνική που αυτόματα επανασχεδιάζει το μοντέλο με περισσότερη ή λιγότερη ομοιόμορφη λεπτομέρεια, κάνοντας χρήση του open source λογισμικού *OpenVDB*.

4.2.3 UV χαρτογράφηση

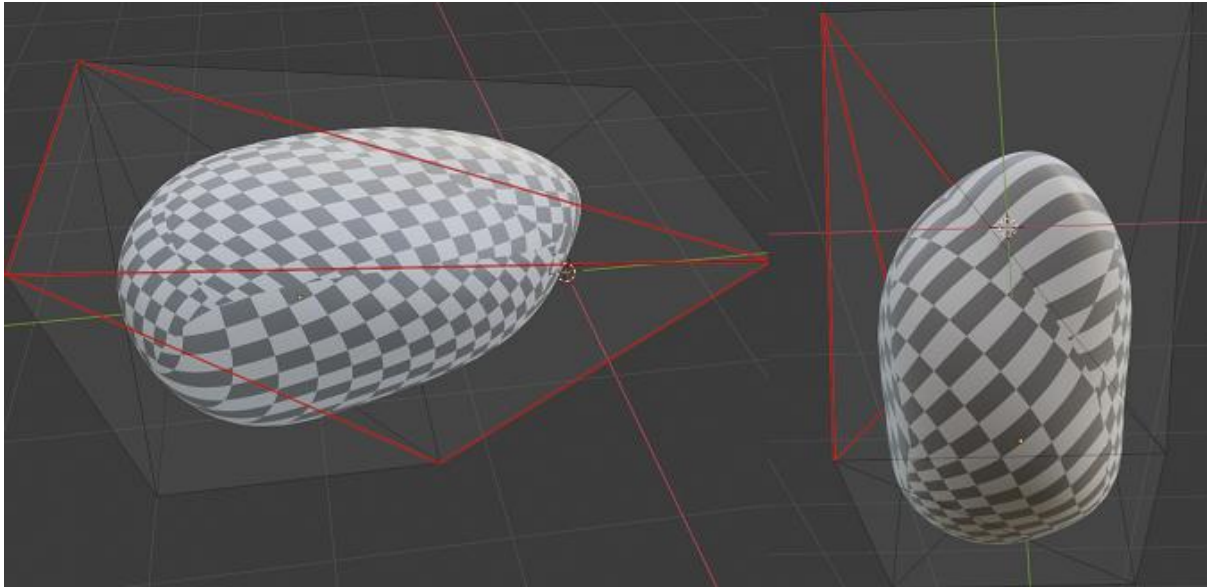
Μετά από την ολοκληρωτική επεξεργασία του μοντέλου από μεριάς σχεδίασης και τροποποίησης δεν συνίσταται να χρησιμοποιηθεί άμεσα από τον σχεδιαστή στην μηχανή γραφικών. Το γεγονός ότι προστέθηκε τροπολογία για καλύτερη ανάλυση, είναι ταυτόχρονα ο λόγος, που το μοντέλο είναι μη κατάλληλο, μιας και θα είναι αρκετά κοστοβόρο για επεξεργασία. Το μοντέλο που θα προστεθεί στην μηχανή στην πραγματικότητα είναι ένα αντίγραφο χαμηλότερης ανάλυσης και είναι σημαντικό να

μην υπάρχει καμία ορατή διαφορά με το αρχικό μοντέλο. Αυτό γίνεται με την προσθήκη υφών που έχουν δημιουργηθεί από το αρχικό μοντέλο και περνάνε στο αντίγραφο, επιτρέποντας την δημιουργία υλικού ή material για να κάνει πιο οργανική την εμφάνισή του 3D asset στο παιχνίδι. Οι υφές που θα προστεθούν είναι δισδιάστατες εικόνες, που θα πρέπει να προβάλλονται σε τρισδιάστατα αντικείμενα και θα πρέπει να υποστούν επεξεργασία UV χαρτογράφησης (UV mapping). Ένας UV χάρτης, είναι η αναπαράσταση ενός τρισδιάστατου μοντέλου σε δύο διαστάσεις, ώστε να είναι εφικτό οι 2D εικόνες να μπορούν να τυλιχτούν γύρω του. Η διαδικασία δημιουργίας ενός χάρτη UV ονομάζεται ξετύλιγμα UV.



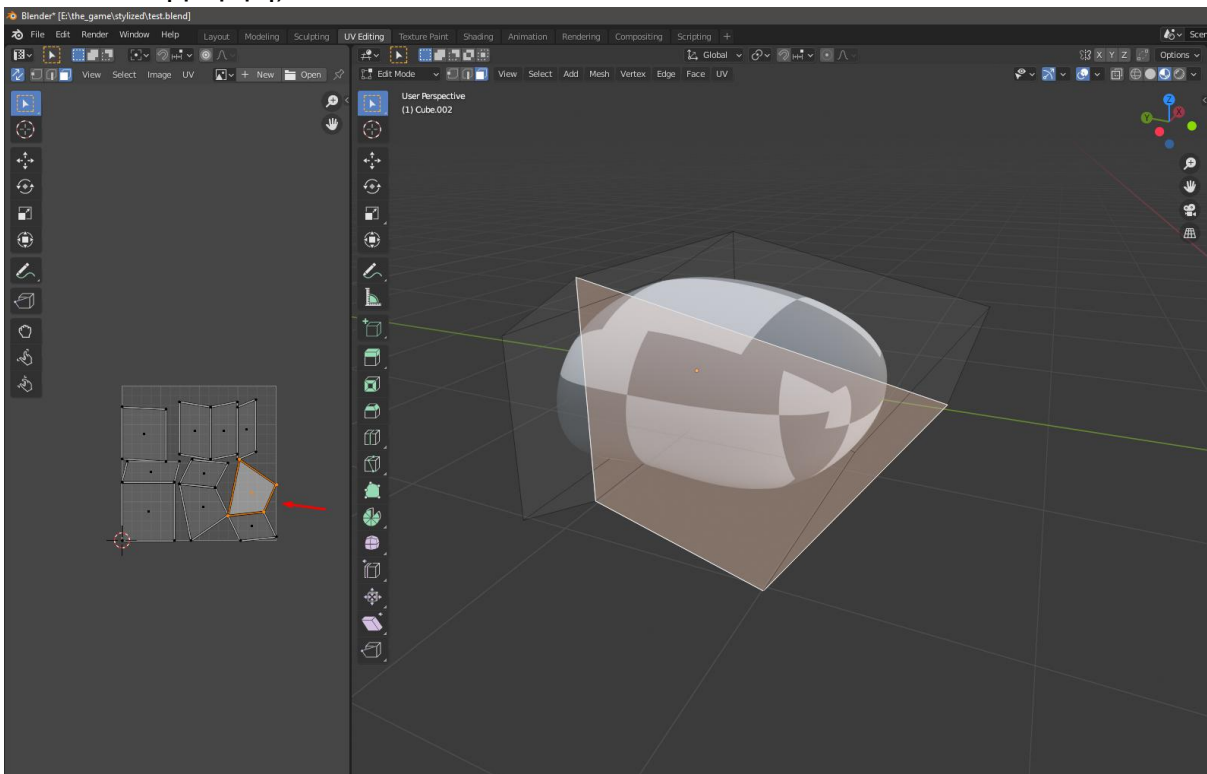
Εικόνα 4.13 UV Unwrap

Σε απλά γεωμετρικά σχήματα ή σε μοντέλα με μικρό δείκτη κορυφών η χαρτογράφηση UV είναι αρκετά εύκολη. Στα μοντέλα που έχουν αρκετές εναλλαγές στην γεωμετρία τους, πρέπει να τεθούν ραφές ή seams που περικλείουν ένα συγκεκριμένο μέρος της επιφάνειας του μοντέλου, ώστε να είναι πιο ακριβής ο υπολογισμός του UV χάρτη. Για την βοήθεια και καθοδήγηση του καλλιτέχνη ως προς την ποιότητα του UV, τα 3D προγράμματα, απεικονίζουν στις επιφάνειες του αντικειμένου κάποιους δείκτες, όπως τετράγωνα ή αριθμοί, που ο καλλιτέχνης μπορεί να έχει σαν οδηγό. Μία ίση αναλογία στα τετράγωνα υποδηλώνει και καλή χαρτογράφηση UV με αποτέλεσμα την καλύτερη τοποθέτηση της 2D εικόνας στο μοντέλο.



Εικόνα 4.15 Ραφές

Σε τέτοιες περιπτώσεις ο σχεδιαστής πρέπει να αποφασίσει ποιές θα είναι άκρες που θα ορίσει σαν ραφές για να μπορέσει να τροποποιήσει την υφή UV. Η διαδικασία αυτή είναι αρκετά χρονοβόρα, αλλά δεν μπορεί να παραλειφθεί διότι έχει άμεσο αντίκτυπο στην τελική μορφή του μοντέλου αφού προστεθούν υφές. Στο πρόγραμμα Blender υπάρχει μια γρήγορη και αποτελεσματική διαδικασία υπολογισμού της UV υφής με όνομα Smart UV projection. Ο υπολογισμός γίνεται με βάση την κλίση της κάθε γωνίας στο μοντέλο και το χωρίζει σε επιμέρους τμήματα (σε δισδιάστατη μορφή).



Εικόνα 2.16 UV επεξεργασία

4.3 Βελτιστοποίηση και επεξεργασία μοντέλου στην Unreal

Ο τρόπος και η διαδικασία ανάπτυξης ενός 3D μοντέλου είναι ένας σημαντικός παράγοντας στην πορεία ανάπτυξης ενός παιχνιδιού, καθώς αποτελεί την βάση για ένα οργανικό, αλλά και ταυτόχρονα, διασκεδαστικό με ενδιαφέρον κόσμο. Παράλληλα είναι ένα αρκετά τεχνικό κομμάτι, που απαιτεί επίβλεψη, για μία καλή ροή της ανάπτυξης ενός παιχνιδιού. Σε αρκετές περιπτώσεις υπήρξαν παιχνίδια με αρκετά “κακοφτιαγμένα” 3D μοντέλα όχι από άποψης αισθητικής, αλλά άλλων ποιοτικών χαρακτηριστικών, με το σημαντικότερο να είναι η πυκνή γεωμετρία τους. Σε προχωρημένα στάδια ανάπτυξης ηλεκτρονικών παιχνιδιών, είναι αρκετά κοστοβόρο, για μία ομάδα να δημιουργήσει ένα νέο ποιο λειτουργικό μοντέλο και να το τοποθετήσει ξανά στα σημεία κλειδιά, που προϋπήρχε ο πρόγονός του.

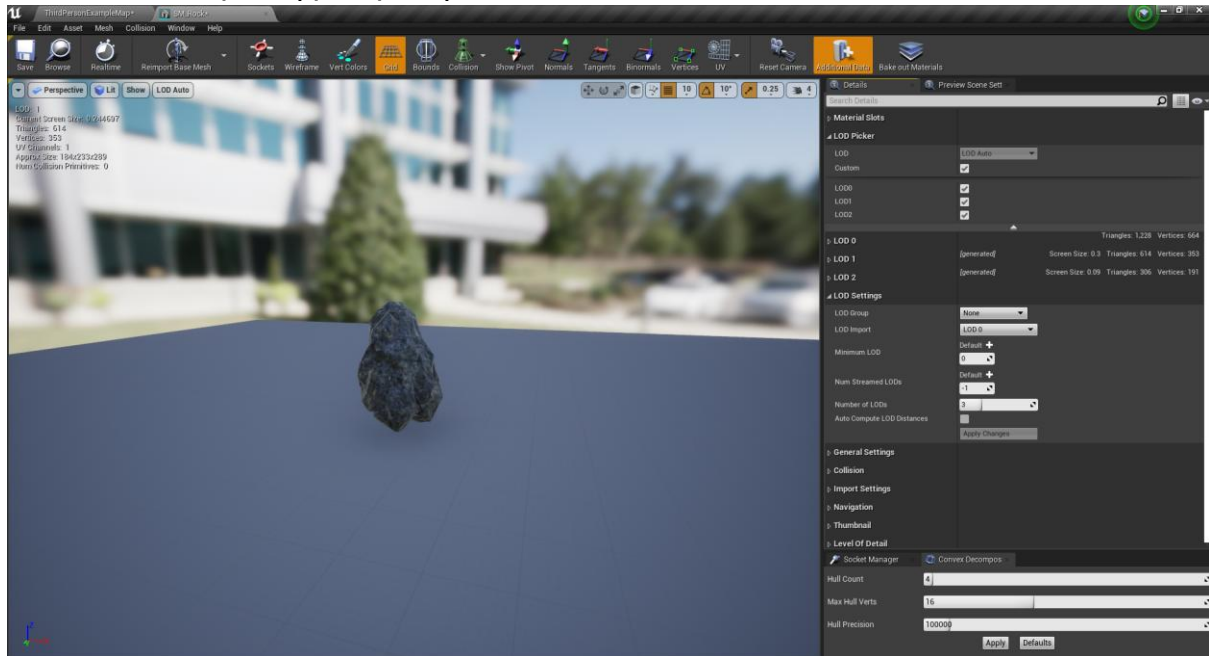
4.3.1 Επίπεδο λεπτομέρειας μοντέλου - LOD

Το γεγονός ότι ο αριθμός των κορυφών είναι ένας σημαντικός παράγοντας για μία καλή εκτέλεση ενός παιχνιδιού, οι σχεδιαστές μπορούν να το εκμεταλλευτούν στην δυνατότητα που τους δίνει η μηχανή γραφικών, σχετικά με το επίπεδο ανάλυσης του κάθε μοντέλου που έχει εισαχθεί. Γνωστό εργαλείο είναι το LOD (Level Of Detail), που επιτρέπει την δυναμική συγχώνευση πολυγώνων, σε αντικείμενα με στατικά πλέγματα, όπως είναι τα 3D assets. Αυτή η ενοποίηση του πλέγματος λειτουργεί υπολογίζοντας το μέγεθος της οπτικής διαφοράς που θα έχει, με την ένωση δύο κορυφών. Αν αυτή η ένωση δεν έχει μεγάλη επίπτωση στην προβολή του μοντέλου, το λογισμικό αποφασίζει την καταλληλότερη θέση, για να προσθέσει την καινούρια συγχωνευμένη κορυφή, αφαιρώντας τυχόν τρίγωνα που είχαν επίσης καταρρεύσει. Η διαδικασία θα συνεχίσει να επαναλαμβάνεται μέχρι να φτάσει ο αριθμός των τριγώνων στο απαιτούμενο ποσό, που μπορεί να οριστεί χειρωνακτικά για κάθε μοντέλο από τον σχεδιαστή ή αυτόματα από την μηχανή γραφικών.

Στην unreal engine στο παράθυρο content browser εμφανίζεται όλο το περιεχόμενο που έχει το συγκεκριμένο project, εκ των οποίων και τα 3D assets που έχουν γίνει import ή τα assets από τον φάκελο Starter Content. Τα τελευταία θα υπάρχουν μόνο αν κατά την δημιουργία του project, έχει οριστεί μέσω των settings. Τα 3D μοντέλα έχουν μία γαλάζια γραμμή στο κάτω μέρος του εικονιδίου τους, ώστε να ξεχωρίζουν από άλλες οντότητες που μπορεί να υπάρχουν στον παρόντα φάκελο.

Αν πατηθεί διπλό αριστερό κλικ σε ένα asset ο σχεδιαστής θα μεταφερθεί σε ένα νέο παράθυρο αποκλειστικά σχετιζόμενο με το τρέχον asset. Στην δεξιά μεριά υπάρχει το παράθυρο Details, που μέσω αυτού μπορούν να οριστούν και να τροποποιηθούν, ιδιότητες και χαρακτηριστικά του μοντέλου, με το μεγαλύτερο ποσοστό να είναι επικεντρωμένα στα LODs. Πριν την ανάθεση των LODs από τον σχεδιαστή, στην ρύθμιση LOD Group η unreal, δίνει επτά προρυθμισμένες ομάδες με

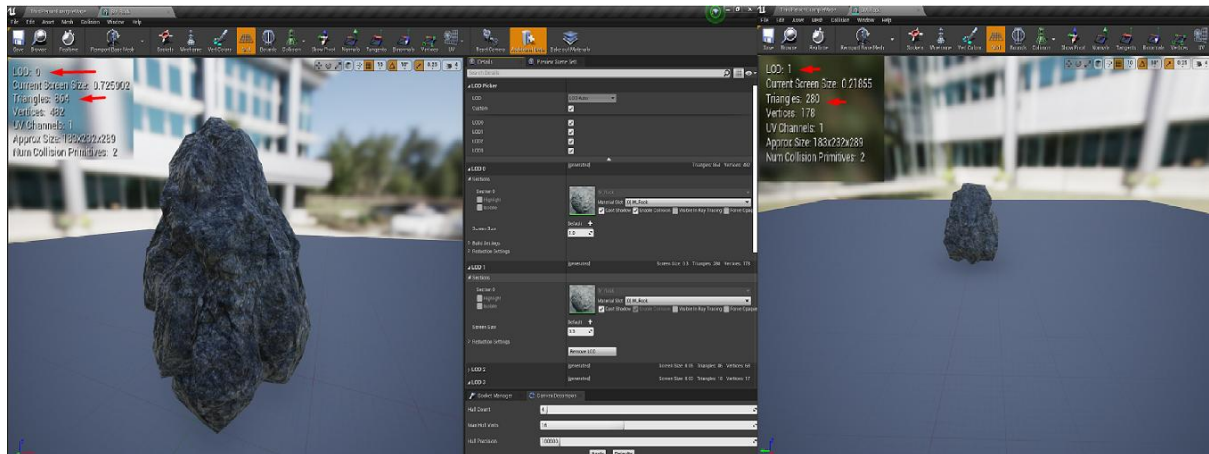
τα δικά τους επίπεδα λεπτομέρειας και μία κενή επιλογή, με την προϋπόθεση ότι θα την αξιοποιήσει ο σχεδιαστής με τις δικές του ρυθμίσεις. Η αρχική προϋπόθεση για όλα τα assets είναι να έχουν μόνο ένα LOD, το LOD0, που σημαίνει ότι δεν υπάρχει τροποποίηση γεωμετρίας ανάλογα την απόσταση της κάμερας. Η ρύθμιση Number of LODs επιτρέπει την αύξηση ή μείωση του αριθμού των επιπέδων λεπτομέρειας και με την ανάθεση τιμής False (unchecked το κουτί) του Auto Compute LOD Distances επιτρέπει στον σχεδιαστή να ορίσει χειρωνακτικά το μέγεθος που θα πρέπει να έχει το asset ώστε να γίνει η μετάβαση από το ένα LOD στο άλλο.



Εικόνα 2.9 LOD Settings

Ειδικότερα το κάθε LOD έχει μία κατηγορία Section. Σε αυτήν η βασικότερη ρύθμιση είναι το Cast Shadow. Ο λόγος που πρέπει να δοθεί βαρύτητα σε αυτό το σημείο είναι η παραπάνω μνήμη που μπορεί να χρειαστεί ο υπολογιστής, για να υπολογίσει τις σκιές του asset αν τα LODs δεν έχουν ρυθμιστεί σωστά. Στα LODs που το screen size είναι σχετικά μεγάλος αριθμός θα πρέπει για λόγους αληθοφάνειας και σωστής απεικόνισης να υπάρχουν προβαλλόμενες σκιές από τα μοντέλα. Όταν η κάμερα είναι πιο απομακρυσμένη από το μοντέλο μπορεί να παραλείπεται ο υπολογισμός σκιάς για βελτιστοποίηση του επιπέδου. Στην κατηγορία Reduction Settings μπορούν να βρεθούν ρυθμίσεις για την μείωση των τριγώνων ή των κορυφών του μοντέλου. Η λογική πίσω από την ανάθεση τιμών στο Percent Triangles είναι η ίδια με αυτή που ακολουθηθηκε για τον ορισμό των σκιών, λαμβάνοντας υπόψη την απόσταση του αντικειμένου από την κάμερα. Αν γίνει απότομη αλλαγή της τιμής η παραμόρφωση του μοντέλου θα είναι διακριτή από τον παίκτη. Για να αποφευχθούν τέτοιες περιπτώσεις είναι καλό να ορίζονται 4 LODs και πάνω με το καθένα να έχει μία μείωση 25 % στα τρίγωνα ή στις κορυφές. Μεγάλα μοντέλα όπως σπίτια ή πέτρες και δέντρα θα χρειαστεί να τα προσεγγίσει κάποιος με διαφορετικό τρόπο, σε σχέση με μικρότερα μοντέλα. Μεγάλα μοντέλα πρέπει να έχουν μικρότερες τιμές στο screen size σε αντίθεση με ένα μικρότερο μοντέλο που μετά από μερικά μέτρα δεν χρειάζεται να

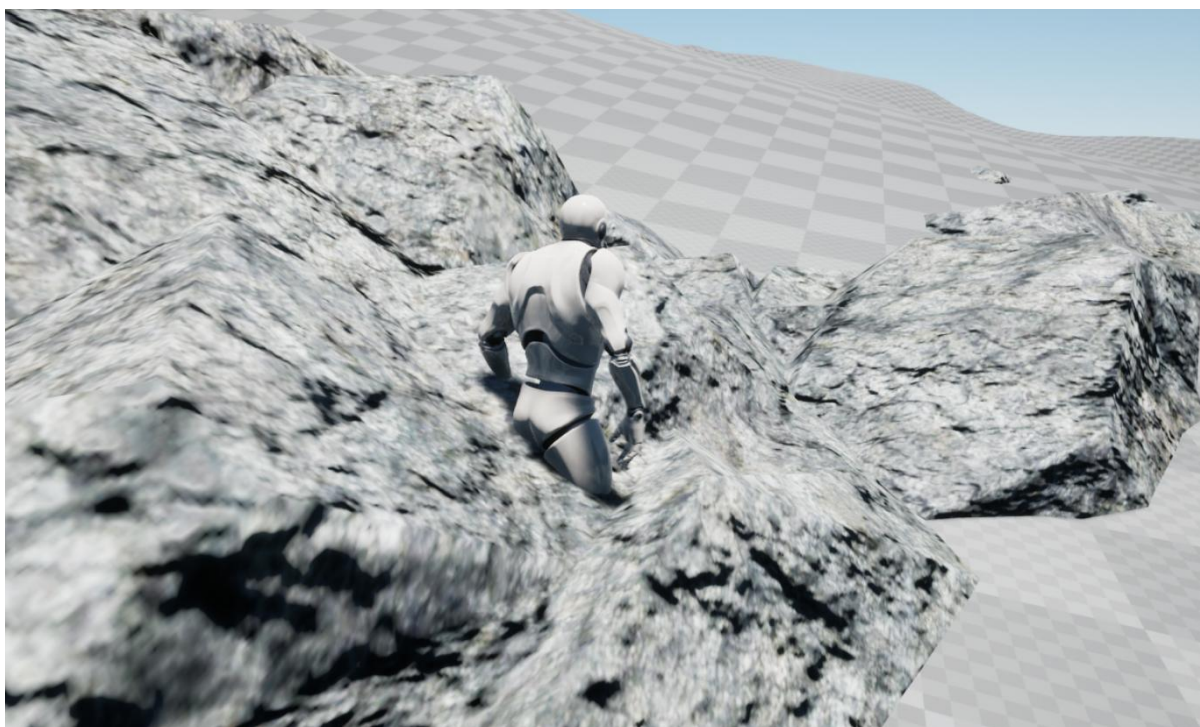
είναι ορατό. Για παράδειγμα ένας βράχος που αποτελεί ένα φυσικό εμπόδιο για τον παίκτη για να μην βγει εκτός χάρτη πρέπει να είναι ορατό από μακρινές αποστάσεις, σε αντίθεση με ένα πετραδάκι που θα είναι ορατό μόνο σε κοντινή απόσταση.



Εικόνα 2.10 Μείωση Τριγώνων με βάση τα LOD

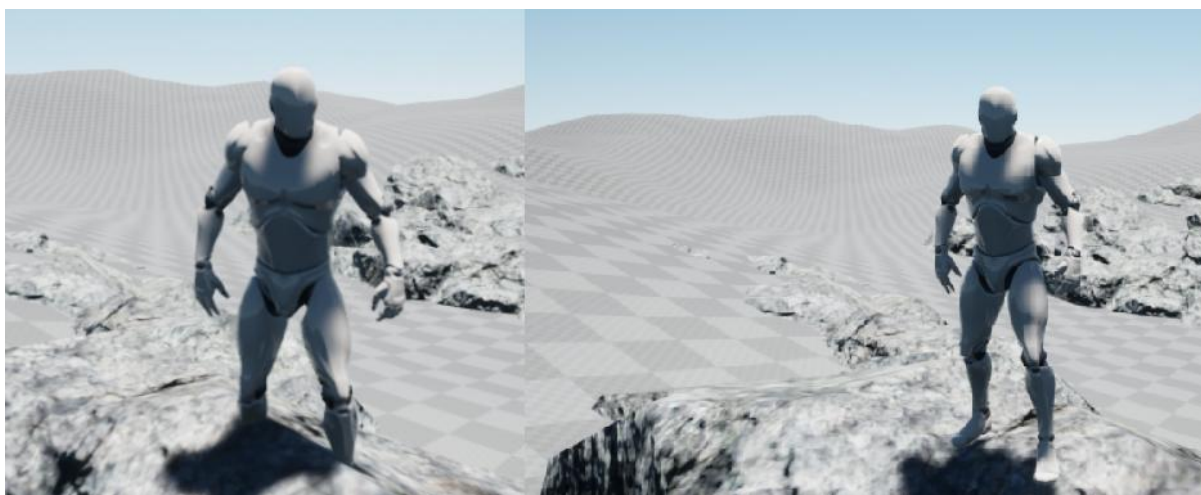
4.3.2 Σύγκρουση - Collision

Πατώντας το κουμπί Play, ανάλογα με το πρότυπο που έχει επιλεγεί για το είδος του παιχνιδιού (first person, third person), θα εμφανιστεί ένας χαρακτήρας που ο σχεδιαστής ή ο προγραμματιστής θα έχει τον έλεγχό του. Έχοντας την δυνατότητα να κινηθεί και να αντιδράσει με το περιβάλλον μπορεί να διαπιστώσει πως τα μοντέλα αδυνατούν να τον αποτρέψουν από το να προχωρήσει. Για παράδειγμα φυσικά εμπόδια όπως δέντρα ή βράχοι θα πρέπει να σταματάνε την κίνηση του χαρακτήρα αν έρθει σε επαφή με αυτά. Τα μοντέλα είναι προκαθορισμένα από την μηχανή να μην έχουν κάποιου είδους σύγκρουση ή collision, οπότε ο χαρακτήρας δεν επιτρέπεται από το να περάσει από μέσα τους. Η διαδικασία της δημιουργίας σύγκρουσης, λαμβάνει χώρα στο ίδιο παράθυρο που ορίστηκαν και τα LODs και μπορεί να γίνει μόνο με ενέργεια του σχεδιαστή.

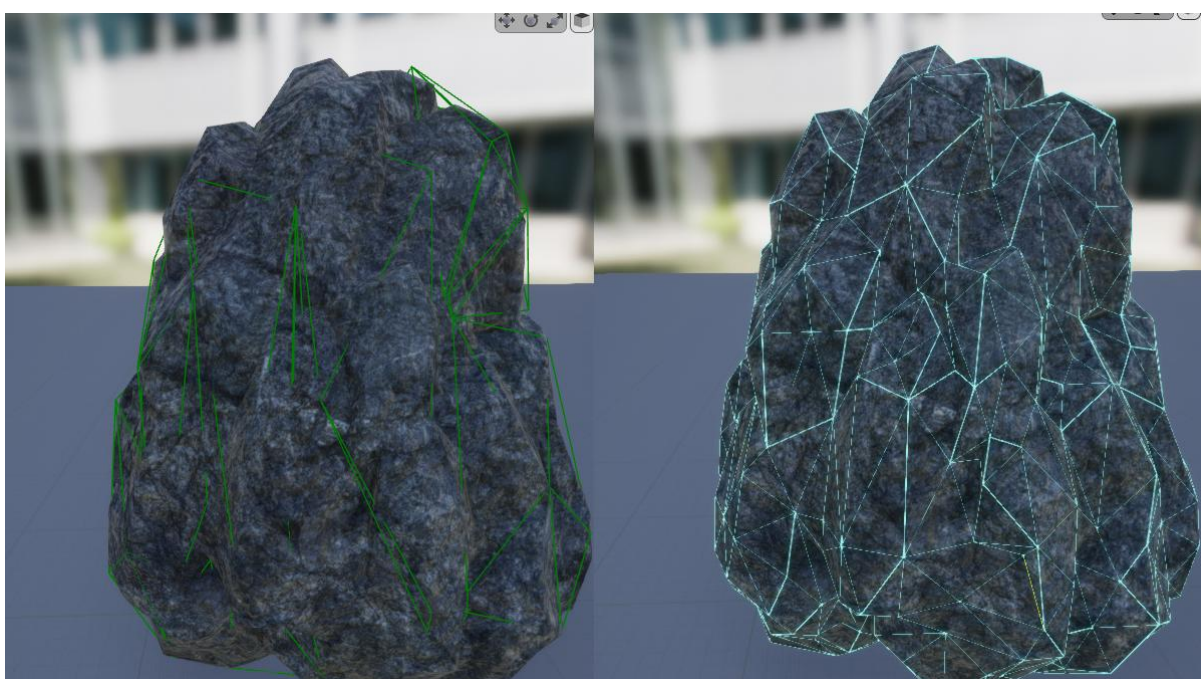


Εικόνα 2.13 Μοντέλα χωρίς σύγκρουση

Τα Collision Presets ορίζουν με ποιους δράστες θα μπορεί ο παίκτης να συγκρουστεί και με ποιους όχι. Κάτω από Object Type που ορίζει το είδος του αντικειμένου κατά την ώρα κίνησης ή σύγκρουσης, υπάρχουν πλαίσια ελέγχου που αναδεικνύουν ποιοι δράστες θα αγνοούν, επικαλύπτουν και θα μπλοκάρουν, τον παίκτη. Οι επιλογές Block All κι No Collision επιτρέπουν και αποτρέπουν αντίστοιχα την σύγκρουση του παίκτη και άλλων δραστών, με τον συγκεκριμένο μοντέλο, ενώ η επιλογή Custom, αφήνει τον σχεδιαστή να επιλέξει χειρωνακτικά τους δράστες που θα αντιδρούν με το συγκεκριμένο μοντέλο. Η μηχανή καταλαβαίνει αν ένας δράστης συγκρούεται με το αντικείμενο από την πληροφορία των τριγώνων που αποτελούν το πεδίο σύγκρουσης. Με την αλλαγή των τιμών της παραμέτρου Collision Complexity, μπορούν να δοθούν τρεις διαφορετικές τεχνικές, για την δημιουργία επιπέδου σύγκρουσης. Το Use Simple Collision As Complex δημιουργεί ένα μικρό πλήθος τριγώνων με αποτέλεσμα μερικές φορές να είναι ανακριβές, αν το μοντέλο είναι σχετικά μεγάλο σε μέγεθος, αλλά ταυτόχρονα είναι πιο βέλτιστο στον υπολογισμό. Αντίθετα το Use Complex As Simple είναι πολύ πιο ακριβές και χρησιμοποιείται κυρίως σε ογκώδες μοντέλα, που έχουν πολύπλοκη τοπολογία τριγώνων, με το μειονέκτημα ότι είναι κοστοβόρο αν χρησιμοποιηθεί επανειλημμένα μιάς και γίνεται επιπλέον υπολογισμός μεγάλου πλήθους τριγώνων.



Εικόνα 2.14 Ανακρίβια σύγκρουσης



Εικόνα 2.15 Simple και Complex Collision

4.3.1 Εξαφάνιση μοντέλου - Culling

Η ιδέα πίσω από την τεχνική απομάκρυνσης ή Culling είναι η επεξεργασία και αποτύπωση των γραφικών που είναι απαραίτητα για την δημιουργία μιας σκηνής. Η τεχνική απομάκρυνσης σαρώνει την σκηνή κατά την εκτέλεση και δημιουργεί ιεραρχία πιθανών μοντέλων που θα ζωγραφιστούν. Η μείωση του αριθμού των μοντέλων, μπορεί να προσεγγισθεί με τέσσερις διαφορετικούς τρόπους στην Unreal Engine 4, με τον καθένα, τελικά, να πετυχαίνει το επιθυμητό αποτέλεσμα απομάκρυνσης .

Distance Culling: Η πρώτη μέθοδος απομάκρυνσης, είναι με βάση την απόσταση του μοντέλου από την κάμερα και λειτουργεί σαν LOD λαμβάνοντας υπόψη τις τιμές

των πεδίων Min Draw Distance και Desired Max Draw Distance, που δηλώνουν την μικρότερη και μεγαλύτερη απόσταση, από την κάμερα αντίστοιχα, για την δημιουργία ή απομάκρυνσης του μοντέλου. Οι τιμές αυτών των πεδίων μπορούν να αλλάξουν για κάθε μοντέλο ξεχωριστά ή να οριστούν μαζικά για όσα μοντέλα επιλέξει ο σχεδιαστής. Αυτό καθιστά δυνατή την μαζική εξαφάνιση μεγάλου πλήθους αντικειμένων που δεν χρειάζεται να είναι ορατά από ένα σημείο και μετά. Με βάση την απόσταση λειτουργεί και το Cull Distance Volume (όγκος αποστάσεως απομάκρυνσης) που επιτρέπει την εξαφάνιση των μοντέλων τα οποία βρίσκονται μέσα σε αυτό, χρησιμοποιώντας ένα array που περιέχει τιμές για το μέγεθος του μοντέλου και την απόσταση απομάκρυνσης (cull distance). Αυτό είναι αρκετά χρήσιμο σε περιπτώσεις που χρειάζεται άμεση εξαφάνιση μοντέλου για εξοικονόμηση πόρων. Καλό παράδειγμα είναι τα μοντέλα που μπορεί να βρίσκονται μέσα σε ένα σπίτι. Όταν ο παίκτης απομακρυνθεί από αυτό δεν θα χρειαστεί να σχεδιαστούν από την GPU ή CPU.

Frustum Culling: Frustum culling ή απομάκρυνση κόλουρης χρησιμοποιεί το οπτικό πεδίο της κάμερας (field of view) για τον σχεδιασμό των αντικειμένων που είναι ορατά σε αυτό. Ταυτόχρονα το frustum culling χρησιμοποιεί και δύο επιπλέον επίπεδα που δεν είναι ορατά, το near clipping plane και το far clipping plane, που παίζουν τον ρόλο συνόρων και ενημερώνουν την κάμερα για την κοντινότερη και πιο μακρινή απόσταση, που με βάση αυτή θα πρέπει να σχεδιαστούν αντικείμενα. Ίσως και το πιο αποτελεσματικό είδος απομάκρυνσης, χρησιμοποιείται από την μηχανή με την δημιουργία ενός project και δεν χρειάζεται να δημιουργηθεί από τον προγραμματιστή.

Dynamic Occlusion: Σε τρισδιάστατες σκηνές όταν μοντέλα καλύπτονται από άλλα μεγαλύτερα σε όγκο και δεν έχουν κάποια βαρύτητα στην σκηνή, δεν χρειάζεται να τα ζωγραφίσει η κάρτα γραφικών ή ο επεξεργαστής. Για κάθε μοντέλο ή δράστη που βρίσκεται στην σκηνή υπάρχει και η κατάσταση ορατότητας του, που με κάθε δευτερόλεπτο θα καλείται για να ενημερώσει την CPU ή GPU αν πρέπει να το σχεδιάσει. Για να μειωθούν οι κλήσεις ελέγχου ορατότητας χρησιμοποιούνται ευρετικοί αλγόριθμοι αυξάνοντας τη συνολική αποτελεσματικότητα και απόδοση της απομάκρυνσης.

Precomputed Visibility: Ανάλογα με την συσκευή είναι δύσκολο αλγόριθμοι όπως η δυναμική απόφραξη (dynamic occlusion) να τρέξουν σε πραγματικό χρόνο, γι αυτό η προϋπολογιζόμενη ορατότητα δημιουργείται με το μαγείρεμα του φωτισμού. Αυτή η μέθοδος χρησιμοποιείται κυρίως για παιχνίδια κινητών συσκευών, που ο παίκτης έχει περιορισμένη πρόσβαση στο επίπεδο και ο φωτισμός των μοντέλων είναι στατικός για λόγους εξοικονόμησης πόρων της συσκευής. Η προϋπολογιζόμενη ορατότητα κάνει χρήση όγκων για την αποθήκευση της κατάστασης ορατότητας των δραστών ή των απλών μοντέλων στον κόσμο με βάση τη θέση του παίκτη ή της κάμερας. Η μηχανή θα τοποθετήσει από μόνη της τους όγκους ορατότητας σε όσα μοντέλα προβάλλουν στατική σκιά. Λόγο του τρόπου κατασκευής της προϋπολογιζόμενης ορατότητας θα αυξηθεί η ώρα μαγειρέματος του φωτισμού, αλλά το παιχνίδι θα καταλήξει να είναι πιο ελαφρύ κατά την εκτέλεση.

κεφάλαιο 5 Υφές και Υλικά

Στα αρχικά στάδια ανάπτυξης ηλεκτρονικών παιχνιδιών είναι σύνηθες να παραλείπονται απαιτήσεις όπως λεπτομερειακά μοντέλα, υφές και υλικά, μιάς και πρώτα θα πρέπει ελεγχθεί η βασική μηχανική του παιχνιδιού. Τα αρχικά επίπεδα φτιάχνονται με βασικά γεωμετρικά σχήματα, για να δώσουν μία πρώτη αίσθηση στους σχεδιαστές και στους προγραμματιστές, γνωστό και ως blocking out. Αφού έχουν καταλήξει στο στυλ του παιχνιδιού, θα περάσουν στην διαδικασία μοντελοποίησης που έχει προαναφερθεί στο προηγούμενο κεφάλαιο. Κατά την διάρκεια ή μετέπειτα αυτής της διαδικασίας, θα πρέπει να δοθούν στα μοντέλα υφές ώστε να προσεγγίσουν το καλλιτεχνικό όραμα των concept artist και του δημιουργού. Η έννοια των υφών είναι ένας συνηθισμένος όρος στον ψηφιακό κόσμο μιας και σε αυτές βασίζονται οι μορφές, που θα έχουν τα μοντέλα. Θα μπορούσε να πει κάποιος ότι είναι το βασικότερο κομμάτι της ανάπτυξης ενός παιχνιδιού από πλευράς εμφάνισης, μιας και τα μοντέλα θα είναι αυτά που θα γεμίζουν το περιβάλλον και θα πρέπει να είναι άριστα υλοποιημένα, με τις κατάλληλες υφές. Παρόλο που η γεωμετρία καθορίζει την σύσταση, η υφή είναι αυτή που θα προσδιορίσει την τελική εμφάνιση του μοντέλου και θα το κάνει να είναι έτοιμο για τοποθέτηση στο παιχνίδι. Ένα μοντέλο δεν περιορίζεται σε μία μόνο υφή. Στην πραγματικότητα οι εικόνες που χρησιμοποιούνται σαν υφές, είναι διαφορετικών ειδών πληροφορίες σχετιζόμενες με το χρώμα, το φωτισμό, την τραχύτητα και την ανάκλαση του μοντέλου.

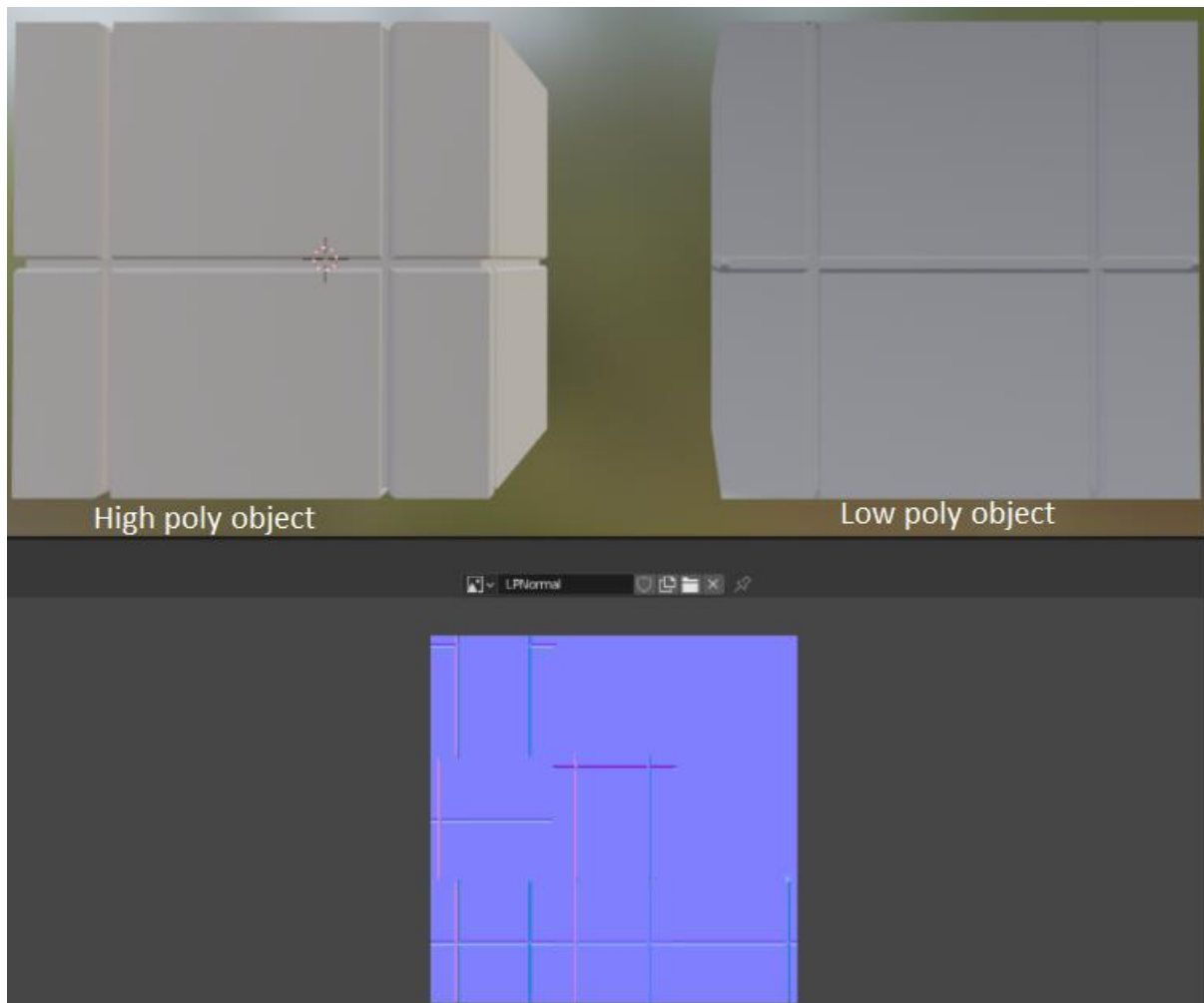
5.1 Υφές

Η υφή albedo περιέχει μόνο χαρτογράφηση χρώματος και είναι υπεύθυνη για την απεικόνιση του σχεδίου που θα έχει το μοντέλο. Σε αυτήν την υφή δεν θα πρέπει να περιέχονται πληροφορίες σχετικά με τον φωτισμό διότι μπορεί να επηρεάσουν το τελικό χρώμα. Αυτό το πρόβλημα ήταν αρκετά σύνηθες πριν την ροή εργασίας “απόδοσης βασισμένη σε φυσική” ή physically based rendering που όλη η πληροφορία σχετικά με το χρώμα και τον φωτισμό ήταν σωσμένη στην χαρτογράφηση diffuse ή χάρτης διάχυσης. Από μόνη της η υφή albedo δεν έχει αντίθεση και είναι αρκετά ανιαρή, αλλά με τον συνδυασμό των άλλων υφών το αποτέλεσμα που προκύπτει είναι αρκετά κοντά στην πραγματικότητα.



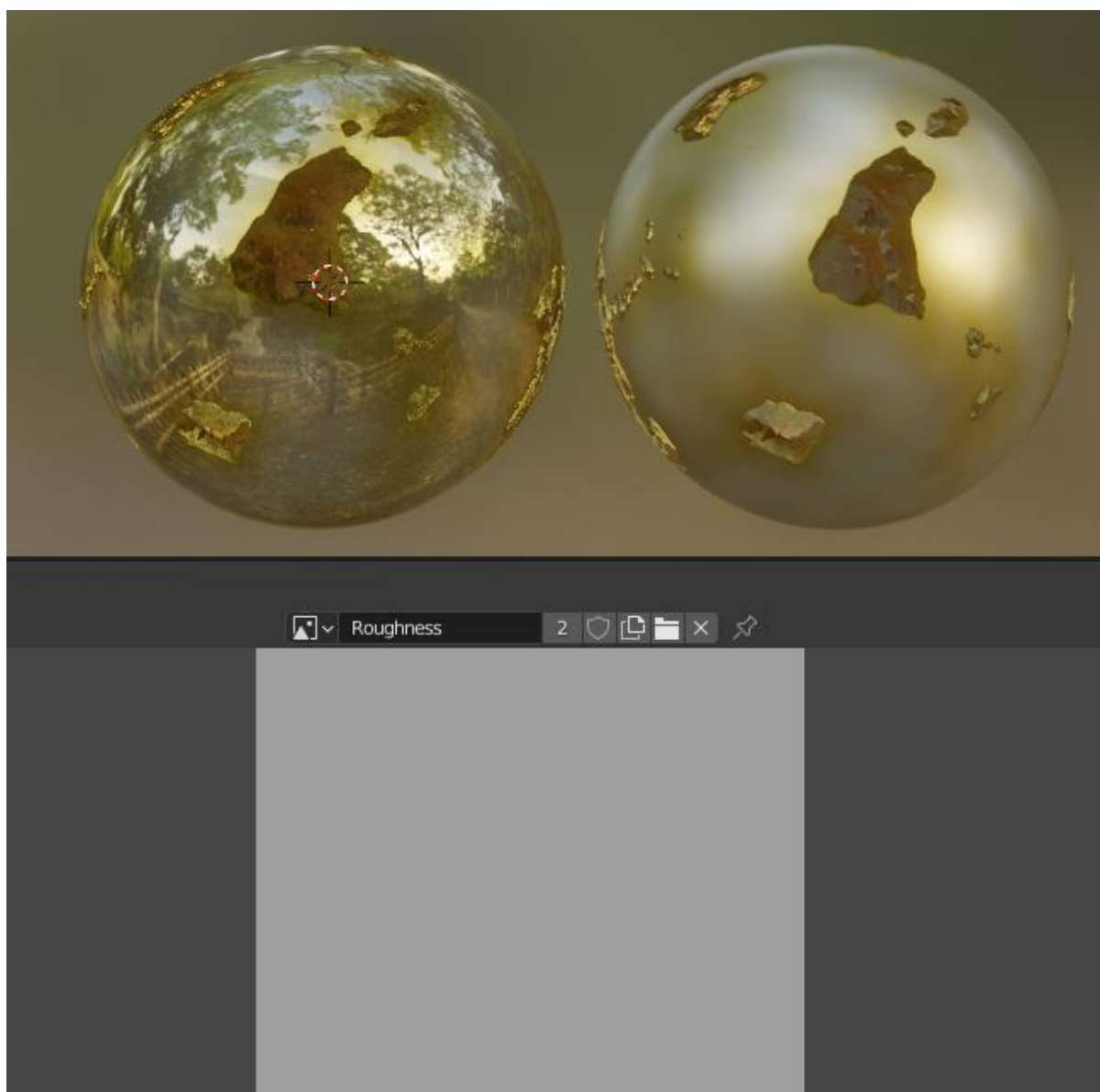
Εικόνα 3.1 Albedo texture

Οι λεπτομέρειες στην επιφάνειας του μοντέλου αποθηκεύεται σε ξεχωριστή υφή που ονομάζεται normal. Με βάση την χρωματική υπόσταση της εικόνας από τις τιμές RGB (red,green,blue) που αντιστοιχούν στις συντεταγμένες x,y,z του μοντέλου μπορούν να παραχθούν εναλλαγές στον τρόπο με τον οποίο το φως θα αντιδρά με το μοντέλο. Αυτή η διαδικασία αντιστοίχισης χρώματος και συντεταγμένων γίνεται ανά ρίxel και δίνει την ψευδαίσθηση λεπτομέρειας χωρίς να τροποποιηθεί η γεωμετρία του μοντέλου. Με αυτόν το τρόπο μοντέλα με μικρό πλήθος κορυφών, μπορούν να έχουν την ίδια ακριβός εμφάνιση με μοντέλα που έχουν πολλαπλάσιο πλήθος κορυφών και είναι πιο κοστοβόρα.



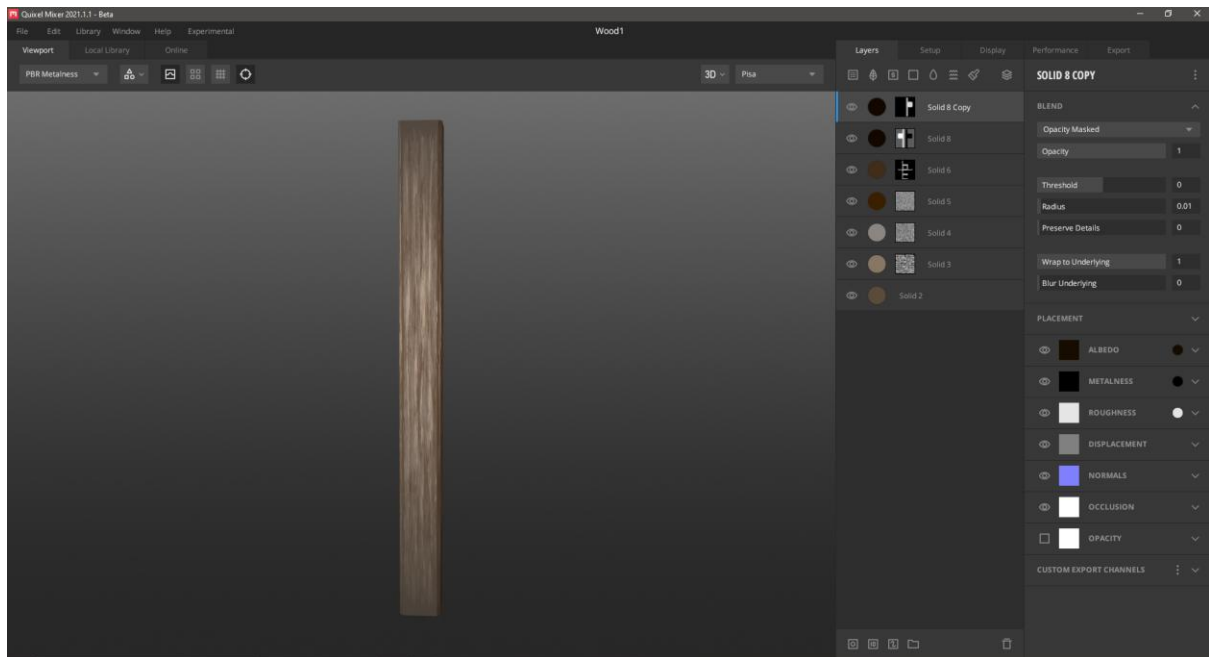
Εικόνα 3.2 Normal texture

Το κάθε μοντέλο ανάλογα με το υλικό που του αντιστοιχεί έχει και συγκεκριμένη μονάδα τραχύτητας επιφάνειας ή roughness value. Η τραχύτητα επηρεάζει τον τρόπο με τον οποίο ένα αντικείμενο θα αλληλεπιδρά με τον περιβάλλοντα φωτισμό. Αν η υφή έχει μικρό δείκτη τραχύτητας το ανακλώμενο φως θα διαχέεται σε λιγότερες κατευθύνσεις και θα κάνει την επιφάνεια πιο ανακλαστική ενώ μεγάλες τιμές θα της δώσουν μια πιο σαγρέ μορφή.

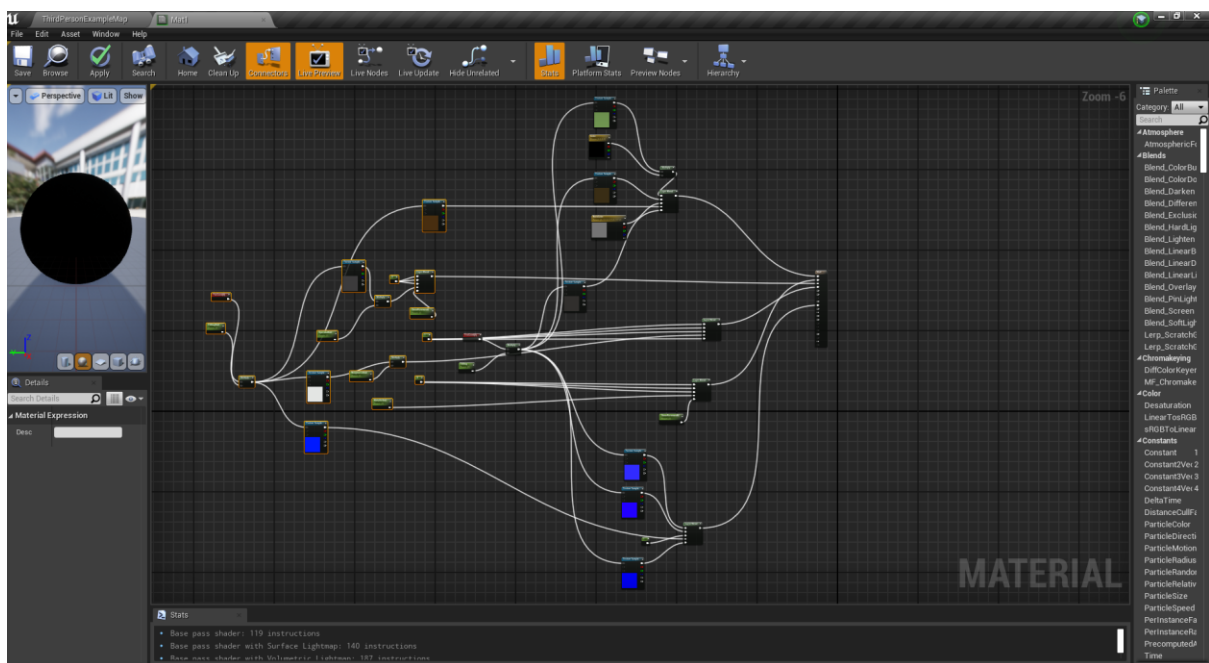


Εικόνα 3.3 Roughness-Τραχύτητα

Όταν οι υφές συνδυαστούν θα αποτελέσουν μία νέα οντότητα που ονομάζεται υλικό ή material. Η κεντρική ιδέα πίσω από ένα υλικό είναι να τοποθετηθεί απευθείας σε έναν δρόστη. Η δημιουργία υλικών ποικίλει ανάλογα με το πρόγραμμα που χρησιμοποιείται. Τα λογισμικά σχεδιασμένα για την δημιουργία υφών και υλικών για την μετέπειτα χρήση τους σε μηχανές γραφικών ή σε 3D λογισμικά προσεγγίζουν μία ροή εργασίας παρόμοια με αυτή του photoshop. Θα υπάρχει ένα μοντέλο που θα έχει εισαχθεί στο λογισμικό και πάνω του θα προστίθενται στρώματα από μάσκες που η κάθε μία θα προσθέτει και ένα επίπεδο λεπτομέρειας. Αντίθετα οι μηχανές γραφικών είναι σχεδιασμένες για μία γρήγορη και εύκολη δημιουργία υλικών και ο σχεδιαστής το μόνο που έχει να κάνει, είναι να αντιστοιχίσει τις υφές με τις αρμόδιες παραμέτρους στον κόμβο που θα λειτουργήσει σαν το σώμα του υλικού.



Εικόνα 3.4 Υλικό από το λογισμικό Quixel Mixer

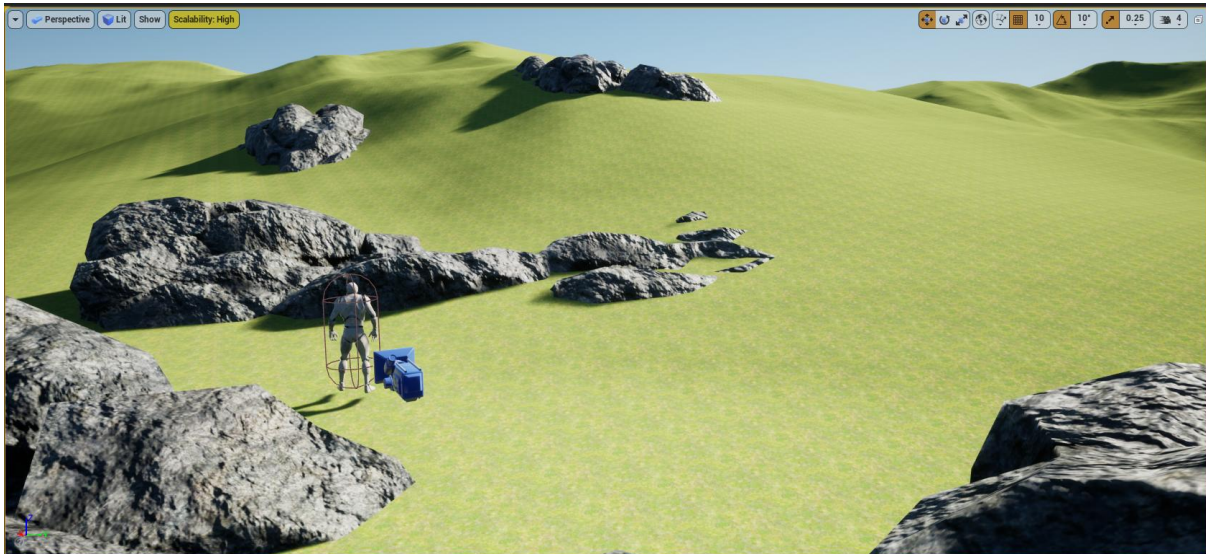


Εικόνα 3.5 Υλικό φτιαγμένο στην μηχανή γραφικών με εισαγόμενες υφές

5.2 Υλικά Εδάφους

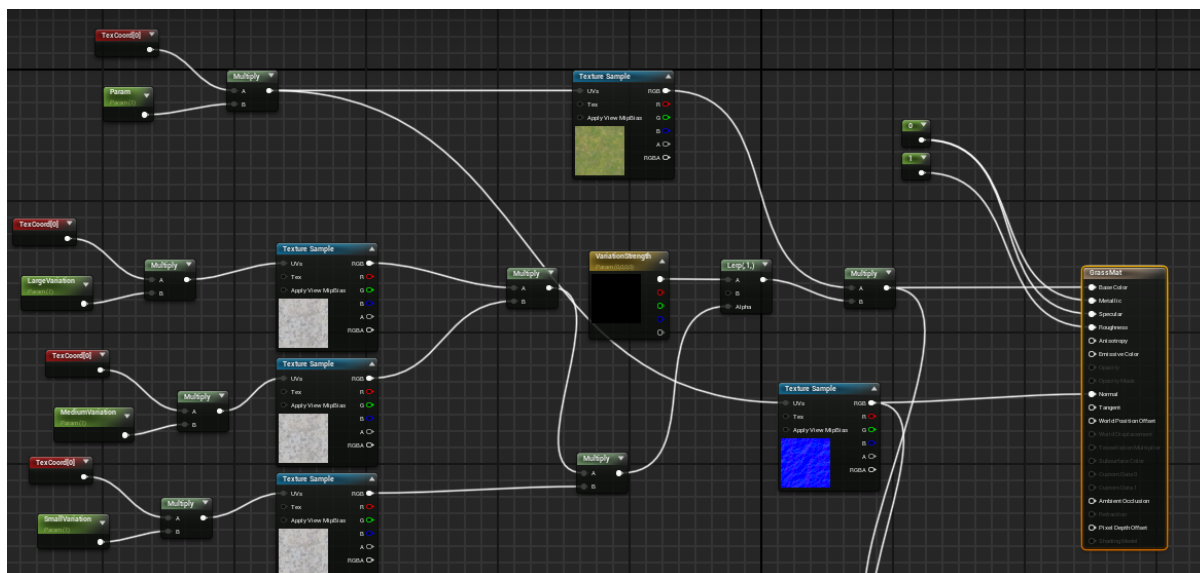
Η λεπτομέρεια στην ανάλυση των υφών και ο καλός σχεδιασμός τους είναι ένα μεγάλο κομμάτι της καλής αισθητικής που θα πρέπει να έχει ένα παιχνίδι. Παράλληλα θα πρέπει να αποφεύγεται η συχνή επανάχρηση του υλικού για την μείωση της μονοτονίας στο περιβάλλον. Ο περιορισμός του ενός υλικού ανά δράστη δεν είναι ιδιαίτερο πρόβλημα στα 3D μοντέλα, μιας και το καθένα είναι μοναδικό και πιθανότατα θα έχει διακριτές διαφορές με τα υπόλοιπα. Αντίθετα όταν προστίθεται ένα υλικό στο

έδαφος ενός παιχνιδιού παρατηρούνται άμεσα προβλήματα. Το πρώτο είναι η επανάληψη σε οποιαδήποτε υφή είναι τοποθετημένη στο υλικό. Το υλικό δεν γνωρίζει το μέγεθος του εδάφους οπότε δεν μπορεί από μόνο του να έχει την σωστή κλίμακα για να μην κάνει ορατή την επαναληψιμότητα στις υφές του. Σε αυτό το σημείο ο σχεδιαστής θα ξεκινήσει την αναγκαία και χρήσιμη διαδικασία εξοικείωσης του, με τους κόμβους υλικών της Unreal engine 4, που με βάση μαθηματικών πράξεων και μεθόδων του επιτρέπουν να αναπτύξει ιδανικά υλικά για οποιοδήποτε επίπεδο. Αφού έχει επιλέξει τις υφές που θέλει να χρησιμοποιήσει θα χρειαστεί να τροποποιήσει το σκέλος των υφών, που πρέπει, για να έχει ένα καλύτερο αποτέλεσμα.



Εικόνα 3.6 Texture tiling

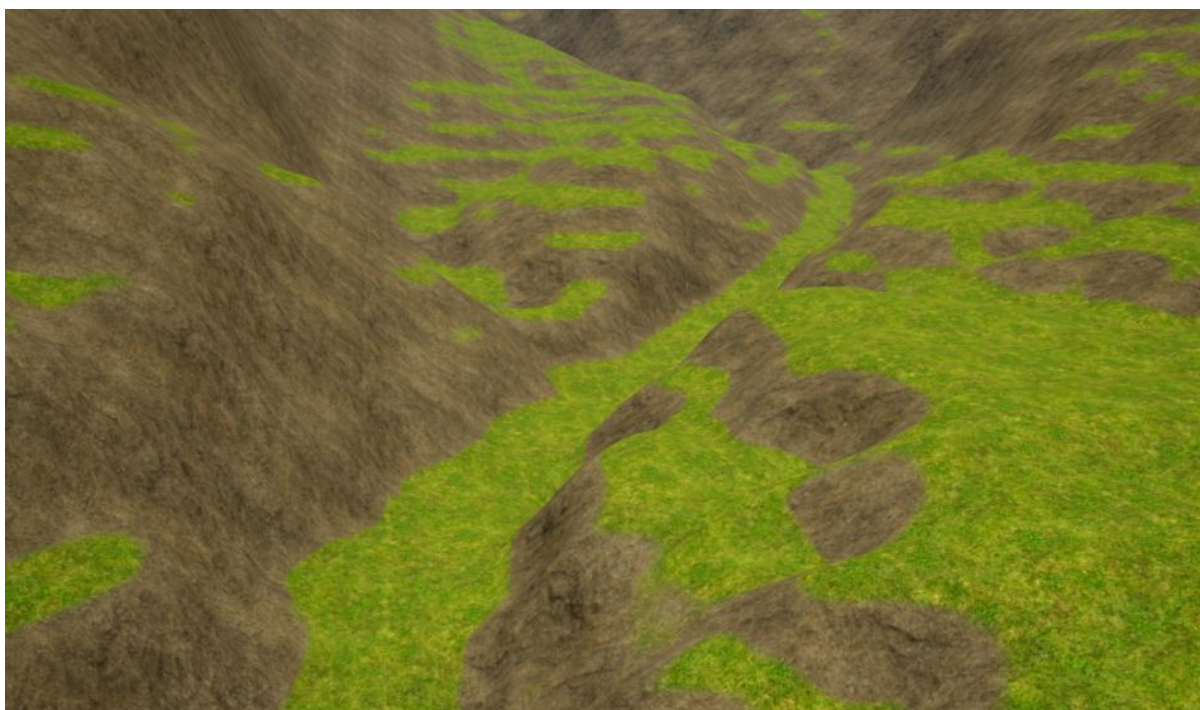
Ο κόμβος Texture Coordinates μεταβάλλει τις υφές ανάλογα με τις τιμές UV που του έχουν ανατεθεί και τις χρησιμοποιεί, σαν χάρτη, για την απεικόνιση των υφών στο έδαφος. Η αλλαγή κλίμακας UV απλώνει τις υφές σε μεγαλύτερη έκταση στην επιφάνεια του εδάφους. Παρόλα αυτά δεν υπάρχει καμία περαιτέρω αλλαγή στην υφή για να κάνει το υλικό πιο ενδιαφέρον οπτικά. Η διαδικασία πολλαπλασιασμού υφών, είναι ένας γρήγορος και αποτελεσματικός τρόπος για να παραχθεί ένα ικανοποιητικό αποτέλεσμα διαφοροποίησης του υλικού. Όπως και στην βασική υφή το ίδιο και σε αυτές που προστίθενται, θα πρέπει να είναι συνδεδεμένες με κόμβο Texture Coordinates για να μπορούν να αλλάξουν σκέλος και να έχει η κάθε μια διαφορετική βαρύτητα στο τελικό αποτέλεσμα.



Εικόνα 3.7 Ένωση υφών

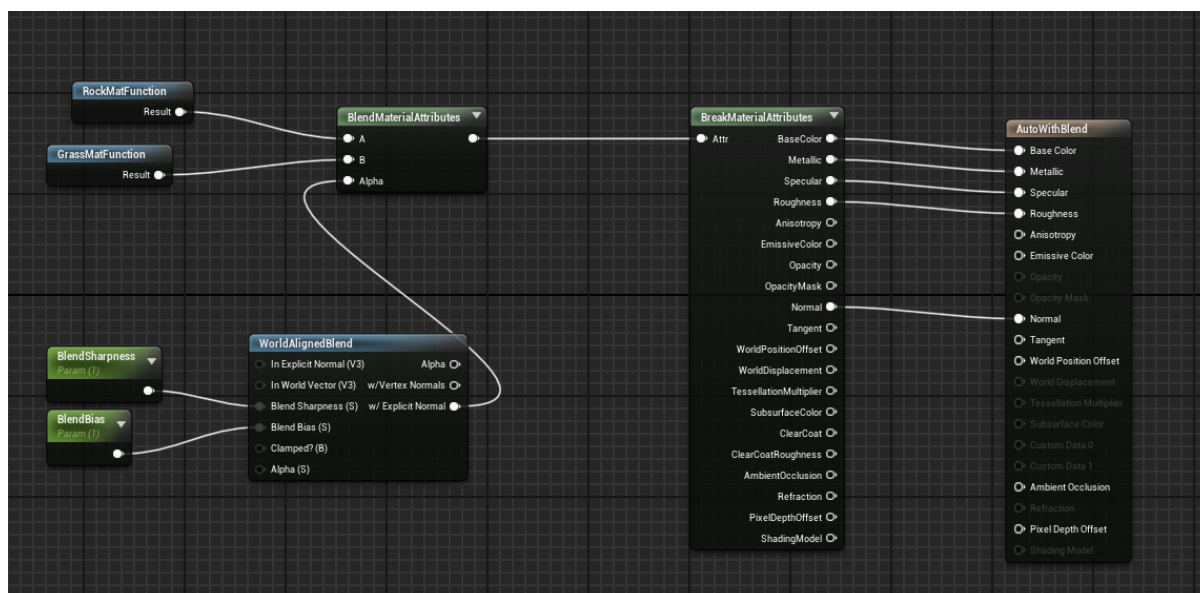
Η χρήση υλικών που επιτρέπουν την πολλαπλή τοποθέτηση υφών στο έδαφος είναι ένα πολύ δυνατό εργαλείο για τους σχεδιαστές. Η επιλογή για τον τρόπο που θα επιτευχθεί αυτή η τοποθέτηση είναι άμεσα εξαρτώμενη από την πολυπλοκότητα του εδάφους και τον χρόνο που μπορεί να αφιερωθεί για ένα καλό αποτέλεσμα. Σε παιχνίδια που το περιβάλλον είναι πιο ανοιχτό, εξερευνίσσιμο και έχει μεγάλο σκέλος, θα χρειαστεί διαφορετική προσέγγιση σε σχέση με ένα πιο γραμμικό παιχνίδι, που ο παίκτης θα είναι αναγκασμένος να ακολουθήσει ένα συγκεκριμένο μονοπάτι. Στην πρώτη περίπτωση οι σχεδιαστές δεν θα κάτσουν να τοποθετήσουν χειρωνακτικά τις υφές, αλλά θα δημιουργήσουν ένα υλικό, που θα τοποθετείται από μόνο του συγκεκριμένες υφές σε συγκεκριμένα σημεία του εδάφους. Το υλικό θα εκμεταλλεύεται το heightmap του εδάφους και ανάλογα με την κλίση του θα τοποθετεί τις απαραίτητες υφές. Αντίθετα στην δεύτερη περίπτωση οι σχεδιαστές θα μπορούν να επιλέξουν μόνο τα σημεία που θα είναι προσβάσιμα από τον παίκτη και θα τοποθετήσουν χειρωνακτικά τις υφές. Παρόλα αυτά αυτές οι δύο μέθοδοι μπορούν να χρησιμοποιηθούν παράλληλα.

Το κάθε επίπεδο ή η κάθε περιοχή, μπορεί να έχει το δικό της μοτίβο και στυλ με μία συγκεκριμένη υφή να είναι πιο συχνή από τις άλλες, αλλά θα χρειαστεί σίγουρα να χρησιμοποιηθεί παραπάνω από μία υφή albedo. Σε σημεία όπως βάλτοι και δάση υπάρχει πληθώρα από διαφορετικές υφές όπως χρώμα, λάσπη, φύλλα και ρίζες που κάνουν το περιβάλλον πιο ενδιαφέρον.



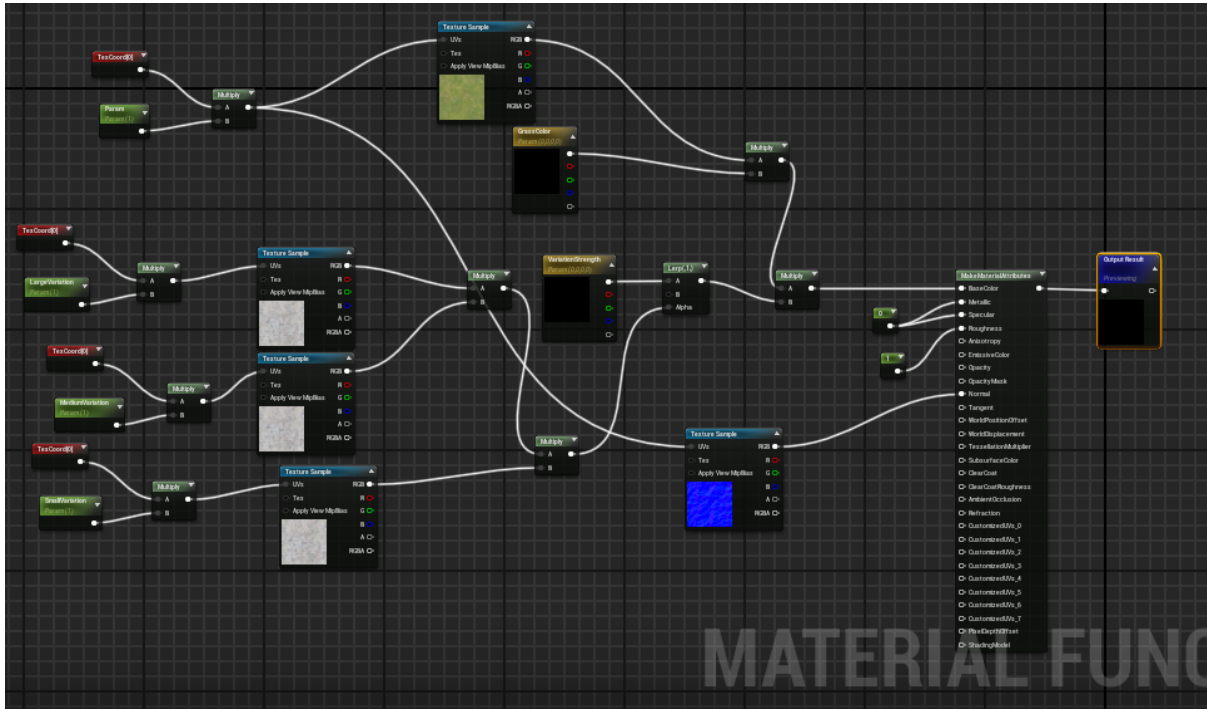
Εικόνα 3.8 Αυτόματο υλικό

Στην παραπάνω εικόνα φαίνεται η τοποθέτηση δύο υφών στο έδαφος. Τα απότομα σημεία έχουν υφή πέτρας και τα επίπεδα έχουν υφή χορταριού. Ενώ είναι άμεση και βέλτιστη η τοποθέτηση υφών ταυτόχρονα δεν μπορεί να υπάρξει διαχωρισμός σε επιφάνειες με παρόμοια μορφολογία π.χ. στο σημείο που υπάρχει το μονοπάτι, δεν γίνεται να προστεθούν περαιτέρω υφές γιατί θα επηρεάσει και τα υπόλοιπα σημεία που δεν χρειάζεται κάποια τροποποίηση. Στη συγκεκριμένη περίπτωση θα μπορούσε να προστεθεί υφή από χώμα ή από πατημένο γρασίδι, υποδηλώνοντας ότι είναι ένα συχνό πέρασμα. Άρα στην περίπτωση αυτή το αυτόματο υλικό δεν έχει την ικανότητα τοποθέτησης υφών σε τόσο ποικιλόμορφα σημεία, καταλήγοντας στην χειρωνακτική τοποθέτηση υφών.



Εικόνα 3.9 Κόμβοι αυτόματου υλικού

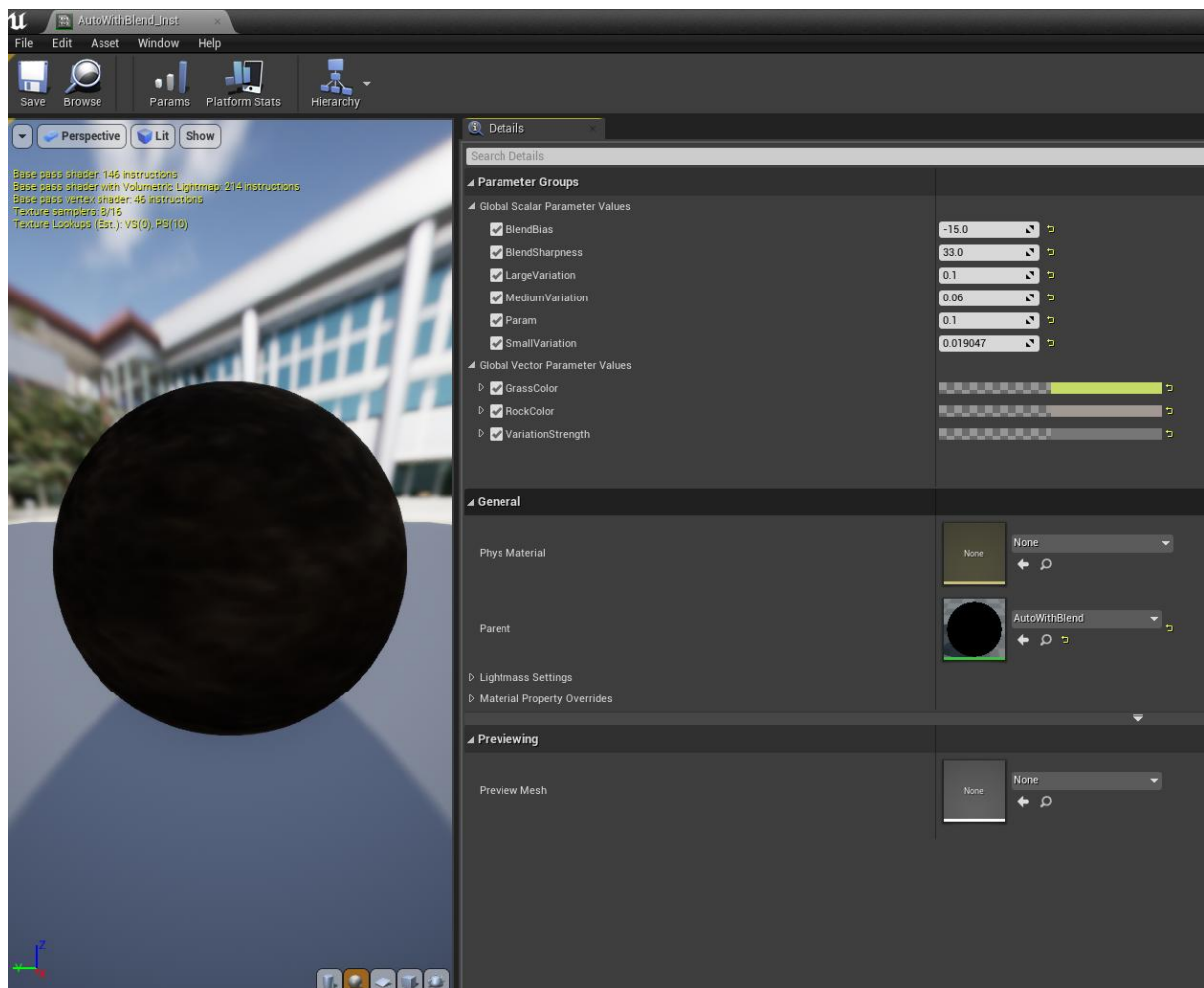
Στην συγκεκριμένη περίπτωση ο κόμβος Blend Material Attributes δέχεται σαν είσοδο δύο material functions, που έχουν αποθηκευμένη πληροφορία υφής που μπορεί να τοποθετηθεί άμεσα σε οποιοδήποτε υλικό. Ένα material function έχει τον ίδιο τρόπο στησίματος με ένα απλό υλικό (ένωση κόμβων σε ένα output) με την διαφορά, ότι δεν μπορεί να τοποθετηθεί απευθείας στο έδαφος, αλλά να χρησιμοποιηθεί σαν input για μία άλλη υφή όπως φαίνεται στην παραπάνω εικόνα.



Εικόνα 3.10 Material Function

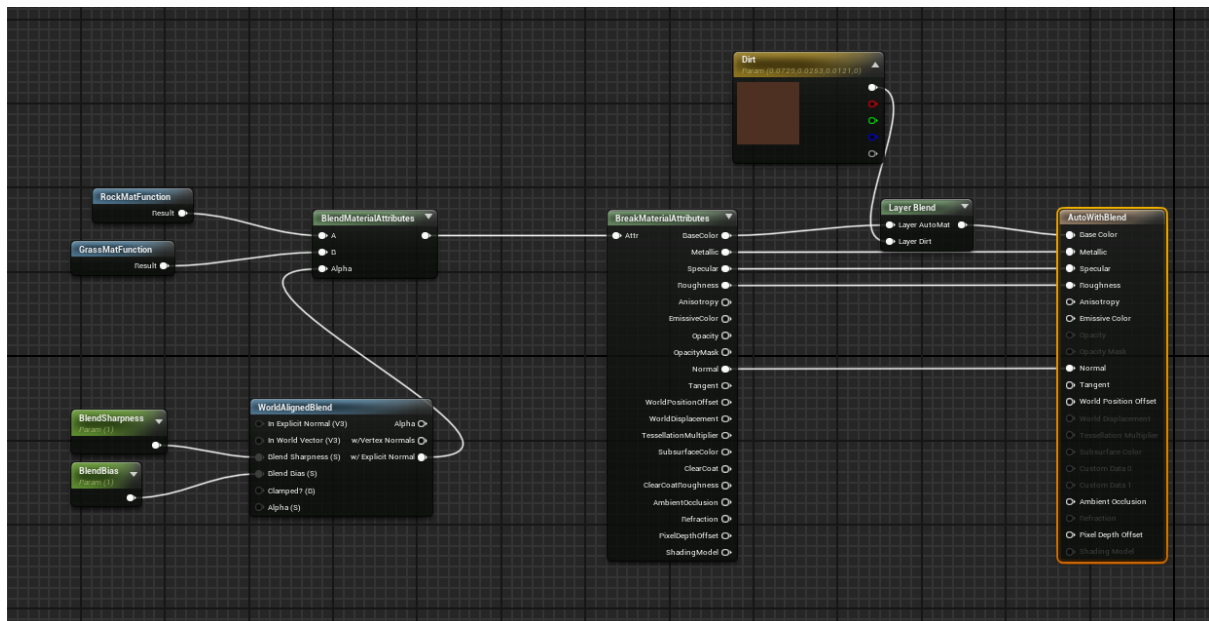
Η τρίτη είσοδος Alpha στον κόμβος Blend Material Output δέχεται χρωματική πληροφορία μαύρου-άσπρου και είναι αυτή που θα κάνει τον διαχωρισμό των δύο υλικών. Αυτή η πληροφορία θα προέρχεται από τον συνδυασμό τριών κόμβων με τον σημαντικότερο να είναι ο World Aligned Blend, που υπολογίζει το ύψος και την κλίση που έχει το έδαφος επιτρέποντας την τοποθέτηση υφών στα κατάλληλα σημεία. Για την συμβολή και την οξύτητα που θα έχει η μίξη των υφών ή των υλικών, μπορούν να προστεθούν παράμετροι, που θα δέχονται τιμές από τον σχεδιαστή, για να υπάρχει ένας καλύτερος έλεγχος.

Σε περιπτώσεις που ο σχεδιαστής θα χρειαστεί να πειραματιστεί και να τροποποιήσει κάποιες παραμέτρους ενός υλικού είναι καλύτερο να χρησιμοποιηθεί ένα material instance. Παραγόμενο από το αρχικό υλικό το material instance, δεν έχει interface με κόμβους, αλλά ένα παράθυρο με όλες τις παραμέτρους του αρχικού υλικού. Το material instance, δεν χρειάζεται να περάσει από compiler και όταν γίνει κάποια αλλαγή το αποτέλεσμα είναι ορατό άμεσα σε αντίθεση με το αρχικό, που με κάθε αλλαγή θα πρέπει να ξαναγίνει μεταγλώττιση.



Εικόνα 3.11 Material Instance

Για να μετατραπεί το αυτόματο υλικό, σε υλικό με επίπεδα, πρέπει απλά να προστεθεί ο κόμβος Landscape Layer Blend. Ο συγκεκριμένος δίνει τη δυνατότητα να συνδυαστούν πολλαπλές υφές ή υλικά, ώστε να μπορούν να χρησιμοποιηθούν ως διαφορετικά επίπεδα. Το Landscape Layer Blend χρησιμοποιεί έναν πίνακα για να αποθηκεύει πληροφορίες σχετικά με το επίπεδο. Στον πίνακα μπορούν να προστεθούν πολλαπλά στοιχεία, το καθένα με την δική του ονομασία και τύπο μίγματος (blend type). Η μοναδική ονομασία του κάθε επιπέδου είναι σημαντική σε περίπτωση που χρησιμοποιηθούν κόμβοι όπως το Landscape Grass Output, που ετυμολογικά αντιλαμβανόμαστε ότι θα προσθέσει μοντέλα τύπου foliage.



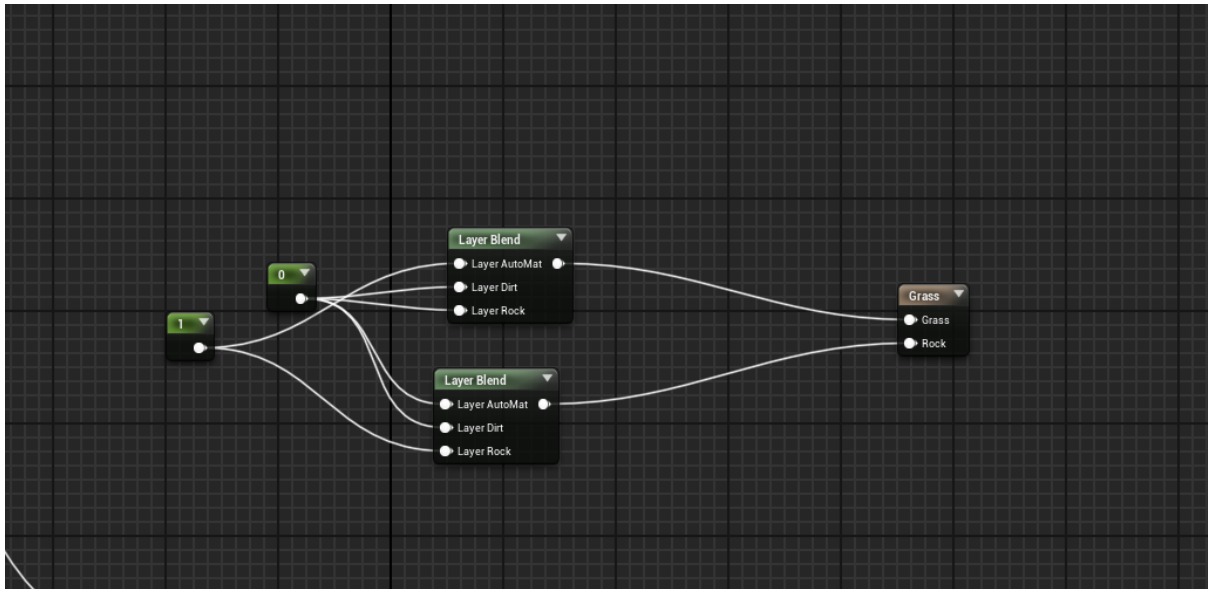
Εικόνα 3.12 Landscape Layer Blend

Το Landscape Layer Blend έχει τρεις διαφορετικές μίξης υφών. Η πρώτη είναι το Weight Blend που χρησιμοποιεί ένα ποσοστό συνόλου υφών που καθορίζει την βαρύτητα, που θα έχει η κάθε μία στο έδαφος. Για παράδειγμα όταν υπάρχει ένα πλήθος υφών μπορούν κάλλιστα να προστεθούν με οποιαδήποτε σειρά επιθυμεί ο σχεδιαστής, με την κάθε υφή που προτίθεται να συμβάλει, στο συνολικού ποσοστού μίξης. Όταν τοποθετείται μία υφή το ποσοστό όλων των υπόλοιπων υφών θα μειωθεί και αυτής θα αυξηθεί. Ο δεύτερος τρόπος μίξης είναι το Alpha Blend. Παρόλο που δεν χρησιμοποιείται τόσο συχνά όσο το Weight and Height Blend είναι αρκετά χρήσιμο για την αντιμετώπιση προβλημάτων που προκύπτουν, από τυχόν υφές που σε συγκεκριμένα σημεία του εδάφους έχουν ταυτόχρονα μηδενική χρωματική τιμή δημιουργώντας μαύρα σημεία. Ο τρόπος με τον οποίο το Alpha Blend επιτρέπει τον χρωματισμό εδάφους είναι ανάλογα με την σειρά που έχουν τοποθετηθεί οι υφές το Layer Blend βρόχο με την πρώτη να θεωρείται η βάση. Τέλος υπάρχει το Height Blend, που λειτουργεί ακριβώς με τον ίδιο τρόπο όπως και το Weight Blend αλλά με την δυνατότητα εισαγωγής πληροφορία ύψους της υφής. Σε μία υφή που έχει κενά ή χαράγματα όπως πλακόστρωτες πέτρες μπορεί να χρησιμοποιηθεί το heightmap της υφής σαν επιπλέον είσοδος στο Layer Blend βρόχο, δίνοντας την απαραίτητη πληροφορία για την μίξη της υφής με μία άλλη ανάλογα με το ύψος της.



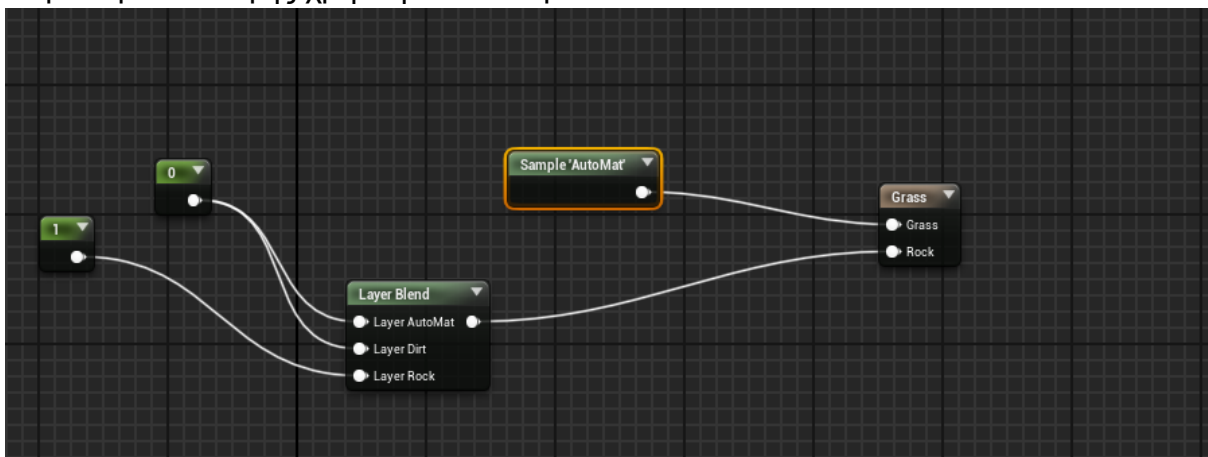
Εικόνα 3.13 Height Blend

Λόγω της χρήσης του Landscape Layer Blend ο διαχωρισμός των υφών μπορεί να χρησιμοποιηθεί για την τοποθέτηση foliage μοντέλων αυτόματα, χωρίς να επέμβει περαιτέρω ο σχεδιαστής. Για να στηθεί μία τέτοια λειτουργία θα χρειαστεί μία έξοδος Landscape Grass Output που υποστηρίζει ένα array με landscape grass types. Παρόλο που τα ονόματα του κόμβου και του τύπου foliage περιέχουν την λέξη φύλλομα, δεν σημαίνει ότι περιορίζεται ο σχεδιαστής σε κάποιο συγκεκριμένο είδος μοντέλου, απλά χρησιμοποιείται σαν όρος μιας και τα φύλλα είναι πολυάριθμα μοντέλα. Η είσοδος του Landscape Grass Output μπορεί να οριστεί με δύο διαφορετικούς τρόπους. Ο πρώτος είναι με την αντιγραφή του Landscape Layer Blend τοποθετώντας σαν είσοδό μία σταθερά 1 ή 0 στην υφή που ο σχεδιαστής θέλει να υπάρχει ή όχι το foliage μοντέλο.



Εικόνα 3.14 Landscape Grass Output ανά υφή (Layer Blend)

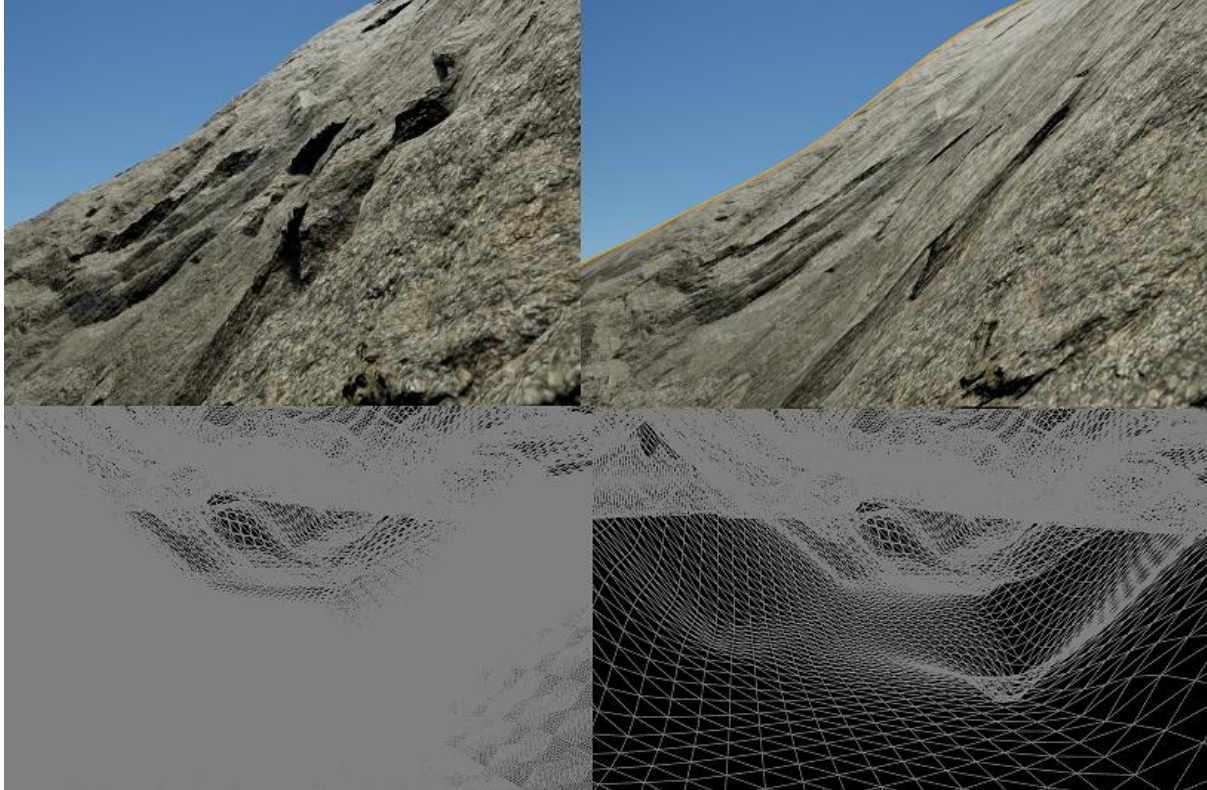
Ο δεύτερος τρόπος χρήσης του Landscape Grass Output είναι με τον βρόχο Landscape Layer Sample που δέχεται απλά ένα string του ονόματος του layer που ο σχεδιαστής θέλει να υπάρχει foliage. Μπορούν και οι δύο τρόποι να χρησιμοποιηθούν παράλληλα, αλλά συνίσταται να εφαρμόζεται μόνο ο δεύτερος μιας και κρατάει το παράθυρο διεπαφής χρήστη πιο καθαρό.



Εικόνα 3.15 Landscape Grass Output ανά υφή (Layer Sample)

Στα υλικά παρόλο που το Normal Map είναι ένα από τα πιο σημαντικά texture, δεν είναι η μοναδική υφή που μπορεί να χρησιμοποιηθεί για την πρόσθεση λεπτομέρειας. Το displacement texture, είναι μία υφή που μεταβάλλει την επιφάνεια του εδάφους ανάλογα με την υφή albedo δίνοντας βάθος στο ίδιο το έδαφος. Παρόλα αυτά είναι αρκετά επιβαρυντικό για τον υπολογιστή, να κάνει σωστή μετατόπιση μιας και χρειάζεται την απαραίτητη γεωμετρία. Στην παρούσα φάση το έδαφος αντιμετωπίζεται σαν ένα μοντέλο τριών διαστάσεων με τον αριθμό κορυφών να είναι το βασικότερο κομμάτι αυτής της διαδικασίας. Μία μεγάλη τιμή στην παράμετρο Tessellation Multiplier μπορεί να προκαλέσει μεγάλη μείωση των καρέ ανά δευτερόλεπτο. Η λεπτομέρεια και

ο ρεαλισμός που προσφέρει η displacement υφή είναι αδιαμφισβήτητη, αλλά είναι σπάνιο να την συναντήσει κάποιος να χρησιμοποιείται σε AAA παιχνιδιού. Το κόστος υλικού που απαιτείται είναι αρκετά μεγάλο και το σημαντικότερο για ένα παιχνίδι είναι, να μπορέσει να το τρέξει ο υπολογιστής. Με μία πρώτη ματιά μπορεί να μην φανεί τόσο αισθητή η μείωση των καρέ αν η σκηνή δεν περιέχει πολλά 3d μοντέλα αλλά με την συνεχή παραλλαγή και εμπλουτισμού του επιπέδου-παιχνιδιού μπορεί να είναι ένας από τους μεγαλύτερους παράγοντες κακής επεξεργασία και εκτέλεσης.



Εικόνα 3.16 Displacement και Tessellation

5.3 Υλικά Μοντέλων

5.3.1 Ζωγραφιστές υφές

Σε κάθε παιχνίδι ο κεντρικός στόχος των μοντέλων είναι να προσφέρουν μεγαλύτερο βάθος στο περιβάλλον, ενώ ταυτόχρονα μέσω αυτών ο παίκτης συναναστρέφεται με τον ψηφιακό κόσμο του παιχνιδιού. Παρόλο που το υλικό του εδάφους προσφέρει πολλαπλότητα στο παιχνίδι τα 3d μοντέλα είναι αυτά, που θα κεντρίσουν το ενδιαφέρον του παίκτη. Απλές λεπτομέρειες όπως ξερά φύλλα στο έδαφος ή πολυκαιρισμένες πέτρες κάποιου παλιού και ξεχασμένου δρόμου, μπορούν να μεταμορφώσουν ένα επίπεδο. Ανάλογα με το παιχνίδι οι σχεδιαστές μπορούν να δημιουργήσουν αντικείμενα που θα προσφέρουν την απαραίτητη ετερογένεια στην σκηνή. Σε κάθε περίπτωση οι υφές και τα υλικά που θα χρησιμοποιηθούν θα πρέπει να είναι καλά σχεδιασμένα για να υπάρχει το κατάλληλο και επιθυμητό αποτέλεσμα.

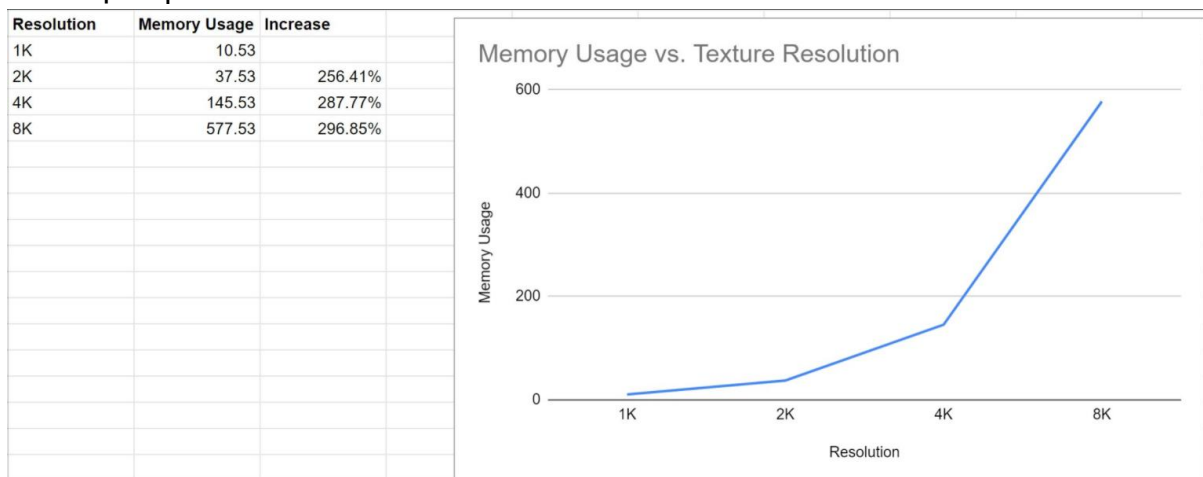
Το γεγονός ότι μέσω των υφών μπορούν να δημιουργηθούν λεπτομέρειες στο υλικό δίχως, να χρειαστεί να αλλάξει η τοπολογία του μοντέλου, επιτρέπει την μείωση της γεωμετρίας σε αυτά, βελτιστοποιώντας την επεξεργασία του παιχνιδιού.

Παρόλα αυτά δεν είναι τόσο απλό να τοποθετηθούν υφές σε 3d μοντέλα προς την ακολουθία κάποιων συγκεκριμένων βημάτων που έχουν αναφερθεί στο 2ο κεφάλαιο (uv-mapping, uv-unwrapping, δημιουργία αντιγράφου μοντέλου με λιγότερες κορυφές). Ταυτόχρονα ο σχεδιαστής θα πρέπει να αποφασίσει τον τρόπο με τον οποίο θα κατασκευάσει τις υφές, με τον καθένα να εφαρμόζεται ανάλογα με την εμφάνιση που χρειάζεται να έχει το μοντέλο. Για την δημιουργία της albedo ή defuse υφής, δηλαδή αυτής που παρέχει την χρωματική πληροφορία του μοντέλου μπορεί να δημιουργηθεί με δύο τρόπους. Ο πρώτος και πιο παραδοσιακός είναι με τον απευθείας ζωγραφισμό της επιφάνειας, με την χρήση λογισμικών όπως Blender, Zbrush, Substance Painter και Photoshop. Στην συγκεκριμένη περίπτωση κατά τη διάρκεια δημιουργίας μιας υφής albedo, ο σχεδιαστής επιλέγει διαφόρων ειδών πινέλα, χρώματα και άλλων ειδών εργαλεία, για να τοποθετήσει λεπτομέρειες στην επιφάνεια. Από την στιγμή που ο ίδιος ο σχεδιαστής επιλέγει το χρώμα του μοντέλου και τις λεπτομέρειές του, πρέπει να μελετήσει προσεκτικά τις ιδέες που έχουν δοθεί από τους concept artists. Αυτή η τεχνική συναντάται συνήθως σε παιχνίδια που φέρουν πιο κοντά στο καρτουνιστικο-comic book ή στυλιζαρισμένο στυλ όπως οι σειρές Darksiders, Dishonored και Borderlands που δεν προσπαθούν να μιμηθούν το πραγματικό υλικό ενός μοντέλου αλλά να το κάνουν απλά να ξεχωρίζει τόσο, ώστε να ο παίκτης να καταλαβαίνει τι βλέπει. Το καλό με τα υλικά που αποτελούνται από υφές που τις έχει ζωγραφίσει ο σχεδιαστής, είναι η πληροφορία που είναι απευθείας τοποθετημένη στην albedo-diffuse υφή, όπως οι σκιές και το ambient occlusion με αποτέλεσμα το ίδιο το υλικό να είναι πιο ελαφρύ στην επεξεργασία του. Ο χρόνος που μπορεί να αφιερωθεί στον ζωγραφισμό της albedo υφής, μπορεί να διαφέρει ανά παιχνίδι και πλατφόρμα πώλησης του. Ένα καλό παράδειγμα είναι αρκετά mobile games. Σε αυτά ο παίκτης δεν έρχεται σε πολύ κοντινή επαφή με την οθόνη ενώ ταυτόχρονα η ίδια είναι αρκετά μικρότερη σε μέγεθος από μία οθόνη υπολογιστή δεν χρειάζεται να παραχθούν αρκετά λεπτομερειακά υλικά για τα 3D assets.



Εικόνα 3.17 Ζωγραφισμένες υφές

Σημαντική σημείωση είναι ότι και οι ίδιες οι υφές και τα υλικά τους καταλαμβάνουν χώρο στην κύρια μνήμη κατά την εκτέλεση του παιχνιδιού. Πλέον είναι αρκετά εύκολο να παραχθούν εικόνες 4k και 8k αλλά αν η κάθε υφή μοντέλου είναι στην υψηλότερη ανάλυση το θα υπάρχει μεγάλη επιβάρυνση στο υλικό του υπολογιστή.

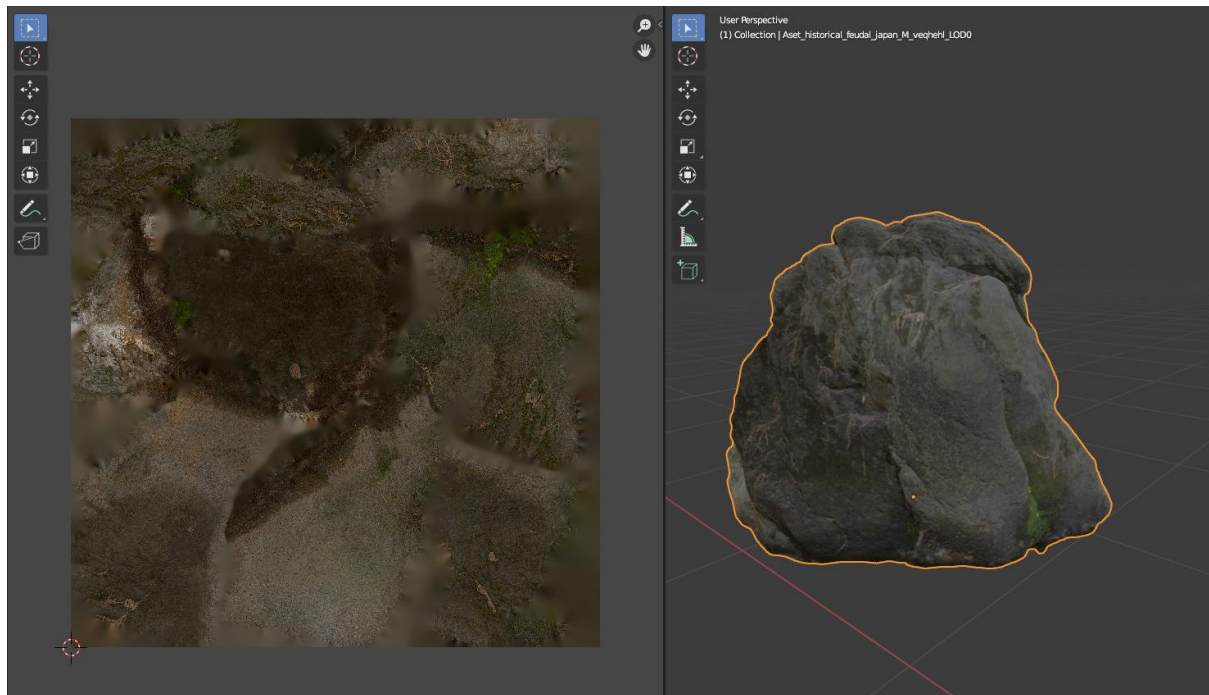


Εικόνα 3.18 Χρήση μνήμης ανάλογα την ανάλυση της υφής.

Ανάλογα την συσκευή που προορίζεται να εκτελεστεί το παιχνίδι προβλέπονται κάποιες προϋποθέσεις και περιορισμοί, που θα πρέπει να ακολουθήσουν οι σχεδιαστές. Μία από αυτές είναι, το μέγεθος ανάλυσης των υφών ή texture space. Για να είναι πιο συμπιεσμένες οι φωτογραφίες υφών χρησιμοποιείται μόνο μία για κάθε παράμετρο του υλικού (albedo, normal, roughness) και όλα τα μέλη που απαρτίζουν ένα μοντέλο έχουν κοινό uv-map. Αυτή η τεχνική ονομάζεται trim texture και βοηθάει τους σχεδιαστές να τοποθετήσουν άμεσα υφές απλά τοποθετώντας τους uv χάρτες

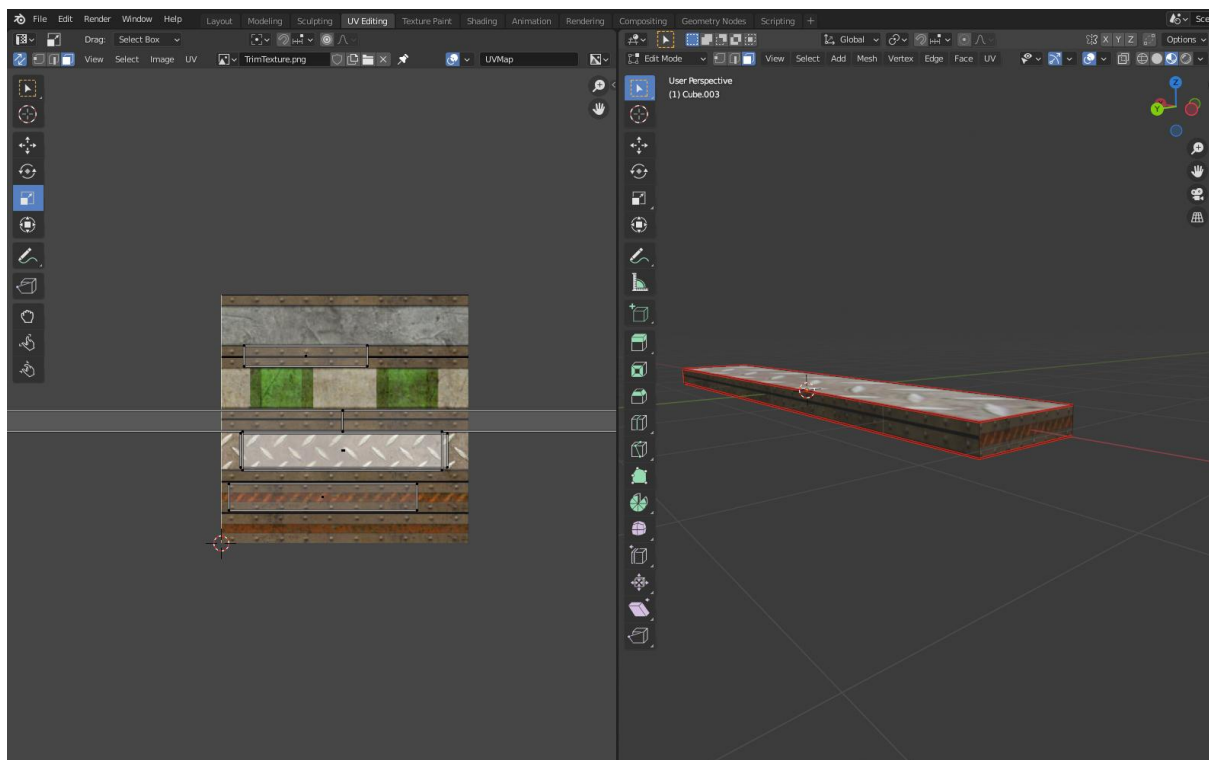
A vertical strip of various textures including metal plates, rivets, green foliage, and a diamond plate pattern.

Άλλος ένας τρόπος τοποθέτησης και αποθήκευσης υφής είναι το atlas texture που είναι και αυτός που συναντάται περισσότερο στα παιχνίδια. Σε αντίθεση με το trim texture ο atlas είναι σχεδιασμένος μόνο για ένα αντικείμενο ή για ένα μέρος του αντικειμένου. Αυτό είναι αρκετά βολικό, όταν το μοντέλο είναι μικρό σε μέγεθος μιας και δεν χρειάζεται η ανάλυση της υφής να είναι υψηλή. Σε περίπτωση που χρησιμοποιείτε μόνο ένα atlas texture για μεγάλα μοντέλα, όπως κτίσματα, θα πρέπει η ανάλυση να είναι αρκετά υψηλή, ώστε να μην φαίνονται τα pixels, αλλά θα υπάρχει τεράστια επιβάρυνση στον υπολογιστή, ειδικά αν είναι επαναλαμβανόμενη η χρήση τους. Για να λυθεί αυτό το πρόβλημα θα πρέπει να οριστούν διαφορετικά atlas textures για κάθε επιφάνεια του 3D asset, ώστε να είναι αρκετά πιο εύκολο για τον σχεδιαστή, να ελέγξει τις υφές και να της τροποποιήσει.



Εικόνα 3.20 Atlas Texture

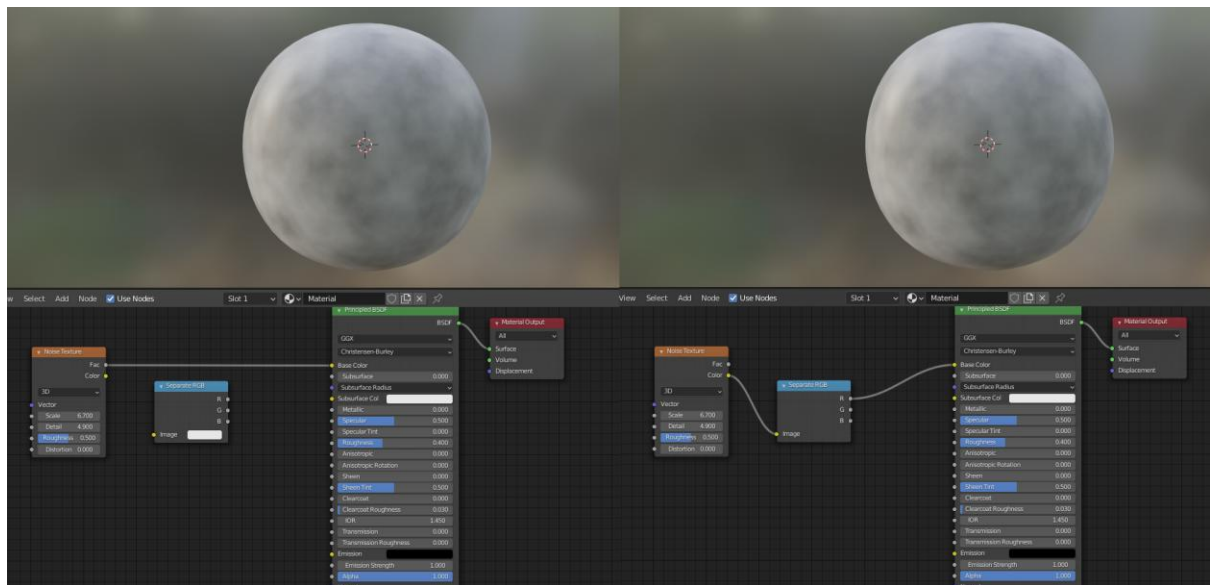
Για να μπορέσει ο σχεδιαστής να αλλάξει την κλίμακα της υφής ή να μετατοπίσει τελείως τις επιφάνειες που επιθυμεί να αλλάξει η υφή τους, θα χρειαστεί να τροποποιήσει το uv-map του μοντέλου. Για αυτόν τον λόγο είναι αρκετά σημαντικό να υπάρχουν καλά τοποθετημένες ραφές, ώστε να επιλέγεται μόνο το επιθυμητό μέρος του μοντέλου. Με την αλλαγή κλιμάκωσης των επιφανειών η υφή θα είναι λιγότερο ή περισσότερο λεπτομερής και επαναλαμβανόμενη. Η μεγάλη κλιμάκωση έχει ως αποτέλεσμα την σμίκρυνση της υφής, ενώ μικρή κλιμάκωση επιφανειών καταλήγει σε μεγέθυνση της υφής.



Εικόνα 3.21 Αλλαγή uv για μετάθεση υφής.

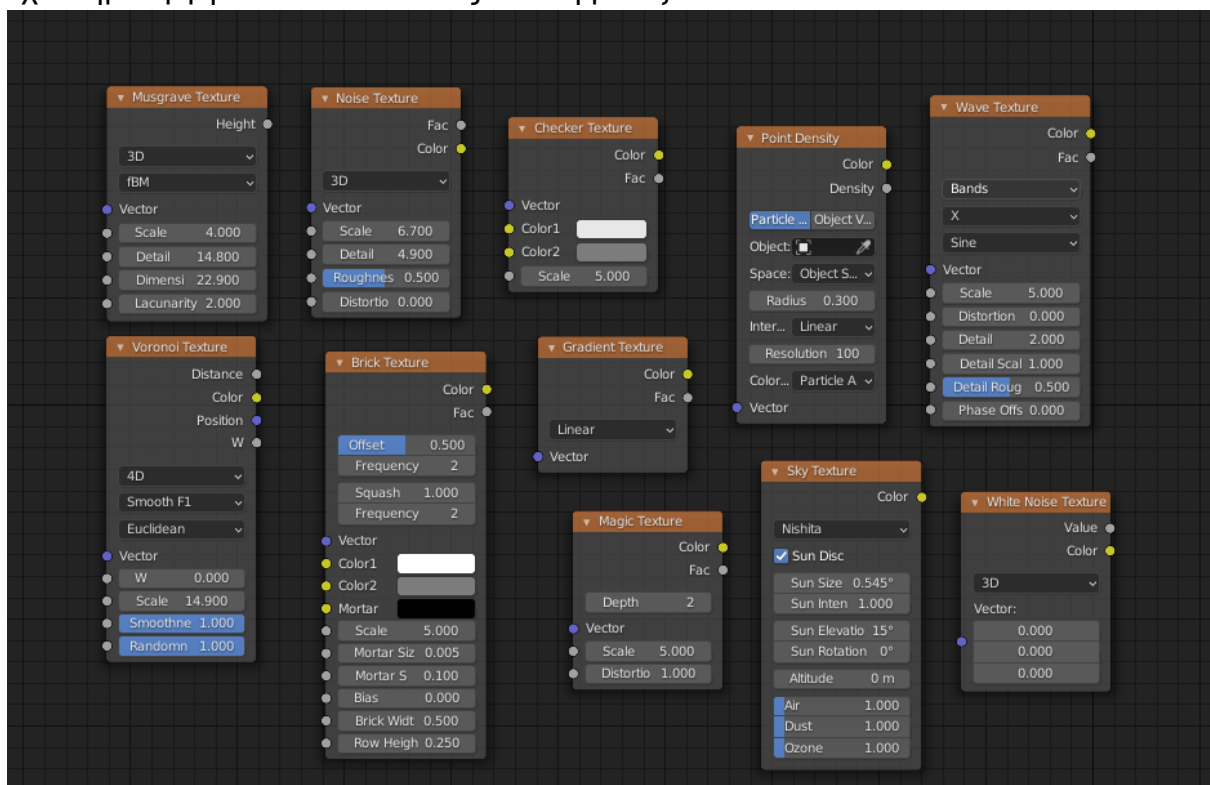
5.3.2 Procedural generated textures-Διαδικαστικά δημιουργημένες υφές

Οι υφές που είναι βασισμένες σε στατικές εικόνες όπως οι ζωγραφισμένες υφές ή οι υφές από φωτογραφίες, αφού δημιουργηθούν δεν μπορούν να υποστούν περαιτέρω επεξεργασία. Αυτό περιορίζει τους σχεδιαστές, αναγκάζοντας τους να μεταφερθούν σε άλλα λογισμικά, όπως photoshop, αν θέλουν να τις τροποποιήσουν περαιτέρω, αυξάνοντας τον χρόνο παραγωγής. Μέσω μαθηματικών εξισώσεων αρκετά λογισμικά, όπως Blender και Substance Designer, έχουν την δυνατότητα να κατασκευάσουν υφές που έχουν άπειρη λεπτομέρεια χωρίς να υπάρχει πρόβλημα επαναληψιμότητας, ενώ ταυτόχρονα το κόστος αποθήκευσης είναι αρκετά μειωμένο. Οι διαδικαστικές υφές ή procedural textures, έχουν την δυνατότητα να τοποθετηθούν σε οποιοδήποτε μοντέλο ασχέτως πολυπλοκότητας γεωμετρίας. Σαν έξοδο αυτές οι υφές μπορούν να δώσουν χρωματική πληροφορία RGB ή μάσκα, με τιμές ανάμεσα από το 0 έως το 1. Με την χρήση της color εξόδου (RGB) δημιουργούνται τρεις διαφορετικές εκδοχές της υφής και ο μέσος όρος τους είναι η έξοδος.

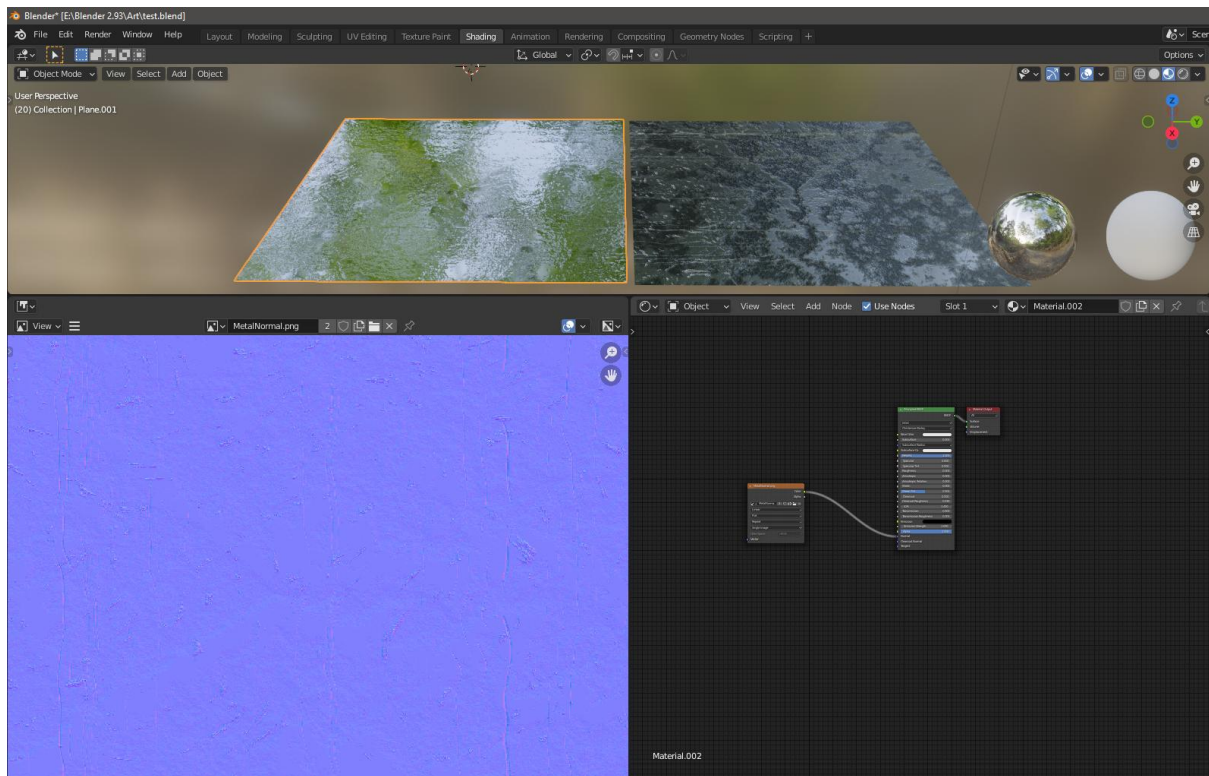


Εικόνα 3.22 Έξοδος χρώματος και παράγοντα(color,factor).

Όλες οι υφές έχουν κάποιες νόρμα εισόδου, όπως το σκέλος και η λεπτομέρεια, που μπορούν να τροποποιηθούν από άλλους κόμβους ή να πάρουν απευθείας τιμές, μέσω του πλαισίου του κειμένου. Αυτοί οι παράμετροι παίζουν βασικό ρόλο στην τελική εμφάνιση της υφής, ενώ παράλληλα μπορούν να επηρεάσουν άλλες υφές αν έχει δημιουργηθεί κάποιου είδους ένωση μεταξύ του.



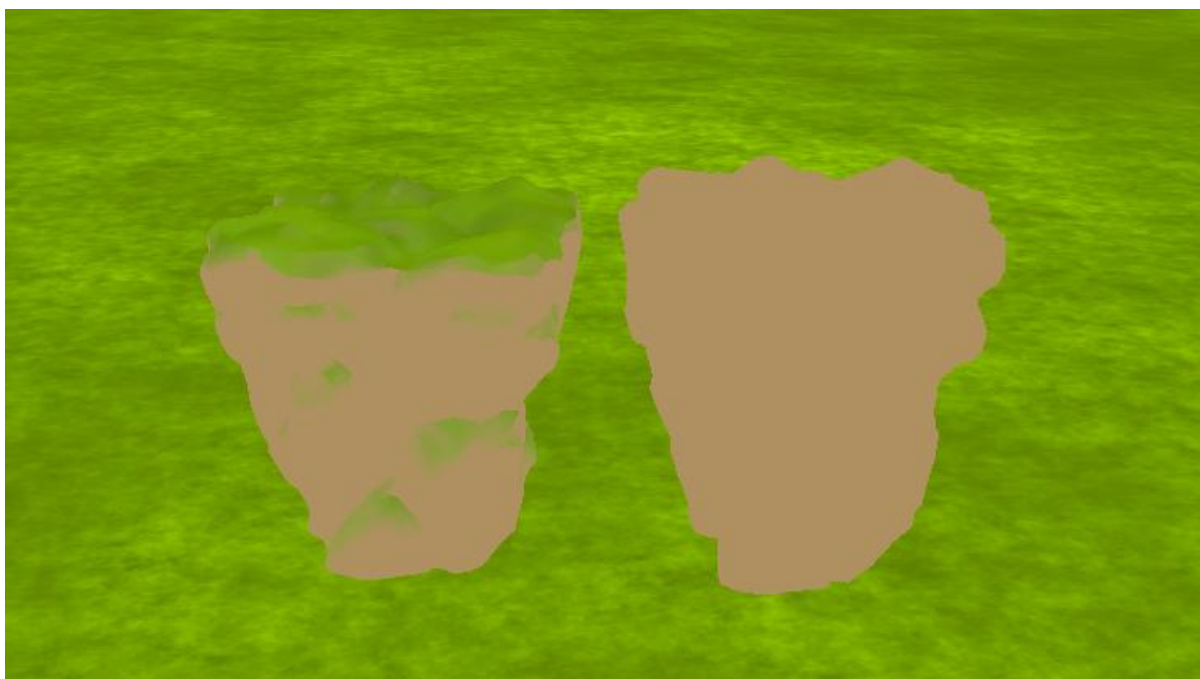
Εικόνα 3.23 Διαδικαστικές υφές στο πρόγραμμα Blender.



Εικόνα 3.25 Εικόνα normal βασισμένη σε διαδικαστική υφή.

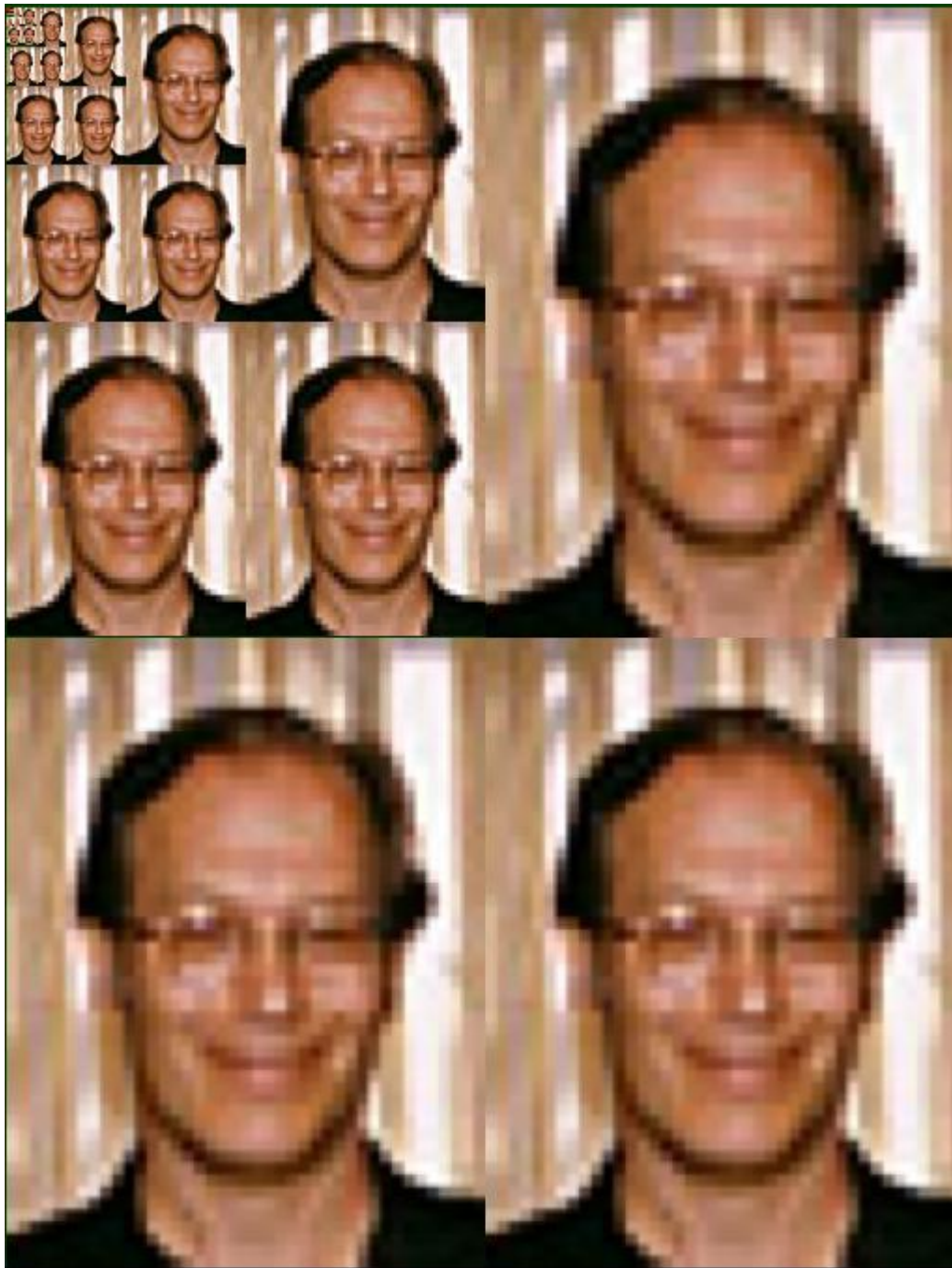
5.3.3 Virtual Textures

Από την εκδόσεις την Unreal engine 4.23 και μετά μπορεί να γίνει χρήση εικονικών υφών για την μετάδοση χρωματικής πληροφορίας στα μοντέλα από το έδαφος .Μοντέλα που τοποθετούνται σε μία σκηνή πρέπει να ταιριάζουν με το περιβάλλον και να μην δημιουργούν έντονες παραλλαγές σε αυτό. Ανάλογα με το περιβάλλον που θα πρέπει να δημιουργηθεί οι σχεδιαστές κατασκευάζουν και τις απαραίτητες υφές, στα 3D μοντέλα για μία καλύτερη μίξη με το περιβάλλον. Σε περίπτωση που κάποιο μοντέλο θα πρέπει να χρησιμοποιηθεί σε άλλο σκηνικό, θα χρειαστεί να σχεδιαστούν εξ ολοκλήρου οι υφές και το υλικό του. Για να μειωθεί ο χρόνος παραγωγής, τα μοντέλα μπορούν να πάρουν πληροφορία από το περιβάλλον και να την τοποθετήσουν στο ήδη υπάρχον υλικό.



Εικόνα 3.26 Runtime Virtual texture σε 3D asset.

Πρακτικές όπως η αναφερόμενη αναδεικνύει την λειτουργικότητα των virtual textures, αλλά ο σκοπός κατασκευής τους είναι πιο τεχνικός, λειτουργώντας καθαρά για την βελτιστοποίηση και την ταχύτερη επεξεργασία ενός project. Η κεντρική ιδέα βασίζεται στην απεικόνιση εικόνων υψηλής ανάλυσης σε πραγματικό χρόνο, ανάλογα με την απόσταση που θα προβληθούν από την οθόνη, λειτουργούν σαν LODs ή level of detail των υφών. Κάθε εικόνα υποδιαιρείται και προβάλλεται στο σημείο που έχει ορίσει ο σχεδιαστής. Με την απλή χρήση υφών το μέγεθος στην μνήμη παραμένει ίδιο άσχετα με το που θα προβληθεί η εικόνα και θα χρειάζεται πάντα η ίδια ισχύς, από την κάρτα γραφικών για το σχεδιασμό της. Αντίθετα οι εικονικές υφές θα υπο διαιρεθούν με συντελεστή 2 της προηγούμενης υφής και θα προβληθούν ανάλογα. Η βασική υφή θα είναι και αυτή ορατή αλλά μόνο σε περιπτώσεις που ο σχεδιασμός της είναι υποστηριζόμενος. Αυτή η μέθοδος ονομάζεται mipmapping και η προέλευση του όρου είναι αρχικισμός της λατινικής φράσης multum in parvo «πολύ σε μικρό χώρο». Το δεύτερο έναυσμα των εικονικών υφών που το αναφέρει και το ίδιο το όνομα είναι η ύπαρξή τους μόνο σε ένα ποσοστό στην μνήμη. Η εικονική υφή αποθηκεύεται σε σκληρό δίσκο και μεταδίδεται σε ροή στην κάρτα γραφικών, κατά το χρόνο εκτέλεσης.



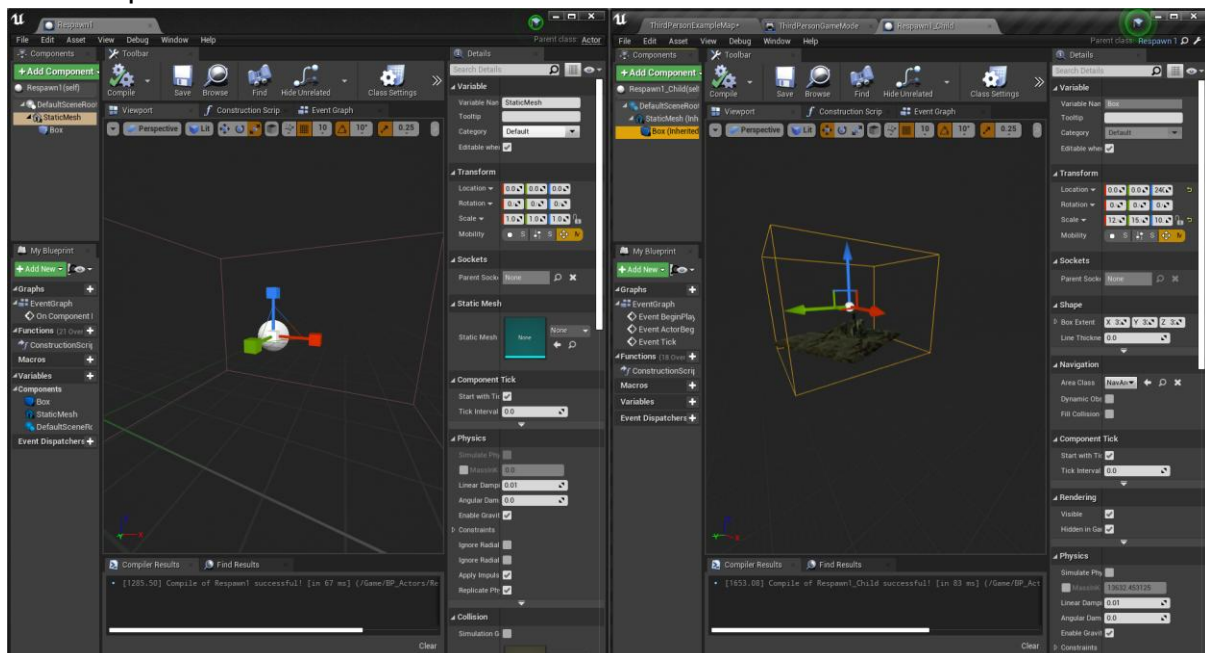
Εικόνα 4.1 Lance Joseph Williams, δημιουργός του mip-mapping.

Κεφάλαιο 6 Δράστες

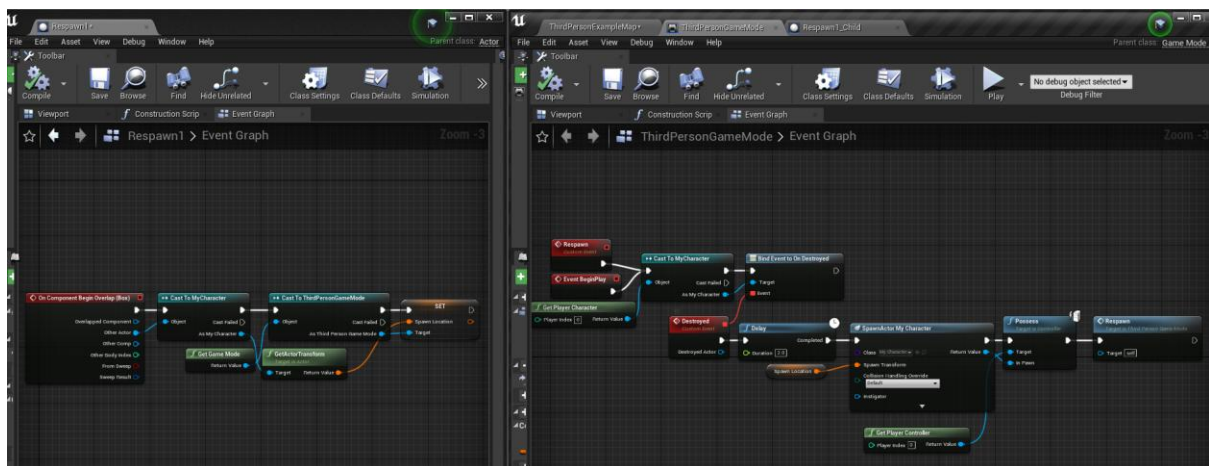
6.1 blueprint actors και user interface

Με την ολοκλήρωση του επιπέδου και του εμπλουτισμού του από μοντέλα, το επόμενο στάδιο είναι η προσθήκη του παίκτη και των δραστών που θα το κατοικούν. Η Unreal Engine προσφέρει έτοιμες κλάσεις δραστών που ο προγραμματιστής ή ο σχεδιαστής μπορεί να τροποποιήσει μέσω του Blueprint Editor ή με την χρήση αντικειμενοστραφούς κώδικα σε C++. Ανάλογα με τον τύπο του δράστη, ο σχεδιαστής,

μπορεί να δημιουργήσει μηχανισμούς που εμπλουτίζουν το παιχνίδι π.χ. αντικείμενα που ο παίκτης μπορεί να επιδράσει με αυτά, όπως όπλα και αντικείμενα αναβάθμισης, ή αναλώσιμα υλικά, που θα τον βοηθήσουν κατά την διάρκεια του παιχνιδιού. Τις περισσότερες φορές οι δράστες αποτελούνται από δύο ή περισσότερα χαρακτηριστικά. Τα δύο βασικότερα είναι το τρισδιάστατο μοντέλο, που θα είναι ορατό στον κόσμο και το εύρος επίδρασης, που θα επιτρέπει στον παίκτη να κάνει μία συγκεκριμένη ενέργεια όταν έρθει σε επαφή με αυτό. Αρκετά σύνθηες σε παιχνίδια ανοιχτού κόσμου και εξερεύνησης, είναι η χρήση σημείων ελέγχου και αναγέννησης (checkpoint, re-spawn point) που χρησιμοποιούν τα προαναφερόμενα χαρακτηριστικά. Με έναν παρόμοιο τρόπο λειτουργούν και τα αναλώσιμα υλικά που μαζεύει ο παίκτης και τοποθετεί σε κάποιο είδους καταγραφής πόρων (inventory system). Ο τρόπος στησίματος αυτών των δραστών διαφέρει στον τρόπο λειτουργία αλλά χρησιμοποιεί παρόμοια χαρακτηριστικά. Σε περιπτώσεις που ένας δράστης έχει παραπάνω από μία χρήσεις ή επαναχρησιμοποιείται με αλλαγμένα χαρακτηριστικά, δεν χρειάζεται ο σχεδιαστής ή ο προγραμματιστής να κάνει εξ ολοκλήρου μία νέα δημιουργία. Με την κατασκευή ενός κεντρικού δράστη μπορούν να δημιουργηθούν παιδιά που κληρονομούν τα χαρακτηριστικά του και μετέπειτα μπορούν να τροποποιηθούν μέσω του Blueprint Editor.

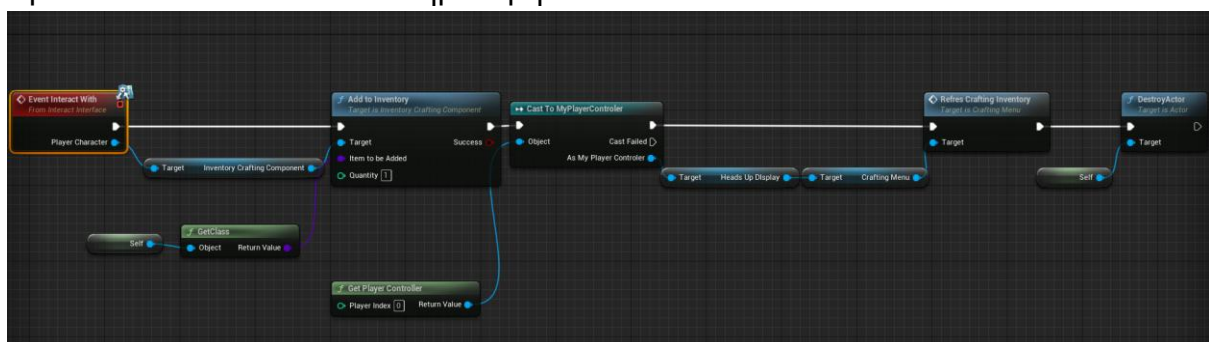


Εικόνα 4.1 Check/Re-spawn δράστης.



Εικόνα 4.2 Checkpoint/Re-spawn κώδικας.

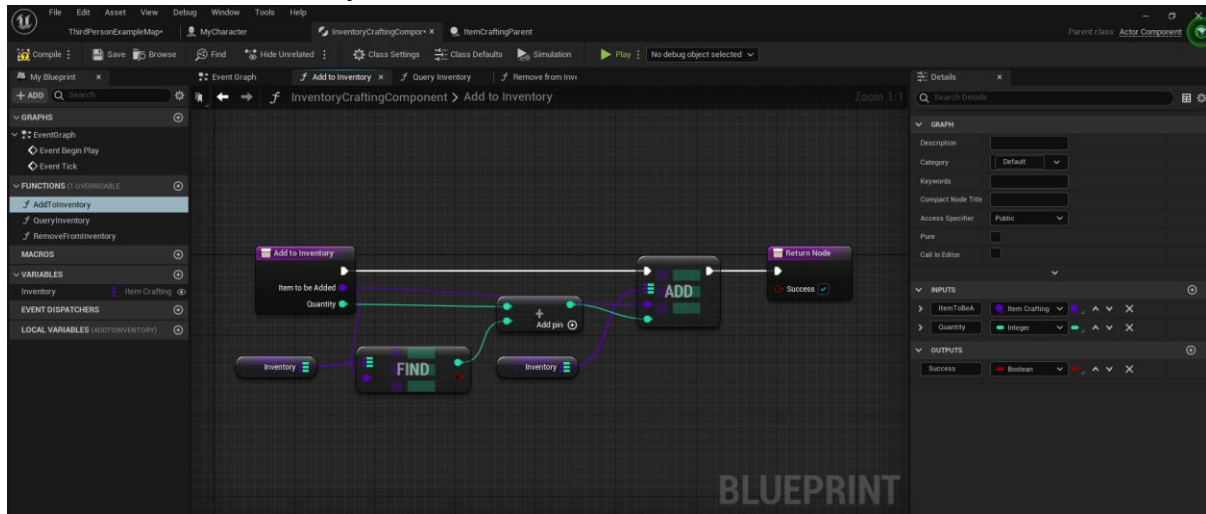
Με την χρήση γεωμετρικών σχημάτων όπως το box μπορεί ο δρᾶστης να καταλάβει αν ο παίκτης είναι σε επαφή με αυτό, επιτρέποντας την εκτέλεση του κώδικα. Ο κόμβος on component begin overlap, είναι είδους event ή συμβάν. Τα συμβάντα είναι κόμβοι που καλούνται από τον κώδικα, για να ξεκινήσουν την εκτέλεση ενός μεμονωμένου δικτύου εντολών εντός του Event Graph. Οι δρᾶστες που χρησιμοποιούνται για πιο σύνθετες λειτουργίες, όπως στην καταγραφή πόρων, δεν μπορούν να λειτουργήσουν αυτόνομα για να παράγουν το αποτέλεσμα καταγραφής, αλλά θα πρέπει να δουλέψουν συνδυαστικά με άλλες λειτουργίες της μηχανής. Η βασική μέθοδος για την αποτύπωση πληροφορίας είναι με την χρήση των user interfaces και των heads up display, που επιτρέπουν την ικανοποίηση της πληροφορίας. Στην προκειμένη περίπτωση μιλώντας για καταγραφή θα πρέπει να υπάρχει ένα παράθυρο που θα ενημερώνει τον παίκτη, για ποσότητα και για είδος αντικειμένου που θα βρίσκεται στον πίνακα καταγραφής. Αφού δημιουργηθεί ένας δρᾶστης γονιός με κατάλληλα χαρακτηριστικά και παράμετροι, ο προγραμματιστής θα προσθέσει κώδικα που θα κληρονομηθεί και στα παιδιά.



Εικόνα 4.3 Χρήση διεπαφής(interface).

Το γεγονός που βρίσκεται στην αρχή του κώδικα είναι μία λειτουργία που έχει δημιουργηθεί σε μία διεπαφή και ο προγραμματιστής την καλεί στον συγκεκριμένο δράστη. Οι διεπαφές χρησιμοποιούνται για την δημιουργία λειτουργιών, με όσους δράστες έχουν πρόσβαση σε αυτές και τους επιτρέπει, να κάνουν χρήση των συγκεκριμένων λειτουργιών, άσχετα με το αν έχουν κοινό πρόγονο. Με παρόμοιο

τρόπο μπορεί να δημιουργηθεί και ένα συστατικό ή component που θα τοποθετηθεί στον παίκτη, επιτρέποντας του να χρησιμοποιήσει τις λειτουργίες του. Στην παραπάνω εικόνα ο κόμβος add to inventory είναι η λειτουργία που δημιουργήθηκε στο προσχέδιο του Inventory Component και επιτρέπει την καταχώρηση δραστών τύπου item στο inventory.



Εικόνα 4.4 Χρήση συστατικού.

Με την ολοκλήρωση της διαδικασίας καταγραφής θα πρέπει ο παίκτης να έχει πρόσβαση στα αντικείμενα που έχει συλλέξει. Τα heads up display τα οποία θα κατοικούν ένα user interface. Το inventory θα παρέχουν αυτή την πληροφορία, αλλά η χρησιμότητα τους δεν σταματάει στην απλή ενημέρωση του παίκτη. Πίσω από κάθε ui(user interface) μπορούν να πραγματοποιηθούν πολύπλοκες διαδικασίες όπως κατασκευή αντικειμένων και αναβάθμιση του χαρακτήρων. Όπως και με τους δράστες η βάση της ευφυΐας πίσω από κάθε ενέργεια που θα κάνει ένα user interface κρύβεται στις λειτουργίες, που θα εκτελούνται ανάλογα με την είσοδο που θα δίνει ο παίκτης. Είναι απαραίτητο να υπάρχουν διαφορετικά ui, ανάλογα με το τι αποσκοπεί το καθένα. Για την δημιουργία τους οι σχεδιαστές αναπτύσσουν σε κάποιο πρόγραμμα όπως το photoshop γραφικά, με συγκεκριμένες προδιαγραφές, που τους έχουν ανατεθεί, με βάση την γενική αίσθηση και ατμόσφαιρα του παιχνιδιού για ένα πιο οργανικό αποτέλεσμα.



Εικόνα 4.5 User Interface.

Ανάλογα με την πληροφορία που χρειάζεται να επικοινωνήσει στον παίκτη επηρεάζεται και η πολυπλοκότητα του ui. Σε παιχνίδια που ο χαρακτήρας έχει ποσοστά ζωής, αντοχής, εμπειρίας και άλλες παρόμοιες μεθόδους μέτρησης χαρακτηριστικών του, που πρέπει να είναι άμεσα διαθέσιμες στον παίκτη, θα πρέπει το ui να είναι σχετικά λιτό ειδικά όταν προορίζεται να είναι συνεχώς ανοιχτό κατά την διάρκεια του παιχνιδιού. Ο στόχος της απλής εικονοποίησης αυτών των στοιχείων επιτρέπει στον παίκτη να έχει ένα “καθαρό περιβάλλον”, που του επιτρέπει να ευχαριστηθεί το παιχνίδι και την διαδραστικότητα του, χωρίς να υπάρχει πληθώρα από παράθυρα στην οθόνη.

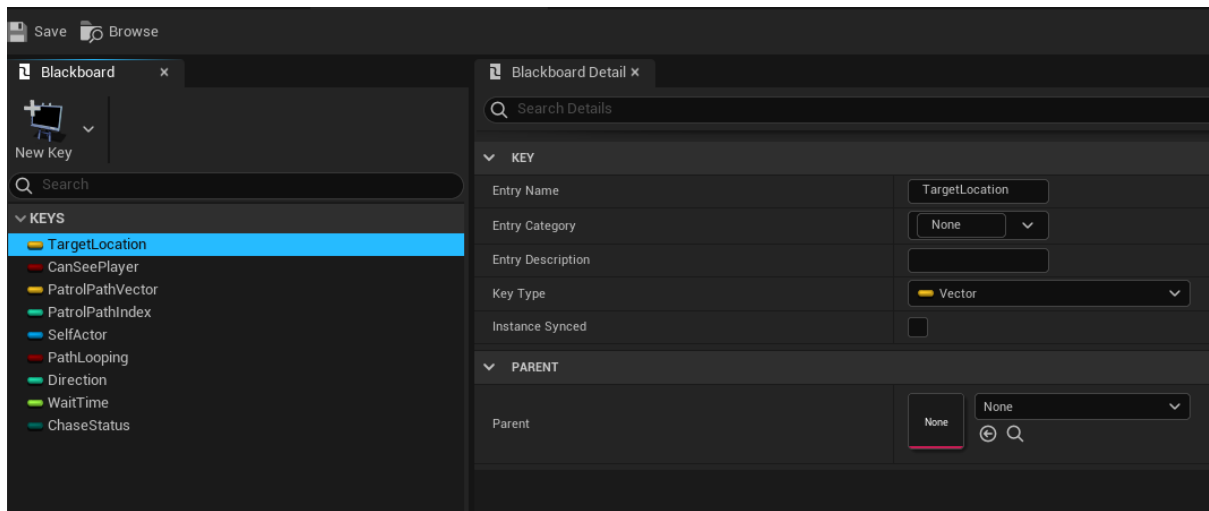


Εικόνα 4.6 Στατικό User Interface στην κεντρική οθόνη.

6.2 Τεχνητή νοημοσύνη

6.2.1 Blackboard

Με την εισαγωγή στον κόσμο της τεχνητής νοημοσύνης, μερικοί από τους πρώτους αλγόριθμους που συναντάει κάποιος είναι αυτοί της αναζήτησης δέντρων, για την αναζήτηση και εντοπισμό κλαδιών ή κόμβων, μέσα σε ένα σετ. Κάθε αλγόριθμος, έχει συγκεκριμένο τρόπο εύρεσης του στόχου, με διαφορετικό τρόπο προσπέλασης των κλαδιών του δέντρου. Σε αυτή την ιδέα βασίζονται τα δέντρα συμπεριφοράς στην Unreal Engine με την δυνατότητα να προστεθούν περαιτέρω λειτουργίες, για τον έλεγχο των αποφάσεων που θα πάρει. Για να δημιουργηθεί ένας αντίπαλος που θα έχει από πίσω του λογική, θα πρέπει να λαμβάνονται αποφάσεις με βάση κάποιων δεδομένων, που θα έχει ορίσει ο προγραμματιστής. Στην Unreal Engine αυτά τα δεδομένα στεγάζονται σε ένα blackboard που μπορεί να χρησιμοποιείται από έναν ή περισσότερους αντιπάλους, για την άμεση πρόσβαση σε αυτά.



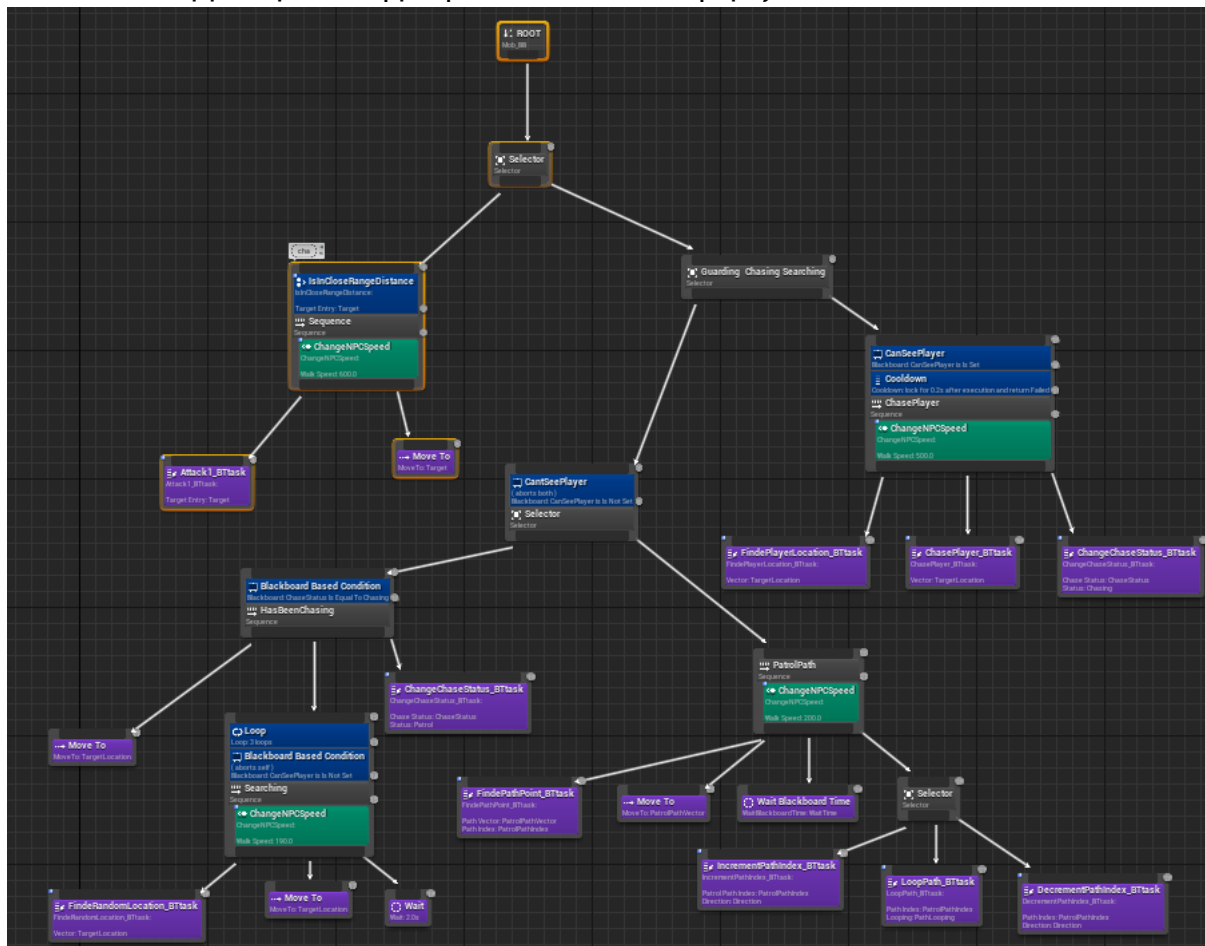
Εικόνα 4.7 Blackboard.

6.2.2 Δέντρα συμπεριφοράς

Για κάθε ενέργεια που θα κάνει ο αντίπαλος θα αντιστοιχεί σε ένα καθήκον του δέντρου που κάποιος μπορεί να το φανταστεί σαν τα φύλλα του. Το κάθε καθήκον είναι μοναδική οντότητα, που έχει το δικό της Blueprint και μπορεί να προγραμματιστεί ανεξάρτητα από τα άλλα. Τα καθήκοντα χρησιμοποιούν δεδομένα του blackboard και εκτελούν κομμάτια κώδικα, που αρμόζουν στην κάθε ενέργεια αλλά δεν είναι τα ίδια που αποφασίζουν το πότε θα εκτελεστούν. Εδώ παίζει ρόλο ο διακοσμητής που έχει βάλει ο προγραμματιστής για να ελέγχει το πότε θα μπορεί ο αλγόριθμος να προχωρήσει στο επόμενο κλαδί ή να επιλέξει ένα άλλο μονοπάτι. Οι διακοσμητές λειτουργούν σαν πύλες που ελέγχουν την τωρινή κατάσταση του αντιπάλου και καθοδηγούν τον αλγόριθμο, στο σωστό καθήκον. Για να μπορούν να οδηγήσουν την ροή του δέντρου στα σωστά φύλλα, θα πρέπει να τοποθετηθούν στους σωστούς κόμβους. Οι ενδιαμέσοι κόμβοι είναι πιο σύνθετοι και έχουν συγκεκριμένη μεθοδολογία, στην τοποθέτησή τους. Ο πρώτος είναι ο selector ή εκλέκτορα και λειτουργεί σαν γονέας, που επιλέγει, πιο θα είναι το εκτελέσιμο παιδί. Το παιδί θα αποτελείται από τα φύλλα και από το δεύτερο είδος ενδιαμέσου κόμβου που βρίσκεται πριν από αυτά, τον κόμβο αλληλουχίας. Η αλληλουχία επιτρέπει την εκτέλεση φύλλων του δέντρου το ένα μετά το άλλο σαν ένα κλειστό κύκλωμα που θα αντιμετωπίζεται σαν μία λειτουργία.

Ένας αντίπαλος μπορεί να έχει δύο βασικές καταστάσεις στηριζόμενες στο αν βλέπει τον παίκτη ή όχι και με αυτόν τον τρόπο το δέντρο μπορεί να εκτελέσει συγκεκριμένα καθήκοντα. Οι περισσότεροι εχθροί σε παιχνίδια έχουν την αρχική κατάσταση συμπεριφοράς ως κάποιου είδους περιπολίας μίας περιοχή ή φρούρησής της. Όταν ο εχθρός έρθει σε επαφή με τον παίκτη, θα παραμερίσει την κατάσταση φρούρησης και θα περάσει σε μία κατάσταση καταδίωξης. Σε ένα τέτοιο δέντρο μπορούν να συνυπάρχουν αλληλουχίες κάτω από τον ίδιο εκλέκτορα, κάνοντας τον αντίπαλο να φαίνεται πιο έξυπνος. Τέτοιες συμπεριφορές μπορεί να συναντήσει

κάποιος σε παιχνίδια όπως στην far cry σειρά, όπου αφού ο παίκτης ξεφύγει από τους εχθρούς, θα συνεχίσουν για ένα συγκεκριμένο χρονικό διάστημα να ψάχνουν στην τριγύρω περιοχή για να τον εντοπίσουν. Αυτά τα δύο καθήκοντα είναι υπό την αιγίδα του ίδιου εκλέκτορα π.χ. φρούρησης-καταδίωξης-αναζήτησης, αλλά με βάση το αν βλέπει τον παίκτη, θα εκτελέσει ένα από τα δύο. Στις πρώτες εκδοχές των δέντρων συμπεριφοράς, κάθε δευτερόλεπτο η ροή του δέντρου επέστρεφε στην ρίζα για να επανεκτίμηση ποιοι κόμβοι πρέπει να είναι ενεργοί. Αυτό μπορεί να γίνει πολύ κοστοβόρο για τον υπολογιστή ειδικά αν το δέντρο είναι αρκετά μεγάλο. Για την επίλυση αυτού του προβλήματος, σε πιο εξελιγμένες εκδόσεις, τα δέντρα κρατάνε αναφορά για το ποιος είναι ο ενεργός κόμβος και ο ίδιος θα εκτελέσει την διαδικασία αλλαγής ροής με βάση τον εκλέκτορα. Αφού τελειώσει η συμπεριφορά ή τα καθήκοντα αυτού του κόμβου γονέα, η ροή θα επανέλθει στην ρίζα.



Εικόνα 4.8 Δέντρο συμπεριφοράς.

6.2.3 Μηχανή πεπερασμένης κατάστασης

Η χρήση των δέντρων συμπεριφοράς, παρά την ευχέρεια που δίνουν στους προγραμματιστές, είναι μία τεχνολογία που τα τελευταία δέκα χρόνια έχει ξεκινήσει και χρησιμοποιείται σαν μία από τις σταθερές της βιομηχανίας. Ο προκάτοχος αυτής της σταθεράς ήταν η μηχανή πεπερασμένης κατάστασης, που ακόμα και τώρα χρησιμοποιείται σε αρκετούς AAA τίτλους. Όπως το δέντρο έχει καθήκοντα η μηχανή

έχει καταστάσεις που λένε στον αντίπαλο τι πρέπει να κάνει μία συγκεκριμένη χρονική στιγμή. Η μηχανή κρατάει μία κατάσταση που είναι ενεργή μέχρι να διαβάσει μία είσοδο που αναγνωρίζει ώστε να μεταβεί σε μία άλλη. Οι προγραμματιστές είναι αυτοί που ελέγχουν την εγκυρότητα της εισόδου που θα προκαλέσει αυτήν την μετάβαση αλλά και ποια, θα είναι η κατάσταση που θα ενεργοποιηθεί. Παρόλα αυτά οι πεπερασμένες μηχανές είναι ικανές να αντιδράσουν και σε γεγονότα, που πραγματοποιούνται σε πραγματικό χρόνο στο περιβάλλον, από τον ίδιο τον παίκτη ή από άλλους αντιπάλους ή δράστες.

6.2.4 Λόγοι μεταφοράς από μηχανές πεπερασμένων καταστάσεων σε δέντρα συμπεριφοράς

Οι μηχανές πεπερασμένων καταστάσεων και τα δέντρα συμπεριφοράς καταλήγουν στο ίδιο τελικό αποτέλεσμα, αλλά υπάρχουν μερικές διαφορές που δικαιολογούν την επιλογή της βιομηχανίας να περάσει σε πιο απλουστερους και εύχρηστους τρόπους δημιουργίας τεχνητής νοημοσύνης, όπως τα δέντρα. Αρχικά υπάρχει το πρόβλημα μεγάλου φόρτου εργασίας κατασκευής και διαδικασίας αποσφαλμάτωσης. Σε περίπτωση που η μηχανή έχει πολλές καταστάσεις αυξάνεται η πιθανότητα ύπαρξης bug, αλλά και ο αριθμός των μεταβάσεων που θα πρέπει να λαμβάνουν υπόψη οι προγραμματιστές. Το γεγονός ότι το σκέλος αυξάνεται δραματικά με την προσθήκη νέων καταστάσεων, καθιστά την υποστήριξη ολοένα και πιο δύσκολη. Μία μέθοδος που εφαρμόστηκε για να εξομαλύνει αυτό το πρόβλημα είναι η χρήση της ιεραρχικής μηχανής πεπερασμένης κατάστασης, που επιτρέπει την μετάβαση σε συλλογές ή ομάδες από καταστάσεις. Με αυτόν τον τρόπο μπορούν να δημιουργηθούν μικρότερου σκέλους μηχανές, που μεταβαίνουν η μία στην άλλη, αλλά στην πραγματικότητα το πρόβλημα υποστήριξης μεταφέρεται ένα επίπεδο υψηλότερα στην αρχιτεκτονική. Αντίθετα τα δέντρα λόγω της δομής τους, που ξεκινάνε από την ρίζα και φτάνουν στα φύλλα, είναι πιο κατανοητή και αρθρωτή, επιτρέποντας την ύπαρξη άμεσων αλλαγών στους κόμβους. Ο κύριος παράγοντας στην ευελιξία της κατασκευής ενός δέντρου, είναι ότι χρησιμοποιεί κλήσεις συναρτήσεων που εκτελούν προκαθορισμένα καθήκοντα αλλά δεν απαιτούν να γνωρίζουν σε ποιους κόμβους θα μεταβούν. Το αποτέλεσμα του κάθε καθήκοντος είναι επιτυχία ή αποτυχία, με τη ροή του δέντρου να αλλάζει, ανάλογα με το αποτέλεσμα.

Πηγές και Βιβλιογραφία

- ❖ [Learn Arch Viz](#) Complete UE4 Lighting Workflow for Arch Viz Artists | Light Types / Light Bake / Lightmass Settings
 - [Complete UE4 Lighting Workflow for Arch Viz Artists | Light Types / Light Bake / Lightmass Settings](#)

- ❖ [NerdyUnrealDev](#) UE4 Skylight Basics
 - [UE4 Skylight Basics](#)
- ❖ [A WAY OF LIFE STUDIOS](#) How to paint landscapes in Unreal Engine 4.18.3
 - [How to paint landscapes in Unreal Engine 4.18.3. For super new people.](#)
- ❖ [Nitrogen](#) 3 Levels of Procedural Grass in UE4 | UE4 Tutorial
 - [3 Levels of Procedural Grass in UE4 | UE4 Tutorial](#)
- ❖ [Ryan Laley](#) Unreal Engine 4 Tutorial - AI
 - [Unreal Engine 4 Tutorial - AI - Part 1 The Behaviour Tree](#)
- ❖ [Unreal Sensei](#) How to HIDE Texture REPETITION in Unreal Engine - UE4 Tutorial
 - [How to HIDE Texture REPETITION in Unreal Engine - UE4 Tutorial](#)
- ❖ [Unreal Sensei](#) How to INSTANTLY TEXTURE your landscapes in UE4 - Unreal Engine tutorial
 - [How to INSTANTLY TEXTURE your landscapes in UE4 - Unreal Engine tutorial](#)
- ❖ [WorldofLevelDesign](#) UE4: Easy 5-Step Process for Hand-Sculpting Perfect Landscapes Entirely in UE4
 - [UE4: Easy 5-Step Process for Hand-Sculpting Perfect Landscapes Entirely in UE4 - Tutorial](#)
- ❖ Άλμπερτ Τζούλιαν Μάγιερ. Γραφικά υπολογιστή/ψηφιακή επεξεργασία εικόνας. Σχολή Επιστήμης Υπολογιστών στο Τεχνικό Πανεπιστήμιο της Βιέννης
- ❖ Wikipedia. Normal mapping
 - https://en.wikipedia.org/wiki/Normal_mapping

- ❖ Wikipedia. Unreal Engine
 - https://en.wikipedia.org/wiki/Unreal_Engine
- ❖ Unreal Engine 4 Documentation. Get Started with UE4
 - <https://docs.unrealengine.com/4.27/en-US/Basics/GettingStarted/>
- ❖ Packtpub. An overview of Unreal Engine
 - <https://hub.packtpub.com/overview-unreal-engine/>
- ❖ Unreal Engine 4 Documentation. Directional Lights
 - <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LightingAndShadows/LightTypes/Directional/>
- ❖ Unreal Engine 4 Documentation. Sky Light
 - <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LightingAndShadows/LightTypes/SkyLight/>
- ❖ Unreal Engine 4 Documentation. Spot Light
 - <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LightingAndShadows/LightTypes/Spot/>
- ❖ Unreal Engine 4 Documentation. Point Lights
 - <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LightingAndShadows/LightTypes/Point/>
- ❖ Unreal Engine 4 Documentation. Lighting Basics
 - <https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/LightingAndShadows/Basics/>