



Τμήμα Τεχνών Ήχου και Εικόνας, Ιόνιο Πανεπιστήμιο

«Έρευνα και Ανάπτυξη Έργου μέσω Εφαρμοσμένων Τεχνικών Τρισδιάστατης
Μοντελοποίησης και Σχεδιασμού»

Μαγουλάς Γεώργιος TX2014052

Επιβλέπων Καθηγητής: Τηλιγάδης Κωνσταντίνος

Κέρκυρα,
Φεβρουάριος 2022

Συνοπτική περίληψη

Η παρούσα εργασία έχει σκοπό την ανάλυση και την κατανόηση των βασικών τεχνικών τρισδιάστατης μοντελοποίησης αντικειμένων που προορίζονται για χρήση σε βιντεοπαιχνίδια και εφαρμογές πραγματικού χρόνου. Αποτελεί συνοδευτικό κείμενο της δημιουργίας ενός ολοκληρωμένου τρισδιάστατου μοντέλου αεροσκάφους τύπου Boeing 767-300ER και μιας διαδραστικής εφαρμογής περιήγησης πρώτου προσώπου στον εσωτερικό και εξωτερικό χώρο του αεροπλάνου, με δυνατότητα πτήσης. Το εκτελέσιμο πρόγραμμα προορίζεται για υπολογιστές λογισμικού Windows. Το εσωτερικό σε μια πρώιμη έκδοση του παραχωρήθηκε στο φοιτητή, του τμήματος, Ανδρέα Βουλιάκη, για την εκπόνηση της ατομικής του εργασίας με θέμα τη δημιουργία του βιντεοπαιχνιδιού Last Chance.

Abstract

This paper aims to analyze and examine the basic techniques of 3D modeling of objects intended for use in video games and real-time applications. It operates as companion text to the creation of a 3D model of a Boeing 767-300ER aircraft and an interactive first-person, flight-enabled, tour application of the interior and exterior of the aircraft. The executable program is intended for Windows software computers. An early version of the interior was provided to the student, Andreas Vouliakis, for his individual project on the creation of the video game Last Chance.

Περίληψη

Στο πρώτο κεφάλαιο θα αναλυθούν οι βασικές μεθοδολογίες τρισδιάστατης μοντελοποίησης. Θα γίνει αναφορά στις περιπτώσεις εφαρμογής τους, στα θετικά και αρνητικά τους στοιχεία. Οι τεχνικές αυτές είναι: η Πολυγωνική, η Nurbs, η Ψηφιακή Γλυπτική, η Διαδικαστική, η Φωτογραμμετρία, η 3D Σάρωση και οι Προσομοιώσεις.

Στο δεύτερο κεφάλαιο θα αναλυθεί η διαδικασία του Rendering, καθώς και οι δύο βασικοί του τύποι, ο Offline και ο Real-time. Ακόμα θα γίνει αναφορά στους αλγορίθμους, Raytrace και Rasterization, καθώς και πώς οι τελευταίες εξελίξεις της τεχνολογίας έχουν αναβαθμίσει τα γραφικά και προμηνύουν μεγάλες αλλαγές στη διαδικασία παραγωγής.

Στο τρίτο, θα δοθεί μεγάλη σημασία στην τεχνική “High to Low Polygon Modeling”. Θα γίνει λεπτομερής επεξήγηση των διαδικασιών που απαιτούνται για την επίτευξη ενός λειτουργικού μοντέλου που προορίζεται για χρήση σε Real-time, πραγματικού χρόνου, εφαρμογές. Με πρώτη, τη θεμελιώδη τεχνική του Baking, ο αναγνώστης θα γνωρίσει πως μέσω προγραμμάτων

δημιουργούνται ειδικοί χάρτες, οι οποίοι μετατρέπουν πληροφορίες γεωμετρίας σε εικόνες, pixels, προκειμένου να αποδευτεθεί υπολογιστική ισχύς και μνήμη. Θα μάθει να αναγνωρίζει τους βασικούς αυτούς χάρτες και θα κατανοήσει τις ανάγκες που εξυπηρετεί ο καθένας. Ακόμα θα γίνει αναφορά στα προβλήματα που προκύπτουν κατά τη διαδικασία αυτής, καθώς και τους τρόπους αντιμετώπισής τους. Έπειτα θα μάθει για τα προγράμματα εφαρμογής υφών, textures, τη δημιουργία υλικών, καθώς και επιμέρους τεχνικές εφαρμογής εφέ σκόνης και φθοράς στα υλικά, που ενισχύουν την αίσθηση ρεαλισμού. Επιπλέον, θα γίνει αναφορά στην μέθοδο LOD, όπου δημιουργούνται αντικείμενα με πολλαπλά επίπεδα γεωμετρίας, με σκοπό τη βελτιστοποίηση της ταχύτητας εκτέλεσης των ζωντανού χρόνου εφαρμογών. Τέλος θα δοθούν ενδεικτικές μεθοδολογίες προετοιμασίας και οργάνωσης των εργασιών για μια σωστή και αξιόπιστη εκπόνηση των παραπάνω διαδικασιών.

Στο τέταρτο κεφάλαιο γίνεται ανάλυση των βασικών αρχών λειτουργίας των προγραμμάτων δημιουργίας βιντεοπαιχνιδιών. Ως πρόγραμμα αναφοράς χρησιμοποιήθηκε το Unity, στο οποίο δημιουργήθηκε και η εφαρμογή περιήγησης. Ξεκινώντας με την εισαγωγή των στοιχείων στο πρόγραμμα ο χρήστης θα μάθει να οργανώνει το περιβάλλον εργασίας του, ώστε να μπορεί να τα ταξινομεί και να τα αναζητεί εύκολα παρά την πολυπλοκότητα του “project” στο οποίο εργάζεται. Θα μάθει για τη δημιουργία σκηνών που θα περιέχουν τα μοντέλα με τις υφές και τα υλικά που έχει σχεδιάσει προηγουμένως. Έπειτα αναλύοντας τα διαθέσιμα συστήματα φωτισμού, εφαρμογής ουρανών, δημιουργίας τοπίων, της χρήσης καμερών και επιμέρους εργαλείων θα μπορέσει να δημιουργήσει ένα μοναδικό ψηφιακό περιβάλλον, το οποίο θα εμψυχώσει με τη χρήση προγραμματισμού και animation. Ακόμα θα μάθει για τους παράγοντες που επιβαρύνουν την επεξεργαστική ισχύ σε σχέση με την ποιότητα των γραφικών. Έτσι θα μπορεί να δημιουργήσει τα βέλτιστα γραφικά για την συσκευή αναπαραγωγής που προορίζεται το παιχνίδι ή η εφαρμογή του.

Στο πέμπτο κεφάλαιο θα γίνει μια πλήρης αναφορά στο Game Play, δηλαδή το παραδοτέο πρόγραμμα και τις οδηγίες περιήγησης και διάδρασης σε αυτό.

Περιεχόμενα

Εισαγωγή.....	6
Κεφάλαιο 1 ^ο : Βασικές Τεχνικές Τρισδιάστατης Μοντελοποίησης	7
Πολυγωνική Μοντελοποίηση (Polygon)	7
NURBS (Non-Uniform Rational B-Spline).....	8
Ψηφιακή Γλυπτική (Digital Sculpting).....	9
Διαδικαστική Μοντελοποίηση (Procedural).....	11
Photogrammetry/ 3D Scanning.....	12
Προσομοιώσεις (Simulations)	13
Κεφάλαιο 2 ^ο : Rendering	14
Κεφάλαιο 3 ^ο : Τεχνικές σχεδιασμού μοντέλων χαμηλής γεωμετρίας	16
High-Low poly.....	16
Ενδεικτική μεθοδολογία	17
LOD (Level Of Detail).....	17
UV Mapping και Unwrap	18
Baking	19
Βασικοί Χάρτες.....	20
Παράμετροι Baking και Προβληματικές	23
Texturing.....	29
Materials.....	29
Κεφάλαιο 4 ^ο : Πρόγραμμα δημιουργίας βιντεοπαιχνιδιών	33

Οργάνωση και εισαγωγή αντικειμένων	33
Scene	33
Render Pipeline	34
Material	34
Prefab	35
LOD και Texture Resolution	35
Camera	36
Environment.....	36
Lighting.....	36
Baked Mixed layer mask.....	38
Reflection Probe.....	38
Decal Projector.....	38
Post Process	38
Scripts	39
Terrain.....	39
Κεφάλαιο 4°: Game Play.....	40
Συμπεράσματα	40
Βιβλιογραφία	42
Ιστογραφία	42
Παράρτημα.....	43

Εισαγωγή

Η δημιουργία τρισδιάστων αντικειμένων αποτελεί μια διαδικασία εξειδικευμένη και αρκετά απαιτητική. Οι μέθοδοι και οι τεχνικές σχεδιασμού ποικίλουν και η διαδικασία του σχεδιασμού μπορεί να γίνει αρκετά αυστηρή, τεχνική και χρονοβόρα. Τα στάδια είναι αρκετά συγκεκριμένα και αλληλένδετα. Ένα αμελητέο λάθος στα αρχικά επίπεδα σχεδιασμού μπορεί να αποδειχθεί κρίσιμο και μη επιλύσιμο σε ένα πιο προχωρημένο στάδιο. Ένας νέος σχεδιαστής είναι πολύ πιθανό να συγχυστεί και να αφιερώσει πολύ περισσότερο χρόνο από ότι προέβλεπε ή να καταλήξει πολλές φορές σε αντικείμενα πολύ μεγάλης γεωμετρίας, σύνθετα, τα οποία είναι μη λειτουργικά και δύσκολο να επεξεργαστούν. Είναι σημαντικό λοιπόν να γνωρίζει τις βασικές αρχές του τρισδιάστατου σχεδιασμού και μοντελοποίησης, ώστε να μπορεί να οργανώσει ένα πλάνο εργασίας λειτουργικό και αποτελεσματικό.

Πεδίο έρευνας, και εφαρμογής της διπλωματικής εργασίας, είναι τα στάδια και οι διαδικασίες που απαιτούνται για τη δημιουργία τρισδιάστατων αντικειμένων που προορίζονται για εφαρμογές εκτέλεσης σε πραγματικό χρόνο. Η επιλογή του συγκεκριμένου θέματος έγινε συνειδητά, μιας και πρόκειται για μια από τις πιο εξειδικευμένες διαδικασίες μοντελοποίησης και περιλαμβάνει μεγάλο εύρος τεχνικών, εξαιρετικά χρήσιμων για όλα τα είδη της τρισδιάστατης μοντελοποίησης. Ωστόσο η επιλογή ενός αεροπλάνου αποδείχθηκε ιδιαίτερα απαιτητική σε αντίθεση με μια πιο «ελεύθερη» σκηνή ανοιχτού περιβάλλοντος. Λόγω της πολυπλοκότητας και της αυστηρότητας που απαιτείται τόσο κατά τον σχεδιασμό όσο και κατά την έρευνα, τη συλλογή εικόνων, σχεδίων και αρχείων αναφοράς. Για τη δημιουργία και τη δυνατότητα “συναρμολόγησης” πολλών μεμονωμένων αντικειμένων, τα οποία πρέπει να συντελέσουν μια πολύ συγκεκριμένη ολότητα, τίθεται εξαιρετικά σημαντική η τήρηση και η κατανόηση των επιπέδων και διαδικασιών του σχεδιασμού που πρόκειται να αναλυθούν. Ως αποτέλεσμα, αυτής της μελέτης, έρχεται η γνώση της λειτουργίας αρκετών διαφορετικών προγραμμάτων, η διεύρυνση των ικανοτήτων μοντελοποίησης, σχεδιασμού και οργάνωσης ενός αποδοτικού πλάνου εργασίας, καθώς ακόμα, η εύρεση των κατάλληλων πηγών για επίλυση των προβλημάτων που προκύπτουν.

Έτσι, σκοπός της εργασίας αυτής είναι η παρουσίαση και η κατανόηση των διαδικασιών αυτών από τον αναγνώστη, σχεδιαστή, ώστε να μπορέσει να οργανώσει ένα πλάνο εργασίας λειτουργικό, αποδοτικό και κυρίως ευχάριστο, αφήνοντας περισσότερο χρόνο στην ανάπτυξη καλλιτεχνικής δημιουργικότητας.

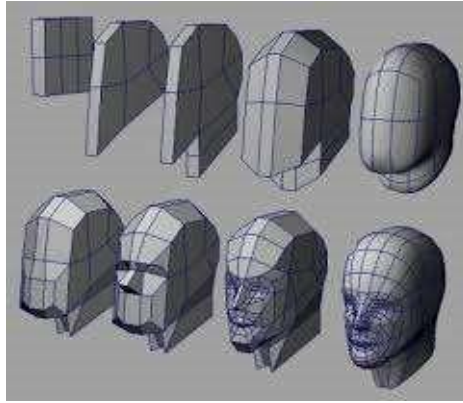
Για την υλοποίηση της εργασίας χρησιμοποιήθηκαν τα εξής προγράμματα: Adobe Photoshop, Unity, Substance Painter, Autodesk 3ds Max, Blender Studio, Quixel Bridge, Substance Alchemist.

Κεφάλαιο 1^ο:

Βασικές Τεχνικές Τρισδιάστατης Μοντελοποίησης

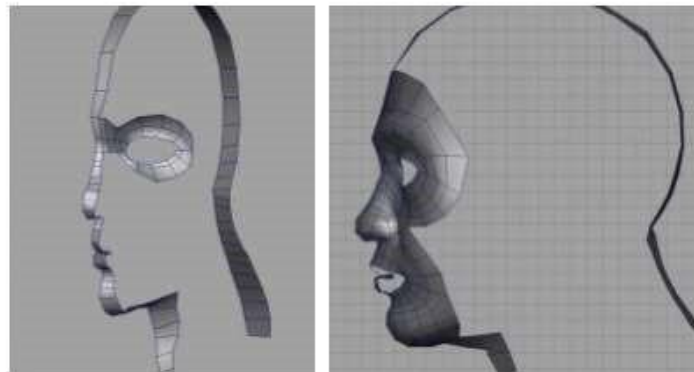
Πολυγωνική Μοντελοποίηση (Polygon)

Πρόκειται για τον πλέον ευρέως διαδεδομένο τρόπο μοντελοποίησης στον τομέα της διαφήμισης, του κινηματογράφου και των βίντεο παιχνιδιών. Ανοίγοντας ένα πρόγραμμα τρισδιάστατης μοντελοποίησης το πρώτο που θα παρατηρήσει κανείς είναι το μενού των “Primitive Objects”, που δημιουργεί βασικά σχήματα όπως ο κύβος, η σφαίρα, ο κύλινδρος, ο ρόμβος κ.ο.κ. Αναλύοντας τα μέρη ενός σχήματος, όπως ο κύβος μπορεί κανείς να καταλάβει πώς προκύπτουν οι βασικοί όροι για την συγκεκριμένη πρακτική μοντελοποίησης. Ο κύβος αποτελείται από έξι έδρες (faces). Οι γραμμές που περιβάλλουν τα πολύγωνα και τα ενώνουν μεταξύ τους ονομάζονται ακμές (edges). Ενώ τα σημεία που τέμνουν τις ακμές ονομάζονται κορυφές (vertices). Αυτά είναι τα δομικά στοιχεία ενός πολυγώνου (polygon), face, edge, vertices. Είναι σημαντικό να σημειωθεί πως για να υπάρξει ένα πολύγωνο πρέπει να υπάρχουν τουλάχιστον τρεις ακμές και τρεις κορυφές, ώστε να δημιουργηθεί τουλάχιστον ένα τρίγωνο. Οπότε, πέρα από τον συγκεκριμένο περιορισμό τα πολύγωνα μπορούν να έχουν οποιονδήποτε αριθμό ακμών και κορυφών (n-gons). Ωστόσο, συνηθίζεται να χρησιμοποιούνται πολύγωνα με τέσσερις ακμές (quads) σε περιπτώσεις offline rendering και τριγωνικά (triangle) σε realtime. Ο σχεδιαστής καλείται να δημιουργήσει τα αντικείμενα με την επεξεργασία των παραπάνω στοιχείων χρησιμοποιώντας συγκεκριμένες μεθόδους. Με τη μέθοδο πολυγωνικής μοντελοποίησης Box/Subdivision ο σχεδιαστής χρησιμοποιεί ως βάση ένα primitive object. Έπειτα χρησιμοποιώντας βασικές εντολές, όπως extrude (εξαγωγή), bridge (ένωση), bevel, chamfer (διαίρεση) κ.ο.κ, δημιουργεί μοντέλα χαμηλής γεωμετρίας τα οποία υποδιαιρεί, ώστε να εξομαλυνθεί και να επεξεργαστεί εκ νέου δημιουργώντας επίπεδα μεγαλύτερης λεπτομέρειας. Πρόκειται για την πιο απλή και εύκολη μέθοδο για να ξεκινήσει κανείς να ασχολείται με την τρισδιάστατη μοντελοποίηση.



Εικόνα 1. Box/Subdivision Modeling

Μια άλλη τεχνική είναι η Edge/Contour. Πρόκειται για έναν τρόπο με τον οποίο το μοντέλο χτίζεται κομμάτι-κομμάτι, έως ότου να προκύψει το τελικό ολοκληρωμένο σχήμα. Δεν απαιτεί την χρήση ενός primitive object ως βάση, αλλά ένα οποιοδήποτε πολύγωνο, plane. Στη συνέχεια δημιουργούνται βασικές ομάδες πολυγώνων, οι οποίες δημιουργούν τη βασική ροή της επιθυμητής γεωμετρίας σύμφωνα με της ανάγκες του μοντέλου. Έπειτα συμπληρώνονται τα κενά που υπάρχουν μεταξύ των ακμών, ώστε να δημιουργηθεί η συνολική γεωμετρία.



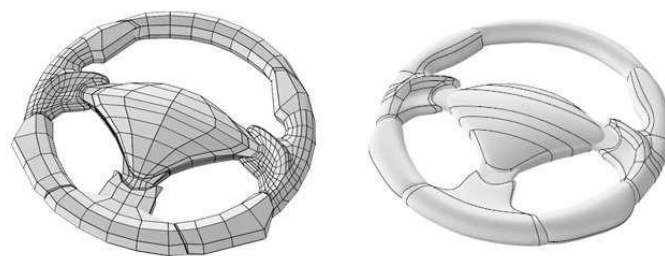
Εικόνα 2. Edge/Contour Modeling

NURBS (Non-Uniform Rational B-Spline)

Βασίζεται πάνω σε ένα παλιό μαθηματικό μοντέλο, το οποίο χρησιμοποιούσαν οι σχεδιαστές για να δημιουργήσουν και αναπαραστήσουν καμπύλες στο χαρτί.¹ Στη συγκεκριμένη τεχνική οι επιφάνειες ορίζονται από καμπύλες, η ροή των οποίων επηρεάζεται από σταθμισμένα σημεία ελέγχου. Μετακινώντας και περιστρέφοντας τα σημεία ελέγχου καθορίζεται η ροή της καμπύλης και της επιφάνειας. Πρόκειται για πραγματικές καμπύλες και όχι για ένα σύνολο από ευθείες με ακμές. Επομένως οι επιφάνειες δεν αποδίδονται με πολύγωνα, οπότε το αντικείμενο δεν έχει όψεις, ακμές και άκρα. Η λογική της συγκεκριμένης τεχνικής θυμίζει αυτή των

¹ Πιερ Μπεζιέ(1962), καμπύλη Bézier

προγραμμάτων που λειτουργούν με διανύσματα, όπως το Adobe Illustrator. Το μεγάλο της προτερήμα είναι ότι δεν τίθενται θέματα ποιότητας κατά την μεγέθυνση του μοντέλου, λόγω της έλλειψης ακμών. Αντίθετα με τα πολυγωνικά μοντέλα που σε περιπτώσεις μεγέθυνσης η ποιότητά τους “πέφτει” και θα πρέπει να υποστούν υποδιαιρέσεις, ώστε να έρθουν ξανά στο σημείο που δεν παρατηρούνται ακμές τους. Αυτό καθιστά τα μοντέλα τύπου NURBS να δεσμεύουν λιγότερη υπολογιστική μνήμη και ισχύ. Επιπλέον δεν τίθενται θέματα τοπολογίας σε αντίθεση με τα πολυγωνικά, που η σωστή διαχείριση της είναι κρίσιμη. Η ένωση και η αφαίρεση τμημάτων γίνεται χωρίς παραμορφώσεις με τη χρήση απλών εντολών. Η συγκεκριμένη τεχνική βρίσκει εφαρμογή κυρίως σε ιατρικούς, βιομηχανικούς και μηχανολογικούς τομείς, όπου η ακρίβεια των μοντέλων είναι απαραίτητη. Ακόμα χρησιμοποιείται ιδιαίτερα σε περιπτώσεις που το μοντέλο θα πρέπει να εκτυπωθεί μέσω 3D εκτυπωτή ή να χαραχτεί με cnc ή laser cutter. Ένα από τα αρνητικά της συγκεκριμένης τεχνικής είναι ότι προκειμένου να χρησιμοποιηθεί ένα NURBS μοντέλο για Render, θα πρέπει να υποστεί πολλές μετατροπές, προκειμένου να αποδοθούν σωστά οι ανακλάσεις του φωτός στην επιφάνειά του, μιας και οι περισσότεροι αλγόριθμοι υπολογίζουν το φως σε σχέση με το πως προσπίπτει πάνω σε όψεις και όχι σε καμπύλες.²



Subdivision surface

NURBS surfaces

Εικόνα 3. Subdivision / NURBS μοντέλα

Ψηφιακή Γλυπτική (Digital Sculpting)

Η ψηφιακή γλυπτική αποτελεί την πιο καλλιτεχνική εκδοχή της τρισδιάστατης μοντελοποίησης. Η λογική των προγραμμάτων είναι η προσομοίωση της κλασσικής γλυπτικής με πηλό. Χρησιμοποιεί επίσης κορυφές, ακμές και όψεις, όπως οι πολυγωνικές μέθοδοι. Ωστόσο η μεγάλη διαφορά, η οποία την καθιστά πολύ βολική στη χρήση είναι τα εργαλεία επεξεργασίας που προσφέρει. Μέσω βουρτσών (brushes) ο χρήστης μπορεί να ωθεί ή να εξωθήσει επιφάνειες, να τις ομαλοποιήσει ή να τις κάνει πιο τραχιές, Ορίζοντας μια περιοχή

² Chopine, Ami (2011). 3D Art Essentials, The Fundamentals of 3D Modeling, Texturing, and Animation: p. 51

επιρροής και έντασης, αναδιαμορφώνει τη γεωμετρία με βάση τον τύπο και τις ρυθμίσεις της βούρτσας. Στην ουσία δίνει μια πολύ πιο ζωγραφική αίσθηση ειδικά με την χρήση γραφίδας, που θεωρείται απαραίτητο εργαλείο για την συγκεκριμένη τεχνική. Ακόμα δίνει τη δυνατότητα στον χρήστη να έχει πολύ γρήγορα και εύκολα αποτελέσματα σε αρκετά σύνθετες δημιουργίες. Ωστόσο, είναι σημαντικό να σημειωθεί πως η ψηφιακή γλυπτική ενδείκνυται κατά βάση για τη δημιουργία οργανικών αντικειμένων και όχι τόσο για hard surface, με ισχυρές ακμές, μοντελοποίηση. Πολύ συχνά προγράμματα γλυπτικής χρησιμοποιούνται για την επιμέρους χάραξη λεπτομερειών, σε μοντέλα που έχουν δημιουργηθεί με την παραδοσιακή πολυγωνική τεχνική, μιας και με αυτή θα ήταν αρκετά χρονοβόρο και δύσκολο. Όπως και στην πολυγωνική μοντελοποίηση έτσι και στην γλυπτική υπάρχουν διάφορες τεχνικές που εφαρμόζονται ανάλογα την συνθήκη. Η πιο βασική είναι απευθείας χάραξη και διαμόρφωση ενός αντικειμένου, όπου θα έχει οριστεί εξ αρχής το ποσοστό της γεωμετρίας του, ανάλογα με τις ανάγκες της τελικής του ανάλυσης, ώστε οι πληροφορίες να είναι εμφανείς. Μια πιο έξυπνη και λειτουργική τεχνική που μας προσφέρουν τα σύγχρονα προγράμματα, είναι αυτή της δυναμικής υποδιαίρεσης (dynamic subdivide), όπου η βασική φόρμα μπορεί να είναι χαμηλής γεωμετρίας και όταν δημιουργηθεί η ανάγκη για μεγαλύτερη ανάλυση, επιτρέπει τη δημιουργία υποδιαίρεσης, καθώς όμως διατηρεί και το επίπεδο της χαμηλής γεωμετρίας.³ Οι υποδιαίρεσεις μπορούν να είναι πολλές, αλλά όλες αλληλοεπηρεάζονται και η πρόσβαση στο κάθε επίπεδο λεπτομέρειας παραμένει ανεξάρτητα από το πλήθος τους. Ο χρήστης έχει λοιπόν πρόσβαση στα διάφορα επίπεδα χωρίς να χρειάζεται να υπολογίσει την τελική ανάλυση του αντικειμένου εξ αρχής. Συνεπώς, όταν θέλει να κάνει μαζικές αλλαγές δουλεύει στο επίπεδο της χαμηλής γεωμετρίας, ενώ όταν θέλει να κάνει μικρές αλλαγές, που χρειάζεται λεπτομέρεια, δουλεύει σε μεγαλύτερα.

Μια άλλη σημαντική λειτουργία είναι αυτή της δυναμικής τοπολογίας (dynamic topology/tessellation), με την οποία προστίθεται ή αφαιρείται γεωμετρία μόνο στα σημεία από όπου έχει περάσει η βούρτσα.⁴ Η συγκεκριμένη τεχνική λειτουργεί μόνο με τριγωνικά πολύγωνα, οπότε τα προγράμματα προειδοποιούν πριν τη χρήση ότι η τοπολογία του μοντέλου θα μετατραπεί αυτόματα σε τριγωνική. Τέλος, τις περισσότερες φορές τα μοντέλα που έχουν δημιουργηθεί με γλυπτική καταλήγουν να έχουν μεγάλα επίπεδα γεωμετρίας, με λανθασμένη ροή. Αυτό, τα καθιστά βαριά στη διαχείρισή τους από τον υπολογιστή, αλλά και μη λειτουργικά για περαιτέρω διαδικασίες, όπως για παράδειγμα το animation. Έτσι, η τεχνική

³ <http://docs.pixologic.com/user-guide/3d-modeling/modeling-basics/dynamic-subdivision/>

⁴ https://docs.blender.org/manual/en/latest/sculpt_paint/sculpting/adaptive.html

που ακολουθείται είναι αυτή της αναδιαμόρφωσης της τοπολογίας (retopology). Αυτό γίνεται είτε χειροκίνητα, είτε αυτοματοποιημένα μέσω παραμετροποιήσιμων αλγορίθμων, τους remesher.⁵

Διαδικαστική Μοντελοποίηση (Procedural)

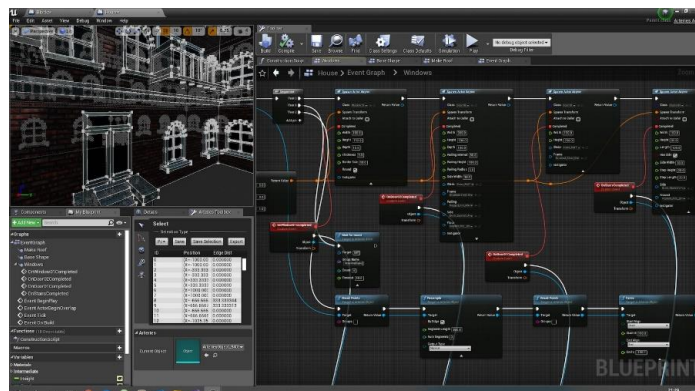


Εικόνα 4. Παράδειγμα Ψηφιακής Γλυπτικής

Πρόκειται για έναν τρόπο μοντελοποίησης πολύ διαφορετικό από όσους έχουν ήδη αναφερθεί. Η δημιουργία των μοντέλων γίνεται μέσω αλγορίθμων και ο σχεδιαστής δεν έχει την άμεση επαφή κατά την επεξεργασία τους, όπως στα πολυγωνικά ή στα NURBS μοντέλα. Μέσω παραμέτρων και κανόνων μπορεί να δημιουργήσει πολύπλοκα αντικείμενα ακόμα και ολόκληρες σκηνές σε ελάχιστο χρόνο σε σχέση με τις κλασικές τεχνικές. Επειδή τα αντικείμενα είναι παραμετροποιήσιμα, χρησιμοποιείται ιδιαίτερα σε περιπτώσεις όπου τα στοιχεία της σκηνής είναι παρόμοια, αλλά όχι τέλειοι κλώνοι. Ένα τέτοιο παράδειγμα είναι η δημιουργία ενός δάσους ή μιας πόλης. Στις περιπτώσεις αυτές ο σχεδιασμός των μοντέλων, με τις κλασικές τεχνικές, θα ήταν δραματικά χρονοβόρος. Συγκριμένα για τη δημιουργία οργανικών αντικειμένων, όπως τα δέντρα, χρησιμοποιούνται παράμετροι και κανόνες που βασίζονται στη φρακταλική γεωμετρία. Υπάρχουν δύο βασικές κατηγορίες στην συγκεκριμένη τεχνική. Η πρώτη βασίζεται στα εργαλεία (tool based). Τα εργαλεία αυτά, είτε τα έχει δημιουργήσει ο χρήστης ορίζοντας τους κανόνες και τις παραμέτρους τους, είτε τα βρίσκει διαθέσιμα στα προγράμματα. Ένα παράδειγμα έτοιμου εργαλείου που συναντάει κανείς στα κλασικά προγράμματα τρισδιάστατης μοντελοποίησης, όπως το 3dsmax, είναι αυτό της δημιουργία σκαλιών και παραθύρων. Η δεύτερη κατηγορία είναι αυτή των geometry shaders. Πρόκειται για ένα είδος μικρών προγραμμάτων που επηρεάζουν τη γεωμετρία ή τις υφές κατά τη διαδικασία της φωτοσκίασης (render). Η συγκεκριμένη τεχνική χρησιμοποιείται σε μεγάλο βαθμό για την αναπαράσταση εδάφους σε διαδραστικές εφαρμογές. Με τη χρήση ενός plane

⁵ <http://docs.pixologic.com/user-guide/3d-modeling/topology/zremesher/>

μεγάλης γεωμετρίας ως βάση, μέσω παραμετροποιήσιμων κόμβων (nodes) και μαθηματικών σχέσεων εξωθείται η γεωμετρία, ώστε να δημιουργηθεί το επιθυμητό ανάγλυφο (displacement). Οι πλέον εξελιγμένες κάρτες γραφικών που διαθέτουν DirectX 10 ικανότητες και άνω, όπως geometry shader, stream output, render to 3D texture, μπορούν να χρησιμοποιηθούν για να παράγουν τις τελικές εικόνες τοπίων μεγάλης πολυπλοκότητας με τη συγκεκριμένη τεχνική.⁶ Γνωστά προγράμματα γέννησης τοπίων είναι τα: Terragen, Bryce, and Vue. Ακόμα ένα πρόγραμμα που χρησιμοποιείται σε μεγάλο βαθμό στον κινηματογράφο, τη διαφήμιση και τα ειδικά εφέ, και λειτουργεί εξ' ολοκλήρου με διαδικαστική λογική είναι το Houdini, καθώς και το υποπρόγραμμα Tyflow, στο 3dsMax.⁷ Τέλος, υπάρχουν προγράμματα που λειτουργούν με την αντίστοιχη φιλοσοφία για την παραγωγή υφών. Ορισμένα από αυτά είναι το Mixer της Quixel και το Substance Alchemist της Adobe. Πρόκειται για δύο προγράμματα που ανταγωνίζονται πιο θα υπερισχύσει στην αγορά. Η βάση δεδομένων και των δύο προέρχεται από τρισδιάστατα σαρωμένα αντικείμενα.



Εικόνα 5. Παράδειγμα Procedural μοντελοποίησης, Unreal Engine

Photogrammetry/ 3D Scanning

Η φωτογραμμετρία και η τρισδιάστατη σάρωση είναι δύο τεχνικές που προσπαθούν να πετύχουν το ανάλογο αποτέλεσμα.

Η φωτογραμμετρία είναι μια διαδικασία διαδοχικής φωτογράφισης του αντικειμένου ανά λίγες μοίρες, καλύπτοντας όλες τις όψεις του, οι οποίες θα συντεθούν σε ειδικό πρόγραμμα το οποίο θα δημιουργήσει στη συνέχεια ένα τρισδιάστατο αντικείμενο όμοιο του αρχικού. Το μοντέλο αυτό εμφανίζει κοινά χαρακτηριστικά με το αρχικό, σε σχήμα και υφή, ωστόσο η γεωμετρία του πάντα θα εμφανίζει κάποιες ατέλειες, οι οποίες ανάλογα με τη χρήση για την οποία

⁶ <https://developer.nvidia.com/gpugems/gpugems3/part-i-geometry/chapter-1-generating-complex-proceduralterrains-using-gpu>

⁷ <http://docs.tyflow.com/why/>

προορίζεται, θα πρέπει να βελτιστοποιηθούν κατάλληλα. Η φωτογραμμετρία αν και πολύπλοκη πολλές φορές και χρονοβόρα μπορεί να εφαρμοστεί από οποιονδήποτε διαθέτει έναν ηλεκτρονικό υπολογιστή και μια φωτογραφική μηχανή.

Η τρισδιάστατη σάρωση είναι μια πιο σύγχρονη τεχνική η οποία χρησιμοποιεί έναν ειδικό σαρωτή που, όπως και στην φωτογραμμετρία, χρειάζεται διαδοχικά δείγματα από όλες τις όψεις του αντικειμένου προκειμένου να αναπαράγει ένα τρισδιάστατο αντίγραφο. Η ποιότητα και η ταχύτητα επεξεργασίας με τη χρήση του τρισδιάστατου σαρωτή είναι κατά πολύ ανώτερη σε σχέση με την φωτογραμμετρία. Ωστόσο, είναι πολύ πιο ακριβή μέθοδος. Τα τρισδιάστατα αυτά αντίγραφα και στις δύο περιπτώσεις μπορεί αντιμετωπίσουν θέματα ποιότητας στη γεωμετρία. Η πιο συνήθης διαδικασία είναι η δημιουργία ενός μοντέλου με σωστή τοπολογία που να αγκαλιάζει το αρχικό μοντέλο και στη συνέχεια η εφαρμογή της τεχνικής baking (ψησίματος), ώστε να ανακτηθούν πληροφορίες γεωμετρίας που θα αποδοθούν ως υφές.

Υπάρχουν πολλές και μεγάλες συλλογές σαρωμένων αντικειμένων και υφών που διατίθενται είτε για σκιαγράφηση (render), είτε για πραγματικού χρόνου (real-time) εφαρμογές. Μια από αυτές είναι η συλλογή Megascan της Quixel, η οποία παρέχεται δωρεάν υπό ορισμένες προϋποθέσεις⁸

Προσομοιώσεις (Simulations)

- Physics
- Cloth
- Soft body
- Fluid
- Fire and smoke
- Ocean
- Particles



Εικόνα 6. Προσομοίωση νερού, 3ds Max, Vray Renderer. Σκάφος Cristal 25 μ.

Πρόκειται για μια διαδικασία κατά την οποία ο χρήστης ορίζει παραμέτρους, για τις φυσικές ιδιότητες των αντικειμένων. Βάση αυτών ο υπολογιστής, υπολογίζει κατά ποιον τρόπο τα αντικείμενα θα κινηθούν και θα αλληλοεπιδράσουν ανά καρέ. Στη συνέχεια το τελικό αποτέλεσμα μπορεί να χρησιμοποιηθεί για τη δημιουργία animation, είτε στατικής εικόνας. Το πιο συχνό και απλό παράδειγμα προσομοίωσης είναι για τη δημιουργία υφασμάτων. Αντί, να μοντελοποιήσει κανείς το ύφασμα με έναν από τους κλασσικούς χρονοβόρους τρόπους, με το

⁸ <https://quixel.com/megascans/home>

simulation θα έχει ένα πολύ πιο γρήγορο αποτέλεσμα. Ειδικά σε πιο σύνθετες περιπτώσεις, όπως εκρήξεις, καπνούς, κύματα, φωτιές, υγρά στοιχεία, αφρούς κοκ. η επιλογή της τεχνικής της προσομοίωσης αποτελεί μονόδρομο. Τα περισσότερα προγράμματα παρέχουν διάφορα μενού για προσομοιώσεις. Ωστόσο, υπάρχουν και προγράμματα σχεδιασμένα ειδικά για προσομοιώσεις, όπως το Phoenix FD.⁹

Κεφάλαιο 2^ο:

Rendering

Οι δύο βασικοί αλγόριθμοι render σύμφωνα με τους οποίους ορίζουμε το πολυγωνικό πλέγμα των μοντέλων μας, είναι ο offline και ο real-time.

Ο offline renderer χρησιμοποιείται κατά βάση για την παραγωγή εικόνων ή βίντεο και αφορούν συνήθως μοντέλα μεγάλης γεωμετρίας με πολλές υποδιαιρέσεις. Η παραγωγή της τελικής εικόνας ή του κάθε καρέ μπορεί να διαρκέσει από λίγα δευτερόλεπτα, λεπτά έως και αρκετές ώρες ανάλογα την πολυπλοκότητα της σκηνής και της υπολογιστικής ισχύος που διαθέτει το μηχάνημά μας. Οι offline renderers χρησιμοποιούν σύνθετους αλγόριθμους όπως, ο raytracing, που προσπαθούν να προσομοιώσουν τις ανακλάσεις του φωτός, τις σκιές με μια ρεαλιστική προσομοίωση. Ο συγκεκριμένος αλγόριθμος καθώς και άλλοι, μπορούν να φτάσουν την παραγόμενη εικόνα τόσο κοντά στον ρεαλισμό, που ο παρατηρητής να μην αντιλαμβάνεται ότι πρόκειται για εικόνα φτιαγμένη στον υπολογιστή. Ωστόσο, αυξάνουν δραματικά τις απαιτήσεις υπολογιστικής ισχύος και τους χρόνους διαδικασίας του rendering. Η συγκεκριμένη μέθοδος βρίσκει μεγάλη εφαρμογή στον κινηματογράφο, στα αρχιτεκτονικά έργα στο βιομηχανικό σχεδιασμό και τη διαφήμιση, όπου δεν υπάρχει ανάγκη παραγωγής εικόνων και βίντεο σε πραγματικό χρόνο.

Ο real-time renderer καλύπτει αυτό το κενό. Χρησιμοποιείται κατά κύριο λόγο σε βιντεοπαιχνίδια και διαδραστικές εφαρμογές. Οι εικόνες παράγονται όσο ο παίχτης-χρήστης, τρέχει την εφαρμογή. Ο στόχος του real-time render είναι να δημιουργεί τόσες εικόνες ανά δευτερόλεπτο, όσος ο αριθμός του βασικού ρυθμού ανανέωσης των οθονών, δηλαδή τα 60hz, άρα τουλάχιστον 60 καρέ ανά δευτερόλεπτο, για εφαρμογές και βιντεοπαιχνίδια. Ενώ για το βίντεο στόχος είναι τα 24 ή τα 30 καρέ ανά δευτερόλεπτο.¹⁰ Για την επίτευξη του συγκεκριμένου στόχου, είναι σημαντικό να υπολογιστεί κατά τον σχεδιασμό, ο βαθμός της πολυπλοκότητας της γεωμετρίας των αντικειμένων, καθώς και οι υπολογιστικές δυνατότητες

⁹ <https://www.chaosgroup.com/phoenix-fd/3ds-max>

¹⁰ <https://www.chaosgroup.com/blog/real-time-ray-traced-and-rasterized-rendering-explained>

που θα διαθέτει η συσκευή για την οποία προορίζεται. Για να γίνει αυτό κατανοητό, τα γραφικά που μπορεί να τρέξει ένας σύγχρονος ηλεκτρονικός υπολογιστής, που διαθέτει έναν επεξεργαστή με υψηλή συχνότητα και αρκετούς πυρήνες, καθώς και μια ισχυρή κάρτα γραφικών, θα είναι κατά πολύ ανώτερα από αυτά που μπορεί να τρέξει ένα κινητό τηλέφωνο εξίσου σύγχρονο. Βέβαια και στις δύο κατηγορίες σε σχέση με τα προηγούμενα χρόνια υπάρχει ραγδαία εξέλιξη, όσον αφορά την ποιότητα των γραφικών.

Μια τεχνική που έχει χρησιμοποιηθεί σε μεγάλο βαθμό ιστορικά, στους real-time renderers, είναι αυτή του rasterization. Πρόκειται για μια διαδικασία κατά την οποία ο υπολογιστής μετατρέπει τα πολύγωνα των τρισδιάστατων αντικειμένων σε pixel ή κουκίδες. Το κάθε pixel έχει μια αρχική τιμή χρώματος, η οποία επηρεάζεται ανάλογα με την απόσταση του από την πηγή του φωτός.¹¹ Συχνά δεν αρκεί μια φωτιστική πηγή, αλλά πολλαπλές, για να αποδοθεί μια φυσική αίσθηση σκιών και φωτός, διότι ο συγκεκριμένος αλγόριθμος δεν λειτουργεί όπως ο raytrace, που προσομοιώνει την πραγματική ανάκλαση του φωτός. Η διαδικασία του rasterization, μιας και δεν απαιτεί μεγάλη υπολογιστική ισχύ, μειώνει κατά πολύ μεγάλο βαθμό τον χρόνο του rendering και γι' αυτό μέχρι πρότινος αποτελούσε μονόδρομο, για τα γραφικά πραγματικού χρόνου. Οι offline renderers χρησιμοποιούνταν για μέγιστη απόδοση ρεαλισμού, σε στατικές εικόνες ή βίντεο, και οι real-time για εφαρμογές πραγματικού χρόνου, αλλά με κατώτερα γραφικά, αφού δεν ήταν δυνατή η χρήση του raytracing από συμβατικά μηχανήματα. Ωστόσο, μια πρόσφατη τεχνολογική επίτευξη έχει επιτρέψει στις πιο σύγχρονες κάρτες γραφικών NVIDIA και AMD να τρέχουν αλγορίθμους πραγματικού χρόνου raytracing. Αυτό το γεγονός έχει αποτελέσει μεγάλο πλεονέκτημα για τους real-time renderers, αφού πλέον μπορούν να δημιουργήσουν μια πολύ πιο ρεαλιστική αναπαράσταση φωτός, σκιών, διαφανειών και αντανακλάσεων, ανάλογη των offline. Αυτό έχει κεντρίσει ιδιαίτερα το ενδιαφέρον των αρχιτεκτόνων και των σχεδιαστών βιομηχανικών προϊόντων, που μέχρι πρότινος ασχολούνταν αποκλειστικά με τους offline. Η συγκεκριμένη τεχνολογία αν και σε πρώιμο στάδιο έχει ήδη εφαρμοστεί σε ορισμένα παιχνίδια, που υπάρχουν στην αγορά, αν και απαιτούν την χρήση των ανάλογων καρτών γραφικών. Πρόκειται για μια τεχνολογία πολλά υποσχόμενη που πιθανότατα στο μέλλον, θα αλλάξει ριζικά τη διαδικασία παραγωγής. Ήδη μεγάλες εταιρείες offline renderers, έχουν βγάλει δοκιμαστικές εκδόσεις real-time, Chaos Vantage, Blender EEVEE.¹²

¹¹ <https://blogs.nvidia.com/blog/2018/03/19/whats-difference-between-ray-tracing-rasterization/>

¹² <https://www.chaosgroup.com/vantage>, https://wiki.blender.org/wiki/Reference/Release_Notes/2.80/EEVEE

Κεφάλαιο 3^ο:

Τεχνικές σχεδιασμού μοντέλων χαμηλής γεωμετρίας

Για τη διασφάλιση της σωστής ροής των γραφικών, ειδικά στα βιντεοπαιχνίδια, ο σχεδιαστής, καλλιτέχνης, οφείλει να ακολουθήσει μια σειρά από συγκεκριμένες διαδικασίες ώστε να δημιουργήσει ένα τελικό αντικείμενο χαμηλής γεωμετρίας.

High-Low poly

Συνήθως ξεκινά με την δημιουργία ενός μοντέλου υψηλής γεωμετρίας, με οποιαδήποτε τεχνική επιθυμεί ή και με τον συνδυασμό τους, χωρίς κανέναν περιορισμό. Σε οποιοδήποτε στάδιο της μοντελοποίησης δημιουργεί ένα όμοιο αντικείμενο, χαμηλότερης γεωμετρίας. Είναι σημαντικό τα δύο αντικείμενα να έχουν τα ίδια κέντρα και οι επιφάνειές τους να εφάπτονται όσο το δυνατόν περισσότερο. Η ποιότητα του μοντέλου χαμηλής γεωμετρίας, βασίζεται στην κρίση του σχεδιαστή και επηρεάζεται από τις προδιαγραφές της συσκευής για την οποία προορίζεται καθώς και από ορισμένα προβλήματα που μπορεί να προκύψουν στις επόμενες διαδικασίες, που θα αναλυθούν στη συνέχεια. Ο λόγος για τον οποίο κατασκευάζονται δύο μοντέλα είναι ο εξής. Το μοντέλο υψηλής γεωμετρίας θα χρησιμοποιηθεί ως μήτρα, της οποίας οι γεωμετρικές λεπτομέρειες θα αποδοθούν στο μοντέλο χαμηλής γεωμετρίας ως υφή. Έτσι το τελικό αντικείμενο θα εμφανίζει πληροφορίες του υψηλής γεωμετρίας, μέσω της χρήσης ειδικών χαρτών.



Εικόνα 7. Παράδειγμα High και Low Poly μοντέλων

Ενδεικτική μεθοδολογία

Για τη δημιουργία των low poly μοντέλων της εργασίας αυτής χρησιμοποιήθηκε η παρακάτω τεχνική τεσσάρων σταδίων. Πρώτα δημιουργούταν ένα αρχικό αντικείμενο σε χαμηλή γεωμετρία. Σε δεύτερο στάδιο δημιουργούταν ένας κλώνος αυτού του αντικειμένου και η γεωμετρία του προσαρμοζόταν κατάλληλα ώστε να υποδιαιρεθεί και να δημιουργηθεί το μοντέλο υψηλής γεωμετρίας. Σε τρίτο στάδιο γινόταν επιστροφή στο αρχικό μοντέλο και εκ νέο δημιουργία κλώνου. Τέλος γινόταν η τελική επεξεργασία της γεωμετρίας ώστε οι δυο κλώνοι να εφάπτονται, η αφαίρεση των περιττών πλευρών, edges, και η τριγωνοποίηση της γεωμετρίας.

LOD (Level Of Detail)

Πρόκειται για μία τεχνική η οποία έχει ως στόχο την βελτιστοποίηση της διαδικασίας του real-time rendering καθώς και την αποδέσμευση υπολογιστικής ισχύος. Κατά την συγκεκριμένη τεχνική χρησιμοποιούνται συνήθως τέσσερις με πέντε διαφορετικές εκδοχές του ίδιου μοντέλου σε διαφορετικά επίπεδα ποιότητας. Ο χρήστης αφού έχει ολοκληρώσει τη μοντελοποίηση, τη διαδικασία του unwrap UVW και συχνά το baking και το texturing, θα προβεί σε κάποιο πρόγραμμα σχεδιασμού και θα χρησιμοποιήσει ένα εργαλείο με το οποίο θα μπορέσει να μειώσει το βαθμό της γεωμετρίας του αντικειμένου περίπου 20-30% ανά επίπεδο. Το επίπεδο με τη μεγαλύτερη γεωμετρία θα το ονομάσει “ONOMA_LOD0”, αυτό της αμέσως μικρότερης “ONOMA_LOD1”, το επόμενο “ONOMA_LOD2” κοκ. Στη συνέχεια θα εξάγει αυτά τα επίπεδα ως ένα αρχείο. Μόλις στείλει το αρχείο σε ένα πρόγραμμα real-time τρισδιάστατης ανάπτυξης, όπως το Unity ή το Unreal Engine, εκείνο αυτόματα θα αναγνωρίσει ότι πρόκειται για αντικείμενο που περιέχει διαφορετικά επίπεδα λεπτομέρειας. Έτσι όταν η κάμερα, θα βλέπει το αντικείμενο από μια κοντινή απόσταση, θα εμφανίζει το αντικείμενο της μεγαλύτερης γεωμετρίας και όσο θα απομακρύνεται θα το αντικαθιστά με όλο και μικρότερης. Είναι συχνό φαινόμενο, στα βιντεοπαιχνίδια, αυτή η μετάβαση γίνεται αισθητή. Ωστόσο, αν η διαδικασία γίνει ορθά τότε θα είναι ανεπαίσθητη. Η βελτίωση που προσφέρει η χρήση των LODs στη ρυθμό ανανέωσης της εικόνας είναι μεγάλη και δεν πρέπει να υποτιμάται η σημασία τους. Αντίστοιχη, διαδικασία μπορεί να χρησιμοποιηθεί και στα texture maps, με τη χρήση χαρτών διαφορετικών επιπέδων ανάλυσης και θα αναλυθεί σε επόμενο κεφάλαιο. Στη συγκεκριμένη εργασία η δημιουργία των LODs έγινε στο Blender Studio με τη χρήση του decimate modifier.

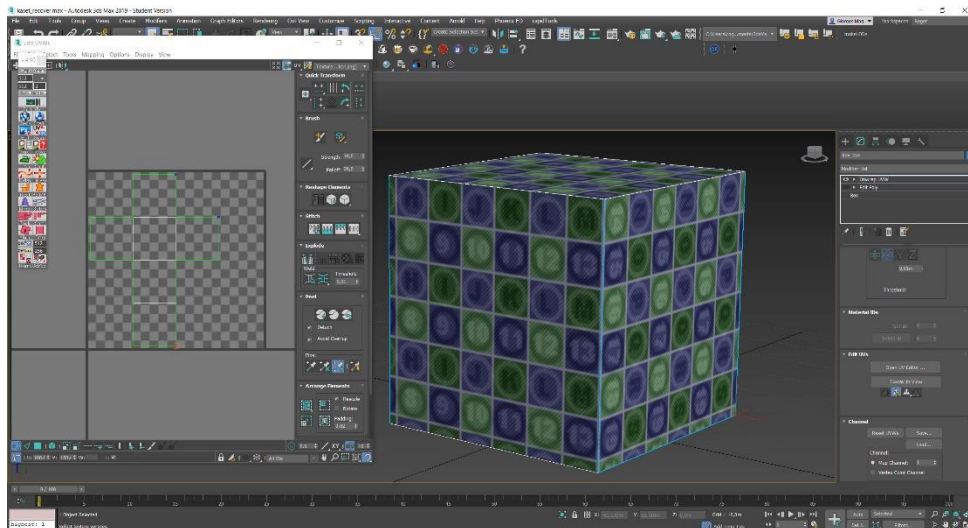


Εικόνα 8. Παράδειγμα μοντέλου με LOD

UV Mapping και Unwrap

Προκειμένου να μπορέσει ο σχεδιαστής να προβάλει μια δισδιάστατη εικόνα πάνω σε ένα τρισδιάστατο μοντέλο πρέπει να μπορεί να προσδιορίσει τη θέση κάθε σημείου της επιφάνειας του, μέσω ενός συστήματος συντεταγμένων. Αυτό επιτυγχάνεται με τη χρήση των συντεταγμένων UV, όπου το U αναπαριστά το άξονα των τετμημένων, οριζόντιο, και το V των τεταγμένων, κατακόρυφο. Συχνά χρησιμοποιείται και ένας τρίτος άξονας, ο W. Οι συντεταγμένες UVW είναι ανάλογες των x,y,z. Σε πρώτο και βασικό στάδιο μπορεί να χρησιμοποιηθεί μια εντολή προβολής, ώστε να προβληθεί μια εικόνα πάνω σε ένα αντικείμενο. Υπάρχουν διαφορά είδη προβολών όπως σφαιρική, κυβική, επίπεδη, κυλινδρική κ.ο.κ. Ωστόσο, συνήθως απευθύνονται σε αντικείμενα ή μέρη αντικειμένων που έχουν αντίστοιχο σχήμα, primitives objects. Σε περιπτώσεις σύνθετων αντικειμένων και πιο συγκεκριμένα, που πρέπει να εφαρμοστεί η τεχνική του baking είναι απαραίτητη η δημιουργία χαρτών UV, UVW Maps. Στην ουσία ο χρήστης καλείται να δημιουργήσει ένα ανάπτυγμα, unwrap, δηλαδή να αποδώσει το τρισδιάστατο αντικείμενο σε δύο διαστάσεις. Ένα ίσως οικείο παράδειγμα ανάλογο της συγκεκριμένης διαδικασίας, είναι η απεικόνιση της υδρόγειου σφαίρας πάνω σε χάρτινους ή ψηφιακούς χάρτες. Με τη χρήση διάφορων προβολών, όπως η γνωμονική ή η Μερκατωρική, ο τρισδιάστατος πλανήτης, απεικονίζεται πάνω σε ένα δισδιάστατο επίπεδο, όπου με τη χρήση συντεταγμένων μπορεί κανείς να προσδιορίσει τη θέση οποιουδήποτε σημείου πάνω στην επιφάνειά του. Η διαδικασία του unwrap προϋποθέτει την εφαρμογή διάφορων κανόνων προκειμένου να μη δημιουργούνται παραμορφώσεις στις υφές. Είναι μια διαδικασία που αρχικά μπορεί να φανεί περίπλοκη στο χρήστη, αλλά με την εξοικείωση πρόκειται για μια διαδικασία ρουτίνας. Υπάρχουν πολλά πλήρως ή ημι-αυτοματοποιημένα εργαλεία καθώς και τελείως αυτόνομα προγράμματα για τη συγκεκριμένη διαδικασία. Ο πιο βασικός κανόνας για έναν ποιοτικό UV χάρτη είναι, οι επιφάνειες να εμφανίζουν όσο το δυνατόν λιγότερη

παραμόρφωση και να διατηρούν τις ίδιες αναλογίες μεγέθους, σε σχέση με αυτές του τρισδιάστατου μοντέλου. Ακόμα πολύ σημαντικό είναι οι πλευρές που σχηματίζουν μεταξύ τους μεγάλες γωνίες να χωρίζονται. Ο έλεγχος για τη σωστή απόδοση των χαρτών γίνεται με ειδικές δοκιμαστικές εικόνες, τις εικόνες checker.



Εικόνα 9. Αριστερά: Unwrap κόβου, Δεξιά: εφαρμογή Checker

Baking

Πρόκειται για μια θεμελιώδη και σημαντική διαδικασία που χρησιμοποιείται κυρίως στα γραφικά που προορίζονται για βιντεοπαιχνίδια ή για animation. Κατά το baking οι πληροφορίες που σχετίζονται με ένα τρισδιάστατο αντικείμενο αποτυπώνονται σε μια δισδιάστατη εικόνα (bitmap).¹³ Υπάρχουν διάφορα είδη baking και θα αναλυθούν στη συνέχεια της εργασίας. Ωστόσο, το πιο απαιτητικό είναι αυτό που αφορά baking πληροφοριών από το μοντέλο υψηλής γεωμετρίας προς το μοντέλο χαμηλής (baking High to Low) .

Το πρόγραμμα που χρησιμοποιήθηκε για τη συγκεκριμένη τεχνική είναι το Substance Painter, version 7.1.1. Οι πληροφορίες, χάρτες που παράγονται κατά το baking στο συγκεκριμένο πρόγραμμα είναι οι περιβαλλοντικές σκιές (ambient occlusion), οι πληροφορίες επιφάνειας (normal), η καμπυλότητα (curvature), η ταυτότητα υλικού (material ID), positioning και thickness. Προκειμένου να αποτυπωθούν σωστά αυτές οι πληροφορίες οι bakers χρησιμοποιούν ειδικά κελιά, cage, και παραμέτρους επιρροής τους. Τα κελιά αυτά ορίζουν από ποιο μέχρι ποιο σημείο θα χρησιμοποιηθεί πληροφορία. Κατά τη διαδικασία, ο αλγόριθμος

¹³ <https://docs.substance3d.com/bake/what-is-baking-172818449.html>

στέλνει ακτίνες που σαρώνουν και αναλύουν το μοντέλο υψηλής γεωμετρίας, High Poly, συλλέγοντας πληροφορίες ώστε να δημιουργήσει τους κατάλληλους χάρτες.. Ορισμένα ακόμα δημοφιλή baking και texturing προγράμματα είναι τα Marmoset, Modo, Armor και Mari.

Βασικοί Χάρτες

Bump Map

Είναι ασπρόμαυροι χάρτες που περιορίζονται σε 8-bit πληροφορία χρώματος. Κατά συνέπεια μπορούν να αποδώσουν 256 τονικές διαβαθμίσεις από το απόλυτο μαύρο έως το απόλυτο λευκό. Χρησιμοποιούνται για να προσομοιάσουν λεπτομέρειες ύψους και βάθους στις επιφάνειες των μοντέλων, χωρίς όμως να επηρεάζουν τη γεωμετρία τους. Το απόλυτο άσπρο δημιουργεί την αίσθηση της εξώθησης, ενώ το μαύρο της ώθησης.

Normal Map

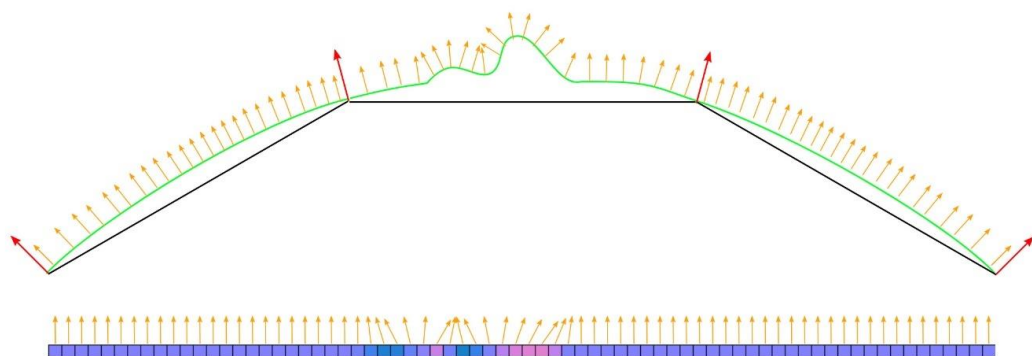
Οι συγκεκριμένοι χάρτες αποτελούν μια πιο εξελιγμένη και ποιοτική εκδοχή των απλών ασπρόμαυρων Bump Maps. Χρησιμοποιούν τρία κανάλια χρώματος κόκκινο, πράσινο, μπλε, RGB, που αντιστοιχούν με τους άξονες X, Y, Z του τρισδιάστατου χώρου. Έτσι κάθε χρώμα επιτρέπει στα προγράμματα να γνωρίζουν πως πρέπει να σκιαγραφηθεί η κάθε λεπτομέρεια ώστε να αποδοθεί μια ρεαλιστική αίσθηση βάθους. Υπάρχουν τρία βασικά είδη Normal χαρτών. Αυτοί που χρησιμοποιούνται περισσότερο καθώς είναι οι πιο «ευέλικτοι», είναι οι tangent space, που χαρακτηρίζονται από μωβ-μπλε αποχρώσεις και χρησιμοποιούνται κυρίως για αντικείμενα που προορίζονται για εφαρμογές animation και πρόκειται να παραμορφωθούν. Οι object space χαρακτηρίζονται από πολλές αποχρώσεις, οι οποίοι συχνά εμφανίζουν καλύτερη ποιότητα και απαιτούν λιγότερη υπολογιστική ισχύ, ωστόσο δεν μπορούν να χρησιμοποιηθούν σε επιφάνειες που πρόκειται να παραμορφωθούν. Οι world space, είναι ακόμα πιο ελαφριοί υπολογιστικά, αλλά μπορούν να χρησιμοποιηθούν σε αντικείμενα που δεν πρόκειται να παραμορφωθούν, αλλά να μετακινηθούν. Είναι σημαντικό να αντιληφθεί κανείς ότι οι Normal, όπως και οι Bump χάρτες δημιουργούν προσεγγίσεις, ψευδαίσθηση γεωμετρίας, χωρίς να την επηρεάζουν. Αυτό σημαίνει, ότι θα υπάρχουν σφάλματα, ιδιαίτερα σε μοντέλα πολύ χαμηλής γεωμετρίας από συγκεκριμένες όψεις.



Εικόνα 10. Αποτέλεσμα χρήσης Normal Map



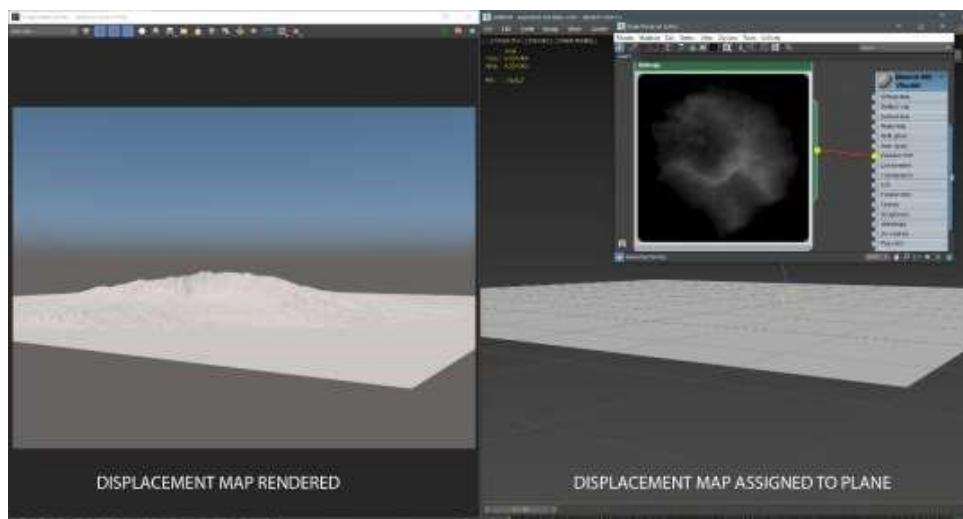
Εικόνα 11. Normal Map



Εικόνα 12. Αναπαράσταση γεωμετρίας ως Normal Map

Displacement Maps

Οι συγκεκριμένοι χάρτες είναι ασπρόμαυροι, αντίστοιχης λογικής με τους Bump, η μεγάλη τους διαφορά όμως είναι ότι χρησιμοποιούνται ώστε να προκαλέσουν πραγματικές εξωθήσεις στην γεωμετρία των αντικειμένων. Ο χρήστης χρησιμοποιώντας ένα plane μεγάλης γεωμετρίας ή ειδικά displacement planes και έναν χάρτη μπορεί να δημιουργήσει μια νέα γεωμετρία η οποία θα επηρεάζεται από αυτόν. Η συγκεκριμένη τεχνική χρησιμοποιείται σε μεγάλο βαθμό για την παραγωγή εδαφών (terrain) και shaders.



Εικόνα 13. Παράδειγμα χρήσης Displacement Map σε γεωμετρία

Ambient Occlusion Maps

Οι χάρτες Ambient Occlusion (AO) χρησιμοποιούνται για να δημιουργήσουν σκιές πάνω στα αντικείμενα αντίστοιχες με αυτές που θα δημιουργούνταν σε μια συνθήκη διάχυτου φωτός. Οι συγκεκριμένοι χάρτες ενισχύουν την αίσθηση ρεαλισμού και είναι επίσης ασπρόμαυροι.

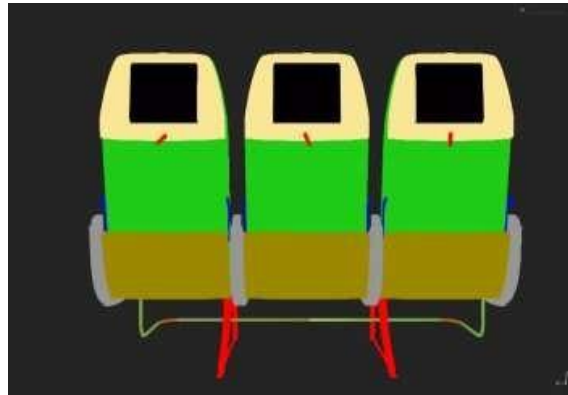


Εικόνα 14. Αποτέλεσμα χρήσης Ambient Occlusion χάρτη

Material ID Maps

Προκειμένου να μπορέσει κανείς να επιλέξει και να επεξεργαστεί μεμονωμένα σημεία του Low Poly μοντέλου, είναι σημαντικό να έχει διαχωρίσει αυτές τις περιοχές στο High Poly πριν γίνει εξαγωγή από το πρόγραμμα σχεδιασμού. Αυτό γίνεται με τη χρήση διαφορετικών υλικών ή αριθμών ταυτότητας στις περιοχές που θέλει ο χρήστης να διαχωρίσει. Κατά το baking παράγεται έναν χάρτη όπου σε κάθε υλικό ή αριθμό, θα δοθεί ένα συγκεκριμένο χρώμα. Έτσι είναι εύκολο να δημιουργηθούν μάσκες με την εντολή color selection, για την μεμονωμένη επεξεργασία επιθυμητών περιοχών. Αν η διαδικασία αυτή παραληφθεί και δεν δημιουργηθούν χάρτες ταυτότητας, ο χρήστης θα πρέπει να ζωγραφίσει χειροκίνητα με τις βούρτσες τις μάσκες. Σε περίπτωση που οι λεπτομέρειες που έχουν αποδοθεί στο Low poly, από το High

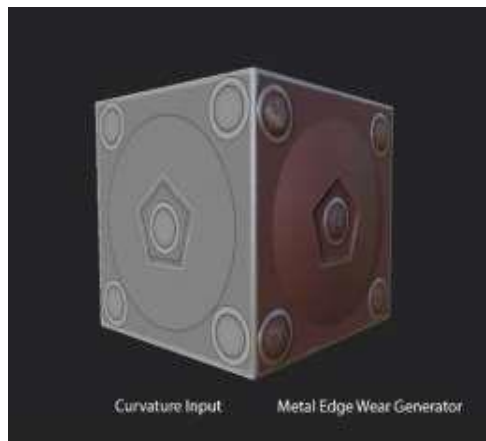
poly, είναι πολλές και συνθέτες, τότε ο διαχωρισμός τους χειροκίνητα θα είναι μια πολύ χρονοβόρα διαδικασία. Επομένως, ειδικά σε σύνθετα μοντέλα που θα αποδοθούν πολλές υφές και υλικά η δημιουργία material ID map είναι απαραίτητη.



Εικόνα 15. Χάρτης Material ID

Curvature Maps

Οι συγκεκριμένοι χάρτες είναι κατά βάση ασπρόμαυροι και σκοπός τους είναι η απεικόνιση των κοίλων και των κυρτών επιφανειών ενός μοντέλου. Οι πληροφορίες κοιλότητας απεικονίζονται με τόνους από το απόλυτο λευκό ως το μέσο γκρι, ενώ της κυρτότητας από το μέσο γκρι έως το απόλυτο μαύρο. Μπορούν να δημιουργηθούν είτε από πληροφορίες γεωμετρίας, είτε από πληροφορίες των normal maps. Οι συγκεκριμένοι χάρτες χρησιμοποιούνται κατά κύριο λόγο για τη δημιουργία масκών για εφαρμογή εφέ φθοράς και σκόνης στις ακμές των μοντέλων.

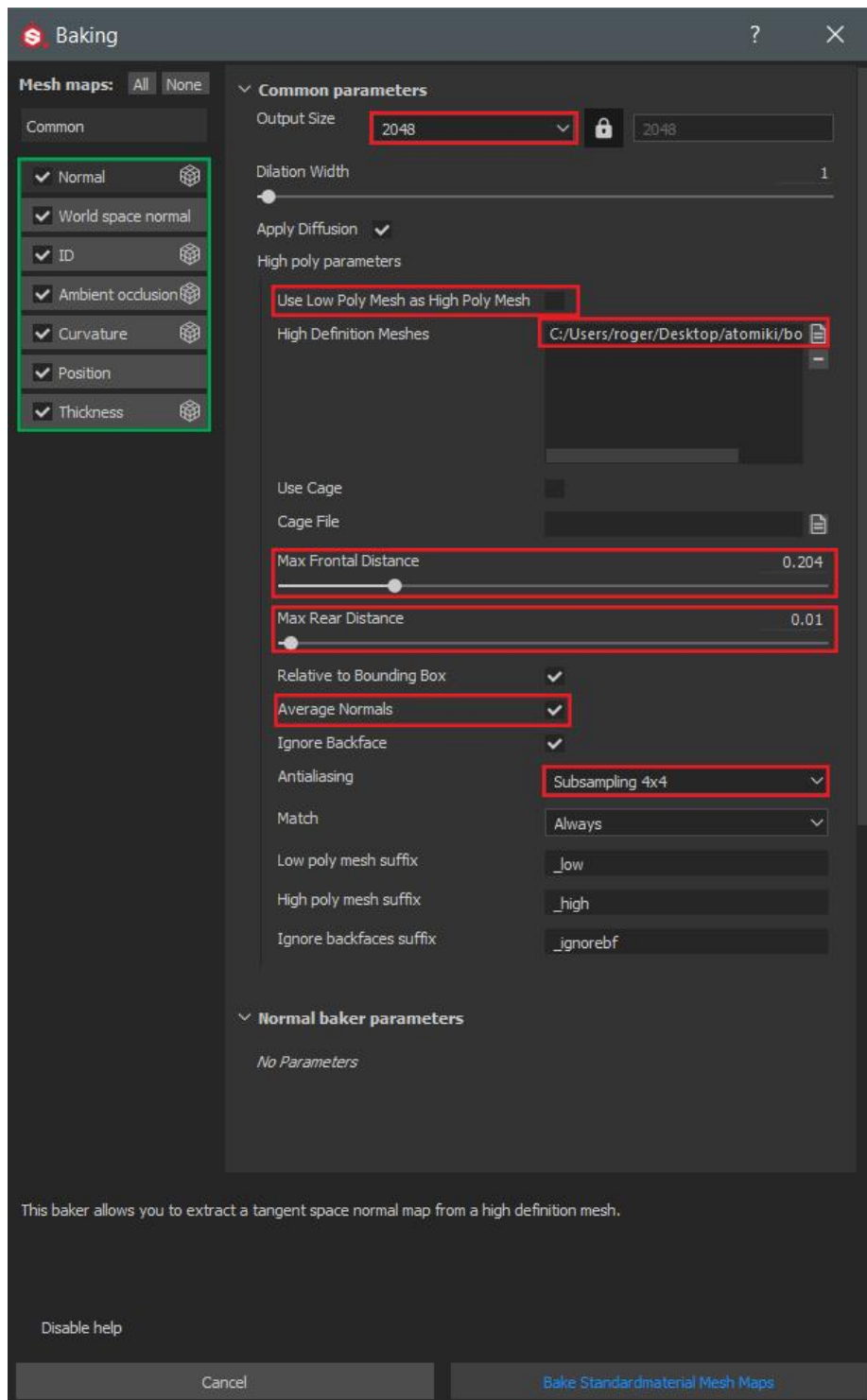


Εικόνα 16: Χρήση χάρτη Curvature για παραγωγή φθοράς

Παράμετροι Baking και Προβληματικές

Κατά τη διαδικασία του baking οι χάρτες που δημιουργούνται είναι προσεγγίσεις. Είναι πιθανό, ειδικά σε μοντέλα πολύ χαμηλής γεωμετρίας, με σκληρές ακμές, να υπάρξουν ορισμένα

προβλήματα, όπως προαναφέρθηκε. Ο χρήστης είναι σημαντικό να γνωρίζει τον τρόπο λειτουργίας των “Bakers”, ώστε να μπορεί να προνοήσει και να αντιμετωπίσει τα προβλήματα αυτά. Ανοίγοντας κανείς το παράθυρο “Baking” σε ένα πρόγραμμα, όπως το Substance Painter, θα παρατηρήσει μια σειρά από επιλογές, όπως φαίνονται στη Εικόνα 17. Παρακάτω θα αναλυθούν οι βασικές ρυθμίσεις που καλείται ο χρήστης να επιλέξει, καθώς και πως αυτές επιδρούν στο τελικό αποτέλεσμα.



Εικόνα 17. Baker, Substance Painter

Output Size

Πρόκειται για την ανάλυση σε pixel, των χαρτών που πρόκειται να παραχθούν από τη διαδικασία του baking. Όσο πιο μεγάλη θα είναι η ανάλυση τόσο καλύτερη θα είναι η ποιότητα των υφών. Ωστόσο, όσο θα μεγαλώνει η ποιότητα τόσο μεγαλύτερη υπολογιστική ισχύς χρειάζεται για να την δημιουργήσει και να την απεικονίσει. Ένα πλεονέκτημα του συγκεκριμένου προγράμματος είναι πως ο χρήστης μπορεί να επιλέξει μια χαμηλή ποιότητα κατά το baking, αλλά στη συνέχεια να τη μεγαλώσει στο view port ή κατά το output.

High poly parameter

Η συγκεκριμένη επιλογή προσφέρει δύο δυνατότητες. Η μια είναι η Use Low Poly Mesh As High Poly Mesh, που επιτρέπει στον αλγόριθμο να χρησιμοποιήσει το ίδιο το Low Poly ως High κατά τη διαδικασία του baking και να αποτυπώσει από αυτό πληροφορίες, Ο λόγος που θα χρησιμοποιήσει κανείς τη συγκεκριμένη επιλογή είναι διότι διαθέτει μόνο ένα μοντέλο και ενδιαφέρεται για τη δημιουργία χαρτών των ambient occlusion, curvature, position ή thickness, αλλά όχι των normal map. Η συγκεκριμένη τεχνική χρησιμοποιείται συνήθως κατά την δημιουργία υφών σε μοντέλα υψηλής γεωμετρίας, τα οποία πιθανότατα θα χρησιμοποιηθούν για offline rendering εφαρμογές. Η δεύτερη επιλογή είναι αυτή του High Definition Mesh. Ο χρήστης εδώ ορίζει την περιοχή που είναι αποθηκευμένο το μοντέλο υψηλής γεωμετρίας από το οποίο θα αποδοθούν πληροφορίες στο μοντέλο χαμηλής γεωμετρίας που έχει εισαχθεί στο πρόγραμμα.

Cage

Πρόκειται για την εισαγωγή ενός προκατασκευασμένου κελιού επιρροής από δευτερεύον πρόγραμμα που συνοδεύει το High Poly μοντέλο.

Max Frontal Distance

Ορίζει την απόσταση επιρροής από την επιφάνεια του χαμηλής γεωμετρίας μοντέλου (low poly) και προς τα έξω, κατά την οποία ο "Baker" θα αντλήσει πληροφορίες του υψηλής γεωμετρίας, High Poly. Αν η απόσταση αυτή είναι πιο μικρή από την απόσταση του High Poly-Low Poly τότε θα υπάρξει έλλειψη πληροφορίας.

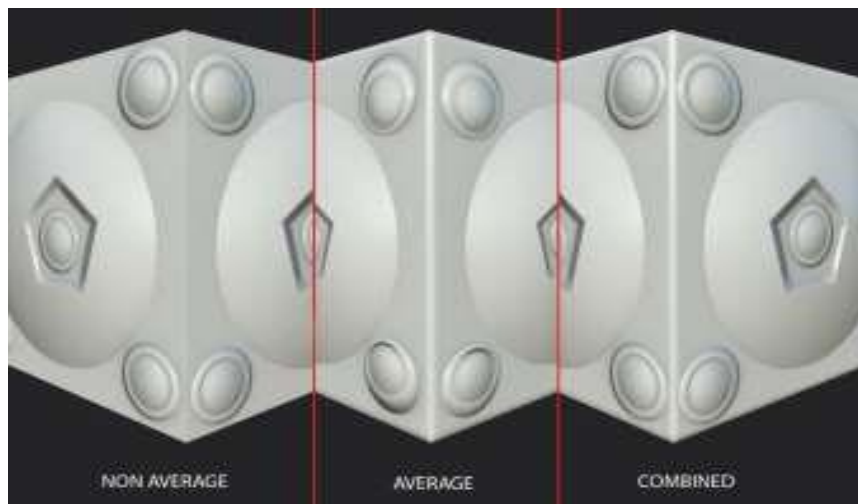
Max Rear Distance

Λειτουργεί ακριβώς όπως η παραπάνω επιλογή, Max Front Distance, με τη διαφορά ότι ορίζει την απόσταση άντλησης πληροφορίας από την επιφάνεια του Low Poly αντικείμενου και ως

προς το εσωτερικό του. Κατά συνέπεια, αν η απόσταση Max Rear Distance είναι μικρότερη από τη Low Poly-High Poly τότε θα υπάρξει απώλεια πληροφορίας.

Average Normals

Πρόκειται για μια επιλογή που συνήθως, ειδικά σε μοντέλα πολύ χαμηλής γεωμετρίας, θα δημιουργήσει προβλήματα είτε την επιλέξει ο χρήστης, είτε όχι. Πιο συγκεκριμένα, όπως φαίνεται στην *Εικόνα 18* με τη χρήση της επιλογής Average Normal γίνεται μια προσπάθεια ακολουθίας της ροής της γεωμετρίας του high poly μοντέλου. Αυτό δημιουργεί ποιοτικούς χάρτες στις ακμές του low poly, ωστόσο δημιουργεί και αλλοιώσεις κατά την αποτύπωση στοιχείων στις περιοχές που βρίσκονται κοντά τους.



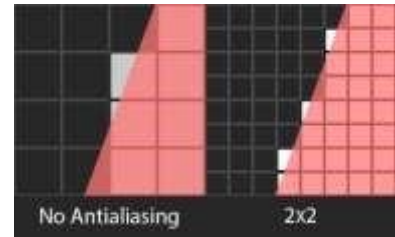
Εικόνα 18. Non Average-Average-Συνδυασμός

Ακριβώς το αντίθετο, συμβαίνει στην περίπτωση που ο χρήστης δεν επιλέξει να κάνει bake με την επιλογή Average Normals. Έτσι, δεν αποδίδεται σωστά η πληροφορία στις ακμές του μοντέλου. Συγκεκριμένα για το Substance Painter το συγκεκριμένο πρόβλημα αντιμετωπίζεται με τρεις τεχνικές. Η πρώτη είναι με τη δημιουργία δύο χαρτών, ενός Average κι ενός Non-Average, η εξαγωγή τους από το πρόγραμμα με διαφορετική ονομασία, για να μην τους επικαλύψει το πρόγραμμα, και η σύνθεσή τους σε κάποιο εξωτερικό πρόγραμμα επεξεργασίας εικόνας, όπως το Photoshop. Με τη χρήση μασκών ο χρήστης κρατάει τα περιγράμματα των Average Normal χαρτών και τις εσωτερικές πληροφορίες των Non Average. Έπειτα εισάγεται στο πρόγραμμα ο διορθωμένος χάρτης. Η δεύτερη και λιγότερο αποτελεσματική είναι με τη χρήση διάφορων script και plug-in που μπορούν να εισαχθούν στο πρόγραμμα. Λόγω των αυτοματοποιημένων διαδικασιών συνήθως δεν επιτυγχάνεται το επιθυμητό αποτέλεσμα σε όλες τις περιοχές του μοντέλου. Τέλος, η τρίτη και ίσως πιο αποτελεσματική μέθοδος γίνεται μέσα στο ίδιο το πρόγραμμα. Ο χρήστης μετονομάζει το texture set στο οποίο βρίσκεται το

αντικείμενο ενδεικτικά σε “Average”. Επιλέγοντας την επιλογή Average Normal κάνει Bake. Έπειτα στο παράθυρο “TEXTURE SET LIST” μετονομάζει το texture set στο οποίο βρίσκεται το αντικείμενο σε “Non Average” και απενεργοποιεί την επιλογή Average Normal στο Baker και κάνει ξανά Bake. Στο παράθυρο “Sub-Self Project” πλέον υπάρχουν και οι δύο χάρτες. Στη συνέχεια δημιουργεί έναν φάκελο στο παράθυρο των layers τον οποίο μπορεί να ονομάσει “Normal Correction” στον οποίο μέσα θα δημιουργήσει δύο νέα fill-layers. Σε κάθε ένα από αυτά θα δώσει τα ανάλογα ονόματα και θα τοποθετήσει τους χάρτες που δημιούργησε στις περιοχές Normal των material, απενεργοποιώντας τα υπόλοιπα κανάλια. Είναι βολικό στην ιεραρχία να είναι υψηλότερα το layer που περιέχει τον χάρτη “Non Average” και να προσαρμοστεί σε αυτό μια μαύρη μάσκα. Έπειτα πρέπει στο παράθυρο “TEXTURE SET SETTINGS” να αλλάξει την επιλογή “Normal mixing” από “Combine” σε “Replace”. Τέλος, ο χρήστης αλλάζει την επιλογή Base Color σε Normal στο παράθυρο των “LAYERS” και σε κάθε ένα από τα layer και τον φάκελο αλλάζει το Blending Mode από Normal Map Detail σε Normal. Πλέον μπορεί με την επεξεργασία της μάσκας να εμφανίσει ή να αποκρύψει τις ατέλειες των δύο χαρτών βλέποντας άμεσα το αποτέλεσμα. Επιπλέον, με τον τρόπο αυτό μπορεί να διορθώσει ατέλειες ζωγραφίζοντας απευθείας πάνω στον χάρτη των Normal δημιουργώντας ένα νέο “Paint layer” εφαρμόζοντας και σε αυτό το τελευταίο στάδιο της διαδικασίας, δηλαδή από “Normal Map Detail” σε “Normal”. Ανάλογη είναι η διαδικασία για τη διόρθωση των χαρτών Ambient Occlusion. Άλλα αντίστοιχα προγράμματα διαθέτουν εργαλεία αποκλειστικά για την αντιμετώπιση αυτού του ζητήματος. Η συγκεκριμένη τεχνική προέκυψε ύστερα από άφθονη έρευνα και χρόνο στη προσπάθεια επίλυσης του συγκεκριμένου ζητήματος και αποτελεί ένα σύνολο μεθοδολογιών και κατανόησης της λειτουργίας του προγράμματος. Είναι ιδιαίτερα σημαντική καθώς επιτρέπει την επεξεργασία των χαρτών μέσα στο ίδιο το πρόγραμμα και σε επίπεδο που επιτρέπει οποιαδήποτε παραμετροποίηση.

Antialiasing

Το aliasing είναι ένα είδος προβλήματος που συμβαίνει κατά τη διαδικασία του baking, στα σημεία που ενώνονται οι ραφές, seams των UVs, ιδιαίτερα όταν η γεωμετρία τους δεν είναι ευθυγραμμισμένη με τους άξονες του πλέγματος. Στην ουσία ο baker στην προσπάθειά του να μετατρέψει την γεωμετρία σε pixels συχνά αντιμετωπίζει θέματα ποιότητας. Έτσι το



Εικόνα 21. Antialiasing

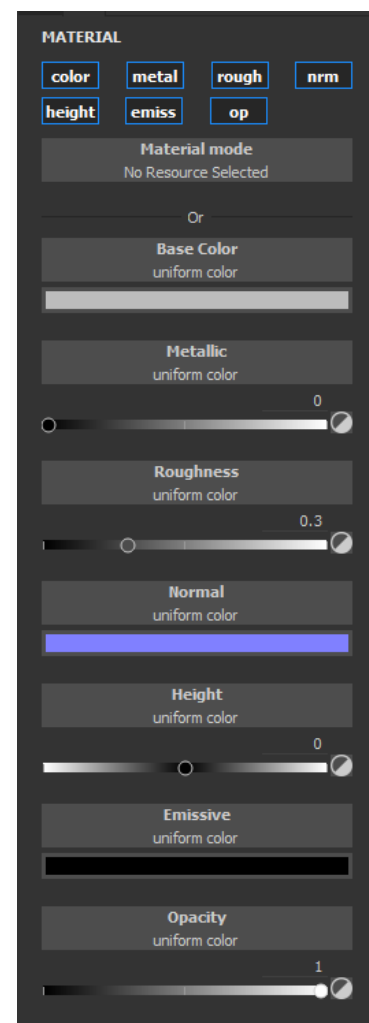
antialiasing φίλτρο, θα πολλαπλασιάσει τον συντελεστή που θα του ορίσει ο χρήστης με την ανάλυση του Output Size. Για παράδειγμα, στην περίπτωση που ο συντελεστής antialiasing είναι 2X2 και το Output Size 2048X2048, κατά τη διαδικασία του baking οι χάρτες θα υπολογιστούν σε ανάλυση 4096X4096 και στην συνέχεια θα μετατραπούν σε 2048X2048. Η συγκεκριμένη διαδικασία μπορεί να καθυστερήσει σε μεγάλο βαθμό το baking, ειδικά όταν χρησιμοποιούνται μεγάλοι συντελεστές. Ωστόσο, στις περισσότερες περιπτώσεις η χρήση της είναι απαραίτητη.

Texturing

Αφού ολοκληρωθεί η διαδικασία του baking ο δημιουργός συνεχίζει με την παραγωγή των υφών, textures. Πρόκειται για μια αρκετά εικαστική και δημιουργική διαδικασία κατά την οποία, θα “ζωγραφίσει” τους χάρτες που θα συντελέσουν το τελικό υλικό του αντικείμενου. Με αυτή τη διαδικασία το αντικείμενο θα αποκτήσει “χαρακτήρα” και “προσωπικότητα”. Τα περισσότερα προγράμματα παραγωγής υφών, όπως το Substance Painter, έχουν αντίστοιχη λογική με το Photoshop. Έτσι ένας εξοικειωμένος χρήστης δεν θα αντιμετωπίσει ιδιαίτερες δυσκολίες. Η βασική μέθοδος εργασίας γίνεται με τη χρήση επιπέδων (layers) και μασκών.

Materials

Δημιουργώντας ένα Fill ή ένα Paint Layer, εμφανίζεται το παράθυρο Material, που διαθέτει διάφορες θυρίδες των οποίων η λειτουργία αναλύεται παρακάτω. Μέσω αυτού ο χρήστης θα μπορέσει να διαμορφώσει το εκάστοτε υλικό. Με τη χρήση



Εικόνα 22. Material. Substance Painter

μασκών μπορεί να απομονώσει περιοχές. Μια καλή μέθοδος οργάνωσης της εργασίας είναι με τη χρήση φακέλων και μιας γενικής μάσκας σε αυτούς για κάθε περιοχή που πρέπει να απομονωθεί. Μέσα στο φάκελο τοποθετούνται τα επιμέρους layers που αφορούν το συγκεκριμένο κομμάτι ή περιοχή του μοντέλου. Με τον τρόπο αυτό η εύρεση στοιχείων και η ταξινόμησή του project ειδικά όταν περιλαμβάνει ένα μοντέλο με πολλά διαφορετικά και σύνθετα υλικά παραμένει εύκολη και εργονομική.

Base Color

Ρυθμίζει το χρώμα του υλικού.

Metallic

Ορίζει τον βαθμό κατά τον οποίο το υλικό θα είναι μεταλλικό. Στο 0, απόλυτο άσπρο δεν θα είναι καθόλου μεταλλικό, ενώ στο 1, απόλυτο μαύρο θα είναι σαν ακατέργαστο μέταλλο. Η αρχική του τιμή είναι το μέσο γκρι.



Εικόνα 23. Metallic Map

Roughness

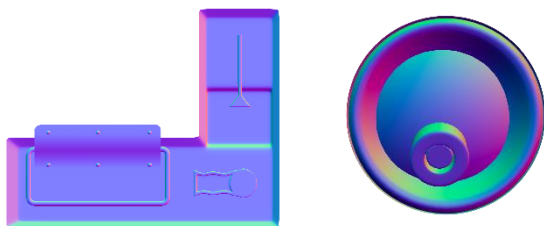
Είναι ένα στοιχείο που αν χρησιμοποιηθεί σωστά μπορεί να αποδώσει μεγάλο ρεαλισμό στο υλικό. Ορίζει στην ουσία το βαθμό της γυαλάδας και της τραχύτητας. Στο 1, απόλυτο μαύρο το αντικείμενο είναι 100% τραχύ, δηλαδή δεν ανακλά καθόλου ακτίνες φωτός, ενώ στο 0 απόλυτο λευκό είναι 100% γυαλιστερό. Χρησιμοποιείται ιδιαίτερα για τη δημιουργία αίσθησης φθοράς, σκόνης και υγρασίας, μέσω масκών και ειδικών χαρτών.



Εικόνα 24. Roughness Map

Normal

Ο χρήστης στο συγκεκριμένο πεδίο μπορεί να επεξεργαστεί πληροφορίες Normal. Στο μενού Hard Surface μπορεί να βρει πολλές έτοιμες λεπτομέρειες τις οποίες με τη χρήση βούρτσας να τις προσθέσει στο αντικείμενό του. Ωστόσο, μπορεί να παράγει και τις δικές του Hard Surface λεπτομέρειες μέσω τρισδιάστατων μοντέλων που έχει δημιουργήσει και έχει κάνει bake πάνω σε ένα τετράγωνο plane. Εξάγοντας τους χάρτες Normal και ID σε ένα πρόγραμμα επεξεργασίας εικόνας όπως το photoshop απομονώνει την πληροφορία με τη χρήση του ID χάρτη ως μάσκα σε ένα κανάλι Alpha.



Εικόνα 25. Λεπτομέρειες Normal παραγμένες από γεωμετρία

Height

Επιτρέπει την επεξεργασία πληροφοριών ύψους. Όπως και στα Normal, ο χρήστης μπορεί να σχεδιάσει πάνω στο αντικείμενο με τη χρήση βουρτσών. Οι πληροφορίες του ύψους μετατρέπονται σε Normal κατά την εξαγωγή.



**Εικόνα 26: Normal-
Height-Normal**

Opacity

Επηρεάζει την διαφάνεια του αντικειμένου. Ένα παράδειγμα χρήσης είναι για τη δημιουργία γυάλινων επιφανειών.

Emission

Χρησιμοποιείται για τη δημιουργία λάμψης.



Εικόνα 27. Εφαρμογή πληροφορίας Emission

Alpha Brush

Τα Alpha είναι εικόνες ασπρόμαυρες που λειτουργούν σαν σφραγίδες για τη δημιουργία σχημάτων. Ο χρήστης μπορεί να σχεδιάσει ένα σχήμα στο Photoshop και να το εισάγει στο Substance Painter προκειμένου να το αποτυπώσει πάνω στο αντικείμενο. Μπορεί να τα χρησιμοποιήσει ως βούρτσες και να επηρεάσει οποιοδήποτε από τα παραπάνω στοιχεία του

υλικού που δέχεται ασπρόμαυρες πληροφορίες. Τα πινέλα Alpha χρησιμοποιούνται σε μεγάλο βαθμό και στην Ψηφιακή Γλυπτική.



Εικόνα 28. Χρήση εικόνας ως Alpha Brush

Generators and Grulinge

Κανένα αντικείμενο στο πραγματικό χώρο δεν είναι απόλυτα καθαρό. Προκειμένου να υπάρξει ρεαλιστική απόδοση υλικών, είναι σημαντικό να προστεθούν πληροφορίες όπως σκόνη, δαχτυλιές, υγρασία, σκουριά, γρατζουνιές και άλλα δείγματα φθοράς και χρήσης, προκειμένου να ενταχθεί το αντικείμενο στο περιβάλλον του. Η διαδικασία που ακολουθείται συνήθως είναι η δημιουργία ενός fill layer και στη συνέχεια μιας μάσκας. Έπειτα χρησιμοποιείται μια από τις εντολές Add generator, Add paint, Add fill που



Εικόνα 29: Generator Metal Wear and Tear

διαμορφώσουν τη μάσκα. Οι generators εκμεταλλεύονται πληροφορίες ύψους, normal, curvature, thickness προκειμένου να δημιουργήσουν ρεαλιστικά αποτελέσματα. Με την επιλογή paint ο χρήστης ζωγραφίζει μόνος του τις πληροφορίες της μάσκας, ενώ με το Fill χρησιμοποιεί έτοιμους χάρτες. Όλες αυτές οι διαδικασίες μπορούν να συνδυαστούν, ώστε το αποτέλεσμα να είναι πιο σύνθετο και λεπτομερές.

Anchor points

Σε περίπτωση που ο σχεδιαστής έχει προσθέσει λεπτομέρειες ύψους ή Normal μετά τη διαδικασία του baking, χειροκίνητα, δεν θα τις συμπεριλάβει ένας Generator. Για να αντιμετωπίσει αυτό το ζήτημα πρέπει να δημιουργήσει ένα anchor point στο layer των πληροφοριών και να το επιλέξει στο μενού του Generator.

Κεφάλαιο 4°:

Πρόγραμμα δημιουργίας βιντεοπαιχνιδιών

Για την υλοποίηση της διαδραστικής περιήγησης της εργασίας χρησιμοποιήθηκε το πρόγραμμα Unity, 2020.23.25f1. Πρόκειται για ένα επαγγελματικό πρόγραμμα παραγωγής βιντεοπαιχνιδιών, εφαρμογών και διαδραστικών περιβαλλόντων, με χρήση τρισδιάστατων και δισδιάστατων γραφικών. Ορισμένα αντίστοιχα προγράμματα ευρέως διαδεδομένα είναι το Unreal Engine και το Codot. Στο συγκεκριμένο κεφάλαιο θα αναλυθούν οι βασικές αρχές και λειτουργίες που χρειάζεται να γνωρίζει κανείς προκειμένου να δημιουργήσει μια τρισδιάστατη διαδραστική σκηνή στο συγκεκριμένο πρόγραμμα. Προτού ξεκινήσει ένα καινούριο project είναι σημαντικό να επιλέξει ο χρήστης την κατάλληλη έκδοση του προγράμματος που επιθυμεί να δουλέψει. Προτείνεται η χρήση κάποιας έκδοσης LTS (Long Term Support), η οποία κατά την εταιρεία θεωρείται η πλέον πιο σταθερή και μπορεί να παρέχει υποστήριξη μέσω της ιστοσελίδας της. Η επιλογή κάποια πειραματικής έκδοσης είναι ιδιαίτερα πιθανό πως θα δημιουργήσει προβλήματα ειδικά όσο η σκηνή γίνεται πιο σύνθετη. Είναι σημαντικό ο χρήστης να αποφύγει την εγκατάσταση κάποιας άλλης έκδοσης, σε προχωρημένο στάδιο της εργασίας του, ακόμα και αν η έκδοση αυτή είναι LTS. Είναι σχεδόν βέβαιο πως θα αντιμετωπίσει ασυμβατότητες μεταξύ των δυο εκδόσεων και η επίλυση τους θα χρειαστεί αρκετό χρόνο και έρευνα που θα τον αποπροσανατολίσουν από τη δημιουργική και παραγωγική διαδικασία.

Οργάνωση και εισαγωγή αντικειμένων

Προκειμένου να υπάρξει μια λειτουργική και καλά οργανωμένη μέθοδος εργασίας είναι σημαντικό να δημιουργηθούν οι κατάλληλοι φάκελοι στον τομέα Asset. Έτσι πριν εισαχθεί στο πρόγραμμα ένα αντικείμενο ο χρήστης πρέπει να έχει δημιουργήσει τους κατάλληλους φακέλους ανάλογα με τη φύση του αντικειμένου. Μπορεί διαχωρίσει τα αντικείμενά του σε φακέλους όπως Materials, Models, Textures, Scenes, ώστε να μπορεί να ανατρέξει σε αυτά εύκολα.

Scene

Προκειμένου να ξεκινήσει η τοποθέτηση αντικειμένων στο χώρο πρέπει να δημιουργηθεί μια σκηνή. Το ίδιο το πρόγραμμα δημιουργεί αυτόματα μια αρχική σκηνή. Ανάλογα τις ανάγκες της εφαρμογής ή του παιχνιδιού ο χρήστης θα επιλέξει πόσες σκηνές τον εξυπηρετούν. Στην παρούσα εργασία υπάρχουν **τρεις** βασικές σκηνές. Η σκηνή του αρχικού μενού, του

σταθμευμένου αεροπλάνου και του αεροπλάνου σε λειτουργία πτήσης. Στο παράδειγμα θα μπορούσε κάποιος να επιλέξει στη ίδια σκηνή που το αεροπλάνο είναι σταθμευμένο, ο χρήστης να το ενεργοποιεί και να το οδηγεί ο ίδιος στον αεροδιάδρομο και να ξεκινά την πτήση. Ωστόσο, η διαδικασία αυτή είναι πολύ πιο σύνθετη προγραμματιστικά και θα επηρέαζε και την ποιότητα της φωτιστικής συνθήκης όπως θα αναλυθεί στη συνέχεια.

Render Pipeline

Το συγκεκριμένο πρόγραμμα παρέχει τρεις βασικές επιλογές προφίλ rendering. Το Build-in Render Pipeline (BRP), το Universal Render Pipeline (URP) και το High Definition Render Pipeline (HDRP). Οι renderer όπως ήδη έχει αναφερθεί εκτελούν διαδικασίες ανάλυσης των στοιχείων της σκηνής ώστε να παράγουν την τελική εικόνα. Ο Build-in Render Pipeline είναι ο απλούστερος στη χρήση με τις πιο λίγες παραμέτρους και τα λιγότερο ρεαλιστικά αποτελέσματα. Ο URP είναι πλήρως παραμετροποιήσιμος και χρησιμοποιείται σε εφαρμογές που απευθύνονται σε ένα μεγάλο εύρος συσκευών από κινητά έως υπολογιστές υψηλού επιπέδου. Ο HDRP είναι επίσης πλήρως παραμετροποιήσιμος και αφορά κυρίως υπολογιστές υψηλού επιπέδου επεξεργαστικής ισχύος. Ο τελευταίος παράγει τα πιο ρεαλιστικά αποτελέσματα γραφικών και απόδοσης του φωτός.

Material

Ο χρήστης μπορεί να εφαρμόσει υλικά στα αντικείμενά του εισάγοντας τα textures, υφές, που έχει δημιουργήσει σε κάποιο πρόγραμμα, όπως το Substance Painter, στις θυρίδες, slots των material. Ωστόσο, σε αντικείμενα που πρόκειται να αποδοθεί ένα απλό και ενιαίο υλικό, αυτό μπορεί να δημιουργηθεί μέσα στο ίδιο πρόγραμμα μέσω των materials ρυθμίζοντας τις παραμέτρους κατάλληλα. Εισάγοντας τα textures στο πρόγραμμα ο χρήστης μπορεί να ορίσει την επιθυμητή τους ανάλυση και συγκεκριμένα για τους Normal χάρτες πρέπει να οριστεί το είδος τους, ειδικά οι πληροφορίες τους δεν θα εμφανίζονται ορθά. Ένας τρόπος εργασίας που θα φανεί αρκετά βοηθητικός σε περίπτωση που πρέπει να γίνουν αλλαγές και διορθώσεις στα textures, είναι η απευθείας εξαγωγή τους στη διεύθυνση του φάκελου όπου τα έχει αποθηκεύσει το Unity. Έτσι κάθε φορά που θα γίνεται εξαγωγή, export, από το πρόγραμμα texturing, τα νέα textures θα αντικαθιστούν τα προηγούμενα και θα φορτώνουν απευθείας χωρίς να χρειάζεται ο χρήστης να ορίσει εκ νέου την ποιότητα και το είδος των χαρτών. Μια επιλογή που προσφέρεται στο παράθυρο των Material και είναι ιδιαίτερα σημαντική όταν υπάρχουν ίδια μοντέλα που μοιράζονται τα ίδια υλικά, είναι η επιλογή Enable GPU Instancing.

Με την ενεργοποίησή της η κάρτα γραφικών κάνει render όλα αυτά τα αντικείμενα την ίδια χρονική στιγμή, αντί να τα κάνει μεμονωμένα. Έτσι αποδεδειγμένα μεγάλο ποσοστό υπολογιστικής ισχύος.

Prefab

Είναι μια λειτουργία προενεργοποιημένη κατά την εισαγωγή ενός αντικείμενου στο πρόγραμμα. Όταν ο χρήστης σύρει αυτό το αντικείμενο στη σκηνή θα παρατηρήσει στην ιεραρχία της σκηνής ότι ο τίτλος του αναγράφεται με μπλε γραμματοσειρά. Με την λειτουργία αυτή ενεργοποιημένη κάθε κλώνος του αντικείμενου που δημιουργείται επηρεάζει ο ένας τον άλλον. Όταν ένα αντικείμενο το οποίο αποτελείται από μεμονωμένα μικροαντικείμενα είναι prefab¹⁴, τότε δεν μπορεί να επεξεργαστεί ή να διαγραφεί κάποιο από μεμονωμένα αντικείμενα που περιέχει. Για να τα επεξεργαστεί λοιπόν πρέπει να επιλέξει “Prefab” και στη συνέχεια “Unpack”. Έτσι όμως χάνεται η λειτουργία επιρροής. Ο χρήστης μπορεί να δημιουργήσει νέα σύνολα αντικειμένων τα οποία θα ορίσει ως Prefabs. Η συγκεκριμένη τεχνική είναι ιδιαίτερα χρήσιμη όταν υπάρχουν πολλοί κλώνοι σε μια σκηνή και θέλουμε να τους επηρεάσουμε χωρίς να χρειάζεται να τους επιλέξουμε όλους μαζί, όπως τα δέντρα σε μια σκηνή δάσους.

LOD και Texture Resolution

Όπως ήδη έχει αναφερθεί με τη συγκεκριμένη τεχνική εξοικονομείται μεγάλη επεξεργαστική ισχύς στις εφαρμογές. Ωστόσο, υπάρχουν ορισμένες ιδιαιτερότητες κατά τη χρήση των LODs που θα αναλυθούν στη συνέχεια. Όταν εισάγεται ένα μοντέλο που περιέχει LOD με την κατάλληλη κατάληξη ονομασίας το πρόγραμμα το αναγνωρίζει αυτόματα και το μετατρέπει σε Prefab. Ο χρήστης μπορεί να ορίσει την απόσταση μεταξύ κάμερας και αντικείμενου κατά την οποία θα αντικαθίσταται κάθε επίπεδο (Εικόνα 30). Ο χρήστης μπορεί να δημιουργήσει επίπεδα ανάλυσης και στις υφές των μοντέλων. Θα πρέπει να εισάγει εκ νέου τα textures ή να δημιουργήσει αντίγραφα στον φάκελο αποθήκευσης τόσες φορές όσα και τα επίπεδα που θέλει να δημιουργήσει και να ορίσει διαφορετικές αναλύσεις κατά την εισαγωγή. Στη συνέχεια θα φτιάξει άλλα τόσα νέα material στα οποία θα προσαρμόσει διαφορετικές αναλύσεις. Τέλος, θα εφαρμόσει το ανάλογο επίπεδο ανάλυσης υφών στο ανάλογο επίπεδο γεωμετρίας.

LOD 0 100%	LOD 1 13%	LOD 2 9%	LOD 3 6%	LOD 4 3%	Culled 1%
---------------	--------------	-------------	-------------	-------------	--------------

Εικόνα 30. LOD σύστημα, Unity

¹⁴ Prefab (Prefabrication)=Προκατασκευασμένος

Camera

Για τη παραγωγή εικόνας ο χρήστης καλείται να χρησιμοποιήσει ένα συστήματα καμερών. Έχει πρόσβαση σε ρυθμίσεις όπως η ταχύτητα κλείστρου, το διάφραγμα, η εστιακή απόσταση, το iso, η clipping plane, το anti-aliasing φίλτρο κ.α.

Environment

Όπως σε κάθε άλλο πρόγραμμα έτσι και στο Unity είναι δυνατή η δημιουργία ουρανού και ατμόσφαιρας. Το συγκεκριμένο πρόγραμμα προσφέρει ένα μεγάλο φάσμα επιλογών πλήρως παραμετροποιήσιμων. Το πιο βασικό που εισάγει αυτόματα το πρόγραμμα σε κάθε νέα σκηνή είναι το σύστημα Gradient Sky. Η δημιουργία του ουρανού γίνεται με τη χρήση τριών βασικών χρωμάτων. Έπειτα με τη χρήση του HDRP Sky είναι δυνατή η χρήση εικόνων HDRI, μεγάλου δυναμικού εύρους 360 μοιρών. Η απόδοση τόσο του φωτός όσο και της εικόνας που παράγεται είναι σε πολύ μεγάλο βαθμό ρεαλιστική. Η ίδια η εικόνα αναλαμβάνει το φωτισμό της σκηνής. Έτσι αν γίνει επιλογή ενός ηλιοβασιλέματος στα αντικείμενα θα αποδοθεί η ανάλογη φωτιστική χροιά. Με τη χρήση του Physical Based Sky γίνεται η προσπάθεια απόδοσης πλανητικής ατμόσφαιρας δύο επιπέδων. Ανάλογα το υψόμετρο αλλάζει η σύσταση της ατμόσφαιρας με αποτέλεσμα το φωτιστικό αποτέλεσμα να αγγίζει επίσης μεγάλα επίπεδα ρεαλισμού. Επιπλέον μπορεί να οριστούν παράμετροι όπως το μέγεθος του ηλίου, η διάθλαση του φωτός, η εφαρμογή ομίχλης, νεφών κ.α.

Lighting

Ένα από τα απαραίτητα εφόδια που πρέπει να διαθέτει ένας καλλιτέχνης που ασχολείται με την εικόνα είναι η διαχείριση και η κατανόηση του φωτός. Όπως και σε ρεαλιστικές συνθήκες έτσι και σε οποιοδήποτε πρόγραμμα τρισδιάστατων εφαρμογών ένας από τους πιο σημαντικούς παράγοντες που θα αποδώσει την αίσθηση ρεαλισμού είναι η φωτιστική συνθήκη. Ακόμα και σε μοντέλα Stylized,¹⁵ η σωστή χρήση του φωτός είναι αυτή που εμψυχώνει τα αντικείμενα και τους δίνει μια σύσταση και σημασία στο χώρο. Η κατανόηση της χρήσης των φωτιστικών πηγών για τη δημιουργία συνθηκών φωτός και σκιάς, ανοίγουν το δρόμο για μια από τις πιο δημιουργικές διαδικασίες της τρισδιάστατης τέχνης. Ακόμα κι ένας κινηματογραφιστής ή ένας φωτογράφος που πιθανότατα δεν έχουν σκοπό να ασχοληθούν με την τρισδιάστατη τέχνη, έχοντας μονάχα τις βασικές γνώσεις διαχείρισης ενός περιβάλλοντος 3D, θα μπορούσαν να

¹⁵ μοντέλα εξαιρετικά χαμηλής γεωμετρίας με υφές που παραπέμπουν σε δισδιάστατα γραφικά

χρησιμοποιήσουν τα προγράμματα αυτά ως στούντιο πειραματισμού για να διευρύνουν τις ιδέες τους, τη φωτιστική τους αντίληψη και την δημιουργικότητά τους.

Στις real-time εφαρμογές τίθεται ένα σημαντικό ζήτημα που αφορά τον φωτισμό και την υπολογιστική ισχύ. Όπως στα μοντέλα υψηλής και χαμηλής γεωμετρίας υπάρχουν περιορισμοί, αντίστοιχα υπάρχουν και στα φώτα με τα Real Time και τα Baked. Τα Real Time είναι φώτα που η επίδρασή τους υπολογίζεται σε πραγματικό χρόνο, ανά καρέ. Λόγω αυτού καταναλώνουν μεγάλα ποσοστά επεξεργαστικής ισχύος, ωστόσο η χρήση τους είναι απαραίτητη όταν υπάρχουν αντικείμενα ή χαρακτήρες που κινούνται και παράγουν φως ή σκιές στο χώρο. Αντίθετα τα Baked είναι φώτα των οποίων η επίδραση έχει προϋπολογιστεί και αποθηκευτεί σε ειδικούς χάρτες, εικόνες, τους Lightmaps και κατά την εκτέλεση της τελικής εφαρμογής δεν γίνεται κανένας υπολογισμός για αυτά. Κατά συνέπεια, ένα Baked φως του οποίου η πληροφορία είναι αποθηκευμένη στον υπολογιστή καταναλώνει ελάχιστη ισχύ. Ωστόσο δεν μπορεί να χρησιμοποιηθεί για κινούμενα αντικείμενα παρά μόνο σε σταθερά. Ένα εργαλείο που χρησιμοποιείται για να καλύψει ένα μέρος αυτού του προβλήματος είναι τα Light Probe. Πρόκειται για ένα εργαλείο το οποίο έχει συλλέξει δείγματα φωτός από την περιοχή που ο χρήστης το έχει τοποθετήσει και όταν ένα αντικείμενο περάσει από κοντά του, τότε εκείνο του αποδίδει τις ανάλογες πληροφορίες χρώματος και τονικότητας. Όσο περισσότερα Light Probe τοποθετηθούν τόσο καλύτερη θα είναι η μετάβαση και ποιότητα του φωτός στο κινούμενο αντικείμενο. Ωστόσο το θέμα τις έλλειψης δημιουργίας σκιάς από το κινούμενο αντικείμενο προς τον περιβάλλοντα χώρο παραμένει. Επιπλέον χρήση των Light Probe πρέπει γίνεται στις σκηνές που περιέχουν LOD μοντέλα και χρησιμοποιούνται φώτα Baked. Μια λύση για μια υψηλή ποιότητα φωτός και αντιμετώπιση των παραπάνω ζητημάτων είναι η επιλογή Mixed Light, δηλαδή χρήση Real Time αλλά και Baked φώτων. Τέλος ένα σημαντικό πλεονέκτημα που ενισχύει τον ρεαλισμό είναι η χρήση των Volumetric Light, που εφαρμόζεται μόνο σε Real Time φώτα. Πρόκειται για τη δημιουργία ακτίδων φωτός με τη χρήση ατμοσφαιρικής σκόνης. Ένα παράδειγμα φωτιστικής συνθήκης που μιμούνται τα συγκεκριμένα φώτα είναι η δέσμες φωτός που δημιουργούνται μεταξύ χαραμιάδων ή αιχμών.

Static Object

Ως στατικό αντικείμενο ορίζεται κάθε αντικείμενο το οποίο δεν πρόκειται να μετακινηθεί κατά την διάρκεια της εφαρμογής ή να εφαρμοστεί σε αυτό animation. Τα αντικείμενα αυτά συμπεριλαμβάνονται στις διαδικασίες του Light Baking.

Baked Mixed layer mask

Προκειμένου να απομονωθεί ή να συμπεριληφθεί η επίδραση μιας φωτιστικής πηγής πάνω σε ένα αντικείμενο ο χρήστης μπορεί να χρησιμοποιήσει ειδικές μάσκες. Στα φώτα το μενού των μασκών είναι κρυφό και μπορεί να ενεργοποιηθεί έχοντας επιλεγμένο το επιθυμητό φως και πατώντας το κουμπί με το γρανάζι (more options) στις γενικές, General, ρυθμίσεις του inspector. Αφού εμφανιστεί ο τομέας Light Layer ο χρήστης μπορεί να ορίσει ποια θα είναι τα επίπεδα τα οποία θα φωτίζει το συγκεκριμένο φως. Ενώ στο επιλεγμένο αντικείμενο η αντίστοιχη επιλογή βρίσκεται φανερά στις επιλογές, additional setting, του inspector.

Reflection Probe

Τοποθετώντας ένα αντικείμενο στο περιβάλλον της Unity αυτόματα λαμβάνει αντανakλάσεις μόνο από το περιβάλλον, ουρανό. Ακόμα και αν βρίσκεται σε εσωτερικό χώρο ή κοντά σε άλλα αντικείμενα, πάνω στην επιφάνειά του δεν θα αποδίδονται ανακλάσεις των άλλων αντικειμένων πέρα του ουρανού που έχει οριστεί. Τα reflection probe χρησιμοποιούνται για να ορίσουν το εύρος της περιοχής επίδρασης ανακλάσεων μεταξύ αντικειμένων. Ωστόσο και εδώ θα συναντήσει κανείς το θέμα του Real Time και του Bake. Αν είναι real time αποδίδονται οι αντανakλάσεις όλων των αντικειμένων είτε στατικών, είτε ενεργών με κόστος την επεξεργαστική ισχύ. Στην αντίθετη περίπτωση περιλαμβάνονται μόνο τα στατικά αλλά η πληροφορία αποθηκεύεται σε εικόνες, χάρτες οπότε απαιτείται λιγότερη επεξεργαστική ισχύς.

Decal Projector

Πρόκειται για ένα εργαλείο που έχει εφαρμοστεί τα τελευταία χρόνια στο πρόγραμμα Unity και επιτρέπει την προβολή υλικών και υφών πάνω σε επιφάνειες χωρίς να χρειάζεται να έχει προηγηθεί unwrap.

Post Process

Πρόκειται για ένα σύνολο εφέ που εφαρμόζονται συνήθως σε τελικό στάδιο της διαδικασίας στην κάμερα ώστε να δώσουν μια πιο ζωντανή και μοναδική χροιά στην εικόνα. Υπάρχουν διάφορα εφέ όπως color correction, bloom, vignette, contrast, depth in view, motion blur, antialiasing, fog, chromatic aberration, white balance. Επιπλέον ο καλλιτέχνης μπορεί να αποδώσει διαφορετικά εφέ σε διαφορετικές κάμερες ή σημεία του χώρου με τη χρήση script.

Scripts

Είναι κομμάτια κώδικα τα οποία συντάσσει ή βρίσκει έτοιμα ο χρήστης και μπορούν να χρησιμοποιηθούν στο πρόγραμμα για τη δημιουργία διάδρασης, εργαλείων, εντολών κίνησης αντικειμένων, εντολών εισόδου ή οποιαδήποτε άλλη διαδικασία θέλει ο χρήστης. Το Unity χρησιμοποιεί γλώσσα C#.

Terrain

Για τη δημιουργία του τοπίου χρησιμοποιήθηκε το εργαλείο terrain. Πρόκειται για ένα εργαλείο του προγράμματος το οποίο επιτρέπει τη δημιουργία εξωθήσεων και αυλακώσεων εδάφους, την εφαρμογή γρασιδιού, λουλουδιών, δέντρων, υφών στο έδαφος και την αίσθηση ανέμου στη σκηνή. Είναι εύκολο στη χρήση και σε μεγάλο βαθμό παραμετροποιήσιμο.

Κεφάλαιο 4^ο :

Game Play

Το πρακτικό κομμάτι της εργασίας αποτελείται από ένα εκτελέσιμο αρχείο με το οποίο ο χρήστης μπορεί να περιηγηθεί στον εξωτερικό και εσωτερικό χώρο ενός αεροπλάνου το οποίο βρίσκεται σταθμευμένο σε ένα αεροδρόμιο. Εκτελώντας το πρόγραμμα κατευθύνεται στη σελίδα του αρχικού μενού. Εκεί δίνονται οι οδηγίες περιήγησης και διάδρασης με τα αντικείμενα του χώρου, καθώς και οι οδηγίες πτήσης. Με τα πλήκτρα w,s ο χρήστης μπορεί να κινηθεί μπρος και πίσω, ενώ με τα a,d μπορεί να κατευθυνθεί αριστερά και δεξιά ανάλογα. Πατώντας το πλήκτρο space μπορεί να κάνει άλματα. Πατώντας το εικονίδιο “Play”, ο χρήστης μεταφέρεται στη βασική σκηνή εξωτερικά του αεροπλάνου, από όπου μπορεί να ξεκινήσει την περιήγηση. Ενώ πατώντας το εικονίδιο “Quit” πραγματοποιεί έξοδο από το πρόγραμμα. Η συγκεκριμένη εφαρμογή είναι πρώτου προσώπου, fps, δηλαδή η κάμερα είναι τοποθετημένη με τρόπο ώστε να προσομοιώνει τα μάτια του παίχτη. Η κίνηση της κάμερας επιτυγχάνεται με τη χρήση του ποντικιού. Μετακινώντας το κινείται και η κάμερα προς ανάλογη κατεύθυνση καλύπτοντας ένα πεδίο 360 μοιρών. Στο κέντρο της εικόνας υπάρχει μόνιμα μία λευκή κουκίδα, στόχος. Όταν αυτή περάσει πάνω από ένα αντικείμενο με το οποίο ο χρήστης μπορεί να διαδράσει μετατρέπεται σε κόκκινη. Πατώντας αριστερό κλικ δίνεται η εντολή διάδρασης. Με αυτόν το τρόπο ο παίχτης μπορεί να ανοιγοκλείσει την πόρτα του πιλοτηρίου και των τουαλετών ώστε να εισέλθει στους συγκεκριμένους χώρους. Η μόνη πόρτα που είναι κλειδωμένη είναι αυτή της μπροστινής αριστερής τουαλέτας, στην οποία ο χρήστης δεν μπορεί να εισέλθει. Υπάρχει ανάλογη κόκκινη ένδειξη στα φώτα διαθεσιμότητας και την κλειδαριά της. Πατώντας το πλήκτρο “Escape” επιστρέφει στο αρχικό menu. Όταν χρήστης μπει στο πιλοτήριο εμφανίζεται ένα μήνυμα το οποίο τον παραπέμπει να πατήσει το πλήκτρο “F” ώστε να μεταβεί στη σκηνή πτήσης. Με τα πλήκτρα w,s μπορεί να κατευθύνει το αεροπλάνο προς τα πάνω ή προς τα κάτω ενώ με τα πλήκτρα a,s δεξιά και αριστερά. Η αυξομείωση της ταχύτητας γίνεται με το αριστερό και το δεξί πλήκτρο του ποντικιού. Τέλος, με το αριστερό πλήκτρο “Alt” γίνεται χρήση των αερόφρενων. Σε περίπτωση που το αεροπλάνο πέσει στη θάλασσα ή ο χρήστης πατήσει το πλήκτρο “Escape” γίνεται μετάβαση στο αρχικό Menu.

Συμπεράσματα

Η πληθώρα των τεχνικών και των σύνθετων μεθοδολογιών μοντελοποίησης μπορεί να μπερδέψουν και να αποτρέψουν έναν νέο χρήστη να ασχοληθεί σε βάθος με τον συγκεκριμένο τομέα. Ωστόσο, η κατανόησή τους και η τριβή με το αντικείμενο, θα βοηθήσουν σε έναν

οργανωμένο και ώριμο καταμερισμό των εργασιών, που θα βελτιστοποιήσει τον χρόνο παραγωγής και θα δώσει περισσότερο χώρο στην ανάπτυξη της δημιουργικότητας και της καλλιτεχνικής έκφρασης. Είναι σημαντικό, σ' αυτόν τον σύνθετο και αυστηρά τεχνικό τομέα, ο σχεδιαστής-καλλιτέχνης να πειραματιστεί με τις διαφορετικές τεχνικές, εργαλεία και προγράμματα, με στόχο να διαμορφώσει τη δική του μεθοδολογία, τεχνική και να καταφέρει να αποδώσει την προσωπική του αισθητική χροιά στα έργα του. Συχνά, η τρισδιάστατη τέχνη συγχέεται εσφαλμένα με την αντίληψη ότι παράγει μια τυπική, διακριτή και ενιαία αισθητική. Η αντίληψη αυτή έχει δημιουργηθεί λόγω της αφομοίωσης της από τη βιομηχανία του θεάματος. Οι ευρύτερες απαιτήσεις της αγοράς έφεραν στο προσκήνιο την ανάγκη για τη δημιουργία ενός μέσου που θα μπορούσε να αναπαραστήσει περιβάλλοντα, αντικείμενα και συνθήκες με ρεαλιστικούς όρους, ώστε να μειωθεί το κόστος παραγωγής. Το ρόλο αυτό έχει υιοθετήσει ο 3D σχεδιασμός. Ωστόσο, οι εφαρμογές του μέσου είναι ανεξάντλητες και η εξέλιξη της τεχνολογίας έχει επιτρέψει στον καλλιτέχνη, με τη χρήση ακόμη κι ενός συμβατικού υπολογιστή, να οπτικοποιήσει και να αποδώσει, σχεδόν οτιδήποτε μπορεί να φανταστεί. Όπως στις εικαστικές τέχνες, δυο καλλιτέχνες που τους τίθεται να αποδώσουν ένα κοινό ζητούμενο, θα καταλήξουν σε δύο διαφορετικές και μοναδικές δημιουργίες, έτσι και στην πιο αυστηρά ρεαλιστική τρισδιάστατη αποτύπωση ενός αντικειμένου, θα προηγηθεί μια τελείως διαφορετική έρευνα, πρακτική και προσέγγιση υλοποίησης, μεταξύ δύο σχεδιαστών. Ανακεφαλαιώνοντας, η παρούσα εργασία έχει ως στόχο την παρουσίαση των διάφορων τεχνικών και μεθοδολογιών, ώστε να διευκολύνουν κάποιον που θα ήθελε να ασχοληθεί με τον συγκεκριμένο τομέα να εκφράσει πιο εύκολα τις καλλιτεχνικές του ιδέες σε βάθος χρόνου. Τέλος, τον προτρέπει να αντιμετωπίσει την τρισδιάστατη μοντελοποίηση σαν μέσο έκφρασης και όχι σαν απλό εργαλείο αναπαραγωγής θεάματος.

Βιβλιογραφία

- 1.Chopine, Ami (2011). 3D Art Essentials, The Fundamentals of 3D Modeling, Texturing, and Animation, Taylor & Francis.
- 2.H. Eberly, David (2006). 3D Game Engine Design, Second Edition: A Practical Approach to Real-Time Computer Graphics, Sun Francisco: Morgan Kaufmann.
- 3.Ahead, Luke (November 2017). 3D Game Environments, Create Professional 3D Game Worlds, Second Edition, New York: A K Peters/CRC Press
- 4.Scherer, Manuel (March 2011). ZBrush 4 Sculpting for Games, Beginner's Guide, Birningham: Packt
- 5.Mooney, Thomas (October 2012). 3ds Max Speed Modeling for 3D Artists, Flex your speed modeling muscles using 3ds Max, Birningham: Packt
- 6.Söderlund, Anders (June 2015). Procedural Modeling of Rocks, Uppsala Universitet, https://academic.oup.com/dsh/article/30/suppl_1/i148/361687?ijkey=PyyjAyKj0Z3HxVv&keytype=ref
- 7.Saldaña, Marie (December 2015). An Integrated Approach to the Procedural Modeling of Ancient Cities and Buildings. Digital Scholarship in the Humanities, Volume 30, Issue suppl_1

Ιστογραφία

<https://artisticrender.com/10-different-types-of-3d-modeling-techniques/>
<https://www.chaosgroup.com/blog/real-time-ray-traced-and-rasterized-rendering-explained>
<https://knowledge.autodesk.com/support/alias-products/getting-started/caas/CloudHelp/cloudhelp/2019/ENU/Alias-Tutorials/files/GUID-366304CB-16FF-46F9-9F64-D7385358D855-htm.html> <https://www.magellan-solutions.com/blog/3d-modeling-techniques/> <https://docs.substance3d.com/bake/what-is-baking-172818449.html>
https://docs.knaldtech.com/doku.php?id=thhttps://docs.knaldtech.com/doku.php?id=the_bake_r_knald_1.1e_baker_knald_1.1 <https://help.poliigon.com/en/articles/1712652-what-are-the-different-texture-maps-for>
https://docs.blender.org/manual/en/latest/sculpt_paint/sculpting/adaptive.html
https://docs.blender.org/manual/en/latest/render/shader_nodes/vector/vector_displacement.html <https://developer.nvidia.com/gpugems/gpugems3/part-i-geometry/chapter-1->

[generatingcomplex-procedural-terrains-using-gpu](https://cgicoffee.com/blog/2018/01/unity-real-time-rendering-vs-offline-cgi) <https://unity.com/ray-tracing>
<https://cgicoffee.com/blog/2018/01/unity-real-time-rendering-vs-offline-cgi>
<https://oneirosvr.com/real-time-rendering-vs-offline-rendering-in-architecture/>
<https://www.chaosgroup.com/blog/real-time-ray-traced-and-rasterized-rendering-explained>
<https://conceptartempire.com/uv-mapping-unwrapping/>
<https://www.chaosgroup.com/blog/understanding-metalness>
<https://academy.substance3d.com/courses/the-pbr-guide-part-2>
<https://docs.unity3d.com/Manual/Prefabs.html>
<https://docs.unity3d.com/Manual/optimizing-draw-calls.html>
<https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@7.1/manual/Override-Physically-Based-Sky.html>

Παράρτημα

EspaceMenuActivator:

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

```
public class EspaceMenuActivator : MonoBehaviour
{
```

```
    void Update()
```

```
    {
```

```
        if (Input.GetKeyDown(KeyCode.Escape))
```

```
        {
```

```
            SceneManager.LoadScene(0);
```

```
        }
```

```
    }
```

```
    void Start()
```

```
    {
```

```

        Cursor.visible = true;
        Cursor.lockState = CursorLockMode.None;
    }
}

```

LevelLoader:

```

.using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class LevelLoader : MonoBehaviour
{
    public GameObject loadingScreen;
    public Slider slider;
    public Text progressText;

    public void LoadLevel (int sceneIndex)
    {
        StartCoroutine(LoadAsynchronously(sceneIndex));

    }

    IEnumerator LoadAsynchronously (int sceneIndex)
    {
        AsyncOperation operation = SceneManager.LoadSceneAsync(sceneIndex);

        loadingScreen.SetActive(true);
    }
}

```

```

while (!operation.isDone)
{
    float progress = Mathf.Clamp01(operation.progress / .9f);

    slider.value = progress;

    progressText.text = progress * 100f + "%";
    yield return null;

}

}

}

MainMenu:

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class MainMenu : MonoBehaviour
{
    public void PlayGame()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

    }
    public void QuitGame()
    {
        Debug.Log("Quit");
        Application.Quit();
    }
}

```

```
}
```

Mouse refresh:

```
using System.Collections;
```

```
using System.Collections.Generic;
```

```
using UnityEngine;
```

```
public class mouse : MonoBehaviour  
{
```

```
    // Use this for initialization
```

```
    void Start ()
```

```
{
```

```
    Cursor.visible = true;
```

```
    Cursor.lockState = CursorLockMode.None;
```

```
}
```

```
    // Update is called once per frame
```

```
    void Update ()
```

```
{
```

```
}
```

```
}
```

Raycast:

```
using System.Collections;
```

```
using System.Collections.Generic;
```

```
using UnityEngine;
```

```
using UnityEngine.UI;
```

```

public class DoorRaycast : MonoBehaviour
{
    [SerializeField] private int rayLength = 5;
    [SerializeField] private LayerMask layerMaskInteract;
    [SerializeField] private string excludeLayerName = null;

    private MyDoorController raycastedObj;

    [SerializeField] private KeyCode openDoorKey = KeyCode.Mouse0;

    [SerializeField] private Image crosshair = null;
    private bool isCrosshairActive;
    private bool doOnce;

    private const string interactableTag = "InteractiveObject";

    private void Update()
    {
        RaycastHit hit;
        Vector3 fwd = transform.TransformDirection(Vector3.forward);

        int mask = 1 << LayerMask.NameToLayer(excludeLayerName) |
layerMaskInteract.value;

        if (Physics.Raycast(transform.position, fwd, out hit, rayLength, mask))
        {
            if (hit.collider.CompareTag(interactableTag))
            {
                if (!doOnce)
                {
                    raycastedObj =
hit.collider.gameObject.GetComponent<MyDoorController>();
                    CrosshairChange(true);
                }
            }
        }
    }
}

```

```

        }

        isCrosshairActive = true;
        doOnce = true;

        if (Input.GetKeyDown(openDoorKey))
        {
            raycastedObj.PlayAnimation();
        }
    }
}

else
{
    if (isCrosshairActive)
    {
        CrosshairChange(false);
        doOnce = false;
    }
}
}

void CrosshairChange(bool on)
{
    if (on && !doOnce)
    {
        crosshair.color = Color.red;
    }
    else
    {
        crosshair.color = Color.white;
        isCrosshairActive = false;
    }
}
}

```

```
}
```

DoorController:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class MyDoorController : MonoBehaviour
{
    private Animator doorAnim;

    private bool doorOpen = false;

    private void Awake()
    {
        doorAnim = gameObject.GetComponent<Animator>();
    }

    public void PlayAnimation()

    {
        if (!doorOpen)
        {
            doorAnim.Play("DoorOpen", 0, 0.0f);
            doorOpen = true;
        }
        else
        {
            doorAnim.Play("DoorClose", 0, 0.0f);
            doorOpen = false;
        }
    }
}
```

Game Over:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class game_over : MonoBehaviour
{
    private void OnTriggerStay(Collider other)
    {
        if (other.gameObject.tag == "Player" )
        {

            SceneManager.LoadScene(0);
        }
    }
}
```

Press “F” to fly text:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class PressF : MonoBehaviour
{
    public GameObject UiObject;

    void Start()
    {
        UiObject.SetActive(false);
    }
}
```

```

void OnTriggerStay(Collider other)
{
    if (other.tag == "Player")
    {
        UiObject.SetActive(true);
    }
    else
    {
        UiObject.SetActive(false);
    }
}

void OnTriggerExit(Collider other)
{
    UiObject.SetActive(false);
}

}

```

Press F change Scene:

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class FlightSceneLoader : MonoBehaviour

{

    // Update is called once per frame

```

```
private void OnTriggerStay(Collider other)
{
    if (other.gameObject.tag == "Player" && Input.GetKeyDown(KeyCode.F))
    {

        SceneManager.LoadScene(2);
    }
}

}
```