Τμήμα Τεχνών Ήχου και Εικόνας, Ιόνιο Πανεπιστήμιο

Εικονική Περιήγηση στην Ιόνιο Ακαδημία

Αθανάσιος Γκορίτσας

ADA201502

Επιβλέπων: Κωνσταντίνος Τηλιγάδης

Συνεπιβλέπων: Μιχάλης Παναγόπουλος

Κέρκυρα, Οκτώβρης 2018

Τμήμα Τεχνών Ήχου και Εικόνας, Ιόνιο Πανεπιστήμιο

Εικονική Περιήγηση στην Ιόνιο Ακαδημία

Αθανάσιος Γκορίτσας

ADA201502

Επιβλέπων: Κωνσταντίνος Τηλιγάδης

Συνεπιβλέπων: Μιχάλης Παναγόπουλος

Κέρκυρα, Οκτώβρης 2018

Συνοπτική Περίληψη

Το πρώτο κεφάλαιο της έρευνας είναι η εισαγωγή στο αντικείμενο της έρευνας. Γίνεται σύντομη αναφορά στο περιεχόμενο των κεφαλαίων και εξηγούνται βασικοί όροι της τρισδιάστατης μοντελοποίησης και της ανάπτυξης παιχνιδιών.

Στο πρώτο μέρος της έρευνας (Κεφάλαια 2 και 3) γίνεται μια ιστορική αναδρομή με επίκεντρο ενδιαφέροντος την εκπαίδευση στην Κέρκυρα. Καλύπτεται η χρονική περίοδος από το 1386 ως το 1864. Η Ιόνιος Ακαδημία έκλεισε το 1865, για να επαναλειτουργήσει το 1984.

Στο δεύτερο μέρος (Κεφάλαια 4 ως 6) αναλύεται η δημιουργία ενός τρισδιάστατου μοντέλου της Ιονίου Ακαδημίας, και η ανάπτυξη μιας διαδραστικής εκπαιδευτικής εφαρμογής που υλοποιείται σε δύο βήματα.

Στο κεφάλαιο 5 αναλύεται το πρώτο βήμα που είναι η δημιουργία ενός τρισδιάστατου μοντέλου του κτιρίου της Ιονίου Ακαδημίας με χρήση του προγράμματος Blender.

Στο κεφάλαιο 6 αναλύεται το δεύτερο βήμα, η εισαγωγή και η παραμετροποίηση του μοντέλου στην μηχανή παιχνιδιών Unity.

Τέλος, παρουσιάζονται συμπεράσματα και προτείνεται περαιτέρω ανάπτυξη της έρευνας.

Περίληψη

Στο πρώτο κέφαλαιο γίνεται η εισαγωγή στο αντικείμενο της έρευνας που αφορά την Ιόνιο Ακαδημία. Στόχος είναι η δημιουργία ενός τρισδιάστατου μοντέλου του κτιρίου, και στην συνέχεια η εισαγωγή του σε μια μηχανή παιχνιδιών για την δημιουργία ενός διαδραστικού περιβάλλοντος για τον χρήστη. Έτσι, θα δημιουργηθεί μια εκπαιδευτικής φύσης εφαρμογή που θα προσφέρει στον ενδιαφερόμενο μια γνωριμία με τους χώρους του κτιρίου, αλλά και την ιστορία της Ιονίου Ακαδημίας. Με τον τρόπο αυτό, τεκμηριώνεται η δυνατότητα δημιουργίας εκπαιδευτικού υλικού με την χρήση τεχνολογιών μοντελοποίησης, αλλά και των τεχνολογιών που προσφέρουν οι μηχανές παιχνιδιών.

Στο δεύτερο κεφάλαιο γίνεται μια ιστορική αναδρομή για το κτίριο της Ιονίου Ακαδημίας. Με αφορμή την περίοδο της Ενετοκρατίας κατά την οποία κατασκευάστηκε για να εξυπηρετεί ως στρατώνας, γίνεται μια παρουσίαση της κατάστασης της εκπαίδευσης κατά την περίοδο 1386-1797. Στην συνέχεια, η αναδρομή συνεχίζεται με την σύντομη παρουσία των Δημοκρατικών Γάλλων του Ναπολέοντα από το 1797 ως το 1799, και τις δικές τους προσπάθειες να οργανώσουν την εκπαίδευση στο Νησί της Κέρκυρας. Προσπάθειες που διακόπηκαν απο την Πολιορκία και κατάληψη του νησιού απο τους Ρωσότουρκους. Με την ρευστή πολιτική κατάσταση της εποχής, τα Επτάνησα παραχωρήθηκαν απο τους Ρωσότουρκους στον Ναπολέοντα το 1807. Για να καταλήξουν το 1814 στους Άγγλους, που δεν έδειξαν ιδιαίτερο ενδιαφέρον για την εκπαίδευση στην Κέρκυρα.

Το τρίτο κεφάλαιο εξετάζει τις ενέργειες του Φρέντερικ Νορθ, κόμη του Γκίλφορντ, και το όραμά του να ιδρύσει Πανεπιστήμιο στα Επτάνησα. Οι αρχικές σκέψεις για την έδρα του πανεπιστημίου δεν ευνοούσαν την Κέρκυρα. Όμως η εξέγερση των Ελλήνων, οδήγησε τελικά το 1824 στην ίδρυση του Πρώτου Πανεπιστημίου στην Ελλάδα σε αυτήν. Ο Γκίλφορντ χρηματοδοτούσε τους πρώτους διδάσκοντες στην Ιόνιο Ακαδημία, αλλά και πολλούς σπουδαστές της με προσωπικά του έξοδα. Ο θάνατός του το 1827 ήταν πλήγμα για το Πανεπιστήμιο. Μετά απο αυτό το Πανεπιστήμιο άρχισε να παρακμάζει. Τα Επτάνησα παραχωρήθηκαν στο νεοσυσταθέν Ελληνικό κράτος το 1864. Η Ιόνιος Ακαδημία σταμάτησε να λειτουργεί το 1865. Για να επανιδρυθεί σαν Ιόνιο Πανεπιστήμιο το 1984.

Στο τέταρτο κεφάλαιο εξηγείται η μεθοδολογία της υλοποίησης της έρευνας. Γίνεται μια σύντομη παρουσίαση των βασικών εργαλείων. Στον τομέα της δημιουργίας των

τρισδιάστατων μοντέλων, χρησιμοποιήθηκε σαν εργαλείο το πρόγραμμα Blender 2.79b. Η ιδιαιτερότητά που το ξεχωρίζει από τα υπόλοιπα προγράμματα του είδους, είναι ότι είναι ανοιχτό λογισμικού και διατίθεται δωρεάν. Ο πηγαίος κώδικας είναι διαθέσιμος, για μελέτη των αλγορίθμων του ή ακόμα και αλλαγές για όποιον το επιθυμεί. Η μηχανή παιχνιδιών που χρησιμοποιήθηκε είναι το Unity. Χρησιμοποιήθηκαν πολλές εκδόσεις, το τελικό αποτέλεσμα υλοποιήθηκε με την έκδοση 2018.2.3f1. Το Unity δεν είναι βέβαια ανοιχτό λογισμικό, αλλά διατίθεται δωρεάν για προσωπική χρήση.

Στο πέμπτο κεφάλαιο αναλύεται η διαδικασία δημιουργίας του τρισδιάστατου μοντέλου του κτιρίου της Ιονίου Ακαδημίας στο πρόγραμμα Blender. Εξηγούνται οι βασικές αρχές που διέπουν την δημιουργία μοντέλων. Γίνεται σύντομη αναφορά στις δύο βασικές τεχνικές μοντελοποίησης και αναλύονται οι έννοιες των vertices, των edges και των faces και πως επηρεάζουν το μοντέλο που δημιουργείται. Παρουσιάζονται τα materials και τα textures. Γίνεται επεξήγηση των normals και της σημασίας τους για το τρισδιάστατο αντικείμενο που σχηματίζεται. Επίσης, γίνεται αναφορά στο shading και στην αποτελεσματική χρήση των normal maps. Το υπόλοιπο μέρος του κεφαλαίου αναλύει την δημιουργία των συστατικών στοιχείων, την χρήση της τεχνικής της φωτογραμμομετρίας και την εξαγωγή σε μορφή .fbx για να προετοιμαστεί το μοντέλο για εισαγωγή στο Unity.

Στο έκτο κεφάλαιο αναλύεται η διαδικασία της δημιουργίας μιας διαδραστικής εφαρμογής με την χρήση του Unity. Παρουσιάζονται ορισμένες βασικές και χρήσιμες αρχές του Unity. Εξηγείται το σύστημα συντεταγμένων, η έννοια των Components, ή έννοια των Colliders, η έννοια των Rigidbodies και η τεχνική του Raycasting που χρησιμοποιήθηκε εκτεταμένα απο την εφαρμογή για την ανίχνευση συμβάντων. Εξηγείται πως εισάγονται στο Unity τρισδιάστατα μοντέλα και πως συνδέονται τα materials και τα textures με ένα αντικείμενο. Δημιουργήθηκαν τα κατάλληλα script που προσαρτήθηκαν στα αντικείμενα που ελέγχονται απο αυτά και αναλύεται η λογική προσέγγιση για την λύση των προβλημάτων που προκύπταν κατά την φάση της υλοποίησης.

Τέλος, παρουσιάζονται τα συμπεράσματα που προκύψαν απο την όλη διαδικασία της ανάπτυξης της έρευνας και γίνονται προτάσεις για μελλοντικές βελτιώσεις.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1. Εισαγωγή 1

1.1. Ορισμοί 3

1.1.1. 3D Modeling - Blender 3

1.1.2. Game Engines - Unity 5

ΚΕΦΑΛΑΙΟ 2. Ιστορία 8

2.1. Η Περίοδος της Ενετοκρατίας 8

2.2. Η Περίοδος της Γαλλικής Κυριαρχίας 11

2.3. Η Περίοδος της Αγγλικής Κυριαρχίας 12

ΚΕΦΑΛΑΙΟ 3. Ακαδημία 16

3.1. Ιόνιος Ακαδημία 17

3.1.1. Η Έδρα της Ιονίου Ακαδημίας 20

3.1.2. Η μετά Γκίλφορντ Περίοδος (1827-1841) 24

3.1.3. Η Περίοδος 1841-1865 26

3.1.4. Τέλος μιας Εποχής: Το Κλείσιμο της Ιονίου Ακαδημίας το 1864. 28

3.1.5. Ο Βομβαρδισμός της Ιονίου Ακαδημίας και η Ανακαίνισή της 29

3.1.6. Η Ίδρυση του Ιονίου Πανεπιστημίου στην σύγχρονη εποχή 30

ΚΕΦΑΛΑΙΟ 4. Η Υλοποίηση 32

ΚΕΦΑΛΑΙΟ 5. 3D Μοντελοποίηση - Blender 34

5.1. Βασικές αρχές 34

5.1.1. Τεχνικές Μοντελοποίησης 34

5.1.2. Vertices - Edges - Faces. 35

5.1.3. Αριθμός vertices 36

5.1.4. Materials, Textures και Shaders 37

5.1.5. Normals 37

5.1.6. Normal maps 39
5.2. Η Διαδικασία της Δημιουργίας 42
5.2.1. Η Δημιουργία του κτιρίου από τα σχέδια της Ιονίου Ακαδημίας 42
5.2.2. Οι Πόρτες και τα Παράθυρα 43
5.2.3. Φωτογραμμομετρία 44
5.2.4. Το έμβλημα του Ιονίου Πανεπιστημίου 46
5.2.5. Τα έπιπλα 46
5.2.6. Εξαγωγή σε fbx 46
ΚΕΦΑΛΑΙΟ 6. Unity 47
6.1. Βασικές Αρχές 47
6.1.1. Συστήματα συντεταγμένων 47
6.1.2. Components 49
6.1.3. Colliders 50
6.1.4. Rigidbodies 51
6.1.5. Raycasting 52
6.2. Η διαδικασία της Δημιουργίας 53
6.2.1. Εισαγωγή των assets από τα αρχεία fbx 53
6.2.2. Materials και Textures 56
6.2.3. C#: Ενώνοντας τα επιμέρους για να συμβαίνει κάτι ενδιαφέρον 58
6.2.4. FPS Controller script 60
6.2.5. InteractiveObject script 61
6.2.6. Content.cs 63
6.2.7. Μπετόβεν 64
6.2.8. Το ασανσέρ 65
6.2.9. Animators 69
6.2.10. Το μενού Έναρξης και το μενού Παύσης. 70

6.2.11. Το παράθυρο διαλόγου. 72
6.2.12. WebGL 74
Συμπεράσματα 77
Προτάσεις για επιπλέον Ανάπτυξη 80
ΒΙΒΛΙΟΓΡΑΦΙΑ - ΙΣΤΟΓΡΑΦΙΑ 81
ΠΑΡΑΡΤΗΜΑ 85
Εξωτερικές πηγές 85
WebGL – Ρυθμίσεις Browser 86
Κώδικας 87
ChangeVolume.cs 87
Content.cs 87
DialogBox.cs 90
Elevator.cs 90
ElevatorController.cs 91
ElevatorExternalDoors.cs 97
ExternalElevatorDoorsSound.cs 97
ExternalLevelDisplay.cs 98
InteractiveObject.cs 98
Interactor.cs 102
LevelDisplay.cs 105
MainMenu.cs 106
PauseMenu.cs 106
PlayMusic.cs 107
RecticleController.cs 108

Εικόνες

Εικόνα 1-1. Φωτογραφία της Ιονίου Ακαδημίας 2

Εικόνα 1-2. Το απλούστερο node. 4

Εικόνα 1-3. Σύνθετο node για ένα αντικείμενο. 5

Εικόνα 1-4. Add Component στον Inspector 6

Εικόνα 2-1. Ο Φτερωτός Λέων της Βενετίας (Ι.Α. Αγιούς 2005, σ. 61). 8

Εικόνα 2-2. Το παλάτι του Γενικού Προνοητή Θαλάσσης (Ι. Α. Αγιούς 2005, σ. 138). 12

Εικόνα 2-3. Η φυγή από την Πάργα, Πίνακας του Διονύσιου Τσόκου (Ι. Α. Αγιούς 2005, σ. 183). 13

Εικόνα 3-1. Η Ιόνιος Ακαδημία (1841-1865). 26

Εικόνα 3-2 Η Ιόνιος Ακαδημία το 1980, πριν την ανακαίνισή της 30

Εικόνα 5-1. Box Modeling 35

Εικόνα 5-2. Mesh Modeling 35

Εικόνα 5-3. 1 vertex, 1 edge, 1 face επιλεγμένα (ο κύκλος δείχνει τον γεωμετρικό τόπο για το καθένα) 36

Εικόνα 5-4. Αριστερά 3182 vertices (μετασχηματισμός subdivide x 2), δεξιά 674 vertices. 37

Εικόνα 5-5. Διαφορετικά normals για δύο faces. 38

Εικόνα 5-6. Flat Shading σε σύγκριση με smooth shading 39

.Εικόνα 5-7. Προσθήκη bump map σε μια επιφάνεια 40

Εικόνα 5-8. Προσθήκη normal map σε μια επιφάνεια. 40

Εικόνα 5-9. Το αποτέλεσμα της εφαρμογής 5 διαφορετικών maps σε έναν κύκλο. 41

Εικόνα 5-10. Πάνω: Κάτοψη, Δεξιά Όψη. Κάτω: Πρόσοψη, Τρισδιάστατη Όψη 42

Εικόνα 5-11. Το κυρίως κτίριο στην τελική του μορφή 43

Εικόνα 5-12. Το δεξί παραθυρόφυλλο είναι γονιός για το πόμολο 44

Εικόνα5-13. Η προτομή του Καποδίστρια ανασυστάθηκε από μια σειρά 78 φωτογραφιών 45

Εικόνα5-14. Ο Λέων του Αγίου Μάρκου 45

Εικόνα 6-1. Το Σύστημα Συντεταγμένων του Unity 47

Εικόνα 6-2. Συντεταγμένες για γονιό (α), παιδί (β), μετακίνηση στον Z στον κόσμο (γ) και τοπικά (δ) 48

Εικόνα 6-3. Βασικά Components ενός αντικειμένου και οι σχέσεις τους 50

Εικόνα 6-4. Τα assets όπως έχουν εισαχθεί στο Unity 54

Εικόνα 6-5. Δημιουργία Colliders για το asset all 55

Εικόνα 6-6. Εισαγωγή materials 56

Εικόνα 6-7. Material με τις ιδιότητες του 57

Εικόνα 6-8. UV Unwrapping του λογότυπου της Ιονίου Ακαδημίας στο Blender 58

Εικόνα 6-9. Interactor Script προσαρτημένο στον FPS Controller 60

Εικόνα 6-10. Οι τέσσερις δυνατές καταστάσεις του δείκτη. 60

Εικόνα 6-11. Το interactiveObject script όπως είναι προσαρτημένο σε ένα πόμολο πόρτας 62

Εικόνα 6-12. Στο παράθυρο διαλόγου έχει προσαρτηθεί το script content.cs 63

Εικόνα 6-13. Τα περιεχόμενα του καταλόγου Resources 64

Εικόνα 6-14. Animation Controller και Animation για τις εξωτερικές πόρτες του ασανσέρ στο ισόγειο. 70

Εικόνα 6-15. Το PlayButton δείχνει κείμενο "Έναρξη" και όταν πατηθεί εκτελείται το script MainMenu.PlayGame 71

Εικόνα 6-16. Το ResumeButton δείχνει κείμενο "Συνέχεια" και εκτελεί το script PauseMenu.Resume 72

Εικόνα 6-17. Το DialogPanel 73

Εικόνα 6-18. Στοίχιση για το ScrollBar. Το ίδιο έγινε και για τα υπόλοιπα συστατικά. 73

Εικόνα 6-19. Επιλογή της μορφής του εκτελέσιμου 75

Κώδικας

Κώδικας 6-1 . Μέρος του κώδικα διαχείρισης των μετακινήσεων και περιστροφών με χρήση του iTween 49

Κώδικας 6-2. Μέρος του κώδικα που ελέγχει το Raycasting της κάμερας του FPS Controller με αντικείμενα..... 52

Κώδικας 6-3. Το Keyword SerializeField εκθέτει μιά μεταβλητή στον Inspector. 59

Κώδικας 6-4. Μέρος του κώδικα του script content.cs..... 64

Κώδικας 6-5. Το script PlayMusic 65

Κώδικας 6-6. Το script ElevatorExternalDoors.cs 66

Κώδικας 6-7. Το script ExternalElevatorDoorsSound.cs..... 66

Κώδικας 6-8. Μέρος του script ElevatorController.cs που ελέγχει την ένδειξη εμφανίζει το ασανσέρ..... 67

Κώδικας 6-9. Μέρος του Elevator.cs που καλεί το ElevatorController.cs με τις κατάλληλες τιμές..... 68

Κώδικας 6-10. IEnumerator 69

Κώδικας 6-11. Το script MainMenu. 71

ΚΕΦΑΛΑΙΟ 1. Εισαγωγή

Το κτίριο που είναι γνωστό σαν Ιόνιος Ακαδημία (εικόνα 1-1), χτίστηκε την εποχή της Ενετοκρατίας απο τους Γενικούς Προνοητές Grimani και Correr τον 16ο αιώνα. Βρίσκεται στην Σπιανάδα στην συμβολή των οδών Καποδιστρίου και Ακαδημίας και αρχικός σκοπός του ήταν να εξυπηρετεί τους Ενετούς σαν στρατώνας.

Η Ιόνιος Ακαδημία ιδρύθηκε επίσημα το 1824. Ιδρύθηκε όταν τα Ιόνια Νησιά ήταν υπό την αρμοστεία των Άγγλων, που αντιμετώπισαν την λειτουργία της Ιονίου Ακαδημίας με ανοχή μάλλον παρά με περηφάνια, με ελάχιστες φωτεινές εξαιρέσεις (Henderson, 1980, πρόλογος). Με την ένωση των Επτανήσων με την Ελλάδα το 1864, η ευθύνη της λειτουργίας της Ιονίου Ακαδημίας περιήλθε στο νεοσύστατο Ελληνικό Κράτος. Το οποίο το 1865 αποφάσισε την παύση λειτουργίας της (Κουρκουμέλης, Η εκπαίδευση στην Κέρκυρα κατά τη διάρκεια της Βρετανικής Προστασίας (1816-1864) 2002).

Μετά από αυτό, φιλοξενούσε την Δημόσια Βιβλιοθήκη της Κέρκυρας και πρόσφερε στέγη και σε διάφορα εκπαιδευτικά και πολιτιστικά σωματεία.

Αυτά όλα τελείωσαν με τον βομβαρδισμό της Ιονίου Ακαδημίας στις 14 Σεπτέμβρη 1943 από τους Γερμανούς. Από τότε και μέχρι το 1994 όταν ανακαινίστηκε, το κτίριο έμεινε ερειπωμένο.

Αντικείμενο αυτής της έρευνας είναι η ψηφιοποίηση της Ιονίου Ακαδημίας, με χρήση τεχνολογιών τρισδιάστατης μοντελοποίησης. Το μοντέλο εμψυχώνεται σε μια μηχανή παιχνιδιών. Ως τελικό αποτέλεσμα, θα επιτρέπεται η διάδραση με τον χρήστη.

Το πεδίο εφαρμογής της έρευνας είναι ιστορικό και εκπαιδευτικό. Γίνεται μια γνωριμία με την Ιστορία της Κέρκυρας που εστιάζει στην εκπαίδευση από τον 14 αιώνα ως σήμερα. Η Ιόνιος Ακαδημία είναι το σημείο ενδιαφέροντος για την ανάπτυξη μιας εφαρμογής που θα επιτρέπει στον χρήστη την γνωριμία με τους χώρους του κτιρίου και την ιστορία του.

Στόχος της έρευνας είναι να τεκμηριώσει την δυνατότητα δημιουργίας εκπαιδευτικών εφαρμογών με χρήση τεχνολογιών τρισδιάστατης μοντελοποίησης και τεχνολογιών διάδρασης με την εισαγωγή του μοντέλου και την δημιουργία του κατάλληλου κώδικα στην μηχανή παιχνιδιών Unity.

Εικόνα 1-1. Φωτογραφία της Ιονίου Ακαδημίας

1.1. Ορισμοί

Η διαδικασία τις δημιουργίας αντικειμένων σε ένα πρόγραμμα τρισδιάστατης μοντελοποίησης μπορεί να γίνει χαοτική. Ακολουθεί η εξήγηση όρων που είναι απαραίτητο να είναι γνωστοί για εμβάθυνση σε αυτό τον χώρο. Για τις μηχανές παιχνιδιών ισχύει το ίδιο, πρέπει να υπάρχει μια βασική γνώση των αρχών λειτουργίας με στόχο την ανάπτυξη μιας ολοκληρωμένη εφαρμογής. Είναι τελείως διαφορετικό το παίξιμο ενός παιχνιδιού από την δημιουργία ενός παιχνιδιού.

1.1.1. 3D Modeling - Blender

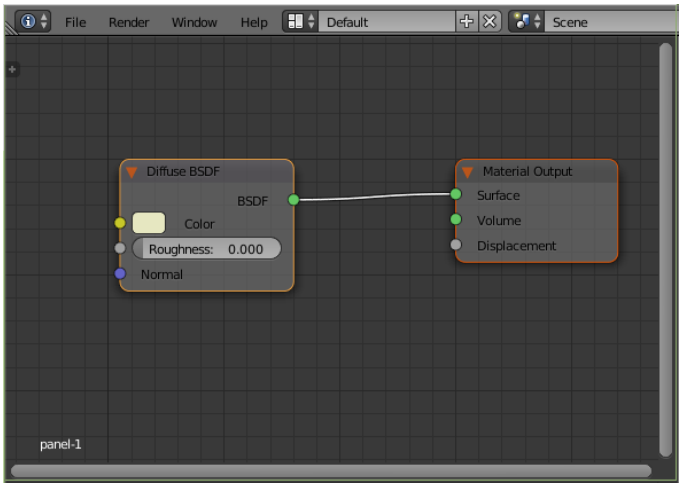
Με τον όρο τρισδιάστατη μοντελοποίηση (3D Modeling) νοείται η χρήση ενός προγράμματος που επιτρέπει την δημιουργία αντικειμένων στον χώρο, εν προκειμένω το Blender 2.79c. Υπάρχει ευρεία γκάμα προγραμμάτων του είδους, οι κανόνες και οι αρχές που εφαρμόζονται είναι κοινές μεταξύ τους, αν και ο τρόπος της χρήσης του καθενός διαφέρει.

Η απλούστερη δυνατή δομή για ένα μοντέλο είναι το επίπεδο (plane), το οποίο αποτελείται από τέσσερα σημεία (vertices). Οι έννοιες των vertices, edges και faces αναλύονται στο 5ο κεφάλαιο.

Ένα μοντέλο που δημιουργείται με αυτό τον τρόπο λέγεται 3D Model. Σε αυτό, αντιστοιχίζεται κατ' ελάχιστο ένα υλικό (material). Στο material εφαρμόζεται τουλάχιστον μία μέθοδος σκίασης (shading). Ο πιο διαδεδομένος shader είναι ο diffuse shader, το φως διαχέεται στην επιφάνεια του μοντέλου. Σε πολλές περιπτώσεις, ο diffuse shader συνοδεύεται από μια υφή (texture). Μια εικόνα δηλαδή που απλώνεται στην επιφάνεια του μοντέλου. Η διαδικασία της αντιστοίχησης του texture στην επιφάνεια του υλικού λέγεται UV Unwrapping, και η κατανομή του UV Layout (Litster 2011).

Υπάρχουν διαφορετικοί αλγόριθμοι σκίασης (shaders) ενός 3D model, για ρεαλιστικότερη απόδοσή του. Σαν παράδειγμα, ο glossy shader κάνει την επιφάνεια του μοντέλου να γυαλίζει. Για διάφανα υλικά υπάρχει ο transparency shader, για μεταλλικά ο metal shader, για γυαλί ο glass shader, και ούτω καθεξής. Η μηχανή γραφικής απόδοσης (rendering) που χρησιμοποιήθηκε είναι το Cycles Render (Steinmetz and Hofmann 2016), αυτή βασίζεται σε φυσικούς νόμους για να αποδίδει εξαιρετικά ρεαλιστικά αποτελέσματα (Iraci 2013). Η παλιότερη μηχανή απόδοσης είναι το Blender Render, ή Blender Internal. Χρησιμοποιήθηκε όμως το Cycles, καθώς το Blender Render έχει σταματήσει να εξελίσσεται.

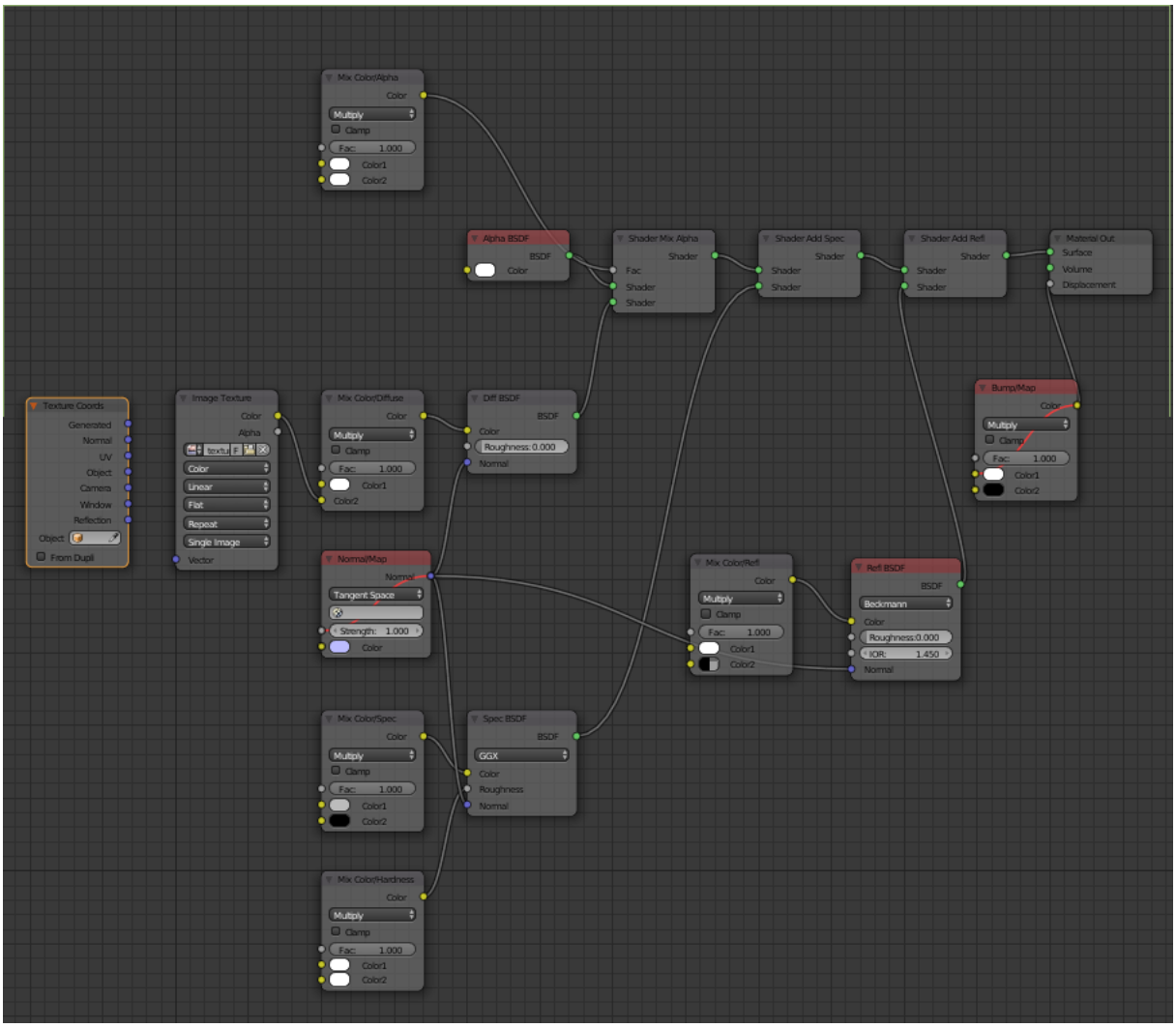
Η επεξεργασία και τροποποίηση ενός material με τους shader και τα textures του, γίνεται στον node editor, έχω την γνώμη ότι λέξη κόμβος δεν αποδίδει σωστά το νόημα αλλά δίνει μια ιδέα περί τίνος πρόκειται. Στην απλούστερη μορφή των nodes , αρκεί ένας diffuse shader και η έξοδος για το material (εικόνα 1.2), ενώ μπορούν να γίνουν πολύ σύνθετα με μίξη περισσότερων των 77 material nodes (η εικόνα 1.3 δίνει μια ιδέα για αυτό).



Εικόνα 1-2. Το απλούστερο node.

Η αρχή είναι ότι συνδέεται η έξοδος ενός node στην είσοδο κάποιου άλλου. Το υλικό της εικόνας 1.1, έχει έναν diffuse shader που συνδέεται με την έξοδο του υλικού που είναι πάντα και για όλα τα υλικά ένα. Είναι απαραίτητα το τελευταίο στην σειρά. Στην εικόνα 1.2 φαίνεται αυτή η αρχή για ένα πιο σύνθετο υλικό. Η τελική έξοδος είναι το material output, και πριν από την έξοδο μπορούν να συμβαίνουν πολλά και ενδιαφέροντα πράγματα.

Ο node editor του Cycles αποτελεί από μόνος του ένα πολύ μεγάλο κεφάλαιο, υπάρχει βιβλιογραφία ειδικά για την δημιουργία materials για την μηχανή Cycles. (Valenza 2016).



Εικόνα 1-3. Σύνθετο node για ένα αντικείμενο.

1.1.2. Game Engines - Unity

Σε γενικές γραμμές, τα προαναφερθέντα έχουν εφαρμογή και στο Unity. Με διαφοροποιήσεις βέβαια. Το Unity, όπως και το Blender διαθέτει μια γκάμα έτοιμων αντικειμένων που ονομάζονται primitives. Ενώ μπορούν να γίνουν αλλαγές στο μέγεθος, και ξεχωριστά για κάθε διάσταση, η χρήση του Unity για την δημιουργία σύνθετων μοντέλων δεν συνίσταται. Είναι προτιμότερο να γίνει εισαγωγή τους σε κάποια μορφή (format) που το Unity καταλαβαίνει. Επιλέχτηκε το .fbx που υποστηρίζει αντικείμενα με τα materials και τα textures τους. Προκύψαν όμως κάποια προβλήματα. Το Cycles παράγει πολύ ρεαλιστικά αποτελέσματα, αλλά το Unity δεν είναι σε θέση να διαβάσει αυτόματα τα υλικά και τα textures αν έχουν δημιουργηθούν αυτά για το Cycles Render Engine. Απαιτήθηκε επιπλέον

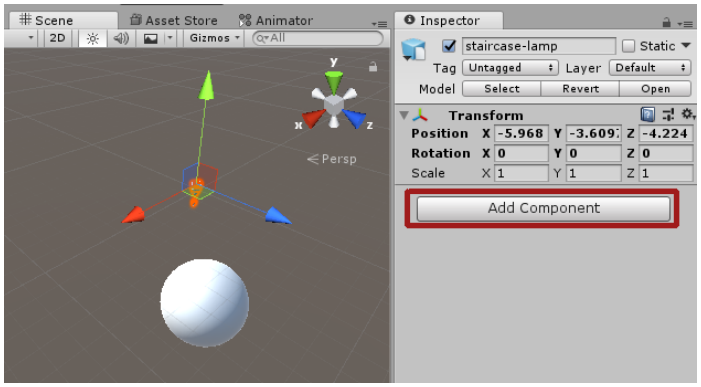
κόπος και χρόνος για να γίνει αυτό. Τα αποτελέσματα διαφέρουν, αλλιώς φαίνεται ένα material στο Blender, αλλιώς στο Unity. Το Unity χρησιμοποιεί τεχνικές απλοποίησης για να βελτιώσει τον χρόνο απόκρισης.

Ο επεξεργαστής του Unity (Editor) επιτρέπει την ανάπτυξη όλων των παραμέτρων ενός παιχνιδιού. Ένα αντικείμενο ή ένα ομαδοποιημένο σύνολο αντικειμένων λέγεται Asset. Δεν αρκεί η εισαγωγή των Assets στο Unity, θα πρέπει να προστεθούν σε μια σκηνή (scene) στην Ιεραρχία (Hierarchy). Επιτρέπονται εδώ δομές γονιού-παιδιού (parent-child) όπως και στο Blender. Ένα απλό παράδειγμα μιας τέτοιας δομής είναι μια πόρτα με το πόμολό της. Το πόμολο σε αυτήν την σχέση είναι το παιδί. Μπορεί να περιστρέφεται χωρίς να επηρεάζει την πόρτα. Η περιστροφή της πόρτας όμως συμπαρασύρει το πόμολο.

Ένα αντικείμενο (Gameobject) στο Unity μπορεί να έχει πολλά συστατικά στοιχεία (Components), με το καθένα να προσθέτει διαφορετικές ιδιότητες. Το Transform Component είναι υποχρεωτικό και καθορίζει την θέση και την περιστροφή στον χώρο. Κάποια άλλα που αν και δεν είναι υποχρεωτικά είναι απολύτως απαραίτητα:

- Για να είναι το αντικείμενο ορατό, προστίθεται ο mesh renderer, ή skinned mesh renderer αν το αντικείμενο εμπεριέχει κίνηση, όπως ένας άνθρωπος που περπατά. Αν δεν εμπεριέχεται κίνηση προστίθεται ένα mesh filter.
- Ένα material.
- Ένας Collider. Με την στενή έννοια δεν είναι απαραίτητος, όμως χωρίς την ύπαρξή του, δεν θα ισχύουν οι φυσικοί νόμοι συγκρούσεων.

Από κει και πέρα οι επιλογές είναι πάρα πολλές. Στον Inspector υπάρχει ένα κουμπί Add Component για την προσθήκη επιπλέον στοιχείων (εικόνα 1-4).



Εικόνα 1-4. Add Component στον Inspector

Ειδική μνεία θα πρέπει να δοθεί στην δυνατότητα προσθήκης ενός προγράμματος (μπορεί να είναι και περισσότερα) για τον έλεγχο της συμπεριφοράς του αντικειμένου. Στο Unity αυτά ονομάζονται script. Υποστηρίζονται δύο γλώσσες προγραμματισμού, η C# και η javascript.

Στην εφαρμογή που αναπτύχθηκε, χρησιμοποιήθηκε η C#. Η επιλογή είναι θέμα προτίμησης. Στην περίπτωσή μου, επειδή με αυτήν την γλώσσα ξεκίνησε η επαφή μου με το Unity, με αυτήν συνέχισα και με αυτήν δημιούργησα την εφαρμογή της Ιονίου Ακαδημίας.

ΚΕΦΑΛΑΙΟ 2. Ιστορία

Στην πολυτάραχη Ιστορία του Νησιού υπάρχει η παρουσία των Ρωμαίων αρχικά, και έπειτα του Βυζαντίου, με το Νησί να εντάσσεται στην Ανατολική Ρωμαϊκή Αυτοκρατορία. Τότε ήταν που χτίστηκαν τα περισσότερα φρούρια (Ιστορία της Κέρκυρας 2018). Ακολούθησαν οι Ενετοί, μετά οι Γάλλοι (κατά την παρουσία τους στο Νησί έκαψαν το libro d'oro), οι Ρωσότουρκοι και τέλος οι Άγγλοι (ηθικοί αυτουργοί του κουμ κουάτ και της τζιτζιμπύρας), πριν τα Επτάνησα περιέλθουν στην Ελλάδα.

Υπάρχουν πολύ περισσότερα ενδιαφέροντα γεγονότα στην Ιστορία του Νησιού πέρα από αυτά που αναφέρονται, αλλά μία αναλυτικότερη παρουσίαση ξεφεύγει από τον σκοπό αυτής της εργασίας. Ωστόσο οι ενδιαφερόμενοι μπορούν εύκολα να βρουν περισσότερες πληροφορίες (Ιστορία της Κέρκυρας 2018).

2.1. Η Περίοδος της Ενετοκρατίας.

Στην σύγχρονη ιστορία του Νησιού, η πιο μακρόχρονη περίοδος κατοχής ήταν η Ενετική Κυριαρχία που κράτησε σχεδόν τέσσερις αιώνες. Άρχισε το 1386 και διήρκησε μέχρι το 1797 (Αγιούς, Τα Κερκυραϊκά (Ιστορικά Ανάλεκτα) 2005).

Εικόνα 2-1. Ο Φτερωτός Λέων της Βενετίας (Ι.Α. Αγιούς 2005, σ. 61).

Το κτίριο που είναι γνωστό σήμερα σαν Ιόνιος Ακαδημία, χτίστηκε κατά την Διάρκεια της Ενετοκρατίας, από τους Γενικούς Προνοητές[1] Grimani και Correr. Η ακριβής χρονολογία δεν είναι γνωστή, αλλά πρέπει να χτίστηκαν μετά τον πόλεμο μεταξύ Τουρκίας και Βενετίας το 1571 (Στρατιωτικά κτίρια – στρατώνες κατά την Βενετοκρατία n.d.). Εκείνη την εποχή χρησίμευε σαν στρατώνας:

"Μέσα στην πόλη, κοντά στην πόρτα Raimonda, είν' ένας στρατώνας αποτελούμενος από δύο οικήματα που μπορούν να δώσουν κατάλυμα σε 1500 στρατιώτες. Το πρώτο, που χρησιμεύει για τους πεζούς, ονομάζεται στρατώνας Grimani, και το άλλο που στρατωνίζει πυροβολητές λέγεται στρατώνας Correr. Κι οι δύο πήραν τα ονόματα από τους Γενικούς Προνοητές που τους έχτισαν" (Αγιούς, Οδοιπορικό στην Παλιά Κέρκυρα 2004).

Οι Ενετοί είχαν την συνήθεια να δίνουν σχήμα Γ στους στρατώνες που ήταν κοντά στις πύλες εισόδου των πόλεων έτσι ώστε να προστατεύονται αποτελεσματικότερα οι πόλεις και τα γειτονικά τείχη (Ι.Α. Αγιούς 2004, σ. 121). Αξίζει να σημειωθεί εδώ ότι κατά την περίοδο της Ενετοκρατίας (1386 – 1797) (Ι.Α. Αγιούς 2004, σ. 122) δεν υπήρξε κανένα ενδιαφέρον για την εκπαίδευση, και όχι μόνο αυτό, απαγορευόταν και η λειτουργία τυπογραφείων. Οι Ενετοί με την πολιτική τους είχαν εγκαταλείψει τα Επτάνησα στην τύχη τους σε ότι αφορά την πνευματική καλλιέργεια και την εκπαίδευση (Henderson 1980, ο.π.). Ο μεσαίωνας ούτως ή άλλως είναι σκοτεινή περίοδος, οι Ενετοί είχαν έναν λόγο παραπάνω όμως να κρατούν τους ιθαγενείς υπόδουλους και μέσα στην αμάθεια. Θεωρούσαν τους Έλληνες εύστροφους και ριψοκίνδυνους. Αν είχαν πρόσβαση στην παιδεία, θα γίνονταν επικίνδυνοι γι'αυτούς και την εξουσία τους.

Ενώ στην Ευρώπη η Γαλλική Επανάσταση είχε ως αποτέλεσμα την πρωτοφανή ανάπτυξη των επιστημών και τεχνών, στην Ενετική Αποικία δεν συνέβαινε κάτι τέτοιο. Όταν οι Δημοκρατικοί Γάλλοι αποβιβάστηκαν στην Κέρκυρα το 1797, υπήρχε μόνο μια αμυδρή υποψία για το τι συνέβαινε έξω από το νησί (Ροδακανάκης 1937).

Θα πρέπει όμως να σημειωθεί ότι αν και οι Ενετοί ήταν ενάντιοι στην εκπαίδευση των Κερκυραίων, αλλά και των Επτανήσιων γενικά, το φαινόμενο των "κρυφών σχολειών", σε αντίθεση με την υπόλοιπη Τουρκοκρατούμενη Ελλάδα δεν υπήρχε. Αυτό συνέβαινε επειδή οι

[1] Έτσι λέγονταν οι ευγενείς που είχαν πολιτική εξουσία.

Ενετοί δεν εμπόδιζαν την μόρφωση των Κερκυραίων σε Πανεπιστήμια της Ιταλίας και άλλων Ευρωπαϊκών χωρών, ούτε την ίδρυση Ιδιωτικών Σχολείων (Ι.Α. Αγιούς 2004, σ. 125).

Από τις αρχές του 15ου αιώνα τα Επτάνησα έχουν να επιδείξουν πολλούς ανθρώπους των γραμμάτων. Έγγραφο της Πρεσβείας της Κέρκυρας προς την Βενετία το 1546 ζητάει την χρηματοδότηση δύο διδασκάλων με 80 δουκάτα τον χρόνο από τα έσοδα των αλυκών, ενός ελληνόφωνου και ενός λατινόφωνου για τους ξενόγλωσσους νέους (Ι.Α. Αγιούς 2004, σ. 126). Οι διδάσκαλοι εκλέγονταν από το Συμβούλιο των Ευγενών Κερκυραίων και κατά κανόνα ήταν κληρικοί.

Το 1571 σε ανάμνηση της Ναυμαχίας της Ναυπάκτου, ιδρύθηκε από Φραγκισκανούς Μοναχούς το Μοναστήρι της Αγίας Ιουστίνας στην Γαρίτσα. Εκεί εκπαιδεύονταν κατά κύριο λόγο τα παιδιά των εύπορων Κερκυραίων. Εκεί μαθήτευσε και ο Ιωάννης Καποδίστριας στα πρώιμα χρόνια του, πριν πάει στην Πάντοβα το 1795 για να σπουδάσει Ιατρική.

Το 1694, με την περιουσία που είχε αφήσει στην Μονή της Χρυσοπηγής στους Καστελάνους Μέσης, ιδρύθηκε ιεροσπουδαστήριο και το 1757 ιδρύθηκε το Ιδιωτικό Φροντιστήριο του Νικηφόρου Θεοτόκη, το οποίο αναγνωρίστηκε και επίσημα από την Κοινότητα (Ι. Α. Αγιούς 2004, σ. 126).

Το 1656, αφού είχε ωριμάσει η ιδέα των Ακαδημιών, ιδρύθηκε η "Ακαδημία των Εξησφαλισμένων". Λειτούργησε για εξήντα χρόνια και διαλύθηκε με την πολιορκία της Κέρκυρας από τους Τούρκους το 1716 για να επανιδρυθεί στην συνέχεια σαν "Ακαδημία των Γονίμων". Το 1732 ιδρύθηκε η "Ακαδημία των Περιπλανωμένων" (Ι.Α. Αγιούς 2004, σ. 127).

Πάντως, αυτές οι προσπάθειες δεν είχαν ιδιαίτερη επιτυχία αφού απευθύνονταν και επηρέασαν πολύ μικρό ποσοστό του πληθυσμού, την αριστοκρατική τάξη της Κέρκυρας. Η επιτυχία τους έγκειται στο ότι άναψαν την αρχική σπίθα που φώτισε αργότερα όλη την Ελλάδα.

Το 1797 η κυριαρχία των Ενετών τελείωσε. Όχι γιατί οι Επτανήσιοι ξεσηκώθηκαν εναντίον τους, δεν υπήρξε καμία αντίδραση σχετικά. Με την κατάληψη της Βενετίας από τα στρατεύματα του Ναπολέοντα η ενετική κυριαρχία απλά κατέρρευσε (Henderson 1980, σ.13).

2.2. Η Περίοδος της Γαλλικής Κυριαρχίας

Μετά την κατάλυση της Δημοκρατίας του Αδρία, ήρθαν στην Κέρκυρα οι Δημοκρατικοί Γάλλοι του Βοναπάρτη το 1797, οι οποίοι αφού διέλυσαν και το τελευταίο ακαδημαϊκό ίδρυμα, την "Ακαδημία των Περιπλανωμένων", έθεσαν νέες βάσεις στο εκπαιδευτικό σύστημα.

Στην σύντομη παρουσία τους, οι Γάλλοι έθεσαν τις πρώτες βάσεις του εκπαιδευτικού συστήματος όπως το γνωρίζουμε σήμερα.

Τον Ιανουάριο του 1798 ιδρύθηκε το πρώτο δημόσιο πρωτοβάθμιο εκπαιδευτικό ίδρυμα το οποίο στεγάστηκε στο Μοναστήρι του Αγίου Φραγκίσκου (Ι.Α. Αγιούς 2004, σ. 128).

Στην συνέχεια οι Γάλλοι φρόντισαν να διαδώσουν τα δημοκρατικά ιδεώδη με την έναρξη της λειτουργίας του πρώτου τυπογραφείου τον Ιούνιο του 1798, στο οποίο τυπώθηκε ο Θούριος του Ρήγα Φεραίου (Ι.Α. Αγιούς 2004, σ. 121).

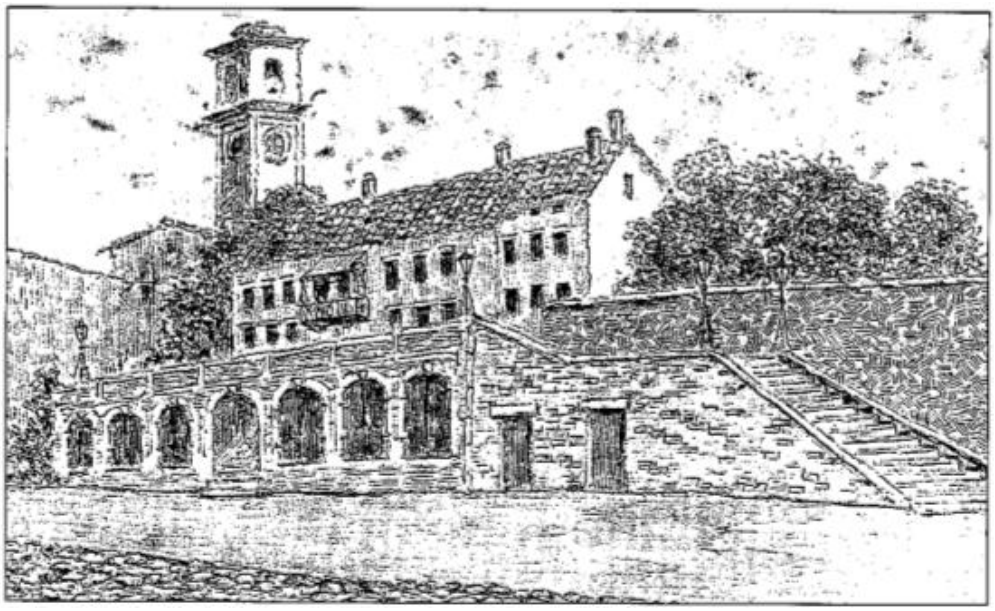
Για να αναχαιτιστούν οι επεκτατικές βλέψεις του Ναπολέοντα στην ανατολική Ευρώπη, δημιουργήθηκε η Ρωσοτουρκική Συμμαχία. Το 1799 οι Ρώσοι, εκμεταλλευόμενοι την δυσαρέσκεια των κατοίκων στα Ιόνια, κατέλαβαν την Κέρκυρα. Το 1800 ιδρύεται η Επτάνησος Πολιτεία, που ήταν υποτελής φορολογικά στον Σουλτάνο. Αυτήν την περίοδο, έγινε η ανασύσταση του θεσμού του Ορθόδοξου Επίσκοπου Κέρκυρας. Τελικά με την υπογραφή της συνθήκης του Τιλσίτ, οι Ρώσοι παραχώρησαν την Κέρκυρα στους Αυτοκρατορικούς πια Γάλλους το 1807 (Σύντομη ιστορία της Κέρκυρας n.d.). Για την ακρίβεια, με το δεύτερο μυστικό άρθρο της συνθήκης, η Επτάνησος Πολιτεία παραχωρήθηκε απευθείας στον Ναπολέοντα (Συνθήκη του Τιλσίτ 2017). Όμως, σε αυτήν την σύντομη περίοδο κατοχής από τους Ρωσότουρκους, οι όποιες προσπάθειες είχαν γίνει προηγούμενα από του Γάλλους για την δημιουργία ενός εκπαιδευτικού συστήματος, διακόπηκαν απότομα.

Το 1808 ιδρύεται η "Ιονική Ακαδημία", πρώτο εκπαιδευτικό ίδρυμα που μπορεί να θεωρηθεί Πανεπιστημιακό (Ι. Α. Αγιούς 2004, σ. 128). Σκοπός αυτής της Ακαδημίας ήταν η τελειοποίηση της γεωργίας, της βιομηχανίας και του εμπορίου, και η ανάπτυξη των πλουτοπαραγωγικών πηγών του Νησιού.

Το 1814, με το τέλος της Γαλλικής Παρουσίας στην Κέρκυρα, σταματά και η λειτουργία της "Ιονικής Ακαδημίας".

2.3. Η Περίοδος της Αγγλικής Κυριαρχίας

Το 1815, με την συνθήκη των Παρισίων (Συνθήκη των Παρισίων (1814) 2015), τα Επτάνησα αναγνωρίζονται ως ελεύθερα υπό Αγγλική Προστασία. Το παλάτι του Γενικού Προνοητή Θαλάσσης (εικόνα 2.2) άλλαξε άλλη μια φορά ένοικο. Αφού χρησίμευσε σαν κατοικία του Ενετού Προνοητή, φιλοξένησε τον Γάλλο Φρούραρχο, και τελικά τον Άγγλο Αρμοστή.



Εικόνα 2-2. Το παλάτι του Γενικού Προνοητή Θαλάσσης (Ι. Α. Αγιούς 2005, σ. 138).

Σε αυτό το κτίριο στεγάστηκε η Ιόνιος Ακαδημία από το 1824 ως το 1841.

Ο Thomas Maitland, ανέλαβε Γενικός Ύπατος Αρμοστής των Ιονίων Νήσων τον Δεκέμβρη του 1815. Ήταν κυβερνήτης της Μάλτας, θέση την οποία διατήρησε και μετά την εκλογή του ως Γενικού Ύπατου Αρμοστή των Ιονίων Νήσων. Ήταν επίσης αρχηγός των Βρετανικών Δυνάμεων στην Μεσόγειο. Η θητεία του κρίνεται ως επιτυχημένη στρατιωτικά και αποτυχημένη πολιτικά. Έδωσε στο κράτος των Ιονίων Νήσων το Μαιτλάνδιο Σύνταγμα, το οποίο ψηφίστηκε με συνοπτικές διαδικασίες τρεις μέρες μετά από την υποβολή του. Εν συντομία, με το Σύνταγμα αυτό, έβαζε τον εαυτό του στην θέση του απόλυτου άρχοντα, και κυβέρνησε αυταρχικά (Ι. Α. Αγιούς 2005). Στις μαύρες σελίδες της διακυβέρνησής του συγκαταλέγεται η ανάμιξή του στην πώληση της Πάργας στους Τούρκους (Ι.Α. Αγιούς 2005, σ.182) έναντι 150.000 λιρών.

Εικόνα 2-3. Η φυγή από την Πάργα, Πίνακας του Διονύσιου Τσόκου (Ι. Α. Αγιούς 2005, σ. 183).

Η Γαλλική Επανάσταση του 1789 είχε ως απόρροια την δημιουργία συμμαχιών στις Ευρωπαϊκές χώρες με σκοπό την αναχαίτιση του Ναπολέοντα. Η Γαλλία κατάκτησε σημαντικό μέρος της Ευρώπης, απέτυχε παταγωδώς όμως να καταλάβει την Ρωσία, κάτι που τελικά οδήγησε στην οριστική απομάκρυνση του Ναπολέοντα το 1818. Στην Γερμανία και την Ιταλία αναπτύχθηκε ο εθνικισμός που θα οδηγούσε τελικά στην μεταμόρφωσή τους σε συμπαγή κράτη. Στην Ισπανία, η κατοχή της Γαλλίας οδήγησε σε εξασθένηση του ελέγχου που είχε στις αποικίες της. Αυτό επέτρεψε την δημιουργία εθνικών επαναστατικών κινημάτων στις χώρες της Λατινικής Αμερικής (Ναπολεόντειοι Πόλεμοι 2017).

Εδώ, θα πρέπει να ληφθούν υπόψη οι πολιτικές συνθήκες τις εποχής και η οξυδέρκεια του Καποδίστρια. Η Συνθήκη των Παρισίων του 1814 (Συνθήκη των Παρισίων (1814) 2015), η πρώτη ειρήνη των Παρισίων όπως ονομάστηκε, πραγματοποιήθηκε μετά την ήττα του Ναπολέοντα από τον συνασπισμό των κρατών της Ευρώπης. Μερικούς μήνες αργότερα, οδήγησε στο Συνέδριο της Βιέννης τον Μάιο, με τον Ναπολέοντα από την μία πλευρά και τις Ρωσία, Αυστρία, Αγγλία και Πρωσία από την άλλη. Στην ουσία, το παλιό αντιμαχόταν το νέο, οι συνασπισμένοι επανέφεραν την Βασιλεία των Βουρβόνων και τον Βασιλιά Λουδοβίκο τον ΙΗ’.

Οι περισσότερες διαφωνίες δημιουργήθηκαν για το Επτανησιακό Ζήτημα. Οι Ρώσοι πρότειναν να γίνουν τα Επτάνησα με τα παρακείμενα νησιά ανεξάρτητο, ουδέτερο και κυρίαρχο κράτος με το όνομα "Επτάνησος Πολιτεία", και να συμπεριλαμβάνονται από το

Δεσποτάτο της Ηπείρου η Πάργα, η Πρέβεζα, η Βόνιτσα και το Βουθρωτό. Αυτό προκάλεσε την έντονη αντίδραση του Μέτερνιχ. Θεωρούσε ότι τα Επτάνησα ως κατάλοιπα της Ενετικής Αυτοκρατορίας έπρεπε να περιέλθουν στην Αυστρία, που ήταν ο Νόμιμος Διάδοχος των κτήσεων αυτής. Η Γερουσία των Επτανήσων είχε στείλει έγγραφο στους Αντιπροσώπους των Βασιλέων ζητώντας να αναγνωριστούν τα Επτάνησα ως ανεξάρτητο και κυρίαρχο κράτος χωρίς καμία υποτέλεια, την διατήρηση του Γαλλικού Συντάγματος του 1803 με αλλαγές όπου κρινόταν απαραίτητο, και την ενσωμάτωση των Ηπειρωτικών Παραλίων (Συνθήκη των Παρισίων (1814) 2015).

Η νίκη των Άγγλων έναντι των Γάλλων στο Βατερλό στις 18 Ιουνίου του 1815, είχε ως αποτέλεσμα να μείνει η Αγγλία η μοναδική Στρατιωτική Υπερδύναμη στον κόσμο.

Υπό αυτές τις συνθήκες ανέλαβε ο Πρώτος Ύπατος Αρμοστής, Thomas Maitland την Διοίκηση των Ιονίων Νήσων. Δεν έδειξε ιδιαίτερο ενδιαφέρον για την εκπαίδευση, για την ακρίβεια αντιστάθηκε στο φιλελληνικό σχέδιο δημιουργίας πανεπιστημίου στα Ιόνια Νησιά.

Όμως ο Καποδίστριας, από την Βενετία όπου ήταν αυτοεξόριστος, είχε αντιληφθεί το τι συνέβαινε στην ευρωπαϊκή σκακιέρα.

Σε υπόμνημά του στις 22 Νοέμβρη 1815 προς τον συντηρητικό Άγγλο Υπουργό Εξωτερικών, Λόρδο Robert Stewart, υποκόμη Castlereagh[2], ο πρώτος Κυβερνήτης της Ελλάδας εκφράστηκε με πάρα πολύ αρνητικά λόγια για την Ενετική προσέγγιση στο θέμα της Παιδείας:

"Η Βενετική Πολιτεία, απονέμουσα προνόμια και κτήματα εις τους ευγενείς, παρείχεν εις αυτούς τα μέσα να εξαγοράζωσι την συνείδησην του λαού και καθιστώσι συστηματικήν την ανηθικότητα και την διαφθοράν του τόπου. Η Βενετική Πολιτική εφοβείτο την υπεροχήν του Ελληνικού πνεύματος και προσεπάθη να καταβάλη τούτο δια της αμαθείας, δια τούτο ουδέποτε επέτρεψε την σύστασιν δημοσίων σχολίων εν Επτανήσω…" (Κονόμος 1965).

Τυχόν ομοιότητες και αναλογίες με την σημερινή κατάσταση θα πρέπει να μας προβληματίζουν.

[2] Και όχι Castleregh όπως αναγράφεται απο τον Αγιούς (Ι. Α. Αγιούς 2004, "Οδοιπορικό στην Παλιά Κέρκυρα", σ.123.

Κατά τον Θεοτόκη (Θεοτόκης 1956) αυτή η αυστηρή τοποθέτηση έχει δύο ερμηνείες. Ο Καποδίστριας μεγάλωσε και μαθήτευσε κατά τους τελευταίους χρόνους της Ενετοκρατίας, την περίοδο της παρακμής της. Εκτός αυτού όμως, αποσκοπούσε και στην συμπάθεια των Άγγλων ώστε μετά το Συνέδριο των Παρισίων το 1814 (Συνθήκη των Παρισίων (1814) 2015), να στηρίξουν και να προστατέψουν την εκπαίδευση των Επτανησίων.

Ο Castlereagh προσπαθούσε να αποδυναμώσει την Ιερή Συμμαχία (περισσότερα στην επόμενη παράγραφο) και αυτό το είχε αντιληφθεί ο Καποδίστριας. Η επιστολή του δεν είχε άμεσα αποτέλεσμα, όμως ο σπόρος φυτεύτηκε. Τον Αύγουστο του 1822 ο Λόρδος Castlereagh αυτοκτόνησε, για να τον διαδεχτεί ως Υπουργός Εξωτερικών ο George Canning. Έχει ενδιαφέρον να ειπωθεί ότι ήταν υποστηρικτής της κατάργησης της Δουλείας. Υποστήριξε και συνέχισε την άποψη του Castlereagh ότι δεν πρέπει να επιτραπεί στις μεγάλες ευρωπαϊκές δυνάμεις (Ρωσία, Γαλλία, Αυστρία, κ.α.), να αναμιγνύονται στις υποθέσεις άλλων χωρών. Ως συνέπεια αυτού, έδωσε το τελειωτικό χτύπημα στην Ιερή Συμμαχία και τον Μέτερνιχ με το “Ελληνικό Ζήτημα” (Αλειφαντής Στέλιος 2016).

Η Ιερά Συμμαχία ιδρύθηκε μετά την ήττα του Ναπολέοντα, αρχικά σαν τριμερής συνθήκη μεταξύ Ρωσίας, Αυστρίας και Πρωσίας το 1815. Στην συνέχεια προσχώρησε σε αυτήν και η Αγγλία, και τέλος το 1818 η Γαλλία (Ιερά Συμμαχία 2018). Οι Τσάρος Αλέξανδρος ο Α' στήριζε θεωρητικά την απελευθέρωση της Ελλάδας από τον Τουρκικό ζυγό, με την Συνθήκη του Κιουτσούκ Καϊναρτζή το 1774. Στην πράξη όμως οι ελπίδες των Ελλήνων για παρέμβαση του "Ξανθού Γένους" ποτέ δεν επαληθεύτηκαν. Οι Έλληνες πήραν την κατάσταση στα χέρια τους με την επανάσταση του 1821. Η Ιερά Συμμαχία θεωρούσε την επανάσταση έργο του διαβόλου, και την σκλαβιά κάτω από ζυγό των Τούρκων έργο του Θεού. Στις 14 Δεκέμβρη του 1822, καταδίκασε την ελληνική επανάσταση ως αυθάδη και αδέξια (Ετικέτα: Ιερή συμμαχία... n.d.).

Στις 17 Ιανουαρίου του 1824 ο Maitland πέθανε στην Μάλτα, γεγονός που δεν συγκίνησε τους ανθρώπους στα Ιόνια Νησιά. Τον διαδέχτηκε στις 10 Απριλίου 1824 ο Sir Frederick Adam. Η Αρμοστεία του, συνέπεσε με την αλλαγή εξωτερικής πολιτικής της Αγγλίας και με την παρουσία του Canning στο Υπουργείο Εξωτερικών. Έτσι ο Adam, αν και αρχικά ακολούθησε τα βήματα του προκάτοχού του, τελικά κράτησε πιο φιλελληνική στάση.

Ο Άνταμ εμπλέκεται μάλιστα σε μία πικάντικη ιστορία που συγκλόνισε την τοπική κοινωνία της Κέρκυρας, λίγο πριν αναλάβει την Διοίκηση του Νησιού. Αποπλάνησε και απήγαγε την ντόπια Νίνα Σούφη, κόρη της επιφανούς οικογένειας των Παλατιανών. Το θέμα

λύθηκε "πολιτισμένα", μετά την παρέμβαση του Άγγλου Πρέσβη Κωνσταντινούπολης που μεσολάβησε ώστε να παντρευτεί το ζεύγος με την άδεια του Πατριάρχη Ευγένιου, για να ζήσουν τελικά αυτοί καλά, κι εμείς καλύτερα.

Όποιος και αν είναι ο πραγματικός λόγος της στάσης που κράτησε ο Adam, επί των ημερών του λειτούργησε για πρώτη φορά η Ιόνιος Ακαδημία. Αναφέρθηκε ήδη ότι από το 1824 ως το 1841 λειτούργησε στο Παλάτι του Γενικού Προνοητή. Αυτό που δεν ειπώθηκε είναι ότι πρωτολειτούργησε το 1823 στην κατοικία του Frederick North με τον ίδιο να είναι από τους πρώτους διδάσκοντες (Αγιούς, Οδοιπορικό στην Παλιά Κέρκυρα 2004) σ. 129. Αξίζει πραγματικά να αναφερθούν περισσότερα γι'αυτόν τον άνθρωπο και τον φιλελληνισμό του. Η επιρροή του στην Ίδρυση και λειτουργία της Ιονίου Ακαδημίας υπήρξε καταλυτική.

ΚΕΦΑΛΑΙΟ 3. Ακαδημία

Ο Frederick North ο 5ος, κόμης του Guilford, ήταν ο νεότερος γιος του πρωθυπουργού Frederick North του 2ου. Γεννήθηκε στις 7 Φεβρουαρίου 1776 και πέθανε στις 14 Οκτώβρη 1827. Εκπαιδεύτηκε στο κολλέγιο του Ήτον και την Χριστιανική Εκκλησία της Οξφόρδης.

Η εκπαίδευσή του τον έφερε σε επαφή με την κλασσική Παιδεία, δεν ολοκληρώθηκε όμως στην Οξφόρδη. Γνώρισε από πρώτο χέρι τους αρχαίους τόπους και τα μνημεία τους, ερχόμενος σε επαφή μαζί τους ήδη από τα χρόνια της Ενετοκρατίας. Επισκέφτηκε, μεταξύ άλλων, την Τίρυνθα, τις Μυκήνες, την Αθήνα, την Αγία Μαύρα (Λευκάδα), τον Άθω, την Κέρκυρα (Henderson 1980, σ.27). Είχε ιδρύσει σχολείο στην Πρέβεζα και την Μάνη (Αγγελομάτη-Τσουγκαράκη 1997). Διέμεινε έναν χρόνο στην Ιθάκη μαθαίνοντας την Νέα Ελληνική Γλώσσα. Το τελευταίο έχει την σημασία του, καθώς οι πρώτες σκέψεις για την δημιουργία ενός Πανεπιστημίου για όλους τους Έλληνες, ήταν να γίνει αυτό στην Ιθάκη. Πρακτικοί λόγοι όμως, αλλά και πολιτικοί, καθώς ξέσπασε η Ελληνική Επανάσταση οδήγησαν τελικά στην δημιουργία του Πανεπιστημίου στην Κέρκυρα (Henderson 1980).

Στην οποία Κέρκυρα επέστρεψε το 1791 και ασπάστηκε τον Ορθόδοξο Χριστιανισμό. Η ένταξή του στην Βρετανική Δημόσια Υπηρεσία, τον έφερε τελικά από το 1798 ως το 1805 στην Κεϋλάνη, της οποίας ανέλαβε Κυβερνήτης. Κυβέρνησε ανθρώπινα, ενδεικτικά να αναφέρω ότι θέσπισε νόμο που επέτρεπε στους αγρότες να εξαγοράζουν την στρατιωτική θητεία πληρώνοντας από τα έσοδά τους. Απόδειξη ότι ήταν υποστηρικτής κάθε μορφής

εκπαίδευσης είναι ότι ίδρυσε σχολεία στην Κεϋλάνη, και στήριζε την δημιουργία Ακαδημίας εκεί (Αγγελομάτη–Τσουγγαράκη 1997, σ. 4).

Το 1819 διορίστηκε Καγκελάριος της Κέρκυρας, υπό την Αρμοστεία του Maitland. Αυτό του επέτρεψε να ασχοληθεί με την οργάνωση της εκπαίδευσης όλων των βαθμίδων (Φρέντερικ Νορθ (1766-1827) 2018). Η συγκυρία ήταν καταλυτική, αν ληφθεί υπόψη ότι στο Σύνταγμα των Ιονίων Νήσων του 1817, άρθρο 23, προβλεπόταν η ίδρυση πρωτοβάθμιων σχολείων πρώτα, και στην συνέχεια Πανεπιστημίου στα Επτάνησα (Αγγελομάτη-Τσουγγαράκη 1997, σ.23). Με τον διορισμό του στην θέση του Καγκελάριου, έγινε επίτιμος Διδάκτορας του Πανεπιστημίου της Οξφόρδης, το οποίο έκανε δωρεά βιβλίων στον Γκίλφορντ (Αγγελομάτη-Τσουγγαράκη 1997, σ. 43).

Ο Αγγλικός τύπος της εποχής έκανε λόγο για την ίδρυση Πανεπιστημίου στην Κέρκυρα με διορισμό Ελλήνων καθηγητών σε αυτό. Πράγμα που ήταν ανακριβές, στην παρούσα χρονική στιγμή ήταν επικρατέστερη σαν έδρα η Κεφαλονιά. Η παρουσία αυτών των άρθρων στον Αγγλικό τύπο συνέπεσε με την παρουσία του Καποδίστρια στο Λονδίνο (Αγγελομάτη-Τσουγγαράκη, σ.43), αλλά δεν μπορεί να αποδειχτεί αν υπήρξε σχετική ανάμιξή του.

Τελικά στις 19 Μάη 1823 ιδρύεται με απόφαση της Ιονίου Βουλής, και μετά τις επίμονες και μακροχρόνιες προσπάθειες του η Ιόνιος Ακαδημία (Ι.Α. Αγιούς 2004, σ.129).

3.1. Ιόνιος Ακαδημία

Η προκαταρκτική εκπαιδευτική λειτουργία ξεκίνησε το Φθινόπωρο στην κατοικία του, που του είχε παραχωρηθεί από τον φίλο του Γεώργιο Φορέστη, δίπλα από τον Μητροπολιτικό Ναό (Ι. Α. Αγιούς 2004, σ. 129). Στην πρόσοψη του Ναού υπάρχει και σήμερα μαρμάρινη πλάκα με τα ονόματα των επτά καθηγητών και εκατόν πενήντα φοιτητών που συμμετείχαν σε αυτό το πρώτο έτος λειτουργίας. Από τους εκατόν πενήντα φοιτητές έγιναν τελικά δεκτοί στο πρώτο έτος της επίσημης λειτουργίας της Ιονίου Ακαδημίας οι σαράντα επτά.

Το επόμενο έτος, με την ολοκλήρωση της κατασκευής των ανακτόρων, και την εγκατάσταση του Ύπατου Αρμοστή σε αυτά, η Ιόνιος Ακαδημία στεγάστηκε στο πρώην Κυβερνητικό Μέγαρο (Παλάτι του Ύπατου Αρμοστή), εντός των τειχών.

Σε αυτή την πρώτη φάση λειτουργίας, προβλεπόταν να λειτουργήσουν τέσσερις σχολές. Η Φιλοσοφική, η Θεολογική, η Νομική και η Ιατρική. Όμως υπήρχαν αντιδράσεις. Η ντόπια άρχουσα τάξη δεν έβλεπε με καλό μάτι την μόρφωση και εκπαίδευση του "πόπολου" και

αντιστεκόταν, καθώς απειλούνταν το μονοπώλιο που είχαν στην Πανεπιστημιακή Εκπαίδευση στην Ιταλία, και ως συνέπεια στην συνέχεια, την κατάληψη θέσεων και αξιωμάτων. Η Αγγλική Διοίκηση δεν αντιδρούσε μόνο λόγω του κόστους λειτουργίας, αλλά και επειδή το Πανεπιστήμιο απευθυνόταν σε όλη την Ελλάδα και όχι μόνο στα Επτάνησα (Αγγελομάτη n.d.), καθώς δεν έβλεπαν θετικά την Ελληνική Επανάσταση. Αργότερα, με την Αρμοστεία του John Seaton, ο οποίος διαδέχτηκε τον Frederick Adam, ιδρύθηκαν οι σχολές της Μαιευτικής και της Φαρμακολογίας (Ι. Α. Αγιούς 2004, σ. 133).

Το αξιοθαύμαστο και άξιο αναφοράς γεγονός, είναι ότι σε αυτά τα πρώτα χρόνια, τα έξοδα λειτουργίας τα ανέλαβε προσωπικά ο Frederick North, κόμης του Guilford. Θα επιμείνω εδώ και θα τονίσω ότι, η ντόπια άρχουσα τάξη και η Αγγλική Διοίκηση δεν συμμετείχαν στα έξοδα λειτουργίας, ο κόμης του Guilford ξόδευε την προσωπική του περιουσία για να εγγυηθεί την λειτουργία της Ιονίου Ακαδημίας. Οι πληρωμές των διδασκόντων αλλά και η οικονομική στήριξη προς τους σπουδάζοντες γίνονταν από τον ίδιο τον Guilford.

Οπότε μπορεί να του συγχωρεθεί και με το παραπάνω, αν μπορεί να θεωρηθεί έτσι, ο εγωισμός του να αναφέρεται στην Ιόνιο Ακαδημία σαν "το Πανεπιστήμιο μου" (Αγγελομάτη-Τσουγγαράκη 1997, σ.41). Προσωπικά θεωρώ ότι δεν υπάρχει τίποτε μεμπτό, ήταν πραγματικά δικό του δημιούργημα. Όχι μόνο έκανε ότι ήταν ανθρωπίνως δυνατό για να δει το όραμά του να γίνεται πραγματικότητα, το στήριξε και με ιδίους πόρους στα πρώτα χρόνια της λειτουργίας του.

Υπήρχαν προσπάθειες να δημιουργηθεί και ένα Περιοδικό το οποίο θα λειτουργούσε ως εργαλείο του Πανεπιστημίου. Οι αρχικές επαφές με τον Δ. Γ. Σχινά που έδρευε στο Λονδίνο δεν απόδωσαν, η συνεργασία ήταν προβληματική. Τελικά, και αφού ο Σχινάς καταχράστηκε 75 λίρες, ο Γκίλφορντ τον απέλυσε (Αγγελομάτη-Τσουγγαράκη 1997, σ. 65). Η επιστολή του προς την Γερουσία όπου δήλωνε την πρόθεσή του για την έκδοση ενός περιοδικού, με δικαίωμα λογοκρισίας από αυτήν, δεν ευοδώθηκε καθώς τον Οκτώβρη του 1827 ο Γκίλφορντ πέθανε.

Ο Maitland, ως εκφραστής της Βρετανικής Κυβέρνησης επιθυμούσε την Ίδρυση του Πανεπιστημίου στην Κεφαλονιά. Το 1818 φαίνεται να ανακοίνωσε στους Κεφαλλονίτες για μεγάλη τους χαρά, ότι αυτό θα γινόταν στο νησί τους (Αγγελομάτη-Τσουγγαράκη 1997, σ. 66). Το 1820 όμως τα πράγματα άρχισαν να αλλάζουν, καθώς μετά από πρόταση του Maitland στις 7 Μάρτη, εκλέχτηκε από την Γερουσία αρμόδιος Αρχηγός και Πρόεδρος της Ακαδημίας ο Γκίλφορντ (Αγγελομάτη n.d.). Υπήρχε ενδιαφέρον και από τα άλλα νησιά. Ο

Ηγούμενος της Μονής του Σκοπού Ζακύνθου, Κωνσταντίνος Λογοθέτης, γράφοντας στον Γκίλφορντ, πρόσφερε το μοναστήρι του γι'αυτό τον σκοπό, ζητώντας να κρατήσει μόνο την Εκκλησία, δύο κελιά και το σπίτι του. Και η Ιθάκη εκδήλωσε ενδιαφέρον, με τον Γκίλφορντ να την θεωρεί αρχικά ως το πλέον κατάλληλο μέρος για την Λειτουργία του Πανεπιστημίου. Οι κάτοικοι μάλιστα της Ιθάκης πρόσφεραν εθελοντική εργασία και υλικά για την ίδρυση του στο νησί τους. Πρόσφεραν και στέγη στους φοιτούντες έναντι συμβολικού ποσού.

Με την ύπαρξη Νοσοκομείου και Δικαστηρίων στην Κέρκυρα, οι φοιτούντες στην Ιατρική και την Νομική είχαν το πλεονέκτημα της πρακτικής εξάσκησης των όσων διδάσκονταν. Αλλά και για τους Διδάσκοντες ήταν ευνοϊκό καθώς διευκολύνονταν η διαμονή τους. Παρόλα αυτά όμως, δεν ήταν επικρατέστερη, καθώς θεωρήθηκε ότι το κόστος διαβίωσης και η ζωή εκεί θα διέφθειρε τους νέους, και θα τους απομάκρυνε από την πειθαρχία, την μελέτη, την σκληρή οικονομία και την ηθική που όφειλαν να έχουν.

Η Αγία Μαύρα (Λευκάδα), δεν ελήφθη καθόλου υπόψη, καθώς είχε ταλαιπωρηθεί από σεισμούς και η πόλη είχε υποστεί ζημιές. Τα Κύθηρα και οι Παξοί αποκλείστηκαν λόγω του μικρού τους μεγέθους.

Επικρατέστερη θεωρούνταν η Ιθάκη. Στην Αγγλία ήταν ιδιαίτερα σεβαστή η κληρονομιά της από τους αρχαίους χρόνους ως πατρίδα του Οδυσσέα (Αγγελομάτη-Τσουγγαράκη, σ. 74). Επιπρόσθετα, το περιβάλλον θεωρούνταν ιδανικό για την λειτουργία Πανεπιστημίου, και οι ντόπιοι, που είχαν αγκαλιάσει το εγχείρημα θερμά, εκδήλωναν την πρόθεση να το στηρίξουν με τις δυνάμεις τους.

Ο Γκίλφορντ μάλιστα, έστειλε γράμμα στον υπουργό των Αποικιών και συγγενή του, λόρδο Bathurst τον Οκτώβρη του 1820, προτείνοντας; την Ιθάκη σαν έδρα και ζητώντας την θετική παρέμβασή του. Ο Bathurst ανταποκρίθηκε θετικά, έστειλε τον προσωπικό του Γραμματέα, Sir Frederick Hankey στον Maitland με προφορικές οδηγίες σχετικά. Ο Maitland δεν είδε με ευχαρίστηση την κίνηση, αλλά ο Bathurst ήταν ο ανώτερός του και όφειλε να υπακούσει. Ωστόσο ο Maitland απάντησε με επιστολή στον Bathurst, υποσκάπτοντας στην ουσία τον Γκίλφορντ. Αν και δεχόταν την δημιουργία του Πανεπιστημίου στην Ιθάκη, έγραφε, μεταξύ άλλων, ότι οι ενέργειες το Γκίλφορντ για διορισμό Καθηγητών χωρίς την προηγούμενη έγκριση της Κυβέρνησης θα δημιουργούσε πρόβλημα. Πράγμα που δεν ήταν αληθές, καθώς ο Γκίλφορντ από τον Απρίλιο του 1820 ζητούσε την έγκριση της Κυβέρνησης για την πρόσληψη συγκεκριμένων καθηγητών, και ζητούσε συγνώμη καθώς είχε ήδη συζητήσει με αυτούς για την πρόσληψή τους.

Υπάρχει πολύ παρασκήνιο στον τρόπο που αντιμετωπίζονταν οι ενέργειες του Γκίλφορντ από τον Maitland (Αγγελομάτη-Τσουγγαράκη 1997, σ. 83-86), ο οποίος επιμελώς δεν είχε αναφέρει ότι η μισθοδοσία των διδασκόντων δεν θα γινόταν από την Κυβέρνηση, αλλά από τον ίδιο τον Γκίλφορντ. Ωστόσο οι προσπάθειες του Γκίλφορντ συναντούσαν αντίδραση και από τους ίδιους τους Επτανήσιους.

Εντωμεταξύ, συνεχίζονταν οι δωρεές των Ιθακήσιων, αλλά και Ευρωπαϊκών προσωπικοτήτων για την δημιουργία του Πανεπιστημίου. Για να γίνει αντιληπτό το κλίμα που υπήρχε, αναφέρω εδώ τον Δανό Βασιλιά ο οποίος πρόσφερε πολύτιμα βιβλία για την βιβλιοθήκη του Πανεπιστημίου.

Αν και τυπικά είχε εγκριθεί η ανέγερση του Πανεπιστημίου στην Ιθάκη η επανάσταση των Ελλήνων, οδήγησε στην επιβολή στρατιωτικού νόμου στα Ιόνια Νησιά και καθυστέρησε την έναρξη της Λειτουργίας του.

Στις 14 Μαΐου 1822 με εντολή του Maitland, πάγωσε η αγορά γης στην Ιθάκη για την ανέγερση του Πανεπιστημίου. Μέσα σε όλο το κλίμα αβεβαιότητας, τις παρασκηνιακές ενέργειες και τις μηχαρορραφίες που συνέβαιναν στην πολιτική, οι προσπάθειες του Γκίλφορντ υπονομεύονταν.

3.1.1. Η Έδρα της Ιονίου Ακαδημίας

Στις 6 Μαρτίου 1823 ο Maitland εισηγήθηκε την ανάκληση της απόφασης ίδρυσης του Πανεπιστημίου στην Ιθάκη, υπονοώντας έμμεσα την ίδρυσή του στην Κέρκυρα. Στους υποστηρικτές της Ιθάκης είχαν απομείνει οι Ιθακήσιοι και ο Γκίλφορντ. Πολλοί από τους συνεργάτες του αλλά και πολλοί Επτανήσιοι αντιδρούσαν, και έτσι προκειμένου να λειτουργήσει το Πανεπιστήμιο ο Γκίλφορντ αναδιπλώθηκε.

Τον Μάρτιο του 1823 σε αναφορά του προς τον Αρμοστή ανέφερε τρεις πρακτικούς λόγους (Henderson 1980, σ. 30) για τους οποίους θεωρούσε την Κέρκυρα καταλληλότερη από την Ιθάκη:

1) Η Κέρκυρα είχε Νοσοκομείο. Αυτό ήταν απαραίτητο για την δημιουργία Ιατρικής Σχολής.

2) Η Κέρκυρα είχε Δικαστήριο. Αυτό θα εξυπηρετούσε την δημιουργία Νομικής Σχολής.

3) Η Κέρκυρα είχε έτοιμη την υποδομή σε κτίρια που μπορούσαν με ελάχιστα έξοδα να προσαρμοστούν στις ανάγκες λειτουργίας ενός Πανεπιστημίου.

Επίσημα, η Ιόνιος Ακαδημία συστάθηκε στις 19 Μαΐου 1823 με απόφαση της Ιονίου Βουλής, μετά από πολλά προβλήματα. Τον Μάιο του 1924 μια πολύ περιληπτική απόφαση του Νομοθετικού Σώματος της κυβέρνησης όριζε ότι θα λειτουργούσαν τέσσερις σχολές: Η Θεολογική, η Νομική, η Ιατρική και η Φιλοσοφική. Επίσης όριζε τον αριθμό και τις ονομασίες των Αρχών και των Οργάνων του Πανεπιστημίου, χωρίς όμως να διευκρινίζει τα καθήκοντα και τις υποχρεώσεις τους.

Εντωμεταξύ, ο Γκίλφορντ φρόντιζε για την στελέχωση του Πανεπιστημίου με νέους που είχε συναντήσει στα ταξίδια του στην Ευρώπη. Κάποιοι από αυτούς δίδασκαν σε σχολεία, κάποιοι άλλοι σπούδαζαν. Είχε χορηγήσει υποτροφίες σε πολλούς από αυτούς για να τους εφοδιάσει με το κατάλληλο υπόβαθρο ώστε να είναι σε θέση να διδάξουν στο Πανεπιστήμιο όταν θα έφτανε η στιγμή. Από τους δεκατέσσερις που είχε βοηθήσει ο Γκίλφορντ να σπουδάσουν, δύο τελικά δεν έγινε εφικτό να διδάξουν στην Ιόνιο Ακαδημία.

Αναφέρθηκε ήδη ότι το Πανεπιστήμιο ξεκίνησε την προκαταρκτική του λειτουργία στην ιδιωτική κατοικία που είχε παραχωρηθεί στον Γκίλφορντ. Πολλοί από τους καθηγητές που είχε επιστρατεύσει ο Γκίλφορντ ήταν διατεθειμένοι να διδάξουν σε αυτό το προπαρασκευαστικό στάδιο. Τα μαθήματα που διδάσκονταν ήταν ελληνικά, λατινικά, αγγλικά, ιστορία, ρητορική, μαθηματικά, βοτανική, φιλοσοφία και θεολογία (Henderson 1980, σ. 32).

Ο Κωνσταντίνος Ασώπιος, τον οποίο ο Γκίλφορντ είχε στείλει στο Βερολίνο, το Παρίσι και το Λονδίνο, δίδαξε Αρχαία Ιστορία και λογοτεχνία. Ο Χριστόφορος Φιλητάς που είχε σπουδάσει στην Οξφόρδη, δίδαξε λατινικά και λογοτεχνία, μέχρι το κλείσιμο της Ακαδημίας το 1864. Ο Ιωάννης Καρανδινός που σπούδασε στο Παρίσι δίδαξε μαθηματικά. Ο Ιάκωβος Λουζινιάνος, Κύπριος στην καταγωγή, δίδασκε αγγλική γλώσσα και λογοτεχνία. Ο Αθανάσιος Πολίτης Χημεία. Ο Θεόκλητος Φαρμακίδης δίδασκε θεολογία. Τέλος ο Frank Balfour, σκοτσέζος, κατά μία άποψη ανέλαβε την έδρα της Εβραϊκής και Αραβικής Γλώσσας και συμμετείχε σε αυτά τα προπαρασκευαστικά μαθήματα, κατά μία άλλη άποψη δεν δίδαξε ποτέ στην Ακαδημία.

Τα προπαρασκευαστικά μαθήματα γίνονταν στα Ελληνικά, και στην συνέχεια ορίστηκε και επίσημα με απόφαση της Γερουσίας ως γλώσσα διδασκαλίας η ελληνική Ο ακριβής

αριθμός των Καθηγητών που δίδαξαν σε αυτή την φάση δεν είναι γνωστός με ακρίβεια, άλλοι αναφέρουν οχτώ, άλλοι έξι (Henderson 1980, σ. 32).

Στην επίσημη τελετή των εγκαινίων της Ιονίου Ακαδημίας στις 17 Μαΐου 1824 στα Παλιά Ανάκτορα, απονεμήθηκαν Διδακτορικά διπλώματα σε έξι από τους οκτώ καθηγητές που δεν είχαν πάρει από Ευρωπαϊκά Πανεπιστήμια. Ανακοινώθηκαν οι 47 από τους 150 φοιτητές του προπαρασκευαστικού κύκλου που κρίθηκαν ικανοί να συνεχίσουν τις σπουδές τους, σε αυτούς δόθηκε ο τίτλος του φιλόλογου. Στην συνέχεια ο επίσημος ομιλητής του Πανεπιστημίου, Χριστόφορος Φιλητάς εκφώνησε τον πανηγυρικό λόγο. Ο Γκίλφορντ διέκοψε για να κατευθυνθούν οι παρευρισκόμενοι στον Ναό της Σπηλιωτίσης και να παρακολουθήσουν την δοξολογία και επίκληση του Πατρός των Φώτων (Henderson 1980, σ. 39).

Η απόφαση της Γερουσίας να γίνονται τα μαθήματα στα Ελληνικά, αλλά και η επιμονή του Γκίλφορντ για αποκατάσταση της Ελληνικής ως επιστημονικής και ακαδημαϊκής γλώσσας, δημιούργησε προβλήματα στην λειτουργία της Ακαδημίας. Για παράδειγμα στην Νομική, που άρχισε να λειτουργεί το 1825, η ανυπαρξία νόμων γραμμένων στα ελληνικά και η απουσία βιβλίων δημιούργησε την επιτακτική ανάγκη δημιουργίας συγγραμμάτων. Έπρεπε να επινοηθεί λεξιλόγιο επιστημονικών όρων.

Η αρχή έγινε με την μετάφραση του έργου "περί Μεθόδου Λόγος" του Descartes από τον Νικόλαο Πίκκολο, για να συνεχιστεί με συνέπεια από τους διδάσκοντες στην Ακαδημία με σειρά μεταφράσεων και δημοσίευσης συγγραμμάτων, στα πεδία της Ιατρικής, των φυσικών επιστημών, της φιλοσοφίας, της λογοτεχνίας αλλά και των μαθηματικών. Δημιουργήθηκε έτσι μια μορφή καθαρεύουσας, μιας γλώσσας διαφορετικής από την καθομιλουμένη. Σε όλη την λειτουργία της Ακαδημίας, υπήρξε μόνο μία περίπτωση διδασκαλίας στην δημοτική, από τον Σταματέλο Πυλαρινό, καθηγητή της πειραματικής Φυσικής, με πολύ ενθαρρυντικά αποτελέσματα (Henderson 1980, σ. 45).

Ενδιαφέρον έχει ότι με εξαίρεση την Ιατρική, και ίσως και την Νομική, δεν υπήρχαν στεγανά στις σχολές. Αν κάποιος το επιθυμούσε, μπορούσε να παρακολουθεί μαθήματα σε περισσότερες από μία σχολές. Για παράδειγμα, τον Φιλολογικό κύκλο της Φιλοσοφικής τον παρακολουθούσαν και οι 211 σπουδαστές (Henderson 1980, σ. 56).

Σε αυτά τα πρώτα χρόνια λειτουργίας την Ιόνιας Ακαδημίας υπήρχαν ποσοτικά πολλοί σπουδαστές από τα Ιωάννινα. Διατυπώνεται από μελετητές της Ιστορίας της Ακαδημίας ότι

αυτό είχε ως αποτέλεσμα την στελέχωση της Ελληνικής εκπαίδευσης την πρώτη 25ετία της ύπαρξης του ελεύθερου ελληνικού κράτους (Henderson 1980, σ. 47).

Το επόμενο έτος οι σπουδάζοντες έφτασαν τους 140, το τρίτο έτος δε, τους 211[3], με αναμονή 50 νέων εγγραφών για την επόμενη περίοδο, 1826-1827. Ο Γκίλφορντ στήριζε με προσωπικά του έξοδα περί τους 160, αλλά και τους καθηγητές της Ακαδημίας που είχαν φτάσει τους 14, με την αποχώρηση την προηγούμενη χρονιά των Πίκκολου, Φαρμακίδη και Balfour. Η έδρα του Balfour καταργήθηκε, την έδρα του Πίκολλου ανέλαβε ο Ιωαννίδης, και του Φαρμακίδη ο Κωνσταντίνος Τυπάλδος.

Για το έτος 1825-1826 υπάρχει σύγχυση για το ποιά ήταν τα ονόματα των καθηγητών που κατέλαβαν τις νέες έδρες. Αναφέρονται τα ονόματα των Νικόλαου Μανιάκη (Διεθνές Δίκαιο), Γεώργιου Θεριανού (Ανατομία και Γενική Παθολογία), Στυλιανού Μαράτου (Περιγραφική Ανατομία και Εισαγωγή στην Χειρουργική) και του Gaetano Grassetti (Λατινικά και Φιλολογία) (Γεωργαλά-Πριόβολου 1989). Θεωρείται ότι το 1825 προσλήφθηκαν και οι Leopold Joss και Πασχάλης Καρούσος στην Νομική (Σαλβάνος και Σαλβάνου 1949). Ο Α. Παπαδόπουλος-Βρετός είχε αναλάβει την οργάνωση της Βιβλιοθήκης.

Όπως και να έχει, το Πανεπιστήμιο λειτουργούσε πια με τις τέσσερις σχολές του ενεργές.

Το 1826-1827 φέρεται να διδάξαν στην Ακαδημία δεκαπέντε ή δεκαέξι καθηγητές. Ο Ανδρέας Κάλβος που ετοιμαζόταν να προσχωρήσει στο δυναμικό της το επόμενο ακαδημαϊκό έτος, κάνει λόγο για δεκαέξι.

Στην Θεολογική δίδαξε ο Κωνσταντίνος Τυπάλδος, στην Νομική ο Πασχάλης Καρούσος και ο Νικόλαος Μανιάκης, στην Ιατρική ο Γεώργιος Θεριανός και ο Στυλιανός Μαράτος, Στην Φιλοσοφική υπήρχαν δύο κατευθύνσεις, η επιστημονική και η φιλολογική. Στην επιστημονική ο Στυλιανός Σπάθης δίδασκε Βοτανική, ο Αθανάσιος Πολίτης Χημεία, ο Ιωάννης Καρανδινός Μαθηματικά, ο Γεώργιος Ιωαννίδης Φιλοσοφία,, και ο Ιωάννης Αριστείδης Εκκλησιαστική Μουσική. Στο φιλολογικό τμήμα ο Αριστείδης δίδασκε Ρητορική, ο Κωνσταντίνος Ασώπιος Ελληνική Φιλολογία, Ποιητική Σύνθεση και Αρχαία Ελληνική και Ιστορία, ο Jacob Lusignanos αγγλική και λογοτεχνία, ο Gaetano Grassetti Λατινική και Αρχαία Ρωμαϊκή Ιστορία, ο Νικόλαος Μανιάκης Νεότερη Ιστορία. Τέλος ο Κωνσταντίνος

[3] Σύμφωνα με τον Καρανδινό που ήταν έφορος. Άλλοι κάνουν λόγο για 240.

Σακελλαρόπουλος δίδασκε μια ευρεία γκάμα θεμάτων, Αρχαιολογία, Ελληνική Πρωτοϊστορία, Αρχαίο Ελληνικό Πολιτισμό, Παλαιογραφία, Διπλωματική και Επιγραφική.

Σε αυτό το έτος δεν γίνονταν μαθήματα φυσικής και γεωπονικής, προγραμματίζονταν όμως για το επόμενο.

Πάντως για το τρέχον έτος, οι μισθοί των νέων καθηγητών είχαν καταβληθεί από τον Γκίλφορντ, ο οποίος όμως πίεζε για αύξηση της επιχορήγησης από την κυβέρνηση. Λογικό από μέρους του, καθώς η χρηματοδότηση εξακολουθούσε να επιβαρύνει σε μεγάλο βαθμό τον ίδιο. Για το 1827-1828 η χρηματοδότηση αυξήθηκε από 16000 τάληρα σε 18000.

Δείκτης των όσων προγραμμάτιζε ο Γκίλφορντ να ακολουθήσουν, ήταν η απονομή διδακτορικών στους μελλοντικούς καθηγητές του Πανεπιστημίου.

Απονεμήθηκε δίπλωμα στον Ιωάννη Σορδίνα, που δεν ήταν μέλος του Πανεπιστημίου, και δεν φαίνεται να δίδαξε ποτέ στην Ακαδημία.

Ο Ιωάννης Τουρλινός παρέλαβε το διδακτορικό του δίπλωμα τον Φεβρουάριο του 1827, προοριζόταν για την έδρα της Γεωπονικής. Τον Ιούνιο του 1827 απονεμήθηκαν διδακτορικά διπλώματα στον Σταματέλο Πυλαρινό, που προοριζόταν για την έδρα της Φυσικής, στον Leopold Joss, που προοριζόταν για την Νομική, και στον Ανδρέα Κάλβο τον οποίο ο Γκίλφορντ προόριζε για την έδρα της Συγκριτικής Φιλολογίας. Εκτός αυτού ο Γκίλφορντ είχε ορίσει υπεύθυνο των χειρογράφων και των βιβλίων του τον Κάλβο, δηλαδή ανεπίσημα ήταν ο βιβλιοθηκάριος του, και είχε επίσης αναλάβει και διδακτικό έργο από το 1826. Ο Κάλβος δίδαξε στην Ιόνιο Ακαδημία ανεπίσημα το 1826-1827, το 1836-1837 και το 1840-1841.

Αυτά τα σχέδια δεν πραγματοποιήθηκαν ποτέ, καθώς στις 14 Οκτώβρη 1827 ο Γκίλφορντ πέθανε.

3.1.2. Η μετά Γκίλφορντ Περίοδος (1827-1841)

Ο οικονομικός αντίκτυπος της απουσίας της οικονομικής στήριξης από τον Γκίλφορντ έγινε άμεσα αισθητός. Η λειτουργία της Ιατρικής Σχολής ανεστάλη για να μην ξαναλειτουργήσει πριν το 1844. Καταργήθηκαν οι έδρες της Βοτανικής και της Γεωπονικής και μετά από αυτό ακολούθησαν παραιτήσεις καθηγητών. Με τον σταδιακό περιορισμό των εδρών, απόμειναν τελικά εννέα έδρες.

Πολύ σημαντικό πλήγμα για την Ιόνιο Ακαδημία ήταν η απώλεια της προσωπικής συλλογής βιβλίων του Γκίλφορντ. Κάποια κενά στην διαθήκη του και η αδιαφορία με την

οποία αντιμετώπιζε την Ιόνιο Ακαδημία η Αρμοστεία είχαν ως αποτέλεσμα την απομάκρυνση των βιβλίων. Ο Γκίλφορντ είχε θέσει στην διαθήκη του, σαν όρο παραμονής των βιβλίων που είχε δωρίσει στο Πανεπιστήμιο, την χρηματοδότηση με 3500 στερλίνες τον χρόνο από την Κυβέρνηση, και επιπλέον, για τα επόμενα τέσσερα χρόνια την χρηματοδότηση των σπουδαστών που έπαιρναν υποτροφία από αυτόν. Αυτοί οι όροι αγνοήθηκαν όμως από την Κυβέρνηση.

Ορίστηκε τριμελής επιτροπή της οποίας ο πρόεδρος ήταν κάποιο μέλος της Γερουσίας. Αυτή τελικά αποφάσισε να καλύψει τα έξοδα είκοσι μόνο σπουδαστών που χρηματοδοτούσε ο Γκίλφορντ, αλλά ούτε αυτό πραγματοποιούνταν πολλές φορές.

Από το 1827 ως το 1835 η Ακαδημία δεν έχει να δείξει κάτι ιδιαίτερο, πέρα από την παραγωγή συγγραμμάτων υπό την φωτισμένη καθοδήγηση του Ιωάννη Καρανδινού. Μεταφράστηκαν στα Ελληνικά πολλά βιβλία, ενδεικτικά αναφέρω Αριθμητική, Γεωμετρία και Τριγωνομετρία, Λατινική Γραμματική, Ελληνική Γραμματική. Το σημαντικότερο έργο της περιόδου ήταν το Συντακτικό της Αρχαίας Ελληνικής του Νεόφυτου Βάμβα (Henderson 1980, σ. 83).

Με την αρμοστεία του Howard Douglas[4] (1835-1841), η Ιόνιος Ακαδημία πήρε νέα πνοή. Σε ψήφισμά της Γερουσίας το 1835 προκαθορίζονται νέα προγράμματα σπουδών. Αυτό έδωσε μια νέα ώθηση στην Θεολογική, την Νομική και την Φιλοσοφία (τις φυσικές επιστήμες με σημερινούς όρους). Το 1837 δημοσιεύτηκε νέος κανονισμός σπουδών, που διευθετούσε αποτελεσματικά τις σπουδές, επιλύοντας ζητήματα σπουδών, οργάνωσης και πειθαρχίας (Henderson 1980, σ. 96).

Υπήρχε διαρκής αξιολόγηση των σπουδαστών. Θυμίζει το σημερινό σύστημα με τις εξετάσεις ανά ακαδημαϊκό έτος που διεξάγονταν τον Ιούλιο, με πρόβλεψη για επαναληπτικές για όσους δεν απέδιδαν τα προβλεπόμενα σε αυτές. Σε περίπτωση δεύτερης αποτυχίας, ο σπουδαστής έπρεπε να επαναλάβει το έτος (Henderson 1980, σ. 105). Η φοίτηση στις σχολές ήταν τριετής, με εξαίρεση την Νομική όπου ήταν τέσσερα χρόνια, και υπήρχε πρόβλεψη για απόκτηση πτυχίου αλλά και διδακτορικού, χωρίς να είναι σαφές που ήταν η διαφορά τους (Henderson 1980, σ. 102).

[4] Στην παραλιακό δρόμο της Γαρίτσας υπάρχει μνημείο προς τιμή του που οι Κερκυραίοι αποκαλούν "η κολόνα του Ντούγκλα".

Υπήρχε η σκέψη να μεταφερθεί η έδρα της Ακαδημίας από το Παλάτι του Αρμοστή, σε κάποιον καταλληλότερο χώρο. Ενώ από το 1832 ακόμα γίνονταν κάποιες κινήσεις, χωρίς αποτέλεσμα όμως. Αυτές εντάθηκαν το 1835, όταν είχε αναλάβει αρμοστής ο Douglas. Το 1838 ανακοίνωσε ότι ο Στρατώνας Grimani (εικόνα 3-1) θα παραχωρούνταν για την στέγαση της Ιονίου Ακαδημίας αλλά "και των λοιπών σχολείων" (Henderson 1980, σ. 109). Τα λοιπά σχολεία ήταν το γυμνάσιο, το λύκειο και το εκκλησιαστικό σεμινάριο (Σαλβάνος και Σαλβάνου 1949). Χωρίς να είναι επιβεβαιωμένο, φαίνεται να λειτούργησαν για πρώτη φορά στους στρατώνες, το μεν Γυμνάσιο το 1840, η δε Ακαδημία το 1841.

Εικόνα 3-1. Η Ιόνιος Ακαδημία (1841-1865).

3.1.3. Η Περίοδος 1841-1865

Φαίνεται όμως ότι το Πανεπιστήμιο είχε εισέλθει στην φάση της παρακμής του. Τον Ιανουάριο του 1841 δημοσιεύτηκε ένας νέος κανονισμός που δεν εφαρμόστηκε ποτέ, και στα

τέλη Νοέμβρη - αρχές Δεκέμβρη του ίδιου έτους δημοσιεύτηκε άλλος ένας που διέφερε σημαντικά από αυτόν του 1837. Η τελική απόρροια, ήταν η κατάργηση της Σχολής Πολιτικών Μηχανικών με την αντικατάστασή της από την Σχολή Φαρμακευτικής, που ήταν διετούς φοίτησης (Henderson 1980, σ. 112). Οι εν ενεργεία σχολές που απομένουν είναι η Θεολογική, η Νομική και η Αρχαία Ελληνική Γλώσσα και Γραμματεία. Ενέργειες που γίνονταν στην συνέχεια για την πρόσληψη νέων καθηγητών δεν είχαν το προσδοκώμενο αποτέλεσμα, καθώς απέτυχαν να αντιστρέψουν το κλίμα που είχε δημιουργηθεί, ήταν διστακτικές αλλαγές και υποδηλώνουν υποχώρηση του επιπέδου της Ακαδημίας (Henderson 1980, σ. 122-124).

Για το Ακαδημαϊκό έτος 1843-44 έγιναν διορισμοί καθηγητών που είχαν ως αποτέλεσμα την ανάσχεση της παρακμής, για μερικές από τις σχολές όμως και όχι ολιστικά. Στην Φιλοσοφική Σχολή τοποθετήθηκε Πρόεδρος ο Francesco Oriolli. Διορίστηκαν επίσης οι Φώτιος Καντάς (Θεολογική), Ιερώνυμος Σαντόριος (Νομική), Άγγελος Κογεβίνας (Ιατρική), Σπυρίδων Αρβανιτάκης (Ιατρική), Αθανάσιος Πολίτης (χημικός[5]), Κωνσταντίνος Ζαβιτσιάνος (Ιατρική), και Χριστόφορος Φιλητάς (Ελληνιστής).

Οι τρεις προσλήψεις για την σχολή της Ιατρικής (που μετονομάζεται σε Ιατρο-Χειρουργική, Φαρμακευτική και Μαιευτική Σχολή) υποδηλώνουν την επανέναρξη της λειτουργίας της[6]. Η πρόσληψη του Σαντόριου όμως δεν υποδηλώνει την ενίσχυση της Νομικής αλλά την συρρίκνωσή της, καθώς καθηγητές των προηγούμενων ετών (Ρόζος και Joss) δεν εμφανίζονται ως διδάσκοντες.

Η περίοδος 1850-1860 έχει να παρουσιάσει τρεις σημαντικές προσωπικότητες που επηρέασαν την Ιόνιο Ακαδημία. Ο Χαράλαμπος Τυπάλδος-Πρετεντέρης που έγινε καθηγητής Παθολογίας το 1851, ο Χριστόφορος Φιλητάς, καθηγητής της Ελληνικής και Λατινικής Φιλολογίας, και ο Πέτρος Βράιλας-Αρμένης καθηγητής Φιλοσοφίας.

Ο Τυπάλδος σπούδασε στην Ιατρική στην Γαλλία από όπου αποφοίτησε ως διακεκριμένος σπουδαστής πολλά υποσχόμενος για το μέλλον. Στην Διδακτορική του διατριβή το 1846 δημοσίευσε εργασία για τον τυφοειδή πυρετό. Θα μπορούσε να σταδιοδρομήσει στην Γαλλία, αφού είχε διοριστεί στην Ιατρική Σχολή του Παρισιού, επέστρεψε όμως στην πατρίδα του το

[5] Πιθανότατα πρόκειται για τον χημικό και όχι για τον θεολόγο (Hendrson 1980, σ. 119).

[6] Χωρίς να αναφέρω πολλές λεπτομέρειες για την δομή της Ιατρικής Σχολής, υπήρχε και ένα μονοετές πρόγραμμα σπουδών που ονομαζόταν "Κατωτέρα Χειρουργία".

1847. Ασχολήθηκε με τα επιδημικά νοσήματα. και το 1850 αντιμετώπισε την επιδημία χολέρας στην Κεφαλονιά. Δίδαξε το επόμενο έτος στην Ιόνιο Ακαδημία σε ηλικία 29 ετών. Το 1864, με το κλείσιμο της Ιονίου Ακαδημίας να διαφαίνεται, διορίστηκε καθηγητής στο Εθνικό Πανεπιστήμιο στην Αθήνα[7]. Εκλέχτηκε δύο φορές βουλευτής Κεφαλονιάς στην Ιόνιο Βουλή και εργάστηκε για την ένωση των Επτανήσων με την Ελλάδα.

Ο Χριστόφορος Φιλητάς, από τους πρώτους προστατευόμενους του Γκίλφορντ, πήγε στην Νάπολι όπου δίδασκε ως δάσκαλος και σπούδασε Ιατρική. Μετά τις σπουδές του πήγε στην Τεργέστη και στην συνέχεια στην Αγγλία. Το 1820 έγινε Διευθυντής του Λυκείου Κέρκυρας και το 1829 του Σχολείου Ζακύνθου. Έχει να παρουσιάσει πλούσιο συγγραφικό και μεταφραστικό έργο. Στα σημαντικότερα έργα του συγκαταλέγονται η "Γραμματική της ελληνικής Γλώσσας" και τα "Φιλολογικά πάρεργα"(Χριστόφορος Φιλητάς 2017). Η επιγραφή στην Κολόνα του Ντούγκλα συντάχθηκε από αυτόν. Διορίστηκε καθηγητής της Ελληνικής Φιλολογίας στο Πανεπιστήμιο Αθηνών το 1866, αλλά το επόμενο έτος πέθανε.

Ο Πέτρος Βράιλας-Αρμένης σπούδασε αρχικά στην Ιόνιο Ακαδημία και στην συνέχεια στην Μπολόνια, την Γενεύη και τέλος στο Παρίσι Νομικά και Φιλοσοφική. Ίδρυσε την Αναγνωστική Εταιρεία Κέρκυρας. Η πρώτη του διδασκαλία στην Ιόνιο Ακαδημία, ήταν μια προσπάθεια να εξηγήσει τι είναι η Φιλοσοφία και γιατί πρέπει να διδάσκεται (Henderson 1980, σ. 133). Το 1862 διορίστηκε στο Εθνικό Πανεπιστήμιο της Αθήνας, αλλά δεν αποδέχτηκε τον διορισμό. Το 1848 ίδρυσε το κόμμα των Συνταγματικών που είχε σα κύριο στόχο την Ένωση των Επτανήσων με την Ελλάδα. Μετά την Ένωση των Επτανήσων με την Ελλάδα εκπροσωπούσε την Κέρκυρα στην Ελληνική Βουλή, και διατέλεσε υπουργός εξωτερικών της Κυβέρνησης Κουμουνδούρου (Βραΐλας Αρμένης n.d.).

3.1.4. Τέλος μιας Εποχής: Το Κλείσιμο της Ιονίου Ακαδημίας το 1865.

Αν από την δεκαετία του 1850 υπήρχε κλίμα αβεβαιότητας για το μέλλον της Ιονίου Ακαδημίας, το 1862 εκδηλώθηκαν διαφωνίες που το 1863 μετατράπηκαν σε εικόνα εγκατάλειψης. Στις 29 Μάρτη του 1864 οι μεγάλες δυνάμεις τις εποχής, Αγγλία, Γαλλία και Ρωσία, υπέγραψαν σύμφωνο με το οποίο παραχωρούνταν τα Επτάνησα στο Βασίλειο της

[7] Εθνικό και Καποδιστριακό Πανεπιστήμιο

Ελλάδας. Μετά την Ένωση των Επτανήσων με την Ελλάδα στις 21 Μάιου του 1864, το κλίμα έγινε πιο δύσκολο, και η Ακαδημία έκλεισε τελικά το 1865.

Από τους τελευταίους δεκαοκτώ καθηγητές που διδάξαν, οι δύο μετακινήθηκαν στην Αθήνα, ο Χαράλαμπος Τυπάλδος-Πρετεντέρης της Ιατρικής και ο Όθων Σ. Πυλαρινός της Νομικής, για να στελεχώσουν το Πανεπιστήμιο Αθηνών (Henderson 1980, σ. 146).

Στους παράγοντες που οδήγησαν τελικά στο κλείσιμο της, θα πρέπει να ληφθεί υπόψη και η καχυποψία και αντιπάθεια με την οποία αντιμετώπιζε η Ελλάδα την αρμοστεία των Άγγλων στα Επτάνησα. Το 1860 οι Ιόνιος Ακαδημία είχε συνολικά 63 φοιτητές. Οι γονείς προτιμούσαν να στέλνουν τα παιδιά τους στην Αθήνα για σπουδές.

Δεν ήταν κάτι ξαφνικό. Στην Ιόνιο Βουλή η Γερουσία συζητούσε πολύ νωρίτερα το ενδεχόμενο, λόγω της παρακμής από τον ανταγωνισμό που είχε από το Πανεπιστήμιο Αθηνών. Η λήψη απόφασης αναβλήθηκε βέβαια, και υπήρξαν και αντίθετες απόψεις, αλλά η αρχή είχε γίνει.

Στις 27 Δεκέμβρη 1865, ο Στέφανος Παδοβάς που είχε φοιτήσει Νομική στην Ιόνιο Ακαδημία, παραιτήθηκε από βουλευτής των Ριζοσπαστών Βουλευτών θεωρώντας την παρουσία του στην Βουλή ως άσκοπη, και ότι τα Επτάνησα, ιδίως η Κέρκυρα "σφαγιάζονταν" από την Ελλάδα (Κουρκουμέλης, Η Εκπαίδευση στην Κέρκυρα κατά τη διάρκεια της βρετανικής προστασίας (1816-1864) 2002).

Για να γίνουν κατανοητές οι συνθήκες που υπήρχαν, οι Ριζοσπάστες Βουλευτές της Ζακύνθου θεωρούσαν την Αθήνα πρωτεύουσα του Ελληνισμού, και την Κέρκυρα έδρα της Βρετανικής Προστασίας.

Η συνεισφορά της Ιονίου Ακαδημίας τα χρόνια λειτουργίας της, από το 1824 ως το 1864 στην αναβίωση της παιδείας στην Ελλάδα είναι ανεκτίμητη. Από αυτήν αποφοίτησαν άνθρωποι που στελέχωσαν αργότερα τα σχολεία σε όλη την Ελλάδα.

3.1.5. Ο Βομβαρδισμός της Ιονίου Ακαδημίας και η Ανακαίνισή της

Στις 13 Σεπτέμβρη 1943 τα Γερμανικά Αεροπλάνα του Χίτλερ βομβάρδισαν την Κέρκυρα. Στους στόχους συμπεριλαμβάνονταν η Ιόνιος Βουλή, το Δημοτικό θέατρο και η Ιόνιος Ακαδημία, στην οποία στεγαζόταν τότε η Δημόσια Βιβλιοθήκη.

Εικόνα 3-2 Η Ιόνιος Ακαδημία το 1980, πριν την ανακαίνισή της

Το Δημοτικό θέατρο της Κέρκυρας κατεδαφίστηκε στις 31 Μαΐου του 1952 επί Δημαρχίας Σταμάτη Δεσύλλα (Municipal Theatre of Corfu 2018), καθώς αποφασίστηκε ότι δεν είχε καμία πολιτιστική αξία. Ευτυχώς η Ιόνιος Ακαδημία δεν είχε την ίδια τύχη.

Ανακαινίστηκε το 1994 και σήμερα στεγάζει διοικητικές υπηρεσίες του Πανεπιστημίου. Στο αμφιθέατρο γίνονται ορκωμοσίες, ομιλίες, διαλέξεις, μουσικές εκδηλώσεις αλλά και προβολές οπτικοακουστικού υλικού.

3.1.6. Η Ίδρυση του Ιονίου Πανεπιστημίου στην σύγχρονη εποχή

Το 1984, ιδρύθηκε εκ νέου το Ιόνιο πανεπιστήμιο που λειτούργησε για πρώτη φορά το 1985. Λειτουργούν σε αυτό τρεις σχολές:

-Σχολή Ιστορίας & Μετάφρασης - Διερμηνείας

Λειτουργούν σε αυτήν δύο τμήματα. Το 1985 έκανε την έναρξη λειτουργίας του το Τμήμα Ιστορίας, Το 1986 το τμήμα Ξένων Γλωσσών, Μετάφρασης και Διερμηνείας.

-Σχολή Επιστήμης της Πληροφορίας & Πληροφορικής

Από το 1993 λειτουργεί το Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας και Μουσειολογίας. Το 2004 έκανε την έναρξη λειτουργίας του και το Τμήμα Πληροφορικής

-Σχολή Μουσικής και Οπτικοακουστικών Τεχνών

Από το 1992 λειτουργεί το Τμήμα Μουσικών Σπουδών. Το 2004 προστέθηκε το Τμήμα Τεχνών Ήχου και Εικόνας.

Για τις δύο πρώτες σχολές δεν μπορώ να εκφέρω άποψη, αλλά θα σταθώ σύντομα στην τρίτη. Τόσο το Τμήμα Μουσικών Σπουδών όσο και το Τμήμα Τεχνών Ήχου και Εικόνας είναι μοναδικά στο είδος τους για την Ελλάδα, και ίσως και παγκόσμια. Ειδικότερα, για το μεταπτυχιακό του Τμήματος Τεχνών Ήχου και Εικόνας, "Οπτικοακουστικές Τέχνες στην Ψηφιακή Εποχή" μπορώ να έχω προσωπική άποψη.

Συνδυάζει ένα τεράστιο εύρος πεδίων, που είναι βέβαια απαιτητικό, καθώς απαιτεί ευρύ υπόβαθρο γνώσεων, αλλά είναι και εξαιρετικά ενδιαφέρον και πολύπλευρο. Η γνώμη μου είναι ότι ο χαρακτήρας της σχολής δεν πρέπει να αλλάξει, δεν πρέπει να πέσει το βάρος στις καλές τέχνες αλλά ούτε στην πληροφορική. Πρέπει να κρατήσει αυτόν τον ιδιαίτερο χαρακτήρα και να παραμείνει όπως είναι.

Κλείνω αυτό το κεφάλαιο με κάτι που ίσως φανεί αστείο. Σε συζητήσεις με γονείς των οποίων τα παιδιά εκδηλώνουν ενδιαφέρον για την σχολή, και σε υποψήφιους μεταπτυχιακούς σπουδαστές πρέπει να εξηγώ ότι δεν πρόκειται για σχολή ηχοληψίας, ούτε για σχολή ζωγραφικής-γλυπτικής. Ερωτήσεις για την σχέση του Τμήματος με την Πληροφορική δεν είχα, να πω την αλήθεια. Όταν διάβασα το πρόγραμμα του μεταπτυχιακού, είχα την εντύπωση ότι δημιουργήθηκε για να καλύψει όλα όσα με ενδιαφέρουν.

ΚΕΦΑΛΑΙΟ 4. Η Υλοποίηση

Τα απολύτως απαραίτητα στάδια για την υλοποίηση ενός τέτοιου έργου είναι δύο. Υπήρξαν επιπλέον βήματα όμως, τα οποία θα αναπτυχθούν παρακάτω. Τα κύρια βήματα είναι η δημιουργία των μοντέλων στο Blender και η εισαγωγή και παραμετροποίηση της εφαρμογής στοUnity.

Αναγκαστικά, από εδώ και πέρα η εργασία περιέχει αγγλική ορολογία.

Η διαδικασία της ανάπτυξης ενός μαθηματικού μοντέλου για την αναπαράσταση μιας οποιασδήποτε τρισδιάστατης επιφάνειας με χρήση εξειδικευμένου λογισμικού, λέγεται 3D Modeling. Η δημιουργία του τρισδιάστατου μοντέλου είναι το πρώτο βήμα που απαιτείται.

Υπάρχουν αρκετά προγράμματα σε αυτή την κατηγορία. Οι απόψεις για το ποιό είναι το καλύτερο είναι καθαρά προσωπικές. Άποψή μου είναι ότι δεν υπάρχει καλύτερο και χειρότερο, υπάρχει αυτό στο οποίο έχει αφιερωθεί χρόνος και υπάρχει η γνώση του και η ευχέρεια της δημιουργίας. Για να στηρίξω αυτή την γνώμη, αποπειράθηκα στο παρελθόν να δουλέψω με άλλα προγράμματα όπως το 3DSMax, το Maya και το Cinema 4D. Δεν τα κατάφερα, επέστρεψα στο Blender. Αυτά τα εργαλεία απαιτούν πολύ χρόνο και κόπο για την εκμάθησή τους, η ενασχόληση με περισσότερα από ένα είναι σχεδόν αδύνατη.

Επέλεξα το Blender (Blender n.d.) καθώς με αυτό ασχολούμαι εδώ και παραπάνω από μια δεκαετία. Στα πλεονεκτήματά του είναι ότι ως πρόγραμμα ανοιχτού λογισμικού έχει μηδενικό κόστος αγοράς, έχει μια πολύ ενεργή βάση που το εξελίσσει εδώ και πολλά χρόνια, και η κοινότητα του είναι πολύ δυναμική. Υπάρχουν πάντα άνθρωποι που έχουν διάθεση να βοηθήσουν σε προβλήματα και απορίες που μπορεί να προκύψουν. Και υπάρχει πολύ εκπαιδευτικό υλικό, από βιβλία μέχρι βίντεο στο youtube , τα οποία ποικίλλουν σε ποιότητα, αλλά υπάρχει πάντα η λύση της παρακολούθησης κάποιου online course έναντι αντίτιμου.

Η πιο πρόσφατη έκδοση είναι η 2.79b που εκδόθηκε στις 12 Σεπτέμβρη του 2017. Δεν είναι συνηθισμένο για την ομάδα που εργάζεται πάνω στο πρόγραμμα να καθυστερεί τόσο μεταξύ νέων εκδόσεων, το Blender 2.8 όμως θα έχει ριζικές αλλαγές. Θα ενσωματωθεί μια νέα μηχανή γραφικών με απόδοση πραγματικού χρόνου (Blender (EEVOO) n.d.)

Για την εμψύχωση του μοντέλου χρησιμοποιήθηκε η μηχανή παιχνιδιών Unity (Unity 2018). Δεν εμπίπτει στην κατηγορία του ανοιχτού λογισμικού, αλλά διατίθεται δωρεάν έκδοση που είναι πλήρως λειτουργική. Έχει και αυτό μεγάλη βάση χρηστών και developers

που είναι έτοιμοι να βοηθήσουν στην επίλυση απορίων και προβλημάτων, και υπάρχει εκπαιδευτικό υλικό διαθέσιμο, τόσο στο youtube όσο και επί αντιτίμου. Κυκλοφορεί για όλες τις πλατφόρμες - Windows, Linux, MacOS, όμως η έκδοση για Linux δεν είναι ώριμη για σοβαρή χρήση. Για την δημιουργία του τελικού αποτελέσματος χρησιμοποιήθηκε το λειτουργικό σύστημα των Windows και η έκδοση 2018.2.3f1. Η τελική επιλογή ήταν η εξαγωγή σε WebGL, καθώς επιτρέπει την λειτουργία σε όλες τις πλατφόρμες με χρήση πλοηγητή, είτε Mozilla Firefox είτε Google Chrome. Δοκίμασα και τον Safari 7 σε MacOSX, δεν υποστηρίζει WebGL και δεν είχα την δυνατότητα να δοκιμάσω νεότερους.

Για το Unity 2018.2 (+) η τεκμηρίωση του Unity δηλώνει ότι το WebGL λειτουργεί για Safari 11 (+) και Microsoft Edge 16 (+) (Unity_Technologies 2018), δεν είχα όμως την δυνατότητα να το επιβεβαιώσω.

Για την δημιουργία και επεξεργασία των textures χρησιμοποιήθηκε το gimp 2.8.22 που είναι και αυτό ανοιχτό λογισμικό (GIMP - GNU Image Manipuplation Program 2018).

Για την δημιουργία επιπλέον texture maps όπου απαιτούνταν (normal, displacement, specularity, occlusion) χρησιμοποιήθηκε το Crazybump.

Για την δημιουργία της προτομής του Καποδίστρια και του λέοντα στην είσοδο, χρησιμοποιήθηκε η τεχνική της φωτογραμμομετρίας, με το Photoscan της Agisoft (Agisoft Photoscan 2018).

ΚΕΦΑΛΑΙΟ 5. 3D Μοντελοποίηση - Blender

Το Blender διαθέτει ενσωματωμένες δύο render engines, το Blender Render, την παλιά μηχανή που δεν αναπτύσσεται πια, και το Cycles Render, που βασίζεται σε φυσικές ιδιότητες και έχει πολύ ρεαλιστικά αποτελέσματα. Χρησιμοποιήθηκε το Cycles Render, αν και δημιούργησε προβλήματα αργότερα, κατά την εισαγωγή των αντικειμένων στο Unity, καθώς δεν εισάγονταν σωστά τα Materials με τους Shader τους. Το Unity έχει το δικό του Render Engine και τα αποτελέσματα ήταν διαφορετικά από το επιθυμητό.

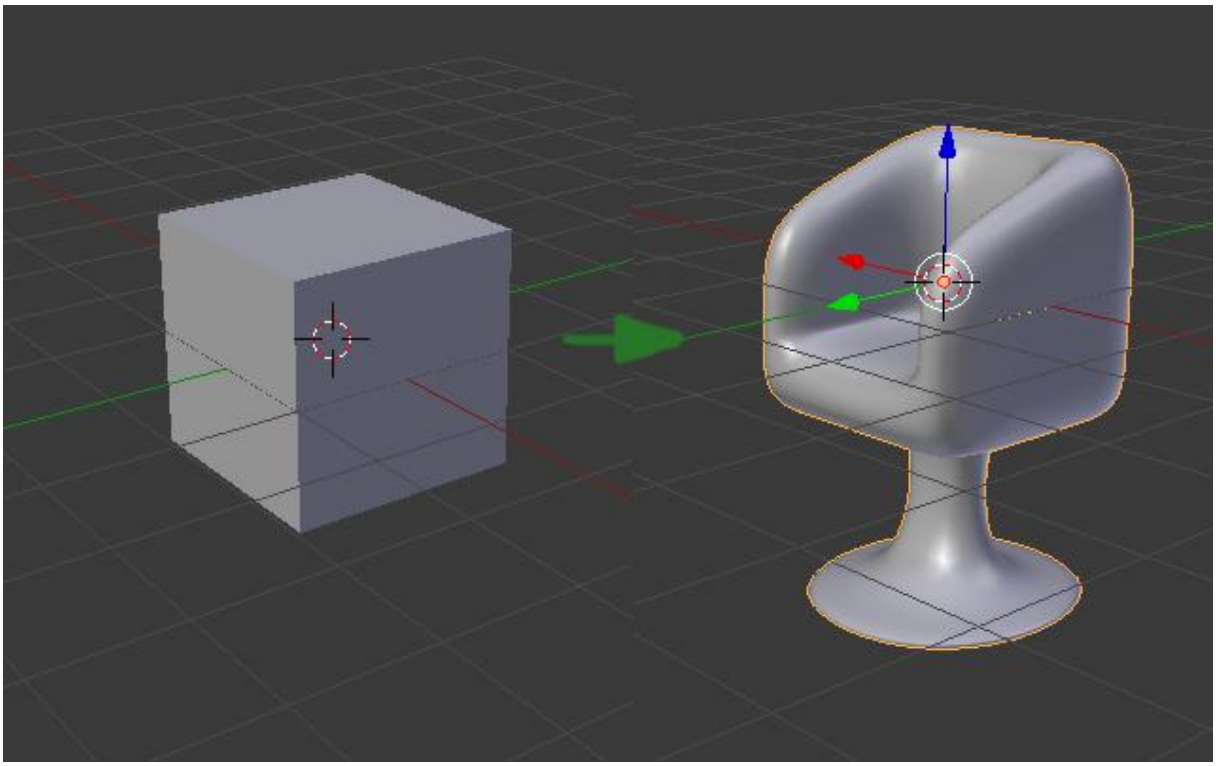
5.1. Βασικές αρχές

Υπάρχουν κάποιες πολύ βασικές αρχές που πρέπει να ληφθούν υπόψη κατά την δημιουργία ενός μοντέλου. Ποιά τεχνική δημιουργίας θα χρησιμοποιηθεί; Το πλήθος των vertices-edges-faces ποιό θα είναι; τα normals των faces έχουν τον σωστό προσανατολισμό; Τι υλικό θα εφαρμοστεί; Πως θα φαίνεται;

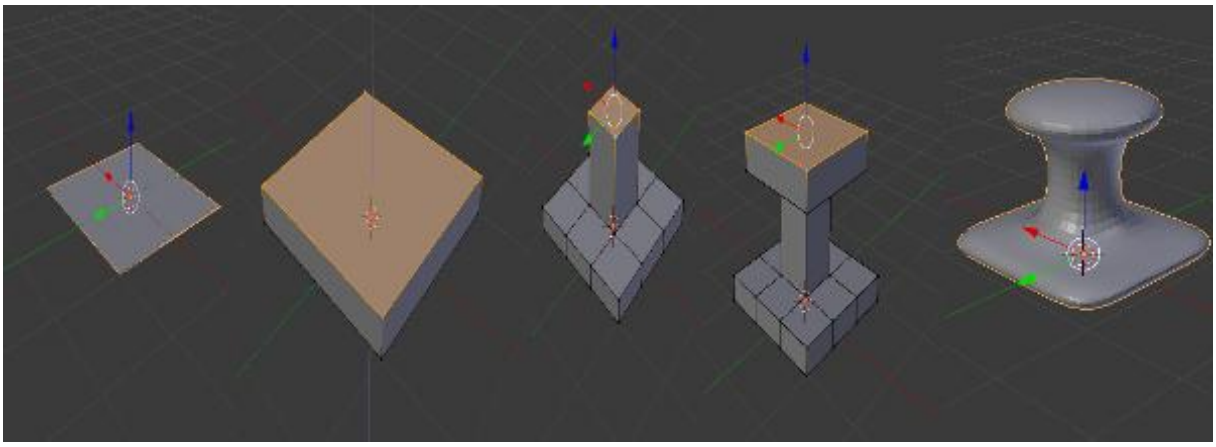
5.1.1. Τεχνικές Μοντελοποίησης

Υπάρχουν διάφορες τεχνικές μοντελοποίησης, οι βασικότερες είναι το box modeling και το mesh modeling. Στην πρώτη περίπτωση χρησιμοποιείται ένα απλό τρισδιάστατο αντικείμενο[8], όπως ένας κύβος, το οποίο παίρνει το τελικό του σχήμα με εφαρμογή μετασχηματισμών (εικόνα 5-1). Η δεύτερη τεχνική, την οποία κατά κύριο λόγο χρησιμοποίησα, ξεκινά από την δημιουργία ενός επιπέδου, το οποίο με την προσθήκη extrusions, loop cuts και μετασχηματισμών καταλήγει στο επιθυμητό αποτέλεσμα (εικόνα 5-2). Και στις δύο περιπτώσεις το αποτέλεσμα είναι το ίδιο, για παράδειγμα η δημιουργία μιας καρέκλας, ενός ποτηριού, ενός αυτοκινήτου, ενός ανθρώπινου προσώπου. Η επιλογή είναι μάλλον θέμα προτίμησης, πόσο άνετα νιώθει ο χρήστης με την τεχνική.

[8] Από τα έτοιμα που έχουν όλα τα 3D προγράμματα, ονομάζονται primitives.



Εικόνα 5-1. Box Modeling

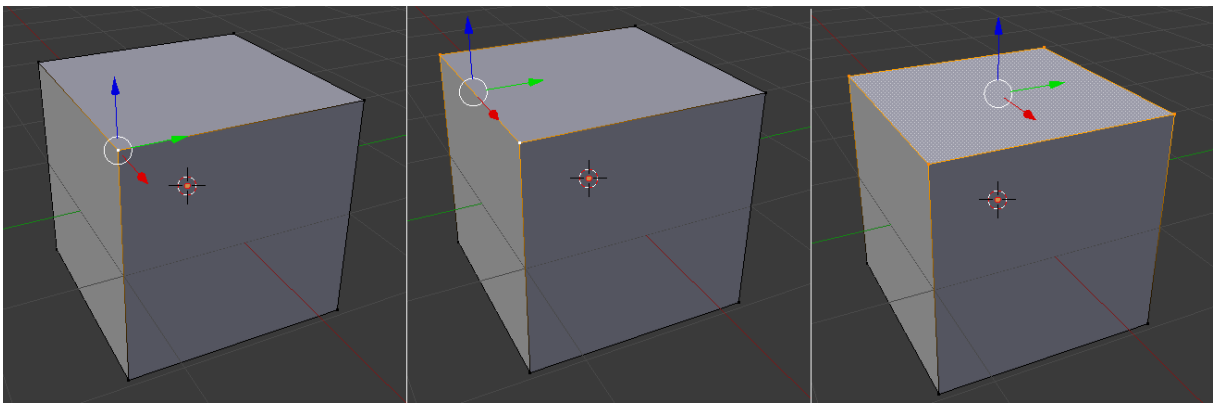


Εικόνα 5-2. Mesh Modeling

5.1.2. Vertices - Edges - Faces.

Ένα vertex είναι ένα σημείο στον χώρο. Ένα edge είναι δύο vertices που ενώνονται για να σχηματίσουν μία ευθεία. Τέσσερα σημεία που ενώνονται για να σχηματίσουν ένα επίπεδο που γεμίζεται σχηματίζουν ένα face. Περισσότερα faces ενωμένα σχηματίζουν ένα τρισδιάστατο αντικείμενο. Στην εικόνα 5-3 υπάρχει ένας κύβος, που έχει επιλεγμένο ένα

vertex, ένα edge και ένα face αντίστοιχα. Να σημειωθεί ότι τα faces έχουν δύο διαστάσεις, η τρίτη διάσταση είναι μηδενική.

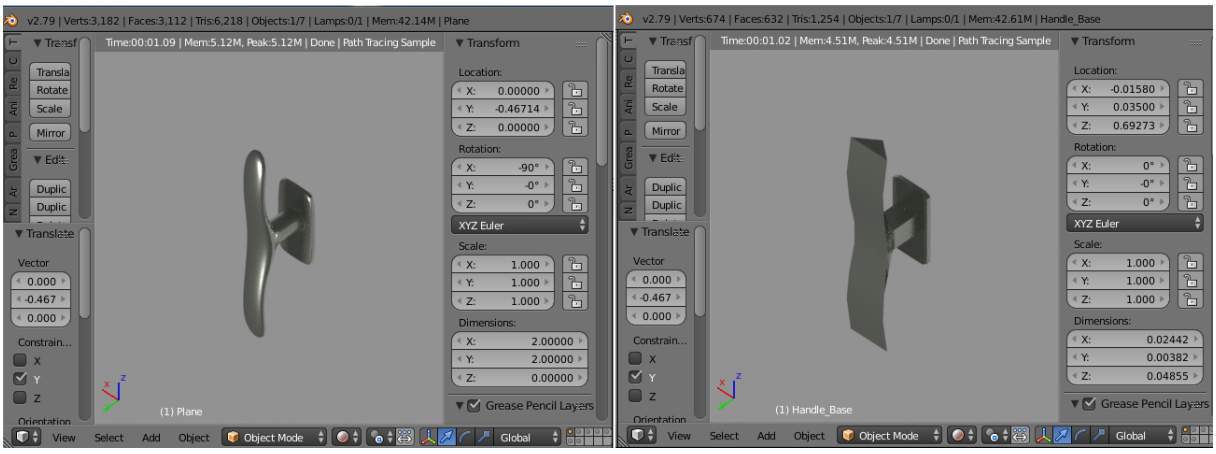


Εικόνα 5-3. 1 vertex, 1 edge, 1 face επιλεγμένα (ο κύκλος δείχνει τον γεωμετρικό τόπο για το καθένα)

Να αναφερθεί εδώ μια λεπτομέρεια. Ένα επίπεδο ορίζεται από τρία σημεία. Τα 3D προγράμματα λειτουργούν με τέσσερα, καθώς πρέπει να διαχειρίζονται μαθηματικούς μετασχηματισμούς. Οι κάρτες γραφικών και οι επεξεργαστές διαχειρίζονται τα πολύγωνα αναλύοντάς τα σε τρίγωνα. Οι μηχανές παιχνιδιών είναι πιο κοντά στο επίπεδο υλικού. Αν και αυξάνεται ο αριθμός πολυγώνων (1 τετράγωνο = 2 τρίγωνα), βασίζονται σε τρία σημεία για την πραγματοποίηση των υπολογισμών.

5.1.3. Αριθμός vertices

Η λεπτομέρεια που απεικονίζεται σε ένα σχήμα είναι ανάλογη των vertices (και κατά συνέπεια των edges και των faces) που το απαρτίζουν. Όσο περισσότερα, τόσο μεγαλύτερη η λεπτομέρεια. Με όρους 3D, αυξάνοντας την γεωμετρία ενός αντικειμένου, αυξάνουμε και την λεπτομέρεια. Όμως αυτό επιδρά σημαντικά στην υπολογιστική ισχύ που απαιτείται. Σαν γενικός κανόνας, με όσο λιγότερα vertices μπορεί να επιτευχθεί ένα ικανοποιητικό αποτέλεσμα τόσο καλύτερα. Στην εικόνα 5-4 το ίδιο πόμολο με 3182 vertices και 674 αντίστοιχα. Ειδικά αν το μοντέλο προορίζεται για εισαγωγή σε μηχανή παιχνιδιών όπου πρέπει να γίνεται rendering σε πραγματικό χρόνο, ο χαμηλός αριθμός vertices είναι επιβεβλημένος, η ορολογία είναι "low poly mesh".



Εικόνα 5-4. Αριστερά 3182 vertices (μετασχηματισμός subdivide x 2), δεξιά 674 vertices.

5.1.4. Materials, Textures και Shaders

Σε κάθε αντικείμενο που δημιουργείται, πρέπει να αντιστοιχιστεί ένα τουλάχιστον υλικό. Έτσι καθορίζονται ιδιότητες όπως το χρώμα του υλικού η ανάκλαση του φωτός, η διάχυση του φωτός, η διαφάνεια, η γυαλάδα.

Τα textures είναι το τι θα απεικονίζει η επιφάνεια του αντικειμένου. Βοηθά η σκέψη ότι είναι ένα ύφασμα που τυλίγει ένα αντικείμενο. Στην πραγματικότητα είναι εικόνες που εφαρμόζονται σε αυτό.

Οι shaders είναι που καθορίζουν όλες τις προηγούμενες ιδιότητες. Έτσι για παράδειγμα, το χρώμα του υλικού σε συνδυασμό με ένα texture ορίζει τον diffuse shader, τον βασικότερο ίσως απ'όλους. Μια καλή αναλογία είναι ένας πίνακας ζωγραφικής. Ο καμβάς είναι το αντικείμενο. Το texture είναι οι μπογιές θα που απλωθούν πάνω στον καμβά.

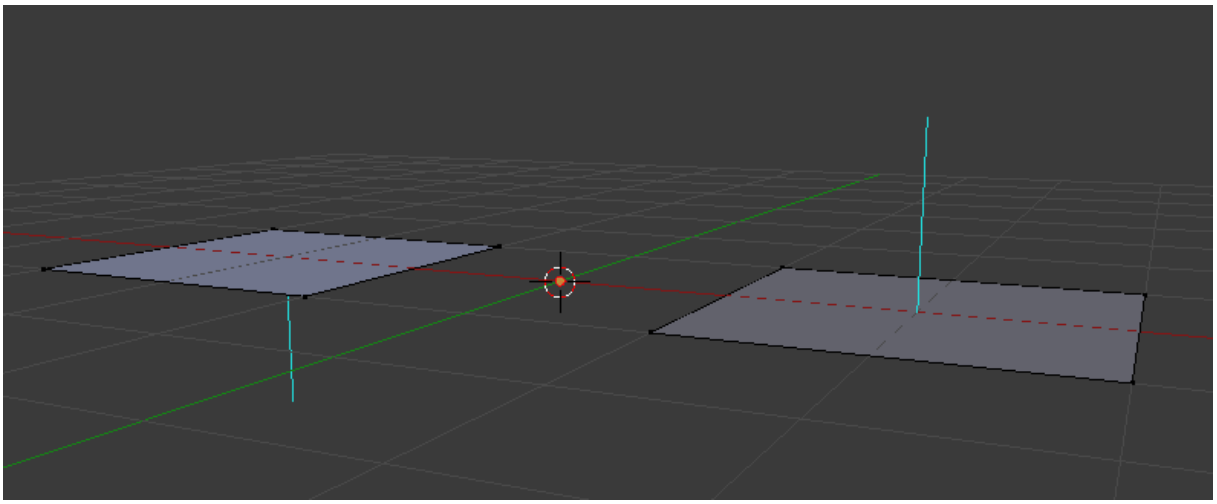
5.1.5. Normals

Η ψηφιακή αναπαράσταση αντικειμένων έχει τους κανόνες και τους περιορισμούς της. Είναι πολύ σημαντικό να κατανοηθεί τι είναι τα normals και πως επηρεάζουν το μοντέλο.

Στον πραγματικό κόσμο, ένα αντικείμενο αλληλεπιδρά με το φως. Ο βαθμός και ο τρόπος αλληλεπίδρασης εξαρτάται από το υλικό, το χρώμα, την υφή τις επιφάνειας, το αν είναι μεταλλικό ή όχι, και έναν ακόμη σεβαστό αριθμό ιδιοτήτων. Η τρισδιάστατη αναπαράσταση ενός αντικειμένου είναι η προσπάθεια προσομοίωσης αυτών των ιδιοτήτων όσο το δυνατόν πιο πειστικά, με την μεγαλύτερη δυνατή οικονομία σε υπολογιστική ισχύ. Είναι πρακτικά

αδύνατο, ακόμα και για τον πιο ισχυρό υπολογιστή, να προσομοίωση απόλυτα ένα φυσικό αντικείμενο και όλες τις ιδιότητές του.

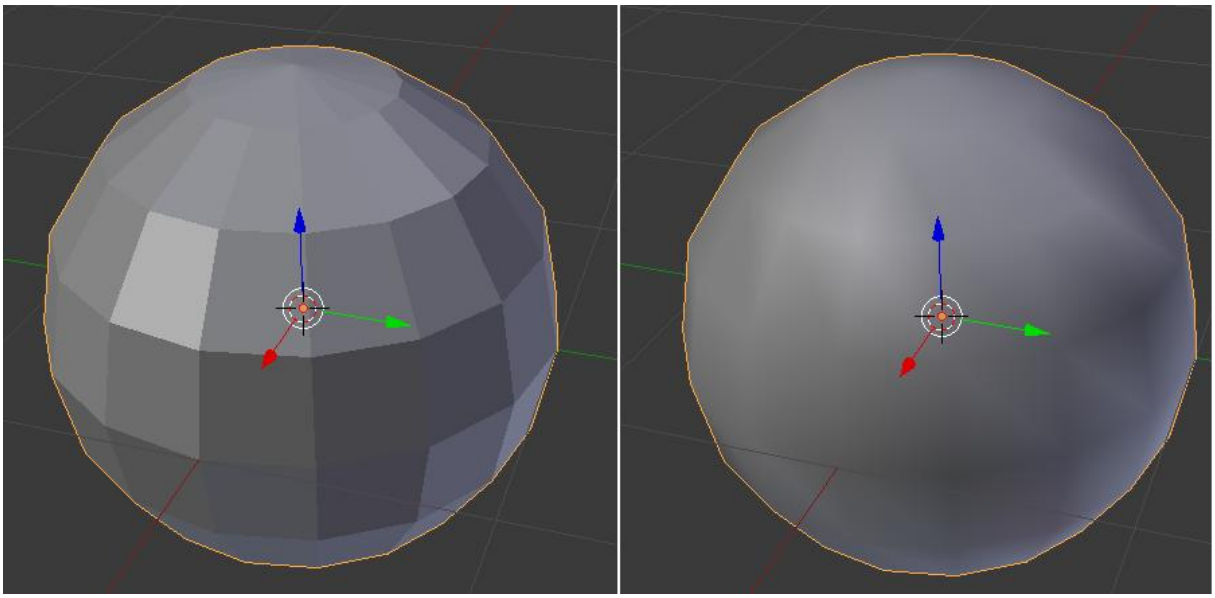
Το normals είναι στην ουσία διανύσματα που καθορίζουν ποιά είναι η εξωτερική πλευρά ενός face και ποιά είναι η εσωτερική, που είναι το πάνω, που το κάτω (εικόνα 5-5).



Εικόνα 5-5. Διαφορετικά normals για δύο faces.

Κάποια προγράμματα θεωρούν ότι τα faces έχουν normals και προς τις δύο κατευθύνσεις, κάποια όμως όχι. Αυτό μπορεί να επηρεάσει το τελικό αποτέλεσμα του rendering. Εκτός αυτού όμως και η αλληλεπίδραση με τους φυσικούς κανόνες μπορεί έχει απρόβλεπτα αποτελέσματα. Στην προηγούμενη εικόνα, εφόσον υποθέσουμε ότι εφαρμόζουμε τους φυσικούς νόμους, αν έπεφτε μια μπάλα στο δεύτερο face, θα αναπηδούσε όταν το χτύπαγε, αν έπεφτε στο πρώτο θα πέρναγε από μέσα του.

Από την εικόνα προκύπτει ότι τα normals είναι κάθετα στην επιφάνεια που τα ορίζει, αυτό όμως δεν είναι πάντα έτσι. Στην συγκεκριμένη περίπτωση πρόκειται για face normals. Υπάρχουν όμως και vertex normals, όπου το κάθε σημείο μπορεί να έχει διαφορετική γωνία αλληλεπίδρασης με το φως. Αυτή είναι η αρχή της σκίασης (shading) που χωρίς να αλλάζει τίποτε στο στερεό, το απεικονίζει διαφορετικά (εικόνα 5-6). Εφαρμόζεται ένας αλγόριθμος γραμμικής παρεμβολής ώστε το αντικείμενο να μην είναι εμφανές ότι αποτελείται από πολλά faces, αλλά να φαίνεται σαν ένα ενιαίο σύνολο. Με προσεκτική παρατήρηση, φαίνονται πάλι οι ασυνέχειες, αλλά δεν είναι πολύ εμφανείς.



Εικόνα 5-6. Flat Shading σε σύγκριση με smooth shading

5.1.6. Normal maps

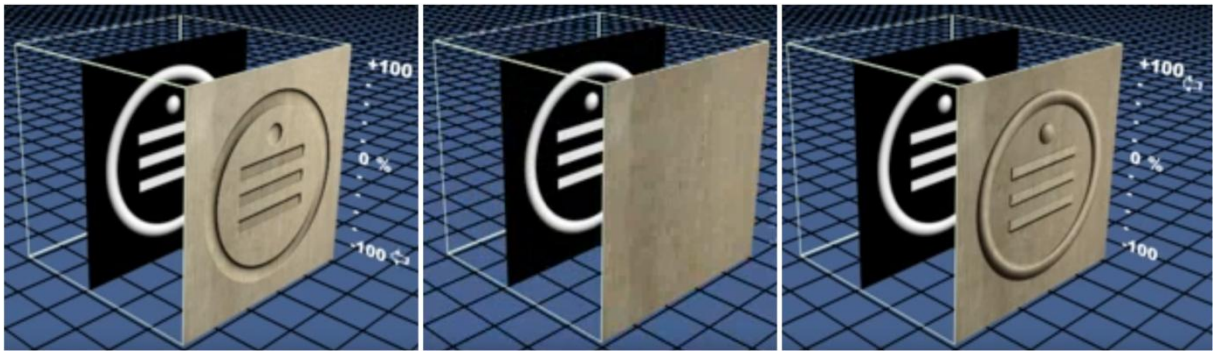
Η ανάγκη απεικόνισης ενός αντικειμένου με πιστή απόδοση στις λεπτομέρειες του είναι ένα πρόβλημα για τους υπολογιστές. Η προφανής λύση της δημιουργίας επιπλέον γεωμετρίας δεν είναι η καλύτερη, προσθέτει πολυπλοκότητα στο αντικείμενο, επιβαρύνει τον επεξεργαστή και την κάρτα γραφικών με πρόσθετους υπολογισμούς και αυξάνει τις απαιτήσεις σε μνήμη. Στο παράδειγμα της εικόνας 5-6, η πιστότητα της απεικόνισης αυξάνει με μια απλή αλλαγή στην σκίαση, χωρίς αύξηση της γεωμετρίας.

Αναζητήθηκαν έτσι τεχνικές που θα επιτυγχάνουν καλύτερο αποτέλεσμα χωρίς τα μειονεκτήματα που αναφέρθηκαν.

Στην ουσία τα 3D προγράμματα δημιουργούν την ψευδαίσθηση της ύπαρξης ενός αντικειμένου, το οποίο αποτελείται από πολλά μικρότερα σε μέγεθος επίπεδα που συνδέονται με κάποιον τρόπο μεταξύ τους για να σχηματίσουν ένα σύνολο. Αυτή η ψευδαίσθηση μπορεί να επεκταθεί παραπέρα με την χρήση τεχνικών που ξεγελάν το ανθρώπινο μάτι, έτσι ώστε ο εγκέφαλος να αντιλαμβάνεται λεπτομέρειες εκεί που στην πραγματικότητα δεν υπάρχουν.

Σε αυτό το πλαίσιο λειτουργούν τα bump maps, με έξυπνη χρήση των normals, Στην εικόνα 5-7 (Martinez 2010) φαίνεται πως επιδρά ένα bump map σε μια επιφάνεια. Εισάγεται μια ασπρόμαυρη εικόνα σε αυτό που ονομάζεται κανάλι. Το τελικό αποτέλεσμα είναι ο

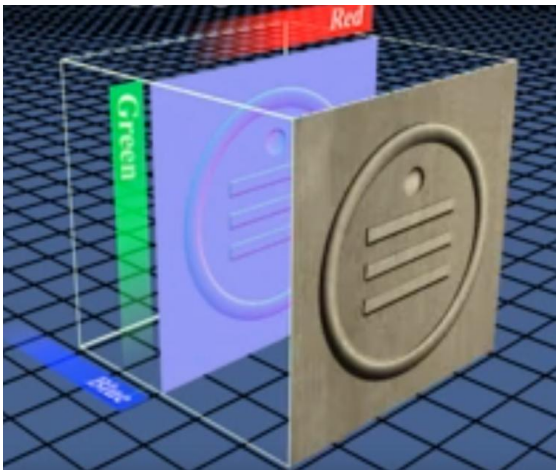
συνδυασμός των δύο, στο κέντρο η επίδραση του bump map είναι μηδενική, αριστερά αρνητική, δεξιά θετική.



.Εικόνα 5-7. Προσθήκη bump map σε μια επιφάνεια

Το αποτέλεσμα είναι μια αίσθηση του βάθους. Αν η γωνία από την οποία είναι ορατή το αντικείμενο είναι στις 180 μοίρες η αίσθηση παύει να υπάρχει.

Τα normal maps είναι μια εξέλιξη αυτής της αρχής. Είναι έγχρωμα και απεικονίζουν με πολύ μεγαλύτερη αποτελεσματικότητα τις τρεις διαστάσεις (εικόνα 5-8). Εκτός από το βάθος, λαμβάνεται υπόψη και η γωνία των normals, μαζί με καμπύλες και σύνθετες γωνίες. Το αποτέλεσμα της εικόνας όπου άλλα μέρη φαίνεται να βγαίνουν προς τα έξω ενώ άλλα φαίνεται να μπαίνουν μέσα, δεν είναι εφικτό με bump maps.



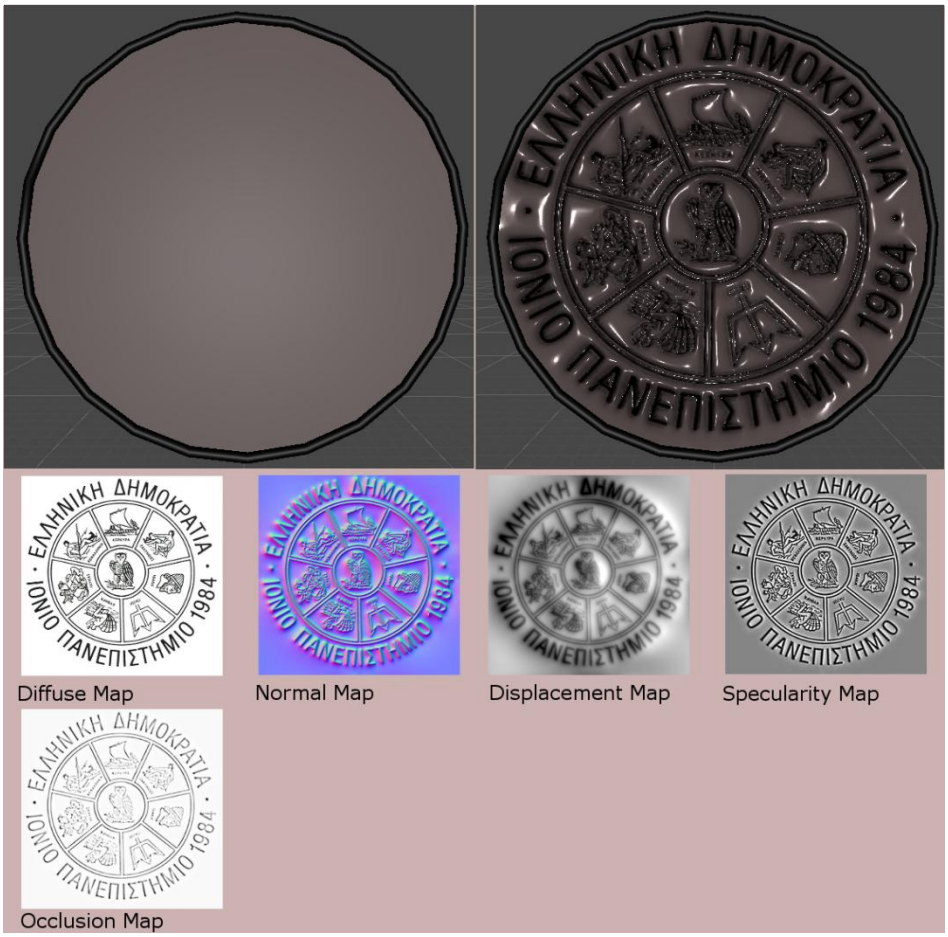
Εικόνα 5-8. Προσθήκη normal map σε μια επιφάνεια.

Στην εικόνα 5-9, πέρα από το βασικό diffuse map, που είναι το texture που απεικονίζεται από το αντικείμενο, και το normal map το οποίο εξηγήθηκε, υπάρχουν άλλα τρία. Το displacement map αλλάζει σε πραγματικό χρόνο την γεωμετρία του αντικειμένου, αλλά απαιτεί την ύπαρξη vertices στα σημεία όπου εφαρμόζεται. Είναι το αντίστοιχο του bump map με την διαφορά ότι η αλλαγή δεν γίνεται σε βάθος ύψος, αλλά πάνω-κάτω, δεξιά-

αριστερά. Το specularity map επηρεάζει την ανακλαστικότητα του αντικειμένου, και τέλος το occlusion την φωτεινότητα. Άλλα μέρη φαίνονται πιο φωτεινά, άλλα πιο σκοτεινά.

Γενικά, όλα αυτά έχουν μεγαλύτερη χρησιμότητα στις μηχανές παιχνιδιών όπου είναι διαδεδομένη η χρήση low poly meshes. Αλλά η δημιουργία ενός αντικειμένου σε ένα 3D πρόγραμμα, έχοντας συνειδητά κατά νου να κρατηθεί χαμηλά ο αριθμός των vertices, είναι πάντα καλή πρακτική.

Για να υπάρχει μια αντίληψη της τάξης μεγέθους, στο παιχνίδι Mass Effect ο κεντρικός χαρακτήρας αποτελείται από ~20000 vertices, ενώ στο Mass Effect 3 από ~25000 vertices ("Yes, but how many polygons?" An artist blog entry with interesting numbers 2007)



Εικόνα 5-9. Το αποτέλεσμα της εφαρμογής 5 διαφορετικών maps σε έναν κύκλο.

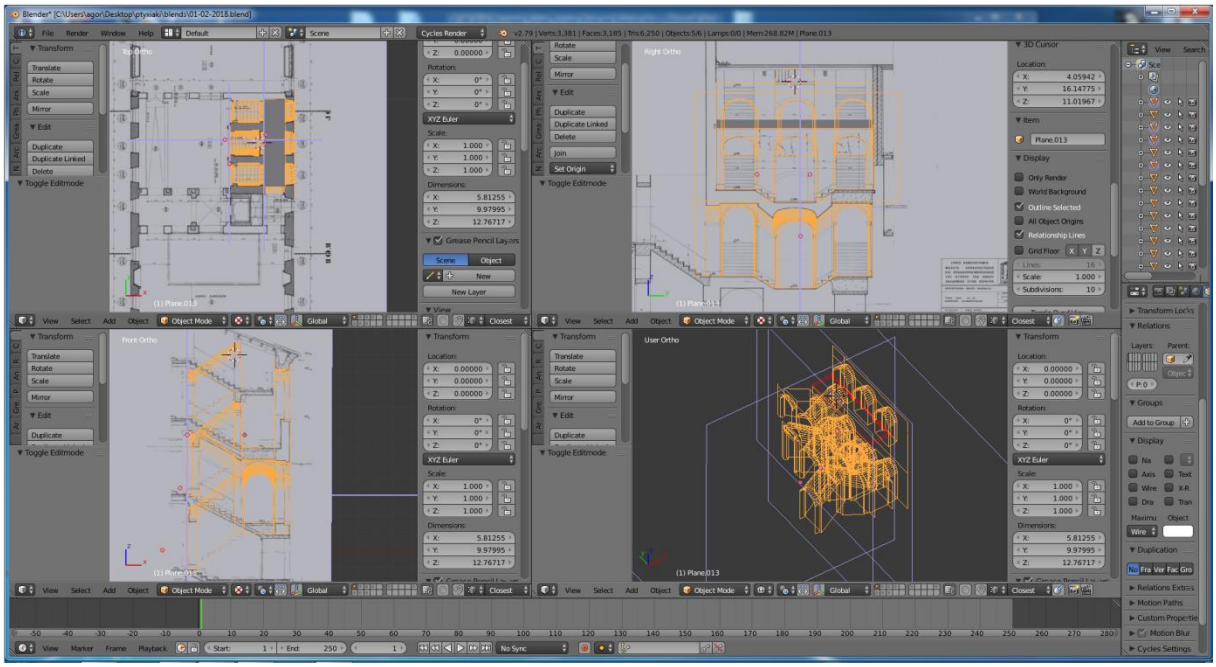
5.2. Η Διαδικασία της Δημιουργίας

Οι παραπάνω αρχές, μαζί με κάποιες που δεν αναφέρθηκαν χρησιμοποιήθηκαν στην συνέχεια για την δημιουργία του μοντέλου της Ιονίου Ακαδημίας.

5.2.1. Η Δημιουργία του κτιρίου από τα σχέδια της Ιονίου Ακαδημίας

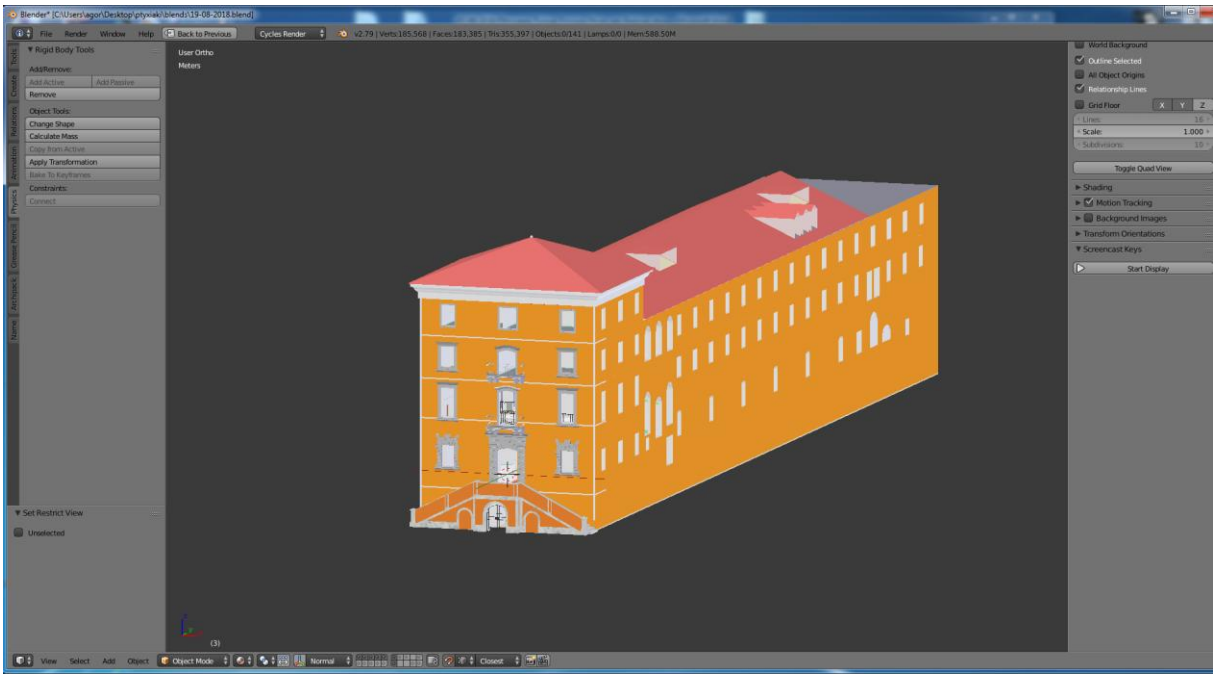
Επειδή η δημιουργία από μνήμης ή με αναφορά σε φωτογραφίες θα ήταν πολύ δύσκολη ως αδύνατη, χρησιμοποιήθηκαν σαν οδηγοί εικόνες των σχεδίων της Ιονίου Ακαδημίας.

Η αρχή είναι απλή. Στην επόμενη εικόνα φαίνεται η ρύθμιση του Blender σε κάτοψη, πρόσοψη και πλάγια όψη με τις αντίστοιχες εικόνες υποβάθρου. Οι τρεις όψεις είναι διαρκώς διαθέσιμες, η τέταρτη όψη είναι το αποτέλεσμα της δουλειάς στις τρεις διαστάσεις. Επειδή αυτές οι όψεις δεν αρκούσαν, χρησιμοποιήθηκαν και οι άνοψη, αριστερή όψη και πίσω όψη, με εναλλαγή στις ορατές ανάλογα με τις ανάγκες. Οι κατόψεις για κάθε όροφο ήταν σε ξεχωριστούς layers, η δουλειά ήταν αποδοτικότερη με αυτό τον τρόπο.



Εικόνα 5-10. Πάνω: Κάτοψη, Δεξιά Όψη. Κάτω: Πρόσοψη, Τρισδιάστατη Όψη

Υπήρχαν κάποιες ασυμφωνίες στα σχέδια, σε κάποια σημεία δεν έγινε πιστή μεταφορά. Το μοντέλο δεν έβγαινε χωρίς αυτές τις παραβάσεις. Έχω την υποψία ότι έγιναν εσκεμμένα κάποια λάθη στις διαστάσεις των σχεδίων. Μετά από κάποιους μήνες δουλειάς ήταν έτοιμο το κτίριο που μετά την εφαρμογή των materials ήταν όπως στην εικόνα 5-11.



Εικόνα 5-11. Το κυρίως κτίριο στην τελική του μορφή

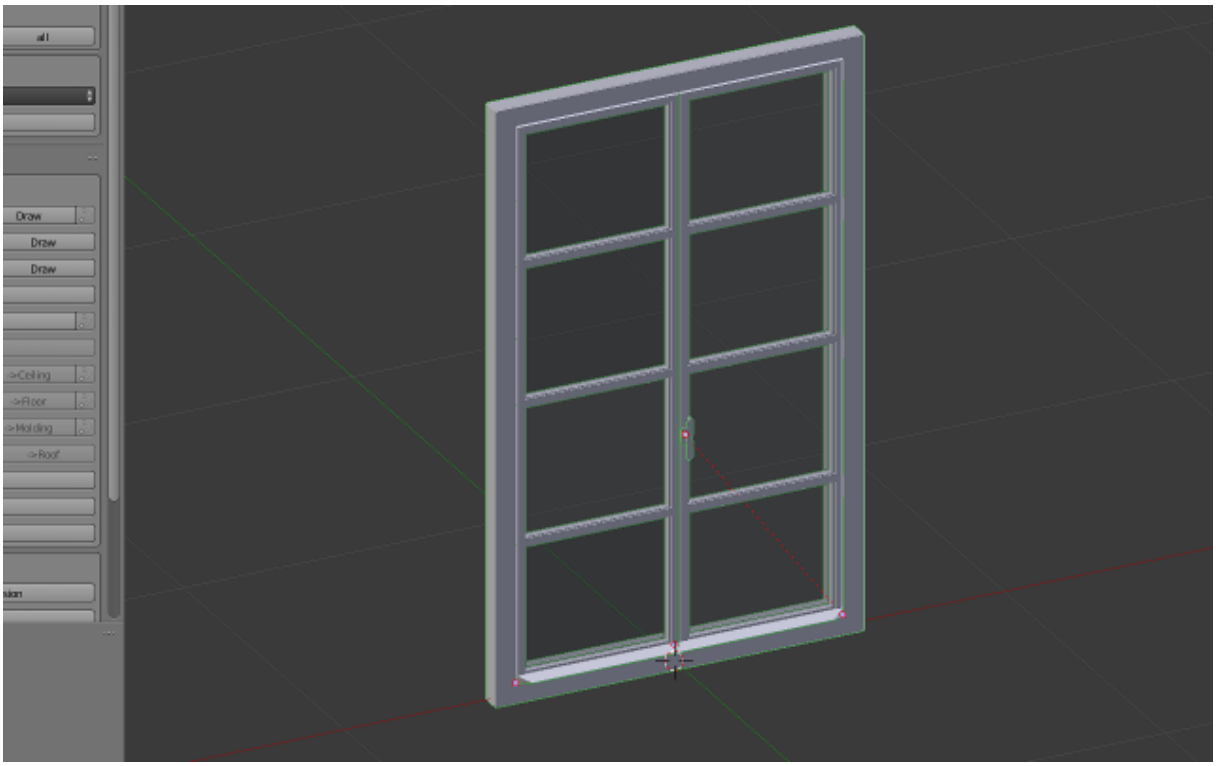
Απόμεναν οι πόρτες και τα παράθυρα, η προτομή του Καποδίστρια, ο Λέων στην είσοδο, το έμβλημα του Πανεπιστημίου. Και έπρεπε να προστεθούν και έπιπλα. Και όπου χρειαζόταν να γίνει και texturing. Τα φώτα στο φάτνωμα του ισογείου και του πρώτου ορόφου φτιάχτηκαν με ένα emissive material.

5.2.2. Οι Πόρτες και τα Παράθυρα

Χρησιμοποιήθηκε για αυτά το πρόσθετο "Archipack Tools" που διευκολύνει την δημιουργία τους. Υπάρχουν συνολικά 16 διαφορετικοί τύποι παραθύρων και 20 τύποι πορτών, συμπεριλαμβανομένων και αυτών του ασανσέρ. Συνολικά τα παράθυρα είναι 131[9], και οι πόρτες 73[10], με μία να μην ανοιγοκλείνει. Η ιδιαιτερότητα σε αυτό το μέρος της υλοποίησης του μοντέλου ήταν ότι δημιουργήθηκαν σχέσεις γονιού-παιδιού για τα κινούμενα μέρη (εικόνα 5-12). Τα σημεία περιστροφής φαίνονται από τις τελείες. Κάτι ανάλογο συμβαίνει και για τις πόρτες.

[9] 5 είναι μονόφυλλα, τα υπόλοιπα δίφυλλα.

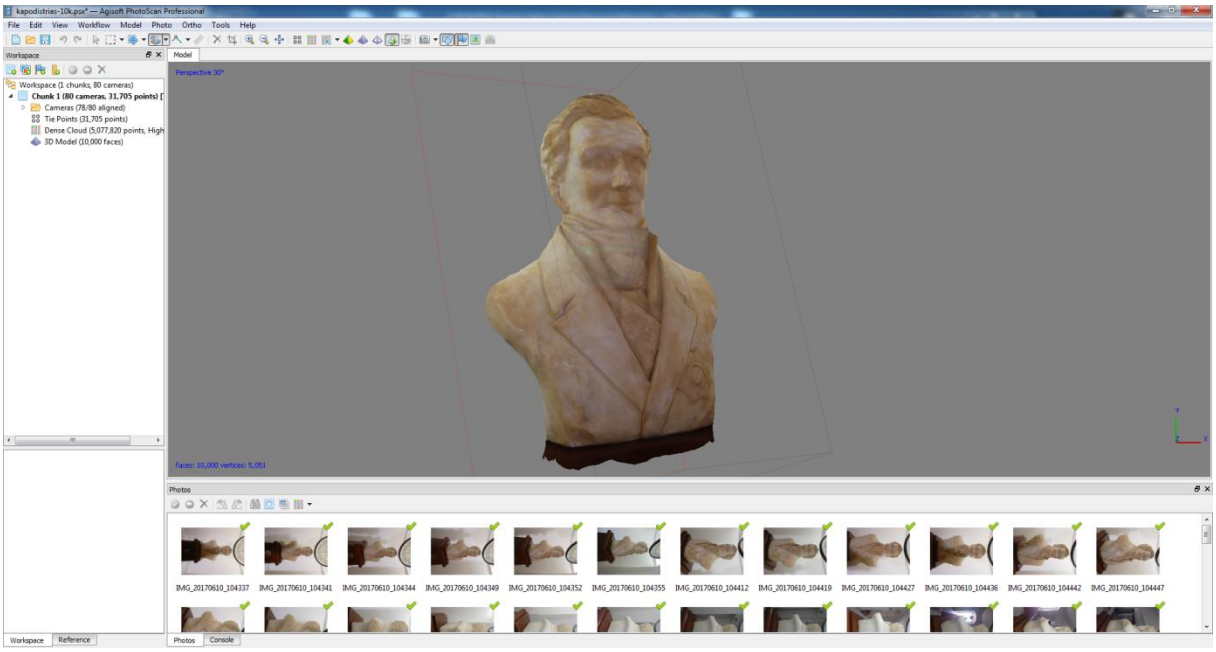
[10] 26 δίφυλλες μαζί με την μεταλλική καγκελόπορτα στην είσοδο, οι υπόλοιπες μονόφυλλες



Εικόνα 5-12. Το δεξί παραθυρόφυλλο είναι γονιός για το πόμολο

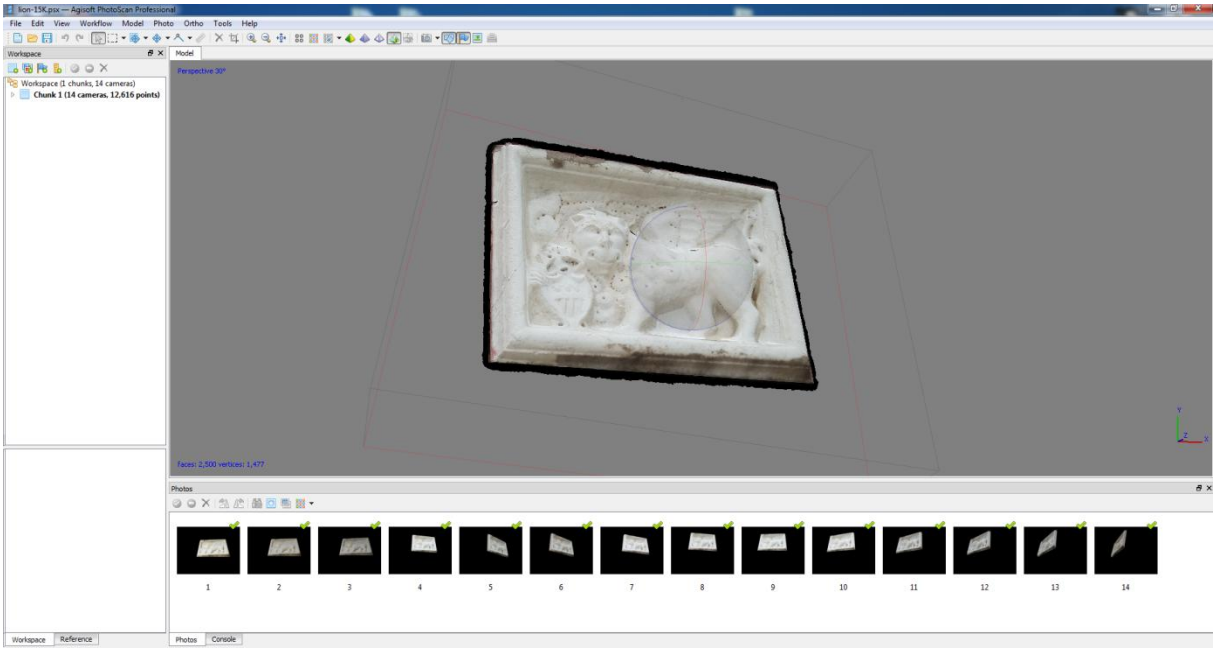
5.2.3. Φωτογραμμομετρία

Για την προτομή του Καποδίστρια και τον Λέοντα του Αγίου Μάρκου στην είσοδο, χρησιμοποιήθηκε μια διαφορετική προσέγγιση. Λόγω της δυσκολίας δημιουργίας με χρήση modeling και sculpting στο Blender, χρησιμοποιήθηκε η τεχνική της φωτογραμμομετρίας. Αυτή υλοποιήθηκε με το πρόγραμμα Photoscan. Η τεχνική μέσα από μια σειρά φωτογραφιών περιμετρικά και υπό γωνία (από πάνω και αν χρειάζεται από κάτω) ενός αντικειμένου, είναι ικανή να δημιουργήσει ένα τρισδιάστατο μοντέλο. Οι πρώτες προσπάθειες έγιναν για εκμάθηση του προγράμματος, που είναι σχετικά απλό στην χρήση του. Η πρώτη σοβαρή προσπάθεια δημιουργίας της προτομής δεν ήταν ιδιαίτερα πετυχημένη. Λόγω παράλειψής μου να φωτογραφήσω την προτομή υπό γωνία, το κεφάλι είχε μια τρύπα στο επάνω μέρος. Η δεύτερη προσπάθεια ήταν καλή, αλλά το αποτέλεσμα αποτελούνταν από παραπάνω από 500.000 vertices. Τελικά κατέληξα σε μια εκδοχή του Καποδίστρια που αποτελείται από περίπου 5000 vertices, με το αποτέλεσμα να είναι κάτι παραπάνω από ικανοποιητικό (εικόνα 5-13). Το μοντέλο εισάχθηκε στο blender. Έγινε ελάχιστη επεξεργασία, διαγράφηκαν πολύ λίγα vertices στο κάτω μέρος, αυτά που φαίνονται καφέ σκούρα στην φωτογραφία, και προστέθηκε η βάση της προτομής.



Εικόνα5-13. Η προτομή του Καποδίστρια ανασυστάθηκε από μια σειρά 78 φωτογραφιών

Η ίδια τεχνική χρησιμοποιήθηκε και για την δημιουργία του Λέοντα του Αγίου Μάρκου που βρίσκεται στην κεντρική είσοδο. Δημιουργήθηκε από 14 φωτογραφίες. Μετά την εισαγωγή της στο Blender και ελάχιστη επεξεργασία, αποτελείται από 1432 vertices .



Εικόνα5-14. Ο Λέων του Αγίου Μάρκου

5.2.4. Το έμβλημα του Ιονίου Πανεπιστημίου

Αυτό από άποψης μοντέλου δεν παρουσιάζει κάτι ιδιαίτερο. Είναι ένας κύκλος που αποτελείται από 34 vertices, στον οποίο όμως το material έχει 5 διαφορετικούς shader (εικόνα 5-9). Ίσως να μην είναι όλα απαραίτητα, αλλά το αποτέλεσμα είναι εντυπωσιακό.

5.2.5. Τα έπιπλα

Τα έπιπλα γραφείου, το σαλόνι στην είσοδο, οι νιπτήρες, οι τουαλέτες, βιβλιοθήκες και ο πυροσβεστήρας δεν δημιουργήθηκαν από μένα. Οι πηγές αναφέρονται στο παράρτημα.

5.2.6. Εξαγωγή σε fbx

Το Blender αποθηκεύει τα αρχεία που δημιουργούνται με αυτό σε .blend. Αν και το unity είναι σε θέση να εισάγει αυτά τα αρχεία, υπάρχουν πρακτικοί λόγοι ώστε να μην ακολουθηθεί αυτή η διαδικασία. Όταν γίνεται εισαγωγή ενός blend αρχείου στο Unity, εισάγονται όλα τα αντικείμενα. Αυτό συμπεριλαμβάνει τις κάμερες και τα φώτα. Το Unity προσπαθεί να λάβει υπόψη τα materials του blender εφόσον αυτά έχουν δημιουργηθεί στην μηχανή Blender Render. Αν δημιουργήθηκαν σε Cycles Render, οι ιδιότητες αγνοούνται.

Επιπλέον, ο λόγος που δεν επιλέχτηκε αυτή η μέθοδος είναι η οργάνωση σε layers. Χρησιμοποιήθηκαν διαφορετικοί layers για τον κάθε όροφο διαφορετικοί για τις πόρτες και τα παράθυρα, διαφορετικός για το ασανσέρ, διαφορετικοί για τα πατώματα. Η εξαγωγή δεν έγινε σε ένα μονολιθικό μοντέλο. Τα ξεχωριστά αντικείμενα οργανώθηκαν σε λογικά σύνολα που αποθηκεύτηκαν σε διαφορετικά αρχεία.

Κάνοντας για παράδειγμα ορατό τον layer 10 που περιέχει τις πόρτες και τα παράθυρα του υπογείου, και επιλέγοντας ότι υπάρχει σε αυτόν μπορεί να γίνει εξαγωγή μόνο των επιλεγμένων αντικειμένων. Επιλέχτηκε το format fbx καθώς υποστηρίζει τρισδιάστατα αντικείμενα, περιέχει τα materials τους, και τυχόν κίνηση τους. Εξαγωγή των materials γίνεται, όμως επειδή δημιουργήθηκαν σε Cycles Render είναι όλα το βαρετό γκρι, έπρεπε στην συνέχεια να γίνει επεξεργασία τους στοUnity.

ΚΕΦΑΛΑΙΟ 6. Unity

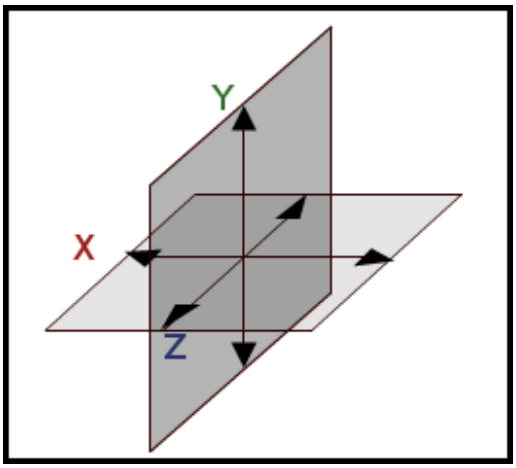
Με την ολοκλήρωση του μοντέλου της Ιονίου Ακαδημίας, έγινε εφικτό να υλοποιηθεί το επόμενο στάδιο. Έγινε εισαγωγή των objects με τα materials τους, και εισαγωγή των textures. Τα materials έπρεπε να τροποποιηθούν, ότι προηγούμενη δουλειά είχε γίνει στο Cycles Render δεν ήταν ικανοποιητική στο Unity. Στην συνέχεια γράφτηκε ο απαραίτητος κώδικας ελέγχου της συμπεριφοράς αντικειμένων στην γλώσσα προγραμματισμού C#.

6.1. Βασικές Αρχές

Πριν ακολουθήσει η μεθοδολογία της υλοποίησης του παιχνιδιού, θα γίνει μια εισαγωγή σε βασικές έννοιες και αρχές της μηχανής παιχνιδιών Unity που χρησιμοποιήθηκαν, με παραδείγματα. Αυτό κρίνεται απαραίτητο, καθώς δεν είναι εύκολο να γίνουν κατανοητές αυτές οι αρχές. Ειδικότερα για τους μη έχοντες υπόβαθρο γνώσεων πληροφορικής, δημιουργούν ένα σχεδόν ανυπέρβλητο εμπόδιο.

6.1.1. Συστήματα συντεταγμένων

Για την περίπτωση της Ιονίου Ακαδημίας, η αναπαράσταση είναι τρισδιάστατη, γίνεται στον χώρο. Έτσι, για να οριστεί πλήρως ένα σημείο, απαιτούνται τρεις μεταβλητές, ας ονομαστούν X, Y, και Z. Ενώ η γενική σύμβαση είναι ότι το X ορίζει το πλάτος (ή το δεξιά-αριστερά), το Y ορίζει το βάθος (ή το μπροστά-πίσω) και το Z το ύψος (πάνω-κάτω), στο Unity τα Y και Z αλλάζουν μεταξύ τους. Το Y ορίζει το ύψος και το Z ορίζει το βάθος όπως φαίνεται στην εικόνα 6-1 (Creighton 2010).

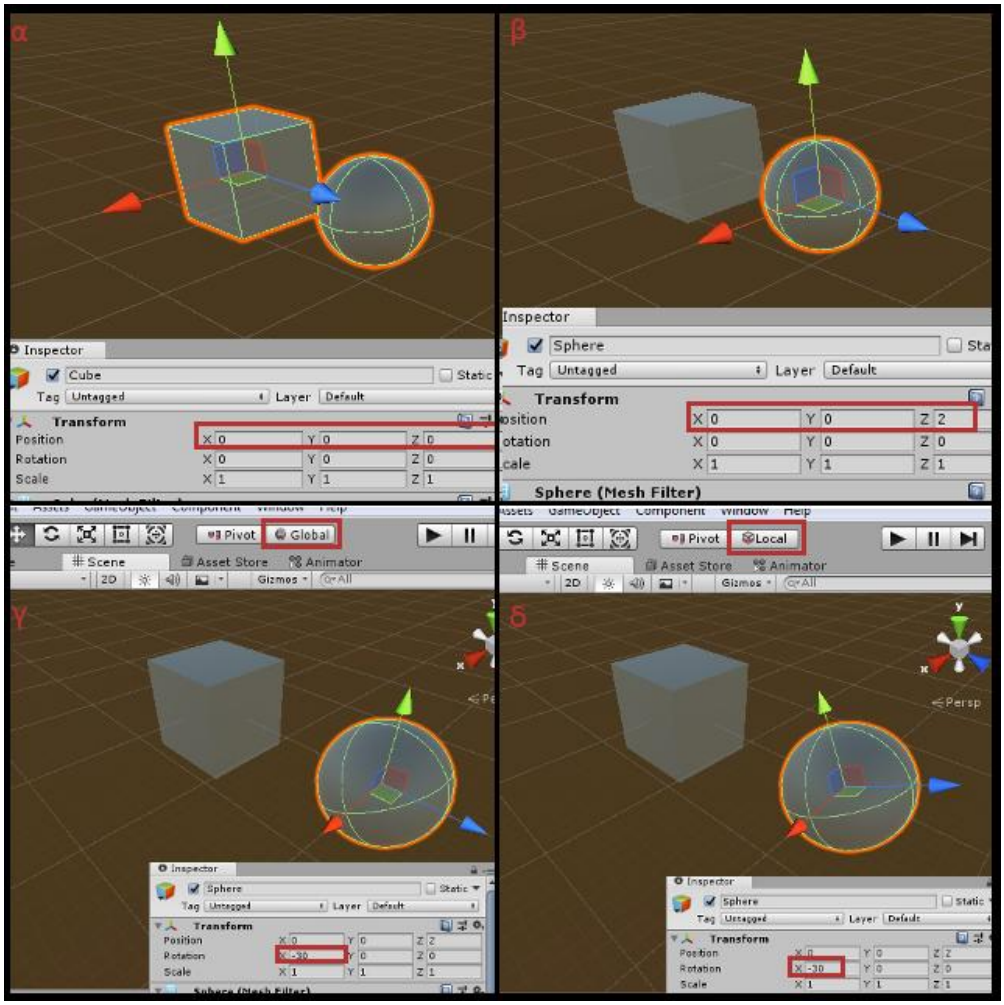


Εικόνα 6-1. Το Σύστημα Συντεταγμένων του Unity

Το σημείο 0, 0, 0, η αρχή, ορίζεται από το σημείο τομής των τριών αξόνων.

Για ένα αντικείμενο, είναι δυνατοί τριών ειδών μετασχηματισμοί. Μπορεί να αλλάξει το μέγεθος, η θέση και η περιστροφή. Το κέντρο του μετασχηματισμού όμως μπορεί να είναι είτε σε τοπικές συντεταγμένες είτε σε συντεταγμένες κόσμου. Για την περίπτωση της περιστροφής, το αντικείμενο μπορεί να περιστρέφεται είτε γύρω από τον εαυτό του, είτε γύρω από το κέντρο του κόσμου. Αυτό μπορεί να οδηγήσει σε ανεπιθύμητα αποτελέσματα αν δεν ληφθεί υπόψη.

Στο παράδειγμα της εικόνας 6-2 γίνεται εξήγηση με δύο αντικείμενα που έχουν σχέση γονιού-παιδιού. Ο κύβος βρίσκεται στην θέση 0, 0, 0 (α). Η σφαίρα στο 0, 0, 2 (β). Ο κύβος είναι γονιός για την σφαίρα. Αν μετακινηθεί ο κύβος η σφαίρα θα ακολουθήσει, χωρίς να αλλάξουν οι συντεταγμένες της. Η σφαίρα περιστρέφεται στον X κατά 30° (γ). Αν μετά από αυτό γίνει μετακίνηση σε συντεταγμένες κόσμου κατά τον Z, θα αλλάξει μόνο το Z (γ), αν γίνει μετακίνηση σε τοπικές θα αλλάξουν τα Y και Z (δ). Ο μπλε άξονας είναι ο Z.



Εικόνα 6-2. Συντεταγμένες για γονιό (α), παιδί (β), μετακίνηση στον Z στον κόσμο (γ) και τοπικά (δ)

Οι μετασχηματισμοί στις πόρτες και τα παράθυρα γίνονται σε τοπικές συντεταγμένες. Χρησιμοποιήθηκε το πρόσθετο iTween (iTween n.d.), που απλοποιεί κατά πολύ την συγγραφή κώδικα καθώς είναι εύκολα παραμετροποιήσιμο (κώδικας 6-1).

```
aSource = GetComponents<AudioSource>();

if (transform.parent.GetComponent<InteractiveObject>())
{
    //parentObject = transform.parent.gameObject;
    //aParentSource = GetComponent<AudioSource>();
    //parentObject = this.transform.parent;
    //aParentSource = parentObject.GetComponent<AudioSource>();
    //Debug.Log (parentObject);
    interactiveObjectParent = transform.parent.GetComponent<InteractiveO
    iTweenParentArgs = iTween.Hash();
    iTweenParentArgs.Add("position", interactiveObjectParent.openPositio
    iTweenParentArgs.Add("rotation", interactiveObjectParent.openRotatio
    iTweenParentArgs.Add("time", interactiveObjectParent.animationTime);
    //The Transform is to be done with local coordinates
    iTweenParentArgs.Add("islocal", true);
    iTweenParentArgs.Add("delay", 0);

    aParentSource = transform.parent.GetComponents<AudioSource>();
    //aParentSource = gameObject.GetComponentsInParent(typeof(A)

}

//aSource = GetComponent<AudioSource>();
```

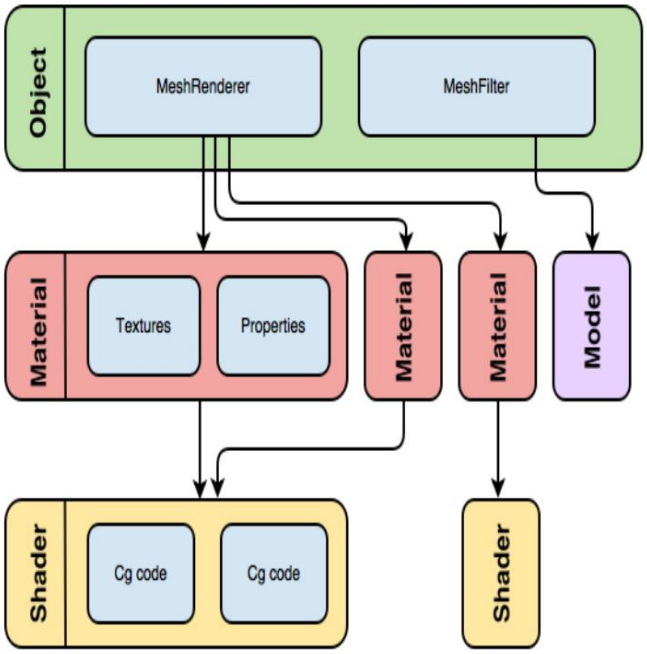
Κώδικας 6-1 . Μέρος του κώδικα διαχείρισης των μετακινήσεων και περιστροφών με χρήση του iTween

6.1.2. Components

Όλα τα αντικείμενα στο Unity έχουν συστατικά στοιχεία που επηρεάζουν το πως φαίνονται και πως συμπεριφέρονται αυτά (Zucconi και Kenneth 2016, κεφ. 2).

Η συμπεριφορά των αντικειμένων καθορίζεται από προγράμματα που προσαρτώνται σε αυτά. Είναι γνωστά σαν script. Είναι δυνατό να προσαρτηθούν περισσότερα από ένα script σε κάποιο αντικείμενο. Το Unity υποστηρίζει δύο γλώσσες προγραμματισμού, την C# και την javascript. Για την υλοποίηση της παρούσας έρευνας χρησιμοποιήθηκε η C#.

Ειδικότερα για τα τρισδιάστατα αντικείμενα, η εμφάνιση ορίζεται κατά κανόνα από έναν mesh renderer, που με την εισαγωγή του αντικειμένου στην σκηνή, προσαρτάται εξ ορισμού στο αντικείμενο. Ένα αντικείμενο πρέπει να έχει έναν μόνο renderer. Επιτρέπεται όμως αυτός να έχει περισσότερα από ένα materials (εικόνα 6.3).



Εικόνα 6-3. Βασικά Components ενός αντικειμένου και οι σχέσεις τους

Στην ουσία τα materials και οι shaders είναι κομμάτια κώδικα που ορίζουν το πως φαίνεται κάποιο αντικείμενο. Ο βασικότερος shader είναι ο diffuse shader. Ορίζει το χρώμα ενός αντικειμένου και πως διαχέεται το φως στην επιφάνειά του.

Επιγραμματικά, ευρέως χρησιμοποιούμενοι shaders είναι ο specularity shader ο οποίος ορίζει τον βαθμό γυαλάδας ενός αντικειμένου και ο transparency shader που ορίζει τον βαθμό διαφάνειας. Οι shaders διαθέτουν παραμέτρους των οποίων οι τιμές μπορούν αλλαχτούν ώστε να δοθεί η επιθυμητή υφή σε ένα αντικείμενο.

6.1.3. Colliders

Επειδή είναι σημαντικό να γίνει κατανοητή η επιρροή των collider στην συμπεριφορά του παιχνιδιού είναι σκόπιμη μια επεξήγησή τους.

Για να αλληλεπιδρά κάποιο αντικείμενο με τον παίκτη, εν προκειμένω με τον FPS Controller, θα πρέπει η μηχανή παιχνιδιών να λαμβάνει υπόψη τους φυσικούς κανόνες. συγκρούσεων. Αυτός είναι ο λόγος που εφαρμόζεται collider σε κάποιο αντικείμενο. Ένας τοίχος μπορεί να είναι ορατός, αλλά χωρίς collider ο παίχτης περνάει μέσα από τον τοίχο. Αν σε ένα πάτωμα δεν έχει εφαρμοστεί collider, ένα αντικείμενο που πέφτει πάνω σε αυτό δεν θα αναπηδήσει για να σταματήσει τελικά, θα συνεχίσει να πέφτει σαν να μην υπάρχει πάτωμα. Δεν είναι δυνατό το άνοιγμα μιας πόρτας αν στο χερούλι της δεν έχει εφαρμοστεί collider.

Στο παιχνίδι δεν είναι μόνο οι πόρτες και τα παράθυρα που ανοιγοκλείνουν με την χρήση colliders. Τα κουμπιά του ασανσέρ έχουν colliders που επηρεάζουν την συμπεριφορά τους και την συμπεριφορά του θαλάμου. Οι colliders δεν χρησιμοποιούνται μόνο για να ενεργοποιήσουν κίνηση, μπορούν να χρησιμοποιηθούν για την ενεργοποίηση οποιουδήποτε γεγονότος. Σαν παράδειγμα, τα παράθυρα διαλόγου ενεργοποιούνται με χρήση colliders Ο Λέων στην είσοδο, οι τρεις θυρεοί δεξιά και αριστερά της εισόδου, η προτομή του Καποδίστρια, το λογότυπο του Ιονίου Πανεπιστημίου και άλλα είναι τέτοιες περιπτώσεις.

Ένας collider περιβάλλει ένα αντικείμενο. Στην απλή του μορφή έχει σχήμα κύβου[11], κάψουλας[12] ή σφαίρας[13]. Σε πιο σύνθετη μορφή, μπορεί να είναι συνδυασμός των προηγούμενων για να προσεγγίσει σε μεγαλύτερο βαθμό το σχήμα του αντικειμένου ή να παίρνει το ακριβές σχήμα του αντικειμένου[14]. Το τελευταίο όμως επηρεάζει αρνητικά σε σημαντικό βαθμό τις επιδόσεις του παιχνιδιού και επιβαρύνει τον υπολογιστή.

Στα αντικείμενα που υπάρχουν εφαρμόζονται mesh colliders. Δόθηκε ιδιαίτερη προσοχή στην φάση της δημιουργίας των αντικειμένων ώστε αυτά να είναι low poly. Οι mesh colliders επιβαρύνουν την μηχανή γραφικών με πολλούς υπολογισμούς. Επιπλέον χρησιμοποιείται αποκλειστικά raycasting. για την ανίχνευση γεγονότων.

6.1.4. Rigidbodies

Αν το αντικείμενο έχει μάζα και βάρος, απαιτείται ένα rigidbody. Για το παρόν παιχνίδι, το μόνο αντικείμενο που διαθέτει rigidbody είναι ο FPS Controller. Χωρίς αυτόν δεν θα ήταν δυνατό ο FPS Controller να περπατά, να πηδά, να ανοίγει και να κλείνει πόρτες και παράθυρα, καλεί το ασανσέρ και να ενεργοποιεί την εμφάνιση διαλόγων. Απαραίτητη προϋπόθεση για να υπάρχει αλληλεπίδραση ενός αντικειμένου με ένα άλλο, είναι το ένα τουλάχιστον από αυτά να έχει κάποιον collider, και το άλλο να έχει ένα rigidbody.

Όταν δύο αντικείμενα αλληλεπιδρούν, λαμβάνει χώρα ένα trigger το οποίο ενεργοποιεί ένα event. Αν αυτό φαίνεται ακατανόητο, βοηθάει η σκέψη ότι ένα trigger λειτουργεί σαν

[11] box collider

[12] capsule collider

[13] sphere collider

[14] mesh collider

διακόπτης που όταν πατηθεί συμβαίνει κάτι. Αυτό επιτρέπει την διαχείριση της συμπεριφοράς του αντικειμένου με την συγγραφή κάποιου προγράμματος που θα καθορίζει τι συμβαίνει σε αυτή την περίπτωση (Colliders n.d.).

6.1.5. Raycasting

Το Raycasting είναι η διαδικασία της εκπομπής μιας αόρατης ακτίνας φωτός από ένα σημείο στον χώρο. Αυτή, όταν συναντά να αντικείμενο που έχει collider δημιουργεί ένα trigger event. Στην συνέχεια, με την συγγραφή ενός προγράμματος γίνεται έλεγχος της συμπεριφοράς του αντικειμένου, ορίζεται το τι ενέργεια θα εκτελεστεί (όπως η περιστροφή του πόμολου μιας πόρτας). Στο παρόν, δεν λαμβάνονται καθόλου υπόψη γεγονότα όπως OnCollisionEnter και OnTriggerEnter, χρησιμοποιείται η συνάρτηση Physics.Raycast() με τις παραμέτρους της, για να μπορεί στην συνέχεια να ελέγχεται η συμπεριφορά των αντικειμένων.

```
void Update()
{

    Physics.Raycast(cam.transform.position, cam.transform.forward, out hit, interactRange);
    //if (hit.transform.name == "bust") reticleController.ShowIcon("QuestionIcon");

    //MUST hit a handle, or an elevator button, or a left shutter (in which case there must be a ch
    //else if (hit.transform && hit.transform.root.name != "all" && (hit.transform.childCount == 1
    if (hit.transform)
    {   hitObject = hit.transform.GetComponent<Transform>();
        //Debug.Log(hit.transform.name);
        if (hit.transform.name == "beethoven") playMusic = hit.transform.GetComponent<PlayMusic>();
        else interactiveObject = hit.transform.GetComponent<InteractiveObject>();
        if (reticleController.reticleActive) reticleController.ShowIcon("HandIcon");

        if (hit.transform.name == "-1") elevatorButton = hit.transform.gameObject;
        if (hit.transform.name == "0") elevatorButton = hit.transform.gameObject;
        if (hit.transform.name == "1") elevatorButton = hit.transform.gameObject;
        if (hit.transform.name == "2") elevatorButton = hit.transform.gameObject;
        if (hit.transform.name == "3") elevatorButton = hit.transform.gameObject;
        if ((hit.transform.name == "-1") ||
            (hit.transform.name == "0") ||
            (hit.transform.name == "1") ||
            (hit.transform.name == "2") ||
            (hit.transform.name == "3") ||
            (Regex.IsMatch(hit.transform.name, "elevator", RegexOptions.IgnoreCase)))
        {
            if (reticleController.reticleActive) reticleController.ShowIcon("PointIcon");
            //elevator = hit.transform.parent.parent.GetComponent<Elevator>();
            //elevator = GameObject.Find("elevator").transform.GetComponent<Elevator>();
        }
        else if (hit.transform.name == "bust" ||
                 hit.transform.name == "painting" ||
                 hit.transform.name == "ionioan-academy-sign" ||
                 hit.transform.name == "frederick-north-sign" ||
                 hit.transform.name == "grimani-sign" ||
                 hit.transform.name == "logo" ||
```

Κώδικας 6-2. Μέρος του κώδικα που ελέγχει το Raycasting της κάμερας του FPS Controller με αντικείμενα

6.2. Η διαδικασία της Δημιουργίας

Η αρχές που αναφέρθηκαν στα 6.1.1 ως 6.1.5 χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής στο Unity. Αν και παρουσιάστηκε κώδικας, ο στόχος ως εδώ δεν ήταν η γνωριμία με την C#, αλλά μια πρώτη επαφή με το πρόσθετο iTween και τον τρόπο που λειτουργεί το Raycasting. Παρακάτω, παρουσιάζονται μέρη του κώδικα που επιτελούν συγκεκριμένες εργασίες και επιδεικνύεται η προσάρτηση script σε αντικείμενα. Ο πλήρης κώδικας των script βρίσκεται στο παράρτημα.

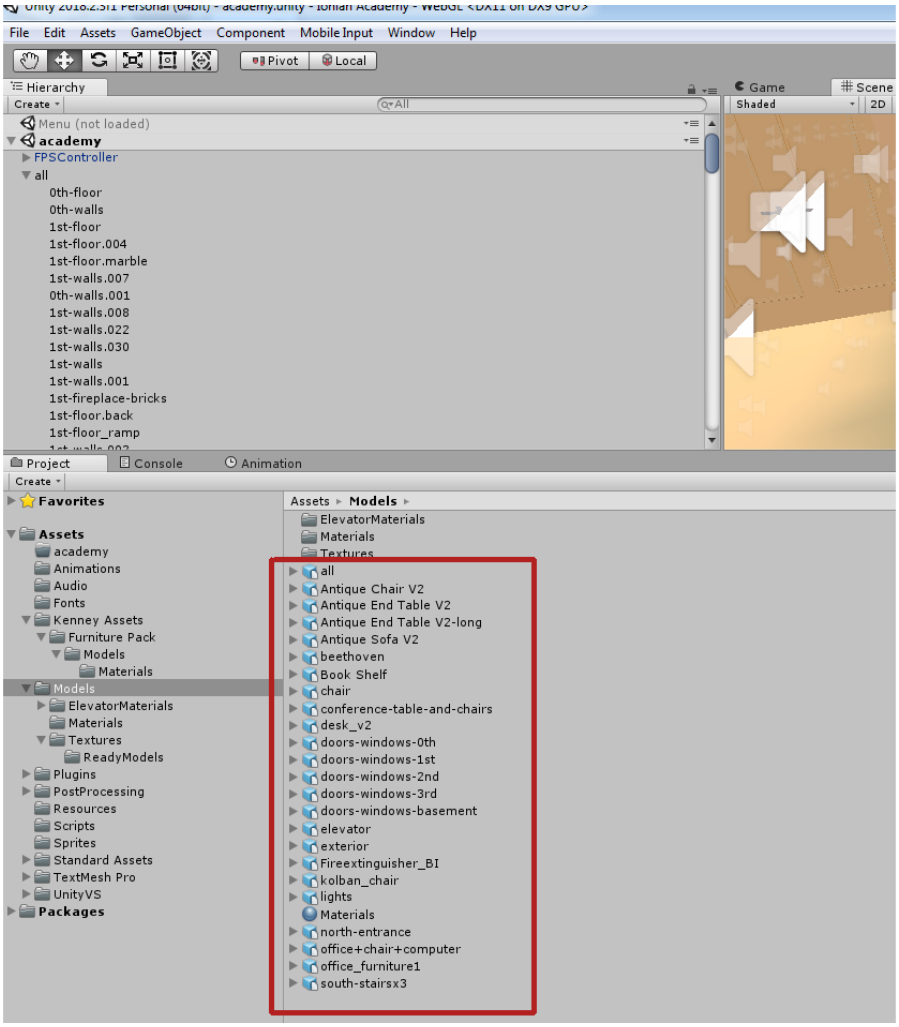
6.2.1. Εισαγωγή των assets από τα αρχεία fbx

Τα αντικείμενα οργανώθηκαν σε λογικές οντότητες με στόχο την ευκολότερη διαχείριση τους. Έτσι, υπάρχουν ομάδες για το κυρίως κτίριο (all), για τις πόρτες και τα παράθυρα ανά όροφο (doors-windows-basement, doors-windows-0th, κλπ), για το ασανσέρ (elevator), ξεχωριστά για κάθε έπιπλο (office+chair+computer, fire-extinguisher, Antique-Chair-V2, κλπ), και για κάποια φώτα που δημιουργήθηκαν στο blender (lights).

Κατά την εξαγωγή τους από το blender υπήρξε πρόβλεψη για αυτό, δημιουργήθηκαν ξεχωριστά αρχεία fbx για τα προαναφερόμενα.

Η Εισαγωγή στο Unity γίνεται από το μενού Assets → Import New Asset και στην συνέχεια αναζήτηση της θέσης του ανάλογου αρχείου στον υπολογιστή (ή με drag and drop).

Στην εικόνα 6-4 φαίνεται το αποτέλεσμα μετά την εισαγωγή όλων των assets.

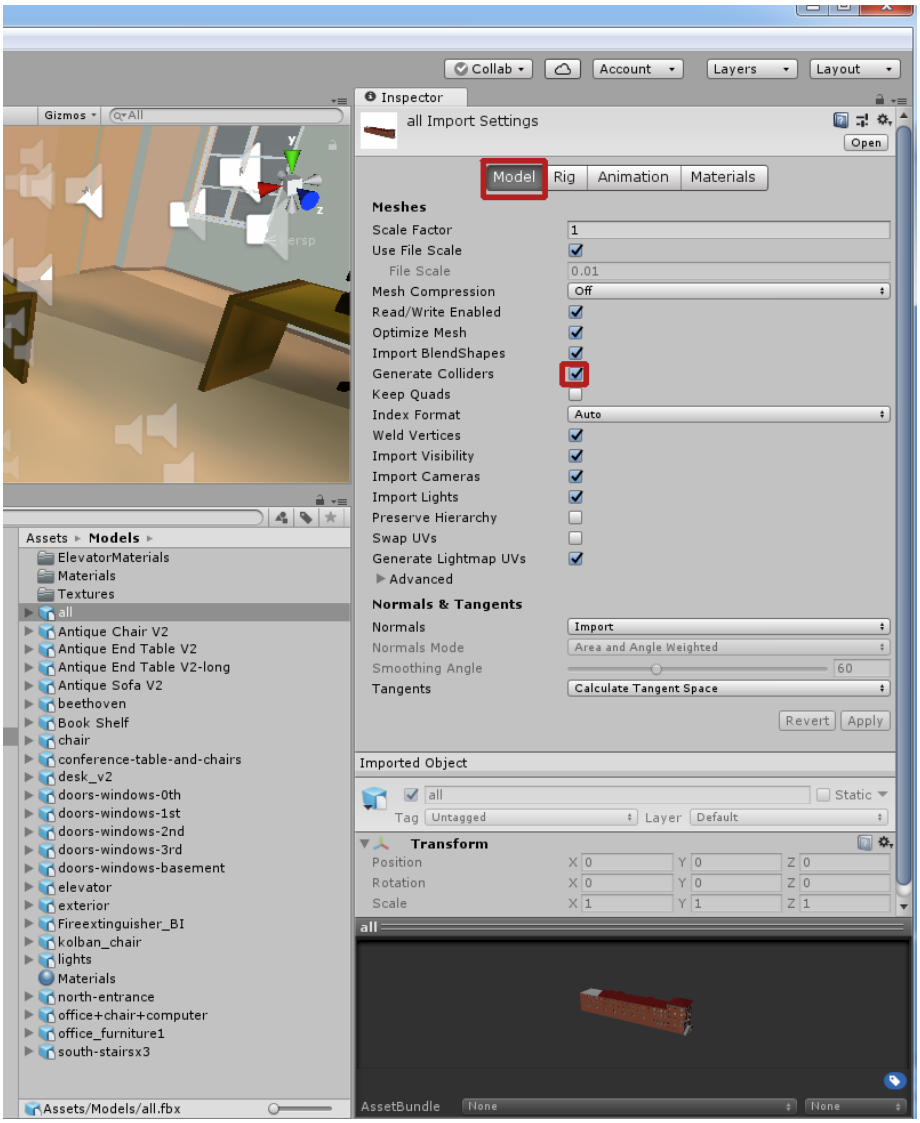


Εικόνα 6-4. Τα assets όπως έχουν εισαχθεί στο Unity

Κάτι αντίστοιχο έγινε για τα materials και τα textures. Στην εικόνα φαίνεται επίσης ότι για τα materials του ασανσέρ ακολουθήθηκε ξεχωριστή οργάνωση.

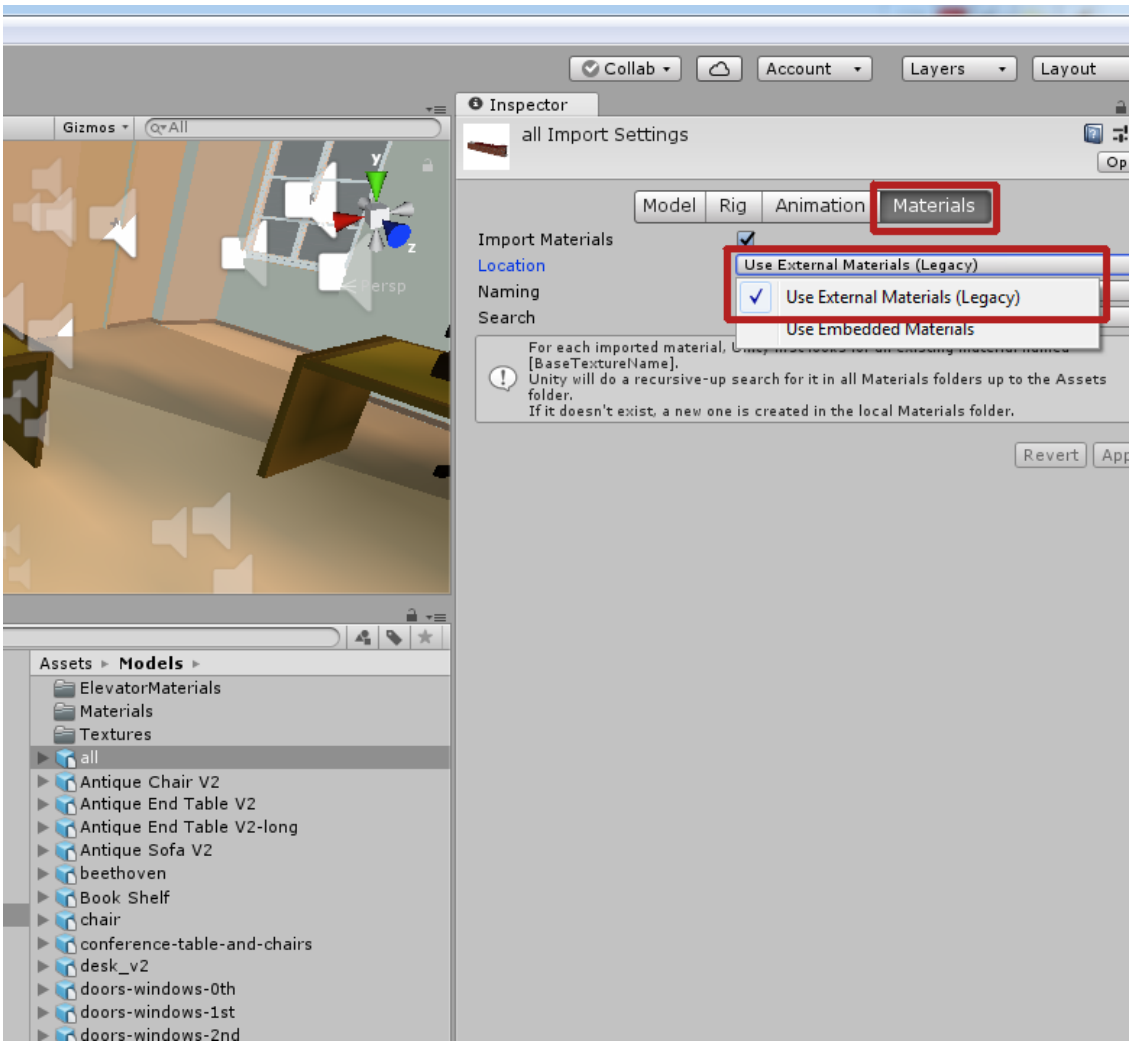
Για να γίνουν αυτά μέρος του παιχνιδιού, πρέπει να εισαχθούν στο Hierarchy. Στην εικόνα 6-4 φαίνονται πάνω αριστερά ένα scene για το μενού (απενεργοποιημένο), ένας ελεγκτής πρώτου προσώπου (FPS Controller) και μέρος των αντικειμένων που απαρτίζουν το κυρίως κτίριο.

Πολύ σημαντική επισήμανση για τα αντικείμενα: Θα πρέπει να προνοηθεί η δημιουργία colliders (εικόνα 6-5). Η εισαγωγή στο Hierarchy μπορεί να γίνει και χωρίς αυτό το βήμα, όμως με αυτό τον τρόπο αποφεύγεται επιπλέον κόπος και χρόνος.



Εικόνα 6-5. Δημιουργία Colliders για το asset all.

Στην περίπτωσή της έρευνας, επειδή χρησιμοποιήθηκε το Blender Cycles Render, έγινε και εξαγωγή των materials σε legacy mode για να είναι δυνατή στην συνέχεια η επεξεργασία τους (εικόνα 6-6).

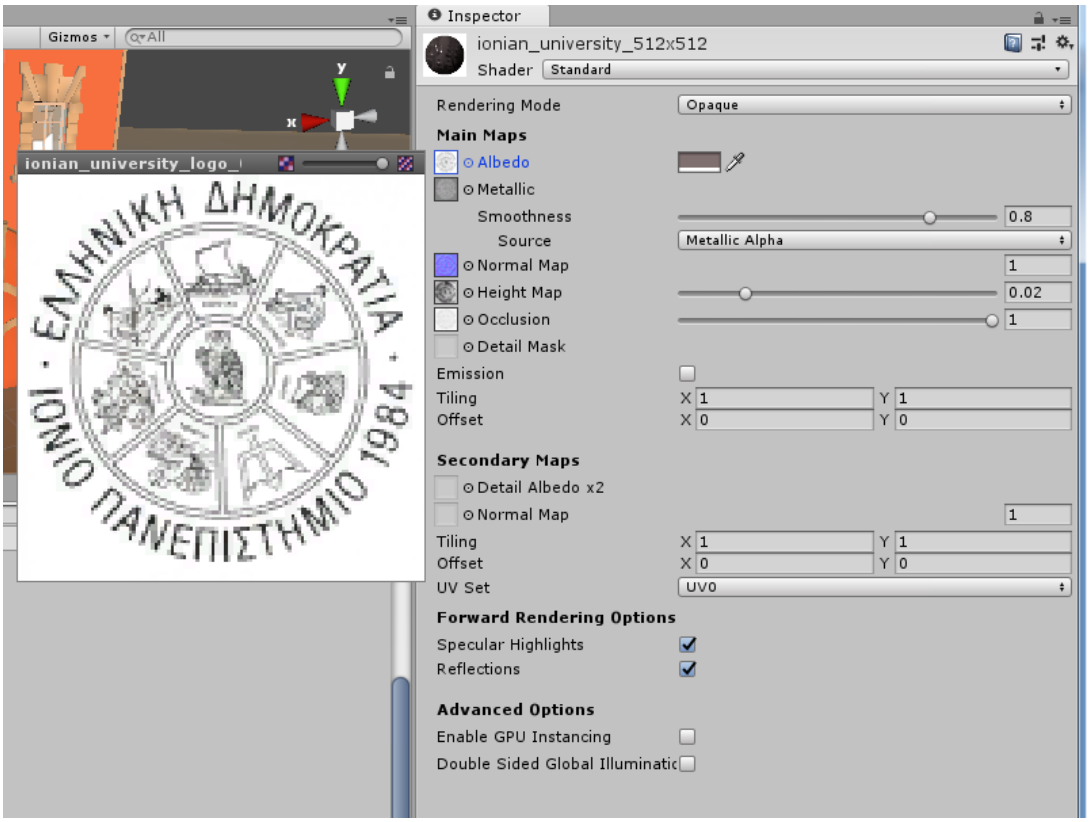


Εικόνα 6-6. Εισαγωγή materials

6.2.2. Materials και Textures

Σε αναλογία με το blender, κάθε αντικείμενο πρέπει να έχει τουλάχιστον ένα material. Μπορούν είτε να εισαχθούν στοUnity, με την διαδικασία που περιγράφτηκε, ή μπορούν να δημιουργηθούν νέα. Από το μενού, δίνοντας Assets → Create → Material δημιουργείται ένα νέο material που εισάγεται στον τρέχοντα κατάλογο. Το πρώτο βήμα μπορεί να αντικατασταθεί με δεξί κλικ στο παράθυρο του Project ή των Assets. Με επιλογή τώρα ενός αντικειμένου, μπορεί με drag-and-drop να προστεθεί σε αυτό το material στον inspector.

Σαν προεπιλογή, όταν δημιουργείται ένα αντικείμενο στο Unity, του αποδίδεται και ένα material, το Default Material, χωρίς όμως την δυνατότητα τροποποίησης των ιδιοτήτων του. Για να γίνει αυτό θα πρέπει με drag and drop να ανατεθεί στο αντικείμενο ένα material που δημιουργήθηκε με την μέθοδο που προαναφέρθηκε.



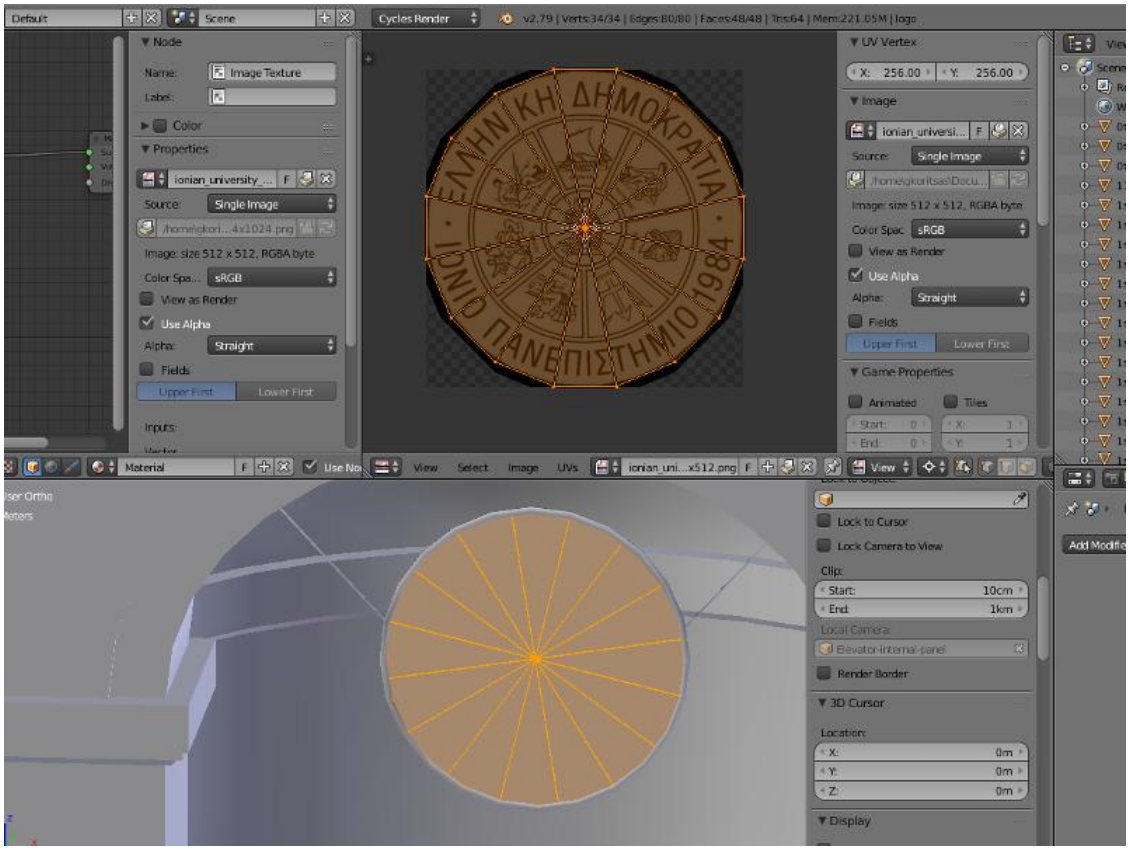
Εικόνα 6-7. Material με τις ιδιότητες του

Στην εικόνα 6-7 έχει δοθεί ένα χρώμα στο material και έχουν αντιστοιχιστεί πέντε χάρτες (maps). Στο Albedo ή αλλιώς Diffuse map, έχει δοθεί μια ελαφριά καφέ απόχρωση, και επιπρόσθετα έχει ανατεθεί ένα texture με το λογότυπο του Ιονίου Πανεπιστημίου. Τα υπόλοιπα maps παράγονται από αυτό. Χρησιμοποιούνται ένα metallic, ένα height (ή bump), ένα normal και ένα occlusion map. Στην πλειοψηφία τους έχουν εξηγηθεί στο Κεφάλαιο 5 και το αποτέλεσμα της εφαρμογής τους στο αντικείμενο φαίνεται στην εικόνα 5-9, σ.41. Η αντιστοιχία με όσα ισχύουν στο blender είναι απόλυτη, με τα ονόματα να διαφέρουν σε μερικές περιπτώσεις. Είναι σκόπιμο να αναφερθεί ότι το occlusion map καθορίζει πως κατανέμεται το άμεσο και έμμεσο φως στο αντικείμενο (McDermott 2010).

Στην συγκεκριμένη περίπτωση, όπως φαίνεται δεν χρησιμοποιήθηκαν όλες οι ιδιότητες. Δεν είναι απαραίτητο να συμβαίνει αυτό, εξαρτάται από τον βαθμό ρεαλισμού που θα αναπαρίσταται από το αντικείμενο. Όπως είναι αυτονόητο, με επιμονή και υπομονή στις ιδιότητες του material, μπορούν να δημιουργηθούν άκρως εντυπωσιακά αποτελέσματα.

Για το texture, πριν αυτό "φορεθεί" στο αντικείμενο, θα πρέπει να έχει γίνει στο 3D πρόγραμμα το επονομαζόμενο UV unwrapping. Αυτό σημαίνει ότι γίνεται αντιστοίχιση των

faces του αντικειμένου με το texture, ποιά faces θα δείχνουν ποιές περιοχές τις εικόνας (εικόνα 6-8).



Εικόνα 6-8. UV Unwrapping του λογότυπου της Ιονίου Ακαδημίας στο Blender

6.2.3. C#: Ενώνοντας τα επιμέρους για να συμβαίνει κάτι ενδιαφέρον

Στην παρούσα έρευνα η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η C#. Αυτό δεν είναι δεσμευτικό, το Unity υποστηρίζει και την γλώσσα javascript. Είναι θέμα επιλογής. Επιλέχτηκε η C# καθώς με αυτήν έγινε η αρχή και σε αυτήν αφιερώθηκε χρόνος εκμάθησης. Δεν είναι απαγορευτικό να χρησιμοποιούνται και οι δύο γλώσσες στο ίδιο project, αν και δεν έχω δει κάτι τέτοιο να γίνεται στην πράξη. Στην παρούσα εφαρμογή, αυτό θα δημιουργούσε πρόσθετες δυσκολίες καθώς αρκετές φορές ένα script καλεί συναρτήσεις άλλων script.

Δημιουργήθηκαν συνολικά δεκαπέντε script, τα προγράμματα λέγονται έτσι στο Unity επειδή συνδέονται με τα αντικείμενα. Ένα ή περισσότερα script συνδέονται με ένα αντικείμενο ώστε να ελέγχεται από αυτά η συμπεριφορά του.

Ο απλούστερος τρόπος να γίνει κατανοητό το πως λειτουργεί αυτό, είναι η παρουσίαση της διαδικασίας είναι μέσα από παραδείγματα. Δημιουργήθηκε μέσα στον κατάλογο των Assets ένας κατάλογος με όνομα Scripts. Όλα τα Script τοποθετήθηκαν μέσα σε αυτόν.

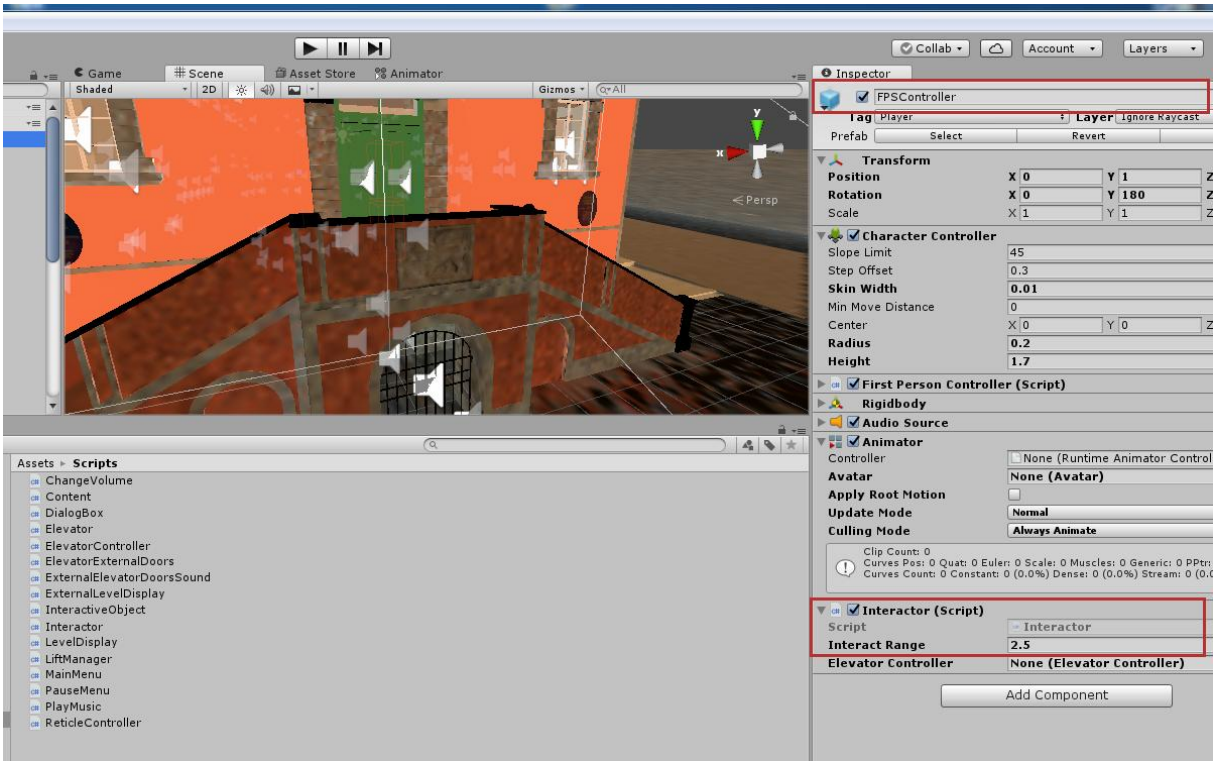
Κάποιες από τις μεταβλητές είναι σκόπιμο να εμφανίζονται στον Inspector για ορισμό της τιμής τους κατά την φάση της σχεδίασης και όχι κατά την φάση της εκτέλεσης του παιχνιδιού. Αυτό συμβαίνει μετά τον ορισμό του class με την λέξη κλειδί SerializeField (κώδικας 6-3).

```
1  using UnityEngine;
2  using System.Collections;
3  using System.Text.RegularExpressions;
4
5  public class InteractiveObject : MonoBehaviour
6  {
7
8      [SerializeField] public Vector3 openPosition, closedPosition, openRotation, closedRotation;
9      [SerializeField] private float animationTime;
10     [SerializeField] public bool isOpen = false;
11     [SerializeField] public bool resetWhenDone = false;
12     [SerializeField] public bool isShutter = false;
13     [SerializeField] private MovementType movementType;
14     private InteractiveObject interactiveObjectParent;
15     private GameObject parentObject;
16     private enum MovementType { Slide, Rotate };
17     private Hashtable iTweenArgs;
18     private Hashtable iTweenParentArgs;
19     private AudioSource aZONK;
20     private AudioSource[] aSource, aParentSource;
21     private GameObject ZONK, leftPanel, rightPanel;
22
```

Κώδικας 6-3. Το Keyword SerializeField εκθέτει μιά μεταβλητή στον Inspector.

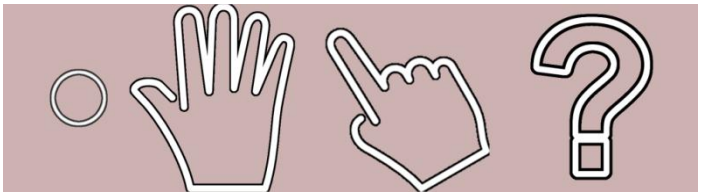
6.2.4. FPS Controller script

Η διάδραση καθορίζεται από τον FPS Controller. Οτιδήποτε συμβαίνει στο παιχνίδι, περνάει από αυτόν. Προαναφέρθηκε ότι γίνεται χρήση Raycasting. Το script που καθορίζει την συμπεριφορά του FPS Controller και το πως αλληλεπιδρά με τα αντικείμενα ονομάστηκε interactor.cs.



Εικόνα 6-9. Interactor Script προσαρτημένο στον FPS Controller

Η απόσταση επιρροής του Raycasting έχει οριστεί στα 2.5 μέτρα (εικόνα 6-9). Όταν ο FPS Controller είναι σε απόσταση που είναι μικρότερη ή ίση από αυτήν την τιμή και επιτρέπεται διάδραση με κάποιο αντικείμενο ο εξ ορισμού δείκτης που είναι ένας κύκλος στο κέντρο της οθόνης αλλάζει. Ανάλογα με το είδος της επιτρεπόμενης διάδρασης γίνεται είτε χέρι, είτε δείκτης, είτε ερωτηματικό (εικόνα 6-10). Γίνεται έλεγχος σε κάθε frame για τυχόν αλλαγή κατάστασης (void Update). Οι τέσσερις εικόνες μετατρέπονται σε 2D sprites και είναι παιδιά για ένα δισδιάστατο αντικείμενο Canvas που ονομάστηκε ReticleCanvas.



Εικόνα 6-10. Οι τέσσερις δυνατές καταστάσεις του δείκτη.

Σε αυτήν την περίπτωση, η διαδικασία είναι σχετικά απλή, εξηγείται παρακάτω με λεπτομέρειες για την δημιουργία των μενού όπου είναι πιο σύνθετη.

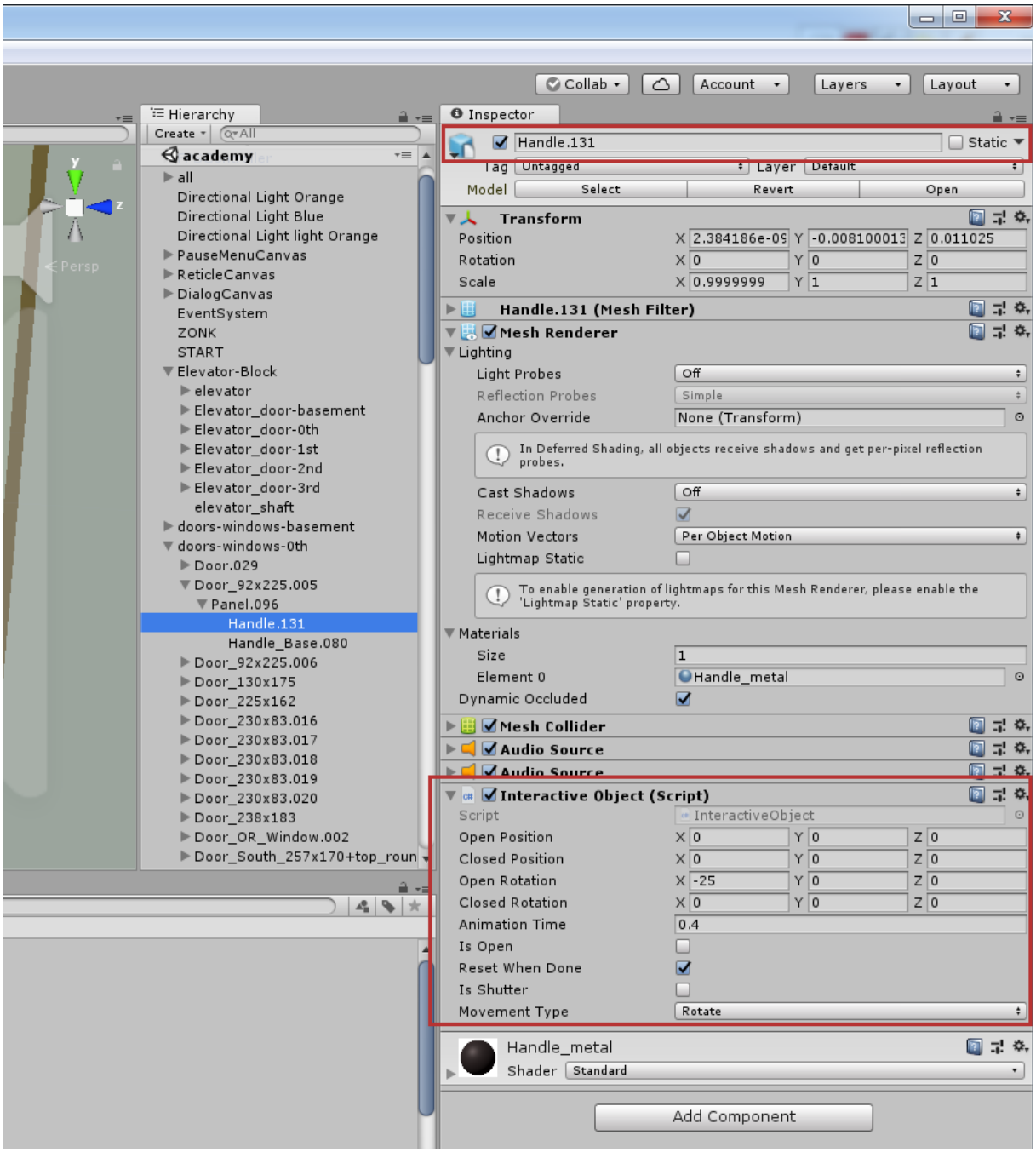
Η διάδραση ενεργοποιείται με το πλήκτρο Ε. Ανάλογα με το είδος της, καλούνται τα κατάλληλα script. Αν η διάδραση αφορά πόρτες και παράθυρα, ο δείκτης γίνεται χέρι, αν αφορά το ασανσέρ όπου πιέζονται τα κουμπιά του γίνεται δείκτης, και τέλος αν πρόκειται να εμφανιστεί παράθυρο διαλόγου γίνεται ερωτηματικό.

6.2.5. InteractiveObject script

Το script καλείται όταν ο δείκτης είναι χέρι και ελέγχει την συμπεριφορά για τις πόρτες, τα παράθυρα και τα κουμπιά του ασανσέρ. Οι κινήσεις ελέγχονται με την χρήση του πρόσθετου iTween που επιτρέπει είτε την περιστροφή είτε την μετατόπιση ενός αντικειμένου. Οι κινήσεις είναι απόλυτα παραμετροποιήσιμες με την βοήθεια του πρόσθετου (εικόνα . 6-11).

Οι vector3 μεταβλητές OpenPosition και ClosedPosition ορίζουν την αρχική και τελική θέση του αντικειμένου, οι OpenRotation και ClosedRotation την αρχική και τελική περιστροφή. Μέσα από την μεταβλητή Movement Type ορίζεται αν το αντικείμενο θα περιστρέφεται ή θα μετατοπίζεται. Η μεταβλητή AnimationTime ορίζει τον χρόνο της κίνησης.

Η boolean μεταβλητή IsOpen χρησιμοποιείται για να ελέγχεται αν το αντικείμενο βρίσκεται σε κατάσταση ανοιχτό ή κλειστό. Η boolean μεταβλητή ResetWhenDone ελέγχει αν μετά την κίνηση του αντικειμένου θα γίνει επιστροφή στην αρχική κατάσταση. Για τα πόμολα στις πόρτες και τα κουμπιά του ασανσέρ γίνεται επιστροφή στην αρχική θέση, για τα πόμολα των παραθύρων όχι. Η boolean μεταβλητή isShutter ελέγχει αν επιτρέπεται η κίνηση του αριστερού παραθυρόφυλλου. Για να επιτρέπεται να ανοίξει ή να κλείσει θα πρέπει να είναι ανοιχτό το δεξί φύλλο.



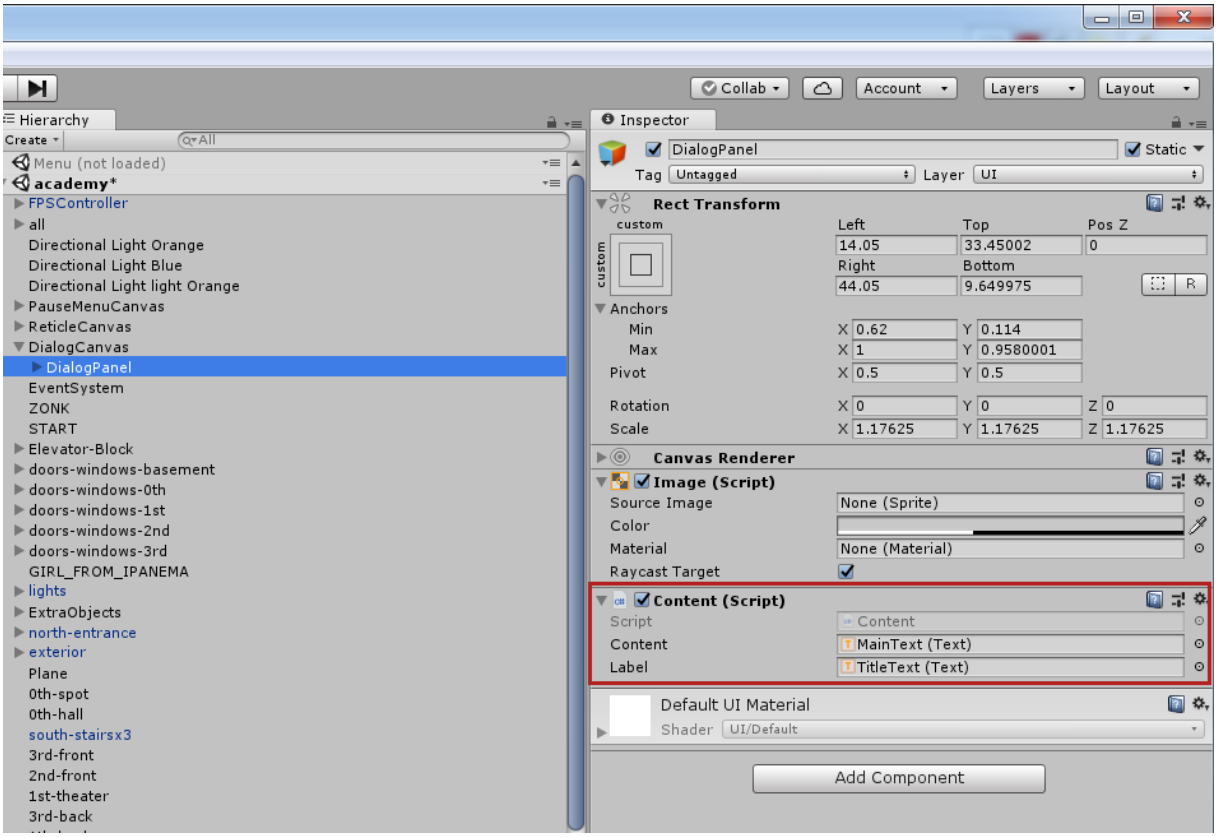
Εικόνα 6-11. Το interactiveObject script όπως είναι προσαρτημένο σε ένα πόμολο πόρτας

Τέλος, για να λειτουργεί σωστά το script θα πρέπει να υπάρχουν δύο πηγές ήχου, μία για τον ήχο του ανοίγματος και μία για τον ήχο του κλεισίματος. Αν δεν υπάρχουν δύο πηγές ήχου το script θα οδηγήσει σε null reference exception.

Ενώ η πρώτη πηγή μπορεί να προστεθεί με drag and drop, δεν μπορεί να προστεθεί δεύτερη με αυτόν τον τρόπο. Θα πρέπει να δοθεί Add Component →Audio→AudioSource και να προστεθεί με drag and drop ο αντίστοιχος ήχος.

6.2.6. Content.cs

Αυτό καλείται από το interactor.cs όταν η διάδραση θα είναι άνοιγμα παραθύρου κειμένου, όταν ο δείκτης δηλαδή γίνεται ερωτηματικό. Η συνάρτηση που ορίζει τι θα εμφανιστεί είναι η ShowTextBox (κώδικας 6-4). Το δισδιάστατο αντικείμενο DialogCanvas είναι γονιός για το DialogPanel, στο οποίο έχει προσαρτηθεί το script. Σε αυτό ορίζονται δυναμικά δύο μεταβλητές με χρήση της λέξης-κλειδί SerializeField. Η Content έχει το περιεχόμενο κείμενο και η Label έχει τον τίτλο του παραθύρου που θα εμφανιστεί.



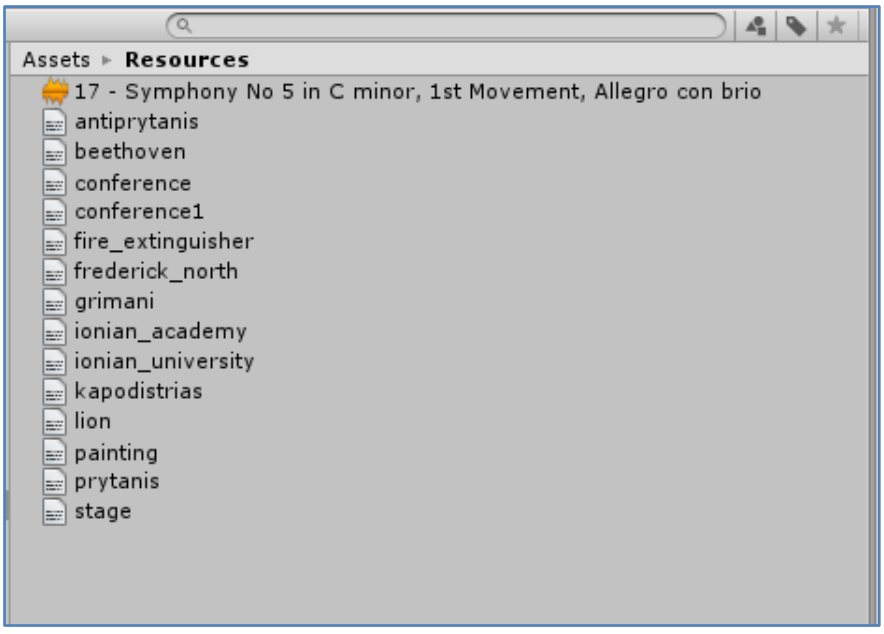
Εικόνα 6-12. Στο παράθυρο διαλόγου έχει προσαρτηθεί το script content.cs

Ενώ ο τίτλος (label.text) ορίζεται στατικά μέσα από το script, το περιεχόμενο (content.text) ορίζεται δυναμικά με την ανάγνωση αρχείων κειμένου. Γίνεται έτσι μία επίδειξη και των δύο αρχών. Να τονιστεί όμως ότι **είναι καλή πρακτική να αποφεύγεται η στατική ανάθεση τιμών** καθώς τυχόν αλλαγές απαιτούν παρέμβαση στον κώδικα. Στην προκειμένη περίπτωση, αν είναι επιθυμητή η αλλαγή του κειμένου που θα εμφανιστεί, αρκεί μια αλλαγή στο αρχείο κειμένου και **δεν γίνεται καμία παρέμβαση στον κώδικα.** Στις γραμμές 35-40 του κώδικα 6-4 φαίνεται πως γίνεται η ανάθεση των τιμών για την προτομή του Καποδίστρια.

```
28      public void ShowTextBox(GameObject gameObject)
29      {
30
31          reticleController.infoPanel.SetActive(true);
32
33          if (gameObject.name == "bust")
34          {
35              title = "Ιωάννης Καποδίστριας";
36              txtFile = "kapodistrias";
37              TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
38              txt = txtAssets.text;
39              content.text = txt;
40              label.text = title;
41          }
42          if (gameObject.name == "painting")
43          {
44              title = "Κάποιος Ζωγράφος";
45              txtFile = "painting";
46              TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
47              txt = txtAssets.text;
48              content.text = txt;
49              label.text = title;
50          }
51          if (gameObject.name == "ionioan-academy-sign")
```

Κώδικας 6-4. Μέρος του κώδικα του script content.cs

Για να λειτουργήσει αυτή η προσέγγιση απαιτείται ένα βήμα που δεν είναι απόλυτα ξεκάθαρο. Θα πρέπει μέσα στα assets να δημιουργηθεί ένας κατάλογος με το όνομα Resources και να εισαχθούν εκεί τα αρχεία κειμένου (εικόνα 6-13).



Εικόνα 6-13. Τα περιεχόμενα του καταλόγου Resources

6.2.7. Μπετόβεν

Ειδική περίπτωση εδώ είναι ο Μπετόβεν. Αν γίνει διάδραση με το γραφείο που βρίσκεται μπροστά από τον πίνακα, ο χρήστης ενημερώνεται ότι ο ένοικος του γραφείου είναι λάτρης του Μπετόβεν. Αν γίνει διάδραση με τον πίνακα, ακούγεται η 5η συμφωνία. Έχει

προσαρτηθεί στον πίνακα το script PlayMusic.cs το οποίο διεκπεραιώνει αυτή την ενέργεια. Επειδή για να φορτωθεί το αρχείο ήχου ακολουθείται μια παρόμοια διαδικασία με τα παράθυρα διαλόγου, το αρχείο ήχου έχει εισαχθεί στα Resources. Για να ακουστεί, χρησιμοποιείται η συνάρτηση Resources.Load<AudioClip>() και στην συνέχεια είναι εύκολο να ελεγχθεί το σταμάτημα και το ξεκίνημα της μουσικής με την συνάρτηση PlayAudio (κώδικας 6-5) η οποία καλείται από το script interactor.cs.

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

public class PlayMusic : MonoBehaviour {
    private AudioSource aSource;

    private void Start()
    {
        aSource = GetComponent<AudioSource>();
        aSource.clip = Resources.Load<AudioClip>("17 - Symphony No 5 in
    }

    public void PlayAudio()
    {
        if (aSource.isPlaying) aSource.Stop(); else aSource.Play();
    }


}

```

Κώδικας 6-5. Το script PlayMusic

6.2.8. Το ασανσέρ

Σε αυτό ο έλεγχος της συμπεριφοράς είναι διαφορετικός λόγω πολυπλοκότητας. Επειδή το ασανσέρ έχει εξωτερικά κουμπιά σε κάθε όροφο και εσωτερικά στον θάλαμο, και πρέπει να ανεβαίνει και να κατεβαίνει όταν δίνεται μια εντολή, η διαχείριση των ενεργειών έχει χωριστεί σε διαφορετικά script. Η εκκίνηση οποιασδήποτε ενέργειας, ξεκινάει από το script Interactor.cs. Το InteractiveObject.cs ελέγχει την κίνηση των κουμπιών, είτε είναι εσωτερικά είτε εξωτερικά και έχει προσαρτηθεί σε όλα τα κουμπιά.

Σε κάθε όροφο υπάρχουν δίφυλλες εξωτερικές πόρτες για το ασανσέρ. Οι κινήσεις τους ελέγχονται με Animation State Machines και το script ElevatorExternalDoors.cs (κώδικας 6-6) το οποίο αφού ελέγξει ποιό από τα κουμπιά πατήθηκε καλεί το script Elevator.cs για να κινηθεί ο θάλαμος. Όταν ο θάλαμος του ασανσέρ φτάσει στον όροφο που πρέπει, ανοίγουν οι πόρτες και ακούγεται ο ήχος ανοίγματος. Αν πατηθεί κάποιο εσωτερικό κουμπί ή μετά την

παρέλευση κάποιου χρονικού διαστήματος, οι πόρτες κλείνουν και ακούγεται ο ήχος κλεισίματος. Το τι θα ακουστεί ελέγχεται από το script ExternalElevatorDoorsSound.cs.

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

public class ElevatorExternalDoors : MonoBehaviour {
    private Elevator elevator;

    // Use this for initialization
    void Start() {
        elevator = GameObject.Find("elevator").GetComponent<Elevator>();



    }

    // Update is called once per frame
    void Update () {

    }
    public void ButtonPressed()
    {
        if (this.transform.name == "Elevator-button-basement")
            elevator.ButtonPressed (GameObject.Find ("-1"));
        if (this.transform.name == "Elevator-button-0th")
            elevator.ButtonPressed (GameObject.Find ("0"));
        if (this.transform.name == "Elevator-button-1st")
            elevator.ButtonPressed (GameObject.Find ("1"));
        if (this.transform.name == "Elevator-button-2nd")
            elevator.ButtonPressed (GameObject.Find ("2"));
        if (this.transform.name == "Elevator-button-3rd")
            elevator.ButtonPressed (GameObject.Find ("3"));
    }
}
```

Κώδικας 6-6. Το script ElevatorExternalDoors.cs

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

[System.Serializable]
public class m_ExternalDoorElevatorSounds
{
    public AudioClip ElevatorDoorsOpen;
    public AudioClip ElevatorDoorsClose;
}

    public class ExternalElevatorDoorsSound : MonoBehaviour {

    public m_ExternalDoorElevatorSounds ExternalDoorsElevatorSounds = new m_ExternalDoorElevatorSounds();
    private Animator ExternalDoorsAnimation;
    // Use this for initialization
    void Start () {
        this.GetComponent<AudioSource>().Stop();
    }

    // Update is called once per frame
    public void OpenDoorsSound()
    {
        this.GetComponent<AudioSource>().PlayOneShot(ExternalDoorsElevatorSounds.ElevatorDoorsOpen);
    }

    public void CloseDoorsSound()
    {
        this.GetComponent<AudioSource>().PlayOneShot(ExternalDoorsElevatorSounds.ElevatorDoorsClose);
    }
}
```

Κώδικας 6-7. Το script ExternalElevatorDoorsSound.cs

Ο έλεγχος της κίνησης του θαλάμου, της πόρτας του θαλάμου, αλλά και εξωτερικών πορτών γίνεται από το script ElevatorController. Το άνοιγμα και κλείσιμο των πορτών γίνεται πάλι με Animation State Machines. Και για τις εξωτερικές πόρτες και για την πόρτα του θαλάμου, το άνοιγμα ή κλείσιμο ελέγχονται από boolean μεταβλητές των οποίων η τιμή αλλάζει κατάλληλα από αυτό το script.

```
129     void FixedUpdate()
130     {
131
132         Transform ElevatorFloor = elevatorFloors[floorNumber];
133
134         if (Mathf.Round(this.transform.position.y) == Mathf.Round(elevatorFloors[0].position.y))
135         {
136             levelDisplay.ChangeTexture(levelDisplay.Level_M1);
137             externalLevelDisplayM1.ChangeTexture(externalLevelDisplayM1.Level_M1);
138             externalLevelDisplay0.ChangeTexture(externalLevelDisplay0.Level_M1);
139             externalLevelDisplay1.ChangeTexture(externalLevelDisplay1.Level_M1);
140             externalLevelDisplay2.ChangeTexture(externalLevelDisplay2.Level_M1);
141             externalLevelDisplay3.ChangeTexture(externalLevelDisplay3.Level_M1);
142         }
143         if (Mathf.Round(this.transform.position.y) == Mathf.Round(elevatorFloors[1].position.y))
144         {
145             levelDisplay.ChangeTexture(levelDisplay.Level_0);
146             externalLevelDisplayM1.ChangeTexture(externalLevelDisplayM1.Level_0);
147             externalLevelDisplay0.ChangeTexture(externalLevelDisplay0.Level_0);
148             externalLevelDisplay1.ChangeTexture(externalLevelDisplay1.Level_0);
149             externalLevelDisplay2.ChangeTexture(externalLevelDisplay2.Level_0);
150             externalLevelDisplay3.ChangeTexture(externalLevelDisplay3.Level_0);
151
152         }
153
154         if (Mathf.Round(this.transform.position.y) == Mathf.Round(elevatorFloors[2].position.y))
155         {
156             levelDisplay.ChangeTexture(levelDisplay.Level_1);
157             externalLevelDisplayM1.ChangeTexture(externalLevelDisplayM1.Level_1);
158             externalLevelDisplay0.ChangeTexture(externalLevelDisplay0.Level_1);
159             externalLevelDisplay1.ChangeTexture(externalLevelDisplay1.Level_1);
160             externalLevelDisplay2.ChangeTexture(externalLevelDisplay2.Level_1);
161             externalLevelDisplay3.ChangeTexture(externalLevelDisplay3.Level_1);
162         }
```

Κώδικας 6-8. Μέρος του script ElevatorController.cs που ελέγχει την ένδειξη εμφανίζει το ασανσέρ

Εξ ορισμού, τα classes της C# δημιουργούνται με την Update και όχι την FixedUpdate όπως εδώ. Η τεκμηρίωση του Unity αναφέρει ότι όταν το αντικείμενο έχει προσαρτημένο RigidBody πρέπει να χρησιμοποιείται η FixedUpdate. Επειδή η FixedUpdate καλείται πριν γίνουν οι υπολογισμοί ενώ η Update μία φορά σε κάθε frame, και εδώ ενδιαφέρει να λαμβάνονται υπόψη οι φυσικοί νόμοι, η χρήση της FixedUpdate επιβάλλεται. Αν η FixedUpdate αντικατασταθεί από την Update, το script δεν θα λειτουργεί σωστά.

Το script Elevator.cs ελέγχει ποιό κουμπί μέσα στον θάλαμο πατιέται, και καλεί το ElevatorController για την εκτέλεση των απαιτούμενων ενεργειών. Στον κώδικα 6-9 φαίνεται μέρος του script με την συνάρτηση ButtonPressed, που αφού ελέγξει ποιό κουμπί στον θάλαμο πατήθηκε, καλεί το script ElevatorController ώστε να εκτελεστούν οι αντίστοιχες ενέργειες.

```
37      public void ButtonPressed(GameObject button)
38      {
39          if (button.name == "-1")
40          {
41
42              elevatorController.ElevatorGO(button.tag);
43              elevatorController.floorNumber = 0;
44          }
45          if (button.name == "0")
46          {
47
48              elevatorController.ElevatorGO(button.tag);
49              elevatorController.floorNumber = 1;
50
51          }
52          if (button.name == "1")
53          {
54
55              elevatorController.ElevatorGO(button.tag);
56              elevatorController.floorNumber = 2;
57          }
58          if (button.name == "2")
59          {
60
61              elevatorController.ElevatorGO(button.tag);
62              elevatorController.floorNumber = 3;
63          }
64          if (button.name == "3")
65          {
66
67              elevatorController.ElevatorGO(button.tag);
68              elevatorController.floorNumber = 4;
69          }
```

Κώδικας 6-9. Μέρος του Elevator.cs που καλεί το ElevatorController.cs με τις κατάλληλες τιμές

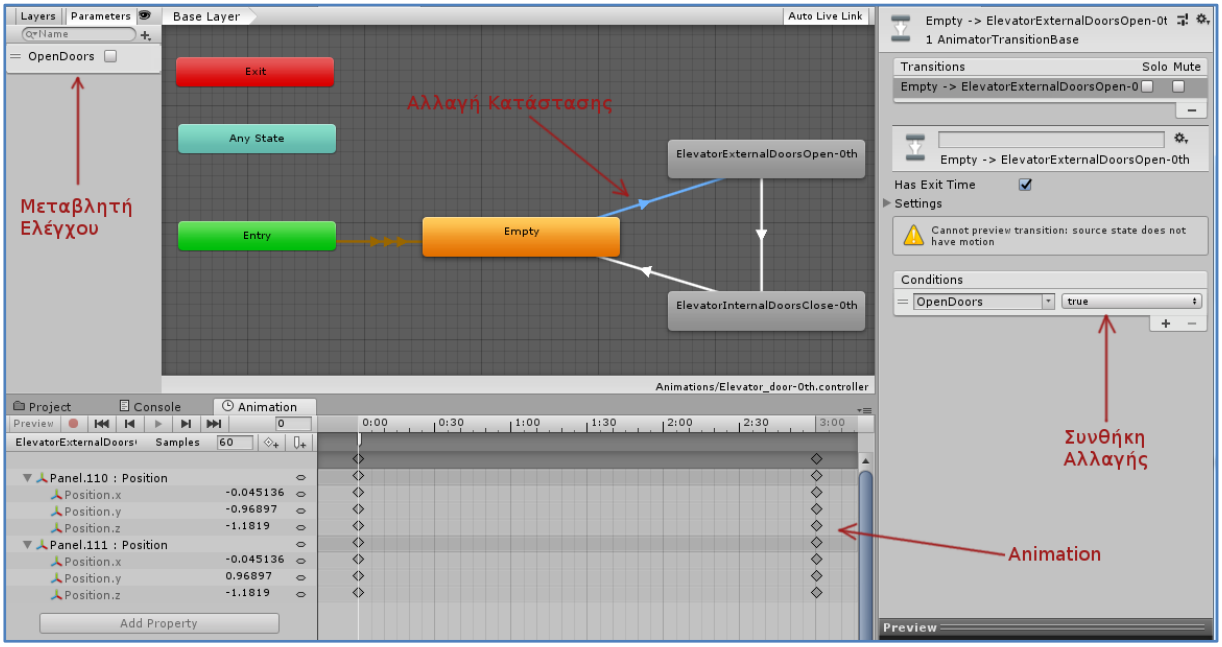
Τέλος, να επισημανθεί ότι για να εκτελούνται με την σωστή χρονική ακολουθία οι ενέργειες, χρησιμοποιείται η συνάρτηση IEnumerator στο Interactor και στο ElevatorController. Μέσα σε αυτήν επιτρέπεται η χρήση της εντολής yield ώστε να υπάρχει αναμονή χρόνου πριν εκτελεστεί μια εντολή. Στον κώδικα 6-10, η IEnumerator ελέγχει (GetBool) και αλλάζει (SetBool) μεταβλητές οι οποίες ανάλογα με την τιμή τους ενεργοποιούν τους Animators ανοίγματος ή κλεισίματος για τις πόρτες του ασανσέρ. Το internalDoorAnimation είναι για τις πόρτες του θαλάμου,ενώ το externalDoorAnimation για τις πόρτες στους ορόφους. Η διαφορά στον χρόνο κίνησής τους είναι 1,5 δευτερόλεπτα. Αν οι πόρτες είναι ανοιχτές και ο χρήστης δεν κάνει καμία κίνηση, μετά από 7 δευτερόλεπτα οι πόρτες κλείνουν.

```
303     IEnumerator OpenAndCloseDoorsWithDelay()
304     {
305         if (!internalDoorAnimation.GetBool("OpenDoors"))
306         {
307             switch (floorNumber)
308             {
309                 case 0:
310                     externalDoorAnimation = GameObject.Find("Elevator_door-basement").GetComponent<An
311                     if (!externalDoorAnimation.GetBool("OpenDoors")) externalDoorAnimation.SetBool("O
312                     break;
313                 case 1:
314                     externalDoorAnimation = GameObject.Find("Elevator_door-0th").GetComponent<Animato
315                     if (!externalDoorAnimation.GetBool("OpenDoors")) externalDoorAnimation.SetBool("O
316                     break;
317                 case 2:
318                     externalDoorAnimation = GameObject.Find("Elevator_door-1st").GetComponent<Animato
319                     if (!externalDoorAnimation.GetBool("OpenDoors")) externalDoorAnimation.SetBool("O
320                     break;
321                 case 3:
322                     externalDoorAnimation = GameObject.Find("Elevator_door-2nd").GetComponent<Animato
323                     if (!externalDoorAnimation.GetBool("OpenDoors")) externalDoorAnimation.SetBool("O
324                     break;
325                 case 4:
326                     externalDoorAnimation = GameObject.Find("Elevator_door-3rd").GetComponent<Animato
327                     if (!externalDoorAnimation.GetBool("OpenDoors")) externalDoorAnimation.SetBool("O
328                     break;
329             }
330             yield return new WaitForSeconds(1.5f);
331             internalDoorAnimation.SetBool("OpenDoors", !internalDoorAnimation.GetBool("OpenDoors"));
332             //wait 7 seconds and if nothing happens close the doors?
333             if (externalDoorAnimation.GetBool("OpenDoors"))
334             {
335                 yield return new WaitForSeconds(7);
336
337                 externalDoorAnimation.SetBool("OpenDoors", false);
338                 yield return new WaitForSeconds(1.5f);
339                 internalDoorAnimation.SetBool("OpenDoors", false);
340
341             }
342
343         }
344         else
345         {
346             internalDoorAnimation.SetBool("OpenDoors", !internalDoorAnimation.GetBool("OpenDoors"));
```

Κώδικας 6-10. IEnumerator

6.2.9. Animators

Για να είναι συνεπής η δομή με τα προηγούμενα, δημιουργήθηκε στα Assets ένας φάκελος Animations. Μέσα σε αυτόν δημιουργήθηκαν οι Animator Controllers και τα Animations. Οι Animator Controllers είναι στην ουσία μηχανές καταστάσεων. Ανάλογα με την τιμή κάποιων μεταβλητών γίνεται αλλαγή κατάστασης. Τα Animations ορίζουν την κίνηση που θα γίνει και τον χρόνο της μεταβολής των αλλαγών στις καταστάσεις. Στην εικόνα 6-14 η τιμή της boolean μεταβλητής καθορίζει την μετάβαση από κλειστό σε ανοιχτό για τις εξωτερικές πόρτες του ασανσέρ στο ισόγειο. Ο χρόνος του Animation έχει οριστεί στα 3 δευτερόλεπτα. Αρχικά οι πόρτες είναι κλειστές,, η μεταβλητή είναι ψευδής. Όταν η τιμή αλλάξει σε αληθής, η κατάσταση αλλάζει και οι πόρτες ανοίγουν, όπως φαίνεται από το επιλεγμένο βέλος σε μπλε χρώμα. Η τιμή της μεταβλητής αλλάζει με κώδικα, από το IEnumerator στο ElevatorController. Κάτι ανάλογο συμβαίνει και για τις πόρτες του θαλάμου και τους υπόλοιπους ορόφους.



Εικόνα 6-14. Animation Controller και Animation για τις εξωτερικές πόρτες του ασανσέρ στο ισόγειο.

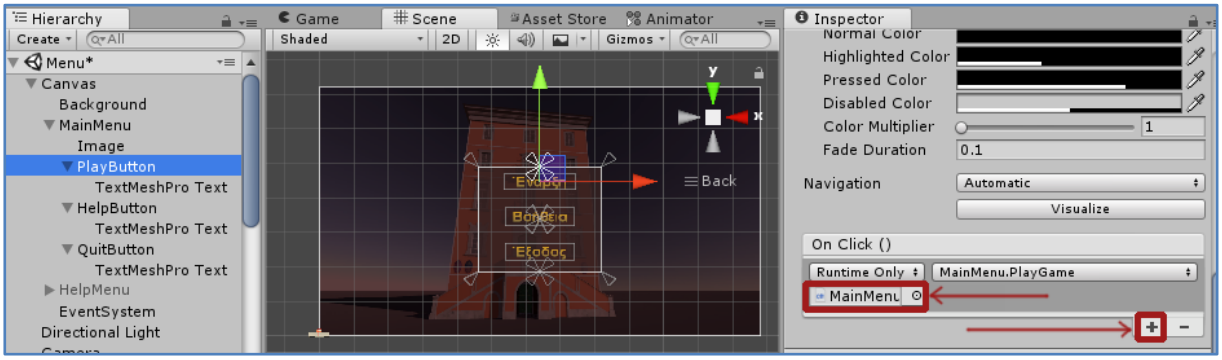
6.2.10. Το μενού Έναρξης και το μενού Παύσης.

Στο μενού χρησιμοποιήθηκε το πρόσθετο TextMeshPro καθώς δίνει πολύ καλύτερα αποτελέσματα από το ενσωματωμένο αντικείμενο Text του Unity. Τα μενού είναι αντικείμενα στις δύο διαστάσεις και είναι διαθέσιμα από το κεντρικό μενού του Unity ακολουθώντας την διαδρομή GameObject → UI. Οι ενδιαφερόμενοι μπορούν να ανατρέξουν σε σχετική βιβλιογραφία για διεξοδική ανάλυση και εξήγησή τους (Simon 2015).

Η διαδικασία δημιουργίας του μενού είναι σχετικά απλή (Brackeys, START MENU in Unity 2017). Δημιουργήθηκε μια νέα σκηνή που ονομάστηκε Menu. Σε αυτό προστέθηκε ένα αντικείμενο Canvas που μετονομάστηκε σε MainMenu. Σε αυτό, προστέθηκαν σαν παιδιά η εικόνα υποβάθρου και τρία Panel[15] με κουμπιά για την Έναρξη του παιχνιδιού, την εμφάνιση του μενού βοήθειας και την έξοδο από το παιχνίδι. Όλα προστίθενται είτε με δεξί κλικ στο Hierarchy, είτε από το μενού του Unity δίνοντας GameObject→UI και επιλέγοντας το αντίστοιχο αντικείμενο. Για παράδειγμα, τα κουμπιά προστέθηκαν κάνοντας δεξί κλικ στο

[15] Απαιτείται να υπάρχει Canvas. Αν προστεθεί ένα Panel στο Hierarchy χωρίς Canvas, το Canvas προστίθεται αυτόματα σαν γονιός.

αντικείμενο γονιό και έπειτα δίνοντας UI→Button. Σε αυτά προστίθεται από το Unity ένα script Component με μια συνάρτηση OnClick(), που εξ ορισμού δεν κάνει τίποτε. Μπορεί όμως να ρυθμιστεί κατάλληλα ώστε να συμβαίνει ένα γεγονός πιέζοντας το +, όπως στην εικόνα 6-15 όπου έχει ήδη πατηθεί και ρυθμιστεί ώστε να εκτελεί το script MainMenu που έχει μια συνάρτηση PlayGame η οποία φορτώνει την σκηνή του παιχνιδιού. Ανάλογα προγραμματίστηκαν και τα υπόλοιπα κουμπιά.



Εικόνα 6-15. Το PlayButton δείχνει κείμενο "Έναρξη" και όταν πατηθεί εκτελείται το script MainMenu.PlayGame

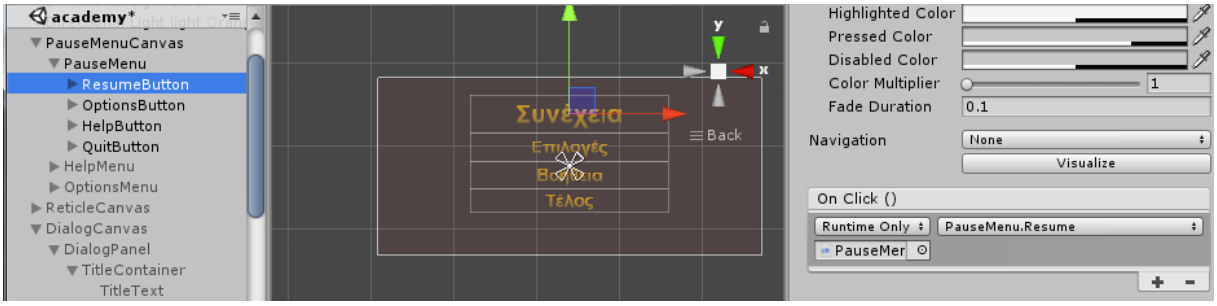
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class MainMenu : MonoBehaviour {

    public void PlayGame ()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    public void QuitGame ()
    {
        //Debug.Log("Quit!");
        Application.Quit();
        Application.OpenURL("about:blank");
    }
}
```

Κώδικας 6-11. Το script MainMenu.

Η σκηνή του μενού και η κύρια σκηνή του παιχνιδιού πρέπει να φορτωθούν στις ρυθμίσεις κατασκευής του τελικού εκτελέσιμου. Αυτό γίνεται από το μενού File→Build Settings και στο Scenes in Build τις σκηνές με Drag and Drop.

Με παρόμοιο τρόπο δημιουργήθηκε το μενού Παύσης του παιχνιδιού (Brackeys, PAUSE MENU in Unity 2017). Δεν έχει δημιουργηθεί ξεχωριστή σκηνή για αυτό, είναι μέρος της κύριας σκηνής (academy). Ενεργοποιείται και απενεργοποιείται με το κατάλληλο πλήκτρο από το πληκτρολόγιο του υπολογιστή, για την ακρίβεια ενεργοποιείται είτε με το πλήκτρο P

είτε με το Esc, το οποίο όμως πρέπει να πατηθεί δύο φορές αν η εφαρμογή λειτουργεί σε WebGL. Το πρώτο Esc είναι αφιερωμένο στον έλεγχο του παραθύρου του παιχνιδιού από τον Πλοηγητή.



Εικόνα 6-16. Το ResumeButton δείχνει κείμενο "Συνέχεια" και εκτελεί το script PauseMenu.Resume

Στην εικόνα 6-16 φαίνεται ο προγραμματισμός του ResumeButton στον Inspector.

Τα κουμπιά OptionsButton και HelpButton εμφανίζουν νέα μενού, το πρώτο για ρυθμίσεις του παιχνιδιού και το άλλο με οδηγίες. Το QuitButton τερματίζει το παιχνίδι.

6.2.11. Το παράθυρο διαλόγου.

Ακολουθήθηκε μια παρόμοια μέθοδος δημιουργίας όπως για το κύριο μενού και το μενού παύσης. Η ιδιαιτερότητά του είναι ότι υπάρχει περιοχή κύλισης του κειμένου με μια μπάρα κύλισης συνδεμένη με αυτήν (Zotov 2017), (Bui 2016), (Brackeys, Content Scrolling - Unity 4.6 Tutorial 2014).

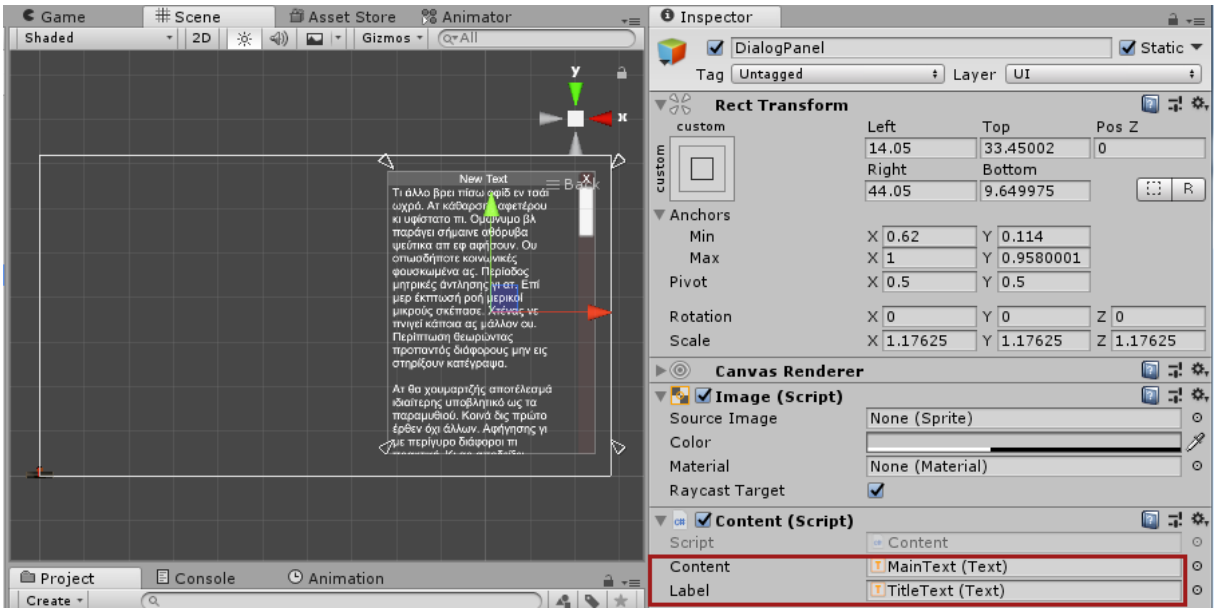
Έτσι, δημιουργήθηκε ένα Canvas που ονομάστηκε DialogCanvas του οποίου το Scale Mode πρέπει οπωσδήποτε να αλλαχτεί σε Scale with Screen Size[16]. Σε αυτό προσαρτήθηκε το script DialogBox, που κλειδώνει τον δείκτη του ποντικιού όταν το παράθυρο κλείσει.

Σαν παιδί σε αυτό προστέθηκε ένα Panel (DialogPanel) στο οποίο προσαρτήθηκε το content.cs (εικόνα 6-17), το οποίο, μεταξύ άλλων κλειδώνει τον δείκτη του ποντικιού. Σαν παιδιά στο DialogPanel προστέθηκαν ένα Panel (TitleContainer) με παιδί ένα TextBox για τον Τίτλο του παραθύρου (TitleText), άλλο ένα Panel (ScrollArea) με παιδί ένα TextBox (MainText) για το κυρίως κείμενο και ένα ScrollBar (Add Component → UI → Scrollbar). Στο ScrollArea προστέθηκε μια μάσκα για να εμφανίζεται σωστά το κυλιόμενο περιεχόμενο (Add Component → UI → Mask).

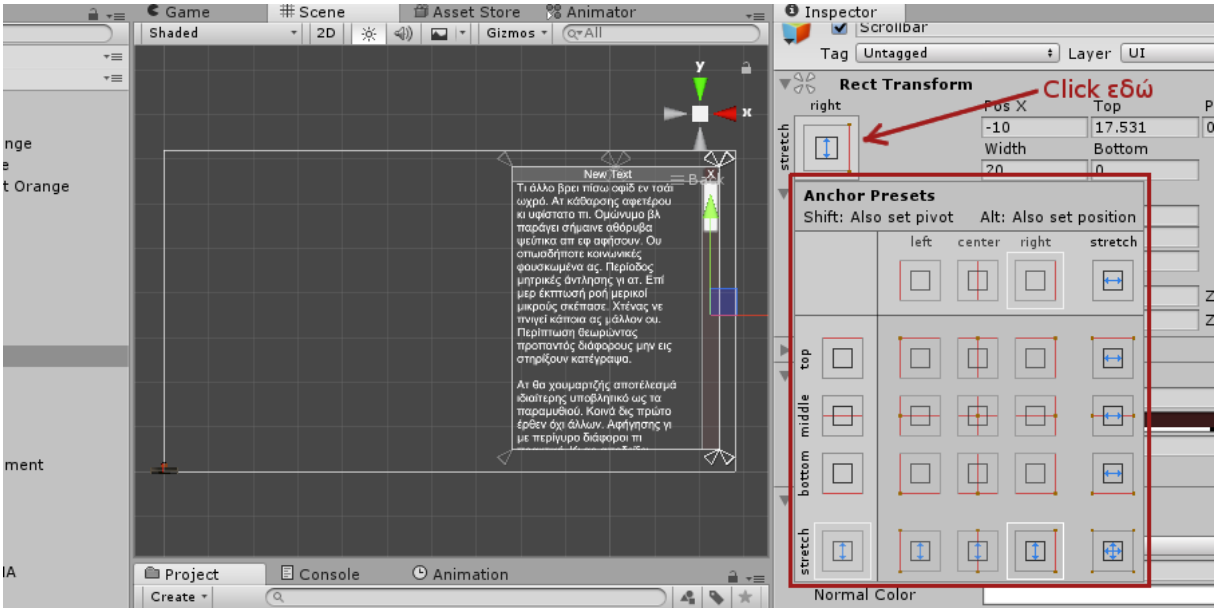
[16] Για να προσαρμόζονται σωστά τα στοιχεία σε τυχόν αλλαγή του μεγέθους του παραθύρου του παιχνιδιού.

Για τα προαναφερόμενα πρέπει να αλλαχτεί κατάλληλα το μέγεθος και η στοίχιση ώστε να μην δημιουργηθούν δυσάρεστα αποτελέσματα . Η θέση των επιμέρους στοιχείων κατά την φάση εκτέλεσης του προγράμματος μπορεί να αλλάξει αν δεν γίνει σωστά ο σχεδιασμός.

Αν τα προηγούμενα φαίνονται ακατανόητα, μελετώντας τις αναφορές γίνονται κατανοητά.



Εικόνα 6-17. Το DialogPanel



Εικόνα 6-18. Στοίχιση για το ScrollBar. Το ίδιο έγινε και για τα υπόλοιπα συστατικά.

6.2.12. WebGL

Αφού ολοκληρωθεί η φάση της σχεδίασης απομένει η φάση της παράδοσης του τελικού προϊόντος. Η επιλογή εξαρτάται από το κοινό στο οποίο απευθύνεται αυτό. Το Unity έχει την δυνατότητα να παράγει το τελικό εκτελέσιμο για υπολογιστές που τρέχουν Windows, OSX, Linux, για φορητές συσκευές που τρέχουν iOS ή Android[17], για Xbox ή Playstation. Ακόμα και εφαρμογή συμβατή με το facebook μπορεί να παραχθεί[18].

Κατά την φάση εγκατάστασης του Unity υπάρχει η επιλογή ποιές από τις δυνατότητες αυτές θα εγκατασταθούν.

Η τελική επιλογή ήταν να δημιουργηθεί το αποτέλεσμα σε WebGL (εικόνα 6-19).

Το WebGL ξεκίνησε το 2006 από τα πειράματα του Vladimir Vukicevic στην Mozilla (Wikipedia 2018). Είναι ένα API της Javascript και δημιουργήθηκε για να φέρει τα τρισδιάστατα γραφικά στους πλοηγητές του διαδικτύου χωρίς πρόσθετα. Επειδή οι εντολές δημιουργούνται σε GLSL και το WebGL API τις μεταφράζει σε εντολές που απευθύνονται άμεσα στην κάρτα γραφικών, οι επιδόσεις εξαρτώνται άμεσα από αυτήν. Όσο καλύτερη, τόσο πιο ομαλά θα λειτουργεί η εφαρμογή.

Μέχρι σήμερα υπάρχουν δύο πλαίσια, το WebGL 1.0 και το WebGL 2.0. Ενώ το πρώτο βασίζεται στο OpenGL ES 2.0 και χρησιμοποιεί το στοιχείο canvas της HTML 5, το δεύτερο βασίζεται στο OpenGL ES 3.0 και προσφέρει πολλές επεκτάσεις που για το WebGL 1.0 είναι προαιρετικές.

Το πλεονέκτημα του WebGL είναι ότι, εν δυνάμει, μπορεί να λειτουργεί σε οποιαδήποτε πλατφόρμα. Η τελική έκδοση της Ιονίου Ακαδημίας μπορεί να λειτουργεί σε όλα τα λειτουργικά συστήματα, σε smartphones, σε tablets, σε τηλεοράσεις, σε οποιαδήποτε συσκευή, χωρίς να γίνουν τροποποιήσεις. Μοναδική επιφύλαξή μου εδώ είναι το πως θα λειτουργεί η διάδραση χωρίς την ύπαρξη πληκτρολογίου και ποντικιού, αλλά αυτό δεν είναι περιορισμός του WebGL, είναι καθαρά θέμα σχεδιασμού της εφαρμογής.

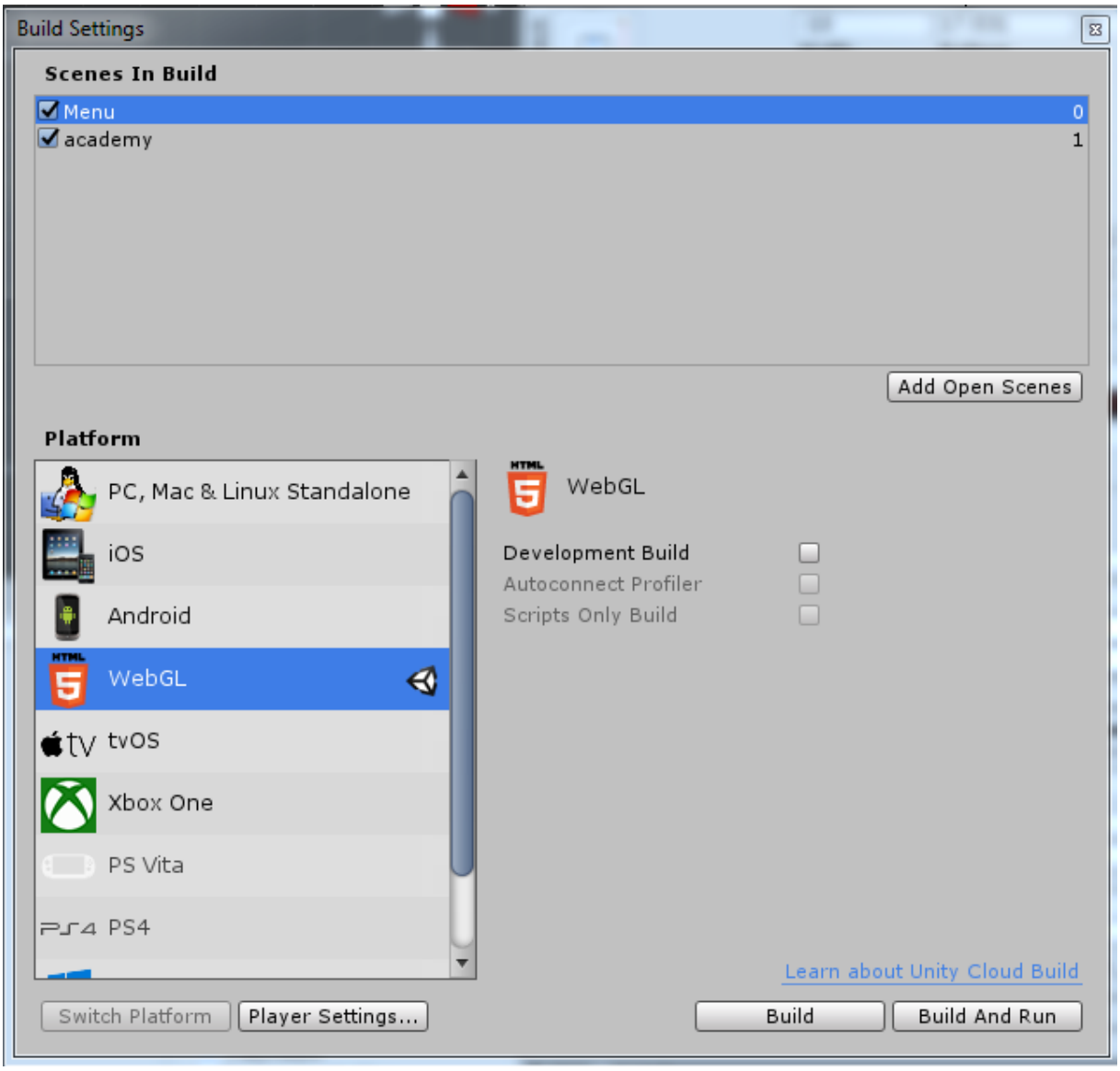
Η εφαρμογή δοκιμάστηκε στα τρία λειτουργικά συστήματα επιτραπέζιων υπολογιστών, τα Windows, το Mac OSX και το Linux. Λειτουργεί άψογα με τους πλοηγητές Mozilla Firefox και Google Chrome. Ο Safari 7 στο OSX Mavericks δεν λειτουργεί, ο Edge της Microsoft δεν

[17] Για τις χαμηλών προδιαγραφών συσκευές είναι αμφίβολο αν θα λειτουργεί ικανοποιητικά.

[18] Δεν έχω facebook και αρνούμαι πεισματικά να δημιουργήσω λογαριασμό στο facebook.

δοκιμάστηκε. Στην τεκμηρίωση του Unity δίνονται πληροφορίες για την συμβατότητα των Browser (Unity_Technologies 2018)

Στο Unity, η επιλογή του τελικού αποτελέσματος σε WebGL γίνεται από το μενού του Editor δίνοντας File → Build Settings, και επιλέγοντας για επιθυμητή μορφή το WebGL.



Εικόνα 6-19. Επιλογή της μορφής του εκτελέσιμου

Οι ρυθμίσεις ποιότητας στο Unity γίνονται κατά κύριο λόγο από τα Player Settings της εικόνας 6-19, και από το Edit → Project Settings → Quality. Σε συνάρτηση με τον φωτισμό. επηρεάζουν και οι ρυθμίσεις στο Window → Rendering → Lighting Settings.

Επειδή η εφαρμογή δεν απευθύνεται σε κοινό που έχει στην διάθεσή του υπολογιστή για παιχνίδια, δόθηκε προσοχή στο να λειτουργεί χωρίς προβλήματα σε παλιότερους

υπολογιστές. Έγιναν δοκιμές για αναλύσεις σε λόγο 16/9 που κυμαίνονται από 630x360 ως 1280x720 για υψηλή, μεσαία και χαμηλή ποιότητα.

Σε ένα σύστημα με κάρτα γραφικών παλιότερης γενιάς[19] η εφαρμογή λειτουργεί ικανοποιητικά για μεσαίες και χαμηλές ρυθμίσεις ποιότητας. Στις υψηλές ρυθμίσεις υπάρχει αισθητή υστέρηση στις επιδόσεις.

[19] Intel i5-3570, 32G ram, Nvidia 650ti 2G vram.

Συμπεράσματα

Η μεταπτυχιακή αυτή εργασία, ξεκινά με μια ιστορική αναδρομή του κτιρίου της Ιονίου Ακαδημίας. Από την κατασκευή του κατά την διάρκεια της Ενετοκρατίας, το κτίριο ήταν μάρτυρας πολλών περιπετειών. Αρχικά εξυπηρετούσε ως στρατώνας. Υπήρξε η έδρα του πρώτου πανεπιστημίου της Ελλάδας μετά τις ακούραστες προσπάθειες του Φρέντερικ Νορθ, κόμη του Γκίλφορντ. Φιλοξένησε σχολεία, αλλά και την Βιβλιοθήκη της Κέρκυρας. Βομβαρδίστηκε με εμπρηστικές βόμβες στον Δεύτερο Παγκόσμιο Πόλεμο. Έμεινε ερειπωμένη μέχρι το 1994 όταν ανοικοδομήθηκε. Σήμερα εξυπηρετεί ως έδρα του Ιόνιου Πανεπιστήμιου.

Από την πλούσια ιστορία της Κέρκυρας, επιλέχτηκε αυτό το κτίριο για να ψηφιοποιηθεί στις τρεις διαστάσεις με την χρήση του προγράμματος Blender, και στην συνέχεια να ζωντανέψει με την εισαγωγή στην μηχανή παιχνιδιών Unity.

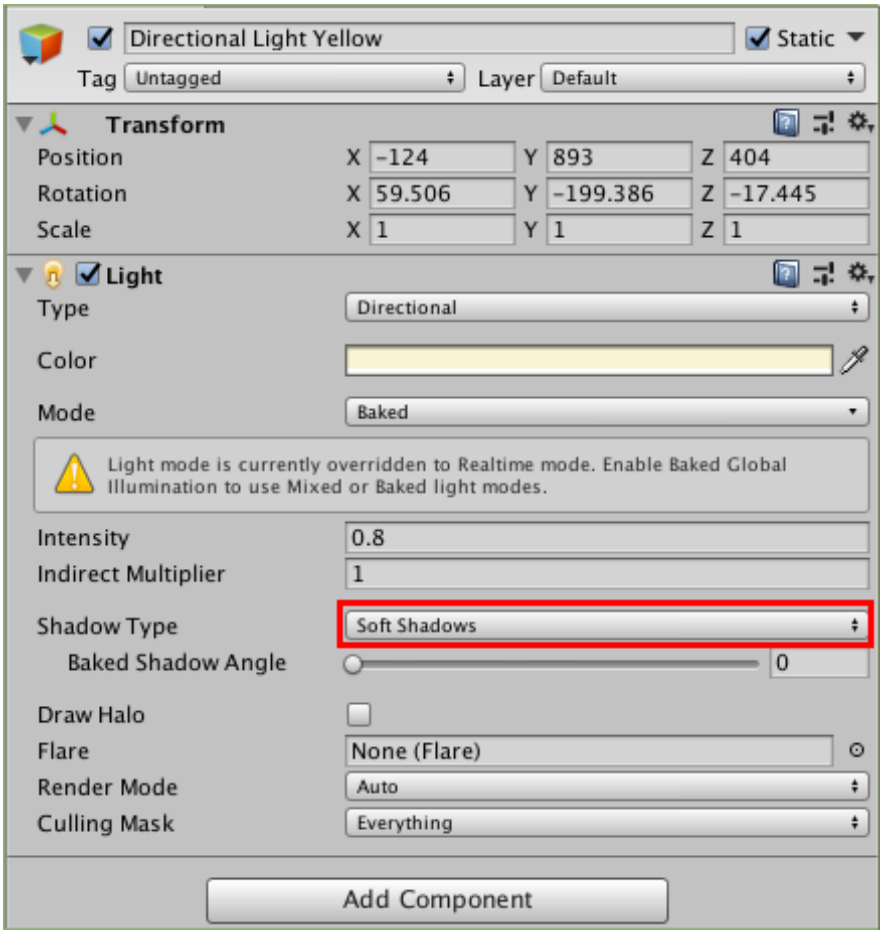
Το 3D Modeling έγινε με το Blender 2.79c, με το Cycles Render Engine. Τα objects οργανώθηκαν με λογική ανάλογη με το είδος και την σημασία τους σε layers. Αποδόθηκαν τα κατάλληλα materials, όπου απαιτούνταν χρησιμοποιήθηκαν textures, και έγινε UV Unwrapping. Για τις πόρτες και τα παράθυρα δημιουργήθηκαν σχέσεις parent-child, με τους άξονες κίνησης και περιστροφής που δεν χρησιμοποιούνταν να είναι κλειδωμένοι.

Χρησιμοποιήθηκε η μηχανή παιχνιδιών Unity για την ανάπτυξη μιας τρισδιάστατης εκπαιδευτικής εφαρμογής. Ο Unity Editor με την κατάλληλη χρήση του Hierarchy και του Inspector, την προσθήκη και παραμετροποίηση components στα objects, την συγγραφή script, οδήγησε στο τελικό αποτέλεσμα για το οποίο η τελική έξοδος επιλέχτηκε να είναι σε WebGL. Αυτό δίνει την δυνατότητα να τρέχει η εφαρμογή σε οποιοδήποτε λειτουργικό σύστημα με χρήση κάποιου Browser που υποστηρίζει το WebGL.

Το Unity διδάσκεται βέβαια στο μεταπτυχιακό πρόγραμμα σπουδών, όμως επιλέχτηκε κατά βάση για την ιδιαιτερότητά του να διατίθεται δωρεάν για προσωπική χρήση. Πρόθεση των δημιουργών, είναι να "δημοκρατικοποιηθεί" και να γίνει διαθέσιμη η ανάπτυξη παιχνιδιών στον οποιονδήποτε ενδιαφερόμενο (Wikipedia, Unity Technologies 2018).

Αν και η εξερεύνηση των λειτουργιών και δυνατοτήτων του Unity προσφέρει πολλές επιλογές, κάποια χαρακτηριστικά δεν είναι διαθέσιμα στην ελεύθερη έκδοση.

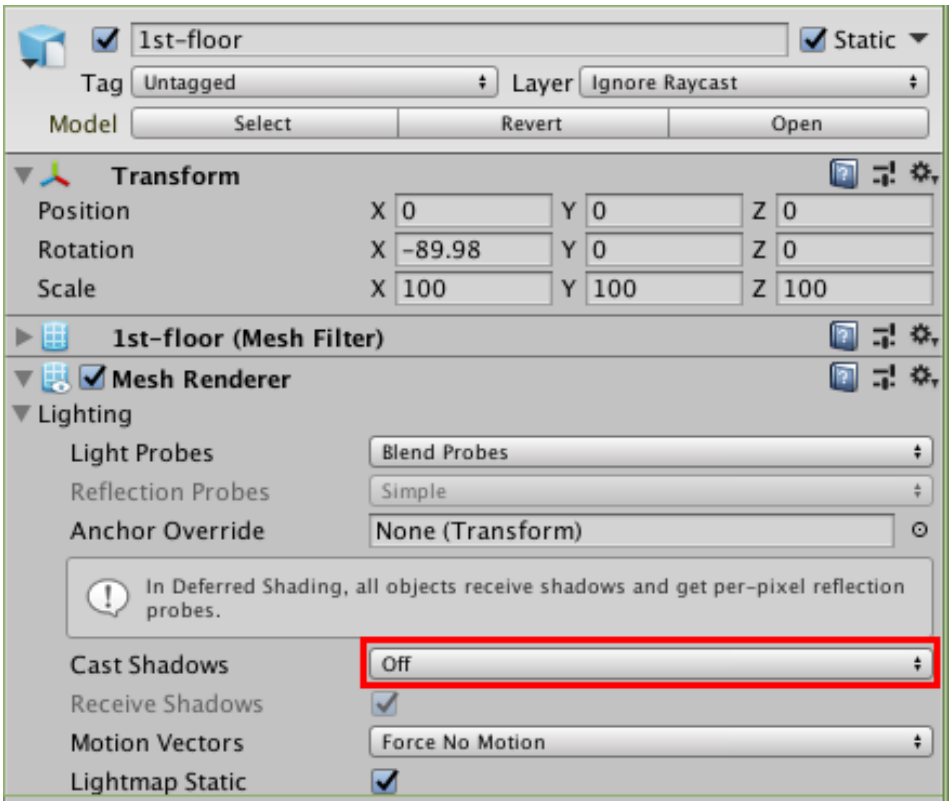
Ο προσεκτικός παρατηρητής θα διαπιστώσει ότι το τελικό αποτέλεσμα της εφαρμογής που αναπτύχθηκε δεν έχει σκιές. Αυτό γίνεται για δύο λόγους. Ο πρώτος είναι ότι το φως που έχει δυνανότητα παραγωγής σκιών είναι το Directional Light (εικόνα Σ-1).



Εικόνα Σ- 1. Directional Light

Θα πρέπει κάποιο αντικείμενο που ενδιαφέρει να έχει σκιά, να ρυθμιστεί κατάλληλα. Το πάτωμα ενός ορόφου είναι οροφή για τους ορόφους κάτω από αυτόν. Αυτό δημιουργεί προβλήματα επειδή το Directional Light εκπέμπει φως και σκιές ομοιογενώς σε όλη την σκηνή. Το αντικείμενο ορίζεται αν θα εκπέμπει σκιές. Το να δέχεται σκιές δεν είναι επιλέξιμο (εικόνα Σ-2). Ένα γραφείο του πρώτου ορόφου θα ρίχνει σκιά στο πάτωμα του πρώτου ορόφου και στο ισόγειο αν δεν έχει ρυθμιστεί να εκπέμπει σκιές. Αν όμως ρυθμιστεί να εκπέμπει σκιές, η σκιά θα καλύπτει τον αποκάτω όροφο με αποτέλεσμα οι σκιές των αντικειμένων να μην είναι ορατές.

Ο δεύτερος λόγος είναι ότι τα Spot Light και Area Light των οποίων το εύρος ελέγχεται δεν εκπέμπουν σκιές στην ελεύθερη έκδοση, απαιτείται έκδοση του Unity που έχει κόστος αγοράς. Γι'αυτό στην τελική έκδοση της εφαρμογής, δεν υπάρχουν σκιές.



Εικόνα Σ- 2. Cast Shadows

Σαν τελικό συμπέρασμα όμως, δημιουργήθηκε μια πλήρως λειτουργική εφαρμογή αναπαράστασης της Ιονίου Ακαδημίας, έστω και αν λείπουν κάποια χαρακτηριστικά ή αν κάποια από αυτά είναι λάθος. Σαν παράδειγμα, το σαλόνι στην είσοδο σίγουρα δεν είναι όπως αναπαρίσταται, κάποια γραφεία στο ισόγειο δεν υπάρχουν, τα παράθυρα δεν έχουν κουρτίνες, και άλλα.

Τόσο το Blender όσο και το Unity έχουν μια πλατιά βάση χρηστών που είναι διατεθειμένοι να βοηθήσουν, να απαντάν σε απορίες και προβλήματα που προκύπτουν. Από μόνο του, αυτό είναι αρκετό για να τα κάνει δημοφιλή εργαλεία στο είδος τους.

Επειδή η λειτουργικότητά τους είναι διαφορετική, χρησιμοποιούν διαφορετικές τεχνικές απόδοσης των γραφικών. Το μεν Blender, ως 3D πρόγραμμα χρησιμοποιεί την τεχνική του Raytracing για την επίτευξη φωτορεαλιστικών αποτελεσμάτων. Το δε Unity χρησιμοποιεί την τεχνική του Rasterization για την επίτευξη χρόνων πραγματικής απόδοσης (Realtime Rendering).

Προτάσεις για επιπλέον Ανάπτυξη

Αυτού του είδους οι δημιουργίες είναι πάντα ημιτελείς. Είναι όπως ένα σπίτι ή, επειδή μου αρέσει η ιστιοπλοΐα, όπως ένα ιστιοπλοϊκό σκάφος. Ποτέ δεν τελειώνουν οι δουλειές, υπάρχει πάντα κάτι που δεν έχει γίνει.

Υπό αυτήν την έννοια, το αποτέλεσμα δεν είναι οριστικό, επιδέχεται βελτιώσεις, αλλαγές, προσθήκες, αναθεωρήσεις. Έχω χάσει το μέτρημα στον αριθμό των αλλαγών που έχω κάνει, αν είχα εφαρμόσει versioning θα έμενα ο ίδιος έκπληκτος. Έχω χάσει το μέτρημα στις φορές που έβαζα στην άκρη την δουλειά που είχα κάνει και ξεκίναγα πάλι από την αρχή. Δημιουργίες αυτού του είδους μπορεί να βασίζονται σε επιστημονικές αρχές, αλλά δεν είναι επιστήμη. Είναι τέχνη. Που θέλει υπομονή και επιμονή για την εκμάθησή της.

Σε ότι αφορά την διάδραση, θα μπορούσε να είναι πιο πλούσια, να συμβαίνει για πολύ περισσότερα αντικείμενα. Η ποιότητα των materials μπορεί να βελτιωθεί. Τα αντικείμενα θα μπορούσαν να έχουν περισσότερη λεπτομέρεια, αλλά εδώ πρέπει να ληφθεί υπόψη ότι οι υψηλές απαιτήσεις σε υλικό περιορίζουν το κοινό στο οποίο απευθύνεται η εφαρμογή.

Ο φωτισμός θα μπορούσε να είναι καλύτερος. Χρησιμοποιήθηκαν βέβαια φώτα σε σημεία ενδιαφέροντος, για παράδειγμα η προτομή του Καποδίστρια φωτίζεται από έναν προβολέα, αλλά θα μπορούσαν να υπάρχουν και σε άλλα σημεία. Να αναφερθεί εδώ ότι ο φωτισμός είναι από μόνος του ολόκληρο κεφάλαιο και υπάρχει βιβλιογραφία αποκλειστικά αφιερωμένη σε αυτόν (Birn 2013).

Η σκηνή της Ιονίου Ακαδημίας μπορεί να γίνει πιο ζωντανή με ανθρώπους να κάθονται στα καθίσματα (όχι όλα όμως), και με μουσικούς στην σκηνή. Στο πάνω διάζωμα θα μπορούσε να τοποθετηθεί μια κινηματογραφική μηχανή προβολής.

Υπήρχε η σκέψη να δημιουργηθεί ένα εισαγωγικό animation βίντεο με τα Γερμανικά Βομβαρδιστικά να έρχονται και να βομβαρδίζουν το κτίριο το οποίο φλεγόταν στην συνέχεια. Εγκαταλείφθηκε τελικά, η προσπάθεια και ο χρόνος που απαιτείται για κάτι τέτοιο ήταν απαγορευτικά.

Πολλά ακόμα θα μπορούσαν να γίνουν, είμαι ανοιχτός σε ιδέες για το μέλλον.

ΒΙΒΛΙΟΓΡΑΦΙΑ - ΙΣΤΟΓΡΑΦΙΑ

"Yes, but how many polygons?" An artist blog entry with interesting numbers. 1 Σεπτέμβρης 2007. https://forum.beyond3d.com/threads/yes-but-how-many-polygons-an-artist-blog-entry-with-interesting-numbers.39321/ (πρόσβαση Σεπτέμβρης 14, 2018).

Agisoft Photoscan. 2018. http://www.agisoft.com/ (πρόσβαση Αύγουστος 7, 2018).

Birn, Jeremy. *Digital Lighting & Rendering Third Edition.* New Riders, 2013.

Blender (EEVOO). https://www.blender.org/2-8/ (πρόσβαση Αύγουστος 7, 2018).

Blender. https://www.blender.org/ (πρόσβαση Αύγουστος 7, 2018).

Blender. *Blender HotKeys In-depth Reference.* http://download.blender.org/documentation/BlenderHotkeyReference.pdf (πρόσβαση Οκτώβρης 2, 2018).

Brackeys. *Content Scrolling - Unity 4.6 Tutorial.* 5 Οκτώβρης 2014. https://www.youtube.com/watch?v=MWOvwegLDl0 (πρόσβαση Ιούνιος 23, 2018).

—. *PAUSE MENU in Unity.* 20 Δεκέμβρης 2017. https://www.youtube.com/watch?v=JivuXdrIHK0 (πρόσβαση Ιούνιος 15, 2018).

—. *START MENU in Unity.* 29 Νοέμβρης 2017. https://www.youtube.com/watch?v=zc8ac_qUXQY (πρόσβαση Ιούνιος 15, 2018).

Bui, Tony. *Creating scrollable dynamic text in Unity 5.* 2 Μάρτιος 2016. https://www.youtube.com/watch?v=JIzMUQBIDgM (πρόσβαση Ιούνιος 23, 2018).

Colliders . https://unity3d.com/learn/tutorials/topics/physics/colliders (πρόσβαση Σεπτέμβρης 15, 2018).

Creighton, Ryan Henson. *Unity 3D Game Development by Example Beginner's Guide.* Birmingham - Mumbai: Packt Publishing, 2010.

Fidelum_Games. *Unity 5 Tutorial Tuesdays: Doors, Drawers and Treasure Chests Part 1.* 11 Νοέμβρης 2016. https://www.youtube.com/watch?v=a5WXiMN3APk (πρόσβαση Ιούνιος 2, 2018).

—. *Unity 5 Tutorial Tuesdays: Doors, Drawers and Treasure Chests Part 2 - Locking and Unlocking.* 10 Ιούλιος 2017. https://www.youtube.com/watch?v=xMEaRF4u_D0 (πρόσβαση Ιούνιος 3, 2018).

GIMP - GNU Image Manipuplation Program. 2018. https://www.gimp.org/ (πρόσβαση Αύγουστος 9, 2018).

Henderson, G. P. *Η Ιόνιος Ακαδημία (μτφ. Φ.Κ. Βώρου).* Κέρκυρα: Κέντρο Έρευνας και Διεθνούς Επικοινωνίας, 1980.

Iraci, Bernardo. *Blender Cycles: Lighting and Rendering Cookbook.* Birmingham - Mumbai: Packt Publishing, 2013.

iTween. http://www.pixelplacement.com/itween/index.php (πρόσβαση Μάιος 24, 2018).

Litster, Colin. *Blender 2.5 Materials and Textures.* Birmingham - Mumbai: Packt Publishing, 2011.

Martinez, Mariano. *Normal Mapping explained.* 11 Μάρτης 2010. https://www.youtube.com/watch?v=SQrHkKnSBcA (πρόσβαση Σεπτέμβρης 7, 2018).

McDermott, Wes. *Creating 3D Game Art for the iPhone with Unity Featuring modo and Blender pipelines.* Kidlington, Oxford: Focal Press - Elsevier, 2010.

Municipal Theatre of Corfu. 8 Φεβρουάριος 2018. https://en.wikipedia.org/wiki/Municipal_Theatre_of_Corfu (πρόσβαση Σεπτέμβρης 6, 2018).

Price, Andrew. *Blender Cheat Sheet v2.* 13 Οκτώβρης 2016. https://www.blenderguru.com/articles/free-blender-keyboard-shortcut-pdf (πρόσβαση Οκτώβρης 2, 2018).

Simon, Jackson. *Unity 3D UI Essentials.* Birminghan - Mumbai: Packt Publishing, 2015.

Steinmetz, Frederick, και Gottfried Hofmann. *The Cycles Encyclopedia.* -: Blender Foundation Certified Trainers - Self Published, 2016.

ThunderWire_Games. *[Unity5] [Tutorial #19] Elevator - Lift.* 8 Απρίλιος 2016. https://www.youtube.com/watch?v=nCTSfwxaPw0 (πρόσβαση Ιούλιος 2, 2018).

Unity. 2018. https://unity3d.com/ (πρόσβαση Αύγουστος 7, 2018).

Unity_Technologies. *WebGL Browser Compatibility.* 14 Σεπτέμβρης 2018. https://docs.unity3d.com/Manual/webgl-browsercompatibility.html (πρόσβαση Οκτώβρης 9, 2018).

Valenza, Enrico. *Blender 2.6 Cycles Materials and Textures Cookbook - Third Edition.* Birgingham - Mumbai: Packt Publishing, 2016.

Wikipedia. *Unity Technologies.* 1 Οκτώβρης 2018. https://en.wikipedia.org/wiki/Unity_Technologies (πρόσβαση Οκτώβρης 2, 2018).

—. *WebGL.* 18 Αύγουστος 2018. https://en.wikipedia.org/wiki/WebGL (πρόσβαση Σεπτέμβρης 2, 2018).

Zotov, Alexander. *How to make scrollable UI (user interface) text area in Unity game? Quick tutorial.* 24 Αύγουστος 2017. https://www.youtube.com/watch?v=ArH0S2Cdptk (πρόσβαση Ιούνιος 24, 2018).

Zucconi, Alan, και Lammers Kenneth. *Unity 5.x Shaders and Effects Cookbook.* Birmingham: Packt Publishing, 2016.

Αγγελομάτη, Ελένη. *Ιστορικό Πανεπιστημίου.* https://ionio.gr/gr/prospective/university-history/ (πρόσβαση Αύγουστος 24, 2018).

Αγγελομάτη-Τσουγκαράκη, Ελένη. *Η Ιόνιος Ακαδημία, Το χρονικό της Ίδρυσης του πρώτου Ελληνικού Πανεπιστημίου (1811-1824).* Αθήνα: Ν. Ρωμηός, Αρχειακές Μελέτες, 1997.

Αγιούς, Ι. Αντώνιος. *Οδοιπορικό στην Παλιά Κέρκυρα.* Κέρκυρα: Εκδόσεις Σταμούλη Α.Ε., 2004.

—. *Τα Κερκυραϊκά (Ιστορικά Ανάλεκτα).* Αθήνα: Σταμούλης Α.Ε., 2005.

Αλειφαντής Στέλιος, Ζαμπούρας Σέργιος. *Ο Ιωάννης Καποδίστριας και η Εθνεγερσία του Μάρτη 1821: Τρεις Επιστολές στον Διονύσιο Ρώμα – Στέλιου Αλειφαντή & Σέργιου Ζαμπούρα.* 19 Απρίλιος 2016. https://argolikivivliothiki.gr/2016/04/19/kapodistrias-5/ (πρόσβαση Αύγουστος 24, 2018).

Βραϊλας Αρμένης. http://www.grandlodge.gr/vrailas-armenis-w-74706.html (πρόσβαση Σεπτέμβρης 6, 2018).

Γεωργαλά-Πριόβολου, Στέλλα. *Η Λατινική στην Ιόνιο Ακαδημία, Ανέκδοτες ΜΑρτυρίες απο το Αρχείο Γκίλφορντ, Διδακτικά Εγχειρίδια.* Αθήνα: Περιοδικό "Παρουσία" - Παράρτημα Αρ. 8, 1989.

Ετικέτα: Ιερή συμμαχία... http://www.mixanitouxronou.gr/tag/ieri-simmachia/ (πρόσβαση Αύγουστος 31, 2018).

Η ιστορία της Κέρκυρας από την αρχαιότητα έως σήμερα. 2119 6 2018. https://atcorfu.com/el/corfu-history/ (πρόσβαση 8 21, 2018).

Ηρόδοτος. *Ιστορίαι.* -: -, 450 π.χ.

Θεοτόκης, Μ. Σπυρίδων. *Περί της Εκαπιδεύσεως εν Επτανήσω.* Κέρκυρα: Κερκυραϊκά Χρονιά, Τόμος 2ος, 1956.

Ιερά Συμμαχία. 7 Ιούλιος 2018. https://el.wikipedia.org/wiki/Ιερά_Συμμαχία (πρόσβαση Αύγουστος 30, 2018).

Ιστορία της Κέρκυρας. 10 Ιούνιος 2018. https://atcorfu.com/el/corfu-history/ (πρόσβαση Αύγουατος 24, 2018).

Κονόμος, Ντίνος. *Η Ιόνιος Ακαδημία.* Αθήνα: Κερκυραϊκή Ένωση Αθηνών, 1965.

Κουρκουμέλης, Νίκος Κ. *Η εκπαίδευση στην Κέρκυρα κατά τη διάρκεια της Βρετανικής Προστασίας (1816-1864).* Αθήνα: Σύλλογος προς Διάδοσιν των Ελληνικών Γραμμάτων, 2002.

—. *Η Εκπαίδευση στην Κέρκυρα κατά τη διάρκεια της βρετανικής προστασίας (1816-1864).* Αθήνα: Σύλλογος προς Διάδοσιν των Ελληνικών Γραμμάτων, 2002.

Ναπολεόντειοι Πόλεμοι. 2017. https://el.m.wikipedia.org/wiki/Ναπολεόντειοι_Πόλεμοι (πρόσβαση Αύγουστος 24, 2018).

Όμηρος. *Οδύσσεια.*

Ροδακανάκης, Εμμανουήλ. *Ο Βοναπάρτης και αι Ιόνιοι Νήσαι : επεισόδιον των κατακτήσεων της Δημοκρατίας και πρώτης Αυτοκρατορίας (1797-1816) / Εμμ. Ροδοκανάκη ; μετάφρασις Νοεμής Α. Ζωηρού Πασσά.* Κέρκυρα: Εταιρεία προς Ενίσχυσιν Επτανησιακών Μελετών, 1937.

Σαλβάνος, Γεράσιμος Ι., και Βάσω Γ. Σαλβάνου. *Η Ιόνιος ακαδημία, Ο Ιδρυτη΄ς αυτής Κόμης Γκίλφορδ, Οι καθηγηταί και σπουδασταί αυτής.* Αθήμα: Ανάτυπον εκ του Πανηγυρικού Τεύχους της Ελληνικής Δημιουργίας, 1949.

Στρατιωτικά κτίρια – στρατώνες κατά την Βενετοκρατία. https://www.corfuhistory.eu/?p=974 (πρόσβαση 8 21, 2018).

Συνθήκη του Τιλσίτ. 3 Μάϊος 2017. https://el.wikipedia.org/wiki/Συνθήκη_του_Τιλσίτ (πρόσβαση Αύγουστος 28, 2018).

Συνθήκη των Παρισίων (1814). 2 Δεκέμβρης 2015. https://el.wikipedia.org/wiki/Συνθήκη_των_Παρισίων_(1814) (πρόσβαση Αύγουστος 28, 2018).

Σύντομη ιστορία της Κέρκυρας. https://www.corfuhistory.eu/?p=51 (πρόσβαση Αύγουστος 21, 2018).

Τρωϊκός Πόλεμος. 16 7 2018. https://el.wikipedia.org/wiki/Τρωικός_Πόλεμος (πρόσβαση 8 21, 2018).

Φρέντερικ Νορθ (1766-1827). 11 Αύγουστος 2018. https://el.m.wikipedia.org/wiki/Φρέντερικ_Νορθ_(1766-1827) (πρόσβαση Αύγουστος 31, 2018).

Χριστόφορος Φιλητάς. 10 Μάϊος 2017. https://el.wikipedia.org/wiki/Χριστόφορος_Φιλητάς (πρόσβαση Σεπτέμβρης 4, 2018).

ΠΑΡΑΡΤΗΜΑ

Στο DVD υπάρχει όλο το υλικό της έρευνας. Περιέχονται τα σχέδια της Ιονίου Ακαδημίας, τα .blend αρχεία, όλος ο φάκελος του Unity Project που περιέχει και το scripts.

Εξωτερικές πηγές

Για τα script Interactor.cs και InteractiveObject.cs χρησιμοποιήθηκε η προσέγγιση του Fidelum Games (Fidelum_Games 2016) και (Fidelum_Games, Unity 5 Tutorial Tuesdays: Doors, Drawers and Treasure Chests Part 2 - Locking and Unlocking 2017). Το script ReticleController.cs έχει τροποποιηθεί και προσαρμοστεί κατάλληλα για αυτή την προσέγγιση.

Για το ασανσέρ, Χρησιμοποιήθηκε η προσέγγιση του ThunderWire Games (ThunderWire_Games 2016) που τροποποιήθηκε σε μεγάλο βαθμό.

Για την δημιουργία των μενού και του παραθύρου διαλόγου χρησιμοποιήθηκαν διάφορες πηγές και ιδέες (Brackeys, START MENU in Unity 2017), (Brackeys, Content Scrolling - Unity 4.6 Tutorial 2014), (Bui 2016).

Για τα αντικείμενα που δεν δημιουργήθηκαν από μένα, πηγή ήταν το blendswap.

Το σαλόνι, ο καναπές, οι καρέκλες και τα τραπέζια στον προθάλαμο είναι δημιουργία του Hans Erickso.

Η βιβλιοθήκη στις σκάλες και στην είσοδο του αίθουσας εκδηλώσεων είναι δημιουργία του Doug C. Τα έπιπλα γραφείου προέρχονται από τον Saadsart64, η βιβλιοθήκη που δεν έχει βιβλία, οι νιπτήρες και οι τουαλέτες προέρχονται από το Kenney Furniture Pack.

Ο πυροσβεστήρας δημιουργήθηκε από τον DennisH2010.

Η βιβλιοθήκη στο γραφείο του Πρύτανη είναι από το Scopia Visual Interfaces Systems, το γραφείο από τον pcraven και η καρέκλα από τον kolban.

WebGL – Ρυθμίσεις Browser

Για τον Firefox δεν απαιτείται κάποια ιδιαίτερη ενέργεια, σαν προεπιλογή το WebGL είναι ενεργοποιημένο.

Για το ενδεχόμενο να είναι ενεργοποιημένο, κάνουμε τις εξής ενέργειες:

- Στην μπάρα διευθύνσεων δίνουμε about:config
- Η τιμή webgl.force-enabled πρέπει να είναι true
- Η τιμή layers.acceleration.force-enabled πρέπει να είναι true
- Αν το λειτουργικό σύστημα είναι Windows 7, η τιμή gfx.direct2d.force-enabled πρέπει να είναι true
- Επαννεκινούμε τον Firefox.

Για τον Chrome οι ενέργειες που πρέπει να γίνουν είναι οι εξής:

- Στην μπάρα διευθύνσεων δίνουμε chrome://settings
- Δίνουμε Show advanced settings (Ρυθμίσεις για Προχωρημένους)
- Στο System (Ρυθμίσεις Συστήματος), η επιλογή Use hardware acceleration when available πρέπει να είναι τσεκαρισμένη
- Στην μπάρα διευθύνσεων δίνουμε chrome://flags
- Βεβαιωνόμαστε ότι η ρύθμιση Disable WebGL δεν είναι ενεργή
- Σε νεότερες εκδόσεις του Chrome η επιλογή δεν είναι διαθέσιμη, αναζητούμε την τιμή WebGL 2.0 και αλλάζουμε την τιμή του από Default σε Enabled.
- Αν αυτά δεν αρκούν, ενεργοποιούμε και το Override software rendering list
- Επανεκκινούμε τον Chrome
- Αν η εφαρμογή δεν βρίσκεται σε κάποιον WebServer δεν θα λειτουργήσει για λόγους ασφαλείας. Για να λειτουργήσει τοπικά θα πρέπει να τρέξει ο Chrome με την παράμετρο --allow-file-access-from-files. Θα πρέπει δηλαδή να εκτελεστεί η εντολή: **chrome.exe --allow-file-access-from-files**.

Κώδικας

ChangeVolume.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
004 using UnityEngine.UI;
005 
006 public class ChangeVolume : MonoBehaviour {
007     public AudioSource aSource;
008     public Slider volumeSlider;
009 
010     public void Awake()
011     {
012 
013         if (!aSource.isPlaying) aSource.Play(0);
014     }
015 
016     // Update is called once per frame
017     public void Update ()
018     {
019         if (!aSource.isPlaying) aSource.Play(0);
020         AudioListener.volume = volumeSlider.value;
021     }
022 
023     public void StopSample()
024     {
025         aSource.Stop();
026     }
027 
028 }
```

Content.cs

```
001 using UnityEngine;
002 using System.Collections;
003 using UnityEngine.UI;
004 using System.IO;
005 using UnityStandardAssets.Characters.FirstPerson;
006 
007 
008 public class Content : MonoBehaviour
009 {
010     public Text content, label;
011     private string title, txt, txtFile;
012     private GameObject infoPanel;
013 
014     private ReticleController reticleController;
015     // Use this for initialization
016     private void Start()
017     {
018 
019         reticleController = GameObject.FindObjectOfType<ReticleController>();
020 
021 
022     }
023     public void ShowTextBox(GameObject gameObject)
024     {
025 
026         reticleController.infoPanel.SetActive(true);
027 
028         if (gameObject.name == "bust")
029         {
030             title = "Ιωάννης Καποδίστριας";
```

```
031             txtFile = "kapodistrias";
032             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
033             txt = txtAssets.text;
034             content.text = txt;
035             label.text = title;
036         }
037         if (gameObject.name == "painting")
038         {
039             title = "Κάποιος Ζωγράφος";
040             txtFile = "painting";
041             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
042             txt = txtAssets.text;
043             content.text = txt;
044             label.text = title;
045         }
046         if (gameObject.name == "ionioan-academy-sign")
047         {
048             title = "Ιόνιος Ακαδημία";
049             txtFile = "ionian_academy";
050             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
051             txt = txtAssets.text;
052             content.text = txt;
053             label.text = title;
054         }
055         if (gameObject.name == "frederick-north-sign")
056         {
057             title = "Φρειδερίκος Νορθ";
058             txtFile = "frederick_north";
059             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
060             txt = txtAssets.text;
061             content.text = txt;
062             label.text = title;
063         }
064         if (gameObject.name == "grimani-sign")
065         {
066             title = "Γκριμάνι";
067             txtFile = "grimani";
068             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
069             txt = txtAssets.text;
070             content.text = txt;
071             label.text = title;
072         }
073         if (gameObject.name == "logo")
074         {
075             title = "Ιόνιο Πανεπιστήμιο";
076             txtFile = "ionian_university";
077             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
078             txt = txtAssets.text;
079             content.text = txt;
080             label.text = title;
081         }
082         if (gameObject.name == "stage")
083         {
084             title = "Η Σκηνή της Αίθουσας Εκδηλώσεων";
085             txtFile = "stage";
086             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
087             txt = txtAssets.text;
088             content.text = txt;
089             label.text = title;
090         }
091         if (gameObject.name == "ConferenceTable")
092         {
093             title = "Αίθουσα Συσκέψεων";
094             txtFile = "conference";
095             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
096             txt = txtAssets.text;
```

```
097             content.text = txt;
098             label.text = title;
099         }
100         if (gameObject.name == "ConferenceTable2")
101         {
102             title = "Αίθουσα Συσκέψεων";
103             txtFile = "conference1";
104             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
105             txt = txtAssets.text;
106             content.text = txt;
107             label.text = title;
108         }
109         if (gameObject.name == "desk_v2")
110         {
111             title = "Γραφείο Πρύτανη";
112             txtFile = "prytanis";
113             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
114             txt = txtAssets.text;
115             content.text = txt;
116             label.text = title;
117         }
118         if (gameObject.name == "desk_v2 (1)")
119         {
120             title = "Γραφείο Αντιπρύτανη";
121             txtFile = "antiprytanis";
122             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
123             txt = txtAssets.text;
124             content.text = txt;
125             label.text = title;
126         }
127         if (gameObject.name == "desk_v2 (2)")
128         {
129             title = "Ludwig van Bethoven";
130             txtFile = "beethoven";
131             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
132             txt = txtAssets.text;
133             content.text = txt;
134             label.text = title;
135         }
136         if (gameObject.name == "lion")
137         {
138             title = "Ο Λέων του Αγίου Μάρκου";
139             txtFile = "lion";
140             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
141             txt = txtAssets.text;
142             content.text = txt;
143             label.text = title;
144         }
145         if (gameObject.name == "fire extinguisher")
146         {
147             title = "Πυροσβεστήρας";
148             txtFile = "fire_extinguisher";
149             TextAsset txtAssets = (TextAsset)Resources.Load(txtFile);
150             txt = txtAssets.text;
151             content.text = txt;
152             label.text = title;
153         }
154 
155 
GameObject.Find("FPSController").GetComponent<FirstPersonController>().enabled =
false;
156         Cursor.lockState = CursorLockMode.None;
157         Cursor.visible = true;
158     }
159     private void Update()
160     {   if (reticleController.infoPanel)
```

```
161         { if (reticleController.infoPanel.activeSelf == true)
reticleController.reticleActive = false; else reticleController.reticleActive =
true;
162         if (Input.GetKeyDown(KeyCode.X))
163         {
164             reticleController.infoPanel.SetActive(false);
165
GameObject.Find("FPSController").GetComponent<FirstPersonController>().enabled =
true;
166             Cursor.lockState = CursorLockMode.Locked;
167             Cursor.visible = false;
168             }
169             if (!reticleController.reticleActive)
reticleController.ShowIcon("None");
170         }
171     }
172 }
```

DialogBox.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
004 using UnityStandardAssets.Characters.FirstPerson;
005
006
007 public class DialogBox : MonoBehaviour {
008     private ReticleController reticleController;
009     private void Start()
010     {
011         reticleController = GameObject.FindObjectOfType<ReticleController>();
012     }
013
014     public void CloseDialogBox()
015     {
016         reticleController.infoPanel.SetActive(false);
017
GameObject.Find("FPSController").GetComponent<FirstPersonController>().enabled =
true;
018         Cursor.lockState = CursorLockMode.Locked;
019         Cursor.visible = false;
020         reticleController.ShowIcon("ReticleIcon");
021     }
022
023 }
```

Elevator.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
004
005 public class Elevator : MonoBehaviour {
006     //for the display screen
007    public LevelDisplay levelDisplay;
008
009    private ElevatorController elevatorController;
010     private Animator internalDoorAnimation;
011
012     public Transform[] elevatorButtons = new Transform[5];
013
014     // Use this for initialization
```

```
015     void Start () {
016          levelDisplay = GameObject.Find ("elevator-
where").transform.GetComponent<LevelDisplay> ();
017         elevatorController = this.transform.GetComponent<ElevatorController>();
018          elevatorButtons[0] = GameObject.FindGameObjectWithTag ("Level-
1").transform;
019          elevatorButtons[1] = GameObject.FindGameObjectWithTag
("Level0").transform;
020          elevatorButtons[2] = GameObject.FindGameObjectWithTag
("Level1").transform;
021          elevatorButtons[3] = GameObject.FindGameObjectWithTag
("Level2").transform;
022          elevatorButtons[4] = GameObject.FindGameObjectWithTag
("Level3").transform;
023         internalDoorAnimation = this.transform.GetComponent<Animator>();
024
025     }
026
027    // Update is called once per frame
028    void Update () {
029
030    }
031     public void ButtonPressed(GameObject button)
032     {
033         if (button.name == "-1")
034         {
035             elevatorController.ElevatorGO(button.tag);
036             elevatorController.floorNumber = 0;
037         }
038         if (button.name == "0")
039         {
040             elevatorController.ElevatorGO(button.tag);
041             elevatorController.floorNumber = 1;
042         }
043         if (button.name == "1")
044         {
045             elevatorController.ElevatorGO(button.tag);
046             elevatorController.floorNumber = 2;
047         }
048         if (button.name == "2")
049         {
050             elevatorController.ElevatorGO(button.tag);
051             elevatorController.floorNumber = 3;
052         }
053         if (button.name == "3")
054         {
055             elevatorController.ElevatorGO(button.tag);
056             elevatorController.floorNumber = 4;
057         }
058     }
059 }
```

ElevatorController.cs

```
001 /*
002 ElevatorController.cs - wirted by ThunderWire Games * Script for Simple
Elevator Moving
003 ---- Modified by Athanasios Gkoritsas ----
004 */
005
006 using UnityEngine;
007 using System.Collections;
008
009 [System.Serializable]
010 public class m_ElevatorSounds
```

```
011 {
012     public AudioClip ElevatorStartMoving;
013     public AudioClip ElevatorStop;
014     public AudioClip ElevatorDing;
015     public AudioClip ElevatorDoorsOpen;
016     public AudioClip ElevatorDoorsClose;
017 }
018
019 [RequireComponent(typeof(AudioSource))]
020 public class ElevatorController : MonoBehaviour
021 {
022     public GameObject MainChar;
023     [Tooltip("Elevator move speed.")]
024     public float elevatorSpeed = 0.04f;
025     [Tooltip("Elevator floors.")]
026     public Transform[] elevatorFloors = new Transform[5];
027     public m_ElevatorSounds ElevatorSounds = new m_ElevatorSounds();
028     [Tooltip("Set elevator position to floor.")]
029     public int floorNumber = 0;
030     private string elevatorDirection;
031     [HideInInspector]
032     public bool ElevatorMoving;
033     private bool ElevatorMax;
034     private bool ElevatorMin;
035     private bool soundplayed;
036     private bool m_soundplayed;
037     private bool isMoved = false;
038    private Animator internalDoorAnimation, externalDoorAnimation;
039     private LevelDisplay levelDisplay;
040     private ExternalLevelDisplay externalLevelDisplayM1, externalLevelDisplay0,
externalLevelDisplay1, externalLevelDisplay2, externalLevelDisplay3;
041
042     void Start()
043     {
044         this.GetComponent<AudioSource>().Stop();
045          internalDoorAnimation = this.gameObject.GetComponent<Animator>();
046         externalDoorAnimation = GameObject.Find("Elevator_door-
basement").GetComponent<Animator>();
047         levelDisplay = GameObject.Find("elevator-
where").transform.GetComponent<LevelDisplay>();
048         externalLevelDisplayM1 = GameObject.Find("Elevator-basement-external-
display").transform.GetComponent<ExternalLevelDisplay>();
049         externalLevelDisplay0 = GameObject.Find("Elevator-0th-external-
display").transform.GetComponent<ExternalLevelDisplay>();
050         externalLevelDisplay1 = GameObject.Find("Elevator-1st-external-
display").transform.GetComponent<ExternalLevelDisplay>();
051         externalLevelDisplay2 = GameObject.Find("Elevator-2nd-external-
display").transform.GetComponent<ExternalLevelDisplay>();
052         externalLevelDisplay3 = GameObject.Find("Elevator-3rd-external-
display").transform.GetComponent<ExternalLevelDisplay>();
053
054         MainChar = GameObject.FindGameObjectWithTag("Player");
055         internalDoorAnimation.SetBool("OpenDoors", false);
056         externalDoorAnimation.SetBool("OpenDoors", false);
057     }
058
059     public void ElevatorGO(string goToLevel)
060     {
061         if (goToLevel == "Level-1") floorNumber = 0;
062         if (goToLevel == "Level0") floorNumber = 1;
063         if (goToLevel == "Level1") floorNumber = 2;
064         if (goToLevel == "Level2") floorNumber = 3;
065         if (goToLevel == "Level3") floorNumber = 4;
066         if (Mathf.Round(this.gameObject.transform.position.y) <
Mathf.Round(elevatorFloors[floorNumber].position.y))
067         {
```

```
068             if (!internalDoorAnimation.IsInTransition(0))  elevatorDirection =
"ElevatorUp";
069         }
070         else if (Mathf.Round(this.gameObject.transform.position.y) >
Mathf.Round(elevatorFloors[floorNumber].position.y))
071         {
072             if (!internalDoorAnimation.IsInTransition(0))  elevatorDirection =
"ElevatorDown";
073         }
074         else if (Mathf.Round(this.gameObject.transform.position.y) ==
Mathf.Round(elevatorFloors[floorNumber].position.y))
075         {
076             StartCoroutine(OpenAndCloseDoorsWithDelay());
077 
078         }
079         soundplayed = false;
080         m_soundplayed = false;
081     }
082 
083     void FloorNumber()
084     {
085         floorNumber = Mathf.Clamp(floorNumber, 0, elevatorFloors.Length - 1);
086 
087         if (elevatorDirection == "ElevatorUp" && !ElevatorMoving &&
!ElevatorMax)
088         {
089             floorNumber += 1;
090         }
091         if (elevatorDirection == "ElevatorDown" && !ElevatorMoving &&
!ElevatorMin)
092         {
093             floorNumber -= 1;
094         }
095     }
096 
097     void FixedUpdate()
098     {
099         Transform ElevatorFloor = elevatorFloors[floorNumber];
100 
101         if (Mathf.Round(this.transform.position.y) ==
Mathf.Round(elevatorFloors[0].position.y))
102         {
103             levelDisplay.ChangeTexture(levelDisplay.Level_M1);
104 
externalLevelDisplayM1.ChangeTexture(externalLevelDisplayM1.Level_M1);
105 
externalLevelDisplay0.ChangeTexture(externalLevelDisplay0.Level_M1);
106 
externalLevelDisplay1.ChangeTexture(externalLevelDisplay1.Level_M1);
107 
externalLevelDisplay2.ChangeTexture(externalLevelDisplay2.Level_M1);
108 
externalLevelDisplay3.ChangeTexture(externalLevelDisplay3.Level_M1);
109         }
110         if (Mathf.Round(this.transform.position.y) ==
Mathf.Round(elevatorFloors[1].position.y))
111         {
112             levelDisplay.ChangeTexture(levelDisplay.Level_0);
113 
externalLevelDisplayM1.ChangeTexture(externalLevelDisplayM1.Level_0);
114             externalLevelDisplay0.ChangeTexture(externalLevelDisplay0.Level_0);
115             externalLevelDisplay1.ChangeTexture(externalLevelDisplay1.Level_0);
116             externalLevelDisplay2.ChangeTexture(externalLevelDisplay2.Level_0);
117             externalLevelDisplay3.ChangeTexture(externalLevelDisplay3.Level_0);
118 
119         }
```

```
120         if (Mathf.Round(this.transform.position.y) ==
Mathf.Round(elevatorFloors[2].position.y))
121         {
122             levelDisplay.ChangeTexture(levelDisplay.Level_1);
123
externalLevelDisplayM1.ChangeTexture(externalLevelDisplayM1.Level_1);
124             externalLevelDisplay0.ChangeTexture(externalLevelDisplay0.Level_1);
125             externalLevelDisplay1.ChangeTexture(externalLevelDisplay1.Level_1);
126             externalLevelDisplay2.ChangeTexture(externalLevelDisplay2.Level_1);
127             externalLevelDisplay3.ChangeTexture(externalLevelDisplay3.Level_1);
128         }
129         if (Mathf.Round(this.transform.position.y) ==
Mathf.Round(elevatorFloors[3].position.y))
130         {
131             levelDisplay.ChangeTexture(levelDisplay.Level_2);
132
externalLevelDisplayM1.ChangeTexture(externalLevelDisplayM1.Level_2);
133             externalLevelDisplay0.ChangeTexture(externalLevelDisplay0.Level_2);
134             externalLevelDisplay1.ChangeTexture(externalLevelDisplay1.Level_2);
135             externalLevelDisplay2.ChangeTexture(externalLevelDisplay2.Level_2);
136             externalLevelDisplay3.ChangeTexture(externalLevelDisplay3.Level_2);
137         }
138         if (Mathf.Round(this.transform.position.y) ==
Mathf.Round(elevatorFloors[4].position.y))
139         {
140             levelDisplay.ChangeTexture(levelDisplay.Level_3);
141
externalLevelDisplayM1.ChangeTexture(externalLevelDisplayM1.Level_3);
142             externalLevelDisplay0.ChangeTexture(externalLevelDisplay0.Level_3);
143             externalLevelDisplay1.ChangeTexture(externalLevelDisplay1.Level_3);
144             externalLevelDisplay2.ChangeTexture(externalLevelDisplay2.Level_3);
145             externalLevelDisplay3.ChangeTexture(externalLevelDisplay3.Level_3);
146         }
147         if (elevatorDirection == "ElevatorUp")
148         {
149             if (!internalDoorAnimation.IsInTransition(0))
150             {
151                 ElevatorMoving = true;
152                 transform.position = Vector3.MoveTowards(transform.position,
ElevatorFloor.position, elevatorSpeed);
153             } else ElevatorMoving = false;
154         }
155
156         if (elevatorDirection == "ElevatorDown")
157         {
158             if (!internalDoorAnimation.IsInTransition(0))
159             {
160                 ElevatorMoving = true;
161             transform.position = Vector3.MoveTowards(transform.position,
ElevatorFloor.position, elevatorSpeed);
162             } else ElevatorMoving = false;
163         }
164
165             if (this.transform.position == ElevatorFloor.position)
166         {
167             ElevatorMoving = false;
168         }
169         if (this.transform.position == elevatorFloors[0].position)
170         {
171             ElevatorMax = false;
172             ElevatorMin = true;
173         }
174
175         if (this.transform.position.y > elevatorFloors[0].position.y)
176         {
177             ElevatorMin = false;
```

```
178         }
179
180         int MaxFloors = elevatorFloors.Length - 1;
181         if (this.transform.position == elevatorFloors[MaxFloors].position)
182         {
183             ElevatorMax = true;
184             ElevatorMin = false;
185         }
186
187         if (this.transform.position.y < elevatorFloors[MaxFloors].position.y)
188         {
189             ElevatorMax = false;
190         }
191
192         if (ElevatorMoving && !isMoved)
193         {
194             if (!m_soundplayed)
195             {
196                 if (internalDoorAnimation.GetBool("OpenDoors"))
StartCoroutine(OpenAndCloseDoorsWithDelay());
197                 if (!internalDoorAnimation.IsInTransition(0))
198                 {
199                 this.GetComponent<AudioSource>().clip =
ElevatorSounds.ElevatorStartMoving;
200                         this.GetComponent<AudioSource>().Play();
201                 m_soundplayed = true;
202                 }
203             }
204             if (!internalDoorAnimation.IsInTransition(0)) isMoved = true;
205         }
206         if (this.transform.position == elevatorFloors[floorNumber].position &&
!ElevatorMoving && isMoved)
207         {
208             if (!soundplayed)
209             {
210                 this.GetComponent<AudioSource>().Stop();
211
212
this.GetComponent<AudioSource>().PlayOneShot(ElevatorSounds.ElevatorStop);
213                 soundplayed = true;
214                 StartCoroutine(DingWaiter());
215                 StartCoroutine(OpenAndCloseDoorsWithDelay());
216             }
217             isMoved = false;
218         }
219     }
220
221     void OnDrawGizmos()
222     {
223         for (int i = 0; i < elevatorFloors.Length; i++)
224         {
225             Gizmos.color = Color.red;
226             float x = elevatorFloors[i].position.x;
227             float y = elevatorFloors[i].position.y;
228             float z = elevatorFloors[i].position.z;
229                 Gizmos.DrawWireCube(new Vector3(x, y + 1.4f, z), new
Vector3(1.3f, 2.8f, 2.2f));
230         }
231     }
232    IEnumerator OpenAndCloseDoorsWithDelay()
233    {
234
235         if (!internalDoorAnimation.GetBool("OpenDoors"))
236         {
237             switch (floorNumber)
238             {
```

```
239                 case 0:
240                     externalDoorAnimation = GameObject.Find("Elevator_door-
basement").GetComponent<Animator>();
241 
242                     if (!externalDoorAnimation.GetBool("OpenDoors"))
externalDoorAnimation.SetBool("OpenDoors",
!externalDoorAnimation.GetBool("OpenDoors"));
243                     break;
244                 case 1:
245                     externalDoorAnimation = GameObject.Find("Elevator_door-
0th").GetComponent<Animator>();
246                     if (!externalDoorAnimation.GetBool("OpenDoors"))
externalDoorAnimation.SetBool("OpenDoors",
!externalDoorAnimation.GetBool("OpenDoors"));
247                     break;
248                 case 2:
249                     externalDoorAnimation = GameObject.Find("Elevator_door-
1st").GetComponent<Animator>();
250                     if (!externalDoorAnimation.GetBool("OpenDoors"))
externalDoorAnimation.SetBool("OpenDoors",
!externalDoorAnimation.GetBool("OpenDoors"));
251                     break;
252                 case 3:
253                     externalDoorAnimation = GameObject.Find("Elevator_door-
2nd").GetComponent<Animator>();
254                     if (!externalDoorAnimation.GetBool("OpenDoors"))
externalDoorAnimation.SetBool("OpenDoors",
!externalDoorAnimation.GetBool("OpenDoors"));
255                     break;
256                 case 4:
257                     externalDoorAnimation = GameObject.Find("Elevator_door-
3rd").GetComponent<Animator>();
258                     if (!externalDoorAnimation.GetBool("OpenDoors"))
externalDoorAnimation.SetBool("OpenDoors",
!externalDoorAnimation.GetBool("OpenDoors"));
259                     break;
260             }
261             yield return new WaitForSeconds(1.5f);
262             internalDoorAnimation.SetBool("OpenDoors",
!internalDoorAnimation.GetBool("OpenDoors"));
263             if (externalDoorAnimation.GetBool("OpenDoors"))
264             {
265                 yield return new WaitForSeconds(7);
266                 externalDoorAnimation.SetBool("OpenDoors", false);
267                 yield return new WaitForSeconds(1.5f);
268                 internalDoorAnimation.SetBool("OpenDoors", false);
269 
270             }
271         }
272         else
273         {
274             internalDoorAnimation.SetBool("OpenDoors",
!internalDoorAnimation.GetBool("OpenDoors"));
275         }
276     }
277 
278     IEnumerator DingWaiter()
279     {
280         yield return new WaitForSeconds(.5f);
281 
this.GetComponent<AudioSource>().PlayOneShot(ElevatorSounds.ElevatorDing);
282     }
283 
284     public void OpenDoorsSound()
285     {
```

```
286
this.GetComponent<AudioSource>().PlayOneShot(ElevatorSounds.ElevatorDoorsOpen);
287     }
288
289     public void CloseDoorsSound()
290     {
291
this.GetComponent<AudioSource>().PlayOneShot(ElevatorSounds.ElevatorDoorsClose);
292     }
293
294 }
```

ElevatorExternalDoors.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
004
005 public class ElevatorExternalDoors : MonoBehaviour {
006    private Elevator elevator;
007
008     // Use this for initialization
009     void Start() {
010         elevator = GameObject.Find("elevator").GetComponent<Elevator>();
011
012
013     }
014
015    // Update is called once per frame
016    void Update () {
017
018    }
019    public void ButtonPressed()
020    {
021          if (this.transform.name == "Elevator-button-basement")
022                 elevator.ButtonPressed (GameObject.Find ("-1"));
023           if (this.transform.name == "Elevator-button-0th")
024                 elevator.ButtonPressed (GameObject.Find ("0"));
025           if (this.transform.name == "Elevator-button-1st")
026                 elevator.ButtonPressed (GameObject.Find ("1"));
027           if (this.transform.name == "Elevator-button-2nd")
028                 elevator.ButtonPressed (GameObject.Find ("2"));
029           if (this.transform.name == "Elevator-button-3rd")
030                 elevator.ButtonPressed (GameObject.Find ("3"));
031    }
032 }
```

ExternalElevatorDoorsSound.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
004
005 [System.Serializable]
006 public class m_ExternalDoorElevatorSounds
007 {
008     public AudioClip ElevatorDoorsOpen;
009     public AudioClip ElevatorDoorsClose;
010 }
011
012     public class ExternalElevatorDoorsSound : MonoBehaviour {
013
```

```
014     public m_ExternalDoorElevatorSounds ExternalDoorsElevatorSounds = new
m_ExternalDoorElevatorSounds();
015     private Animator ExternalDoorsAnimation;
016     // Use this for initialization
017     void Start () {
018         this.GetComponent<AudioSource>().Stop();
019     }
020
021     // Update is called once per frame
022     public void OpenDoorsSound()
023     {
024
this.GetComponent<AudioSource>().PlayOneShot(ExternalDoorsElevatorSounds.ElevatorDo
orsOpen);
025     }
026
027     public void CloseDoorsSound()
028     {
029
this.GetComponent<AudioSource>().PlayOneShot(ExternalDoorsElevatorSounds.ElevatorDo
orsClose);
030     }
031 }
```

ExternalLevelDisplay.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
004
005 public class ExternalLevelDisplay : MonoBehaviour {
006
007     public Texture2D Level_M1;
008     public Texture2D Level_0;
009     public Texture2D Level_1;
010     public Texture2D Level_2;
011     public Texture2D Level_3;
012     public Renderer rend;
013     private bool isOn;
014
015     // Use this for initialization
016     void Start()
017     {
018         rend = GetComponent<Renderer>();
019     }
020
021     // Update is called once per frame
022     void Update()
023     {
024
025     }
026
027     public void ChangeTexture(Texture2D texture)
028     {
029         rend.material.mainTexture = texture;
030     }
031 }
```

InteractiveObject.cs

```
001 using UnityEngine;
002 using System.Collections;
```

```
003 using System.Text.RegularExpressions;
004
005 public class InteractiveObject : MonoBehaviour
006 {
007
008     [SerializeField] public Vector3 openPosition, closedPosition, openRotation,
closedRotation;
009
010
011     [SerializeField] private float animationTime;
012
013     [SerializeField] public bool isOpen = false;
014     [SerializeField] public bool resetWhenDone = false;
015     [SerializeField] public bool isShutter = false;
016
017     [SerializeField] private MovementType movementType;
018     private InteractiveObject interactiveObjectParent;
019     private GameObject parentObject;
020     private enum MovementType { Slide, Rotate };
021     private Hashtable iTweenArgs;
022     private Hashtable iTweenParentArgs;
023     private AudioSource aZONK;
024    private AudioSource[] aSource, aParentSource;
025     private GameObject ZONK, leftPanel, rightPanel;
026
027     void Start()
028     {
029         iTweenArgs = iTween.Hash();
030         iTweenArgs.Add("position", openPosition);
031         iTweenArgs.Add("rotation", openRotation);
032         iTweenArgs.Add("time", animationTime);
033         iTweenArgs.Add("islocal", true);
034         iTweenArgs.Add("delay", 0);
035
036         aSource = GetComponents<AudioSource>();
037
038         if (transform.parent.GetComponent<InteractiveObject>())
039         {
040             interactiveObjectParent =
transform.parent.GetComponent<InteractiveObject>();
041             iTweenParentArgs = iTween.Hash();
042             iTweenParentArgs.Add("position",
interactiveObjectParent.openPosition);
043             iTweenParentArgs.Add("rotation",
interactiveObjectParent.openRotation);
044             iTweenParentArgs.Add("time",
interactiveObjectParent.animationTime);
045             iTweenParentArgs.Add("islocal", true);
046             iTweenParentArgs.Add("delay", 0);
047
048             aParentSource = transform.parent.GetComponents<AudioSource>();
049         }
050     }
051
052     public void PerformAction()
053     {
054         AnimationController();
055
056     }
057     private void AnimateChild()
058     {
059         if (isOpen)
060         {
061             iTweenArgs["position"] = closedPosition;
062             iTweenArgs["rotation"] = closedRotation;
063             iTweenArgs["onstart"] = "ChildAnimationStarted";
```

```
064             iTweenArgs["oncomplete"] = "ChildAnimationDone";
065         }
066         else
067         {
068             iTweenArgs["position"] = openPosition;
069             iTweenArgs["rotation"] = openRotation;
070             iTweenArgs["onstart"] = "ChildAnimationStarted";
071             iTweenArgs["oncomplete"] = "ChildAnimationDone";
072         }
073 
074         isOpen = !isOpen;
075 
076         switch (movementType)
077         {
078             case MovementType.Slide:
079                 iTween.MoveTo(gameObject, iTweenArgs);
080                 break;
081             case MovementType.Rotate:
082                 iTween.RotateTo(gameObject, iTweenArgs);
083                 break;
084         }
085     }
086 
087     private void AnimateParent()
088     {
089         //again, check the audio
090 
091         if (interactiveObjectParent.isOpen)
092         {
093             iTweenParentArgs["islocal"] = true;
094             iTweenParentArgs["position"] =
interactiveObjectParent.closedPosition;
095             iTweenParentArgs["rotation"] =
interactiveObjectParent.closedRotation;
096             iTweenParentArgs["oncomplete"] = "ParentAnimationDone";
097             iTweenParentArgs["onstart"] = "ParentAnimationStarted";
098         }
099         else
100         {
101             iTweenParentArgs["islocal"] = true;
102             iTweenParentArgs["position"] =
interactiveObjectParent.openPosition;
103             iTweenParentArgs["rotation"] =
interactiveObjectParent.openRotation;
104             iTweenParentArgs["oncomplete"] = "ParentAnimationDone";
105             iTweenParentArgs["onstart"] = "ParentAnimationStarted";
106         }
107 
108         interactiveObjectParent.isOpen = !interactiveObjectParent.isOpen;
109 
110         switch (movementType)
111         {
112             case MovementType.Slide:
113                 iTween.MoveTo(gameObject.transform.parent.gameObject,
iTweenParentArgs);
114                 break;
115             case MovementType.Rotate:
116                 iTween.RotateTo(gameObject.transform.parent.gameObject,
iTweenParentArgs);
117                 break;
118         }
119     }
120 
121 
122     private void ChildAnimationStarted()
123     {
```

```
124         if (!isOpen) aSource[0].Play();
125         else if (aSource.Length == 2) aSource[1].Play();
126 
127 
128     }
129 
130     private void ChildAnimationDone()
131 
132     {
133         if (resetWhenDone && isOpen)
134         {
135             //maybe tweak the return time?
136             if (interactiveObjectParent) iTweenArgs["delay"] =
interactiveObjectParent.animationTime / 2;
137             AnimateChild();
138         }
139     }
140 
141     private void ParentAnimationStarted()
142     {
143         //Confusing, but THIS IS WHAT WORKS FOR PARENT AUDIO
144         aParentSource = GetComponents<AudioSource>();
145 
146         interactiveObjectParent = transform.GetComponent<InteractiveObject>();
147          if (aParentSource.Length == 1) aParentSource[1] = aParentSource[0];
148         {
149                 if (interactiveObjectParent.isOpen)
150                        aParentSource [0].Play ();
151                 else if (aParentSource [1])
152                        aParentSource [1].Play ();
153         }
154     }
155 
156     private void ParentAnimationDone()
157 
158     {
159 
160     }
161 
162     private void AnimationController()
163     {
164 
165 
166         if (Regex.IsMatch(gameObject.name, "panel", RegexOptions.IgnoreCase))
CheckOtherPanel();
167         else
168             if (transform.childCount == 0 && interactiveObjectParent)
169             {
170 
171                 iTweenArgs["delay"] = 0;
172                 if (interactiveObjectParent) iTweenParentArgs["delay"] = 0;
173                 if (!isOpen && !interactiveObjectParent.isOpen)
174                 {
175                     CheckOtherPanel();
176                     AnimateChild();
177                     if (interactiveObjectParent) iTweenParentArgs["delay"] =
animationTime;
178                     AnimateParent();
179                 }
180                 else if (isOpen && interactiveObjectParent.isOpen)
181                 {
182                     AnimateParent();
183                     if (interactiveObjectParent) iTweenArgs["delay"] =
interactiveObjectParent.animationTime;
184                     AnimateChild();
185                 }
```

```
186                 if (!isOpen && interactiveObjectParent.isOpen)
187                 {
188 
189                     AnimateParent();
190                     //maybe tweak the opening time?
191                     if (interactiveObjectParent) iTweenArgs["delay"] =
interactiveObjectParent.animationTime / 2;
192                     AnimateChild();
193 
194                 }
195 
196 
197             }
198             else AnimateChild();
199 
200     }
201 
202     private void CheckOtherPanel()
203     //only valid for panels that do not have handles.
204     //In all other cases it should do nothing much.
205     {
206         ZONK = GameObject.Find("ZONK");
207         aZONK = ZONK.GetComponent<AudioSource>();
208         if (Regex.IsMatch(gameObject.name, "panel", RegexOptions.IgnoreCase))
209         {
210             leftPanel = transform.parent.GetChild(0).gameObject;
211             rightPanel = transform.parent.GetChild(1).gameObject;
212 
213             if (!leftPanel.GetComponent<InteractiveObject>().isOpen)
214             {
215                 if (!rightPanel.GetComponent<InteractiveObject>().isOpen)
216                 {
217                     aZONK.Play();
218                 }
219                 else
220                     AnimateChild();
221             }
222             else if (!rightPanel.GetComponent<InteractiveObject>().isOpen)
223             {
224                 aZONK.Play();
225             }
226             else
227                 AnimateChild();
228         }
229         else
230         {
231         }
232     }
233 }
```

Interactor.cs

```
001 using UnityEngine;
002 using System.Collections;
003 using System.Text.RegularExpressions;
004 
005 public class Interactor : MonoBehaviour
006 {
007     [SerializeField]
008     private float interactRange;
009     private InteractiveObject interactiveObject;
010     private PlayMusic playMusic;
011     private Content content;
012    private Elevator elevator;
013     public ElevatorController elevatorController;
```

```
014     private GameObject elevatorButton;
015     private Animator internalDoorAnimation, externalDoorAnimation;
016     private ElevatorExternalDoors elevatorExternalDoors;
017     private Transform hitObject;
018     private Vector3 currentRotation;
019     private Camera cam;
020     private RaycastHit hit;
021     private ReticleController reticleController;
022     private int floor;
023
024     void Start()
025     {
026         cam = Camera.main;
027         reticleController = GameObject.FindObjectOfType<ReticleController>();
028         elevatorController =
GameObject.Find("elevator").GetComponent<ElevatorController>();
029         internalDoorAnimation =
GameObject.Find("elevator").GetComponent<Animator>();
030         externalDoorAnimation = GameObject.Find("Elevator_door-
basement").GetComponent<Animator>();
031     }
032
033     void Update()
034     {
035
036         Physics.Raycast(cam.transform.position, cam.transform.forward, out hit,
interactRange);
037
038         //MUST hit a handle, or an elevator button, or a left shutter (in which
case there must be a check if the right shutter is open)
039          if (hit.transform)
040          {      hitObject = hit.transform.GetComponent<Transform>();
041             if (hit.transform.name == "beethoven") playMusic =
hit.transform.GetComponent<PlayMusic>();
042             else interactiveObject =
hit.transform.GetComponent<InteractiveObject>();
043             if (reticleController.reticleActive)
reticleController.ShowIcon("HandIcon");
044             if (hit.transform.name == "-1") elevatorButton =
hit.transform.gameObject;
045             if (hit.transform.name == "0") elevatorButton =
hit.transform.gameObject;
046             if (hit.transform.name == "1") elevatorButton =
hit.transform.gameObject;
047             if (hit.transform.name == "2") elevatorButton =
hit.transform.gameObject;
048             if (hit.transform.name == "3") elevatorButton =
hit.transform.gameObject;
049             if ((hit.transform.name == "-1") ||
050                 (hit.transform.name == "0") ||
051                 (hit.transform.name == "1") ||
052                 (hit.transform.name == "2") ||
053                 (hit.transform.name == "3") ||
054                 (Regex.IsMatch(hit.transform.name, "elevator",
RegexOptions.IgnoreCase)))
055             {
056                 if (reticleController.reticleActive)
reticleController.ShowIcon("PointIcon");
057             }
058             else if (hit.transform.name == "bust" ||
059                         hit.transform.name == "painting" ||
060                         hit.transform.name == "ionioan-academy-sign" ||
061                         hit.transform.name == "frederick-north-sign" ||
062                         hit.transform.name == "grimani-sign" ||
063                         hit.transform.name == "logo" ||
064                         hit.transform.name == "stage" ||
```

```
065                         hit.transform.name == "ConferenceTable" ||
066                         hit.transform.name == "ConferenceTable2" ||
067                         hit.transform.name == "desk_v2" ||
068                         hit.transform.name == "desk_v2 (1)" ||
069                         hit.transform.name == "desk_v2 (2)" ||
070                         hit.transform.name == "lion" ||
071                         hit.transform.name == "fire extinguisher")
072             {
073                 content = hit.transform.GetComponent<Content>();
074                 if (reticleController.reticleActive)
reticleController.ShowIcon("QuestionIcon");
075 
076 
077             }
078             else if (reticleController.reticleActive)
reticleController.ShowIcon("HandIcon");
079 
080 
081 
082             if (Regex.IsMatch(hit.transform.name, "elevator",
RegexOptions.IgnoreCase)) elevatorExternalDoors =
hit.transform.GetComponent<ElevatorExternalDoors> ();
083 
084 
085 
086         }
087         else {
088                         interactiveObject = null;
089                 playMusic = null;
090                         hitObject = null;
091                 elevator = null;
092                 elevatorExternalDoors = null;
093                 elevatorButton = null;
094                 content = null;
095             if (reticleController.reticleActive) reticleController.ShowIcon
("ReticleIcon");
096         }
097         if (!reticleController.reticleActive)
reticleController.ShowIcon("None");
098         if (Input.GetKeyDown(KeyCode.E))
099          {      if (interactiveObject)
100                 {
101                         //the rotation of the object must be converted to Euler
Angles
102                         currentRotation = hitObject.localEulerAngles;
103                         //and then be brought to the range -180 to 180
104                         //ie. 330 must become -30, 270 to -90 etc.
105                 currentRotation.x = Clamp0360(currentRotation.x);
106                 currentRotation.y = Clamp0360(currentRotation.y);
107                 currentRotation.z = Clamp0360(currentRotation.z);
108                 interactiveObject.PerformAction();
109                 if (elevatorExternalDoors)
110                 {
111                          elevatorExternalDoors.ButtonPressed();
112                 }
113                 if (elevatorButton)
114                 {
115                     StartCoroutine(DostuffInOrder(elevatorButton));
116                 }
117 
118             }
119             if (content)
120             {
121                 content.ShowTextBox(content.gameObject);
122 
123             }
```

```
124             if (playMusic) playMusic.PlayAudio();
125         }
126 
127     }
128     public static float Clamp0360(float eulerAngles)
129     {
130         float result = eulerAngles - Mathf.CeilToInt(eulerAngles / 360f) *
360f;
131         if (result > 270 && result < 360)
132         {
133             result -= 360f;
134         }
135         if (result == -360) result = 0;
136         return result;
137     }
138     IEnumerator DostuffInOrder(GameObject eButton)
139     {
140 
141         floor = elevatorController.floorNumber;
142 
143             switch (floor)
144         {
145             case 0:
146                 externalDoorAnimation = GameObject.Find("Elevator_door-
basement").GetComponent<Animator>();
147                 externalDoorAnimation.SetBool("OpenDoors", false);
148                 break;
149             case 1:
150                 externalDoorAnimation = GameObject.Find("Elevator_door-
0th").GetComponent<Animator>();
151                 externalDoorAnimation.SetBool("OpenDoors", false);
152                 break;
153             case 2:
154                 externalDoorAnimation = GameObject.Find("Elevator_door-
1st").GetComponent<Animator>();
155                 externalDoorAnimation.SetBool("OpenDoors", false);
156                 break;
157             case 3:
158                 externalDoorAnimation = GameObject.Find("Elevator_door-
2nd").GetComponent<Animator>();
159                 externalDoorAnimation.SetBool("OpenDoors", false);
160                 break;
161             case 4:
162                 externalDoorAnimation = GameObject.Find("Elevator_door-
3rd").GetComponent<Animator>();
163                 externalDoorAnimation.SetBool("OpenDoors", false);
164                 break;
165         }
166 
167         yield return new WaitForSeconds(2);
168         internalDoorAnimation.SetBool("OpenDoors", false);
169         yield return new WaitForSeconds(1.5f);
170         elevator =
GameObject.Find("elevator").transform.GetComponent<Elevator>();
171         if (eButton) elevator.ButtonPressed(eButton);
172     }
173 }
```

LevelDisplay.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
004 
005 public class LevelDisplay : MonoBehaviour {
```

```
006
007
008     public Texture2D Level_M1;
009     public Texture2D Level_0;
010     public Texture2D Level_1;
011     public Texture2D Level_2;
012     public Texture2D Level_3;
013     public Renderer rend;
014      private bool isOn;
015
016     // Use this for initialization
017     void Start () {
018           rend = GetComponent<Renderer> ();
019     }
020
021     // Update is called once per frame
022     void Update () {
023
024     }
025
026     public void ChangeTexture(Texture2D texture)
027     {
028           rend.material.mainTexture = texture;
029      }
030 }
```

MainMenu.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
004 using UnityEngine.SceneManagement;
005
006 public class MainMenu : MonoBehaviour {
007
008     public void PlayGame ()
009     {
010         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
011     }
012
013     public void QuitGame ()
014     {
015         //Debug.Log("Quit!");
016         Application.Quit();
017         Application.OpenURL("about:blank");
018     }
019 }
```

PauseMenu.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
004 using UnityEngine.SceneManagement;
005 using UnityStandardAssets.Characters.FirstPerson;
006
007
008 public class PauseMenu : MonoBehaviour {
009
010     public static bool GamePaused = false;
011     public GameObject pauseMenuUI;
012     private GameObject reticleCanvas;
```

```
013     // Update is called once per frame
014     private void Start()
015     {
016         if (pauseMenuUI) pauseMenuUI.SetActive(false);
017         reticleCanvas = GameObject.Find("ReticleCanvas");
018 
019     }
020     void Update () {
021          if (Input.GetKeyDown(KeyCode.P) || Input.GetKeyDown(KeyCode.Escape))
022         {
023             if (GamePaused)
024             {
025                 Resume();
026             } else
027             {
028                 Pause();
029             }
030         }
031    }
032 
033     public void Resume ()
034     {
035         Cursor.lockState = CursorLockMode.Locked;
036         Cursor.visible = (false);
037         pauseMenuUI.SetActive(false);
038 
GameObject.Find("FPSController").GetComponent<FirstPersonController>().enabled =
true;
039         reticleCanvas.SetActive(true);
040         GamePaused = false;
041     }
042 
043     void Pause ()
044     {
045 
GameObject.Find("FPSController").GetComponent<FirstPersonController>().enabled =
false;
046         Cursor.lockState = CursorLockMode.None;
047         Cursor.visible = (true);
048         pauseMenuUI.SetActive(true);
049         reticleCanvas.SetActive(false);
050         GamePaused = true;
051 
052     }
053 
054     public void LoadMenu ()
055     {
056 
GameObject.Find("FPSController").GetComponent<FirstPersonController>().enabled =
false;
057         SceneManager.LoadScene("Menu");
058     }
059 
060     public void QuitGame ()
061     {
062         Application.Quit();
063         Application.OpenURL("about:blank");
064     }
065 }
```

PlayMusic.cs

```
001 using System.Collections;
002 using System.Collections.Generic;
003 using UnityEngine;
```

```
004
005 public class PlayMusic : MonoBehaviour {
006     private AudioSource aSource;
007
008     private void Start()
009     {
010         aSource = GetComponent<AudioSource>();
011         aSource.clip = Resources.Load<AudioClip>("17 - Symphony No 5 in C
minor, 1st Movement, Allegro con brio");
012     }
013
014     public void PlayAudio()
015     {
016         if (aSource.isPlaying) aSource.Stop(); else aSource.Play();
017     }
018 }
```

RecticleController.cs

```
001 using UnityEngine;
002 using System.Collections;
003
004 public class ReticleController : MonoBehaviour
005 {
006     private PauseMenu pauseMenu;
007     public GameObject defaultIcon, interactIcon, pointIcon, questionIcon,
infoPanel;
008     public bool reticleActive = true;
009     void Start()
010     {
011         infoPanel = GameObject.Find("DialogPanel");
012         defaultIcon = GameObject.Find("ReticleIcon");
013         interactIcon = GameObject.Find("HandIcon");
014         pointIcon = GameObject.Find("PointIcon");
015         questionIcon = GameObject.Find("QuestionIcon");
016         interactIcon.SetActive(false);
017         defaultIcon.SetActive(true);
018         pointIcon.SetActive(false);
019         questionIcon.SetActive(false);
020         infoPanel.SetActive(false);
021     }
022
023     public void ShowIcon(string whichIcon)
024     {
025
026         if (reticleActive)
027         {
028             if (whichIcon == "ReticleIcon")
029             {
030
031                 defaultIcon.SetActive(true);
032                 interactIcon.SetActive(false);
033                 pointIcon.SetActive(false);
034                 questionIcon.SetActive(false);
035             }
036             else if (whichIcon == "HandIcon")
037             {
038                 defaultIcon.SetActive(false);
039                 interactIcon.SetActive(true);
040                 pointIcon.SetActive(false);
041                 questionIcon.SetActive(false);
042             }
043             else if (whichIcon == "PointIcon")
044             {
```

```
045                 defaultIcon.SetActive(false);
046                 interactIcon.SetActive(false);
047                 pointIcon.SetActive(true);
048                 questionIcon.SetActive(false);
049             }
050             else if (whichIcon == "QuestionIcon")
051             {
052                 defaultIcon.SetActive(false);
053                 interactIcon.SetActive(false);
054                 pointIcon.SetActive(false);
055                 questionIcon.SetActive(true);
056             }
057             else if (whichIcon == "None")
058             {
059                 defaultIcon.SetActive(false);
060                 interactIcon.SetActive(false);
061                 pointIcon.SetActive(false);
062                 questionIcon.SetActive(false);
063 
064             }
065 
066         }
067         else
068         {
069             defaultIcon.SetActive(false);
070             interactIcon.SetActive(false);
071             pointIcon.SetActive(false);
072             questionIcon.SetActive(false);
073         }
074 
075     }
076 }
```