Ελληνικό Μεσογειακό Πανεπιστήμιο

Σχολή Μηχανικών

Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

Πτυχιακή Εργασία

Τίτλος: Design & Implementation of a Horror-type game in Unity

Καμνάκης Στυλιανός (ΑΜ: ΤΠ4345)

Επιβλέπων εκπαιδευτικός : Παχουλάκης Ιωάννης

Επιτροπή αξιολόγησης :

Ημερομηνία παρουσίασης : 27/09/2019

Abstract

The subject of the present thesis is the creation of a horror-type video game using Unity. The main goal of this thesis is the production of high-quality graphics. While working on the project, a variety of software is used, such as Blender for modelling and UV mapping, Substance Painter for texturing and Unity 2019.3 HDRP for the materials, effects post-processing and rendering. In the present thesis, all the needed technologies to produce the visuals are described as well as the workflow and the software that was used and why. Even though the main goal is the creation of the visuals of the game, algorithms for the player's movement and interaction with dynamic objects were developed as well as sound effects wherever necessary.

Σύνοψη

Το θέμα της παρούσας πτυχιακής εργασίας είναι η δημιουργία ενός παιχνιδιού τρόμου με την χρήση του Unity. Ο βασικός στόχος της πτυχιακής εργασίας είναι η παραγωγή υψηλής ποιότητας γραφικών για παιχνίδια. Στα πλαίσια της πτυχιακής εργασίας χρησιμοποιούνται τα παρακάτω εργαλεία το Blender για μοντελοποίηση και δημιουργία UV Maps, το Substance Painter για την παραγωγή των texture και το Unity 2019.3 HDRP για την δημιουργία των materials, effects και το τελικό rendering. Στην συνέχεια περιγράφονται οι τεχνολογίες που χρειάστηκαν για την ανάπτυξη των γραφικών, τρόποι για την παραγωγή ρεαλιστικών μοντέλων και σκηνών, καθώς και τα λογισμικά που χρησιμοποιούνται και ο λόγος που χρησιμοποιούνται. Γίνονται επίσης αναφορές στην μεθοδολογία που ακολουθείται για να επιτευχθεί το τελικό αποτέλεσμα. Παρόλο που ο βασικός στόχος είναι η δημιουργία των οπτικών του παιχνιδιού, αναπτύσσονται και περιγράφονται βασικοί αλγόριθμοι για την κίνηση του παίχτη αλλά και την αλληλεπίδραση του με δυναμικά μοντέλα, καθώς και ηχητικά εφέ όπου ήταν απαραίτητο.

Πίνακας περιεχομένων

1 Γραφικά υπολογιστών 10
1.1 Τα βασικά των γραφικών υπολογιστών 10
1.2 Τύποι γραφικών υπολογιστή 10
2 3D Modelling 11
2.1 Τα βασικά της τρισδιάστατης μοντελοποίησης 11
2.2 3D Μοντέλο (3D Model) 11
2.2.1 Τύποι 3D μοντέλων 11
2.3 Face Normals 12
2.4 Object Primitives 12
2.5 3D Χώρος (3D Space) 13
2.6 Modeling operations 13
2.7 Non-Destructive Modeling 14
2.8 Τοπολογία (Topology) 17
2.9 Sculpting 18
3 UV Mapping 21
3.1 Τα βασικά του UV Mapping 21
3.2 Τα συστατικά του UV Mapping 21
3.3 Ένα καλό UV Map 22
4 Texture Maps 23
4.1 Τι είναι ένα Texture 23
4.2 Οι διάφοροι τύποι Texture 23
4.3 Normals Baking 26
5 Materials & Shaders 27
5.1 Shaders 27
5.1.1 Shader Graph 27
5.1.2 Vertex Shader 27
5.1.3 Tessellation Shader & LOD System 28
5.2 Materials 29
6 Lighting 30
6.1 Φωτισμός στα γραφικά υπολογιστών 30
6.2 Πηγές φωτός 30
6.3 Environment Lighting 33
6.4 Volumetric Lighting 33

6.5 Baked Lighting 34
6.6 Global Illumination 34
6.7 Light Probes 35
6.8 Reflection Probes 36
7 Rendering 37
7.1 Τύποι Rendering 37
7.1.1 Realtime Rendering 37
7.1.2 Non-Realtime Rendering 37
7.2 PBR (Physically Based Rendering) 38
7.3 Occlusion Culling 39
8 Environment Art for Games 40
8.1 Τι είναι Environment Art 40
8.2 Terrain System 40
8.3 Modular Assets 41
8.3.1 Modular Models 41
8.3.2 Trim Textures 41
8.4 Props 42
8.5 Particle Systems 43
8.6 Post-Processing 44
9 Τα λογισμικά που θα χρησιμοποιηθούν 46
9.1 Blender 46
9.2 Materialize 47
9.3 Substance Painter 47
9.4 Unity 48
10 Διαδικασία ανάπτυξης γραφικών 50
10.1 Δημιουργία αντικειμένων 50
10.1.1 Συλλογή πληροφοριών και εικόνων αναφοράς 50
10.1.2 Μοντελοποίηση και UV unwrapping 51
10.1.3 Texturing 54
10.1.4 Import στην μηχανή παιχνιδιού 56
10.2 Τα βήματα που ακολουθήθηκαν 57
10.2.1 Συλλογή πληροφοριών 57
10.2.2 Δημιουργία βασικού level 58
10.2.3 Import level στην μηχανή παιχνιδιού 60

10.2.4 Δημιουργία βασικού φωτισμού και εφέ 61
10.2.5 Δημιουργία props 61
10.2.6 Import props στην μηχανή παιχνιδιού 63
10.2.7 Δημιουργία ολοκληρωμένου level 64
10.2.8 Δημιουργία player, scripts και animations 64
10.2.9 Βελτιστοποίηση παιχνιδιού 68
11 Αποτελέσματα 70
11.1 Το παιχνίδι 70
11.2 Δυσκολίες κατά την παραγωγή 74
11.3 Πιθανές μελλοντικές εξελίξεις και διορθώσεις 74
12 Βιβλιογραφία 75

Πίνακας εικόνων

Παράδειγμα 1 Face Normals 12
Παράδειγμα 2 Πρωτόγονα Αντικείμενα 12
Παράδειγμα 3 3D χώρος 13
Παράδειγμα 4Απλός Κύβος 13
Παράδειγμα 5 3D μοντέλο από κύβο 13
Παράδειγμα 6 Subdivision Surface Modifier 15
Παράδειγμα 7 Boolean Modifier 15
Παράδειγμα 8 Bevel Modifier 16
Παράδειγμα 9 Solidify Modifier 16
Παράδειγμα 10 Array Modifier 17
Παράδειγμα 11 Bad vs Good Topology 17
Παράδειγμα 12 Sculpt Brush - Alpha Texture 19
Παράδειγμα 13 Sculpted Sphere 19
Παράδειγμα 14 Sculpted Rocks 20
Παράδειγμα 15 UV Mapping 21
Παράδειγμα 16 How a good UV Map looks Like 22
Παράδειγμα 17 Diffuse vs Albedo Map 23
Παράδειγμα 18 Ambient Occlusion Map 24
Παράδειγμα 19 Normal Map 24
Παράδειγμα 20 Height ή Bump Map 24
Παράδειγμα 21 Roughness, Metallic Maps 25
Παράδειγμα 22 Smoothness, Specular Map 25
Παράδειγμα 23 Environment Texture 25
Παράδειγμα 24 Normals Baking 26
Παράδειγμα 25 Shader Graph 27
Παράδειγμα 26 How vertex shader works 28
Παράδειγμα 27 LOD System in Unity 28

Παράδειγμα 28 Materials applied on Spheres 29
Παράδειγμα 29 Unity's Lit Shader parameters 29
Παράδειγμα 30 Point Light 30
Παράδειγμα 31 Area Light 31
Παράδειγμα 32 Spot Light 31
Παράδειγμα 33 Directional Lighting 32
Παράδειγμα 34 Emissive Materials 32
Παράδειγμα 35 Environment Lighting 33
Παράδειγμα 36 Volumetric Lighting 33
Παράδειγμα 37 Baked Lighting 34
Παράδειγμα 38 Global Illumination 35
Παράδειγμα 39 Light Probes 35
Παράδειγμα 40 Reflection Probes 36
Παράδειγμα 41 Final Render vs Solid Color with wireframe view 38
Παράδειγμα 42 Non-PBR vs PBR Shader 39
Παράδειγμα 43 Occlusion Culling 39
Παράδειγμα 44 Terrain created with Unity 40
Παράδειγμα 45 Modular Assets of a building 41
Παράδειγμα 46 Trim Textures 42
Παράδειγμα 47 Trim Textures Applied 42
Παράδειγμα 48 Game Props 1 43
Παράδειγμα 49 Game Props 2 43
Παράδειγμα 50 Particles Fire Effect 44
Παράδειγμα 51 Post-Processing 45
Παράδειγμα 52 Blender Layout 46
Παράδειγμα 53 Simple Image 47
Παράδειγμα 54 All texture maps generated and applied 47
Παράδειγμα 55 Substance Painter Software 48
Παράδειγμα 56 Unity Layout 49
Παράδειγμα 57 Occlusion Areas & Portals in Unity 69
Παράδειγμα 58 Occlusion Culling Visualization 69

Μεθοδολογία 1 Το παράδειγμα που θα χρησιμοποιηθεί 50
Μεθοδολογία 2 Φωτογραφία αναφοράς παραδείγματος 51
Μεθοδολογία 3 Πληροφορίες παραδείγματος 51
Μεθοδολογία 4 Μοντελοποίηση #1 52
Μεθοδολογία 5 Μοντελοποίηση #2 52
Μεθοδολογία 6 Μοντελοποίηση #3 52
Μεθοδολογία 7 Μοντελοποίηση #4 52
Μεθοδολογία 8 Μοντελοποίηση #5 52
Μεθοδολογία 9 Μοντελοποίηση #6 52
Μεθοδολογία 10 UV Unwrapping 53
Μεθοδολογία 11 UV map της πόρτας 53
Μεθοδολογία 12 UV map του συρταριού 53
Μεθοδολογία 13 Baked Mesh Maps 54
Μεθοδολογία 14 Wood Material 55

Μεθοδολογία 15 Wood Material Preview 55
Μεθοδολογία 16 Fully textured model 55
Μεθοδολογία 17 Exported Textures 55
Μεθοδολογία 18 Game Ready Object 56
Μεθοδολογία 19 Unity HDRP Material Properties 56
Μεθοδολογία 20 Level Παράδειγμα #1 57
Μεθοδολογία 21 Level Παράδειγμα #2 57
Μεθοδολογία 22 Level Παράδειγμα #3 57
Μεθοδολογία 23 Level Παράδειγμα #4 57
Μεθοδολογία 24 Modular Parts 58
Μεθοδολογία 25 Modular Parts United #1 58
Μεθοδολογία 26 Modular Parts United #2 59
Μεθοδολογία 27 Modular Parts United #3 59
Μεθοδολογία 28 Modular Parts United #4 59
Μεθοδολογία 29 Level Layout in Unity 60
Μεθοδολογία 30 No Light Scene 61
Μεθοδολογία 31 Directional Light & Volumetric 61
Μεθοδολογία 32 Scene with Lights 61
Μεθοδολογία 33 Light Bleed Solution 61
Μεθοδολογία 34 Παράδειγμα prop #1 62
Μεθοδολογία 35 Παράδειγμα prop #2 62
Μεθοδολογία 36 Παράδειγμα prop #3 62
Μεθοδολογία 37 Παράδειγμα prop #4 62
Μεθοδολογία 38 Παράδειγμα prop #5 62
Μεθοδολογία 39 Παράδειγμα prop #6 62
Μεθοδολογία 40 Παράδειγμα prop #7 62
Μεθοδολογία 41 Παράδειγμα prop #8 62
Μεθοδολογία 42 Παράδειγμα prop #9 63
Μεθοδολογία 43 Παράδειγμα prop #10 63
Μεθοδολογία 44 Παράδειγμα prop #11 63
Μεθοδολογία 45 Παράδειγμα prop #12 63
Μεθοδολογία 46 Mansion Props 63
Μεθοδολογία 47 Props inserted to level 64
Μεθοδολογία 48 Unity Default FPS Controller 65
Μεθοδολογία 49 Simple door animation 65
Μεθοδολογία 50 Open-Close Door Animator 66

Gameplay 1 Hall 70
Gameplay 2 Hall from Top 71
Gameplay 3 Room #1 71
Gameplay 4 Room #2 72
Gameplay 5 Library 72
Gameplay 6 Green Room 73
Gameplay 7 Music Room 73

1 Γραφικά υπολογιστών

1.1 Τα βασικά των γραφικών υπολογιστών

Τα γραφικά (Computer Graphics ή CG) είναι ένας επιστημονικός κλάδος της πληροφορικής που αφορά την οπτική αναπαράσταση εικόνων, σχημάτων, βίντεο, διεπαφών χρήστη-υπολογιστή αλλά και δεδομένων. Σκοπός των γραφικών είναι δώσουν στον τελικό χρήστη μια πληροφορία.

Τα γραφικά έχουν μπει για τα καλά σε πολλούς τύπους πολυμέσων και εφαρμογών όπως ταινίες, animation, παιχνίδια, διαφημίσεις καθώς και στα CAD. Για αυτό το λόγο υπάρχουν πολλές νέες τεχνολογίες για την δημιουργία και επεξεργασία γραφικών με την μορφή εξειδικευμένων λογισμικών, που ποικίλουν ανάλογα τον τύπο των γραφικών, π.χ. απλά λογισμικά επεξεργασίας εικόνας, λογισμικά τρισδιάστατης μοντελοποίησης αλλά και συνδυασμός πολλών τύπων. (Γραφικά Υπολογιστών, n.d.)

1.2 Τύποι γραφικών υπολογιστή

Δυσδιάστατα γραφικά (2D Graphics) : Τα 2D γραφικά αναπαριστούν απλές εικόνες δύο διαστάσεων στην οθόνη μιας συσκευής. Εφαρμόζονται για την δημιουργία διεπαφών χρήστη-υπολογιστή αλλά και για εικονογραφήσεις βιβλίων καθώς και για την δημιουργία απλούστερων κινουμένων σχεδίων.

Τρισδιάστατα γραφικά (3D Graphics) : Τρισδιάστατα γραφικά αναπαριστούν αντικείμενα τριών διαστάσεων στις οθόνες ψηφιακών συσκευών. Το γεγονός ότι η αναπαράσταση χρησιμοποιεί τρεις διαστάσεις τα καθιστά περισσότερο ρεαλιστικά. Χρησιμοποιεί μαθηματικούς τύπους και αλγορίθμους σε ένα σύστημα συντεταγμένων τριών διαστάσεων. Χρησιμοποιούνται εκτενώς στα παιχνίδια νέας γενιάς αλλά και σε ταινίες όπως και στην εικονική πραγματικότητα (VR). Κατηγοριοποιούνται σε δύο είδη, στα στατικά γραφικά υπολογιστή και στα γραφικά πραγματικού χρόνου.

Τα **στατικά γραφικά υπολογιστή** αποτελούνται από σκηνές που περιέχουν αντικείμενα, φωτισμό και χρώμα. Αυτά τα γραφικά είναι προϋπολογισμένα και προ επεξεργασμένα. Δηλαδή το τελικό αποτέλεσμα είναι στατικό και δεν αποδίδεται την ώρα της εκτέλεσης της εφαρμογής. Δεν είναι δια-δραστικά εφόσον στην ουσία είναι αρχεία βίντεο που έχουν αποθηκευτεί από το λογισμικό που τα έχει δημιουργήσει.

Τα **γραφικά πραγματικού χρόνου** είναι σκηνές με αντικείμενα τοποθετημένα σε κάποιες συντεταγμένες ενός χώρου τριών διαστάσεων. Η διαφορά τους είναι ότι αναπαρίστανται κατά την διάρκεια εκτέλεσης μια εφαρμογής. Τα γραφικά ενός παιχνιδιού νέας γενιάς είναι γραφικά πραγματικού χρόνου. Είναι πιο απαιτητικά σε πόρους αφού απαιτούν την χρήση του επεξεργαστή και της κάρτας γραφικών. Για την προβολή τους είναι αναγκαία η χρήση μιας μηχανής πραγματικού χρόνου ή μιας παιχνιδομηχανής. Τεχνολογίες αναγκαίες για την δημιουργία αυτής της κατηγορίας γραφικών είναι βιβλιοθήκες όπως Direct3D και OpenGL. (Γραφικά Υπολογιστών, n.d.)

2 3D Modelling

2.1 Τα βασικά της τρισδιάστατης μοντελοποίησης

Στα γραφικά υπολογιστή, μοντελοποίηση σε 3D είναι η διαδικασία κατά την οποία παράγεται μια μαθηματική αναπαράσταση μιας οποιασδήποτε επιφάνειας ενός αντικειμένου (έμψυχου η άψυχου) σε τρείς διαστάσεις με κάποιο ειδικευμένο λογισμικό. Το αποτέλεσμα ονομάζεται μοντέλο 3D. Ο δημιουργός αυτών τον μοντέλων λέγεται 3d artist. Ένα 3D μοντέλο μπορεί είτε να αναπαρασταθεί σε δυσδιάστατη εικόνα με μία διαδικασία που λέγεται 3D rendering, είτε να ενσωματωθεί σε εφαρμογές που υποστηρίζουν την αναπαράσταση 3D αντικειμένων, είτε και να εκτυπωθεί από τρισδιάστατους εκτυπωτές. Η 3D μοντελοποίηση γίνεται από ειδικά λογισμικά τρισδιάστατης μοντελοποίησης που συνήθως είναι μέρος λογισμικών που περιέχουν περισσότερα πεδία της δημιουργίας γραφικών υπολογιστών.

2.2 3D Μοντέλο (3D Model)

Ένα μοντέλο 3D είναι μια μαθηματική αναπαράσταση ενός οποιουδήποτε τρισδιάστατου αντικειμένου του πραγματικού κόσμου (άψυχο ή έμψυχο) σε ένα περιβάλλον 3D λογισμικού. Εν αντιθέσει με τα 2D, τα 3D μοντέλα μπορούν να προβληθούν σε ειδικά λογισμικά από οποιαδήποτε οπτική γωνία καθώς και να μεγεθυνθούνε, να περιστραφούν και να επεξεργαστούν.

2.2.1 Τύποι 3D μοντέλων

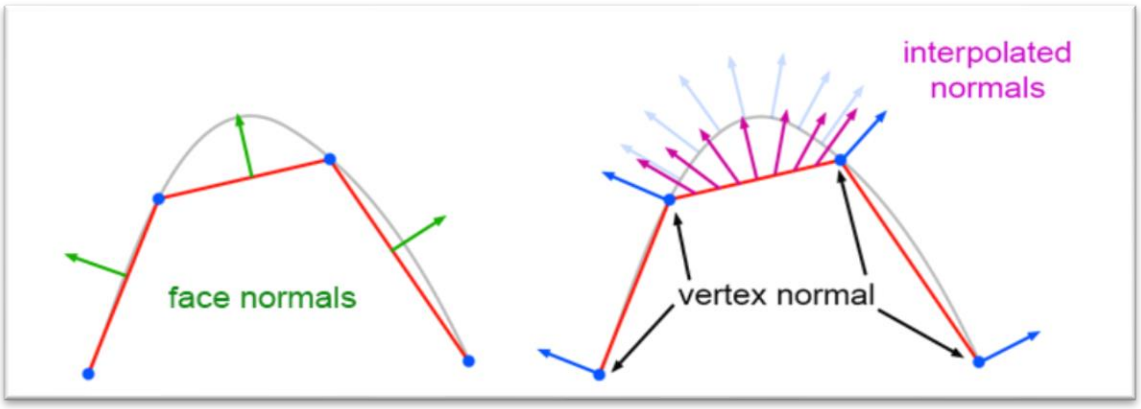
Υπάρχουν δύο βασικοί τύποι 3D μοντέλων που χρησιμοποιούνται στην βιομηχανία παιχνιδιών και ταινιών. Οι κυριότερες διαφορές τους είναι στον τρόπο που δημιουργούνται και χειρίζονται παρά στο τελικό αποτέλεσμα.

NURBS Surface : Μια επιφάνεια NURBS (non-uniform rational basis spline surface) είναι ένα μοντέλο με λεία επιφάνεια που δημιουργήθηκε με την χρήση Bezier καμπύλων. Για να δημιουργηθεί μια NURBS επιφάνεια, ο καλλιτέχνης (χρήστης του λογισμικού) σχεδιάζει δύο ή περισσότερες καμπύλες στον 3D χώρο, οι οποίες μπορούν να χειραγωγηθούν μετακινώντας “λαβές” που ονομάζονται κορυφές χειρισμού (control vertices) στους άξονες x,y,z. Το λογισμικό παρεμβάλει την απόσταση μεταξύ των καμπύλων και δημιουργεί ένα πλέγμα ανάμεσα τους. Οι επιφάνειες NURBS έχουν μεγαλύτερη μαθηματική ακρίβεια και συνεπώς χρησιμοποιούνται περισσότερο στην αυτοκινητοβιομηχανία.

Πολυγωνικό μοντέλο (Polygonal Model) : Το πολυγωνικό μοντέλο είναι η πιο συνηθισμένη μορφή 3D μοντέλου στην βιομηχανία ταινιών και παιχνιδιών. Το βασικό αντικείμενο ενός πολυγωνικού μοντέλου είναι ένα **σημείο (vertex)**, το οποίο έχει κάποιες συντεταγμένες στον 3D χώρο (x, y, z). Όταν δύο σημεία ενωθούν τότε έχουμε ένα **άκρο (edge)** το οποίο είναι μια ευθεία γραμμή. Όταν τρία ή περισσότερα σημεία ενωθούν τότε έχουμε ένα **πολύγωνο (polygon).** Ένα πολύγωνο με τρία σημεία (**triangle**) είναι απλούστερη μορφή πολυγώνου. Όταν ένα πολύγωνο έχει 4 σημεία λέγεται τετράγωνο τύπου (**quad).** Μια συλλογή από πολύγωνα με κοινά σημεία λέγεται **στοιχείο (element) ή πλέγμα (mesh)**. Κάθε πολύγωνο ενός στοιχείου λέγεται **πρόσωπο (face)**. (3D Μοντελοποίηση, n.d.)

2.3 Face Normals

Τα **Face Normals** δείχνουν την κατεύθυνση στην οποία θα είναι ορατό ένα πρόσωπο (face) και είναι απαραίτητο για την μεταφορά του φωτισμού και την ανίχνευση των ακτινών του. Ορισμένα συστήματα χρησιμοποιούν **vertex normal** (σημείων) για μεγαλύτερη ακρίβεια του φωτισμού όμως έχει περισσότερο κόστος πόρων. Κάθε face έχει δύο face normals όμως το ένα θεωρείται έγκυρο ενώ το άλλο ονομάζεται backface (πίσω επιφάνεια). Στα πιο σύγχρονα λογισμικά μπορεί να επιλέξει ο χρήστης εάν το backface θα είναι ορατό ή όχι.

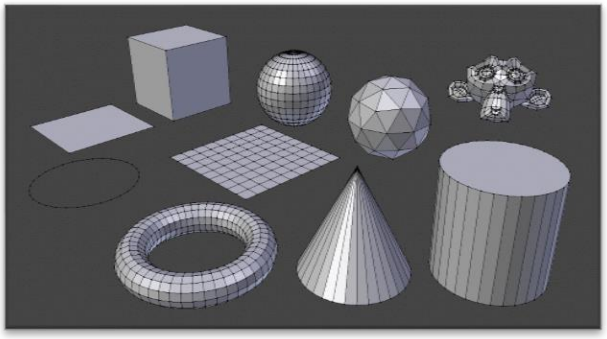


Παράδειγμα 1 Face Normals

https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/shading-normals

2.4 Object Primitives

Πρωτόγονα αντικείμενα (primitive objects) είναι αντικείμενα που έχουν βασικά σχήματα όπως κύβος, σφαίρα, κύλινδρος κτλ. Αυτά τα αντικείμενα χρησιμοποιούνται ως ένα εύκολος τρόπος να ξεκινήσει ο χρήστης να μοντελοποιεί το αντικείμενο του. Αφού ξεκινήσει με το βασικό σχήμα που θέλει, το οποίο δημιουργεί το λογισμικό, μετά με διάφορες λειτουργίες μπορεί να το τροποποιήσει και να δημιουργήσει κάτι εντελώς καινούργιο.

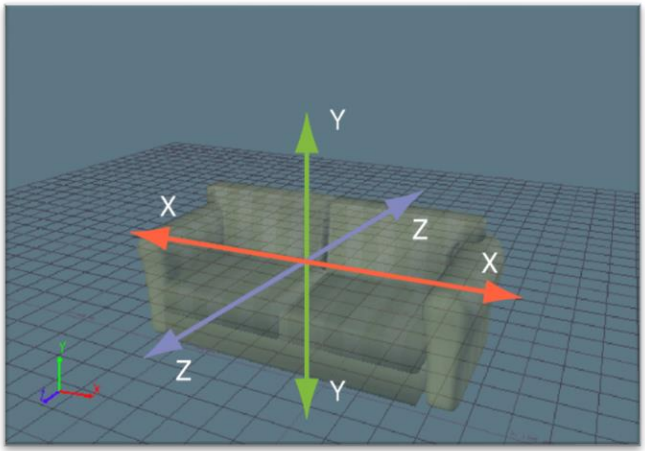


Παράδειγμα 2 Πρωτόγονα Αντικείμενα

https://docs.blender.org/manual/en/latest/modeling/meshes/primitives.html

2.5 3D Χώρος (3D Space)

Ο τρισδιάστατος χώρος (3D Space) αποτελείται από τρείς τιμές-παραμέτρους x, y, z (συντεταγμένες) οι οποίες είναι αναγκαίες για να καθοριστεί η θέση ενός αντικειμένου. Συνήθως η τιμή του x αφορά το πλάτος, η τιμή του y το ύψος και η τιμή του z το βάθος. Ο άξονας που αναπαριστά κάθε τιμή αλλάζει από λειτουργικό σε λειτουργικό καθώς σε ορισμένα μπορεί και να τροποποιηθεί από τον χρήστη ώστε να βολεύει τις ανάγκες του. Μια 3D σκηνή αποτελείται από ένα σύνολο 3D μοντέλων στον τρισδιάστατο χώρο.

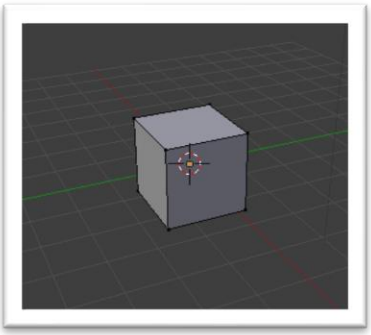


Παράδειγμα 3 3D χώρος

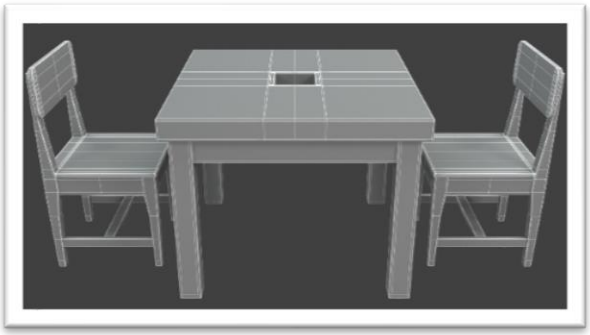
http://help.autodesk.com/view/MOBPRO/2017/ENU/?guid=GUID-F90A9BF3-1A41-4FB7-AE58-53D73BDDEF6B

2.6 Modeling operations

Κάθε πρόγραμμα μοντελοποίησης έχει τις δικές του λειτουργίες τις οποίες το κάθε ένα υλοποιεί με διαφορετικούς αλγορίθμους. Υπάρχουν χιλιάδες λειτουργίες σε κάθε πρόγραμμα ωστόσο στην πολυγωνική μοντελοποίηση υπάρχουν κάποια βασικά εργαλεία που βοηθούν τον χρήστη να φτάσει ένα πρωτόγονο αντικείμενο σε ένα πολύ προχωρημένο επίπεδο.



Παράδειγμα 4Απλός Κύβος



Παράδειγμα 5 3D μοντέλο από κύβο

Βασικές λειτουργίες μοντελοποίησης (Modeling operations) :

- **Περιήγηση στον 3D χώρο (View tools)**. Είναι η βασική λειτουργία μοντελοποίησης εφόσον χρησιμοποιείται συνέχειας για να μπορεί ο χρήστης να βλέπει και να τροποποιεί το πλέγμα από οποιαδήποτε οπτική, περιλαμβάνει μετακίνηση, περιστροφή και μεγέθυνση (drag, rotate, zoom).
- **Μετακίνηση (Translate)**. Κάθε αντικείμενο έχει ένα σημείο προέλευσης (origin point). Αυτό το σημείο συνήθως είναι στο κέντρο της μάζας αλλά μπορεί να τοποθετηθεί οπουδήποτε στον 3D χώρο. Όταν μετακινείς ένα αντικείμενο στην ουσία μετακινείς το σημείο προέλευσης και μαζί με αυτό μετακινούνται και τα σημεία του πλέγματος (vertices).
- **Περιστροφή (Rotate)**. Κάθε αντικείμενο μπορεί να περιστραφεί ώστε να έχει την κατεύθυνση που θέλει ο δημιουργός. Η περιστροφή γίνεται επίσης από το σημείο προέλευσης.
- **Μεγέθυνση – Σμίκρυνση (Scale)**. Μια λειτουργία που επίσης γίνεται από το σημείο προέλευσης και βοηθάει τον δημιουργό να μεγαλώσει η να μικρύνει ένα αντικείμενο.

Λειτουργίες τροποποίησης πλέγματος

- Μετακίνηση σημείων, άκρων, προσώπων (vertices, edges, faces).
- Περιστροφή άκρων, προσώπων.
- Διαγραφή σημείων, άκρων, προσώπων.

Μερικά βασικά εργαλεία :

- Knife tool: Με αυτό το εργαλείο μπορείς να προσθέσεις να ένα ή παραπάνω σημεία σε ένα η παραπάνω άκρα.
- Loop cut: Με αυτό το εργαλείο μπορεί να προσθέσεις άκρα σε ένα η παραπάνω πρόσωπο προς κάποια κατεύθυνση.
- Add face: Υπάρχει η περίπτωση να υπάρχουν 3 η παραπάνω σημεία που να σχηματίζουν ένα πολύγωνο αλλά να μην υπάρχει το πρόσωπο που κάνει το πολύγωνο ορατό. Με αυτή τη λειτουργία προστίθεται ένα πρόσωπο στο πολύγωνο
- Extrude: Διπλασιάζει τα επιλεγμένα σημεία αλλά τα κρατάει συνδεμένα στο πλέγμα.
- Inset: Το εργαλείο αυτό παίρνει τα επιλεγμένα πρόσωπα και δημιουργεί μια εισροή σε αυτό με το επιλεγμένο πάχος και βάθος πάντα σχετικό με τα επιλεγμένα πρόσωπα ακόμα και σε περίπλοκα πλέγματα.

2.7 Non-Destructive Modeling

Μη-καταστροφική μοντελοποίηση (non-destructive modeling) είναι όταν κατά την διάρκεια την μοντελοποίησης ο καλλιτέχνης χρησιμοποιεί εργαλεία που του προσφέρει το λογισμικό ώστε να επεξεργάζεται το αντικείμενο χωρίς να το αλλάζει στην πραγματικότητα. Έτσι μπορεί να επιστρέψει σε οποιοδήποτε στάδιο θέλει, όποτε θέλει, γρήγορα και εύκολα. Τα εργαλεία που επιτρέπουν την επίτευξη της μη-καταστροφικής μοντελοποίησης λέγονται **τροποποιητές (modifiers)**. Είναι μια τεχνική που χρησιμοποιείται πολύ συχνά στην δημιουργία αντικειμένων για παιχνίδια και ταινίες, καθώς προσφέρει αποδοτικότητα και "ελαστικότητα". Αυτό που κάνει τους τροποποιητές τόσο δημοφιλής είναι ότι αλλάζοντας μερικές παραμέτρους μπορείς να κάνεις τεράστιες αλλαγές αλλά και μικρές στο αντικείμενο.

Οι τροποποιητές διαφέρουν από λειτουργικό σε λειτουργικό ωστόσο κάποιοι “standards” υπάρχουν σε όλα καθώς χρησιμοποιούνται πολύ συχνά.

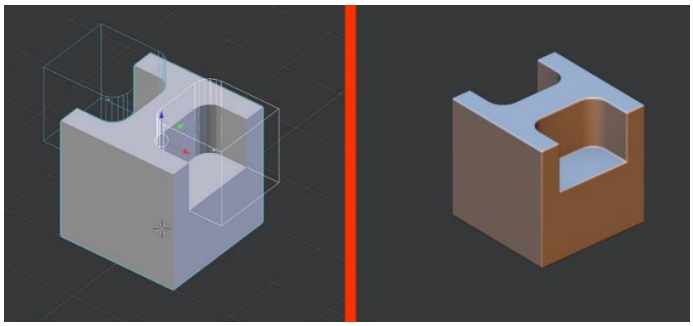
Οι πιο βασικοί modifiers είναι οι εξής :

- **Subdivision Surface Modifier (Sub-surf Modifier)**: Είναι πιθανότατα ο περισσότερο χρησιμοποιημένος τροποποιητής, αφού σου δίνει την δυνατότητα να μοντελοποιήσεις περίπλοκες λείες επιφάνειες με απλά και low-poly (χαμηλό αριθμό πολυγώνων) πλέγματα. Αυτό κάνει την διαδικασία της μοντελοποίησης ευκολότερη καθώς δεν χρειάζεται ο δημιουργός να χειρίζεται τεράστια μάζα δεδομένων για κάθε αλλαγή που θέλει να κάνει και το αντικείμενο έχει λεία επιφάνεια. (Blender Documentation, n.d.)

Παράδειγμα 6 Subdivision Surface Modifier

https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/subdivision_surface.html

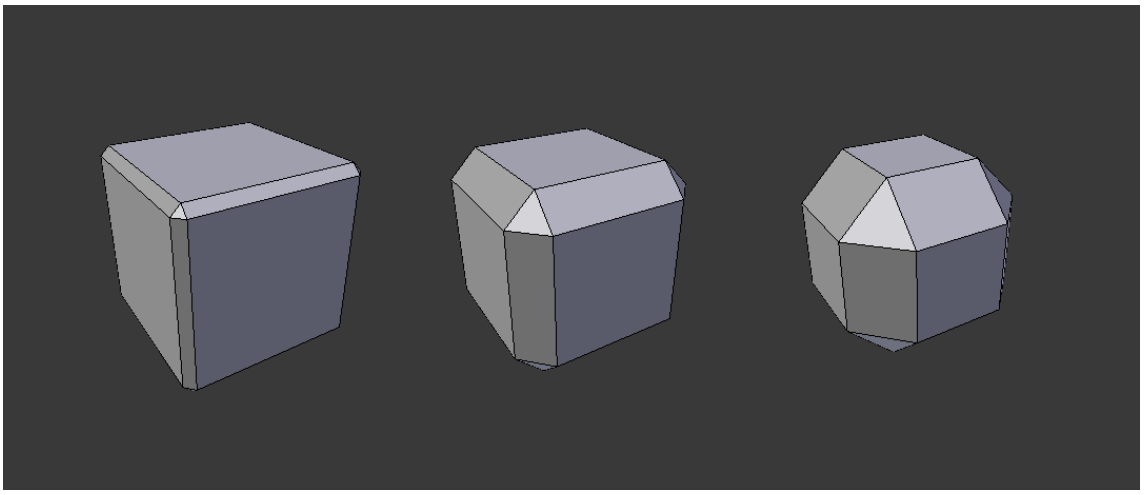
- **Boolean Modifier**: Ένας επίσης πολύ βασικός τροποποιητής αφού σου επιτρέπει να πετύχεις αλλαγές που θα ήταν πολύ δύσκολο να πετύχεις χωρίς αυτόν. Ο Boolean modifier χρησιμοποιεί ένα δεύτερο αντικείμενο από το οποίο παίρνει δεδομένα όπως η θέση, η περιστροφή, το μέγεθος και το αντικείμενο αυτό καθαυτό. Το δεύτερο αντικείμενο λέγε **στόχος (target).** Ο Boolean modifier έχει τρείς βασικές λειτουργίες. (Blender Documentation, n.d.)
  - **Difference (Διαφορά)**: Όπου το αντικείμενο αφαιρείται από τον στόχο.
  - **Union (Ένωση)**: Όπου ο στόχος προστίθεται στο αντικείμενο.
  - **Intersect (Τομή)**: Όπου ο στόχος αφαιρείται από το αντικείμενο.



Παράδειγμα 7 Boolean Modifier

https://www.youtube.com/watch?v=RDKt_lNAHss

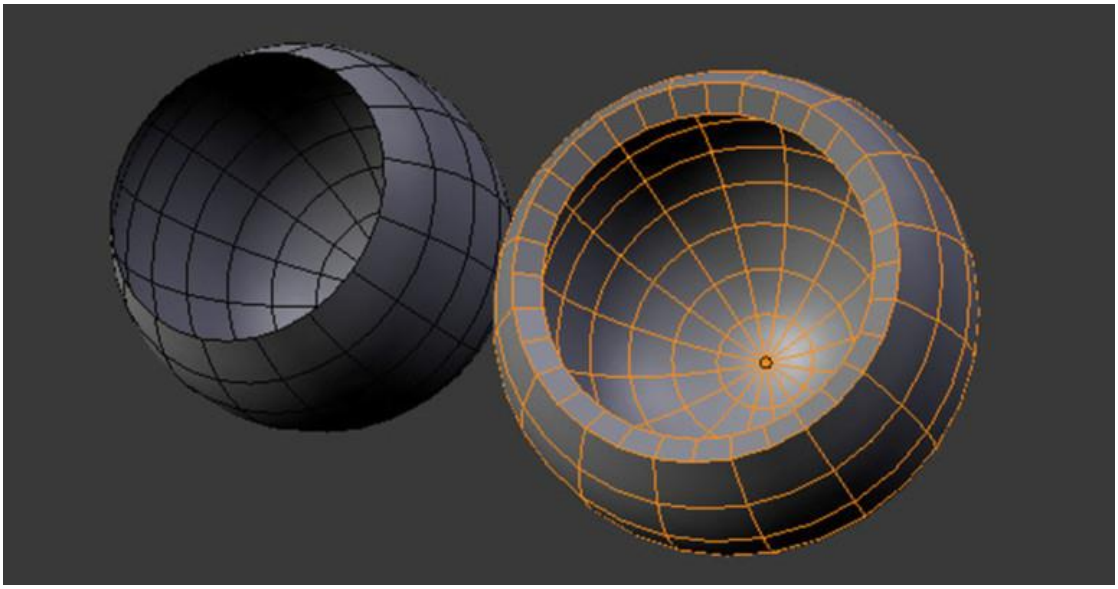
- **Bevel Modifier**: Αυτός ο τροποποιητής χρησιμοποιείται συνήθως όταν ο δημιουργός θέλει να πετύχει ρεαλιστικά άκρα στο αντικείμενο (καθώς καμία γωνία στον πραγματικό κόσμο δεν είναι τέλεια -90°-). Αυτό το πετυχαίνει κόβοντας λοξά τις γωνίες του αντικειμένου με το επιθυμητό πλάτος καθώς και το πλήθος των κοψιμάτων. (Blender Documentation, n.d.)



Παράδειγμα 8 Bevel Modifier

https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/bevel.html

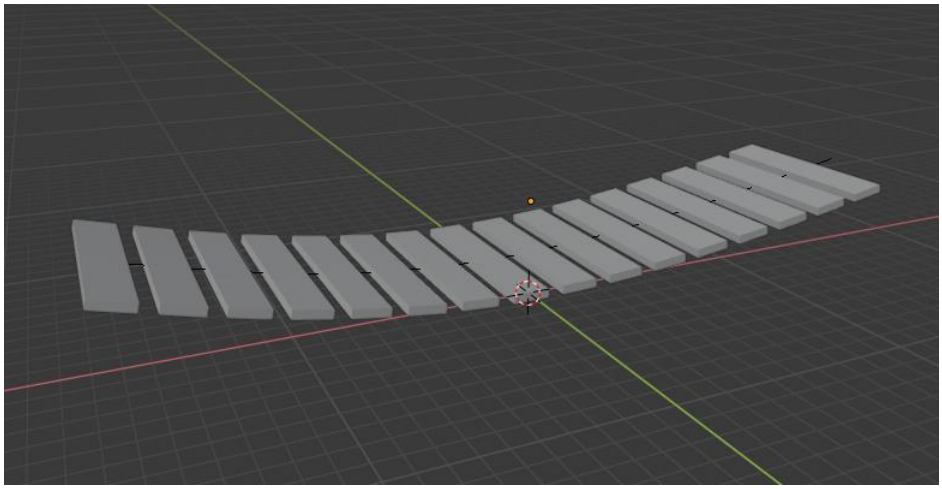
- **Solidify Modifier**: Αυτός ο τροποποιητής παίρνει ένα αντικείμενο και του προσθέτει βάθος. Ο δημιουργός έχει την επιλογή της κατεύθυνσης όπου θα προστεθεί το βάθος αλλά και το πόσο βαθύ θα κάνει το αντικείμενο. (Blender Documentation, n.d.)



Παράδειγμα 9 Solidify Modifier

https://spolearninglab.com/curriculum/software/3d_modeling/blender/blender_svg.html

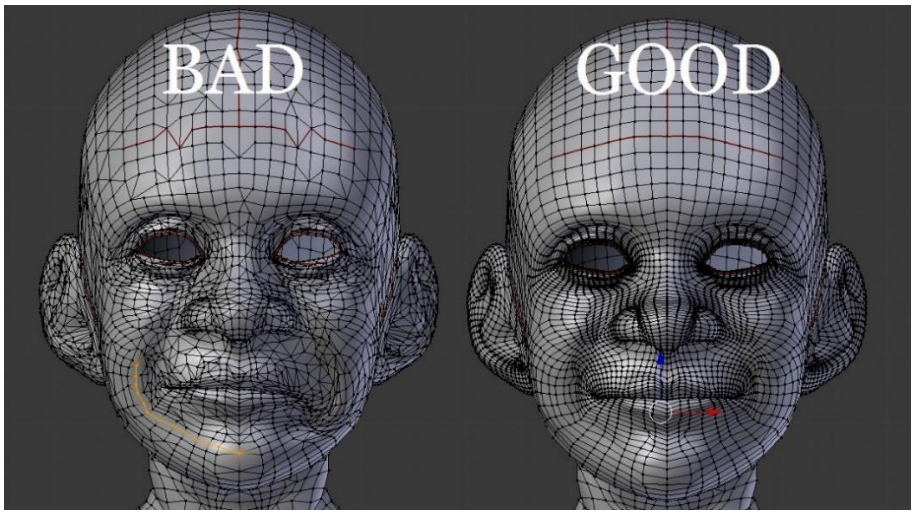
- **Array Modifier**: Χρησιμοποιείται για να πολλαπλασιάσει ένα αντικείμενο n φορές προς μία κατεύθυνση. Είναι αρκετά χρήσιμο σε αντικείμενα που έχουν μια βάση η οποία πολλαπλασιάζεται. Ένα παράδειγμα θα μπορούσε να είναι μια αλυσίδα. (Blender Documentation, n.d.)



Παράδειγμα 10 Array Modifier

2.8 Τοπολογία (Topology)

Κάτι πολύ σημαντικό στην διαδικασία της μοντελοποίησης είναι η τοπολογία (topology). Τοπολογία είναι η οργάνωση, η ροή και δομή των σημείων, άκρων και προσώπων ενός μοντέλου. Αυτό που κάνει μια τοπολογία καλή είναι όταν τα σημεία είναι καλά οργανωμένα ώστε το πλέγμα να είναι "καθαρό", λεπτομερές και να φέρει το επιθυμητό αποτέλεσμα.



Παράδειγμα 11 Bad vs Good Topology

http://thilakanathanstudios.com/2016/09/why-do-we-need-topology-in-3d-modeling/

Τα προνόμια μιας καλής τοπολογίας είναι πολλά.

- Είναι ευκολότερο για τον καλλιτέχνη να δουλέψει πάνω σε αυτό.
- Δεν υπάρχουν παραμορφώσεις στο αντικείμενο.
- Βοηθάει και κάποιον τρίτο να δουλέψει με αυτό το αντικείμενο (π.χ. animator)
- Χρησιμοποιεί λιγότερη μνήμη. Όσο λιγότερα σημεία υπάρχουν τόσο πιο ελαφρύ είναι το μοντέλο.

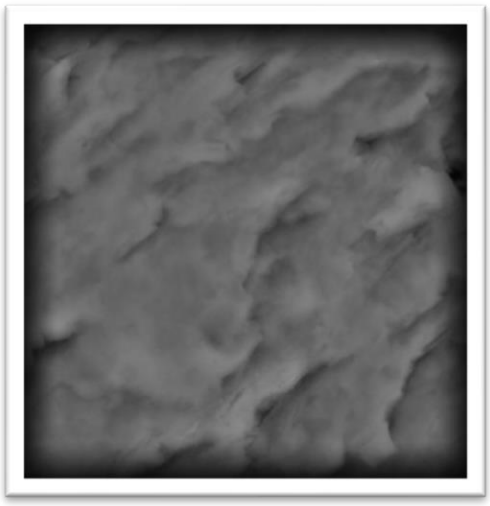
Κάποιοι άγραφοι κανόνες που μπορεί να ακολουθήσει ένα καλλιτέχνης για να πετύχει μια καλή τοπολογία είναι οι εξής :

- Να μην υπάρχουν άσχετα σημεία, άκρα.
- Να μην υπάρχουν διπλά σημεία, άκρα, επιφάνειες (vertices, edges, faces). Αυτό είναι κάτι πολύ σημαντικό και μπορεί να συμβεί πολύ εύκολα χωρίς να το αντιληφθεί ο δημιουργός. Τα πιο σύγχρονα λογισμικά περιέχουν αυτοματισμούς που διορθώνουν αυτό το πρόβλημα σε κάποιες περιπτώσεις, αλλά πολλές φορές ο χρήστης πρέπει να το κάνει μόνος του.
- Να υπάρχουν όσο το δυνατό λιγότερα σημεία (vertices) για το επιθυμητό αποτέλεσμα. Χρήση subsurface modifier αν είναι αναγκαία περισσότερη λεπτομέρεια.
- Τα πολύγωνα του πλέγματος να είναι όλα quads (να έχουν 4 σημεία). Είναι πολύ σημαντικό καθώς εξαφανίζει περίεργες παρενέργειες πάνω στο μοντέλο όπως “μυταλάκια”. Είναι επίσης απαραίτητο για να αντιδρά σωστά το φως επάνω στο μοντέλο κατά την διαδικασία του φωτισμού.
- Να έχουν όλα τα quads περίπου το ίδιο μέγεθος. Σε κομμάτια του μοντέλου που θέλουμε περισσότερη ακρίβεια, όπως ένα πρόσωπο είναι σημαντικό, τα quads που το απαρτίζουν να έχουν περίπου το ίδιο μέγεθος. Αυτό βέβαια δεν είναι κανόνας για όλα τα μοντέλα και μπορεί να παραληφθεί για απλούστερα αντικείμενα όπως ένα τραπέζι ή μια καρέκλα.

Retopologizing είναι η διαδικασία κατά την οποία ο δημιουργός διορθώνει τις ατέλειες του πλέγματος ώστε να έχει μια καλή τοπολογία. Αυτό γίνεται εφόσον κατά την μοντελοποίηση δεν έχει κρατήσει καθαρό το πλέγμα, αλλά πολύ περισσότερο χρησιμοποιείται μετά από ένα άλλος είδος μοντελοποίησης που λέγεται **γλυπτική (sculpting)**.

2.9 Sculpting

3D γλυπτική (3D Sculpting ή Digital Sculpting) είναι μια πολύ δημοφιλής μορφή μοντελοποίησης που χρησιμοποιείται ευρέως στην βιομηχανία παιχνιδιών και ταινιών καθώς βοηθάει να πετύχεις πολύ λεπτομερές και ρεαλιστικά μοντέλα. Υπάρχουν διάφορα λογισμικά που περιέχουν εργαλεία για ψηφιακή γλυπτική. Η 3D γλυπτική θυμίζει πολύ την πραγματική γλυπτική μιας και χρησιμοποιεί βούρτσες (brushes) με τις οποίες μπορείς να “σκάψεις” και να “τραβήξεις” λεπτομέρειες πάνω στο μοντέλο. Οι βούρτσες εισάγονται στο κατάλληλο λογισμικό σε μορφή εικόνας με διαφανές φόντο καθώς οι λεπτομέρειες καθορίζονται από αποχρώσεις του γκρι.



Παράδειγμα 12 Sculpt Brush - Alpha Texture

https://www.textures.com/download/3dbrush0100/133294

Παράδειγμα 13 Sculpted Sphere

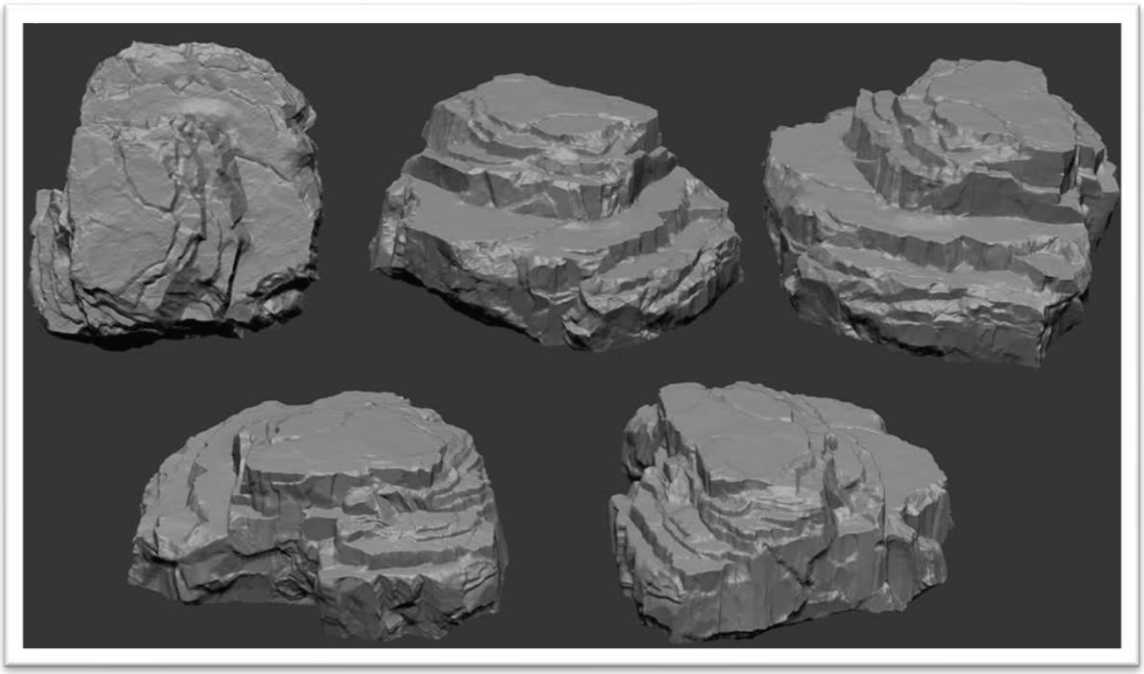
https://www.textures.com/download/3dbrush0100/133294

Με τον τρόπο που λειτουργεί ο συγκεκριμένος τύπος μοντελοποίησης το πλέγμα πρέπει να έχει αρκετά πολύγωνα, έτσι ώστε τα εργαλεία γλυπτικής να μπορούν να πάρουν αυτά πολύγωνα και να τα τροποποιήσουν ώστε να πετύχει ο δημιουργός αυτό που θέλει.

Ωστόσο, όσο αποτελεσματική και να είναι η ψηφιακή γλυπτική, έχει και κάποια μειονεκτήματα. Το βασικότερο είναι ότι προσθέτει πολύ πληροφορία (πολύγωνα) στο μοντέλο έτσι ώστε να πετύχει την λεπτομέρεια που είναι επιθυμητή, αυτό λύνεται με μια διαδικασία που λέγεται Normals Baking, την οποία θα περιγράψουμε σε επόμενο κεφάλαιο. Ένα επίσης αρνητικό της ψηφιακής γλυπτικής είναι ότι καθώς προσθέτει πληροφορία (πολύγωνα) στο πλέγμα, δεν φροντίζει να ακολουθεί τους κανόνες τοπολογίας, προσθέτει δηλαδή πολύγωνα με περισσότερα από 4 σημεία, καθώς τοποθετεί σημεία (vertices) και άκρα (edges) στην ίδια θέση.

Όλα τα σύγχρονα λογισμικά περιέχουν μια συγκεκριμένη λειτουργία που ονομάζεται **“Διαγραφή διπλών” (Remove doubles)**, η οποία παίρνει τις θέσει όλων των σημείων και διαγράφει το ένα από τα δύο σημεία που βρίσκονται στην ίδια θέση. Όσο αφορά την κακή τοπολογία που δημιουργείται με την ψηφιακή γλυπτική, ο χρήστης πρέπει να “καθαρίσει” το πλέγμα με το χέρι. Κάτι που κάνει πολλούς καλλιτέχνες να σκέφτονται καλά πριν ξεκινήσουν να δίνουν λεπτομέρεια στο μοντέλο τους με αυτόν τον τρόπο, αλλά είναι και κάτι που νέοι καλλιτέχνες δυσκολεύονται να καταφέρουν καθώς απαιτεί αρκετή εμπειρία.

Ωστόσο, καθώς τα λογισμικά γίνονται όλο και καλύτερα με τον χρόνο, στα πιο σύγχρονα, έχει αναπτυχθεί μια λειτουργία πάνω στην 3D γλυπτική η οποία λέγεται Dynamic Topology (Δυναμική τοπολογία) επιτρέπει στον χρήστη να χρησιμοποιεί τα εργαλεία της 3D γλυπτικής αλλά με το λογισμικό να φροντίζει τα πολύγωνα να είναι είτε tris (τριών σημείων) είτε quads (τεσσάρων σημείων). Αυτό έχει λύσει τα χέρια σε πολλούς δημιουργούς και έχει κάνει την γλυπτική περισσότερο δημοφιλή.



Παράδειγμα 14 Sculpted Rocks

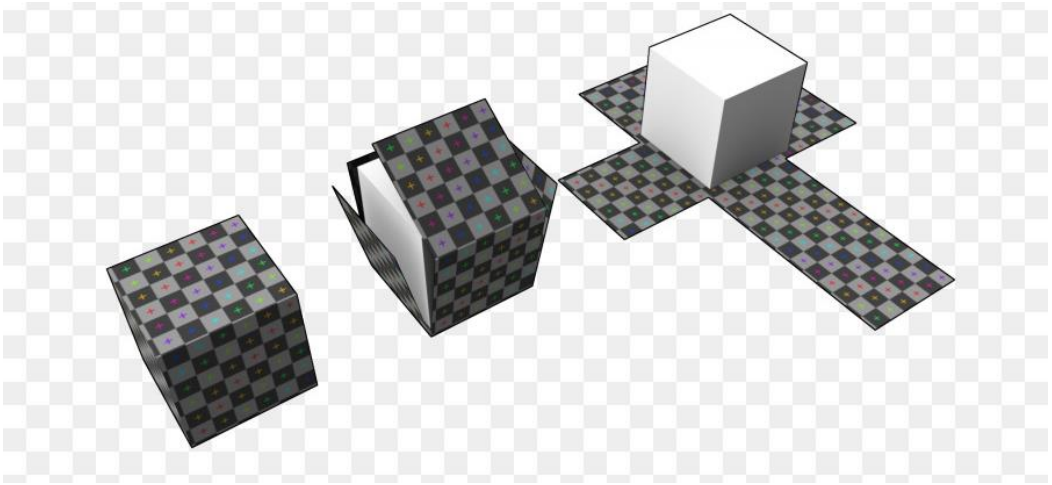
https://polycount.com/discussion/83605/frustrated-and-jealous-rock-sculpting-help

3 UV Mapping

3.1 Τα βασικά του UV Mapping

UV Map (Χάρτης UV) είναι η δυσδιάστατη απεικόνιση της επιφάνειας ενός τρισδιάστατου μοντέλου. Η διαδικασία που ακολουθείται για την δημιουργία ενός UV Map λέγεται UV unwrapping. Τα UVs αναπαριστούν τους άξονες του δυσδιάστατου χώρου, οριζόντια και κάθετα. Αυτή η διαδικασία γίνεται αφότου τελειώσει η μοντελοποίηση του αντικειμένου. Ο χάρτης UV είναι απαραίτητο να δημιουργηθεί έτσι ώστε να μπορέσει το λογισμικό να πάρει τις κατάλληλες πληροφορίες κατά την διάρκεια του Texturing, που θα δούμε αργότερα. Ένας χάρτης UV περιέχει τα πολύγωνα του αντικειμένου που έγινε unwrapped τοποθετημένα σε μια επίπεδη επιφάνεια.

Η διαδικασία αυτή πρέπει να θεωρηθεί ως "ξεδίπλωμα". Έστω ότι το αντικείμενο είναι ένας κύβος από χαρτί, κόβονται οι άκρες του κύβου που είναι απαραίτητες και έτσι ξεδιπλώνεται το χαρτί στην επίπεδη επιφάνεια. Όσο πιο περίπλοκα γίνονται τα αντικείμενα, τόσο πιο δύσκολη γίνεται και η διαδικασία UV Mapping. Το UV Mapping γίνεται επίσης στο λογισμικό μοντελοποίησης.



Παράδειγμα 15 UV Mapping

https://www.cleanpng.com/png-uv-mapping-texture-mapping-cube-3d-modeling-cube-1181894/

3.2 Τα συστατικά του UV Mapping

Για την διαδικασία αυτή υπάρχουν πολλά εργαλεία και το κάθε ένα χρησιμοποιείται για άλλος σκοπό. Υπάρχουν εργαλεία που παράγουν ένα ολοκληρωμένο UV Map από κάποιο μοντέλο, ωστόσο δεν είναι χρήσιμο για περίπλοκα μοντέλα καθώς δεν είναι καθόλου ακριβής καθιστώντας το τελικό αποτέλεσμα άτοπο. Αυτό είναι ένα πρόβλημα, πόσο μάλλον αν το αντικείμενο αυτό είναι ένα βασικό στοιχείο (focused element) της τελικής εικόνας ή βίντεο ή ό,τι παράξει ο καλλιτέχνης.

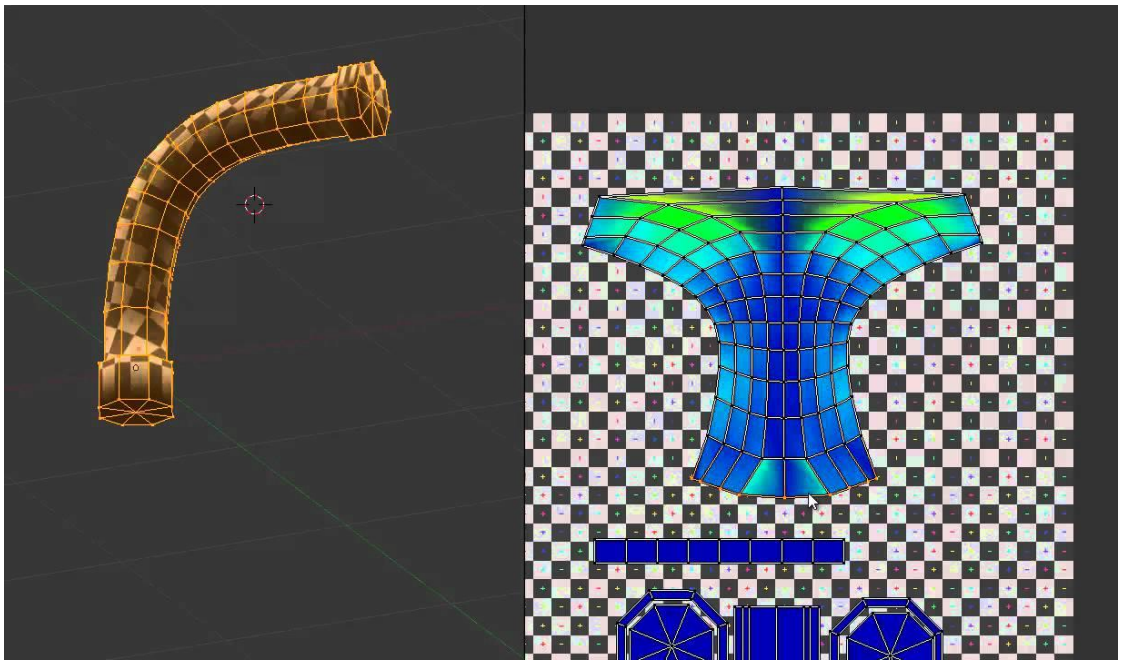
Για να παραχθεί ένα καλός χάρτης UV η καλύτερη επιλογή είναι να γίνει χειροκίνητα από τον δημιουργό. Αυτό γίνεται με την χρήση των seams. Ο δημιουργός επιλέγει τα άκρα τα οποία θέλει να κοπούν ώστε γίνει το unwrap και τις ορίζει ως seams. Το λογισμικό κατά την διαδικασία του unwrap διαβάζει τα δεδομένα του αντικειμένου και ψάχνει για αυτά τα seams ώστε να κάνει τους κατάλληλου υπολογισμούς έτσι ώστε να παραχθεί ένας καλός χάρτης UV. Αν και είναι μια διαδικασία που μπορεί να φανεί περίπλοκη στην αρχή, με την κατάλληλη εμπειρία ένας δημιουργός μπορεί να την κάνει αρκετά γρήγορα.

3.3 Ένα καλό UV Map

Όταν κάποιος αρχίσει να μπαίνει στην διαδικασία του unwrap, κατά πάσα πιθανότατα θα φτάσει στο σημείο που δεν θα ξέρει αν έχει τελειώσει και εάν το αποτέλεσμα είναι καλό. Αυτό το φαινόμενο εμφανίζεται συνήθως σε μοντέλα περίπλοκα, με πολλά πολύγωνα και πολλές γωνίες.

Ένα εργαλείο που έχουν τα περισσότερα λογισμικά λέγεται UV Stretch. Χρωματίζει κατάλληλα τα πολύγωνα που έχουν γίνει unwrapped ώστε να καταλάβει ο δημιουργός που υπάρχουν ατέλειες όπως "τέντωμα" και παραμορφώσεις του πλέγματος. Είναι ένα πολύ χρήσιμο εργαλείο καθώς σε βοηθάει να καταλάβεις πότε έχεις τελειώσει αλλά και το πόσο καλό είναι το αποτέλεσμα.

Καλό είναι ο δημιουργός να περιορίζει τα δεδομένα του UV χάρτη στον χώρο 1:1 για να μην υπάρξουν παραμορφώσεις κατά την διάρκεια του texturing. Αυτό ωστόσο είναι ένα πρόβλημα που έχει λυθεί από τις πιο σύγχρονες εκδόσεις των περισσότερων λογισμικών. Παρόλα αυτά είναι καλό να συνεχίσει να τηρείται αυτός ο κανόνας ώστε τα αντικείμενα να μπορούν να εισαχθούν σε όλα τα λογισμικά.



Παράδειγμα 16 How a good UV Map looks Like

https://www.youtube.com/watch?v=8VgLlurusWc

4 Texture Maps

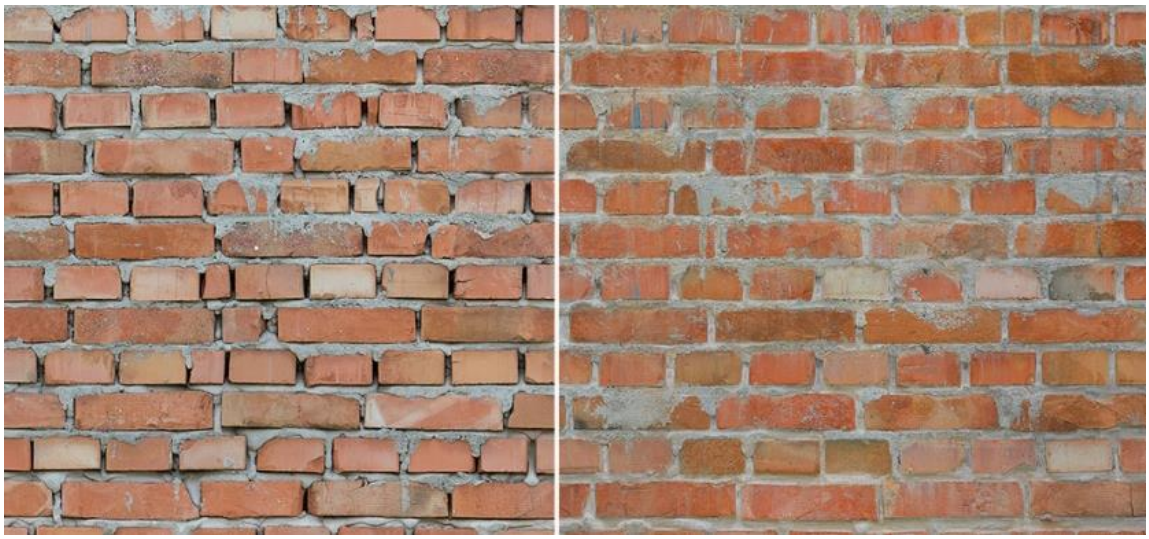
4.1 Τι είναι ένα Texture

Texture είναι μια εικόνα που χρησιμοποιείται για να δώσει στο αντικείμενο μια πιο ρεαλιστική επιφάνεια καθώς του προσθέτει χρώμα αλλά και δεδομένα που το βοηθούν να αντιδράει καλύτερα στον φωτισμό και στο περιβάλλον του αντικειμένου. Texture mapping είναι η διαδικασία κατά την οποία ένας καλλιτέχνης ενσωματώνει τα textures σε ένα 3D μοντέλο. Υπάρχουν πολλά είδη texture και το κάθε ένα δίνει δεδομένα στο λογισμικό με άλλο τρόπο και για άλλο λόγο. Να σημειωθεί ότι κάθε λογισμικό χρειάζεται και διαφορετικά είδη textures για να κάνει τους υπολογισμούς και να παράξει ένα ρεαλιστικό αποτέλεσμα.

4.2 Οι διάφοροι τύποι Texture

Όπως ειπώθηκε και προηγουμένως, τα είδη texture που υπάρχουν είναι πολλά και το κάθε ένα έχει άλλο σκοπό. Παρακάτω θα γίνει μια περιγραφή των κυριότερων ειδών texture που χρησιμοποιούνται από τα περισσότερα λογισμικά.

- **Albedo Map ή Diffuse Map**: Είναι το texture που περιέχει τα δεδομένα για το χρώμα του αντικειμένου. Η διαφορά του diffuse map με του albedo map είναι το albedo δεν περιέχει σκιές από φωτισμό καθώς χρησιμοποιείται περισσότερο σε PBR Workflow που θα περιγράφει σε επόμενο κεφάλαιο.



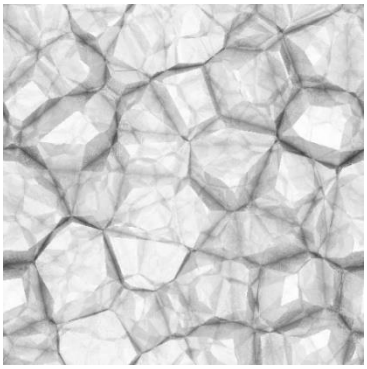
Παράδειγμα 17 Diffuse vs Albedo Map

https://blendermarket.com/products/friendly-shade-bundle-01

- **Ambient Occlusion (AO)**: Ένα AO Texture χρησιμοποιείται για να δώσει στο αντικείμενο έξτρα πληροφορία για το που θα έπρεπε να υπάρχουν σκιές και συνήθως συνδυάζεται με τον albedo map για να παράξουν ένα πιο ρεαλιστικό αποτέλεσμα. Το

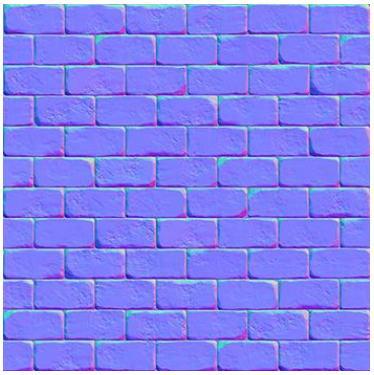
άσπρο χρώμα δείχνει που δεν θα υπάρχουν σκιές και οι αποχρώσεις του γκρι δείχνουν που θα υπάρχει και πόσο σκοτεινή θα είναι ανάλογα πόσο σκούρα είναι η απόχρωση.

- **Normal Map**: Ένα normal map καθορίζει χρησιμοποιείται για να καθορίσεις τις μικρές ατέλειες μια επιφάνειες όπως "βαθουλώματα" ή "εξογκώματα". Όταν ενσωματωθεί στο αντικείμενο το πάει σε άλλο επίπεδο καθώς αυτές οι ατέλειες το κάνουν να δείχνει πολύ πιο ρεαλιστικό. Η μορφή του είναι περίεργη για το μάτι αλλά τέλεια για το λογισμικό. Απαρτίζεται από αποχρώσεις του μωβ όπου κάθε απόχρωση αντιπροσωπεύει και έναν άξονα. Έτσι το λογισμικό ξέρει ακριβώς πως πρέπει να δείχνουν οι ατέλειες πάνω στο μοντέλο.



Παράδειγμα 18 Ambient Occlusion Map

https://www.filterforge.com/filters/10500-ambientocclusion.html



Παράδειγμα 19 Normal Map

https://learnopengl.com/Advanced-Lighting/Normal-Mapping

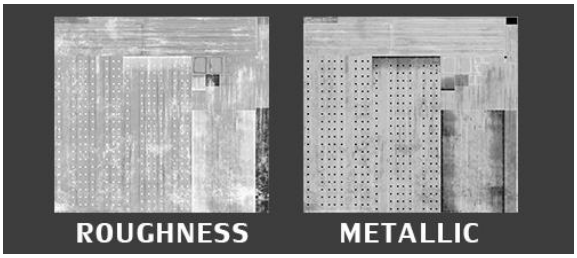
- **Height Map ή Bump Map**: Έχει παρόμοια χρήση με το normal map αλλά δίνει πληροφορία μόνο για το ύψος των λεπτομερειών και όχι για διάφορες γωνίες.

Παράδειγμα 20 Height ή Bump Map

https://kcclemo.neocities.org/creating-height-and-normal-maps/

- **Roughness Map**: Καθορίζει πόσο διεσπαρμένο θα αναπηδήσει το φως από την επιφάνεια. Όσο πιο "τραχιά" είναι η επιφάνεια (άσπρο χρώμα) τόσο ευρύτερη θα είναι η λάμψη.

- **Metallic Map**: Καθορίζει το πόσο μεγάλη θα είναι η αντανάκλαση. Όσο πιο ανοιχτό είναι το χρώμα τόσο μεγαλύτερη η αντανάκλαση.
- **Smoothness Map**: Είναι το αντίθετο του Roughness Map αφού το άσπρο καθορίζει πόσο λεία θα είναι η επιφάνεια.
- **Mask Map**: Χρησιμοποιείται μόνο από την μηχανή παιχνιδιών Unity 3D και μόνο για τις νέες εκδόσεις που περιέχουν την High-Definition Render Pipeline. Θα χρησιμοποιηθεί πολύ στο πρακτικό κομμάτι της παρούσας πτυχιακής εργασίας. Ουσιαστικά είναι η μίξη τεσσάρων διαφορετικών texture maps ένα για κάθε κανάλι των RGBA. Στο κόκκινο κανάλι μπαίνει το Metallic Map, στο πράσινο κανάλι μπαίνει ο Ambient Occlusion Map, στο μπλε κανάλι μπαίνει ο Detail Map (Height Map) ενώ στο Alpha κανάλι μπαίνει ο Smoothness Map. Αυτό γίνεται είτε με κάποιο Add-on του Unity είτε σε κάποιο πρόγραμμα επεξεργασίας εικόνας όπως Photoshop είτε παράγεται από λογισμικό όπως το Substance Painter.



Παράδειγμα 21 Roughness, Metallic Maps

https://blog.backspinestudios.com/from-specular-to-metallic/

Παράδειγμα 22 Smoothness, Specular Map

https://blog.backspinestudios.com/from-specular-to-metallic/

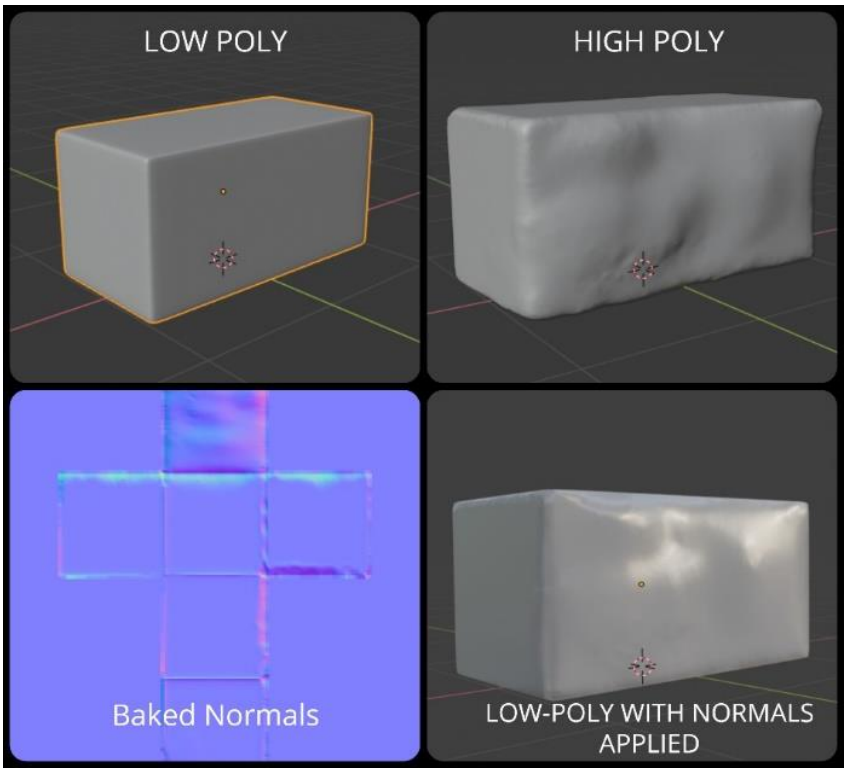
- **Environment Texture**: Είναι ένα αρχείο εικόνας που χρησιμοποιείται για να δώσει στην 3D σκηνή ένα φόντο το οποίο χρησιμεύει και σαν πηγή φωτός αλλά και για διάφορες αντανακλάσεις ορισμένων αντικειμένων.

Παράδειγμα 23 Environment Texture

https://texturify.com/category/environment-panoramas.html

4.3 Normals Baking

Baking είναι μια πολύ δημοφιλής διαδικασία στην βιομηχανία παιχνιδιών και ταινιών. Αυτό γιατί επιτρέπει την μεταφορά μεγάλης λεπτομέρειας από ένα high-poly mesh (πλέγμα πολλών πολυγώνων) σε ένα Normal Map ώστε να ενσωματωθεί σε ένα low-poly mesh (πλέγμα λίγων πολυγώνων). Μπορεί δηλαδή να παραχθεί μοντέλο με πολύ λεπτομέρεια χωρίς να κοστίζει απολύτως τίποτα σε πόρους κατά την διάρκεια του τρεξίματος. Αυτή η διαδικασία γίνεται συνήθως έως πάντα μετά την μοντελοποίηση με 3D γλυπτική (Sculpting). Τα δεδομένα από το αποτέλεσμα αυτής της διαδικασίας αποθηκεύονται σε μία εικόνα (ένα Normal Map).



Παράδειγμα 24 Normals Baking

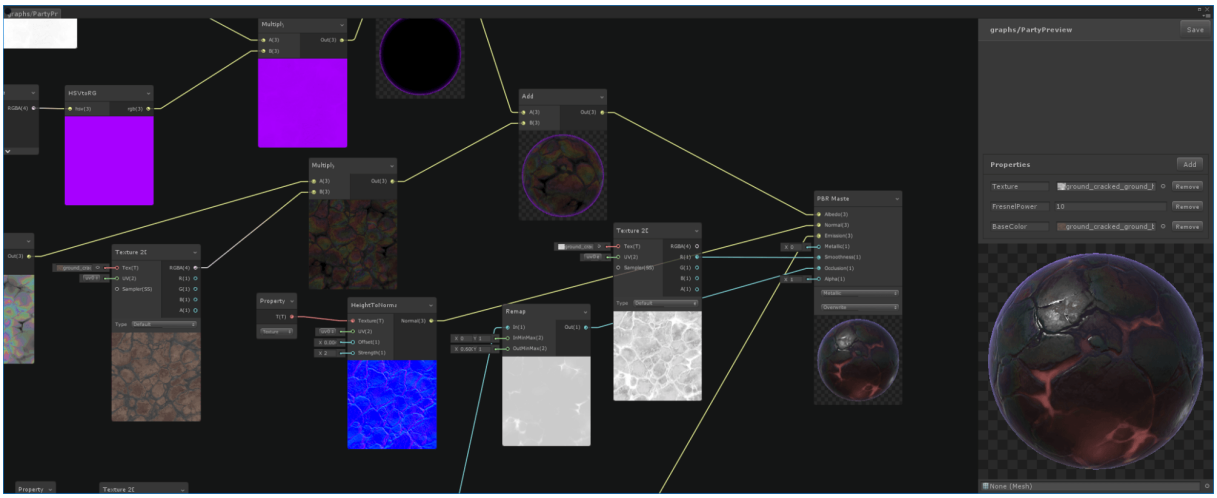
5 Materials & Shaders

5.1 Shaders

Shader είναι ένας αλγόριθμος που καθορίζει τις ιδιότητες ενός Material. Καθορίζει δηλαδή το πώς θα υπολογίζονται οι σκιές, το φως, το χρώμα και πολλά άλλα σε ένα μοντέλο που θα έχει ενσωματωμένο το συγκεκριμένο material. Οι Shaders δεν αφορούν μόνο το οπτικό αποτέλεσμα ενός 3D μοντέλου αλλά χρησιμοποιείται πολύ και για την αναπαράσταση ειδικών εφέ καθώς και στην διαδικασία post-processing. Υπάρχουν πολλοί τύποι shaders και ο κάθε ένας χρησιμοποιεί και διαφορετικά δεδομένα για να κάνει τους υπολογισμούς και να πετύχει το αποτέλεσμα που θέλει.

5.1.1 Shader Graph

Επειδή ο προγραμματισμός ενός Shader μπορεί να γίνει αρκετά περίπλοκος αν χρησιμοποιήσει κανείς μια γλώσσα προγραμματισμού αλλά και επειδή με το να δοθεί η δυνατότητα σε δημιουργούς να δημιουργήσουν τους δικούς τους shaders μπορεί να επιτευχθούν τρομερά αποτελέσματα. Οι εταιρίες που δημιουργούν 3D λογισμικά έχουν δημιουργήσεις τους επονομαζόμενους Shader Graphs. Αυτοί δίνουν την δυνατότητα στον δημιουργό να φτιάξει τους δικούς του shader αλλά σε ένα οπτικό περιβάλλον με την χρήση nodes, είναι μια πολύ δημοφιλής διαδικασία και έχει πάει το επίπεδο των γραφικών παιχνιδιών και ταινιών σε άλλο επίπεδο.

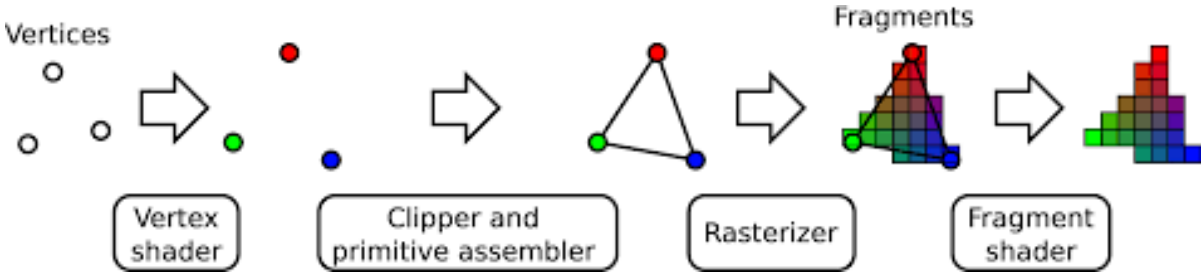


Παράδειγμα 25 Shader Graph

https://blogs.unity3d.com/2018/02/27/introduction-to-shader-graph-build-your-shaders-with-a-visual-editor/

5.1.2 Vertex Shader

Vertex shader είναι ο πιο συνηθισμένος Shader και τρέχει όλες τις εντολές μια φορά για κάθε vertex που φαίνεται στην οθόνη. Ένας vertex shader μπορεί να χειριστεί πολλές ιδιότητες ενός vertex όπως θέση στο 3D χώρο, συντεταγμένες πάνω στο texture, χρώμα κλπ..

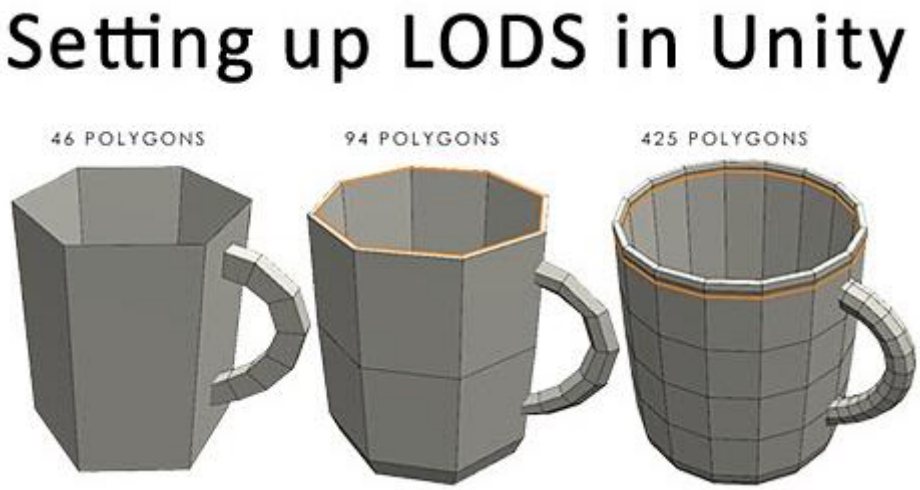


Παράδειγμα 26 How vertex shader works

https://www.google.com/search?biw=1366&bih=635&tbm=isch&sa=1&ei=XqVuXZOBMJ-HjLsPwYKBiA4&q=vertex+shader&oq=verte&gs_l=img.3.0.35i39j0i67l2j0j0i67j0j0i67j0l3.17746.18505..19033...0.0..0.143.672.0j5.....0....1..gws-wiz-img.MWguVYRw9go#imgrc=Bw0U2kDlwL-SjM:

5.1.3 Tessellation Shader & LOD System

Tessellation Shader είναι ένα νέο είδος shader που δίνει την δυνατότητα να αλλαχτεί η θέση των vertices και να τραβηχτούν προς την κατεύθυνση των normal τους. Αυτό επιτρέπει στο πλέγμα να πάρει περισσότερη λεπτομέρεια και συνεπώς να είναι πιο ρεαλιστικό. Επίσης επιτρέπει σε απλά πλέγματα να αλλάζουν ανάλυση at runtime. Αυτό γίνεται υπολογίζοντας την απόσταση του αντικειμένου από την κάμερα του λογισμικού. Όσο πιο μακριά είναι το αντικείμενο από την κάμερα τόσο μικρότερη ανάλυση θα έχει, ενώ όσο πιο κοντά είναι τόσο πιο μεγάλη ανάλυση θα αποκτήσει. Αυτό παλαιότερα γινόταν χειροκίνητα με την τεχνική LOD (level-of-design), όπου ο δημιουργός έπρεπε να μοντελοποιήσει διάφορες εκδοχές του αντικειμένου από πολύ μικρή ανάλυση μέχρι πολύ μεγάλη ανάλυση έτσι ώστε να αλλάζει το αντικείμενο ανάλογα με την απόσταση του από την κάμερα. Αυτή η τεχνική ωστόσο χρησιμοποιείται ακόμα για μοντέλα με πολύ λεπτομέρεια καθώς δεν γίνεται ακόμα να επεξεργαστούν σωστά at runtime.



Παράδειγμα 27 LOD System in Unity

https://www.youtube.com/watch?v=IzlU_xvTK3Y

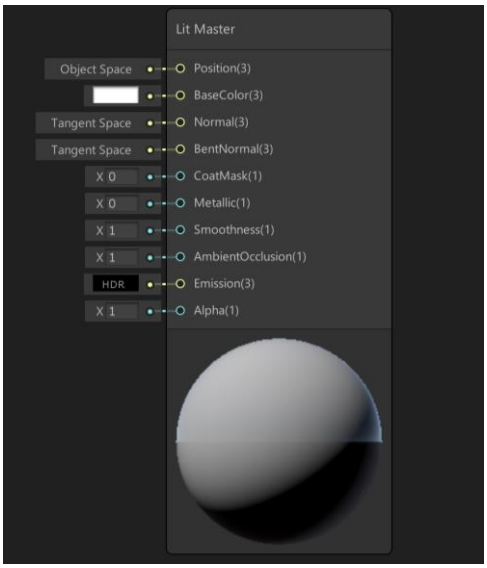
5.2 Materials

Materials είναι μια τεχνολογία που χρησιμοποιείται από τους καλλιτέχνες για να δώσουν σε ένα 3D μοντέλο τις ιδιότητες ενός υλικού του πραγματικού κόσμου, όπως ξύλο, μέταλλο, νερό κλπ. Όταν ένα μοντέλο παίρνει τις ιδιότητες ενός material, αλλάζει χρώμα, αντιδράει αλλιώς στο φως και αλλάζει τελείως όψη. Αυτό γίνεται για να γίνει το αντικείμενο πιο ρεαλιστικό και συνεπώς να χρησιμοποιηθεί για ότι θέλει ο δημιουργός του, όπως κάποιο παιχνίδι ή μια ταινία. Κάθε material για να λειτουργήσει σωστά πρέπει να βασίζεται σε ένα shader ή σε μια μίξη παραπάνω από ενός shader.

Ένα material περιέχει τους διάφορους παραμέτρους ενός shader που πρέπει να αλλαχτούν για να επιτευχθεί το αποτέλεσμα που θέλει ο χρήσης. Παραδείγματα αυτών των παραμέτρων είναι τα αρχεία εικόνα του κάθε texture map όπως albedo, mask map, normal map. Αλλά και άλλους παραμέτρους που διαφέρουν ανάλογα το Shader.

Παράδειγμα 28 Materials applied on Spheres

http://www.reynantemartinez.com/cycles-material-studies.html



Παράδειγμα 29 Unity's Lit Shader parameters

https://blogs.unity3d.com/2018/12/19/unity-2018-3-shader-graph-update-lit-master-node/

6 Lighting

6.1 Φωτισμός στα γραφικά υπολογιστών

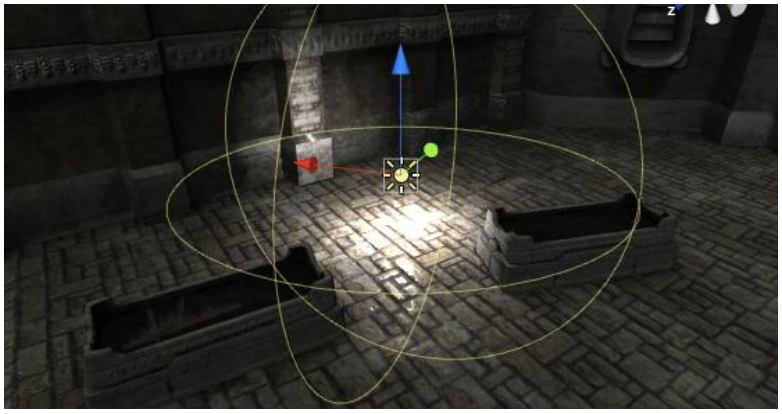
Lighting είναι η προσομοίωση του φωτός στα γραφικά υπολογιστών. Το λογισμικό χρησιμοποιεί ειδικούς αλγορίθμους για να καταφέρει αυτήν την προσομοίωση και αυτοί οι αλγόριθμοι διαφέρουν από λογισμικό σε λογισμικό. Υπάρχουν προσομοιώσεις που χρησιμοποιούν πολύ ακριβείς τεχνικές που συνδέονται άμεσα με το πώς παρουσιάζεται το φως στον πραγματικό κόσμο, αυτός ο τρόπος φωτισμού ωστόσο είναι αρκετά χρονοβόρος αλλά και πολύ ακριβός σε πόρους, για αυτό το λόγο δεν χρησιμοποιείται σε ταινίες και παιχνίδια. Υπάρχουν ωστόσο και άλλες προσομοιώσεις που με διαφορετικούς μεθόδους, δημιουργούν την ψευδαίσθηση του φωτός και κάνουν το φωτισμό πιο φθηνό σε πόρους και συνεπώς κατάλληλο για παιχνίδια, όπου απαιτείται ο υπολογισμός του φωτισμού σε πραγματικό χρόνο. Αν και ο τρόπος υλοποίησης των δύο τεχνικών είναι διαφορετικός, το αποτέλεσμα είναι εξίσου αποτελεσματικό και ρεαλιστικό.

6.2 Πηγές φωτός

Για να αρχίσει η διαδικασία προσομοίωσης του φωτός από ένα λογισμικό πρέπει να υπάρχει στην σκηνή μια πηγή φωτός, υπάρχουν διάφορα είδη πηγών και το κάθε ένα έχει και τις δικές του ιδιότητες. Για να επιτευχθεί μια ρεαλιστική προσομοίωση φωτός σε μία σκηνή, συχνά είναι απαραίτητο να χρησιμοποιηθούν πολλά είδη πηγών φωτός μαζί, καθώς και στον πραγματικό κόσμο, οι πηγές φωτός είναι πολλές. Η διαδικασία φωτισμού με πηγές φωτός λέγε direct lighting (άμεσος φωτισμός).

Τα βασικότερα είδη πηγών είναι τα εξής :

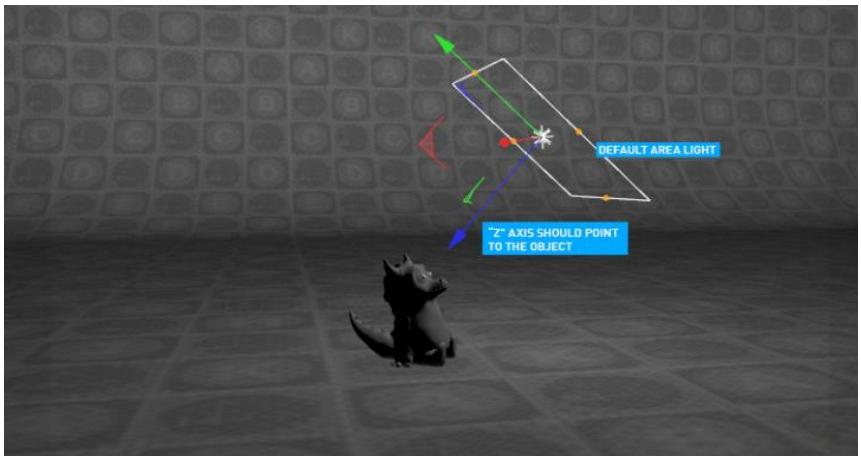
- Point Light: Είναι μια πηγή της οποίας οι ακτίνες φωτός διασκορπίζονται προς όλες τις κατευθύνσεις. Τα αντικείμενο που είναι κοντά στην πηγή φωτίζονται περισσότερο ενώ αυτά που είναι πιο μακριά, λιγότερο έως και καθόλου. Χρησιμοποιείται συνήθως για να προσομοιώσει το φως μιας λάμπας, ενός κεριού κλπ. Οι παράμετροι που συνήθως αλλάζουν οι δημιουργοί μιας σκηνής είναι η ένταση του φωτός, το χρώμα αλλά και το εύρος. (Unity Documentation, n.d.)



Παράδειγμα 30 Point Light

https://docs.unity3d.com/Manual/Lighting.html

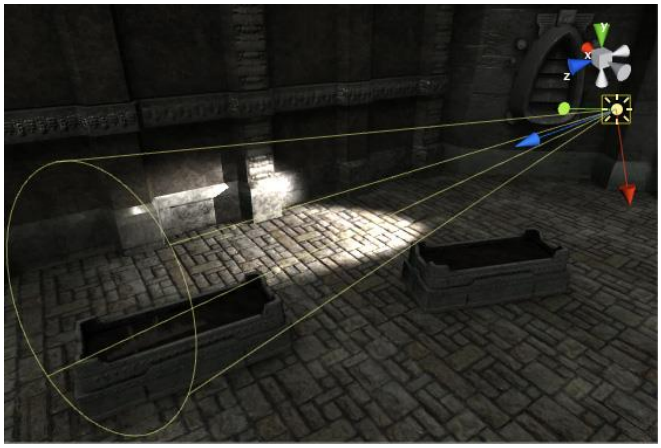
- Area Light: Στον πραγματικό κόσμο, το φως πολλές φορές παράγεται από επιφάνειες με διάφορα σχήματα και τα αποτελέσματα διαφέρουν ανάλογα την επιφάνεια. Τέτοιες επιφάνειες μπορεί να είναι ο ουρανός, η τηλεόραση, μια LED πινακίδα κλπ.. Στα γραφικά υπολογιστών για να πετύχεις μια έγκυρη προσομοίωση φωτός τέτοιων επιφανειών, απαιτείται η χρήση ενός Area Light. Αυτό το είδος φωτός έχει κάποιο σχήμα, όπως τετράγωνο, ορθογώνιο, έλλειψη κλπ. και στέλνει κάθετες ακτίνες προς μια κατεύθυνση. Οι βασικοί παράμετροι ενός area light είναι, το σχήμα, η ένταση, το εύρος και το χρώμα. (Unity Documentation, n.d.)



Παράδειγμα 31 Area Light

http://www.aoktar.com/octane/OCTANE%20HELP%20MANUAL.html?MakingSpotLight.html

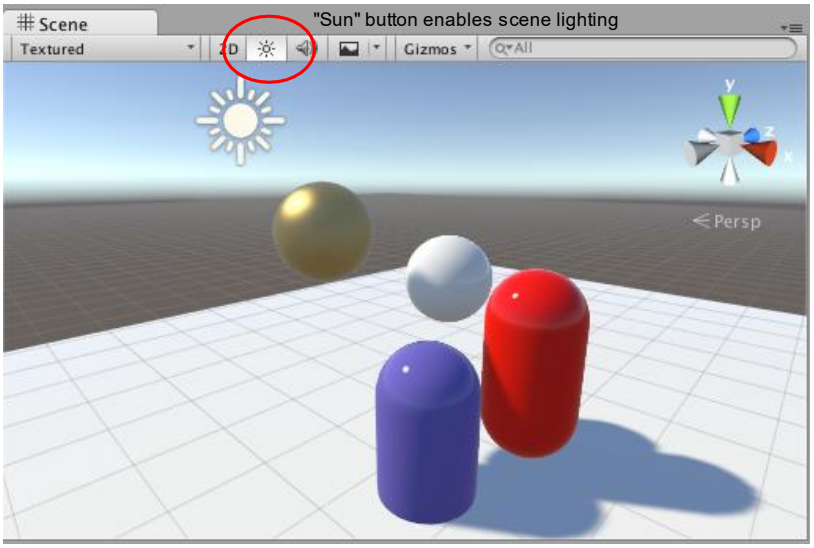
- Spot Light: Το spot light εκπέμπει ακτίνες σε σχήμα κώνου. Εστιάζει προς μια κατεύθυνση και ένα σημείο που βρίσκεται στο κέντρο του κώνου. Χρησιμοποιείται για να προσομοιώσει την παραγωγή φωτός ενός φακού, μια λάμπας δρόμου, των φώτων ενός αυτοκινήτου κλπ. Οι παράμετροι του είναι επίσης το χρώμα, η ένταση και το εύρος αλλά και το μέγεθος του κώνου. (Unity Documentation, n.d.)



Παράδειγμα 32 Spot Light

https://docs.unity3d.com/Manual/Lighting.html

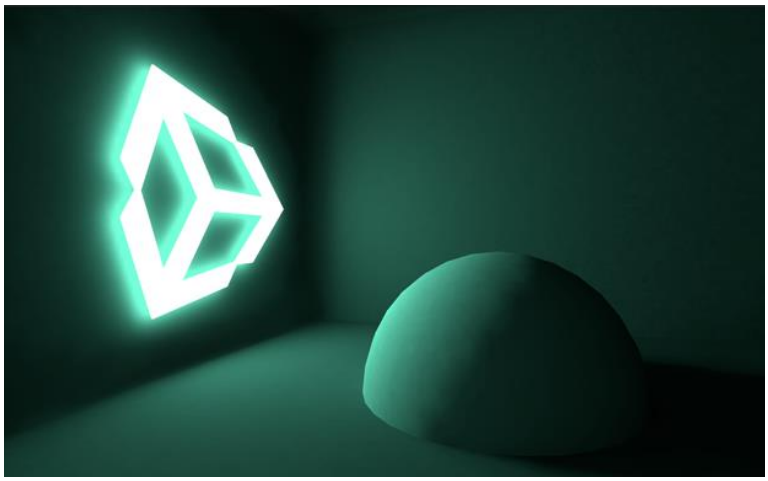
- Directional Light: Παράγει φως προς μια κατεύθυνση χωρίς να παίζει ρόλο η θέση του στον χώρο. Όλα τα σημεία της σκηνής λαμβάνουν την ίδια ένταση φωτός. Οι παράμετροι του είναι η περιστροφή του, η ένταση, το χρώμα κλπ., όχι όμως το εύρος. Συνηθίζεται να χρησιμοποιείται για να προσομοιώσει το φως του ήλιου ή του φεγγαριού. (Unity Documentation, n.d.)



Παράδειγμα 33 Directional Lighting

https://docs.unity3d.com/Manual/UsingLights.html

- Emissive Objects: Emissive Object είναι αντικείμενα των οποίων τα Material χρησιμοποιούν την emissive τεχνική ώστε να παράγουν φως. Το φως εκπέμπεται από κάθε πολύγωνο του αντικείμενο προς την κατεύθυνση των normals του. Οι παράμετροι είναι η ένταση και το χρώμα. (Unity Documentation, n.d.)



Παράδειγμα 34 Emissive Materials

https://unity3d.com/kr/learn/tutorials/topics/graphics/emissive-materials?playlist=48370

6.3 Environment Lighting

Environment Lighting είναι η διαδικασία φωτισμού μιας σκηνής (συνήθως εξωτερικό περιβάλλον) χρησιμοποιώντας ένα environment texture. Αυτή η διαδικασία παίρνει το επιλεγμένο texture και το βάζει σε μια σφαίρα που περικυκλώνει την σκηνή. Μετά διαβάζει τα δεδομένα του texture, όπως χρώμα, ένταση χρώματος, και τα χρησιμοποιεί για να φωτίσει την σκηνή. Επιπροσθέτως χρησιμοποιεί το texture για να υπολογίσει τις αντανακλάσει από διάφορες επιφάνειες της σκηνής, όπως νερό, μέταλλο κλπ.. Αυτό έχει ένα πολύ ρεαλιστικό αποτέλεσμα και χρησιμοποιείται πολύ συχνά στην παραγωγή παιχνιδιών καθώς είναι πολύ φθηνό σε πόρους.

Παράδειγμα 35 Environment Lighting

https://www.thegnomonworkshop.com/tutorials/environment-lighting-for-production

6.4 Volumetric Lighting

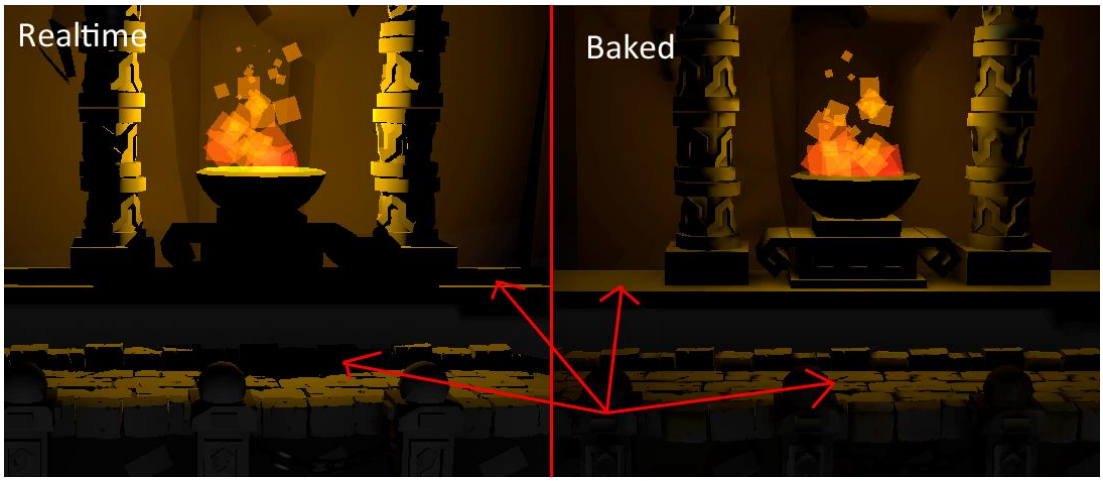
Volumetric Lighting είναι μια τεχνική στα γραφικά υπολογιστή που δημιουργεί το εφέ της σκόνης σε μια σκηνή. Χρησιμοποιείται πολύ συχνά στην παραγωγή παιχνιδιών καθώς κάνει την σκηνή πιο ρεαλιστική αφού φαίνεται πιο “ζωντανή”. Χρειάζεται ένα έξτρα αντικείμενο για να λειτουργήσει και αυτό είναι ένας κύβος που περικυκλώνει την σκήνη στα σημεία που υπάρχουν πηγές φωτός που θέλει ο δημιουργός να προσθέσει αυτό το εφέ. Αυτό το αντικείμενο λέγεται volume και περιέχει παραμέτρους όπως το πόσο πυκνή θα είναι η σκόνη, το χρώμα κλπ.

Παράδειγμα 36 Volumetric Lighting

https://github.com/SlightlyMad/VolumetricLights

6.5 Baked Lighting

Baked Lighting είναι μια λειτουργία που χρησιμοποιούν τα 3D λογισμικά για να αποθηκεύσουν τα δεδομένα του φωτισμού κατά την διάρκεια της ανάπτυξης της σκηνής ώστε να πετύχουν μεγαλύτερη λεπτομέρεια χωρίς να καθυστερούν και να "βαραίνουν" την σκηνή σε πραγματικό χρόνο. Με το baked lighting μπορεί να επιτευχθεί μεγάλη ανάλυση φωτισμού και σκίασης χωρίς να είναι ακριβό σε πόρους. Η πληροφορία αυτή αποθηκεύεται σε κάποια αρχεία εικόνας που λέγονται lightmaps. Ωστόσο έχει και τα μειονεκτήματα του καθώς με τη χρήση του, περιορίζει τα φώτα στο να μείνουν σταθερά σε πραγματικό χρόνο. Είναι μια τεχνολογία που χρησιμοποιείται κυρίως στα παιχνίδια για να κάνει την εκτέλεση τους πιο ομαλή.



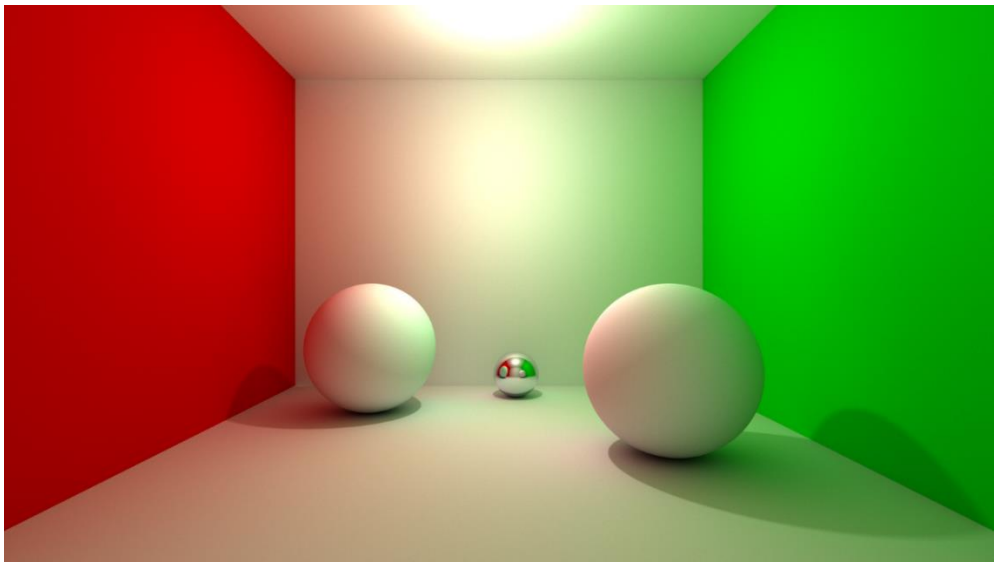
Παράδειγμα 37 Baked Lighting

https://forum.unity.com/threads/unity-5-baked-point-light.326668/

6.6 Global Illumination

Global Illumination είναι ένα σύστημα που καθορίζει το πώς το φως θα αναπηδήσει από μια επιφάνεια σε μια άλλη. Αυτό λέγεται έμμεσος φωτισμός και είναι κάτι που πολύ σημαντικό για να γίνει μια σκηνή που περιέχει μόνο άμεσο φωτισμό πιο ρεαλιστική. Μια σκηνή με έμμεσο φωτισμό είναι αυτόματα πιο ρεαλιστική καθώς στον πραγματικό κόσμο το φως αναπηδά από τις επιφάνειες των αντικειμένων που χτυπάει και καταλήγει σε άλλες επιφανείς και ούτω καθεξής. Το σύστημα αυτό αναλαμβάνει επίσης να κάνει και τις σκιές που υπάρχουν στην σκηνή πιο soft, δηλαδή να μην σταματάνε απότομα αλλά να σβήνουν σιγά-σιγά. Αυτό είναι επίσης κάτι πολύ σημαντικό για μια ρεαλιστική σκηνή.

Είναι μια τεχνική που για να γίνει in real-time είναι πολύ χρονοβόρα και πολύ ακριβή σε πόρους. Ωστόσο μιας και η υλοποίηση της σε παιχνίδια νέας γενιάς κρίθηκε αναγκαία, τα λογισμικά έχουν αναπτύξει μια τεχνική που λέγεται Baked Global Illumination (Baked GI). Αυτή η τεχνική διαβάζει τα αντικείμενα κατά την διάρκεια της ανάπτυξης της σκηνής και αποθηκεύει τα δεδομένα που χρειάζεται σε αρχεία εικόνα που λέγονται Lightmaps. Έτσι κάνει τις απαραίτητες ενέργειες με αυτά τα δεδομένα ώστε να γίνει η σκηνή πιο ρεαλιστική όταν τρέχει το παιχνίδι. Ένα μειονέκτημα του Baked GI είναι ότι περιορίζεται σε αντικείμενα της σκηνής που έχουν οριστεί ως στατικά, δηλαδή απαγορεύεται να κουνηθούν κατά την διάρκεια του παιχνιδιού. (Unity Documentation, n.d.)



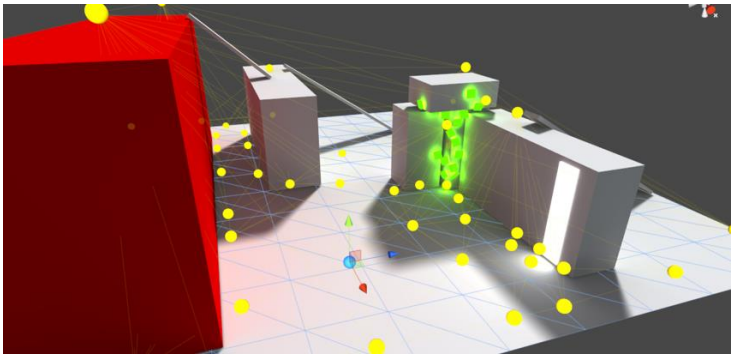
Παράδειγμα 38 Global Illumination

https://imgur.com/gallery/5iE3m

6.7 Light Probes

Light Probes είναι ένα σύστημα που επιτρέπει την αποθήκευση πληροφορίας που αφορά το φωτισμό που περνάει ανάμεσα από τον κενό χώρο μια σκηνής. Αποθηκεύει τα δεδομένα που χρειάζεται κατά την διάρκεια της ανάπτυξης του παιχνιδιού (Baked Data) και τα χρησιμοποιεί για να παράξει υψηλής ποιότητας φωτισμό σε αντικείμενα που κινούνται στο χώρο. Είναι δηλαδή η λύση στο μειονέκτημα του Baked Global Illumination, τον φωτισμό δηλαδή που αναπηδάει από επιφάνεια σε επιφάνεια που περιορίζεται σε στατικά αντικείμενα μόνο.

Ουσιαστικά είναι καλά τοποθετημένα σημεία στον 3D χώρο που αποθηκεύουν πληροφορία για τον φωτισμό γύρω από αυτά. Ένα παράδειγμα χρήσης είναι όταν ένας χαρακτήρας με άσπρα ρούχα να περνάει δίπλα από ένα μεγάλο μπλε τοίχο και το σύστημα αυτό καταφέρνει να φωτίσει τον χαρακτήρα που κινείται και να του δώσει την απόχρωση του χρώματος του τοίχου. (Unity Documentation, n.d.)



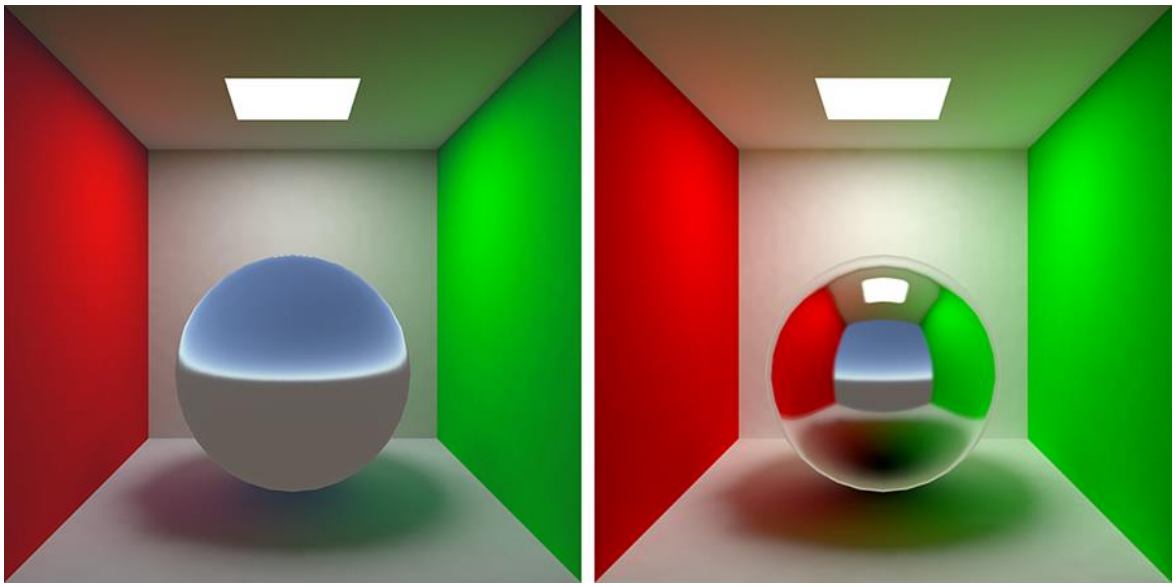
Παράδειγμα 39 Light Probes

https://docs.unity3d.com/540/Documentation/Manual/LightProbes.html

6.8 Reflection Probes

Reflection Probe είναι ένα αντικείμενο που τοποθετείται στην σκηνή και λαμβάνει δεδομένα για τα αντικείμενα που είναι γύρω του. Αυτά τα δεδομένα τα αποθηκεύει σε ένα cube map για να χρησιμοποιηθεί από αντικείμενα που χρησιμοποιούν αντανακλάσεις ώστε αυτές να είναι πιο έγκυρες. Ένα παράδειγμα μπορεί να είναι ότι σε ένα δωμάτιο υπάρχει ένας καθρέπτης και πίσω από αυτόν ένα κρεβάτι, το reflection probe θα αποθηκεύσει στο cube map τα αντικείμενα του δωματίου (το κρεβάτι) και έτσι από τον καθρέπτη θα είναι ορατό το κρεβάτι.

Οι εργοστασιακές αντανακλάσεις ενός αντικειμένου σχεδόν σε όλα τα λογισμικά προέρχονται είτε ένα απλό χρώμα είτε υπολογίζονται από το environment texture. Ένα reflection probe έχει ένα πεδίο ορισμού μέσα στο οποίο διαβάζει δεδομένα και αυτό πρέπει να τοποθετηθεί σωστά ώστε να είναι έγκυρες οι αντανακλάσεις. Τα δεδομένα μπορούν να προϋπολογιστούν κατά την διάρκεια της ανάπτυξης του παιχνιδιού αλλά και να υπολογιστούν σε πραγματικό χρόνο. (Unity Documentation, n.d.)



Παράδειγμα 40 Reflection Probes

https://unity3d.com/ru/learn/tutorials/topics/graphics/reflections

7 Rendering

Rendering είναι η διαδικασία παραγωγής μιας 2D εικόνας ή βίντεο από μια 3D σκηνή. Μπορεί να το δει κανείς ως την φωτογράφιση μια εικόνας ή την λήψη ενός βίντεο όταν ολοκληρωθεί το σκηνικό. Υπάρχουν πολλοί μέθοδοι rendering και ο κάθε ένα χρησιμοποιείται για διαφορετικό σκοπό. Αυτή η διαδικασία μπορεί να διαρκέσει από ένα κλάσμα του δευτερολέπτου μέχρι και μέρες για την παραγωγή μια σκηνής. Ο χρόνος που χρειάζεται για να ολοκληρωθεί διαφέρει ανάλογα τον αλγόριθμο που χρησιμοποιεί, το μέγεθος της σκηνής, την ανάλυση των αντικειμένων αλλά και πολλούς άλλους παραμέτρους.

Αυτό που ξεχωρίζει μια καλή μέθοδο rendering είναι ο τρόπος που υλοποιεί κάποια βασικά χαρακτηριστικά. Αυτά τα χαρακτηριστικά είναι οι αντανακλάσεις των επιφανειών καθώς και η σκίαση των αντικειμένων, αλλά δεν χειρίζεται μόνο αυτά. Μια διαδικασία rendering πρέπει να λαμβάνει υπόψιν και το πως θα παρουσιαστεί το φως, την όψη που θα παρουσιαστούν τα αντικείμενα (προοπτική ή ορθογραφική) και ένα σωρό άλλες ρυθμίσεις που ορίζουν το πώς θα παραχθεί το τελικό αποτέλεσμα.

Η διαδικασία αυτή δεν αναλαμβάνει μόνο να παράγει μια εικόνα ή βίντεο που περιλαμβάνει φωτισμό και 3D μοντέλα αλλά και διάφορα ειδικά εφέ που δίνουν μια πιο ρεαλιστική εμφάνιση στο τελικό αποτέλεσμα. Τέτοια εφέ μπορεί να αφορούν την ένταση του φωτός, την απόχρωση των χρωμάτων, vignette, motion blur, depth of field αλλά και πολλά άλλα. (Unity Documentation, n.d.)

7.1 Τύποι Rendering

7.1.1 Realtime Rendering

Realtime Rendering είναι όταν η διαδικασία rendering γίνεται σε πραγματικό χρόνο. Κάτι που είναι απαραίτητο για εφαρμογές όπως παιχνίδια αφού υπάρχει διάδραση μεταξύ του χρήστη και των αντικειμένων της σκηνής. Στο real-time rendering η σκηνή παράγεται κάθε frame. Ένα frame μπορεί να γίνει από πολύ μικρό χρόνο μέχρι και μεγαλύτερο ανάλογα τον επεξεργαστή και την κάρτα γραφικών του υπολογιστή που εκτελείται η εφαρμογή. Συνήθως σε ένα δευτερόλεπτο εκτελούνται από 20 έως 120 frame περίπου. Ένα παιχνίδι πρέπει να τρέχει περίπου στα 60 frames ανά δευτερόλεπτο για να έχει μια ομαλή εμπειρία ο χρήστης. Ένας real-time renderer είναι το νέο Eevee που είναι μέρος της νέας έκδοσης του Blender 2.80.

7.1.2 Non-Realtime Rendering

Non-Realtime Rendering είναι η διαδικασία rendering που δεν γίνεται σε πραγματικό χρόνο, αλλά μέχρι να ολοκληρωθεί απαιτείται αρκετός χρόνος ανάλογα την σκηνή. Αυτό χρησιμοποιείται κυρίως σε ταινίες και εικόνες αρχιτεκτονικής όπου απαιτούνται πολύ υψηλά επίπεδα ρεαλισμού. Κάνει όλους του απαραίτητους υπολογισμούς και προσομοιώνει πολλά φυσικά φαινόμενα που συμβαίνουν στον πραγματικό κόσμο για πετύχει τον απαραίτητο ρεαλισμό.

Χρησιμοποιεί τεχνικές όπως ray tracing, path tracing, photon mapping και radiosity που έχουν αναπτυχθεί ειδικά για αυτό το επίπεδο ποιότητας. Λαμβάνει υπόψιν και άλλες αλληλεπιδράσεις της ύλης με το φως όπως volumetric lighting για σκόνη και ομίχλη, όπως subsurface scattering για αντικείμενα που απορροφούν φως όπως το δέρμα αλλά δίνει και

μεγάλη έμφαση στην απόδοση των particle systems που χρησιμοποιούνται για να υλοποιηθούν φυσικά φαινόμενα όπως βροχή, φωτιά, χιόνι, καπνός κλπ. Τα παραπάνω φυσικά και υλοποιούνται και στο rendering πραγματικού χρόνου ωστόσο εκεί χρησιμοποιούνται τεχνικές που δίνουν την ψευδαίσθηση αυτών των φαινομένων ενώ σε αυτήν την περίπτωση (non-real-time) το λογισμικό χρησιμοποιεί έγκυρες διαδικασίες για να παράξει ένα ακριβές αποτέλεσμα. Ένας non-realtime renderer είναι ο Cycles του Blender.

Παράδειγμα 41 Final Render vs Solid Color with wireframe view

https://www.liveenhanced.com/3d-rendering-techniques-an-overview-of-various-techniques/

7.2 PBR (Physically Based Rendering)

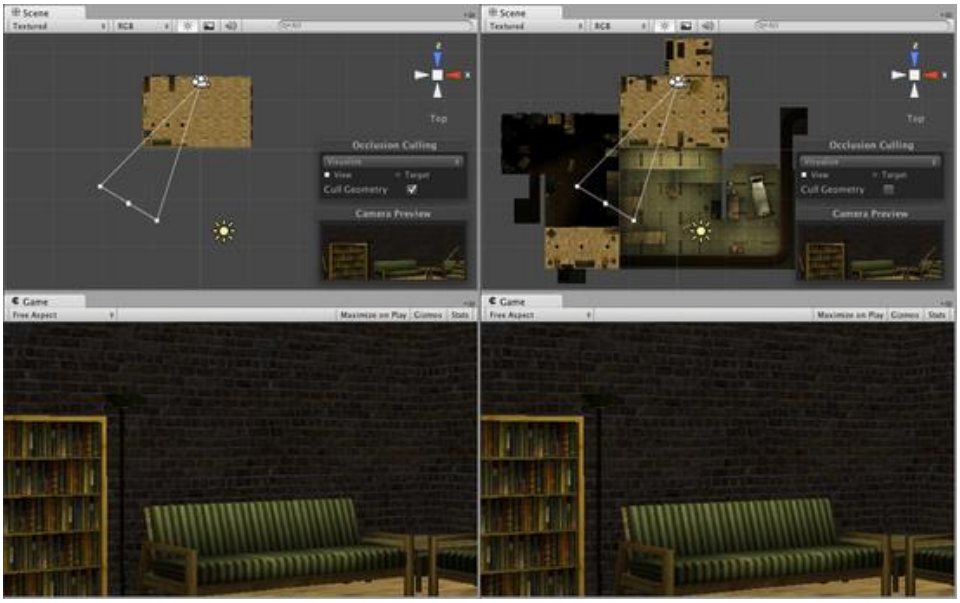
Physically Based Rendering είναι μια νέα τεχνολογία που έχει ως στόχο να αποδώσει τα γραφικά με τέτοιο τρόπο ώστε όλα τα μοντέλα να λαμβάνουν, ανακλούν, απορροφούν φως με τον πιο έγκυρο τρόπο, όπως ακριβώς συμβαίνει και στον πραγματικό κόσμο. Αυτό έχει ως αποτέλεσμα τα τελικά αποτελέσματα, ότι και αν είναι αυτά, να είναι πολύ πιο ρεαλιστικά. Αυτό που κάνει το PBR τόσο δημοφιλής, εκτός φυσικά από το πολύ καλό αποτέλεσμα, είναι ότι μπορεί να χρησιμοποιηθεί και για real-time εφαρμογές όπως παιχνίδια ή διαδραστικές εφαρμογές. Έχει αναπτυχθεί από πολλά λογισμικά, μερικά από αυτά είναι τα Unity, Unreal Engine, Blender, Sketchfab κ.α.. Το PBR πετυχαίνει αυτό το αποτέλεσμα, δίνοντας έμφαση στο πώς θα επεξεργαστούν ενέργειες που αφορούν κυρίως στα material ενός μοντέλου. Δίνει έμφαση στο πως θα υλοποιηθούν πολλά μέρη ενός material όπως τα reflections, transparency, translucency, diffusion, metalness και άλλα.

Παράδειγμα 42 Non-PBR vs PBR Shader

https://marmoset.co/posts/pbr-texture-conversion/

7.3 Occlusion Culling

Occlusion Culling είναι ένα χαρακτηριστικό που δίνει την δυνατότητα στην κάμερα να μην κάνει render αντικείμενα που δεν είναι ορατά σε αυτήν. Είτε επειδή είναι εκτός του πεδίου προβολής της είτε επειδή κρύβονται από άλλα αντικείμενα. Το Occlusion Culling είναι από τα βασικότερα συστατικά για την βελτιστοποίησης της εκτέλεσης του παιχνιδιού αφού αν υλοποιηθεί σωστά ανεβάζει κατακόρυφα τα FPS (Frames per second) του παιχνιδιού. (Unity Documentation, n.d.)



Παράδειγμα 43 Occlusion Culling

https://dev.rbcafe.com/unity/unity-3.4.2/Documentation/Manual/Occlusion%20Culling.html

8 Environment Art for Games

8.1 Τι είναι Environment Art

Environment Art στην βιομηχανία παιχνιδιών είναι η διαδικασία παραγωγής του κόσμου που θα διαδραματίζεται το παιχνίδι. Περιλαμβάνει την δημιουργία των terrain, το εσωτερικό και το εξωτερικό κτηρίων, απλά αντικείμενα που περιλαμβάνει μια σκηνή (κρεβάτια, τραπέζια κλπ.), οχήματα, βλάστηση (γρασίδι, δέντρα, λουλούδια) κλπ.. Όλα δηλαδή τα γραφικά που χρειάζεται ένα παιχνίδι εκτός των χαρακτήρων και των ειδικών εφέ. Ένας environment artist πρέπει να δημιουργήσει τα μοντέλα, τα UV Maps, τα materials, το φωτισμό και πολλά άλλα. Το τελικό αποτέλεσμα πρέπει να είναι έτοιμο για εισαγωγή στην μηχανή παιχνιδιών που χρησιμοποιεί η ομάδα που αναπτύσσει το παιχνίδι.

8.2 Terrain System

Ένα Terrain System περιέχει τα απαραίτητα εργαλεία που χρειάζεται ένας environment artist για να δημιουργήσει τα εδάφη για το παιχνίδι που δημιουργεί. Συνήθως αυτό το σύστημα είναι μέρος μια παιχνιδομηχανής όπως Unity ή Unreal Engine και τα εργαλεία διαφέρουν από την μια στην άλλη. Αυτό που κάνει τους δημιουργούς να χρησιμοποιούν αυτό το σύστημα δεν είναι μόνο για την πληθώρα εργαλείων που προσφέρει, αλλά και επειδή τα terrains που δημιουργούνται με αυτά τα συστήματα είναι έτσι σχεδιασμένα ώστε να μην χρειάζονται πολλούς πόρους κατά την διάρκεια του rendering. Αυτό δεν γίνεται να το καταφέρει κανείς αν δημιουργήσει εδάφη με τον κλασσικό τρόπο μοντελοποίησης αφού τα εργαλεία terrain είναι σχεδιασμένα για να δημιουργούν εδάφη για κάθε παιχνιδομηχανή ξεχωριστά.

Τα διάφορα εργαλεία που περιέχει ένα σύστημα terrain είναι brushes για σκάλισμα του εδάφους, εργαλεία για ζωγραφική textures πάνω στο πλέγμα, εργαλεία τοποθέτησης αντικειμένων όπως δέντρα, γρασίδι κλπ.

Παράδειγμα 44 Terrain created with Unity

http://www.tanukidigital.com/tenkoku/

8.3 Modular Assets

8.3.1 Modular Models

Modular Assets είναι αντικείμενα-κομμάτια αντικειμένων τα οποία μπορεί να ενώσει ο δημιουργός με διάφορους τρόπους ώστε να παράξει διάφορα αποτελέσματα. Ένα κλασσικό παράδειγμα modular αντικειμένων είναι ένα κομμάτι τοίχου, παράθυρου, ταβανιού, πατώματος, τα οποία ενώνει ο δημιουργός για να φτιάξει ένα σπίτι με μέγεθος και διάταξη της επιθυμίας του, ύστερα ένας άλλος δημιουργός μπορεί να πάρει το ίδιο πακέτο αντικειμένων και να παράξει ένα άλλο σπίτι.

Αυτή είναι μια πολύ καλή τεχνική για διάφορους λόγους. Αρχικά αν για παράδειγμα όλα τα σπίτια ενός παιχνιδιού αποτελούνται από τα ίδια αντικείμενα κάνει αυτόματα το αποτέλεσμα ομοιόμορφο καθώς δεν υπάρχουν μεγάλες διαφορές στον χρωματισμό και το σχήμα των σπιτιών πράγμα που συμβαίνει συχνά σε χωριά και πόλεις, που μπορεί να θέλει να αναπαραστήσει ο δημιουργός. Ένα επίσης μεγάλο πλεονέκτημα είναι ότι δεν χρειάζεται το λογισμικό να υπολογίσει πολλά διαφορετικά αντικείμενα και να φορτώσει πολλά textures γιατί όλα χρησιμοποιούν τα ίδια, έτσι το παιχνίδι κάνει οικονομία σε πόρους άρα είναι και πιο ελαφρύ. Το πιο βασικό βέβαια πλεονέκτημα είναι ότι ο δημιουργός μπορεί ανά πάσα στιγμή να κάνει οποιαδήποτε αλλαγή θέλει άρα και να πετύχει ένα καλύτερο αποτέλεσμα. Αυτό δεν θα ήταν δυνατό αν για παράδειγμα όλο το σπίτι ήταν ένα συμπαγές πλέγμα.

Παράδειγμα 45 Modular Assets of a building

https://gr.pinterest.com/pin/534802524499540916/?lp=true

8.3.2 Trim Textures

Trim Texture είναι ένα texture που περιλαμβάνει παραπάνω από μια επιφάνειες έτσι ώστε να χρησιμοποιηθεί για παραπάνω από ένα αντικείμενα. Είναι επίσης μιας μορφής modularity και έχει τα ίδια πλεονεκτήματα με τα modular αντικείμενα. Τα trim textures δημιουργούνται σε λογισμικά επεξεργασίας εικόνας, και συνήθως ο δημιουργός του είναι και ο δημιουργός των αντικειμένων που θα ενσωματωθούν. Αυτό γιατί για να λειτουργήσει ένα trim texture πρέπει οι θέσεις των πολυγώνων στο UV Map ενός αντικειμένου να συμβαδίζουν με το κάθε κομμάτι του texture.

Παράδειγμα 46 Trim Textures

https://www.youtube.com/watch?v=IziIY674NAw&t=70s

Παράδειγμα 47 Trim Textures Applied

https://www.youtube.com/watch?v=IziIY674NAw&t=70s

8.4 Props

Props είναι αντικείμενα που δεν είναι βασικό μέρους μιας σκηνής, ούτε μέρη του βασικού χαρακτήρα, ωστόσο είναι αυτά τα αντικείμενα που γεμίζουν την σκηνή και της δίνουν την δική της ταυτότητα. Ένα prop μπορεί να είναι από μια γλάστρα μέχρι την κουρτίνα ενός παραθυριού, από ένα διαλυμένο αυτοκίνητο μέχρι ένα πεσμένο κορμό δέντρου, οτιδήποτε γεμίζει την σκηνή αλλά δεν είναι απαραίτητο για εκτέλεση του παιχνιδιού. Παρόλο που δεν είναι απαραίτητα για την εκτέλεση του παιχνιδιού, είναι απαραίτητα για να δώσει ένα χαρακτήρα και μια ρεαλιστική αίσθηση στο περιβάλλον που περιηγείται ο παίχτης. Τα props διακρίνονται σε διάφορες κατηγορίες που κάθε μια έχει και άλλα χαρακτηριστικά τα οποία δείχνουν πότε και που πρέπει να υπάρχουν στη σκηνή. (Nelson, n.d.)

Αυτές οι κατηγορίες είναι οι εξής :

- **Common Props**: Είναι όλα τα αυτά τα αντικείμενα που δεν κουνιούνται και δεν μπορεί ο παίχτης να αλληλεπιδράσει με αυτά. Υπάρχουν για να γεμίζουν την σκηνή και να δίνουν την αίσθηση ότι κάθε περιβάλλον έχει κάποια ιστορία.
- **Dynamic Props**: Είναι αυτά τα αντικείμενα με τα οποία ο παίχτης μπορεί να αλληλεπιδράσει, μπορεί να εκτελέσει ενέργειες πάνω σε αυτά και συνεπώς να υπάρξει κάποιο αποτέλεσμα από αυτές τις ενέργειες. Κάποια παραδείγματα μπορεί να είναι, οι διακόπτες που ανοιγοκλείνουν ένα φως, ένα συρτάρι που ανοιγοκλείνει, ένα βαρέλι με βενζίνη που εκρήγνυται κλπ.. Συχνά έως πάντα αυτά τα αντικείμενα ανταποκρίνονται στους νόμους της φυσικής όπως βαρύτητα, συγκρούσεις και άλλα, συνεπώς περιέχουν κάποιο physics modifier που βέβαια διαφέρει από παιχνιδομηχανή σε παιχνιδομηχανή.
- **Supportive Props**: Ονομάζονται αλλιώς και pick-ups. Είναι αντικείμενα που σχετίζονται άμεσα με την εξέλιξη του παιχνιδιού αφού μπορεί να έχουν πολλές

επιδράσεις πάνω στον χαρακτήρα ή στην εκπλήρωση κάποιας αποστολής. Μπορεί να είναι κάποιο κλειδί που χρειάζεται ο παίχτης για να συνεχίσει το παιχνίδι, κάποιο αντικείμενο που δίνει ζωή στον παίχτη, αντικείμενα που βελτιώνουν τις ικανότητες του παίχτη ή του εξοπλισμού του και πάει λέγοντας.

- **Interactive Props**: Σχετίζονται με τα dynamic pros, ωστόσο έχουν κάποιες διαφορές, Είναι αντικείμενα με τα οποία αλληλοεπιδράει ο χαρακτήρας και έχουν άμεση επίδραση σε αυτόν και όχι στο περιβάλλον. Αυτά μπορεί να είναι, ένα όπλο, ένα μεταφορικό μέσο, ένα ασανσέρ κλπ.. Ο σκοπός τους είναι να βοηθήσουν τον παίχτη να εκτελέσουν κάποια αποστολή αλλά δεν είναι απαραίτητα για την εκπλήρωση της. Συνήθως είναι περίπλοκα αντικείμενα και βαριά για το παιχνίδι, συνεπώς δεν υπάρχουν πολλά στον χάρτη, και αν υπάρχουν χρησιμοποιούν το ίδιο μοντέλο με διαφορετικά textures.

Παράδειγμα 48 Game Props 1

https://www.coroflot.com/nigeqb/Game-Props

Παράδειγμα 49 Game Props 2

http://www.shapesandlines.com/generic-gallery-test02/low-poly-game-props-004/

8.5 Particle Systems

Ένα Particle System είναι ένα σύστημα παραγωγής σωματιδίων με σκοπό την οπτική αναπαράσταση υγρών ή άμορφων οντοτήτων. Αυτό επιτυγχάνετε αφού το σύστημα παράγει μικρά σωματίδια που μπορεί να είναι μια εικόνα ή ένα αντικείμενο, και ο συνδυασμός πολλών σωματιδίων φέρει το επιθυμητό αποτέλεσμα. Ένα τέτοιο σύστημα έχει πολλούς παραμέτρους και προσφέρει μεγάλη ευκαμψία στην χρήση του. Λόγω αυτού μπορούν να δημιουργηθούν πάρα πολλών ειδών εφέ. Ένα παράδειγμα χρήσης μπορεί να είναι ένα σύννεφο καπνού. Κάθε σωματίδιο αναπαριστά ένα μικροσκοπικό σύννεφο καπνού ενώ ο συνδυασμός πολλών σωματιδίων που αναπαράγονται με την κατάλληλη ταχύτητα, πυκνότητα και σε ένα κατάλληλο σχήμα, σχηματίζουν ένα μεγάλο σύννεφο καπνού.

Κάθε σωματίδιο έχει περιορισμένη διάρκεια ζωής που ορίζεται από τους παραμέτρους του συστήματος. Άλλοι παράμετροι είναι το χρώμα, το μέγεθος, η ταχύτητα και πολλά άλλα. Η χρήση αυτού του συστήματος είναι πολύ συχνή στα παιχνίδια αφού δίνει την δυνατότητα δημιουργίας πολλών εφέ που είναι απαραίτητα για ένα όμορφο και ρεαλιστικό περιβάλλον.

Παράδειγμα 50 Particles Fire Effect

https://80.lv/articles/best-vfx-for-unity/

8.6 Post-Processing

Post-Processing είναι μια διαδικασία που επιτρέπει στον δημιουργό να προσθέσει ειδικά εφέ σε μια ήδη rendered εικόνα ή βίντεο, ακόμα και στο αποτέλεσμα που βλέπει η κάμερα σε ένα παιχνίδι. Ο σκοπός αυτής της διαδικασία είναι να κάνει πιο όμορφο το τελικό αποτέλεσμα αλλά και πιο ρεαλιστικό πολλές φορές. Αυτό που κάνει το Post-Processing πολύ δημοφιλής είναι ότι είναι πολύ φθηνό σε πόρους και προφέρει πολύ ποιοτικά αποτελέσματα. Αυτή η διαδικασία διαφέρει από λογισμικό σε λογισμικό και έχει πολλούς παραμέτρους πράγμα που προσφέρει μεγάλη ευκαμψία στις επιλογές του δημιουργού.

Μερικές επιλογές είναι :

- Vignette: Που σκουραίνει τις γωνίες της εικόνας και δίνει έμφαση στο κέντρο της εικόνας.
- SSR: Δημιουργεί αντανακλάσεις στις κατάλληλες επιφάνειες και δίνει ένα παραπάνω ρεαλισμό στην εικόνα. Είναι χαμηλότερης ποιότητας αντανακλάσεις από αυτές των Reflection Probes, ωστόσο είναι ιδανικό για να ελαττωθεί η διαρροή φωτός.
- Motion Blur: Δημιουργεί ένα εφέ κίνησης στην κάμερα όταν αυτή κινείται θολώνοντας τις γωνίες.
- Grain: Δημιουργεί ένα εφέ κακής ποιότητας κάμερα, που μπορεί να χρησιμοποιηθεί σε διάφορες περιπτώσεις.
- Auto Exposure: Δημιουργεί το εφέ της προσαρμογής του ματιού στο φως. Αυτό συμβαίνει όταν η κάμερα από ένα σκοτεινό σημείο κοιτάξει απότομα σε ένα φωτεινό.

- Depth of Field: Βοηθάει την κάμερα να κάνει focus σε ένα αντικείμενο και να θολώνει τα υπόλοιπα. Συχνό εφέ στην φωτογράφιση στον πραγματικό κόσμο.
- Bloom: Δίνει παραπάνω ένταση στα φωτεινά σημεία της σκηνής.
- Color Grading: Βοηθάει στην αλλαγή αποχρώσεων του χρώματος της εικόνας.
- Anti-Aliasing: Δίνει στα γραφικά μια πιο λεία εμφάνιση αφού κάνει όλες τις γωνίες πιο λείες.

Παράδειγμα 51 Post-Processing

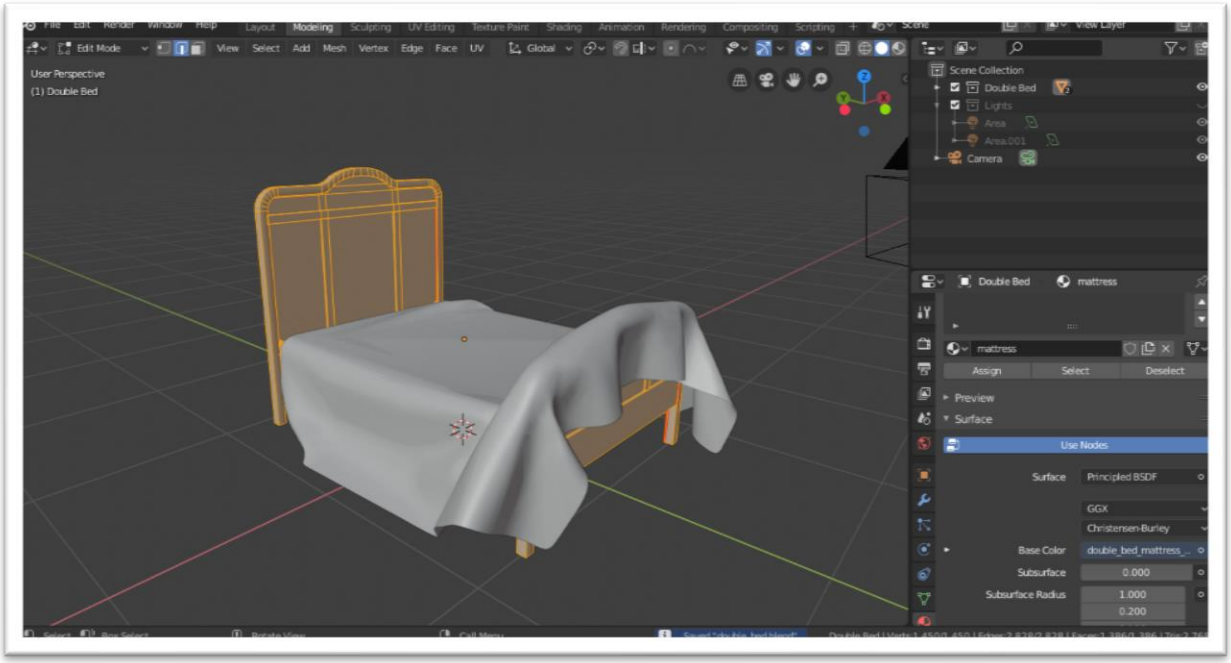
https://forum.kerbalspaceprogram.com/index.php?/topic/168763-17x-ks3p/

9 Τα λογισμικά που θα χρησιμοποιηθούν

9.1 Blender

Το Blender είναι ένα δωρεάν και open source λογισμικό για την παραγωγή 3D προϊόντων. Περιέχει κάθε είδους συστήματα επεξεργασίας 3D πληροφορίας, όπως modelling, rigging, animation, simulation, rendering, compositing, motion tracking, video editing και game creation. Πολλά άτομα ανά τον κόσμο αλλά και μικρά studios χρησιμοποιούν το Blender και επωφελούνται από το γεγονός ότι είναι open source. Αυτό γιατί εκτός απ' το ότι είναι δωρεάν, προσφέρει την δυνατότητα σε κάποιον προγραμματιστή να κάνει αλλαγές στο λογισμικό και να το φέρει σε μια μορφή που θα βοηθήσει στην ανάπτυξη των προϊόντων που θέλει να δημιουργήσει αυτός ή και η ομάδα. Πολλά addons έχουν δημιουργηθεί από προγραμματιστές που λύνουν τα χέρια σε πολλούς χρήστες του λογισμικού.

Στην παρούσα πτυχιακή εργασία, θα χρησιμοποιηθεί για διάφορους σκοπούς. Πρώτα απ' όλα θα χρησιμοποιηθεί για την μοντελοποίηση όλων των αντικειμένων που υπάρχουν στο παιχνίδι, από τους τοίχους του σπιτιού μέχρι το ταβάνι και το πάτωμα αλλά και όλα τα αντικείμενα που θα γεμίζουν την σκηνή, όπως κρεβάτια, τραπέζια, βιβλία, λάμπες κλπ.. Έπειτα με το ίδιο λογισμικό θα δημιουργηθούν τα UV Maps για κάθε αντικείμενο. Όταν τελειώσει αυτή η διαδικασία θα γίνει η εξαγωγή του κάθε αντικειμένου ξεχωριστά για να εισαχθεί στο πρόγραμμα δημιουργίας texture. Μετά αφού τελειώσει αυτή η διαδικασία θα ενωθούν όλα τα αντικείμενα μαζί για να δημιουργηθεί το επίπεδο που θα διαδραματίζεται το παιχνίδι. Τέλος θα γίνει η εξαγωγή των αντικειμένων για να εισαχθούν στην παιχνιδομηχανή.

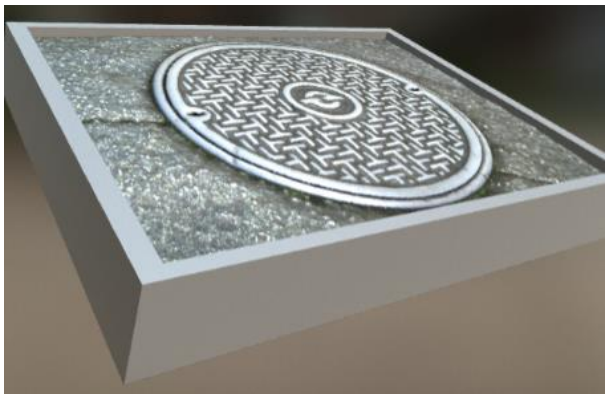


Παράδειγμα 52 Blender Layout

9.2 Materialize

Materialize είναι ένα δωρεάν και open source εργαλείο που παράγει όλα τα απαραίτητα texture maps από μία απλή εικόνα. Βοηθάει δηλαδή στην δημιουργία όλων των απαραίτητων συστατικών που χρειάζονται για να δημιουργηθεί ένα υψηλής ποιότητας PBR Material. Είναι ένα εργαλείο με πολλές επιλογές και δίνει μεγάλη ευκαμψία στο δημιουργό, αφού σε κάθε μέρος της διαδικασίας δημιουργίας ενός texture map προσφέρει πάρα πολλές επιλογές για να παραχθεί το κατάλληλο αποτέλεσμα. Στην παρούσα πτυχιακή εργασία θα χρησιμοποιηθεί για την δημιουργία ορισμένων texture που θα χρειαστούν κάποια αντικείμενα ώστε να τελειοποιηθούν. Αυτά τα texture είτε θα εισαχθούν σε άλλο λογισμικό για να χρησιμοποιηθούν είτε θα ενσωματωθούν αυτούσια στα Material των αντικειμένων.

Παράδειγμα 53 Simple Image



Παράδειγμα 54 All texture maps generated and applied

9.3 Substance Painter

Πρόκειται για ένα λογισμικό της Adobe με πλούσια λειτουργικότητα πάνω στην διαδικασία Texturing και Texture Painting. Σκοπός της χρήσης του είναι η δημιουργία των texture ενός μοντέλου. Καθώς περιέχει πολλά εργαλεία όπως masking, procedural tools, layers κ.α. επιτρέπει την δημιουργία πολύπλοκων αποτελεσμάτων που χωρίς αυτό θα ήταν αν όχι αδύνατον, πολύ δύσκολο και χρονοβόρο να επιτευχθούν.

Χρησιμοποιεί κάθε είδους PBR Workflow και συνεπώς είναι συμβατό με όλα τα 3D λογισμικά που μπορεί να χρησιμοποιήσει κάνεις, όπως Blender, Unity, Unreal Engine κλπ. Περιέχει ενσωματωμένα pre-set materials που βοηθούν τον χρήστη να ξεκινήσει. Υποστηρίζει μέχρι και 8K textures. Είναι ένα εργαλείο που εκτός των πολύ περίπλοκων και ποιοτικών αποτελεσμάτων που παράγει, βοηθάει τον χρήστη να πετύχει αυτά τα αποτελέσματα αρκετά γρήγορα και με τέτοια ευκαμψίας ώστε να μπορούν να γίνουν οποιεσδήποτε αλλαγές οποιαδήποτε στιγμή. Ένα επίσης πολύ μεγάλο πλεονέκτημα είναι ότι υπάρχουν ήδη πολλά έτοιμα materials online για οποιονδήποτε, πράγμα που βοηθάει στην επίτευξη κάποιων αποτελεσμάτων αν αυτά υποβληθούν σε επεξεργασία.

Είναι πλέον ένα standard λογισμικό στην βιομηχανία παιχνιδιών. Έτσι στην παρούσα πτυχιακή εργασία θα χρησιμοποιηθεί για την παραγωγή σχεδόν κάθε texture κάθε

αντικειμένου. Αυτό που χρειάζεται το συγκεκριμένο λογισμικό είναι ένα πολυγωνικό μοντέλο με τα UV Maps του, πράγμα που θα γίνει στο Blender.

Παράδειγμα 55 Substance Painter Software

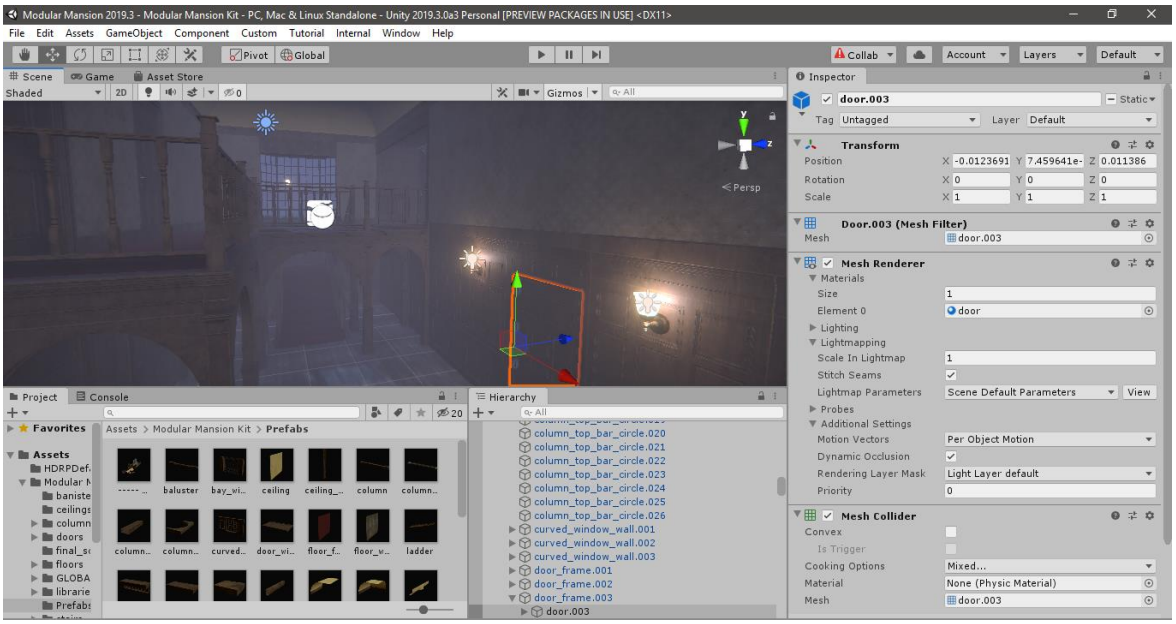
https://www.toolfarm.com/tutorial/substance_painter_texturing_for_beginners/

9.4 Unity

Το Unity είναι μια μηχανή παιχνιδιών. Μια μηχανή παιχνιδιών είναι το λογισμικό που απαιτείται για την δημιουργία ενός παιχνιδιού. Περιέχει όλα τα κατάλληλα εργαλεία για την ανάπτυξη ενός πλήρως λειτουργικού παιχνιδιού. Σε αυτό το λογισμικό εισάγονται όλα τα γραφικά που απαρτίζουν μια σκηνή, δημιουργούνται animations. Γράφεται ο κώδικας του παιχνιδιού για κάθε σύστημα, αντικείμενο και χαρακτήρα. Αναλαμβάνει επίσης το rendering, την φυσική που απαιτείται για τα αντικείμενα, όπως βαρύτητα, συγκρούσεις κλπ., καθώς και την διαχείριση μνήμης και φυσικά την εξαγωγή του παιχνιδιού σε ένα εκτελέσιμο πρόγραμμα.

Μια μηχανή παιχνιδιών περιέχει πέντε βασικά συστατικά. Το βασικό πρόγραμμα που αναλαμβάνει να διαχειριστεί την λογική του παιχνιδιού. Μια μηχανή rendering που είναι αναγκαία για να παράγει τα γραφικά και να τα εμφανίσει στη οθόνη. Μια μηχανή ήχου, που διαχειρίζεται ότι αφορά τον ήχο στο παιχνίδι. Μια μηχανή φυσικής, που αναλαμβάνει να υλοποιήσει και να υπολογίσει ότι είναι απαραίτητο για να αλληλοεπιδρούν σωστά τα αντικείμενα. Και ένα AI σύστημα, για όλη την τεχνητή νοημοσύνη που μπορεί να χρειαστεί ένα παιχνίδι.

Στην παρούσα πτυχιακή εργασία, το Unity θα χρησιμοποιηθεί αρχικά για την εισαγωγή των γραφικών, την δημιουργία των material κάθε αντικειμένου. Την δημιουργία της σκηνής που θα διαδραματίζεται το παιχνίδι. Τον ήχο που θα χρειαστεί το παιχνίδι. Τα scripts του χαρακτήρα για την κίνηση του, τα scripts για τα αντικείμενα και το πως θα αλληλοεπιδρούν με τον χαρακτήρα. Καθώς και την εξαγωγή του παιχνιδιού για Windows, Linux ή MAC.



Παράδειγμα 56 Unity Layout

10 Διαδικασία ανάπτυξης γραφικών

Στο συγκεκριμένο κεφάλαιο θα περιγραφούν όλη η μεθοδολογία και οι τεχνικές που ακολουθήθηκαν σε κάθε στάδιο της πτυχιακής εργασίας. Από το πως να δημιουργηθεί ένα καλό μοντέλο, στην διαδικασία texturing του μοντέλου, μέχρι και πώς να συνδυαστούν όλα τα μοντέλα μαζί για να παραχθεί μια ολοκληρωμένη σκηνή. Μετέπειτα θα περιγραφούν όλες οι διαδικασίες που ήταν απαραίτητες για την δημιουργία των γραφικών της παρούσας πτυχιακής εργασίας, όπως φωτισμός, animation, ήχος, προγραμματισμός κλπ..

10.1Δημιουργία αντικειμένων

Σε αυτό το υπό κεφάλαιο θα περιγραφούν οι τεχνικές και οι διαδικασίες που είναι απαραίτητες για την δημιουργία ενός μοντέλου που θα είναι έτοιμο για να εισαχθεί σε μια μηχανή παιχνιδιών. Καθώς όμως θα ήταν αδύνατο να περιγραφεί η διαδικασία για κάθε μοντέλο που δημιουργήθηκε στην πτυχιακή εργασία, θα γίνει μόνο για ένα συγκεκριμένο παράδειγμα. Το παράδειγμα που θα χρησιμοποιηθεί θα είναι μια ντουλάπα.

Μεθοδολογία 1 Το παράδειγμα που θα χρησιμοποιηθεί

10.1.1Συλλογή πληροφοριών και εικόνων αναφοράς

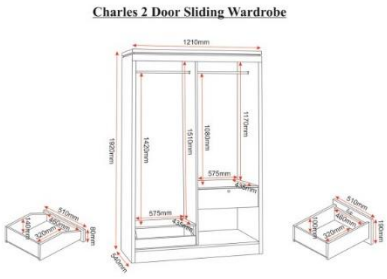
Πριν ξεκινήσει κανείς να σχεδιάζει αντικείμενα, καλό να προηγηθεί μια μικρή έρευνα για αυτό που πρόκειται να δημιουργήσει. Αυτό γιατί συχνά ο άνθρωπος, ενώ είναι σίγουρος ότι θυμάται την όψη ενός αντικειμένου, όταν το κοιτάει παρατηρεί κάποιες λεπτομέρειες που το μυαλό είναι αδύνατον να θυμάται. Κάτι επίσης απαραίτητο που πρέπει να ξέρει ο δημιουργός πριν αρχίσει την μοντελοποίηση είναι οι πραγματικές διαστάσεις του αντικειμένου ώστε τα

σχετικά μεγέθη των αντικειμένων να είναι αυτά που έχει συνηθίσει το μάτι και η σκηνή να είναι ρεαλιστική.

Με μια απλή αναζήτηση στο διαδίκτυο, βρίσκει κανείς όλες τις απαραίτητες πληροφορίες που χρειάζεται. Στην περίπτωση της ντουλάπας υπήρχαν πολλές επιλογές και πολλές πληροφορίες. Αυτές που επιλέχθηκαν φαίνονται στις παρακάτω εικόνες.

Μεθοδολογία 2 Φωτογραφία αναφοράς παραδείγματος

http://bensherra.info/old-wardrobe-closet/



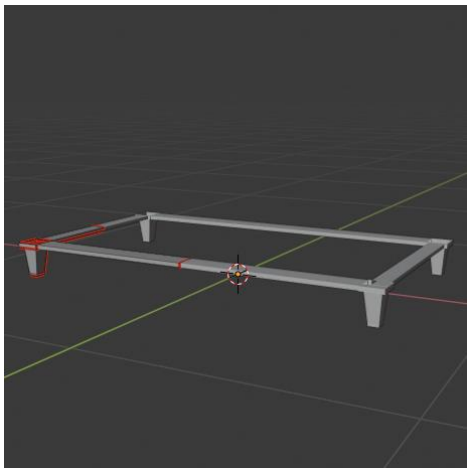
Μεθοδολογία 3 Πληροφορίες παραδείγματος

https://furniture123.co.uk/charles-2-door-sliding-wardrobe-in-oak_34700

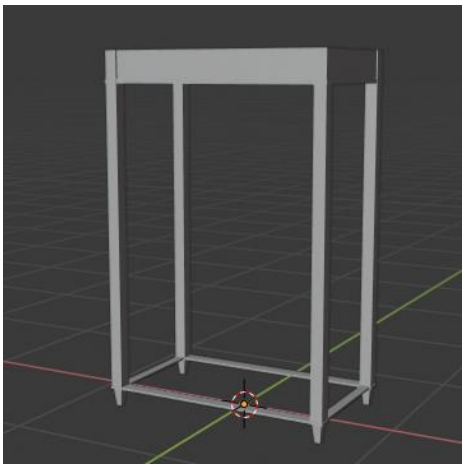
10.1.2 Μοντελοποίηση και UV unwrapping

Όπως ειπώθηκε και στα προηγούμενα κεφάλαια, η μοντελοποίηση και το UV unwrapping στην παρούσα πτυχιακή εργασία θα γίνουν στο Blender (έκδοση 2.8). Για το συγκεκριμένο παράδειγμα, είναι πιο βολικό να αρχίσει η μοντελοποίηση, πρώτα από τα πόδια της ντουλάπας, ύστερα, από τα πλάγια, και το πίσω μέρος, μετά από το κάτω και το πάνω μέρος του συρταριού και τέλος από το πάνω μέρος της ντουλάπας, τις πόρτες και το συρτάρι. Όπως προ ειπώθηκε, σε αυτήν την διαδικασία θα χρησιμοποιηθούν τεχνικές για non-destructive modeling. Στο συγκεκριμένο παράδειγμα αυτές οι τεχνικές θα είναι οι mirror και bevel modifier. Αφού ολοκληρωθεί το βασικό σχήμα του αντικειμένου, είναι ώρα να προστεθούν οι μικρές λεπτομέρειες, όπως τα χερούλια της πόρτας και του συρταριού καθώς και οι μεντεσέδες.

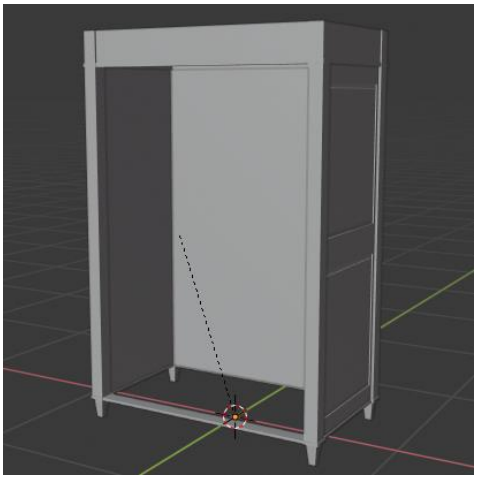
Αφού ολοκληρωθεί αυτή η διαδικασία, το επόμενο βήμα είναι να δημιουργηθούν τα UV maps ώστε να μπορέσει ύστερα το λογισμικό texturing (Substance Painter) να πάρει τις κατάλληλες πληροφορίες ώστε να παράγει το επιθυμητό αποτέλεσμα. Αυτή η διαδικασία γίνεται με τον εξής τρόπο. Επιλέγονται τα edges που πρέπει να θεωρηθούν ως seams (να κοπούν για να ξεδιπλωθεί η επιφάνεια) και ύστερα εκτελείται η εντολή Unwrap του Blender (Στο blender τα edges φαίνονται με κόκκινο χρώμα). Είναι αναγκαίο να μπουν όσο το δυνατόν λιγότερα seams και να μην υπάρχουν στρεβλώσεις στο UV map. Καθώς και τα απομονωμένα κομμάτια να βρίσκονται στις σωστές θέσεις σε σχέση με τα άλλα για να μην υπάρχουν μεγάλες διαφορές ανάμεσα στις δύο επιφάνειες όταν ενσωματωθεί το texture.



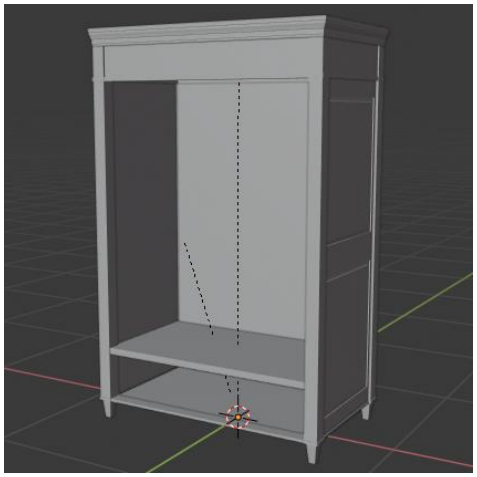
Μεθοδολογία 4 Μοντελοποίηση #1



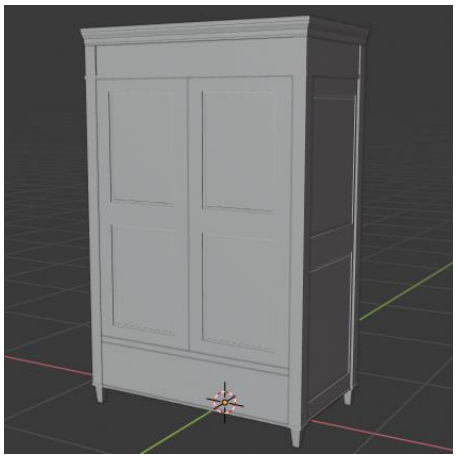
Μεθοδολογία 5 Μοντελοποίηση #2



Μεθοδολογία 6 Μοντελοποίηση #3

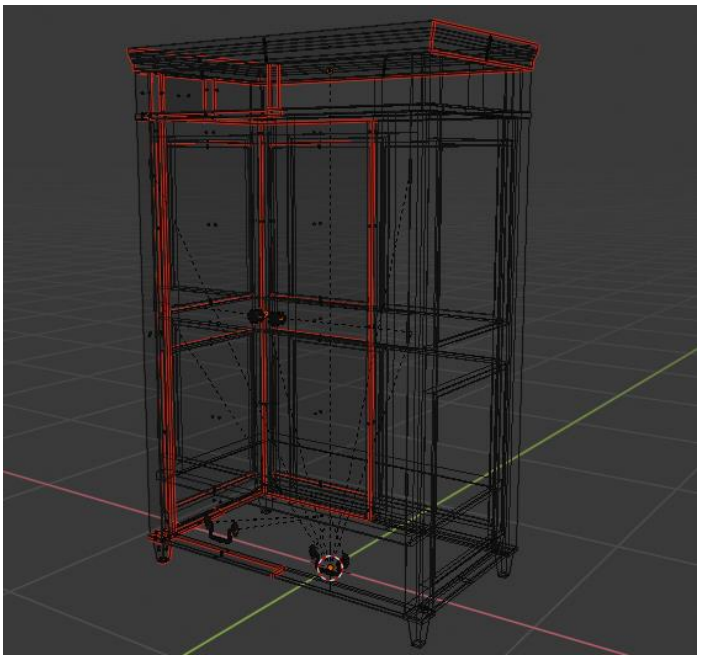


Μεθοδολογία 7 Μοντελοποίηση #4

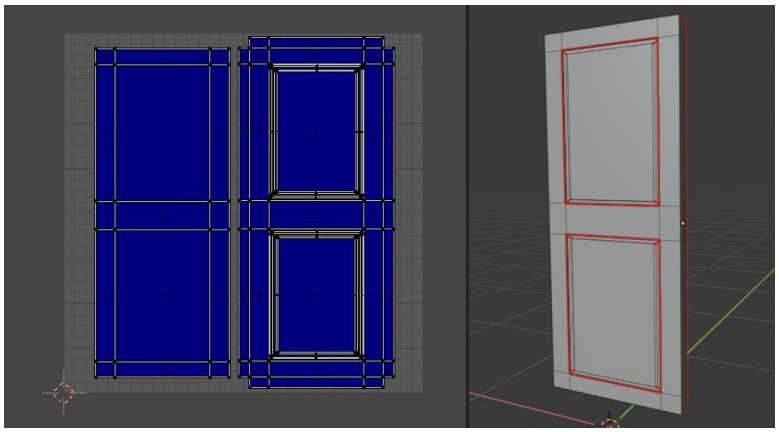


Μεθοδολογία 8 Μοντελοποίηση #5

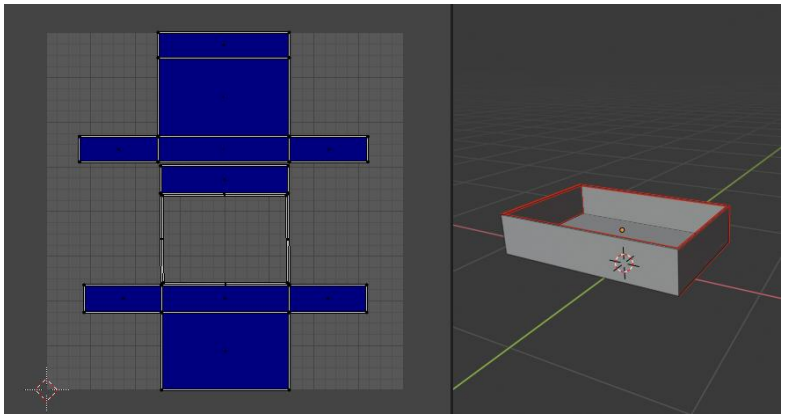
Μεθοδολογία 9 Μοντελοποίηση #6



Μεθοδολογία 10 UV Unwrapping



Μεθοδολογία 11 UV map της πόρτας

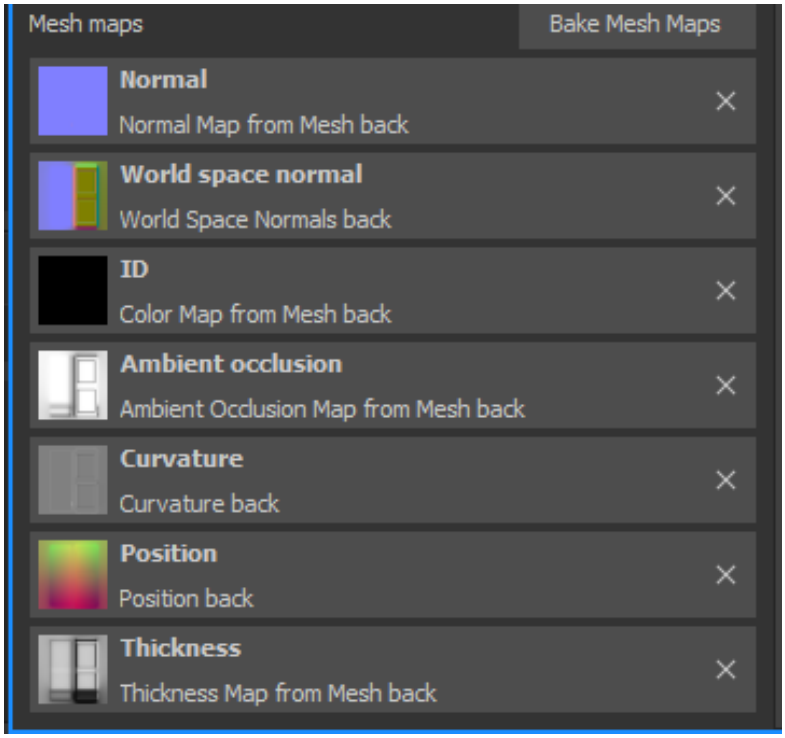


Μεθοδολογία 12 UV map του συρταριού

10.1.3 Texturing

Αφού ολοκληρωθούν οι διαδικασίες, modelling και UV unwrap. Πρέπει να γίνει η εξαγωγή του μοντέλου σε αρχείο .fbx για να εισαχθεί στο Substance Painter. Την εξαγωγή αναλαμβάνει να κάνει το λογισμικό μοντελοποίησης. Όταν γίνει αυτό, το αρχείο εισάγεται σε ένα νέο project του Substance Painter, όπου και θα γίνει το texturing.

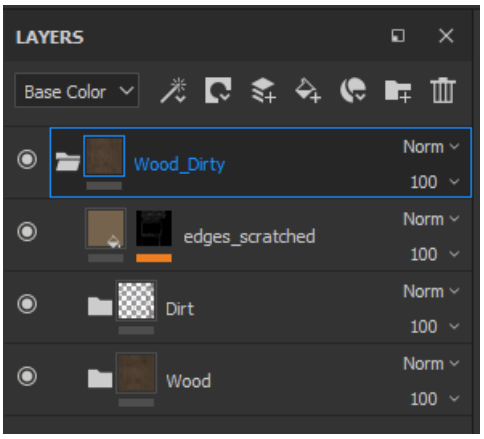
Το πρώτο πράγμα που πρέπει να γίνει αφού δημιουργηθεί το project είναι να δημιουργηθούν τα mesh maps. Τα mesh maps είναι διάφορες πληροφορίες σε μορφή εικόνας για το πλέγμα. Θα τις χρησιμοποιήσει το λογισμικό για να ξέρει που βρίσκονται καμπύλες, άκρα και άλλες λεπτομέρειες πάνω στο πλέγμα.



Μεθοδολογία 13 Baked Mesh Maps

Αυτό πρέπει να γίνει για κάθε Texture Set, δηλαδή για κάθε material του blender. Κάτι στο οποίο πρέπει να δοθεί προσοχή είναι να μην έχουν το ίδιο material, faces που έχουν overlapping UVs. Να μην είναι δηλαδή το ένα πάνω στο άλλο. Αυτό γιατί το Substance Painter θα μπερδευτεί και θα δημιουργηθούν ατέλειες στο texture.

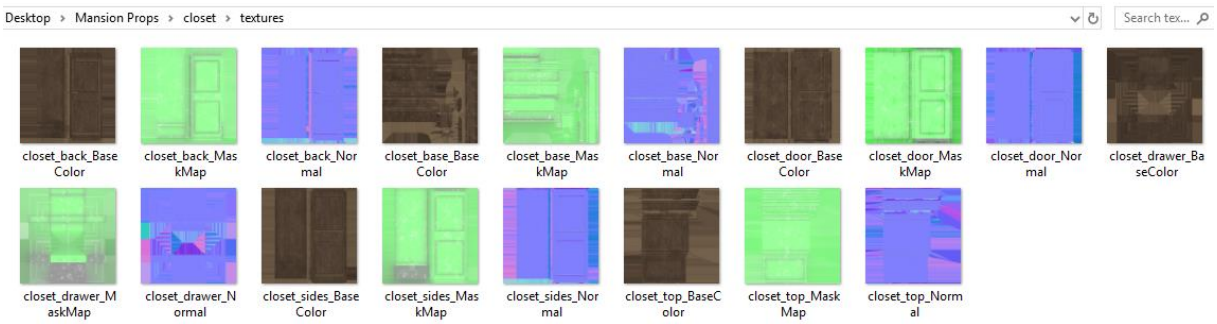
Αφού δημιουργηθούν τα mesh maps, μπορούν να γίνουν οι κατάλληλες ενέργειες για την δημιουργία των material. Αυτές οι ενέργειες είναι πολλές στο Substance Painter και για αυτό το λόγο είναι και τόσο δυναμικό. Αυτό που χρησιμοποιήθηκε στο παρών αντικείμενο είναι ένα wood material του substance painter, στο οποίο αλλάχτηκαν κάποιοι παράμετροι και προστέθηκαν κάποια smart masks. Αφού ολοκληρωθούν όλα τα texture sets τότε αυτά γίνονται export για να εισαχθούν μαζί με το μοντέλο στην μηχανή παιχνιδιού (Unity). Αυτό που πρέπει να γίνει export για να εισαχθεί στο Unity και στο HDRP είναι το Base Color το Normal Map και το Mask Map. Αυτά επιλέγονται στο Substance Painter στις ρυθμίσεις του exporter.



Μεθοδολογία 14 Wood Material

Μεθοδολογία 15 Wood Material Preview

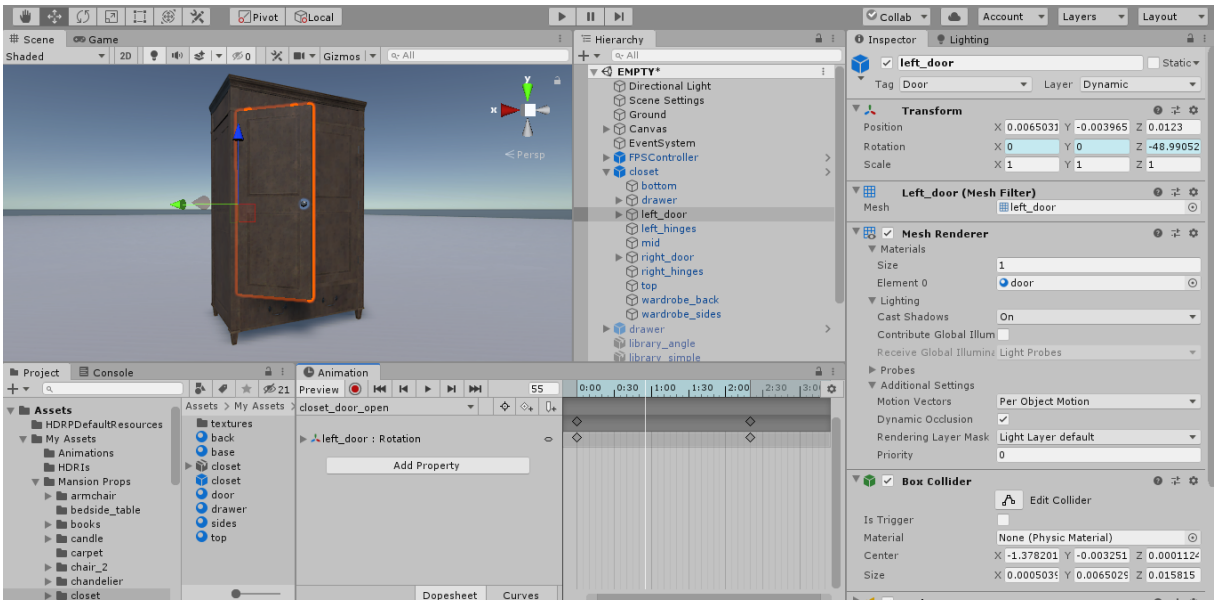
Μεθοδολογία 16 Fully textured model



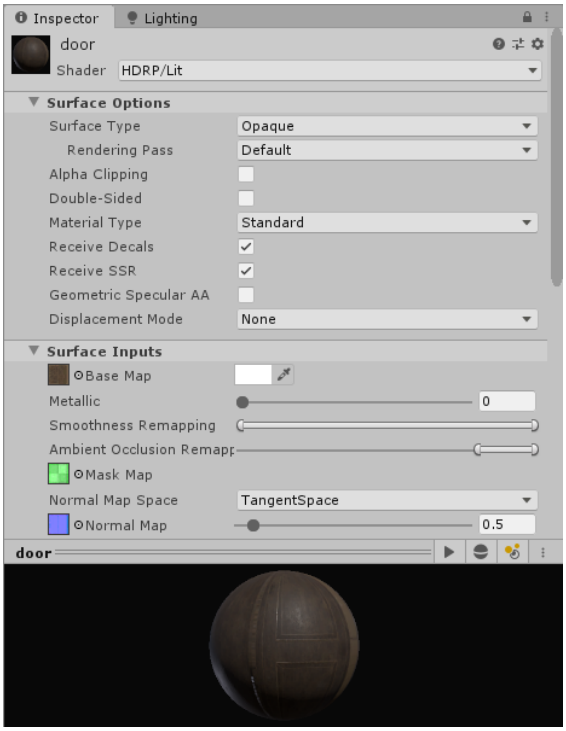
Μεθοδολογία 17 Exported Textures

10.1.4 Import στην μηχανή παιχνιδιού

Τέλος πρέπει να γίνει import του μοντέλου και των texture στο Unity για την τελική χρήση τους. Στο Unity θα δημιουργηθούν τα materials που θα περιέχουν τα texture καθώς και οι colliders και τα animation των πορτών και του συρταριού.



Μεθοδολογία 18 Game Ready Object



Μεθοδολογία 19 Unity HDRP Material Properties

10.2 Τα βήματα που ακολουθήθηκαν

10.2.1Συλλογή πληροφοριών

Το πρώτο πράγμα που έπρεπε να γίνει για να ξεκινήσει η πτυχιακή εργασία ήταν η συλλογή πληροφοριών σχετικά με το θέμα του level που έπρεπε να δημιουργηθεί. Εφόσον το θέμα ήταν ένα παιχνίδι τρόμου, μια ωραία επιλογή θα ήταν η δημιουργία ενός σπιτιού το οποίο θα έπρεπε ωστόσο να είναι αρκετά μεγάλο ώστε το παιχνίδι να έχει πολλές επιλογές. Για αυτό το λόγο αποφασίστηκε να δημιουργηθεί ένα παλιό εγκαταλελειμμένο μέγαρο. Έτσι, έγινε μια έρευνα για φωτογραφίες του συγκεκριμένου θέματος.

Μεθοδολογία 20 Level Παράδειγμα #1

https://cellcode.us/quotes/victorian-manor-hallway.html

Μεθοδολογία 21 Level Παράδειγμα #2

https://www.mountvernon.org/the-estate-gardens/the-mansion/the-mansion-room-by-room/

Μεθοδολογία 22 Level Παράδειγμα #3

https://www.mountvernon.org/the-estate-gardens/the-mansion/the-mansion-room-by-room/

Μεθοδολογία 23 Level Παράδειγμα #4

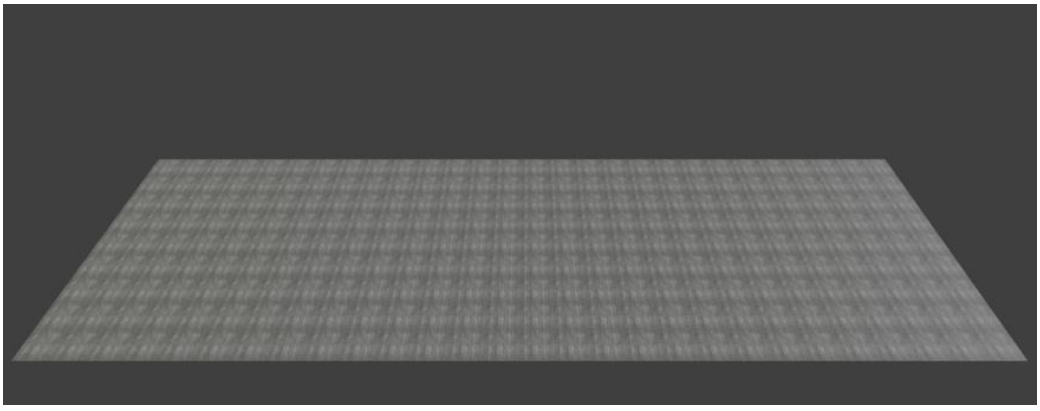
https://www.mountvernon.org/the-estate-gardens/the-mansion/the-mansion-room-by-room/

10.2.2 Δημιουργία βασικού level

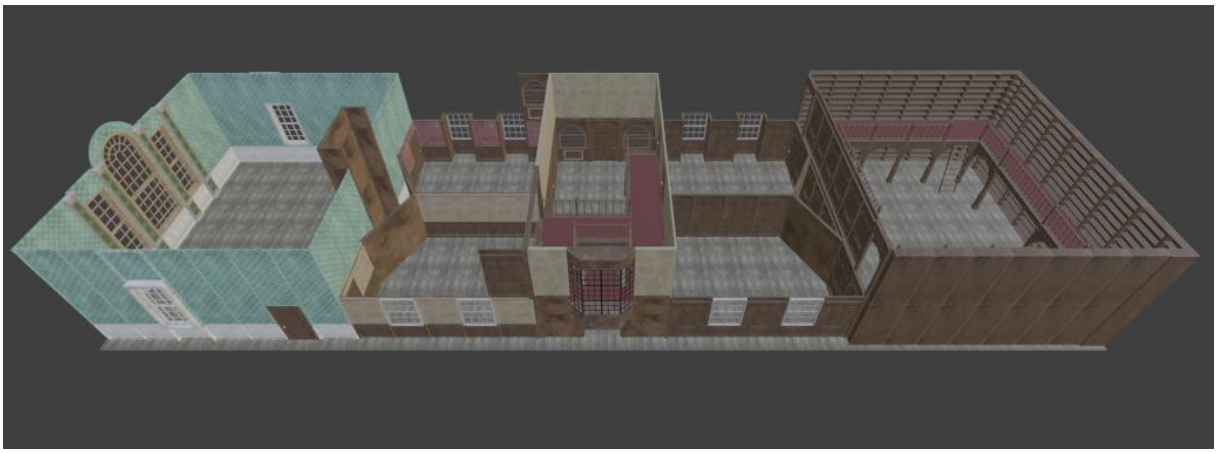
Αφού συλλέχτηκαν οι απαραίτητες πληροφορίες και αποφασίστηκε τι ακριβώς είναι επιθυμητό να δημιουργηθεί, έπρεπε να δημιουργηθούν τα όλα τα modular κομμάτια τα οποία όταν ενωθούν θα μπορούν να δημιουργήσουν ένα ολοκληρωμένο level στο οποίο θα μπορούν να γίνουν αλλαγές πολύ εύκολα. Τα κομμάτια αυτά ήταν πολλά, αλλά τα πιο κύρια ήταν ένας κομμάτι τοίχου, πατώματος, ταβανιού, πόρτας, παράθυρου κλπ.

Μεθοδολογία 24 Modular Parts

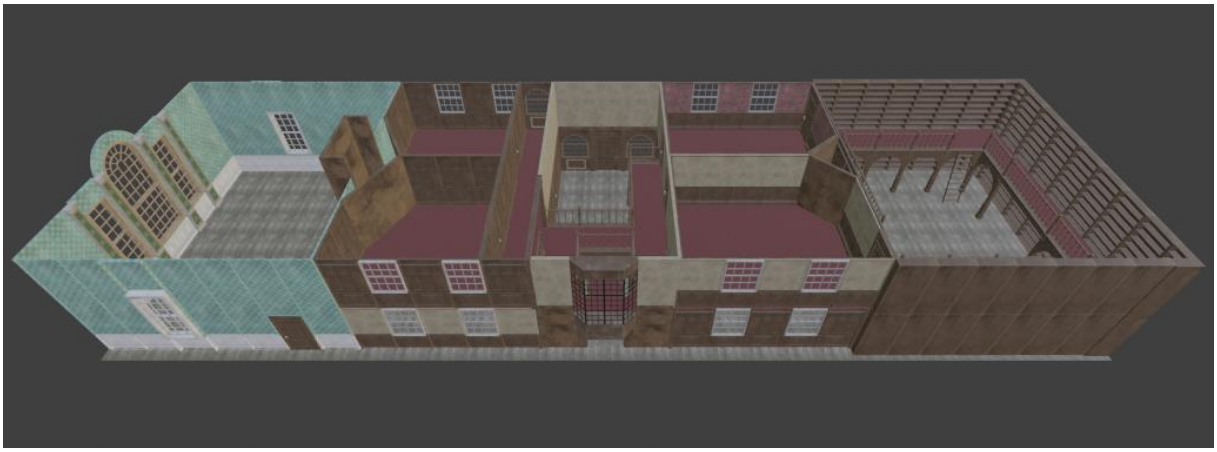
Αφού δημιουργήθηκαν όλα τα απαραίτητα μέρη του σπιτιού, ενώθηκαν για να φτιάξουν ένα ολοκληρωμένο σπίτι το οποίο και θα ήταν το βασικό κομμάτι του level. Εκεί θα έπαιζε ο παίχτης και πάνω σε αυτό θα έμπαιναν όλα τα αντικείμενα που θα συμπλήρωναν το παιχνίδι. Αυτό στο οποίο έπρεπε να δοθεί μεγάλη προσοχή είναι στην διαδικασία texturing των συγκεκριμένων αντικειμένων αφού έπρεπε όταν έμπαιναν δίπλα δίπλα δύο αντικείμενα να φαίνεται σαν να είναι ένα και όχι να αλλάζει. Για παράδειγμα, όπως τελείωνε η δεξιά μεριά του τοίχου έτσι έπρεπε να αρχίζει η αριστερή.



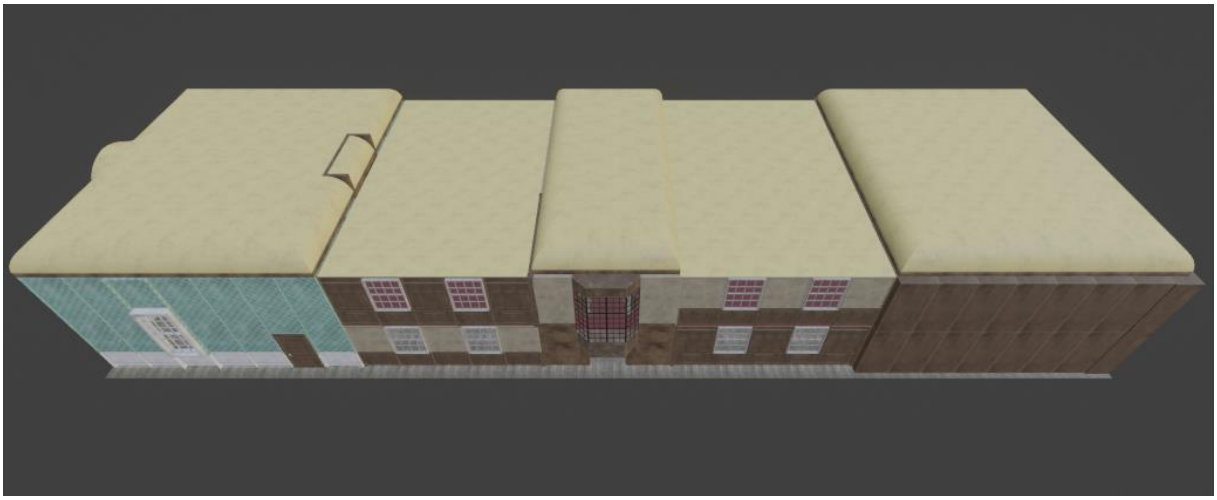
Μεθοδολογία 25 Modular Parts United #1



Μεθοδολογία 26 Modular Parts United #2



Μεθοδολογία 27 Modular Parts United #3

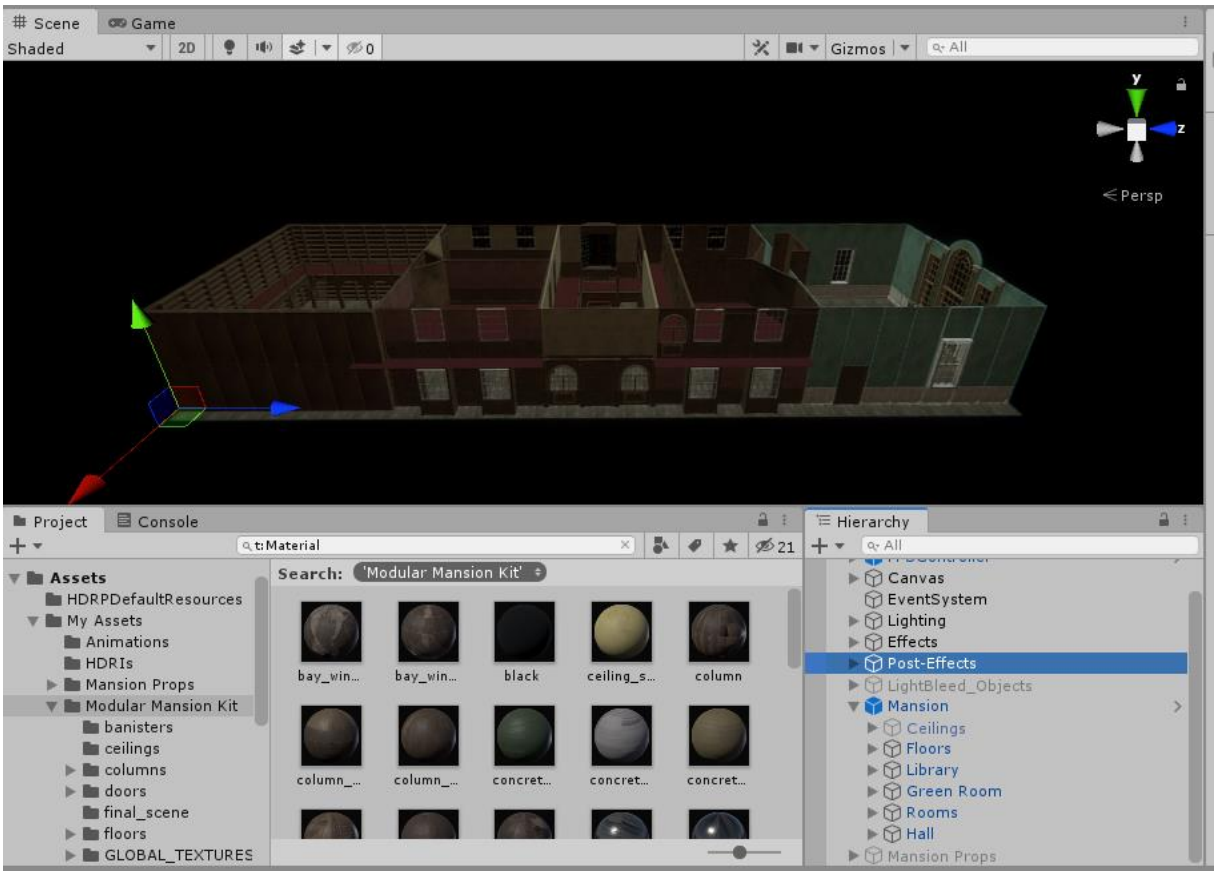


Μεθοδολογία 28 Modular Parts United #4

Η διαδικασία της ένωσης των κομματιών θα μπορούσε να είχε γίνει είτε στην μηχανή παιχνιδιού, αφού γινόταν η εισαγωγή του κάθε κομματιού ξεχωριστά, είτε στο Blender και να εισαγόταν όλο μαζί στην μηχανή παιχνιδιού. Επιλέχθηκε η δεύτερη επιλογή, πράγμα που δεν αλλάζει το τελικό αποτέλεσμα.

10.2.3 Import level στην μηχανή παιχνιδιού

Αφού δημιουργήθηκαν τα modular μέρη και το βασικό level, έγιναν export σε αρχείο .fbx και εισάχθηκαν μαζί με τα απαραίτητα texture σε ένα νέο project του Unity 2019.3 με HRDP Template. Στο Unity δημιουργήθηκαν όλα τα απαραίτητα materials που περιείχαν τα textures και αυτά ενσωματώθηκαν στο κατάλληλο αντικείμενο.



Μεθοδολογία 29 Level Layout in Unity

Αφού εισαχθούν και γίνουν οι κατάλληλες ρυθμίσεις είναι πολύ εύκολο να αλλάξεις την θέση ή το μέγεθος ενός αντικείμενο ακόμα και να δημιουργήσεις αντίγραφα αντικειμένων. Συνεπώς μπορούν να γίνουν τεράστιες αλλαγές στο level χωρίς περεταίρω modelling, texturing κλπ. Αυτό είναι πολύ σημαντικό όταν σε ένα project δουλεύουν πάνω από ένα άτομα, αφού συνήθως άλλος φτιάχνει τα μοντέλα, άλλος τα texture και άλλος το level design.

Το επόμενο πράγμα που έπρεπε να γίνει είναι να δημιουργηθούν τα colliders του level για να μπορούν να γίνονται οι κατάλληλοι υπολογισμοί σε νόμους της φυσική κλπ. Αυτό είναι κάτι που γίνεται πολύ εύκολα στο Unity καθώς έχει την επιλογή να το δημιουργήσει μόνο του και δεν χρειάζεται να το κάνει ο δημιουργός χειροκίνητα.

10.2.4 Δημιουργία βασικού φωτισμού και εφέ

Το επόμενο βήμα για να επιτευχθεί ο επιθυμητός ρεαλισμός στο level είναι να προστεθούν τα κατάλληλα εφέ και φωτισμοί, αυτό γίνεται μέσα στο Unity. Το πρώτο πράγμα που θα δημιουργηθεί είναι ένα Directional Light που θα αναπαριστά το φως που παράγει το φεγγάρι (εφόσον είναι νύχτα). Ύστερα θα γίνουν οι απαραίτητες ρυθμίσεις για να δημιουργηθεί το εφέ του Volumetric Light, ώστε να υπάρχει η αίσθηση της σκόνης και της ομίχλης. Εκτός του directional light δημιουργήθηκαν και οι υπόλοιποι φωτισμοί που θα προέρχονται από άλλες πηγές όπως λάμπες κλπ. Κάτι επίσης βασικό, είναι ότι πολλές φορές σε λεπτά αντικείμενα το φως περνάει από μέσα και προκαλεί ένα φαινόμενο που λέγεται light bleed. Αυτό λύνεται τοποθετώντας έξτρα αντικείμενα γύρο από άλλα αντικείμενα τα οποία όμως δεν θα είναι ορατά αλλά περιορίζουν το φως στο να περάσει.

Μεθοδολογία 30 No Light Scene

Μεθοδολογία 31 Directional Light & Volumetric

Μεθοδολογία 32 Scene with Lights

Μεθοδολογία 33 Light Bleed Solution

10.2.5 Δημιουργία props

Αφού ολοκληρωθεί ο φωτισμός και τα εφέ σειρά έχει η δημιουργία όλων αυτών των αντικειμένων που θα γεμίζουν την σκηνή και θα αλληλοεπιδρούν με τον παίχτη. Αυτά είναι πολλά αντικείμενα, όπως κρεβάτια, τραπέζια, κεριά, βιβλία κλπ.

Μεθοδολογία 34 Παράδειγμα prop #1

Μεθοδολογία 35 Παράδειγμα prop #2

Μεθοδολογία 36 Παράδειγμα prop #3

Μεθοδολογία 37 Παράδειγμα prop #4

Μεθοδολογία 38 Παράδειγμα prop #5

Μεθοδολογία 39 Παράδειγμα prop #6

Μεθοδολογία 40 Παράδειγμα prop #7

Μεθοδολογία 41 Παράδειγμα prop #8

Μεθοδολογία 42 Παράδειγμα prop #9

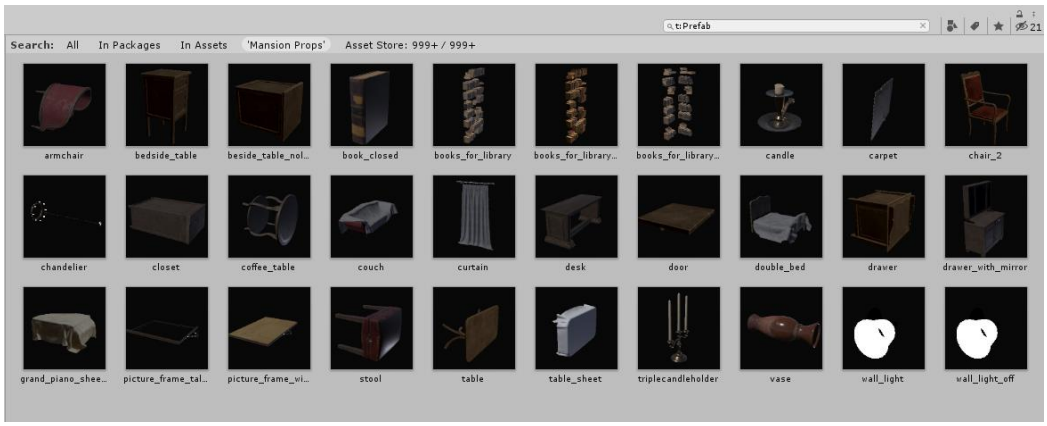
Μεθοδολογία 43 Παράδειγμα prop #10

Μεθοδολογία 44 Παράδειγμα prop #11

Μεθοδολογία 45 Παράδειγμα prop #12

10.2.6 Import props στην μηχανή παιχνιδιού

Αφού δημιουργηθούν όλα τα props καθώς και τα textures τους, πρέπει να εισαχθούν στο Unity. Όπως και για το level, έτσι για τα props, πρέπει να γίνει import το μοντέλο και τα textures. Μετά από αυτό πρέπει να δημιουργηθούν τα materials και να ενσωματωθούν στα μοντέλα. Αφού γίνει και αυτό, για κάθε μοντέλο πρέπει να δημιουργηθεί ένα prefab που θα το περιέχει. Prefab στο Unity είναι ένα αντικείμενο που περιέχει άλλα αντικείμενα όλα όμως έχουν την ίδια ρίζα. Αν δηλαδή γίνει μια αλλαγή στο prefab, όλα τα στιγμιότυπα αυτού του prefab στην σκηνή ενημερώνονται. Αυτό γλυτώνει τον δημιουργό από πολύ χρόνο αλλά είναι και πιο φθηνό σε πόρους.



Μεθοδολογία 46 Mansion Props

10.2.7 Δημιουργία ολοκληρωμένου level

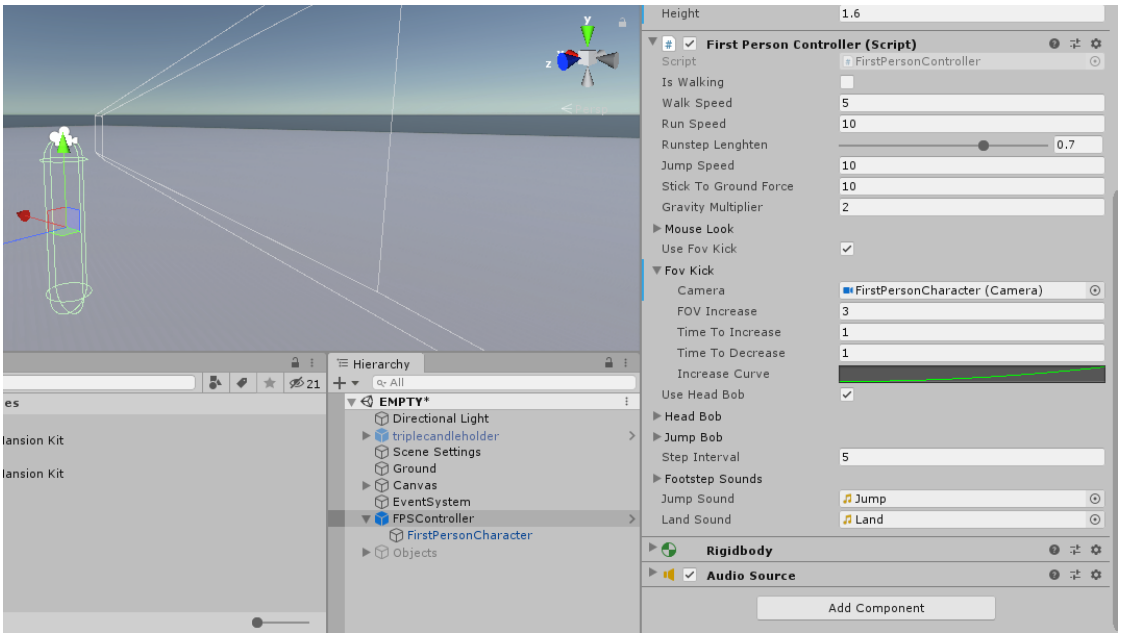
Αφού εισαχθούν όλα τα props στο Unity και γίνουν οι κατάλληλες ρυθμίσεις στο κάθε ένα, το επόμενο βήμα είναι να τοποθετηθούν στην σκηνή. Μετά από αυτό κάτι πολύ σημαντικό, είναι τα αντικείμενα που δεν πρόκειται να κουνηθούν, όπως καρέκλες, κουρτίνες κλπ. να σημειωθούν ως «static», έτσι ώστε να συμβάλλουν στο Global Illumination όταν ο φωτισμός γίνει baked.

Μεθοδολογία 47 Props inserted to level

10.2.8 Δημιουργία player, scripts και animations

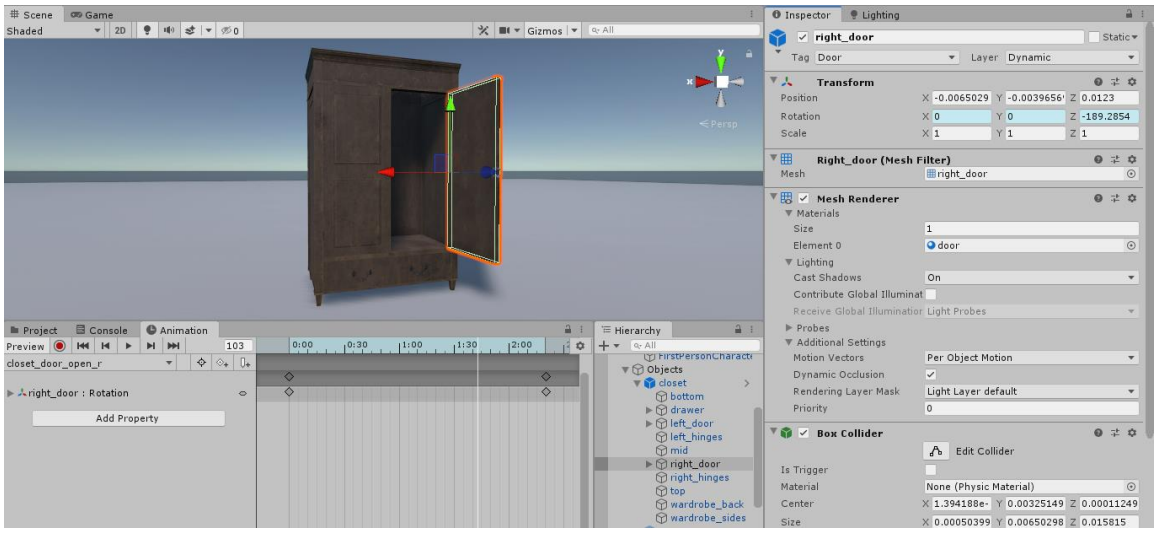
Εφόσον έχει ολοκληρωθεί το επίπεδο, το επόμενο βήμα ήταν η δημιουργία του παίχτη και το πως θα κινείται στον χώρο, μετά η δημιουργία των animations των αντικειμένων, για παράδειγμα το άνοιγμα μιας πόρτας και τέλος τα Scripts που χρειάζονται και καθορίζουν το πότε θα παίζονται αυτά τα animations αλλά και άλλες λειτουργίες του παιχνιδιού. Ωστόσο μιας και το βασικό θέμα της πτυχιακής εργασία ήταν η υλοποίηση των γραφικών και του level και όχι το gameplay, δεν δόθηκε μεγάλη έμφαση σε αυτό το κομμάτι.

Αρχικά για τον παίχτη χρησιμοποιήθηκε το default FPS Controller του Unity. Το βρίσκει κανείς στο Asset Store στο Standard Assets από την Unity Technologies. Λίγες ήταν οι ρυθμίσεις που αλλάχτηκαν από αυτό μιας και οι εργοστασιακές του ρυθμίσεις ήταν κατάλληλες για το επιθυμητό αποτέλεσμα. Το μόνο που έπρεπε να προσαρμοστεί ήταν η ταχύτητα του παίχτη το ύψος του και η δύναμη που πηδούσε. Αυτό έγινε πολύ εύκολα αλλάζοντας τις τιμές στον Inspector.



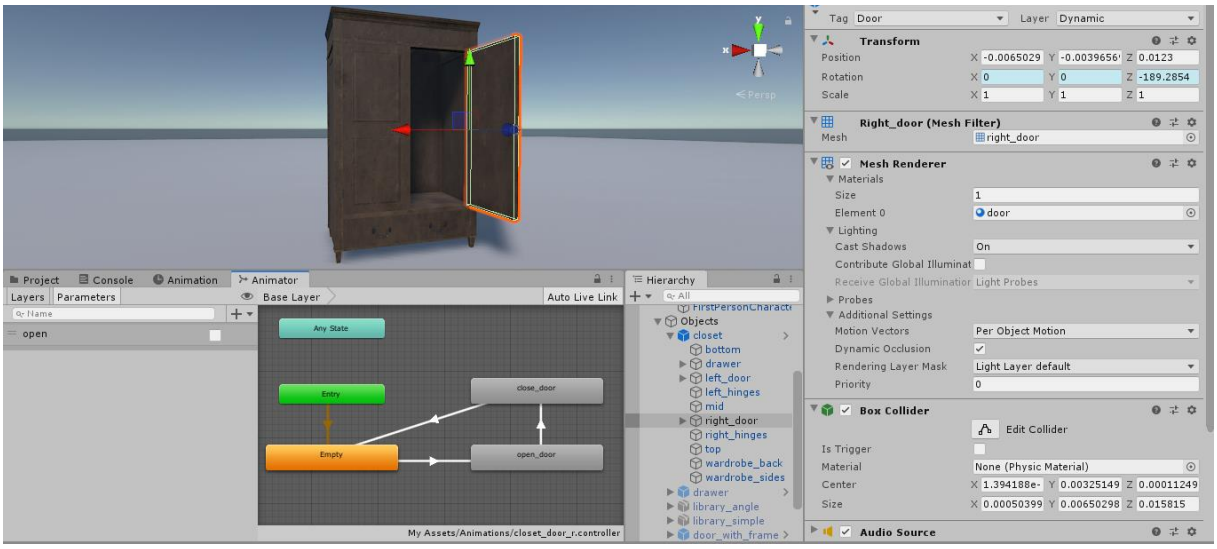
Μεθοδολογία 48 Unity Default FPS Controller

Όσο αφορά την δημιουργία των animation, ήταν κάτι σχετικά απλό αφού δεν είχε να κάνει με περίπλοκες κινήσεις χαρακτήρων αλλά απλές μετακινήσεις και περιστροφές αντικειμένων. Έτσι ως παράδειγμα την ντουλάπα, εφόσον κατά την μοντελοποίηση είχε οριστεί ως σημείο προέλευσης η άκρη της πόρτας και με την χρήση του συστήματος Animation του Unity έγινε ένα απλό rotation 120 μοιρών.



Μεθοδολογία 49 Simple door animation

Αφού δημιουργήθηκαν όλα τα animations για κάθε αντικείμενο έπρεπε να οριστεί πότε θα παιζόταν αυτά τα animation. Τι θα έπρεπε να κάνει δηλαδή ο παίχτης για να γίνει trigger το animation. Αποφασίστηκε ότι αυτό θα γινόταν αν ο παίχτης ήταν αρκετά κοντά στο αντικείμενο και πατούσε αριστερό κλικ στο ποντίκι. Αυτό υλοποιήθηκε με την χρήση του Animator και ενός απλού script.



Μεθοδολογία 50 Open-Close Door Animator

Ο συγκεκριμένος Animator λειτουργεί ως εξής. Όταν μία παράμετρος τύπου Boolean γίνει true, τότε η πόρτα ανοίγει, εφόσον είναι κλειστή, εάν αυτή είναι ανοιχτή και η παράμετρος γίνει false τότε κλείνει και η κατάσταση γίνεται πάλι idle. Η παράμετρος θα αλλάζει από τα εξής scripts.

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

public class PlayerActions : MonoBehaviour
{

    RaycastHit hit;

    void Update()
    {
        int layerMask = 1 << 9;

        if (Input.GetMouseButtonDown(0))
        {
            if(Physics.Raycast(transform.position, transform.forward, out hit, 1.75f,
layerMask))
            {
                if (hit.collider.gameObject.CompareTag("Door"))
                {

hit.collider.transform.gameObject.GetComponent<open_close>().OpenClose();
                }
            }
        }

    }
}
```

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

public class open_close : MonoBehaviour
{

    private Animator _animator;
    private AudioSource _audioSource;
    public bool _isOpen, _isLocked;
    public AudioClip open, close, locked;
    public Text message;

    private void Awake()
    {
        _animator = gameObject.GetComponent<Animator>();
        _audioSource = gameObject.GetComponent<AudioSource>();
        _isOpen = false;
    }

    public void OpenClose()
    {
        if (!_isLocked)
        {
            _isOpen = !_isOpen;
            _animator.SetBool("open", _isOpen);
            if (!_isOpen)
            {
                _audioSource.clip = close;
            }
            else
            {
                _audioSource.clip = open;
            }
        }
        else
        {
            _audioSource.clip = locked;
            StartCoroutine(writeMsg("Oh! It's locked!", 0));
        }
        _audioSource.Play(0);
    }

    private IEnumerator writeMsg(string msg, float waitTime)
    {
            yield return new WaitForSeconds(waitTime);
            message.text = msg;
            yield return new WaitForSeconds(3);
            message.text = "";
    }
}
```

Το πρώτο Script εισάγεται στον παίχτη και κάνει τρία πράγματα, αρχικά διαβάζει εάν ο παίχτης πάτησε αριστερό κλικ, εφόσον έγινε αυτό ρίχνει μια ακτίνα από το κέντρο της κάμερας και σε απόσταση 1.75 (μονάδες unity). Αν αυτή η ακτίνα χτυπήσει collider όπου το αντικείμενο

έχει οριστεί ως Dynamic και έχει ετικέτα Door τότε εκτελείται η συνάρτηση OpenClose(); του αντικειμένου αυτού. Αυτή η εντολή υπάρχει στο δεύτερο Script που έχει ενσωματωθεί σε κάθε δυναμικό αντικείμενο.

Το δεύτερο script αναλαμβάνει να ανοίξει (trigger το animation) την πόρτα αν αυτή είναι κλειστή και ξεκλείδωτη και να παίξει τον κατάλληλο ήχο καθώς και να την κλείσει αν αυτή είναι ανοιχτή. Αναλαμβάνει επίσης να δείξει στον παίχτη το κατάλληλο μήνυμα, αν για παράδειγμα είναι κλειδωμένη.

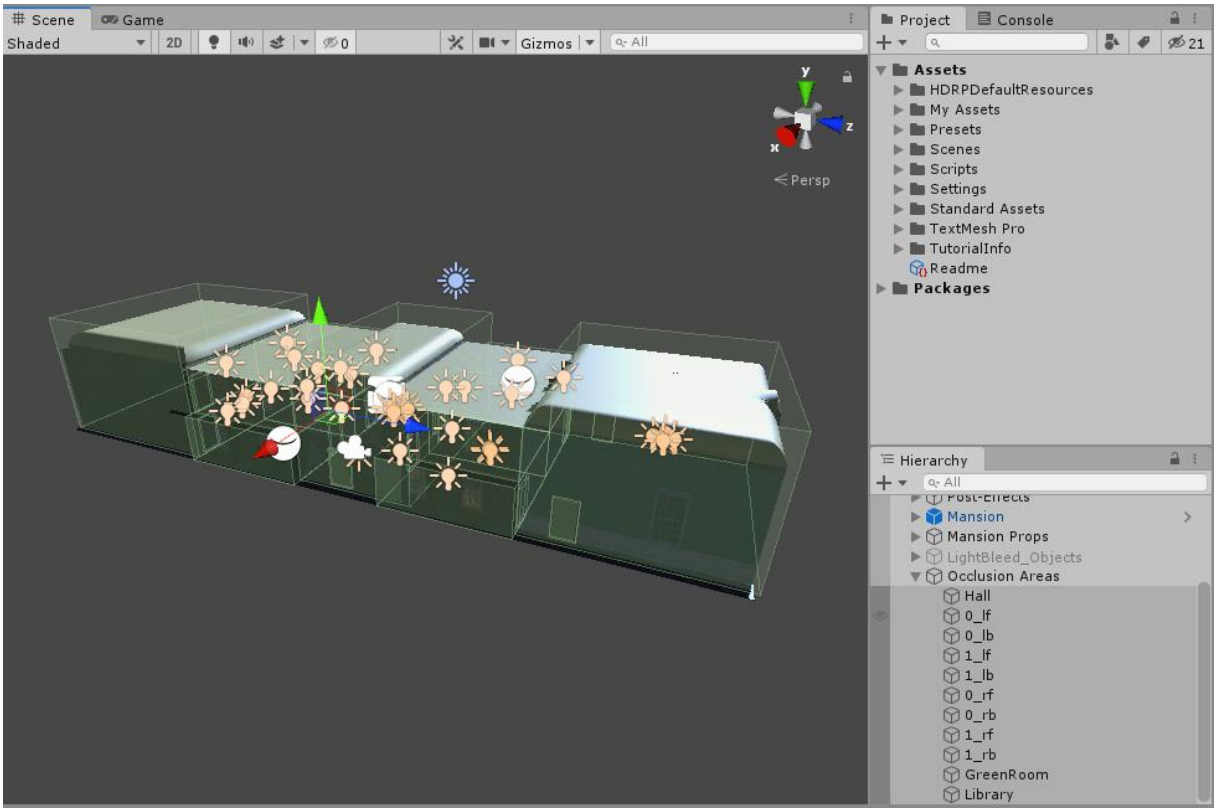
Ένα ακόμα script που δημιουργήθηκε ήταν αυτό που θα δείχνει στον παίχτη αν είναι αρκετά κοντά σε ένα δυναμικό αντικείμενο. Αυτό έγινε ρίχνοντας ξανά μια ακτίνα προς την ίδια κατεύθυνση και απόσταση και αλλάζοντας το χρώμα ενός μικρού δείκτη που είναι πάντα στο κέντρο της οθόνης.

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

public class pointerFeedback : MonoBehaviour
{
    private RaycastHit hit;
    public Camera cam;


    void Update()
    {
        int layerMask = 1 << 9;

        if (Physics.Raycast(cam.transform.position, cam.transform.forward, out hit,
1.75f, layerMask))
        {
            gameObject.GetComponent<Image>().color = new Color32(255, 185, 26, 100);
        }
        else
        {
            gameObject.GetComponent<Image>().color = Color.white;
        }
    }
}
```

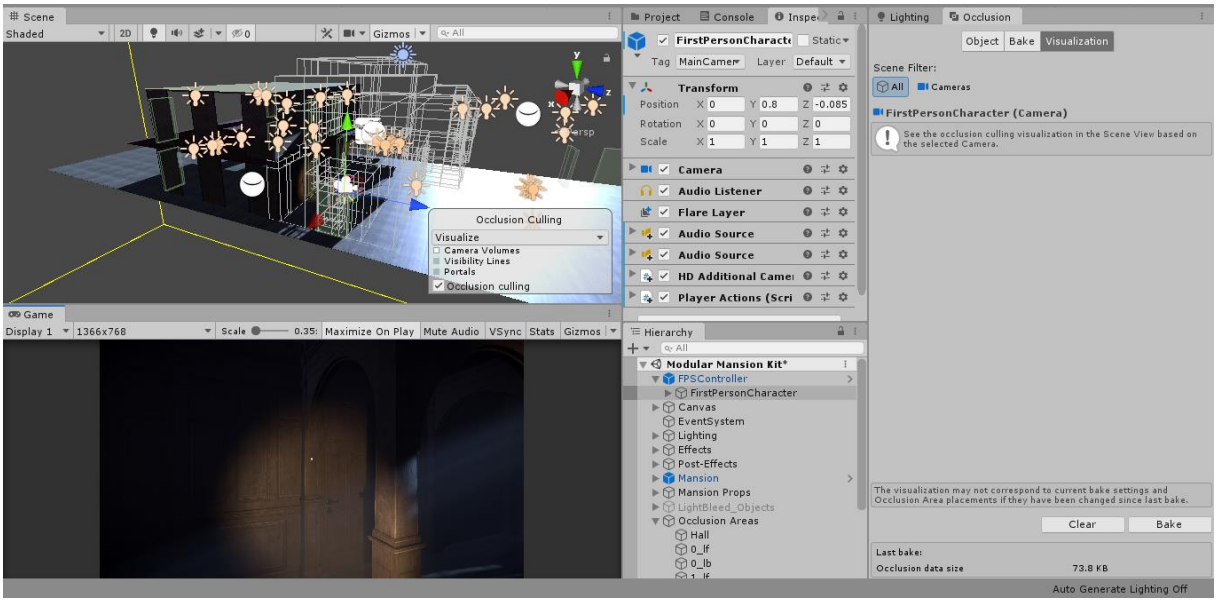
10.2.9 Βελτιστοποίηση παιχνιδιού

Το τελευταίο βήμα που έγινε ήταν να περιοριστούν τα γραφικά που γίνονται render κάθε frame. Αυτό έγινε με την χρήση του Occlusion Culling. Για να επιτευχθεί έπρεπε να τοποθετηθούν Occlusion Areas σε κάθε δωμάτιο καθώς και Occlusion Portals σε κάθε πόρτα αφού ήταν επιθυμητό να μην φαίνονται τα αντικείμενα πίσω από την πόρτα όταν αυτή είναι κλειστή και να είναι ορατά όταν είναι ανοιχτή. Αφού τοποθετήθηκαν κατάλληλα τα παραπάνω προστέθηκαν δύο μικρές συναρτήσεις στον κώδικα της πόρτας, Αυτές οι συναρτήσεις εκτελούνται όταν αρχίζει το animation που ανοίγει η πόρτα και όταν τελειώνει το animation που κλείνει την πόρτα. Η χρήση του είναι για να αλλάζει την τιμή του Boolean “open” που

περιέχει το Occlusion Portal Component και καθορίζει αν το portal θεωρείται ανοιχτό η κλειστό.



Παράδειγμα 57 Occlusion Areas & Portals in Unity



Παράδειγμα 58 Occlusion Culling Visualization

11 Αποτελέσματα

11.1 Το παιχνίδι

Το παιχνίδι που υλοποιήθηκε περιέχει ένα πλήρες επίπεδο όσο αφορά τα γραφικά. Παράχθηκε ένα mansion τριών μεγάλων δωματίων και 8 μικρότερων. Παράχθηκαν αρκετά props για να γεμίσει το επίπεδο όπου αρκετά εξ αυτών είναι dynamic, συνεπώς ο παίχτης μπορεί να αλληλεπιδράσει με αυτά. Υλοποιήθηκε ο απαραίτητος φωτισμός για το θέμα που ήταν επιθυμητό να υλοποιηθεί καθώς και τα απαραίτητα εφέ για τον ίδιο λόγο. Υλοποιήθηκε ένα πλήρες σύστημα κίνησης παίχτη καθώς και δυνατότητες αλληλεπίδρασης με δυναμικά αντικείμενα. Προστέθηκαν ορισμένοι ήχοι που δίνουν μια πιο ωραία εμπειρία στον χρήστη καθώς είναι κάτι που επίσης βοηθάει στην παρουσίαση του θέματος του παιχνιδιού. Έχει επιτευχθεί ένα σεβαστό επίπεδο ρεαλισμού.

Gameplay 1 Hall

Gameplay 2 Hall from Top

Gameplay 3 Room #1

Gameplay 4 Room #2

Gameplay 5 Library

Gameplay 6 Green Room

Gameplay 7 Music Room

11.2 Δυσκολίες κατά την παραγωγή

Υπήρξαν αρκετές δυσκολίες κατά την παραγωγή των γραφικών και του παιχνιδιού. Η πιο βασική δυσκολία είχε να κάνει με την εξοικονόμηση των πόρων, πράγμα που οδήγησε σε ένα βαρύ παιχνίδι που χρειάζεται ένα αρκετά δυνατό σύστημα για να τρέξει. Αυτό συνέβη λόγο απειρίας καθώς δεν είχα δουλέψει ξανά σε μια τόσο μεγάλη σκηνή. Άλλες δυσκολίες που προέκυψαν ήταν κατά την μοντελοποίηση πολύ λεπτομερών επιφανειών (π.χ. κομψοί καθρέπτες που συναντάει κανείς σε βίλες). Αυτό συνέβη λόγω έλλειψής εξάσκησης σε τέτοιου είδους επιφάνειες καθώς απαιτούν πολύ χρόνο για να μπορεί κανείς να πετύχει ένα καλό αποτέλεσμα. Αρκετά προβλήματα επίσης προέκυψαν κατά την δημιουργία των UV Maps ορισμένων πολύπλοκων μοντέλων πράγμα που δημιούργησε αλλοιώσεις κατά την διαδικασία bake του φωτισμού.

11.3 Πιθανές μελλοντικές εξελίξεις και διορθώσεις

Μιας και ο βασικός στόχος της πτυχιακής εργασίας ήταν η παραγωγή υψηλής ποιότητας γραφικών, μια πιθανή εξέλιξη της εφαρμογής θα μπορούσε να είναι η ανάπτυξη ενός κανονικού παιχνιδιού με βάση το επίπεδο που αναπτύχθηκε και το θέμα του παιχνιδιού. Αυτό θα μπορούσε να συμβεί με την υλοποίηση αποστολών, στόχων αλλά και την προσθήκη άλλων χαρακτήρων που θα λειτουργούσαν ως εμπόδια για την επίτευξη του τελικού στόχου.

Κάποιες διορθώσεις που θα μπορούσαν να γίνουν είναι η βελτιστοποίηση κάποιων μοντέλων τα οποία έχουν μεγάλο αριθμό σημείων πράγμα που κάνουν αρκετά βαρύ το παιχνίδι αλλά και η δημιουργία trim textures για την αποφυγή πολλών μεγάλων texture που επίσης βαραίνουν το παιχνίδι.

12 Βιβλιογραφία

Έργα που αναφέρονται

3D Μοντελοποίηση. (χ.χ.). Ανάκτηση από Wikipedia: https://el.wikipedia.org/wiki/3D_%CE%BC%CE%BF%CE%BD%CF%84%CE%B5%CE%BB%CE%BF%CF%80%CE%BF%CE%AF%CE%B7%CF%83%CE%B7

Blender Documentation. (χ.χ.). Ανάκτηση από Blender: https://docs.blender.org/manual/en/latest/

Nelson, S. (χ.χ.). *Using Photoshop to Create Art for a Mobile Game*. Ανάκτηση από Peachpit: http://www.peachpit.com/articles/article.aspx?p=2245728&seqNum=5

Unity Documentation. (χ.χ.). Ανάκτηση από Unity : https://docs.unity3d.com/Manual/index.html

What to know when Creating next Gen Assets. (χ.χ.). Ανάκτηση από CG Masters: https://cgmasters.net/free-tutorials/what-to-know-when-creating-next-gen-assets/

Γραφικά Υπολογιστών. (χ.χ.). Ανάκτηση από Wikipedia: https://el.wikipedia.org/wiki/%CE%93%CF%81%CE%B1%CF%86%CE%B9%CE%BA%CE%AC_%CF%85%CF%80%CE%BF%CE%BB%CE%BF%CE%B3%CE%B9%CF%83%CF%84%CF%8E%CE%BD

Images

https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-shading/shading-normals
https://docs.blender.org/manual/en/latest/modeling/meshes/primitives.html
http://help.autodesk.com/view/MOBPRO/2017/ENU/?guid=GUID-F90A9BF3-1A41-4FB7-AE58-53D73BDDEF6B
https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/subdivision_surface.html
https://www.youtube.com/watch?v=RDKt_lNAHss
https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/bevel.html
https://spolearninglab.com/curriculum/software/3d_modeling/blender/blender_svg.html
http://thilakanathanstudios.com/2016/09/why-do-we-need-topology-in-3d-modeling/
https://www.textures.com/download/3dbrush0100/133294
https://polycount.com/discussion/83605/frustrated-and-jealous-rock-sculpting-help
https://www.cleanpng.com/png-uv-mapping-texture-mapping-cube-3d-modeling-cube-1181894/
https://www.youtube.com/watch?v=8VgLlurusWc
https://blendermarket.com/products/friendly-shade-bundle-01
https://www.filterforge.com/filters/10500-ambientocclusion.html
https://learnopengl.com/Advanced-Lighting/Normal-Mapping
https://kcclemo.neocities.org/creating-height-and-normal-maps/
https://blog.backspinestudios.com/from-specular-to-metallic/
https://texturify.com/category/environment-panoramas.html

https://blogs.unity3d.com/2018/02/27/introduction-to-shader-graph-build-your-shaders-with-a-visual-editor/
https://www.google.com/search?biw=1366&bih=635&tbm=isch&sa=1&ei=XqVuXZOBMJ-HjLsPwYKBiA4&q=vertex+shader&oq=verte&gs_l=img.3.0.35i39j0i67l2j0j0i67j0j0i67j0l3.17746.18505..19033...0.0..0.143.672.0j5.....0....1..gws-wiz-img.MWguVYRw9go#imgrc=Bw0U2kDlwL-SjM:
https://www.youtube.com/watch?v=IzlU_xvTK3Y
http://www.reynantemartinez.com/cycles-material-studies.html
https://blogs.unity3d.com/2018/12/19/unity-2018-3-shader-graph-update-lit-master-node/
https://docs.unity3d.com/Manual/Lighting.html
http://www.aoktar.com/octane/OCTANE%20HELP%20MANUAL.html?MakingSpotLight.html
https://docs.unity3d.com/Manual/UsingLights.html
https://unity3d.com/kr/learn/tutorials/topics/graphics/emissive-materials?playlist=48370
https://www.thegnomonworkshop.com/tutorials/environment-lighting-for-production
https://github.com/SlightlyMad/VolumetricLights
https://forum.unity.com/threads/unity-5-baked-point-light.326668/
https://imgur.com/gallery/5iE3m
https://docs.unity3d.com/540/Documentation/Manual/LightProbes.html
https://unity3d.com/ru/learn/tutorials/topics/graphics/reflections
https://www.liveenhanced.com/3d-rendering-techniques-an-overview-of-various-techniques/
https://marmoset.co/posts/pbr-texture-conversion/
https://dev.rbcafe.com/unity/unity-3.4.2/Documentation/Manual/Occlusion%20Culling.html
http://www.tanukidigital.com/tenkoku/
https://gr.pinterest.com/pin/534802524499540916/?lp=true
https://www.youtube.com/watch?v=IziIY674NAw&t=70s
http://www.shapesandlines.com/generic-gallery-test02/low-poly-game-props-004/
https://www.coroflot.com/nigeqb/Game-Props
https://80.lv/articles/best-vfx-for-unity/
https://forum.kerbalspaceprogram.com/index.php?/topic/168763-17x-ks3p/
https://www.toolfarm.com/tutorial/substance_painter_texturing_for_beginners/
http://bensherra.info/old-wardrobe-closet/
https://furniture123.co.uk/charles-2-door-sliding-wardrobe-in-oak_34700
https://www.mountvernon.org/the-estate-gardens/the-mansion/the-mansion-room-by-room/
https://cellcode.us/quotes/victorian-manor-hallway.html
https://www.textures.com/search?q=books