



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΙΟΥΡΓΙΑ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΕΦΑΡΜΟΓΗΣ
ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΗΝ
ΕΚΜΑΘΗΣΗ ΤΟΥ ΜΑΘΗΜΑΤΟΣ ΙΣΤΟΡΙΑΣ ΤΗΣ Δ'
ΔΗΜΟΤΙΚΟΥ



Του φοιτητή
Θεοδώρου Ιωάννη
Αρ. Μητρώου: 144266

Επιβλέπων
Ευκλείδης Κεραμόπουλος
Καθηγητής

Θεσσαλονίκη 2023

Τίτλος Π.Ε. Δημιουργία εκπαιδευτικής εφαρμογής επαυξημένης πραγματικότητας για την εκμάθηση του μαθήματος Ιστορίας της Δ' Δημοτικού

Κωδικός Π.Ε. 23182

Όνοματεπώνυμο φοιτητή Ιωάννης Θεοδώρου

Όνοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος

Ημερομηνία ανάληψης Π.Ε. 29-03-2023

Ημερομηνία περάτωσης Δ.Ε. 18-09-2023

Βεβαιώνω ότι είμαι η συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ιωάννη Θεοδώρου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Τα τελευταία χρόνια, η τεχνολογία έχει περάσει από την επιστημονική φαντασία στον χώρο της καθημερινότητας, επανασχεδιάζοντας τον τρόπο που αλληλεπιδρούμε με τον κόσμο γύρω μας. Ένα από τα πολλά επιτεύγματα της, είναι το Augmented Reality (AR) ή αλλιώς Επαυξημένη Πραγματικότητα. Πλέον, με την ευρεία χρήση των έξυπνων συσκευών, η υλοποίηση ιδεών μέσω αυτής της τεχνολογίας είναι πιο εύκολη από ποτέ χάρη στα τεχνολογικά άλματα των τελευταίων ετών. Υπάρχουν πλέον εφαρμογές για ευρύ φάσμα τομέων όπως, η ιατρική, η μηχανική, η βιομηχανία, το εμπόριο και φυσικά η ψυχαγωγία. Ωστόσο δεν θα πρέπει να παραλείψουμε τις εκπληκτικές δυνατότητες που μπορεί να προσφέρει στην εκπαίδευση. Μέσω της επαυξημένης πραγματικότητας το πραγματικό περιβάλλον συνδυάζεται με το εικονικό και ενδυναμώνεται από αυτό. Οι μαθητές μπορούν να δουν σε πραγματικό χρόνο και να έχουν την δυνατότητα να αλληλεπιδράσουν διαδραστικά με το επαυξημένο περιβάλλον διαφόρων θεματολογιών, κάτι το οποίο είτε θα ήταν ασύμφορο με συμβατικά μοντέλα ή στις περισσότερες περιπτώσεις αδύνατο να συμβεί. Οι εκπαιδευτικοί μπορούν πλέον να παρουσιάσουν εικονικά μοντέλα ιδεών και καταστάσεων χωρίς να βασίζονται μόνο σε προφορικό λόγο.

Έτσι, επιτυγχάνεται βαθύτερη κατανόηση από τον μαθητή καθώς ουσιαστικά βιώνει το περιεχόμενο του μαθήματος αντί απλά να το αποστηθίζει. Επιπλέον παρέχει και μια μορφή διασκέδασης για τις μικρότερες ηλικίες διατηρώντας το ενδιαφέρον τους. Μια από αυτές τις περιπτώσεις καλείται να παρουσιάσει και η συγκεκριμένη πτυχιακή εργασία, συγκεκριμένα για το μάθημα της Ιστορίας Δ' δημοτικού.

Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως στόχο την δημιουργία μιας εφαρμογής επαυξημένης πραγματικότητας με θέμα την ύλη της Ιστορίας της Δ' δημοτικού. Η χρήση της θα μπορεί να γίνει μέσα από οποιαδήποτε συσκευή που χρησιμοποιεί το λειτουργικό σύστημα Android και θα υποστηρίζεται από εύλογα παλιές συσκευές. Γίνεται χρήση τόσο 2D όσο και 3D στοιχείων, τα οποία συνδυάζονται για να παρουσιάσουν στον χρήστη την ύλη του βιβλίου, με στόχο την βελτίωση της εκπαιδευτικής διαδικασίας.

Έγινε χρήση του περιβάλλοντος Blender για τα τρισδιάστατα μοντέλα και για την καθαυτή ανάπτυξη της εφαρμογής, η πλατφόρμα Unity. Το Visual Studio Code χρησιμοποιήθηκε για τη συγγραφή και διαχείριση του κώδικα σε C# και η βιβλιοθήκη Vuforia για την επαυξημένη αναπαράσταση. Τα φυσικά αντικείμενα ταυτοποίησης που χρησιμοποιεί η Vuforia είναι εικόνες από το βιβλίο, που σκανάρονται μέσω κάμερας. Στα πρώτα κεφάλαια γίνεται ιστορική αναδρομή, παρουσίαση της οικογενείας τεχνολογιών της επαυξημένης πραγματικότητας, και έπειτα ανάλυση των προγραμμάτων και βημάτων που πραγματοποιήθηκαν για την υλοποίηση της εφαρμογής.

Augmented reality educational application for learning the 4th grade History course

Theodorou Ioannis

Abstract

The aim of this thesis is to create an augmented reality application on the subject of 4th grade History. It can be used from any device that uses the Android operating system and will be supported by reasonably old devices. It uses both 2D and 3D elements which are combined to present the material of the book to the user with the aim of improving the educational process. The Blender environment was used for the 3D models and for the application development, the Unity platform. Visual Studio Code was used to write and manage the code in C# and the Vuforia library was used for augmented representation. The physical identification objects used by Vuforia are images from the book, scanned through a camera. In the first chapters there is a historical review, a presentation of the family of augmented reality technologies and then an analysis of the programs and steps carried out to implement the application.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω αρχικά τον επιβλέποντα καθηγητή κύριο Ευκλείδη Κεραμόπουλο που μου έδωσε την ευκαιρία για ανάληψη της εργασίας και δυνατότητα ενασχόλησης με μια πολύ ενδιαφέρουσα και αναπτυσσόμενη τεχνολογία καθώς και με ένα κομμάτι της Ιστορίας αρκετά αγαπημένο για εμένα.

Τέλος, θα ήθελα να ευχαριστήσω τα μέλη της εξεταστικής επιτροπής που δέχτηκαν να αφιερώσουν χρόνο για να αξιολογήσουν την πτυχιακή εργασία μου.

Περιεχόμενα

| | |
|---|-----|
| Πρόλογος | iii |
| Περίληψη | iv |
| Abstract | v |
| Ευχαριστίες | vi |
| Περιεχόμενα | vii |
| Κατάλογος Σχημάτων | xi |
| Συντομογραφίες | xiv |
| Κεφάλαιο 1ο : Εισαγωγή και Σύντομη Αναδρομή | 1 |
| 1.1 Εισαγωγή | 1 |
| 1.2 Η Εργασία και η Δομή της | 2 |
| 1.3 Επίλογος | 2 |
| Κεφάλαιο 2ο : Επταυξημένη και Εκτεταμένη Πραγματικότητα | 3 |
| 2.1 Εισαγωγή | 3 |
| 2.2 Επταυξημένη Πραγματικότητα | 4 |
| 2.3 Μεικτή και Επταυξημένη Εικονικότητα | 5 |
| 2.4 Εικονική Πραγματικότητα | 5 |
| 2.5 Εκτεταμένη Πραγματικότητα | 5 |
| 2.6 Εφαρμογές της Εκτεταμένης Πραγματικότητας | 6 |
| 2.6.1 Μάρκετινγκ και Αγορές | 6 |
| 2.6.2 Υγεία | 7 |
| 2.6.3 Παιδεία, Ψυχαγωγία και Εκπαίδευση | 8 |
| 2.7 Το Μέλλον | 10 |
| 2.8 Θέσεις Εργασίας και Ευκαιρίες | 10 |
| 2.9 Κίνδυνοι και Δυσκολίες | 11 |
| 2.10 Επίλογος | 12 |
| Κεφάλαιο 3ο : Πλατφόρμες και εργαλεία υλοποίησης | 13 |
| 3.1 Εισαγωγή | 13 |
| 3.2 Unity Hub και Unity | 13 |
| 3.2.1 Δημιουργία και Ρύθμιση Νέου Project | 14 |
| 3.3 Unity Package Manager | 17 |
| 3.3.1 TextMeshPro | 17 |

| | |
|---|----|
| 3.3.2 ProBuilder | 17 |
| 3.3.3 Vuforia Engine | 17 |
| 3.3.4 Dreamteck Splines | 18 |
| 3.3.5 Lean Touch | 18 |
| 3.4 Vuforia Engine Setup | 18 |
| 3.4.1 Vuforia Engine Key | 18 |
| 3.4.2 Vuforia Engine Database | 19 |
| 3.4.3 Vuforia Image Target | 21 |
| 3.5 Blender | 21 |
| 3.6 Επίλογος | 22 |
| Κεφάλαιο 4ο: Εφαρμογή και Εισαγωγική Σκηνή | 23 |
| 4.1 Εισαγωγή | 23 |
| 4.2 Εικονίδιο | 23 |
| 4.3 Η Εφαρμογή | 24 |
| 4.4 Εισαγωγική Σκηνή | 25 |
| 4.4.1 Πληροφορίες | 27 |
| 4.4.2 Level Select Panel | 28 |
| 4.5 Επίλογος | 30 |
| Κεφάλαιο 5ο: Γεωμετρικά Χρόνια | 31 |
| 5.1 Εισαγωγή | 31 |
| 5.2 Panel Εισαγωγής | 31 |
| 5.3 Δημιουργία Περιβάλλοντος | 32 |
| 5.3.1 Animation 3D Αντικειμένων | 34 |
| 5.3.1.1 Δημιουργία Μονοπατιού | 34 |
| 5.3.1.2 Κίνηση στο Μονοπάτι | 36 |
| 5.3.2 Λοιπά Αντικείμενα | 39 |
| 5.4 User Interface | 41 |
| 5.4.1 Υλοποίηση Λειτουργικότητας Animation | 43 |
| 5.4.2 Αλληλεπίδραση με τα Αντικείμενα | 45 |
| 5.4.3 Λειτουργικότητα Αλλαγής Σελίδας και Κατάστασης Αντικειμένων | 46 |
| 5.5 Επίλογος | 51 |
| Κεφάλαιο 6ο : Αρχαϊκά Χρόνια | 53 |
| 6.1 Εισαγωγή | 53 |

| | |
|---|-----|
| 6.2 Panel Εισαγωγής | 53 |
| 6.3 Δημιουργία Περιβάλλοντος | 54 |
| 6.4 User Interface | 55 |
| 6.4.1 Animation αντικειμένων UI | 58 |
| 6.4.2 Λειτουργικότητα Quiz και Υλοποίηση | 60 |
| 6.4.2 Κώδικας χειρισμού σκηνής | 63 |
| 6.5 Επίλογος | 66 |
| Κεφάλαιο 7ο : Κλασικά Χρόνια(ΜΑΚΕΔΟΝΙΑ) | 67 |
| 7.1 Εισαγωγή | 67 |
| 7.2 Panel Εισαγωγής | 67 |
| 7.3 Δημιουργία Περιβάλλοντος | 68 |
| 7.3.1 Clashing Animation | 71 |
| 7.4 Δημιουργία Περιβάλλοντος Δεύτερου Image Target | 73 |
| 7.5 User Interface και Λειτουργικότητα | 77 |
| 7.6 Επίλογος | 82 |
| Κεφάλαιο 8ο : Ελληνιστικά Χρόνια - Οι Διάδοχοι του Μ.Αλεξανδρού | 83 |
| 8.1 Εισαγωγή | 83 |
| 8.2 Panel Εισαγωγής | 83 |
| 8.3 Δημιουργία Περιβάλλοντος | 84 |
| 8.4 Δημιουργία Περιβάλλοντος Δεύτερου Image Target | 85 |
| 8.5 Δημιουργία Περιβάλλοντος Τρίτου Image Target | 87 |
| 8.6 User Interface | 90 |
| 8.7 Επίλογος | 91 |
| Κεφάλαιο 9ο : Ελληνιστικά Χρόνια - Ρώμη και Ελλάδα | 93 |
| 9.1 Εισαγωγή | 93 |
| 9.2 Panel Εισαγωγής | 93 |
| 9.3 Δημιουργία Περιβάλλοντος | 94 |
| 9.4 Δημιουργία Περιβάλλοντος Δευτέρου Image Target | 96 |
| 9.5 User Interface | 98 |
| 9.6 Επίλογος | 99 |
| Κεφάλαιο 10ο : Συμπεράσματα και Προτάσεις Βελτίωσης | 101 |
| ΒΙΒΛΙΟΓΡΑΦΙΑ | 102 |
| ΠΑΡΑΡΤΗΜΑ Α: script “ToggleModelsBasedOnPages” | 106 |

| | |
|---|-----|
| ΠΑΡΑΡΤΗΜΑ Β: script “QuizForScene3” | 110 |
| ΠΑΡΑΡΤΗΜΑ Γ: script “QuizForScene4” | 115 |
| ΠΑΡΑΡΤΗΜΑ Δ: script “Scene2CheckForBoats” | 118 |

Κατάλογος Σχημάτων

| | | |
|------------|--|----|
| Σχήμα 1.1 | : Χρονική Εξέλιξη της Ε.Π. | 1 |
| Σχήμα 2.1 | : Εκτεταμένη Πραγματικότητα | 3 |
| Σχήμα 2.2 | : Βαθμοί Ελευθερίας | 6 |
| Σχήμα 2.3 | : Εφαρμογή Ικέα | 7 |
| Σχήμα 2.4 | : Virtual Dressing Room | 7 |
| Σχήμα 2.5 | : Accuvein | 8 |
| Σχήμα 2.6 | : Ανατομία Ανθρώπινου Σώματος με AR Headset | 9 |
| Σχήμα 3.1 | : Unity Hub Version Install | 13 |
| Σχήμα 3.2 | : Εγκατάσταση Έκδοσης με Android SDK | 14 |
| Σχήμα 3.3 | : Δημιουργία Νέου Project | 14 |
| Σχήμα 3.4 | : Unity Preferences | 15 |
| Σχήμα 3.5 | : Build Settings | 16 |
| Σχήμα 3.6 | : Project Settings, Player Tab | 16 |
| Σχήμα 3.7 | : Vuforia Create Key | 18 |
| Σχήμα 3.8 | : Vuforia Create Key and Paste Key | 19 |
| Σχήμα 3.9 | : Δημιουργία Βάσης | 20 |
| Σχήμα 3.10 | : Προσθήκη Εικόνων στην Βάση | 20 |
| Σχήμα 3.11 | : Παράδειγμα Επιλογής Εικόνας σαν Image Target | 21 |
| Σχήμα 3.12 | : Αρχική Σκηνή Blender | 22 |
| Σχήμα 4.1 | : Εικονίδιο Εφαρμογής | 23 |
| Σχήμα 4.2 | : Σκηνές της Εφαρμογής | 24 |
| Σχήμα 4.3 | : Εισαγωγική Σκηνή | 25 |
| Σχήμα 4.4 | : Κώδικας Αλλαγής Κατάστασης Αντικειμένων | 26 |
| Σχήμα 4.5 | : Κώδικας για SerializeField | 26 |
| Σχήμα 4.6 | : onClick() ,Custom Inspector Field | 27 |
| Σχήμα 4.7 | : Info Panel | 27 |
| Σχήμα 4.8 | : Level Select Panel | 28 |
| Σχήμα 4.9 | : Scene Manager | 29 |
| Σχήμα 4.10 | : Κώδικας Αλλαγή Σκηνής | 29 |
| Σχήμα 4.11 | : Ιεραρχία Αντικειμένων StartScene | 29 |
| Σχήμα 5.1 | : Intro Panel | 31 |
| Σχήμα 5.2 | : Εικόνα Πρώτου Image Target | 32 |
| Σχήμα 5.3 | : Blender Export Path Mode | 32 |
| Σχήμα 5.4 | : Unity Material Extraction | 33 |
| Σχήμα 5.5 | : Unity Texture Type Change | 33 |
| Σχήμα 5.6 | : Εικόνες για τα 3D models | 34 |
| Σχήμα 5.7 | : Spline Computer | 35 |
| Σχήμα 5.8 | : Δημιουργία path | 35 |
| Σχήμα 5.9 | : Παράδειγμα Trigger Properties | 36 |
| Σχήμα 5.10 | : Spline Follower | 37 |
| Σχήμα 5.11 | : Spline Follower Part 2 | 38 |
| Σχήμα 5.12 | : Αρχικό Κομμάτι της Πρώτης Σκηνής | 39 |
| Σχήμα 5.13 | : Υπόλοιπα αντικείμενα πρώτης Σκηνής | 39 |
| Σχήμα 5.14 | : Decimate Modifier | 40 |
| Σχήμα 5.15 | : Canvas Scaler | 41 |
| Σχήμα 5.16 | : UI Text Anchor Points | 41 |
| Σχήμα 5.17 | : Κουμπί Επιστροφής | 42 |
| Σχήμα 5.18 | : UI Πρώτης Σκηνής | 43 |

| | | |
|--------------|---|----|
| Σχήμα 5.19 : | Κώδικας ενεργοποίησης animation..... | 44 |
| Σχήμα 5.20: | Script Αλλαγής Σελίδας..... | 47 |
| Σχήμα 5.21 : | Αλλαγή Κατάστασης με Βάση την Σελίδα..... | 48 |
| Σχήμα 5.22 : | Μέθοδοι και Script του BTN Nextpage..... | 49 |
| Σχήμα 5.23 : | Ιεραρχία Αντικειμένων Πρώτης Σκηνής..... | 50 |
| Σχήμα 5.24 : | Υπόλοιπα Αντικείμενα Πρώτης Σκηνής..... | 50 |
| Σχήμα 5.25 : | Ανασκόπηση Πρώτης Σκηνής..... | 51 |
| Σχήμα 6.1 : | Δεύτερο Intro Panel..... | 53 |
| Σχήμα 6.2 : | Εικόνα Δεύτερου Image Target..... | 54 |
| Σχήμα 6.3 : | Αντικείμενα Δεύτερης Σκηνής..... | 54 |
| Σχήμα 6.4: | Εικόνα Πολιτευμάτων..... | 56 |
| Σχήμα 6.5 : | Δημιουργία Τμημάτων Sprites..... | 56 |
| Σχήμα 6.6 : | Τελικά Sprites..... | 57 |
| Σχήμα 6.7 : | Αρχική Κατάσταση Quiz..... | 57 |
| Σχήμα 6.8 : | Ιεραρχία Αντικειμένων UI..... | 58 |
| Σχήμα 6.9 : | Animation TAB Create .anim..... | 59 |
| Σχήμα 6.10 : | Animation Tab..... | 59 |
| Σχήμα 6.11 : | Animator Tab..... | 60 |
| Σχήμα 6.12 : | Inspector για το Animation..... | 60 |
| Σχήμα 6.13 : | Κώδικας Λειτουργίας Quiz..... | 62 |
| Σχήμα 6.14 : | Το Αντικείμενο για το script..... | 63 |
| Σχήμα 6.15 : | Παράδειγμα OnClick() από την Ολιγαρχία..... | 63 |
| Σχήμα 6.16 : | Update method του script ToggleONOFF..... | 64 |
| Σχήμα 6.17 : | Κώδικας Ελέγχου Αντικειμένων με πάτημα αφής..... | 65 |
| Σχήμα 6.18 : | Ιεραρχία Αντικειμένων Δεύτερης Σκηνής..... | 65 |
| Σχήμα 6.19 : | Ανασκόπηση Δεύτερης Σκηνής..... | 66 |
| Σχήμα 7.1 : | Τρίτο Intro Panel..... | 67 |
| Σχήμα 7.2 : | Εικόνα Τρίτου Image Target..... | 68 |
| Σχήμα 7.3 : | ProBuilder Object..... | 69 |
| Σχήμα 7.4 : | Material για το Κράτος της Πέλλας..... | 69 |
| Σχήμα 7.5 : | Sword Material..... | 70 |
| Σχήμα 7.6 : | Sword Pivot..... | 71 |
| Σχήμα 7.7 : | Crossed Sword..... | 72 |
| Σχήμα 7.8 : | Αντικείμενο για το Κεφάλαιο 30..... | 72 |
| Σχήμα 7.9 : | Ιεραρχία για το Κεφάλαιο 30..... | 73 |
| Σχήμα 7.10: | Εικόνα Τετάρτου ImageTarget..... | 74 |
| Σχήμα 7.11 : | Clashing Swords Audio Controller..... | 75 |
| Σχήμα 7.12 : | Αντικείμενο για τα Κεφάλαια 31-33..... | 76 |
| Σχήμα 7.13 : | Ιεραρχία για τα Κεφάλαια 31-33..... | 76 |
| Σχήμα 7.14 : | Κώδικας για Ενεργοποίηση Animation..... | 78 |
| Σχήμα 7.15: | Κώδικας για Σταμάτημα Animation..... | 78 |
| Σχήμα 7.16 : | UI Τρίτης Σκηνής..... | 79 |
| Σχήμα 7.17 : | Μέθοδος που καλούν τα UI κουμπιά στο quiz..... | 81 |
| Σχήμα 7.18 : | Ιεραρχία UI Τρίτης Σκηνής..... | 81 |
| Σχήμα 7.19 : | Ανασκόπηση Τρίτης Σκηνής..... | 82 |
| Σχήμα 8.1 : | Τέταρτο Intro Panel..... | 83 |
| Σχήμα 8.2 : | Εικόνα Πέμπτου Image Target..... | 84 |
| Σχήμα 8.3 : | Δημιουργία Αντικειμένου για το Βασίλειο της Συρίας..... | 85 |
| Σχήμα 8.4: | Τελική Μορφή Πρώτου Χάρτη..... | 85 |
| Σχήμα 8.5 : | Εικόνα Έκτου Image Target..... | 86 |
| Σχήμα 8.6 : | Τελική Μορφή Δευτέρου Χάρτη..... | 87 |

| | |
|--|----|
| Σχήμα 8.7 : Εικόνα Εβδόμου Image Target..... | 87 |
| Σχήμα 8.8 :Χρωματισμός του Κολοσσού..... | 88 |
| Σχήμα 8.9 : UV Editing της Βάσης..... | 89 |
| Σχήμα 8.10 : Ιεραρχία Μοντέλων Σκηνής..... | 89 |
| Σχήμα 8.11 : Ανασκόπηση Τέταρτης Σκηνής..... | 90 |
| Σχήμα 9.1 : Πέμπτο Intro Panel..... | 93 |
| Σχήμα 9.2 : Εικόνα Ογδού Image Target..... | 94 |
| Σχήμα 9.3: Τελική Μορφή Πρώτου Χάρτη της Πέμπτης Σκηνής..... | 95 |
| Σχήμα 9.4 : Ιεραρχία Πρώτου Image Target..... | 95 |
| Σχήμα 9.5 : Εικόνα Ενάτου Image Target..... | 96 |
| Σχήμα 9.6 : Σχήμα 9.6 :Τελική Μορφή Δεύτερου Χάρτη της Πέμπτης Σκηνής Χωρίς Κείμενο..... | 97 |
| Σχήμα 9.7 : Ιεραρχία Δεύτερου Image Target..... | 97 |
| Σχήμα 9.8 : Ανασκόπηση της Πέμπτης Σκηνής..... | 98 |

Συντομογραφίες

| | |
|---------|------------------------------|
| Π.Ε | Πτυχιακή Εργασία |
| ΔΙ.ΠΑ.Ε | Διεθνές Πανεπιστήμιο Ελλάδος |
| VR | Virtual Reality |
| AR | Augmented Reality |
| MR | Mixed Reality |
| XR | Extended Reality |
| AI | Artificial Intelligence |
| APK | Android Package Kernel |
| UI | User Interface |
| TMP | TextMeshPro |
| 3D | Three Dimensional |
| Ε.Π | Επαυξημένη Πραγματικότητα |
| π.χ. | Παραδείγματος Χάρη |
| κ.α. | Και Άλλα |

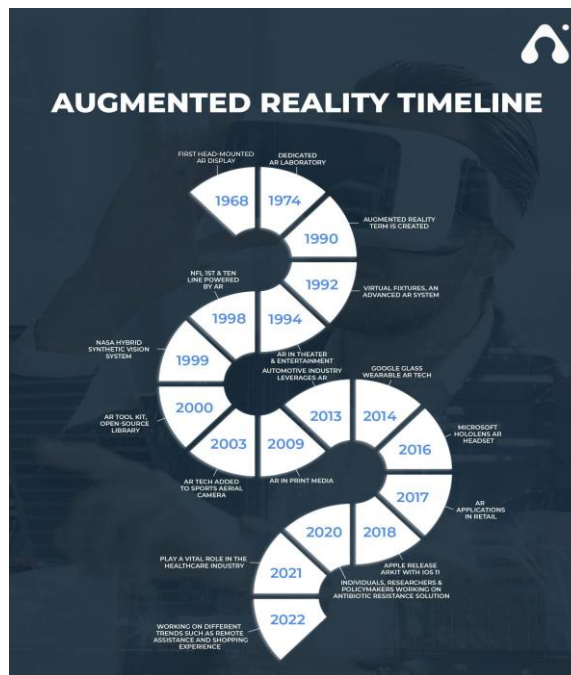
Κεφάλαιο 1ο : Εισαγωγή και Σύντομη Αναδρομή

1.1 Εισαγωγή

Η έννοια της Επαυξημένης Πραγματικότητας υπάρχει από αρκετά παλιά και βασίζεται στην ανάγκη του ανθρώπου να αντλήσει όσο το δυνατόν περισσότερη πληροφορία για το περιβάλλον του μέσω της κύριας και πιο σημαντικής αίσθησης του, της όρασης.

Σαν ιδέα έχει προαναφερθεί από το 1584 στο βιβλίο “Magia Naturalis” του Giambattista Della Porta περιγράφοντας γυαλιά τα οποία μέσω ανάκλασης του φωτός θα δείχνουν αντικείμενα τα οποία δεν βρίσκονται πραγματικά στην θέση που φαίνονται ότι είναι. Επόμενο βήμα ήταν το 1968 από τον Ivan Sutherland με το πρώτο σύστημα οθόνης, το οποίο τοποθετούνταν στο κεφάλι. Ουσιαστική πρόοδος σημειώνεται την δεκαετία του 1990 από ερευνητές της Boeing, Tom Caudell και David Mizell, όπου και αναφέρθηκε για πρώτη φορά ο όρος. Την ίδια περίοδο δημιουργήθηκαν πρωτότυπα για την Αμερικανική Πολεμική Αεροπορία που στόχο είχαν επίσης την βελτίωση εκπαίδευσης των νέων πιλότων. Προς το τέλος της δεκαετίας η Ε.Π άρχισε να κάνει την εμφάνιση της και στα μέσα ψυχαγωγίας, όμως με αρκετά υψηλό κόστος ενσωμάτωσης[1].

Από την νέα χιλιετία του 21ου αιώνα η Ε.Π σημείωσε τεράστια πρόοδο. Σημαντικότερα σημεία είναι η δημιουργία της βιβλιοθήκης ανοιχτού κώδικα AR Toolkit το 2000, όπου χρησιμοποιώντας την ο καθένας μπορεί να φτιάξει εφαρμογές. Έπειτα η δημιουργία των πρώτων φορητών γυαλιών την επομένη δεκαετία από την Google, Microsoft και HoloLens μετέτρεψε ουσιαστικά σε πραγματικότητα την αρχική ιδέα από το 1584. Ενδιάμεσα ωστόσο δεν θα πρέπει να παραληφθεί ότι η καθίδρυση των έξυπνων κινητών ως την πιο σημαντική και διαδεδομένη συσκευή, με συνεχώς αυξανόμενες ικανότητες, έκανε την Handheld AR πιο εύκολα και οικονομικά προσβάσιμη σε όλο τον πλανήτη. Το αποτέλεσμα αυτή την στιγμή είναι να αποτελεί το κύριο μέσο ανάπτυξης και προώθησης της τεχνολογίας διεθνώς[1][2]. Στην παρακάτω εικόνα (Σχήμα 1.1) διαγράφεται η αναδρομή της εξέλιξης της Ε.Π με μερικά από τα κυριότερα στιγμιότυπα ανά τα χρόνια.



Σχήμα 1.1 : Χρονική Εξέλιξη της Ε.Π

1.2 Η Εργασία και η Δομή της

Το κείμενο της πτυχιακής εργασίας αποτελείται από δέκα κεφάλαια. Αρχικά όπως είδαμε έγινε μια μικρή εισαγωγή και αναφορά στην ιστορία της Επαυξημένης Πραγματικότητας. Στο δεύτερο κεφάλαιο θα δούμε τις υπόλοιπες τεχνολογίες που απαρτίζουν τον κύκλο της Εκτεταμένης Πραγματικότητας καθώς επίσης διάφορες εφαρμογές και πεδία, στα οποία μπορεί να ενσωματωθεί με εκπληκτικά αποτελέσματα. Έπειτα στο τρίτο κεφάλαιο γίνεται ανάλυση των προγραμμάτων και πακέτων που χρησιμοποιήθηκαν στην εφαρμογή. Ακολουθούν τα βήματα για την ολοκλήρωση των βασικών ρυθμίσεων. Στο τέταρτο κεφάλαιο υλοποιούμε την εισαγωγική σκηνή που βλέπει ο χρήστης και το βασικό μενού της εφαρμογής.

Από το πέμπτο μέχρι και το ένατο κεφάλαιο κάνουμε ανάλυση των σκηνών που αποτελούν τον βασικό κορμό της εφαρμογής και γίνεται παρουσίαση των βημάτων που ακολουθήθηκαν για την δημιουργία τους. Η κάθε μια αναπαριστά μια διαφορετική ενότητα του σχολικού βιβλίου και περιλαμβάνει διάφορα κεφάλαια του βιβλίου αυτού. Περαιτέρω ανάλυση γίνεται στο τέταρτο κεφάλαιο της Π.Ε.

Τέλος, στο τελευταίο δέκατο κεφάλαιο έχουμε τα συμπεράσματα που προκύπτουν μετά το πέρας της δημιουργίας της εφαρμογής.

1.3 Επίλογος

Σε αυτό το πρώτο κεφάλαιο έγινε μια ιστορική αναδρομή της Ε.Π και παρουσιάστηκε με σύντομη αναφορά η δομή της Π.Ε που έχει σαν θέμα την Ιστορία της Δ' Δημοτικού.

Κεφάλαιο 2ο : Επαυξημένη και Εκτεταμένη Πραγματικότητα

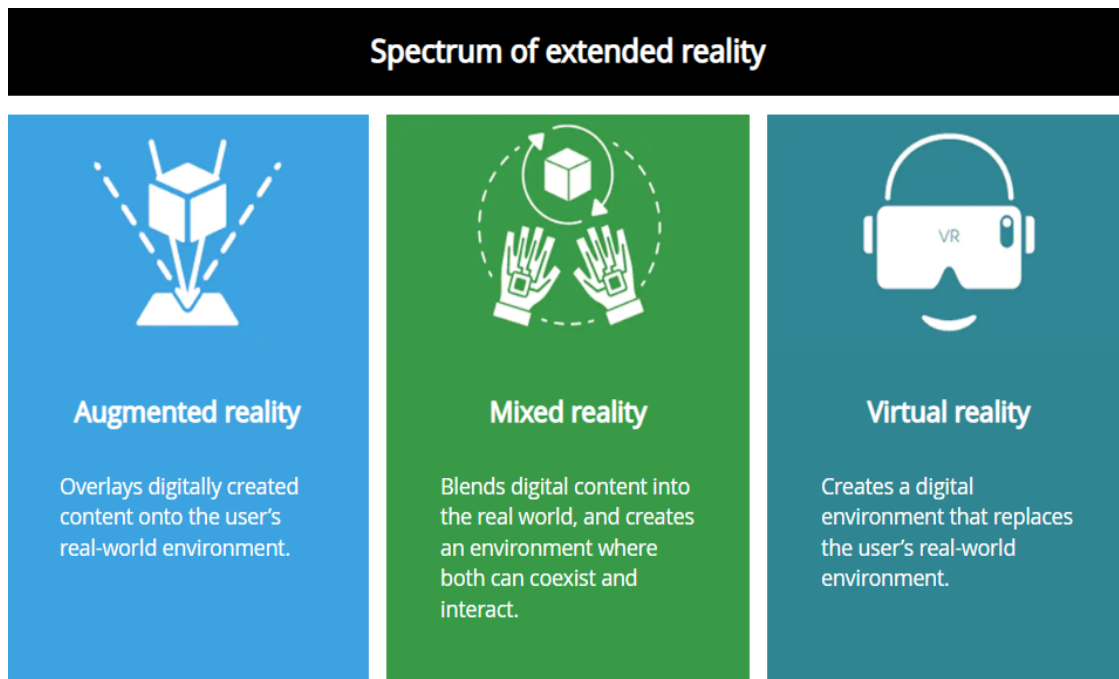
2.1 Εισαγωγή

Η Επαυξημένη πραγματικότητα (Augmented Reality) είναι μία από τις τεχνολογίες που υπόκεινται στο φάσμα της Πραγματικότητας-Εικονικότητας (Σχήμα 2.1) μεταξύ του πραγματικού και του ψηφιακού κόσμου. Οι υπόλοιπες αναφορικά είναι, η Επαυξημένη Εικονικότητα (Augmented Virtuality) ή αλλιώς Μεικτή Πραγματικότητα (Mixed Reality) και η Εικονική Πραγματικότητα (Virtual Reality). Όλες αυτές οι τεχνολογίες αναπτύσσονται, βελτιώνονται ραγδαία και βρίσκουν εφαρμογές που εκμεταλλεύονται τις δυνάμεις τους. Παρόλο που χρησιμοποιούνται επί το πλείστον παρόμοιες μέθοδοι και τεχνικές σε όλες τις τεχνολογίες της Εκτεταμένης Πραγματικότητας κάθε μία από αυτές προσφέρει μια διαφορετική προσέγγιση στα διάφορα ζητήματα που καλείται να λύσει. Έτσι, ανάλογα το ζητούμενο αποτέλεσμα κάποια αποτελεί την βέλτιστη επιλογή. Η παρούσα εργασία κάνει χρήση της Ε.Π για την οποία θα γίνει αναφορά μαζί με τις υπόλοιπες πραγματικότητες [1,2].

Για την επίτευξη της επαύξεσης, αναγκαίο είναι ένα μέσο εισαγωγής πληροφοριών από το περιβάλλον. Πρωταρχικό μέσο είναι κάποια κάμερα ή κάμερες και σαν επιπλέον δυνατότητα διάφοροι αισθητήρες όπως, επιταχυνσιόμετρο (accelerometer) ή και γυροσκοπία ώστε να μπορεί να γίνει κατανοητή από την εφαρμογή η θέση της συσκευής στον χώρο. Τα τελευταία χρόνια η χρήση του GPS (Global Positioning System) είναι πολύ διαδεδομένη, καθώς προσφέρει ακριβή, γρήγορη και έμπιστη πηγή πληροφοριών για εφαρμογές στις οποίες απαιτείται γνώση ευρείας χωροθέτησης μιας περιοχής[1][2].

Τέλος, απαιτείται μια μορφή εμφάνισης των αποτελεσμάτων στον χρήστη, που συνήθως είναι υπό την μορφή κάποιας οθόνης, αλλά υπάρχει και η δυνατότητα προβολής της εικόνας μέσω κάποιου προβολέα (projector).

Όσον αφορά την επεξεργασία των δεδομένων αυτό γίνεται μέσω AR software πχ ARToolKit, ArCore με σημαντική παρατήρηση την αύξηση χρήσης υποβοήθησης από την Τεχνητή Νοημοσύνη (AI) η οποία μειώνει τον χρόνο επεξεργασίας και παρέχει νέες δυνατότητες όπως φωνητική εντολή στις εφαρμογές[3].



Σχήμα 2.1 : Εκτεταμένη Πραγματικότητα

2.2 Επαυξημένη Πραγματικότητα

Σύμφωνα με τον Ronald T. Azuma, διδάκτωρ στο πανεπιστήμιο της Βόρειας Καρολίνα και πρωτοπόρο ερευνητή στον τομέα της Ε.Π: “Η Επαυξημένη Πραγματικότητα (AR) είναι μια παραλλαγή Εικονικού Περιβάλλοντος (VE) ή Εικονική Πραγματικότητα όπως αποκαλείται πιο συχνά. Οι τεχνολογίες VE βυθίζουν πλήρως τον χρήστη μέσα σε ένα συνθετικό περιβάλλον. Ο χρήστης δεν μπορεί να δει τον πραγματικό κόσμο γύρω του. Αντίθετα, το AR επιτρέπει στον χρήστη να δει τον πραγματικό κόσμο, με εικονικά αντικείμενα που υπερτίθενται, επικαλύπτονται ή συνδυάζονται με τον πραγματικό κόσμο. Επομένως, η AR συμπληρώνει την πραγματικότητα, αντί να την αντικαθιστά πλήρως. Στην ιδανική περίπτωση, θα φαινόταν στον χρήστη ότι τα εικονικά και τα πραγματικά αντικείμενα συνυπήρχαν στον ίδιο χώρο”[4].

Για την αποφυγή περιορισμού της Ε.Π σε συγκεκριμένες τεχνολογίες, μπορούμε να διακρίνουμε τα εξής τρία χαρακτηριστικά:

1. Συνδυάζει πραγματικό και εικονικό.
2. Είναι διαδραστική σε πραγματικό χρόνο.
3. Αξιοποιούνται και οι τρεις διαστάσεις από τα αντικείμενα επαύξεσης.

Έτσι δεν αποκλείονται τεχνολογίες όπως γυαλιά ή οθόνες τοποθετημένες σε κεφάλι (HMD) ενώ διατηρείται η ουσία πίσω από την ιδέα της Ε.Π [4].

Η τεχνολογία αυτή μπορεί να χωριστεί σε δυο κατηγορίες, την βασισμένη σε δείκτη (Marker-based AR) και χωρίς δείκτη (Marker-less AR). Κάθε μια από αυτές λύνει διαφορετικά προβλήματα και είναι στην ευχέρεια του δημιουργού της εφαρμογής να λάβει υπόψη του τις δυνατότητες και αδυναμίες της.

Η πρώτη δημιουργείται χρησιμοποιώντας την αναγνώριση εικόνας για τον εντοπισμό αντικειμένων που έχουν ήδη προγραμματιστεί στη συσκευή ή την εφαρμογή. Χρησιμοποιούνται ως σημεία αναφοράς, που βοηθούν την AR συσκευή να καθορίσει τη θέση και τον προσανατολισμό της κάμερας. Όταν ανιχνεύεται ένας δείκτης συγκρίνεται με τους υπόλοιπους που βρίσκονται σε βάση δεδομένων. Μόλις υπάρχει ταίριασμα, η συσκευή χρησιμοποιεί αυτά τα δεδομένα για να προσδιορίσει μαθηματικά την θέση του δείκτη, και να τοποθετήσει το μοντέλο ή την εικόνα AR στο σωστό σημείο [3].

Η αντίθετη περίπτωση είναι πιο περίπλοκη, καθώς δεν υπάρχει συγκεκριμένο σημείο στο οποίο θα επικεντρωθεί η συσκευή. Λόγω αυτού, θα πρέπει να αναγνωρίζει τα αντικείμενα όπως εμφανίζονται στην οθόνη. Χρησιμοποιώντας έναν αλγόριθμο αναγνώρισης, αναζητά χρώματα, μοτίβα και παρόμοια χαρακτηριστικά για να καθορίσει ποιο είναι το κάθε αντικείμενο. Στη συνέχεια, τις συνδυάζει με πληροφορίες από το επιταχυνσιόμετρο, την πυξίδα, και το GPS για να προσανατολιστεί. Τέλος, η εφαρμογή χρησιμοποιεί την κάμερα για να υπερθέσει το μοντέλο ή εικόνα AR στον πραγματικό κόσμο μέσω της οθόνης [3].

Ένα από τα δημοφιλέστερα παραδείγματα είναι το παιχνίδι Pokemon Go. Οι δημιουργοί της εφαρμογής εκμεταλλεζόμενοι την διασημότητα και φύση του παιχνιδιού, που είναι η αιχμαλωσία ψηφιακών φανταστικών ζώων, που αναπαρίστανται ως 3D μοντέλα πάνω στο χώρο ενώ ο παίκτης περιπλανιέται στον κόσμο. Η Ε.Π ήταν η βέλτιστη εφαρμογή τεχνολογίας για τον συνδυασμό του φανταστικού αυτού κόσμου με τον πραγματικό δικό μας και μέσω των handheld devices έγινε γνωστή σε εκατομμύρια ανθρώπους[5]. Γνωστά παραδείγματα είναι επίσης σκηνές από τις ταινίες της Marvel, του Ironman ή ακόμα και στο Terminator όπου ψηφιακά στοιχεία εμπλουτίζουν το αναλογικό περιβάλλον ή επαυξάνουν την ζωντανή μετάδοση βίντεο.

2.3 Μεικτή και Επαυξημένη Εικονικότητα

Η κατηγορία αυτή βρίσκεται στο μέσο του φάσματος Πραγματικότητας-Εικονικότητας και σκοπό έχει την μεταφορά των εικονικών στοιχείων όσο πιο απρόσκοπτα γίνεται στον πραγματικό κόσμο. Η διαφορά είναι ότι σε αυτή την τεχνολογία τα εικονικά αντικείμενα αλληλεπιδρούν τόσο με τον χρήστη όσο και το περιβάλλον στο οποίο βρίσκονται, σαν να ήταν πραγματικά, και καταλαμβάνουν όγκο στον χώρο. Η αλληλεπίδραση του χρήστη μαζί τους μπορεί να γίνει με διάφορα τηλεχειριστήρια ή κινήσεις δακτύλων και χεριών.

Συχνά υπάρχει παρανόηση μεταξύ της Ε.Π και της Μεικτής Πραγματικότητας καθώς με μια πρώτη μάτια φαινομενικά είναι παρόμοιες, ωστόσο υπάρχει ειδοποιός διαφορά μεταξύ τους. Η πρώτη απλά επαυξάνει με κάποιου είδους πληροφορία ή αντικείμενο τον χώρο μέσα από κάποιο μέσο προβολής (οθόνης η projector). Η δεύτερη αποτελεί μέρος του χώρου και τα αντικείμενα που δημιουργούνται επηρεάζονται πλήρως, εκτός από εμάς, και από το περιβάλλον γύρω μας, σε πραγματικό χρόνο.

Με τον όρο Επαυξημένη Εικονικότητα αναφερόμαστε στην περίπτωση όπου στοιχεία του πραγματικού κόσμου έχουν εισαχθεί στην Εικονική Πραγματικότητα (VR) όπως κάποια φωτογραφία ή βίντεο τα οποία μπορούμε να δούμε στον εικονικό κόσμο [6].

2.4 Εικονική Πραγματικότητα

Η Ε.Π και η Μεικτή Πραγματικότητα συνδυάζουν τον αναλογικό με τον εικονικό κόσμο φέρνοντας τα ψηφιακά στοιχεία στο πραγματικό περιβάλλον γύρω μας με στόχο την ενσωμάτωση τους όσο πιο κοντά γίνεται στην πραγματικότητα. Σε αντίθεση η Εικονική Πραγματικότητα έχει ως σκοπό την πλήρη βύθιση του ατόμου σε έναν διαφορετικό εικονικό κόσμο μεταφέροντας ουσιαστικά τον χρήστη σε περιβάλλον που είναι πλήρως ψηφιακό. Για την επίτευξη αυτή χρειάζεται να αποκοπεί η εισροή ερεθισμάτων από τον περιβάλλοντα χώρο και να αντικατασταθεί από τον εικονικό.

Αυτό επιτυγχάνεται παραπληροφώνοντας τις κύριες αισθήσεις της όρασης και την ακοής. Το δεύτερο είναι πιο εύκολο καθώς γίνεται χρήση ακουστικών, όμως για την προαναφερθείσα χρησιμοποιούνται ειδικά κράνη με οθόνες που ονομάζονται headset ή αλλιώς head mounted display(HMD), οι οποίες έχουν τον ρόλο να δείχνουν στερεοσκοπικά την εικόνα στον χρήστη δημιουργώντας έτσι μια ψευδαίσθηση βάθους. Να σημειωθεί ότι θα πρέπει να αποκόπτουν εντελώς την όραση του πραγματικού κόσμου, για να μπορούν να συμπεριληφθούν σε αυτή την κατηγορία [7].

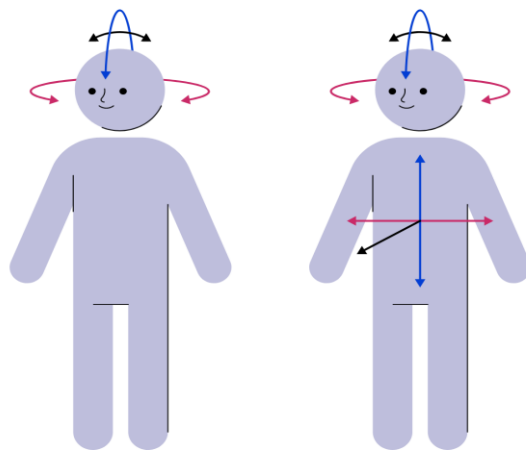
2.5 Εκτεταμένη Πραγματικότητα

Χρησιμοποιώντας τον όρο Extended Reality(XR) αναφερόμαστε σε όλες τις τεχνολογίες οι οποίες υπόκεινται στο φάσμα της Εικονικότητας. Κατ' επέκταση μπορούμε να πούμε ότι συμπεριλαμβάνει όλες τις νέες τεχνολογίες που δεν έχουν δημιουργηθεί ακόμα, καθώς είναι ένας κλάδος που εξελίσσεται ραγδαία, και δεν έχουμε δει ακόμα το απόγειο της. Τεχνολογίες όπως το 5G, cloud computing, και η μηχανική μάθηση είναι δεδομένο ότι θα συμβάλουν στην βελτίωση τόσο της τεχνολογίας, όσο και των τελικών εφαρμογών τα επόμενα χρόνια [8-10].

Κάποιες από τις ιδιότητες και ανάγκες που τείνουν να χαρακτηρίζουν τις τεχνολογίες αυτές, και ως επακόλουθο τις συσκευές που τις αξιοποιούν, είναι:

Επαυξημένη και Εκτεταμένη Πραγματικότητα

1. **Χαμηλή Καθυστέρηση** τόσο μεταξύ της εναλλαγής των σκηνών και φόρτωση των αναγκαίων μοντέλων αλλά όσο και στο μέσο προβολής της τελικής εικόνας. Η συνεχής διατήρηση του ρυθμού μετάδοσης των εικόνων (frame rate) πρέπει να είναι σταθερός και ιδανικά πάνω από 90 εικόνες ανά δευτερόλεπτο (FPS)
2. **Υψηλή Ανάλυση** ειδικά για τις VR εφαρμογές καθώς σε διαφορετική περίπτωση η αίσθηση της συνοχής και εμπάθυνσης στον κόσμο χάνεται.
3. **Οπτικό Πεδίο** το οποίο βέλτιστα θα πρέπει να είναι κοντά στις 200-220 μοίρες που αντιλαμβάνεται το ανθρώπινο μάτι ώστε, να αποτυπώνεται όσο πιο φυσικά γίνεται η εικόνα και η πληροφορία.
4. **Βαθμοί Ελευθερίας** όπου και υπάρχουν δύο περιπτώσεις όπως φαίνεται στην παρακάτω εικόνα (Σχήμα 2.2). Αρχικά είναι το επίπεδο που αναγνωρίζει περιστροφή μόνο στους τρεις άξονες X-Y-Z οπότε ονομάζεται με τρεις βαθμούς ελευθερίας (3DoF). Δεν υπάρχει καταγραφή της κίνησης του χρήστη στον χώρο, οπότε δεν μπορεί να περπατήσει, να σκύψει ή να χοροπηδήσει. Αποτελεί παλαιότερη τεχνολογία και συναντάται συνήθως σε headsets που είναι πιο οικονομικά ή χαμηλότερης ποιότητας. Η επόμενη κατηγορία ονομάζεται έξι βαθμών ελευθερίας(6DoF) προσφέροντας πλέον την δυνατότητα των προαναφερθέντων και επιπλέον με πλήρη μετακίνηση του χρήστη στον χώρο προς οποιαδήποτε κατεύθυνση και αν επιθυμεί [8-10] .



Σχήμα 2.2 : Βαθμοί Ελευθερίας

2.6 Εφαρμογές της Εκτεταμένης Πραγματικότητας

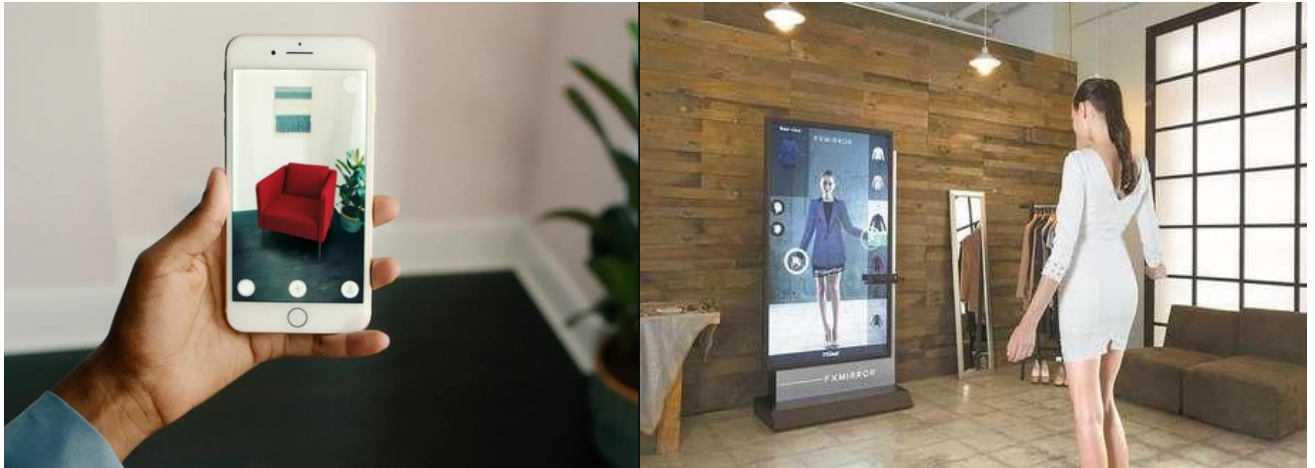
Αρχικά το σύνολο των τεχνολογιών που απαρτίζουν την XR από πειραματικά, ερευνητικά μοντέλα μεταφέρθηκαν σε εφαρμογές παιχνιδιών, ψυχαγωγίας και διασκέδασης.

Τα τελευταία χρόνια όμως οι νέες εφαρμογές τείνουν ολοένα να μετακινηθούν σε άλλους τομείς, σημειώνοντας εκπληκτικά αποτελέσματα και δημιουργώντας προσδοκίες τόσο για την καθημερινότητα, όσο και για την επιχειρηματικότητα. Στην παρούσα υποενότητα, θα αναφερθούν κάποιοι από τους σημαντικότερους τομείς, ενώ έχουν ήδη αρχίσει να εμφανίζονται οι πρώτες εφαρμογές.

2.6.1 Μάρκετινγκ και Αγορές

Το μεγαλύτερο αντίκτυπο αναμφίβολα, καταγράφοντας και την μεγαλύτερη αντίδραση από το κοινό, έχει υπάρξει στον τομέα του εμπορίου. Εταιρείες κυρίως ένδυσης, υπόδησης, καλλυντικών και επίπλων παρέχουν υπηρεσίες τόσο AR όσο και MR δίνοντας την ευκαιρία στους υποψηφίους πελάτες τους να

“δοκιμάζουν” επάνω τους ή στο χώρο τους, τα προϊόντα προτού προβούν σε κάποια αγορά. Κάτι τέτοιο είναι ιδιαίτερα χρήσιμο σε μια αγορά που οι καταναλωτές προτιμούν να αγοράζουν μέσω ιστοσελίδων. Έχοντας μέχρι τώρα μόνο εικόνες σαν γνώμονα αγοράς, πάνω από το 60% προτιμά μαγαζιά που προσφέρουν τέτοιες υπηρεσίες. Κάποια παραδείγματα αποτελεί εφαρμογές όπως μπορούμε να δούμε παρακάτω (Σχήμα 2.3,2.4) της L’Oreal, Warby Parker, Lenskart, Sephora, IKEA και το Dressing Room από την Gap τόσο και από άλλες εταιρίες [11-12] .



Σχήμα 2.3,2.4 : Εφαρμογή Ικέα(αριστερά) και Virtual Dressing Room(δεξιά)

Ακόμα χρησιμοποιείται για την δημιουργία AR επαγγελματικών καρτών, προβάλλοντας έτσι επιπλέον πληροφορίες και βίντεο κάτι που δεν θα μπορούσε να χωρέσει σε μια συμβατική κάρτα, κερδίζοντας το ενδιαφέρον του υποψήφιου πελάτη. Επίσης για την πλοήγηση σε χώρους εταιριών, καταλυμάτων και για την αγορά ακινήτων, όπως σπιτιών ή διαμερισμάτων. Μέσω μια AR εφαρμογής ο ενδιαφερόμενος μπορεί να έχει μια πρώτη άποψη για τον χώρο χωρίς να χρειαστεί η φυσική του μετακίνηση εκεί, ώστε να αποφασίσει αν του ταιριάζει η όχι.

Τέλος, μπορεί να δώσει μια παραπάνω διάσταση και στον τομέα της εστίασης, μέσω των προστιθέμενων εικονικών στοιχείων, ο πελάτης θα μπορεί να δει πως μοιάζει ένα πιάτο πριν το παραγγείλει και ίσως να δει τις διατροφικές αξίες και τα συστατικά του. Προσπάθειες έχουν γίνει ωστόσο και με εικονικούς μπάρμαν που αναλύουν την προέλευση και γεύση των ποτών [13-14] .

2.6.2 Υγεία

Ένας άλλος κύριος τομέας στον οποίο έχει αρχίσει ήδη να φαίνεται η συμβολή και χρησιμότητα των τεχνολογιών της XR είναι αυτής της επιστήμης της Ιατρικής και της Υγείας. Οι δυνατότητες του VR μπορούν να χρησιμοποιηθούν για τη δημιουργία ειδικών πλήρως ελεγχόμενων περιβαλλόντων με σκοπό την θεραπεία ατόμων που αντιμετωπίζουν διάφορες φοβίες ή ψυχικές διαταραχές, όπως διαταραχή μετατραυματικού στρες (PTSD). Έτσι, θα δημιουργηθούν νέες θεραπείες που δεν θα μπορούσαν να υλοποιηθούν στον πραγματικό κόσμο. Με τις δυνατότητες του MR οι γιατροί θα μπορούν εύκολα ακόμα και σε καταστάσεις τηλεδιάσκεψης και απομακρυσμένων θεραπειών να επικοινωνήσουν, δείχνοντας το πρόβλημα σε εικονικά ανατομικά μοντέλα, βοηθώντας την κατανόηση της κατάστασης αντί να βασίζονται απλά στην ομιλία[4,7,10-15].

Επαυξημένη και Εκτεταμένη Πραγματικότητα

Επίσης σημαντικό ρόλο παίζει και στην εκπαίδευση των γιατρών. Οι εκπαιδευόμενοι γιατροί μπορούν να εκπαιδευτούν πάνω στο σώμα μέσω 3D μοντέλων και ολογραμμάτων, βλέποντας εσωτερικά σημεία του σώματος και την λειτουργικότητα τους, αποκτώντας μια καλύτερη γνώση κάτι το οποίο δεν θα μπορούσε να γίνει πάνω σε κάποιον άνθρωπο λόγω ηθικών ζητημάτων [4,7,10-15].

Με την χρήση της AR τεχνολογίας και την χρήση ειδικών γυαλιών θα μπορεί ο γιατρός να βλέπει την κατάσταση του ασθενή, όλες τις πληροφορίες του φακέλου του ακόμα και κατά τη διάρκεια χειρουργείων, απελευθερώνοντας έτσι τα χέρια του. Χρήσεις όπως η 3D απεικόνιση υπερήχων κατευθείαν πάνω στην περιοχή της εξέτασης βοηθούν τον εξεταστή να έχει μια πιο πλήρη εικόνα της κατάστασης του ασθενή [4,7,10-15].

Τέλος, ένα ακόμα παράδειγμα της εφαρμογής είναι το Accuvein (Σχήμα 2.1). Πρόκειται για μια συσκευή που εντοπίζει τις φλέβες και τις προβάλλει πάνω στο δέρμα του ασθενή, βοηθώντας στην λήψη αίματος και την τοποθέτηση φλεβοκαθετήρων. Μειώνεται έτσι ο πόνος κατά 39% και η πιθανότητα πρόκλησης τραυματισμού [4,7,10-15].



Σχήμα 2.5 : Accuvein

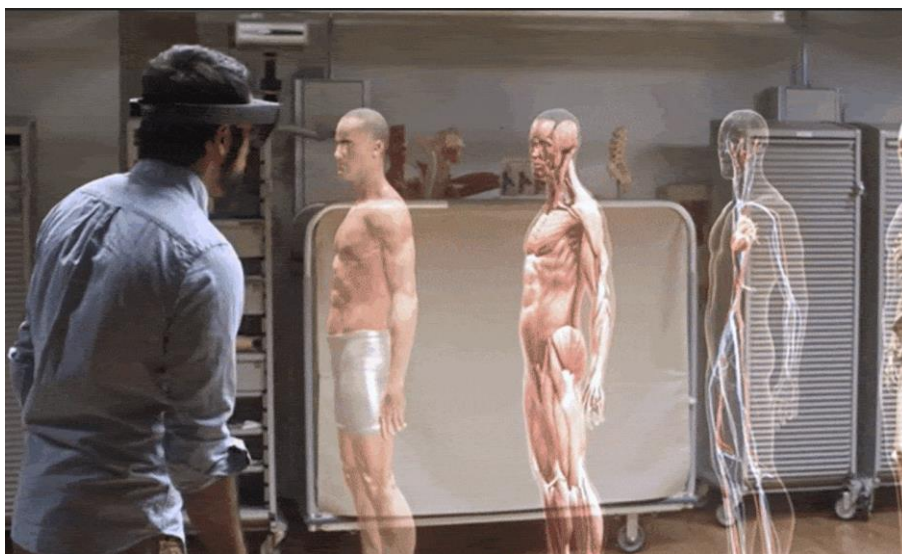
2.6.3 Παιδεία, Ψυχαγωγία και Εκπαίδευση

Η ψυχαγωγία είναι ο τομέας που το ευρύ κοινό είχε την μεγαλύτερη επαφή με την XR, κυρίως υπό την μορφή του AR μέχρι την άνθηση του στον τομέα του εμπορίου. Σημαντικός σύμμαχος υπήρχε το smartphone το οποίο μπορεί να αντεπεξέλθει στις απαιτήσεις της τεχνολογίας βοηθώντας την ανάπτυξη του AR gaming.

Διασημότερο παράδειγμα αποτελεί το παιχνίδι Pokémon GO που δημοσιεύτηκε το 2016. Αξιοποίησε συνδυασμό του AR με την γεωγραφική τοποθεσία μέσω GPS. Οι χρήστες έπρεπε να κινηθούν στον πραγματικό κόσμο ψάχνοντας με την κάμερα του κινητού τους ψηφιακά πλάσματα που μπορούσαν να συναντήσουν τα οποία εμφανίζονται στην οθόνη του κινητού τους με σκοπό να τα πιάσουν. Η κίνηση αυτή απέφερε στην εταιρία πάνω από 550 εκατομμύρια δολάρια τον πρώτο χρόνο. Πλέον τα έσοδα της ξεπερνάνε τα 5.5 δισεκατομμύρια δολάρια έχοντας πάνω από 1 δισεκατομμύριο λήψεις μέχρι σήμερα. Επιπλέον παραδείγματα αποτελούν άλλα διάσημα παιχνίδια όπως Temple Treasure Hunt και Sharks in the Park [12,16].

Σημαντικό βήμα αποτελεί η δημιουργία και πλέον η δυνατότητα αγοράς απτικών γαντιών (haptic gloves) η οποία ήταν υπό έρευνα και ανάπτυξη την τελευταία πενταετία. Χρησιμοποιώντας τα, προσομοιάζεται η αλληλεπίδραση και επαφή με τα εικονικά αντικείμενα σαν να ήταν πραγματικά με την δυνατότητα αναπαραγωγής ακόμα και της υφής. Τελευταία γίνεται προσπάθεια ώστε να μπορέσει να προσομοιάσει και η θερμική διαφορά, δίνοντας την αίσθηση της θερμής ή της κρύας αφής στον χρήστη.[17,18]

Δεν πρέπει ωστόσο να υποτιμήσουμε το σημαντικό ρόλο που μπορεί να παίζει στο κομμάτι της εκπαίδευσης. Μέσω των δυνατοτήτων που παρέχει η XR επιτρέπει στις επιχειρήσεις και τους οργανισμούς να μειώσουν το κόστος που απαιτείται για την κατάρτιση και εκπαίδευση. Επίσης, να μετριάσουν τους κινδύνους που μπορεί να ελλοχεύουν κατά την διάρκεια της. Κάτι τέτοιο είναι εμφανές ιδίως όταν πρόκειται για εργασίες με επικίνδυνες καταστάσεις και υλικά, αφού μπορούν να προσομοιαστούν εικονικά. Μέσα από την αλληλεπίδραση με τα εικονικά στοιχεία αποφεύγεται η χρήση και σπατάλη πραγματικών πόρων και επιτυγχάνεται η καλύτερη προσέλευση στο αντικείμενο της εκπαίδευσης. Παραδείγματα στον ιατρικό τομέα αποτελούν The Human Anatomy Atlas App με πάνω από 10.000 ανατομικά μοντέλα και πληροφορίες για αυτά ή παρόμοιες εφαρμογές (Σχήμα 2.6) [12,14].



Σχήμα 2.6 : Ανατομία Ανθρώπινου Σώματος με AR Headset

Τέλος, ιδιαίτερο ενδιαφέρον παρουσιάζουν εφαρμογές πλοήγησης σε μουσεία, ιστορικούς χώρους και τοπογραφικά μέρη τα οποία είναι είτε δυσπρόσιτα στο ευρύ κοινό (Εβερρεστ, Αμαζόνιος) είτε λόγω ότι αποτελούν ιστορικά μνημεία που διασώζεται μόνο μέρος τους. Παραδείγματα τέτοιων εφαρμογών είναι το smartify που δίνει δυνατότητα επαύξησης των μουσειακών εκθεμάτων μαζί με άλλες δυνατότητες[19]. Σημαντική αναφορά είναι η νέα εφαρμογή από την Cosmote, το CHRONOS App, όπου με την βοήθεια τεχνολογιών όπως ARKit και ARCore ζωντανεύει ξανά μνημεία όπως η Ακρόπολη και εκθέματα του Μουσείου της με την πλήρη μορφή που είχαν αιώνες πριν, δίνοντας την δυνατότητα να πλοηγηθούμε μέσα τους, ενώ αναπαριστάνονται στην οθόνη του κινητού ενώ κινούμαστε μέσα στον χώρο μας[20].

2.7 Το Μέλλον

Καθώς βρισκόμαστε στο κατώφλι της τέταρτης βιομηχανικής επανάστασης, τεχνολογίες που αναπτύσσονται τα τελευταία χρόνια αρχίζουν σιγά σιγά να αποφέρουν καρπούς, καθώς είναι σε θέση πλέον να υλοποιήσουν εφαρμογές με ουσιαστικά αποτελέσματα. Η καθημερινότητα μετασχηματίζεται σε ψηφιακή, χωρίς να αποκλείεται κανένας κλάδος από τις αλλαγές και τις βελτιώσεις που προσφέρει η XR συνδυαστικά με άλλα επιτεύγματα όπως, την καθίδρυση του 5G, της ρομποτικής, της τεχνητής νοημοσύνης και τις νέες προοπτικές που φέρνουν οι διάφορες συσκευές προβολής [12,21].

Πιο συγκεκριμένα χάρη στο 5G μπορούμε να μεταφέρουμε τεράστιο όγκο πληροφοριών γρήγορα. Αυτό σημαίνει ότι δεν χρειάζεται να αποθηκεύει τοπικά η συσκευή τα δεδομένα για τις εφαρμογές πράγμα που ήταν αναγκαστικό πρωτότερα. Ως αποτέλεσμα έχει ότι μπορούν παραπάνω από μια συσκευές να αλληλεπιδρούν με το ψηφιακό περιβάλλον της MR. Μπορεί να γίνει έτσι αξιοποίηση και άλλων τεχνολογιών όπως η ρομποτική, και ο εξ' αποστάσεως χειρισμός της. Σε εγχειρήματα όπως αυτά κύριο ρόλο θα παίξει η τεχνητή νοημοσύνη η οποία θα υποβοηθά τον χρήστη τόσο στην απλούστευση των αναγκαίων χειρισμών όσο και στην αλληλεπίδραση, π.χ. μέσω φωνητικών εντολών ή κινήσεων των χεριών, και επεξεργασία των πληροφοριών πριν την παρουσίαση τους.

Το μόνο σίγουρο είναι ότι η δημοτικότητα και ζήτηση αυτής της τεχνολογίας θα συνεχίσει να αναπτύσσεται ανά τα χρόνια. Συγχρόνως γίνεται πιο προσιτή στους καταναλωτές προσφέροντας νέες ευκαιρίες αλλά και προκλήσεις στις εταιρείες και τους ερευνητές. Με τις υπάρχουσες προβλέψεις υπολογίζεται ότι μέχρι το 2028 η αγορά της Αμερικής μόνο για εφαρμογές AR θα φτάσει τα 97.76 δισεκατομμύρια δολάρια [13,21-23].

2.8 Θέσεις Εργασίας και Ευκαιρίες

Λόγω λοιπόν της τεράστιας ανάπτυξης και ενδιαφέροντος πίσω από αυτό τον κλάδο της πληροφορικής όπως είναι αναμενόμενο ανοίγουν πολλές νέες θέσεις εργασίας. Δημιουργούνται ευκαιρίες για ανάπτυξη καριέρας με πλούσιες απολαβές ειδικά για τους πιο καταξιωμένους στον χώρο. Φυσικά οι μεγαλύτερες ευκαιρίες βρίσκονται στον κλάδο της πληροφορικής καθώς αποτελεί τον κύριο κλάδο ανάπτυξης των τελικών “προϊόντων”[24-25].

Βασικές δεξιότητες που είναι απαραίτητες για την δημιουργία τέτοιων εφαρμογών είναι η γνώση πλατφορμών όπως Unity, Unreal engine, Android Studio αφού είναι τα κύρια περιβάλλοντα ανάπτυξης που χρησιμοποιούνται μέχρι στιγμής. Ειδικά για το κομμάτι του VR απαιτείται γνώση των Oculus, Vive, πλατφόρμες οι οποίες ηγούνται στην τεχνογνωσία και ανάπτυξη εικονικών περιβαλλόντων με τον αντίστοιχο υλικό εξοπλισμό [24-25].

Για τον προγραμματισμό χρησιμοποιούνται κατά κύριο λόγο γλώσσες C++, C#, Objective-C και Swift, οπότε ο προγραμματιστής που ενδιαφέρεται να ενταχθεί σε αυτόν τον κλάδο οφείλει να είναι άριστος γνώστης τουλάχιστον μιας από αυτές. Η γνώση διαφόρων λειτουργικών συστημάτων κρίνεται επίσης χρήσιμη [24-25].

Σε βαθύτερο επίπεδο γίνονται έρευνες για την ανάπτυξη νέων περιβαλλόντων ανάπτυξης, αλγορίθμων και διεπαφών προγραμματισμού εφαρμογών, που θα αξιοποιούν βέλτιστα τα νέα επιτεύγματα στον τομέα των hardware. Αναφορικά παραδείγματα είναι τα Vulkan, OpenGL, DirectX και ως επακόλουθο υπάρχει η προϋπόθεση γνώσης Java, Python και C [24-25].

Τέλος, όσον αφορά το κομμάτι των δημιουργών, αναγκαίος είναι ο ρόλος γραφιστών και 3D Artists/Graphics Engineer καθώς είναι αυτοί που κατά κύριο λόγο έχουν τις δεξιότητες να δημιουργήσουν πιστά και ρεαλιστικά τρισδιάστατα μοντέλα. Δίνοντας τους κίνηση και ρεαλιστικά χρώματα μπορούν να χρησιμοποιηθούν για την δημιουργία μιας πιο απρόσκοπτης εμπειρίας. Παραδείγματα τέτοιων προγραμμάτων αυτού του χώρου είναι Blender, Maya, 3ds Max, SketchUP, Autodesk 3D [24-25].

Ανάγκες για θέσεις εργασίας θα προκύψουν στο κομμάτι συντήρησης και υποστήριξης τόσο των συσκευών όσο και των προγραμμάτων. Ειδικότερα για εφαρμογές που απευθύνονται στον βιομηχανικό τομέα με σκοπό την ομαλή λειτουργία της παραγωγής, την σωστή εκπαίδευση των εργαζομένων, και την πρόληψη σφαλμάτων του υλικοτεχνικού εξοπλισμού [24-25].

2.9 Κίνδυνοι και Δυσκολίες

Παρόλα τα οφέλη και ευκολίες που παρέχουν οι τεχνολογίες της XR αυτό δεν σημαίνει ότι έρχονται και χωρίς κάποια αρνητικά στοιχεία και επιπτώσεις. Αρχικά το πιο εύκολα αντιληπτό είναι τα επιπλέον έξοδα που αναγκάζονται να προβούν, τόσο οι εταιρίες όσο και οι καταναλωτές. Αρχικά για τους χρήστες μέχρι στιγμής οι περισσότερες εφαρμογές Ε.Π είναι δωρεάν, ωστόσο για το VR χρειάζεται να καλύψουν τα έξοδα του εξειδικευμένου εξοπλισμού που είναι ακόμα πολύ ακριβός και του προγράμματος που θα χρησιμοποιήσουν. Όσον αφορά τις εταιρίες, υπάρχει το κόστος για την άδεια χρήσης του περιβάλλοντος ανάπτυξης και το κόστος δημιουργίας του λογισμικού της εφαρμογής.

Για την δημιουργία μιας τέτοιας εφαρμογής μπορεί να ξοδέψουν μερικές χιλιάδες δολάρια. Σε περίπτωση πλήρους ανάπτυξης από το μηδέν, δεν είναι απίθανο για το κόστος, να φτάσει ακόμα και στα εξαψήφια νούμερα. Κάτι τέτοιο μπορεί να μην είναι τόσο σημαντικό για τις εταιρίες κολοσσούς, ωστόσο φέρνει σε μειονεκτική θέση τις μικρομεσαίες επιχειρήσεις που αδυνατούν να καλύψουν τέτοιο κόστος μειώνοντας έτσι την ανταγωνιστικότητά τους.

Κομμάτι της παραπάνω δυσκολίας έρχεται να εντείνει η έλλειψη προτύπων, με αποτέλεσμα η ανάπτυξη να καθυστερεί. Πολλές φορές επίσης, οι διαφορετικές λύσεις δεν είναι συμβατές μεταξύ τους ώστε να συνδυαστούν. Έτσι αυξάνεται το κόστος και η σύγχυση αφού πολλές φορές πρέπει να βρεθούν διαφορετικές λύσεις για το ίδιο πρόβλημα.

Λόγω της φύσης του XR είναι αναγκαία η συνεχής λειτουργία κάποιας κάμερας ή και σε πολλές περιπτώσεις επιπλέον αισθητήρων που καταγράφουν τον χώρο και την γεωγραφική τοποθεσία που βρισκόμαστε. Με την καθιέρωση πλέον του 5G, τα δεδομένα θα μεταβαίνουν στο cloud για επεξεργασία πριν την επαναποστολή στην συσκευή μας, πράγμα που τα καθιστά ευάλωτα σε κυβερνοεπιθέσεις, κλοπή-διάρρευση πληροφοριών και ταυτότητας. Επίσης σημαντική ευπάθεια είναι η αποστολή ψευδών δεδομένων με σκοπό την παραπληροφόρηση ή πρόκληση ζημιάς, για χρήσεις όπως πλοήγηση ή ρομποτικές εφαρμογές. Επιπλέον υπάρχει το πρόβλημα της δηκτικότητας καθώς έχοντας ανοιχτή συνέχεια την κάμερα καταγράφουμε τους περαστικούς και άτομα στο background. Ο τρόπος χειρισμού τέτοιου είδους δεδομένων θα αποτελέσει θέμα συζήτησης για την ηθική και νομική επίπτωση [26,27,28].

Τέλος σημαντικό ζήτημα είναι αυτό της υγείας. Καθώς είναι μια νέα τεχνολογία η οποία δεν έχει προλάβει ακόμα να δοκιμαστεί εκτενώς και να ωριμάσει, δεν γνωρίζουμε τις μακροχρόνιες επιπτώσεις που μπορεί να υπάρξουν. Εκδηλώνονται ανησυχίες για την πρόκληση διαφόρων προβλημάτων υγείας, κυρίως στην όραση. Ειδικότερα στην περίπτωση του VR έχουν ήδη παρατηρηθεί ζαλάδες, πονοκέφαλοι, καταπόνηση των ματιών και φόβοι για εθισμό και κοινωνική απομόνωση [26,27,28].

2.10 Επίλογος

Παρόλο που υπάρχουν προκλήσεις που πρέπει να καθοριστούν και λυθούν, η χρήση του XR τείνει ολοένα στο να καθιερωθεί στην καθημερινότητα μας. Προσφέρει πολλές νέες δυνατότητες σε όλους τους κλάδους και τομείς τόσο της καθημερινότητας όσο και της επιχειρηματικότητας. Η εκρηκτική πορεία εξέλιξης της, την τελευταία πενταετία μας προδιαθέτει για μεγάλες και καινοτόμες αλλαγές που δεν θα σταματήσουν να μας εκπλήσσουν σε κάθε νέο βήμα.

Κεφάλαιο 3ο : Πλατφόρμες και εργαλεία υλοποίησης

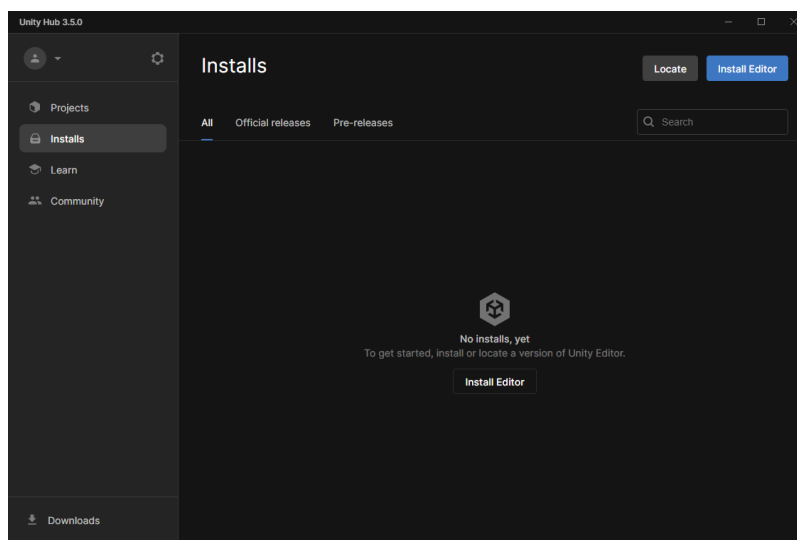
3.1 Εισαγωγή

Για να επιτευχθεί η ανάπτυξη μιας AR εφαρμογής απαιτούνται διάφορα εργαλεία και λογισμικά, ανάλογα με το ζητούμενο αποτέλεσμα μπορεί να είναι λίγα ή πιο εκτεταμένα. Στο παρόν κεφάλαιο θα αναφερθούν οι πλατφόρμες ανάπτυξης, τα πακέτα και εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας πτυχιακής εργασίας καθώς η διαδικασία για τις ρυθμίσεις τους και τα βήματα που έγιναν.

Αναφορικά έγινε χρήση των Blender [29] για την δημιουργία και επεξεργασία των 3D μοντέλων και Unity [30] σαν περιβάλλον ανάπτυξης της εφαρμογής. Το Visual Studio Code [31] χρησιμοποιήθηκε για τη συγγραφή και διαχείριση του κώδικα σε C# και το MS Paint για την τροποποίηση των εικόνων. Επιπλέον ανάλυση θα γίνει στις επόμενες υποενότητες.

3.2 Unity Hub και Unity

Κατά την χρονική περίοδο συγγραφής αυτής της εργασίας το Unity βρίσκεται πρώτο σε δημοτικότητα καθώς είναι δωρεάν και εύκολο στην εκμάθηση για αρχάριους χρήστες. Προσφέρει μια πλήρη γκάμα εργαλείων για ανάπτυξη, σε μια πληθώρα εφαρμογών και παιχνιδιών, ανάμεσα τους και AR, με τεράστια κοινότητα που το υποστηρίζει και βοηθάει στην εξέλιξη του διαρκώς. Είναι cross-platform, προσφέρει δηλαδή δυνατότητα για ανάπτυξη τόσο σε Desktop όσο και Android, iOS. Ακόμα ενδείκνυται και για κονσόλες όπως PlayStation αλλά και για headsets στο κομμάτι του VR. Τα στοιχεία αυτά λοιπόν συνέβαλαν στην επιλογή του, σε σύγκριση με κάποια άλλη πλατφόρμα όπως το Unreal Engine ή το Android Studio.

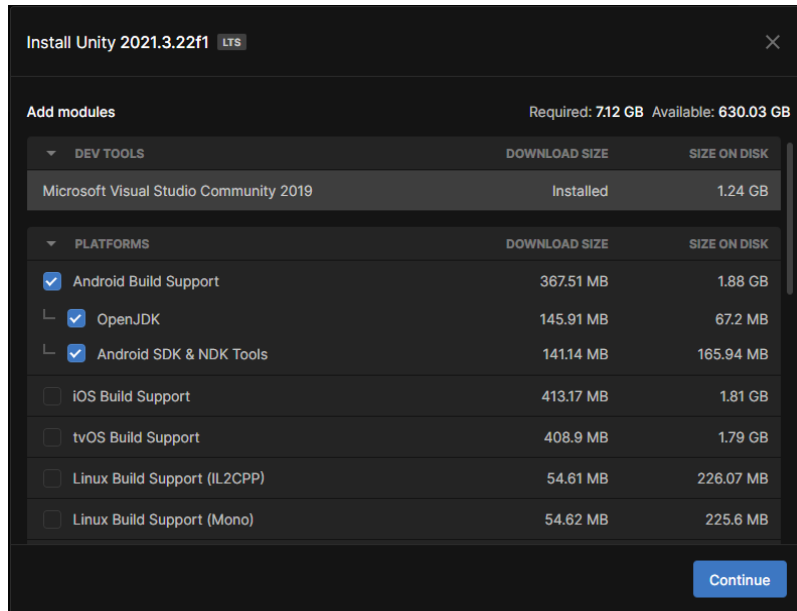


Σχήμα 3.1 : Unity Hub Version Install

Η εγκατάσταση της μηχανής Unity γίνεται μέσα από το Unity Hub, το οποίο μπορούμε να κατεβάσουμε δωρεάν μέσω της επίσημης ιστοσελίδας του. Έπειτα μπορούμε να επιλέξουμε την έκδοση που επιθυμούμε μέσω της χρήσης του γραφικού περιβάλλοντος του μενού (Σχήμα 3.1). Για την παρούσα πτυχιακή εργασία

Πλατφόρμες και εργαλεία υλοποίησης

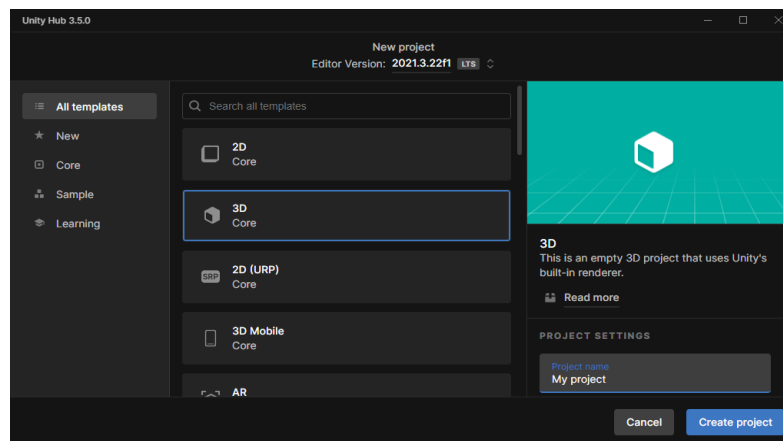
επιλέχθηκε η έκδοση 2021.3.22f1, καθώς υπήρχαν προβλήματα σε άλλες εκδόσεις με το πακέτο Vuforia, που θα αναφερθεί παρακάτω. Ταυτόχρονα με το κατέβασμα και την εγκατάσταση της έκδοσης χρειάστηκε και το Android Build Support module που περιέχει τα απαραίτητα εργαλεία για την ανάπτυξη της εφαρμογής σε Android (Σχήμα 3.2).



Σχήμα 3.2 : Εγκατάσταση Έκδοσης με Android SDK

3.2.1 Δημιουργία και Ρύθμιση Νέου Project

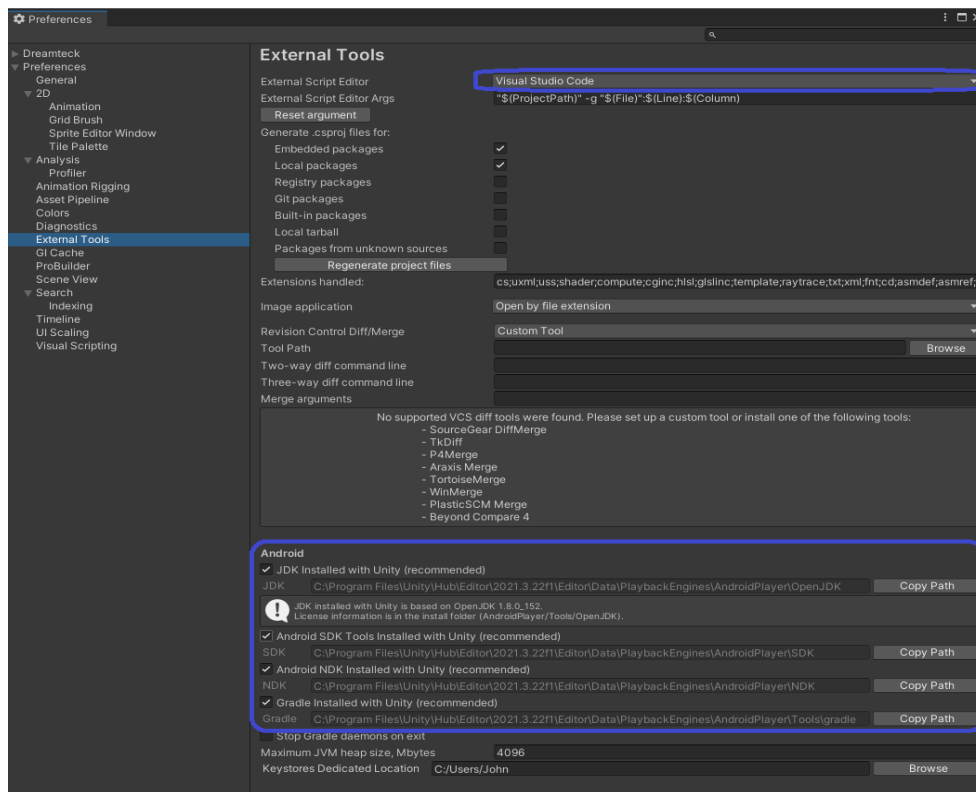
Στην συνέχεια από το tab Projects πατάμε New Project και γίνεται επιλογή του 3D Core. Δίνουμε το επιθυμητό όνομα και μετά το πάτημα του Create Project αναμένουμε ώστε να γίνουν οι αναγκαίες διεργασίες (Σχήμα 3.3).



Σχήμα 3.3 : Δημιουργία Νέου Project

Κατά την ολοκλήρωση μπορούμε να δούμε τα δυο βασικά αντικείμενα, την βασική κάμερα την οποία για την συγκεκριμένη περίπτωση δεν θα την χρειαστούμε και το Directional Light. Η συγκεκριμένη έκδοση έρχεται με προεγκατεστημένα packages όπως το device simulator. Μέσω αυτού τρέχουμε την εφαρμογή τοπικά, αλλά συμπεριφέρεται σαν να ήταν σε κάποια πραγματική Android συσκευή. Αξιοποιείται στον έλεγχο των αλλαγών κατά την διάρκεια ανάπτυξης της εφαρμογής χωρίς να χρειάζεται να κάνουμε διαρκώς build.

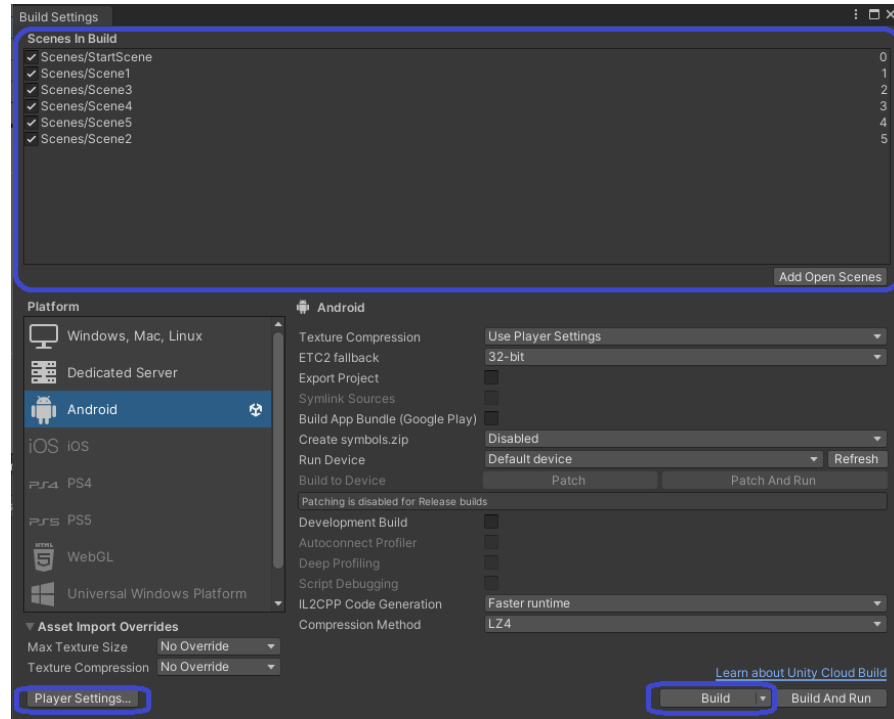
Έπειτα μπορούμε από το Edit → Preferences → External Tools να αλλάξουμε διάφορες ρυθμίσεις. Στο External Script Editor επιλέγουμε Visual Studio Code ώστε να γνωρίζει το Unity να το χρησιμοποιεί για να ανοίξει και να επεξεργαστεί τα C# scripts. Στο κάτω μέρος μπορούμε να δούμε τα αναγκαία SDK και JDK για να δουλέψει η εφαρμογή σε Android, που επιλέχθηκε να εγκατασταθούν μαζί με την Unity έκδοση. Μπορούν να τροποποιηθούν και αλλαχθούν αν υπάρχει κάποιο πρόβλημα όπως φαίνεται από το Σχήμα 3.4.



Σχήμα 3.4 : Unity Preferences

Επόμενα βήμα είναι File → Build Settings όπου επιλέγουμε το tab platforms Android και πατάμε Switch platform. Μόλις γίνει η αλλαγή το build θα εξάγεται σε APK, το οποίο μπορούμε να μεταφέρουμε στο smartphone μας. Σημειώνεται ότι όλες οι σκηνές πρέπει να έχουν προστεθεί στο Scenes In Build μέσω του κουμπιού Add Open Scenes, αλλιώς σε διαφορετική περίπτωση θα προκύψουν σφάλματα (Σχήμα 3.5).

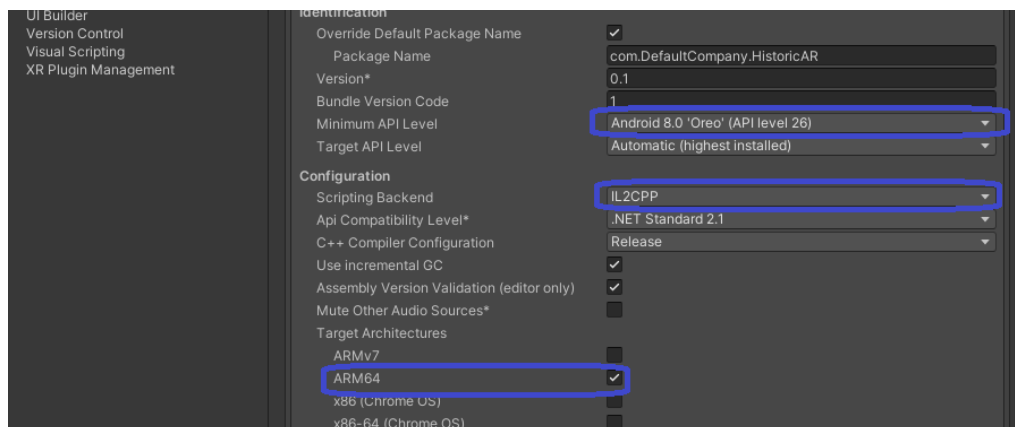
Πλατφόρμες και εργαλεία υλοποίησης



Σχήμα 3.5 : Build Settings

Τέλος, μέσω του Edit → Project Settings → Player ή πατώντας το κουμπί Player Settings αλλάζουμε τις τελευταίες ρυθμίσεις. Κάνουμε επιλογή της ελάχιστης έκδοσης το Android 8.0 Oreo ώστε, να καλύπτεται μια πολύ μεγάλη γκάμα παλαιότερων συσκευών, διατηρώντας ωστόσο τις δυνατότητες των νεότερων Android εκδόσεων.

Επιλέγουμε ως Scripting Backend το IL2CPP. Επίσης, ως Target Architectures αναγκαία είναι η επιλογή μόνο της ARM64, διότι σε διαφορετική περίπτωση λαμβάνουμε error (Σχήμα 3.6).



Σχήμα 3.6 : Project Settings ,Player Tab

3.3 Unity Package Manager

Με το Package Manager, που μπορούμε να βρούμε πηγαίνοντας Window → Package Manager, έχουμε την δυνατότητα να κατεβάσουμε διάφορα πακέτα τα οποία δεν έχουν έρθει προεγκατεστημένα με την έκδοση που έχουμε επιλέξει. Μέσα από το παράθυρο μπορούμε να δούμε ποια έχουν εγκατασταθεί, ποια έκδοση έχουν και επιπλέον να επιλέξουμε, κατεβάσουμε και εγκαταστήσουμε κάποια προεπιλεγμένα. Από την ιστοσελίδα Unity Asset Store μπορούμε να βρούμε επιπλέον πακέτα που δεν βρίσκονται στο παράθυρο ή ακόμα και αλλά customs τα οποία έχουν δημιουργηθεί από χρήστες. Στην συνέχεια θα αναφερθούν τα επιπλέον πακέτα που χρησιμοποιήθηκαν.

3.3.1 TextMeshPro

Μέσω αυτού του πακέτου το οποίο έχει ως στόχο να αντικαταστήσει τα UI Text και Text Mesh μπορούμε να γράψουμε κείμενο. Είναι ένα ισχυρό και εύχρηστο εργαλείο που χρησιμοποιεί προηγμένες τεχνικές απόδοσης κειμένου με σύνολο από shaders και έτσι πετυχαίνει βελτίωση στην οπτική ποιότητα, ενώ παράλληλα δίνει στους χρήστες απίστευτη ευελιξία όσον αφορά το στυλ και το texture του κειμένου. Παρέχει πάνω από 30 ετικέτες εμπλουτισμένου κειμένου και υποστήριξη διαφόρων Font & Sprites, Custom Styles και πολλών άλλων. Μέσα σε αυτά συμπεριλαμβάνεται και η Ελληνική Γλώσσα. Συναντήθηκαν ωστόσο προβλήματα όπου τα sprites διαφόρων γραμμάτων δεν υπήρχαν ή δεν ανιχνεύονταν. Χρειάστηκε η χειροκίνητη προσθήκη τους μέσα από το TextMeshPro → Font Asset Creator. Δίνουμε τον δεκαεξαδικό (HEX) κωδικό του γράμματος και κάνουμε generate τον Font Άτλα εμπλουτίζοντάς τον.

3.3.2 ProBuilder

Το πακέτο ProBuilder είναι ένα υβριδικό εργαλείο που προσφέρει την δυνατότητα για δημιουργία 3D σχημάτων, αντικειμένων και επιπέδων. Μπορεί να χρησιμοποιηθεί για την δημιουργία απλής γεωμετρίας, αλλά είναι ικανό για λεπτομερή επεξεργασία. Χρησιμοποιώντας το, μπορούμε γρήγορα να δημιουργήσουμε δομές και σύνθετες κατασκευές όπως δέντρα, όπλα, οχήματα. Εκμεταλλεύεται τις δυνατότητες του Unity και συνδυάζεται με αλλά παρόμοια εργαλεία όπως το Maya, οπότε ο χρήστης μπορεί να γυαλίσει επιπλέον τα ψηφιακά του μοντέλα. Θα γίνει επιπλέον αναφορά σε επόμενο κεφάλαιο όπου έγινε χρήση του.

3.3.3 Vuforia Engine

Το Vuforia Engine AR είναι ένα πολύ δημοφιλές πλέον software development kit (SDK) δηλαδή κιτ ανάπτυξης λογισμικού το οποίο προσφέρει την δυνατότητα δημιουργίας εφαρμογών επαυξημένης πραγματικότητας τόσο για φορητές συσκευές όσο και για ψηφιακά γυαλιά. Με το SDK, προστίθεται προηγμένη λειτουργικότητα υπολογιστικής όρασης στην εφαρμογή, επιτρέποντάς της να αναγνωρίζει εικόνες, αντικείμενα και χώρους. Με εύχρηστες επιλογές μπορεί να διαμορφωθεί η εφαρμογή ώστε να αλληλεπιδρά με τον πραγματικό κόσμο. Παρέχει δυνατότητα για ανάπτυξη σε Android, Studio, iOS, Unity, MagicLeapOS καθώς και UWP. Προσφέρει παρακολούθηση μιας ποικιλίας αντικειμένων και χώρων που κατηγοριοποιούνται σε Εικόνες, Αντικείμενα και Περιβάλλοντα [33-34] .

Μπορούμε να το κατεβάσουμε δωρεάν από την επίσημη ιστοσελίδα αφού πρώτα δημιουργήσουμε έναν λογαριασμό. Για την παρούσα εργασία, έγινε λήψη της έκδοσης Vuforia-package-10.15.4, για την πλατφόρμα Unity. Η εγκατάσταση μπορεί να γίνει είτε μέσω του Package manager, είτε απλά με ένα drag n drop στο Asset tab του project, και πατώντας το κουμπί import.

3.3.4 Dreamteck Splines

Ένα ακόμα πακέτο που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής είναι το πακέτο Dreamteck Splines το οποίο μπορούμε να προμηθευτούμε δωρεάν από το Unity Asset Store. Μας δίνει την δυνατότητα μέσα από ειδικά components, ανάμεσα και σε πολλές άλλες χρήσεις, να σχηματίσουμε μονοπάτια και γραμμές τα οποία μπορούν να ακολουθηθούν από άλλα αντικείμενα στον χώρο [35].

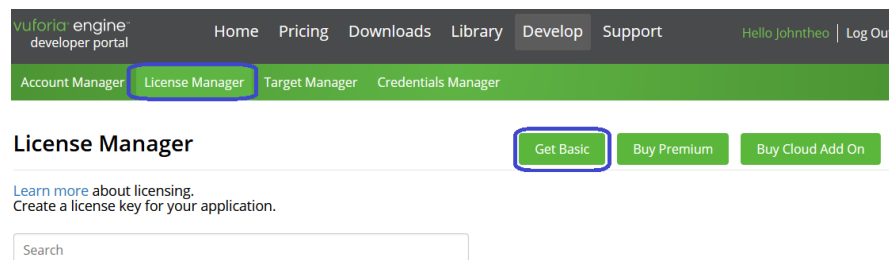
3.3.5 Lean Touch

Τέλος, ένα ακόμα πολύ δημοφιλές δωρεάν third party package το οποίο μπορούμε να προμηθευτούμε επίσης από το Unity Asset Store και προσφέρει πολλές δυνατότητες σχετικές με το input manager και input control σε όλες τις πλατφόρμες είναι το Lean Touch. Μέσω των διαφόρων scripts του μπορούμε να χειριστούμε μετακινώντας ή αλλάζοντας κλίμακα τα τρισδιάστατα μοντέλα μας στον χώρο [36].

3.4 Vuforia Engine Setup

Μετά την εισαγωγή του πακέτου Vuforia στο Unity μπορούμε να διαγράψουμε την main camera που δημιουργήθηκε μαζί με την πρώτη σκηνή και να πάρουμε στην νέα λίστα για εισαγωγή, που δημιουργήθηκε στο MenuBar, από το GameObject → Vuforia → ARCamera. Το ίδιο μπορεί να γίνει και μέσω του παράθυρου Hierarchy κάνοντας δεξί κλικ. Έπειτα αλλάζουμε το tag στον inspector της ARCamera σε MainCamera.

3.4.1 Vuforia Engine Key

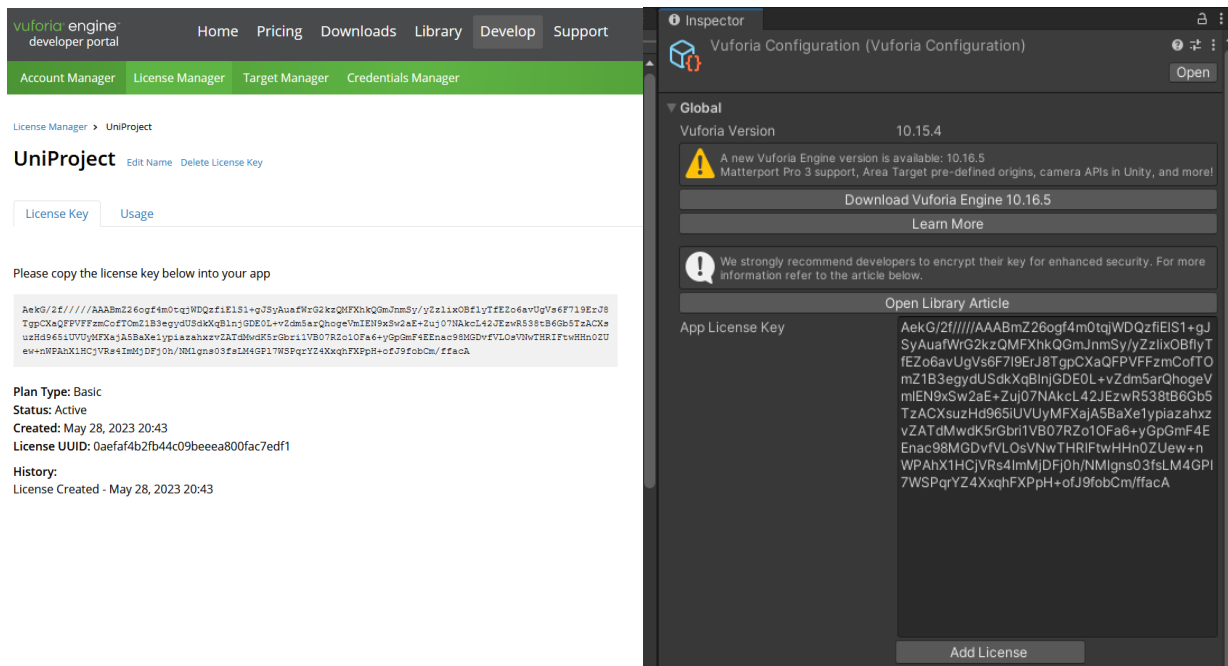


Σχήμα 3.7 : Vuforia Create Key

Επόμενο βήμα για να λειτουργήσει είναι να λάβουμε άδεια χρήσης. Στην ιστοσελίδα Developer πηγαίνουμε στο tab License Manager όπου πατάμε το κουμπί Get Basic.(Σχήμα 3.7)

Επιλέγουμε το όνομα που θέλουμε και αφού αποδεχτούμε τους όρους χρήσης ολοκληρώνουμε την διαδικασία. Μόλις δημιουργηθεί το νέο entry μπορούμε να το δούμε σαν νέα γραμμή ακριβώς από κάτω. Επιλέγοντας την, μας ανοίγει νέα σελίδα όπου και μπορούμε να βρούμε το license key και να το αντιγράψουμε.

Επιστρέφοντας πίσω στο project του Unity στον inspector της ARCamera επικολλούμε το κλειδί μέσω του Open Vuforia Engine Configuration → VuforiaConfiguration → Global → App License Key→ Add License (Σχήμα 3.8)

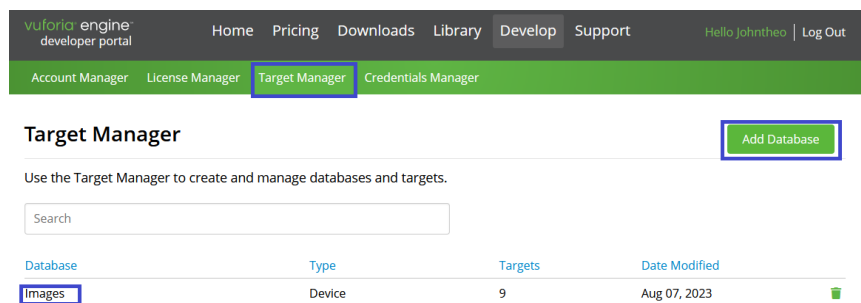


Σχήμα 3.8 : Vuforia Create Key and Paste Key

3.4.2 Vuforia Engine Database

Η λειτουργία της επαύξησης μοντέλων στον χώρο μέσω του Vuforia επιτυγχάνεται αναγνωρίζοντας κάποιο αντικείμενο στον χώρο που βλέπει η κάμερα. Για να συμβεί αυτό θα πρέπει να υπάρχει κάποια βάση δεδομένων που θα περιέχει τις πληροφορίες των στόχων, ώστε να μπορούν να συγκριθούν και να ταυτοποιηθούν. Η βάση αυτή δημιουργείται ξανά μέσω της επίσημης ιστοσελίδας.

Πρώτο βήμα είναι να πάμε από την σελίδα Developer στο tab Target Manager. Εκεί επιλέγουμε το κουμπί Add database και επιλέγουμε τον τύπο που θέλουμε. Για την εργασία χρησιμοποιήθηκε η επιλογή Device. Αφού την ονοματίσουμε επιλέγουμε το create και περιμένουμε να εμφανιστεί (Σχήμα 3.9).



Σχήμα 3.9 : Δημιουργία Βάσης

Για να προσθέσουμε στόχους επιλέγουμε την βάση που δημιουργήσαμε και στην νέα σελίδα πατάμε Add Target. Μας δίνεται μια πληθώρα από τύπους αντικειμένων που μπορούμε να επιλέξουμε, από απλή εικόνα μέχρι και αντικείμενα, κάθε ένα από τα οποία καλύπτει και μια διαφορετική ανάγκη. Διαλέγουμε την επιλογή εικόνα(Image) και αφού βρούμε την επιθυμητή από το file system, δίνουμε πλάτος και ονομασία. Την προσθέτουμε επιλέγοντας το Add. Ύστερα από μερικά δευτερόλεπτα η εικόνα μας ανεβαίνει στην βάση και τροποποιείται ώστε να λειτουργήσει ως στόχος (Σχήμα 3.10).

Η διαδικασία αυτή ακολουθήθηκε αρκετές φορές μέχρι να συμπεριληφθούν όλες οι απαραίτητες εικόνες για τα υλοποιημένα κεφάλαια.

Τέλος, μπορούμε να κατεβάσουμε την βάση μας, πατώντας το κουμπί Download Database(All) και να την κάνουμε import στο Unity project.

Add Target

Type:

Image Multi Cylinder Object

File:

Choose File

.jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

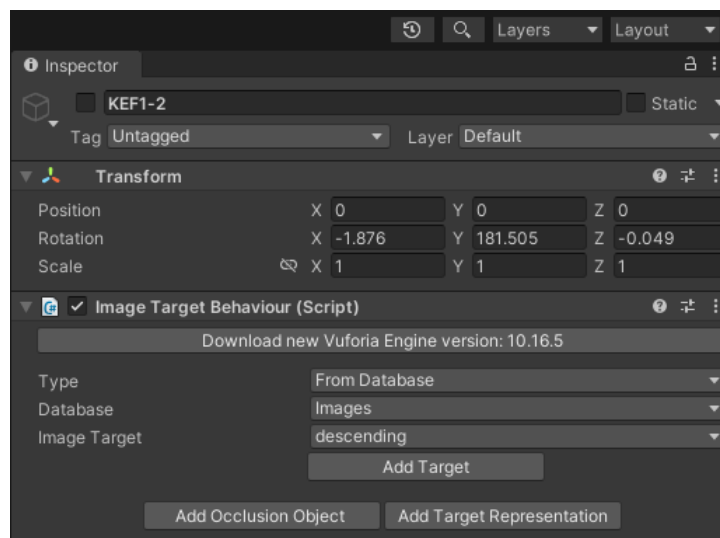
Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Σχήμα 3.10 : Προσθήκη Εικόνων στην Βάση

3.4.3 Vuforia Image Target

Έχοντας καλύψει την ανάγκη της δημιουργίας βάσης, επόμενη ενέργεια είναι να δημιουργήσουμε ImageTarget που θα περιέχει τα δεδομένα της βάσης υπό μορφή εικόνων, τα οποία θα εντοπίσει η AR κάμερα.

Δημιουργούμε ένα νέο GameObject μέσω του GameObject → Vuforia Engine → ImageTarget και από τον inspector αλλάζουμε τις ρυθμίσεις στο Image Target Behavior Script. Επιλέγουμε σαν Type την επιλογή FromDatabase και στην επιλογή Database, την βάση που έχουμε κάνει import. Τέλος, επιλέγουμε την εικόνα που θέλουμε να οριστεί ως στόχος (Σχήμα 3.11). Η διαδικασία αυτή επαναλαμβάνεται κάθε φορά που θέλουμε να δημιουργήσουμε και να ορίσουμε έναν νέο στόχο.

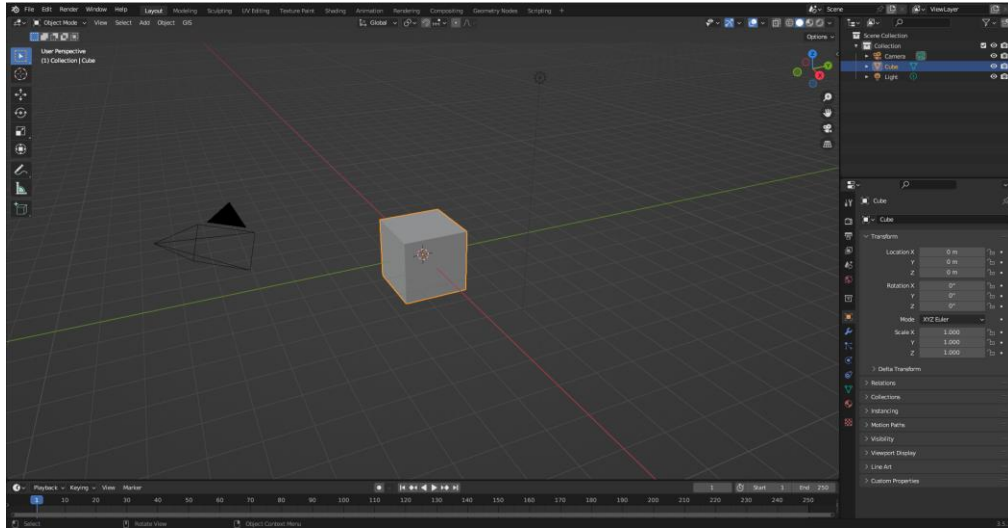


Σχήμα 3.11: Παράδειγμα Επιλογής Εικόνας σαν Image Target

3.5 Blender

Το Blender (Σχήμα 3.12) είναι δωρεάν, ανοιχτού κώδικα και αποτελεί το πιο δημοφιλές πρόγραμμα για την δημιουργία τρισδιάστατων ψηφιακών στοιχείων, καθώς παρέχει και υποστηρίζει ολόκληρο το 3D pipeline από την αρχή της μοντελοποίησης, rigging, animation, προσομοίωση, rendering, επεξεργασία βίντεο και πολλά άλλα.[37]

Χρησιμοποιήθηκε η έκδοση Blender 3.5 για την δημιουργία, τροποποίηση και επεξεργασία των 3D μοντέλων και εξαγωγή τους σε τύπο FBX, το οποίο μπορεί να διαχειριστεί το Unity.



Σχήμα 3.12 : Αρχική Σκηνή Blender

3.6 Επίλογος

Σε αυτό το κεφάλαιο παρατέθηκε η διαδικασία για την ολοκλήρωση των βασικών προγραμμάτων και εργαλείων που αξιοποιήθηκαν. Μετά από αυτές τις ενέργειες μπορούμε να αρχίσουμε να αναπτύσσουμε το ουσιαστικό κομμάτι της εφαρμογής που είναι οι σκηνές.

Κεφάλαιο 4ο: Εφαρμογή και Εισαγωγική Σκηνή

4.1 Εισαγωγή

Στο παρόν κεφάλαιο θα περιγραφεί η βασική ιδέα για την υλοποίηση της εφαρμογής και δημιουργία της πρώτης εισαγωγικής σκηνής. Υπάρχει αναφορά στις υπόλοιπες σκηνές που απαρτίζουν το σύνολο της εφαρμογής οι οποίες θα αναλυθούν στα επόμενα κεφάλαια.

4.2 Εικονίδιο

Το πρώτο πράγμα που βλέπει κάποιος χρήστης πριν ανοίξει και περιέλθει σε κάποια εφαρμογή είναι το εικονίδιο της. Έπειτα από αναζήτηση στο διαδίκτυο βρέθηκε ένα επιθυμητό δωρεάν εικονίδιο το οποίο τροποποιήθηκε με την χρήση του 3D Paint και έπειτα τοποθετήθηκε το όνομα της εφαρμογής σε σημείο που μπορεί να αναγνωστεί και να μην κρύβει την εικόνα. Το εικονίδιο φαίνεται στο σχήμα 4.1.



Σχήμα 4.1 : Εικονίδιο Εφαρμογής

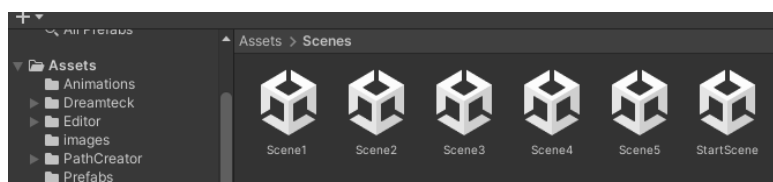
Για την χρήση του σαν ορισμένο εικονίδιο της εφαρμογής, χρειάστηκε να δημιουργηθεί ένας φάκελος Images στα Assets του Unity όπου και θα χρησιμοποιηθεί για την τοποθέτηση και των επόμενων εικόνων. Αφού προστέθηκε ακολουθούμε τα βήματα File → BuildSettings → PlayerSettings → Player → DefaultIcon και το επιλέγουμε από την λίστα.

4.3 Ιστορία Στα Αρχαία Χρόνια

Η δομή του εκπαιδευτικού υλικού δημιουργήθηκε με βάση το σχολικό βιβλίο εκμάθησης Ιστορία της Δ' δημοτικού, το οποίο και προμηθεύτηκε σε ηλεκτρονική μορφή από την ιστοσελίδα του Υπουργείου Παιδείας, Θρησκευμάτων και Αθλητισμού[38].

Έγινε διαχωρισμός σε τέσσερις ενότητες με βάση την Ιστορική χρονολογική περίοδο. Η τελευταία χωρίστηκε σε δυο σκηνές λόγω του εκτενούς αριθμού κεφαλαίων. Προέκυψαν πέντε σκηνές (Σχήμα 4.2). Οι ενότητες αναφορικά είναι:

1. ΓΕΩΜΕΤΡΙΚΑ ΧΡΟΝΙΑ
2. ΑΡΧΑΪΚΑ ΧΡΟΝΙΑ
3. ΚΛΑΣΙΚΑ ΧΡΟΝΙΑ(ΜΑΚΕΔΟΝΙΑ)
4. ΕΛΛΗΝΙΣΤΙΚΑ ΧΡΟΝΙΑ
 - 4.1. ΟΙ ΔΙΑΔΟΧΟΙ ΤΟΥ Μ. ΑΛΕΞΑΝΔΡΟΥ
 - 4.2. ΡΩΜΗ ΚΑΙ ΕΛΛΑΔΑ



Σχήμα 4.2 : Σκηνές της Εφαρμογής

Το κείμενο κάθε κεφαλαίου αποτελεί κατά βάση την ύλη στην οποία βασίστηκε η συγγραφή των κειμένων που παρουσιάζεται. Έγινε περίληψη και σύμπτυξη των σημαντικότερων πληροφοριών που αναφέρονται και όπου κρίθηκε χρήσιμο συμπεριλήφθηκαν και πληροφορίες από τα παραθέματα. Η επιλογή των τρισδιάστατων μοντέλων, και η δημιουργία των animation, έγινε ώστε να αντικατοπτρίζονται τα κυριότερα στοιχεία του κειμένου. Μέσω αυτών επιδιώκεται, ο μαθητής που χρησιμοποιεί την εφαρμογή να έχει μια πολύπλευρη εμπειρία εκμάθησης, καθώς δεν μαθαίνει απλώς διαβάζοντας, αλλά βλέπει να γεγονότα να συμβαίνουν μπροστά του.

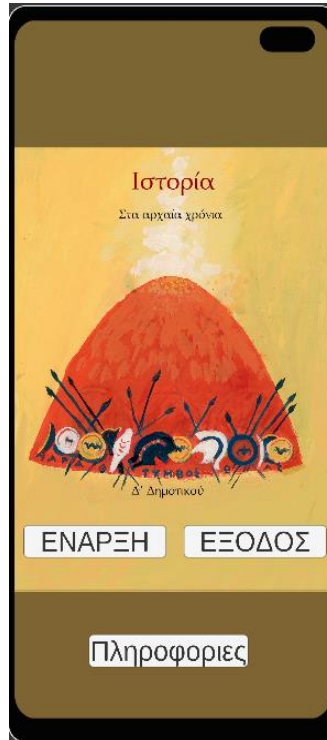
4.3 Η Εφαρμογή

Στην εφαρμογή, ο χρήστης θα έχει την δυνατότητα μετάβασης στις διάφορες σκηνές, μέσα από ένα μενού επιλογής επιπέδων. Στην αρχή κάθε σκηνής θα υπάρχει ένα πάνελ με κείμενο, όπου θα αναφέρουμε την ύλη του βιβλίου στην οποία είναι βασισμένη η σκηνή. Επίσης στο ίδιο κείμενο, θα ενημερώνουμε τον χρήστη για την σελίδα όπου βρίσκεται η εικόνα-στόχος, που θα πρέπει να σκανάρει με την κάμερα της συσκευής, ώστε να εμφανιστούν τα τρισδιάστατα μοντέλα. Έπειτα από την ολοκλήρωση του μαθήματος-σκηνής θα κάνουμε υπενθύμιση των βασικών σημείων που είδε. Αυτό θα γίνεται μέσα από κείμενο σε ένα νέο πάνελ, και θα του δίνεται η δυνατότητα να προχωρήσει στην επομένη σκηνή. Στην περίπτωση της τελευταίας σκηνής θα επιστρέψει στην εισαγωγική σκηνή (Σχήμα 4.3).

Έγινε προσπάθεια ο κώδικας που θα γραφτεί να μπορεί να αξιοποιηθεί σε διάφορες σκηνές και να τροποποιηθεί μόνο σε περιπτώσεις που δεν μπορεί να δημιουργηθεί κάποιο παραπλήσιο function. Ακόμα, έγινε προσπάθεια ώστε κάθε script να επιτελεί μια ξεχωριστή λειτουργία ώστε να είναι ξεκάθαρη η τοποθέτηση και χρήση του καθενός από αυτά. Τέλος, έγινε επαναχρησιμοποίηση κάποιων 3D αντικειμένων, ειδικά στην περίπτωση των χαρτών, καθώς τα περισσότερα ιστορικά γεγονότα που καλύπτονται συνέβησαν σε κοντινές γεωγραφικά περιοχές.

4.4 Εισαγωγική Σκηνή

Μετά το άνοιγμα της εφαρμογής η πρώτη σκηνή που θα αντικρίσει ο χρήστης είναι η εισαγωγική σκηνή (Σχήμα 4.3). Μπορεί να παρατηρήσει ότι σαν φόντο είχε το εξώφυλλο του σχολικού βιβλίου, συνοδευόμενο από το UI που είναι τρία κουμπιά.



Σχήμα 4.3 : Εισαγωγική Σκηνή

Για την δημιουργία του αλλάζουμε το όνομα της default σκηνής σε StartScene. Έπειτα, μέσω του παράθυρου Hierarchy κάνουμε δεξί κλικ UI → RAW Image. Αυτό έχει ως αποτέλεσμα την δημιουργία ενός GameObject Canvas το οποίο έχει ένα child που είναι η RAWImage που θέλουμε να φαίνεται. Μέσω του inspector αλλάζουμε το όνομα σε StartUpImage και επιλέγουμε την εικόνα εξώφυλλο του βιβλίου.

Για τις πάνω και κάτω περιοχές της οθόνης που δεν καλύπτονται από την εικόνα, δημιουργούμε δυο αντικείμενα Image UI, τα οποία τοποθετήσαμε στις αντίστοιχες περιοχές. Μέσω του property Color Inspector αλλάζουμε από το RGBA το χρώμα σε ανοιχτό καφέ. Για να επιτευχθεί το ημι-διαφανές εφέ επιλέχτηκε για το Alpha value το 100.

Τέλος, προστέθηκαν τρία κουμπιά μέσω του UI → Button τα οποία εκτελούν τις χρήσεις εξόδου από την εφαρμογή, εμφάνιση πληροφοριών και έναρξης. Η εμφάνιση των λέξεων στα κουμπιά έγινε μέσω του TextMeshPro και χρήσης της Ελληνικής γλώσσας με Font LiberationSans. Το ίδιο font αξιοποιήθηκε για το κείμενο που εμφανίζεται σε όλα τα κουμπιά καθώς και τα Text(TMP). Για να είναι πιο εμφανές χρησιμοποιήθηκε η επιλογή Outline με τιμή 0.211 thickness.

Για την λειτουργικότητα των κουμπιών δημιουργήθηκαν δυο script από το Create → C# Script και τοποθετήθηκαν μέσα στον φάκελο Scripts όπου και θα μπουν μελλοντικά και όλα τα επόμενα. Το επιθυμητό κάθε φορά script προστίθεται ως component του GameObject, και στην συγκεκριμένη περίπτωση επειδή είναι κουμπί καλείται από την προϋπάρχουσα μέθοδο OnClick().

Η έξοδος από την εφαρμογή επιτυγχάνεται με την προϋπάρχουσα μέθοδο Application.Quit() που καλείται από την QuitTheGame() που είναι τύπου void.

Με τα υπόλοιπα δυο κουμπιά θα αξιοποιήσουμε την μέθοδο SetActive() ώστε να εμφανίζουμε και να εξαφανίζουμε αντικείμενα από το περιβάλλον. Αρχικά, προσθέτουμε δυο νέα panel από το UI → Panel στο υπάρχον Canvas. Το πρώτο, με όνομα PanelWithLevelButtons, θα περιέχει τα κουμπιά για το μενού αλλαγής σκηνών, και το άλλο PanelInfoText, κείμενα για τις πληροφορίες. Θέλουμε το πρώτο να είναι εντελώς αδιαφανές, οπότε αλλάζουμε το Alpha value σε 255. Στην συνέχεια δημιουργούμε το script ToggleONOFF (Σχήμα 4.4), το οποίο θα ενεργοποιεί ή θα απενεργοποιεί τα GameObject του Hierarchy.

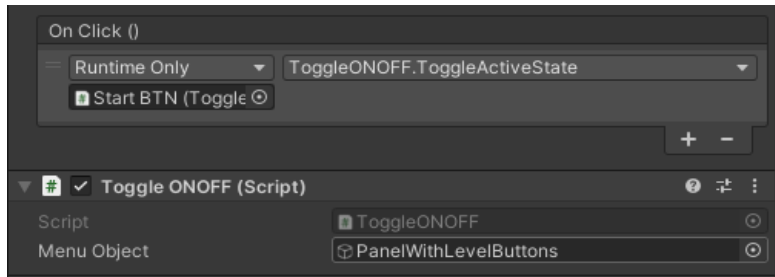
```
public void ToggleActiveState()
{
    bool currentState = menuObject.activeSelf;
    Debug.Log(currentState);
    menuObject.SetActive(!currentState);
}
```

Σχήμα 4.4 :Κώδικας Αλλαγής Κατάστασης Αντικειμένων

Για να γνωρίζει ο κώδικας ποιο αντικείμενο πρέπει να ελέγξει και να τροποποιήσει, δημιουργούμε μια μεταβλητή τύπου GameObject στην οποία περνάμε το επιθυμητό στοιχείο μέσα από τον inspector. Για να φανεί το πεδίο πρέπει να υπάρχει στον κώδικα το [SerializeField] (Σχήμα 4.5) ή να είναι τύπου public.

```
public class ToggleONOFF : MonoBehaviour
{
    [SerializeField]
    public GameObject menuObject;
```

Σχήμα 4.5 : Κώδικας για SerializeField

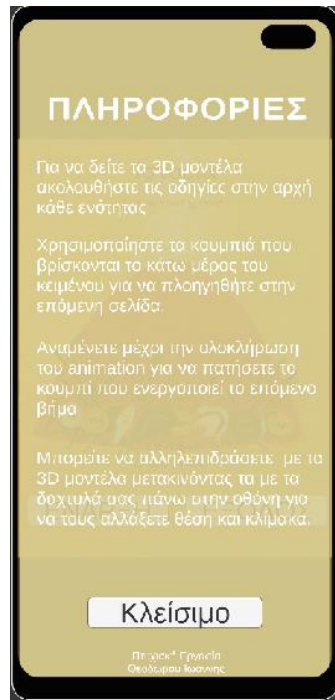


Σχήμα 4.6: OnClick() ,Custom Inspector Field

Προσθέτουμε το script ως component στο κουμπί “ Έναρξη” με παράμετρο το PanelWithLevelButtons. Έπειτα, το τοποθετούμε στην μέθοδο OnClick() του κουμπιού (Σχήμα 4.6). Έτσι, κάθε φορά που πατιέται το κουμπί, η κατάσταση του αντικειμένου αλλάζει από active (εμφανίζεται), σε not active(αποκρύπτεται από την σκηνή) και το αντίστροφο. Ανάλογα, κάνουμε και για το PanelInfoText με τις πληροφορίες στο κουμπί “ Πληροφορίες”.

4.4.1 Πληροφορίες

Πατώντας το κουμπί “Πληροφορίες” ενεργοποιείται το panel (Σχήμα 4.7) στο οποίο εμπεριέχονται τα TMP (TextMeshPro) με τα κείμενα τους.



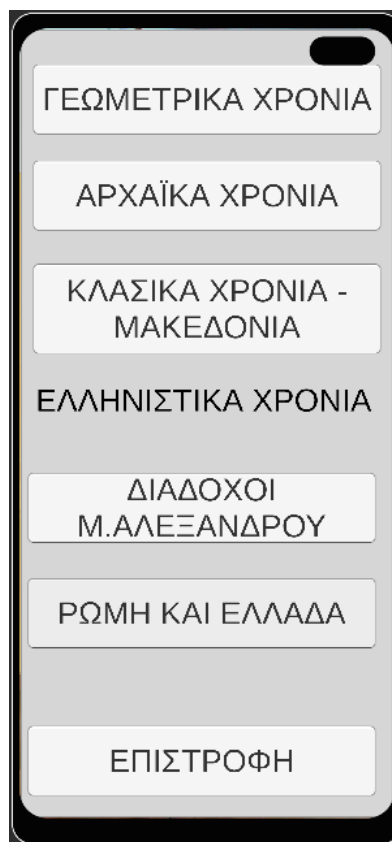
Σχήμα 4.7 : Info Panel

Για την απόκρυψη του πάνελ υπάρχει, στο κάτω μέρος του ένα κουμπί που λειτουργεί ανάλογα με τα προαναφερθέντα. Ο χρωματισμός του έγινε από το property Color χρησιμοποιώντας RGBA τιμές με Alpha value 100.

4.4.2 Level Select Panel

Με το πάτημα του κουμπιού “Έναρξη” εμφανίζεται το μενού με τα κουμπιά, που φορτώνουν την κάθε μια σκηνή της εφαρμογής (Σχήμα 4.8). Δημιουργήθηκε αντίστοιχα με το πάνελ για τις πληροφορίες, με την διαφορά ότι αυτή την φορά έχουμε επιπλέον script, το οποίο είναι υπεύθυνο για την φόρτωση και άνοιγμα της επιλεγμένης σκηνής (Σχήμα 4.10). Χρησιμοποιήθηκαν δυο διαφορετικές τεχνικές, η μια για την πρώτη σκηνή και η δεύτερη για τις υπόλοιπες.

Στην πρώτη περίπτωση δημιουργήθηκε ένα άδειο GameObject στο οποίο προστέθηκε το script και καλείται από την μέθοδο OnClick() του κουμπιού. Η δεύτερη είναι να προστεθεί κατευθείαν στο υπάρχον Button το script, σαν ένα ακόμα component. Δεν εντοπίστηκε κάποια ουσιαστική διαφορά όποτε προτιμήθηκε ο δεύτερος τρόπος στις επόμενες φορές.



Σχήμα 4.8 : Level Select Panel

Για να μπορέσει να εκτελεστεί ο κώδικας, ως προαπαιτούμενο είναι να υπάρχει στο script η βιβλιοθήκη SceneManager (Σχήμα 4.9), που μπορούμε να κάνουμε import. Τέλος, φαίνεται η ιεραρχία όλων των αντικειμένων της σκηνής (Σχήμα 4.11).

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

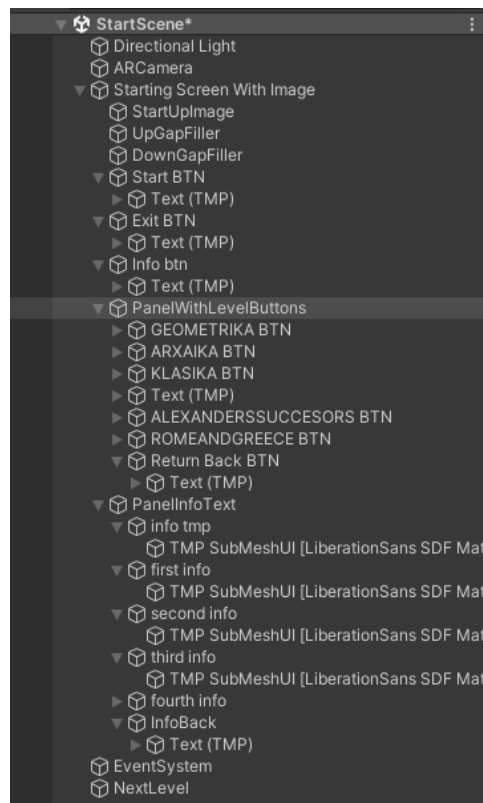
public class LevelSelectButton : MonoBehaviour

[SerializeField]
public string sceneName;
```

Σχήμα 4.9 : Scene Manager

```
public void changeScene()
{
    SceneManager.LoadScene(sceneName);
}
```

Σχήμα 4.10 : Κώδικας Αλλαγή Σκηνής



Σχήμα 4.11 : Ιεραρχία Αντικειμένων StartScene

4.5 Επίλογος

Σε αυτό το κεφάλαιο είδαμε την υλοποίηση της αρχικής εισαγωγικής σκηνής. Επίσης, παρουσιάστηκε ο τρόπος χειρισμού των αντικειμένων στην εφαρμογή, μέσω της δυνατότητας αλλαγής της ενεργούς κατάστασης που παρέχει το Unity, από την μέθοδο `SetActive()`.

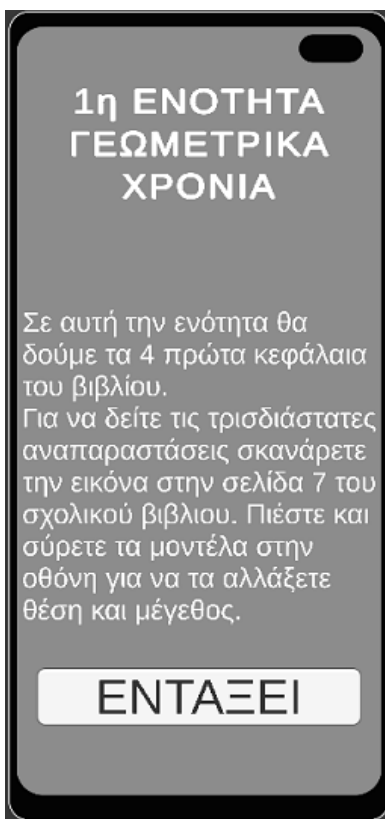
Κεφάλαιο 5ο: Γεωμετρικά Χρόνια

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα δούμε την υλοποίηση της πρώτης ενότητας του σχολικού βιβλίου, η οποία θα έχει πλέον και ουσιαστική χρήση της AR τεχνολογίας. Με την ARCamera, που προσφέρει η βιβλιοθήκη Vuforia, χρησιμοποιώντας εικόνες του βιβλίου ως στόχους για επαύξηση, θα ζωντανέψουμε τα πρώτα κεφάλαια μέσα από ένα μικρό animation και έπειτα θα δείξουμε τρισδιάστατα μοντέλα. Βάση αυτής της ιδέας για την πρώτη σκηνή θα δημιουργηθούν και οι επόμενες, που θα αναπτυχθούν στα επόμενα κεφάλαια. Μετά την δημιουργία κάθε σκηνής επαναλαμβάνεται η διαδικασία που αναφέρθηκε στο 3.4 Vuforia Engine Setup για την αντικατάσταση της default κάμερας από την ARCamera.

5.2 Panel Εισαγωγής

Μόλις φορτωθεί η σκηνή ο χρήστης αντικρίζει ένα πάνελ καλωσορίσματος (Σχήμα 5.1) το οποίο είναι ημιδιαφανές. Σε αυτό αναφέρουμε μερικές πληροφορίες και επιπλέον, την σελίδα του βιβλίου που βρίσκεται η εικόνα για να την σκανάρει με την κάμερα. Στο κάτω μέρος του πάνελ υπάρχει κουμπί που το απενεργοποιεί. Ταυτόχρονα ενεργοποιούνται ο Canvas, ο στόχος για επαύξηση, καθώς και το βασικό αντικείμενο που καλύπτει τα δύο πρώτα κεφάλαια της ενότητας.



Σχήμα 5.1 : Intro Panel

5.3 Δημιουργία Περιβάλλοντος

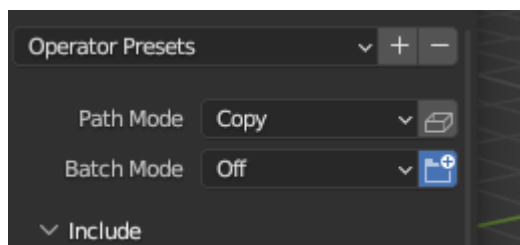
Για να γνωρίζει η ARCamera ότι θέλουμε να επαυξηθεί κάποιος στόχος και όχι απλά ένα σημείο του περιβάλλοντος που φαίνεται στην οθόνη χρειάζεται να εισάγουμε ένα ImageTarget (Σχήμα 5.2). Αυτό μπορούμε να το κάνουμε ακολουθώντας τα βήματα που αναφέρθηκαν στο 3.4.3 Vuforia Image Target, και είναι μια διαδικασία που θα επαναληφθεί αρκετές φορές κατά την διάρκεια ανάπτυξης της εφαρμογής.



Σχήμα 5.2 : Εικόνα Πρώτου Image Target

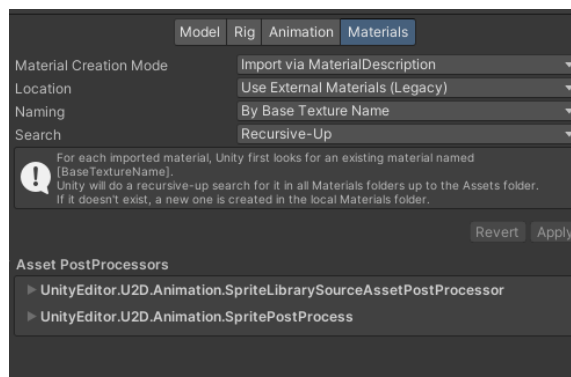
Για τον χάρτη της Ελλάδας χρησιμοποιήθηκε δωρεάν μοντέλο που βρέθηκε από το Sketchfab[40] και επεξεργάστηκε στο Blender ώστε να μπορεί να εισαχθεί στο εικονικό περιβάλλον του Unity. Μετά την επεξεργασία του, κατά την εξαγωγή του σε τύπο FBX χρησιμοποιούμε την επιλογή Copy για το Path Mode (Σχήμα 5.3) ώστε να συμπεριληφθούν μαζί και τα materials, φωτογραφίες, normals κ.α., τα οποία θα χρειαστεί να μεταφέρουμε στο Unity για τα εμφανιστούν τα textures των μοντέλων.

Το μοντέλο εισάγεται στον φάκελο Assets του Unity. Αν δεν γίνει αυτόματα η εξαγωγή και δημιουργία των materials, εάν υπάρχουν στο μοντέλο, μπορούμε να τα δοκιμάσουμε να τα εξάγουμε χειροκίνητα. Επιλέγουμε το επιθυμητό μοντέλο και έπειτα, Inspector → Materials. Δοκιμάζουμε τις επιλογές από το Location και Naming (Σχήμα 5.4) και κάνουμε click το Apply.



Σχήμα 5.3 : Blender Export Path Mode

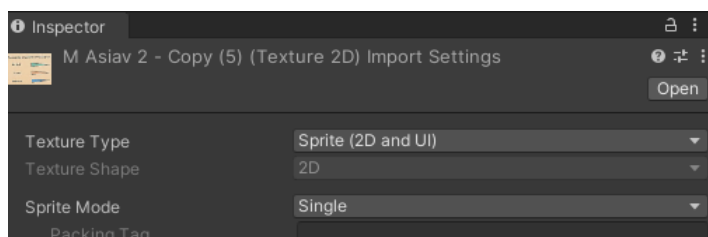
Ο χάρτης χρησιμεύει τόσο ως μοντέλο επαύξησης για τον γεωγραφικό χώρο της Ελλάδας και της Μ. Ασίας αλλά και ως βάση ParentObject κάτω από το οποίο θα εισαχθούν όλα τα υπόλοιπα αντικείμενα για τα δυο πρώτα κεφάλαια. Το αντικείμενο τοποθετείται ως ChildObject του ImageTarget στην ιεραρχία των αντικειμένων.



Σχήμα 5.4 : Unity Material Extraction

Με στόχο την καλύτερη γεωγραφική αντίληψη των περιοχών μπορούμε να τοποθετήσουμε κείμενο πάνω από το 3D μοντέλα μας, μέσω του Hierarchy, κάνοντας δεξί κλικ, 3D Object → Text-TextMeshPro. Συμβουλευόμενοι τον χάρτη, τοποθετούμε τις ονομασίες των κυριότερων περιοχών και πόλεων που αναφέρονται στα δύο πρώτα κεφάλαια. Σαν ιεραρχία τοποθετηθήκαν κάτω από το αντικείμενο του χάρτη ως ChildObjects, ώστε να μπορούν να μεταφέρονται μαζί στον χώρο. Ακολουθώντας τον χρωματικό κώδικα του βιβλίου, μέσω από τις επιλογές των attributes που μας δίνει το component TextMeshPro –Text, έγινε χρωματισμός των γραμμάτων ώστε να αντικατοπτρίζουν αυτών του βιβλίου.

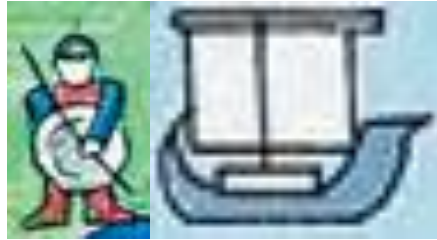
Για την βέλτιστη κατανόηση των Ελληνικών φύλων που αντιπροσωπεύουν τα χρώματα, δημιουργήθηκε εικονίδιο με βάση το υπόμνημα του χάρτη της σελίδας 9 του σχολικού βιβλίου. Το εικονίδιο χρειάστηκε να μετατραπεί σε sprite για να μπορέσουμε να το τοποθετήσουμε στον χάρτη. Αυτό μπορεί να γίνει από τον inspector του αλλάζοντας το Texture Type σε Sprite(2D and UI)(Σχήμα 5.5).



Σχήμα 5.5 : Unity Texture Type Change

Ολοκληρώνοντας το περιβάλλον με τα τρισδιάστατα μοντέλα, χρειάζεται να προσθέσουμε τα αντικείμενα τα οποία θα μετακινηθούν σε αυτό, ακολουθώντας τις διαδρομές που αναφέρονται στο μάθημα. Τα μοντέλα δημιουργήθηκαν με βάση την εικόνα του στρατιώτη, και των χρωματιστών πλοίων(Σχήμα 5.6), από τους δύο πρώτους χάρτες του βιβλίου, και κάνοντας χρήση του εργαλείου MonsterMash [39].

Έπειτα από την δημιουργία τους, επεξεργαστήκαν στο Blender όπου έγινε χρήση των Flatten, Scrape, DrawSharp, Grab για τροποποίηση του mesh ,και το TexturePaint για αλλαγή και βελτίωση των χρωμάτων των εικόνων που χρησιμοποιήθηκαν για να δημιουργηθούν τα materials.



Σχήμα 5.6 : Εικόνες για τα 3D models

5.3.1 Animation 3D Αντικειμένων

Σε αυτό το κομμάτι, θα δούμε πως έγινε η υλοποίηση του animation για τα διάφορα μοντέλα πάνω στους χάρτες. Η διαδικασία που θα περιγράψει χωρίζεται στην δημιουργία του path, και τις παραμέτρους που τροποποιούνται στο μοντέλο που το διατρέχει.

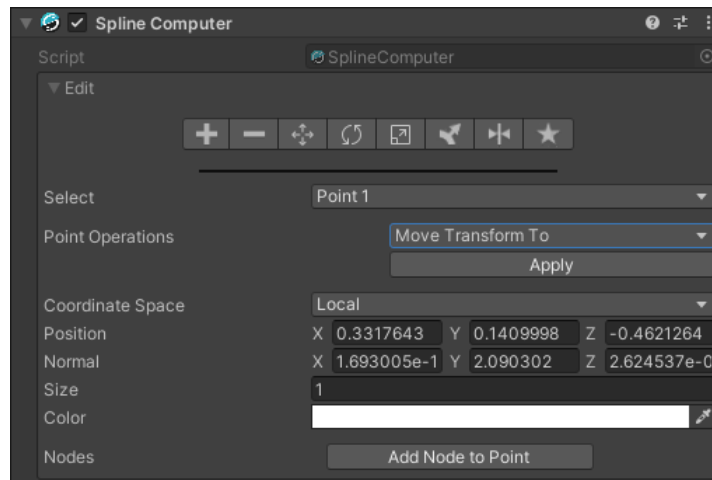
Την ίδια μεθοδολογία ακολουθούμε και για τις επόμενες σκηνές, με μερικές όμως τροποποιήσεις, ανάλογα τις ιδιαιτερότητες κάθε σκηνής.

5.3.1.1 Δημιουργία Μονοπατιού

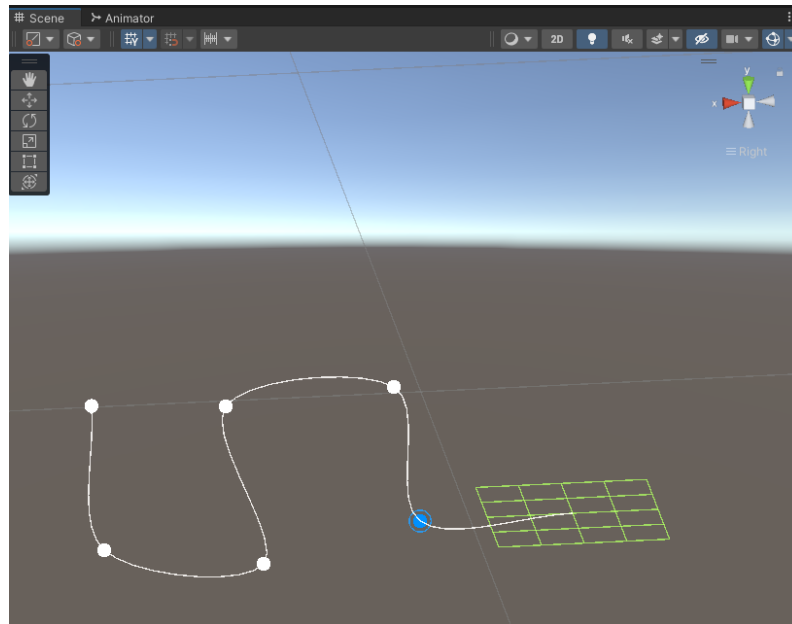
Σχετικά με την δημιουργία του animation χρησιμοποιήθηκε το package Dreamtech Splines. Δημιουργούμε ένα νέο άδειο GameObject και προσθέτουμε στα component Spline Computer (Σχήμα 5.7). Αυτό μας δίνει την δυνατότητα να τοποθετήσουμε σημεία στον χώρο τα οποία συνδέονται μεταξύ τους δημιουργώντας έτσι ένα μονοπάτι (Σχήμα 5.8). Μπορούμε μέσω του Inspector να διαχειριστούμε διάφορες ιδιότητες αυτού του component, όπως να αυξήσουμε ή να μειώσουμε τον αριθμό των σημείων, να επιλέξουμε ένα ή περισσότερα για αλλαγή συντεταγμένων, καθώς και να περιστρέψουμε τα normals. Με τον όρο normals αναφερόμαστε στο διάνυσμα το οποίο δείχνει προς την εξωτερική πλευρά της επιφάνειας 3D πλέγματος του αντικειμένου και είναι κάθετο σε αυτή.

Η μετακίνηση των σημείων στο χώρο μπορεί να γίνει και μέσα από το περιβάλλον τόσο μεμονωμένα, όσο και μαζικά, μετακινώντας τα με τον κέρσορα.

Τέλος, οι γραμμές που ενώνουν τα σημεία μπορούν να τροποποιηθούν, είτε μέσω της προεπιλογής Type(Hermite, Bezier, B-Spline, Linear), είτε στο περιβάλλον χειροκίνητα. Έτσι, οι καμπύλες που δημιουργούνται μπορούν να μετατραπούν σε πιο αμβλείες ή οξείες.

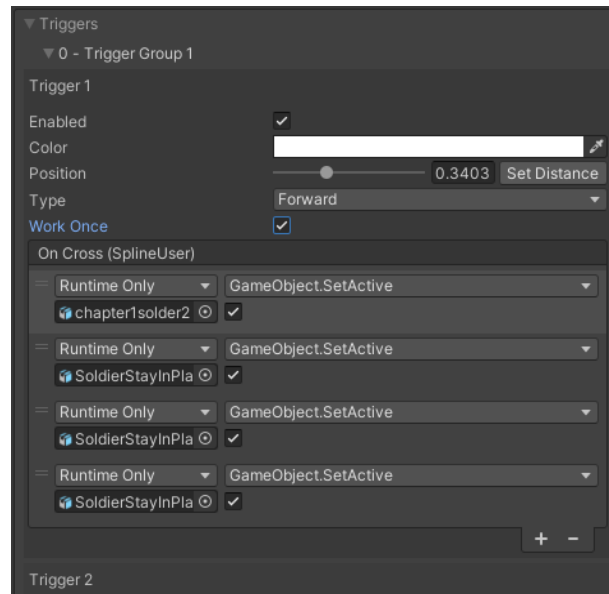


Σχήμα 5.7 : Spline Computer



Σχήμα 5.8 : Δημιουργία path

Το επόμενο από τα properties που μπορούμε να χρησιμοποιήσουμε είναι η δημιουργία triggers. Τα spline triggers είναι ένας τρόπος για να καλέσουμε συμβάντα, όταν ένα αντικείμενο φτάσει σε ένα συγκεκριμένο σημείο κατά μήκος του spline. Έτσι, μπορούμε να ενεργοποιήσουμε και να απενεργοποιήσουμε αντικείμενα, components ή και καλέσουμε scripts. Επιπλέον μας δίνονται και άλλες επιλογές όπως η συνεχής ή μεμονωμένη ενεργοποίηση καθώς και για μόνο προς την μια ή και τις δύο κατευθύνσεις κίνησης (Σχήμα 5.9).



Σχήμα 5.9 : Παράδειγμα Trigger Properties

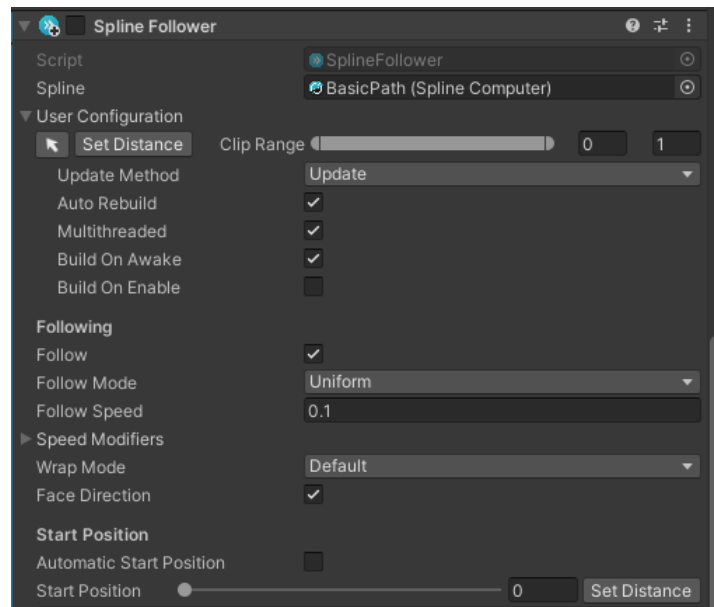
Χρησιμοποιήθηκαν για την εμφάνιση των 3D μοντέλων στον χώρο, όταν το αντικείμενο που εκτελεί το animation περάσει από συγκεκριμένα σημεία(Σχήμα 5.9). Έτσι, φαίνονται οπτικά τα γεγονότα που εξιστορούνται στο κείμενο. Μέσω trigger ενεργοποιούμε και ένα δεύτερο μικρότερο μονοπάτι, στο οποίο θα κινηθεί ένας διαφορετικός στρατιώτης.

Επίσης, δημιουργήθηκαν τρία ακόμα διαφορετικά μονοπάτια για τα πλοία τα οποία αντιπροσωπεύουν την μετανάστευση προς την Μ. Ασία. Κάθε ένα από τα οποία έχει και αυτό τα δικά του triggers που αλλάζουν την κατάσταση άλλων μοντέλων σε active. Η υλοποίηση εκεί έχει γίνει αντίστοιχα με τα Σχήματα 5.7-5.9.

5.3.1.2 Κίνηση στο Μονοπάτι

Τέλος, αφού έχουμε σχηματίσει το μονοπάτι της διαδρομής με βάση το οποίο θα γίνει το animation, μένει να ετοιμάσουμε και το GameObject που θα το ακολουθήσει. Αυτό γίνεται με την χρήση ενός άλλου component από το Dreamtech Splines package, το οποίο ονομάζεται Spline Follower (Σχήμα 5.10).

Διαλέγουμε το αντικείμενο που επιθυμούμε και προσθέτουμε το component από τον inspector του. Μπορούμε να παρατηρήσουμε πληθώρα επιλογών (Σχήμα 5.10,5.11).



Σχήμα 5.10: Spline Follower

Αρχικά επιλέγουμε να εκτελεστεί η πορεία ενός μονοπατιού στο πεδίο Spline, το οποίο δημιουργήσαμε με το Spline Computer, και το αντικείμενο “κλειδώνει” πάνω στην γραμμή της διαδρομής.

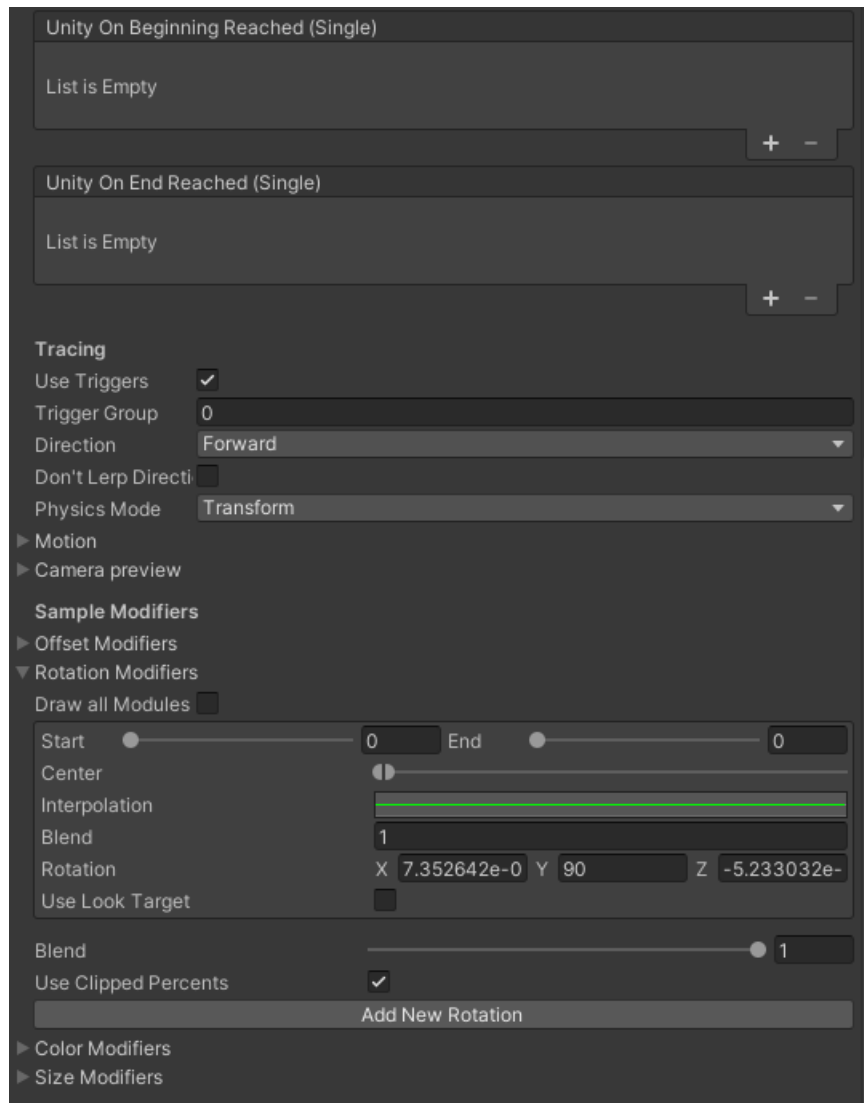
Μέσω του Clip Range μπορούμε να ορίσουμε το ποσοστό της διαδρομής στο οποίο θα κινηθεί. Επιλέχτηκε να διανυθεί ολόκληρο το μήκος της από το αρχικό μέχρι το τελικό σημείο. Έπειτα, μέσω του variable Follow Speed μπορούμε να ελέγξουμε την ταχύτητα με την οποία θα κινηθεί. Μετά από πειραματισμό επιλέχτηκε μια μικρή τιμή καθώς και η κλίμακα των μοντέλων είναι μικρή. Σε διαφορετική περίπτωση το animation ολοκληρώνεται πολύ γρήγορα. Δίνεται επίσης η δυνατότητα διαμόρφωσης της ταχύτητας με βάση τον χρόνο ή τον αριθμό των σημείων της διαδρομής (Σχήμα 5.10).

Ακόμα επιλέχτηκε το FaceDirection ώστε τα μοντέλα να είναι στραμμένα πάντα προς την φορά της κίνησης με ανάγκη διαμόρφωσης των normals που θα αναφερθεί παρακάτω.

Επομένη κίνηση είναι να ενεργοποιήσουμε την δυνατότητα να χρησιμοποιηθούν τα trigger που έχουμε φτιάξει μέσω της Boolean μεταβλητής Use Trigger. Έπειτα καθορίζουμε το group που θέλουμε, το οποίο είναι το πρώτο, δηλαδή 0 (μηδέν) (Σχήμα 5.11).

Τέλος, λόγω ιδιαίτερων τιμών περιστροφής και διαστάσεων, κάποια από τα μοντέλα δεν τοποθετήθηκαν σωστά πάνω στην γραμμή κίνησης, με αποτέλεσμα ο προσανατολισμός τους να μην είναι ο επιθυμητός.

Αυτό μπορούμε να το τροποποιήσουμε από την επιλογή RotationModifiers. Αλλάζουμε την επιλογή Rotation και σε συνδυασμό με την μεταβολή των normals του μονοπατιού, καταφέραμε το επιθυμητό αποτέλεσμα. Μέσω του Interpolation μπορούμε να τροποποιήσουμε τον ρυθμό μεταβολής σε σημεία, ώστε να είναι πιο ραγδαίος ή πιο ομαλός. Στην συγκεκριμένη περίπτωση επιλέχθηκε ένας σταθερός ρυθμός μεταβολής της περιστροφής (Σχήμα 5.11). Το τελικό αποτέλεσμα φαίνεται στο Σχήμα 5.12



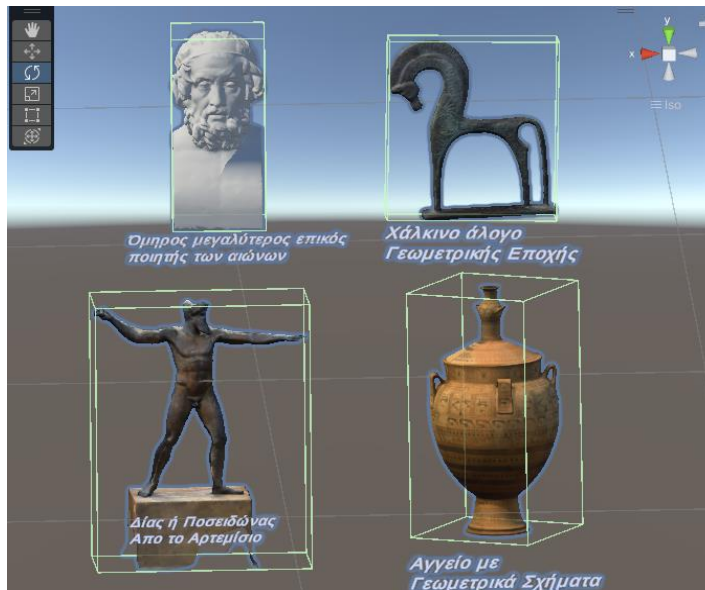
Σχήμα 5.11 : Spline Follower Part 2



Σχήμα 5.12 : Αρχικό Κομμάτι της Πρώτης Σκηνής

5.3.2 Λοιπά Αντικείμενα

Στην επόμενη φάση σχεδιασμού της πρώτης ενότητας, λήφθηκε υπόψη το περιεχόμενο των υπόλοιπων κεφαλαίων, και έγινε χρήση 3D μοντέλων για την παρουσίαση και αναπαράσταση τους(Σχήμα 5.13). Δίνουν μια πιο σφαιρική εικόνα στον μαθητή αφού μπορεί να αλληλεπιδράσει μαζί τους και να τα παρατηρήσει από διαφορετικές οπτικές γωνίες, πράγμα αδύνατο μέσα από ένα κείμενο ή μια εικόνα.



Σχήμα 5.13 :Υπόλοιπα αντικείμενα πρώτης Σκηνής

Γεωμετρικά Χρόνια

Τα μοντέλα προμηθεύτηκαν από σελίδες όπως:

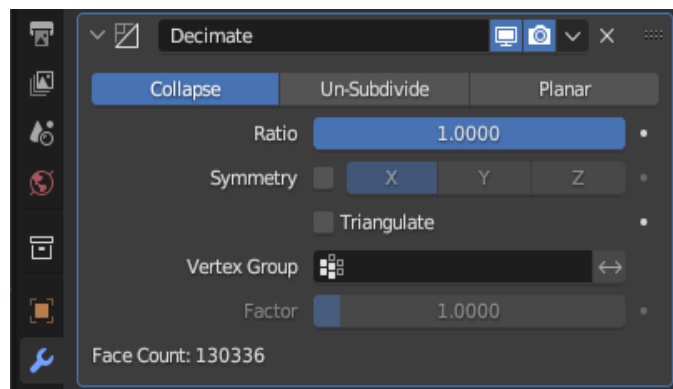
- Sketchfab [40]
- Thingiverse [41]
- Cgtrader [42]
- Turbosquid [43]
- 3dexport [44]

Καθώς πολλά από αυτά δεν προορίζονταν για χρήση σε μηχανές κατασκευής παιχνιδιών όπως είναι το Unity, χρειάστηκε να γίνει επεξεργασία τους στο Blender.

Αρχικά για την βελτίωση της απόδοσης και την μείωση της επεξεργαστικής ισχύος που θα χρειαστεί να δαπανηθεί για την εμφάνιση των μοντέλων, έγινε προσπάθεια μείωσης του όγκου τους.

Αυτό επιτεύχθηκε με την χρήση του Blender Decimate modifier [45] (Σχήμα 5.14). Μειώνει τον αριθμό τριγώνων μέσω της διαγραφής των vertices (ακμών), αφαιρώντας επιπλέον τις περιττές ακμές και μειώνοντας την πολυπλοκότητα των συνδέσεων τους. Ωστόσο αυτό έχει ως αποτέλεσμα να χαθεί μέρος της λεπτομέρειας του μοντέλου. Για τον λόγο αυτό, έγινε προσπάθεια της εύρεσης κατάλληλης ποσοστιαίας μείωσης, με διατήρηση όσο το δυνατόν μεγαλύτερης λεπτομέρειας γίνεται.

Επιπλέον μικραίνουμε την κλίμακα των αντικειμένων και τα εξάγουμε σε μορφή FBX που είναι αναγνωρίσιμη από το Unity.



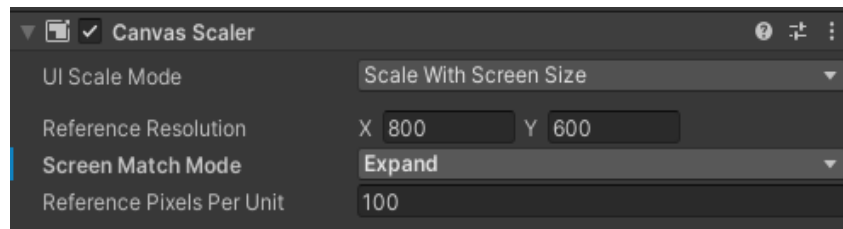
Σχήμα 5.14 :Decimate Modifier

Με την είσοδο τους στο Unity και έπειτα στην σκηνή, κάνουμε χρήση του Box Collider component και παρατηρούμε την εμφάνιση ενός πράσινου πλαισίου που περιβάλλει το αντικείμενο. Εάν η αυτόματη εισαγωγή του κουτιού-πλαισίου δεν γίνει αυτόματα σωστά στο αντικείμενο, αλλάζουμε χειροκίνητα τις διαστάσεις του. Τα μοντέλα τελικά είναι όπως φαίνονται στο σχήμα 5.11.

5.4 User Interface

Έχοντας ολοκληρώσει το κομμάτι του περιβάλλοντος της σκηνής που βρίσκονται τα 3D αντικείμενα, μπορούμε πλέον να ασχοληθούμε με το κομμάτι της διεπαφής χρήστη, στο οποίο θα βρίσκεται και το κείμενο.

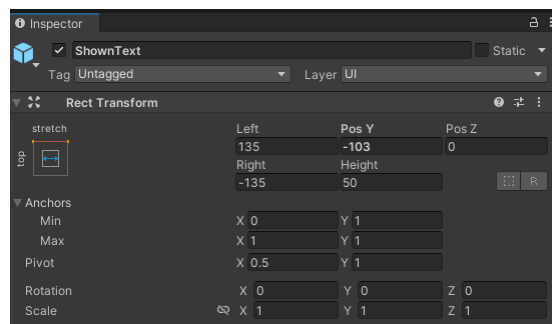
Αρχικά δημιουργούμε έναν καμβά μέσω του Hierarchy UI → Canvas. Θα αποτελέσει την βάση και το parent object μέσα στο οποίο θα τοποθετήσουμε τα υπόλοιπα στοιχεία. Για να μπορεί να αποτυπωθεί σωστά σε διαφορετικού είδους οθόνες αλλάζουμε από το CanvasScaler, την επιλογή UI Scale Model, σε ‘Scale With Screen Size’ (Σχήμα 5.15). Επίσης, αλλάζουμε το Screen Match Mode σε Expand, ώστε τα UI στοιχεία να επεκταθούν κατά το πλάτος της οθόνης, είτε σε Portrait είτε Landscape Mode.



Σχήμα 5.15 : Canvas Scaler

Στην συνέχεια προσθέτουμε ένα αντικείμενο κειμένου από το UI → Text-TextMeshPro, το οποίο τοποθετούμε στο πάνω μέρος της οθόνης. Για να μπορέσει να διατηρηθεί το σημείο που ορίσαμε αλλάζουμε από το property Rect Transform στον inspector του, τα anchor points σε Top Stretch (Σχήμα 5.16).

Σε αυτό θα τοποθετήσουμε το κείμενο, το οποίο δημιουργήσαμε συμπυκνώνοντας τα βασικότερα κομμάτια των κειμένων από κάθε σελίδα-κεφάλαιο της ενότητας.

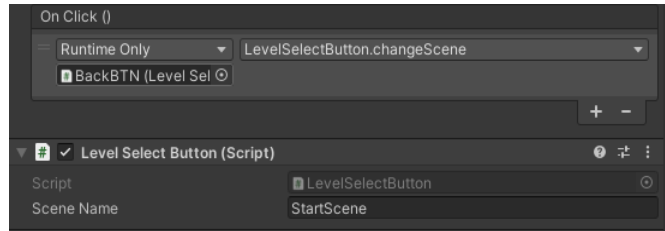


Σχήμα 5.16 : UI Text Anchor Points

Επόμενο βήμα είναι η προσθήκη των κουμπιών όπως ακριβώς έγινε και στην ενότητα “4.4 Εισαγωγική Σκηνή”.

Το ένα το τοποθετούμε στην κάτω αριστερή θέση και αλλάζουμε το anchor point σε bottom left, ώστε να είναι εμφανές, αλλά συγχρόνως να μην εμποδίζει τον χρήστη. Θα το μετατρέψουμε σε κουμπί επιστροφής στην αρχική σκηνή. Για να επιτευχθεί αυτό φορτώνουμε το script που δημιουργήσαμε για αλλαγή σκηνής και στην παράμετρο μεταβλητή ορίζουμε την σκηνή StartScene (Σχήμα 5.17).

Με στόχο την καλύτερη οπτική αναπαράσταση, κάνουμε χρήση ενός εικονιδίου, που απεικονίζει ένα βέλος στρεφόμενο προς την αριστερή διεύθυνση. Για να χρησιμοποιηθεί το αλλάζουμε σε sprite, όπως αναφέρθηκε σε προηγούμενη ενότητα. Τέλος, το προσθέτουμε στο property SourceImage, του component Image.



Σχήμα 5.17 : Κουμπί Επιστροφής

Τα επόμενα δύο κουμπιά, έχουν ως σκοπό την εκκίνηση του animation. Τα τοποθετούμε στην πάνω δεξιά πλευρά της οθόνης με τον προαναφερθέντα τρόπο. Τους δίνουμε τις ίδιες συντεταγμένες, ώστε να είναι το ένα πάνω στο άλλο. Χρησιμοποιώντας το εμφωλευμένο TMP με την Ελληνική γλώσσα, στο ένα γράφουμε “Εκκίνηση”, και στο άλλο “Επόμενο”. Αξιοποιώντας τις επιλογές που προσφέρει το component button μπορούμε να αλλάξουμε το χρώμα που έχει το κουμπί σε διάφορες του καταστάσεις.

Στην συγκεκριμένη περίπτωση χρησιμοποιούμε το property Pressed Color, το οποίο αλλάζουμε σε πράσινο. Έτσι, το χρώμα του κουμπιού αλλάζει σε πράσινο μόλις πατηθεί, προσφέροντας ένα επίπεδο επιβεβαίωσης στον χρήστη ότι η ενεργεία που έκανε έχει εκτελεστεί.

Επομένη κίνηση είναι η προσθήκη των δύο τελευταίων κουμπιών. Με αυτά θα αλλάζουμε τις σελίδες στο κείμενο, που θα προβάλλεται στο UI TMP αντικείμενο.

Για να είναι πιο εμφανής η χρήση τους αξιοποιήθηκαν δύο εικονίδια με βελάκια τα οποία δείχνουν το ένα δεξιά και το άλλο αριστερά. Τα τοποθετούμε κάτω από το UI TMP ώστε να μην κρύβουν τα γράμματα, με τον ίδιο τρόπο όπως τα προηγούμενα.

Λόγω του όγκου του κειμένου, αποφασίστηκε να δημιουργηθεί ένα ακόμα UI TMP μέσω αντιγραφής του ήδη υπάρχον για να εμφανιστεί το κείμενο σε δύο τμήματα.

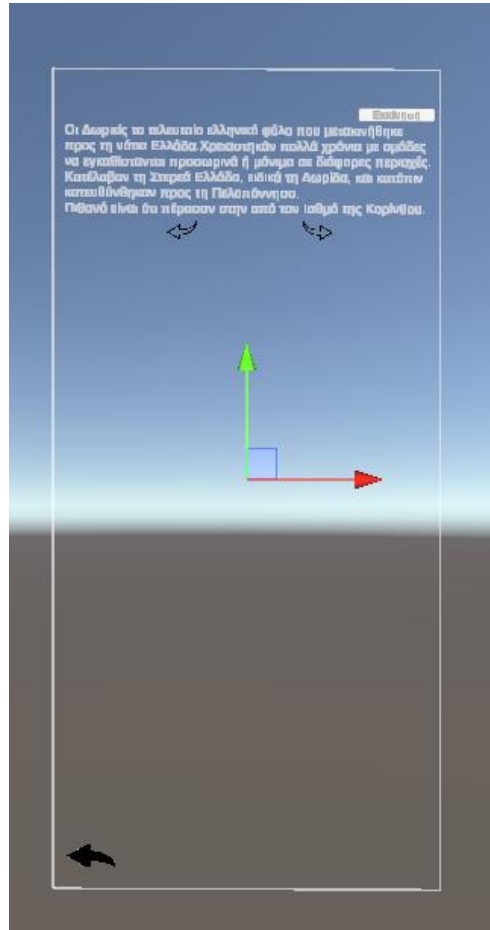
Τελευταία κίνηση είναι να δημιουργήσουμε ένα panel UI → Panel το οποίο είναι και αυτό ChildObject του Canvas. Θέτουμε από το Color για την τιμή Alpha στο 100 ώστε να είναι ελαφρώς ημιδιαφανές. Έπειτα προσθέτουμε ένα κουμπί που θα εισάγουμε την λειτουργικότητα αλλαγή σκηνής. Γράφουμε “ΕΠΟΜΕΝΗ ΕΝΟΤΗΤΑ” με μπλε φόντο στο UI TMP του. Έπειτα προσθέτουμε το script LevelSelectButton ως component με παράμετρο την επομένη σκηνή, και το περνάμε στην μέθοδο OnClick() του.

Τέλος, προσθέτουμε δύο UI TMP στο πάνελ. Στο ένα γράφουμε το μήνυμα-τίτλο “Τι είδαμε σε αυτήν την ενότητα” και το άλλο θα περιέχει μια σύντομη περιγραφή του τι έχει δει ο μαθητής στην κάθε σκηνή επιγραμματικά.

Σε αυτή την φάση έχουμε έτοιμη την υλοποίηση των UI στοιχείων και μπορούμε να τα αποθηκεύσουμε σαν prefab στον αντίστοιχο φάκελο κάνοντας δεξί κλικ Create → Prefab. Έπειτα κάνουμε drag and drop το αντικείμενο που θέλουμε πάνω στο νέο που δημιουργήσαμε και το αποθηκεύουμε. Με τον όρο Prefab

ονομάζουμε ένα GameObject το οποίο μπορούμε να δημιουργήσουμε, να διαμορφώσουμε και να αποθηκεύσουμε με όλα του τα στοιχεία, τιμές των properties άλλα και τα ChildObject. Το νέο αυτό αντικείμενο λειτουργεί ως πρότυπο για την δημιουργία άλλων αντικειμένων. Ωστόσο, μπορούμε να το τροποποιήσουμε και να δημιουργήσουμε διαφορετικές εκδοχές αν υπάρχουν άλλες ανάγκες.

Έτσι μπορούμε να αξιοποιήσουμε το Canvas αυτό στις επόμενες σκηνές, χωρίς να χρειαστεί να προβούμε ξανά στην ίδια διαδικασία για την επαναδημιουργία του. Η μορφή του Canvas φαίνεται στο Σχήμα 5.18.



Σχήμα 5.18 : UI Πρώτης Σκηνής

5.4.1 Υλοποίηση Λειτουργικότητας Animation

Έχοντας ολοκληρώσει το οπτικό κομμάτι με τα αντικείμενα και όλες τις απαραίτητες προεργασίες, μπορούμε πλέον να ασχοληθούμε με το κομμάτι υλοποίησης του κώδικα και της λειτουργικότητας.

Αρχικά ξεκινάμε με τον κώδικα για το κουμπί “Εκκίνηση” (Σχήμα 5.19). Με αυτό αρχίζει το animation του βασικού στρατιώτη που αντιπροσωπεύει τις ομάδες των Δωριέων που κατέβηκαν από τον βορρά. Το component SplineFollower ξεκινάει την λειτουργικότητα του κατευθείαν μόλις φορτώσουν τα δεδομένα του στην σκηνή ή μόλις ενεργοποιηθεί το αντικείμενο στο οποίο έχει προστεθεί. Στην συγκεκριμένη περίπτωση όμως επειδή θέλουμε ο στρατιώτης να είναι εμφανής και να αρχίσει την

Γεωμετρικά Χρόνια

κίνηση του μόλις πατηθεί το κουμπί δεν μπορούμε απλά να ενεργοποιήσουμε το αντικείμενο μέσω της μεθόδου `SetActive()`.

Το Unity όμως παρέχει εκτός από την δυνατότητα για ανίχνευση και αλλαγή κατάστασης των `GameObject` επιπλέον την ίδια δυνατότητα και για τα `component` τους.

Δημιουργούμε ένα νέο script με σκοπό την υλοποίηση αυτής της δυνατότητας. Αρχικά θα χρειαστεί η χρήση της βιβλιοθήκη `DreamteckSplines` για την αναγνώριση του `component SplineFollower` την οποία εισάγεται με την λέξη κλειδί `'using'`.

Έπειτα δημιουργούμε την μεταβλητή στην οποία θα περάσουμε από τον `inspector` το αντικείμενο.

Πρέπει να υπάρχει το attribute `[SerializeField]` για να μπορέσει να έχει πρόσβαση σε αυτό ο `editor`. Επίσης, καθώς θέλουμε να περάσει το `component` και όχι ολόκληρο το `GameObject` θέτουμε ως τύπο της μεταβλητής το `SplineFollower`, ακολουθούμενη από το όνομα της (Σχήμα 5.19).

Τέλος, φτιάχνουμε μια `function` το `StartTheAnimation()` στο οποίο ενεργοποιούμε την μεταβλητή θέτοντας την τιμή της ως `true`.

```
Assets > Scripts > Scene1StartAnimation.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using Dreamteck.Splines;
5
6  public class Scene1StartAnimation : MonoBehaviour
7  {
8      [SerializeField]
9      SplineFollower _soldierNo1;
10
11     // Start is called before the first frame update
12     void Start()
13     {
14         _soldierNo1.enabled = false;
15     }
16     // Update is called once per frame
17     void Update()
18     {
19
20     }
21     public void StartTheAnimation()
22     {
23         _soldierNo1.enabled = true;
24     }
25
26 }
```

Σχήμα 5.19 : Κώδικας ενεργοποίησης animation

Για να απενεργοποιήσουμε το `component SplineFollower`, θέτουμε την μεταβλητή που φτιάξαμε ως `false` στην μέθοδο `Start()`, η οποία είναι η πρώτη μέθοδος που εκτελείται μόλις φορτώνει η σκηνή. Με αυτό τον τρόπο δεν θα ξεκινήσει το animation αφού δεν είναι ενεργό. Θα πρέπει να περάσουμε το script και στο αντικείμενο εκτός από το κουμπί που το ενεργοποιεί.

Θα σημειωθεί ωστόσο ότι την ίδια λειτουργικότητα θα επιτυγχάνει κάνοντας uncheck το κουτί του components SplineFollower του πρώτου στρατιώτη από τον inspector. Η παραπάνω διαδικασία έγινε για την επίδειξη της λειτουργίας της μεθόδου μέσω της Start(). Στα επόμενα κεφάλαια θα προτιμηθεί η περίπτωση απενεργοποίησης από τον instector.

Για να συμπληρώσουμε την χρήση του κουμπιού “Εκκίνηση” στην μέθοδο OnClick() του προσθέτουμε το script που αναφέρθηκε πριν περνώντας με παράμετρο το GameObject του στρατιώτη. Επιπλέον προστέθηκε στην OnClick() η μέθοδος GameObject.SetActive() με σαν true με παράμετρο το κουμπί “Επόμενο” και false με παράμετρο τον εαυτό του. Έτσι μόλις πατηθεί θα εξαφανιστεί από την οθόνη και θα εμφανιστεί στην θέση του το δεύτερο κουμπί.

Η ίδια λογική ακολουθήθηκε και για την ενεργοποίηση του animation των τριών βαρκών από το κουμπί “Επόμενο”, με νέο script τριών μεταβλητών τύπου SplineFollower. Η διαφορά είναι, ότι δεν θα εμφανιστεί κάποιο άλλο κουμπί.

5.4.2 Αλληλεπίδραση με τα Αντικείμενα

Μετά την εισαγωγή τους στο περιβάλλον αν θέλουμε να μπορεί ο χρήστης της εφαρμογής να αλληλεπιδρά με τα μοντέλα αντί να βρίσκονται απλά στατικά στον χώρο θα χρειαστεί να εφαρμόσουμε κάποιο script. Για την υλοποίηση αυτής της δυνατότητας θα κάνουμε χρήση του LeanTouch package.

Πρώτο βήμα είναι η δημιουργία δύο άδειων αντικειμένων. Θα αποτελέσουν τα βασικά controllers για τον χειρισμό των τρισδιάστατων μοντέλων μέσω αφής στην οθόνη. Στο πρώτο κάνουμε εισαγωγή του Lean Touch το οποίο αποτελεί και τον βασικό έλεγχο για την ανίχνευση της αφής.

Καθώς θέλουμε να μπορούμε να αναγνωρίσουμε κάθε ένα από τα αντικείμενα και να τα χειριστούμε ξεχωριστά, εισάγουμε στο δεύτερο άδειο αντικείμενο τα script, Lean Finger Down και Lean Select By Finger. Για να γίνεται αυτό μόνο όσο διατηρούμε την αφή πάνω στο αντικείμενο που θέλουμε να συνδέουμε. Στην μέθοδο OnFinger() του script FingerDown, περνάμε το script SelectByFinger(). Έπειτα επιλέγουμε να καλείται η μέθοδος SelectScreenPosition() κάθε φορά που καλείται η OnFinger().

Με αυτές τις κινήσεις έχουμε διαμορφώσει πλήρως τα δύο touch controllers.

Οι κινήσεις που θέλουμε να μπορούν να κάνουν τα αντικείμενα είναι η μετακίνηση τους στην οθόνη, η αλλαγή κλίμακας, και η περιστροφή γύρω από κάποιο άξονα. Για να μπορούν να κάνουν αυτές τις τοπολογικές μετατροπές στον χώρο, θα χρειαστεί να εισάγουμε σε κάθε ένα από αυτά τα αντίστοιχα scripts μέσα από τον inspector τους. Αναφορικά αυτά είναι το LeanPinchScale, LeanDragTranslate, LeanRotateAxis όπου στο τελευταίο επιλέγουμε τον επιθυμητό άξονα ή άξονες περιστροφής.

Τέλος, για να επιτύχουμε την αυτόνομη κίνηση του κάθε ενός μοντέλου προσθέτουμε και scripts το LeanSelectableByFinger.

Αντίστοιχα τα παραπάνω βήματα ακολουθούνται κάθε φορά για κάθε νέα σκηνή τόσο για την δημιουργία των χειριστών(controllers) όσο και για τα νέα αντικείμενα που θέλουμε να μπορούμε να έχουμε αλληλεπίδραση μέσω αφής. Για αντικείμενα όπως οι χάρτες κάνουμε τις παραπάνω ενέργειες στο ParentObject και λόγω της ιεραρχίας οι μεταβολές των συντεταγμένων ισχύουν και σε όλα τα ChildObject που έχει.

5.4.3 Λειτουργικότητα Αλλαγής Σελίδας και Κατάστασης Αντικειμένων

Η ιδέα πίσω από την εμφάνιση του κειμένου είναι να μπορεί ο χρήστης να διαβάσει και να αντλήσει τις ιστορικές πληροφορίες. Ταυτόχρονα θα βλέπει μπροστά του σε πραγματικό χρόνο τις AR σκηνές να εξελίσσονται μπροστά του μέσα από την οθόνη της android συσκευής του. Αρχική σκέψη ήταν το κείμενο να τοποθετηθεί σε ένα ScrollView καθώς είναι αρκετά εκτενές για να εμφανιστεί ολόκληρο. Κάτι τέτοιο ωστόσο, δεν θα μπορούσε να διαχωρίσει τα κεφάλαια του βιβλίου και θα υπήρχε σύγχυση με τα μοντέλα. Έτσι υπερίσχυσε η ιδέα της εμφάνισης με την μορφή τμημάτων ανά σελίδες. Τέτοια λειτουργικότητα μας δίνει το TMP από το attribute Overflow μέσω της επιλογής Page.

Ενεργοποιώντας αυτή την επιλογή το κείμενο το οποίο δεν μπορεί να εμφανιστεί στα όρια του text area που έχουμε ορίσει, δεν χάνεται αλλά κόβεται και διαχωρίζεται σε “σελίδες”. Στόχος κάθε μιας, είναι να χωράει το κείμενο στην περιοχή εμφάνισης. Αυτό μπορεί να ελέγχει επιπρόσθετα, μέσω της χρήσης του rich tag <page>, που κόβει το κείμενο σπάζοντας το σε μικρότερες σελίδες.

Η δυνατότητα αυτή χρησιμοποιήθηκε για κατακερματισμό του κειμένου σε ποιο ευανάγνωστα τμήματα, τα οποία έχουν λογική συνέχεια και συνδέουν το περιεχόμενο με τα 3D μοντέλα που εμφανίζονται.

Μπορούμε να δούμε τις σελίδες του διαχωρισμένου κειμένου αλλάζοντας την τιμή των ακέραιων αριθμών στο textbox, που εμφανίζεται διπλά ακριβώς από το dropdown list του Overflow, όταν έχουμε διαλέξει την επιλογή Page.

Καθώς θέλουμε να μπορούμε να αλλάζουμε τις σελίδες από τα δύο κουμπιά που έχουμε δημιουργήσει, θα χρειαστεί ένα script το οποίο θα επιτελεί αυτή την ενέργεια (Σχήμα 5.20).

Αρχικά δημιουργούμε ένα νέο αρχείο C# όπως έχει προαναφερθεί, και το τοποθετούμε στον φάκελο με τα υπόλοιπα.

Καθώς θα χειριστούμε αντικείμενα τύπου TMP χρειάζεται η βοήθεια της αντίστοιχης βιβλιοθήκης που εισάγουμε γράφοντας “using TMPPro”. Έπειτα δημιουργούμε μια μεταβλητή τύπου TextMeshProGUI καθώς το αντικείμενο το οποίο θα περαστεί μέσω του inspector είναι τέτοιου τύπου. Αυτό γίνεται για να γνωρίζει ο editor ποιο αντικείμενο θέλουμε να διαχειριστεί. Τέλος, δημιουργούμε δύο μεταβλητές τύπου int, τις

- currentpage
- totalpages

Την πρώτη την αρχικοποιούμε περνώντας τον αριθμό ένα, ώστε να εμφανίζεται πάντα σαν αρχική η πρώτη σελίδα.

Έπειτα δημιουργούμε την μέθοδο ButtonNextPage() η οποία θα καλείται για να πραγματοποιηθεί η αλλαγή.

Μέσω βοηθητικών πεδίων που παρέχει η βιβλιοθήκη TMPro παίρνουμε τις πληροφορίες για τον συνολικό αριθμό των σελίδων με το `textInfo.pageCount`, και για τον αριθμό της τρέχουσας σελίδας με το `.PageToDisplay`. Μέσω των βοηθητικών πεδίων ενημερώνουμε τις μεταβλητές που δημιουργήσαμε στην αρχή, και τις χρησιμοποιούμε ως δείκτες. Ακολουθεί έλεγχος στον οποίο αν η σελίδα που είμαστε είναι μικρότερη ή ίση από τις συνολικές, αυξάνουμε τον αριθμό της `currentpage`, και της προβαλλόμενης σελίδας(μεσω της `.PageToDisplay`) κατά ένα (Σχήμα 5.20).

Με την ίδια λογική δημιουργήσαμε και μια `ButtonPreviousPage()` για το κουμπί της προηγούμενης σελίδας, με την διαφορά ότι δεν θα αφήνει να κατέβει κάτω από την πρώτη.

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

public class TEXTMESH PAGES : MonoBehaviour

[SerializeField]
public TextMeshProUGUI PageText;
//public TMP_Text PageText;
int currentpage = 1;
int totalpages;

public void ButtonNextPage()
{
    totalpages = PageText.textInfo.pageCount;
    currentpage= PageText.pageToDisplay;
    if (currentpage <= totalpages)
    {
        currentpage++;
        PageText.pageToDisplay++;
    }
}

public void ButtonPreviousPage()
{
    totalpages = PageText.textInfo.pageCount;
    currentpage= PageText.pageToDisplay;
    if (currentpage >= totalpages || currentpage >1)
    {
        currentpage--;
        PageText.pageToDisplay--;
    }
}
```

Σχήμα 5.20: Script Αλλαγής Σελίδας

Μετά από αυτό το βήμα, αφού έχουμε μια σαφή μέθοδο για την ανάγνωση και αλλαγή του κειμένου. Μπορούμε λοιπόν να εστιάσουμε στον τρόπο με τον οποίο θα αλλάζουμε τις καταστάσεις των αντικειμένων, ανάλογα την σελίδα που διαβάζει ο χρήστης.

Φτιάχνουμε ένα νέο script, το `ToggleModelsBasedOnPages` (Σχήμα 5.21), εισάγουμε την βιβλιοθήκη TMPro και δημιουργούμε εκτός από τις μεταβλητές που αναφέρθηκαν στο προηγούμενο αρχείο μερικές τύπου `GameObject`. Επίσης δημιουργούμε τις τύπου `int`, `TargetPage`, `TargetPage2`, και ακόμα μια μεταβλητή τύπου `TextMeshProUGUI`. Στην τελευταία θα εισάγουμε το δεύτερο TMP UI του Canvas.

Στη συνέχεια δημιουργούμε την void μέθοδο `ShowNextModel()` η οποία ανάλογα με την σελίδα που βρίσκεται το TMP UI, και αυτές που έχουμε ορίσει εμείς στις `TargetPage`, αλλάζει το state των μοντέλων.

Αρχικά λαμβάνουμε την κατάσταση των αντικειμένων, με τη βοήθεια της Boolean τοπικής μεταβλητής `currentState` και του πεδίου `activeSelf`, για κάθε ένα από αυτά. Ελέγχουμε αν ταυτίζονται οι αριθμοί της

Γεωμετρικά Χρόνια

σελίδας-στόχου, με την τρέχουσα, και αν αυτό ισχύει, τότε αλλάζουμε τα αντικείμενα στην αντίθετη κατάσταση. Για το δεύτερο κείμενο, καθώς δημιουργείται σφάλμα εάν είναι απενεργοποιημένο το αντικείμενο κατά την εκτέλεση του προγράμματος, απενεργοποιούμε το component Text UI. Το ενεργοποιούμε θέτοντας το σαν `.enabled = true` όταν θέλουμε να είναι εμφανές το κείμενο σε συγκεκριμένη σελίδα.

Η μέθοδος αυτή έχει τροποποιηθεί σε διάφορες εκδοχές, για να καλύψουμε τις ανάγκες των αντικειμένων στις υπόλοιπες σκηνές. Επίσης, έχει αυξηθεί ο αριθμός των σελίδων-στόχων αντίστοιχα.

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

public class ToggleModelsBasedOnPages : MonoBehaviour
{
    [SerializeField]
    public TextMeshProUGUI PageText;
    public GameObject menuObject;
    public GameObject menuObject2;
    public GameObject menuObject3;
    public GameObject menuObject4;

    public int TargetPage;
    public int TargetPage2;
    //public TMP_Text Text2; also work
    public TextMeshProUGUI Text2;
    int currentpage;
    int totalpages;

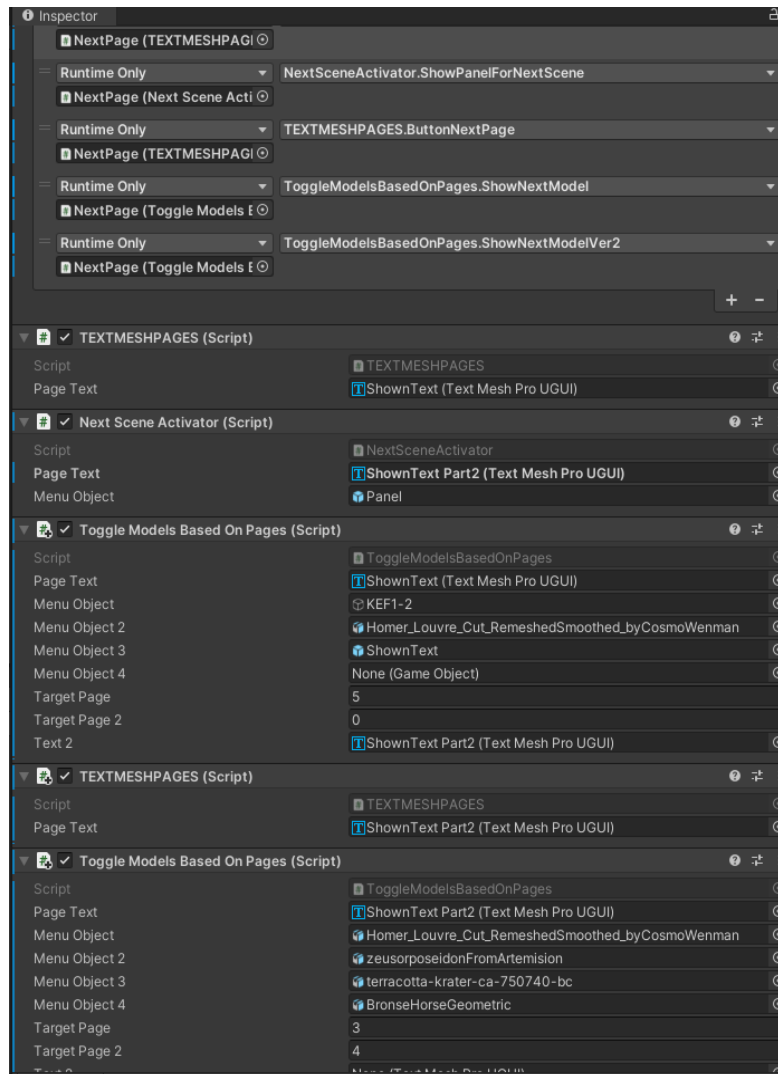
    public void ShowNextModel()
    {
        totalpages = PageText.textInfo.pageCount;
        bool currentState = menuObject.activeSelf;
        bool currentState2 = menuObject2.activeSelf;
        bool currentState3 = menuObject3.activeSelf;
        currentpage = PageText.pageToDisplay;

        if (currentpage == TargetPage)
        {
            menuObject.SetActive(!currentState);
            menuObject2.SetActive(!currentState2);
            menuObject3.SetActive(!currentState3);
            Text2.enabled = true;
        }
    }
}
```

Σχήμα 5.21 : Αλλαγή Κατάστασης με Βάση την Σελίδα

Με αυτό το βήμα έχουμε ολοκληρώσει την λειτουργικότητα του UI. Το μόνο που μένει είναι να εμφανίσουμε το τελικό panel που έχουμε δημιουργήσει στο 5.4 User Interface. Εκεί ανακεφαλαιώνουμε τις πληροφορίες που υπάρχουν στην σκηνή και δίνουμε την δυνατότητα πλοήγησης στην επομένη.

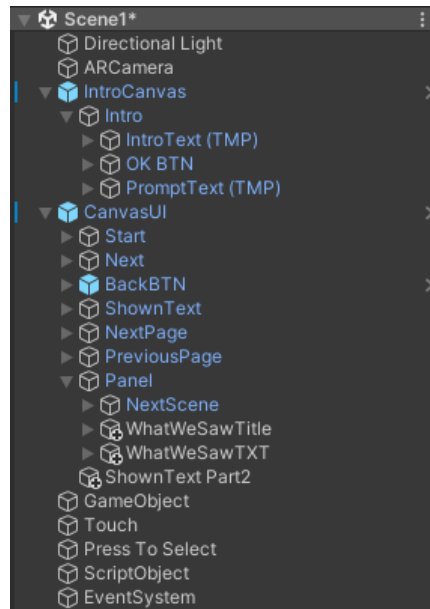
Το τελικό Panel ενεργοποιείται σύμφωνα με την προαναφερθέντα λογική. Συγκεκριμένα όταν ο αριθμός της τρέχουσας σελίδας γίνει ίσος με τον αριθμό των συνολικών σελίδων. Η ενεργοποίηση όλων αυτών των script γίνεται μέσα από την μέθοδο `OnClick()` των κουμπιών `NextPage` (Σχήμα 5.22) και `PreviousPage`.



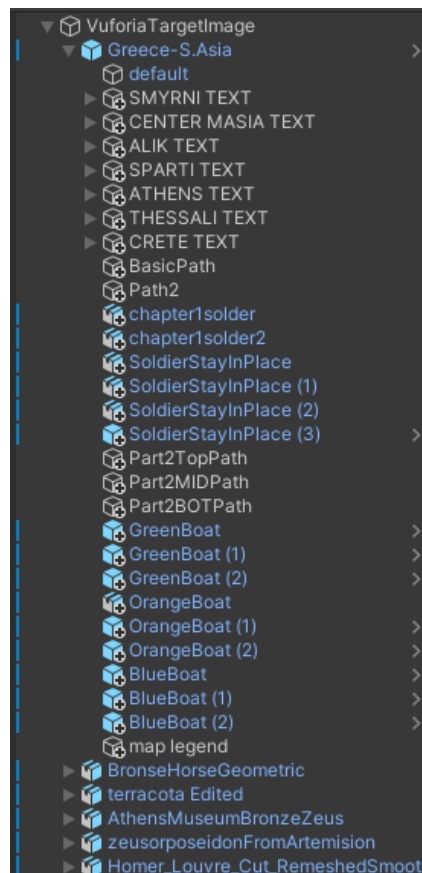
Σχήμα 5.22 : Μέθοδοι και Script του BTN Nextpage

Κατά αντίστοιχο τρόπο διαμορφώνουμε την χρήση και λειτουργικότητα του UI και στις επόμενες σκηνές, με φυσικά ελάχιστες τροποποιήσεις κάθε φορά.

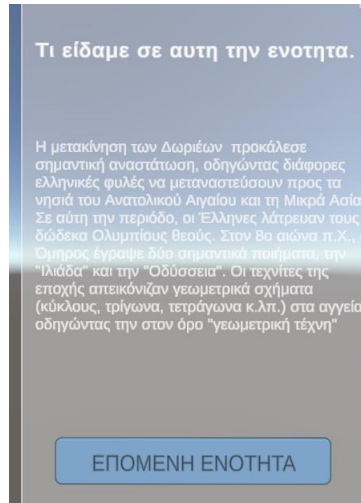
Σε αυτή την φάση έχουμε πλέον ολοκληρώσει την πρώτη σκηνή που αναφέρεται στην πρώτη ενότητα του σχολικού βιβλίου. Η τελική ιεραρχία των αντικειμένων είναι η εξής (Σχήμα 5.23,5.24) :



Σχήμα 5.23 : Ιεραρχία Αντικειμένων Πρώτης Σκηνής



Σχήμα 5.24 : Υπόλοιπα Αντικείμενα Πρώτης Σκηνής



Σχήμα 5.25 : Ανασκόπηση Πρώτης Σκηνης

Στο σχήμα 5.25 φαίνεται η μορφή του τελικού πάνελ στη σκηνής.

5.5 Επίλογος

Η ύλη στην οποία βασίστηκαν οι πληροφορίες που παρουσιάζονται σε αυτό το πέμπτο κεφάλαιο της Π.Ε είναι η πρώτη Ενότητα του σχολικού βιβλίου.

Στο παρόν κεφάλαιο είδαμε την γενική μορφή της εφαρμογής. Επίσης, τον τρόπο λειτουργίας πίσω από το UI, το κείμενο, και χρήση των scripts. Ακόμα, αναλύθηκε η διαμόρφωση, εμφάνιση και κίνηση των μοντέλων. Με βάση αυτό το μοτίβο θα συνεχίσουν να δημιουργούνται και οι υπόλοιπες σκηνές, πάντα τροποποιημένες και διαφοροποιημένες για να καλύπτουν τις ανάγκες της κάθε ενότητας.

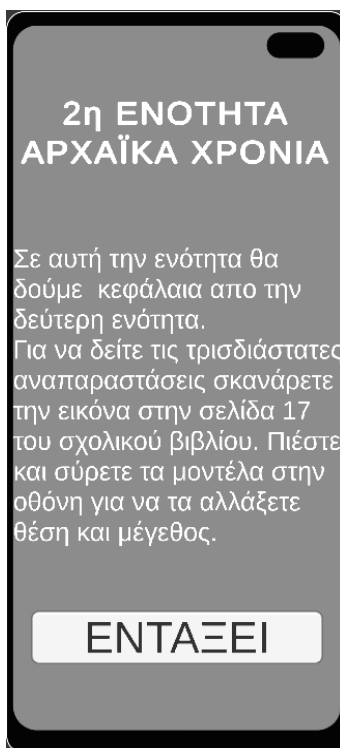
Κεφάλαιο 6ο : Αρχαϊκά Χρόνια

6.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο θα αναφερθούμε στην δεύτερη ενότητα του σχολικού βιβλίου που αφορά τα Αρχαϊκά Χρόνια. Η σκηνή έχει την βασική δομή την οποία αναλύσαμε στο προηγούμενο κεφάλαιο. Ωστόσο, η διαφορά είναι ότι στην συγκεκριμένη υπάρχει μικρό animation εικόνων, εκτός από αυτό των μοντέλων. Επίσης προστέθηκε ένα μικρό quiz για την επιβεβαίωση κατανόησης των πολιτευμάτων, με βάση την απεικόνιση τους στην ενότητα. Επιπλέον, ο χρήστης θα καλείται να επιλέξει ανάμεσα από δύο μοντέλα διαφορετικών ρυθμών τέχνης, και θα ενημερώνεται για την ορθότητα της απάντησής του.

6.2 Panel Εισαγωγής

Αντίστοιχα με το Κεφάλαιο 5 το πρώτο πράγμα που συναντά ο χρήστης είναι το εισαγωγικό panel(Σχήμα 6.1). Εκεί παρουσιάζουμε τις πληροφορίες που θα χρειαστεί για την εμφάνιση των AR στοιχείων. Πατώντας το κουμπί “ΕΝΤΑΞΕΙ” απενεργοποιούμε το αντικείμενο, και ενεργοποιούμε το Canvas και το ImageTarget.



Σχήμα 6.1 : Δεύτερο Intro Panel

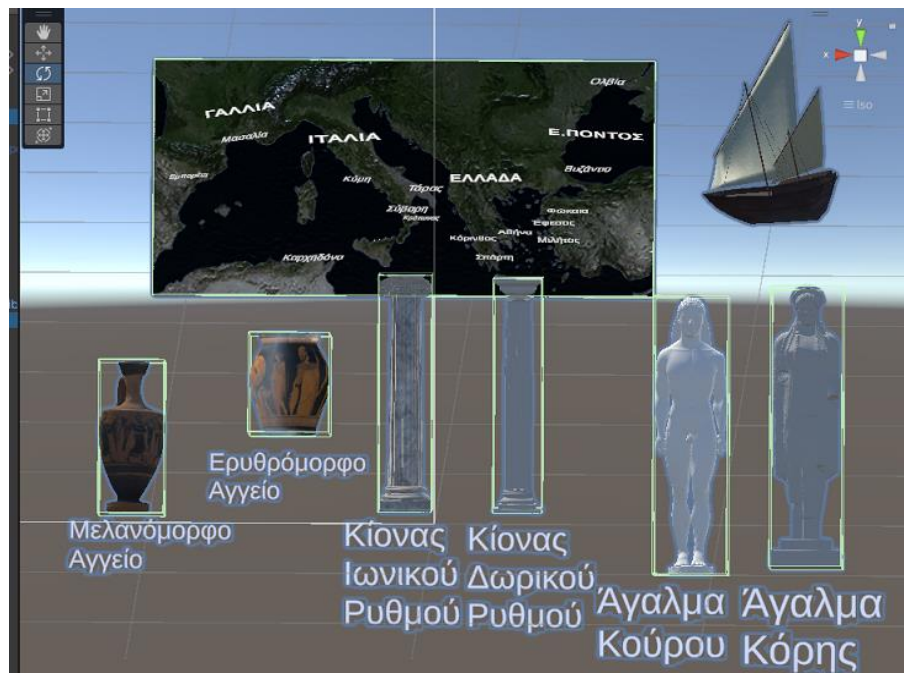
6.3 Δημιουργία Περιβάλλοντος

Αρχικά ακολουθώντας τη διαδικασία του 3.4, αλλάζουμε την default camera σε AR Camera, όπως και στο προηγούμενο κεφάλαιο. Αντίστοιχα χρειαζόμαστε ένα στόχο για εντοπισμό και επαύξηση. Για την δεύτερη ενότητα επιλέχτηκε η εικόνα της σελίδας 17 του σχολικού βιβλίου (Σχήμα 6.2).



Σχήμα 6.2 : Εικόνα Δεύτερου Image Target

Έπειτα, εισάγουμε τα μοντέλα, τα οποία πρώτα τροποποιήθηκαν στο Blender με Decimate modifier. Αφαιρούμε τυχόν επιπλέον mesh, μέσα από το Edit Mode του Blender, τα οποία μπορεί να μπερδεύουν και να επιβαρύνουν την μηχανή Unity. Τέλος, τα εξάγουμε με τύπο FBX.



Σχήμα 6.3 : Αντικείμενα Δεύτερης Σκηνής

Για τον χρωματισμό του μοντέλου του χάρτη χρησιμοποιήσαμε material το οποίο περιλαμβάνει τέσσερις εικόνες στα εξής πεδία. Albedo για το βασικό χρώμα, Specular για την αντανάκλαση του φωτός με Smoothness 100%, NormalMap για το διάνυσμα κατεύθυνσης των normals και Occlusion για την διαπερατότητα. Έπειτα προστέθηκαν όπως και πριν τα 3D TMP για την ονομασία των περιοχών και δημιουργήθηκαν τρία μονοπάτια για να απεικονίσουν κάποιες από τις πορείες των μεταναστεύσεων. Τα διατρέχουν μοντέλα πλοίου(Σχήμα 6.3) με χρήση του component SplineFollower όπως και πριν.

Σε κάθε ένα από τα μοντέλα, εκτός από το πλοίο, κάνουμε χρήση του BoxCollider component. Επιπρόσθετα, το ίδιο κάνουμε στα 3D TMP που αναφέρουν την Γαλλία, Ιταλία και Εύξεινο Πόντο. Στόχος είναι να λειτουργήσουν ως κουμπιά, και να ενεργοποιήσουν το animation μόλις πατηθούν. Η ενημέρωση του χρήστη για αυτά, θα γίνεται μέσα από μήνυμα προτροπής, που θα εμφανίζεται σε συγκεκριμένη σελίδα. Επίσης, θα του υποδεικνύεται πόσα από τα τρία έχει ενεργοποιήσει, ώστε να μπορέσει να προχωρήσει στην επόμενη σελίδα.

Στα 3D αντικείμενα προστέθηκαν ως component, scripts, τα οποία επιτρέπουν την μετακίνηση, περιστροφή και αλλαγή διαστάσεων στο μοντέλο μέσω της αφής της οθόνης. Ο χρήστης θα μπορεί να τα επεξεργαστεί βλέποντάς τα από οποιαδήποτε γωνία επιθυμεί. Έτσι, αυξάνουμε το ενδιαφέρον και την δυνατότητα κατανόησης τους.

Επιπρόσθετα θα κάνουμε τροποποίηση στο script ToggleONOFF που δημιουργήσαμε στο υποκεφάλαιο 5.4, για να μπορούν τα αντικείμενα με BoxCollider να συμπεριφέρονται ως κουμπιά όταν πατηθούν στην οθόνη.

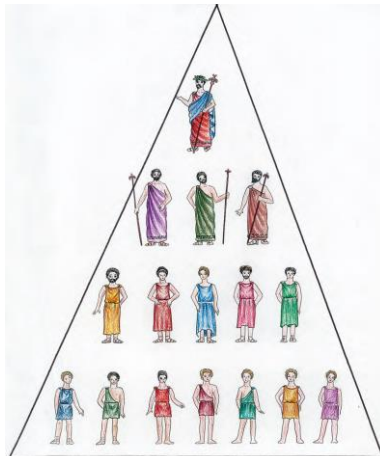
6.4 User Interface

Καθώς έχουμε ολοκληρώσει το οπτικό κομμάτι που αφορά το περιβάλλον της σκηνής, επόμενο σημαντικό βήμα είναι να προχωρήσουμε στην υλοποίηση της λειτουργικότητας τους.

Σχετικά με την προβολή του κειμένου, έχουμε δημιουργήσει το prefab του Canvas, άρα δεν χρειάζεται να ξαναπροβούμε σε όλες αυτές τις διεργασίες που αναφέρθηκαν στην ενότητα 5.4.

Μπορούμε απλά να σύρουμε το αντικείμενο prefab στο περιβάλλον της σκηνής, και έτσι δημιουργούμε έναν νέο αντικείμενο με όλα τα ChildObject, επιλογές και component που είχαμε ορίσει όταν το αποθηκεύσαμε ως prefab.

Όπως έχει αναφερθεί, αυτό δεν σημαίνει ότι πρέπει να μείνει ακριβώς το ίδιο. Όσο δεν αφαιρούμε ή αλλάζουμε τη σειρά της ιεραρχίας στα αντικείμενα του, παραμένει ως έχει. Έχουμε όμως την δυνατότητα να τροποποιήσουμε τις παραμέτρους του, και να προσθέσουμε νέα επιπλέον αντικείμενα στην ιεραρχία του. Με βάση αυτό λοιπόν δημιουργούμε μια νέα εικόνα από το UI → Image, και περνάμε στο SourceImage την παρακάτω εικόνα (Σχήμα 6.4), την οποία πρωτύτερα μετατρέψαμε σε sprite.

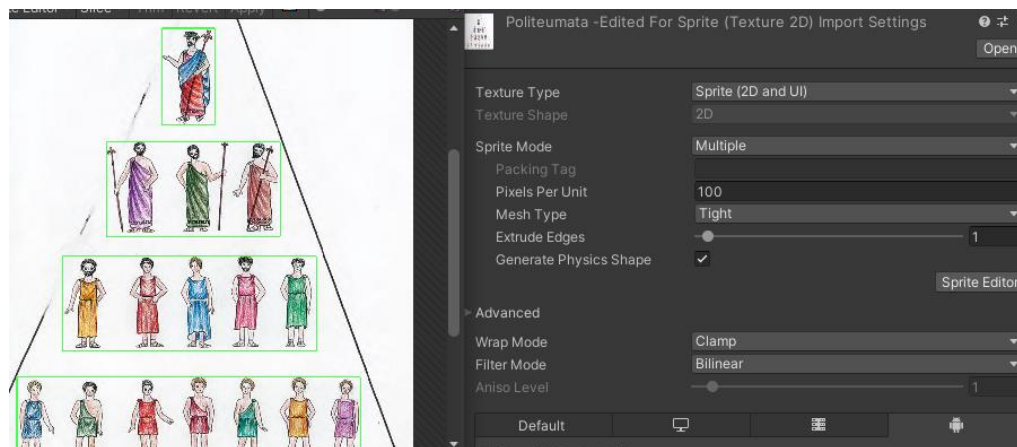


Σχήμα 6.4: Εικόνα Πολιτευμάτων

Η συγκεκριμένη εικόνα θα χρησιμοποιηθεί και για το παρακάτω quiz, όπου έγινε επεξεργασία της στο Paint3D. Αφαιρέσαμε τις πλάγιες μαύρες γραμμές και κάναμε πιο ομαλή τη μεταβολή του χρώματος στο background. Τέλος, την τοποθετήσαμε στον φάκελο των εικόνων μαζί με την παραπάνω μη επεξεργασμένη.

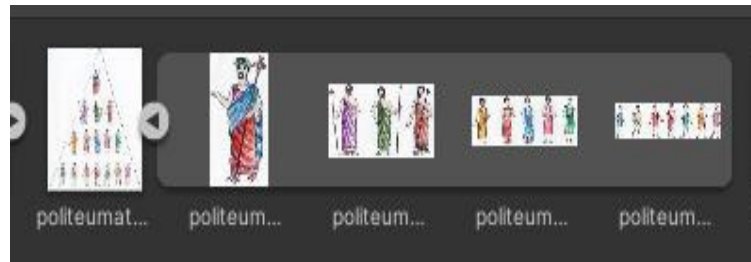
Επομένη διαδικασία είναι, αφού την μετατρέψουμε και αυτή σε Sprite, να αλλάξουμε το SpriteMode σε Multiple.

Αυτό μας δίνει την δυνατότητα να ανοίξουμε το Sprite Editor (Σχήμα 6.5), από και όπου χωρίσαμε την εικόνα σε τέσσερα μέρη, ένα για κάθε πολίτευμα που αντιπροσωπεύει.



Σχήμα 6.5 : Δημιουργία Τμημάτων Sprites

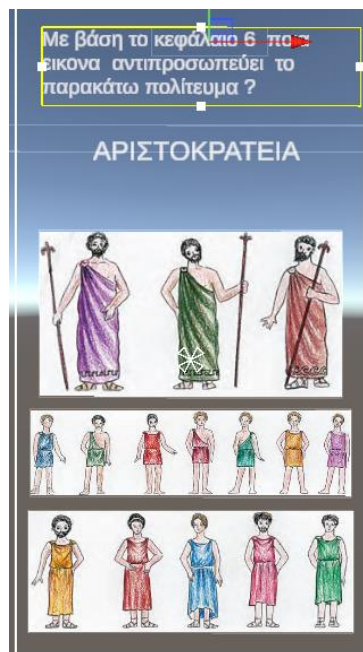
Έτσι όταν σώσουμε τις αλλαγές, μπορούμε εκτός από την ολοκληρωμένη εικόνα να επιλέξουμε τα τμήματα της. Όπως φαίνεται παρακάτω προέκυψαν τέσσερα sprites, τα οποία θα χρησιμοποιήσουμε σαν επιλογές για το quiz (Σχήμα 6.6).



Σχήμα 6.6 : Τελικά Sprites

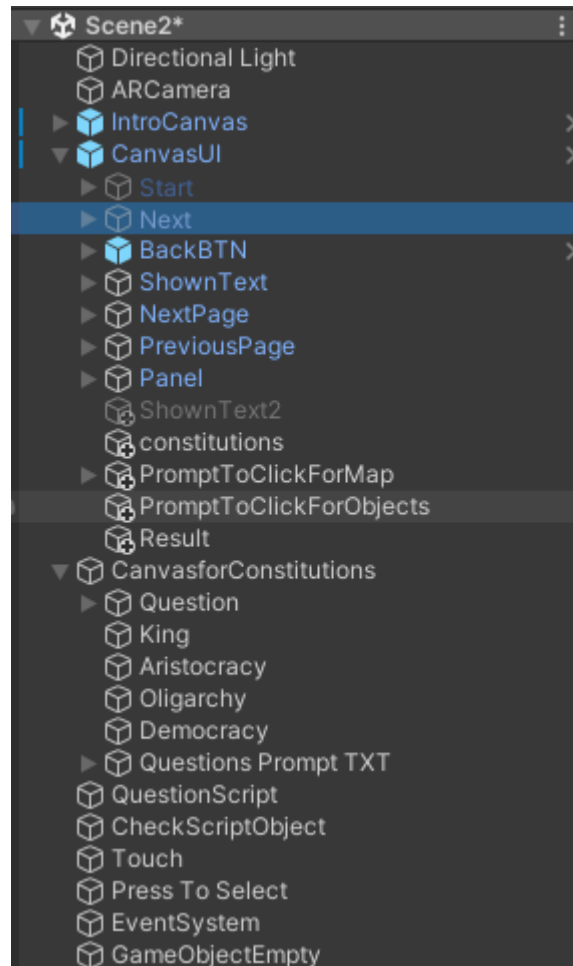
Έχοντας πλέον έτοιμες τις εικόνες (Σχήμα 6.6), δημιουργούμε ένα νέο Canvas, τον CanvasforConstitutions. Μέσα σε αυτό δημιουργούμε τέσσερα νέα αντικείμενα UI Image, και σε κάθε ένα από τα οποία θέτουμε ως SourceImage μια από τις παραπάνω εικόνες-sprites για να προβληθούν. Επίσης, προσθέτουμε το component Button, ώστε να μπορούν να πατηθούν και να λειτουργούν ως κουμπιά. Έπειτα, προσθέτουμε δύο ακόμα UI TMP. Στο ένα θα εμφανίζεται το κείμενο προτροπής-ερώτησης του Quiz. Στο άλλο αλλάζουμε από τα properties του, το overflow σε page. Χρησιμοποιούμε το rich text tag <page>, για να χωρίσουμε σε σελίδες τα ονόματα των πολιτευμάτων μέσα στο Text Input Box του δεύτερου TMP. Στην τελευταία σελίδα βάζουμε το κείμενο “ΠΑΤΗΣΤΕ ΕΔΩ ΓΙΑ ΣΥΝΕΧΕΙΑ”.

Τέλος, προσθέτουμε και εδώ το component button. Έτσι, το UI TMP θα λειτουργήσει ως κουμπί ολοκλήρωσης του quiz. Όταν φτάσει το κείμενο στην τελευταία σελίδα, θα εμφανίζει τις παραπάνω λέξεις παροτρύνοντας τον χρήστη να το πατήσει. Η αρχική κατάσταση του Quiz φαίνεται στο Σχήμα 6.7.



Σχήμα 6.7 : Αρχική Κατάσταση Quiz

Σε αυτό το σημείο έχουμε ολοκληρώσει την υλοποίηση του οπτικού κομματιού του UI όποτε το μόνο που απομένει είναι να περάσουμε στο στάδιο της λειτουργίας. Τα αντικείμενα της σκηνής φαίνονται στο Σχήμα 6.8.



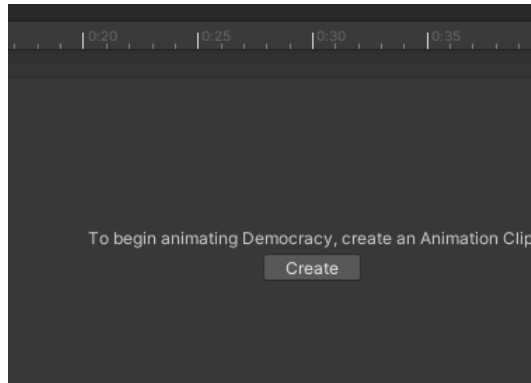
Σχήμα 6.8 : Ιεραρχία Αντικειμένων UI

6.4.1 Animation αντικειμένων UI

Πριν από την ενασχόληση μας με τον κώδικα θα δημιουργήσουμε ένα μικρό animation για την εισαγωγή της εικόνας του σχήματος 6.3, καθώς και της εικόνας για την Αριστοκρατία. Έτσι, δεν θα φανεί απότομη η αλλαγή και θα γίνει πιο ενδιαφέρουσα η παρουσίαση.

Για να πραγματοποιήσουμε το animation αρχικά χρειάζεται να ανοίξουμε το animation TAB (Σχήμα 6.9). Αυτό γίνεται επιλέγοντας το επιθυμητό στοιχείο, και έπειτα Window → Animation → Animation. Το TAB ανοίγει με έτοιμο ως προεπιλογή το αντικείμενο που είχαμε επιλεγμένο.

Ακολουθούμε τις οδηγίες που μας εμφανίζονται και δημιουργούμε ένα νέο αρχείο τύπου .anim. Το αποθηκεύουμε σε ένα νέο φάκελο που ονομάσαμε animations.

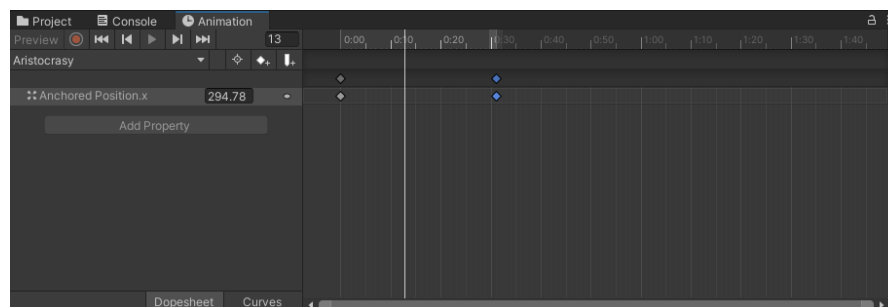


Σχήμα 6.9 : Animation TAB Create .anim

Έχοντας δημιουργήσει το αρχείο μπορούμε να προχωρήσουμε στο animation. Πατάμε το κόκκινο κουμπί στα αριστερά ώστε να αρχίσει να καταγράφει τις κινήσεις μας. Τοποθετούμε το αντικείμενο που θέλουμε, στην προκειμένη περίπτωση την εικόνα, στο αρχικό σημείο. Το αποθηκεύουμε κάνοντας κλικ πάνω στο timeline, στο χρονικό σημείο που θέλουμε. Δημιουργούμε έτσι ουσιαστικά ένα keyframe. Έπειτα το μετακινούμε στο επόμενο σημείο που θέλουμε, εδώ το τελικό, και επαναλαμβάνουμε το παραπάνω βήμα. Τέλος, ξαναπατάμε το κόκκινο αρχικό κουμπί για να σταματήσουν να καταγράφονται νέες θέσεις (Σχήμα 6.10).

Η μετακίνηση του από το αρχικό σημείο στο τελικό γίνεται αυτόματα από τον υπολογιστή, όπου υπολογίζει τα σημεία που πρέπει να περάσει, δημιουργώντας έτσι την αίσθηση της κίνησης. Αν θέλουμε η κίνηση να γίνεται πιο αργά η γρήγορα μπορούμε να μετακινήσουμε το σημείο keyframe στο timeline επιλέγοντας το, κρατώντας το παρατεταμένα και έπειτα μπορούμε να το σύρουμε στον άξονα του χρόνου.

Επιλέγοντας το αρχείο .anim μπορούμε να δούμε στον inspector την επιλογή Loop Time. Μέσω αυτής το animation επαναλαμβάνεται μόλις ολοκληρωθεί. Απενεργοποιούμε την επιλογή, διότι θέλουμε να εκτελεστεί μόνο μια φορά.



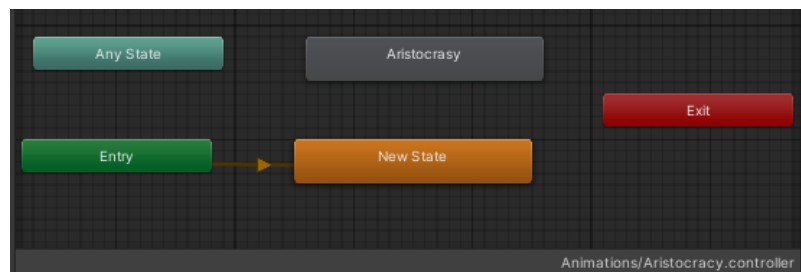
Σχήμα 6.10 : Animation Tab

Επόμενο βήμα είναι η δημιουργία ενός animation controller, από το Create → Animation Controller. Του δίνουμε το ίδιο όνομα με το αρχείο .anim ώστε να μην υπάρχει σύγχυση. Το ανοίγουμε και πηγαίνουμε αυτόματα στο animator TAB. Εκεί φορτώνουμε το αρχείο .anim που δημιουργήσαμε, μέσω drag n drop

(Σχήμα 6.11).Αυτό το αρχείο είναι υπεύθυνο για την ενεργοποίηση της κίνησης και υπό ποιες προϋποθέσεις θα γίνει. Η παραπάνω διαδικασία ακολουθείται και για τις δύο εικόνες, με διαφορά, την κατεύθυνση της κίνησης της κάθε μιας.

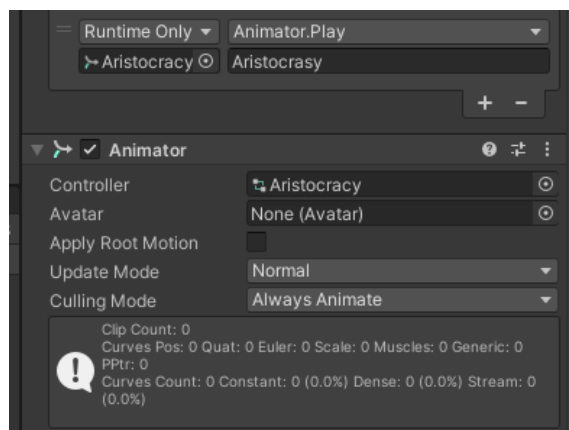
Τα παρακάτω ακολουθηθήκαν μόνο για το sprite της Αριστοκρατίας.

Δημιουργούμε μια καινούργια κενή κατάσταση στο Animation Controller και την θέτουμε σαν Default State. Αν δεν το κάνουμε το animation θα ξεκινάει από μόνο του, κάτι που δεν επιθυμούμε.



Σχήμα 6.11 : Animator Tab

Τέλος, στις εικόνες προσθέτουμε το component Animator και στο πεδίο controller περνάμε τον αντίστοιχο controller που δημιουργήσαμε πριν. Για να ενεργοποιηθεί η εικόνα της Αριστοκρατίας, μόνο όταν κάνουμε κλικ, προσθέτουμε ακόμα μια onclick() μέθοδο. Σε αυτήν περνάμε από την λίστα την μέθοδο Animator.play(string), με παράμετρο string το όνομα του Animator controller της (Σχήμα 6.12).



Σχήμα 6.12 : Inspector για το Animation

6.4.2 Λειτουργικότητα Quiz και Υλοποίηση

Η βασική ιδέα πίσω από quiz είναι, αφού διδαχθεί στον μαθητή το κομμάτι για τα πολιτεύματα, αμέσως μετά να ερωτηθεί σε αυτά, ώστε να τα κατανοήσει πλήρως. Μέσω των προαναφερθέντων script που έχουμε δημιουργήσει όταν φτάσει στην κατάλληλη σελίδα, εμφανίζουμε την εικόνα του σχήματος 6.3. Συνδυαστικά με το κείμενο αποκτάει μια βασική αντίληψη της δομής και εξέλιξης των πολιτευμάτων. Μόλις φτάσουμε την τελευταία σελίδα που τα αφορά, αντί να εμφανίσουμε την επόμενη, το αντικείμενο Canvas απενεργοποιείται και ενεργοποιείται το CanvasforConstitutions που δημιουργήσαμε.

Δίνουμε στον χρήστη τη λέξη του πολιτεύματος, και πρέπει να επιλέξει ανάμεσα από τρεις εικόνες τη σωστή. Για να μπορεί να ξεχωρίσει αν έχει απαντήσει σωστά ή λάθος, η εικόνα αλλάζει χρώμα σε κόκκινη για την λανθασμένη απάντηση, και πράσινη στην σωστή.

Έπειτα από κάθε επιλογή απάντησης, υπάρχει μια μικρή καθυστέρηση για να μπορέσει να αντιληφθεί ο χρήστης το αποτέλεσμα.

Αν είναι σωστή τότε η σελίδα στο TMP αλλάζει εμφανίζοντας το επόμενο πολίτευμα και η χρωματική κατάσταση των εικόνων επανέρχεται σε default, που είναι η λευκή. Εάν δεν υπάρχουν άλλες ερωτήσεις, τότε εμφανίζεται το μήνυμα “ΠΑΤΗΣΤΕ ΕΔΩ ΓΙΑ ΣΥΝΕΧΕΙΑ” στην περιοχή του κειμένου. Εάν συνεχίσουμε να πατάμε τις εικόνες, θα γίνουν κόκκινες και θα παραμείνουν έτσι ώστε να καταλάβει ο χρήστης ότι δεν υπάρχει άλλη ερώτηση.

Κάθε μια φωτογραφία έχει και ένα αριθμό, ο οποίος αντιστοιχεί στον αριθμό της σελίδας, που είναι το κάθε πολίτευμα. Αν υπάρχει ταύτιση μεταξύ των δύο αυτών σημαίνει ότι έχει γίνει σωστή επιλογή.

Για το κομμάτι της υλοποίησης των προαναφερθέντων υπάρχει κώδικας ο οποίος θα ελέγχει τις απαντήσεις σε ένα νέο αρχείο C# script.

Αρχικά καθώς θέλουμε να περάσουμε σαν παράμετρο ένα TMP UI πρέπει να εισάγουμε την βιβλιοθήκη TMPto όπως έχει προαναφερθεί. Επιπλέον θα χρειαστούμε δύο ακόμα επιπλέον βιβλιοθήκες για την υλοποίηση του κώδικα. Αυτές είναι οι :

- UnityEngine.UI
- System.Threading.Tasks

Αρχικά δημιουργούμε τις μεταβλητές για την αλλαγή σελίδων καθώς και την απόσπαση πληροφοριών της τρέχουσας σελίδας και τον αριθμό των συνολικών όπως έχει προαναφερθεί.

Τέλος, θα χρειαστούμε μια λίστα τύπου Button που θα γίνει populate με τις εικόνες-κουμπιά.

Επόμενο βήμα είναι η δημιουργία της ασύγχρονης void μεθόδου, στην οποία θα περνάμε σαν μεταβλητή τύπου INT την επιλογή που έχει γίνει με βάση τον αριθμό που έχουμε δώσει σε κάθε εικόνα.

Η λίστα που έχει τα κουμπιά επιλογών ξεκινάει από το μηδέν, ωστόσο οι σελίδες πρέπει να ξεκινάνε από το ένα. Για αυτό τον λόγο αυξάνουμε τον αριθμό επιλογής του χρήστη κατά μια μονάδα κάθε φορά πριν τον χρησιμοποιήσουμε, ώστε να υπάρχει σωστή αντιστοίχιση.

Μέσα σε έλεγχο if τσεκάρουμε αν υπάρχει αυτή η ταύτιση, και αν υπάρχει, αλλάζουμε την εικόνα σε πράσινο.

Έπειτα από μια μικρή καθυστέρηση αυξάνουμε τον αριθμό της τρέχουσας σελίδας, της σελίδας εμφάνισης και αλλάζουμε όλες τις εικόνες σε άσπρο.

Σε διαφορετική περίπτωση αλλάζουμε την επιλεγμένη εικόνα σε κόκκινο.

Τέλος καθώς είναι τέσσερις οι επιλογές αλλά θα εμφανίζονται μόνο οι τρεις, αποφασίστηκε η πρώτη επιλογή μόλις απαντήσει σωστά και χρωματιστεί πράσινη, να μετακινηθεί προς τα δεξιά με την χρήση animation που έχει προαναφερθεί στο 6.4.2. Έπειτα το αντικείμενο απενεργοποιείται και να γίνεται αντικατάσταση του. Ο κώδικας σε C# φαίνεται παρακάτω στο Σχήμα 6.13.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using UnityEngine.UI;
using System.Threading.Tasks;

public class QuestionAnswers : MonoBehaviour
{
    public TextMeshProUGUI PageText;

    //public GameObject CanvasforConstitutions;
    public List<Button> ImageBTN = new List<Button>();
    //public List<Sprite> ImageBTN = new List<Sprite>();
    int currentpage;
    int Showpage;
    public Color SelectedRed;
    public Color SelectedGreen;

    public async void PressedChoice(int clickedChoice)
    {
        //ColorBlock cb = ImageBTN[clickedChoice].colorBlock;
        //if(currentpage == 0)
        // {Showpage = 1;}
        //button.GetComponent<Button>().colorBlock.pressedColor = new Color(1, 1, 1);

        Showpage = PageText.pageToDisplay;
        Debug.Log("clicked"+clickedChoice);
        Debug.Log("page"+Showpage);

        if (1+clickedChoice== Showpage){
            ImageBTN[clickedChoice].GetComponent<Image>().color = Color.green;
            await Task.Delay(700);
            PageText.pageToDisplay++;
            Showpage = PageText.pageToDisplay;
            Debug.Log(PageText.pageToDisplay);

            ImageBTN[1].GetComponent<Image>().color = Color.white;
            ImageBTN[2].GetComponent<Image>().color = Color.white;
            ImageBTN[3].GetComponent<Image>().color = Color.white;
        }

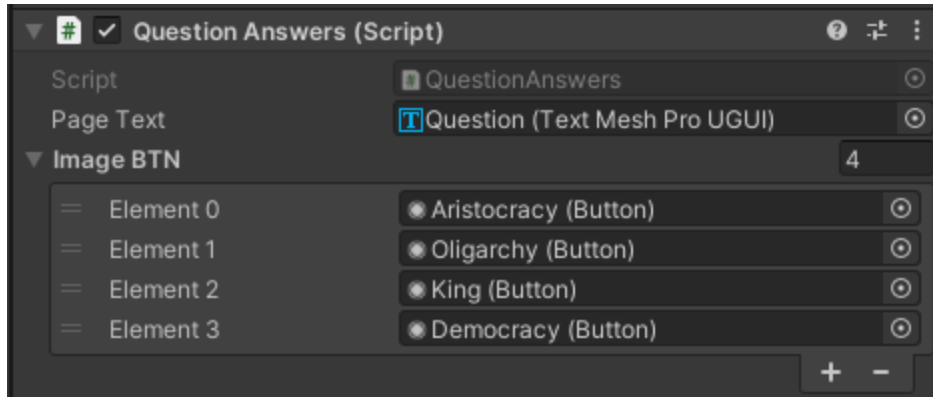
        else {
            //cb.pressedColor = SelectedRed;
            //ImageBTN[clickedChoice].color=cb;
            ImageBTN[clickedChoice].GetComponent<Image>().color = Color.red;
        }
    }
}
```

Σχήμα 6.13 : Κώδικας Λειτουργίας Quiz

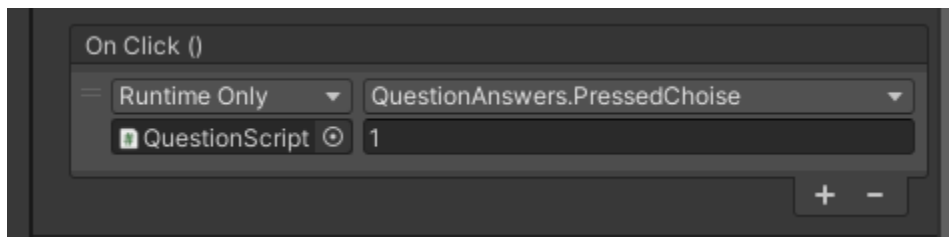
Για την εφαρμογή του script αρχικά χρησιμοποιήθηκε ένα νέο άδειο GameObject όπου περάστηκε σαν component. Στα attributes, περάστηκαν αρχικά σαν μεταβλητή, το αντικείμενο TMP που έχει τις λέξεις ερωτήσεις, και στην λίστα των κουμπιών το κάθε ένα αντικείμενο εικόνα (Σχήμα 6.14). Εισήχθησαν με σειρά ανάλογη της θέσης σελίδας που αντιστοιχεί η απάντηση στο TMP (Σχήμα 6.15).

Έπειτα το προαναφερθέν αντικείμενο περάστηκε στην μέθοδο OnClick() της κάθε εικόνας-κουμπί με επιλεγμένη μέθοδο την PressedChoice(int ClickedChoice) που δημιουργήσαμε.

Καθώς η μέθοδος δέχεται παράμετρο, από δίπλα υπάρχει ένα textbox για να εισάγουμε ακέραιο αριθμό. Βάζουμε αριθμό αντίστοιχο με την θέση που έχει το αντικείμενο στην λίστα με τα κουμπιά(ImageBTN) (Σχήμα 6.14).



Σχήμα 6.14 : Το Αντικείμενο για το script



Σχήμα 6.15 : Παράδειγμα OnClick() από την Ολιγαρχία.

Μόλις ολοκληρωθεί το quiz και πατήσουμε το κουμπί για τον τερματισμό του τότε το CanvasforConstitutions απενεργοποιείται και ενεργοποιεί ξανά τον βασικό Canvas. Έχει πλέον μεταβεί στην επομένη σελίδα από αυτή που είχε διακόψει την ροή του, και ταυτόχρονα εμφανίζει τα νέα αντικείμενα που αναφέρονται στο κείμενο της τρέχουσας σελίδας.

6.4.2 Κώδικας χειρισμού σκηνής

Όπως αναφέραμε στην αρχή του κεφαλαίου, ξεκινάμε με το κομμάτι του κώδικα από το script ToggleONOFF που θα χρειαστεί να τροποποιήσουμε. Μέσω αυτού θα μετατρέψουμε τα 3D αντικείμενα που έχουν BoxCollider, σε toggle κουμπιά αφής.

Όπως φαίνεται στο Σχήμα 6.16, ο κώδικας τοποθετείται στην μέθοδο Update() καθώς θέλουμε να γίνεται έλεγχος σε κάθε frame για τυχόν πάτημα στην οθόνη. Όταν αυτό συμβεί, και μέσω της βοηθητικής μεθόδου Raycast, εάν στο σημείο το οποίο ακούμπησε ο χρήστης στην οθόνη υπάρχει αντικείμενο με BoxCollider τότε καλούμε την ToggleActiveState(). Με αυτόν τον τρόπο πλέον τα αντικείμενα μας μετατρέπονται σε κουμπιά.

```

void Update()
{
    if (Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Began){ // Inp
        Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position); // Input
        RaycastHit hit;
        if( Physics.Raycast(ray, out hit) && hit.collider.gameObject== gameObject ){
            ToggleActiveState();
            // if (hit.transform == transform){ // this also works
            // if (hit.collider.tag == "Object1"){ No need for tag here
            // } }
        }
    }
}

```

Σχήμα 6.16 : Update method του script ToggleONOFF

Στην συνέχεια φορτώνουμε το script μέσω του inspector στα 3D TMP για την Γαλλία, Ιταλία και Εύξεινο Πόντο που είναι στην ιεραρχία του χάρτη. Κάθε ένα τα τρία 3D TMP ενεργοποιεί το animation ενός διαφορετικού πλοίου για την περιοχή που βρίσκεται. Αυτό γίνεται αφού περάσουμε το αντικείμενο σαν παράμετρο στο script για να το διαχειριστεί η ToggleActiveState().

Για να ελέγχουμε εάν έχουν ενεργοποιηθεί τα πλοία και έχει γίνει το animation δημιουργούμε ένα δεύτερο script.

Αρχικά στην τρίτη σελίδα, απενεργοποιούμε το κουμπί μετάβασης στην επόμενη, μόνο για αυτή. Ουσιαστικά δεν αφήνουμε τον χρήστη να προχωρήσει πέραν του συγκεκριμένου σημείου. Ακολουθώς ενεργοποιούμε μήνυμα προτροπής ώστε να πατήσει τα κουμπιά, στο οποίο αλλάζουμε το κείμενο από τον κώδικα. Σε αυτό τον ενημερώνουμε για το πόσα από τα τρία μονοπάτια έχουν εμφανιστεί.

Έπειτα ελέγχουμε την κατάσταση του κάθε ενός από τα τρία πλοία, για το αν είναι ενεργοποιημένο. Εάν αυτό ισχύει τότε θέτουμε τον προσωπικό του μετρητή στην μονάδα όταν αυτό ισχύει. Μόλις ενεργοποιηθούν και τα τρία το κουμπί επόμενης σελίδας ενεργοποιείται ξανά και παραμένει σε αυτή την κατάσταση. Όταν ο χρήστης προχωρήσει στην τέταρτη σελίδα, απενεργοποιούμε το μήνυμα προτροπής-ενημέρωσης.

Όσον αφορά τον κώδικα εμφάνισης των υπόλοιπων μοντέλων η λογική είναι η ίδια με αυτή του Κεφαλαίου 5 της Π.Ε, ωστόσο για αυτή την σκηνή προτιμήθηκε η δημιουργία νέου script καθώς χρειάζονται κάποιες επιπρόσθετες μεταβολές.

Για τις σελίδες εμφάνισης των κίωνων και αμφορέων, εμφανίζουμε μέσω του κώδικα, στο ίδιο UI TMP και για τα δύο είδη αντικειμένων, μήνυμα ερώτησης. Ο χρήστης καλείται να επιλέξει ένα από τα δύο μοντέλα που βλέπει, με βάση τον ρυθμό τέχνης που του ζητάμε. Σε περίπτωση σωστής απάντησης εμφανίζεται το μήνυμα “ΣΩΣΤΟ” με πράσινα γράμματα και λανθασμένης το “ΛΑΘΟΣ” με κόκκινα γράμματα. Ταυτόχρονα εμφανίζονται οι λεζάντες και των δύο. Οι απαντήσεις εμφανίζονται σε ένα διαφορετικό UI TMP από το μήνυμα ερώτησης, ωστόσο και εδώ γίνεται μέσω του κώδικα.

Ο έλεγχος γίνεται μέσω κλήσης της μεθόδου ShowTheAnswer() (Σχήμα 6.17) από την Update() που τρέχει σε κάθε frame. Η αναγνώριση γίνεται με την ίδια λογική κώδικα όπως και αυτής στο σχήμα 6.14, ωστόσο η ταυτοποίηση του κάθε μοντέλου γίνεται μέσω της σύγκρισης Tag. Έχουμε δώσει διαφορετικό Tag σε κάθε μοντέλο, που συγκρίνεται με αυτό του αντικειμένου που πατάει ο χρήστης μέσω αφής, χρησιμοποιώντας την CompareTag().

```

public void ShowTheAnswer(){
    if(currentpage == TargetPage3){
        if (Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Began){
            Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);
            RaycastHit hit;
            if( Physics.Raycast(ray, out hit) && (hit.collider.gameObject.CompareTag("Ionikos"))){
                OptionIonic.SetActive(true);
                OptionDoric.SetActive(true);
                Result.SetActive(true);
                Result.GetComponent<TMP_Text>().text= "ΣΩΣΤΟ";
                Result.GetComponent<TMP_Text>().color = Color.green;
            }
            else if( Physics.Raycast(ray, out hit) && (hit.collider.gameObject.CompareTag("Dorikos"))){
                OptionDoric.SetActive(true);
                OptionIonic.SetActive(true);
                Result.SetActive(true);
                Result.GetComponent<TMP_Text>().text= "ΛΑΘΟΣ";
                Result.GetComponent<TMP_Text>().color = Color.red;
            }
        }
    }
}

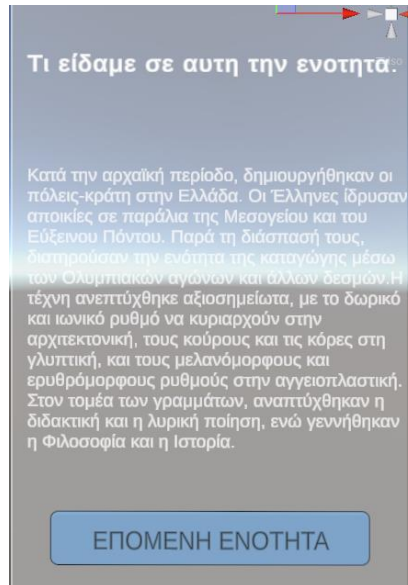
```

Σχήμα 6.17 : Κώδικας Ελέγχου Αντικειμένων με πάτημα αφής

Η ροή συνεχίζεται με τον ίδιο τρόπο που έχουμε αναφέρει και στο 5ο Κεφάλαιο. Όταν ο χρήστης φτάσει στην τελευταία σελίδα και προσπαθήσει να μεταβεί στην επομένη, εμφανίζεται το Panel ανασκόπησης που τον ειδοποιεί ότι έχει ολοκληρωθεί η ενότητα, και τον προτρέπει να μεταβεί στην επομένη σκηνή.



Σχήμα 6.18 : Ιεραρχία Αντικειμένων Δεύτερης Σκηνής



Σχήμα 6.19 : Ανασκόπηση Δεύτερης Σκηνής

Στο Σχήμα 6.18, μπορούμε να δούμε την ιεραρχία αντικειμένων της δεύτερης σκηνής και στο Σχήμα 6.19, φαίνεται η μορφή του panel ανασκόπησης της.

6.5 Επίλογος

Σε αυτό το έκτο κεφάλαιο παρουσιάστηκε η υλοποίηση της δεύτερης ενότητας του σχολικού βιβλίου. Ακολουθήθηκε κατά βάση η διαδικασία ανάπτυξης από το προηγούμενο κεφάλαιο. Κάναμε προσθήκη νέων χαρακτηριστικών, με την μορφή animation UI στοιχείων και ερωτήσεων επιλογής, για επιπλέον πρόκληση ενδιαφέροντος του χρήστη. Τέλος, είδαμε έναν ακόμα τρόπο χρήσης ενεργοποίησης των scripts μέσω ενός κενού GameObject.

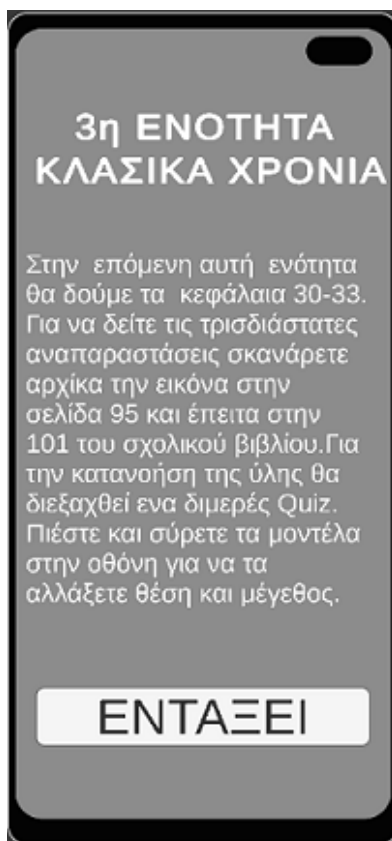
Κεφάλαιο 7ο : Κλασικά Χρόνια(ΜΑΚΕΔΟΝΙΑ)

7.1 Εισαγωγή

Το έβδομο αυτό κεφάλαιο αναφέρεται στην τρίτη ενότητα του σχολικού βιβλίου που αφορά τα Κλασικά Χρόνια και συγκεκριμένα το πέμπτο υποκεφάλαιο για την Μακεδονία. Η διαμόρφωση της σκηνής είναι στα πλαίσια που έχουν οριστεί στα προηγούμενα κεφάλαια με επιπρόσθετα στοιχεία για το οπτικό κομμάτι, και με την ίδια λογική λειτουργίας του UI. Σε αυτήν την σκηνή χρησιμοποιήθηκαν δύο διαφορετικά image target, λόγω της ανάγκης κάλυψης δύο διαφορετικών εκτενών γεωγραφικών περιοχών. Στο δεύτερο χάρτη, κατά την διάρκεια των animation, υπάρχει ένα σημείο στο οποίο γίνεται παύση αυτών. Εμφανίζεται ένα μικρό ερωτηματολόγιο στον χρήστη με ερωτήσεις σύμφωνα με τα κεφάλαια της ύλης. Μόλις αυτό απαντηθεί, τα animation συνεχίζουν. Υπάρχει αξιολόγηση του αριθμού των απαντήσεων στο τελικό panel.

7.2 Panel Εισαγωγής

Όπως και πριν, αρχικά εμφανίζουμε το πάνελ εισαγωγής στον χρήστη, ώστε να τον πληροφορήσουμε σε ποια σελίδα του βιβλίου πρέπει να μεταβεί ώστε να λειτουργήσει η AR Camera(Σχήμα 7.1). Επίσης από αυτό ενημερώνεται και για το περιεχόμενο που θα συναντήσει μέσα σε αυτή την σκηνή.



Σχήμα 7.1 : Τρίτο Intro Panel

7.3 Δημιουργία Περιβάλλοντος

Πρώτη κίνηση είναι να προσθέσουμε την AR Camera που παρέχει το Vuforia engine και να διαγράψουμε την default. Στην συνέχεια εισάγουμε το πρώτο image target, ακολουθώντας τα βήματα από το υποκεφάλαιο 3.4.3 Vuforia Image Target. Αυτό θα είναι και το parent GameObject, όπου κάτω από την ιεραρχία του, θα τοποθετηθούν όλα τα υπόλοιπα αντικείμενα για να εμφανιστούν, μόλις αυτό εντοπιστεί και αναγνωριστεί. Από την βάση που έχουμε κάνει εισαγωγή, επιλέγουμε την εικόνα που φαίνεται στο Σχήμα 7.2.

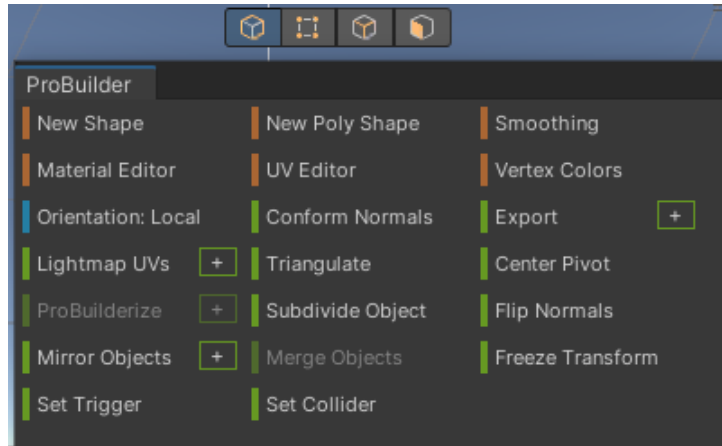


Σχήμα 7.2 : Εικόνα Τρίτου Image Target

Για τον 3D χάρτη απεικόνισης της περιοχής χρησιμοποιήθηκε το ίδιο μοντέλο με αυτό της πρώτης ενότητας που έχουμε αναφέρει στο 5ο Κεφάλαιο της Π.Ε, με τα αντίστοιχα components ώστε να έχουμε αλληλεπίδραση μαζί του.

Επόμενο βήμα είναι να δημιουργήσουμε πάνω στον χάρτη τμήματα, τα οποία τον χωρίζουν στα επιμέρους κράτη που αναφέρονται στο κείμενο του μαθήματος. Αντί να γίνει ένας απλός χρωματισμός της εικόνας στο texture του χάρτη, προτιμήθηκε να δημιουργήσουμε αντικείμενα που θα επικαλύπτουν την περιοχή. Προσθέτουμε έτσι ένα ακόμα επίπεδο στον όγκο του μοντέλου. Για την υλοποίηση αυτή μπορούμε να κάνουμε χρήση του ProBuilder, ώστε να δημιουργήσουμε ακριβώς τα σχήματα που θέλουμε μέσα από το περιβάλλον της σκηνής.

Μπορούμε να έχουμε πρόσβαση στο παράθυρο επιλογών του ProBuilder από το Tools → Probuilder → Probuilder Window (Σχήμα 7.3).

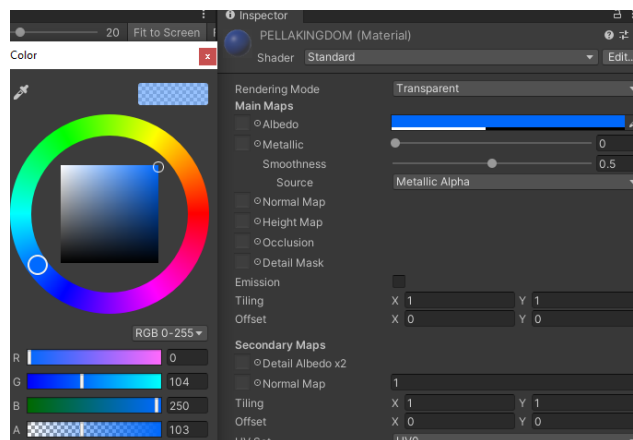


Σχήμα 7.3 : ProBuilder Object

Κάνουμε χρήση του New Poly Shape, με το οποίο επιλέγουμε σημεία στον χώρο τα οποία θα αποτελέσουν τις ακμές (vertices) του αντικειμένου μας και συνδέονται μεταξύ τους. Ακολουθούμε την περίμετρο των κρατών στον χάρτη ώστε η τελευταία ακμή να συμπέσει με την πρώτη. Έτσι ολοκληρώνουμε το σχήμα που θέλουμε. Έχουμε την δυνατότητα να κάνουμε extrude το σχήμα επιλέγοντας διάσταση για το ύψος του. Επιλέγουμε μια πολύ μικρή τιμή ώστε να είναι τρισδιάστατο, αλλά να μην καταλαμβάνει πολύ χώρο και κρύβει το μοντέλο από κάτω.

Τέλος, καθώς το default material δεν έχει χρώμα δημιουργούμε ένα νέο φάκελο, τον SeeThroughMaterials. Μέσα σε αυτόν δημιουργούμε ένα νέο material, και το περνάμε στο αντικείμενο, δίνοντας του το χρώμα που έχει στην εικόνα το κράτος που θέλουμε. Παράδειγμα ενός από αυτά, είναι για το κράτος της Πέλλας στο Σχήμα 7.4.

Καθώς επιθυμούμε να είναι διαφανές για να μπορούμε να εντοπίσουμε τα στοιχεία του χάρτη που βρίσκεται από κάτω, αλλάζουμε το Rendering Mode σε Transparent. Μεταβάλλουμε την τιμή Alpha του χρώματος, προς το 0(μηδέν), μέχρι να βρούμε μια ικανοποιητική τιμή, ώστε να είναι και τα δύο εμφανή και διακριτά.



Σχήμα 7.4 : Material για το Κράτος της Πέλλας

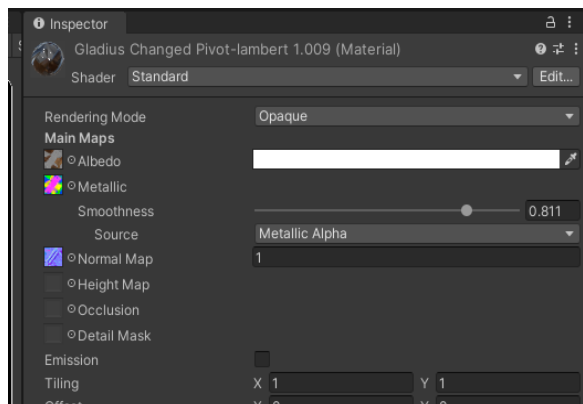
Τέλος, επιλέγοντας το αντικείμενο που μόλις δημιουργήσαμε μέσω του Probuilder, αλλάζουμε το EditMode Vertex Selection και παρατηρούμε ότι αλλάζει και το παράθυρο των επιλογών του Probuilder. Από τις επιλογές, διαλέγουμε το Triangulate, ώστε να χωριστεί το σχήμα στα βασικά του τρίγωνα, και μέσω της επιλογής μετακίνησης των ακμών τοποθετούμε το σχήμα ακριβώς πάνω από το μοντέλο του χάρτη. Τέλος, επιλέγουμε το αντικείμενο από την ιεραρχία, ώστε να μην χρειαστεί η επιλογή κάθε ενός τριγώνου του. Μέσω της επιλογής Smoothing το τροποποιούμε, ώστε να λειανθεί και βελτιωθεί η εμφάνιση του. Με αντίστοιχο τρόπο, έγιναν και τα υπόλοιπα δύο σχήματα για τις περιοχές, που απεικονίζονται στο σχήμα 7.2.

Επίσης η διαδικασία που ακολουθήσαμε για να δημιουργήσουμε το υπόμνημα, είναι να επεξεργαστούμε την εικόνα του σχήματος 7.2, κρατώντας μόνο το σημείο που θέλουμε. Αφού την εισάγουμε στον φάκελο με τις υπόλοιπες εικόνες στο Unity την μετατρέπουμε σε Sprite(2D and UI) από τον inspector. Μετά μπορούμε να την προσθέσουμε στην σκηνή και να την τοποθετήσουμε στο επιθυμητό σημείο. Έτσι, είναι αντιληπτή από τον χρήστη η χρωματική σημασία των αντικειμένων που δημιουργήσαμε, και ποιο κράτος αντιπροσωπεύεται από το κάθε ένα.

Για να ολοκληρώσουμε τον χάρτη, κάνουμε χρήση του αντικείμενου TextMeshPro 3D, ώστε να τοποθετήσουμε τα ονόματα των κρατών και πόλεων, ακριβώς όπως κάναμε και στα προηγούμενα κεφάλαια.

Τέλος, για την απεικόνιση της μάχης η οποία αναφέρεται στον κείμενο θα δημιουργήσουμε ένα αντικείμενο που θα έχει δύο σπαθιά να συγκρούονται και θα είναι animated.

Για το μοντέλο του σπαθιού, ακολουθούμε την διαδικασία επεξεργασίας στο Blender που έχουμε αναφέρει. Επιπλέον αλλάζουμε το σημείο Origin του στην θέση της λαβής, από το μέσο του αντικειμένου που βρίσκεται. Αυτό το επιτυγχάνουμε τοποθετώντας τον κέρσορα του Blender στο σημείο που θέλουμε και κάνοντας δεξί κλικ, SetOrigin → Origin to 3D Cursor. Έπειτα περνάμε το διαμορφωμένο αρχείο στο Unity. Τροποποιούμε το material του, περνώντας τις εικόνες για τον χρωματισμό, metallic και normal στις αντίστοιχες επιλογές. Επιλέγουμε το rendering Mode σε Opaque και αλλάζουμε τις τιμές του Smoothness(Σχήμα 7.5).



Σχήμα 7.5 : Sword Material

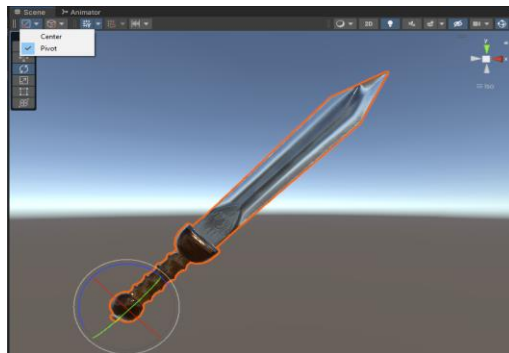
7.3.1 Clashing Animation

Στο Unity το animation λειτουργεί μεταφέροντας τα αντικείμενα που έχουμε ορίσει, από συγκεκριμένες συντεταγμένες στον χώρο την πρώτη χρονική στιγμή (αρχικό keyframe), σε διαφορετικό σετ συντεταγμένων την επομένη χρονική στιγμή που έχουμε ορίσει (επόμενο keyframe). Οι ενδιάμεσες θέσεις τις οποίες πρέπει να διαβεί το αντικείμενο υπολογίζονται από την μηχανή του Unity. Έτσι τα αντικείμενα κλειδώνουν στις θέσεις που έχουν οριστεί τις χρονικές στιγμές των keyframes. Ακόμα και αν αλλάξουμε την θέση τους στον χώρο, όταν αρχίσει η αναπαραγωγή της σκηνής μεταφέρονται από την τελευταία γνωστή τους θέση που τα έχουμε αποθηκεύσει, σε αυτήν που ορίζει το πρώτο keyframe εάν είναι ενεργοποιημένη η δυνατότητα για animation.

Αυτή η ιδιαιτερότητα μπορεί να παρακαμφθεί με την χρήση ενός νέου κενού αντικειμένου. Αφού το δημιουργήσουμε, τοποθετούμε μέσα σαν childObject το μοντέλο που θέλουμε να κάνουμε animate.

Έτσι όταν θέσουμε τις θέσεις στον χώρο που θα κλειδώσουν τα keyframes, αυτό γίνεται συγκριτικά με βάση την θέση του ParentObject, κάτω από την ιεραρχία του οποίου βρίσκεται, και όχι σχετικά με τον χώρο περιβάλλοντος της σκηνής όπως πριν. Επιπλέον αφού οι συντεταγμένες του ParentObject δεν είναι animated, μπορούμε να το τοποθετήσουμε οπουδήποτε στον χώρο της σκηνής, χωρίς αυτό να επηρεάζει το animation του ChildObject του οποίου η σχετική θέση δεν αλλάζει ουσιαστικά.

Αξιοποιώντας αυτή την τεχνική, αρχικά δημιουργούμε το νέο κενό GameObject και μέσα δημιουργούμε το πρώτο σπαθί. Στην συνέχεια αλλάζουμε το Handle Position Tool από Center σε Pivot (Σχήμα 7.6), όπου το αντικείμενο θα περιστρέφεται με άξονα το σημείο που είχαμε καθάρσει σαν origin στο Blender. Ακολουθώντας τα βήματα που έχουν αναφερθεί στο 6.4.1 Animation αντικειμένων UI, δημιουργούμε ένα νέο αρχείο .anim για το σπαθί, και αλλάζουμε τις συντεταγμένες του στον χώρο μέσα από το Rotation tool.



Σχήμα 7.6 : Sword Pivot

Στο tab του animation θέτουμε στο χρονικό σημείο 0 σαν πρώτη θέση keyframe. Για τις συντεταγμένες αλλάζουμε την κλίση στις 45 μοίρες σε έναν άξονα (εδώ προτιμήθηκε ο άξονας Z καθώς είναι ο άξονας του βάθους), που θα αποτελέσει και το σημείο επαφής. Έπειτα, επιλέγουμε στην χρονική θέση 0:30 δευτερόλεπτων ως επόμενο keyframe. Για συντεταγμένες, επιλέγουμε την θέση με περιστροφή στις 110 μοίρες στον Z άξονα, ώστε να δώσουμε την ψευδαίσθηση της κίνησης με ορμή. Τέλος, στο χρονικό σημείο του ενός δευτερόλεπτου, βάζουμε το τελικό keyframe, πάλι στην θέση με συντεταγμένες τις 45 μοίρες στον Z άξονα, για να ολοκληρωθεί ο κύκλος της κίνησης.

Κλασικά Χρόνια(ΜΑΚΕΔΟΝΙΑ)

Επόμενο βήμα όπως και πριν είναι η δημιουργία ενός Animation Controller στο οποίο φορτώνουμε το αρχείο .anim. Τέλος, όπως και στο 6.4.1 προσθέτουμε στο αντικείμενο ένα Animator component και περνάμε σαν παράμετρο το controller που μόλις δημιουργήσαμε.

Αμέσως μετά στο ίδιο άδειο αντικείμενο, προσθέτουμε ένα ακόμα αντικείμενο σπαθί στην ιεραρχία του. Μετατοπίζουμε το νέο σπαθί σε απόσταση ίση με τη μισή του μήκους του, στον άξονα X. Επαναλαμβάνουμε τα ίδια βήματα, αλλά με μοίρες τις 135 για την θέση επαφής και τις 70 για την δεύτερη θέση, δηλαδή τις συμπληρωματικές των προηγούμενων στο ημικύκλιο. Τέλος, κάνουμε και μια ελάχιστη μετατόπιση στον Z άξονα ώστε να μην έχουμε μπλέξιμο και ένωση των meshes κατά την κίνηση. Έτσι επιτυγχάνουμε με το animation την ταυτόχρονη κίνηση και φαινομενική σύγκρουσης τους.



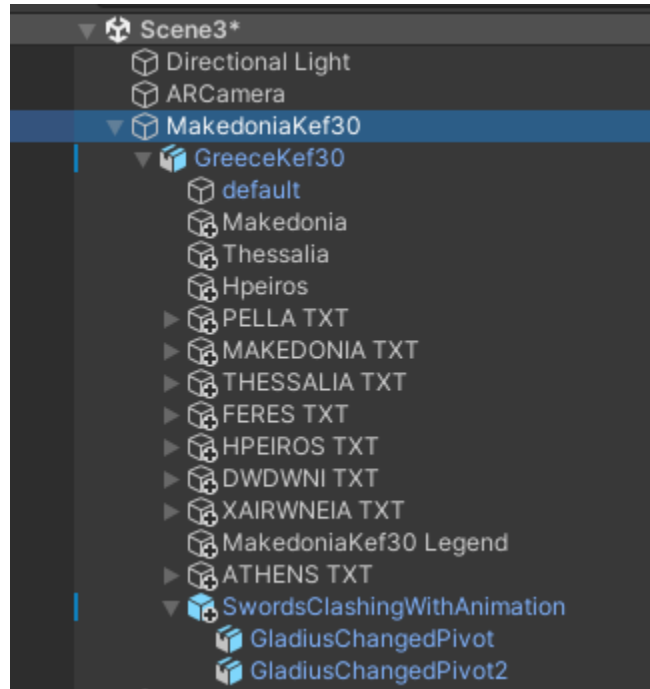
Σχήμα 7.7 : Crossed Sword

Αφού έχουμε τελειώσει με τα animation, δημιουργούμε ένα αντικείμενο prefab με τα βήματα που έχουμε αναφέρει και σώζουμε το parentObject που περιέχει τα δύο σπαθιά. Με τον τρόπο αυτό μπορούμε να το ξανά χρησιμοποιήσουμε χωρίς να χρειαστεί να κάνουμε την διαδικασία. Το προσθέτουμε στον χάρτη, στο σημείο της μάχης που αφηγούμαστε, κάνοντας έτσι πιο ενδιαφέρον το μοντέλο που βλέπει ο χρήστης. Η τελική μορφή πριν ξεκινήσει το animation φαίνεται στο Σχήμα 7.7.



Σχήμα 7.8 : Αντικείμενο για το Κεφάλαιο 30

Με αυτό τον τρόπο έχουμε ολοκληρώσει το πρώτο κομμάτι της σκηνης που αναφέρεται στο τριακοστό κεφάλαιο του βιβλίου για την Μακεδονία(Σχήμα 7.8), τα αντικείμενα για την οποία φαίνονται στο Σχήμα 7.9.



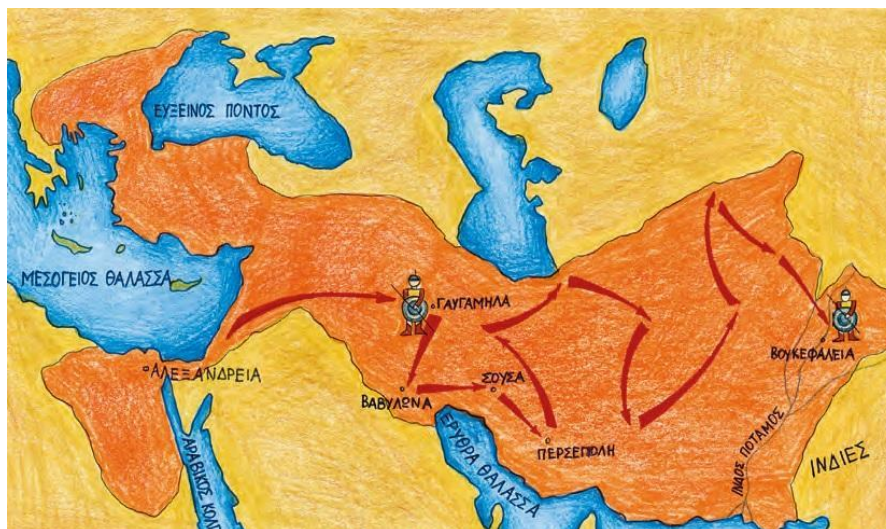
Σχήμα 7.9 : Ιεραρχία για το Κεφάλαιο 30

7.4 Δημιουργία Περιβάλλοντος Δεύτερου Image Target

Έπειτα, καθώς έχουμε ολοκληρώσει το εισαγωγικό κεφάλαιο της τρίτης ενότητας μπορούμε, να συνεχίσουμε στην δημιουργία του τρισδιάστατου κόσμου που καλύπτει τα επόμενα τρία κεφάλαια και περιλαμβάνει την εκστρατεία του Μ. Αλεξάνδρου προς την Ασία.

Αρχικά ακολουθούμε όπως και πριν την δημιουργία του Image Target και την επιλογή της εικόνας για αναγνώριση. Επιλέχθηκε ο χάρτης που είναι στην εικόνα της σελίδας 101 (Σχήμα 7.10), καθώς είναι πιο ολοκληρωμένος και είναι πιο κοντά με το περιεχόμενο και των τριών κεφαλαίων.

Στην συνέχεια εισάγουμε ένα τρισδιάστατο μοντέλο του χάρτη της περιοχής. Πρώτα επεξεργάστηκε στο Blender ανάλογα με τον πρώτο χάρτη πριν εισαχθεί σαν FBX για να είναι αναγνωρισμένο στο Unity. Σαν material χρησιμοποιήθηκε δύο φορές η ίδια φωτογραφία texture σε διαφορετικά κανάλια ώστε να επιτευχθεί πιο βαθύ χρώμα, και να είναι πιο έντονο το ανάγλυφο, ώστε να διακρίνονται καλύτερα οι υψομετρικές διαφορές.



Σχήμα 7.10 : Εικόνα Τετάρτου Image Target

Στην συνέχεια τοποθετηθήκαν με τον ίδιο τρόπο όπως και πριν κείμενα με τα ονόματα των πιο σημαντικών πόλεων που αναφέρονται στην ύλη του βιβλίου. Επιπλέον όπου αναφέρονται στο κείμενο μάχες ή συγκρούσεις προστέθηκε το αντικείμενο με τα συγκρουόμενα ξίφη, για να αναδειχθεί το σημείο που έλαβαν χώρα.

Επιπλέον προστέθηκε μια χρωματισμένη προτομή του Μ. Αλεξάνδρου, αφού πρώτα τροποποιήθηκε στο Blender όπως και πριν. Για το αντικείμενο χρησιμοποιήθηκαν εικόνες για χρώμα, normal, και occlusion για τον φωτισμό, ώστε να είναι πιο αληθοφανές. Ο χρήστης μπορεί να δει πως έμοιαζε στην πραγματικότητα το ιστορικό πρόσωπο για το οποίο μαθαίνει και να αλληλεπιδράσει μαζί του, βλέποντας το από όλες τις γωνίες. Επιπρόσθετα, και στα δύο αντικείμενα υπάρχει δυνατότητα αλληλεπίδρασης του κάθε ενός, αυτόνομα.

Επίσης έγινε προσθήκη μοντέλων για την αναπαράσταση της πορείας της εκστρατείας. Αυτά είναι ενός μακεδόνα οπλίτη, που εκτελεί κίνηση περπατήματος, και μιας μακεδονικής φάλαγγας σαρισσοφόρων. Και τα δύο αναφέρονται στο κείμενο, και μέσω της τρισδιάστατης αναπαράστασης, δίνουμε μια καλύτερη ιδέα στον μαθητή, το πως ακριβώς ήταν αυτά για τα οποία διαβάζει.

Τέλος, το animation για την πορεία των δύο παραπάνω μοντέλων έγινε σε δύο μέρη. Το πρώτο για την μετάβαση του Μ. Αλεξάνδρου από την Πέλλα, την πορεία του στην Μ. Ασία και μέχρι την Αλεξάνδρεια. Το δεύτερο σκέλος καλύπτει την αναχώρηση από την Αλεξάνδρεια, την πορεία προς την Ινδία και την επιστροφή του στην Βαβυλώνα. Η ολοκληρωμένη μορφή του χάρτη φαίνεται στο Σχήμα 7.12.

Για την υλοποίηση τους κάνουμε βήματα αντίστοιχα με αυτή του 5.3.1 τόσο για το μονοπάτι όσο και για την σύνδεση των 3D μοντέλων σε αυτό. Επίσης με τον ίδιο τρόπο, προσθέτουμε triggers, που ενεργοποιούν διάφορα 3D TMP και αντικείμενα σπαθιών, όταν τα μοντέλα περάσουν από τα συγκεκριμένα σημεία. Αυτή την φορά ωστόσο έπρεπε να κινηθούν τα δύο μοντέλα μαζί ταυτόχρονα και στα δύο

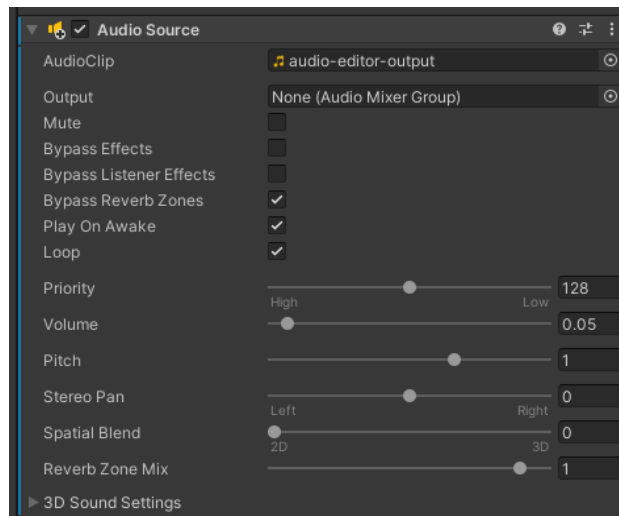
μονοπάτια, όμως διαδοχικά, αρχικά στο πρώτο και έπειτα στο δεύτερο μονοπάτι. Οπότε ήταν αναγκαίο να γίνει τροποποίηση της λογικής του κώδικα από το κεφάλαιο 5.4.1 της Π.Ε που θα αναφερθεί στο επόμενο σκέλος.

Για να συμπληρώσουμε την κίνηση της σύγκρουσης των σπαθιών προσθέτουμε τον ήχο δύο ξιφών που συγκρούονται. Το αρχείο ήταν δωρεάν και προμηθεύτηκε από το rixabay [46].

Επόμενο βήμα είναι η επεξεργασία του ώστε το χρονικό περιθώριο του ηχητικού αρχείου να συμβαδίζει με τον χρονισμό του animation, κάτι που μπορούμε να κάνουμε μέσω online tool που προσφέρουν σελίδες όπως η editor.audio [47] της οποίας έγινε χρήση.

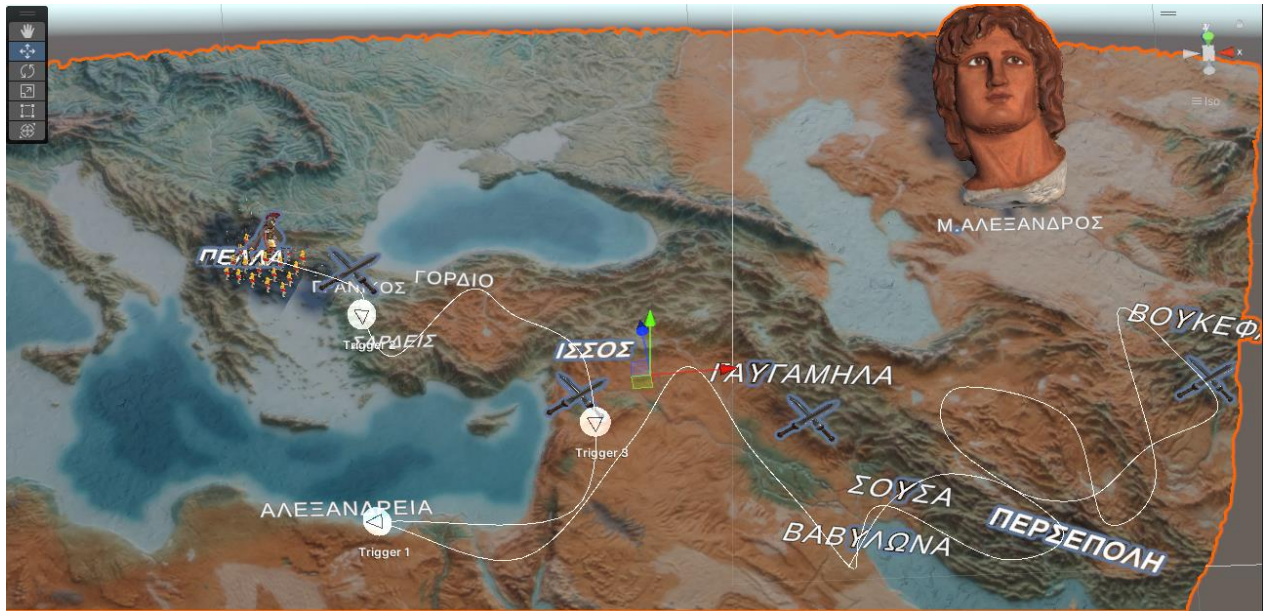
Αφού έχουμε πλέον το τελικό αρχείο ήχου, μπορούμε πλέον να το εισάγουμε στο Unity σε έναν φάκελο sounds για να το αξιοποιήσουμε. Η παραγωγή ήχων από αντικείμενα μπορεί να επιτελεστεί μέσω του component Audio Source. Το προσθέτουμε στο ParentObject όπου κάτω από την ιεραρχία του είναι τα σπαθιά που δημιουργήσαμε πριν. Μπορούμε να προσθέσουμε το αρχείο ήχου στο πεδίο AudioClip είτε σύροντας το αρχείο σε αυτό ή από το Select AudioClip Window(Σχήμα 7.11).

Τέλος, πρέπει να επιλέξουμε properties όπως, play on awake για την αναπαραγωγή μόλις ξεκινήσει η σκηνή, είτε μόλις ενεργοποιηθεί το αντικείμενο. Επίσης το property loop, ώστε να επαναλαμβάνεται η αναπαραγωγή μόλις ολοκληρωθεί η διάρκεια της. Από το component επίσης έχουμε την δυνατότητα μεταβολής παραμέτρων, όπως την ένταση του ήχου ανάμεσα σε άλλες, όπου επιλέχτηκε η τιμή 0.05 ώστε να είναι διακριτικός.



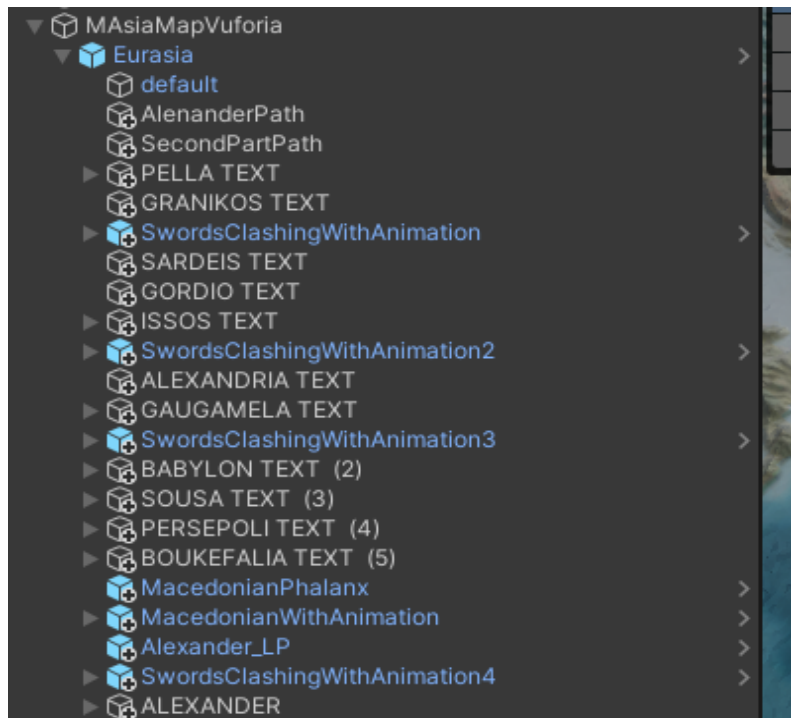
Σχήμα 7.11 : Clashing Swords Audio Controller

Καθώς ο χρήστης μπορεί να μην θέλει να ακούει τον ήχο η να τον διακόψει έπειτα από κάποια χρονική στιγμή για να συγκεντρωθεί στο κείμενο προσθέτουμε ένα toggle button για να μπορεί να γίνει σίγαση. Για την καλύτερη κατανόηση αφαιρέθηκε το text component και αντικαταστάθηκε με ένα image component μέσα από το οποίο εμφανίζεται το εικονίδιο σίγασης. Τέλος, δημιουργούμε ένα άδαιο αντικείμενο που μετονομάστηκε σε Audio Controller όπου και θα τοποθετηθεί script για να ελέγχει την λειτουργία του mute toggle button.



Σχήμα 7.12 : Αντικείμενο για τα Κεφάλαια 31-33

Η παρακάτω εικόνα (Σχήμα 7.13) δείχνει τα αντικείμενα που βλέπει ο χρήστης για τον δεύτερο σκέλος της σκηνης.



Σχήμα 7.13 : Ιεραρχία για τα Κεφάλαια 31-33

7.5 User Interface και Λειτουργικότητα

Για το κομμάτι της αλληλεπίδρασης του χρήστη, τόσο με το κείμενο όσο και με τα κουμπιά που θα ενεργοποιήσουν το animation, χρησιμοποιούμε το Canvas UI από το 5ο Κεφάλαιο, καθώς θέλουμε να διατηρήσουμε την ομοιομορφία της εφαρμογής. Κάτι τέτοιο είναι δυνατό, καθώς έχουμε αποθηκεύσει το αντικείμενο σαν prefab, οπότε δεν χρειάζεται να το ξαναδημιουργήσουμε από την αρχή. Η μόνη προσθήκη που κάναμε, ήταν το toggle button mute, που αναφέραμε στην προηγούμενη υποενότητα.

Ωστόσο μια διαφορά που έχει η σκηνή αυτή, εκτός από το ότι υπάρχουν δύο Image Targets τα οποία τα χειριζόμαστε σαν κανονικά μοντέλα για την ενεργοποίηση τους, είναι η ταυτόχρονη χρήση των δύο ίδιων path από δύο διαφορετικά αντικείμενα, διαδοχικά.

Στο νέο script, που χειρίζεται και ενεργοποιεί τα components των δύο Spline Followers ανά μοντέλο, αρχικά πρέπει να περάσουμε την βιβλιοθήκη Dreamteck Spline.

Στην συνέχεια δημιουργούμε μια μεταβλητή Gameobject, για το αντικείμενο που θέλουμε να διαχειριστούμε, και ένα array τύπου SplineFollower. Μέσα στο τελευταίο θα εισαχθούν τα component του κάθε αντικειμένου του οποίου θέλουμε να ενεργοποιήσουμε σειριακά.

Μέσα από την μέθοδο Start(), η οποία είναι πάντα η πρώτη που τρέχει μόλις αρχίσει η σκηνή, παίρνουμε την λίστα components τύπου SplineFollower από το προαναφερθέν Gameobject αντικείμενο και τις αποθηκεύουμε στο array που δημιουργήσαμε πριν.

Με αυτό τον τρόπο έχουμε πρόσβαση σε κάθε ένα από τα ίδια component και έτσι μπορούμε να τα διαχειριστούμε ξεχωριστά.

Χρησιμοποιώντας αυτήν την δυνατότητα που δημιουργήσαμε γραφούμε δύο νέες μεθόδους την StartCampaign() και την NextCampaign().

Στην πρώτη ενεργοποιούμε το πρώτο SplineFollower, για το path από την Πέλλα μέχρι την Αλεξάνδρεια, που βρίσκεται στην πρώτη θέση του πίνακα, μέσω του enabled =true.

Στην δεύτερη κάνουμε το ίδιο, αλλά για την δεύτερη θέση του πίνακα, που περιέχει το path που δημιουργήσαμε από την Αλεξάνδρεια μέχρι την Ινδία και πίσω στην Βαβυλώνα.

Το script αυτό τοποθετείται ως component στα κουμπιά “Προς Αίγυπτο” και “Προς Ινδία” από δύο φορές μια για κάθε 3D μοντέλο αντικείμενου που θα μετακινηθεί στον χάρτη, το οποίο περνάμε και σαν παράμετρο. Στην OnClick() του πρώτου κουμπιού, “Προς Αίγυπτο”, περνάμε την StartCampaign() μια φορά για κάθε αντικείμενο. Το ίδιο κάνουμε και για την NextCampaign() για το δεύτερο κουμπί, “Προς Ινδία”.

Επίσης το πρώτο κουμπί που ελέγχει την ενεργοποίηση animation ενεργοποιεί και τα μοντέλα, που σαν default είναι απενεργοποιημένα, καθώς επίσης εξαφανίζει τον εαυτό του και εμφανίζει το δεύτερο κουμπί, μέσα από την OnClick() που έχει. Μέρος του κώδικα από το script φαίνεται στο Σχήμα 7.14.

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Dreamteck.Splines;

public class AlexanderCampaignPaths : MonoBehaviour
{
    [SerializeField]
    //public GameObject SplineFollower;
    public GameObject objectToCheck;
    SplineFollower[] thePaths;

    // Start is called before the first frame update
    void Start()
    {
        //SplineFollowers[] SplineFollowers;
        //SplineFollowers = SplineFollower.GetComponent<SplineFollower>();
        thePaths = objectToCheck.GetComponents<SplineFollower>();
    }

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

    public void StartCampaign()
    {
        thePaths[0].enabled = true;
    }

    public void NextCampaign()
    {
        thePaths[1].enabled = true;
    }
}

```

Σχήμα 7.14 : Κώδικας για Ενεργοποίηση Animation

Για να σταματήσουμε το animation περπατήματος του στρατιώτη, το οποίο και έχει επιλεγεί να επαναλαμβάνεται ώστε να είναι διαρκής η κίνηση, αρκεί να απενεργοποιήσουμε το Animator component μέσω script(Σχήμα 7.15). Το script αυτό προστίθεται και στα δύο path και ενεργοποιείται με ένα trigger point τοποθετημένο στο τελευταίο σημείο της διαδρομής. Έτσι, αποδίδεται μια πιο φυσική κίνηση του μοντέλου πάνω στον χάρτη.

```

public void EndTheAnimation()
{
    AnimationObject.GetComponent<Animator>().enabled = false;
}

```

Σχήμα 7.15: Κώδικας για Σταμάτημα Animation

Σε αυτό το σημείο έχοντας ολοκληρώσει το κομμάτι που βλέπει ο χρήστης, μπορούμε να συμπληρώσουμε και να ενώσουμε το κομμάτι το οποίο λαμβάνει τις πληροφορίες και αλληλεπιδρά με την εφαρμογή. Έτσι, προσθέτουμε από τον φάκελο με τα prefabs το CanvasUI που περιέχει τα TMP texts και τα κουμπιά.

Έπειτα προσθέτουμε το νέο toggle button για την σίγαση, μέσα στην ιεραρχία του CanvasUI, και το τοποθετούμε δεξιά από το κουμπί για την επομένη σελίδα ώστε να είναι εύκολα προσβάσιμο.

Η διαδικασία εισαγωγής και διαμόρφωσης του κειμένου που έχει συμπτυκνωθεί είναι αντίστοιχη με αυτή των προηγούμενων σκηνών. Η αλλαγή των σελίδων γίνεται μέσα από τα κουμπιά για την επομένη και προηγούμενη σελίδα μέσω των κατάλληλων scripts που έχουμε αναφέρει. Σε αυτά, περνάμε επιπλέον μαζί τα scripts που χειρίζονται την ενεργοποίηση των αντικειμένων της σκηνής. Η ενεργοποίηση τους αυτή, γίνεται με βάση συγκεκριμένες σελίδες του κειμένου, που διαβάζει ο χρήστης.



Σχήμα 7.16 : UI Τρίτης Σκηνής

Με την πλήρη λειτουργικότητα του UI και των κουμπιών (Σχήμα 7.16), επόμενο βήμα είναι η ενασχόληση με το μικρό ερωτηματολόγιο που θα εμφανίζεται σε συγκεκριμένα σημεία της πορείας των μοντέλων. Αρχικά δημιουργούμε ένα νέο Canvas και μέσα τοποθετούμε δύο UI TMP. Το ένα θα χρησιμοποιηθεί για το στατικό κείμενο μηνύματος “Ας κάνουμε ένα QUIZ” στο οποίο αλλάζουμε το χρώμα.

Στο δεύτερο θα εμφανίζεται δυναμικά το κείμενο της κάθε ερώτησης. Ακόμα, προσθέτουμε δύο κουμπιά UI Button, στα οποία θα εισάγουμε δυναμικά τις απαντήσεις. Τέλος, βάζουμε από ένα νέο trigger point σε κάθε path και το τοποθετούμε περίπου στην μέση της διαδρομής, το οποίο θα ενεργοποιεί το Canvas quiz, θα απενεργοποιεί το CanvasUI του κειμένου και θα κάνει παύση τα animation. Η λειτουργικότητα αυτή πραγματοποιείται από ένα νέο C# script, το QuizForScene3.

Μέσα σε αυτό δημιουργούμε τέσσερις πίνακες τύπου string, οι δυο από τους οποίους είναι για τις ερωτήσεις. Οι άλλοι δυο θα έχουν τις απαντήσεις και είναι διασδιάστατοι, για να μπορέσουμε να καλύψουμε από δυο απαντήσεις για κάθε ερώτηση. Ακόμα, φτιάχνουμε άλλους δυο πίνακες τύπου int, όπου έχουμε τις θέσεις των σωστών απαντήσεων. Επίσης, επιπλέον μια λίστα τύπου button, για να μπορούμε να αριθμήσουμε και να ξεχωρίσουμε τα κουμπιά, καθώς και μια λίστα τύπου string. Σε αυτήν θα τοποθετήσουμε το UI TMP που διαθέτουν τα κουμπιά ώστε να εμφανιστεί το κείμενο των απαντήσεων.

Για την παύση των animation ο κώδικας είναι ανάλογος με αυτόν για την εκκίνηση, με την διαφορά ότι αυτή την φορά γίνεται χρήση της .enable= false. Τα animation ξαναξεκινάνε, όταν η μεταβλητή αλλάζει σε true, μόλις το quiz ολοκληρωθεί.

Όσον αφορά την εμφάνιση των ερωταπαντήσεων, αυτό γίνεται με την μέθοδο foreach που διατρέχει τον πίνακα με τις απαντήσεις με την βοήθεια δεικτών. Εμφανίζουμε την πρώτη διάσταση του πίνακα στο ένα κουμπί, και την δεύτερη στο άλλο. Έπειτα εμφανίζουμε το κείμενο για την ερώτηση, που βρίσκεται στην αντίστοιχη γραμμή του πίνακα των ερωτήσεων. Τέλος, αυξάνουμε τους δείκτες κατά μια μονάδα.

Κάθε φορά που πατάμε ένα από τα δύο κουμπιά, συγκρίνουμε τον αριθμό της θέσης του στην λίστα των κουμπιών με τον αριθμό που έχει ο πίνακας των σωστών απαντήσεων. Αυτό γίνεται μέχρι εμφανιστούν όλες οι γραμμές του πίνακα των ερωτήσεων.

Σε περίπτωση σωστής επιλογής το κουμπί αλλάζει χρώμα σε πράσινο και αυξάνουμε τον μετρητή σωστών απαντήσεων κατά ένα, ενώ σε λανθασμένη το χρώμα γίνεται κόκκινο. Έπειτα, το χρώμα των κουμπιών ξανά επιστρέφει σε λευκό και εμφανίζεται το κείμενο με τις επιλογές της επόμενης ερώτησης.

Η ίδια διαδικασία επαναλαμβάνεται με παρόμοια βήματα για το δεύτερο quiz της σκηνής σε μια άλλη μέθοδο με μόνη διαφορά τον αριθμό των ερωτήσεων. Ωστόσο η εμφάνιση των ερωτήσεων καθώς και των απαντήσεων που έχουν σαν επιλογή γίνεται στα ίδια UI TMP και Button όπως και πριν.

Αυτό το καταφέρνουμε με μια boolean μεταβλητή, την IsActive, με την οποία όσο είναι true η μέθοδος OnClick() καλεί με την σειρά της την πρώτη μέθοδο που δημιουργήσαμε. Μόλις τελειώσει με όλες τις ερωτήσεις που πρέπει να εμφανίσει, αλλάζει την μεταβλητή σε false και άρα από την επομένη φορά η OnClick() θα καλέσει με την σειρά της, την δεύτερη μέθοδο που δημιουργήσαμε. Ο κώδικας φαίνεται στο Σχήμα 7.17.

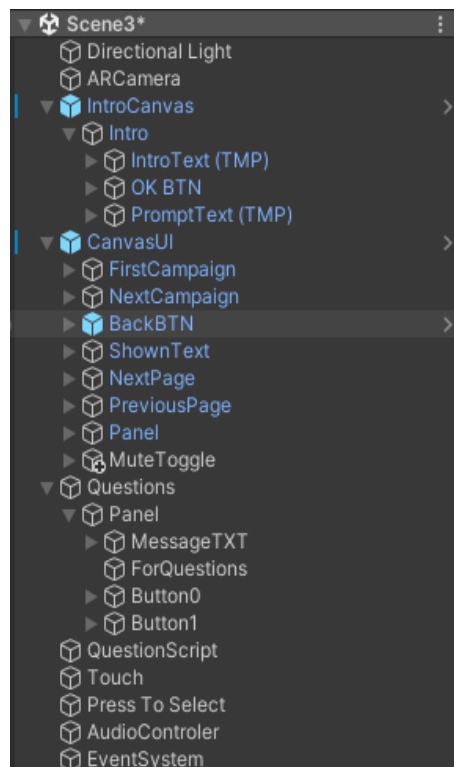
```

public void CalledOnClick(int clickedChoice)
{
    if(IsActive)
    {
        PressedChoisePath1(clickedChoice);
    }
    else
    {
        PressedChoisePath2(clickedChoice);
    }
}

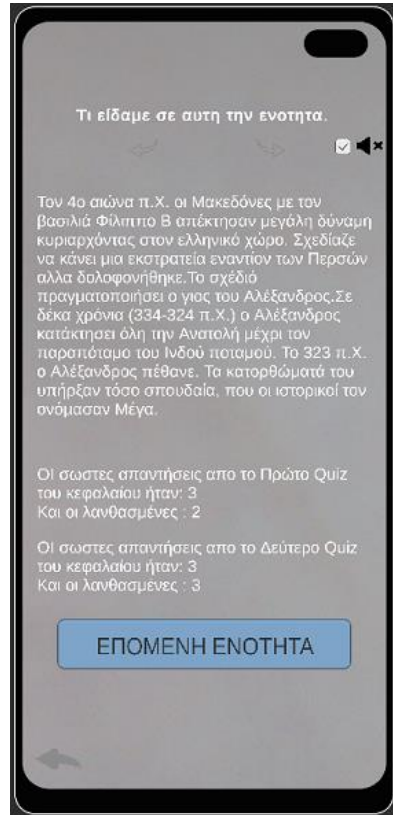
```

Σχήμα 7.17 : Μέθοδος που καλούν τα UI κουμπιά στο quiz

Στο Σχήμα 7.18 μπορούμε να δούμε τα αντικείμενα τα οποία είναι υπεύθυνα για την λειτουργικότητα της σκηνής. Όπως και στο προηγούμενο κεφάλαιο η τελευταία σελίδα του κειμένου ενεργοποιεί το Panel ανασκόπησης (Σχήμα 7.19). Αυτό περιέχει πληροφορίες για την ενότητα, το σύνολο των σωστών και λανθασμένων απαντήσεων του προαναφερθέντος quiz, και στο κάτω μέρος ένα κουμπί μέσω του οποίου προτρέπουμε τον χρήστη να μεταβεί στην επόμενη σκηνή.



Σχήμα 7.18 : Ιεραρχία UI Τρίτης Σκηνής



Σχήμα 7.19 : Ανασκόπηση Τρίτης Σκηνής

7.6 Επίλογος

Στο παρόν κεφάλαιο αναλύθηκε η υλοποίηση της τρίτης σκηνής της εφαρμογής η οποία καλύπτει το κομμάτι της Μακεδονίας από την τρίτη ενότητα του σχολικού βιβλίου. Έγινε χρήση animation 3D μοντέλων με αναφορά στον τρόπο χειρισμού και λειτουργίας τους. Τέλος, προστέθηκε η δυνατότητα παραγωγής ήχου καθώς και σίγασής του.

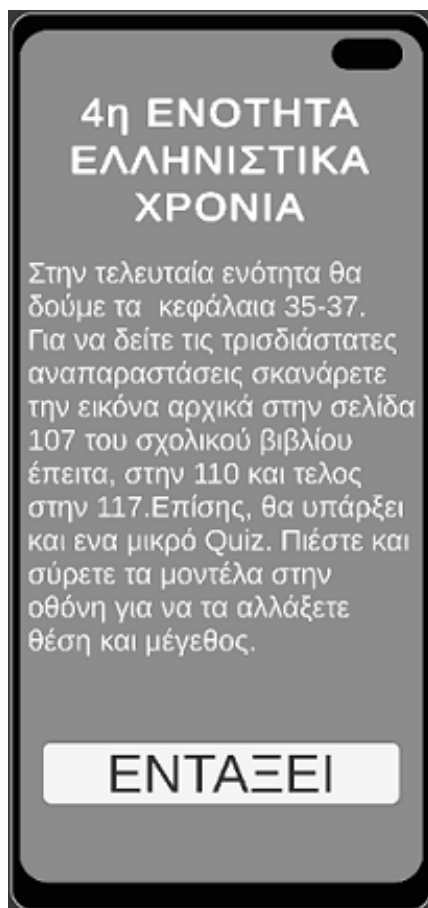
Κεφάλαιο 8ο : Ελληνιστικά Χρόνια - Οι Διάδοχοι του Μ.Αλεξάνδρου

8.1 Εισαγωγή

Στο όγδοο κεφάλαιο θα γίνει αναφορά στην ανάπτυξη της τέταρτης ενότητας του σχολικού βιβλίου της Ιστορίας, και συγκεκριμένα της πρώτης υποενότητας “Διάδοχοι του Μ. Αλεξάνδρου”. Αφορά την χρονική ιστορική περίοδο διαδοχής της αυτοκρατορίας του Μ. Αλεξάνδρου και την ζωή σε αυτήν. Όπως και πριν και σε αυτό το κεφάλαιο έγινε χρήση τριών διαφορετικών Image Targets καθώς το περιεχόμενο της ύλης κάνει αναφορά σε διαφορετικές περιοχές και θέματα. Προτιμήθηκε λοιπόν ο διαχωρισμός τους για μεγαλύτερη σαφήνεια. Κατά την διάρκεια διεξαγωγής του animation στο δεύτερο αντικείμενο, εμφανίζουμε ένα μικρό quiz για τον χρήστη, με ερωτήσεις του κεφαλαίου.

8.2 Panel Εισαγωγής

Κρατώντας το μοτίβο ροής της εφαρμογής, αρχικά μόλις φορτωθεί η σκηνή για παρουσίαση δείχνουμε το εισαγωγικό πάνελ (Σχήμα 8.1). Με αυτό πληροφορούμε τον χρήστη για την ύλη που θα καλυφτεί, το περιεχόμενο καθώς και ποιες εικόνες πρέπει να σκανάρει.

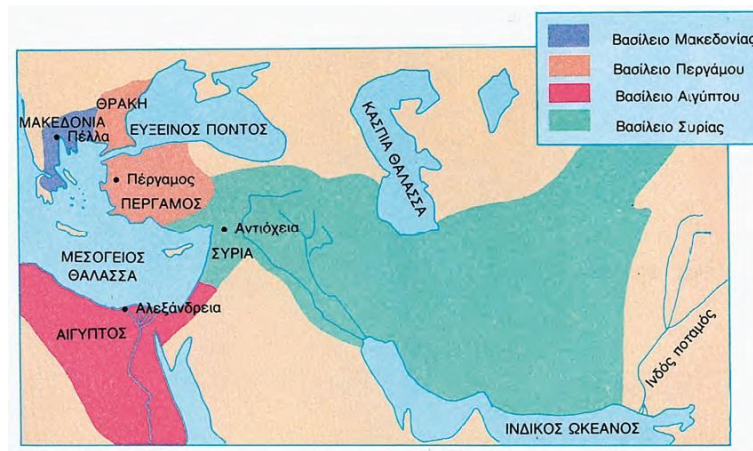


Σχήμα 8.1 : Τέταρτο Intro Panel

8.3 Δημιουργία Περιβάλλοντος

Όπως και πριν, χρειάζεται αρχικά να ακολουθήσουμε τα βήματα που αναφέρθηκαν στην ενότητα 3.4.3 Vuforia Image Target, και να αντικαταστήσουμε την κάμερα με αυτή που παρέχει το Vuforia Engine.

Η εικόνα που χρησιμοποιήθηκε για το πρώτο Image Target μέσα από την βάση είναι η ακόλουθη και φαίνεται στο Σχήμα 8.2.



Σχήμα 8.2 : Εικόνα Πέμπτου Image Target

Αφού έχουμε πλέον την βάση, μπορούμε να τοποθετήσουμε κάτω από την ιεραρχία της το μοντέλο του χάρτη που θα χρησιμοποιήσουμε. Επιλέχθηκε να ξαναχρησιμοποιηθεί το ίδιο με την υποενότητα 7.4 του προηγούμενου κεφαλαίου, καθώς η περιοχή που γίνεται αναφορά στην ύλη είναι η ίδια με πριν, και έχουμε ήδη επεξεργαστεί το 3D μοντέλο και το material του.

Για να γίνει ανάδειξη των διαφόρων βασιλείων στα οποία χωρίστηκε η αυτοκρατορία του Μ. Αλεξάνδρου, κάνουμε χρήση της ίδιας τεχνικής μέσω του ProBuilder, καθώς με τον τρόπο αυτό είναι σαφές οι περιοχές που θέλουμε να δείξουμε, χωρίς να επηρεάζουμε το μοντέλο κάθε αυτό.

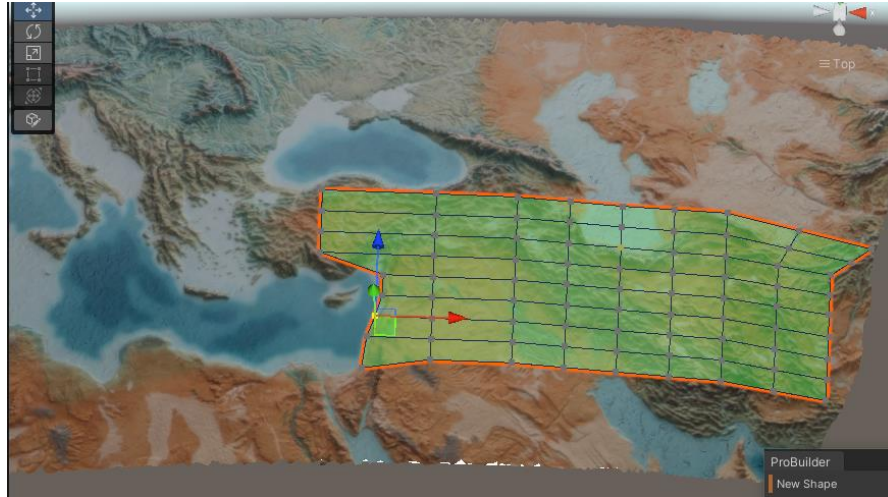
Όπως και στο 7.3 ακολουθούμε τα όρια που αναγράφονται στο σχήμα και δημιουργούμε τα νέα αντικείμενα ιχνηλατώντας τα με σημεία. Μόλις ολοκληρωθεί το περίγραμμα, και το τελευταίο σημείο τοποθετηθεί πάνω στο πρώτο, το αντικείμενο έχει ολοκληρωθεί. Παράδειγμα της διαδικασίας φαίνεται στο Σχήμα 8.3

Με στόχο να έχουμε καλύτερο έλεγχο του αντικειμένου όταν το μετακινήσουμε ακριβώς πάνω από το σχήμα του χάρτη για να το επικαλύψουμε, κάνουμε χρήση του SubdivideObject. Έτσι, δημιουργούμε περισσότερα Vertices και έχουμε πιο λεπτομερή χειρισμό. Έπειτα, μέσω της επιλογής Smoothing λειαίνουμε την επιφάνεια για καλύτερο οπτικό αποτέλεσμα.

Τέλος, αλλάζουμε το material από το default σε αυτό που έχουμε δημιουργήσει στον φάκελο SeeThroughMaterials από το 7.3, ώστε να αντικατοπτρίζει το χρώμα του, αυτό του βασιλείου στον χάρτη. Έτσι, πετυχαίνουμε να υπάρχει συνοχή μεταξύ του βιβλίου και της εφαρμογής.

Αντίστοιχα δημιουργούμε και τα υπόλοιπα αντικείμενα για τα άλλα τρία βασίλεια.

Τελική κίνηση είναι η τοποθέτηση 3D TMP τα οποία χρησιμοποιούνται σαν λεζάντες για την εμφάνιση της ονομασίας καθενός από αυτά. Αλλάζουμε το χρώμα των γραμμάτων ώστε να ταιριάζουν στην περιοχή που αντιστοιχούν (Σχήμα 8.4).



Σχήμα 8.3 : Δημιουργία Αντικείμενου για το Βασίλειο της Συρίας



Σχήμα 8.4: Τελική Μορφή Πρώτου Χάρτη

8.4 Δημιουργία Περιβάλλοντος Δεύτερου Image Target

Για το επόμενο τμήμα της σκηνής, ακολουθούμε πάλι την ίδια διαδικασία για την εισαγωγή ImageTarget σαν νέο αντικείμενο, και ορίζουμε την ακόλουθη εικόνα (Σχήμα 8.5) από την βάση για εμφάνιση. Για το μοντέλο του χάρτη χρησιμοποιήθηκε το αντίστοιχο με αυτό του έκτου κεφαλαίου της Π.Ε, το οποίο επεξεργάστηκε αρχικά στο Blender πριν περαστεί στο Unity λόγω της ανάγκης για εμφάνιση συγκεκριμένης γεωγραφικής περιοχής. Για την χρήση material χρησιμοποιήθηκε το ήδη υπάρχον το οποίο έχουμε δημιουργήσει.



Σχήμα 8.5:Εικόνα Έκτου Image Target

Επομένη κίνηση είναι η τοποθέτηση μέσω του 3D TMP πάνω στον χάρτη, των ονομάτων των περιοχών ενδιαφέροντος για την μέγιστη γεωγραφική κατανόηση της περιοχής.

Στην συνέχεια για την δημιουργία και προσθήκη των 3D μοντέλων χρησιμοποιούμε την ίδια μέθοδο που αναφέρθηκε για την δημιουργία περιβάλλοντος στο 5.3 της Π.Ε, έχοντας σαν βάση τα εικονογραφημένα μοντέλα της εικόνας. Για την επιπλέον επεξεργασία των μοντέλων έγινε χρήση επίσης των ιδίων εργαλείων Flatten, Scrape, DrawSharp, Grab και το TexturePaint.

Στην αναπαράσταση της πορείας της εκστρατείας του Πύρρου χρησιμοποιήθηκε το πακέτο DreamteckSplines όπως και πριν. Το μονοπάτι αποτελείται από δώδεκα σημεία, και τρία trigger points για την ανάδειξη των μαχών καθώς το μοντέλο του στρατιώτη περνάει από τα σημεία. Επιπλέον κάνουμε χρήση του μοντέλου των σπαθιών από την προηγούμενη σκηνή όπως είδαμε στο 7.3.1 και χρήση ηχητικού εφέ ώστε να υπάρχει πλήρης κατανόηση ότι πρόκειται για σημείο μάχης. Για την ολοκλήρωση αυτού του κομματιού, προσθέτουμε όπως και πριν στο άδειο audio controller το script για την σίγαση του ηχητικού εφέ.

Τέλος, προσθέτουμε το κύριο μοντέλο το οποίο θα διατρέξει και το μήκος του μονοπατιού. Καθώς διαθέτει επιπλέον εσωτερικό animation βήματος χρειάζεται να δημιουργήσουμε ένα animation controller. Σε αυτό θα εισάγουμε το υπάρχον αρχείο κίνησης, και θα υλοποιηθεί μέσω της επιπλέον εισαγωγής ενός animator component στον inspector του αντικειμένου. Στο πεδίο controller από το προαναφερθέν component βρίσκουμε και εισάγουμε το animation controller. Στο τρίτο trigger point που βρίσκεται και στο τελικό σημείο της διαδρομής, προσθέτουμε επιπλέον το script που δημιουργήσαμε για την διακοπή της διεξαγωγής του animation που έχουμε αναφέρει στο Σχήμα 7.14. Σαν παράμετρο περνάμε το συγκεκριμένο αντικείμενο.

Το κομμάτι υλοποίησης του quiz έχει αντίστοιχη λογική με αυτό του έβδομου κεφαλαίου της Π.Ε και ο κώδικας βρίσκεται στο script QuizForScene4. Η διαφορά είναι, ότι το quiz δεν είναι συνεχόμενο όπως πριν, αλλά διεξάγεται σε δυο μέρη.

Σε συγκεκριμένο σημείο του μονοπατιού, τοποθετούμε ένα trigger, το οποίο ενεργοποιεί αντίστοιχο Canvas που περιέχει τα UI στοιχεία του quiz, και σταματά τα animation κίνησης. Στην τρίτη ερώτηση αυτό απενεργοποιείται χωρίς να έχει ολοκληρωθεί το Quiz. Τα animation κίνησης και περπατήματος του στρατιώτη ξανά συνεχίζουν κανονικά προς τον τερματισμό της πορείας. Μόλις ολοκληρώσει την διαδρομή το Canvas Quiz ξαναεμφανίζεται δείχνοντας την επόμενη ερώτηση, και μόλις απαντηθούν και οι υπόλοιπες ερωτήσεις απενεργοποιείται. Τότε η ροή της σκηνής συνεχίζεται κανονικά. Η τελική μορφή του χάρτη φαίνεται στο Σχήμα 8.6.



Σχήμα 8.6: Τελική Μορφή Δευτέρου Χάρτη

8.5 Δημιουργία Περιβάλλοντος Τρίτου Image Target

Όπως και στα δυο προαναφερθέντα υποκεφάλαια, ακολουθούμε την ίδια διαδικασία για το Image Target με εικόνα αυτή στο Σχήμα 8.5.

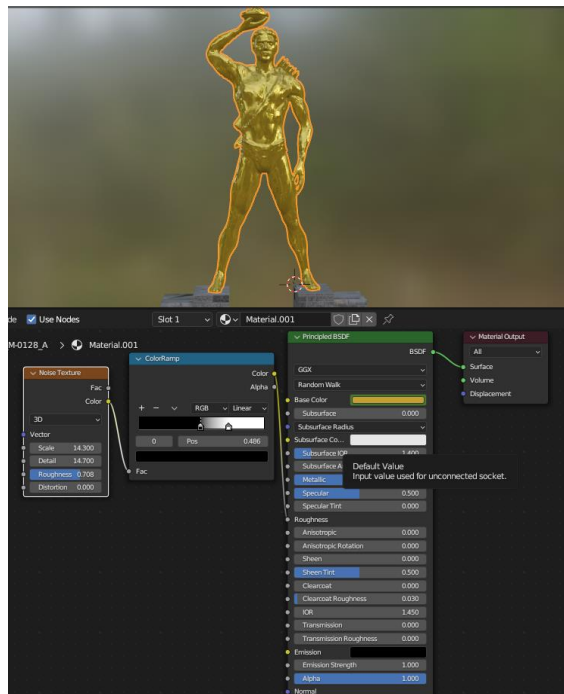


Σχήμα 8.7:Εικόνα Εβδομού Image Target

Ελληνιστικά Χρόνια - Οι Διάδοχοι του Μ.Αλεξάνδρου

Το μοντέλο για τον Κολοσσό της Ρόδου που χρησιμοποιήθηκε βρέθηκε από το Thingiverse[40], αλλά καθώς δεν είχε κάποιο texture, δημιουργήθηκε για αυτό στο Blender και έπειτα έγινε εξαγωγή ως FBX. Αρχικά μέσω του Decimate Modifier μειώθηκαν τα τρίγωνα, ώστε να είναι πιο διαχειρίσιμη η επεξεργασία του. Έπειτα έγινε διαχωρισμός του μοντέλου σε δύο τμήματα, ένα του αγάλματος και δεύτερο της βάσης.

Επόμενο βήμα είναι η δημιουργία νέου material από το Shading tab (Σχήμα 8.5). Μέσω του BaseColor γίνεται η επιλογή του χρώματος, όπου και επιλέγεται μια χρυσή απόχρωση. Για το πιο φυσικό χρώμα του ορείχαλκου προσθέτουμε ένα Noise Texture και ColorRamp nodes, τα οποία συνδέονται στο Roughness. Πειραματιζόμαστε με τις τιμές μέχρι να βρούμε ένα επιθυμητό αποτέλεσμα όπου τελικά και σώζουμε σαν material του πρώτου κομματιού.



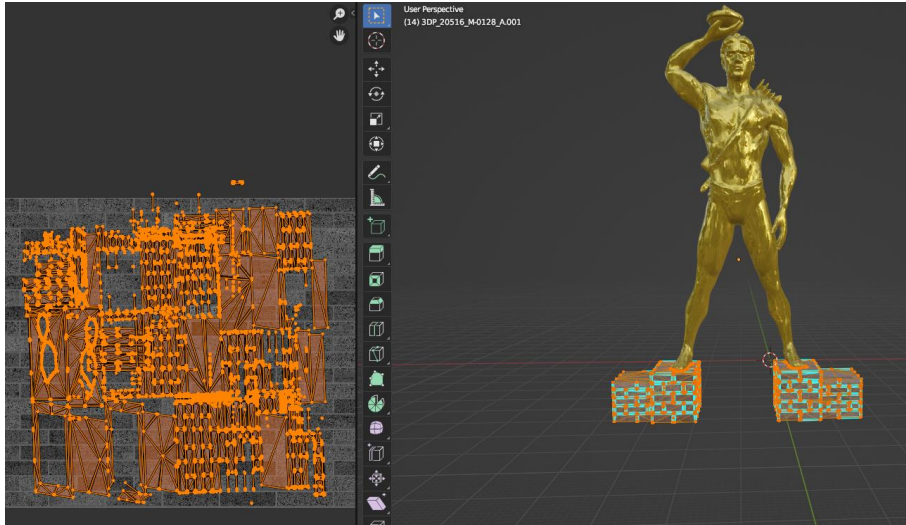
Σχήμα 8.8:Χρωματισμός του Κολοσσού

Για την βάση επιλέχτηκε η χρήση μιας εικόνας από πλίνθινο τοίχο που βρέθηκε στο διαδίκτυο. Έγινε επιλογή του αντικειμένου της βάσης, και μετά επιλέγουμε Edit Mode UV → Smart UV Project ώστε να κάνει unwrap την επιφάνεια του.

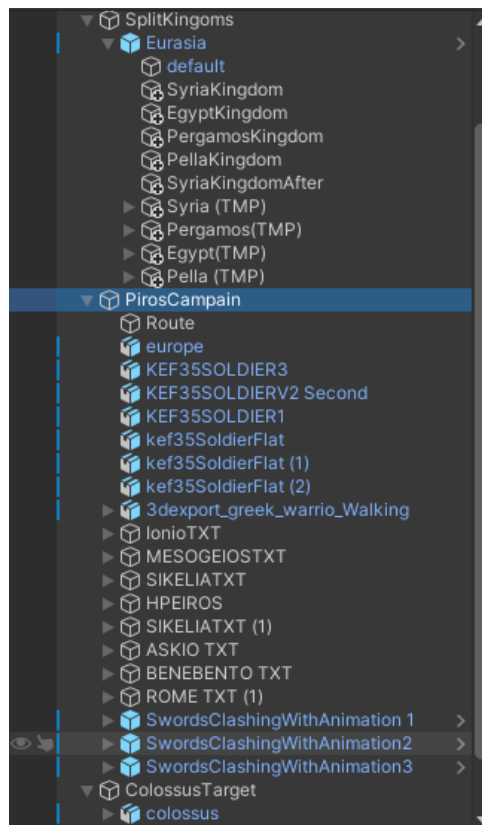
Έπειτα από την καρτέλα UV Editing (Σχήμα 8.6) προστέθηκε και επιλέχτηκε σαν εικόνα φόντου η εικόνα που κατεβάσαμε. Μέσω της επιλογής των vertices από το Modeling tab, παρατηρούμε ότι γίνεται και αντίστοιχη επιλογή των vertices που έχουν ξεδιπλωθεί πάνω στην εικόνα. Επίσης παρατηρούμε ότι μπορούμε να τα μετακινήσουμε. Έτσι τοποθετούμε τα σημεία που επιλέγουμε σε συγκεκριμένες περιοχές της εικόνας, ώστε να προβληθεί πάνω τους ο χώρος της εικόνας που καταλαμβάνουν. Η προαναφερθείσα διαδικασία συνεχίζεται μέχρι να είναι ικανοποιητική η τελική προβολή της εικόνας στο 3D αντικείμενο της βάσης.

Κεφάλαιο 8ο

Τέλος, σώζουμε το material που δημιουργήσαμε και μπορούμε να εξάγουμε το μοντέλο με την διαδικασία που έχουμε αναφέρει σε προηγούμενα κεφάλαια. Όπως και πριν ο χρήστης μπορεί να αλληλεπιδράσει μαζί του και να το παρατηρήσει από διάφορες γωνίες. Το ίδιο γίνεται και με τα προηγούμενα αντικείμενα που δημιουργήσαμε σε αυτό το κεφάλαιο, κάνοντας χρήση των script που έχουμε αναφέρει.



Σχήμα 8.9:UV Editing της Βάσης



Σχήμα 8.10 : Ιεραρχία Μοντέλων Σκηνής

8.6 User Interface

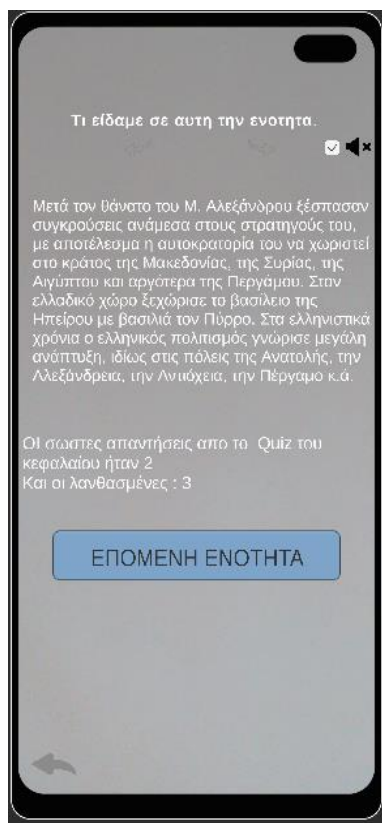
Με το παραπάνω υποκεφάλαιο, έχουμε συμπληρώσει το οπτικό κομμάτι της σκηνης, που περιέχει τα αντικείμενα τα οποία βλέπουμε στο Σχήμα 8.7. Μπορούμε να συνεχίσουμε με το κομμάτι της εμφάνισης της ύλης και την λειτουργικότητα της σκηνης.

Όπως και πριν προσθέτουμε από το prefab το CanvasUI που περιέχει τα διάφορα κουμπιά και UI TMP για την εμφάνιση του κειμένου μας. Επίσης περνάμε στο toggle mute button, το Audio Controller στην OnValueChanged().

Η διαμόρφωση των στοιχείων στον καμβά καθώς και η ιεραρχία του είναι αντίστοιχη με αυτή των προηγούμενων σκηνών, με την διαφορά ότι υπάρχει μόνο ένα κουμπί για εκκίνηση animation, στο δεύτερο Image Target, στην παρούσα σκηνή.

Το κείμενο είναι διαμορφωμένο σε οχτώ σελίδες που έγιναν μέσω της χρήσης rich text tag <page> . Από αυτές η τρίτη και η πέμπτη χρησιμοποιούνται σαν σημεία για την ενεργοποίηση και απενεργοποίηση διαφόρων στοιχείων της ιεραρχίας. Η ενεργοποίηση των scripts, που έχουν αναφερθεί, γίνεται και πάλι μέσω των κουμπιών επομένης και προηγούμενης σελίδας.

Με το πέρας του συνόλου των σελίδων ενεργοποιούμε το outro Panel(Σχήμα 8.8) και μετά από μια μικρή αναδρομή του κεφαλαίου προτρέπουμε τον χρήστη να μεταβεί στο επόμενο κεφάλαιο που αποτελεί και το τελευταίο της εφαρμογής. Επιπλέον σε αυτό δείχνουμε το πλήθος των σωστών και λανθασμένων απαντήσεων από τις ερωτήσεις του Quiz.



Σχήμα 8.11 : Ανασκόπηση Τέταρτης Σκηνης

8.7 Επίλογος

Μέσα από το κεφάλαιο που μόλις ολοκληρώσαμε, είδαμε την ροή δημιουργίας της τέταρτης σκηνής της εφαρμογής. Περιέχει τμήμα της ύλης από την ενότητα τέσσερα του σχολικού βιβλίου. Επίσης κάναμε αναφορά στον τρόπο δημιουργίας materials στο Blender μέσω UV Editing και Shading.

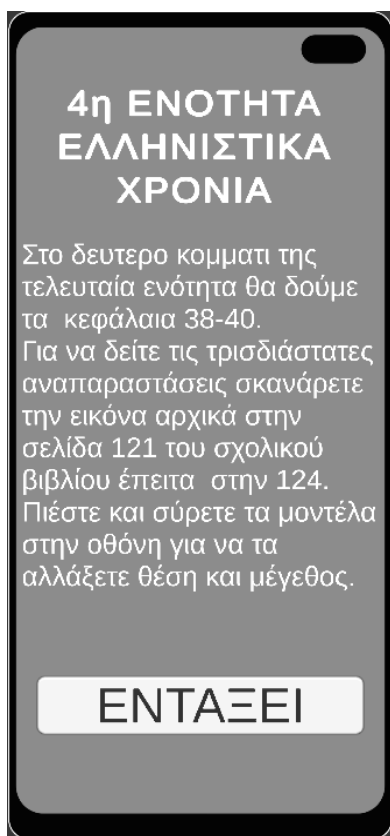
Κεφάλαιο 9ο : Ελληνιστικά Χρόνια - Ρώμη και Ελλάδα

9.1 Εισαγωγή

Στο ένατο κεφάλαιο θα δούμε την υλοποίηση της τελευταίας σκηνής, η οποία βασίζεται στην ύλη του δευτέρου μισού της τέταρτης ενότητας του σχολικού βιβλίου, με θέμα την Ρώμη και την Ελλάδα. Η δομή της σκηνής ακολουθεί το μοτίβο που έχουμε ορίσει από την σκηνή στο πέμπτο κεφάλαιο, και χρησιμοποιούμε δυο Image Targets για την παρουσίαση της ύλης. Επιπρόσθετα, στον τελευταίο χάρτη τα αντικείμενα των στρατιωτών και σπαθιών, λειτουργούν ως κουμπιά. Πατώντας τα εμφανίζουμε κείμενο σχετικό με συγκεκριμένη μάχη. Υπάρχει πλήρης αλληλεπίδραση με τα αντικείμενα όπως και πριν.

9.2 Panel Εισαγωγής

Ξεκινάμε με την δημιουργία του πάνελ εισαγωγής (Σχήμα 9.1) όπως και πριν. Μέσα από αυτό υποδεικνύουμε την σελίδα στην οποία βρίσκονται οι εικόνες-στόχοι, για την επαύξηση τους και εμφάνιση των 3D μοντέλων. Αντίστοιχα με όλα τα προηγούμενα, με το πάτημα του κουμπιού το χρώμα του αλλάζει σε πράσινο για την γνωστοποίηση της εντολής στο χρήστη. Μετά την απενεργοποίηση του εμφανίζεται το πρώτο μοντέλο-χάρτης.



Σχήμα 9.1: Πέμπτο Intro Panel

9.3 Δημιουργία Περιβάλλοντος

Η πρώτη ενέργεια είναι να προσθέσουμε το ImageTarget και την AR Camera όπως έχουμε αναφέρει, με επιλεγμένη την εικόνα του ακόλουθου σχήματος (Σχήμα 9.2) από την βάση.



Σχήμα 9.2 : Εικόνα Ογδού Image Target

Στην συνέχεια προσθέτουμε στην ιεραρχία του ImageTarget ως ChildObject, τον χάρτη που θα αποτελέσει βάση για την παρουσίαση του πρώτου κομματιού της ύλης. Χρησιμοποιούμε το ίδιο μοντέλο με το κεφάλαιο 8.4, όμως μετά την επεξεργασία του στο Blender, έχουμε μόνο την περιοχή της δυτικής πλευράς της Ευρώπης.

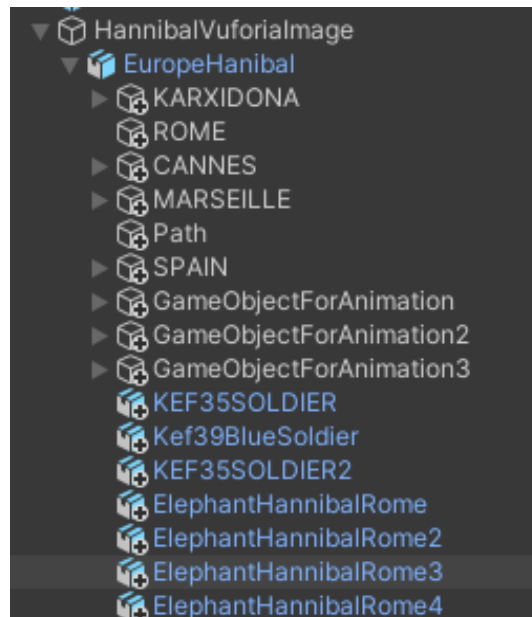
Για την οπτική μορφοποίηση του χρησιμοποιούμε το ίδιο material που δημιουργήσαμε στο προαναφερθέν κεφάλαιο, καθώς πρόκειται για το ίδιο μοντέλο, απλά διαμορφωμένο για άλλη γεωγραφική περιοχή. Στην συνέχεια κάνουμε χρήση του 3D TMP για την τοποθέτηση των ονομάτων των διαφόρων περιοχών που υπάρχουν στον χάρτη του παραπάνω σχήματος. Ο χάρτης ολοκληρώνεται, με την δημιουργία και προσθήκη των εικονιζόμενων μοντέλων στρατιωτών και ελεφάντων που βλέπουμε. Καθώς τα μοντέλα των στρατιωτών έχουν ήδη δημιουργηθεί από προηγούμενες ενότητες χρειάζεται να ασχοληθούμε μόνο με τους ελέφαντες. Κάνουμε χρήση αντίστοιχα του MonsterMash, με την ίδια μέθοδο για δημιουργία του αντικειμένου, όπως προαναφέραμε στο 5ο Κεφάλαιο. Επιπλέον το επεξεργαζόμαστε στο Blender με τα εργαλεία που αναφέραμε στο 8.4 για βελτίωση της εμφάνισης.

Η απεικόνιση της πορείας του Αννίβα που διαγράφεται στην εικόνα, μεταφέρεται στο μοντέλο του χάρτη, με δημιουργία μονοπατιού ακριβώς όπως έχει προαναφερθεί, μέσα από το SpineComputer component. Έχουμε προσθέσει triggers για την εμφάνιση των στρατιωτών και ελεφάντων, στα σημεία που σταμάτησε ο Αννίβας ή διεξήχθησαν μάχες. Επίσης τοποθετήθηκαν μοντέλα συγκρουόμενων σπαθιών με animation από τον φάκελο prefab, που έχουν το ηχητικό εφέ, στα σημεία μαχών. Επιλέχθηκε ένα μοντέλο ελέφαντα για να

ακολουθήσει το μονοπάτι, με το οποίο παρουσιάζουμε τον στρατό του Αννίβα. Το animation γίνεται μέσω του component SplineFollower, το οποίο και σαν default κατάσταση το απενεργοποιούμε. Έτσι με αυτή την τελευταία προσθήκη, έχουμε ολοκληρώσει το πρώτο κομμάτι αυτής της σκηνής με την ακόλουθη μορφή (Σχήμα 9.3) και ιεραρχία αντικειμένων (Σχήμα 9.4).



Σχήμα 9.3: Τελική Μορφή Πρώτου Χάρτη της Πέμπτης Σκηνής



Σχήμα 9.4 : Ιεραρχία Πρώτου Image Target

9.4 Δημιουργία Περιβάλλοντος Δευτέρου Image Target

Σε αυτό το σημείο έχει μείνει η ολοκλήρωση του τελευταίου κεφαλαίου για την εργασία, που αφορά την ενότητα για την υποταγή του Ελληνικού κόσμου από τους Ρωμαίους. Η ανάπτυξη του τελευταίου χάρτη, έχει ακριβώς το ίδιο μοτίβο και μεθοδολογία με τους προηγούμενους που έχουμε αναλύσει μέχρι τώρα. Σαν βάση προστέθηκε ένα ακόμα ImageTarget στο οποίο θα απεικονίζεται το σχήμα από την σελίδα 124 του σχολικού βιβλίου που φαίνεται παρακάτω (Σχήμα 9.5).



Σχήμα 9.5 : Εικόνα Ενάτου Image Target

Ο χάρτης που χρησιμοποιήθηκε για την αναπαράσταση της περιοχής είναι ο ίδιος με αυτόν από την πρώτη σκηνή, στο πέμπτο κεφάλαιο της Π.Ε, καθώς η γεωγραφική περιοχή ενδιαφέροντος είναι η ίδια. Και σε αυτή την περίπτωση επαναχρησιμοποίησης αντικειμένου χάρτη θα χρειαστεί η εκ νέου δημιουργία 3D TMP, για την ονομασία και προσδιορισμό των περιοχών, αφού τα σημεία ενδιαφέροντος είναι διαφορετικά. Ωστόσο κάνουμε χρήση του ίδιου material.

Ακολούθως έγινε η δημιουργία των εικονιζόμενων στρατιωτών με την μεθοδολογία που αναφέρθηκε στο 9.3. Τοποθετήθηκαν για την αναπαράσταση των σημείων που έγιναν μάχες ή και παρατάχθηκαν στρατεύματα, όπως αναφέρεται στην ύλη. Επιπρόσθετα τοποθετήθηκαν αντικείμενα με το animation σύγκρουσης των σπαθιών που δημιουργήσαμε από το 7.3.1 με δυνατότητα παραγωγής του ήχου. Τελική κίνηση είναι η προσθήκη Box Collider και στα δύο είδη αντικειμένων. Η τελική μορφή φαίνεται στο Σχήμα 9.6.

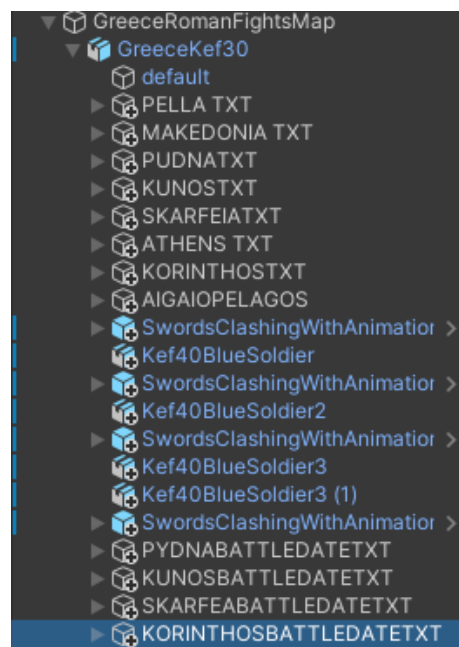
Με αυτή την κίνηση τα μετατρέπουμε σε 3D Toggle Button. Με το πάτημα πάνω τους ενεργοποιούμε κείμενο στο οποίο παρέχουμε επιπλέον πληροφορία για την μάχη την οποία αντιπροσωπεύουν. Όταν ξαναπατηθούν, το νέο 3D TMP που περιέχει το κείμενο απενεργοποιείται. Η υλοποίηση αυτής της λειτουργικότητας έγινε με αντίστοιχο τρόπο με τον οποίο έχουμε αναφέρει για την δεύτερη σκηνή της εφαρμογής στο έκτο κεφάλαιο τη Π.Ε.

Κεφάλαιο 9ο

Η κλίμακα των μοντέλων είναι διαφορετική, συμβολίζοντας την σημασία και μέγεθος της κάθε συμπλοκής ανάμεσα στους δύο στρατούς. Προτιμήθηκε να μην γίνει χρήση animation σε αυτή την σκηνή καθώς η αλληλουχία των γεγονότων πραγματοποιήθηκε σε μεγάλο χρονικό διάστημα. Τα αντικείμενα τα οποία απαρτίζουν τον χάρτη φαίνονται στο Σχήμα 9.7.



Σχήμα 9.6 : Τελική Μορφή Δεύτερου Χάρτη της Πέμπτης Σκηνής Χωρίς Κείμενο



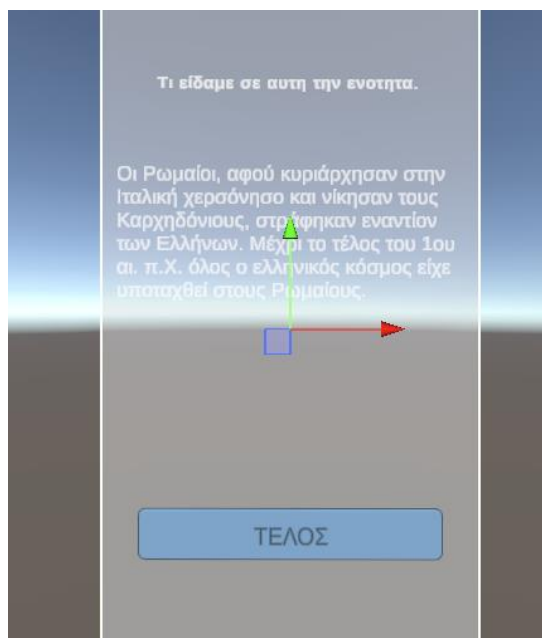
Σχήμα 9.7 : Ιεραρχία ΔεύτερουImage Target

9.5 User Interface

Με το πέρας των παραπάνω διεργασιών, έχουμε συμπληρώσει την επαύξηση της ύλης μέσω τρισδιάστατων μορφών. Μπορούμε πλέον να περάσουμε στο τελικό στάδιο του UI και την εισαγωγή των scripts σε αυτό. Αρχικά προσθέτουμε και πάλι το CanvasUI από τον φάκελο prefabs το οποίο έχει κρατήσει όλες τις επιλογές μας από όταν το αποθηκεύσαμε. Περνάμε στο TMP TEXT UI το συμπυκνωμένο κείμενο της ύλης για προβολή. Κάνουμε διαχωρισμό του κειμένου σε σελίδες, ανάλογα τα κεφάλαια που έχει η ενότητα, χωρισμένα με το rich text tag <page>. Επόμενο βήμα είναι να προσθέσουμε το AudioController με το mute script και το toggle mute button στον καμβά.

Καθώς η ροή της εφαρμογής βασίζεται στην εναλλαγή των σελίδων, βασικό ρόλο και εδώ έχει το κουμπί για την επομένη σελίδα. Σε αυτό φορτώνουμε τα τρία scripts, τα οποία έχουν δημιουργηθεί και αναλυθεί στα παραπάνω κεφάλαια. Αυτά είναι, για την αλλαγή σελίδας, αλλαγή σκηνής κατά την τελευταία σελίδα, και αλλαγή κατάστασης των αντικειμένων με βάση την τρέχουσα σελίδα. Έγινε επιλογή ώστε η τρίτη και η πέμπτη σελίδα να είναι αυτές που ενεργοποιούν, το κουμπί για την εκκίνηση του animation, και την εμφάνιση του δευτέρου Image Target με τον χάρτη, αντίστοιχα. Επίσης υπάρχει η δυνατότητα επιστροφής σε προηγούμενη σελίδα, με ανάλογη λογική αλλά αντίστροφα, με το κουμπί για την προηγούμενη σελίδα.

Τελικό βήμα είναι η δημιουργία του outro Panel(Σχήμα 9.8) . Εκεί κάνουμε την περίληψη της ύλης που παρουσιάστηκε με λίγες λέξεις και δίνουμε την δυνατότητα στον χρήστη να επιστρέψει στην αρχική σκηνή. Αυτό συμβαίνει πατώντας το κουμπί το οποίο αντί για “ΕΠΟΜΕΝΟ”, αναγράφει “ΤΕΛΟΣ”. Με αυτόν τον τρόπο τον ενημερώνουμε ότι η εφαρμογή έχει ολοκληρωθεί. Η εμφάνιση του Panel γίνεται με το πέρας της τελευταίας σελίδας.



Σχήμα 9.8 : Ανασκόπηση της Πέμπτης Σκηνής

9.6 Επίλογος

Στο ένατο κεφάλαιο καλύφθηκε η ύλη του δευτέρου μισού της τέταρτης ενότητας του σχολικού βιβλίου. Αποτελεί την τελευταία σκηνή που εμφανίζεται στην εφαρμογή της Π.Ε. Για την ολοκλήρωση της, αξιοποιήσαμε πληθώρα μεθόδων που καλύφθηκαν στα προηγούμενα κεφάλαια.

Κεφάλαιο 10ο : Συμπεράσματα και Προτάσεις Βελτίωσης

Κάνοντας μια ανασκόπηση από την αρχή, μπορούμε να συμπεράνουμε ότι οι τεχνολογίες της XR έχουν περάσει πλέον από το κομμάτι της φαντασίας, στην απτή πραγματικότητα. Συνεχίζουν να εξελίσσονται με εκθετικούς ρυθμούς ανάπτυξης και συνεχώς νέους τρόπους ενσωμάτωσης. Το κομμάτι της εκπαίδευσης στο οποίο εστίασε αυτή η εργασία είναι πολλά υποσχόμενο σε όλες τις βαθμίδες, αλλά κυρίως για τις μικρότερες ηλικίες. Μπορεί να συνδιάσει το παιχνίδι με την μάθηση, κάνοντας την όλη διαδικασία εκμάθησης ακόμα πιο ενδιαφέρουσα για τον μαθητή. Συγκεκριμένα ο τρόπος επιλογής του storytelling έχει ως σκοπό να συνδυαστεί το απλό κείμενο, που διαβάζει ο χρήστης, με τρισδιάστατα αντικείμενα και κινήσεις, ώστε ουσιαστικά να είναι, σαν να βιώνει μπροστά του την εξέλιξη των γεγονότων που διαβάζει. Με αυτόν τον τρόπο αυξάνεται η κατανόηση των γεγονότων και μειώνεται η πιθανότητα να τα ξεχάσει στο μέλλον.

Παρόλα τα θετικά που μπορεί να προσφέρει μια τέτοια μέθοδος διδασκαλίας στον μαθητή, η υλοποίηση μιας τέτοιας εφαρμογής δεν παύει να έχει και τις δυσκολίες της. Αφενός τόσο στο σχεδιαστικό τομέα πλαισίωσης και συγγραφής του κάθε σεναρίου που θα εμφανιστεί μέσα στην σκηνή, αλλά και αφετέρου, λόγω της πληθώρας των διαφορετικών μοντέλων που χρειάζεται για να καλυφθούν οι απαιτήσεις κάθε χρονικής περιόδου.

Λαμβάνοντας υπόψη τα προαναφερθέντα στοιχεία, υπάρχουν δυνατότητες για βελτίωση της εφαρμογής. Κάνοντας χρήση της νέας τεχνολογίας μετατροπής του κειμένου σε ομιλία, μπορεί να προσφερθεί μια πιο αδιάσπαστη εμπειρία storytelling, αφού δεν θα χρειάζεται η συνεχής ανάγνωση του κειμένου. Έτσι μπορεί ο μαθητής να εστιάζει στα μοντέλα και τις κινήσεις τους. Βασιζόμενοι στο ηχητικό κομμάτι, η προσθήκη μουσικής και επιπλέον ήχων στην εφαρμογή, θα εμπλούτιζαν την μορφή της. Περνώντας στο οπτικό κομμάτι, η προσθήκη περισσότερων εφέ και ακόμα περισσότερων λεπτομερών μοντέλων, θα βοηθούσαν να παρουσιαστούν τα κεφάλαια εκτενέστερα. Μέσω της αλληλεπίδρασης μαζί τους ο χρήστης θα μπορεί να λάβει ειδικές για αυτά πληροφορίες, αυξάνοντας την κατανόηση του για τα ιστορικά γεγονότα και θα έκανε την εφαρμογή ακόμα πιο ενδιαφέρουσα.

Για το προγραμματιστικό κομμάτι έγινε προσπάθεια τμηματοποίησης του κώδικα ώστε κάθε αρχείο να εκτελεί εξατομικευμένη ενέργεια για μεγαλύτερη σαφήνεια. Ωστόσο, υπάρχει δυνατότητα σύμπτυξης του, ώστε μικρότερο κομμάτι κώδικα με λιγότερες μεταβλητές να έχει την ίδια λειτουργικότητα.

Επίσης, πέραν του Android και της δωρεάν μεταφόρτωσης στο Play Store, υπάρχει η δυνατότητα εξαγωγής της εφαρμογής και σε iOS μέσω του Unity. Έτσι, είναι εφικτό να χρησιμοποιήσουν την εφαρμογή και χρήστες συσκευών της Apple.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] The history of augmented reality (AR) and its future Available: <https://vr.x.vr-expert.com/the-history-of-augmented-reality-ar-and-its-future/#b-1584-first-mention-of-ar-in-magia-naturalis-b> [Online]. Last Access July 2023
- [2] A Brief History of Augmented Reality (+ Future Trends & Impact). Available: <https://www.g2.com/articles/history-of-augmented-reality> [Online]. Last Access August 2023
- [3] AR vs. virtual reality vs. mixed reality vs. extended reality Available: <https://dynamics.microsoft.com/en-us/mixed-reality/guides/what-is-augmented-reality-ar/> [Online]. Last Access July 2023
- [4] Azuma, Ronald T. "A survey of augmented reality." Presence: teleoperators & virtual environments 6.4 (1997): 355-385. <https://www.cs.unc.edu/~azuma/ARpresence.pdf> Last Access July 2023
- [5] Pokémon GO Available: <https://www.pokemon.com/us/app/pokemon-go/> [Online]. Last Access August 2023
- [6] Introduction to Mixed Reality Available: <https://thinkpalm.com/blogs/introduction-mixed-reality/> [Online]. Last Access July 2023
- [7] Beyond AR vs. VR: What is the Difference between AR vs. MR vs. VR vs. XR? Available: <https://www.interaction-design.org/literature/article/beyond-ar-vs-vr-what-is-the-difference-between-ar-vs-mr-vs-vr-vs-xr> [Online]. Last Access 2023
- [8] What Is Extended Reality? Every Immersion Counts! Available: <https://www.g2.com/articles/extended-reality> [Online]. Last Access 2023
- [9] Degrees of Freedom (DoF): 3-DoF vs 6-DoF for VR Headset Selection Available: <https://virtualspeech.com/blog/degrees-of-freedom-vr> [Online]. Last Access 2023
- [10] Introduction to extended reality: AR, VR and MR Available: <https://courses.minnalearn.com/en/courses/emerging-technologies/extended-reality-vr-ar-mr/introduction-to-extended-reality-ar-vr-and-mr/> [Online]. Last Access 2023
- [11] How Does Augmented Reality in E-Commerce Transform the Business Sphere? Available: <https://www.appventurez.com/blog/augmented-reality-in-ecommerce/> [Online]. Last Access 2023
- [12] 20 Current Augmented Reality Trends & Predictions for 2022/2023 and Beyond Available: <https://financesonline.com/augmented-reality-trends/> [Online]. Last Access 2023

- [13] 12 Augmented Reality Trends of 2023: New Milestones in Immersive Technology Available: <https://mobidev.biz/blog/augmented-reality-trends-future-ar-technologies> [Online]. Last Access July 2023
- [14] Augmented Reality Examples: How Businesses are Utilising AR Available: <https://daglar-cizmece.com/augmented-reality-examples-how-businesses-are-utilising-ar/> [Online]. Last Access July 2023
- [15] Silva, Rodrigo, Jauvane C. Oliveira, and Gilson A. Giraldi. "Introduction to augmented reality." *National laboratory for scientific computation* 11.1 (2003): 1. Last Access July 2023
- [16] 37 Crucial Pokémon Go Statistics: 2023 Data on Downloads, Revenue & Usage Available: <https://financesonline.com/pokemon-go-statistics/> [Online]. Last Access July 2023
- [17] Thermal display glove for interacting with virtual reality Available: <https://www.nature.com/articles/s41598-020-68362-y> [Online]. Last Access July 2023
- [18] Manus Prime 3 Haptic XR Available: <https://www.manus-meta.com/products/prime-3-haptic-xr> [Online]. Last Access July 2023
- [19] Smartify Your visitor engagement platform Available: <https://partners.smartify.org/> [Online]. Last Access July 2023
- [20] Cosmote Chronos Available: <https://www.cosmote.gr/cs/cosmote/gr/CHRONOS.html> [Online]. Last Access August 2023
- [21] An Introduction to Extended Reality and Why it Matters Available: <https://geekflare.com/extended-reality/> [Online]. Last Access July 2023
- [22] Future Horizons of Extended Reality Available: <https://courses.minnalearn.com/en/courses/emerging-technologies/extended-reality-vr-ar-mr/future-horizons-of-extended-reality/> [Online]. Last Access July 2023
- [23] Leader's Augmented Reality Guide: Everything You Need To Know Available: <https://www.appventurez.com/blog/augmented-reality-guide> [Online]. Last Access July 2023
- [24] XR Jobs In Demand Available: <https://circuitstream.com/blog/xr-jobs-in-demand> [Online]. Last Access July 2023
- [25] How to Get a Job in Virtual or Augmented Reality Available: <https://www.businessnewsdaily.com/10588-get-job-in-virtual-augmented-reality.html> [Online]. Last Access July 2023

- [26] Top 6 Biggest Challenges to Implementing AR Technology Available:
<https://www.cgsinc.com/blog/top-6-biggest-challenges-implementing-ar-technology-2021> [Online]. Last Access July 2023
- [27] What is Extended Reality (XR): Complete Guide for 2023 Available:
<https://research.aimultiple.com/what-is-xr/> [Online]. Last Access July 2023
- [28] 3 Challenges Of Augmented Reality Development Available:
<https://imaginear.com/blog/ar-development-challenges> [Online]. Last Access July 2023
- [29] Blender Simulation Nodes, better UV Packing, performance improvements, and much more! Available: <https://www.blender.org/> [Online]. Last Access March 2023
- [30] Unity Available: <https://unity.com/> [Online]. Last Access March 2023
- [31] Visual Studio Code Available:
<https://code.visualstudio.com/> [Online]. Last Access March 2023
- [32] Unity Asset Store Available:
<https://assetstore.unity.com/> [Online]. Last Access May 2023
- [33] Vuforia Developer Library Available:
<https://library.vuforia.com/getting-started/> [Online]. Last Access March 2023
- [34] Vuforia Developer Library “Vuforia Engine Package for Unity” Available:
<https://library.vuforia.com/getting-started/vuforia-engine-package-unity> [Online]. Last Access April 2023
- [35] Unity Asset Store “Dreamteck Splines” Available :
<https://assetstore.unity.com/packages/tools/utilities/dreamteck-splines-61926> [Online]. Last Access April 2023
- [36] Unity Asset Store “Lean Touch” Available:
<https://assetstore.unity.com/packages/tools/input-management/lean-touch-30111> [Online]. Last Access September 2023
- [37] Blender About Info Available:
<https://www.blender.org/about/> [Online]. Last Access March 2023
- [38] Ιστορία Δ΄ Δημοτικού Στα Αρχαία Χρόνια Available:
http://ebooks.edu.gr/ebooks/v/pdf/8547/584/10-0090-02_Istoria_D-Dimotikou_Vivlio-Mathiti/ [Online]. Last Access March 2023
- [39] Sketchfab Available:
<https://sketchfab.com> [Online]. Last Access July 2023

[40] Monster Mash: A Sketch-Based Tool for Casual 3D Modeling and Animation Available:
<https://ai.googleblog.com/2021/04/monster-mash-sketch-based-tool-for.html> [Online]. Last Access June 2023

[41] Thingiverse Available:
<https://www.thingiverse.com> [Online]. Last Access July 2023

[42] CGTrader Available:
<https://www.cgtrader.com> [Online]. Last Access July 2023

[43] Turbosquid Available:
<https://www.turbosquid.com> [Online]. Last Access July 2023

[44] 3dexport Available:
<https://3dexport.com> [Online]. Last Access July 2023

[45] Blender Manual Decimate Modifier
<https://docs.blender.org/manual/en/latest/modeling/modifiers/generate/decimate.html> [Online]. Last Access April 2023

[46] Pixabay royalty-free sword sound effects Available:
<https://pixabay.com/sound-effects/search/sword/> [Online]. Last Access August 2023

[47] Audio Editor Available:
<https://editor.audio/> [Online]. Last Access August 2023

ΠΑΡΑΡΤΗΜΑ Α: script “ToggleModelsBasedOnPages”

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

public class ToggleModelsBasedOnPages : MonoBehaviour
{
    [SerializeField]
    public TextMeshProUGUI PageText;
    public GameObject menuObject;
    public GameObject menuObject2;
    public GameObject menuObject3;
    public GameObject menuObject4;

    public int TargetPage;
    public int TargetPage2;
    //public TMP_Text Text2; also work
    public TextMeshProUGUI Text2;
    int currentpage;
    int totalpages;

    // Start is called before the first frame update
    void Start()
    {

    }

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

    }
    public void ShowNextModel()
    {
        totalpages = PageText.textInfo.pageCount;
        bool currentState = menuObject.activeSelf;
        bool currentState2 = menuObject2.activeSelf;
        bool currentState3 = menuObject3.activeSelf;

        currentpage = PageText.pageToDisplay;
        // Text2 =GetComponent<TextMeshProUGUI>();
    }
}
```

```

if (currentpage == TargetPage)
{
//Debug.Log("currentpage"+ currentpage);
//Debug.Log("menuObject"+ currentState);
//Debug.Log("menuObject2"+currentState2);
menuObject.SetActive(!currentState);
menuObject2.SetActive(!currentState2);
menuObject3.SetActive(!currentState3);
Text2.enabled = true;
}
}
public void ShowPreviousModel()
{
totalpages = PageText.textInfo.pageCount;
bool currentState = menuObject.activeSelf;
bool currentState2 = menuObject2.activeSelf;
bool currentState3 = menuObject3.activeSelf;
currentpage = Text2.pageToDisplay;
// Text2 =GetComponent<TextMeshProUGUI>();
if (currentpage == TargetPage)
{
menuObject.SetActive(!currentState);
menuObject2.SetActive(!currentState2);
menuObject3.SetActive(!currentState3);
Text2.enabled = false;
}
}

```

```

public void ShowNextModelVer2()
{
totalpages = PageText.textInfo.pageCount;
bool currentState = menuObject.activeSelf;
bool currentState2 = menuObject2.activeSelf;
bool currentState3 = menuObject3.activeSelf;
bool currentState4 = menuObject4.activeSelf;
currentpage = PageText.pageToDisplay;
// Text2 =GetComponent<TextMeshProUGUI>();

```

```

if (currentpage == TargetPage)
{
menuObject.SetActive(!currentState);
menuObject2.SetActive(!currentState2);
}

```

```

if (currentpage == TargetPage2)
{
menuObject2.SetActive(!currentState2);
menuObject3.SetActive(!currentState3);
menuObject4.SetActive(!currentState4);
}
if (currentpage == TargetPage2+2){
    menuObject3.SetActive(!currentState3);
    menuObject4.SetActive(!currentState4);
}
}

public void ShowPreviousModelVer2()
{
totalpages = PageText.textInfo.pageCount;
bool currentState = menuObject.activeSelf;
bool currentState2 = menuObject2.activeSelf;
bool currentState3 = menuObject3.activeSelf;
bool currentState4 = menuObject4.activeSelf;
currentpage = PageText.pageToDisplay;
// Text2 =GetComponent<TextMeshProUGUI>();

if (currentpage == TargetPage)
{
menuObject4.SetActive(!currentState4);
menuObject3.SetActive(!currentState3);
menuObject2.SetActive(!currentState2);
}

if (currentpage == TargetPage2)
{
menuObject2.SetActive(!currentState2);
menuObject.SetActive(!currentState);
}
}

public void ShowNextModelVer3()
{
totalpages = PageText.textInfo.pageCount;
bool currentState = menuObject.activeSelf;
bool currentState2 = menuObject2.activeSelf;
bool currentState3 = menuObject3.activeSelf;

```

```

bool currentState4 = menuObject4.activeSelf;
currentpage = PageText.pageToDisplay;
// Text2 =GetComponent<TextMeshProUGUI>();

if (currentpage == TargetPage)
{
menuObject.SetActive(!currentState);
menuObject2.SetActive(!currentState2);
menuObject3.SetActive(!currentState3);
}

if (currentpage == TargetPage2)
{
menuObject4.SetActive(!currentState4);

}
}
public void ShowNextModelVer4()
{
totalpages = PageText.textInfo.pageCount;
bool currentState = menuObject.activeSelf;
bool currentState2 = menuObject2.activeSelf;
bool currentState3 = menuObject3.activeSelf;
bool currentState4 = menuObject4.activeSelf;
currentpage = PageText.pageToDisplay;
// Text2 =GetComponent<TextMeshProUGUI>();

if (currentpage == TargetPage)
{
menuObject.SetActive(!currentState);
menuObject2.SetActive(!currentState2);
menuObject4.SetActive(!currentState4);
}

if (currentpage == TargetPage2)
{
menuObject2.SetActive(!currentState2);
menuObject3.SetActive(!currentState3);

}
}
}

```

ΠΑΡΑΡΤΗΜΑ Β: script “QuizForScene3”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using UnityEngine.UI;
using System.Threading.Tasks;
using Dreamteck.Splines;

public class QuizForScene3 : MonoBehaviour
{
    int i,j=0;
    int currentpage;
    int Showpage;
    public List<Button> ImageBTN = new List<Button>();
    public List<TMP_Text> ShowAnswersTXT = new List<TMP_Text>();
    public TextMeshProUGUI
    QuestionPageTXT,CorrectNumAnswersTXT,CorrectNumAnswers2TXT;
    public SplineFollower[] thePaths;
    public SplineFollower[] thePaths1;
    public GameObject Soldier,SoldierAnimed,QuestionPanel,CanvasUI;
    int
    questiontxt,row,column,correctanwser,questionrow,row2,column2,correctanwser2,questionrow2;
    public bool IsActive = true;

    private string[] Questions = {
        "Ποιος ήταν ο δάσκαλος του Μ.Αλεξάνδρου;",
        "Ποιον ηρώα που θαύμαζε ο Μ.Αλέξανδρος τίμησε στην Τροία;",
        "Που έγινε η πρώτη μάχη της εκστρατείας κατά των Περσών.",
        "Οι κάτοικοι των πόλεων της Μ.Ασίας αντιμετώπισαν τον Αλέξανδρο ως:",
        "Η/Το ___ της Αιγύπτου που αναδείχτηκε σε σπουδαία πόλη ιδρύθηκε από τον Αλέξανδρο.."
    };

    private string[] Questions2 = {
        "Σε αυτή την μάχη οι Πέρσες κατατροπώθηκαν ολοκληρωτικά .",
        "Ποτέ έγινε η προαναφερθείσα μάχη;",
        "Όταν έφτασαν στον Ινδό ποταμό οι στρατιώτες αποφάσισαν να ___;",
        "Έπειτα από πολύ σκέψη ο Αλέξανδρος αποφάσισε να",
        "Μετά από πολλές δυσκολίες ο στρατός κατέφερε να φτάσει στην",
        "Το 323π.Χ ο Αλέξανδρος πέθανε."
    };

    private string[,] AnswersOption = {
```

```

{"Αριστοτέλης ","Φίλιππος"},
{"Αχιλλέα","Οδυσσέα"},
{"Ισσό","Γρανικό "},
{"Ελευθερωτή ","Κατακτητή "},
{"Κάιρο","Αλεξάνδρεια" }
};

private string[,] AnswersOption2 = {
{"Σούσα","Γαυγάμηλα "},
{"331 π.Χ","333 π.Χ"},
{"Συνεχίσουν","Διαμαρτυρηθούν"},
{"Επιστρέψει","Προχωρήσει"},
{"Κίνα","Βαβυλώνα"},
{"Σωστό","Λάθος"}
};
private int[] Answer = {
0,0,1,0,1
};
private int[] Answer2 = {
1,0,1,0,1,0
};

// Start is called before the first frame update
void Start()
{
ShowTexts();
thePaths = Soldier.GetComponents<SplineFollower>();
thePaths1 = SoldierAnimed.GetComponents<SplineFollower>();

}

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

}

public void CalledOnClick(int clickedChoice)
{
if(IsActive)
{
PressedChoisePath1(clickedChoice);
Debug.Log("PressedChoisePath1");
Debug.Log("IsActive"+IsActive);
}
}

```

```

    }
    else
    {
        PressedChoisePath2(clickedChoice);
        Debug.Log("PressedChoisePath2");
    }
}

public void PausePath1(){

    thePaths[0].enabled = false;
    thePaths1[0].enabled = false;
}

public void PausePath2(){

    thePaths[1].enabled = false;
    thePaths1[1].enabled = false;
}

public async void PressedChoisePath1(int clickedChoice){

    if(clickedChoice==Answer[i]){
        correctanwser++;
        i++;
        ImageBTN[clickedChoice].GetComponent<Image>().color = Color.green;
        await Task.Delay(1000);
    }
    else{
        ImageBTN[clickedChoice].GetComponent<Image>().color = Color.red;
        i++;
        await Task.Delay(1000);
    }

    j++;
    ImageBTN[0].GetComponent<Image>().color = Color.white;
    ImageBTN[1].GetComponent<Image>().color = Color.white;

    Debug.Log("Pressed1I="+i);
    Debug.Log("Pressed1Jis="+j);
    Debug.Log("correctAnwserfor1Pressed"+correctanwser);
}

```

```

        if(j == 5){
            CorrectNumAnswersTXT.text = "Οι σωστές απαντήσεις απο το Πρώτο Quiz του κεφαλαίου ήταν:
"+ correctanwser +
                "\nΚαι οι λανθασμένες : " + (Answer.Length-correctanwser);
            SoldierAnimed.GetComponent<Animator>().enabled = true;
            thePaths[0].enabled = true;
            thePaths1[0].enabled = true;
            QuestionPanel.SetActive(false);
            CanvasUI.SetActive(true);
            i=0;
            j=0;
            ShowTexts2();
            IsActive = false;
            return;
        }
        ShowTexts();
    }

    public async void PressedChoisePath2(int clickedChoice){

        if(clickedChoice==Answer2[i]){
            correctanwser2++;
            i++;
            ImageBTN[clickedChoice].GetComponent<Image>().color = Color.green;
            await Task.Delay(1000);
        }
        else{
            ImageBTN[clickedChoice].GetComponent<Image>().color = Color.red;
            i++;
            await Task.Delay(1000);
        }

        j++;
        ImageBTN[0].GetComponent<Image>().color = Color.white;
        ImageBTN[1].GetComponent<Image>().color = Color.white;

        Debug.Log("Pressed2Jis==" +j);
        //Debug.Log("correctanwser"+correctanwser);

        if(j == 6){
            CorrectNumAnswers2TXT.text = "Οι σωστές απαντήσεις απο το Δεύτερο Quiz του κεφαλαίου
ήταν: "+ correctanwser2 +
                "\nΚαι οι λανθασμένες : " + (Answer2.Length-correctanwser2);
            SoldierAnimed.GetComponent<Animator>().enabled = true;

```

```

thePaths[1].enabled = true;
thePaths1[1].enabled = true;
QuestionPanel.SetActive(false);
CanvasUI.SetActive(true);
return;
}
ShowTexts2();
}

private void ShowTexts(){

if (row< AnswersOption.GetLength(0)) {
ShowAnswersTXT.ForEach(TMPIInButton => TMPInButton.text = AnswersOption[row,
column++]);
column = 0; row++;

}
QuestionPageTXT.text=Questions[questionrow++];
}

private void ShowTexts2(){

if (row2< AnswersOption2.GetLength(0)) {
ShowAnswersTXT.ForEach(TMPIInButton => TMPInButton.text = AnswersOption2[row2,
column2++]);
column2 = 0; row2++;

}
QuestionPageTXT.text=Questions2[questionrow2++];
}
}

```

ΠΑΡΑΡΤΗΜΑ C: script “QuizForScene4”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using UnityEngine.UI;
using System.Threading.Tasks;
using Dreamteck.Splines;

public class QuizForScene4 : MonoBehaviour
{
    int i,j=0;
    int currentpage;
    int Showpage;
    public List<Button> ImageBTN = new List<Button>();
    public List<TMP_Text> ShowAnswersTXT = new List<TMP_Text>();
    public TextMeshProUGUI QuestionPageTXT,CorrectNumAnswersTXT;
    public SplineFollower SoldierPath;
    public GameObject Soldier,QuestionPanel,CanvasUI;
    int questiontxt,row,column,correctanwser,questionrow;

    private string[] Questions = {
        "Με τον όρο «πύρρεια νίκη» αναφερόμαστε σε νίκη με",
        "Γιατί ο Πύρρος δεν πολιορκήσε την Ρωμη",
        "Ο Πύρρος κατάφερε να νικήσει σε όλες τις μάχες στην Ιταλία.",
        "Η εκστρατεία του Πυρρού προς την Σπαρτή στέφτηκε με επιτυχία.",
        "Η τελευταία μάχη του Πυρρού ήταν στο Άργος όπου και σκοτώθηκε."
    };

    private string[,] AnswersOption = {
        {"Πολλές απώλειες","Μεγάλη ευκολία"},
        {"Μεγάλη απόσταση","Ελλιπείς δυνάμεις"},
        {"Σωστό","Λάθος"},
        {"Σωστό","Λάθος"},
        {"Σωστό","Λάθος"}
    };

};

private int[] Answer = {
    0,1,1,1,0
};

// Start is called before the first frame update
```

```

void Start()
{
ShowTexts();
}

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

}
public void PausePath(){

SoldierPath.enabled = false;

}

public async void PressedChoisePurros(int clickedChoice){

if(clickedChoice==Answer[i]){
correctanwser++;
i++;
ImageBTN[clickedChoice].GetComponent<Image>().color = Color.green;
await Task.Delay(1000);
}
else{
ImageBTN[clickedChoice].GetComponent<Image>().color = Color.red;
i++;
await Task.Delay(1000);
}
j++;
ImageBTN[0].GetComponent<Image>().color = Color.white;
ImageBTN[1].GetComponent<Image>().color = Color.white;

Debug.Log("j"+j);
Debug.Log("correctanwser"+correctanwser);
if(j == 3 ){
Soldier.GetComponent<Animator>().enabled = true;
SoldierPath.enabled = true;
QuestionPanel.SetActive(false);
ShowTexts();
return;
}

if(j == 5 ){

```

```

CorrectNumAnswersTXT.text = "Οι σωστές απαντήσεις απο το Quiz του κεφαλαίου ήταν "+
correctanwser +
        "\nΚαι οι λανθασμένες : " + (Answer.Length-correctanwser);
QuestionPanel.SetActive(false);
CanvasUI.SetActive(true);
return;}

ShowTexts();

}

private void ShowTexts(){

if (row< AnswersOption.GetLength(0)) {
ShowAnswersTXT.ForEach(TMPIInButton => TMPInButton.text = AnswersOption[row,
column++]);
column = 0; row++;

}
QuestionPageTXT.text=Questions[questionrow++];
}
}

```

ΠΑΡΑΡΤΗΜΑ D: script “Scene2CheckForBoats”

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

public class Scene2CheckForBoats : MonoBehaviour
{
    public TextMeshProUGUI PageText;
    public GameObject PromptMessage;
    public GameObject BoatFrance;
    public GameObject BoatNorth;
    public GameObject BoatItaly;
    public GameObject NextPageBTN;
    public int TargetPage;
    public int TargetPage2;
    int currentpage;
    int totalpages;
    int a,b,c;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (BoatFrance.activeSelf == true){
            a=1;
        }

        if (BoatNorth.activeSelf == true){
            b=1;
        }
        if (BoatItaly.activeSelf == true){
            c=1;
        }

        if(PromptMessage.activeSelf == true){
            PromptMessage.GetComponent<TMP_Text>().text = "Πατήστε πάνω στις τοποθεσίες των χωρών(έντονα γράμματα) για να δείτε την μετακίνηση."+ "\n" +(a+b+c)+"/3";
        }
    }
}
```

```

    }

    if (BoatFrance.activeSelf == true && BoatNorth.activeSelf == true && BoatItaly.activeSelf ==
true && PageText.pageToDisplay==3 )
    {
        NextPageBTN.SetActive(true);

    }
    if(PageText.pageToDisplay == 4) {
        PromptMessage.SetActive(false);
    }

}

public void ShowNextPageOnShipsDone(){
    totalpages = PageText.textInfo.pageCount;
    currentpage = PageText.pageToDisplay;
    if (currentpage == TargetPage & BoatFrance.activeSelf == false & BoatNorth.activeSelf == false
& BoatItaly.activeSelf == true ){
        NextPageBTN.SetActive(false);
        PromptMessage.SetActive(true);
    }

}

public void ShowNextPageBTNNonPreviousClick(){
    currentpage = PageText.pageToDisplay;
    if (currentpage == TargetPage2 && NextPageBTN.activeSelf==false){
        NextPageBTN.SetActive(true);
    }
}
}

```