

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ



Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Πτυχιακή διατριβή

**Ανάπτυξη εφαρμογής προσομοίωσης εικονικού
μουσείου σε τριδιάστατο περιβάλλον με στοιχεία
adventure game**

Μπολωνάκης Κων/νος

Επιβλέπων Καθηγητής

κ. Μαλάμος Αθανάσιος

Ηράκλειο 2012

Ευχαριστίες

Κατά την διάρκεια υλοποίησης αυτής της προσπάθειας έλαβα στήριξη και βοήθεια από διάφορα πρόσωπα, τα οποία θα ήθελα να ευχαριστήσω με την σειρά μου και να δηλώσω την εκτίμησή μου στον κάθε ένα ξεχωριστά. Θα ήθελα να εκφράσω της ευχαριστίες μου στον καθηγητή κ. Αθανάσιο Μαλάμο για την καθοδήγηση και την συμβολή του σε αυτό το εγχείρημα. Επίσης θα ήθελα να ευχαριστήσω το οικογενειακό και φιλικό μου περιβάλλον, για την στήριξη και την βοήθεια που έλαβα σε διάφορους τομείς, όπου ο κάθε ένας με τον τρόπο του συνέβαλε στην ολοκλήρωση αυτής της εργασίας.

Abstract

Goal of this project is to observe step by step the development of an application by using principal programming tools such as 3D Studio Max and Unity 3D. The development of this application that is being analyzed in the following pages, is the simulation of a virtual museum, placed in a 3D environment, through which we work in modeling, the composition and the programming that is required in order to be created the final working file.

Initially, the presentation of this application starts from the description of the controlling and the control settings of the application. A tour takes place in the museum, as the abilities and the operation offered, are described.

Then, we examine in details tools and methods that has been used starting with 3d studio max program, through of which we present the 3d models creation and learn the abilities offered by this program through examples such as building, character and item modeling. It is also mentioned the unwrapping techniques for textures adjustment in models and the process of creation animation.

Concluding, we present the way through of which is being developed a 3d environment using Unity. We examine the introduction and the manipulation of the models in the Unity environment and how they can gain interactivity via Scripting, using JavaScript and C#. What is more, we mention the main programming tools and all the process needed to reach the final result. Finally, having understand the basic techniques that we follow in order to build a 3d application, we mention some of the tips for the improvement of the project, who it can be more realistic and how we can save system resources.

Σύνοψη

Στόχος αυτού του συγγράμματος είναι να παρατηρήσουμε βήμα προς βήμα την ανάπτυξη μιας εφαρμογής χρησιμοποιώντας σαν κύρια προγραμματιστικά εργαλεία το **3D Studio Max** (πρόγραμμα δημιουργίας τρισδιάστατων γραφικών και animation) και το **Unity 3D** (πρόγραμμα δημιουργίας εφαρμογών τριών διαστάσεων). Η Ανάπτυξη της εφαρμογής που αναλύουμε στις παρακάτω σελίδες, είναι μια προσομοίωση ενός εικονικού μουσείου σε τρισδιάστατο περιβάλλον, μέσα απ'το οποίο ασχολούμαστε με την μοντελοποίηση, την συνθεσή και τον προγραμματισμό που απαιτείται, για να μπορέσει να δημιουργηθεί το τελικό εκτελέσιμο αρχείο της εφαρμογής.

Αρχικά, γίνεται η παρουσίαση της εφαρμογής ξεκινώντας από την περιγραφή του χειρισμού και των στοιχείων ελέγχου της εφαρμογής. Γίνεται μια ξενάγηση στους χώρους του μουσείου, καθώς περιγράφονται οι δυνατότητες και οι λειτουργίες που παρέχονται.

Στην συνέχεια εξετάζονται αναλυτικά τα εργαλεία και οι τεχνικές που χρησιμοποιήθηκαν ξεκινώντας από το προγραμματιστικό εργαλείο 3d studio Max, μέσα απ'όπου παρουσιάζονται μέθοδοι κατασκευής τρισδιάστατων μοντέλων και βλέπουμε δυνατότητες που μας προσφέρει αυτό το πρόγραμμα με παραδείγματα όπως μοντελοποίηση κτιρίων, χαρακτήρων και αντικείμενων. Αναφερόμαστε επίσης στην τεχνική **unwrapping** για την δημιουργία και την προσαρμογή υφών (textures) στα μοντέλα και την διαδικασία αναπτυξής **animation**.

Έπειτα, βλέπουμε τον τρόπο ανάπτυξης ενός 3d περιβάλλοντος με το προγραμματιστικό εργαλείο **Unity**. Εξετάζουμε την εισαγωγή και διαχείριση των 3d μοντέλων στο περιβάλλον του Unity και πως μπορούν να αποκτήσουν διαδραστικότητα μέσω του προγραμματισμού (Scripting), χρησιμοποιώντας **JavaScript** και **C#**. Αναφέρονται τα κυριότερα εργαλεία του προγράμματος και οι διεργασίες εκείνες που χρειάζονται για να φτάσουμε στο τελικό αποτέλεσμα. Τελος έχοντας κατανοήσει της βασικές τεχνικές και διαδικασίες τις οποίες ακολουθούμε για την κατασκευή μιας τρισδιάστατης εφαρμογής, αναφέρονται ορισμένες συμβουλές (tips), για την βελτίωση του project όσον αφορά την απόδοση ρεαλιστικότητας και εξοικονόμησης πόρων του συστήματος.

Περιεχόμενα

Εισαγωγή.....	13
Κεφάλαιο 1	Presentation
Παρουσίαση Εφαρμογής	20
1.1 Χειρισμός	21
1.2 Βασικό μενού.....	22
1.3 Κύριο μέρος της εφαρμογής.....	23
1.3.1 Ξενάγηση στους χώρους του μουσείου	23
1.3.2 Παράθυρο περιγραφής εκθεμάτων.....	26
Κεφάλαιο 2	3D Modeling
3D Studio Max.....	27
2.1 Εισαγωγή στο 3ds Max.....	27
2.2 Τα βασικά εργαλεία του 3ds Max.....	28
2.2.1 Main Toolbar	29
2.2.2 Command Panel	29
2.2.3 Modifier Editable Poly.....	31
2.2.4 TimeLine.....	34
2.2.5 FBX Export Screen.....	35
2.3 Διαδικασία μοντελοποίησης.....	36
2.3.1 Παράδειγμα κατασκευής ενός απλού κτιρίου από Boxes με την μέθοδο Polygon Modeling.....	37
2.3.2 Παράδειγμα κατασκευής ενός δοχείου απο Cylinders με την μέθοδο Polygon Modeling.....	43
2.3.3 Παράδειγμα κατασκευής 3D χαρακτήρα(avatar) με συνδιασμό μέθοδων Box Modeling, Spline Modeling και Polygon Modeling..	48
2.3.3.1 Μοντελοποίηση ανθρωπίνου σώματος με την μέθοδο Box Modeling και Polygon Modeling	48
2.3.3.2 Μοντελοποίηση ανθρώπινου κεφαλιού με την μέθοδο Spline Modeling.....	54
2.4 Προσαρμογή Υφής	58
2.4.1 Modifier Unwrap UVW.....	59
2.4.2 Χρήση της μεθόδου Unwrap UVW	61

2.5 Clothing.....	65
2.5.1 Modifier Cloth	66
2.5.2 Ανάπτυξη ρουχισμού με Cloth modifier	68
2.6 Animation.....	72
2.6.1 Bones και Biped Systems	73
2.6.2 Modifier Skin	74
2.6.3 Προσαρμογή Biped system σε avatar με χρήση του Skin Modifier.....	76
2.6.4 Δημιουργία Animation	78
2.7 Πορεία μοντελοποίησης των στοιχείων της εφαρμογής μέσω του 3DS Max.....	80
2.7.1 Museum Building.....	80
2.7.1.1 Modeling.....	80
2.7.1.2 Textures.....	84
2.7.2 Exhibits(Εκθέματα).....	85
2.7.2.1 Modeling.....	85
2.7.2.2 Textures.....	87

Καφάλαιο 3.....3D Application Building

Unity 3D	89
3.1 Εισαγωγή στο Unity 3D	89
3.2 Το περιβάλλον εργασίας του Unity.....	90
3.3 Ανάπτυξη περιβάλλοντα χώρου.....	91
3.3.1 Terrain.....	92
3.3.2 SkyBox.....	93
3.3.3 Lighting.....	94
3.4 Εισαγωγή και διαχείριση FBX μοντέλων.....	96
3.5 Προβολή και διαδραστικότητα εφαρμογής.....	98
3.5.1 Camera	99
3.5.2 Scripting.....	100
3.5.2.1 Προσαρμογή έτοιμου κώδικα σε αντικείμενο	101
3.5.2.2 Δημιουργία νέου αρχείου κώδικα.....	103

3.6 Πορεία ανάπτυξης της εφαρμογής μέσα από το Unity	108
3.6.1 Δημιουργία νέου Project.....	108
3.6.2 Δημιουργία περιβάλλοντος.....	109
3.6.3 Εισαγωγή μοντέλων.....	110
3.6.4 Εισαγωγή στοιχείων φωτισμού.....	112
3.6.5 Απεικόνιση κίνησης και διαχείριση animation.....	114
3.6.7 Αλληλεπίδραση μεταξύ μοντέλων	117
3.6.7.1 Εμφάνιση περιγραφής εκθεμάτων	117
3.6.7.2 Λειτουργία ανελκυστήρα.....	119
3.7 Tips.....	123
3.7.1 Χρήση Low Poly μοντέλων	123
3.7.2 Χρήση Light Maps.....	124
3.7.3 Εξοικονόμηση όγκου δεδομένων από την χρήση textures	124
Επίλογος.....	125
Βιβλιογραφία.....	126

Πίνακας εικόνων

Κεφάλαιο 1

Εικόνα 1.1	Το Pause Menu της εφαρμογής.....	21
Εικόνα 1.2	Στιγμιότυπα με τις επιλογές του Main Menu.....	22
Εικόνα 1.3	Το παράθυρο Controls και η οθόνη του Loading.....	22
Εικόνα 1.4	Στιγμιότυπα απο τον προάβλιο χώρο του μουσείου.....	23
Εικόνα 1.5	Στιγμιότυπα απο τον προθάλαμο και το wc του μουσείου	23
Εικόνα 1.6	Η κεντρική αίθουσα του μουσείου	24
Εικόνα 1.7	Στιγμιότυπα απο κάποια εκθέματα του μουσείου.....	24
Εικόνα 1.8	Ο ανελκυστήρας στον κεντρικό διαδρομο	25
Εικόνα 1.9	Στιγμιότυπα απο την αίθουσα προσομείωσης, το δωμάτιο με τα δελφίνια και το δωμάτιο του θρόνου.....	25
Εικόνα 1.10	Στιγμιότυπα απο τον πρώτο όροφο	25
Εικόνα 1.11	Η οθόνη περιγραφής των εκθεμάτων	26

Κεφάλαιο 2

Εικόνα 2.1	Περιβάλλον εργασίας 3ds Max2010.....	28
Εικόνα 2.2	Χρήση του “Use NURM Subdivision”.....	34
Εικόνα 2.3	Απεικόνιση κτιρίου σε τρισδιάστατο περιβάλλον.....	42
Εικόνα 2.4	Απεικόνιση δοχείου σε τρισδιάστατο περιβάλλον.....	47
Εικόνα 2.5	Απεικόνιση avatar σε τρισδιάστατο περιβάλλον.....	57
Εικόνα 2.6	Διαδικασία προσαρμογής υφής σε 3d model	58
Εικόνα 2.7	Δημιουργία τομών πάνω στο 3d model κατά την διαδικασία Point To Point Seam	62
Εικόνα 2.8	Διαδικασία unwrapping με χρήση του εργαλείου Pelt.....	62
Εικόνα 2.9	Τοποθέτηση πλεγμάτων στο UVW Panel και δημιουργία Texture Map.....	63
Εικόνα 2.10	Δημιουργία υφής στο Photoshop με χρήση Texture Map.....	64
Εικόνα 2.11	Αποτέλεσμα προσαρμογής υφής στο μοντέλο	64
Εικόνα 2.12	Προσομοίωση υφάσματος σε 3d περιβάλλον.....	65
Εικόνα 2.13	Σχεδιασμός ρούχων με χρήση Splines	68
Εικόνα 2.14	Σύνδεση Panel μπροστινής και πίσω όψης με χρήση Seams.....	69
Εικόνα 2.15	Διαμόρφωση T-Shirt πάνω στο 3d Model	70
Εικόνα 2.16	Τοποθέτηση πλεγμάτων του T-Shirt στο UVW Panel.....	71
Εικόνα 2.17	Τελικό αποτέλεσμα δημιουργίας και εφαρμογής ρούχων στο μοντέλο.....	71
Εικόνα 2.18	Ανάπτυξη Biped και Bone Systems.....	73
Εικόνα 2.19	Περιοχή και κλίμακα επιρροής των Envelopes με χρωματική απεικόνιση.....	74

Εικόνα 2.20	Δημιουργία Biped System	76
Εικόνα 2.21	Τροποποίηση και τοποθέτηση Biped συστήματος στο 3d model	76
Εικόνα 2.22	Προσκόλληση πολυγώνων στην περιοχή επιρροής του Envelope με την χρήση του Abs. Effect	77
Εικόνα 2.23	Χρήση του εργαλείου Footstep mode για προσομοίωση κίνησης τριών ειδών: Walk, Run, Jump.....	78
Εικόνα 2.24	Ακολουθία κινήσεων για την δημιουργία βηματισμού.....	79
Εικόνα 2.25	Πορεία ανάπτυξης μουσείου 1.....	80
Εικόνα 2.26	Πορεία ανάπτυξης σκάλας και διαζωμάτων.....	81
Εικόνα 2.27	Πορεία ανάπτυξης μουσείου 2.....	81
Εικόνα 2.28	Πορεία ανάπτυξης κολώνων μουσείου	82
Εικόνα 2.29	Πορεία ανάπτυξης μουσείου 3.....	83
Εικόνα 2.30	Πορεία ανάπτυξης μουσείου 4.....	83
Εικόνα 2.31	Πορεία ανάπτυξης μουσείου 5.....	83
Εικόνα 2.32	Πορεία ανάπτυξης μουσείου 6.....	84
Εικόνα 2.33	Χάρτες για την προσαρμογή υφών στο μουσείο (τοίχοι, 20πάτωμα και οροφή).....	84
Εικόνα 2.34	Όψεις μουσείου με προσαρμοσμένες υφές πάνω του.....	85
Εικόνα 2.35	Πορεία ανάπτυξης εκθέματος: Kamares Jug.....	85
Εικόνα 2.36	Πορεία ανάπτυξης εκθεμάτων: Lion Rhyton και Bull Rhyton	86
Εικόνα 2.37	Πορεία ανάπτυξης εκθεμάτων: Bees Pendant	86
Εικόνα 2.38	Πορεία ανάπτυξης εκθεμάτων: Snake Goddess και Goddess Terracotta.....	87
Εικόνα 2.39	Δημιουργία τομών πάνω στην επιφάνεια του εκθέματος.....	88
Εικόνα 2.40	Διαδικασία δημιουργίας και προσαρμογής υφών με την χρήση texture maps.....	88

Κεφάλαιο 3

Εικόνα 3.1	Περιβάλλον εργασίας Unity3	91
Εικόνα 3.2	Διαμόρφωση terrain	92
Εικόνα 3.3	Εφαρμογή του skybox στην σκηνή.....	93
Εικόνα 3.4	Απεικόνιση διαφόρων εφέ με τη χρήση Halo και Cookie	95
Εικόνα 3.5	Τοποθέτηση μοντέλου στην σκηνή	98
Εικόνα 3.6	Λήψεις κάμερας με τροποποίηση του Field of View	100
Εικόνα 3.7	Εμφάνιση στοιχείων GUI στην κάμερα.....	100
Εικόνα 3.8	Script για περιστροφή της κάμερας μέσα απο το mouse.....	102
Εικόνα 3.9	Script Editor	103

Εικόνα 3.10	Παράδειγμα διαφορών της C# και JavaScript.....	106
Εικόνα 3.11	Παραδείγματα χρήσης συνθικών if, for και while.....	106
Εικόνα 3.12	Παράδειγμα χρήσης κλάσης	106
Εικόνα 3.13	Παράθυρο Console του Unity	107
Εικόνα 3.14	Παράδειγμα τροποποίησης μεταβλητών σε αντικείμενα που χρησιμοποιούν τη ίδια συμπεριφορά.....	107
Εικόνα 3.15	Παράθυρο του Project Wizard για την δημιουργία νέου project.....	108
Εικόνα 3.16	Διαμόρφωση του terrain.....	109
Εικόνα 3.17	Χρήση εικόνων στην απεικόνιση του ουράνιου θόλου.....	110
Εικόνα 3.18	Εισαγωγή του μουσείου στη σκηνή	110
Εικόνα 3.19	Διαμόρφωση προαύλιου χώρου	111
Εικόνα 3.20	Δημιουργία δαπέδων και προσαρμογή υφών πάνω τους.....	111
Εικόνα 3.21	Εισαγωγή και τοποθέτηση εκθεμάτων στο εσωτερικό του μουσείου	111
Εικόνα 3.22	Εισαγωγή και τοποθέτηση του χαρακτήρα στη σκηνή.....	112
Εικόνα 3.23	Μοντέλα που δημιουργήθηκαν και χρησιμοποιήθηκαν για την απεικόνιση πηγών φωτισμού.....	112
Εικόνα 3.24	Δημιουργία φωτισμού σε εκθέματα.....	113
Εικόνα 3.25	Δημιουργία εφέ λάμπης του θόλου της κεντρικής αίθουσας.....	113
Εικόνα 3.26	Script για τον έλεγχο της κίνησης του χαρακτήρα.....	115
Εικόνα 3.27	Script για την διαχείριση των animation του χαρακτήρα.....	116
Εικόνα 3.28	Texture και ακολουθία εικόνων για την δημιουργία του παράθυρου περιγραφής των εκθεμάτων	117
Εικόνα 3.29	Script για την εμφάνιση του παράθυρου περιγραφής των εκθεμάτων.....	118
Εικόνα 3.30	Χρήση Cubes στις εισόδους και το εσωτερικό του elevator για τον έλεγχο λειτουργίας.....	119
Εικόνα 3.31	Script 1 για τον έλεγχο και λειτουργία του elevator.....	120
Εικόνα 3.32	Script 2 για τον έλεγχο και λειτουργία του elevator.....	121
Εικόνα 3.33	Script 3 για τον έλεγχο και λειτουργία του elevator.....	122
Εικόνα 3.34	Χρήση Normal,Alpha και Bump στην απεικόνιση των μαλλιών του avatar	123
Εικόνα 3.35	Χρήση LightMap για την απόδοση φωτεινότητας.....	124

Πίνακας σχημάτων

Κεφάλαιο 2

Σχήμα 2.1 Command Panel.....	30
Σχήμα 2.2 Modifier Panel	31
Σχήμα 2.3 Modifier Editable Poly Panel.....	32
Σχήμα 2.4 Vertex Tools Panel.....	32
Σχήμα 2.5 Edge Tools Panel	33
Σχήμα 2.6 Border Tools Panel.....	33
Σχήμα 2.7 Polygon Tools Panel	33
Σχήμα 2.8 Element Tools Panel.....	34
Σχήμα 2.9 Geometry Tools Panel	34
Σχήμα 2.10 Surface Parameters Panel.....	35
Σχήμα 2.11 Time Parameters Window	36
Σχήμα 2.12 FBX Export Window.....	36
Σχήμα 2.13 Εφαρμογή Modifier Unwrap UVW	61
Σχήμα 2.14 Unwrap UVW Parameters Panel.....	61
Σχήμα 2.15 Unwrap UVW Map Parameters Panel	61
Σχήμα 2.16 UVW Panel.....	62
Σχήμα 2.17 Εφαρμογή Modifier Cloth.....	67
Σχήμα 2.18 Simulation Tools Panel	67
Σχήμα 2.19 Object Properties Window.....	68
Σχήμα 2.20 Εφαρμογή Modifier Skin.....	76
Σχήμα 2.21 Skin Modifier Parameters	76
Σχήμα 2.22 Envelope και Weight Properties	76
Σχήμα 2.23 Εργαλεία του Footstep Mode.....	78

Κεφάλαιο 3

Σχήμα 3.1 Εργαλεία του terrain editor	93
Σχήμα 3.2 Παράμετροι του material Skybox.....	94
Σχήμα 3.3 Παράμετροι και εφαρμογή του Directional light.....	95
Σχήμα 3.4 Παράμετροι και εφαρμογή του Point light.....	95
Σχήμα 3.5 Παράμετροι και εφαρμογή του Spot light.....	96
Σχήμα 3.6 Παράμετροι των Lights για την δημιουργία σκιάσεων και αλλών εφέ	96
Σχήμα 3.7 Περιοχή “Project.....	97
Σχήμα 3.8 Περιοχή “Hierarchy.....	97
Σχήμα 3.9 Περιοχή “Inspector	97

Σχήμα 3.10	Περιοχή “Inspector” ²	98
Σχήμα 3.11	Παράμετροι κάμερας.....	100
Σχήμα 3.12	Εφαρμογή έτοιμου κώδικα για την περιστροφή της κάμερας.....	102
Σχήμα 3.13	Εφαρμογή του Draw Halo για την απεικόνιση εφέ λάμψης του ήλιου.....	110
Σχήμα 3.14	Παράμετροι του Character Controller	115

Εισαγωγή

Τα γραφικά λειτουργικά συστήματα έγιναν δημοφιλή στις αρχές της δεκαετίας του '90. Τότε, η ανάπτυξη γραφικών τρισδιάστατων εφαρμογών βρισκόταν σε πολύ πρώιμα στάδια, ενώ, η κυρία δυσκολία που αντιμετώπιζε αυτή η μέθοδος ήταν η μεγάλη υπολογιστική ισχύς που χρειαζόταν ένα σύστημα για να μπορέσει να αποτυπώσει τέτοιου είδους γραφικά στην οθόνη. Αυτό συμβαίνει γιατί κάθε 3D αντικείμενο συντίθεται από εκατοντάδες ή και χιλιάδες μικρά πολύγωνα που περιγράφουν τη δομή του και απαιτούνται πολλές μαθηματικές πράξεις για να μετατρέψουν το 3D κόσμο σε εικόνα που μπορεί να εμφανιστεί στην οθόνη του υπολογιστή. Αυτή η διαδικασία μετατροπής τρισδιάστατων γραφικών σε εικόνες δύο διαστάσεων ονομάζεται **Rendering**. Ωστόσο, η πίεση για περισσότερο ρεαλισμό, πιο λεπτομερή γραφικά και μεγαλύτερες ταχύτητες σε τέτοια προγράμματα, όπως είναι τα παιχνίδια δράσης, οι εξομοιωτές και τα σχεδιαστικά πακέτα, σημαίνει περισσότερη 3D εργασία σε μικρότερο χρόνο.

Σήμερα με την τεχνολογία που διαθέτουμε η απεικόνιση τρισδιάστατων γραφικών έχει φτάσει σε ένα αρκετά ικανοποιητικό επίπεδο. Η ψηφιακή απεικόνιση τρισδιάστατων γραφικών χρησιμοποιείται ευρέως στις μέρες μας. Γραφικά τριών διαστάσεων βρίσκουν εφαρμογή σε πολλούς τομείς, όπως είναι ανάπτυξη ηλεκτρονικών παιχνιδιών, δημιουργία ταινιών τρισδιάστατων γραφικών και κατασκευή εφαρμογών προσομοίωσης, ενώ εκτιμάται ότι τα 3D γραφικά θα αποτελέσουν το επόμενο στάδιο εξέλιξης στον χώρο της τεχνολογίας.

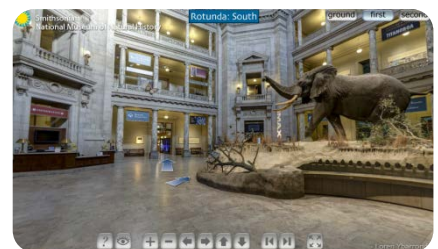
Τα “ζωντανά” χρώματα, τα εφέ και ο ρεαλισμός που προσφέρει σήμερα η τεχνολογία ψηφιακών γραφικών απεικονίσεων, καθιστούν τις τρισδιάστατες εφαρμογές αρκετά ελκυστικές προς το κοινό. Αυτή η τεχνολογία χρησιμοποιείται ήδη σε κάποιους τομείς με μεγάλη επιτυχία, για παράδειγμα στον τομέα της ψυχαγωγίας, τα τρισδιάστατα ηλεκτρονικά παιχνίδια, εμφανίζουν να έχουν μεγάλη απήχηση κυρίως στους νέους, που καταναλώνουν αρκετές ώρες ενασχόλησης. Εκτός από την ψυχαγωγία, τα τρισδιάστατα γραφικά μπορούν να χρησιμοποιηθούν και σε τομείς που αφορούν την παιδεία και τον τουρισμό.

Ποιο συγκεκριμένα οι τεχνολογίες που αφορούν τεχνικές προσομοίωσης θα μπορούσαν να συναντήσουν μεγάλο ενδιαφέρον σε εφαρμογές απεικόνισης χώρων, αντικειμένων, περιβάλλοντος και γενικότερα στοιχείων που προβάλουν και προάγουν τον πολιτισμό, την τέχνη, την κουλτούρα, την ιστορία, την φύση και την τεχνολογία, μιας χώρας. Αυτή η προσέγγιση θα μπορούσε βρει αντίκρισμα σε κοινό, μέσα από την παιδεία, προωθώντας αυτά τα στοιχεία στην διαδικασία της μάθησης και της εκπαίδευσης. Αντίστοιχα θα μπορούσε βρει αντίκρισμα σε κοινό που αφορά τον τουρισμό στα πλαίσια της προβολής αυτών των στοιχείων, συνεπώς και στην συμβολή του στην τόνωση της οικονομίας. Ένας τρόπος προβολής και προώθησης τέτοιων εφαρμογών είναι η διάθεση τους στο κοινό μέσα από το internet, όπου άνθρωποι από όλο τον κόσμο θα έχουν την δυνατότητα να επισκεφθούν εικονικά, από το χώρο του σπιτιού τους, μέσω ενός Η/Υ. Ένας άλλος τρόπος είναι η διάθεση τους στην αγορά και η διανομή τους μέσω CDs ή DVDs.

Σε αυτήν την προσέγγιση έχουν γίνει κάποιες προσπάθειες από διεθνείς οργανισμούς, οργανώσεις, συλλόγους ή άλλους παράγοντες, που αναπτύσσουν τρισδιάστατες εφαρμογές προσομοίωσης, όπου μέσα από αυτές προβάλλονται πληροφορίες σχετικά με μουσεία, μνημεία κτλ. Παρακάτω αναφέρονται κάποια links από όπου κάποιος μπορεί να επισκεφθεί ή να πληροφορηθεί για κάποιες από τις εφαρμογές που ήδη υπάρχουν:



<http://acropolis-virtualtour.gr/acropolisTour.html>



<http://www.mnh.si.edu/vtp/1-desktop/>



<http://www.tradky.com/m-3d-virtual-museums.html>

Μέσα στα πλαίσια που περιγράφηκαν παραπάνω κινήθηκε η προσπάθεια ανάπτυξης της παρούσας εφαρμογής. Στόχος ήταν η δημιουργία ενός τρισδιάστατου μουσείου που θα απεικόνιζε εκθέματα και πληροφορίες που αφορούν τον μινωικό και μυκηναϊκό πολιτισμό, όπου μέσα από την περιήγηση ο χρήστης θα μπορούσε να παρατηρήσει τα εκθέματα και συγχρόνως να έχει την δυνατότητα να λάβει πληροφορίες για αυτά, άλλα και γενικότερα για τον τρόπο ζωής των αρχαίων μινωικών και μυκηναίων. Ακόμα ένα βασικό χαρακτηριστικό της εφαρμογής θα αφορούσε την διαμόρφωση του περιβάλλοντος και τη χρήση αρκετών στοιχείων, ώστε να θυμίζει ηλεκτρονικό παιχνίδι.

Η δημιουργία ενός τρισδιάστατου και ατμοσφαιρικού περιβάλλοντος και η αναπαράσταση κάποιων μουσειακών εκθεμάτων του μινωικού και μυκηναϊκού πολιτισμού, θα μπορούσε να προκαλέσει το ενδιαφέρον του κοινού όλων των ηλικιών αφού συνδυάζει την γνώση, την διαδραστικότητα και τα χαρακτηριστικά που παραπέμπουν σε περιβάλλον σύγχρονου παιχνιδιού. Μέσα από ενέργειες σαν αυτήν θα μπορούσε να προβληθεί παγκοσμίως η πολιτιστική και ιστορική κληρονομία του τόπου μας, να συνεισφέρει στην εκπαίδευση και την γνώση ενώ παράλληλα να δημιουργήσει ένα ενδιαφέρον τεχνολογικού εκσυγχρονισμού σε τομείς που έχουν σχέση με την εκπαίδευση, τον πολιτισμό και τον τουρισμό.

Η ανάπτυξη της εφαρμογής απεικόνισης του μουσείου στο οποίο αναφέρεται το σύγγραμμα θα μπορούσε να χαρακτηριστεί πολύπλοκη και δυσνόητη. Για να μπορέσουμε να κατανοήσουμε καλύτερα την δημιουργία του τρισδιάστατου ψηφιακού κόσμου, πρέπει αρχικά να αναλυθεί στα επιμέρους στοιχεία από τα οποία αποτελείται. Στην συνέχεια περιγράφεται ή δομή των κεφαλαίων ανάλογα με τα στοιχεία που εξετάζει το κάθε ένα.

- **Terrain.** Θα μπορούσαμε να πούμε ότι είναι η πλατφόρμα όπου θα «στηθεί» το 3d περιβάλλον. Ανάλογα με το εργαλείο που θα χρησιμοποιηθεί για την δημιουργία μιας τρισδιάστατης εφαρμογής, το terrain μπορεί να τροποποιηθεί για την απεικόνιση βουνών, κοιλάδων, δρόμων, ποταμών κτλ. Με λίγα λόγια είναι ο κόσμος στον οποίον θα τοποθετηθούν όλα τα αντικείμενα (3d models).

- **3D Models.** Ένα τρισδιάστατο μοντέλο ή αλλιώς mesh είναι ένα σύνολο από πολύγωνα, όπου διαμορφώνονται κατάλληλα, ώστε να αποτελέσουν ένα αντικείμενο που θα τοποθετηθεί στην σκηνή. Ένα τέτοιο αντικείμενο μπορεί να είναι για παράδειγμα ένα αυτοκίνητο, ένα κτίριο, ένα δέντρο, όπως και ένα avatar (κινούμενος χαρακτήρας). Τα μοντέλα αυτά δημιουργούνται χρησιμοποιώντας τα κατάλληλα προγραμματιστικά εργαλεία, όπως είναι το **3d studio Max** και το **Blender**. Τα περισσότερα από αυτά τα προγράμματα προσφέρουν έτοιμα βασικά σχήματα (π.χ. Boxes, Spheres, Cylinders) τα οποία αποτελούν την βάση πάνω στα οποία τα μοντέλα, θα πάρουν την τελική τους μορφή μετά από επεξεργασία και τροποποίηση.
- **Textures (υφές).** Ένα τρισδιάστατο μοντέλο εκτός από την κατάλληλη μορφοποίηση με την διαδικασία της μοντελοποίησης χρειάζεται και μια υφή. Η υφή στο 3d modeling δεν είναι τίποτα άλλο από μια εικόνα (image) συνήθως τύπου png ή jpeg, όπου θα λέγαμε "τυλίγει" το μοντέλο δίνοντάς του την εμφάνιση και την λεπτομέρεια που χρειάζεται, προσφέροντας παράλληλα και μια πιο ρεαλιστική υπόσταση. Όσο πιο σύνθετο και περίπλοκο είναι ένα μοντέλο (mesh) τόσο πιο μεγάλη λεπτομέρεια χρειάζεται κατά την απεικόνιση. Η δημιουργία ενός texture μπορεί να επιτευχθεί με κατάλληλα προγραμματιστικά εργαλεία επεξεργασίας εικόνας όπως είναι το **Adobe Photoshop** και το **Adobe Fireworks**.
- **Lighting.** Μια τρισδιάστατη απεικόνιση χρειάζεται κατάλληλο φωτισμό για να μπορεί να προσφέρει την αίσθηση ρεαλιστικότητας. Ο φωτισμός παίζει μεγάλο ρόλο σε αυτές τις περιπτώσεις ώστε να μπορεί να αποτυπώσει τις κατάλληλες σκιάσεις στα αντικείμενα έτσι ώστε να έχουμε την ψευδαίσθηση της πραγματικότητας. Στα περισσότερα προγραμματιστικά εργαλεία παρέχονται τρεις κύριοι τύποι φωτών (lights): (Point Light), (Directional Light) και (Spot Light). Μέσα από τις κατάλληλες ρυθμίσεις μπορούν να προσφέρουν ένα αρκετά ικανοποιητικό αποτέλεσμα στην εφαρμογή.

- Animation. Σε εφαρμογές που χρειάζονται στοιχεία κίνησης και διαδραστικότητας όπως είναι τα ηλεκτρονικά παιχνίδια μπορούμε να εφαρμόσουμε την κατάλληλη κίνηση στα τρισδιάστατα μοντέλα με τη βοήθεια του Animation. Η κίνηση του αντικειμένου δημιουργείται σε σχέση με τον χρόνο αλλάζοντας την αρχική του κατάσταση (θέση, μέγεθος, περιστροφή). Στη συνέχεια αποθηκεύεται η διαδικασία μετάβασης από την αρχική στην τελική κατάσταση και πλέον αποτελεί στοιχείο του ίδιου του αντικείμενου. Κατά την εισαγωγή του στη σκηνή το μοντέλο θα πραγματοποιεί αυτή τη κίνηση αυτόματα ή όταν εμείς το επιθυμούμε.
- Scripting. Μια εφαρμογή εκτός από όλα αυτά τα προκαθορισμένα στοιχεία που θα εισάγουμε μπορεί να απαιτεί κάποιο είδος διαδραστικότητας ή δυναμικής παρέμβασης από τον χρήστη σε πραγματικό χρόνο (π.χ. κίνηση αντικείμενου κατά βούληση του χρήστη, αλλαγή σκηνής κ.τ.λ.). Αυτό το επιτυγχάνουμε με τη χρήση κώδικα που θα μας επιτρέψει να ελέγξουμε στοιχεία στη σκηνή μας σε πραγματικό ή μη, χρόνο. Υπάρχουν πολλές γλώσσες προγραμματισμού που υποστηρίζουν αυτές τις διεργασίες όπως είναι η Java 3D, JavaScript και η C#.
- Cameras. Ένα από τα βασικά στοιχεία επίσης σε μια τρισδιάστατη εφαρμογή είναι η χρήση κάμερας. Η κάμερα τοποθετείται στη σκηνή και αποτελεί, θα λέγαμε, "τα μάτια του χρήστη" μέσα απ'την οποία ο χρήστης θα μπορεί να βρεθεί εικονικά στη σκηνή που δημιουργήσαμε. Μέσα από αυτήν και την κατάλληλη χρήση κώδικα μπορούμε να περιηγηθούμε και να παρατηρήσουμε το περιβάλλον σε πρώτο πρόσωπο. Ακόμα υπάρχει η επιλογή τοποθέτησης περισσότερων από μια κάμερες στη σκηνή έτσι ώστε να υπάρχει δυνατότητα προβολής από διαφορετικές οπτικές γωνίες. Η χρήση καμερών παρέχεται και υποστηρίζεται από όλες σχεδόν τις πλατφόρμες δημιουργίας τρισδιάστατων εφαρμογών όπως είναι το 3D Studio Max και το Unity.

Στο πρώτο κεφάλαιο παρουσιάζεται η εφαρμογή που δημιουργήθηκε και όλες οι δυνατότητες που παρέχονται μέσα από αυτήν.

Στο δεύτερο κεφάλαιο εξετάζεται αναλυτικά η ανάπτυξη των τρισδιάστατων μοντέλων μέσα από το 3ds Max, καθώς αναλύονται και στοιχεία που αναφέρθηκαν παραπάνω, μέσα από προσαρμογή της υφής και τη δημιουργία animation.

Στο τρίτο και τελευταίο κεφάλαιο διακρίνονται μέσα από το Unity στοιχεία της εφαρμογής όπως το terrain, η κάμερα, η εισαγωγή φωτισμού στην σκηνή και η διαδικασία του scripting για την προσθήκη διαδραστικότητας. Μέσα από αυτό το κεφάλαιο εξετάζεται η σύνθεση και ο συνδυασμός όλων των στοιχείων της εφαρμογής ώστε να επιτευχθεί το τελικό αποτέλεσμα.

Μεθοδολογία Υλοποίησης

Παρακάτω αναφέρεται η μεθοδολογία που ακολουθήθηκε στην ανάπτυξης της εφαρμογής χρησιμοποιώντας τα κατάλληλα προγραμματιστικά εργαλεία σε κάθε βήμα υλοποίησης.

A. Ανάλυση, σχεδίαση και ανάπτυξη 3D μοντέλων με το εργαλείο 3ds Max.

1. Σχεδίαση και ανάπτυξη μουσείου.

Στην δημιουργία του μουσείου αρχικά αναπτύσσεται σταδιακά η κάτοψη του κτιρίου, διαχωρίζοντας τους χώρους και τα δωμάτια από τα οποία αποτελείται. Στην συνέχεια το μουσείο διαμορφώνετε περαιτέρω με την δημιουργία εισόδων, ορόφων, παραθύρων και λεπτομερειών που ίσως χρειάζεται. Τέλος η ανάπτυξη την οροφής ολοκληρώνει το κτιριακό μοντέλο.

2. Ανάπτυξη επιπλέων εξοπλισμού και άλλων πρόσθετων μοντέλων.

Αυτή η περίπτωση περιλαμβάνει την μοντελοποίηση των διαφόρων επιπλέον στοιχείων του μουσείου όπως είναι σκάλες, διαζώματα, αψίδες, πόρτες κτλ. Αυτά τα στοιχεία μοντελοποιούνται ξεχωριστά και προστίθενται στο τελικό κτήριο.

3. Σχεδίαση και ανάπτυξη εκθεμάτων.

Η ανάπτυξη για κάθε ένα από τα μοντέλα πραγματοποιείται ξεχωριστά χρησιμοποιώντας στοιχεία και εικόνες από τα πραγματικά πρωτότυπα εκθέματα για την ρεαλιστική τους απεικόνιση.

4. Σχεδίαση και ανάπτυξη χαρακτήρα/avatar.

Μοντελοποιείται ο χαρακτήρας που μέσα από αυτόν θα πραγματοποιείται η περιήγηση στον χώρο και εφαρμόζονται τεχνικές για την κατασκευή ρούχων και την δημιουργία animation.

B. Σχεδίαση και υλοποίηση υφών με το εργαλείο Photoshop.

1. Σχεδίαση και ανάπτυξη υφών.

Η δημιουργία υφών βασίζεται σε οδηγούς texturemaps που με την βοήθεια τους σχεδιάζονται οι υφές και προσαρμόζονται κατάλληλα στο κάθε μοντέλο.

C. Ανάπτυξη εφαρμογής με το εργαλείο Unity 3d.

1. Σχεδίαση και ανάπτυξη περιβάλλοντος χώρου.

Αρχικά η ανάπτυξη του terrain περιλαμβάνει την χρήση εργαλείων του Unity στην μορφολογία του εδάφους και την δημιουργία βλάστησης στην σκηνή.

2. Εισαγωγή μοντέλων στην σκηνή.

Όλα τα μοντέλα εισάγονται και τοποθετούνται κατάλληλα στην σκηνή. Εισάγεται αρχικά το μουσείο και στην συνέχεια τοποθετούνται μέσα σε αυτό τα εκθέματα που θα απεικονίζονται. Παράλληλα τοποθετούνται επιπλέον μοντέλα στο terrain για την ολοκλήρωση του περιβάλλοντος χώρου και επίσης εισάγετε στη σκηνή και ο χαρακτήρας.

3. Τοποθέτηση κατάλληλου φωτισμού.

Ο φωτισμός τοποθετείται και διαμορφώνεται, ώστε να φωτίζονται κατάλληλα τα στοιχεία της σκηνής και ακόμα δημιουργούνται τα κάποια εφέ στο πλαίσιο της προσέγγισης της ρεαλιστικότητας.

4. Εισαγωγή στοιχείου κάμερας.

Η κάμερα τοποθετείται στην σκηνή και προσαρμόζεται στο avatar, ώστε να το ακολουθεί κατά την διάρκεια της κίνησης.

5. Δημιουργία διάδρασης και αλληλεπίδρασης.

Μέσα από αρχεία κώδικα javascript και C# συντάσσονται εντολές και συναρτήσεις που ελέγχουν την διαδραστικότητα της εφαρμογής.

Παρουσίαση Εφαρμογής

Το παρόν project κατασκευάστηκε για τις ανάγκες της διπλωματικής διατριβής του φοιτητή Μπολωνάκη Κων/νου. Η εφαρμογή αυτή ονομάζεται Minoan Age exhibits, όπου μέσα από ένα τρισδιάστατο κόσμο παρουσιάζεται ένα εικονικό μουσείο που φιλοξενεί εκθέματα του μινωικού και μυκηναϊκού πολιτισμού. Από το πληκτρολόγιο, ο χρήστης μπορεί να περιηγηθεί, σε πρώτο πρόσωπο, στους χώρους του μουσείου, μέσα από ένα avatar/χαρακτήρα, που αντιπροσωπεύει μια νεαρή φοιτήτρια αρχαιολογίας, την Ρέα. Κατά την περιήγηση ο χρήστης μπορεί να παρατηρεί τα τρισδιάστατα εκθέματα και να συλλέγει πληροφορίες από το κάθε ένα που συναντάει, διαμέσου ενός διαδραστικού παραθυρού που εμφανίζεται καθώς τα πλησιάζει.

Το μουσείο αποτελείται από δύο επίπεδα, ισόγειο και πρώτος όροφος. Στο ισόγειο παρουσιάζονται κυρίως μινωικά αγγεία, αγαλματίδια, κοσμήματα και πολλά άλλα αρχαιολογικά ευρήματα από περιοχές όπως, Κνωσός, Φαιστός, Γόρτυνα, Ζάκρος κ.α. Ακόμα, στο εσωτερικό του μουσείου υπάρχει δωμάτιο που προσομοιώνει την αίθουσα του θρόνου από το παλάτι της Κνωσού. Στον πρώτο όροφο παρουσιάζονται μινωικές και μυκηναϊκές τοιχογραφίες από όλη την επικράτεια των δύο αυτών πολιτισμών.

Το περιβάλλον της εφαρμογής είναι αρκετά ατμοσφαιρικό με κατάλληλη background μουσική υπόκρουση και φωτισμό, ενώ υπάρχουν στοιχεία που παραπέμπουν σε adventure game. Παρακάτω περιγράφονται αναλυτικότερα όλα τα μέρη και τα στοιχεία της εφαρμογής ξεχωριστά.

1.1 Χειρισμός

Ο χειρισμός ελέγχου της εφαρμογής είναι παρόμοιος με τον χειρισμό ενός ηλεκτρονικού παιχνιδιού.

Η επιλογή κάποιου στοιχείου που εμφανίζεται στην οθόνη γίνεται με το πάτημα του ποντικιού πάνω του. Για την κίνηση του χαρακτήρα χρησιμοποιείται το πληκτρολόγιο:

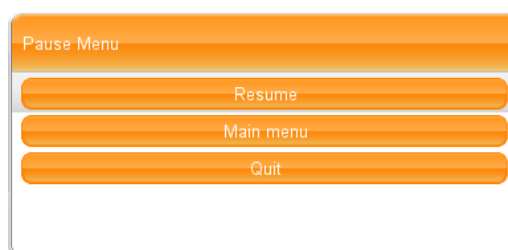
W ή Arrow UP..... για κίνηση μπροστά
S ή Arrow Down..... για κίνηση προς τα πίσω
A ή Arrow Left..... για κίνηση αριστερά
D ή Arrow Right..... για κίνηση δεξιά
Για Run Mode (τρέξιμο) χρησιμοποιείται το πλήκτρο R



Η περιστροφή της κάμερας πραγματοποιείται από το ποντίκι (mouse). Κρατώντας πατημένο το δεξιό ή το αριστερό κουμπί του ποντικιού και μετακινώντας το, η κάμερα περιστρέφεται γύρω από τον χαρακτήρα. Για Zoom In ή Zoom Out χρησιμοποιείται το κεντρικό κουμπί ή η ροδέλα του ποντικιού.



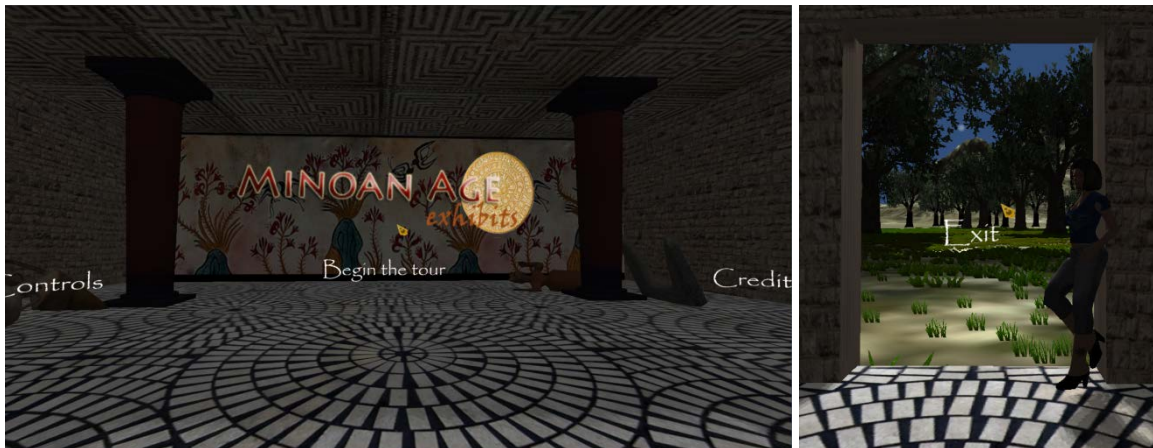
Τέλος το Esc ή το πλήκτρο P ανοίγει το Pause Menu, όπου από εκεί μπορεί να τερματιστεί το παιχνίδι ή να επιστρέψουμε στο κυρίως μενού της εφαρμογής.



Εικόνα 1.1 Το Pause Menu της εφαρμογής

1.2 Βασικό μενού

Εκτελώντας το αρχείο της εφαρμογής, το πρώτο περιβάλλον που συναντάμε είναι το κυρίως μενού. Ο χώρος αυτός απεικονίζει ένα μινωικό κτίριο (παραπέμποντας σε ναό της εποχής) με διασκορπισμένα αγαλματίδια και δοχεία στο εσωτερικό του, ενώ στην είσοδο συναντάμε την Ρέα που μέσα από αυτήν θα μπορούμε αργότερα, να κινηθούμε στο χώρο του μουσείου.



Εικόνα 1.2 Στιγμιότυπα με τις επιλογές του Main Menu

Από το τρισδιάστατο μενού επιλογών μπορούμε να επιλέξουμε το τερματισμό της εφαρμογής, να ανοίξουμε το παράθυρο με την λίστα χειρισμού της, να δούμε τα credits ή να επιλέξουμε να μεταβούμε στο κυρίως μέρος της εφαρμογής που είναι το περιβάλλον του μουσείου. Επιλέγοντας τα ξεκινήσουμε την περιήγηση και κατά την διάρκεια του loading, προβάλλεται στην οθόνη μια εικόνα με το λογότυπο της εφαρμογής.



Εικόνα 1.3 Το παράθυρο Controls και η οθόνη του Loading

1.3 Κύριο μέρος της εφαρμογής

1.3.1 Ξενάγηση στους χώρους του μουσείου

Μόλις μεταβούμε στο κύριο περιβάλλον της εφαρμογής, βρισκόμαστε αρχικά μπροστά από την κεντρική πύλη του μουσείου. Κινώντας τον χαρακτήρα από το πληκτρολόγιο μπορούμε να περιηγηθούμε στους κατάλληλα διαμορφωμένους χώρους του μουσείου, αλλά δεν υπάρχει η δυνατότητα να μετακινηθούμε έξω από τον προαύλιο χώρο.



Εικόνα 1.4 Στιγμιότυπα από τον προαύλιο χώρο του μουσείου

Μόλις εισέλθουμε στον προθάλαμο του μουσείου, ανεβαίνοντας τα σκαλιά ή την ράμπα που βρίσκονται έξω από το κτίριο, συναντάμε το βιβλίο που εξιστορεί την ύπαρξη του μινωικού πολιτισμού από την δημιουργία έως την εξαφάνιση του. Στα δεξιά, υπάρχει ένας διάδρομος που οδηγεί στο wc του κτιρίου.



Εικόνα 1.5 Στιγμιότυπα από τον προθάλαμο και το wc του μουσείου

Στην κύρια αίθουσα συναντάμε το άγαλμα ενός ταύρου, που σαν το ιερότερο ζώο των μινωιτών, αποτελεί ένα βασικό στοιχείο του πολιτισμού τους.



Εικόνα 1.6 Η κεντρική αίθουσα του μουσείου

Δεξιά και αριστερά της κύριας αίθουσας υπάρχουν δύο πτέρυγες που αποτελούνται από τέσσερις αίθουσες η κάθε μια. Στις αίθουσες αυτές απεικονίζονται διάφορα εκθέματα όπως είναι το αγαλματίδιο της θεάς των φιδιών, το κεφάλι του ταύρου, ο κρατήρας με τα κρίνα, ο δίσκος της Φαίστου, ο διπλός πέλεκυς και το περιδέραιο με τις μέλισσες, που είναι μερικά από τα πιο χαρακτηριστικά ευρήματα του μινωικού πολιτισμού.



Εικόνα 1.7 Στιγμιότυπα από κάποια εκθέματα του μουσείου

Μεταβαίνοντας από την κύρια αίθουσα στην πίσω μεριά του μουσείου, συναντάμε την κεντρική σκάλα που οδηγεί στον πρώτο όροφο και τον κεντρικό διάδρομο που οδηγεί στην πίσω μεγάλη αίθουσα του μουσείου.

Στη μέση αυτού του διαδρόμου συναντάμε και έναν ανελκυστήρα που επίσης χρησιμοποιείται για την μετάβαση από το ισόγειο στον πρώτο όροφο και αντίστροφα.



Εικόνα 1.8 Ο ανελκυστήρας στον κεντρικό διαδρόμο

Στην τελευταία αίθουσα του ισογείου προσομοιώνεται ένα μέρος από το παλάτι της Κνωσού όπως είναι το δωμάτιο με τα δελφίνια και η αίθουσα του θρόνου του βασιλιά Μίνωα.



Εικόνα 1.9 Στιγμιότυπα από την αίθουσα προσομείωσης, το δωμάτιο με τα δελφίνια και το δωμάτιο του θρόνου.

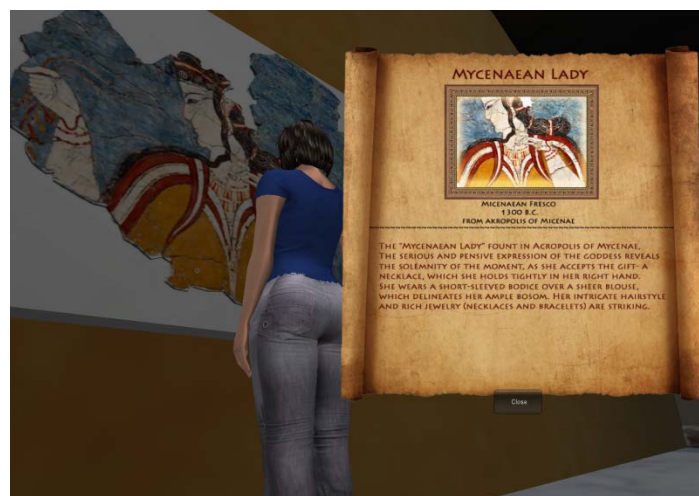
Στον δεύτερο όροφο αρχικά συναντάμε δύο αίθουσες που περιέχουν μινωικές τοιχογραφίες που βρέθηκαν στο νησί της Θήρας και τοιχογραφίες από την μυκηναϊκή περίοδο. Συνεχίζοντας την περιήγηση στον δεύτερο όροφο συναντάμε ένα μεγάλο διάδρομο με μπαλκόνι περιμετρικά του μουσείου. Εκεί, υπάρχουν κυρίως τοιχογραφίες από το παλάτι της Κνωσού, όπως ο πρίγκιπας με τα κρίνα, τα ταυροκαθάψια, οι γαλάζιες πριγκίπισσες κ.α.



Εικόνα 1.10 Στιγμιότυπα από τον πρώτο όροφο

1.3.2 Παράθυρο περιγραφής εκθεμάτων

Καθώς πλησιάζουμε κάποιο έκθεμα εμφανίζεται στην οθόνη η επιλογή Examine που μας παραπέμπει στο παράθυρο που περιέχει της πληροφορίες του εκθέματος που παρατηρούμε. Οι πληροφορίες αυτές αναφέρουν στοιχεία του εκθέματος που σχετίζονται με την χρήση του, την ηλικία του, την τοποθεσία εύρεσης του, καθώς και κάποια γενικότερα στοιχεία περιγραφής του τρόπου ζωής και της κουλτούρας αυτού του πολιτισμού στον οποίο αναφέρεται.



Εικόνα 1.11 Η οθόνη περιγραφής των εκθεμάτων

Στης σελίδες που ακολουθούν αναλύεται ο τρόπος και τα εργαλεία που χρειάζονται για την ανάπτυξη τέτοιου είδους τρισδιάστατων εφαρμογών και περιγράφονται τα βήματα που ακολουθήθηκαν στην πορεία υλοποίησης της συγκεκριμένης εφαρμογής που παρουσιάστηκε στο παρών κεφάλαιο.

3D Studio Max

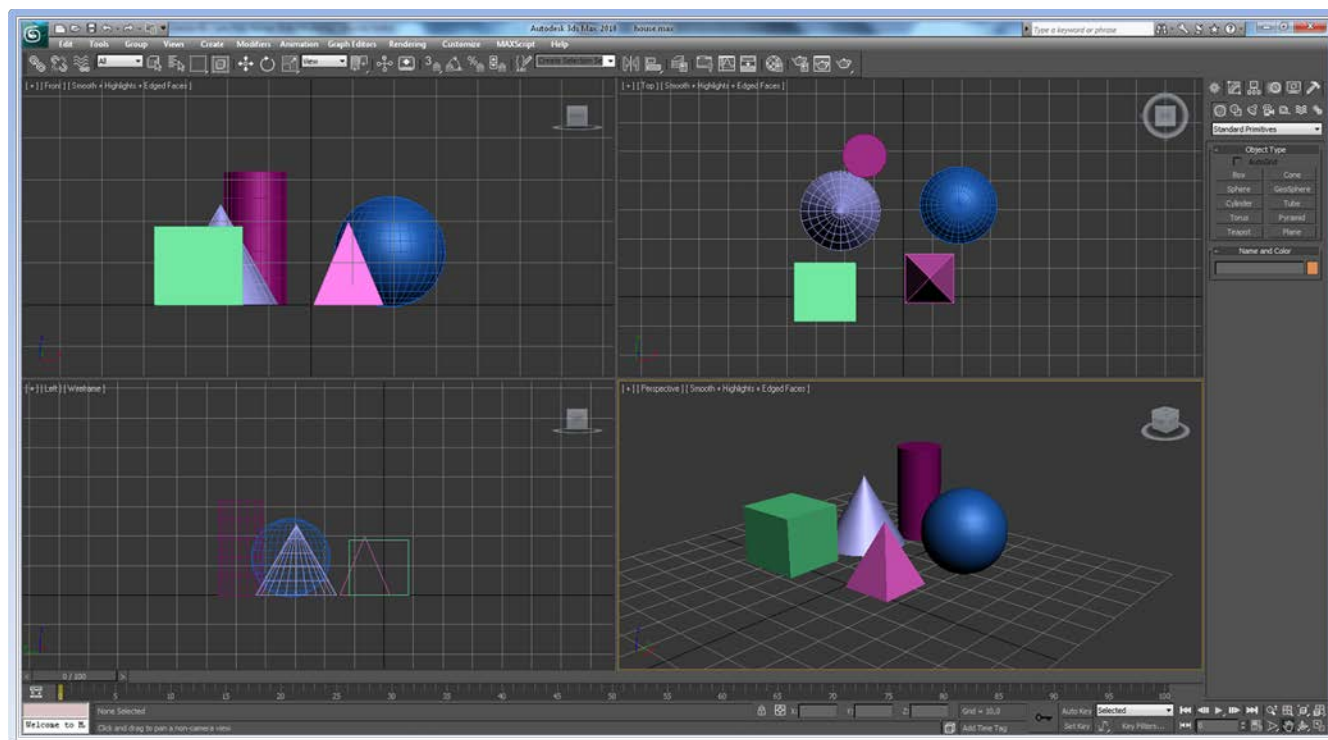
2.1 Εισαγωγή στο 3D Studio Max

Το 3d Studio Max εκδόθηκε το 2008 από την Autodesk, μια εταιρία που κατέχει υψηλή θέση στον κλάδο του λογισμικού δισδιάστατης και τρισδιάστατης σχεδίασης και μηχανολογικού σχεδιασμού. Από το 2008 έως σήμερα έχουν κυκλοφορήσει στην αγορά διάφορες εκδόσεις του προγράμματος: 3ds Max9, 3ds Max 2008, 3ds Max 2009, 3ds Max 2010, 3ds Max 2011, 3ds Max 2012 και αναμένεται το 3ds Max 2013. Το εργαλείο αυτό είναι ένα λογισμικό κατασκευής και ανάπτυξης τρισδιάστατων γραφικών. Χρησιμοποιείται ευρέως για την δημιουργία 3D μοντέλων και animation σε εφαρμογές τρισδιάστατης απεικόνισης, όπως είναι ηλεκτρονικά παιχνίδια, εφαρμογές αρχιτεκτονικού σχεδιασμού και ανάπτυξη γραφικών εφέ. Μέσα απο 3ds Max προσφέρεται η δυνατότητα εισαγωγής και εξαγωγής (import/export) μοντέλων από μια μεγάλη γκάμα διαφορετικών τύπων αρχείων, όπως είναι: .3ds, .fbx, .obj, .x3d. Για την ανάπτυξη 3d γραφικών διαθέτει μια μεγάλη γκάμα επιλογών και δυνατοτήτων στην σχεδίαση τρισδιάστατων μοντέλων και παρέχει πλήρες σύστημα μοντελοποίησης. Τέλος μέσα από αυτήν την πλατφόρμα υπάρχει η δυνατότητα δημιουργίας animation και προσαρμογής υφών (textures) στα μοντέλα, με την τεχνική Unwrap, όπως θα αναλύσουμε παρακάτω.

2.2 Τα βασικά εργαλεία του 3ds Max

Ξεκινώντας την διαδικασία μοντελοποίησης πρέπει να γνωρίζουμε ορισμένα βασικά χαρακτηριστικά. Ένα τρισδιάστατο μοντέλο αποτελείται από: Vertices (κορυφές), Edges (ακμές), Polygons (πολύγωνα) και κάθε ανεξάρτητο σύνολο από πολύγωνα που αποτελούν ένα Element (σύνολο στοιχείων).

Κάθε ένα από αυτά τα στοιχεία μπορούν να τροποποιηθούν ανάλογα με τις ανάγκες μας, μέσα από μια σειρά εργαλείων που μας προσφέρει το 3ds Max. Επίσης στο πρόγραμμα παρέχονται έτοιμα βασικά σχήματα, όπως line, plane, box, sphere, pyramid, cylinder που αποτελούν την βάση, πάνω στα όποια θα δουλέψουμε στην διαδικασία του modeling. Παρακάτω παρουσιάζονται τα βασικότερα εργαλεία του προγράμματος 3ds Max 2010 που θα χρησιμοποιηθούν για διάφορα παραδείγματα μοντελοποίησης που περιγράφονται στις επόμενες σελίδες.



Εικόνα 2.1 Περιβάλλον εργασίας 3ds Max2010

2.2.1 Main Toolbar



(Selection Tool). Εργαλείο επιλογής.



(Select by Name). Δίνει την δυνατότητα επιλογής μεμονωμένων αντικειμένων μέσα από μια λίστα που αναφέρονται όλα τα αντικείμενα που βρίσκονται στη σκηνή ονομαστικά.



(Select and Move). Εργαλείο μετακίνησης.



(Select and Rotate). Εργαλείο περιστροφής.



(Select and Uniform Scale). Εργαλείο τροποποίησης κλίμακας.

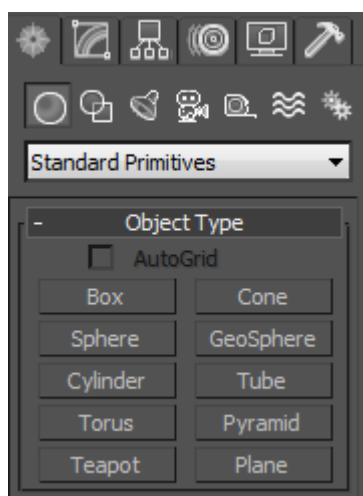


(Mirror). Εργαλείο δημιουργίας ειδώλου.



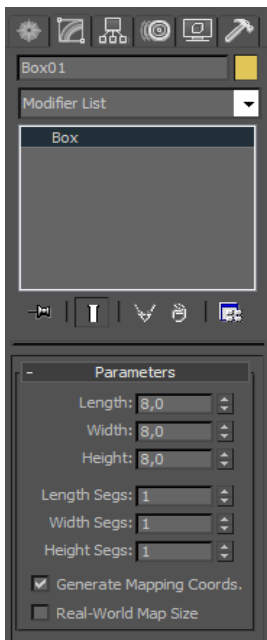
(Render Production). Αποτυπώνει την 3D σκηνή σε εικόνα δυο διαστάσεων.

2.2.2 Command Panel



- (Create). Μέσα από την καρτέλα Create, παρέχεται η δυνατότητα δημιουργίας αντικειμένων απο μια λίστα που περιέχει: βασικά σχήματα, φώτα, κάμερες και συστήματα ανάπτυξης animation και εφέ.

Σχήμα 2.1 Command Panel

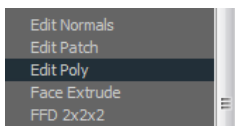


Σχήμα 2.2 Modifier Panel

• (Modify). Από την καρτέλα Modify υπάρχει η δυνατότητα τροποποίησης παραμέτρων και αλλαγής κατάστασης των μοντέλων, μετά την εισαγωγή τους στην σκηνή.

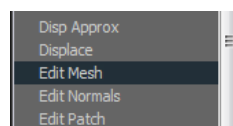
(Modifier List). Μέσα από μια μεγάλη λίστα επιλογών μπορούμε να εφαρμόσουμε στα αντικείμενα διάφορες καταστάσεις και δυνατότητες μορφοποίησης.

Μερικοί χρήσιμοι Modifiers



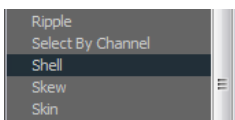
Edit Poly

Μετατροπή του μοντέλου σε μορφή τροποποίησημων πολυγώνων.



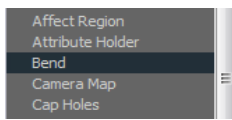
Edit Mesh

Μετατροπή του μοντέλου σε πλέγμα πολυγώνων.



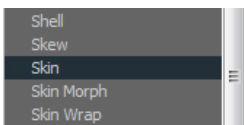
Shell

Δίνει πάχος σε μια επιφάνεια ή μετατρέπει την εσωτερική επιφάνεια των μοντέλων σε τροποποιήσιμα πολύγωνα.



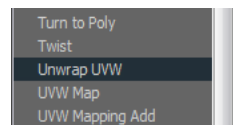
Bend

Δίνει την δυνατότητα σε ένα μοντέλο να λυγίσει, διαγράφοντας μια καμπύλη.



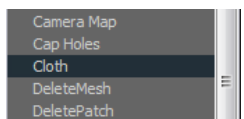
Skin

Χρησιμοποιείται κυρίως για την δημιουργία animation. Εξομαλύνει την κίνηση των πολυγώνων του μοντέλου μέσα από bone systems.



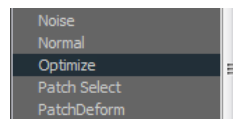
Unwrap UVM

« Ξεδιπλώνει » το 3d μοντέλο ώστε όλα τα πολύγωνα του να είναι εμφανή και δημιουργεί ένα UVM map οδηγό που χρειάζεται για την προσαρμογή υφών στα μοντέλα.



Cloth

Χρησιμοποιείται κυρίως για την δημιουργία ρούχων και γενικότερα υφασμάτων σε 3d. Μετατρέπει ένα μοντέλο σε ένα είδος υφάσματος που μορφοποιείται ανάλογα με την βαρύτητα και τα collision που δέχεται από κάποιο άλλο αντικείμενο.

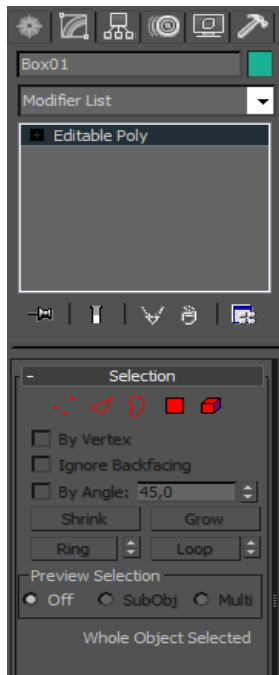


Optimize

Μειώνει τον αριθμό των πολυγώνων ενός μοντέλου. Είναι κατάλληλο για μοντέλα χωρίς μεγάλη λεπτομέρεια.

2.2.3 Modifier Editable Poly

Για να μπορέσουμε να επεξεργαστούμε τα πολύγωνα ενός 3d model, χρειάζεται αυτό να μετατραπεί σε μορφή τροποποιήσεων πολυγώνων (Editable Poly).



Σχήμα 2.3 Modifier Editable Poly Panel

• (Selection). Η Εφαρμογή του Editable Poly modifier μας εμφανίζει μια νέα λίστα εργαλείων για την επεξεργασία στοιχείων του μοντέλου:

▪ *Vertex* (κορυφή)



▪ *Edges* (ακμή)



▪ *Border* (όριο)



▪ *Polygon* (επιφάνεια)

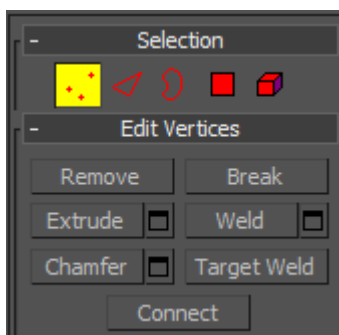


▪ *Element* (στοιχείο)



▪ *Vertex Tools*

Το κουμπί *Vertex* χρησιμοποιείτε για την επιλογή κορυφών των πολυγώνων.



Σχήμα 2.4 Vertex Tools Panel

Remove: Αφαιρεί τις επιλεγμένες κορυφές.

Break: Αποσυνδέει τις επιλεγμένες κορυφές από τις υπόλοιπες του συνόλου.

Extrude: Προεκτείνει επιλεγμένες κορυφές, δημιουργώντας μια προεξοχή.

Weld: Ενώνει δύο ή παραπάνω κορυφές και τις συγχωνεύει σε μια.

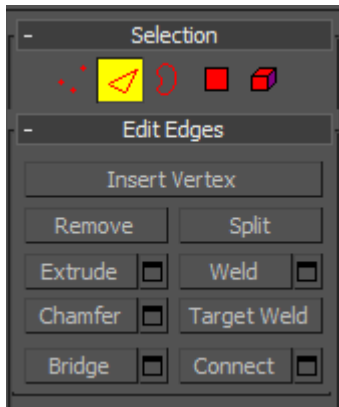
Chamfer: Αφαιρεί τις επιλεγμένες κορυφές δημιουργώντας μια εσοχή.

Target Weld: Συγχωνεύει μια κορυφή σε μια άλλη.

Connect: Συνδέει τις επιλεγμένες κορυφές, προσθέτοντας μια ακμή ανάμεσα τους.

▪ *Edge Tools*

Το κουμπί *Edge* χρησιμοποιείτε για την επιλογή ακμών των πολυγώνων.



Σχήμα 2.5 Edge Tools Panel

Insert Vertex: Εισάγει μια κορυφή κατά μήκος της ακμής

Remove: Αφαιρεί τις επιλεγμένες ακμές.

Split: Αποσυνδέει τις επιλεγμένες ακμές από τις υπόλοιπες του συνόλου.

Extrude: Προεκτείνει επιλεγμένες ακμές, δημιουργώντας μια προεξοχή.

Weld: Ενώνει δύο ή παραπάνω ακμές και τις συγχωνεύει σε μια.

Chamfer: Αφαιρεί τις επιλεγμένες ακμές δημιουργώντας μια εσοχή.

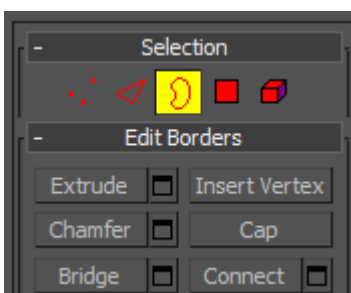
Target Weld: Συγχωνεύει μια κορυφή σε μια άλλη.

Bridge: Δημιουργεί μια γέφυρα μεταξύ δύο ακμών, προσθέτοντας ένα πολύγωνο.

Connect: Συνδέει τις επιλεγμένες ακμές προσθέτοντας άλλη μια ακμή ανάμεσα.

▪ *Border Tools*

Το κουμπί *Border* χρησιμοποιείτε για την επιλογή ενός βρόγχου που δημιουργείται από τις ακμές των πολυγώνων που ορίζουν μια κενή περιοχή του μοντέλου.



Σχήμα 2.6 Border Tools Panel

Extrude: Προεκτείνει τον βρόγχο δημιουργώντας μια προεξοχή.

Insert Vertex: Προσθέτει μια κορυφή στον βρόγχο.

Chamfer: Δημιουργεί έναν δεύτερο βρόγχο ακμών γύρο από τον πρώτο.

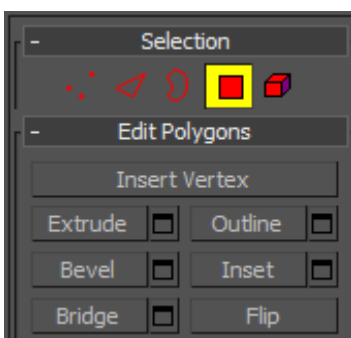
Cap: Καλύπτει τον βρόγχο με ένα πολύγωνο.

Bridge: Δημιουργεί μια γέφυρα από πολύγωνα μεταξύ δύο επιλεγμένων βρόγχων.

Connect: Εισάγει κορυφές στις ακμές του βρόγχου, ώστε να χωριστούν σε τμήματα.

▪ *Polygon Tools*

Το κουμπί *Polygon* χρησιμοποιείτε για την επιλογή των επιφανειών των πολυγώνων.



Σχήμα 2.7 Polygon Tools Panel

Insert Vertex: Εισάγει μια κορυφή μέσα στο πολύγωνο χωρίζοντάς το σε τμήματα.

Extrude: Προεκτείνει τα επιλεγμένα πολύγωνα, δημιουργώντας μια προεξοχή.

Outline: Τροποποιεί το μέγεθος των επιλεγμένων πολύγωνων.

Bevel: Δημιουργεί επιπλέον πολύγωνο μέσα στο επιλεγμένο προεκτείνοντας το.

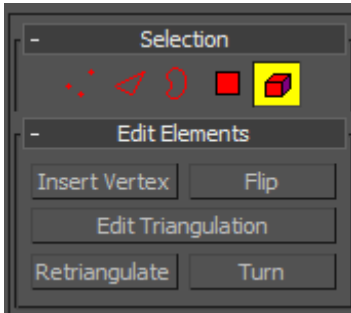
Inset: Δημιουργεί επιπλέον πολύγωνο μέσα στο επιλεγμένο.

Bridge: Δημιουργεί γέφυρα από πολύγωνα μεταξύ δυο επιφανειών πολυγώνων

Flip: Αναστρέφει την επιφάνεια των επιλεγμένων πολυγώνων.

▪ *Element Tools*

Το κουμπί *Element* χρησιμοποιείται για την επιλογή του συνόλου των πολυγώνων που αποτελούν ένα ανεξάρτητο μοντέλο.



Σχήμα 2.8 Element Tools Panel

Insert Vertex: Εισάγει μια κορυφή μέσα σε ένα πολύγωνο ή μια ακμή.

Flip: Αναστρέφει την επιφάνεια του μοντέλου.

Edit Triangulation: Τροποποιεί τα πολύγωνα του μοντέλου σε τρίγωνα.

Retriangulate: Επανατροποποιεί τα πολύγωνα σε τρίγωνα με αντιθετή φορά.

Turn: Αλλάζει φορά σε επιλεγμένα πολύγωνα.



Σχήμα 2.9 Geometry Tools Panel

- (*Edit Geometry*). Πολύ χρήσιμα είναι επίσης τα εργαλεία που βρίσκονται στη λίστα *Edit Geometry*. Αυτά τα εργαλεία μας βοηθούν να τροποποιούμε τα μοντέλα με περισσότερη ευελιξία. Κάποια από αυτά αναφέρονται παρακάτω.

Create: Δημιουργεί επιπλέον στοιχεία (κορυφές, ακμές κτλ.) στην επιθυμητή θέση στη σκηνή.

Collapse: Διαγράφει τα επιλεγμένα στοιχεία.

Attach: Ενώνει δύο διαφορετικά αντικείμενα σε ένα ενιαίο μοντέλο.

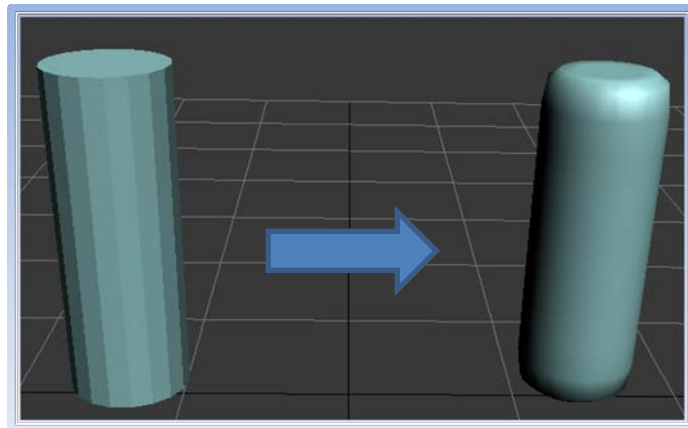
Detach: Αποδεσμεύει τα επιλεγμένα στοιχεία από το μοντέλο, δημιουργώντας ένα νέο αντικείμενο.

Cut: Κόβει τις επιφάνειες πολυγώνων στα επιθυμητά σημεία, χωρίζοντάς τες σε επιμέρους πολύγωνα.



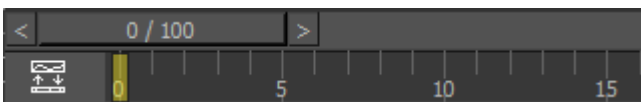
Σχήμα 2.10 Surface Parameters Panel

• (Subdivision Surface). Στην ίδια καρτέλα συναντάμε και τις παραμέτρους για την «λείανση» της επιφάνειας του μοντέλου. Επιλέγοντας το check box “Use NURMS Subdivision” και την κατάλληλη τιμή λείανσης (Smoothness) το μοντέλο από ευδιάκριτα πολύγωνα με εμφανή γωνίες, μετατρέπεται σε λεία επιφάνεια.



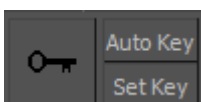
Εικόνα 2.2 Χρήση του “Use NURM Subdivision”

2.2.4 TimeLine



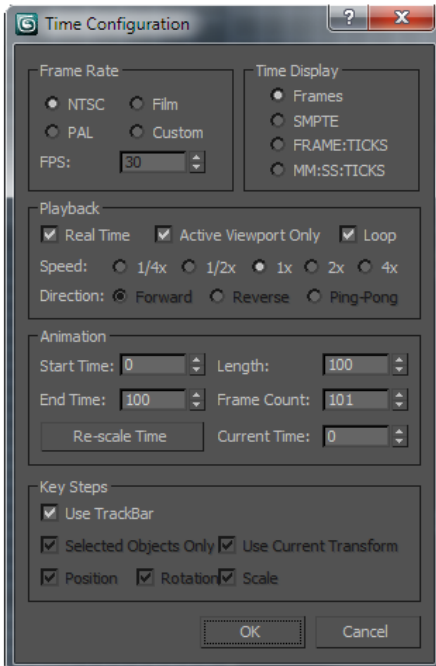
Το Timeline χρησιμοποιείται για την δημιουργία Animation. Εισάγοντας Key frames μπορούμε να απεικονίσουμε και να αποθηκεύσουμε μια αλλαγή κατάστασης ενός μοντέλου σε σχέση με τον χρόνο.

Key framing Tools




Auto Key: Εισάγει αυτόματα Key frames σε κάθε αλλαγή κατάστασης του αντικειμένου.

Set Key: Εισάγει Key frames με επολογή του χρήστη.

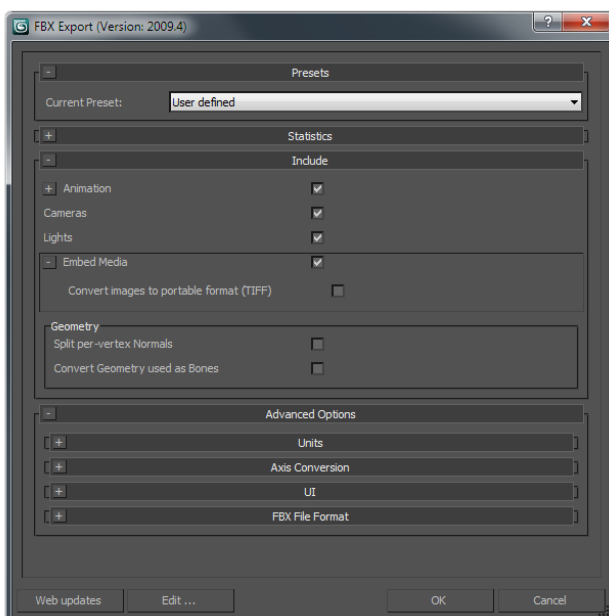


Σχήμα 2.11 Time Parameters Window

- (Time Configuration). Επιλέγοντας το κουμπί  Time Configuration, ανοίγει το παράθυρο με τις παραμέτρους για τον έλεγχο του χρόνου στη διαδικασία του animation. Από το Start και End Time μπορούμε να ορίσουμε το χρονικό περιθώριο που θα έχει το Timeline, ενώ μπορούμε να μειώσουμε ή να αυξήσουμε την ταχύτητα που θα έχει το animation από το πεδίο Playback.

2.2.5 FBX Export Screen

Έχοντας ολοκληρώσει την διαδικασία του modeling, η σκηνή μπορεί να εξαχθεί διάφορους τύπους αρχείων. Ο τύπος αρχείου που χρησιμοποιείτε περισσότερο είναι .fbx



Σχήμα 2.12 FBX Export Window

- (FBX Export). Η εξαγωγή σε τύπο fbx δίνει την δυνατότητα στο αρχείο να συμπεριλαμβάνει animation, κάμερες, φώτα και υφές που μπορεί να υπάρχουν στην σκηνή.

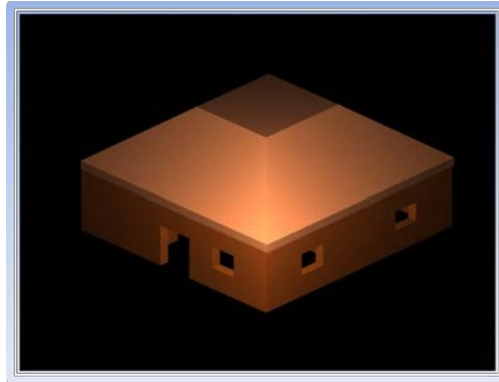
2.3 Διαδικασία μοντελοποίησης

Στο 3d modeling δεν υπάρχει σωστός και λάθος τρόπος που ακολουθείται για την κατασκευή ενός μοντέλου, απλώς έχουν καθιερωθεί κάποιες μεθοδολογίες για τις ανάγκες υλοποίησης διαφορετικών τύπων μοντέλων. Υπάρχουν διάφορες μέθοδοι και τεχνικές που ακολουθούνται στην διαδικασία της μοντελοποίησης. Κάθε μια από αυτές χρησιμοποιείται ανάλογα με το μοντέλο που θα κλιθεί να υλοποιηθεί, ενώ σε πολλές περιπτώσεις χρησιμοποιείται ο συνδυασμός τους για να επιτευχθεί το επιθυμητό αποτέλεσμα. Το 3ds Max έχει την δυνατότητα να καλύψει κάθε ανάγκη που θα προκύψει στην διαδικασία του modeling και υποστηρίζει την χρήση διαφόρων μεθόδων, μέσα από μια μεγάλη γκάμα εργαλείων. Κάποιες βασικές τεχνικές είναι το Polygon Modeling, το Box Modeling και το Spline Modeling.

Στην μέθοδο Polygon Modeling το μοντέλο αναπτύσσεται σταδιακά με την αυξητική διαδικασία πρόσθεσης νέων πολυγώνων στο μοντέλο. Η μεθοδός του Box Modeling βασίζεται στην σταδιακή τροποποίηση των πολυγώνων ενός Box το οποίο αποτελείται από πολλά τμήματα (Segments). Τέλος, με την μέθοδο Spline Modeling χρησιμοποιούνται αντικείμενα δύο διαστάσεων όπως Lines και Circles είτε ως συστατικά, είτε ως οδηγοί για την δημιουργία αντικείμενων τριών διαστάσεων.

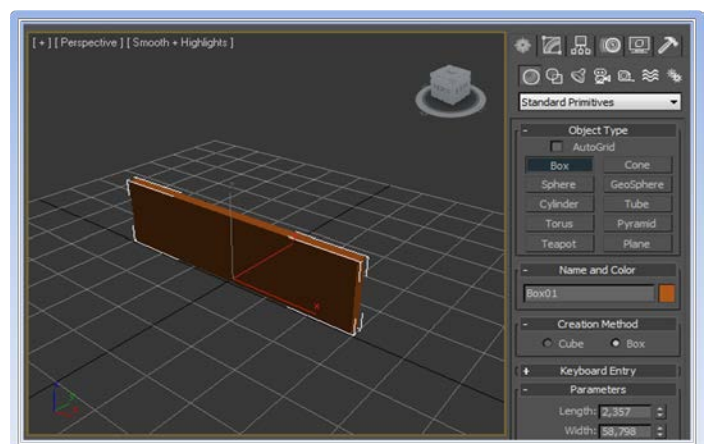
Παρακάτω θα δούμε ορισμένα παραδείγματα μοντελοποίησης στο 3ds max, ενώ περιγράφονται οι βασικοί μέθοδοι που ακολουθήθηκαν προκειμένου να κατασκευαστούν τα 3d μοντέλα της εφαρμογής που παρουσιάζονται στο 1^ο κεφάλαιο.

2.3.1 Παράδειγμα κατασκευής ενός απλού κτιρίου από Boxes με την μέθοδο Polygon Modeling.

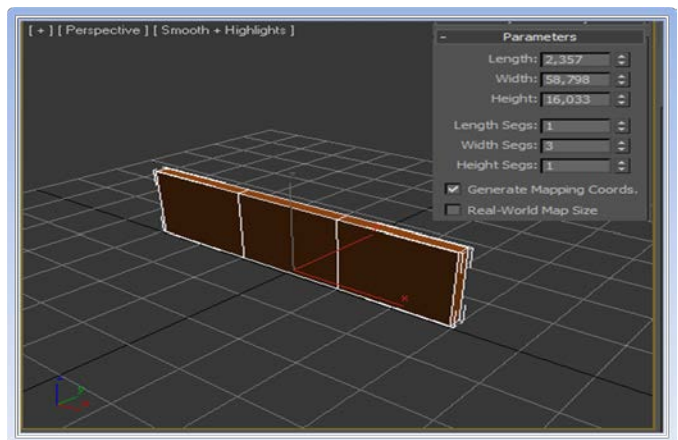


Για την κατασκευή ενός κτιρίου το 3ds Max παρέχει έτοιμα εργαλεία με την επιλογή Windows και AEC Extended από την λίστα του Geometry στην καρτέλα Create. Μέσα από μια σειρά επιλογών μπορούν να δημιουργηθούν διάφοροι τύποι Walls και Windows. Αναφέρεται ότι στο παράδειγμα αυτό ακολουθείται η μέθοδος της χρήσης απλών Boxes, όπου δίνεται η δυνατότητα εξοικείωσης με τις βασικές τεχνικές μοντελοποίησης.

1. Αρχικά απο τήν λίστα Create ή απο το μενού *Create*→*Standard Primitives*→*Box* σχεδιάζουμε ένα απλό **Box** στην επιθυμητή θέση σε σχήμα ορθογώνιου παραλληλεπίπεδου.

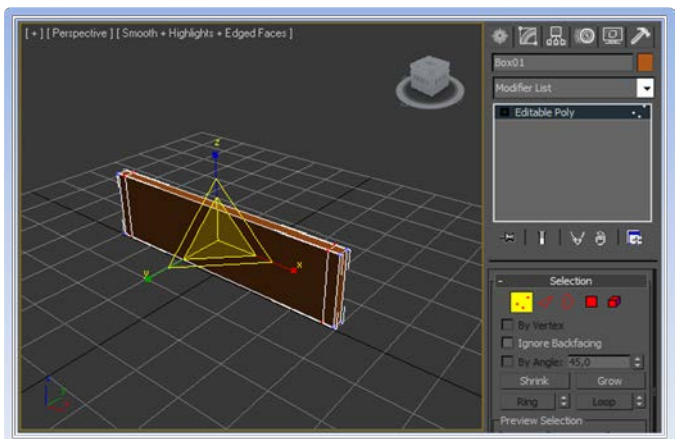


2. Από τις παραμέτρους μπορούμε να χωρίσουμε τις πλευρές του **Box** σε **Segments**. Έτσι κάθε πλευρά του, θα αποτελείται από περισσότερα του ενός πολύγωνα και θα μπορεί να τροποποιηθεί το κάθε ένα από αυτά μεμονωμένα. Στην περίπτωση αυτή επιλέγουμε να χωριστεί το πλάτος του box σε 3 segments.

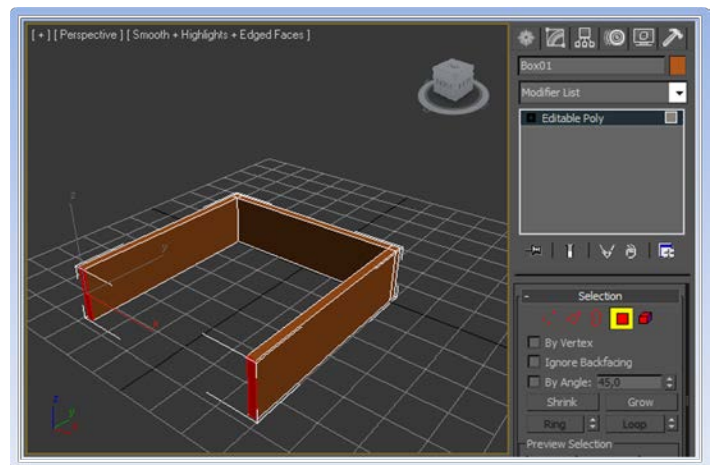


3. Για να μπορέσουμε να επεξεργαστούμε τα πολύγωνα, πρέπει να μετατρέψουμε το αντικείμενο σε **Editable Poly**. Αυτό γίνεται από το *Modifier list* → *Edit Poly* ή *Mouse Right Click* → *Convert To* → *Editable Poly*.

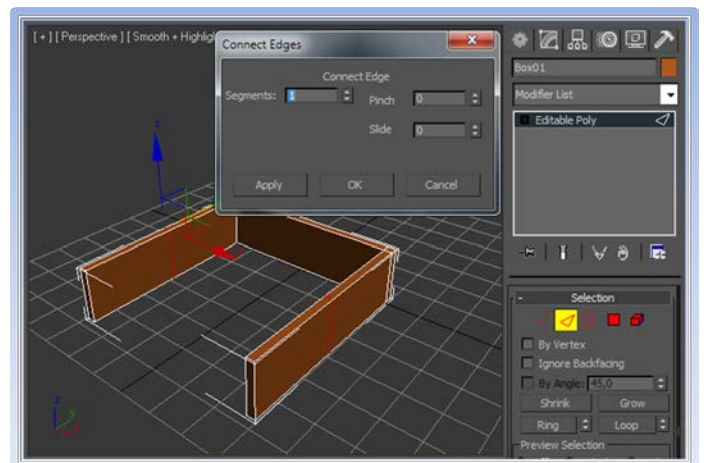
4. Στη συνέχεια επιλέγουμε τις εσωτερικές κορυφές (*Selection* → *Vertex*) και τις μετατοπίζουμε στις άκρες του box με το εργαλείο **Scale**, ούτως ώστε τα εξωτερικά **Segments** να έχουν το ίδιο πάχος με το μήκος (length) του **Box**.



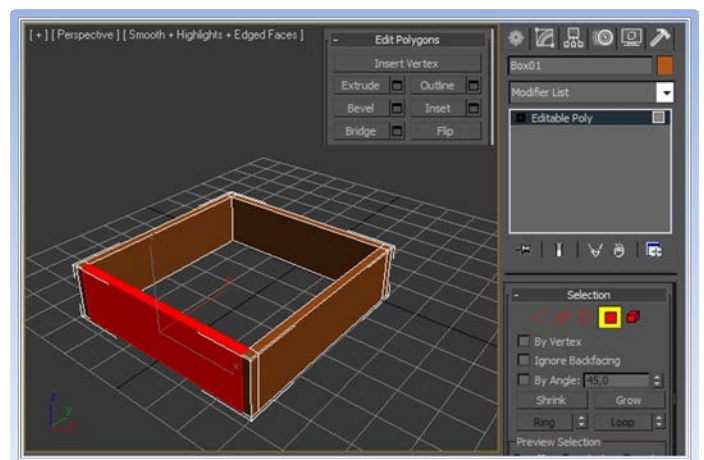
5. Επιλέγοντας τις επιφάνειες των δυο εξωτερικών πολυγώνων (*Selection* → *Polygon*) τις πρόσοψης του σχήματος, επιλέγουμε το εργαλείο **Extrude** (*Edit Polygons* → *Extrude*) και πληκτρολογούμε το επιθυμητό μήκος. Έτσι έχουμε δημιουργήσει τους πλαϊνούς τοίχους του κτιρίου.



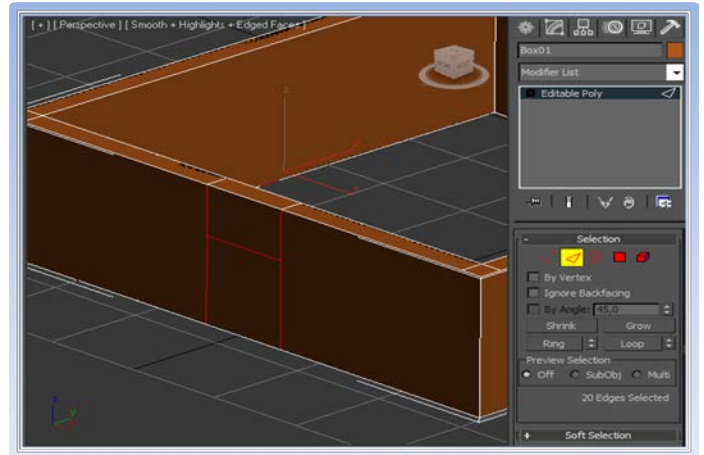
6. Για να κατασκευάσουμε την πρόσοψη επιλέγουμε **Edge** από την λίστα **Selection** και μαρκάρουμε τις οριζόντιες ακμές κατά μήκος των πλαϊνών box που μόλις δημιουργήσαμε. Χρησιμοποιούμε το εργαλείο **Connect** (*Edit Edge* → *Connect*) για να χωρίσουμε τα πλαϊνά box σε δυο τμήματα. Από το πεδίο **Slide** μπορούμε να μετακινήσουμε τις νέες ακμές στην μπροστινή μεριά, έτσι ώστε τα μικρότερα τμήματα να έχουν μήκος όσο το πάχος των υπολοίπων box.



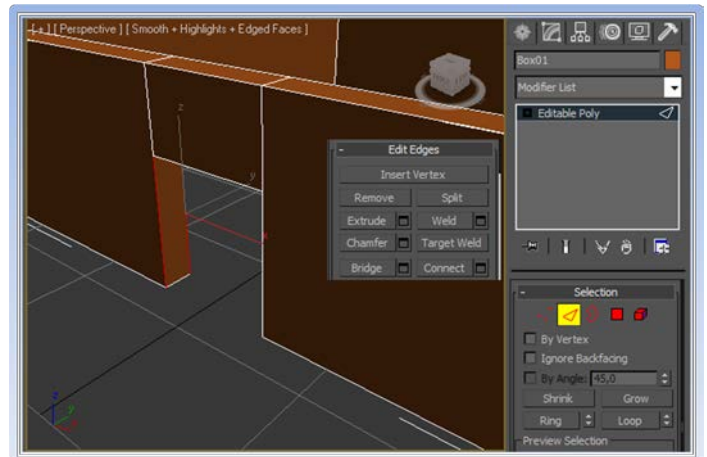
7. Επιλέγουμε τις δυο μικρές εσωτερικές επιφάνειες των νέων τμημάτων και επιλέγουμε το εργαλείο **Bridge** (*Edit Polygons* → *Bridge*). Με αυτό τον τρόπο δημιουργήσαμε ένα τετράγωνο πλαίσιο που αποτελεί τους εξωτερικούς τοίχους του κτιρίου.



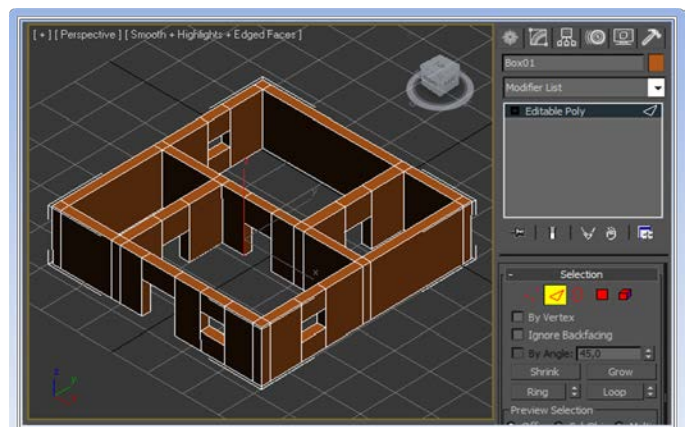
8. Για να κατασκευάσουμε το άνοιγμα της εισόδου, επιλέγουμε τις οριζόντιες ακμές του box που αποτελεί την πρόσοψη και χρησιμοποιούμε το εργαλείο **Connect** όπως περιγράψαμε παραπάνω για να δημιουργήσουμε δυο ακόμα **Segments**. Με την ίδια μέθοδο χωρίζουμε το μεσαίο segment σε δυο οριζόντιες τμήματα και με το εργαλείο Move τα φέρνουμε στην κατάλληλη θέση, ώστε να σχηματίζουν ένα κεντρικό ορθογώνιο παραλληλόγραμμο.



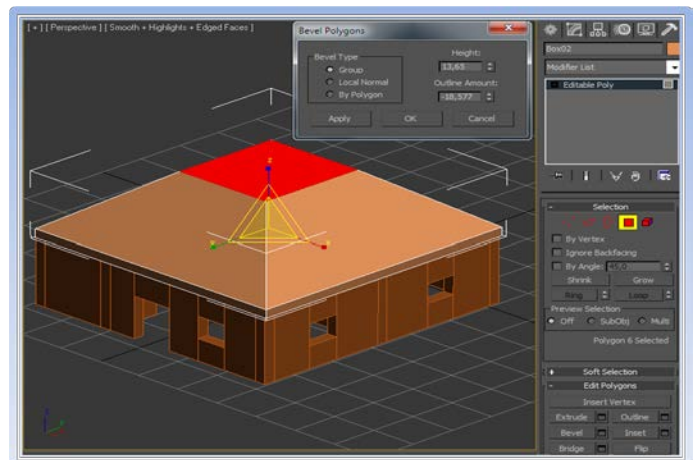
9. Τώρα, για να ανοίξουμε την είσοδο δεν έχουμε παρά να διαγράψουμε τα πολύγωνα του ορθογώνιου που σχηματίζεται, επιλέγοντας τις επιφάνειες τους με το **Polygon** (*Selection*→*Polygon*). Στην συνέχεια καλύπτουμε της κενές περιοχές που έχουν δημιουργηθεί, μαρκάροντας τις παράλληλες ακμές στο εσωτερικό της εισόδου με το εργαλείο **Bridge** (*Edit Edges*→*Bridge*).



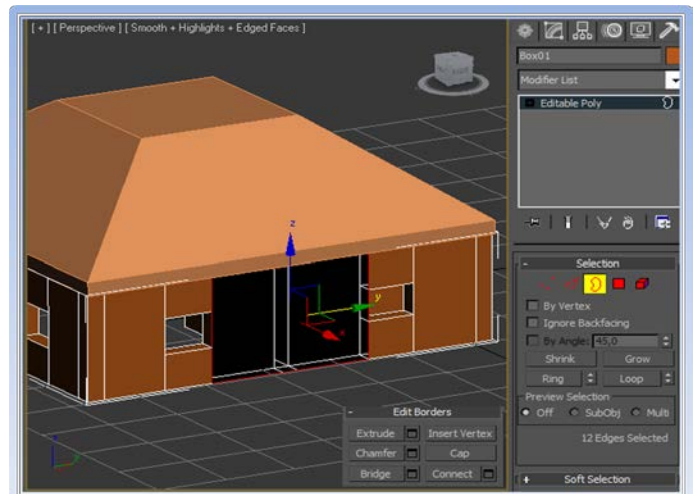
10. Για την κατασκευή παραθύρων χρησιμοποιούμε την ίδια μέθοδο, όπως παραπάνω. Ακόμα, μπορούμε να κατασκευάσουμε εσωτερικούς τοίχους και να χωρίσουμε το κτίριο σε δωμάτια και ορόφους.



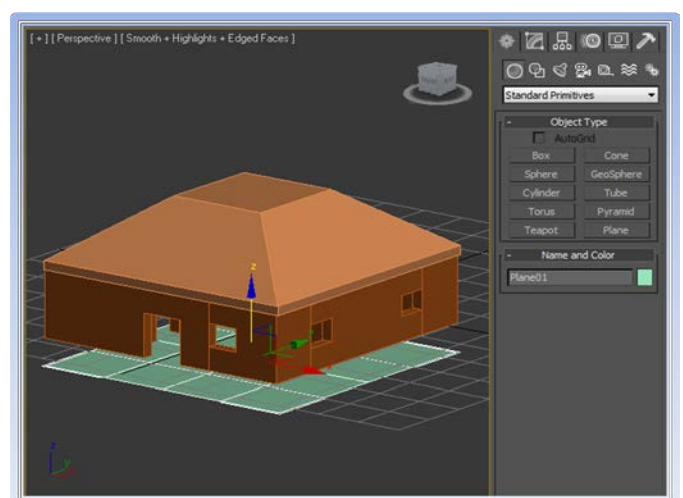
11. Για οροφή μπορούμε απλά να τοποθετήσουμε ένα σχήμα πυραμίδας (*Create* → *Standard Primitives* → *Pyramid*) ή ένα καινούριο σχήμα **Box** που θα το μετατρέψουμε κατάλληλα, ώστε να μοιάζει με σκέπη σπιτιού. Αυτό μπορεί να γίνει με την επιλογή της επάνω επιφάνειας του box και την χρήση του εργαλείου **Bevel** (*Edit Polygons*→*Bevel*), αφού το έχουμε μετατρέψει πρώτα σε **Editable Poly**.



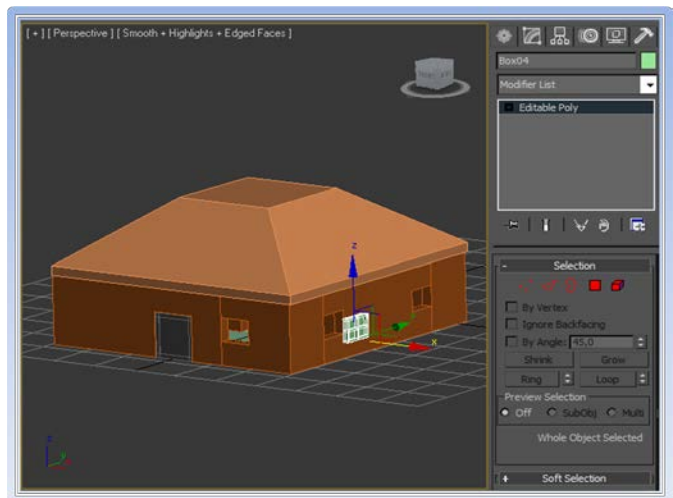
12. Μπορούμε να διαγράψουμε όλα εκείνα τα περιττά πολύγωνα που εξυπηρετήσαν μόνο για την κατασκευή του μοντέλου. Τα επιλέγουμε και πατάμε delete από το πληκτρολόγιο. Στην συνέχεια επιλέγουμε τις κενές περιοχές με το **Border** (*Selection*→*Border*) και χρησιμοποιούμε το εργαλείο **Cap** (*Edit Borders*→*Cap*) για να τις γεμίσουμε με ενιαία πολύγωνα, εξοικονομώντας όγκο δεδομένων.



13. Για δάπεδο μπορούμε να χρησιμοποιήσουμε ένα **Plane** (*Create* → *Standard Primitives* → *Plane*) ή ένα απλό **Box** στο μέγεθος τις κάτοψης του κτιρίου.

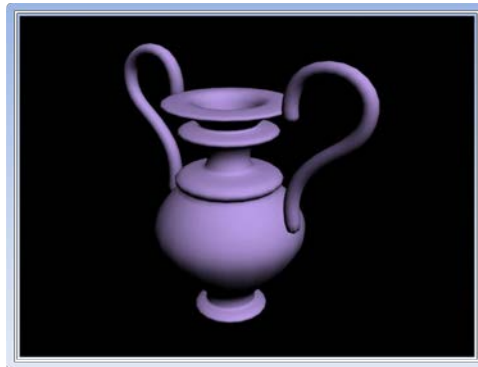


14. Τέλος, έχοντας κατασκευάσει την βασική μορφή του κτιρίου μπορούμε να του προσθέσουμε και άλλα στοιχεία όπως πόρτες, παραθύρα κ.τ.λ. Τα κουφώματα μπορούμε να τα κατασκευάσουμε με άπλα Boxes .



Εικόνα 2.3 Απεικόνιση κτιρίου σε τρισδιάστατο περιβάλλον.

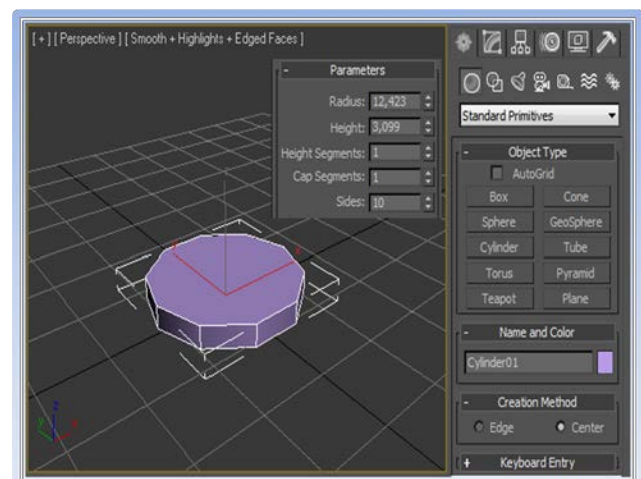
2.3.2 Παράδειγμα κατασκευής ενός δοχείου απο *Cylinders* με την μέθοδο *Polygon Modeling*.



Σε μοντέλα τα οποία περιέχουν καμπύλες απαιτούνται περισσότερα πολύγωνα για την απεικόνισή τους, ενώ μετά την ολοκλήρωση της διαδικασίας κατασκευής, ίσως χρειάζεται να λειανθούν οι επιφάνειες.

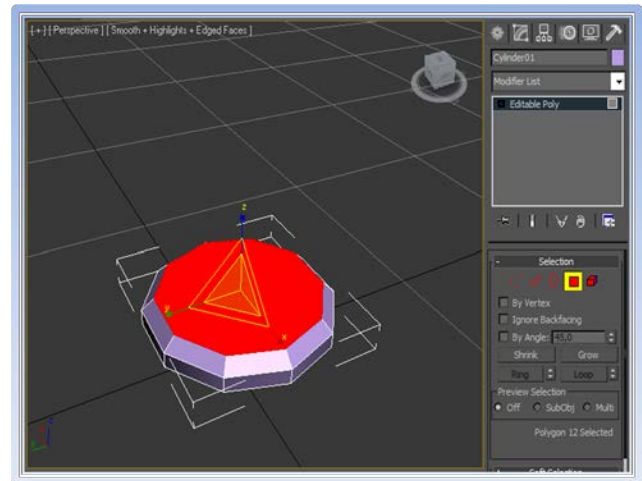
1. Αρχικά κατασκευάζουμε ένα σχήμα σε μορφή δίσκου χρησιμοποιώντας τον κύλινδρο από τα βασικά σχήματα. (*Standard Primitives* → *Cylinder*). Από τις παραμέτρους επιλέγουμε περίπου 10 πλευρές (**Sides**), και 1 **Segment** για το **Height** και **Cap**.

Εδώ αξίζει να αναφέρουμε ότι η εξοικονόμηση όγκου δεδομένων παίζει μεγάλο ρόλο σε 3D εφαρμογές real time. Για τον λόγο αυτό προσπαθούμε να πάρουμε το επιθυμητό αποτέλεσμα χρησιμοποιώντας όσο το δυνατόν λιγότερα πολύγωνα.

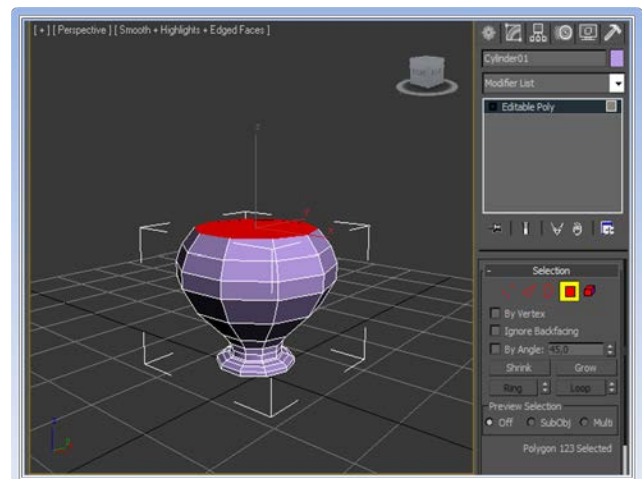


2. Μετατρέπουμε το αντικείμενο σε **Editable Poly** (*Mouse Right Click* → *Convert To* → *Editable Poly* ή *Modifier list* → *Edit Poly*), ούτως ώστε να μπορούμε να διαχειριστούμε τα πολύγωνα.

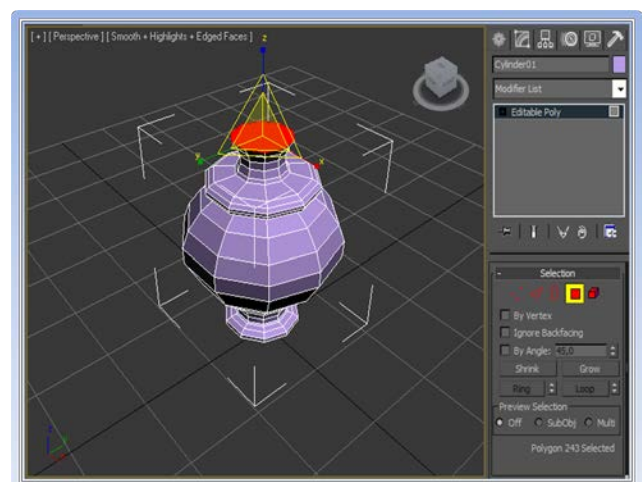
3. Χρησιμοποιώντας το **Polygon Selection** (*Selection*→*Polygon*) επιλέγουμε την πάνω επιφάνεια του δίσκου και με το εργαλείο **Extrude** (*Edit Polygons*→*Extrude*) το προεκτείνουμε στο επιθυμητό σημείο. Μετά από αυτό επιλέγουμε το εργαλείο **Scale** από το **Main Toolbar** και μειώνουμε ελάχιστα την διάμετρο της επιφάνειας για να κατασκευάσουμε τη βάση του δοχείου.



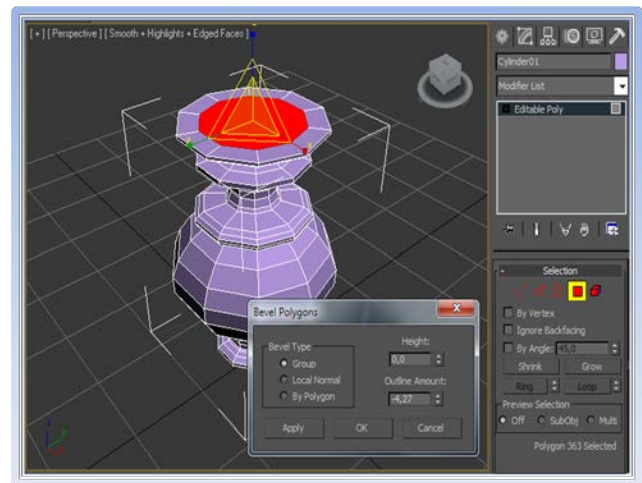
4. Επαναλαμβάνεται η ίδια διαδικασία αυξομειώνοντας όπου χρειάζεται την διάμετρο της επιφάνειας του πάνω πολυγώνου, ανάλογα με την επιθυμητή μορφή που θέλουμε να δώσουμε.



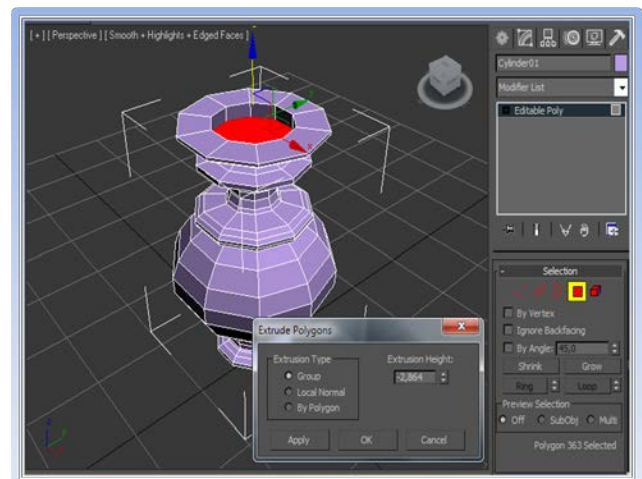
5. Προσπαθούμε να αναπτύξουμε τον σχηματισμό του δοχείου ελαττώνοντας την διάμετρό του από ένα σημείο και μετά, καθώς πλησιάζουμε στο στόμιο. Κάθε νέο επίπεδο πολυγώνων πρέπει να έχει μια αρμονική συνέχεια σε σχέση με το προηγούμενό του.



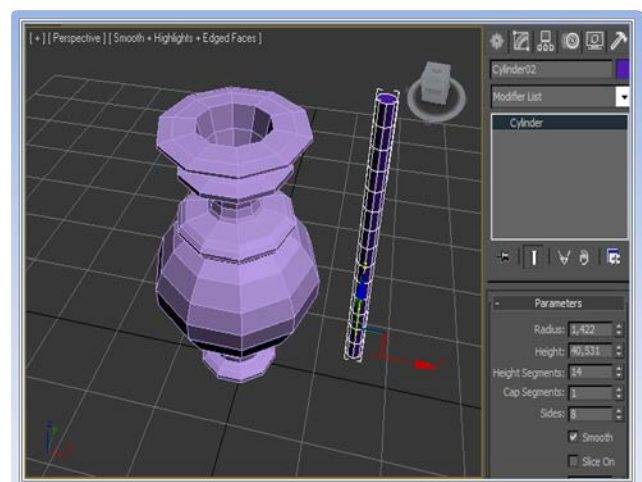
6. Για να δημιουργήσουμε το στόμιο του βάζου χρησιμοποιούμε το εργαλείο **Bevel** (*Edit Polygons*→*Bevel*) και μειώνουμε την διάμετρο του κεντρικού πολυγώνου που δημιουργήθηκε.



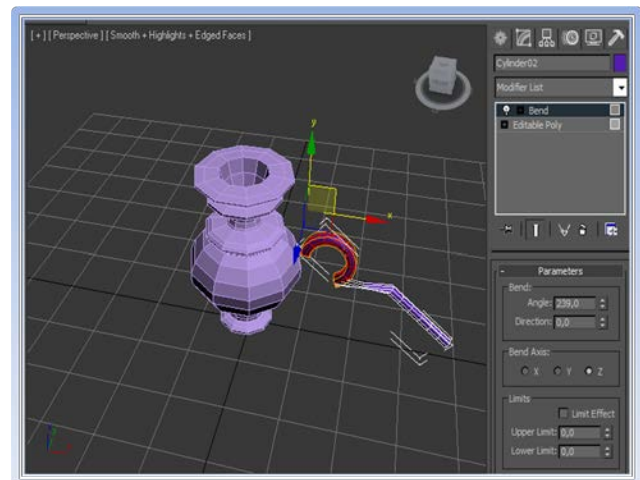
7. Στην συνέχεια επιλέγουμε **Extrude** (*Edit Polygons*→*Extrude*) και επιλέγουμε αρνητική τιμή για να δημιουργήσουμε μια εσοχή, ώστε να φαίνεται ότι το ανοικτό στόμιο του βάζου έχει βάθος. Μπορούμε στη συνέχεια να διαμορφώσουμε κατάλληλα το βάθος που επιθυμούμε, επαναλαμβάνοντας την διαδικασία και χρησιμοποιώντας τα εργαλεία **Scale** και **Move**.



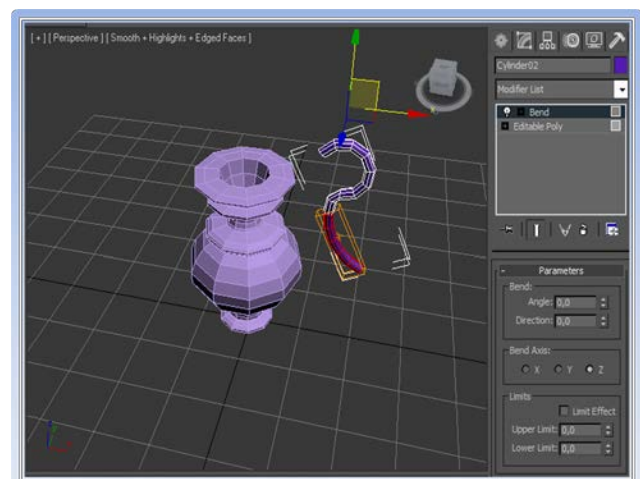
8. Μπορούμε ακόμα να κατασκευάσουμε λαβές αριστερά και δεξιά του βαζού. Για να γίνει αυτό κατασκευάζουμε μια μακρία ράβδο χρησιμοποιώντας ακόμα ένα κύλινδρο. Στις παραμέτρους επιλέγουμε 14 **Height Segments**, ενώ μπορούμε να μειώσουμε ακόμα περισσότερο τις πλευρές (**Sides**) σε 8.



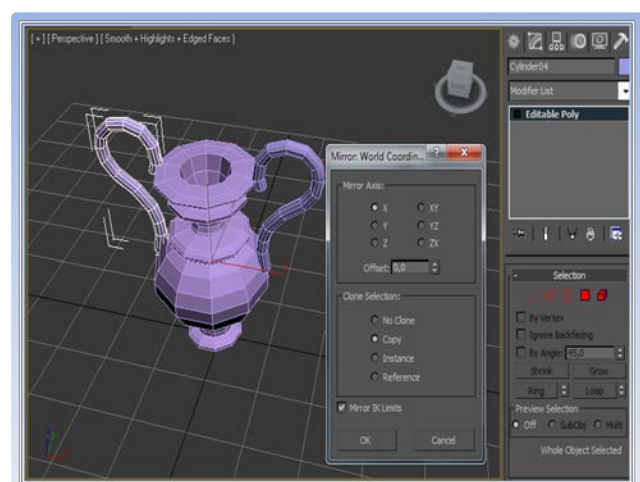
9. Μετατρέπουμε τον κύλινδρο σε **Editable Poly** και επιλέγουμε τα πολύγωνα από το μισό μέρος της ράβδου. Έχοντας επιλεγμένα τα μισά πολύγωνα, εφαρμόζουμε τον modifier **Bend** (*Modifier List* → *Bend*). Τώρα έχουμε την δυνατότητα να τα λυγίσουμε, έτσι ώστε να πάρουν την μορφή μιας καμπύλης.



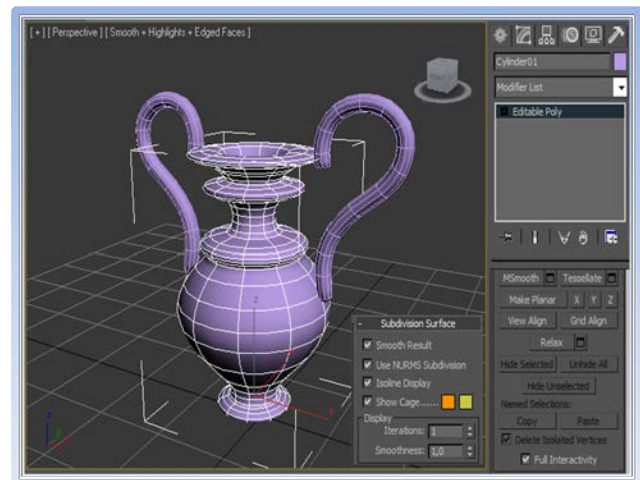
10. Με τον ίδιο τρόπο εφαρμόζουμε αυτόν τον modifier στα πολύγωνα του άλλου μισού μέρους της ράβδου στην αντίθετη κατεύθυνση, έτσι ώστε η ράβδος να πάρει τη μορφή ενός σίγμα(ς). Για περαιτέρω τροποποίηση επιλέγουμε της κατάλληλες ακμές ή κορυφές και χρησιμοποιούμε τα εργαλεία **Scale** και **Move**.



11. Μετά την ολοκλήρωση της κατασκευής της πρώτης λαβής και την κατάλληλη τοποθέτησή της στο πλάι του δοχείου επιλέγουμε το εργαλείο **Mirror** από το **Main Toolbar** και επιλέγουμε copy στην περιοχή του **Clone Selection**. Έτσι έχουμε αντιγράψει ένα είδωλο του αντικειμένου, δημιουργώντας και την δεύτερη λαβή του δοχείου.

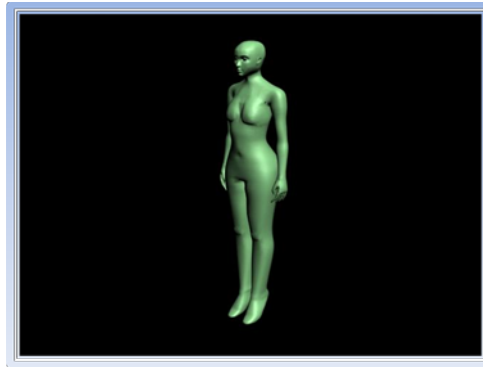


12. Τέλος, επιλέγουμε κάθε ένα από τα αντικείμενα και λειαίνουμε τις επιφάνειές τους τσεκάροντας το **Use NURMS Subdivision** από την λίστα **Subdivision Surface** (*Subdivision Surface* → *Use NURMS Subdivision*). Επιλέγουμε το σύνολο των αντικειμένων και κάνουμε ομαδοποίηση με την επιλογή **Group** από την λίστα εργασιών.



Εικόνα 2.4 Απεικόνιση δοχείου σε τρισδιάστατο περιβάλλον.

2.3.3 Παράδειγμα κατασκευής 3D χαρακτήρα(avatar) με συνδυασμό μέθοδων *Box Modeling*, *Spline Modeling* και *Polygon Modeling*.

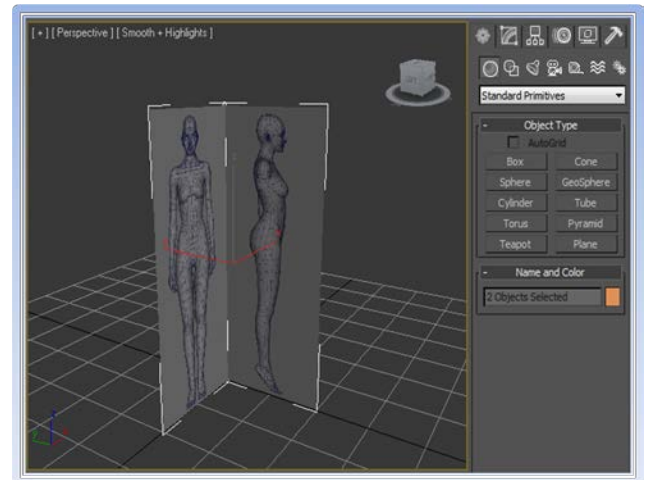


Η μοντελοποίηση ενός ανθρώπινου σώματος είναι μία από τις πιο ενδιαφέρουσες εργασίες σε 3D Modeling. Αλλά είναι επίσης μια από τις πιο δύσκολες διαδικασίες, ιδιαίτερα για την λεπτομερή απεικόνιση ενός προσώπου όπως θα δούμε παρακάτω. Σε αυτήν τη περίπτωση θα ακολουθεί ένας συνδυασμός τριών τεχνικών μοντελοποίησης: *Box Modeling*, *Polygon Modeling*, *Spline Modeling*.

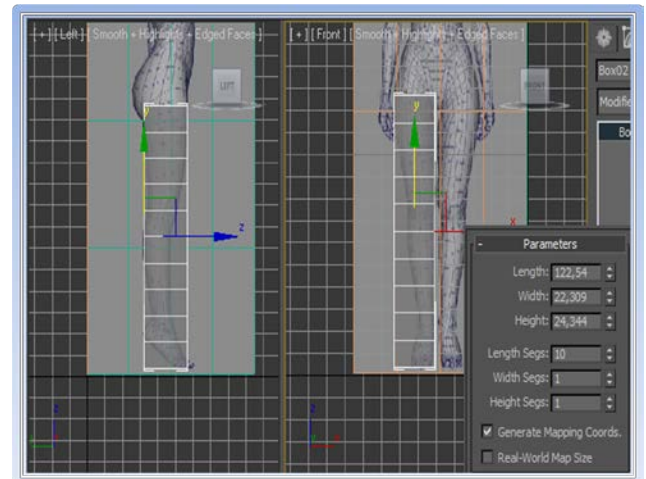
❖ 2.3.3.1 Μοντελοποίηση ανθρώπινου σώματος με την μέθοδο *Box Modeling* και *Polygon Modeling*.

1. Αρχικά χρειαζόμαστε μια εικόνα ενός ανθρώπου, κατά προτίμηση ένα σχέδιο που να φίνονται καθαρά τα μέρη του σώματος. Αν θέλετε, μπορείτε να σχεδιάσετε μόνοι σας ένα. Η εικόνα θα πρέπει να αποτελείται από την μπροστινή και την πλάγια όψη του σώματος. Μετά θα πρέπει να κοπεί σε δύο κομμάτια, το ένα για το Front View και το άλλο για το Side View.

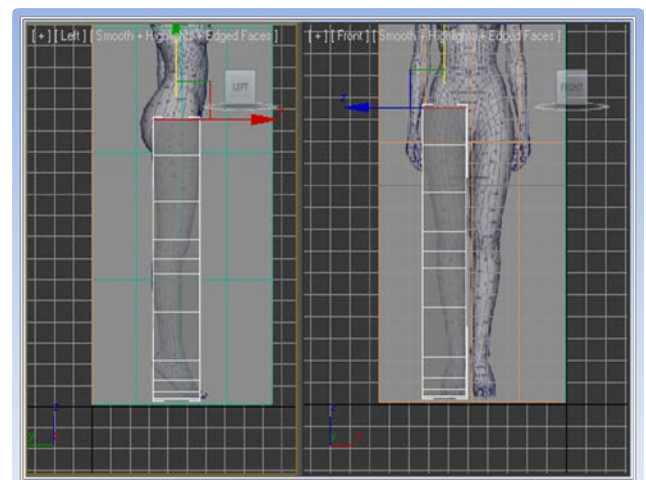
2. Δημιουργούμε δύο Planes (Create →Standard Primitives→Plane) στο μέγεθος της κάθε εικόνας και τα τοποθετούμε στην σκηνή κάθετα μεταξύ τους. Τραβάμε και αφήνουμε (drag and drop) την κάθε εικόνα στο κατάλληλο Plane, έτσι ώστε να φαίνεται η πρόσοψη στο Front View και το προφίλ στο Left View.



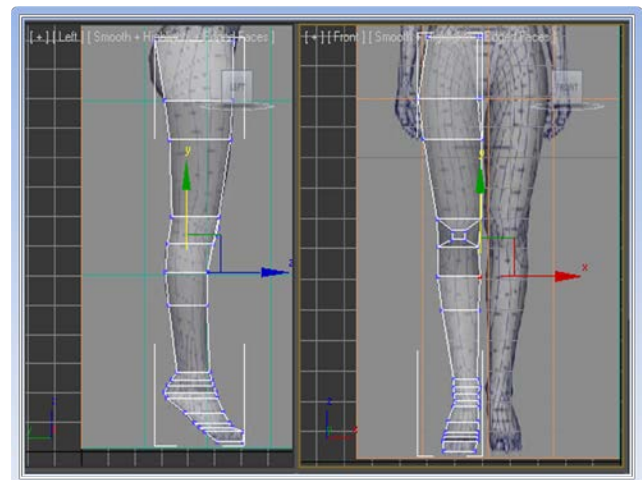
3. Στην συνέχεια κατασκευάζουμε ένα Box κατά μήκος του αριστερού ποδιού και τροποποιούμε τις παραμέτρους, ώστε να έχει περίπου το μέγεθος που χρειάζεται για να καλύπτει το πόδι. Επιλέγουμε να έχει τα κατάλληλα segments αναλόγως την λεπτομέρεια που χρειαζόμαστε και το μετατρέπουμε σε διάφανο σχήμα από το Object Properties του αντικειμένου (Mouse Right Click→ Object Properties →See Through).



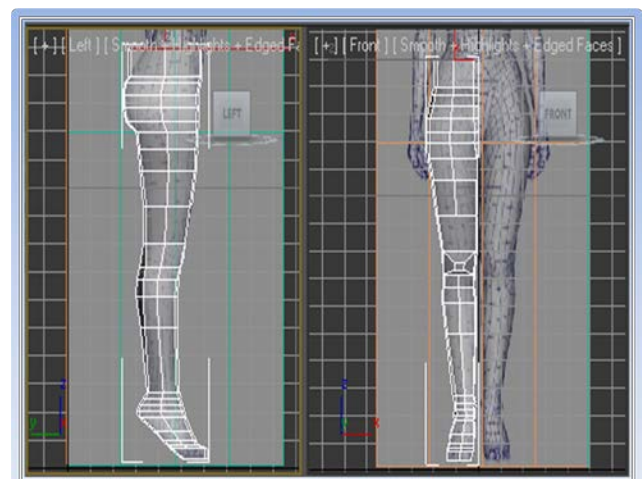
4. Αφού μετατρέψουμε το σχήμα σε Editable Poly, μετακινούμε τις οριζόντιες ακμές, ώστε να πάρουν θέση σε κομβικά σημεία μορφοποίησης του ποδιού. Επίσης προσέχουμε το σχήμα να βρίσκεται ακριβώς στη μέση του σώματος που απεικονίζεται στη φωτογραφία.



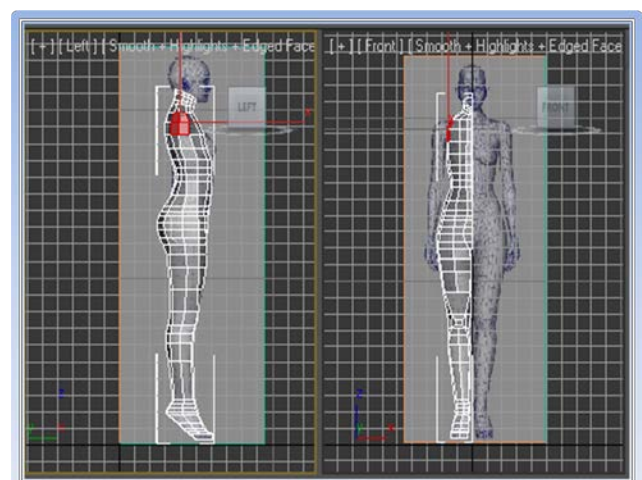
5. Αρχίζοντας την μορφοποίηση του Box μπορούμε να προσθέσουμε ή να αφαιρέσουμε ακμές στα σημεία τα οποία κρίνουμε ότι χρειάζονται με το εργαλείο Connect όπως έχουμε δει. Αρχίζουμε να δίνουμε την κατάλληλη μορφή στο μοντέλο μας μετακινώντας της κορυφές του Box στα όρια του ποδιού από την μπροστινή και πλάγια όψη.



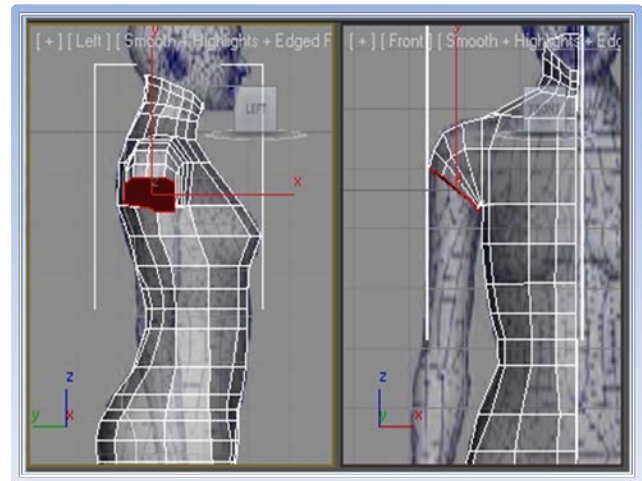
6. Μπορούμε να προσθέσουμε περισσότερη λεπτομέρεια προσθέτοντας ενδιάμεσες ακμές με το εργαλείο Connect ή με το εργαλείο Cut από την λίστα Edit Geometry (Edit Geometry → Cut) κόβοντας όπου χρειάζεται το πολύγωνο. Έχοντας ένα ικανοποιητικό αποτέλεσμα από το πόδι, αναπτύσσουμε το μοντέλο σταδιακά για το υπόλοιπο σώμα, χρησιμοποιώντας το εργαλείο Extrude στην επιφάνεια του πάνω πολυγώνου.



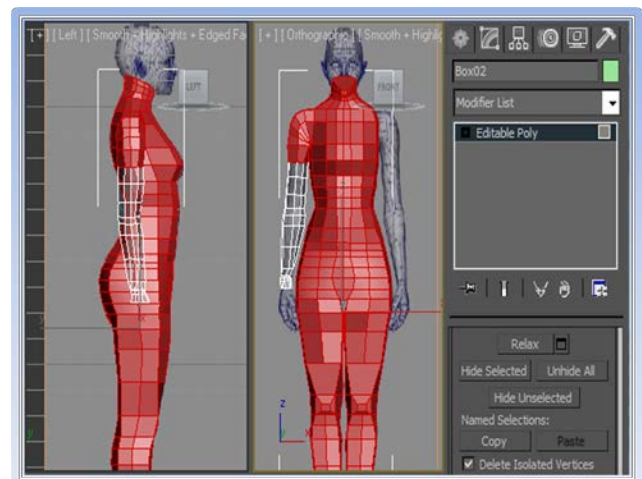
7. Έχοντας φτάσει την ανάπτυξη του σώματος στο σημείο κάτω από το κεφάλι χρησιμοποιώντας τα γνωστά εργαλεία που έχουμε αναφέρει μέχρι τώρα, ξεκινάμε να κατασκευάσουμε το αριστερό χέρι του μοντέλου.



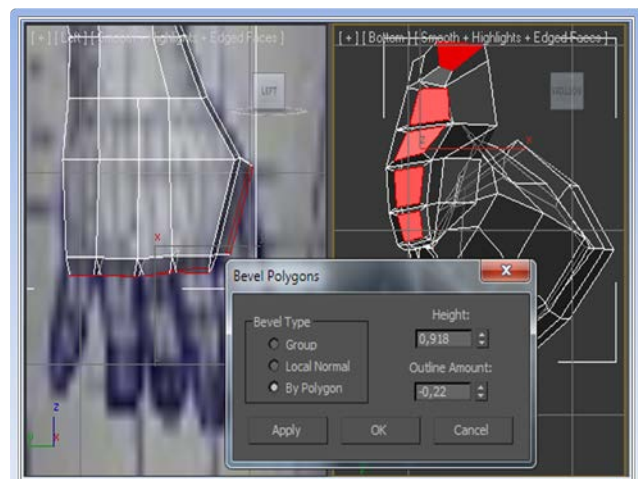
8. Με την ίδια διαδικασία και με το εργαλείο Extrude προεκτείνουμε τα πολύγωνα από το ύψος του ώμου και μετακινούμε τις κορυφές τους στα κατάλληλα σημεία με το εργαλείο Move ή αυξομειώνουμε το μέγεθος των πολυγώνων με το εργαλείο Scale. Προσπαθούμε να δώσουμε μια κυλινδρική μορφή στο χέρι, και παράλληλα να αποτελεί μια αρμονική συνέχεια του σώματος.



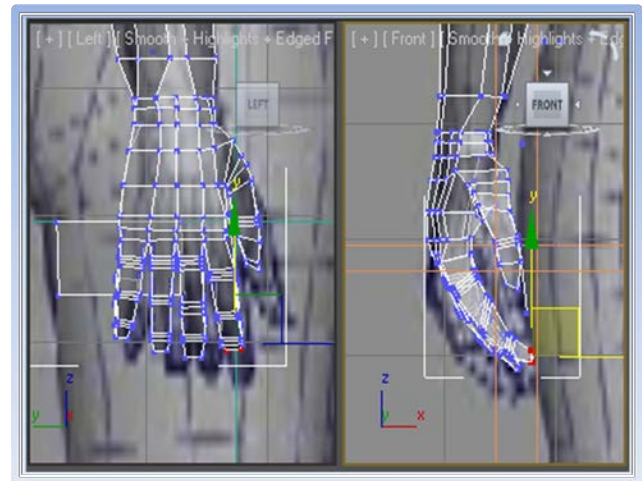
9. Αφού κατασκευάσουμε το χέρι μέχρι το σημείο του καρπού, το δυσκολότερο τμήμα είναι η δημιουργία της παλάμης και των δακτύλων. Εδώ χρειάζεται προσοχή για να έχουμε όσο το δυνατόν πιο φυσικό αποτέλεσμα. Για δική μας ευκολία, μπορούμε προς το παρόν να κρύψουμε τα πολύγωνα που δεν χρειαζόμαστε και που μας δυσκολεύουν να διακρίνουμε αυτά τα όποια μας ενδιαφέρουν. Αυτό μπορούμε να το κάνουμε μαρκάροντας τα πολύγωνα του υπόλοιπου σώματος και επιλέγοντας το κουμπί Hide Selected από την καρτέλα Edit Geometry.



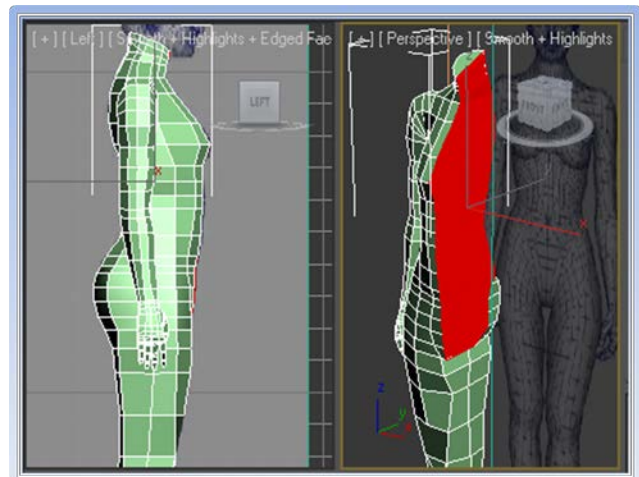
10. Προεκτείνουμε και τροποποιούμε κατάλληλα τα πολύγωνα με τέτοιο τρόπο, ώστε να δημιουργούν προϋποθέσεις για την ανάπτυξη των δακτύλων. Χωρίζουμε την παλάμη σε 5 τμήματα χρησιμοποιούμε το Bevel επιλέγοντας By Polygon στην περιοχή του Bevel Type.



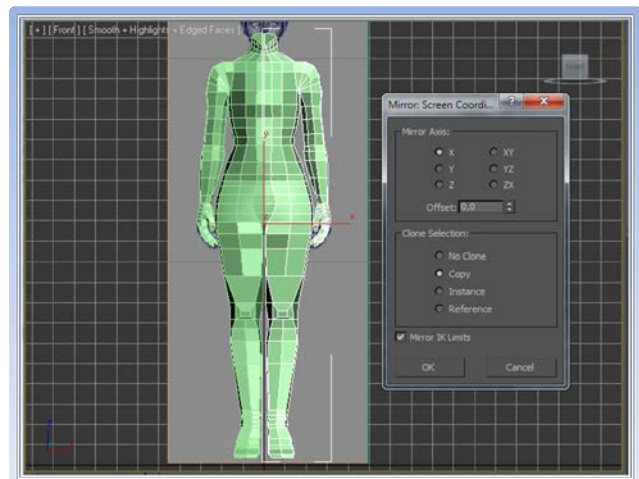
11. Στην συνέχεια προεκτείνουμε τα δάκτυλα και κάνουμε τις τροποποιήσεις που απαιτούνται για να πάρουν την θέση και την μορφή που θέλουμε. Προσπαθούμε να έχουν μια πιο λεπτή μορφή στην κορυφή τους ενώ στις αρθρώσεις τους δημιουργούμε περισσότερα τμήματα, ώστε να μπορούν να λυγίσουν ομαλά αν το επιθυμούμε. Αφού ολοκληρώσουμε το χέρι χρησιμοποιούμε το κουμπί Unhide All από το Edit Geometry για να εμφανιστούν ξανά όλα τα πολύγωνα.



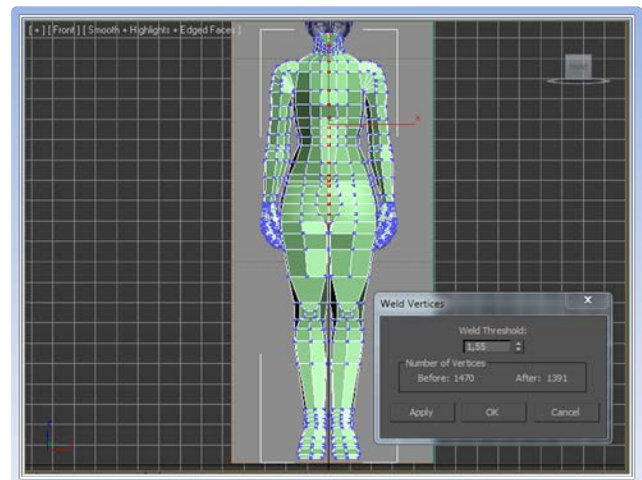
12. Αφού ολοκληρώσαμε το χέρι μπορούμε πλέον να αποεπιλέξουμε το See Through από το παράθυρο του Object Properties, γαι να μπορέσουμε να δούμε ποιο ξεκάθαρα το μοντέλο μας. Μέχρι τώρα έχει κατασκευαστεί το αριστερό κομμάτι του σώματος. Αφού είναι συμμετρικό ως προς το δεξιό του κομμάτι, το μονό που έχουμε να κάνουμε είναι δημιουργήσουμε ένα είδωλο του μοντέλου στην αντίστοιχη δεξιά μεριά και να ενώσουμε τα επιμέρους κομμάτια. Πρίν όμως από αυτό θα πρέπει να διαγράψουμε όλα τα πολύγωνα που βρίσκονται στα σημεία εκείνα όπου τα δύο μοντέλα θα ενωθούν.



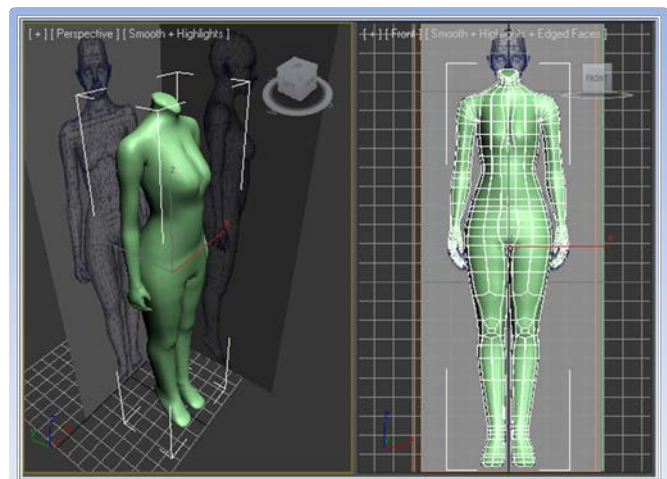
13. Έχοντας διαγράψει τα ενδιάμεσα πολύγωνα επιλέγουμε το εργαλείο Mirror και δημιουργούμε ένα αντίγραφο του μοντέλου στην δεξιά πλευρά. Στην σκηνή τώρα έχουμε δύο αυτόνομα μοντέλα. Αυτό σημαίνει ότι δεν μπορούμε να τα τροποποιήσουμε σαν σύνολο. Γι'αυτό το λόγο επιλέγουμε το αρχικό μοντέλο και πατάμε το εργαλείο Attach από την λίστα Edit Geometry. Στη συνέχεια επιλέγουμε το δεύτερο μισό του σώματος, ούτως ώστε να μετατραπεί σε ένα ενιαίο σύνολο.



14. Για να τα ενώσουμε την δεξιά με την αριστερή πλευρά, επιλέγουμε όλες τις ενδιάμεσες κορυφές των δυο κομματιών και πατάμε Weld από την λίστα Edit Vertices. Έτσι έχουμε πλέον ένα ενιαίο μοντέλο, ολοκληρώνοντας την διαδικασία της μοντελοποίησης.

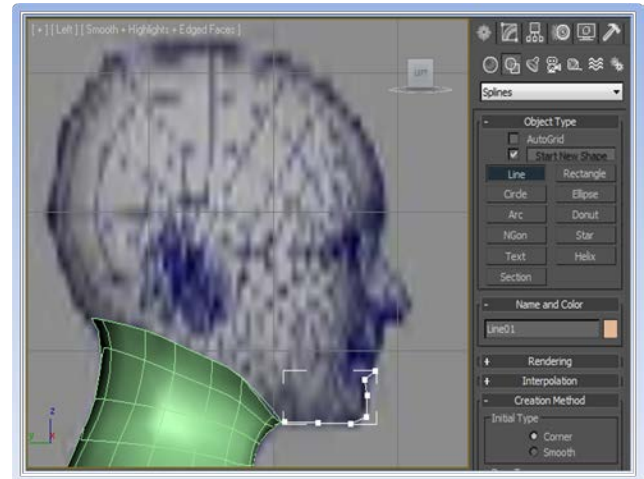


15. Τέλος λειαίνουμε την επιφάνεια του σώματος, τσεκάροντας το Use NURMS Subdivision.

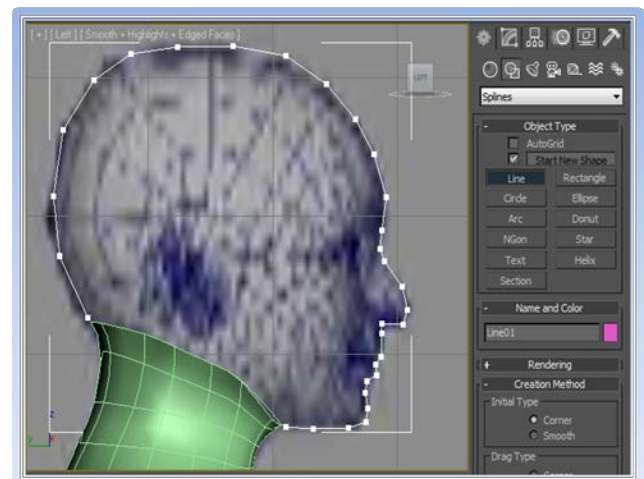


❖ 2.3.3.2 Μοντελοποίηση ανθρώπινου κεφαλιού με την μέθοδο *Spline Modeling*.

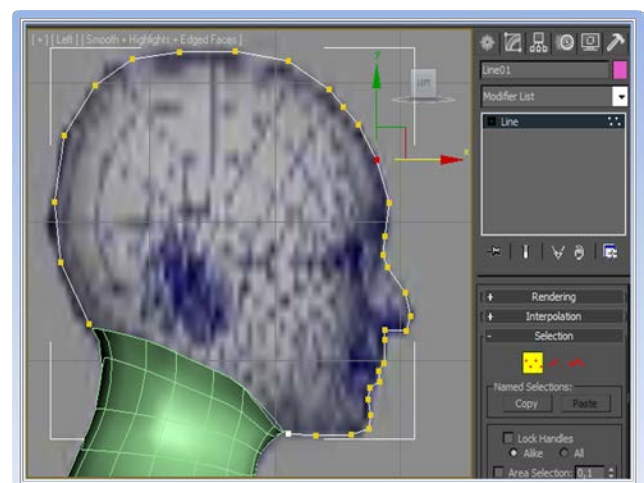
1. Στην διαδικασία μοντελοποίησης με την μέθοδο Spline Modeling θα χρησιμοποιήσουμε μια ακολουθία γραμμών και σημείων για να σχεδιάσουμε την περίμετρο του κεφαλιού. Επιλέγουμε το σχήμα **Line** από την καρτέλα **Create** (*Create*→*Shapes*→*Splines*→*Line*) και επιλέγουμε **Corner** στο πεδίο **Initial Type**. Σχεδιάζουμε την μορφή του κεφαλιού, έχοντας σαν οδηγό την εικόνα που βλέπουμε στο **Left View**.



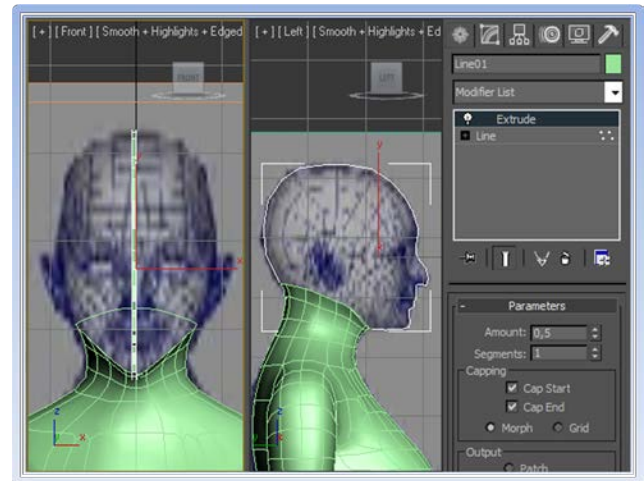
2. Προσπαθούμε να απεικονίσουμε την λεπτομέρεια που χρειάζεται στα χείλη τη μύτη κ.τ.λ. χωρίς όμως να υπερβάλουμε στην χρήση σημείων. Ακολουθούμε την διαδικασία ωστόσο φτάσουμε στο πίσω μέρος του κεφαλιού.



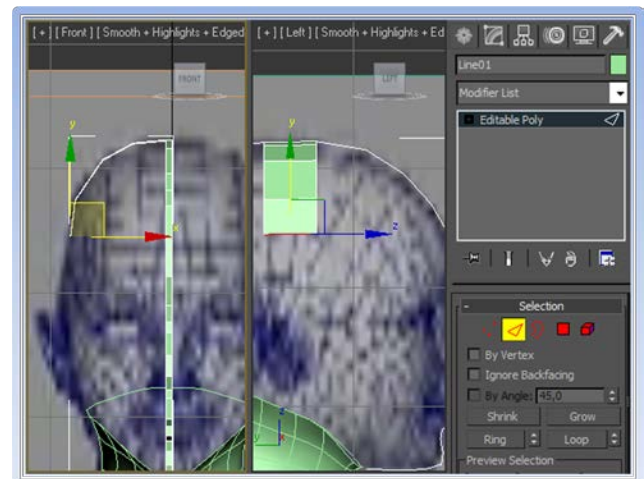
3. Από την καρτέλα **Modify** μας δίνεται η δυνατότητα να τροποποιήσουμε τα σημεία τα οποία δημιουργήσαμε, χρησιμοποιώντας την επιλογή **Vertex** (*Selection* →*Vertex*). Επιλέγοντας το **Segment** (*Selection* →*Segment*) μπορούμε να προσθέσουμε σημεία ανάμεσα στις ακμές, χρησιμοποιώντας το εργαλείο **Divide** από την λίστα **Geometry**.



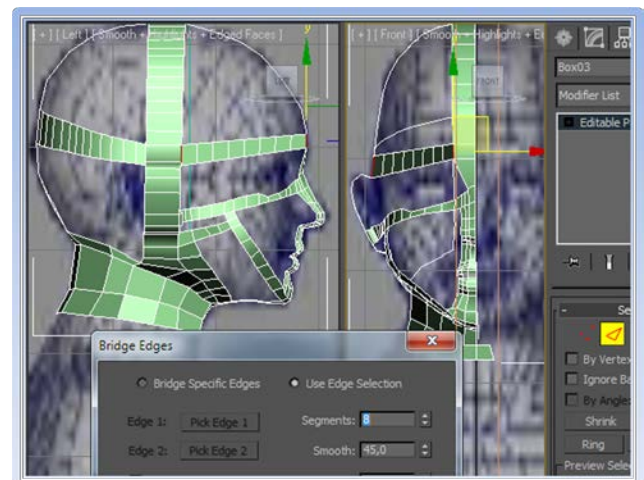
4. Μόλις έχουμε ένα ικανοποιητικό αποτέλεσμα επιλέγουμε Extrude από τη λίστα Modifier (*Modifier List* → *Extrude*) και πληκτρολογούμε μια τιμή στο πεδίο Amount από τις παραμέτρους, ώστε να δημιουργηθεί μια λεπτή λωρίδα πολυγώνων περιμετρικά του κεφαλιού από την γραμμή που σχεδιάσαμε. Στην συνέχεια μετατρέπουμε το σχήμα σε Editable Poly και αν η επιφάνεια των πολυγώνων βρίσκεται από την εσωτερική μεριά, τα επιλέγουμε και χρησιμοποιούμε το εργαλείο Flip (*Edit Polygons* → *Flip*).



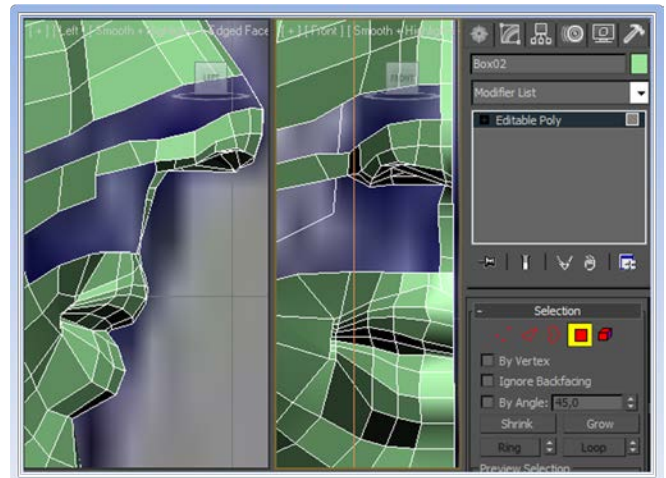
5. Στην συνέχεια από το Front View με το εργαλείο Move επιλέγουμε μια ακμή από την κορυφή του κεφαλιού. Πιέζοντας το Shift από το πληκτρολόγιο τραβάμε οριζόντια στην επιφάνεια του κεφαλιού και προεκτείνουμε το νέο πολύγωνο που δημιουργείτε κάθετα στην λωρίδα που ήδη έχουμε. Επαναλαμβάνουμε την διαδικασία μέχρι να διαγράψουμε μια νέα λωρίδα μέχρι τη βάση του κεφαλιού. Από εδώ και στο εξής θα μορφοποιήσουμε το αριστερό μέρος του κεφαλιού και όταν ολοκληρωθεί θα δημιουργήσουμε ένα είδωλο για το δεξιό μέρος, όπως κάναμε και στην προηγούμενη περίπτωση με το σώμα.



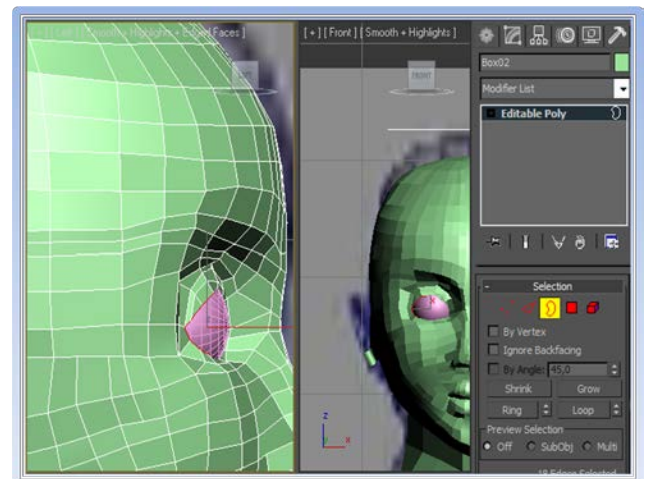
6. Έχοντας δημιουργήσει δύο λωρίδες που διαγράφουν τα βασικά μέρη του αριστερού κομματιού του κεφαλιού, προσπαθούμε να χωρίσουμε τις περιοχές σε επιμέρους τμήματα. Επιλέγουμε τις απέναντι ακμές από τις δυο λωρίδες και χρησιμοποιούμε το εργαλείο Bridge (*Edit Edges* → *Bridge*) για να δημιουργήσουμε "γέφυρες" πολυγώνων μεταξύ τους. Σε κάθε γέφυρα επιλέγουμε τα κατάλληλα Segments, ώστε δώσουμε στη συνέχεια την μορφή που χρειάζεται με τα εργαλεία Move, Scale και Rotation.



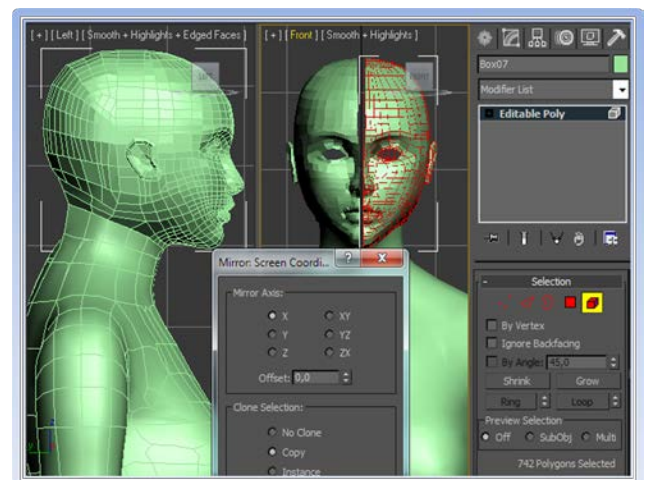
7. Η Δημιουργία του προσώπου είναι αρκετά χρονοβόρα και χρειάζεται αρκετή λεπτομέρεια σε σημεία όπως η μύτη και το στόμα. Τροποποιούμε τα πολύγωνα χρησιμοποιώντας διάφορα εργαλεία απ'αυτά που έχουμε δει μέχρι τώρα, όπως είναι το Cut το Bridge και το Connect.



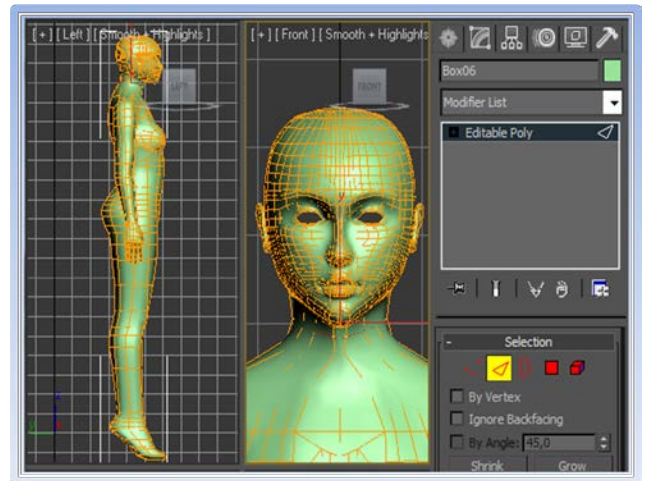
8. Στο σημείο του ματιού χρησιμοποιούμε μια σφαίρα (*Create Standard* → *Primitives Sphere*) σαν οδηγό, για μπορούμε να μορφοποιήσουμε σωστά την περιοχή.



9. Αφού ολοκληρώσουμε το αριστερό μισό του κεφαλιού, χρησιμοποιούμε το Mirror για να δημιουργήσουμε το άλλο μισό. Μετατρέπουμε τα κομμάτια σε ένα ενιαίο σύνολο με το εργαλείο Attach. Όπως είδαμε και παραπάνω μαρκάρουμε τις ενδιάμεσες κορυφές των δύο κομματιών και επιλέγουμε Weld. Αν κάποια σημεία δεν ενωθούν με αυτή την διαδικασία, επειδή ίσως οι κορυφές δεν βρίσκονται στο κατάλληλο σημείο, μπορούμε να χρησιμοποιήσουμε το Target Weld για να τις ενώσουμε μια προς μια.



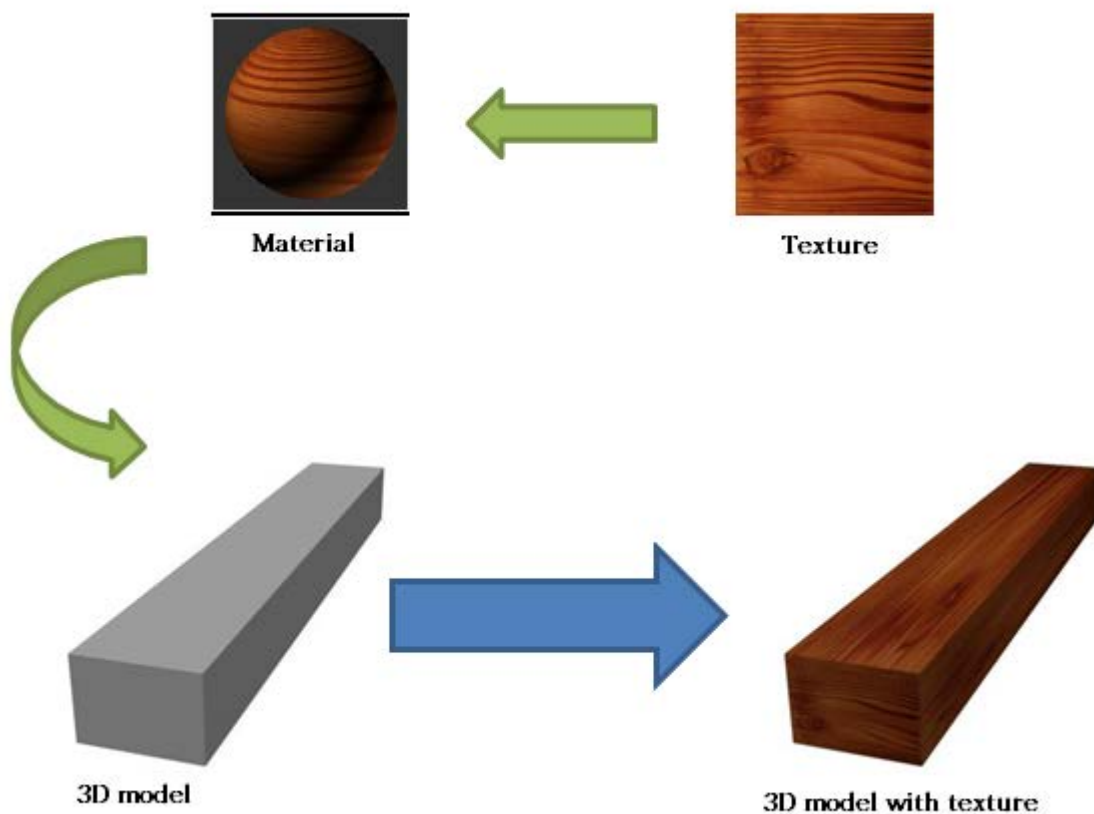
10. Τέλος, επιλέγουμε Use NURMS Subdivision για να λειάνουμε την επιφάνεια και ακολουθούμε την ίδια μέθοδο χρησιμοποιώντας το Attach και το Weld για να ενώσουμε το κεφάλι με το υπόλοιπο σώμα.



Εικόνα 2.5 Απεικόνιση αναταρ σε τρισδιάστατο περιβάλλον.

2.4 Προσαρμογή Υφής

Ένα τρισδιάστατο μοντέλο εκτός από τη βασική μορφή που αποκτάει κατά την διαδικασία της μοντελοποίησης, χρειάζεται περαιτέρω στοιχεία για να ολοκληρωθεί. Το επόμενο βασικό συστατικό απ'το οποίο αποτελείται ένα ολοκληρωμένο μοντέλο είναι η υφή του. Στην ουσία η υφή είναι η εικόνα που θα προσαρμοστεί πάνω του ώστε να αποκτήσει την κατάλληλη λεπτομέρεια που απαιτείται σε κάθε περίπτωση . Σε ένα απλό σχηματικά μοντέλο μπορεί να χρειάζεται να απεικονίσουμε μονάχα το υλικό ή το χρώμα απ' το οποίο αποτελείται. Για παράδειγμα σε ένα μοντέλο που απεικονίζει ένα κομμάτι ξύλο , αρκεί να προσαρμόσουμε πάνω του μια εικόνα μιας ξύλινης επιφάνειας , έχοντας ένα ικανοποιητικό αποτέλεσμα.



Εικόνα 2.6 Διαδικασία προσαρμογής υφής σε 3d model

Στην περίπτωση όμως πιο συνθέτων και πολύπλοκων μοντέλων που θέλουμε να ελέγξουμε την υφή, την εικόνα και την λεπτομέρεια που θα αποτυπώνεται σε κάθε σημείο, χρειαζόμαστε μια διαφορετική προσέγγιση. Μέσα από μια σειρά διαφόρων maps (texture map, normal map, light map, shadow map) που μπορούμε να προσαρμόσουμε σε ένα μοντέλο, επιτυγχάνουμε αρκετά ικανοποιητικά αποτελέσματα σχετικά με την απόδοση λεπτομέρειας και ρεαλιστικότητας. Ένα από τα βασικότερα maps αναφέρεται στην προσαρμογή της υφής (texture map). Ένα texture map στην ουσία είναι ένας οδηγός που βοηθάει να αντιστοιχηθεί κάθε σημείο μιας 2D εικόνας με συγκεκριμένες περιοχές στην επιφάνεια του μοντέλου.

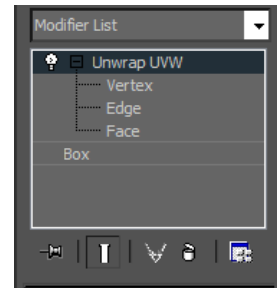
Μέσα από το 3ds Max μπορούμε να ακολουθήσουμε μια τεχνική για την δημιουργία τέτοιων ειδών maps. Στην λίστα των modifiers που μας παρέχει το πρόγραμμα υπάρχει η επιλογή Unwrap UVW. Αυτός ο Modifier μας επιτρέπει να "ξεδιπλώσουμε" το μοντέλο με την διαδικασία unwrapping, σε μια επιφάνεια δύο διαστάσεων, ώστε να είναι εμφανή όλα τα πολύγωνα. Με αυτόν τον τρόπο δημιουργείται ένα UVW map όπου κάθε στοιχείο του αντιπροσωπεύει ένα στοιχείο του μοντέλου και έτσι μπορούμε να εφαρμόσουμε την υφή και το χρωματισμό που επιθυμούμε σε κάθε πολύγωνο ξεχωριστά.

2.4.1 Modifier Unwrap UVW

Κατά την διαδικασία του unwrapping χωρίζεται η επιφάνεια του μοντέλου σε διάφορες περιοχές και "κοβεται" στα σημεία εκείνα που χρειάζεται ώστε να μπορεί να ξεδιπλωθεί το πλέγμα των πολυγώνων πάνω σε ένα πάνελ. Αυτή η διαδικασία μπορεί να υλοποιηθεί με κάποιες έτοιμες τεχνικές που προσφέρονται από το το πρόγραμμα και είναι χρήσιμες συνήθως για άπλα meshes. Σε περιπτώσεις οι οποίες απαιτείτε μεγαλύτερη ακρίβεια υπάρχει η δυνατότητα επιλογής των σημείων κοπής από τον χρήστη και το ξεδίπλωμα των περιοχών από κάθε κορυφή που βρίσκεται στα όρια της ξεχωριστά.

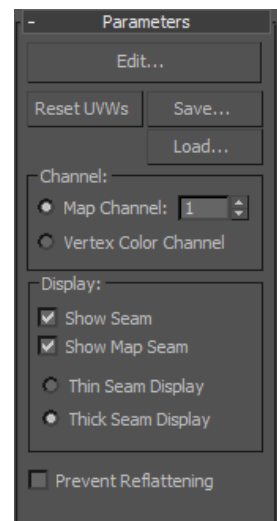
Αναλύοντας περισσότερο την διαδικασία κατά την οποία εφαρμόζουμε τον Unwrap UVW modifier θα δούμε τα βήματα που ακολουθούμε τις επιλογές που μας προσφέρονται.

Σε αυτήν την τεχνική μας δίνεται η δυνατότητα να διαχειριστούμε τις κορυφές, τις ακμές και της επιφάνειες των πολυγώνων του UVW map που δημιουργούμε.



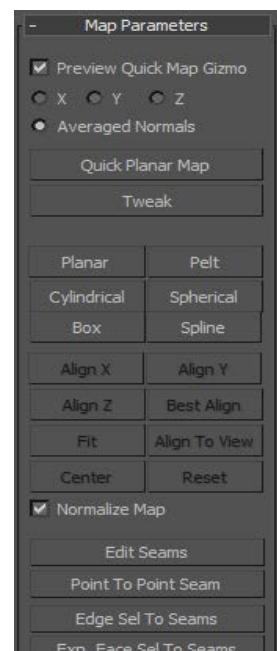
Σχήμα 2.13 Εφαρμογή Modifier Unwrap UVW

Μέσα από τις παραμέτρους δίνονται επιλογές για την διαχείριση της περιοχής του πάνελ πάνω στο οποίο κατασκευάζεται το texture map. Από το Edit εμφανίζεται το πάνελ ενώ υπάρχει η δυνατότητα να αποθηκεύσουμε, να φορτώσουμε ή να επαναπροσδιορίσουμε της τροποποιήσεις στα UVW maps. Από την περιοχή του Channel μπορούμε να αλλάξουμε ή να δημιουργήσουμε περισσότερα κανάλια στα οποία θα εμφανίζονται τα maps, σε περίπτωση που χρησιμοποιούνται περισσότερα του ενός. Στο Display υπάρχουν κάποιες επιλογές που αφορούν την τομές που έχει χωριστεί το μοντέλο στις επιμέρους επιφάνειες.



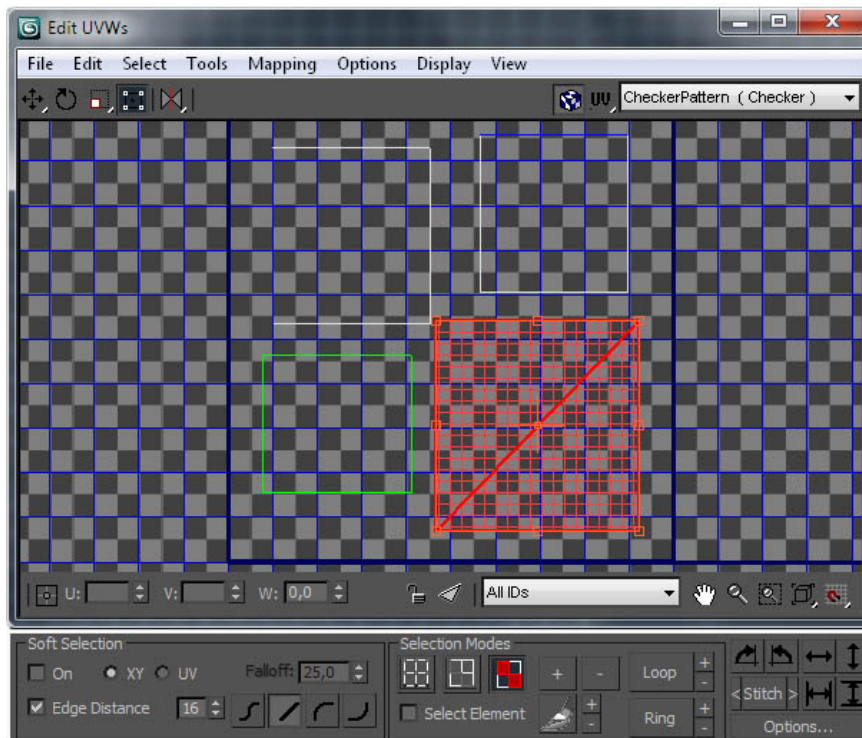
Σχήμα 2.14 Unwrap UVW Parameters Panel

Στις παραμέτρους του χάρτη υπάρχουν εργαλεία για την χρήση έτοιμων gizmos: Planar, Pelt, Cylindrical, Spherical, Box και Spline · όπου υποστηρίζουν την “κοπή” και το “ξεδίπλωμα” της επιφάνειας με συγκεκριμένη τεχνική στους άξονες X, Y και Z, ανάλογα με την επιλογή του χρήστη. Επίσης υπάρχουν εργαλεία που επιτρέπουν την διαχείριση της “κοπής” στα επιθυμητά σημεία επιλέγοντας μια-μια τις κορυφές ή τις ακμές απ’που θα δημιουργηθεί η τομή.



Σχήμα 2.15 Unwrap UVW Map Parameters Panel

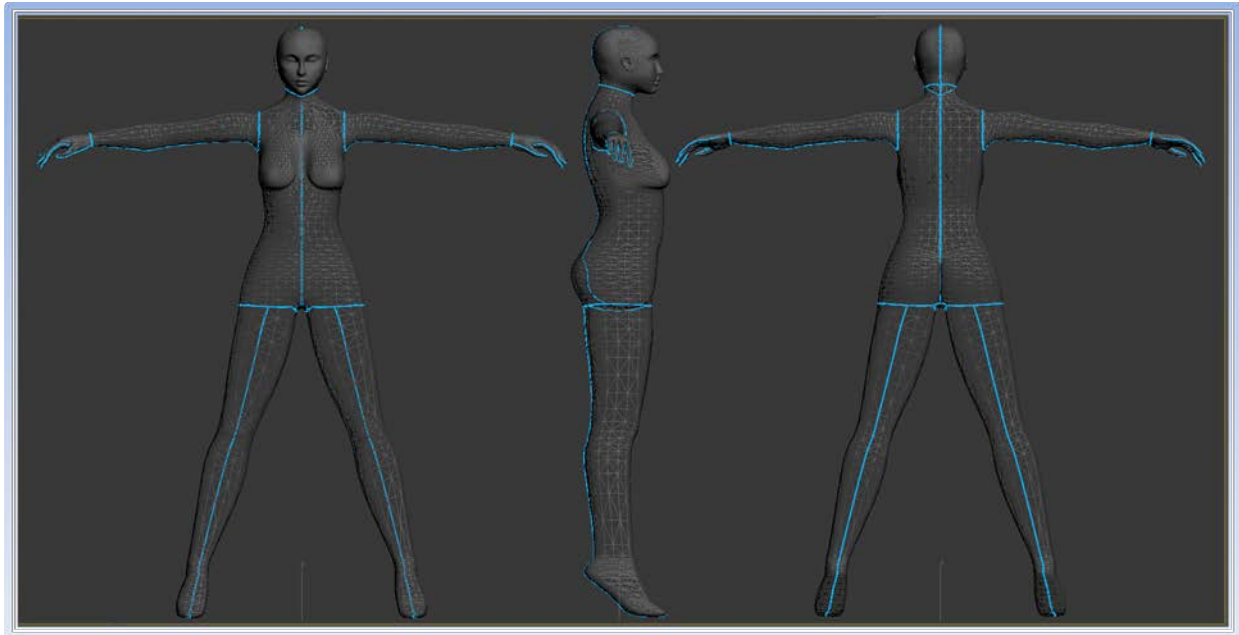
Το πάνελ του UVW επιτρέπει την διαχείριση της επιφάνειας των τμημάτων του μοντέλου που έχουν ξεδιπλωθεί, μέσα από γνωστά εργαλεία, όπως είναι το Move, Scale και Rotation για την σωστή απεικόνιση όλων των πολυγώνων, άλλα και μέσα από πολλά άλλα εργαλεία, που προσφέρουν επιπλέον δυνατότητες.



Σχήμα 2.16 UVW Panel

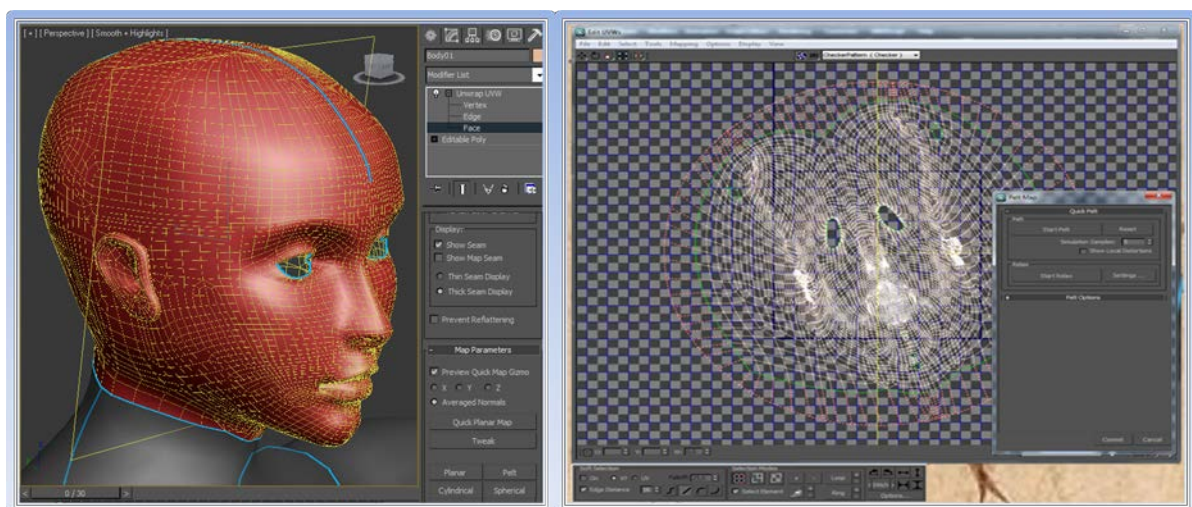
2.4.2 Χρήση της μεθόδου Unwrap UVW

Ξεκινώντας την διαδικασία της δημιουργίας texture map ενός σύνθετου μοντέλου μέσα από τον modifier Unwrap UVM, επιλέγουμε να ορίσουμε εμείς τα σημεία στα όποια θα χωριστούν οι επιφάνειες στο panel. Επιλέγοντας Point To Point Seam διαγράφουμε τις κατάλληλες τομές στο μοντέλο με την χρήση των κορυφών που διαθέτουν τα πολύγωνα.



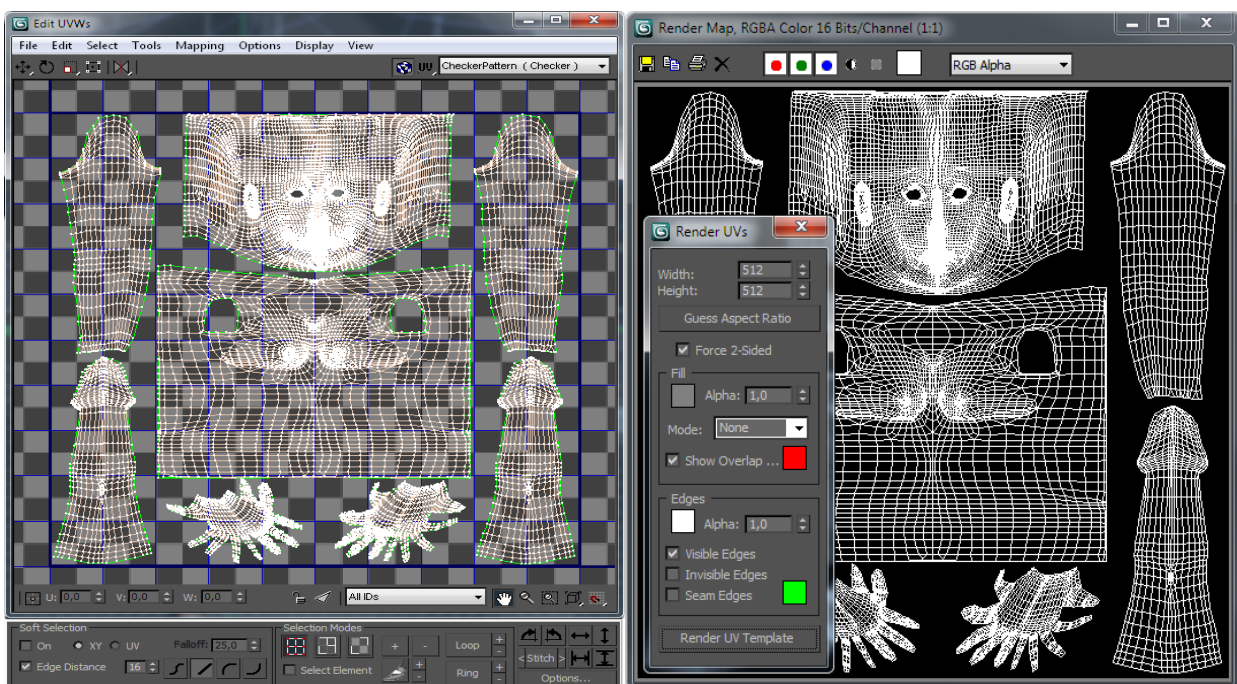
Εικόνα 2.7 Δημιουργία τομών πάνω στο 3d model κατά την διαδικασία Point To Point Seam.

Μόλις προσδιορίσουμε τα σημεία στα οποία θα “κοπούν” τα κομμάτια επιλέγουμε το τμήμα που θέλουμε “ξεδιπλώσουμε” και χρησιμοποιούμε το εργαλείο Pelt και στην συνέχεια πατάμε το κουμπί Start Pelt. Το Pelt θα λέγαμε “τεντωνει” την επιφάνεια του πλέγματος από τις κορυφές που βρίσκονται περιμετρικά. Αφού ολοκληρωθεί αυτή η διαδικασία μπορούμε να χρησιμοποιήσουμε την επιλογή Relax για να δώσουμε στην επιφάνεια μια μορφή που να πλησιάζει αυτήν του τμήματος του μοντέλου αντιπροσωπεύει. Αυτό το κάνουμε περισσότερο για δική μας ευκολία, ώστε να μας βοηθήσει οπτικά κατά την σχεδίαση του texture που θα ακολουθήσει αργότερα.



Εικόνα 2.8 Διαδικασία unwrapping με χρήση του εργαλείου Pelt.

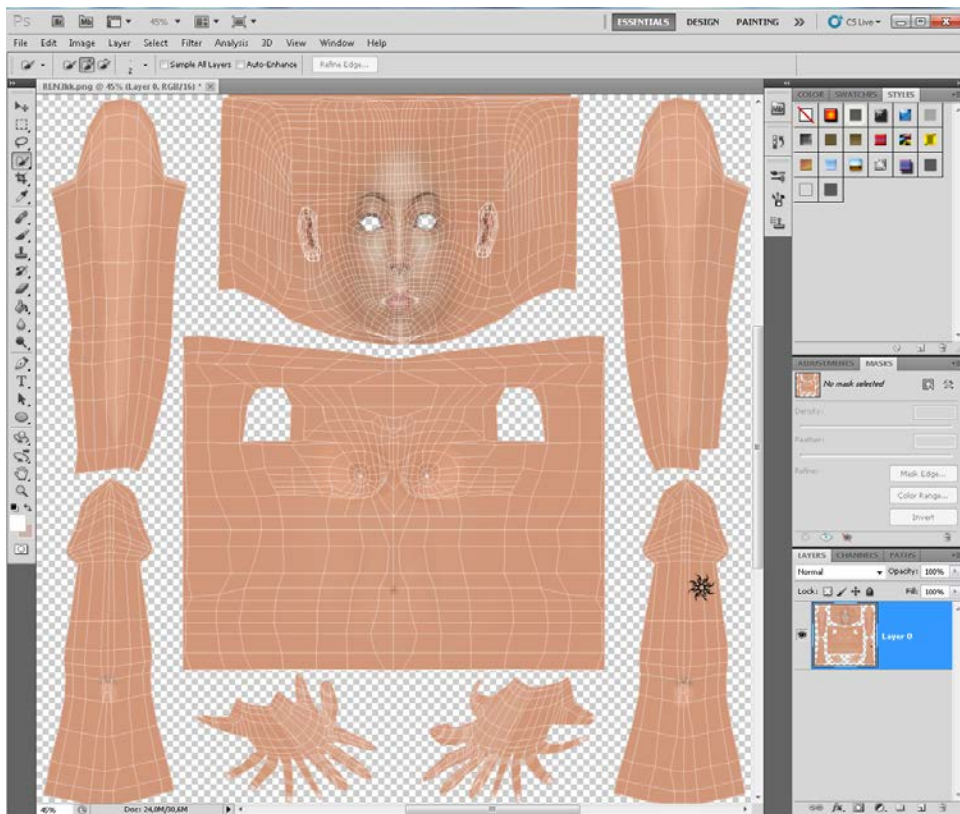
Όταν πλέον έχουμε ολοκληρώσει το unwrapping ακολουθώντας την ίδια διαδικασία για όλα τα τμήματα του μοντέλου, τα τοποθετούμε μέσα στο πλαίσιο του panel σε θέσεις ώστε κανένα από τα κομμάτια να μην καλύπτει κάποιο άλλο. Αυτό μπορούμε να το κάνουμε εμείς ή να το κάνει το πρόγραμμα για εμάς με την επιλογή Pack UVs από το μενού Tools στο Main Toolbar. Μόλις τοποθετηθούν σωστά οι επιφάνειες επιλέγουμε Render UVW Template από το μενού Tools στο Main Toolbar για να εξάγουμε το texture map σαν png εικόνα ή σε ότι άλλο τύπο επιθυμούμε από τις επιλογές που προσφέρονται.



Εικόνα 2.9 Τοποθέτηση πλεγμάτων στο UVW Panel και δημιουργία Texture Map.

Έχοντας τώρα το texture map μπορούμε να το εισάγουμε σε κάποιο πρόγραμμα επεξεργασίας εικόνας όπως είναι το Photoshop και να δημιουργήσουμε την υφή του μοντέλου στα διάφορα μέρη του. Το map σε αυτήν την περίπτωση χρησιμοποιείται σαν οδηγός στον σχεδιασμό και τον χρωματισμό μια εικόνας που θα προσαρμόσουμε στο μοντέλο.

Κατά τον σχεδιασμό της υφής μπορούμε να απεικονίσουμε σκιάσεις και λεπτομέρειες στα διάφορα μέρη, ώστε να δείχνει περισσότερο ρεαλιστικό το αποτέλεσμα.



Εικόνα 2.10 Δημιουργία υφής στο Photoshop με χρήση Texture Map.

Τέλος, αφού ολοκληρώσουμε τον σχεδιασμό της υφής εξάγουμε την εικόνα από πρόγραμμα που χρησιμοποιήσαμε και το προσαρμόζουμε στο μοντέλο.

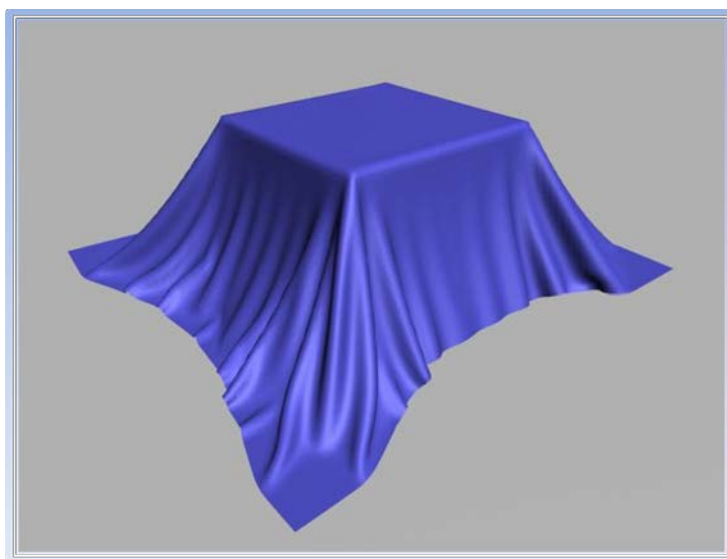


Εικόνα 2.11 Αποτέλεσμα προσαρμογής υφής στο μοντέλο.

2.5 Clothing

Σε πολλές περιπτώσεις κατά την μοντελοποίησή ενός χαρακτήρα θα χρειαστεί να χρησιμοποιήσουμε τεχνικές για να απεικονίσουμε κάποια είδη ένδυσης που θα καλύπτουν κάποια σημεία του σώματος του. Η πρώιμη μέθοδος είναι η επικόλληση εικόνων 2D που απεικονίζουν ρούχα πάνω στο σώμα του μοντέλου όπου κατά την διαδικασία της μοντελοποίησης έχει διαμορφωθεί έτσι ώστε κάποια τμήματα του να έχουν ήδη την μορφή ρουχισμού. Αυτή η μέθοδος είναι απλή, αλλά στερείται διαδραστικότητας και ρεαλιστικότητας. Η δημιουργία ρούχων στον χώρο κατασκευής 3d γραφικών και εφέ έχει αναπτυχθεί αρκετά τα τελευταία χρόνια. Πολλά προγραμματιστικά εργαλεία δημιουργίας τρισδιάστατων γραφικών εμπεριέχουν πλέον τεχνικές για την κατασκευή ρούχων που δίνουν σε ένα μεγάλο βαθμό την ψευδαίσθηση της πραγματικότητας.

Μέσα από εργαλεία που παρέχονται στο 3ds Max μπορούν να δημιουργηθούν αντικείμενα που συμπεριφέρονται σαν επιφάνειες υφασμάτων διαφόρων ειδών και τύπων, δημιουργώντας εφέ επηρεαζόμενα από φυσικές δυνάμεις, όπως η βαρύτητα. Σε αυτό το κομμάτι θα δούμε τον τρόπο με τον οποίο μπορούμε να κατασκευάσουμε κάποιο είδος ρουχισμού μέσα από το 3ds Max χρησιμοποιώντας τον modifier Cloth.



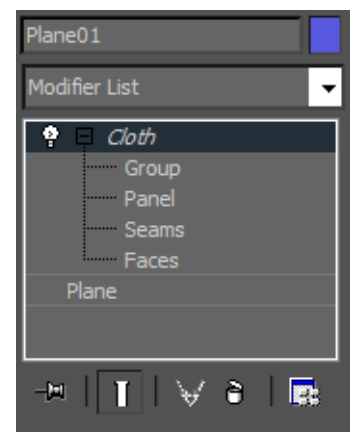
Εικόνα 2.12 Προσομοίωση υφάσματος σε 3d περιβάλλον.

2.5.1 Modifier Cloth

Συνήθως η εφαρμογή του Cloth modifier επιλέγεται σε μια επίπεδη επιφάνεια όπως είναι ένα Plane όπου θα χρησιμοποιηθεί για να απεικονίσει ένα ύφασμα. Στην συνέχεια, με την βοήθεια της βαρύτητας, η επιφάνεια αυτή διαμορφώνεται σε σχέση κάποιο άλλο αντικείμενο της σκηνής το οποίο δημιουργεί collisions σε αυτήν καθώς το καλύπτει. Το ύφασμα θα πάρει την μορφή του αντικειμένου στα σημεία που ήρθε σε επαφή με αυτό, ενώ μπορεί να ενωθεί με άλλα υφάσματα για την δημιουργία ρούχων.

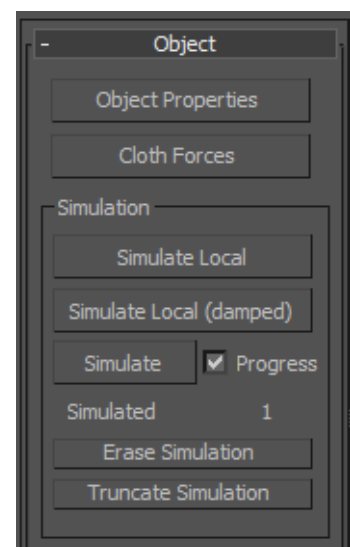
Ας δούμε αναλυτικότερα τα πιο βασικά στοιχεία της εργαλειοθήκης του Cloth modifier και τις επιλογές που μας προσφέρονται.

Όπως στους περισσότερους modifiers έτσι και εδώ έχουμε μια σειρά από επιλογές που αφορούν την επιφάνεια του υφάσματος. Μπορούμε να διαχειριστούμε ξεχωριστά κάποιες κορυφές από την επιφάνεια με το Group ή σαν σύνολο με το Panel. Το Seams χρησιμοποιείται για τις ενώσεις μεταξύ διαφορετικών υφασμάτων αν υπάρχουν και το Face για την διαχείριση των πολυγώνων της επιφάνειας.



Σχήμα 2.17 Εφαρμογή Modifier Cloth

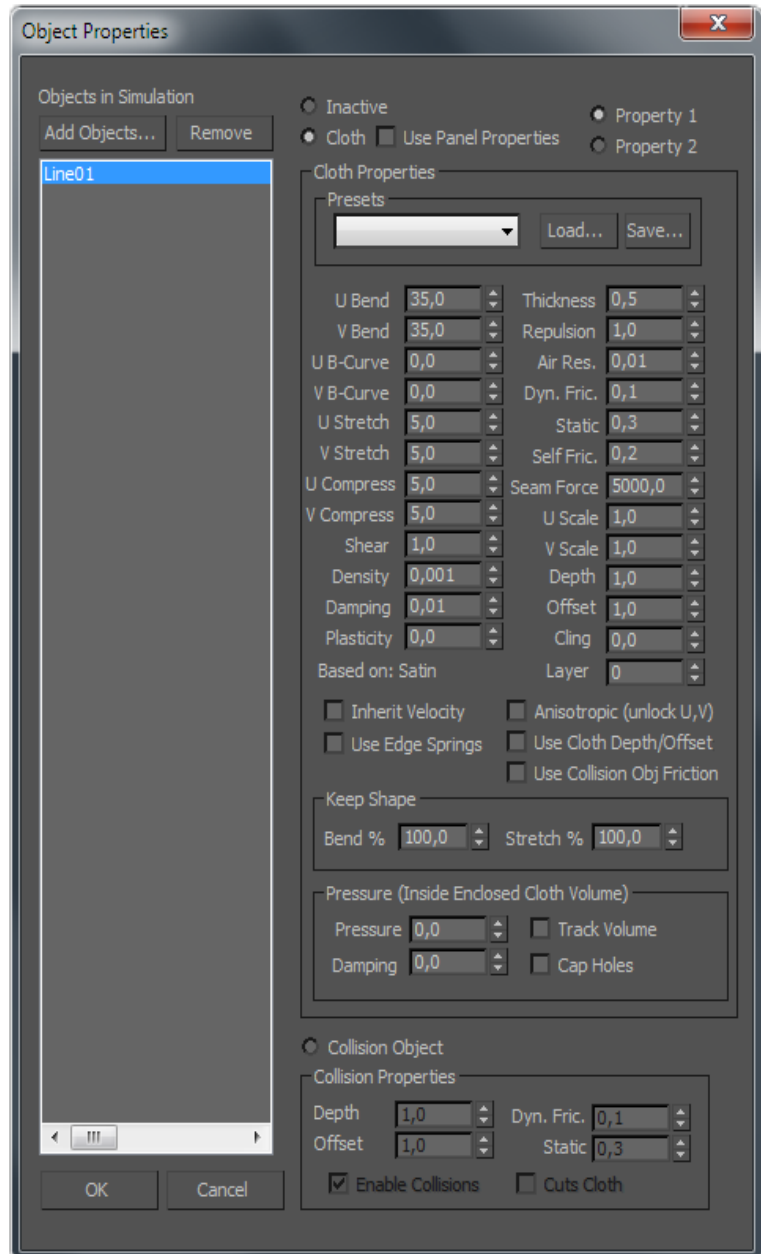
Στις επιλογές της καρτέλας Object διαχειριζόμαστε τους συσχετισμούς των αντικειμένων που εμπλέκονται στην δημιουργία του υφάσματος(Object Properties) και τις δυνάμεις που επιλέγουμε να ασκούνται στα αντικείμενα κατά την διάρκεια της προσομοίωσης(Cloth Forces). Στη περιοχή του Simulation υπάρχουν τα εργαλεία για την διαδικασία της προσομοίωσης. Υπάρχει η δυνατότητα να προσομοιώσουμε μια συγκεκριμένη επιφάνεια σε ένα στατικό αντικείμενο(Simulate Local) ή σε ένα αντικείμενο που εμπεριέχει κάποιο animation σε σχέση με τον χρόνο(Simulate).



Σχήμα 2.18 Simulation Tools Panel

Ακόμα, μπορούμε να επαναφέρουμε την επιφάνεια του υφάσματος στην αρχική του κατάσταση ,αν δεν μας ικανοποιεί το αποτέλεσμα(Erase Simulation).

Στο παράθυρο του Object Properties υπάρχουν παράμετροι που αφορούν την διαμόρφωση υφάσματος και την σχέση μεταξύ των αντικείμενων που εμπλέκονται. Στο πεδίο Objects in Simulation εισάγουμε το μοντέλο που θα μορφοποιήσει την επιφάνεια του υφάσματος. Αυτό θα πρέπει να δηλωθεί σαν Collision Object τσεκάροντας το αντίστοιχο πεδίο στο τέλος του παραθύρου. Ακόμα στο πεδίο υπάρχουν παράμετροι που αφορούν το collision που θα προκύψει. Η επιφάνεια του υφάσματος δηλώνεται σαν cloth στην κορυφή του παραθύρου ενώ αν δεν επιθυμούμε να έχει κάποια ανάμιξη στην προσομοίωση επιλέγουμε Inactive. Τέλος, το πεδίο Cloth Properties περιλαμβάνει έτοιμους τύπους υφασμάτων όπως Silk, Cotton, Leather όπου ανάλογα με την επιλογή αλλάζουν και οι παράμετροι του υφάσματος.

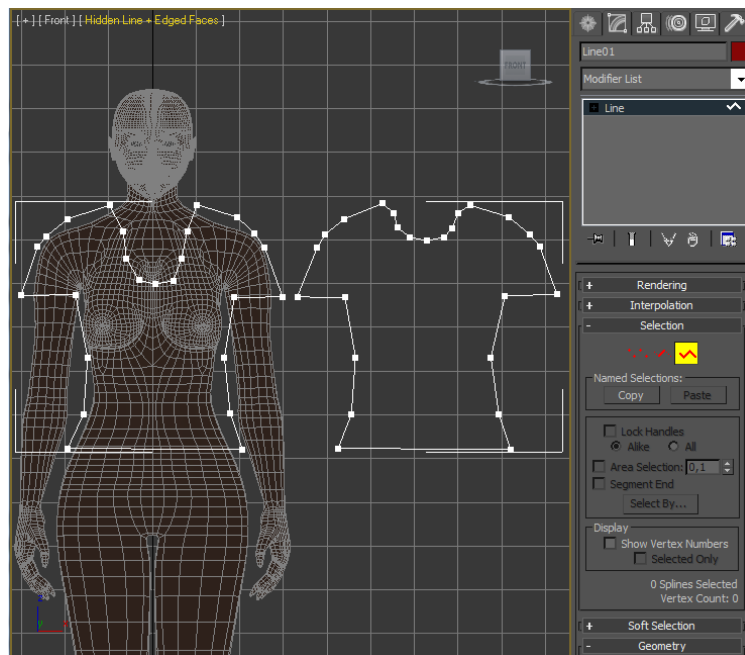


Σχήμα 2.18 Object Properties Window

2.5.2 Ανάπτυξη ρουχισμού με *Cloth modifier*

Η διαδικασία κατασκευής τρισδιάστατων ενδυμάτων που καλύπτουν το σώμα ενός χαρακτήρα είναι αρκετά απλή. Η τεχνική βασίζεται στην χρήση Splines μέσα από την οποία κατασκευάζονται πλαίσια και δίνεται η αρχική μορφή των ρούχων. Στην συνέχεια μετατρέπονται σε επιφάνειες πολυγώνων (panels) και με την χρήση colliders διαμορφώνονται και προσαρμόζονται απευθείας πάνω στο μοντέλο χρησιμοποιώντας τον modifier Cloth.

Πριν την εφαρμογή του Cloth modifier προηγείται η σχεδίαση των μοντέλων με την χρήση απλών Lines. Για παράδειγμα, στην κατασκευή ενός T-Shirt χρειαζόμαστε να σχεδιάσουμε τουλάχιστον δύο σχήματα όπου το ένα θα αφορά την μπροστινή όψη του T-Shirt και το άλλο την πίσω όψη. Μια καλή προσέγγιση είναι η σχεδίαση να γίνεται περιμετρικά του μοντέλου για να μπορούμε αργότερα να έχουμε καλύτερο αποτέλεσμα στην εφαρμογή του ρούχου πάνω του. Ακόμα, πρέπει στο σχέδιο να υπάρχει όσο το δυνατόν περισσότερη συμμετρία ως προς την δεξιά και αριστερή πλευρά του για να προκύψει ομοιομορφία στο τελικό αποτέλεσμα.

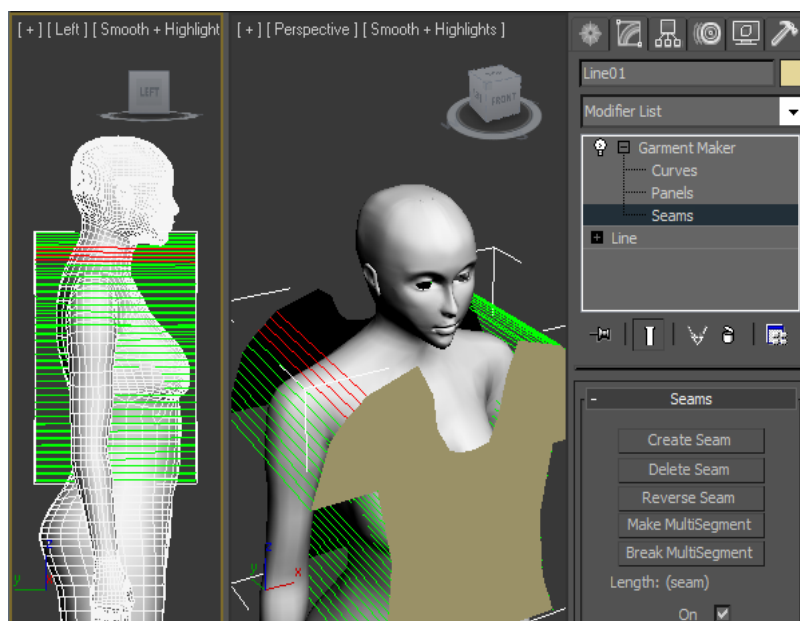


Εικόνα 2.13 Σχεδιασμός ρούχων με χρήση Splines.

Μόλις ολοκληρώσουμε τον σχεδιασμό της μιας όψης μπορούμε να δημιουργήσουμε ένα αντίγραφο του σχήματος και να το τροποποιήσουμε κατάλληλα για την δεύτερη όψη. Αυτό που πρέπει να προσέξουμε είναι να υπάρχει αντιστοιχία των κορυφών ανάμεσα στα δύο κομμάτια.

Αφού έχουμε διαμορφώσει κατάλληλα τα δύο σχήματα τα ομαδοποιούμε με το εργαλείο Attach και επιλέγουμε Garment Maker από την λίστα modifier. Έτσι μετατρέπουμε τους βρόχους σε επιφάνειες που αργότερα θα ενωθούν για να αποτελέσουν το T-Shirt. Τοποθετούμε την μια επιφάνεια μπροστά από τον χαρακτήρα και την άλλη πίσω του, στην αντίστοιχη θέση. Περιστρέφουμε την πίσω όψη 180 μοίρες έτσι ώστε να έχει την σωστή φορά η επιφάνεια και πληκτρολογούμε τον αριθμό των πολυγώνων που θα αποτελείται το κάθε panel από Density.

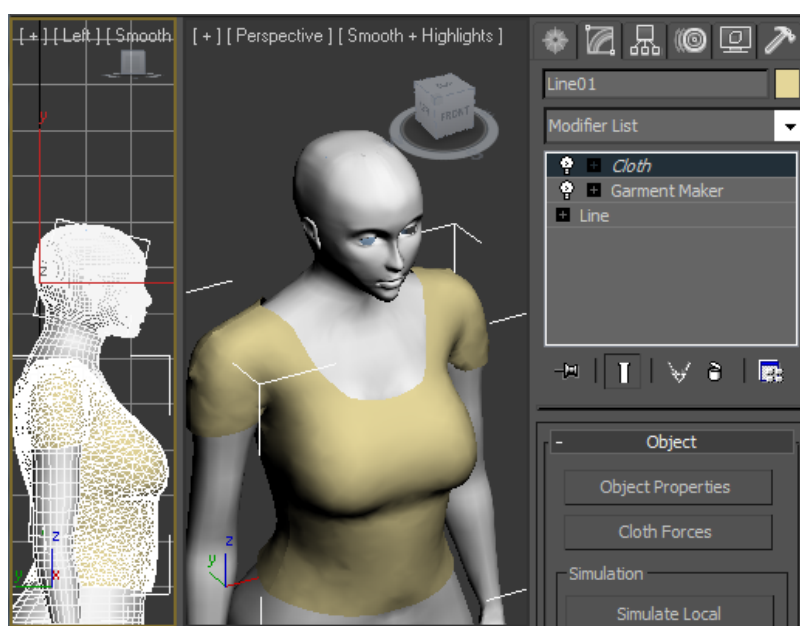
Στην συνέχεια επιλέγουμε Seams από τις επιλογές του modifier Garment Maker και μαρκάρουμε δύο από τις απέναντι ακμές των δύο σχημάτων. Με την επιλογή Create Seams δημιουργούμε συνδέσμους μεταξύ των άκμων και αντίστιχα Delete Seams για να διαγραφουμε . Ακολουθούμε την ίδια διαδικασία για όλα τα αντίστοιχα τμήματα εκτός από τα σημεία που δεν επιθυμούμε να ενωθούν, δηλαδή γύρο από τον λαιμό και τη μέση.



Εικόνα 2.14 Σύνδεση Panel μπροστινής και πίσω όψης με χρήση Seams.

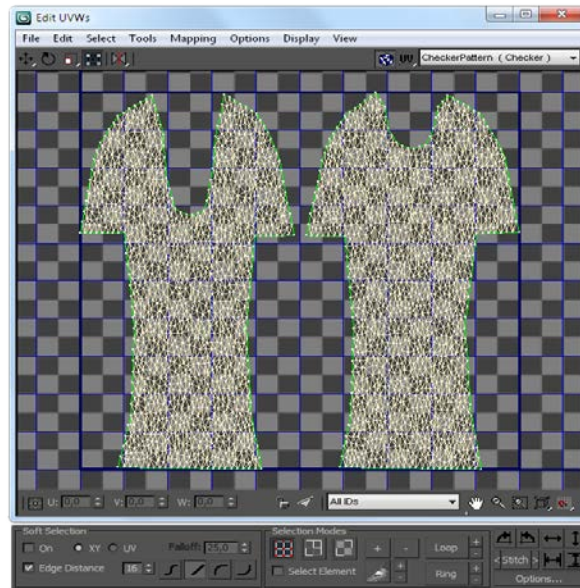
Το αμέσως επόμενο βήμα είναι η χρήση του Cloth modifier. Ανοίγουμε το Object Properties αρχικά και εισάγουμε το μοντέλο του χαρακτήρα στην λίστα που δίνεται από το κουμπί Add Objects. Προσδιορίζουμε την ιδιότητα του, επιλέγοντας Collision Object και τροποποιούμε τις παραμέτρους αν χρειάζεται. Το ίδιο κάνουμε και για την επιφάνεια που θα τροποποιηθεί, επιλέγοντας Cloth. Μπορούμε να επιλέξουμε κάποιον έτοιμο τύπο υφάσματος από το Drop Down List στο Cloth Properties και πατάμε OK. Τώρα επιλέγουμε το εργαλείο Simulate Local και περιμένουμε ,έως ότου διαμορφωθούν οι επιφάνειες γύρο από τον χαρακτήρα δημιουργώντας το T-Shirt. Αν δεν μας ικανοποιεί το αποτέλεσμα επιλέγουμε Erase Simulation και επαναλαμβάνουμε την διαδικασία αλλάζοντας τις παραμέτρους.

Όταν πλέον έχουμε την μορφή που θέλουμε στο T-Shirt αποεπιλέγουμε το Use Sewing Springs και προσομοιώνουμε για μια ακόμα φορά, επιλέγοντας Simulate Local. Με αυτό τον τρόπο ενώνονται τα δύο μέρη από το T-Shirt σε ένα.



Εικόνα 2.15 Διαμόρφωση T-Shirt πάνω στο 3d Model.

Ακόμα μπορούμε να χρωματίσουμε το μπλουζάκι και να του δώσουμε την κατάλληλη υφή με την διαδικασία unrraping όπως είδαμε προηγουμένως.



Εικόνα 2.16 Τοποθέτηση πλεγμάτων του T-Shirt στο UVW Panel.

Τέλος μπορούμε να κατασκευάσουμε επιπλέον ρούχα ακολουθώντας τα βήματα που περιγράψαμε, όπως την κατασκευή ενός Blue Jean παντελονιού για να ντύσουμε και το κάτω μέρος του σώματος του χαρακτήρα.



Εικόνα 2.17 Τελικό αποτέλεσμα δημιουργίας και εφαρμογής ρούχων στο μοντέλο.

2.6 Animation

Σε ένα τρισδιάστατο κόσμο, ένα στατικό μοντέλο μπορεί να “ζωντανέψει” μέσα από την τεχνική του animation. Το animation βασίζεται στην δημιουργία, αποθήκευση και προβολή διαφόρων καταστάσεων του 3d μοντέλου μέσα από πολλά στιγμιότυπα στην διάρκεια του χρόνου. Animation συναντάμε στη δημιουργία τρισδιάστατων ταινιών κινουμένων σχεδίων, στην δημιουργία γραφικών εφέ αλλά και σε 3d εφαρμογές όπως είναι τα ηλεκτρονικά παιχνίδια. Για παράδειγμα, στην απεικόνιση της κίνησης ενός χαρακτήρα (avatar) σε ένα 3d game, χρησιμοποιούνται διάφορα animation clips που προβάλλονται ανάλογα με την πορεία και τον έλεγχο του παιχνιδιού από τον χρήστη.

Το 3ds Max διαθέτει μια σειρά από εργαλεία για την δημιουργία animation που επιτρέπουν στα 3d models να αποκτήσουν κίνηση και διαδραστικότητα. Από το TimeLine μπορούν να οριστούν key frames στην κλίμακα του χρόνου, δηλαδή σημεία αναφοράς για την απεικόνιση της μετάβασης ενός μοντελου από μια κατάσταση σε μια άλλη και συνολικά να αποθηκευτούν σαν ένα animation clip. Ακόμα, μέσα από συστήματα Bones και Biped Skeletons που εισάγονται στα μοντέλα υπάρχει η δυνατότητα διαχείρισης διαφορετικών τμημάτων τους, ώστε να μπορούν να δημιουργηθούν animation από μια σύνθεση διαφορετικών καταστάσεων για κάθε ξεχωριστό τμήμα του μοντέλου.

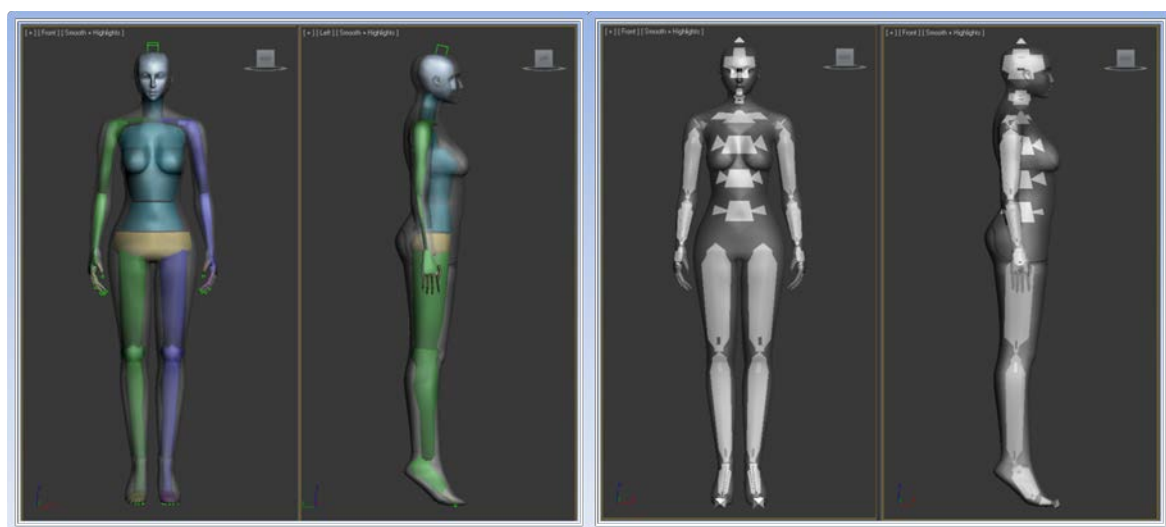
Στις επόμενες σελίδες περιγράφεται ο τρόπος με τον οποίο μπορούν να κατασκευαστούν αυτά τα συστήματα και πως μπορούν να εφαρμοστούν σε ένα μοντέλο χρησιμοποιώντας τον modifier Skin. Επίσης αναφέρεται και η διαδικασία που χρειάζεται ώστε να δημιουργηθεί ένα animation σύνθετης κίνησης σε ένα χαρακτήρα.

2.6.1 Bones και Biped Systems

Τα συστήματα Bones και Biped λειτουργούν σαν ένα σκελετό που προσαρμόζεται στο εσωτερικό του μοντέλου και δίνει την δυνατότητα διαχείρισης διαφορετικών τμημάτων του κατά την διαδικασία του animation. Τα συστήματα αυτά αποτελούνται από σπονδυλωτά στοιχεία (bones) που συσχετίζονται άμεσα μεταξύ τους ακολουθώντας μια ιεραρχική δομή. Κάθε στοιχείο ενός τέτοιου συστήματος είναι υπεύθυνο για την διαχείριση μιας περιοχής πολυγώνων του μοντέλου ενώ επηρεάζει σε έναν βαθμό και γειτονικές περιοχές άλλων στοιχείων, ώστε να διαμορφώνεται η ομαλή μετάβαση μιας κατάστασης, από μια περιοχή σε μια άλλη.

Ένα Biped σύστημα στην ουσία είναι ένας ολοκληρωμένος “δίποδος” σκελετός που παρέχεται έτοιμος από το 3ds Max και χρησιμοποιείται για την προσομοίωση κίνησης σε μοντέλα που διαθέτουν ή προσεγγίζουν ανθρώπινη μορφή.

Το σύστημα Bones κατασκευάζεται χρησιμοποιώντας στοιχεία που θα ονομάζαμε “σπόνδυλους” και αναπτύσσεται σταδιακά στα επιθυμητά σημεία στο εσωτερικό του μοντέλου. Είναι κατάλληλο για μοντέλα τα οποία δεν διαθέτουν ανθρώπινη μορφή ή για περιπτώσεις που η ανάπτυξη ενός σκελετού χρειάζεται περισσότερη ευελιξία ως προς την μορφολογία του.

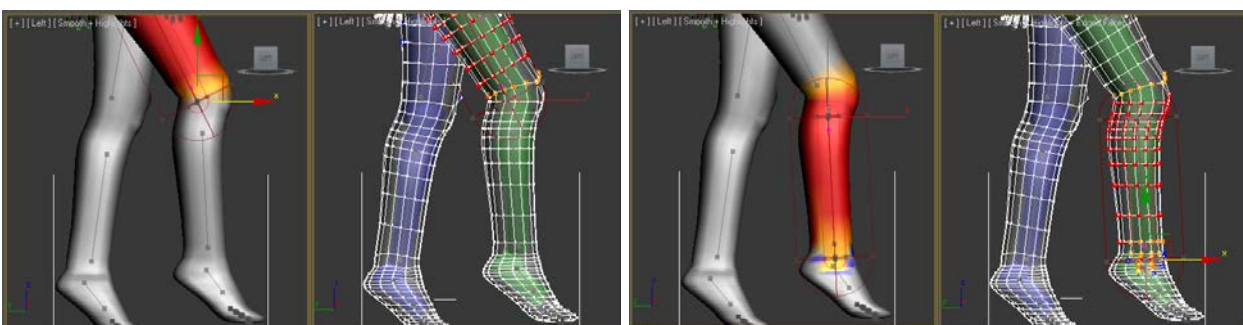


Εικόνα 2.18 Ανάπτυξη Biped και Bone Systems.

2.6.2 Modifier Skin

Ένα μοντέλο για να μπορεί να πραγματοποιήσει μια αλλαγή κατάστασης χρησιμοποιώντας bones η biped systems χρειάζεται να εφαρμοστεί πάνω του ο Skin modifier. Αυτό επιτρέπει στα πολύγωνα (mesh) του μοντέλου να προσκολληθούν πάνω στο σκελετό, ώστε με κάθε κίνηση ενός στοιχείου του σκελετού να ακολουθεί και το συγκεκριμένο τμήμα του μοντέλου στο οποίο έχει οριστεί .

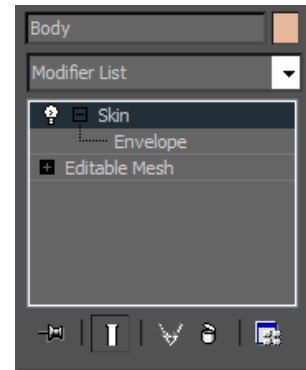
Κάθε στοιχείο του συστήματος (bone) διαθέτει Envelopes (φακέλους) όπου μέσα από αυτούς ορίζονται τα πολύγωνα που θα διαχειρίζεται το κάθε ένα από αυτά. Σε κάθε Envelope υπάρχει μια κλίμακα επιρροής των πολυγώνων, δηλαδή επιλέγεται το κατά πόσο θα επηρεάζεται ένα τμήμα πολυγώνων από κάποιο συγκεκριμένο στοιχείο του σκελετού, καθώς δύο γειτονικά στοιχεία μπορούν να επηρεάζουν ίδια πολύγωνα. Ο βαθμός επιρροής απεικονίζεται με διάφορες χρωματικές διαβαθμίσεις πάνω στην επιφάνεια του μοντέλου . Η θερμότερη χρωματική διαβάθμιση σημαίνει υψηλότερος βαθμός επιρροής, ενώ αντίστοιχα η ψυχρότερη χρωματική διαβάθμιση σημαίνει χαμηλότερος βαθμός επιρροής.



Εικόνα 2.19 Περιοχή και κλίμακα επιρροής των Envelopes με χρωματική απεικόνιση.

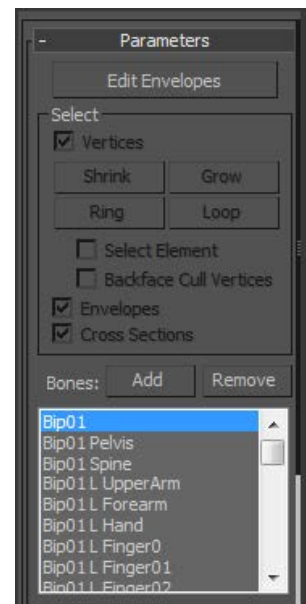
Παρακάτω περιγράφονται τα βασικότερα εργαλεία και επιλογές του modifier Skin ενώ ακολουθεί και ένα παράδειγμα εφαρμογής του στην πράξη.

Στον modifier Skin υπάρχει η επιλογή των φακέλων (Envelope) όπου μέσα από αυτούς οριοθετούμε την περιοχή επηρεασμού που θα έχει το κάθε στοιχείο του σκελετού.



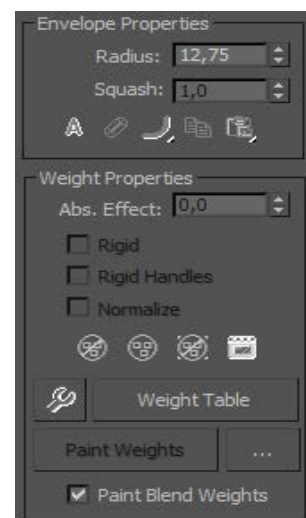
Σχήμα 2.19 Εφαρμογή Modifier Skin

Στην περιοχή Parameters δίνεται η δυνατότητα διαχείρισης και τροποποίησης των φακέλων μέσα από την επιλογή Edit Envelopes. Από το πεδίο Select μπορεί να γίνει η διαχείριση των Envelopes σε σχέση με τις κορυφές των πολυγώνων ενός μοντέλου. Στην λίστα των Bones μπορούν να εισαχθούν τα επιθυμητά στοιχεία του σκελετού που θα εμπλέκονται στην δημιουργία του animation με την επιλογή Add, ενώ αντίστοιχα μπορούν να αφαιρεθούν αυτά που δεν χρειάζονται.



Σχήμα 2.20 Skin Modifier Parameters

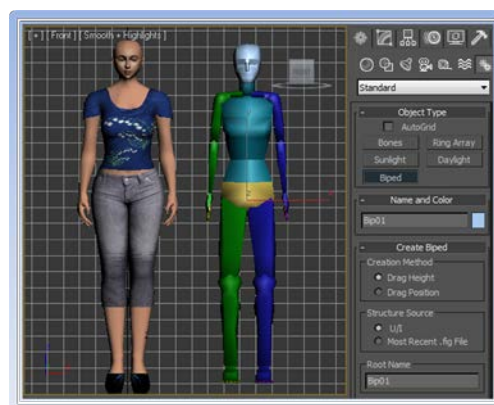
Στην ίδια περιοχή των Parameters υπάρχουν ακόμα τα πεδία Envelope Properties και Weight Properties. Από το πρώτο πεδίο ορίζεται το μέγεθος των φακέλων, δηλαδή η ακτίνα επιρροής στην κάθε περιοχή του μοντέλου, ενώ στο δεύτερο πεδίο ορίζεται η κλίμακα επιρροής του φακέλου σε επιλεγμένα τμήματα ή κορυφές. Υπάρχουν επίσης εργαλεία που βοηθούν στην συνολική και λεπτομερέστερη διαχείριση των φακέλων, όπως είναι το Weight Table, που μέσα από ένα πίνακα που αναφέρει όλα τα στοιχεία του σκελετού, καθορίζονται τα πολύγωνα που επηρεάζονται και ο βαθμός επιρροής τους.



Σχήμα 2.21 Envelope και Weight Properties

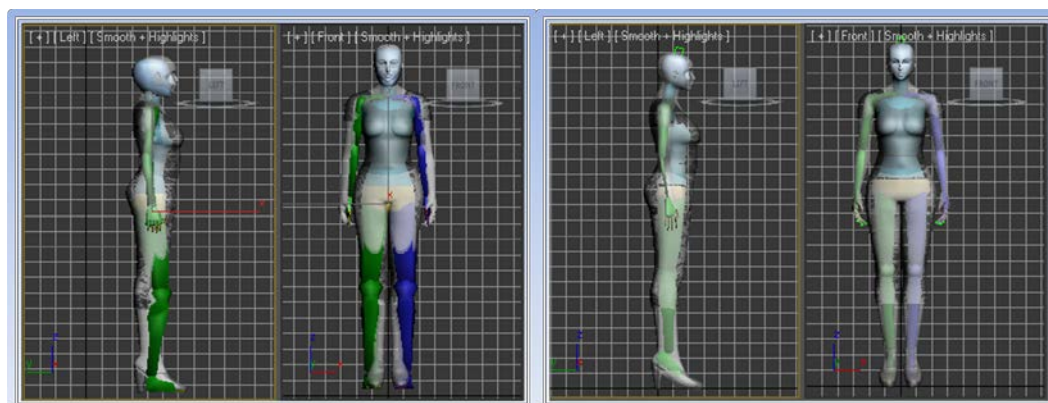
2.6.3 Προσαρμογή Biped system σε avatar με χρήση του Skin modifier .

Για να μπορέσουμε προσαρμόσουμε ένα σύστημα Biped στο μοντέλο αρχικά το εισάγουμε στην σκηνή από την καρτέλα Create→Systems και στην συνέχεια επιλέγουμε Biped. Τραβώντας το mouse στην σκηνή δημιουργούμε το σύστημα στο μέγεθος του σώματος του μοντέλου.



Εικόνα 2.20 Δημιουργία Biped System.

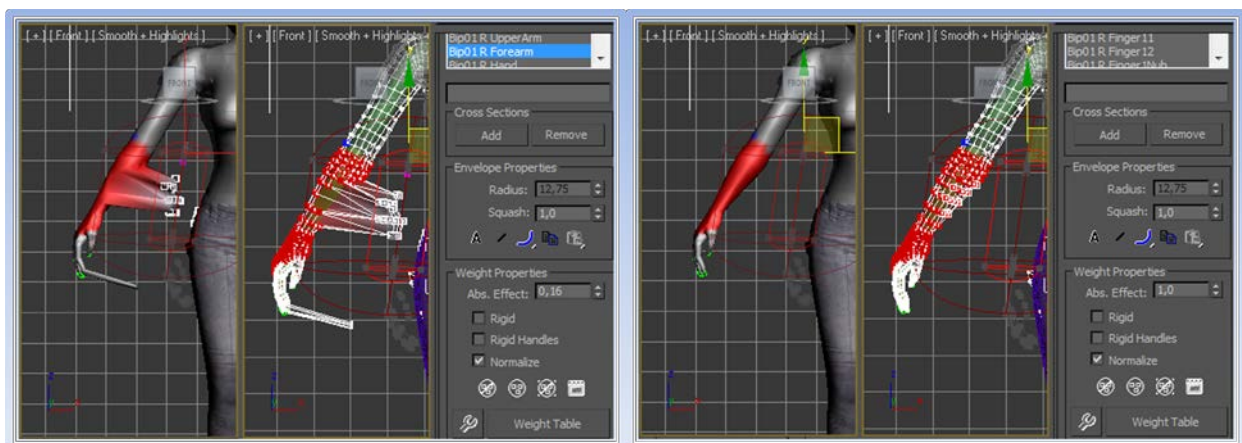
Στην συνέχεια τοποθετούμε το σκελετό στο εσωτερικό του σώματος και μέσα από το Object Properties του μοντέλου επιλέγουμε See Through και Freeze. Αυτό το κάνουμε για διευκόλυνση στην τοποθέτηση και τροποποίηση των στοιχείων του συστήματος, ώστε να μπορούμε να διακρίνουμε το σύστημα εσωτερικά του σώματος και παράλληλα να μην επιτρέπεται η τροποποίηση του μοντέλου που θα μπορούσε να προκύψει λανθασμένα κατά την διαδικασία. Ακόμα, θα μπορούσαμε να αποκρύψουμε ρούχα ή άλλα αξεσουάρ που έχουμε τοποθετήσει στο μοντέλο με την επιλογή Hide Selected, αφού δεν μας χρειάζονται στην συγκεκριμένη περίπτωση. Κάθε bone θα πρέπει να τοποθετείτε απόλυτα μέσα στο σώμα καταλαμβάνοντας όσο περισσότερο χώρο της περιοχής γίνεται. Για να μπορούμε να τροποποιήσουμε το σκελετό πρέπει πρώτα να ενεργοποιήσουμε το Figure Mode που βρίσκεται στην καρτέλα Motion. Έτσι, μέσα από εργαλεία όπως Move, Scale και Rotation φέρνουμε τα τμήματα του σκελετού ακριβώς στο μέγεθος και τη θέση που χρειάζεται.



Εικόνα 2.21 Τροποποίηση και τοποθέτηση Biped συστήματος στο 3d model.

Μετά την ολοκλήρωση της τοποθέτησης του συστήματος στο μοντέλο, από-επιλέγουμε το Freeze από το Object Properties ή μέσω του mouse right click Unfreeze All. Τώρα εφαρμόζουμε στο σώμα τον modifier Skin και εισάγουμε όλα τα στοιχεία του σκελετού στην λίστα Bones από την περιοχή Parameters του modifier. Πλέον το σώμα του μοντέλου ακολουθεί τις κινήσεις του σκελετού μέσω των Envelopes και αυτό μπορούμε να το διαπιστώσουμε μετακινώντας κάποια στοιχεία του συστήματος. Οι κινήσεις που μπορούν να γίνουν μέσα από ένα Biped σύστημα προσομοιώνουν τις δυνατότητες ενός ανθρώπινου σκελετού. Για παράδειγμα, με την κίνηση προς μια κατεύθυνση του bone που βρίσκεται στην παλάμη (Bip01 Hand) του μοντέλου, αντίστοιχα ακολουθεί ολόκληρος ο βραχίονας του χεριού. Αυτές οι δυνατότητες κίνησης εξαρτώνται από τις διασυνδέσεις που έχουν μεταξύ τους τα bones και την ιεραρχική δομή που ακολουθούν. Με την επιλογή Edit Envelopes εμφανίζονται όλοι οι φάκελοι του συστήματος και με χρωματικές απεικονίσεις οι περιοχές που επηρεάζονται.

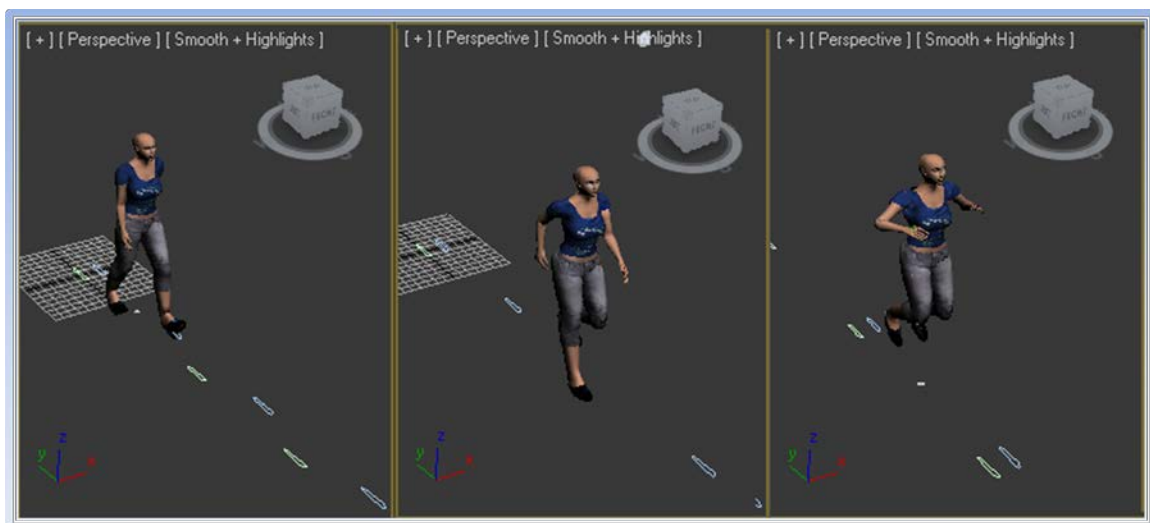
Ωστόσο, συχνά θα χρειαστεί να τροποποιήσουμε κάποιες παραμέτρους των φακέλων για έχουμε μια ομαλή και ρεαλιστική κίνηση. Για να μπορούμε να διακρίνουμε τα σημεία που χρειάζεται να επέμβουμε προκειμένου να διορθώσουμε τις περιοχές που θα επηρεάζει κάθε Envelope, πρέπει να δημιουργήσουμε κάποιες δοκιμαστικές κινήσεις στο μοντέλο έτσι ώστε να παρατηρήσουμε την κατάσταση των πολυγώνων του (mesh). Σε σημεία που παρατηρούμε να χαλάει η συνοχή των πολυγώνων του μοντέλου, μπορούμε να μετακινήσουμε ένα φάκελο ή να αυξομειώσουμε την ακτίνα επιρροής του με τα εργαλεία Move και Scale ή τα εργαλεία στο πεδίο Envelope Properties. Επιπροσθέτως, μπορούμε να καθορίσουμε την επιρροή ενός φακέλου σε μεμονωμένες κορυφές των πολυγώνων από το Weight Properties.



Εικόνα 2.22 Προσκόλληση πολυγώνων στην περιοχή επιρροής του Envelope με την χρήση του Abs. Effect.

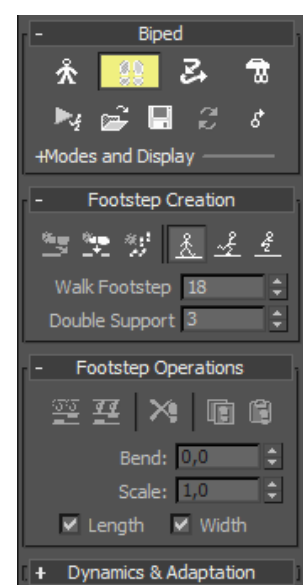
2.6.4 Δημιουργία Animation

Για την ανάπτυξη ενός animation μέσω Biped συστήματος που να προσομοιώνει την κίνηση ενός χαρακτήρα μέσα από το 3ds Max, μπορεί να χρησιμοποιηθεί ένα εργαλείο που ονομάζεται Footstep Mode και βρίσκεται στην καρτέλα Motion. Μεσα από αυτό δίνεται η δυνατότητα δημιουργίας έτοιμων animation κίνησης που μπορούν να προσαρμοστούν σε ένα avatar. Μπορούν να προσομοιωθούν τρεις διαφορετικές καταστάσεις κίνησης: Walk, Run και Jump.



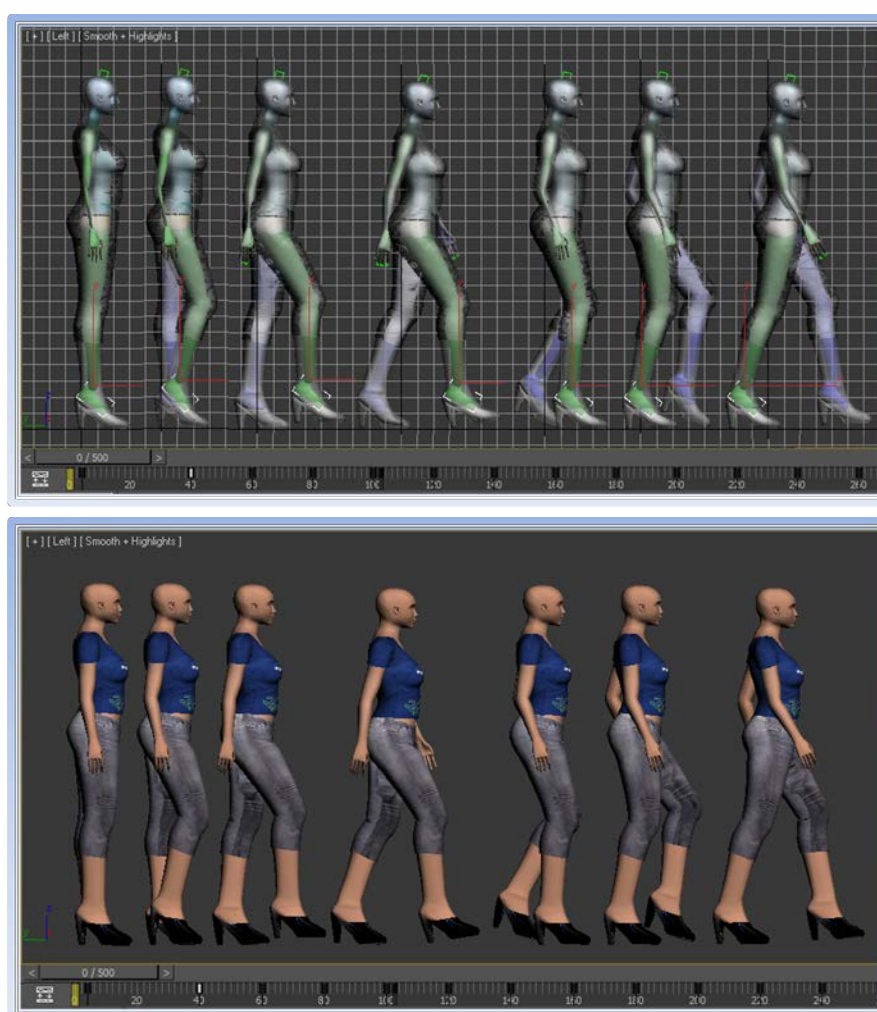
Εικόνα 2.23 Χρήση του εργαλείου Footstep mode για προσομοίωση κίνησης τριών ειδών : Walk, Run, Jump.

Για να δημιουργήσουμε κίνηση από το Footstep Mode επιλέγουμε το είδος της κίνησης(walk,run,jump) και πατάμε Create Multiple Footsteps από το πεδίο Footstep Creation. Επιλέγουμε τον αριθμό των βημάτων που θέλουμε να αποτελείται η κίνηση και στην συνέχεια δημιουργούμε τα κατάλληλα key frames από το Create Keys for Inactive Footsteps που βρίσκεται στο πεδίο Footstep Operations. Επιλέγοντας Play από το TimeLine βλέπουμε να πραγματοποιείται το animation της κίνησης. Ακόμα, μέσα από τις παραμέτρους του Footstep Mode μας δίνεται η δυνατότητα να αλλάξουμε αρκετά στοιχεία της κίνησης, ανάλογα με τις ανάγκες μας, χρησιμοποιώντας τις κατάλληλες ρυθμίσεις.



Σχήμα 2.22 Εργαλεία του Footstep Mode.

Αν θέλουμε να δημιουργήσουμε το δικό μας animation μπορούμε να αλλάζουμε σταδιακά την κατάσταση στην οποία βρίσκεται το μοντέλο εισάγοντας τα αντίστοιχα key frames στην περιοχή του TimeLine. Για να πραγματοποιηθεί μια ολοκληρωμένη κίνηση βηματισμού χρησιμοποιούμε τα στοιχεία του σκελετού που βρίσκονται στα άκρα του χαρακτήρα. Με τα εργαλεία Move και Rotation μεταβάλλουμε την κατάσταση των ποδιών του μοντέλου σε μια διαδικασία βηματισμού ενώ τροποποιούμε κατάλληλα τα χέρια και το σώμα ώστε να διαγράψουν μια αρμονική κίνηση. Προσπαθούμε όσο το δυνατόν περισσότερο να δώσουμε μια φυσικότητα στο animation. Για κάθε μεταβολή του χαρακτήρα επιλέγουμε Set Key από την λίστα Key Info στην καρτέλα Motion και μετατοπίζουμε το timeline Bar στο επόμενο επιθυμητό frame. Μαρκάροντας ένα key frame μπορούμε να το διαγράψουμε ή να το μετατοπίσουμε στον χρόνο. Ακολουθούμε την ίδια διαδικασία έως ότου ολοκληρωθεί η κίνηση και έχουμε ένα ικανοποιητικό αποτέλεσμα.



Εικόνα 2.24 Ακολουθία κινήσεων για την δημιουργία βηματισμού.

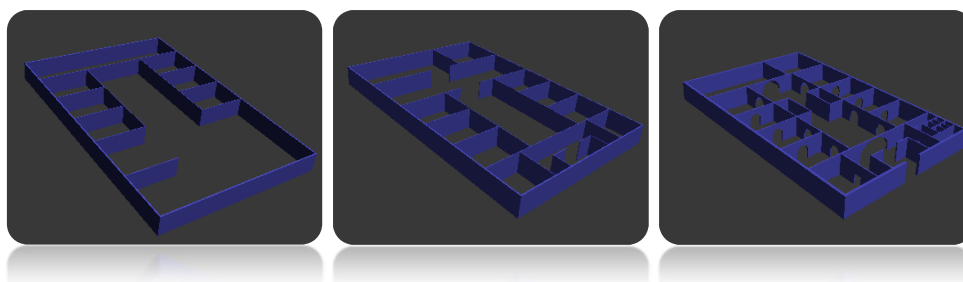
2.7 Πορεία μοντελοποίησης των στοιχείων της εφαρμογής μέσω του 3DS Max.

Αφού είδαμε τα κυριότερα εργαλεία του 3ds Max και εξετάσαμε βασικές τεχνικές και μεθόδους κατά την διαδικασία της μοντελοποίησης, παρουσιάζουμε περιληπτικά και μέσα από εικόνες την πορεία κατασκευής των μοντέλων που χρησιμοποιήθηκαν στην εφαρμογή προσομοίωσης εικονικού μουσείου που αναπτύχθηκε κατά την διατριβή της παρούσας πτυχιακής εργασίας. Παρακάτω παρουσιάζεται η πορεία κατασκευής του μουσείου και των στοιχείων που χρησιμοποιήθηκαν στην ανάπτυξη του, καθώς και κάποια από τα εκθέματα που περιλαμβάνει.

2.7.1 Museum Building

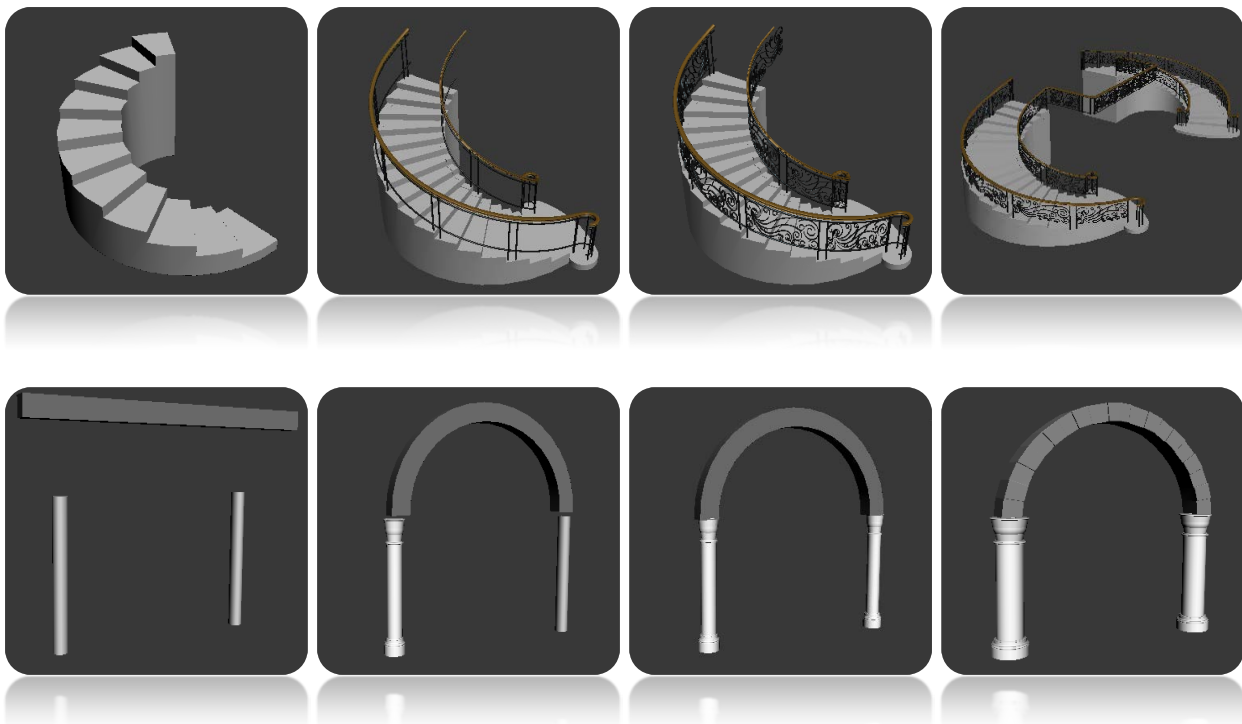
2.7.1.1 Modeling

Το πρώτο βήμα είναι η ανάπτυξη του κτιρίου που θα στεγάσει όλα τα μοντελοποιημένα εκθέματα. Στην διαδικασία αυτή χρησιμοποιήθηκαν τεχνικές polygon και box modeling. Αρχικά σχεδιάζεται το εξωτερικό μέρος του μουσείου και στην συνέχεια κατασκευάζονται οι εσωτερικές αίθουσες και τα δωμάτια με συμμετρική ανάπτυξη.



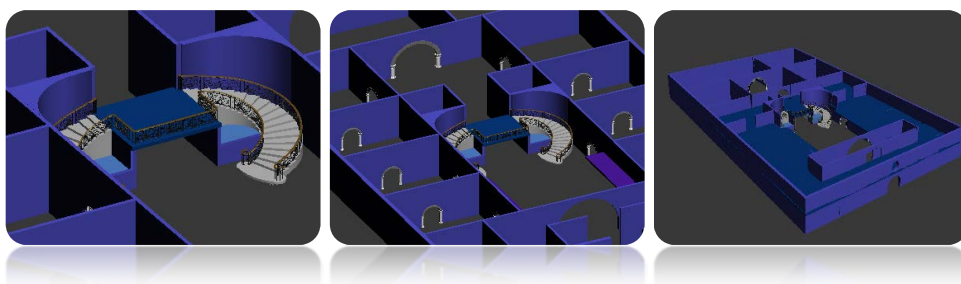
Εικόνα 2.25 Πορεία ανάπτυξης μουσείου 1

Έπειτα από τον βασικό σχεδιασμό αναπτύσσονται τα διαζώματα των εσωτερικών εισόδων των δωματίων και η κεντρική σκάλα που οδηγεί στον δεύτερο όροφο. Παρακάτω παρουσιάζονται εικόνες με την διαδικασία ανάπτυξης των μοντέλων.



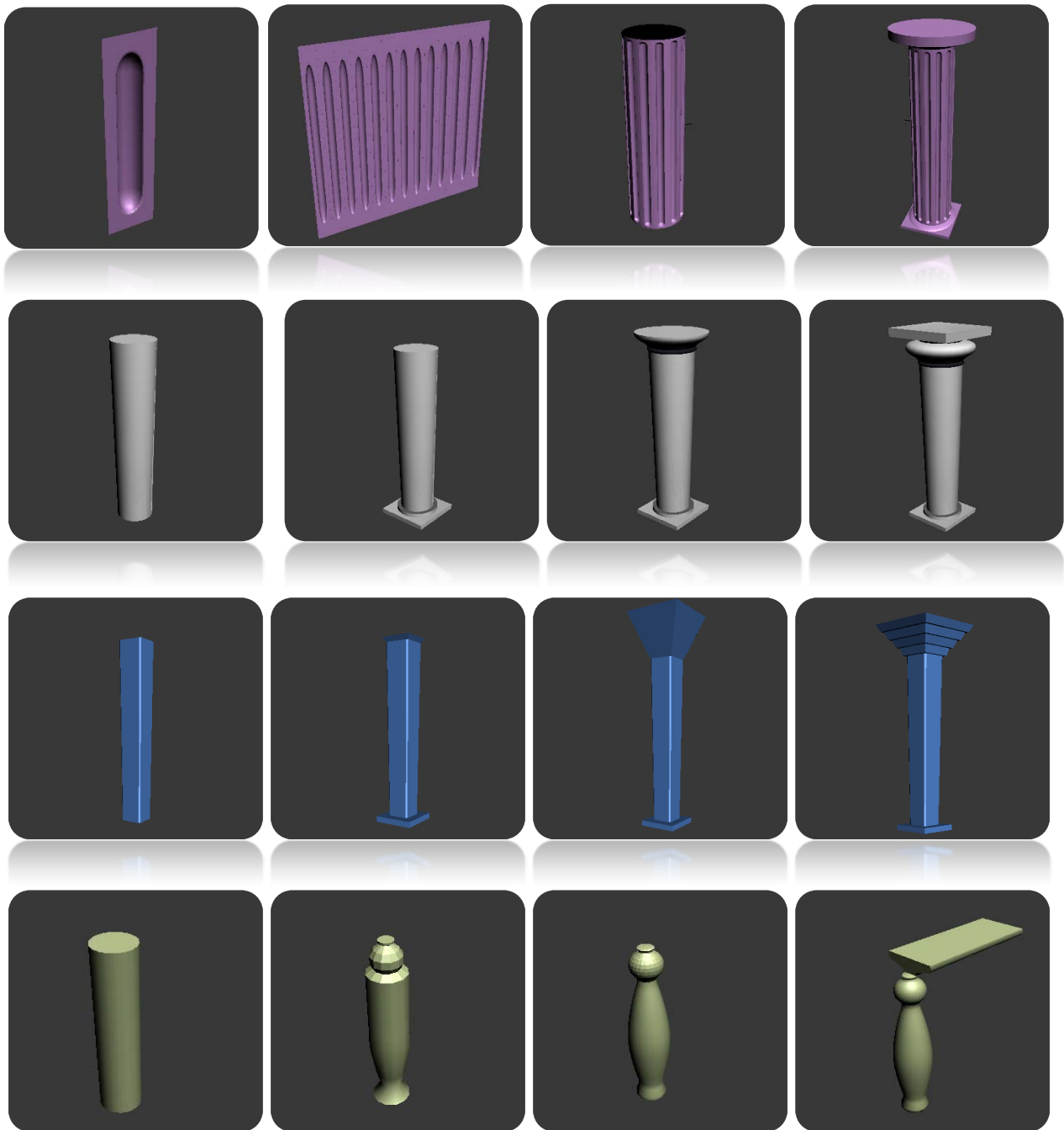
Εικόνα 2.26 Πορεία ανάπτυξης σκάλας και διαζωμάτων

Στην συνέχεια τοποθετούνται τα αντικείμενα κατάλληλα στο εσωτερικό του μουσείου και δημιουργείται το δεύτερο πάτωμα με την χρήση ενός box.



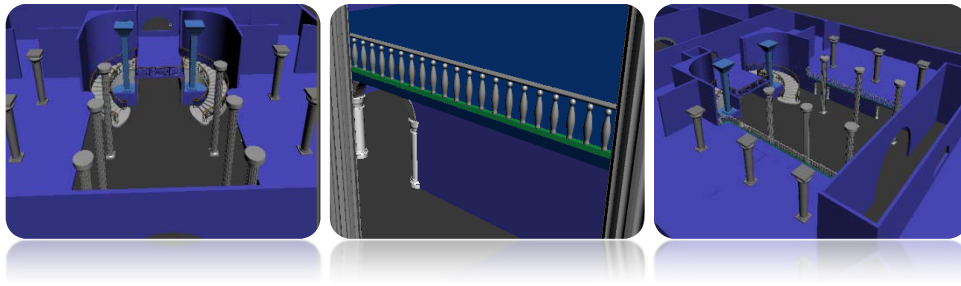
Εικόνα 2.27 Πορεία ανάπτυξης μουσείου 2

Αφού κατασκευάστηκε και το δεύτερο πάτωμα, μοντελοποιούνται οι κολώνες και τα κάγκελα που θα τοποθετηθούν στην δεξιά και αριστερή πλευρά της κεντρικής αίθουσας και του πρώτου ορόφου. Παρακάτω παρουσιάζεται μια ακολουθία από γραφικά στιγμιότυπα της εξέλιξης αυτών των αντικειμένων.



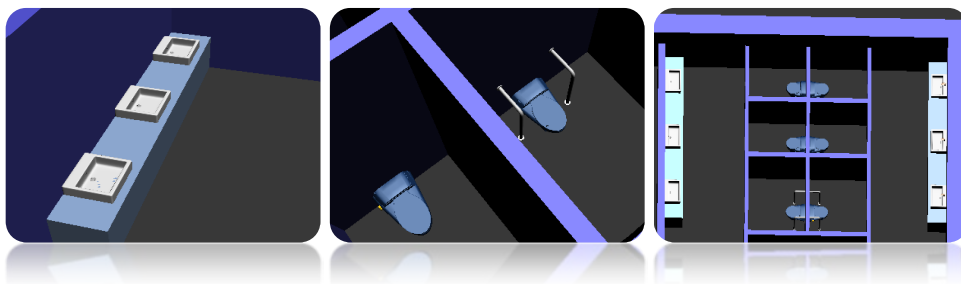
Εικόνα 2.28 Πορεία ανάπτυξης κολώνων μουσείου

Μόλις ολοκληρωθούν τα μοντέλα εισάγονται στην σκηνή και δημιουργείται ο επιθυμητός αριθμός από αντίγραφα, ώστε να τοποθετηθούν όμοια μοντέλα σε διαφορετικά σημεία της σκηνής όπου χρειάζεται.



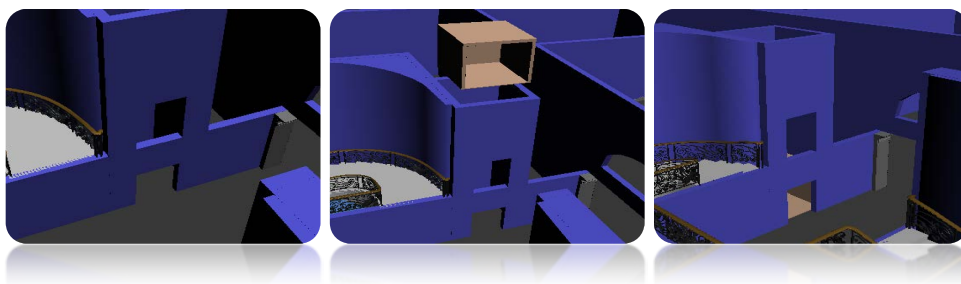
Εικόνα 2.29 Πορεία ανάπτυξης μουσείου 3

Ομοίως αναπτύσσονται και τοποθετούνται τα είδη υγιεινής στην περιοχή του WC.



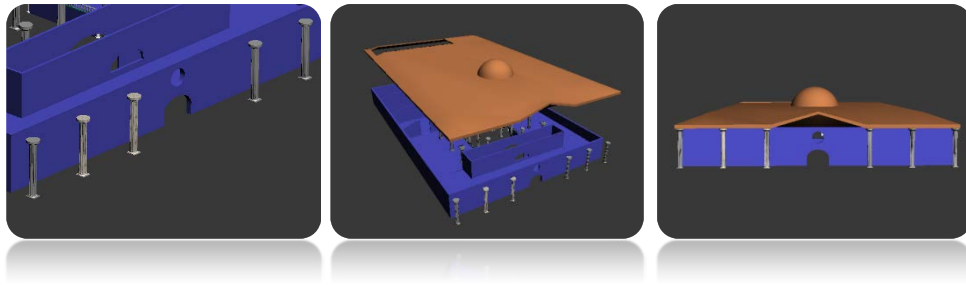
Εικόνα 2.30 Πορεία ανάπτυξης μουσείου 4

Στον κεντρικό διάδρομο αναπτύσσεται ένας χώρος όπου θα τοποθετηθεί ασανσέρ για την μετάβαση στον δεύτερο όροφο. Αφού διαμορφωθεί κατάλληλα ο χώρος προστίθεται ο θάλαμος ο οποίος είναι κατασκευασμένος από boxes.



Εικόνα 2.31 Πορεία ανάπτυξης μουσείου 5

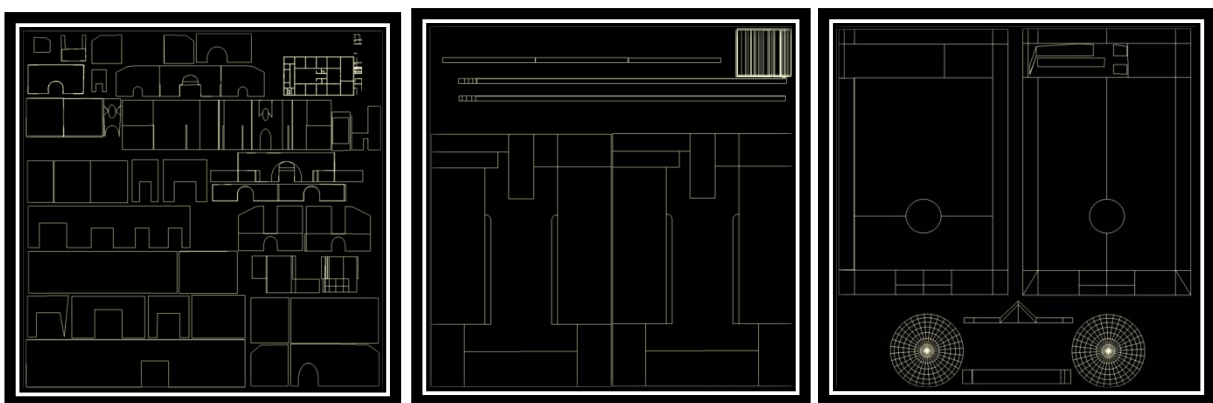
Τέλος, κατασκευάζεται η οροφή του κτιρίου και διαμορφώνεται κατάλληλα ώστε να δημιουργεί έναν θολό στο κέντρο της κύριας αίθουσας. Επίσης προστίθενται έξι ακόμα κίονες στην πρόσοψη του μουσείου.



Εικόνα 2.32 Πορεία ανάπτυξης μουσείου 6

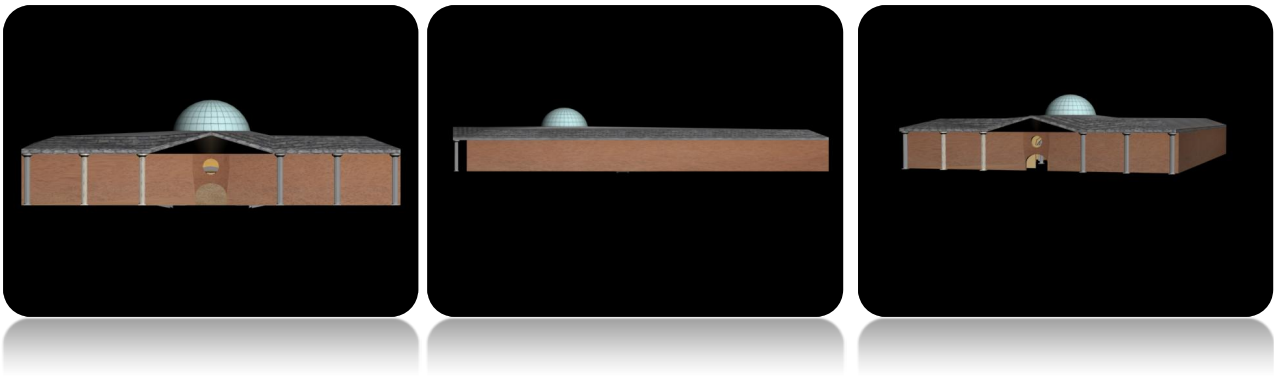
2.7.1.2 Textures

Για την δημιουργία texturemap των τμημάτων του μουσείου ακολουθείται η τεχνική unwrapping με χρήση έτοιμων box gizmos, αφού σχεδόν όλες οι επιφάνειες του κτιρίου είναι επίπεδες και αποτελούνται από παραλληλόγραμμα.



Εικόνα 2.33 Χάρτες για την προσαρμογή υφών στο μουσείο(τοίχοι, 2^ο πάτωμα και οροφή)

Σε όλες τις υφές που δημιουργήθηκαν στην εφαρμογή χρησιμοποιήθηκε το πρόγραμμα επεξεργασίας εικόνας Photoshop.



Εικόνα 2.34 Όψεις μουσείου με προσαρμοσμένες υφές πάνω του

2.7.2 Exhibits(Εκθέματα)

2.7.2.1 Modeling

Για τα εκθέματα χρησιμοποιήθηκαν διάφορες τεχνικές μοντελοποίησης (box, polygon, spline modeling). Για περισσότερη απόδοση ρεαλιστικότητας και λεπτομέρειας των μοντέλων σε σχέση με τα πραγματικά εκθέματα, χρησιμοποιήθηκαν εικόνες σαν οδηγοί κατά τη μοντελοποίηση. Για το λόγο ότι είναι δύσκολο να περιγραφεί η διαδικασία μοντελοποίησης όλων των εκθεμάτων που στεγάζει το μουσείο, παρακάτω παρουσιάζεται ενδεικτικά η πορεία εξέλιξης κάποιων εξ' αυτών.

Στην μοντελοποίηση αγγείων χρησιμοποιήθηκε ο κύλινδρος σαν βασικό συστατικό. Με την αυξητική ανάπτυξη των μοντέλων από την βάση προς την κορυφή, υπήρχε η δυνατότητα σταδιακής διαμόρφωσης σε κάθε νέο επίπεδο.

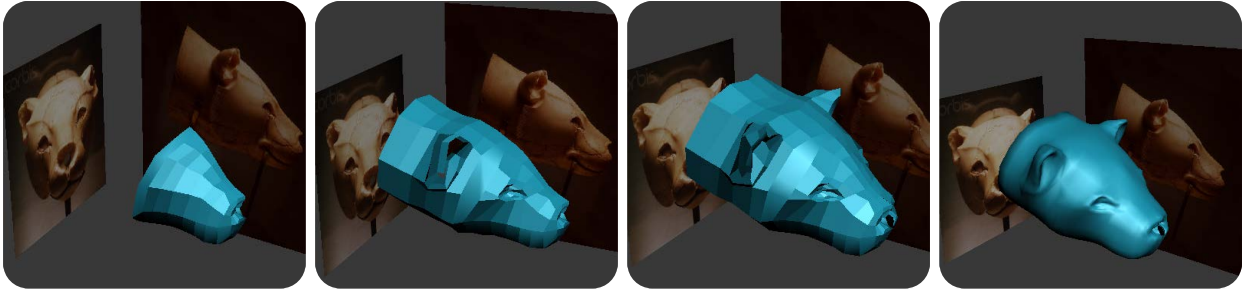
Kamares Jug



Εικόνα 2.35 Πορεία ανάπτυξης μουσείου 5

Στην ανάπτυξη κάποιων εκθεμάτων με την χρήση box σαν αρχικό στοιχείο, ακολουθήθηκε η ίδια διαδικασία σταδιακής μοντελοποίησης σε επίπεδα.

Lion Rhyton



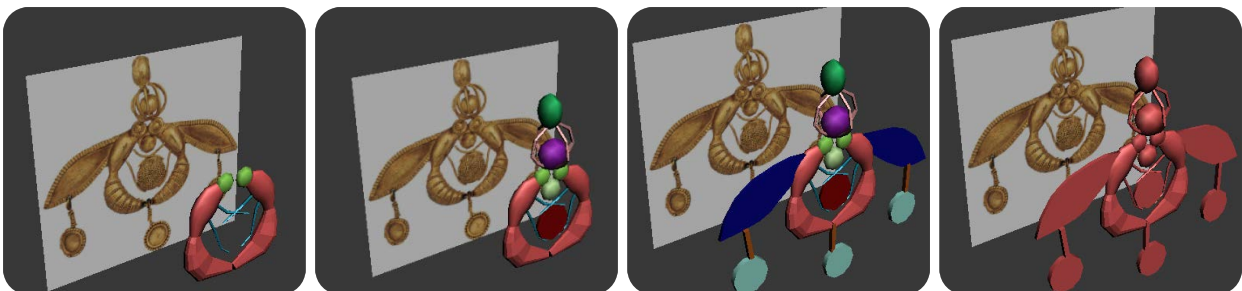
Bull Rhyton



Εικόνα 2.36 Πορεία ανάπτυξης εκθεμάτων: *Lion Rhyton* και *Bull Rhyton*

Σε άλλες περιπτώσεις χρησιμοποιήθηκαν πολλά διαφορετικά σχήματα στην σύνθεση κάποιου εκθέματος, όπως boxes, spheres και cylinders. Στην συνέχεια ενώθηκαν μεταξύ τους για να αποτελέσουν ένα ενιαίο σύνολο.

Bees Pendant



Εικόνα 2.37 Πορεία ανάπτυξης εκθεμάτων: *Bees Pendant*

Τα εκθέματα τα οποία απεικονίζουν ανθρώπινες φιγούρες αναπτύχθηκαν τμηματικά, με την διαμόρφωση του κορμού αρχικά και την διαμόρφωση του κεφαλιού και τον άκρων στη συνέχεια.

Snake Goddess



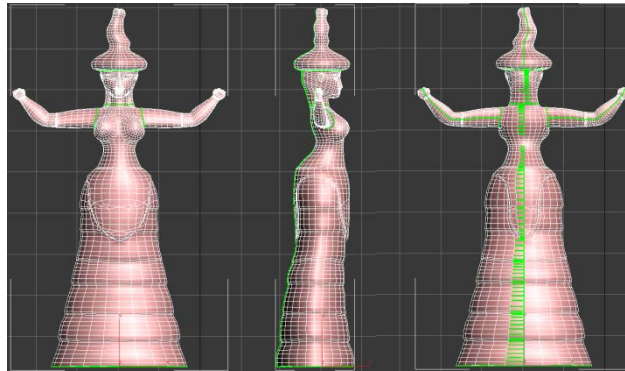
Goddess Terracotta



Εικόνα 2.38 Πορεία ανάπτυξης εκθεμάτων: Snake Goddess και Goddess Terracotta

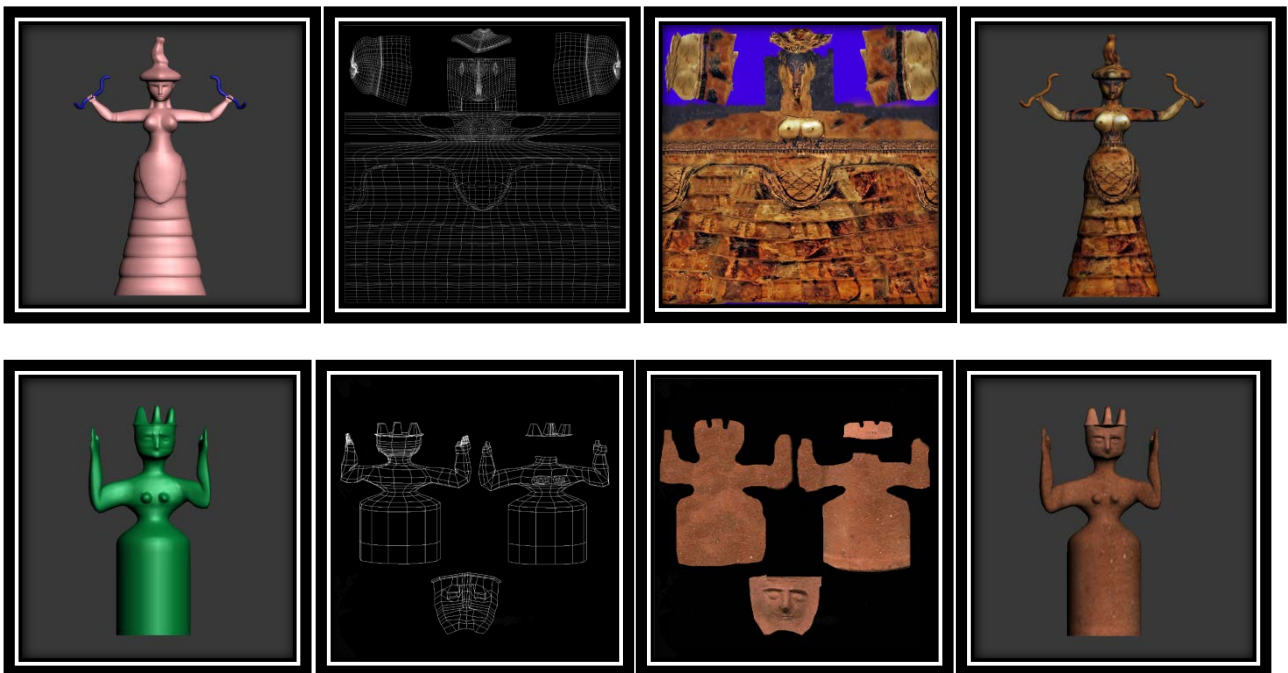
2.7.2.2 Textures

Σε όλα τα εκθέματα χρησιμοποιήθηκε η μέθοδος unwrapping για την δημιουργία και προσαρμογή υφών. Η κατασκευή texture maps πραγματοποιήθηκε με την χρήση έτοιμων gizmos, αλλά σε πολλές περιπτώσεις ήταν απαραίτητη η τμηματοποίηση του μοντέλου με την χρήση τομών στα επιθυμητά σημεία.



Εικόνα 2.39 Δημιουργία τομών πάνω στην επιφάνεια του εκθέματος

Σε πολλές περιπτώσεις χρησιμοποιήθηκαν φωτογραφίες των πραγματικών εκθεμάτων στην δημιουργία των textures για απόδοση περισσότερης ρεαλιστικότητας στα μοντέλα.



Εικόνα 2.40 Διαδικασία δημιουργίας και προσαρμογής υφών με την χρήση texture maps



Unity 3D

3.1 Εισαγωγή στο Unity 3D

Το Unity είναι μια πλήρης πλατφόρμα ανάπτυξης τρισδιάστατων εφαρμογών. Η πρώτη εμφάνιση του προγράμματος έγινε το 2004 από τον David Helgason, Nicholas Francis και Joachim Ante στην Δανία, ενώ από το 2009 άρχισε να προσφέρεται δωρεάν μέσα από μια νέα έκδοση του προϊόντος. Το Unity κατασκευάστηκε για να παρέχει πλήρη υποστήριξη ανάπτυξης 3d εφαρμογών σε λειτουργικά συστήματα και κονσόλες, με αποτέλεσμα να καλύπτει ένα μεγάλο μέρος προγραμματιστικών απαιτήσεων. Μέσα από αυτό μπορούν να κατασκευαστούν εφαρμογές για Direct3D (Windows), OpenGL (Mac, Windows), OpenGL ES (iOS, Android), και APIs (Wii). Ακόμα παρέχει υποστήριξη σε άλλα προγραμματιστικά εργαλεία, όπως 3ds Max, Maya, Softimage, Blender, Modo, ZBrush, Cinema 4D, Cheetah3D, Photoshop, Adobe Fireworks και Allegorithmic Substance. Μέσα από το περιβάλλον εργασίας του Unity προσφέρεται μια μεγάλη γκάμα από εργαλεία και επιλογές για τον αρχιτεκτονικό σχεδιασμό, την ανάπτυξη και την εξαγωγή διαδραστικών τρισδιάστατων εφαρμογών. Σε προγραμματιστικό επίπεδο, η χρήση κώδικα (Scripting) που υποστηρίζει στις τελευταίες εκδόσεις είναι η Boo, η JavaScript και η C#. Ακόμα παρέχει ενσωματωμένη υποστήριξη της Nvidia, της PhysX σε πραγματικό χρόνο και υπάρχει η δυνατότητα ανάπτυξης Particle Systems. Τέλος, μέσα από το Unity παρέχονται τεχνικές Lighting και Lightmapping με πολλές δυνατότητες προσομοίωσης σκιάσεων και ρεαλιστικού φωτισμού.

3.2 Το περιβάλλον εργασίας του Unity.

Το Unity περιέχει μια μεγάλη γκάμα από εργαλεία για την ανάπτυξη και εξαγωγή τρισδιάστατων εφαρμογών. Μέσα από το κεντρικό Menu συναντάμε επιλογές για την δημιουργία βασικών αντικειμένων, φωτισμού, ανάπτυξη γραφικών εφέ, εισαγωγή συστημάτων φυσικής και διαχείρισης για real time εφαρμογές κ.α.

Εκτός από το κύριο μενού, το περιβάλλον εργασίας του Unity θα λέγαμε ότι τμηματοποιείται σε 5 κύρια τμήματα: Scene, Game, Hierarchy, Project και Inspector.

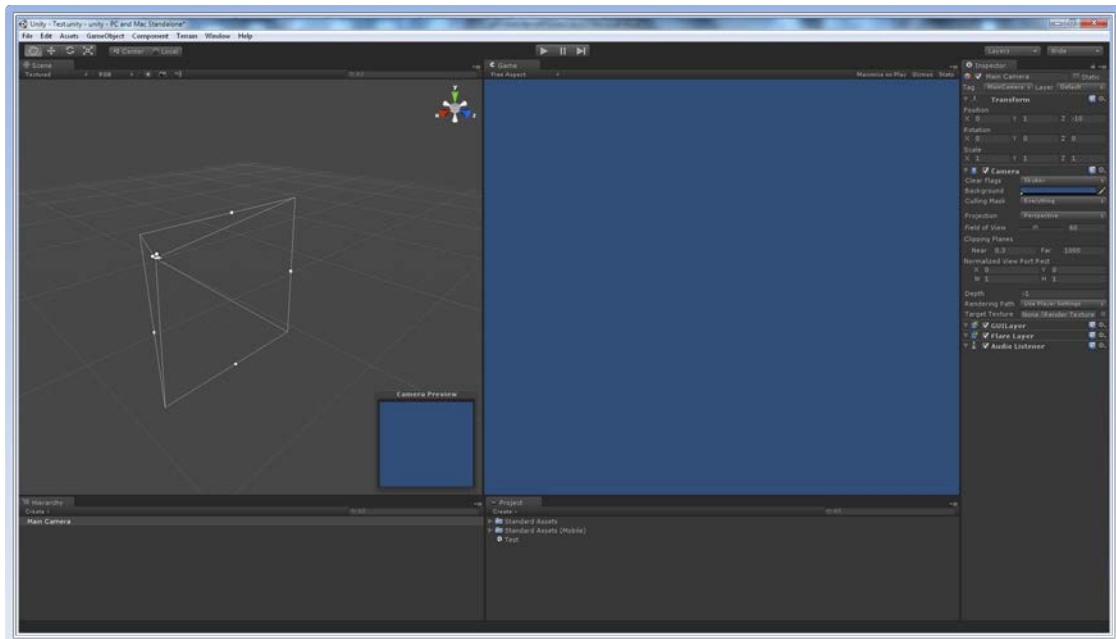
Το Scene αφορά το παράθυρο εργασίας όπου απεικονίζεται γραφικά το project το οποίο βρίσκεται σε εξέλιξη. Μέσα από αυτό μπορεί να γίνει διαχείριση και τροποποίηση όλων των αντικειμένων που βρίσκονται στην σκηνή, έχοντας παράλληλα οπτική αναπαράσταση. Η σκηνή μπορεί προβληθεί από διάφορες οπτικές γωνίες μέσω των X,Y,Z αξόνων.

Το παράθυρο Game προβάλλει την απεικόνιση της σκηνής μέσα από το οπτικό πεδίο της κύριας κάμερας. Τα αντικείμενα που προβάλλονται, περνάνε από την διαδικασία render και το αποτέλεσμα εμφανίζεται στο παράθυρο ενώ δεν υπάρχει δυνατότητα παρέμβασης ή τροποποίησης στοιχείων μέσα από αυτό.

Η καρτέλα Hierarchy αφορά την ονομαστική καταγραφή όλων των αντικειμένων που έχουν εισαχθεί στην σκηνή σε μια λίστα, περιγράφοντας την ιεραρχική τους δομή. Μέσα από αυτήν την λίστα μπορούν να επιλεγθούν αντικείμενα της σκηνής μεμονωμένα και να ελεγχθούν οι μεταξύ τους συσχετισμοί.

Η καρτέλα Project αφορά την ονομαστική παρουσίαση όλων των περιεχομένων του project, συμπεριλαμβανομένων στοιχείων και βιβλιοθηκών του προγράμματος με όλες της δυνατότητες εισαγωγής αντικειμένων, συστημάτων, αρχείων κώδικα κτλ.

Η περιοχή Inspector αφορά την διαχείριση και τροποποίηση κάθε στοιχείου του project ξεχωριστά, μέσα από παραμέτρους και εργαλεία που προσφέρονται για το κάθε ένα.



Εικόνα 3.1 Περιβάλλον εργασίας Unity³.

3.3 Ανάπτυξη περιβάλλοντα χώρου

Σε μια εφαρμογή απεικόνισης ενός τρισδιάστατου κόσμου, όπως είναι για παράδειγμα ένα ηλεκτρονικό παιχνίδι, το περιβάλλον μέσα στο οποίο θα βρίσκεται εικονικά ο χρήστης παίζει σημαντικό ρολό στην ποιότητα του παιχνιδιού, αφού ένα καλοφτιαγμένο και λεπτομερές περιβάλλον, κάνει την χρήση της εφαρμογής περισσότερο ευχάριστη και ελκυστική. Βέβαια η διαμόρφωση του περιβάλλοντα χώρου διαφοροποιείται ανάλογα με τις απαιτήσεις της εφαρμογής και τη κρίση του κατασκευαστή.

Παρακάτω περιγράφεται ο τρόπος με τον οποίον μπορεί να κατασκευαστεί ένα 3D περιβάλλον στο Unity με την χρήση Terrain, Skybox και εισαγωγή Lighting στοιχείων στην σκηνή.

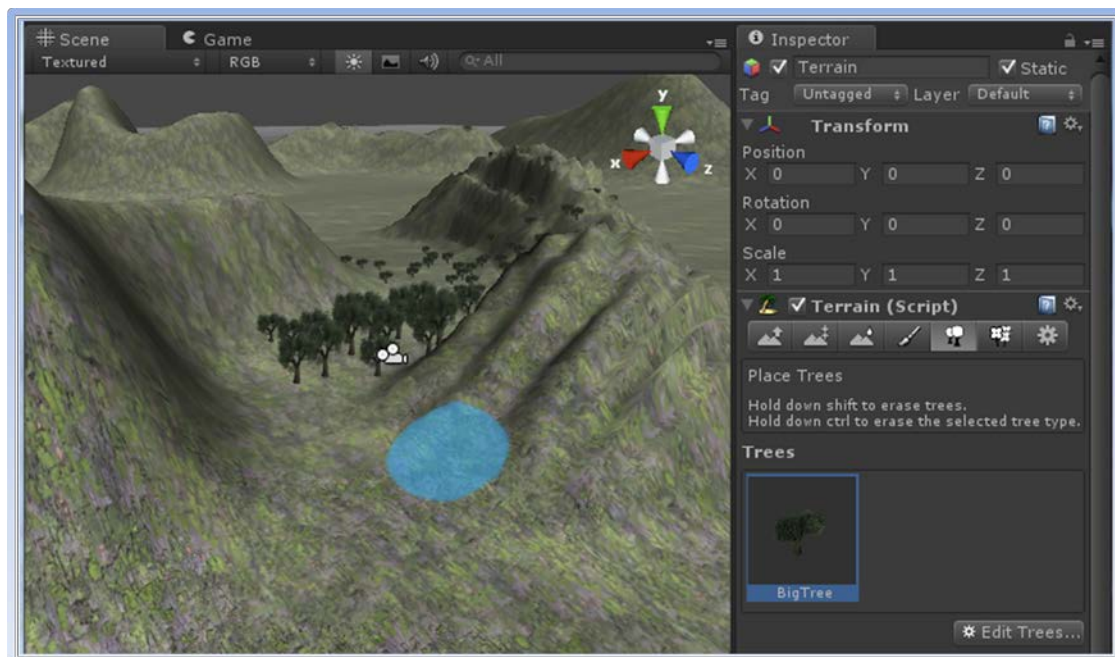
3.3.1 Terrain

Το Terrain μπορούμε να πούμε ότι είναι το περιβάλλον ενός τρισδιάστατου κόσμου πάνω στο οποίο θα τοποθετηθούν όλα τα 3d μοντέλα. Ένα Terrain προσομοιώνει το εξωτερικό περιβάλλον μιας εφαρμογής και μπορεί να δώσει την αίσθηση του φυσικού τοπίου, απεικονίζοντας μορφολογικά χαρακτηριστικά του εδάφους και στοιχεία όπως βουνά, κοιλάδες, θάλασσα, βλάστηση κ.τ.λ.

Για την κατασκευή ενός Terrain μέσω του Unity, επιλέγουμε Terrain από το κύριο μενού και στην συνέχεια Create Terrain. Αυτό που δημιουργείται είναι μια επιφάνεια (Plane) όπου μέσα από τα εργαλεία του Terrain editor μπορούμε να διαχειριστούμε την μορφολογία της επιφάνειας και να προσθέσουμε στοιχεία και αντικείμενα όπως δέντρα, γρασίδι, νερό, βράχια κ.α. Ακόμα, μέσα από υφές που μπορούμε να προσαρμόσουμε, προσομοιώνουμε διάφορα είδη εδάφους στην επιφάνεια του Terrain.



Σχήμα 3.1 Εργαλεία του terrain editor



Εικόνα 3.2 Διαμόρφωση terrain

Μόλις ολοκληρωθεί η μορφολογία του Terrain μπορούμε να εισάγουμε επιπλέον μοντέλα στην σκηνή που θα δώσουν μια πιο ολοκληρωμένη και λεπτομερή μορφή στο περιβάλλον που επιθυμούμε.

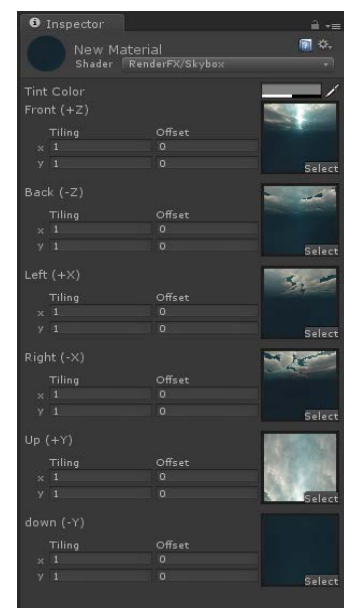
3.3.2 SkyBox

Η τοποθέτηση skybox στην σκηνή δίνει την δυνατότητα απεικόνισης του ουράνιου θόλου και του μακρινού ορίζοντα στον περιβάλλοντα χώρο. Με αυτόν τον τρόπο δίνεται μια αίσθηση συνέχειας στη σκηνή και χρησιμοποιείται για ανοικτούς εξωτερικούς χώρους. Ένα skybox αποτελείται από ένα συνδυασμό έξι εικόνων που συνήθως απεικονίζουν τμήματα του ουρανού προσαρμοσμένα κατάλληλα γύρο από την σκηνή.

Για την εισαγωγή Skybox στην σκηνή αρχικά πρέπει να δημιουργήσουμε έξι εικόνες που να απεικονίζουν τμήματα του ουρανού και να τις εισάγουμε στον φάκελο Assets του project. Σε κάθε εικόνα χρειάζεται να αλλάξουμε το wrap mode στην ένδειξη Clamp. Μέτα από αυτό δημιουργούμε ένα νέο Material (Assets→Create →Material). Στην συνέχεια επιλέγουμε τον Shader drop-down από τον Inspector, και τον ορίζουμε σαν Skybox (Render FX→Skybox). Τώρα για κάθε πλευρά του Skybox, front, back, left, right, up και down επιλέγουμε μια από τις εικόνες αντίστοιχα. Επιλέγουμε Render Settings από το μενού Edit και επιλέγουμε ή τραβάμε (drag and drop) το Material στο slot του Skybox στον Inspector.



Εικόνα 3.3 Εφαρμογή του skybox στην σκηνή



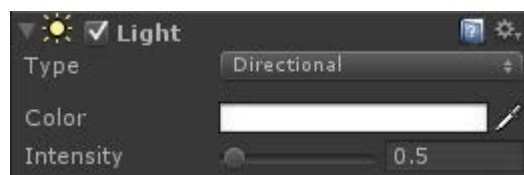
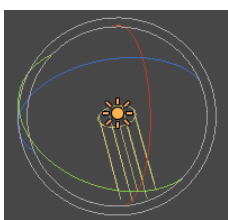
Σχήμα 3.2 Παράμετροι του material Skybox

3.3.3 Lighting

Η χρήση κατάλληλου φωτισμού στην σκηνή, μπορεί να προσδώσει περισσότερη αίσθηση ρεαλιστικότητας στο τρισδιάστατο περιβάλλον. Μέσα από διάφορους τύπους φώτων (lights) αποδίδονται σκιάσεις και φωτισμοί στα αντικείμενα που βρίσκονται σε ένα τρισδιάστατο χώρο.

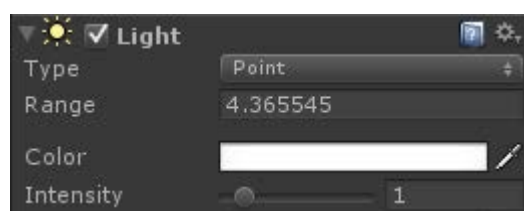
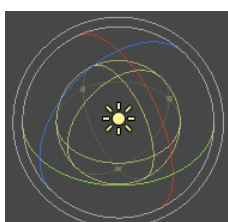
Το Unity, όπως και τα περισσότερα προγραμματιστικά εργαλεία του είδους υποστηρίζει τριών ειδών lights: Directional Light, Point Light και Spot Light. Η εισαγωγή ενός Light γίνεται από το Game Object → Create Other → Directional/Spot/Point Light. Μέτα από αυτό, το light, εμφανίζεται στη λίστα Hierarchy και οι παράμετροί του τροποποιούνται μέσω του Inspector. Για την κατάλληλη τοποθέτηση τους στην σκηνή χρησιμοποιούνται τα εργαλεία Move και Rotation.

Το **Directional Light** χρησιμοποιείται κυρίως για τον γενικό φωτισμό του περιβάλλοντα χώρου. Εκπέμπει φως στην επιθυμητή κατεύθυνση και συνήθως τοποθετείται μακριά από την σκηνή, ώστε να αποκτήσει τα χαρακτηριστικά μιας ισχυρής πηγής φωτός, όπως είναι ο ήλιος. Από τις παραμέτρους μπορεί να οριστεί η ένταση και το χρώμα του φωτός.



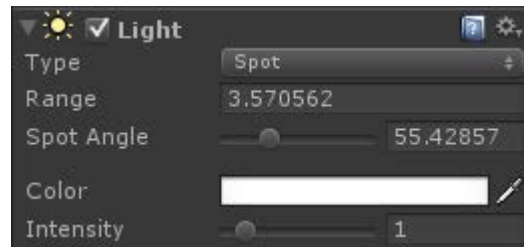
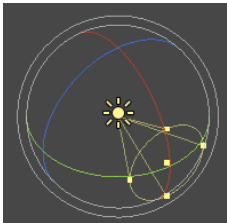
Σχήμα3.3 Παράμετροι και εφαρμογή του Directional light

Το **Point Light** χρησιμοποιείται στο φωτισμό μιας συγκεκριμένης περιοχής και εκπέμπει φως σφαιρικά, εξίσου σε όλες τις κατευθύνσεις. Από τις παραμέτρους ορίζεται η εμβέλεια εκπομπής, η ένταση και το χρώμα του φωτός.



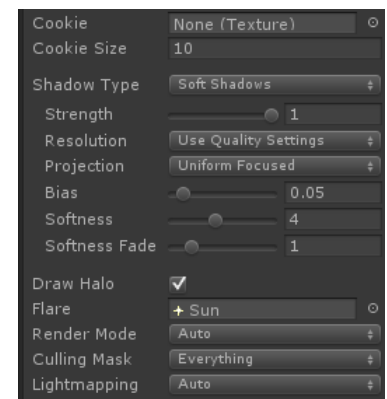
Σχήμα3.4 Παράμετροι και εφαρμογή του Point light

Το **Spot Light** έχει σαν ιδιαιτερότητα ότι φωτίζει συγκεκριμένα σημεία της σκηνής από μια συγκεκριμένη κατεύθυνση. Το φως εκπέμπεται κωνικά και φωτίζει όλα τα στοιχεία που βρίσκονται μέσα στην δέσμη. Από τις παραμέτρους ορίζεται η ένταση, η εμβέλεια η γωνία εκπομπής και το χρώμα του φωτός.



Σχήμα 3.5 Παράμετροι και εφαρμογή του Spot light

Εκτός από τις παραμέτρους που τροποποιούμε σε κάθε τύπο φωτός ανάλογα με την ένταση ή την εμβέλεια του, το Unity παρέχει μια σειρά από επιλογές για την εφαρμογή διαφόρων εφέ φωτισμού. Κάθε τύπος φωτισμού μπορεί να δημιουργεί ανάλογες σκιές πάνω στην σκηνή με την επιλογή Soft ή Hard Shadows. Οι σκιές μπορούν να διαμορφωθούν ανάλογα με την επιθυμητή ένταση και ποιότητα απεικόνισης της σκίασης από τις παραμέτρους. Με την επιλογή Halo και της κατάλληλες ρυθμίσεις μπορεί να δημιουργηθούν εφέ γύρο από την πηγή του φωτός, όπως η εμφάνιση μιας φωτεινής μπάλας ή η απεικόνιση φωτοστέφανου. Τέλος μέσω της προσαρμογής διαφόρων υφών στην περιοχή Cookie, το φως μπορεί να αποκτήσει διάφορα σχήματα και σχέδια.



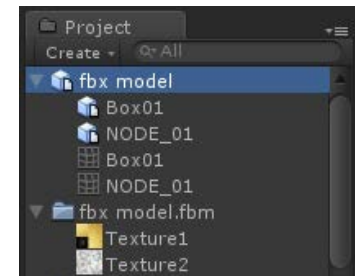
Σχήμα 3.6 Παράμετροι των Lights για την δημιουργία σκιάσεων και άλλων εφέ.



Εικόνα 3.4 Απεικόνιση διαφόρων εφέ με τη χρήση Halo και Cookie

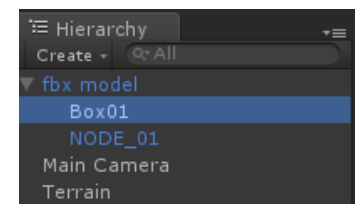
3.4 Εισαγωγή και διαχείριση FBX μοντέλων

Στο περιβάλλον του Unity μπορούν να εισαχθούν 3d models που έχουν δημιουργηθεί από άλλα προγράμματα ανάπτυξης τρισδιάστατων γραφικών όπως είναι το 3ds Max. Με την εισαγωγή τους στο Unity τα μοντέλα αποθηκεύονται στον κατάλογο Assets του τρέχοντος project και εμφανίζονται στην λίστα της περιοχής Project. Ένα μοντέλο μπορεί να αποτελείται από ένα σύνολο αντικειμένων. Ακόμα τα μοντέλα που έχουν αποθηκευτεί σαν αρχεία fbx μπορούν να περιέχουν στοιχεία όπως animations και textures. Αυτά τα στοιχεία αποθηκεύονται σε έναν ξεχωριστό φάκελο με κατάληξη .fbm που συνοδεύει το αρχείο.



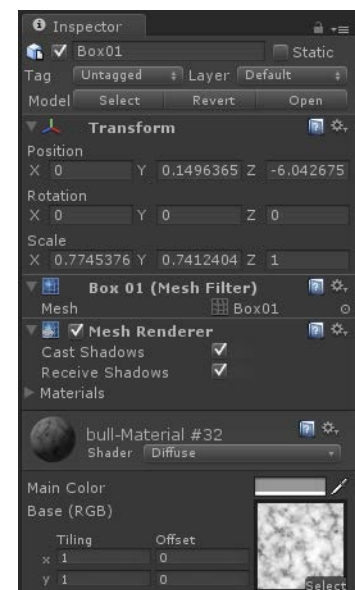
Σχήμα 3.7 Περιοχή "Project"

Για να τοποθετηθεί ένα αντικείμενο στην σκηνή πρέπει να δημιουργηθεί ένα στιγμιότυπο του και να εισαχθεί στην περιοχή Hierarchy. Σε αυτήν την λίστα καταγράφονται τα μοντέλα που υπάρχουν στην σκηνή και παρουσιάζεται ιεραρχικά η δομή από την οποία αποτελούνται. Κάθε αυτόνομο στοιχείο του μπορεί να αποδεσμευτεί από την ιεραρχία ή να εισαχθούν άλλα στοιχεία της σκηνής σε αυτή τη δομή.



Σχήμα 3.8 Περιοχή "Hierarchy"

Η δυνατότητα διαχείρισης των αντικειμένων που βρίσκονται στην σκηνή γίνεται μέσω της περιοχής του Inspector. Από το πεδίο Transform παρέχεται η δυνατότητα τροποποίησης χαρακτηριστικών των μοντέλων που αφορούν τη θέση, τη γωνία και το μέγεθός τους. Από το πεδίο Mesh Render επιλέγεται η δυνατότητα σε ένα μοντέλο να εμφανίζεται ή όχι στην σκηνή, καθώς και να δημιουργεί ή να δέχεται σκιάσεις. Ακόμα, μέσα από αυτό μπορεί να οριστεί η υφή και το shader που θα εφαρμοστεί στο μοντέλο μέσα από την περιοχή του Material.



Σχήμα 3.9 Περιοχή "Inspector"

Μέσω του Inspector υπάρχει επίσης η δυνατότητα διαχείρισης διαφόρων παραμέτρων και τροποποίησης κάποιων στοιχείων για τα πρωτότυπα fbx μοντέλα που βρίσκονται στην περιοχή του Project.

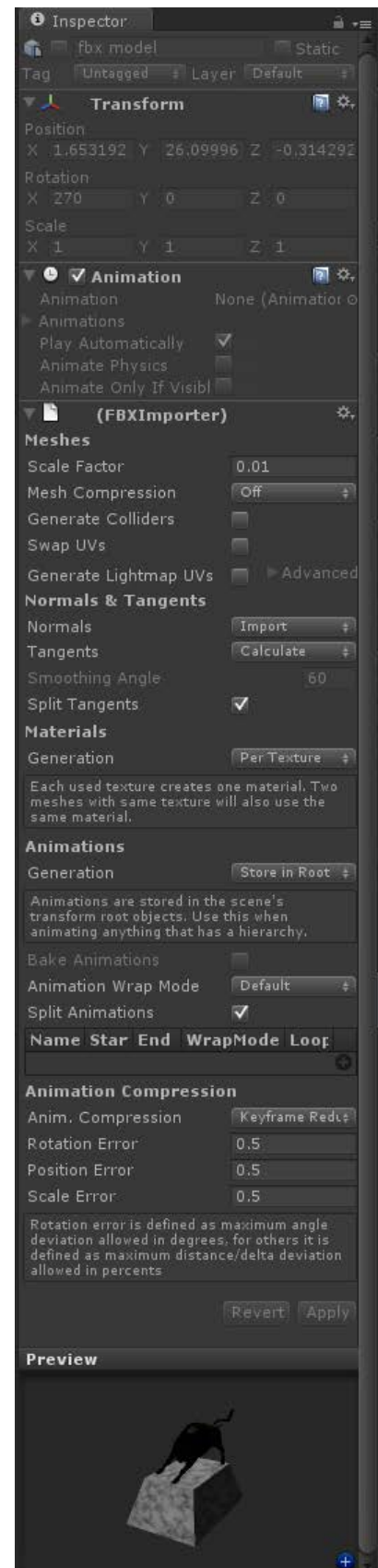
Από τις παραμέτρους του Transform μπορεί να γίνει διαχείριση της θέσης, της κλίμακας, και της γωνίας του μοντέλου που θα εισάγεται στην σκηνή στους άξονες X,Y,Z.

Από το πεδίο του Animation μπορεί να ενεργοποιηθεί ή να απενεργοποιηθεί το animation που πιθανός να περιέχει ένα αντικείμενο. Επίσης υπάρχουν επιλογές που αφορούν την αυτόματη εκτέλεσή του, τη δυνατότητα εφαρμογής στοιχείων φυσικής κατά την εκτέλεση και την επιλογή να εκτελείται μονό εάν το αντικείμενο στην σκηνή είναι εμφανή στην κάμερα.

Στο πεδίο του FBXImporter δίνεται η δυνατότητα διαφόρων επιλογών που αφορούν το μοντέλο, όπως είναι η συρρίκνωση μεγέθους του mesh, η παραγωγή και εφαρμογή colliders, η παραγωγή lightmap και ο υπολογισμός των normal του αντικειμένου. Ακόμα υπάρχουν επιπλέον επιλογές για την διαχείριση του animation, όπως η συρρίκνωση του μεγέθους του, η δυνατότητα να μπορεί να χωριστεί σε μικρότερα τμήματα σε καθορισμένα χρονικά διαστήματα, και ο ορισμός της επανάληψης(loop) στην εκτέλεσή του.

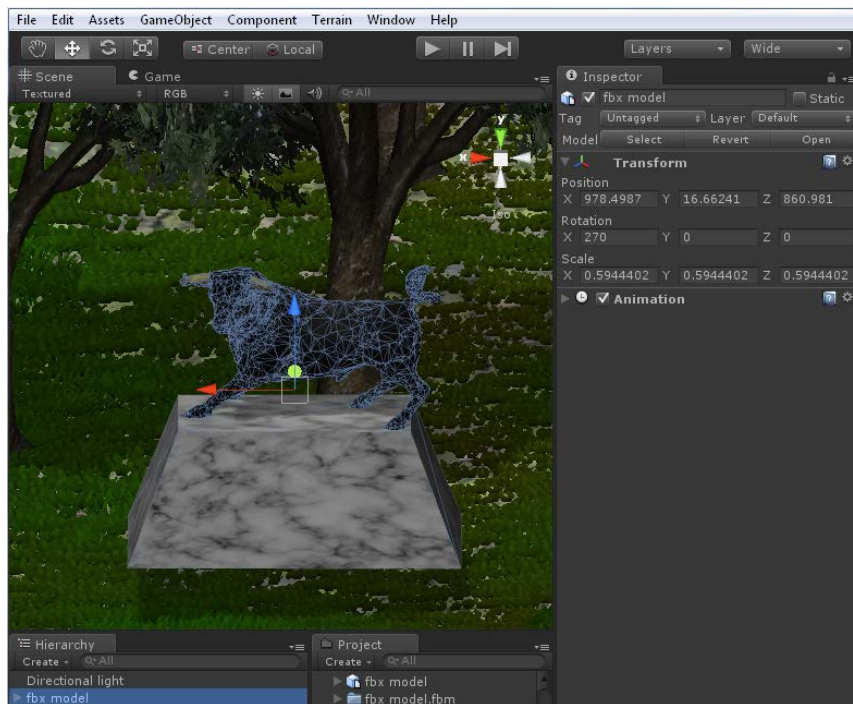
Στο πεδίο Preview απεικονίζεται γραφικά το μοντέλο και υπάρχει η δυνατότητα προβολής από όλες τις πλευρές του κάνοντας drag and move πάνω στην εικόνα.

Μετά την ολοκλήρωση των ρυθμίσεων υπάρχει η επιλογή Revert για να επαναφέρουμε τις ρυθμίσεις στην αρχική τους κατάσταση και η επιλογή Apply για να εφαρμόσουμε τις επιθυμητές αλλαγές.



Σχήμα 3.10 Περιοχή "Inspector"

Μετά την εισαγωγή ενός μοντέλου στην σκηνή, παρουσιάζεται στην οθόνη του Scene. Με τα βασικά εργαλεία, Move και Rotation, μπορούμε να το τοποθετήσουμε στην επιθυμητή θέση και να τροποποιήσουμε το μέγεθός του με το εργαλείο Scale.



Εικόνα 3.5 Τοποθέτηση μοντέλου στην σκηνή

3.5 Προβολή και διαδραστικότητα εφαρμογής.

Η ολοκλήρωση της δημιουργίας ενός τρισδιάστατου κόσμου, διαμορφώνοντας τον χώρο και τοποθετώντας στοιχεία και αντικείμενα στη σκηνή, αποτελεί το πρώτο σκέλος στη διαδικασία ανάπτυξης μιας 3d διαδραστικής εφαρμογής. Το δεύτερο σκέλος αφορά τον τρόπο με τον οποίο θα προβάλλεται η εφαρμογή στην οθόνη και την δυνατότητα του χρήστη να κινείται και να παρεμβαίνει δυναμικά στον χώρο.

3.5.1 Camera

Για να δοθεί η δυνατότητα σε έναν χρήστη να βρεθεί εικονικά σε ένα τρισδιάστατο περιβάλλον που έχει αναπτυχθεί μέσω του Unity, θα πρέπει να χρησιμοποιηθεί η ιδιότητα της κάμερας. Μέσα από την κάμερα παρουσιάζεται το περιβάλλον και τα αντικείμενα της σκηνής κατά την εκτέλεση της εφαρμογής και αποτελεί θα λέγαμε το στοιχείο εκείνο που επιτρέπει την εικονική μετάβαση του χρήστη στην σκηνή.

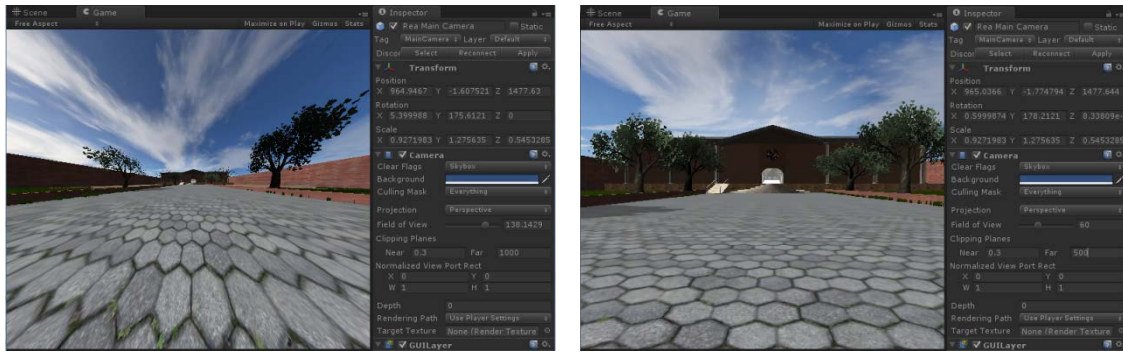
Κατά την δημιουργία ενός νέου project, το Unity εισάγει μια κυρία κάμερα στην σκηνή, όπου επιτρέπει την οπτική αναπαράσταση της εφαρμογής από το παράθυρο του Game σε όλη διάρκεια ανάπτυξης της. Υπάρχει επίσης η δυνατότητα να απεικονίζεται η σκηνή από διαφορετικές οπτικές γωνίες με την χρήση περισσότερων από μια cameras. Αν θέλουμε να εισάγουμε μια νέα κάμερα, επιλέγουμε GameObject από το κύριο μενού εργαλείων και στην συνέχεια Create Other→ Camera.

Στις παραμέτρους της κάμερας, όπως και σε όλα τα αντικείμενα που υπάρχουν στην σκηνή, υπάρχει το πεδίο Transform, όπου μέσα από αυτό μπορεί να τοποθετηθεί στο κατάλληλο σημείο. Στο πεδίο Camera υπάρχουν επιλογές που αφορούν την λήψη της κάμερας. Από εκεί καθορίζεται το είδος, η εμβέλεια, η γωνία και το βάθος λήψης, όπως καθορίζονται και στοιχεία που αφορούν το background της σκηνής.

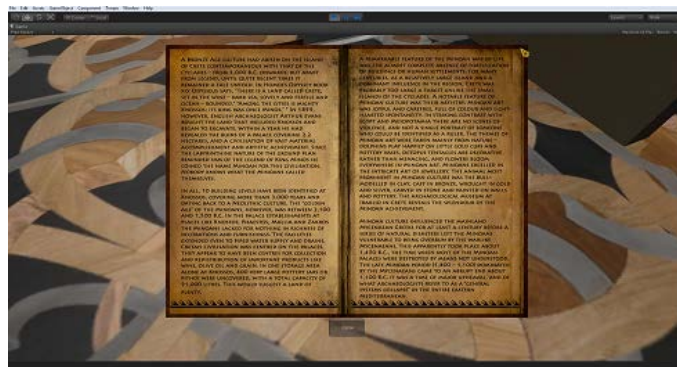
Κάθε camera διαθέτει κάποια Layers, για να υπάρχει μια μορφή ιεραρχίας σε κάποια επιπλέον στοιχεία που μπορούν να εμφανίζονται μέσα από αυτήν, όπως είναι GUI αντικείμενα (buttons, texts, textures, windows κ.τ.λ.) και διάφορα Flare effects. Ακόμα, μια κάμερα διαθέτει έναν Audio Listener, όπου μπορούν να τοποθετηθούν αρχεία ήχου, έχοντας το ρολό background μουσικής κατά τη διάρκεια εκτέλεσης της εφαρμογής.



Σχήμα 3.11 Παράμετροι κάμερας



Εικόνα 3.6 Λήψεις κάμερας με τροποποίηση του Field of View



Εικόνα 3.7 Εμφάνιση στοιχείων GUI στην κάμερα

3.5.2 Scripting

Η χρήση κώδικα είναι απαραίτητη σε 3d εφαρμογές που περιέχουν διαδραστικά στοιχεία. Για παράδειγμα, η κίνηση και ο έλεγχος ενός χαρακτήρα σε ένα 3d game ή γενικότερα η αλληλεπίδραση του χρήστη με μια εφαρμογή “κρύβουν” σειρές κώδικα που άλλοτε παρέχονται έτοιμες από τα προγραμματιστικά εργαλεία ανάπτυξης μέσα από βιβλιοθήκες και άλλοτε χρειάζεται η επέμβαση του κατασκευαστή για την δημιουργία συναρτήσεων που να επιτελούν συγκεκριμένες ενέργειες και συμπεριφορές.

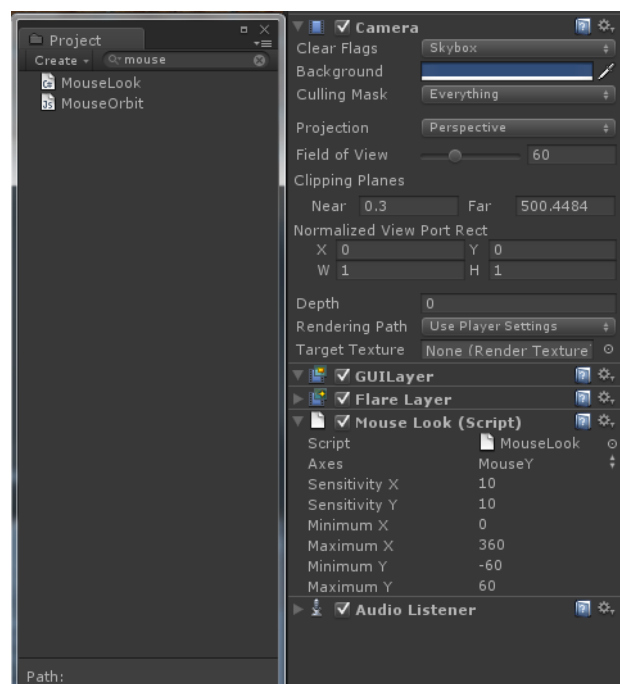
Το Unity υποστηρίζει 3 γλώσσες προγραμματισμού, Boo ,C# και JavaScript που αναπτύσσονται σε MonoDevelop editor και δηλώνονται σαν MonoBehaviour. Κάθε αρχείο κώδικα έχει συγκεκριμένο τύπο κατάληξης , .cs για C#, .js για JavaScript και .boo για την Boo αντίστοιχα. Κάθε αρχείο τέτοιου είδους αποθηκεύεται στον φάκελο Assets του project και εμφανίζεται στην περιοχή

“Project” του περιβάλλοντος εργασίας του Unity. Από εκεί προσαρμόζονται οι ιδιότητες του κώδικα σε αντικείμενα ή σε άλλα στοιχεία της σκηνής και μπαίνουν σε διαδικασία ενεργοποίησης στην διάρκεια εκτέλεσης της εφαρμογής. Διαφορετικά αντικείμενα μπορούν να υιοθετήσουν ίδιες ιδιότητες από ένα αρχείο κώδικα και να προσαρμόσουν την συμπεριφορά τους ανάλογα με τις μεταβλητές που θα δοθούν στο κάθε ένα.

3.5.2.1 Προσαρμογή έτοιμου κώδικα σε αντικείμενο.

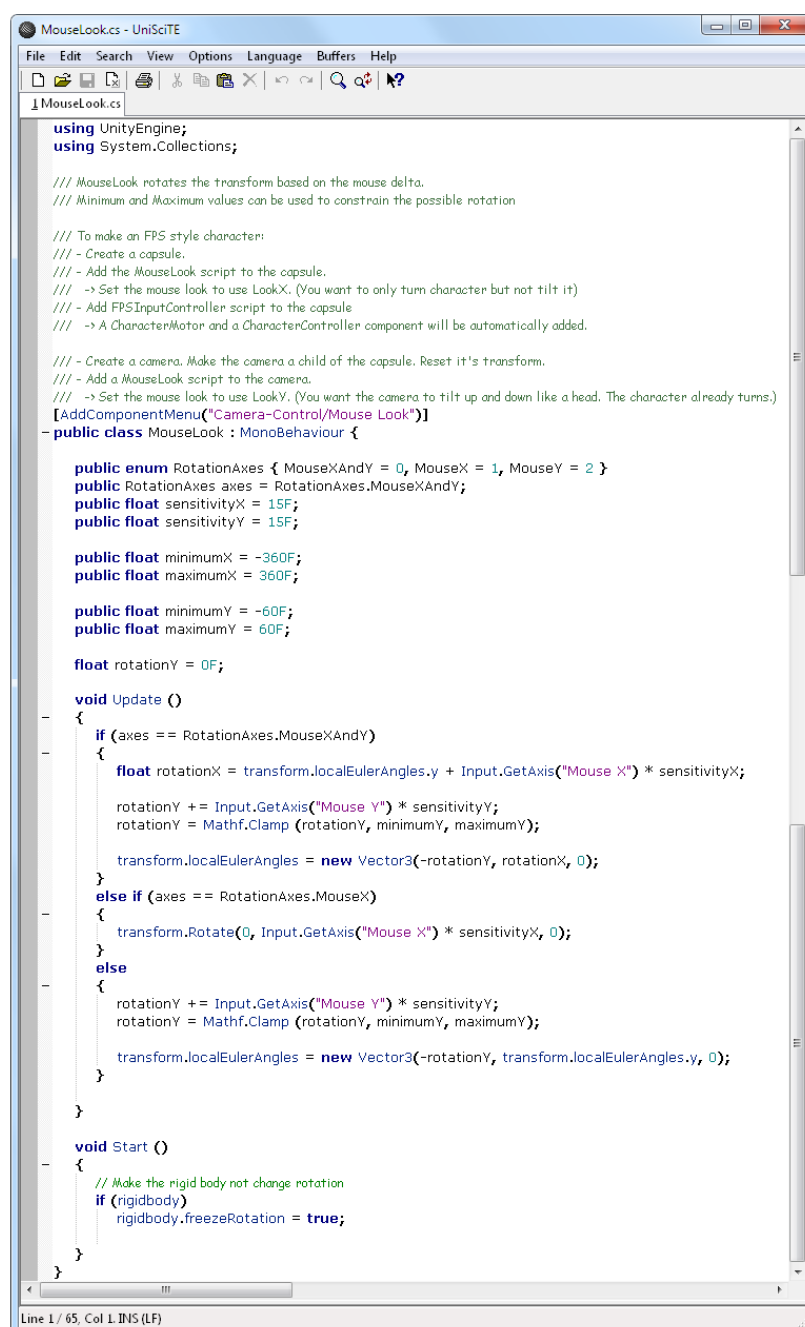
Το Unity περιλαμβάνει βιβλιοθήκες με έτοιμα αρχεία κώδικα που μπορούν να χρησιμοποιηθούν σε αντικείμενα για να εφαρμόσουν βασικές ενέργειες και λειτουργίες. Κάποιες από τις συναρτήσεις που περιέχουν αυτά τα αρχεία έχουν συνταχθεί για να απεικονίσουν βασικές μεθόδους κίνησης μιας κάμερας ή ενός avatar στην σκηνή. Υπάρχουν Script ελέγχου από τον χρήστη για περιπτώσεις που αφορούν κίνηση ενός χαρακτήρα σε πρώτο πρόσωπο(FPSInputController), κίνηση ενός χαρακτήρα σε τρίτο πρόσωπο(ThirdPersonController), έλεγχος κάμερας σε τρίτο πρόσωπο (ThirdPersonCamera), περιστροφή κάμερας στον χώρο (Mouse Look)κ.α.

Η προσαρμογή ενός Script αρχείου σε ένα αντικείμενο μπορεί να γίνει με την διαδικασία drag and drop πάνω του. Αμέσως εμφανίζεται στην περιοχή του Inspector σαν μια νέα νέο πεδίο από όπου μπορούν να τροποποιηθούν ή να αρχικοποιηθούν μεταβλητές του κώδικα εξωτερικά. Με αυτόν τον τρόπο μπορούμε να δώσουμε στο αντικείμενο μια συγκεκριμένη συμπεριφορά ή δυνατότητα που επιθυμούμε. Παραδείγματος χάριν για να μπορούμε να χειριστούμε την κάμερα κατά την διάρκεια εκτέλεσης της εφαρμογής,



Σχήμα 3.12 Εφαρμογή έτοιμου κώδικα για την περιστροφή της κάμερας

ώστε να περιστρέφεται στον χώρο με την κίνηση του ποντικιού χρειάζεται να προσαρμόσουμε ένα κώδικα που να επιτρέπει αυτήν την δυνατότητα. Σε αυτήν την περίπτωση παρέχεται έτοιμος κώδικας από την βιβλιοθήκη του Unity που ονομάζεται “Mouse Look”. Καθώς εφαρμόζεται ο κώδικας στην κάμερα ,εμφανίζονται στον Inspector οι μεταβλητές που χρησιμοποιούνται. Με την τροποποίηση αυτών των μεταβλητών μπορούμε να αλλάξουμε κάποιες παραμέτρους στην ζητούμενη συμπεριφορά της περιστροφής της κάμερας. Όταν η εφαρμογή θα εκτελεσθεί η κάμερα πλέον θα μπορεί να περιστρέφεται ακολουθώντας το mouse.



```

MouseLook.cs - UniSciTE
File Edit Search View Options Language Buffers Help
MouseLook.cs
using UnityEngine;
using System.Collections;

/// MouseLook rotates the transform based on the mouse delta.
/// Minimum and Maximum values can be used to constrain the possible rotation

/// To make an FPS style character:
/// - Create a capsule.
/// - Add the MouseLook script to the capsule.
/// - Set the mouse look to use LookX. (You want to only turn character but not tilt it)
/// - Add FPSInputController script to the capsule
/// - A CharacterMotor and a CharacterController component will be automatically added.

/// - Create a camera. Make the camera a child of the capsule. Reset it's transform.
/// - Add a MouseLook script to the camera.
/// - Set the mouse look to use LookY. (You want the camera to tilt up and down like a head. The character already turns.)
[AddComponentMenu("Camera-Control/Mouse Look")]
public class MouseLook : MonoBehaviour {

    public enum RotationAxes { MouseXAndY = 0, MouseX = 1, MouseY = 2 }
    public RotationAxes axes = RotationAxes.MouseXAndY;
    public float sensitivityX = 15F;
    public float sensitivityY = 15F;

    public float minimumX = -360F;
    public float maximumX = 360F;

    public float minimumY = -60F;
    public float maximumY = 60F;

    float rotationY = 0F;

    void Update ()
    {
        if (axes == RotationAxes.MouseXAndY)
        {
            float rotationX = transform.localEulerAngles.y + Input.GetAxis("Mouse X") * sensitivityX;

            rotationY += Input.GetAxis("Mouse Y") * sensitivityY;
            rotationY = Mathf.Clamp (rotationY, minimumY, maximumY);

            transform.localEulerAngles = new Vector3(-rotationY, rotationX, 0);
        }
        else if (axes == RotationAxes.MouseX)
        {
            transform.Rotate(0, Input.GetAxis("Mouse X") * sensitivityX, 0);
        }
        else
        {
            rotationY += Input.GetAxis("Mouse Y") * sensitivityY;
            rotationY = Mathf.Clamp (rotationY, minimumY, maximumY);

            transform.localEulerAngles = new Vector3(-rotationY, transform.localEulerAngles.y, 0);
        }
    }

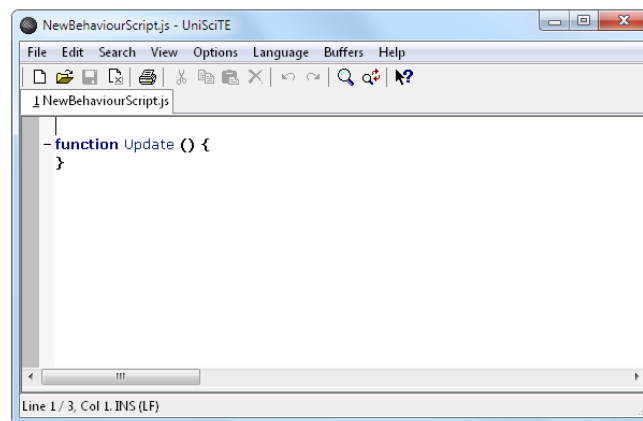
    void Start ()
    {
        // Make the rigid body not change rotation
        if (rigidbody)
            rigidbody.freezeRotation = true;
    }
}
Line 1 / 65, Col 1, INS (LF)

```

Εικόνα 3.8 Script για περιστροφή της κάμερας μέσα από το mouse

3.5.2.2 Δημιουργία νέου αρχείου κώδικα.

Η δημιουργία ενός νέου αρχείου κώδικα πραγματοποιείται από το μενού Assets → Create → JavaScript/C Sharp Script/ Boo Script ή Right Mouse Click πάνω στην περιοχή του Project και στην συνέχεια Create → JavaScript/C Sharp Script/ Boo Script. Μόλις δημιουργηθεί το κενό αρχείο κώδικα, μπορούμε αρχίσαμε να συντάσσουμε εκείνες τις εντολές και τις συναρτήσεις που θα μας δώσουν την επιθυμητή συμπεριφορά.



Εικόνα 3.9 Script Editor

Πριν να αρχίσουμε να γράφουμε κώδικα θα πρέπει να γνωρίζουμε την ιδιότητα και την ιδιαιτερότητα κάποιων βασικών χρήσιμων συναρτήσεων.

- `function Update() { }`

Η συνάρτηση Update καλείται σε κάθε χτύπημα (tic) του ρολογιού της εφαρμογής. Δηλαδή καλείται συνεχόμενα καθόλη την διάρκεια και ελέγχει άμεσα πιθανές αλλαγές σε κάθε frame. Στην update πρέπει να περιλαμβάνονται όλες εκείνες οι γραμμές κώδικα και οι συνθήκες που χειρίζονται τις άμεσες παρεμβάσεις εντολών ή άμεσες κλήσεις άλλων συναρτήσεων που μπορεί να προκύπτουν κατά την εκτέλεση της εφαρμογής.

- `function Start(){ }`

Η συνάρτηση `Start` καλείται στην εκκίνηση εκτέλεσης της εφαρμογής. Αρχικοποιεί κάποιες καταστάσεις αντικειμένων και δίνει την δυνατότητα να παραμείνουν σταθερές μέχρι να υπάρξει κάποια αλληλεπίδραση από κάποιο άλλο γεγονός.

- `function OnGUI () { }`

Μέσα από την συνάρτηση `OnGUI` παρέχεται η δυνατότητα απεικόνισης κάποιων στοιχείων στο πεδίο λήψης της κάμερας. Δηλαδή αυτά τα στοιχεία τα οποία καλούνται μέσα από αυτή τη συνάρτηση εμφανίζονται μονό στην οθόνη της κάμερας χωρίς να συσχετίζονται με το τρισδιάστατο περιβάλλον της εφαρμογής. Τέτοια στοιχεία είναι `GUI buttons`, `textures`, `windows`, `panels`, `texts`, `textboxes` κ.α.

- `function OnTriggerStay(){ }`

Αυτή η συνάρτηση αφορά την αλληλεπίδραση δύο αντικειμένων μεταξύ τους. Εκτελείται όσο ένα αντικείμενο έχει συνεχόμενη επαφή με κάποιο άλλο δημιουργώντας `collision` και παύει να εκτελείται όσο αυτό δεν ισχύει.

- `function OnTriggerEnter(){ }` & `function OnTriggerExit(){ }`

Ομοίως υπάρχουν οι συναρτήσεις `OnTriggerEnter` και `OnTriggerExit`. Η `OnTriggerEnter` εκτελείται μόλις κάποιο αντικείμενο εισέλθει στο χώρο που καταλαμβάνει κάποιο άλλο δημιουργώντας παράλληλα `collision` και η `OnTriggerExit` εκτελείται όταν εξέλθει από αυτόν τον χώρο.

- `function OnCollisionEnter(){ }` & `function OnCollisionExit(){ }`

Όπως και παραπάνω, οι συναρτήσεις `OnCollisionEnter` και `OnCollisionExit`, εκτελούνται σε περιπτώσεις αλληλεπίδρασης με `collision`. Δηλαδή ενεργοποιούνται όταν δύο στοιχεία εισέρχονται σε `collision` ή εξέρχονται από αυτό αντίστοιχα.

- `function OnMouseUp (){ }` & `function OnMouseEnter (){ }`

Οι συναρτήσεις `OnMouseUp` και `OnMouseEnter` αφορούν την αλληλεπίδραση του `mouse cursor` με κάποιο αντικείμενο. Η πρώτη εκτελείται στη περίπτωση που ο `cursor` του `mouse` βρεθεί πάνω από το αντικείμενο και η δεύτερη εκτελείται στη περίπτωση επιλογής του αντικειμένου με το `mouse` (`mouse click`).

Υπάρχουν ακόμα κάποιες `Coroutine` συναρτήσεις που μπορούν να περιλαμβάνονται μέσα σε άλλες και να παρεμβαίνουν στη διαδικασία της εκτέλεσης.

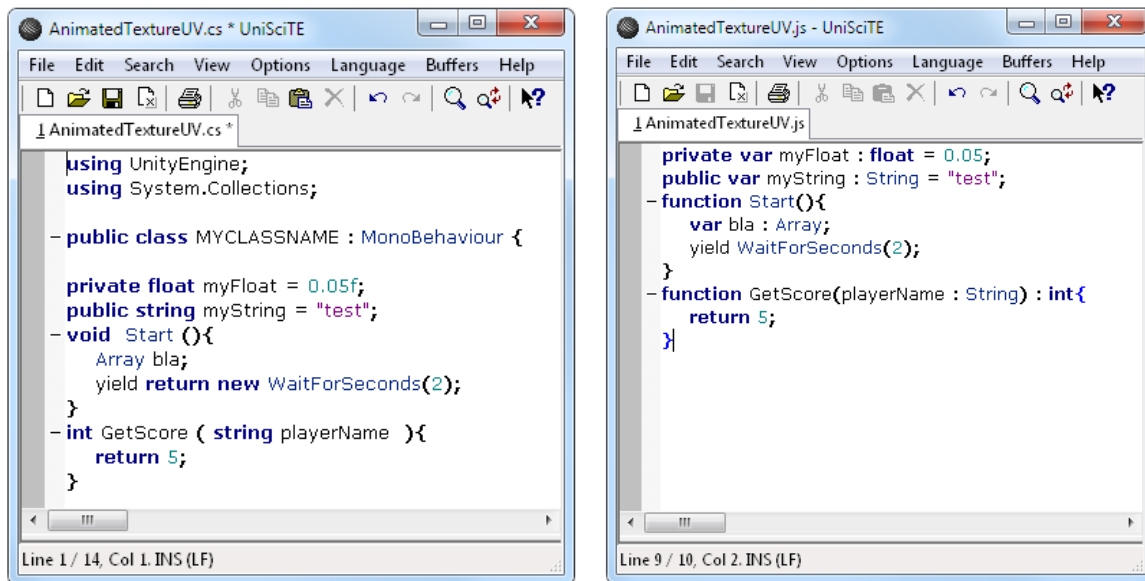
- `yield WaitForSeconds ()`

Η συνάρτηση `WaitForSeconds` παρατείνει την εκτέλεση των εντολών που βρίσκονται κάτω από αυτήν.

- `yield StartCoroutine()`

Η `StartCoroutine` χρησιμοποιείται για την δημιουργία μιας νέας `Coroutine` συνάρτησης.

Όλες οι παραπάνω συναρτήσεις χρησιμοποιούνται από όλες τις προγραμματιστικές γλώσσες που υποστηρίζονται από το Unity. Η επιλογή της γλώσσας που θα συνταχθεί ο κώδικας αφορά κυρίως θέμα προτίμησης του προγραμματιστή. Ένα εκτελέσιμο αρχείο που δημιουργείται μέσα από αυτό το πρόγραμμα μπορεί να περιλαμβάνει χρήση αρχείων κωδικά που έχουν συνταχθεί σε διαφορετικές γλώσσες. Κάποιες διαφορές που συναντώνται ανάμεσα στην JavaScript και την C# είναι για παράδειγμα ο διαφορετικός τρόπος δήλωσης κάποιων συναρτήσεων και μεταβλητών, όπως και κάποια στοιχεία στο συντακτικό της χρήσης `coroutine` συναρτήσεων. Παρακάτω δίνεται ένα παράδειγμα διαφοράς μεταξύ της C# και της JavaScript.



Εικόνα 3.10 Παράδειγμα διαφορών της C# και JavaScript

Ξεκινώντας την κωδικοποίηση με μια από τις γλώσσες που θα επιλέξουμε για να αναπτύξουμε μια συμπεριφορά, αρχικά δηλώνουμε τις μεταβλητές που θα χρησιμοποιηθούν στην πορεία. Στην συνέχεια μέσα από συναρτήσεις θέτουμε τις απαραίτητες εντολές και συνθήκες που θα δημιουργήσουν τις προϋποθέσεις για την αλλαγή καταστάσεων και συμπεριφορών.

```

//if a boolean is true
- if (true) {
    // do something
    statement;
- while (false){
    //while a boolean is false
    //do something
}
}

for ( ; condition; ) - while (true) {
- {
    //do something
    statements;
    //do complicated stuff
    if (someCondition)
        break;
    //more stuff
}
}

```

Εικόνα 3.11 Παραδείγματα χρήσης συνθηκών if, for και while

Κατά την διαδικασία του scripting μπορούν να δημιουργηθούν κλάσεις(classes) που να μπορούν να καλούνται από άλλα αρχεία κώδικα, ώστε να υιοθετούν μεταβλητές ή συναρτήσεις.

```

class Foo
- {
    virtual function DoSomething ()
    {
        Debug.Log("from base class");
    }
}

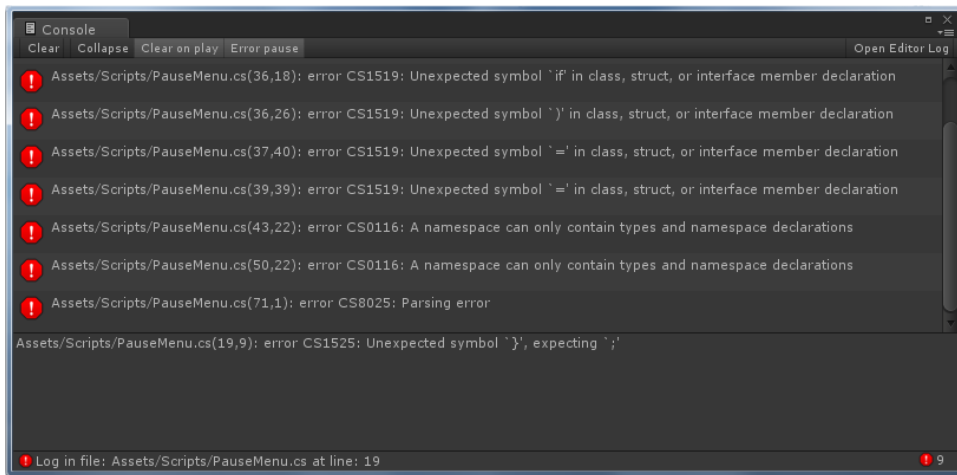
//SubFoo.js
class SubFoo extends Foo
- {
    virtual function DoSomething()
    {
        Debug.Log("from sub class");
    }
}

//Elsewhere
var foo : Foo = new SubFoo();
foo.DoSomething();//prints from sub class

```

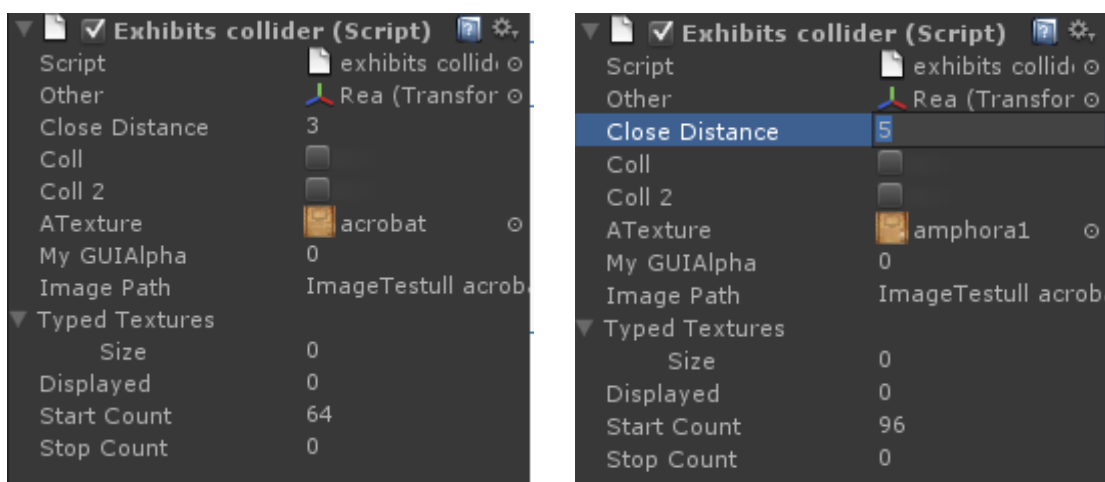
Εικόνα 3.12 Παράδειγμα χρήσης κλάσης

Με την αποθήκευση ενός αρχείου κώδικα στον φάκελο Assets του project ελέγχεται για πιθανά λάθη ή συντακτικά σφάλματα. Τα Errors και τα Warnings που ίσως βρεθούν, εμφανίζονται αναλυτικά στην κονσόλα του unity.



Εικόνα 3.13 Παράθυρο Console του Unity

Όταν ολοκληρωθεί η σύνταξη και η αποθήκευση του αρχείου ,μπορούμε να εφαρμόσουμε τις ιδιότητες του κώδικα με drag and drop του αρχείου πάνω στο αντικείμενο ή στις παραμέτρους του Inspector. Μπορούμε ακόμα να εφαρμόσουμε την ίδια συμπεριφορά και σε άλλα αντικείμενα επιπλέον τροποποιώντας τις κατάλληλες μεταβλητές στον Inspector κάθε αντικειμένου μπορούμε να έχουμε διαφορετικά αποτελέσματα.



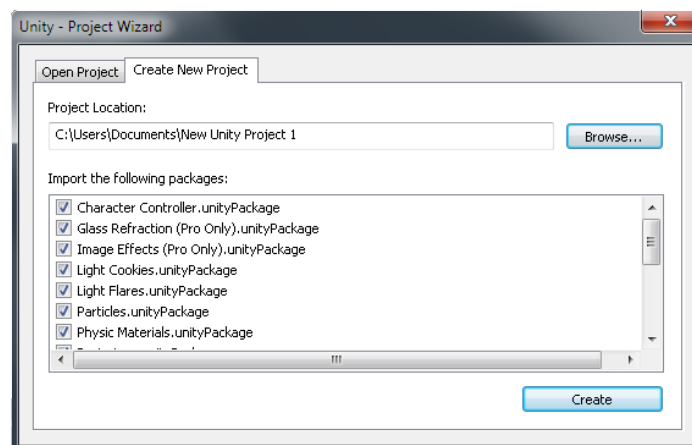
Εικόνα 3.14 Παράδειγμα τροποποίησης μεταβλητών σε αντικείμενα που χρησιμοποιούν τη ίδια συμπεριφορά

3.6 Πορεία ανάπτυξης της εφαρμογής μέσα από το Unity

Παρακάτω περιγράφεται η πορεία ανάπτυξης της εφαρμογής προσομοίωσης εικονικού μουσείου μέσα από το Unity. Αναφέρονται τα βήματα που ακολουθήθηκαν ξεκινώντας από την διαμόρφωση του περιβάλλοντος και την διαχείριση των στοιχείων του. Έπειτα παρουσιάζεται η εισαγωγή των fbx μοντέλων στο πρόγραμμα και η τοποθέτησή τους στη σκηνή. Στη συνέχεια αναλύεται η διαχείριση των animation του χαρακτήρα και όλα τα προγραμματιστικά στοιχεία που χρησιμοποιήθηκαν για την δυνατότητα περιήγησής του μέσα στο χώρο και την αλληλεπίδρασή του με διάφορα αντικείμενα.

3.6.1 Δημιουργία νέου Project

Για την δημιουργία του project χρειάστηκε να εισαχθούν κάποια πακέτα από τις βιβλιοθήκες του Unity που αφορούν εφέ φωτισμού, particle systems, στοιχεία φυσικής, βασικοί controllers κ.α. που ήταν απαραίτητα κατά την διαδικασία της ανάπτυξης της εφαρμογής.



Εικόνα 3.15 Παράθυρο του Project Wizard για την δημιουργία νέου project

3.6.2 Δημιουργία περιβάλλοντος

Στο πρώτο στάδιο ανάπτυξης της εφαρμογής, διαμορφώθηκε κατάλληλα το περιβάλλον πάνω στο οποίο θα μπορούσε να τοποθετηθεί το μουσείο και τα στοιχεία του περιβάλλοντα χώρου.

Αρχικά τοποθετήθηκε ένα Directional Light στην σκηνή που προσομοιώνει το φως του ήλιου και παραμετροποιήθηκε, ώστε να δημιουργεί απαλές σκιές στα αντικείμενα της σκηνής. Ακόμα, από το πεδίο Draw Halo επιλέχθηκε Sun στην λίστα του Falre, ώστε το φως να δημιουργεί ένα εφέ λάμψης γύρω του.



Σχήμα 3.13 Εφαρμογή του Draw Halo για την απεικόνιση εφέ λάμψης του ήλιου

Στην δημιουργία του Terrain χρησιμοποιήθηκαν τα εργαλεία του editor για την προσαρμογή της υφής του εδάφους, την δημιουργία γρασιδιού και την απεικόνιση μερικών δέντρων σε κάποια σημεία διαμορφώνοντας τον προαύλιο χώρο του μουσείου.



Εικόνα 3.16 Διαμόρφωση του terrain

Για την δημιουργία Skybox χρησιμοποιήθηκαν έξι τμήματα από μια φωτογραφία μεγάλης ανάλυσης όπου απεικονίζεται ο ουρανός. Τοποθετήθηκαν κατάλληλα στις πλευρές του skybox material και στη συνέχεια αυτό προσαρμόστηκε στον Renderer της σκηνής.

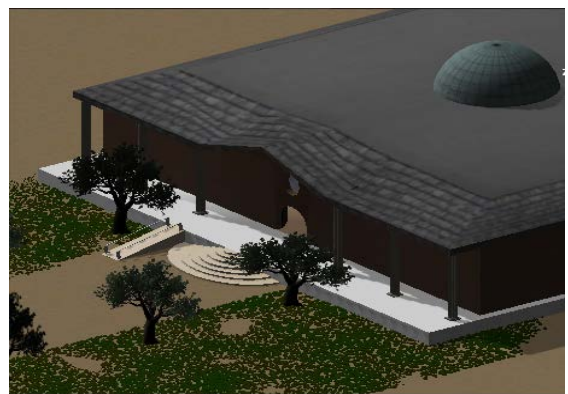


Εικόνα 3.17 Χρήση εικόνων στην απεικόνιση του ουράνιου θόλου

3.6.3 Εισαγωγή μοντέλων

Όλα τα μοντέλα εκείνα που χρειάστηκε να συμπεριληφθούν στην σκηνή μοντελοποιήθηκαν και αποθηκεύτηκαν σαν fbx files στον φάκελο Assets του Project. Τα μοντέλα τα οποία παραμένουν σταθερά στον χώρο, όπως είναι για παράδειγμα το κτίριο του μουσείου, δηλώθηκαν σαν Static αντικείμενα στην περιοχή του Inspector. Επιπροσθέτως, εφαρμόστηκαν collides στα μοντελα όπου θα μπορούσε να προκληθεί επαφή μεταξύ αυτών και κάποιου άλλου αντικειμένου.

Αρχικά τοποθετήθηκε το κτίριο του μουσείου στη σκηνή, προσαρμόζοντάς το στην ανάλογη κλίμακα με το περιβάλλον. Μπροστά του τοποθετήθηκε μια κυκλική σκάλα και δίπλα της μια ράμπα για την πρόσβαση στην είσοδο του μουσείου.



Εικόνα 3.18 Εισαγωγή του μουσείου στη σκηνή

Στην συνέχεια διαμορφώθηκε περαιτέρω ο προαύλιος χώρος, τοποθετώντας την κεντρική πύλη και κάποιους τοίχους από Cubes γύρω από το μουσείο ώστε να περιοριστεί η δυνατότητα περιήγησης έξω από αυτόν τον χώρο. Επίσης, χρησιμοποιήθηκαν Planes με προσαρμοσμένες υφές πάνω τους για την απεικόνιση του δαπέδου.



Εικόνα 3.19 Διαμόρφωση προαύλιου χώρου

Ακολουθήθηκε η ίδια διαδικασία, χρησιμοποιώντας Planes για την απεικόνιση των δαπέδων και στο εσωτερικό του μουσείου, εκτός το πάτωμα του πρώτου ορόφου, όπου έχει προσαρμοστεί ήδη υφή κατά την διαδικασία της μοντελοποίησης.



Εικόνα 3.20 Δημιουργία δαπέδων και προσαρμογή υφών πάνω τους

Αφού ολοκληρώθηκε το κομμάτι που αφορά το κτιριακό μέρος, τοποθετήθηκαν τα μοντέλα που απεικονίζουν τα διάφορα εκθέματα. Κάποια από αυτά χρειάστηκε να τοποθετηθούν πάνω σε βάσεις στις οποίες χρησιμοποιήθηκαν έτοιμα σχήματα του Unity όπως Cubes και Cylinders.



Εικόνα 3.21 Εισαγωγή και τοποθέτηση εκθεμάτων στο εσωτερικό του μουσείου

Τέλος, στη τοποθέτηση των μοντέλων στη σκηνή συμπεριλήφθηκε και η εισαγωγή του χαρακτήρα (avatar) του οποίου η ανάπτυξη περιγράφεται στο προηγούμενο κεφάλαιο. Όπως θα αναλυθεί στις παρακάτω σελίδες, μέσα από αυτόν, ο χρήστης μπορεί να κινηθεί και να περιηγηθεί στον χώρο, καθώς και να αλληλεπιδράσει με τα διάφορα αντικείμενα.



Εικόνα 3.22 Εισαγωγή και τοποθέτηση του χαρακτήρα στη σκηνή

3.6.4 Εισαγωγή στοιχείων φωτισμού

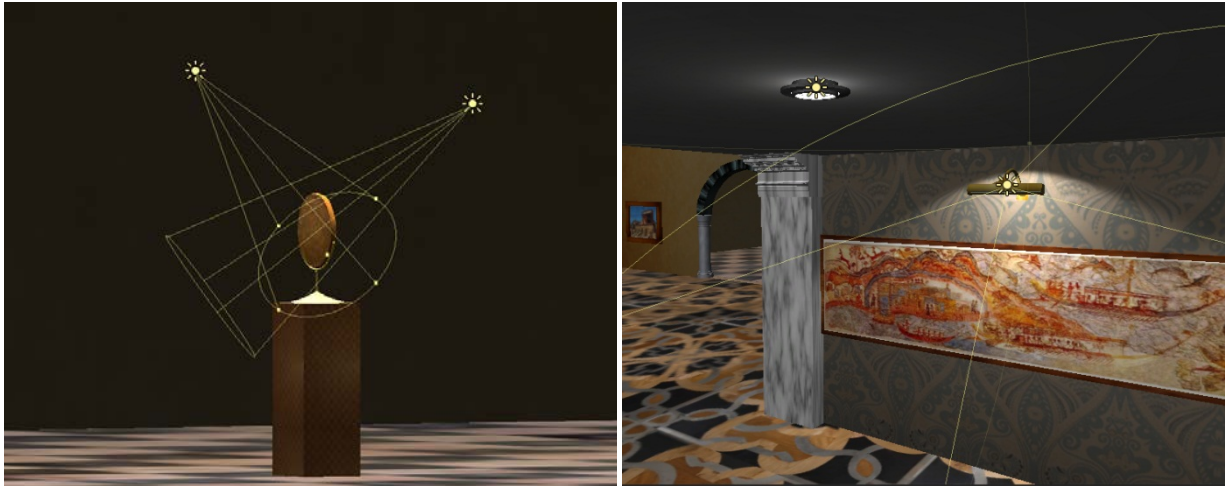
Για την απεικόνιση των πηγών φωτισμού στους χώρους του μουσείου αναπτύχθηκαν και χρησιμοποιήθηκαν μοντέλα, όπως λάμπες τοίχου, spot lights και λάμπες οροφής.



Εικόνα 3.23 Μοντέλα που δημιουργήθηκαν και χρησιμοποιήθηκαν για την απεικόνιση πηγών φωτισμού

Τα μοντέλα αυτά τοποθετήθηκαν στα διάφορα μέρη του μουσείου, ώστε να υπάρχει η ψευδαίσθηση ότι το φως παράγεται μέσα από τους λαμπτήρες που περιέχουν. Στην απόδοση φωτισμού στους χώρους του μουσείου χρησιμοποιήθηκαν δυο ειδών lights, SpotLights και PointLights.

Σε κάθε αίθουσα τοποθετήθηκε ένα τουλάχιστον στοιχείο φωτισμού, ενώ επιπλέον στοιχεία χρησιμοποιήθηκαν σε συγκεκριμένα εκθέματα, για την καλύτερη απόδοση φωτισμού σε αυτά.



Εικόνα 3.24 Δημιουργία φωτισμού σε εκθέματα

Στην οροφή της κεντρικής αίθουσας του μουσείου, απεικονίζεται ένας γυάλινος θόλος. Στη προσπάθεια να δημιουργηθεί η ψευδαίσθηση της εισαγωγής του ηλιακού φωτός μέσα στην αίθουσα, τοποθετήθηκαν lights με επιλεγμένα εφέ, ώστε να απεικονίζουν μια μορφή ήπιας λάμψης στο χώρο.



Εικόνα 3.25 Δημιουργία εφέ λάμψης του θόλου της κεντρικής αίθουσας

3.6.5 Απεικόνιση κίνησης και διαχείριση animation.

Για να μπορέσει το avatar που εισάγαμε στη σκηνή να διαγράψει μια μετακίνηση στο χώρο που να ελέγχεται από τον χρήστη, δημιουργήθηκε ένα αρχείο κώδικα που προσφέρει αυτή τη δυνατότητα. Πριν από αυτό όμως έπρεπε να εισαχθεί στο χαρακτήρα ένα CharacterController σύστημα από τα εργαλεία του Unity το οποίο επιτρέπει τον έλεγχο της κίνησης μέσα από κλασικά buttons του πληκτρολογίου που χρησιμοποιούνται σε πολλές τέτοιου είδους εφαρμογές (W/Front, S/Back, A/Left, D/Right). Επίσης, παρέχονται και άλλες παράμετροι μέσα από standard μεταβλητές που παρέχονται από έτοιμα αρχεία κώδικα.



Σχήμα 3.14 Παράμετροι του Character Controller

Το αρχείο κώδικα που συντάχθηκε χρησιμοποιεί στοιχεία του CharacterController κατά την διαδικασία εκτέλεσης, για την απόδοση της μετακίνησης του αντικειμένου στην σκηνή.

```

1 ReaController.js
var speed = 3.0;
var gravity = 20.0;

var smoothSpeed = 10.0;
var smoothDirection = 10.0;

private var moveDirection = Vector3.zero;
private var verticalSpeed = 0.0;
private var moveSpeed = 0.0;

private var grounded : boolean = false;
private var IsRunning :boolean = false;
private var targetAngle = 0.0;

@script RequireComponent(CharacterController)
// Η Συνάρτηση Awake αποτυπώνει το διάνυσμα της κίνησης προς τα εμπρός
function Awake ()
- {
    moveDirection = transform.TransformDirection(Vector3.forward);
}

```

```

//Η Συνάρτηση UpdateSmoothedMovementDirection χρησιμοποιείται για να
// "προσκολληθεί" η κίνηση με το avatar, ώστε να το ακολουθεί κατά την κίνηση.
function UpdateSmoothedMovementDirection ()
- {
    var cameraTransform = Camera.main.transform;

    var forward = cameraTransform.TransformDirection(Vector3.forward);
    forward.y = 0;
    forward = forward.normalized;

    var right = Vector3(forward.z, 0, -forward.x);

    var targetDirection = Input.GetAxis("Horizontal") * right + Input.GetAxis("Vertical") * forward;

    if (targetDirection != Vector3.zero)
    {
        moveDirection = Vector3.Lerp(moveDirection, targetDirection, smoothDirection * Time.deltaTime);
        moveDirection = moveDirection.normalized;
    }

    var curSmooth = smoothSpeed * Time.deltaTime;

    if (!grounded)
    {
        curSmooth *= 0.5;
    }

    if (Input.GetKeyDown("r"))
        IsRunning = !IsRunning;

    if (IsRunning == true)
        speed = 6.0;
    else
        speed = 3.0;

    moveSpeed = Mathf.Lerp(moveSpeed, targetDirection.magnitude * speed, curSmooth);
}

- function Update() {
    UpdateSmoothedMovementDirection();

    if (grounded) {
        verticalSpeed = 0.0;
    }
    // Απόδοση βαρύτητας
    verticalSpeed -= gravity * Time.deltaTime;

    var movement = moveDirection * moveSpeed + Vector3 (0, verticalSpeed, 0);
    movement *= Time.deltaTime;

    // Χρήση των στοιχείων κίνησης του CharacterController
    var controller : CharacterController = GetComponent(CharacterController);
    var flags = controller.Move(movement);
    grounded = (flags & CollisionFlags.CollidedBelow) != 0;

    transform.rotation = Quaternion.LookRotation(moveDirection);
}
// Η GetSpeed επιστρέφει την ταχύτητα της κίνησης
- function GetSpeed () {
    return moveSpeed;
}

//Η GetDirection επιστρέφει το διάνυσμα της κίνησης
- function GetDirection () {
    return moveDirection;
}

```

Εικόνα 3.26 Script για τον έλεγχο της κίνησης του χαρακτήρα

Στην ανάπτυξη του animation του χαρακτήρα κατά την διαδικασία της μοντελοποίησης, δημιουργήθηκαν τρεις καταστάσεις κίνησης. Η πρώτη, αφορά την κατάσταση αναμονής (wait) του χαρακτήρα, δηλαδή την κίνηση που κάνει ο χαρακτήρας ανά κάποια χρονικά διαστήματα, όταν ο χρήστης δεν χρησιμοποιεί το πληκτρολόγιο. Η δεύτερη, αφορά τον βηματισμό(walk) που εκτελείται κατά την κίνηση του χαρακτήρα προς τα εμπρός. Η τελευταία, αφορά την κατάσταση τρεξίματος (run) του χαρακτήρα με γρηγορότερο βηματισμό κατά την κίνηση.

Μέσω του Unity το animation που αποτελείτε από αυτές τις τρεις καταστάσεις, χωρίστηκε σε επιμέρους τμήματα. Με την χρήση κώδικα δόθηκε η δυνατότητα να εκτελείται το κάθε τμήμα του animation ξεχωριστά, ανάλογα με την κατάσταση στην οποία βρίσκεται το avatar σε συνδυασμό με τον έλεγχο από τον χρήστη.



```
1 ReaAnimation.js
var landBounceTime = 0.6;

function Start ()
- {
  //Ορίζει όλα τα animation σε loop mode
  animation.wrapMode = WrapMode.Loop;
}

function Update ()
- {
  // Καλεί στοιχεία απο το αρχείο ReaController
  var marioController : ReaController = GetComponent(ReaController);
  var currentSpeed = reaController.GetSpeed();

  // Αλλάζει την εκτέλεση του animation ανάλογα με την
  //κατάσταση του avatar από τον έλεγχο του χρήστη
  if (currentSpeed > 0.1)
  - {
    if (currentSpeed > 3)
      animation.CrossFade("run");
    else
      animation.CrossFade("walk");
  }
  else
    animation.CrossFade("wait");
}
}
```

Εικόνα 3.27 Script για την διαχείριση των animation του χαρακτήρα

3.6.7 Αλληλεπίδραση μεταξύ μοντέλων

3.6.7.1 Εμφάνιση περιγραφής εκθεμάτων

Στην εφαρμογή, εκτός από την περιήγηση του χαρακτήρα στο χώρο του μουσείου, δημιουργήθηκε μια διαδραστική δυνατότητα για την εμφάνιση της περιγραφής κάθε εκθέματος ξεχωριστά μόλις το avatar βρεθεί σε μικρή απόσταση από αυτά. Για αυτή τη προσέγγιση χρειάστηκε αρχικά να δημιουργηθεί μια εικόνα/texture στον ρολό ενός pop-out παράθυρου, που να εμφανίζεται στην κάμερα περιέχοντας όλες τις επιθυμητές πληροφορίες. Επίσης, το παράθυρο αυτό διαμορφώθηκε έτσι ώστε να περιέχει στο κέντρο του ένα mini movie clip, που να εμφανίζει το έκθεμα σαν περιστρεφόμενο αντικείμενο. Αυτό έγινε με την γρήγορη εναλλαγή μιας αλληλουχίας εικόνων διαφορετικής λήψης για κάθε αντικείμενο, ώστε να φαίνεται ότι δημιουργείται ένα mini 2d animation clip.



Εικόνα 3.28 Texture και ακολουθία εικόνων για την δημιουργία του παράθυρου περιγραφής των εκθεμάτων

Ο κώδικας που συντάχθηκε για αυτή τη λειτουργία εφαρμόστηκε σε όλα τα εκθέματα και αρχικά ελέγχει την απόσταση του avatar από ένα αντικείμενο. Αν έχει την κατάλληλη τιμή τότε εμφανίζει στην οθόνη της κάμερας ένα GUI button, “Examine”. Με την επιλογή του εμφανίζεται το texture, συμπεριλαμβανομένου του sequence animation και ενός “Close” button. Ο τερματισμός του παραθύρου μπορεί να γίνει από το “Close” button ή απομακρύνοντας τον χαρακτήρα από το αντικείμενο.

```

exhibits collider.js
// JavaScript
var other : Transform;
var closeDistance = 3.0;
var coll : boolean = false;
var coll2 : boolean = false;
var aTexture : Texture;
var MyGUIAlpha = 0.0;
var _typedTextures : Texture2D[] ;
var _displayed : int = 0;
var startCount : int;
var stopCount : int;

--function Update() {
// Ελέγχει την απόσταση του αναταρ από το αντικείμενο
if (other) {
var sqrLen = (other.position - transform.position).sqrMagnitude;
if( sqrLen < closeDistance*closeDistance )
coll=true;
else{
coll=false;
coll2=false;
_displayed=0;
_typedTextures=null;}
}

// Δημιουργεί έναν πίνακα από textures 32 θέσεων.
if (coll2==true) {
var i : int;
_typedTextures = new Texture2D[32];

// Γεμίζει τον πίνακα με τα κατάλληλα textures από τον φάκελο resources
for ( i = 0; i < _typedTextures.Length; i++){
_typedTextures[i]=Resources.Load((i+startCount).ToString());
}
}

//Δημιουργεί ένα μετρητή του πίνακα
if (_typedTextures != null && _typedTextures.Length>0){
_displayed++;

if (_displayed >= _typedTextures.Length){
_displayed = 0;
}
}
}

// Μόσα από την συνάρτηση OnGUI εμφανίζονται τα GUI Buttons και τα Textures στην κάμερα.
--function OnGUI () {
--if (coll==true){
-- if (GUI.Button (Rect ((Screen.width/2)+360,Screen.height/2,100,50), "examine")) {
coll2=true;
}
}
--if(coll2==true){
GUI.DrawTexture(Rect((Screen.width/2)+50,Screen.height/13,
(Screen.width/2)-160,Screen.height-280), aTexture);

// Εμφανίζει την αλληλουχία των εικόνων του animation clip ανάλογα με τον μετρητή του πίνακα
if (_typedTextures != null && _typedTextures.Length > 0){
GUI.DrawTexture(new Rect((Screen.width/2)+306,(Screen.height/13)+117,288, 203),
_typedTextures[_displayed], ScaleMode.StretchToFill, true);

if (GUI.Button (Rect ((Screen.width/2)+368,(Screen.height/13)+717,100,50), "Close")) {
coll2=false;
coll=false;
_displayed=0;
_typedTextures=null;
}
}
}
}
}

```

Εικόνα 3.29 Script για την εμφάνιση του παράθυρου περιγραφής των εκδεμάτων

3.6.7.2 Λειτουργία ανελκυστήρα

Η λειτουργία του ανελκυστήρα πραγματοποιείται με την χρήση *colliders* που προκαλούνται ανάμεσα σε αυτόν και στον χαρακτήρα. Στην προσέγγιση αυτή χρησιμοποιήθηκαν *Cubes* στον ρολό των *triggered* αντικειμένων που αλληλεπιδρούν με το *avatar*. Αρχικά τα *cubes* παραμετροποιήθηκαν ώστε να μην φαίνονται στη σκηνή. Δύο *cubes* τοποθετήθηκαν μπροστά στις εισόδους του ανελκυστήρα στο ισόγειο και στο πρώτο όροφο αντίστοιχα. Αυτό έχει σκοπό να δημιουργηθεί σύγκρουση (*collision*) και να ανοίξει αυτόματα η είσοδος του *elevator* καθώς το πλησιάζει ο χαρακτήρας. Με την απομάκρυνσή του από το σημείο, η είσοδος έχει προγραμματιστεί να κλίνει ξανά έπειτα από ορισμένο χρόνο. Ένα τρίτο *cube* τοποθετήθηκε μέσα στον ανελκυστήρα, ώστε με την σύγκρουση από τον χαρακτήρα που θα έχει εισέλθει μέσα στο θάλαμο, να αλλάξει σταδιακά το *Transform position* του θαλάμου προς τα επάνω ή κάτω ανάλογα με την αρχική του θέση. Για να μπορέσει ο χαρακτήρας να ακολουθήσει την αλλαγή θέσης του *elevator* πρέπει να μετατραπεί σε *child* του θαλάμου. Έπειτα, καθώς αποχωρεί από το ασανσέρ θα χρειαστεί να αναιρεθεί η προηγούμενη μετατροπή και να επανέλθει στην προτέρα κατάσταση.



Εικόνα 3.30 Χρήση *Cubes* στις εισόδους και το εσωτερικό του *elevator* για τον έλεγχο λειτουργίας

```

1 elevator_UpDown.js *
static var eleswitch : int;
var trigger : int=1;
var Player: Transform;

var LDoor : Transform;
var RDoor : Transform;
var drspeed : int = 2;
var to1 : Vector3;
var to2 : Vector3;
var from1 : Vector3;
var from2 : Vector3;

var eleclosed : boolean = true;
var eleopen : boolean = false;
var ele : boolean=false;

//Η μέθοδος OnTriggerStay εκτελείται όσο διαρκεί το collision μεταξύ του avatar και του Cube.
// αναρεί την μετατροπή του elevator σε parent του χαρακτήρα
- function OnTriggerStay( other : Collider) {

    eleswitch= trigger;
    Player.parent = null;
    ele=true;

}

// Ελέγχει αν οι πόρτες του elevator είναι ανοικτές ή κλειστές και της μετακινεί ανάλογα με την
// κατάσταση την οποία βρίσκονται
- function Update() {

- if (ele){

- if(eleclosed){
    if(LDoor.transform.position.z <to1.z)
        LDoor.transform.Translate(Vector3.up*Time.deltaTime*drspeed);
    if(RDoor.transform.position.z >to2.z)
        RDoor.transform.Translate(-Vector3.up*Time.deltaTime*drspeed);
    if((LDoor.transform.position.z>to1.z && RDoor.transform.position.z<to2.z) ||
        (LDoor.transform.position.z==to1.z && RDoor.transform.position.z==to2.z))
        check();
    }

- if(eleopen){
    if(LDoor.transform.position.z >from1.z || LDoor.transform.position.z ==from1.z )
        LDoor.transform.Translate(-Vector3.up*Time.deltaTime*drspeed);
    if(RDoor.transform.position.z <from2.z || RDoor.transform.position.z ==from2.z )
        RDoor.transform.Translate(Vector3.up*Time.deltaTime*drspeed);
    if((LDoor.transform.position.z<from1.z && RDoor.transform.position.z>from2.z) ||
        (LDoor.transform.position.z==from1.z && RDoor.transform.position.z==from2.z)){
        uncheck();
        ele=false;
    }
    }
}

// Παρπαίνει το κλείσιμο της εισόδου κατά 4 sec.
- function check(){
    eleclosed= false;
    yield WaitForSeconds(4);
    eleopen=true;

}

// Παρπαίνει το άνοιγμα της εισόδου κατά 4 sec
- function uncheck(){
    eleopen=false;
    yield WaitForSeconds(4);
    eleclosed=true;

}
}

```

Εικόνα 3.31 Script 1 για τον έλεγχο και λειτουργία του elevator


```

1 elevatorUP.js *
var Player : Transform;
var Elevator : Transform;

var to : Vector3;
var from : Vector3;
var elspeed : int = 5;
var boolswitch : elevator_UpDown;
var pass : boolean = false;
var tin : int;

//Η συνάρτηση OnTriggerStay εκτελείται όσο δημιουργείται collision μεταξύ του Cube και του αναταρ
// Ελέγχει την μεταβλητή από το αρχείο elevator_UpDown που δείχνει την κατάσταση του eleswitch
//Ανάλογα μετακινεί τον θάλαμο στο αντίστοιχο επίπεδο
- function OnTriggerStay(other : Collider) {
boolswitch = FindObjectOfType(elevator_UpDown);
-   if(boolswitch.eleswitch == 1){
tin = 1;
-   if(pass == true)
-       if(Elevator.transform.position.y < from.y){
print(boolswitch.eleswitch);
Player.parent = Elevator;
Elevator.transform.Translate(Vector3.up*Time.deltaTime*elspeed);
-       }
-   }
- }

- function Update(){
if(tin == 1)
foo();
else
pass = false;

disable();
}
//Απενεργοποιεί την λειτουργία του elevator μετά από 6 sec
//( Ο χρόνος που χρειάζεται για να αλλάξει θέση)
- function foo(){
yield WaitForSeconds(6);
pass = true;
}
//Αναβάλει την λειτουργία του elevator για 5 sec.( Μέχρι το κλείσιμο της εισόδου)
- function disable(){
- if(tin == 1 && pass == true){
yield WaitForSeconds(5);
tin = 0;
pass = false;
}
}
}

```

Εικόνα 3.32 Script 2 για τον έλεγχο και λειτουργία του elevator

```

1 elevatorDOWN.js *
var Player : Transform;
var Elevator : Transform;

var to : Vector3;
var from : Vector3;
var elspeed : int = 5;
var boolswitch : elevator_UpDown;
var pass2 : boolean = false;
var tin2 : int;

//Η συνάρτηση OnTriggerStay εκτελείται όσο δημιουργείται collision μεταξύ του Cube και του ανατ
// Ελέγχει την μεταβλητή από το αρχείο elevator_UpDown που δείχνει την κατάσταση του eleswit
//Ανάλογα μετακινεί τον θάλαμο στο αντίστοιχο επίπεδο.
- function OnTriggerStay(other : Collider) {
    boolswitch= FindObjectOfType(elevator_UpDown);

    if(boolswitch.eleswitch == 2){
        tin2=2;
        if(pass2==true)
        if(Elevator.transform.position.y > to.y){
            print(boolswitch.eleswitch);
            Player.parent = Elevator;
            Elevator.transform.Translate(-Vector3.up*Time.deltaTime*elspeed);
        }
    }
}

- function Update(){
    if(tin2==2)
        foo();
    else
        pass2=false;
        disable();
}
//Απενεργοποιεί την λειτουργία του elevator μετά από 6 sec
//( Ο χρόνος που χρειάζεται για να αλλάξει θέση)
- function foo(){
    yield WaitForSeconds(6);
    pass2=true;
}
//Αναβάλλει την λειτουργία του elevator για 5 sec.( Μέχρι το κλείσιμο της εισόδου)
- function disable(){
- if(tin2==2 && pass2 == true){
    yield WaitForSeconds(5);
    tin2=0;
    pass2=false;
}
}
}

```

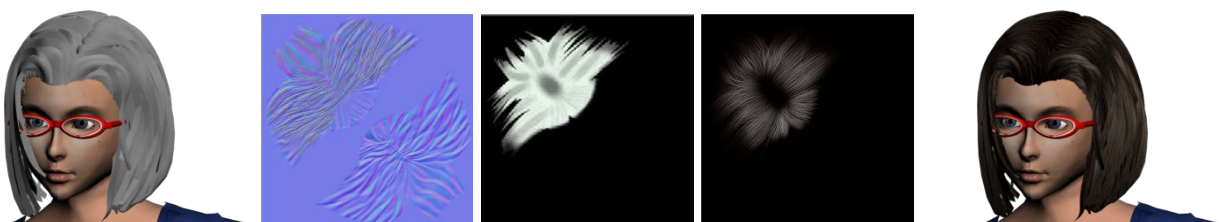
Εικόνα 3.33 Script 3 για τον έλεγχο και λειτουργία του elevator

3.7 Tips

Ορισμένες φορές, η απόδοση περισσότερης ρεαλιστικότητας και η ανάγκη εξοικονόμησης πόρων του συστήματος σε μια 3d εφαρμογή, δημιουργούν ζητήματα που πρέπει να αντιμετωπισθούν με τον καλλίτερο δυνατό τρόπο. Μια ρεαλιστική σκηνή χρειάζεται αρκετή λεπτομέρεια σε επίπεδο περιβάλλοντος και μοντέλων, κατάλληλο φωτισμό και σε συνδυασμό με την χρήση περίπλοκων και μεγάλων αρχείων κώδικα μπορούν να δημιουργήσουν καθυστέρηση κατά την διάρκεια της εκτέλεσης της. Γιαυτό δίνονται παρακάτω ορισμένες συμβουλές για να αποφευχθούν τέτοιου είδους καταστάσεις και συνάμα να δημιουργείται το καλλίτερο δυνατό αποτέλεσμα.

3.7.1 Χρήση Low Poly μοντέλων

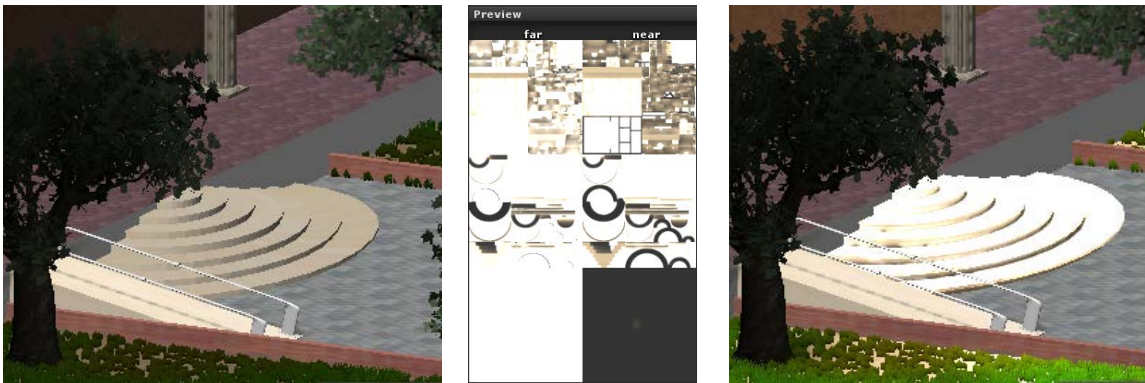
Τα μοντέλα που περιέχουν έναν μεγάλο αριθμό πολυγώνων χρειάζονται περισσότερο χρόνο για να γίνουν render από έναν επεξεργαστή. Η χρήση μοντέλων που αποτελούνται από λιγότερα πολύγωνα (low poly models) είναι μια καλή λύση, αλλά μειώνεται απόδοση στην απεικόνιση λεπτομέρειας. Για να μπορέσει να αποδοθεί η λεπτομέρεια που χρειάζεται σε ένα low poly model, χρειάζεται να εφαρμοστούν πάνω του υφές με την χρήση χαρτών(maps) που να αποτυπώνουν πάνω στο μοντέλο στοιχεία όπως σκιάσεις, λάμπσεις, εσοχές , προεξοχές, χρωματικές διαβαθμίσεις κτλ. που στην πραγματικότητα δεν υπάρχουν, αλλά δίνεται η ψευδαίσθηση ότι αποτελούν μέρος του μοντέλου. Κάποια από τα maps που παρέχουν τέτοιου είδους δυνατότητες είναι τα Normal maps, τα Alpha maps και τα Bump maps.



Εικόνα 3.34 Χρήση Normal, Alpha και Bump στην απεικόνιση τών μαλλιών του avatar

3.7.2 Χρήση Light Maps

Η δημιουργία Lightmap χαρτών χρησιμοποιούνται στην απεικόνιση των σκιών των μοντέλων σε σχέση με τις φωτεινές πηγές που αλληλεπιδρούν με αυτό. Επειδή ο χρόνος που απαιτείτε για να γίνει render μια 3d εικόνα σχετίζεται με την φωτεινότητα και τις σκιές που εκπέμπονται σε μια σκηνή, η χρήση light maps βελτιώνει κατά πολύ αυτόν τον χρόνο. Ένας τέτοιος χάρτης στην ουσία αποτυπώνει την σκιά και την φωτεινότητα του αντικειμένου στην σκηνή σαν να είναι μέρος την υφής που έχει εφαρμοστεί. Έτσι σε μια τρισδιάστατη εφαρμογή υπάρχει η δυνατότητα να αναπτυχθεί ρεαλιστικός φωτισμός, χωρίς να επιβαρύνεται ο επεξεργαστής με επιπλέον πληροφορίες και διεργασίες.



Εικόνα 3.35 Χρήση LightMap για την απόδοση φωτεινότητας

3.7.3 Εξοικονόμηση όγκου δεδομένων από την χρήση textures.

Σε πολλές περιπτώσεις χρειάζεται να χρησιμοποιηθούν textures για τον χρωματισμό κάποιων αντικειμένων σε μια τρισδιάστατη εφαρμογή. Η χρήση μιας εικόνας μεγάλης ανάλυσης για την προσαρμογή υφής μπορεί να επιβαρύνει το σύστημα με περιττά δεδομένα. Για παράδειγμα ένα αντικείμενο μπορεί να έχει ένα ικανοποιητικό αποτέλεσμα με την χρήση υφής μεγάλου όγκου δεδομένων, αλλά θα μπορούσε να έχει εξίσου ικανοποιητικά αποτελέσματα με την χρήση υφής πολύ μικρότερου όγκου δεδομένων. Γενικότερα μοντέλα τα οποία υπάρχουν σε ένα τρισδιάστατο χώρο πρέπει να αξιολογούνται ανάλογα με την συχνότητα, το μέγεθος και την σημαντικότητα εμφάνισης τους στην σκηνή, ώστε να προσαρμόζονται πάνω τους ανάλογες υφές που θα απεικονίζουν μεγαλύτερη ή μικρότερη λεπτομέρεια.

Επίλογος

Για την παρούσα εργασία χρειάστηκαν επτά μήνες εντατικής και πολύωρης ενασχόλησης και έρευνας για την διεκπεραίωσή της. Μέσα από τις δυνατότητες που παρέχει το διαδίκτυο, συλλέκτικαν στοιχεία όπως εικόνες και πληροφορίες που βοήθησαν στην ανάπτυξη και παρουσίαση των εκθεμάτων. Για την ορθή χρήση των προγραμμάτων που συνέβαλαν στην υλοποίηση αυτής της εφαρμογής, δημιουργήθηκε η ανάγκη μελέτης των τεχνικών και των δυνατοτήτων που προσφέρονται από τα προγράμματα αυτά, εστί ώστε να εφαρμοστούν με τον καλύτερο δυνατό τρόπο και να αποδώσουν το παρόν αποτέλεσμα.

Το "Minoan Age exhibits" θα μπορούσε να απευθύνεται σε άτομα που επιθυμούν να γνωρίσουν ή να μάθουν περισσότερα σχετικά με την ιστορία και την κουλτούρα των αρχαίων Κρητών. Μέσα από ένα περιβάλλον γεμάτο χρώματα και αντικείμενα που προκαλούν το ενδιαφέρον του χρήστη, προσφέρεται η δυνατότητα επιλογών μέσου της διαδραστικότητας, όπου μπορεί κάποιος με διασκεδαστικό τρόπο να εμπλουτίσει τις γνώσεις του.

Με τις κατάλληλες τροποποιήσεις και προεκτάσεις, αυτή η εφαρμογή θα μπορούσε να αποκτήσει ψυχαγωγικό χαρακτήρα παραπέμποντας περισσότερο σε 3D παιχνίδι προσφέροντας παράλληλα γνώσεις που αφορούν τον μινωικό ή και άλλους πολιτισμούς.

Βιβλιογραφία

<http://usa.autodesk.com/3ds-max/>

<http://www.youtube.com/user/3dsMaxHowTos>

<http://www.cgarena.com/freestuff/tutorials/tutorials.html>

[http://www.videocopilot.net/tutorials/3d text in 3ds max/](http://www.videocopilot.net/tutorials/3d_text_in_3ds_max/)

<http://cg.tutsplus.com/articles/top-15-applications-for-3d-artists/>

<http://unity3d.com/>

<http://docs.unity3d.com/Documentation>

Copyright 2012

Writer email: kmpolonis@gmail.com