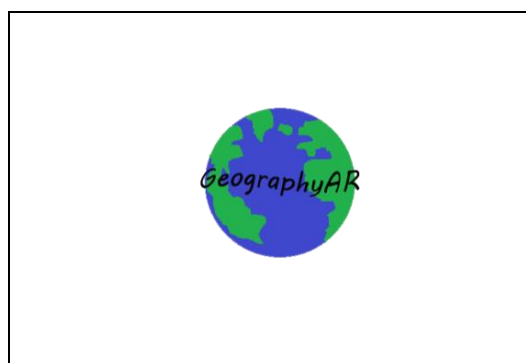




ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ
ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΜΕ
ΣΤΟΧΟ ΤΗΝ ΒΕΛΤΙΩΣΗ ΤΩΝ ΜΑΘΗΤΩΝ
Ε' ΔΗΜΟΤΙΚΟΥ ΣΤΟ ΜΑΘΗΜΑ ΤΗΣ
ΓΕΩΓΡΑΦΙΑΣ



Του φοιτητή
Γεώργιου Χριστόφορου Καζάρης
Αρ. Μητρώου: 164667

Επιβλέπων
Ευκλείδης Κεραμόπουλος
Αναπληρωτής Καθηγητής

Θεσσαλονίκη 2022

Τίτλος Δ.Ε. Ανάπτυξη εκπαιδευτικού παιχνιδιού επαυξημένης πραγματικότητας με στόχο την βελτίωση των μαθητών Ε΄ Δημοτικού στο μάθημα της Γεωγραφίας

Κωδικός Δ.Ε. 21352

Ονοματεπώνυμο φοιτητή/των Γιώργος Χριστόφορος Καζλάρης

Ονοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος

Ημερομηνία ανάληψης Δ.Ε. 14-10-2021

Ημερομηνία περάτωσης Δ.Ε. ...

Βεβαιώνω ότι είμαι ο 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 διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Γεώργιου Χριστόφορου Καζλάρη που την εκπόνησε/αν. 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 απαραίτητα και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

1. Πρόλογος

Ο τομέας της πληροφορικής παίζει καθοριστικό ρόλο στην εξέλιξη της τεχνολογίας σε όλους τους επαγγελματικούς τομείς. Βασικός λόγος της εξέλιξης είναι να κάνει την ζωή του ανθρώπου πολύ πιο εύκολη. Πλέον η τεχνολογία έχει εισχωρήσει στην καθημερινότητα και έχει γίνει τρόπος ζωής. Η εκπαίδευση χρησιμοποιεί όλο και περισσότερο την τεχνολογία σαν μέσο βοήθειας για καλύτερη και πιο αποτελεσματικότερη εκμάθηση. Τα τελευταία χρόνια η επαυξημένη πραγματικότητα έχει ξεκινήσει να αναπτύσσεται σημαντικά τόσο σε επαγγελματικούς τομείς όσο και στην εκπαίδευση. Πρόκειται για μια τεχνολογία η οποία βρίσκεται σε αρχικά στάδια αλλά δείχνει να προσεγγίζει το ενδιαφέρον μεγάλων εταιριών. Μέσω της επαυξημένης πραγματικότητας, οι μαθητές είναι σε θέση να μπορούν να περιηγηθούν μέσα σε ένα περιβάλλον στο οποίο εξερευνούν, μαθαίνουν, και προβληματίζονται. Με την βοήθεια της συγκεκριμένης τεχνολογίας τα μαθήματα θα μπορούν πλέον να είναι πιο διαδραστικά και πρακτικά, κεντρίζοντας το ενδιαφέρον των μαθητών και των μαθητριών, προσφέροντας τους την δυνατότητα να εξερευνήσουν ένα καινούργιο κόσμο, με αποτέλεσμα να μπορέσουν να κατευθυνθούν πιο εύκολα στον τομέα που τους ενδιαφέρει για το μέλλον τους.

2. Περίληψη

Η παρούσα διπλωματική εργασία άπτεται με την δημιουργία ενός εκπαιδευτικού παιχνιδιού για το μάθημα της Γεωγραφίας της Ε' Δημοτικού με την βοήθεια της επαυξημένης πραγματικότητας. Στόχος αυτής της εφαρμογής είναι να βοηθήσει τους μαθητές να κατανοήσουν με έναν πιο διασκεδαστικό και εύκολο τρόπο, το μάθημα της Γεωγραφίας. Η επαυξημένη πραγματικότητα θα δώσει στους μαθητές, την δυνατότητα να περιηγηθούν μέσα σε ένα περιβάλλον το οποίο είναι σχεδιασμένο κατάλληλα για κάθε ένα από τα κεφάλαια του βιβλίου, έτσι ώστε να κατανοούν καλύτερα την κάθε ενότητα του βιβλίου. Τα παιχνίδια έχουν δημιουργηθεί με βάση την ύλη του μαθήματος του σχολικού έτους 2021-2022, με την χρήση της πλατφόρμας Unity και Vuforia. Όλα τα αντικείμενα έχουν δημιουργηθεί με το Blender, ενώ ο κώδικας της εφαρμογής είναι γραμμένος σε κώδικα C#. Στην συγκεκριμένη διπλωματική εργασία γίνεται ανάλυση των παραπάνω, και με την βοήθεια εικόνων, ενώ στο τέλος παρουσιάζονται οι βελτιώσεις της εφαρμογής.

Λέξεις κλειδιά: Επαυξημένη πραγματικότητα, Γεωγραφία, Unity, Vuforia Engine, QR codes

Development of an augmented reality educational game aiming to improve 5th grade pupils in Geography course.

Georgios Christoforos Kazlaris

3. Abstract

The present thesis deals with the creation of an educational game for the Geography course of the 5th grade with the help of augmented reality. The purpose of this app, is to help pupils to understand in more fun and easy way, the Geography course. Augmented Reality will give the opportunity to students, to navigate through an environment that is appropriately designed for each of book's chapters, so that they can understand each section of the book, easier. The games have been created based on the course material of the school year 2021-2022, using the Unity and Vuforia platform. All objects created with Blender, while the app's code is C#. All the above are analyzed in this thesis with the help of images, and finally the improvements of the app are presented.

Keywords: Augmented Reality, Geography, Unity, Vuforia Engine, QR codes

4. Ευχαριστίες

Πρώτα από όλα θα ήθελα να ευχαριστήσω θερμά τον επιβλέπων καθηγητή Κ. Ευκλείδη Κεραμόπουλο, που μου έδωσε την ευκαιρία να ασχοληθώ με ένα τόσο σημαντικό και ενδιαφέρον θέμα, εισπράττοντας πολλές γνώσεις και εμπειρίες.

Επιπλέον θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου, που με στηρίζουν σε κάθε μου προσπάθεια και είναι δίπλα σε κάθε μου βήμα.

Τέλος, θα ήθελα να δώσω ένα μεγάλο ευχαριστώ σε όλα τα άτομα που με βοήθησαν σε αυτή την προσπάθεια να ολοκληρώσω την διπλωματική.

5. Περιεχόμενα

1. Πρόλογος.....	3
2. Περίληψη.....	4
3. Abstract	5
4. Ευχαριστίες.....	6
5. Περιεχόμενα	7
6. Κατάλογος Σχημάτων	10
7. Συντομογραφίες	13
Κεφάλαιο 1ο: Εισαγωγή.....	1
Κεφάλαιο 2ο: Τεχνολογίες στον Εικονικό Κόσμο.....	5
2.1 Εισαγωγή	5
2.2 Διαφορές Επαυξημένης και Εικονικής Πραγματικότητας.....	5
2.3 Εκτεταμένη και Μικτή Πραγματικότητα	6
2.4 Κινούμενη εικόνα στον επαυξημένο χώρο	6
2.5 Βελτίωση της εκπαίδευσης με την βοήθεια της Επαυξημένης Πραγματικότητας	7
2.6 Επίλογος.....	8
Κεφάλαιο 3ο: Κύριες εφαρμογές που χρησιμοποιήθηκαν	9
3.1 Εισαγωγή	9
3.2 Πλατφόρμα Unity	9
3.3 Βασικές ρυθμίσεις Unity.....	9
3.4 Πλατφόρμα Blender	10
3.4.1 Βασικό βοήθημα για την υλοποίηση της εφαρμογής.....	11
3.5 Vuforia και βασικές ρυθμίσεις	13
3.5.1 Εγκατάσταση Vuforia και σύνδεση με Unity.....	13
3.6 Ρυθμίσεις Unity για συσκευές με Android λογισμικό	16
3.7 Επίλογος.....	17
Κεφάλαιο 4ο: Βασικός κορμός παιχνιδιού	19
4.1 Εισαγωγή	19
4.2 Έναρξη εφαρμογής.....	19
4.2.1 Επιλογή κουμπιών	21
4.3 Μουσική υπόκρουση	25
4.4 Βασικά στοιχεία του Project	26
4.5 Επίλογος.....	27

Κεφάλαιο 5ο: Τα είδη των χαρτών	29
5.1 Εισαγωγή	29
5.2 Δομή Παιχνιδιού	29
5.3 Επίλογος.....	38
Κεφάλαιο 6ο: Προσανατολισμός.....	39
6.1 Εισαγωγή	39
6.2 Δομή Παιχνιδιού	39
6.2.1 Υλοποίηση σκηνής Game2	43
6.3 Επίλογος.....	47
Κεφάλαιο 7ο: Οι ακτές της Ελλάδας.....	49
7.1 Εισαγωγή	49
7.2 Η δομή του παιχνιδιού	49
7.3 Επίλογος.....	57
Κεφάλαιο 8ο: Νησιά της Ελλάδας.....	59
8.1 Εισαγωγή	59
8.2 Υλοποίηση παιχνιδιού	59
8.2.1 Υλοποίηση πρώτης ενότητας παιχνιδιού	62
8.2.2 Υλοποίηση ολοκλήρωσης του παιχνιδιού.....	67
8.3 Επίλογος.....	71
Κεφάλαιο 9ο: Τα βουνά της Ελλάδας	73
9.1 Εισαγωγή	73
9.2 Η δομή του παιχνιδιού	73
9.2.1 Υλοποίηση της σκηνής του παιχνιδιού	77
9.2.2 Animations	82
9.3 Επίλογος.....	83
Κεφάλαιο 10ο: Το κλίμα της Ελλάδας.....	85
10.1 Εισαγωγή	85
10.2 Η δομή του παιχνιδιού	85
10.2.1 Χρήση API calls στο παιχνίδι.....	91
10.3 Επίλογος.....	94
Κεφάλαιο 11ο: Γεωγραφική κατανομή του πληθυσμού.....	95
11.1 Εισαγωγή	95
11.2 Υλοποίηση Παιχνιδιού	95
11.2.1 Υλοποίηση κώδικα σκηνής	98
11.3 Επίλογος.....	104

Κεφάλαιο 12ο: Συμπεράσματα και προτάσεις βελτίωσης.....	105
8. ΒΙΒΛΙΟΓΡΑΦΙΑ	106

6. Κατάλογος Σχημάτων

Εικόνα 1.1 Βιβλίο Γεωγραφίας της 5 ^{ης} δημοτικού	3
Εικόνα 2.1 Παιχνίδι Pokemon Go	6
Εικόνα 2.2 Κινούμενες εικόνες στον επαυξημένο κόσμο	7
Εικόνα 2.3 Επαυξημένη πραγματικότητα στην εκπαίδευση	8
Εικόνα 3.1 Λογότυπο της Unity	9
Εικόνα 3.2 Ρυθμίσεις JDK, Gradle, JDK	10
Εικόνα 3.3 Λογότυπο Blender	11
Εικόνα 3.4 Εξαγωγή αρχείου σε FBX	11
Εικόνα 3.5 Μετατροπή χάρτη σε αντικείμενο	12
Εικόνα 3.6 Εισαγωγή ύψους	12
Εικόνα 3.7 QR code	13
Εικόνα 3.8 Αρχική σελίδα Vuforia	14
Εικόνα 3.9 QR codes στην σελίδα της Vuforia	14
Εικόνα 3.10 Κλειδί από την ιστοσελίδα της Vuforia	15
Εικόνα 3.11 Επικόλληση κλειδιού στο Unity	16
Εικόνα 3.12 Σήμα Vuforia στην συσκευή	16
Εικόνα 3.13 Παράθυρο Build Settings	17
Εικόνα 4.1 Εικόνα Εφαρμογής	19
Εικόνα 4.2 Πρώτο Splash Image	19
Εικόνα 4.3 Δεύτερο Splash Image	20
Εικόνα 4.4 Αρχικό Menu	21
Εικόνα 4.5 Κώδικας για κλείσιμο εφαρμογής	21
Εικόνα 4.6 About Scene	22
Εικόνα 4.7 SelectLevel Scene	23
Εικόνα 4.8 Script LevelSelector	24
Εικόνα 4.9 Τοποθέτηση script σε Empty Object	25
Εικόνα 4.10 Παράθυρο Audio Source στο Unity	25
Εικόνα 4.11 Όλα τα assets του Project	26
Εικόνα 4.12 Φάκελος με τα Scenes	26
Εικόνα 4.13 Φάκελος με τα Scripts	27
Εικόνα 4.14 Φάκελος με τα Materials	27
Εικόνα 5.1 Δισδιάστατη σκηνή φάσης πρώτης	29
Εικόνα 5.2 QR code 1 ^{ου} παιχνιδιού στο σχολικό βιβλίο	30
Εικόνα 5.3 Η δεύτερη φάση του παιχνιδιού	31
Εικόνα 5.4 Εισαγωγή βιβλιοθηκών	32
Εικόνα 5.5 Αρχικοποίηση μεταβλητών	32
Εικόνα 5.6 QnA.cs script	33
Εικόνα 5.7 Μέθοδος Start()	33
Εικόνα 5.8 Μέθοδος SetRandomQuestion()	33
Εικόνα 5.9 Εντολή GetMouseButtonDown	34
Εικόνα 5.10 Box Collider	34
Εικόνα 5.11 Έλεγχος επιλογής αντικειμένου και ερώτησης	35
Εικόνα 5.12 Μέθοδος TransitionToNextQuestion()	36
Εικόνα 5.13 Μέθοδος EndGame()	36

Εικόνα 5.14 Σκηνή αποτελεσμάτων	37
Εικόνα 5.15 Results.cs script.....	37
Εικόνα 6.1 Σκηνή IntroGame2	40
Εικόνα 6.2 Κώδικας κουμπιών μαζί με τον ήχο	41
Εικόνα 6.3 Βιβλιοθήκες που χρησιμοποιήθηκαν στο IntroGame2Script.cs	41
Εικόνα 6.4 QR code του παιχνιδιού πάνω στο βιβλίο	42
Εικόνα 6.5 Εμφάνιση παιχνιδιού «Προσανατολισμός»	42
Εικόνα 6.6 Κώδικας script Qlist.cs.....	43
Εικόνα 6.7 Αρχικοποίηση μεταβλητών στο Questions.cs.....	43
Εικόνα 6.8 Μέθοδος Start().....	44
Εικόνα 6.9 Μέθοδος SetRandomQuestion()	44
Εικόνα 6.10 Μέθοδος Default().....	44
Εικόνα 6.11 Μέθοδος Update() στο script Questions.cs.....	45
Εικόνα 6.12 Μέθοδος OnClickAnatolika()	46
Εικόνα 6.13 Σκηνή αποτελεσμάτων	46
Εικόνα 6.14 Μέθοδος Start() στο script Results.cs.....	47
Εικόνα 7.1 Εισαγωγική σκηνή παιχνιδιού	49
Εικόνα 7.2 Κώδικας script IntroButton.cs	50
Εικόνα 7.3 QR code από το παιχνίδι «Οι ακτές της Ελλάδας»	51
Εικόνα 7.4 Κουμπί περιστροφής αντικειμένων	52
Εικόνα 7.5 Αρχικοποίηση μεταβλητών του script ObjectRotation.cs	52
Εικόνα 7.6 Μέθοδος Update του script ObjectRotation.cs	53
Εικόνα 7.7 Αρχικοποίηση μεταβλητών στο script SelectObject.cs	54
Εικόνα 7.8 Μέθοδος Start() του script SelectObject.cs	54
Εικόνα 7.9 Μέθοδος ClickIsthmos().....	55
Εικόνα 7.10 Μέθοδος OnClickPanelButton()	56
Εικόνα 7.11 Μέθοδος NextObject()	56
Εικόνα 7.12 Τελικό Panel παιχνιδιού	57
Εικόνα 8.1 Αρχικό Panel παιχνιδιού.....	59
Εικόνα 8.2 Audio Source μέσα στο SoundObject αντικείμενο	60
Εικόνα 8.3 Τοποθέτηση QR code πάνω στο σχολικό βιβλίο	61
Εικόνα 8.4 Πινακίδα με το όνομα νησιού προς συμπλήρωση.....	62
Εικόνα 8.5 Δισδιάστατα κουμπιά για την συμπλήρωση των λέξεων	62
Εικόνα 8.6 Αρχικοποίηση μεταβλητών στο script KikladesManager.cs	63
Εικόνα 8.7 Μέθοδος Start() από το script KikladesManager.cs.....	63
Εικόνα 8.8 Μέθοδος ClickButton2() ελέγχει το κουμπί που επιλέχθηκε.....	64
Εικόνα 8.9 Μέθοδος GoToNextIslandNaxos()	65
Εικόνα 8.10 Μέθοδος EndKikladesScene()	66
Εικόνα 8.11 Μέθοδος EndKikladesNextButton() στο script ButtonManager.cs.....	66
Εικόνα 8.12 Canvas επόμενης ενότητας	67
Εικόνα 8.13 Αρχικοποίηση μεταβλητών στο script DodekanisaScript.cs	68
Εικόνα 8.14 Μέθοδος SetLetters().....	68
Εικόνα 8.15 Μέθοδος EndDodekanisaScene() από το script DodekanisaScript.cs.....	69
Εικόνα 8.16 Μέθοδος StartSporades().....	69
Εικόνα 8.17 Μέθοδος SetLettersForSkiathos()	70
Εικόνα 8.18 Τελικό Panel του παιχνιδιού	70

Εικόνα 9.1	Πρώτο Panel με τις οδηγίες	73
Εικόνα 9.2	Αρχικοποίηση μεταβλητών στο script CanvasManager.cs.....	74
Εικόνα 9.3	Μέθοδος StartGuideFromCanvas().....	75
Εικόνα 9.4	Δεύτερο εισαγωγικό Panel.....	75
Εικόνα 9.5	QR code τοποθετημένο πάνω στο βιβλίο	76
Εικόνα 9.6	Το επανζημένο παιχνίδι.....	77
Εικόνα 9.7	Αρχικοποίηση μεταβλητών στο script Buttons.cs.....	78
Εικόνα 9.8	Μέθοδος OnMouseDown() στο script MaterialSaver.cs.....	78
Εικόνα 9.9	Μέθοδος OnClickLocationbt1().....	79
Εικόνα 9.10	Αλλαγή χρώματος κουμπιών	80
Εικόνα 9.11	Μέθοδος CheckIfCorrect()	81
Εικόνα 9.12	Εμφάνιση κουμπιού «Πίσω στην αρχική».....	82
Εικόνα 9.13	Δημιουργία Animation	83
Εικόνα 9.14	Τοποθέτηση Animator σε αντικείμενο	83
Εικόνα 10.1	Αρχικό Panel παιχνιδιού.....	85
Εικόνα 10.2	QR code παιχνιδιού πάνω στο σχολικό βιβλίο	86
Εικόνα 10.3	Πινακίδες καιρικών φαινομένων.....	87
Εικόνα 10.4	Αρχικοποίηση μεταβλητών στο script QuestionButton.cs	88
Εικόνα 10.5	Μέθοδος OnClickRainWeather() στο script WeatherManager.cs	89
Εικόνα 10.6	Capsule Collider	90
Εικόνα 10.7	Μέθοδος OnClickButton3() στο script QuestionButton.cs.....	91
Εικόνα 10.8	Μέθοδος Update()	91
Εικόνα 10.9	Ιστοσελίδα Openweathermap.org	92
Εικόνα 10.10	Ανάκτηση κλειδιού από την ιστοσελίδα	92
Εικόνα 10.11	Μέθοδος UpdateWeatherData().....	93
Εικόνα 10.12	Αποτελέσματα πίνακα με API calls	94
Εικόνα 11.1	Πρώτο Panel παιχνιδιού	95
Εικόνα 11.2	QR Code τοποθετημένο πάνω στο σχολικό βιβλίο	96
Εικόνα 11.3	Ελληνική Στατιστική Αρχή (ELSTAT).....	97
Εικόνα 11.4	Αποτέλεσμα σωστής απάντησης.....	97
Εικόνα 11.5	Κουμπί εξόδου παιχνιδιού	98
Εικόνα 11.6	Αρχικοποίηση μεταβλητών στο script CanvasButtonManager.cs	99
Εικόνα 11.7	Μέθοδος OnClickAboutInfoBt().....	100
Εικόνα 11.8	Απόσπασμα της μεθόδου OnClickNextBt()	101
Εικόνα 11.9	Έλεγχος απενεργοποίησης όλων των εικόνων.....	101
Εικόνα 11.10	Αρχικοποίηση μεταβλητών στο script DragDropManager.cs.....	102
Εικόνα 11.11	Μέθοδος OnDrag().....	102
Εικόνα 11.12	Απόσπασμα της μεθόδου OnEndDrag()	103
Εικόνα 11.13	Μέθοδοι για το πάτημα της κάθε πινακίδας	104

7. Σύντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality
XR	Extended Reality
JDK	Java Development Kit

Κεφάλαιο 1ο: Εισαγωγή

Τα τελευταία χρόνια η επαυξημένη πραγματικότητα (Augmented Reality) τραβάει τα βλέμματα πολλών μεγάλων εταιρειών. Όλο και περισσότερες εφαρμογές δημιουργούνται με χρήση της επαυξημένης πραγματικότητας, βάζοντας την με αυτό τον τρόπο στην καθημερινότητα πολλών ανθρώπων. Δεν είναι όμως μόνο αυτή η τεχνολογία που έχει αρχίσει να εξελίσσεται. Υπάρχουν και άλλες τεχνολογίες με διαφορετικά χαρακτηριστικά, όπως για παράδειγμα η Virtual Reality (VR), Extended Reality (XR) και Mixed Reality (MR). Η Virtual Reality τεχνολογία, βάζει τον χρήστη σε έναν ψηφιακό κόσμο χωρίς να έχει οποιαδήποτε επαφή με τον πραγματικό κόσμο. Σε αντίθεση με το VR, η Augmented Reality τεχνολογία συνδέει τον πραγματικό κόσμο με αντικείμενα του ψηφιακού κόσμου.

Οι τεχνολογίες που αναφέρθηκαν παραπάνω έχουν αρχίσει να εφαρμόζονται σε πολλούς επαγγελματικούς τομείς, όπως επίσης και στον τομέα της εκπαίδευσης. Οι νέες αυτές τεχνολογίες μπαίνουν σταδιακά στην εκπαίδευση με στόχο την καλύτερη διεξαγωγή των μαθημάτων και την καλύτερη κατανόηση της ύλης, με μια πιο απλή και διασκεδαστική διαδικασία.

Η παρούσα διπλωματική εργασία έχει να κάνει με ένα εκπαιδευτικό παιχνίδι για το μάθημα της Γεωγραφίας της Ε' δημοτικού με τίτλο «Ανάπτυξη εκπαιδευτικού παιχνιδιού επαυξημένης πραγματικότητας με στόχο την βελτίωση των μαθητών Ε' δημοτικού στο μάθημα της Γεωγραφίας». Αυτή στοχεύει στο να κάνει το μάθημα πιο ευχάριστο και κατανοητό στους μαθητές βελτιώνοντας τις συνθήκες της εκπαίδευσης και ενισχύοντας στην εκμάθηση της Γεωγραφίας. Βασική ιδέα αυτής της διπλωματικής είναι να συνδυάσει την επαυξημένη πραγματικότητα με μαθήματα διαφόρων εκπαιδευτικών βαθμίδων σε πραγματικές συνθήκες διεξαγωγής των μαθημάτων. Αυτό θα έχει ως αποτέλεσμα την καλύτερη κατανόηση της ύλης, το ερέθισμα για εξερεύνηση περεταίρω πληροφοριών, είτε πάνω στο μάθημα, είτε πάνω στην ίδια την τεχνολογία, και τέλος θα αποτελεί μια ιδιαίτερη διαδικασία, ευχάριστη για τον κάθε χρήστη.

Η εφαρμογή η οποία παρουσιάζεται στην παρούσα διπλωματική ονομάζεται «GeographyAR» και χρησιμοποιεί τεχνολογία επαυξημένης πραγματικότητας με την βοήθεια του Vuforia, το οποίο ανακτά όλα τα QR codes που χρησιμοποιούνται μέσα στο βιβλίο για την προβολή του εικονικού περιβάλλοντος σε επαυξημένη πραγματικότητα. Η εφαρμογή καλύπτει την ύλη της Γεωγραφίας της Ε' δημοτικού και έχει σχεδιαστεί κατάλληλα για να βοηθήσει τους χρήστες να καταλαβαίνουν διάφορες έννοιες, τοποθεσίες, φαινόμενα, ονόματα και πολλά ακόμα που μπορούν να ανακαλύψουν μέσω της εφαρμογής [1].

Σε μια σύντομη περιγραφή της δομής αυτής της Διπλωματικής Εργασίας τα κεφάλαια χωρίζονται στις επιμέρους ενότητες.

Το πρώτο κεφάλαιο είναι η Εισαγωγή, όπου γίνεται μια σύντομη παρουσίαση της εφαρμογής που αναπτύχθηκε για το μάθημα της Γεωγραφίας της Ε' δημοτικού, των διάφορων σύγχρονων τεχνολογιών, και τον στόχο που έχει η συγκεκριμένη Διπλωματική Εργασία.

Το δεύτερο κεφάλαιο έχει τον τίτλο «Τεχνολογίες στον Εικονικό Κόσμο» και γίνεται αναφορά στις τεχνολογίες που υπάρχουν για την προσομοίωση διάφορων εικονικών περιβαλλόντων, όσο επίσης και για τις διάφορες που έχουν μεταξύ τους. Επιπλέον αναλύεται ο ρόλος που μπορεί να έχει η Επαυξημένη Πραγματικότητα στην εκπαίδευση και πως μπορεί αυτή να αναβαθμιστεί μέσω της εξέλιξης.

Στο τρίτο κεφάλαιο «Κύριες εφαρμογές που χρησιμοποιήθηκαν» αναλύεται η πλατφόρμα της Unity μαζί με τις βασικές ρυθμίσεις και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του εκπαιδευτικού παιχνιδιού. Επιπλέον περιγράφονται οι λειτουργίες της Vuforia καθώς επίσης και βασικές λειτουργίες που χρησιμοποιήθηκαν για την δημιουργία αντικειμένων από το περιβάλλον του Blender.

Στο τέταρτο κεφάλαιο «Βασικός κορμός παιχνιδιού» περιγράφονται τα περιεχόμενα με τους φακέλους και τις ρυθμίσεις που χρησιμοποιήθηκαν για την εφαρμογή από το περιβάλλον της Unity. Επιπλέον γίνεται εισαγωγή στις πρώτες σκηνές της εφαρμογής και αναλύονται τα βήματα που ακολουθήθηκαν για την δημιουργία τους.

Στο πέμπτο κεφάλαιο «Τα είδη των χαρτών» πραγματοποιείται η ανάλυση του πρώτου παιχνιδιού της εφαρμογής GeographyAR. Αρχικά γίνεται η παρουσίαση του παιχνιδιού και η αναφορά των στόχων του, και στην συνέχεια γίνεται η ανάλυση των λειτουργιών του παιχνιδιού καθώς και του κώδικα που χρησιμοποιήθηκε για την δημιουργία του.

Στο έκτο κεφάλαιο «Προσανατολισμός» παρουσιάζεται το δεύτερο παιχνίδι της εφαρμογής GeographyAR. Γίνεται αναφορά των στόχων και των λειτουργιών του συγκεκριμένου παιχνιδιού και στην συνέχεια γίνεται η ανάλυση του κώδικα που χρησιμοποιήθηκε για την υλοποίηση του.

Στο έβδομο κεφάλαιο «Οι ακτές της Ελλάδας» γίνεται ανάλυση του τρίτου παιχνιδιού της εφαρμογής. Περιγράφονται οι στόχοι του και οι λειτουργίες που παρέχει, και στην συνέχεια αναλύονται όλες οι μέθοδοι και οι μεταβλητές από τα scripts που χρησιμοποιήθηκαν για την υλοποίηση του παιχνιδιού.

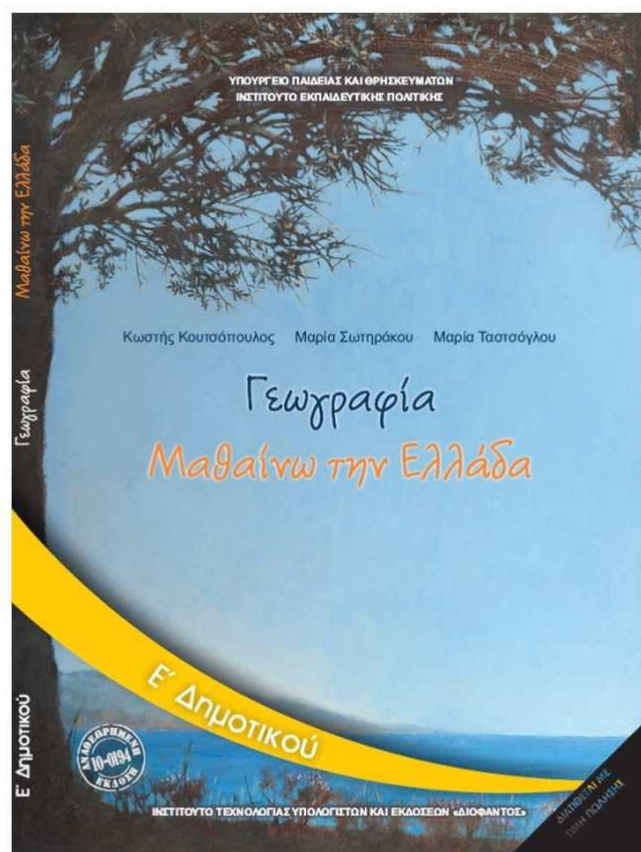
Στο όγδοο κεφάλαιο «Νησιά της Ελλάδας» πραγματοποιείται ανάλυση του τέταρτου παιχνιδιού. Περιγράφονται οι λειτουργίες του καθώς και ο κώδικας που χρησιμοποιήθηκε για να υλοποιηθεί το συγκεκριμένο παιχνίδι.

Στο ένατο κεφάλαιο «Τα βουνά της Ελλάδας» περιγράφεται το πέμπτο παιχνίδι της εφαρμογής. Γίνεται ανάλυση στην δομή του παιχνιδιού καθώς και σε όλα τα κομμάτια του κώδικα που υλοποιήθηκαν για την σωστή λειτουργία του παιχνιδιού.

Στο δέκατο κεφάλαιο «Το κλίμα της Ελλάδας» γίνεται περιγραφή του έκτου παιχνιδιού και γίνεται ανάλυση του κώδικα που χρησιμοποιήθηκε για την υλοποίηση του καθώς και η περιγραφή της δομής του παιχνιδιού.

Το ενδέκατο κεφάλαιο «Γεωγραφική κατανομή του πληθυσμού» αποτελεί το τελικό παιχνίδι της εφαρμογής και γίνεται περιγραφή της δομής του καθώς και του κώδικα που χρησιμοποιήθηκε για να υλοποιηθεί το συγκεκριμένο παιχνίδι.

Το δωδέκατο κεφάλαιο περιέχει το συμπέρασμα και τους τρόπους βελτίωσης της συγκεκριμένης διπλωματικής εργασίας, και στην συνέχεια ακολουθεί η βιβλιογραφία μαζί με παραρτήματα από κομμάτια κώδικα της εφαρμογής.



Εικόνα 1.1 Βιβλίο Γεωγραφίας της 5^{ης} δημοτικού

Κεφάλαιο 2ο: Τεχνολογίες στον Εικονικό Κόσμο

2.1 Εισαγωγή

Η τεχνολογία στην εποχή μας βρίσκεται σε ένα σημείο όπου οι εξελίξεις είναι ραγδαίες. Ο άνθρωπος προσπαθεί καθημερινά να βελτιώσει με την βοήθεια της τεχνολογίας την καθημερινότητά του, αναβαθμίζοντας τα πληροφοριακά συστήματα. Με γοργά βήματα μπαίνουμε στην εποχή όπου τα τρισδιάστατα ψηφιακά αντικείμενα θα συνυπάρχουν με τον πραγματικό κόσμο, ενώνοντας έτσι τον ψηφιακό κόσμο με τον πραγματικό. Ήδη πολλές μεγάλες εταιρίες έχουν στρέψει το βλέμμα τους στις ανερχόμενες τεχνολογίες που αφορούν τον ψηφιακό κόσμο, και οι πλατφόρμες που θα μπορούν να τον υποστηρίξουν έχουν αρχίσει να πληθαίνουν. Στο κεφάλαιο αυτό γίνεται αναφορά ιστορικών σημείων για την εικονική πραγματικότητα, και περιγράφονται οι διάφορες τεχνολογίες μαζί με τις διαφορές που έχουν μεταξύ τους. Τέλος, ολοκληρώνεται το κεφάλαιο με την συνεισφορά της Επαυξημένης Πραγματικότητας (AR) στην βελτίωση της εκπαίδευσης.

2.2 Διαφορές Επαυξημένης και Εικονικής Πραγματικότητας

Η Επαυξημένη Πραγματικότητα (AR) ή αλλιώς ενισχυμένη πραγματικότητα είναι μια τεχνολογία η οποία συνδέει τον ψηφιακό κόσμο με τον πραγματικό. Εισάγει στην ουσία ψηφιακά αντικείμενα στον πραγματικό κόσμο και σε πραγματικό χρόνο. Τα δεδομένα τα οποία επαυξάνονται σε ένα πραγματικό περιβάλλον έχουν τρισδιάστατη μορφή και συνδυάζονται με αντικείμενα που υπάρχουν στον φυσικό χώρο [2]. Οι συσκευές που χρησιμοποιούνται κυρίως για να συνδεθεί ο χρήστης σε ένα εικονικό περιβάλλον με την τεχνολογία της επαυξημένης πραγματικότητας είναι τα κινητά, τα γυαλιά και τα tablets. Πολύ γνωστό παράδειγμα εφαρμογής με χρήση επαυξημένης πραγματικότητας είναι το γνωστό παιχνίδι για smartphones, Pokemon Go από την εταιρία Niantic. Το παιχνίδι αυτό δίνει την δυνατότητα στους χρήστες να δουν μέσα από την κάμερα του κινητού ή του Tablet, ψηφιακά αντικείμενα που είναι τοποθετημένα σε πραγματικό χώρο (Εικόνα 2.1) [3].

Η εικονική πραγματικότητα (VR) είναι μια από τις τεχνολογίες που χρησιμοποιούνται για την δημιουργία ενός ψηφιακού κόσμου. Στην ουσία είναι η προσομοίωση ενός περιβάλλοντος, είτε φανταστικό, είτε πραγματικό, από έναν υπολογιστή δίνοντας έτσι την εντύπωση στον χρήστη ότι περιβάλλεται από αυτό. Ο χρήστης μπορεί να κινηθεί ελεύθερα μέσα στον χώρο και να αλληλεπιδράσει με τρισδιάστατα αντικείμενα. Για αυτό τον λόγο, πρέπει ο χρήστης να έχει μια πλήρως απομονωμένη ορατότητα για να μην αποσπάται από το πραγματικό περιβάλλον. Παρόλα αυτά ο άνθρωπος δεν χρειάζεται να χρησιμοποιεί όλες τις αισθήσεις του για να περιηγηθεί με επιτυχία σε ένα εικονικό περιβάλλον [4]. Οι αισθήσεις που χρειάζονται στην εικονική πραγματικότητα είναι η όραση, η ακοή και η αφή. Για την όραση χρειάζεται να υπάρχει στερεοσκοπική εικόνα για να υπάρχει η δυνατότητα δημιουργίας της αίσθησης του βάθους. Η στερεοσκοπική εικόνα είναι δύο εικόνες, οι οποίες τοποθετούνται σε διαφορετική οπτική γωνία, μια για κάθε μάτι του ανθρώπου. Πολλές φορές, η αφή δεν υποστηρίζεται από το εικονικό περιβάλλον στο οποίο ο χρήστης περιηγείται. Ωστόσο η αφή δεν παίζει τόσο σημαντικό ρόλο όσο οι άλλες δυο αισθήσεις, και αυτό γιατί η εικονική πραγματικότητα θέλει να δώσει την εντύπωση στον χρήστη ότι βρίσκεται σε έναν διαφορετικό κόσμο, και αυτό έχει να κάνει με την αντίληψη [4][5].



Εικόνα 2.1 Παιχνίδι Pokemon Go

2.3 Εκτεταμένη και Μικτή Πραγματικότητα

Η εκτεταμένη πραγματικότητα (XR-extended reality) είναι ένας πιο γενικός όρος, ο οποίος περιλαμβάνει όλες τις νέες τεχνολογίες όπως για παράδειγμα την επαυξημένη (Augmented Reality), την εικονική πραγματικότητα (Virtual Reality) και την Μικτή Πραγματικότητα (Mixed Reality). Πολύ σημαντική έννοια της εκτεταμένης πραγματικότητας είναι το οπτικό πεδίο. Αυτό καθορίζει το πώς βιώνει ένας χρήστης την εκτεταμένη πραγματικότητα. Γενικά το οπτικό πεδίο είναι το μέγεθος του παρατηρήσιμου χώρου σε κάθε δεδομένη στιγμή. Επομένως η γνώση της εμβέλειας του οπτικού πεδίου στον εικονικό κόσμο παίζει μεγάλο ρόλο στην συγκεκριμένη τεχνολογία [6].

Από την άλλη μεριά η μικτή πραγματικότητα (Mixed-Reality) αποτελεί έναν συνδυασμό εικονικών με πραγματικών αντικειμένων για την δημιουργία νέων απεικονίσεων. Πιο συγκεκριμένα, η συγκεκριμένη τεχνολογία ενώνει τις τεχνολογίες της Εικονικής και της Επαυξημένης Πραγματικότητας, συνδυάζοντας έτσι τον εικονικό κόσμο με τον πραγματικό [6][7].

2.4 Κινούμενη εικόνα στον επαυξημένο χώρο

Η δημιουργία μιας κινούμενης εικόνας μέσα στον επαυξημένο κόσμο που αλληλεπιδρά στενά με πραγματικά περιβάλλοντα, είναι μια πολύ ενδιαφέρων λειτουργία που μπορεί να προσφέρει αυτή η τεχνολογία, αλλά αποτελεί και μια δύσκολη διαδικασία για την υλοποίηση. Η χρήση μιας κινούμενης εικόνας στον επαυξημένο κόσμο δίνει περισσότερη ζωή στην εφαρμογή, προσδίδοντας καλύτερη διαδραστικότητα του χρήστη μεταξύ του εικονικού και του φυσικού περιβάλλοντος. Δεδομένου ότι πρέπει να ευθυγραμμιστούν τα εικονικά αντικείμενα με τα πραγματικά, σε διαφορετικές χρονικές στιγμές, καθιστά την δημιουργία κινούμενων σχεδίων στον επαυξημένο χώρο μια αρκετά δύσκολη πρόκληση (Εικόνα 2.2) [8].

Για να αλληλεπιδράσουν σωστά οι κινούμενες εικόνες στον πραγματικό κόσμο και με επιτυχία μέσω της επαυξημένης πραγματικότητας, πρέπει να υπάρχει σωστός συγχρονισμός και ακριβής τοποθέτηση του μοντέλου πάνω στο φυσικό αντικείμενο. Ο σκελετός (Skeletal animation) σε ένα εικονικό αντικείμενο παίζει μεγάλο ρόλο, καθώς μέσω αυτού πραγματοποιείται η ομαλή κίνηση του

αντικειμένου. Στην ουσία είναι μια τεχνική στα κινούμενα σχέδια στην οποία ένας χαρακτήρας αναπαριστάται σε δύο κομμάτια. Το πρώτο είναι η επιφανειακή αναπαράσταση που χρησιμοποιείται για τον σχεδιασμό του χαρακτήρα και ονομάζεται πλέγμα ή δέρμα. Το δεύτερο αποτελείται από ένα ιεραρχικό σύνολο διασυνδεδεμένων μερών όπου ονομάζονται οστά. Τα οστά χρησιμοποιούνται για να ζωντανεύουν το πλέγμα, προσφέροντας στο εικονικό αντικείμενο ευέλικτες κινήσεις και τον έλεγχο της παραμόρφωσης του [9][10]. Γενικά, η μοντελοποίηση τρισδιάστατων αντικειμένων είναι αρκετά πιο εύκολη στις μέρες μας, από ότι ήταν πιο παλιά. Παρόλα αυτά, παραμένει δύσκολη η διαδικασία να «ζωντανέψει» ένα στατικό αντικείμενο, καθώς ο χρήστης θα πρέπει να προσαρμόσει τον χαρακτήρα χειροκίνητα. Αυτό σημαίνει πως θα πρέπει να αναλάβει την χειροκίνητη τοποθέτηση των αρθρώσεων του σκελετού μέσα στο πλέγμα του χαρακτήρα, κάνοντας την διαδικασία της κινούμενης εικόνας πιο απαιτητική και δύσκολη [10].



Εικόνα 2.2 Κινούμενες εικόνες στον επανυζημένο κόσμο

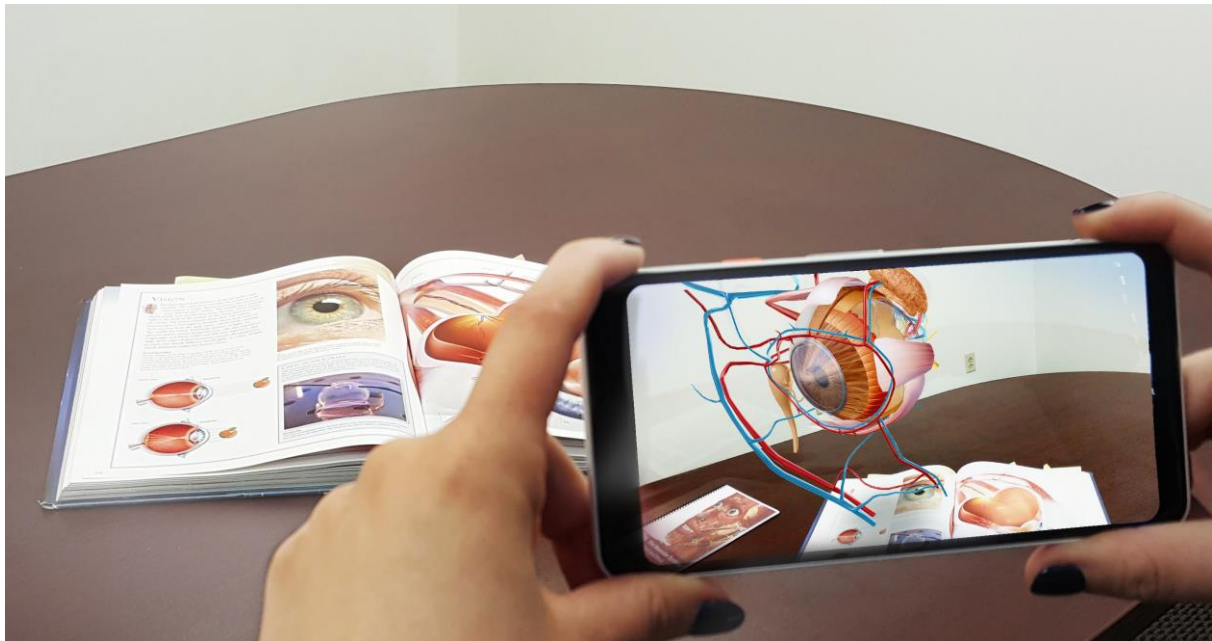
2.5 Βελτίωση της εκπαίδευσης με την βοήθεια της Επαυξημένης Πραγματικότητας

Η εκπαίδευση εκσυγχρονίζεται με αργούς αλλά σταθερού ρυθμούς εντάσσοντας την τεχνολογία στον τομέα της. Με το πέρασμα των χρόνων η τηλεόραση έγινε ένα μέσω προβολής εκπαιδευτικών ταινιών στα σχολεία. Στην συνέχεια ο βιντεοπροβολέας (Projector) προστέθηκε στα εργαλεία της εκπαίδευσης και μέχρι σήμερα οι φορητοί υπολογιστές (laptops) και τα tablets έχουν αντικαταστήσει σε ένα ποσοστό τα βιβλία. Αυτή η εξέλιξη της εκπαίδευσης έγινε σε βάθος χρόνου, παρόλο που η τεχνολογία αναπτύσσεται και εξελίσσεται με γοργούς ρυθμούς.

Καθώς η τεχνολογία της επαυξημένης πραγματικότητας φτάνει σε επίπεδα που μπορεί να ικανοποιήσει σε μεγάλο βαθμό τις ανθρώπινες απαιτήσεις, όλο και περισσότεροι επαγγελματικοί κλάδοι επιλέγουν την τεχνολογία αυτή ως μέσω βελτίωσης του εκπαιδευτικού τομέα. Πολλές είναι οι εφαρμογές μέχρι σήμερα που έχουν αναπτυχθεί με σκοπό να βελτιώσουν την εκπαίδευση σε κάποιο επαγγελματικό τομέα. Για παράδειγμα, έχουν δημιουργηθεί εφαρμογές με την τεχνολογία της επαυξημένης πραγματικότητας, στο τομέα της ιατρικής, για την εκπαίδευση των ειδικευόμενων, πάνω στην γενική αναισθησία. Άλλες εφαρμογές έχουν βοηθήσει εκπαιδευόμενους μηχανικούς να μελετήσουν διάφορες έννοιες και τεχνολογίες στην μηχανολογία. Γενικά, η συγκεκριμένη τεχνολογία

έχει να προσφέρει πολλά στην εκπαίδευση καθώς μπορεί να συσχετίσει τον εικονικό κόσμο με τον πραγματικό, προσφέροντας έτσι μια μοναδική εμπειρία στον χρήστη, και την δυνατότητα να εξερευνεί πράγματα στον επαγγελματικό τομέα που τον ενδιαφέρουν (Εικόνα 2.3). Μια καινοτόμα ιδέα στην εκπαίδευση είναι η χρήση βιβλίων επαυξημένης πραγματικότητας (AR books ή Augmented Books). Τα συγκεκριμένα βιβλία εμφανίζουν ένα τρισδιάστατο κινούμενο εκπαιδευτικό περιβάλλον, όπου επιτρέπουν στους μαθητές να αλληλεπιδρούν με το περιεχόμενο, προσφέροντας μια ξεχωριστή εμπειρία. Τα βιβλία επαυξημένης πραγματικότητας αποτελούν μια σημαντική εξέλιξη στον τομέα της εκπαίδευσης, καθώς οι μαθητές θα έχουν την δυνατότητα να μελετούν με μια ιδιαίτερη διαδικασία [11].

Η επαυξημένη πραγματικότητα έχει προοπτικές να δώσει κίνητρα σε μαθητές και εργαζόμενους να μελετούν με έναν πρωτότυπο τρόπο εκμάθησης. Είναι μια τεχνολογία που μπορεί να προσφέρει στην εκπαίδευση, καινοτομία και εξέλιξη κάνοντας την μάθηση και την εκπαίδευση πιο ευχάριστη και εύκολη. Ο τομέας της εκπαίδευσης μπορεί να εξελιχθεί σε μεγάλο βαθμό και η τεχνολογία της επαυξημένης πραγματικότητας δίνει την βάση για να επιτευχθεί [11][12].



Εικόνα 2.3 Επαυξημένη πραγματικότητα στην εκπαίδευση

2.6 Επίλογος

Ζούμε σε μια εποχή που η τεχνολογία του εικονικού κόσμου αρχίζει να συνυπάρχει με αυτή του πραγματικού προσφέροντας δυνατότητες που είναι πολύτιμες για την εξέλιξη του ανθρώπου. Η τεχνολογία της επαυξημένης πραγματικότητας εξελίσσεται με γρήγορους ρυθμούς προσπαθώντας να εγκατασταθεί σε όλους τους επαγγελματικούς αλλά και τους εκπαιδευτικούς τομείς. Είναι φανερό πως οι νέες τεχνολογίες έχουν πολλά να προσφέρουν, και οι άνθρωποι πολλά να κερδίσουν από αυτές.

Κεφάλαιο 3ο: Κύριες εφαρμογές που χρησιμοποιήθηκαν

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφεται μια εξαιρετικά εύχρηστη και δωρεάν πλατφόρμα η Unity, η οποία είναι γνωστή για την ανάπτυξη διάφορων ηλεκτρονικών παιχνιδιών. Επιπλέον, γίνεται περιγραφή των ρυθμίσεων και των επεκτάσεων (Imports) που χρησιμοποιήθηκαν για την δημιουργία αντικειμένων στο Blender. Τέλος θα γίνει ανάλυση των ρυθμίσεων που χρησιμοποιήθηκαν για την υλοποίηση της AR εφαρμογής.

3.2 Πλατφόρμα Unity

Μια εύχρηστη εφαρμογή, με πολλές δυνατότητες και επιλογές, είναι η πλατφόρμα Unity. Αποτελεί μια από τις καλύτερες πλατφόρμες για την ανάπτυξη ηλεκτρονικών παιχνιδιών και κυρίως για την δυνατότητα ανάπτυξης παιχνιδιών επαυξημένης πραγματικότητας. Μέσω της Unity έχουν δημιουργηθεί παιχνίδια από μεγάλες εταιρίες παγκοσμίως, με εκατομμύρια λήψεις το καθένα. Ένας ακόμα λόγος που κάνει αυτή την πλατφόρμα ιδιαίτερη, είναι η δυνατότητα υλοποίησης παιχνιδιού επαυξημένης πραγματικότητας με εργαλεία που βοηθούν στην καλύτερη χρήση της πλατφόρμας.



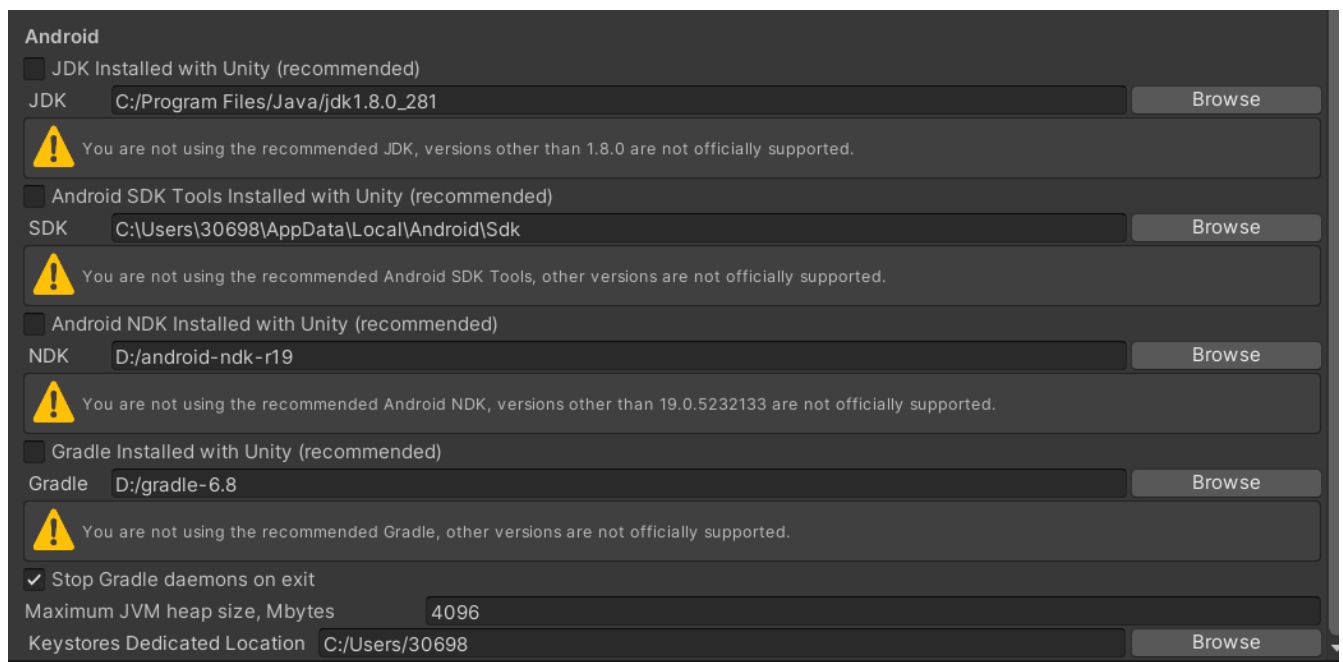
Εικόνα 3.1 Λογότυπο της Unity

3.3 Βασικές ρυθμίσεις Unity

Αρχικά η γλώσσα προγραμματισμού που χρησιμοποιήθηκε και υποστηρίζεται από την Unity είναι η C#. Η C# είναι μια αρκετά απλή και εύκολη στην χρήση, γλώσσα προγραμματισμού και η σύνταξη του κώδικα έγινε με την βοήθεια της πλατφόρμας Visual Studio Code της Microsoft. Από τις πρώτες ρυθμίσεις που θα πρέπει να γίνουν για την ολοκλήρωση της εγκατάστασης Unity, είναι η εγκατάσταση του Java Development Kit (JDK) στον υπολογιστή. Το JDK μαζί με το Software Development Kit (SDK) είναι τα βασικά εργαλεία που χρειάζονται για να μπορέσει να τρέξει Android παιχνίδι στη Unity. Οι ρυθμίσεις που χρησιμοποιήθηκαν για τα παραπάνω βρίσκονται στο

μενού Edit → Preferences → External Tools και φαίνονται παρακάτω στην εικόνα 3.2. Να σημειωθεί πως η επιλογή των εκδόσεων JDK-SDK, είναι αρκετά περίπλοκη καθώς παίζει και μεγάλο ρόλο η έκδοση της Unity που χρησιμοποιείται.

Επιπλέον ρυθμίσεις που χρειάστηκαν να γίνουν έχουν να κάνουν με τα packages της Unity, και πιο συγκεκριμένα με το Unity Package Manager. Αυτό είναι μια δυνατότητα της Unity που επιτρέπει στον χρήστη να αναζητήσει και να εγκαταστήσει διάφορα πακέτα που προσφέρει η πλατφόρμα, ανάλογα με τις δυνατότητες που θέλει να έχει ο χρήστης. Το Unity Package Manager βρίσκεται στο σημείο Window → Package Manager, και χρησιμοποιήθηκαν τα πακέτα Vuforia Engine AR, AR Foundation, AR Subsystems, TextMeshPro, Unity UI, και Visual Studio Code Editor.



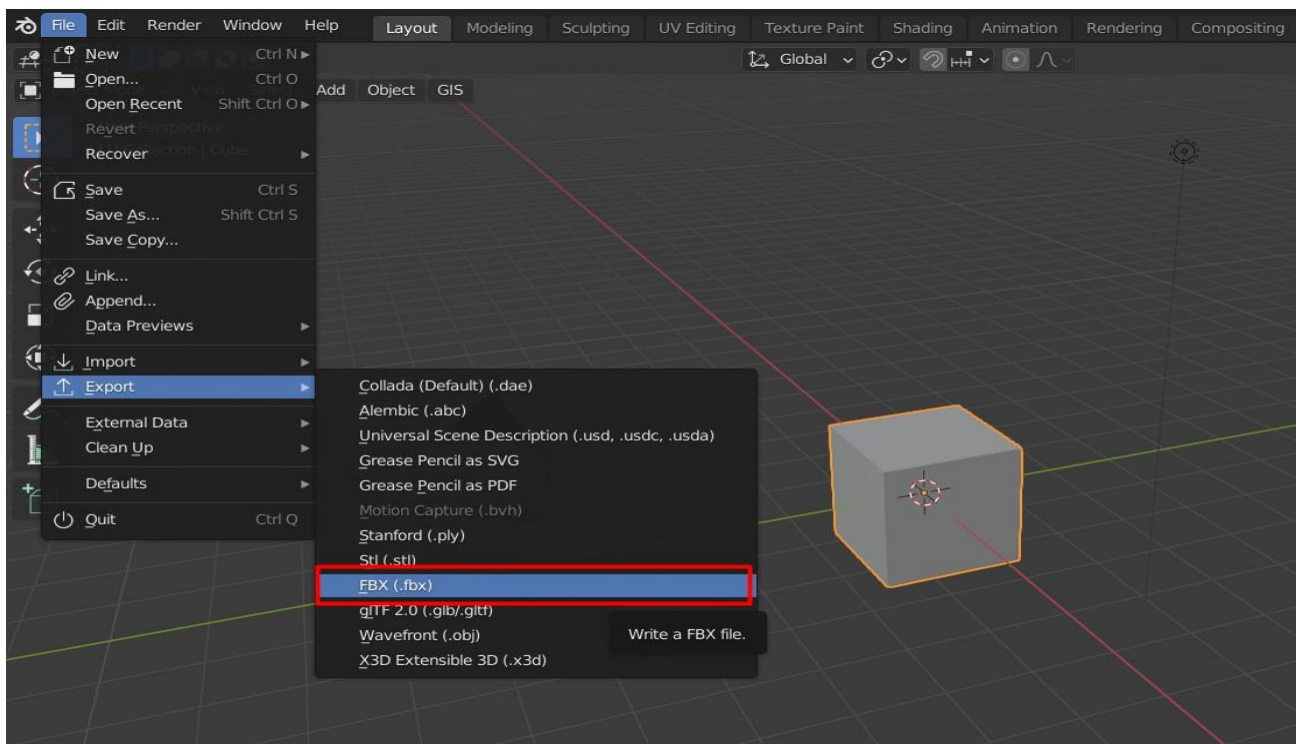
Εικόνα 3.2 Ρυθμίσεις JDK, Gradle, JDK

3.4 Πλατφόρμα Blender

Το Blender αποτελεί ένα δωρεάν σετ εργαλείων γραφικών, ανοιχτού κώδικα και χρησιμοποιείται κυρίως για τη δημιουργία 3D αντικειμένων και ταινιών κινουμένων σχεδίων. Το λογότυπο της εφαρμογής φαίνεται στην εικόνα 3.3. Τα αντικείμενα που δημιουργούνται μέσω του Blender μπορούν με εύκολο τρόπο να εξαχθούν σε άλλο project διαφορετικής εφαρμογής. Τα αντικείμενα που χρησιμοποιούνται στην εφαρμογή έχουν δημιουργηθεί με την βοήθεια του Blender και έχουν εξαχθεί στη Unity για την υλοποίηση των παιχνιδιών. Αυτό γίνεται με μια λειτουργία που προσφέρεται από το Blender η οποία ονομάζεται Export και βρίσκεται στο μενού File → Export → FBX όπως εύκολα παρατηρείται και στο σχήμα 3.4. Το FBX αρχείο (.fbx) είναι μια μορφή που χρησιμοποιείται για την ανταλλαγή τρισδιάστατων αντικειμένων και κινουμένων σχεδίων. Συνήθως αυτά τα αρχεία χρησιμοποιούνται στις ταινίες, τα ηλεκτρονικά παιχνίδια και στην Επαυξημένη Πραγματικότητα.



Εικόνα 3.3 Λογότυπο Blender



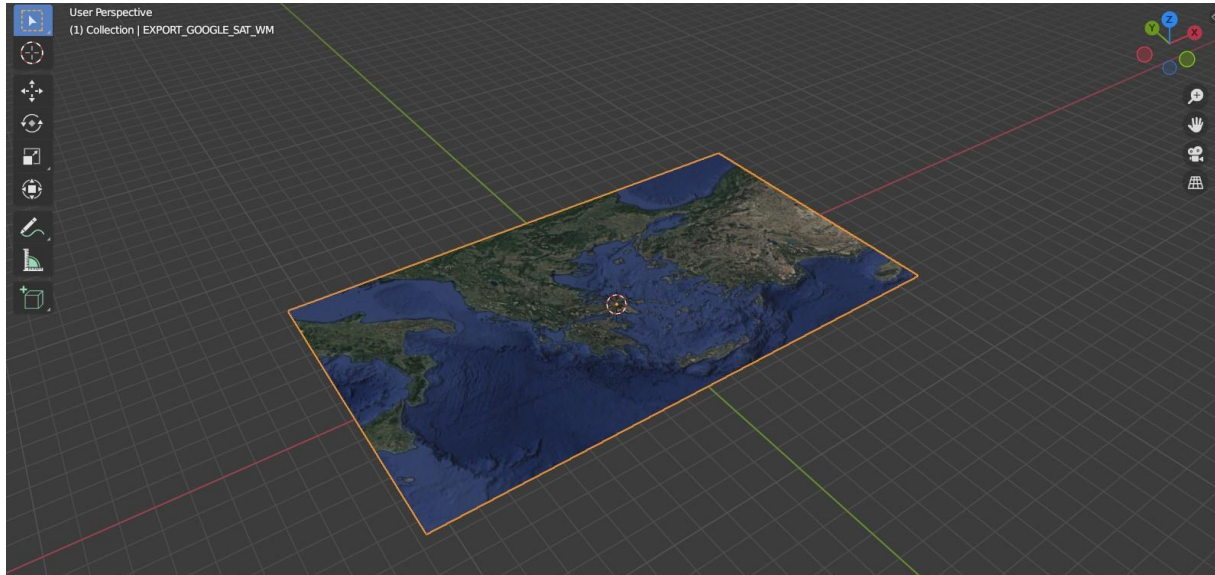
Εικόνα 3.4 Εξαγωγή αρχείου σε FBX

3.4.1 Βασικό βοήθημα για την υλοποίηση της εφαρμογής

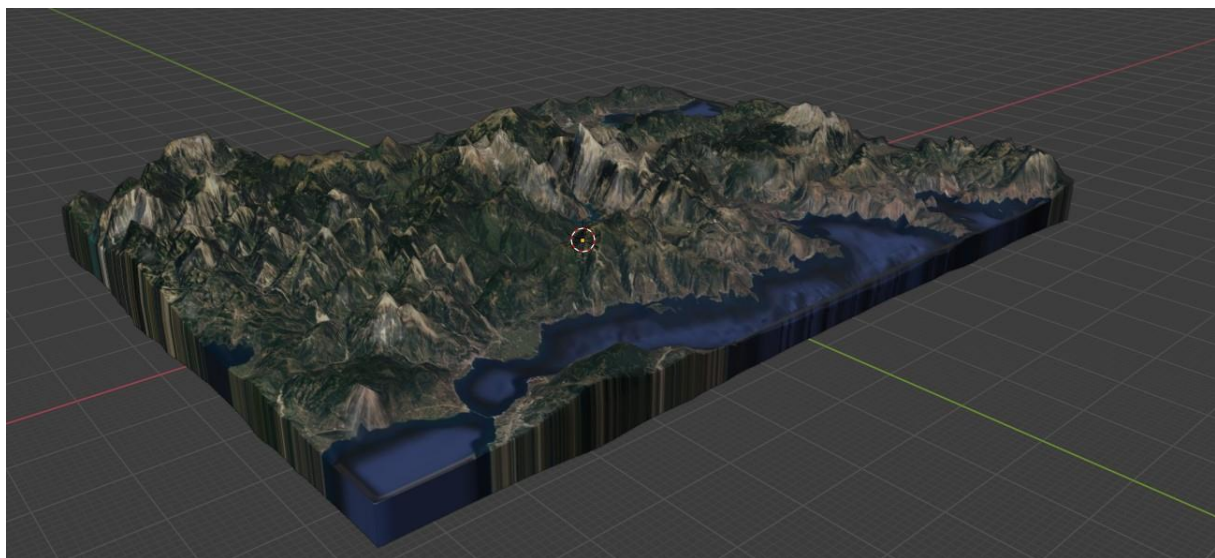
Το Blender προσφέρει μια τεραστία ποικιλία δυνατοτήτων με διάφορα εργαλεία που μπορεί ο κάθε χρήστης να χρησιμοποιήσει δωρεάν. Ένα από αυτά τα εργαλεία είναι και το Blender GIS, το οποίο χρησιμοποιήθηκε για την υλοποίηση πολλών αντικειμένων στην εφαρμογή. Το εργαλείο αυτό δίνει την δυνατότητα στο χρήστη να χρησιμοποιεί τους χάρτες της Google ή της Bing, και με βάση την

Κεφάλαιο 3

τοποθεσία που επιλέγει ο χρήστης, το εργαλείο συλλέγει τα δεδομένα από την τοποθεσία και το μετατρέπει σε αντικείμενο (Εικόνα 3.5). Στην συνέχεια μια ακόμα δυνατότητα που παρέχεται από το εργαλείο είναι να συλλέγει δεδομένα, από την τοποθεσία που επιλέχθηκε, και να δίνει στο αντικείμενο το ύψος στην αντίστοιχη κλίμακα, όπως αυτό υπάρχει στην πραγματικότητα. Παρατηρώντας την Εικόνα 3.6 βλέπουμε το αντικείμενο να έχει πάρει ύψος με τα δεδομένα που συλλέχτηκαν από τους χάρτες.



Εικόνα 3.5 Μετατροπή χάρτη σε αντικείμενο



Εικόνα 3.6 Εισαγωγή ύψους

3.5 Vuforia και βασικές ρυθμίσεις

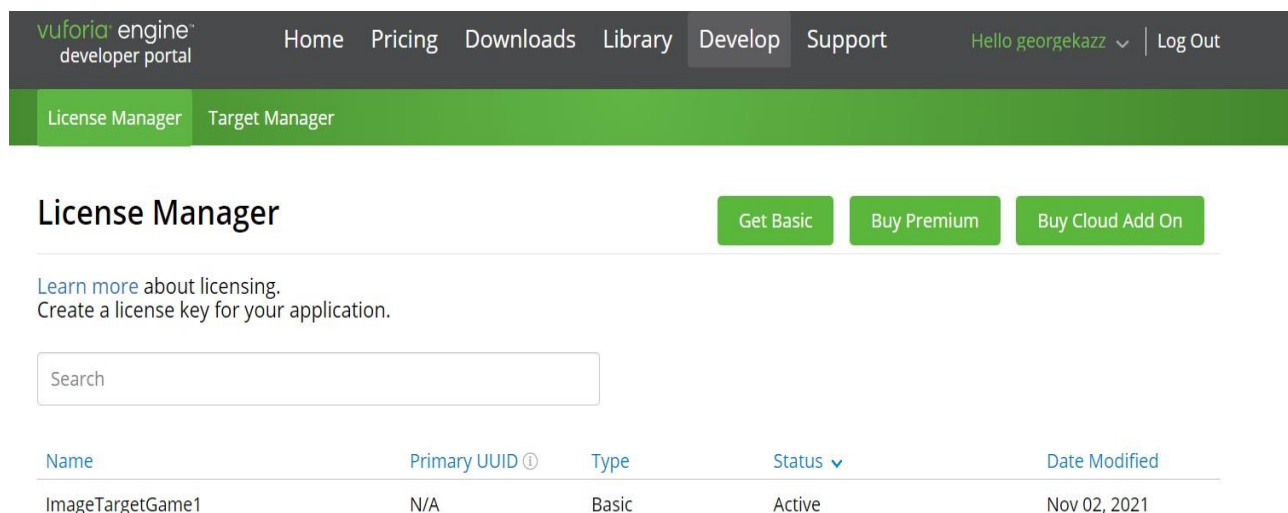
Το Vuforia αποτελεί την αρχή και το τέλος της εφαρμογής, καθώς όλα τα QR codes που χρησιμοποιήθηκαν προέρχονται από αυτό. Έχει αναπτυχθεί από την Qualcomm και με την βοήθεια της τεχνητής νοημοσύνης και της κάμερας του κινητού μπορεί να αναγνωρίζει εικόνες και αντικείμενα. Οι χρήστες μπορούν να επωφεληθούν από αυτές τις δυνατότητες και μέσω διάφορων εικόνων μπορούν να εμφανίζονται επαυξημένα αντικείμενα. Είναι ένα περιβάλλον που μπορεί να συνδέσει το περιβάλλον του Unity με τις δυνατότητες που έχει το Vuforia. Στην συγκεκριμένη εφαρμογή που αναπτύχθηκε χρησιμοποιούνται για εικόνες διάφορα QR codes (σχήμα 3.7) το κάθε ένα ξεχωριστά για κάθε παιχνίδι.



Εικόνα 3.7 QR code

3.5.1 Εγκατάσταση Vuforia και σύνδεση με Unity

Όπως αναφέρθηκε παραπάνω το Vuforia Engine αποτελεί τον κορμό αυτής της εφαρμογής. Για να μπορέσουμε να το χρησιμοποιήσουμε πρώτα από όλα πρέπει να δημιουργήσουμε έναν ιδιωτικό λογαριασμό στην ιστοσελίδα της Vuforia. Μετά την εγγραφή επόμενο βήμα είναι να εμφανιστεί το περιβάλλον της αρχικής σελίδας όπως φαίνεται και στην Εικόνα 3.8. Στην συνέχεια ακολουθεί η δημιουργία βάσης και το ανέβασμα (upload) των εικόνων που θα χρησιμοποιηθούν (εικόνα 3.9). Τα QR codes θα μας επιτρέψουν με την βοήθεια του Vuforia να εμφανίσουμε τον επαυξημένο κόσμο επάνω σε αυτά.



License Manager Get Basic Buy Premium Buy Cloud Add On

[Learn more](#) about licensing.
Create a license key for your application.

Search

Name	Primary UUID ⓘ	Type	Status ▼	Date Modified
ImageTargetGame1	N/A	Basic	Active	Nov 02, 2021

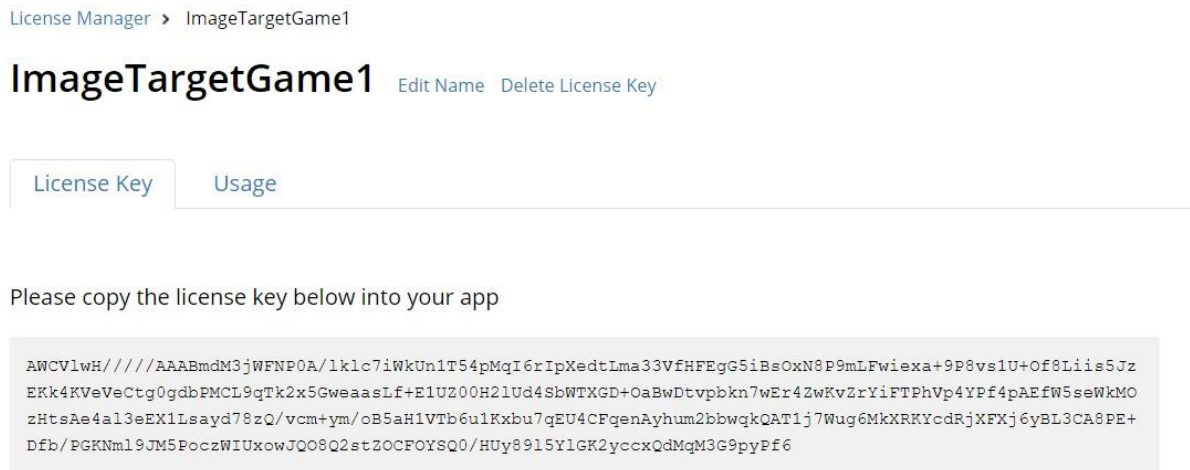
Εικόνα 3.8 Αρχική σελίδα Vuforia

☐ Target Name	Type	Rating ⓘ	Status ▼	Date Modified
☐  ClimateGame	Single Image	★★★★★	Active	Feb 05, 2022 16:36
☐  PopulationGame	Single Image	★★★★★	Active	Jan 22, 2022 20:07
☐  MountainGame	Single Image	★★★★★	Active	Jan 15, 2022 18:15
☐  NisiaGame	Single Image	★★★★★	Active	Dec 31, 2021 12:19
☐  ThesiGame	Single Image	★★★★★	Active	Dec 19, 2021 14:47
☐  game2	Single Image	★★★★★	Active	Dec 06, 2021 20:11
☐  ar_marker2	Single Image	★★★★★	Active	Nov 25, 2021 19:00

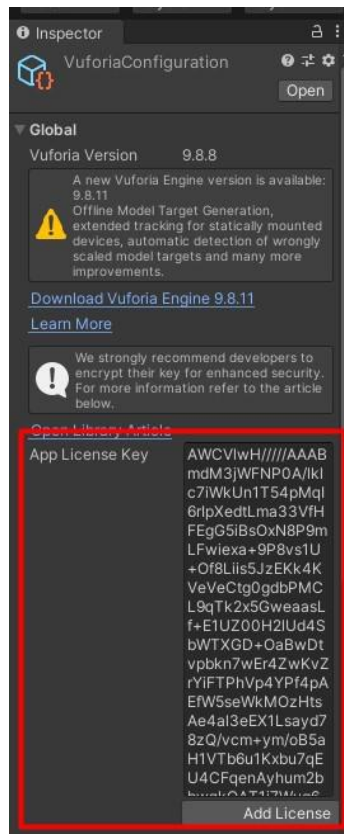
Εικόνα 3.9 QR codes στην σελίδα της Vuforia

Στην συνέχεια, μέσα από την ιστοσελίδα της Vuforia στην κατηγορία Develop και από το μενού License Manager γίνεται η αντιγραφή του License Key. Αυτό το κλειδί θα μας επιτρέψει να ολοκληρώσουμε την επικοινωνία της Vuforia με το Unity (Εικόνα 3.10). Από το περιβάλλον της Unity, σαν πρώτο βήμα, θα πρέπει να τοποθετηθεί μία AR Camera, η οποία βρίσκεται μέσα στην

κατηγορία Vuforia Engine, μέσα στην σκηνή της Unity. Έπειτα, αφού έχουμε αντιγράψει το κλειδί από την σελίδα της Vuforia, μέσω της Unity πάμε στην κατηγορία Window → Vuforia Configuration και μέσα στο αρχείο, στο πεδίο App License Key γίνεται η επικόλληση του License Key (Εικόνα 3.11).



Εικόνα 3.10 Κλειδί από την ιστοσελίδα της Vuforia



Εικόνα 3.11 Επικόλληση κλειδιού στο Unity

Με την ολοκλήρωση των παραπάνω βημάτων μπορεί να γίνει η χρήση της Vuforia μέσω της συσκευής μας, αρκεί να τοποθετηθεί μια AR Camera μέσα στην σκηνή της Unity και στην συνέχεια να γίνει η εγκατάσταση της σε μια συσκευή (κινητό, tablet κλπ.). Η επιτυχής σύνδεση αυτών των δυο θα γίνεται αντιληπτή αν στο κάτω μέρος της οθόνης αναγράφεται το όνομα «Vuforia» όπως φαίνεται στην Εικόνα 3.12.

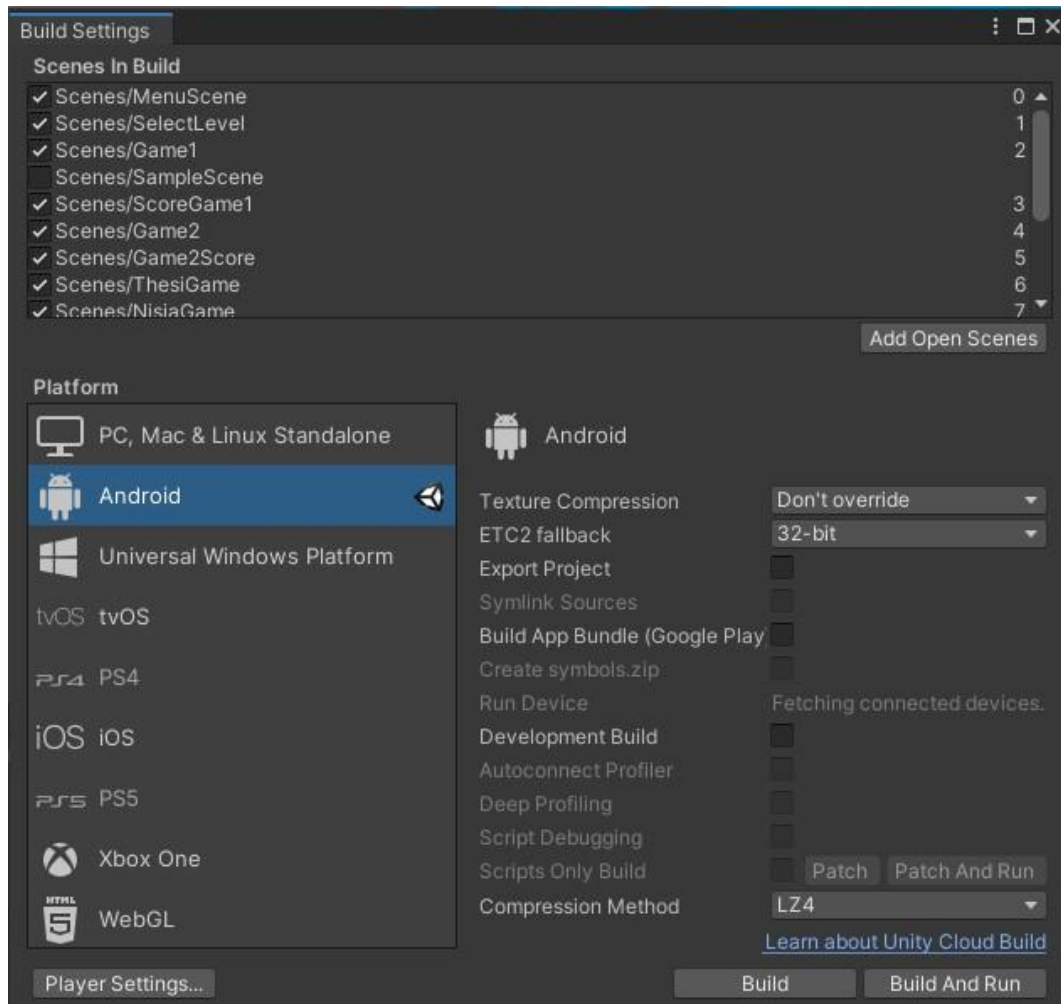


Εικόνα 3.12 Σήμα Vuforia στην συσκευή

3.6 Ρυθμίσεις Unity για συσκευές με Android λογισμικό

Η Unity παρέχει μια λειτουργία μέσα από την οποία μπορεί ο καθένας να χρησιμοποιήσει τις εφαρμογές του, μέσα από διάφορες πλατφόρμες (Mac, Windows, Android, iOS κλπ.). Η εφαρμογή της συγκεκριμένης διπλωματικής έχει δημιουργηθεί για κινητά με Android λογισμικό. Οι ρυθμίσεις που χρειάζονται να γίνουν μέσα στο περιβάλλον της Unity είναι πολύ απλές. Αρχικά, στην κατηγορία

File → Build Settings στο πάνω μέρος του παραθύρου, γίνεται η επιλογή των σκηνών που έχουν δημιουργηθεί για την εφαρμογή, και στο κάτω μέρος γίνεται η επιλογή της πλατφόρμας που θα χρησιμοποιήσουμε. Στην προκειμένη περίπτωση επιλέχτηκε το Android με τις αντίστοιχες σκηνές της εφαρμογής (Εικόνα 3.13).



Εικόνα 3.13 Παράθυρο Build Settings

3.7 Επίλογος

Η πλατφόρμα της Unity αποτελεί μια εξαιρετικά χρήσιμη εφαρμογή για την ανάπτυξη παιχνιδιών και με την βοήθεια της Vuforia Engine γίνεται πιο απλή η ανάπτυξη εφαρμογών με την χρήση της επανξιμένης πραγματικότητας. Η εφαρμογή Blender αποτελεί μια εύχρηστη πλατφόρμα για την δημιουργία τρισδιάστατων αντικειμένων και την δυνατότητα δημιουργίας κινούμενων εικόνων.

Κεφάλαιο 4ο: Βασικός κορμός παιχνιδιού

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει η περιγραφή του βασικού κορμού του παιχνιδιού μαζί με κομμάτια κώδικα που υλοποιήθηκαν για τις βασικές λειτουργίες. Επιπλέον θα γίνει μια μικρή παρουσίαση όλων των παιχνιδιών μαζί με τα κεφάλαια που αντιστοιχούν στο καθένα από αυτά.

4.2 Έναρξη εφαρμογής

Με την εγκατάσταση της εφαρμογής σε μια συσκευή, στην αρχική οθόνη εμφανίζεται η εφαρμογή με το όνομα GeographyAR και με εικόνα εφαρμογής όπως φαίνεται στην εικόνα 4.1. Η εικόνα σχεδιάστηκε και χρωματίστηκε εξ ολοκλήρου με την βοήθεια του Photoshop.

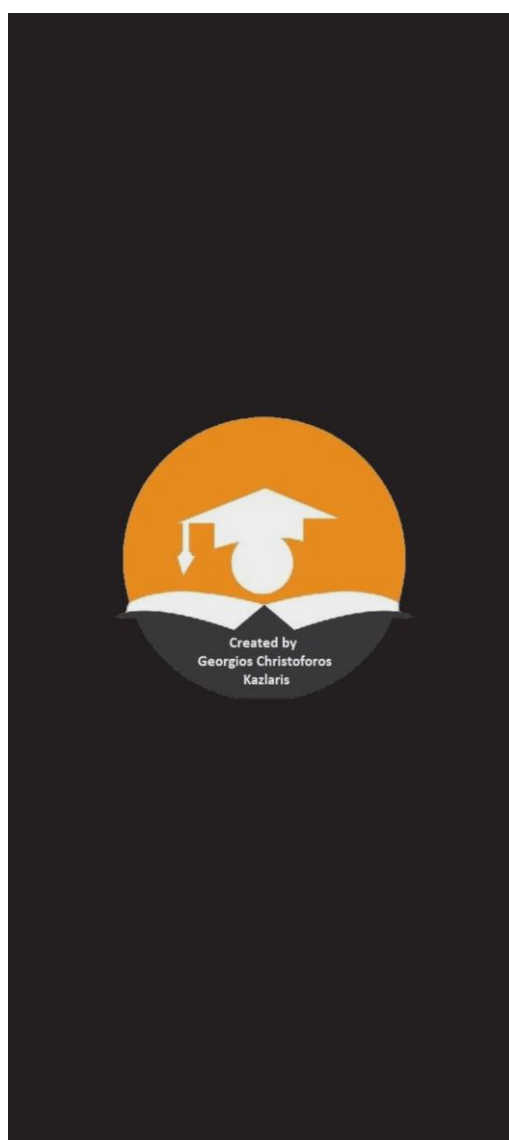


Εικόνα 4.1 Εικόνα Εφαρμογής

Με το που γίνει η έναρξη της εφαρμογής θα εμφανιστούν δύο εικόνες εισαγωγής με διαφορά περίπου ενός δευτερολέπτου η μια από την άλλη. Η πρώτη εικόνα (Εικόνα 4.2) που εμφανίζεται είναι προεπιλογή της Unity, ενώ η δεύτερη (Εικόνα 4.3) η εικόνα που ακολουθεί είναι υλοποιημένη με την βοήθεια του Photoshop. Αυτή η εναλλαγή των εικόνων γίνεται με την βοήθεια μιας λειτουργίας που παρέχεται από την Unity και ονομάζεται Splash Screen. Η λειτουργία αυτή βρίσκεται στο μενού Edit → Project Settings → Splash Image.



Εικόνα 4.2 Πρώτο Splash Image



Εικόνα 4.3 Δεύτερο Splash Image

Το πρώτο πράγμα που αντικρίζει κάποιος αμέσως μετά είναι το αρχικό μενού της εφαρμογής (Εικόνα 4.4). Σε αυτό το σημείο δίνεται στο χρήστη η δυνατότητα να επιλέξει ανάμεσα σε τρία διαφορετικά κουμπιά, και στο επάνω μέρος της οθόνης εμφανίζεται με ένα TextMeshPro Text το όνομα της εφαρμογής. Κεντρικά είναι τοποθετημένο το λογότυπο της εφαρμογής ενώ παράλληλα ακούγεται μουσική, κάνοντας το περιβάλλον που περιηγείται ο χρήστης πιο φιλικό.



Εικόνα 4.4 Αρχικό Menu

4.2.1 Επιλογή κουμπιών

Όπως αναφέρθηκε παραπάνω, στο Αρχικό Μενού υπάρχουν τρία κουμπιά με τα ονόματα «Σχετικά», «Ξεκινάμε» και «Εξοδος». Κάθε κουμπί οδηγεί τον χρήστη στην ανάλογη σκηνή της εφαρμογής. Αρχικά, με το πάτημα του κουμπιού «Εξοδος», κλείνει η εφαρμογή και επιστρέφει τον χρήστη στο Κεντρικό μενού της συσκευής. Αυτό επιτυγχάνεται μέσω του script «QuitGame» με την βοήθεια του κώδικα όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 4.5).

```
void onClickQuitGame() {
    Application.Quit();
}
```

Εικόνα 4.5 Κώδικας για κλείσιμο εφαρμογής

Το κουμπί «Σχετικά» καθοδηγεί τον χρήστη σε μια νέα σκηνή με όνομα AboutScene, όπου εκεί εμφανίζονται τα στοιχεία του δημιουργού της εφαρμογής καθώς και τα copyrights (Εικόνα 4.6).



Εικόνα 4.6 About Scene

Από την άλλη το κουμπί «Ξεκινάμε!» οδηγεί τον χρήστη σε μια νέα σκηνή με όνομα SelectLevel. Αυτή η σκηνή αποτελεί το μενού που περιέχει όλα τα παιχνίδια της εφαρμογής (Εικόνα 4.7). Σε αυτή την σκηνή το κάθε κουμπί αντιστοιχεί σε ένα παιχνίδι. Τα παιχνίδια χωρίζονται σε τρεις ενότητες ανάλογα με το κεφάλαιο στο οποίο βρίσκονται στο βιβλίο.



Εικόνα 4.7 SelectLevel Scene

Η περιφορά του χρήστη από σκηνή σε σκηνή γίνεται με την βοήθεια της εντολής SceneManager η οποία παρέχεται από την Unity. Η εντολή SceneManager χρησιμοποιεί τις βιβλιοθήκες UnityEngine και UnityEngine.SceneManagement. Αντίστοιχα στην σκηνή με τα παιχνίδια, η σύνδεση του χρήστη με το κάθε παιχνίδι γίνεται πάλι με την βοήθεια του SceneManager μέσω του LevelSelector script (Εικόνα 4.8)

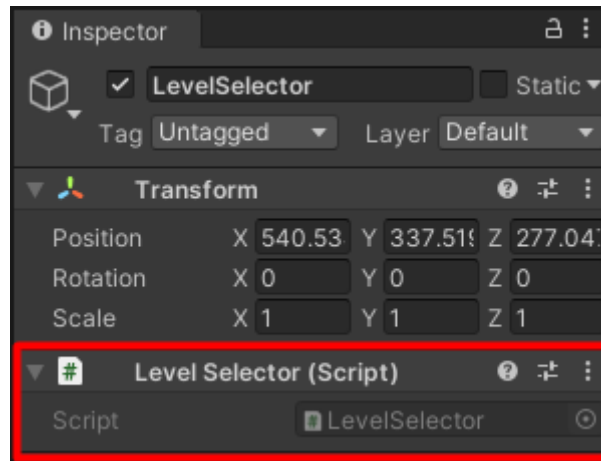
```

C: > GeographyAR > GeographyAR > Assets > scripts > LevelSelector.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class LevelSelector : MonoBehaviour
7  {
8      // push button to go to the next scene
9      public void startGame1() {
10         SceneManager.LoadScene("IntroGame1");
11     }
12
13     public void startGame2() {
14         SceneManager.LoadScene("Game2");
15     }
16
17     public void startGame3() {
18         SceneManager.LoadScene("ThesiGame");
19     }
20
21     public void startGame4() {
22         SceneManager.LoadScene("NisiaGame");
23     }
24
25     public void startGame5() {
26         SceneManager.LoadScene("MountainScene");
27     }
28
29     public void startGame6() {
30         SceneManager.LoadScene("ClimateGame");
31     }
32
33     public void startGame7() {
34         SceneManager.LoadScene("PopulationGame");
35     }
36
37

```

Εικόνα 4.8 Script LevelSelector

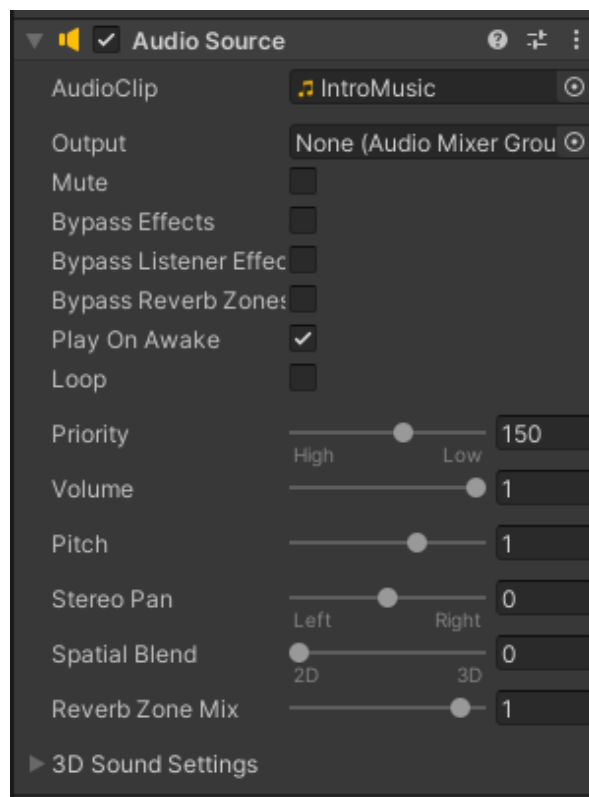
Τα scripts από μόνα τους δεν μπορούν να λειτουργήσουν εκεί που πρέπει και όταν πρέπει. Για αυτό τον λόγο υπάρχουν μέσω της Unity τα Empty Objects. Στην ουσία είναι άδεια αντικείμενα τα οποία αν τοποθετηθούν πάνω στην σκηνή είναι αόρατα στον χρήστη. Παρόλα αυτά, εξυπηρετούν στο να τοποθετούνται μέσα σε αυτά τα scripts. Από εκεί και πέρα η χρήση του script γίνεται μέσω του Empty Object το οποίο είναι καθ' όλη την διάρκεια ενεργοποιημένο όταν ο χρήστης βρίσκεται στην συγκεκριμένη σκηνή. Παρόλα αυτά, υπάρχει και η δυνατότητα ενεργοποίησης και απενεργοποίησης των Empty Objects, και αυτό εξυπηρετεί στο να ενεργοποιούμε ή να απενεργοποιούμε κομμάτια κώδικα (Εικόνα 4.9).



Εικόνα 4.9 Τοποθέτηση script σε Empty Object

4.3 Μουσική υπόκρουση

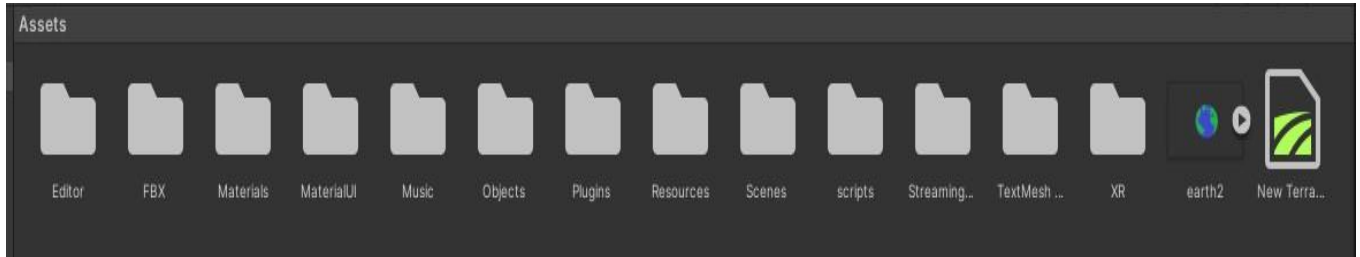
Σε όλες αυτές τις σκηνές υπάρχει μια μουσική υπόκρουση. Η ένταξη μουσικής στην κάθε σκηνή έγινε με την βοήθεια μιας λειτουργίας που προσφέρεται από την Unity μαζί με ένα Empty Object. Αυτή η λειτουργία ονομάζεται Audio Source όπου μέσα τοποθετήθηκε το μουσικό κομμάτι για την μουσική υπόκρουση της εφαρμογής (Εικόνα 4.10).



Εικόνα 4.10 Παράθυρο Audio Source στο Unity

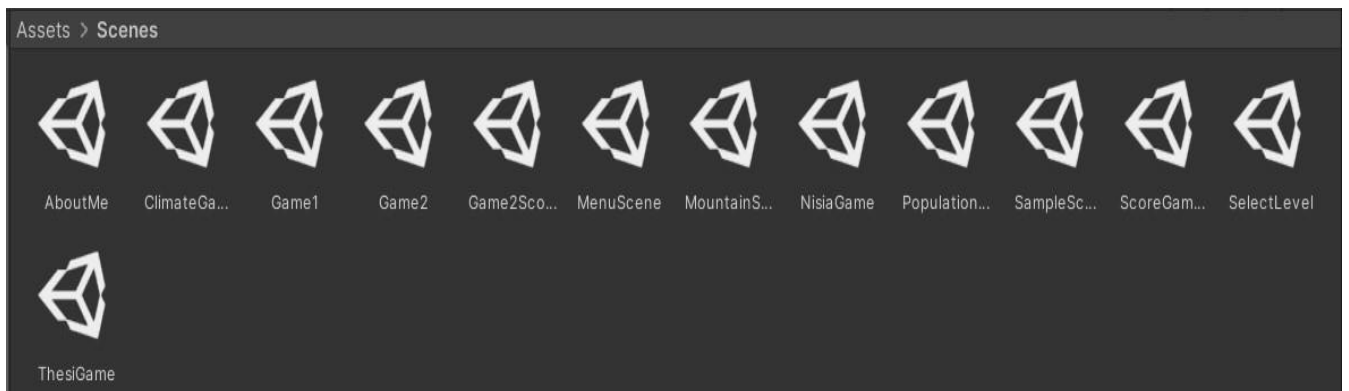
4.4 Βασικά στοιχεία του Project

Μέσα σε ένα Unity Project υπάρχουν πολλά στοιχεία τα οποία ονομάζονται Assets. Από αυτά τα assets αποτελείται όλο το Project και μπορεί να είναι ήχοι, εικόνες, materials, scripts κλπ. Με βάση την Εικόνα 4.11 παρατηρείτε το γεγονός ότι μέσα στα assets του συγκεκριμένου Project υπάρχουν πολλοί φάκελοι οι οποίοι περιέχουν διάφορα στοιχεία ή ακόμα και υπό-φάκελους. Αυτό γίνεται κυρίως για να υπάρχει οργάνωση στο Project ώστε να μπορεί να βρίσκει ο χρήστης εύκολα και γρήγορα τα στοιχεία που χρειάζεται.



Εικόνα 4.11 Όλα τα assets του Project

Όλες οι σκηνές που υπάρχουν στην εφαρμογή βρίσκονται μέσα στο φάκελο Scene (Εικόνα 4.12)



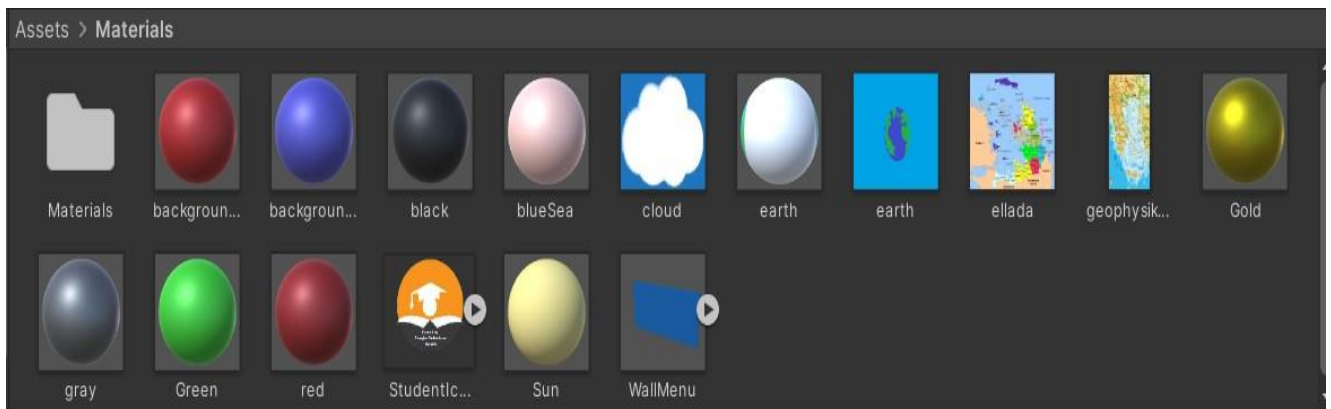
Εικόνα 4.12 Φάκελος με τα Scenes

Τα όλα τα scripts βρίσκονται στον φάκελο scripts, όπου αντίστοιχα εκεί χωρίζονται σε άλλους υπό-φάκελους ανάλογα με το κάθε παιχνίδι (Εικόνα 4.13).



Εικόνα 4.13 Φάκελος με τα Scripts

Όλα τα materials που χρησιμοποιούνται σε αυτό το Project βρίσκονται στον φάκελο Materials όπου και εκεί υπάρχουν άλλοι υπό-φάκελοι οι οποίοι έχουν δημιουργηθεί για το κάθε παιχνίδι ξεχωριστά (Εικόνα 4.14).



Εικόνα 4.14 Φάκελος με τα Materials

4.5 Επίλογος

Όλα τα παιχνίδια χρησιμοποιούν μια συγκεκριμένη σειρά από την αρχή μέχρι το τέλος, και καθοδηγούν το χρήστη σε κάθε παιχνίδι ώστε να μπορέσει να φέρει εις πέρας την λύση του κάθε παιχνιδιού. Με την έναρξη της εφαρμογής γίνεται η εμφάνιση του αρχικού μενού. Στην συνέχεια με την βοήθεια των κουμπιών και των οδηγιών που εμφανίζονται, δίνετε η δυνατότητα στον χρήστη να περιηγηθεί στην εφαρμογή. Στην σκηνή των παιχνιδιών υπάρχουν όλα τα παιχνίδια όπου με την σειρά τους καθοδηγούν τον χρήστη να επιλέξει ένα από αυτά μέχρι να το ολοκληρώσει. Τέλος, με την ολοκλήρωση του κάθε παιχνιδιού ο χρήστης επιστρέφει στο αρχικό μενού (Εικόνα 4.15).

Κεφάλαιο 5ο: Τα είδη των χαρτών

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφεται το παιχνίδι «Τα είδη των χαρτών» το οποίο αποτελεί και το πρώτο παιχνίδι της εφαρμογής. Το κεφάλαιο αυτό βρίσκεται στην σελίδα 13 του βιβλίου, αναλύει τα είδη των χαρτών που υπάρχουν εξηγώντας την χρήση του κάθε χάρτη ξεχωριστά. Τέλος, γίνεται ανάλυση όλων των στοιχείων, των σκηνών και του κώδικα που χρησιμοποιήθηκαν για την υλοποίηση αυτού του παιχνιδιού.

5.2 Δομή Παιχνιδιού

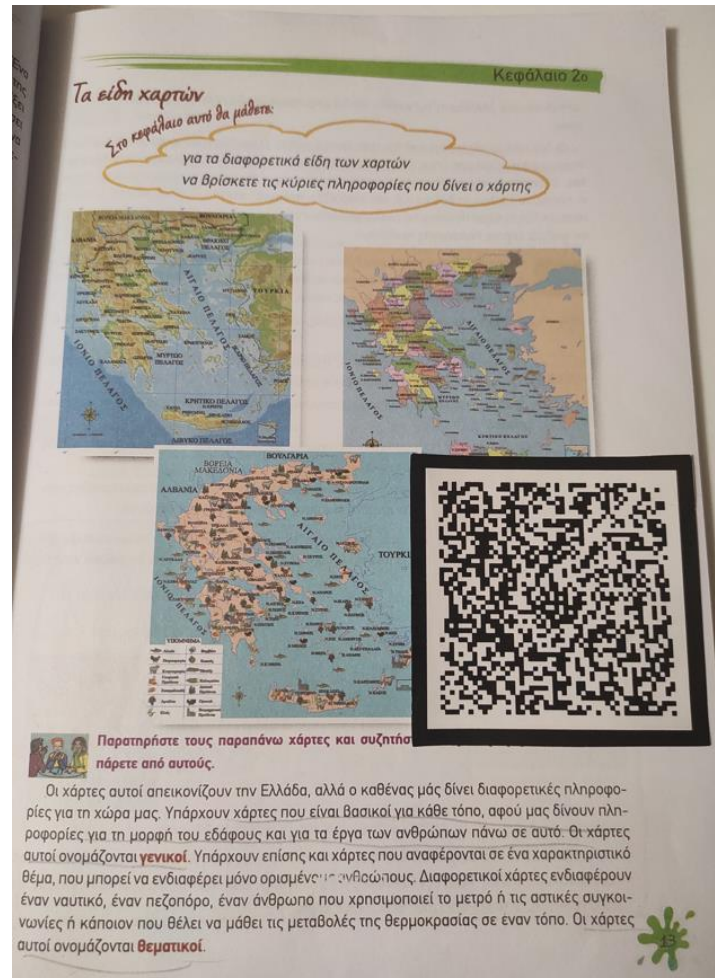
Το παιχνίδι χωρίζεται σε τρία τμήματα, το καλωσόρισμα του χρήστη, το παιχνίδι, και το αποτέλεσμα. Το καλωσόρισμα αποτελείται από μια δισδιάστατη σκηνή που εξηγεί το παιχνίδι. Έπειτα ακολουθεί το παιχνίδι επαυξημένης πραγματικότητας με ερωτήσεις και επιλογές απαντήσεων, και τέλος εμφανίζεται μια δισδιάστατη σκηνή όπου παρουσιάζονται τα αποτελέσματα από το παιχνίδι.

Στην πρώτη φάση του καλωσορίσματος, ο χρήστης με το που συνδεθεί στο παιχνίδι μπαίνει στην πρώτη δισδιάστατη σκηνή όπου και γίνεται το καλωσόρισμα, μαζί με τις οδηγίες του παιχνιδιού για να μπορέσει ο χρήστης να εξοικειωθεί καλύτερα με το περιβάλλον και να καταλάβει την λογική του παιχνιδιού. Οι οδηγίες είναι σε μορφή κειμένου (text) ενώ η εικόνα που εμφανίζεται είναι τοποθετημένη πάνω σε ένα Raw Image. Τα κουμπιά «Ξεκινάμε» και «Πίσω» δίνουν την επιλογή στον χρήστη να συνεχίσει στο παιχνίδι ή να γυρίσει στο Αρχικό Μενού (Εικόνα 5.1).



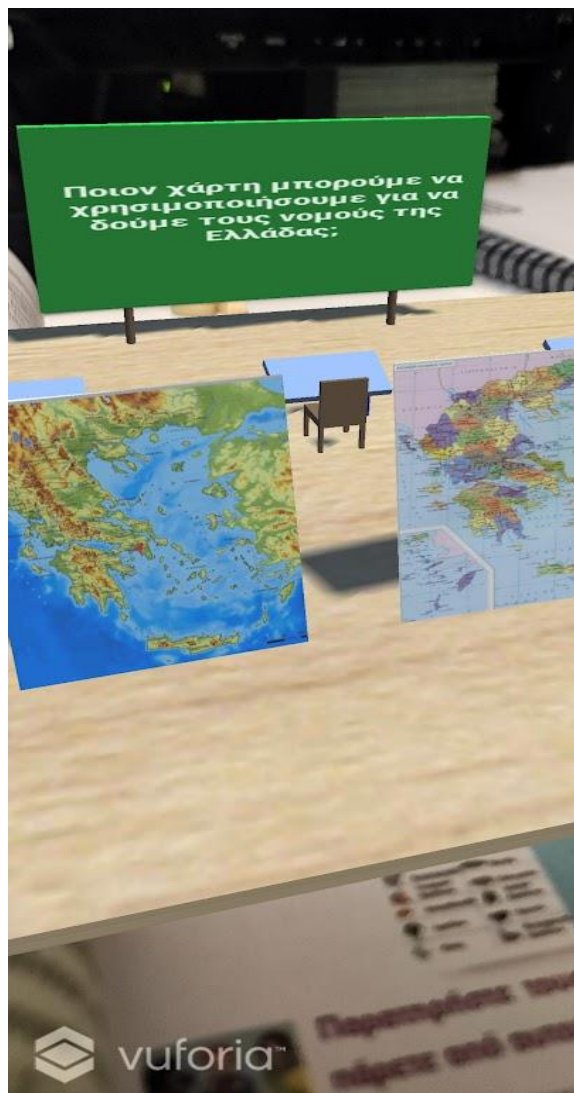
Εικόνα 5.1 Δισδιάστατη σκηνή φάσης πρώτης

Στην δεύτερη φάση που είναι το κύριο παιχνίδι, ο χρήστης θα πρέπει σαν πρώτο βήμα να σκανάρει το QR code το οποίο βρίσκεται στην Α' Ενότητα δεύτερο κεφάλαιο στην σελίδα 13 του βιβλίου (Εικόνα 5.2). Το σκανάρισμα γίνεται από την κάμερα της συσκευής και η υλοποίηση έγινε με την βοήθεια μιας AR Camera και του Vuforia.



Εικόνα 5.2 QR code 1^{ου} παιχνιδιού στο σχολικό βιβλίο

Με το που σκανάρει ο χρήστης το QR code θα εμφανιστεί μπροστά του η σκηνή Game1 η οποία περιέχει το περιβάλλον του παιχνιδιού. Το περιβάλλον αυτό αντικατοπτρίζει μια αίθουσα διδασκαλίας και περιλαμβάνει μερικά θρανία σχολείου, έναν σχολικό πίνακα, και τρεις χάρτες οι οποίοι βρίσκονται στο πίσω μέρος της αίθουσας. Στόχος του πρώτου παιχνιδιού είναι να απαντήσει ο χρήστης σωστά σε 10 ερωτήσεις, με βάση την ύλη αυτού του κεφαλαίου. Οι ερωτήσεις εμφανίζονται με τυχαία σειρά κάθε φορά πάνω στον σχολικό πίνακα. Ο χρήστης αφού διαβάσει την ερώτηση έχει την δυνατότητα να επιλέξει την σωστή απάντηση πατώντας πάνω σε έναν από τους τρεις χάρτες που υπάρχουν μπροστά του. Με το που γίνει η επιλογή της απάντησης υπάρχει αναμονή 2 δευτερολέπτων μέχρι να εμφανιστεί η επομένη τυχαία ερώτηση. (Εικόνα 5.3)



Εικόνα 5.3 Η δεύτερη φάση του παιχνιδιού

Η υλοποίηση της δεύτερης φάσης του παιχνιδιού έγινε με την χρήση του script «QuizManager» το οποίο είναι γραμμένο σε C# όπως και όλα τα scripts γενικά σε αυτή την εφαρμογή. Αρχικά, το script χρησιμοποιεί κάποιες συγκεκριμένες βιβλιοθήκες για να μπορούν να χρησιμοποιηθούν ορισμένα εργαλεία στον κώδικα (Εικόνα 5.4), και στην συνέχεια γίνεται η αρχικοποίηση όλων των μεταβλητών που χρησιμοποιούνται στο script (Εικόνα 5.5). Η μεταβλητή «questions» αποτελεί μία λίστα τύπου QnA (Εικόνα 5.6) όπου περιέχει όλες τις ερωτήσεις του παιχνιδιού μαζί με την σωστή απάντηση που θα πρέπει να δώσει ο χρήστης. Το μέγεθος της λίστας αλλάζει με βάση την τιμή που του δίνεται μέσω της Unity. Η λίστα unansweredQuestions κρατάει όλες τις ερωτήσεις που δεν έχουν απαντηθεί ακόμα από τον χρήστη, ώστε όταν απαντηθούν όλες οι ερωτήσεις, η λίστα αυτή να αδειάζει και ο χρήστης να μεταφέρετε στην επόμενη φάση των αποτελεσμάτων. Οι μεταβλητές geomorfologikosTag, politikosTag και proiontonTag είναι μεταβλητές τύπου Strings οι οποίες χρησιμοποιούνται ως Tags πάνω στους τρεις χάρτες αντίστοιχα που εμφανίζονται μπροστά στον χρήστη. Τα Tags αποτελούν ένα αναγνωριστικό στοιχείο για τα αντικείμενα, και δημιουργούνται μέσα από την Unity. Οι υπόλοιπες μεταβλητές τύπου Text αποτελούν κάποια κείμενα που εμφανίζονται στον χρήστη. Τέλος, υπάρχουν

μεταβλητές τύπου Integers που χρησιμοποιούνται ως μετρητές (counters) για να υπολογίζουν το άθροισμα των σωστών και των λανθασμένων απαντήσεων.

```
C: > GeographyAR > GeographyAR > Assets > scripts > game1 > quizManager.cs
1  using System.Collections;
2  using UnityEngine;
3  using UnityEngine.UI;
4  using System.Collections.Generic;
5  using System.Linq;
6  using TMPro;
7  using Vuforia;
8  using System.Threading;
9  using UnityEngine.Events;
10 using UnityEngine.SceneManagement;
11 using System;
12 using Random=UnityEngine.Random;
```

Εικόνα 5.4 Εισαγωγή βιβλιοθηκών

```
public QnA[] questions;
private static List<QnA> unansweredQuestions;
private QnA currentQuestion;

[SerializeField] private string GeomorfologikosTag = "GeomorfologikosTag";
[SerializeField] private string PolitikosTag = "PolitikosTag";
[SerializeField] private string ProiontonTag = "ProiontonTag";

public GameObject BoardText;
private Text onBoardText;
private Text text;
public int sotesApantisis,lathosApantisis;

[SerializeField]
private Text QuestionText;

[SerializeField]
private float timeBetweenQuestions = 2f;
```

Εικόνα 5.5 Αρχικοποίηση μεταβλητών

```

C: > GeographyAR > GeographyAR > Assets > scripts > game1 > QnA.cs
1
2 [System.Serializable]
3 public class QnA
4 {
5     public string Question;
6     public int isTrue;
7 }
8

```

Εικόνα 5.6 QnA.cs script

Στην συνέχεια του κώδικα, γίνεται η χρήση της μέθοδο Start() τύπου void (Εικόνα 5.7). Αυτή η μέθοδος ενεργοποιείται με το που ξεκινάει το script, και δίνει ένα κείμενο σε Text πάνω στον σχολικό πίνακα. Έπειτα, μέσω της εντολής Invoke, καλεί μετά από 4 δευτερόλεπτα την μέθοδο Update(), ενώ αμέσως μετά καλεί με τον ίδιο τρόπο την μέθοδο SetRandomQuestion όπου σε αυτή την μέθοδο γίνεται η επιλογή, με τυχαία σειρά, μιας ερώτησης (Εικόνα 5.8).

```

35 void Start (){
36
37     onBoardText = BoardText.GetComponent<Text>();
38     Debug.Log(onBoardText);
39
40     onBoardText.text = "Ηρθε η ώρα να ξεκινήσουμε το πρώτο μας παιχνίδι";
41     Invoke("Update",8.0f);
42
43     if (unansweredQuestions == null || unansweredQuestions.Count == 0){
44         unansweredQuestions = questions.ToList<QnA>();
45     }
46
47     Invoke("SetRandomQuestion", 8.0f);
48     int totalQuestions = 15;
49
50 }

```

Εικόνα 5.7 Μέθοδος Start()

```

void SetRandomQuestion () {
    int randomQuestionIndex = Random.Range (0, unansweredQuestions.Count);
    currentQuestion = unansweredQuestions[randomQuestionIndex];

    QuestionText.text = currentQuestion.Question;
}

```

Εικόνα 5.8 Μέθοδος SetRandomQuestion()

Έπειτα, η μέθοδος Update() τύπου void, καλείται σε όλη την διάρκεια που λειτουργεί το script. Σε αυτή την μέθοδο χρησιμοποιείται η εντολή GetMouseButtonDown η οποία περιμένει από τον χρήστη να πατήσει στην οθόνη πάνω σε έναν από τους τρεις χάρτες. Στην συνέχεια, μια μεταβλητή τύπου Ray αναγνωρίζει την ακριβή τοποθεσία που πάτησε ο χρήστης επάνω στην οθόνη και παίρνει την συγκεκριμένη τιμή. Μια επιπλέον συνθήκη If ελέγχει αν το πάτημα που έγινε πάνω στην οθόνη, έγινε σε κάποιο αντικείμενο (Object) της σκηνής και έπειτα ελέγχει αν το αντικείμενο που πατήθηκε είναι ένα από τα αντικείμενα με Tags (Εικόνα 5.9).

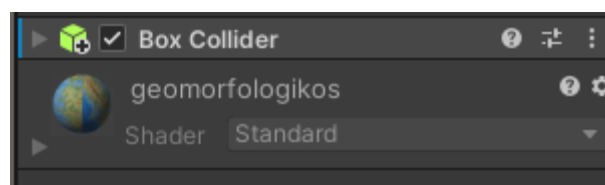
```

52 void Update()
53 {
54     if (Input.GetMouseButtonDown(0))
55     {
56         Debug.Log("Mouse button");
57         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
58         RaycastHit hit;
59
60         if (Physics.Raycast(ray, out hit))
61         {
62             Debug.Log("raycast");
63             var selection = hit.transform;
64         }
65     }
66 }

```

Εικόνα 5.9 Εντολή GetMouseButtonDown

Για να γίνει η αναγνώριση ενός αντικειμένου που πατήθηκε μέσω της οθόνης, δεν αρκεί μόνο να γίνει ο έλεγχος των παραπάνω πραγμάτων. Μέσω του Unity πρέπει να προστεθούν οι κατάλληλες ρυθμίσεις πάνω στα αντικείμενα, για να μπορεί με αυτό τον τρόπο το Unity να ξεχωρίζει τα αντικείμενα που επιλέχτηκαν ξεχωριστά. Το Box Collider (Εικόνα 5.10), αποτελεί ένα εργαλείο (Component) της Unity το οποίο τοποθετείται πάνω στα αντικείμενα, στο κάθε ένα ξεχωριστά, και χρησιμοποιείται για την ενεργοποίηση συμβάντων.



Εικόνα 5.10 Box Collider

Αφού λοιπόν γίνει η επιλογή ενός αντικειμένου μέσω της οθόνης, η Update() με μια επιπλέον If ελέγχει την ερώτηση που εμφανίζεται στον σχολικό πίνακα. Στην συνέχεια, αν ο χάρτης που επιλέχτηκε ταιριάζει με την ερώτηση που υπάρχει στον σχολικό πίνακα τότε ο μετρητής stostesApantisis αυξάνει την τιμή του κατά ένα, δίνοντας έτσι μια επιπλέον μονάδα στις συνολικές σωστές απαντήσεις. Από την άλλη, σε περίπτωση που η επιλογή του χάρτη από τον χρήστη δεν ταιριάζει με την ερώτηση που υπάρχει στον σχολικό πίνακα τότε ο μετρητής lathosApantisis αυξάνει την τιμή του κατά ένα, δίνοντας στις συνολικές λανθασμένες απαντήσεις μια μονάδα. Σε κάθε

περίπτωση, με το που γίνει η επιλογή ενός αντικειμένου καλείται έπειτα η μέθοδος `TransitionToNextQuestion()` (Εικόνα 5.11).

```

if (selection.CompareTag(GeomorfologikosTag)){ //geomorfologikos 1-----
    if(onBoardText.text == "Ποιος από τους χάρτες είναι ο Γεωμορφολογικός;"){
        sotesApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }else if(onBoardText.text == "Θέλω να δω ποια περιοχή είναι πεδινή και ποια είναι ορεινή. Ποια χάρτη θα κοιτάξω;"){
        sotesApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }else{
        lathosApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }
}

else if (selection.CompareTag(PolitikosTag)){ //politikos 2-----
    if(onBoardText.text == "Ποιον χάρτη μπορούμε να χρησιμοποιήσουμε για να δούμε τους νομούς της Ελλάδας;"){
        sotesApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }else if(onBoardText.text == "Ποιος χάρτης είναι ο Πολιτικός;"){
        sotesApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }else if(onBoardText.text == "Μένω στην Θεσσαλονίκη αλλά δεν ξέρω που ακριβώς είναι τα Ιωάννινα για να πάω το σαββατοκύριακο στα φίλα μου. Ποιον χάρτη μπορούμε να χρησιμοποιήσουμε για να δούμε τους νομούς της Ελλάδας;"){
        sotesApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }else if(onBoardText.text == "Ξύλον νομό Ηρακλείου παράγεται πολύ λάδι και τυρί. Ποιον χάρτη θα χρησιμοποιήσω για να δω που βρίσκεται ο νομός αυτός;"){
        sotesApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }else if(onBoardText.text == "Η Θλώρινα είναι ορεινή περιοχή. Ποιον χάρτη μπορώ να δω και να χρησιμοποιήσω που θα μου λέει το που βρίσκεται η Θλώρινα;"){
        sotesApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }else{
        lathosApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }
}

else if (selection.CompareTag(ProiontonTag)) { //proionton 3-----
    if(onBoardText.text == "Θέλω να δω ποια προϊόντα παράγονται στην Θεσσαλία. Ποιον χάρτη θα χρησιμοποιήσω;"){
        sotesApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }else if(onBoardText.text == "Ποιος είναι ο θεματικός χάρτης;"){
        lathosApantisis++;
        StartCoroutine(TransitionToNextQuestion());
    }
}

```

Εικόνα 5.11 Έλεγχος επιλογής αντικειμένου και ερώτησης

Όταν ο χρήστης δώσει την απάντηση του σε μία ερώτηση είτε είναι σωστή είτε είναι λάθος, καλείται η μέθοδος `TransitionToNextQuestion()` (Εικόνα 5.12). Η δουλειά αυτής της μεθόδου είναι να παίρνει την ερώτηση στην οποία απάντησε ο χρήστης και να την αφαιρεί από τις συνολικές ερωτήσεις ώστε να μην υπάρξει το ενδεχόμενο να βγει η ίδια ερώτηση πάνω από μια φορά. Με το που αφαιρεθούν όλες οι ερωτήσεις από την λίστα τότε το κείμενο στο `text` του σχολικού πίνακα αλλάζει ενημερώνοντας τον χρήστη πως το παιχνίδι τελείωσε, και αμέσως μετά καλεί την μέθοδο `EndGame()` (Εικόνα 5.13).

```
IEnumerator TransitionToNextQuestion (){
    unansweredQuestions.Remove(currentQuestion);

    yield return new WaitForSeconds(timeBetweenQuestions);
    if(unansweredQuestions.Count <= 0 ){
        onBoardText.text = "Τέλος Παιχνιδιού!!";
        Invoke("EndGame",4.0f);
    }
    SetRandomQuestion();
}
```

Εικόνα 5.12 Μέθοδος TransitionToNextQuestion()

```
137 public void EndGame () {
138     PlayerPrefs.SetInt("sostesApantisis", sostesApantisis);
139     PlayerPrefs.SetInt("lathosApantisis", lathosApantisis);
140     SceneManager.LoadScene("ScoreGame1");
141 }
142
```

Εικόνα 5.13 Μέθοδος EndGame()

Η EndGame() αποτελεί την τελευταία μέθοδο σε αυτό το script. Αρχικά κρατάει με μια SetInt τους μετρητές για να ξέρει τον αριθμο των σωστών και των λανθασμένων απαντήσεων και στην συνέχεια με την εντολή SceneManager ανοίγει μια καινούργια σκηνή την ScoreGame1 όπου αποτελεί και την τελευταία φάση που είναι τα αποτελέσματα (Εικόνα 5.14). Στην φάση των αποτελεσμάτων ο χρήστης μεταφέρεται αυτόματα στην τελευταία σκηνή μετά την ολοκλήρωση του παιχνιδιού. Η σκηνή αυτή, περιέχει την βαθμολογία των σωστών και των λανθασμένων απαντήσεων σύμφωνα με τις επιλογές του χρήστη. Οι τιμές των αποτελεσμάτων καλούνται μέσω του script Results.cs. Το script περιέχει την μέθοδο Start() όπου με την βοήθεια της εντολής GetInt παίρνει τις τιμές που είχαν γίνει SetInt από το quizManager.cs και τις εμφανίζει πάνω στην σκηνή μέσω των δύο TextMeshProUGUI. Τέλος, δίνεται η δυνατότητα στον χρήστη να επιστρέψει μετά τα αποτελέσματα στο αρχικό μενού της εφαρμογής μέσω ενός κουμπιού «Menu» (Εικόνα 5.15).



Εικόνα 5.14 Σκηνή αποτελεσμάτων

```
14 public class Results : MonoBehaviour
15 {
16     [SerializeField]
17     private TextMeshProUGUI sotes;
18
19     [SerializeField]
20     private TextMeshProUGUI lathos;
21
22     public void Start()
23     {
24         int good, wrong;
25         good = PlayerPrefs.GetInt("sotesApantisis");
26         wrong = PlayerPrefs.GetInt("lathosApantisis");
27
28         String myString = good.ToString();
29         String myString2 = wrong.ToString();
30
31         sotes.text = myString;
32         lathos.text = myString2;
33     }
```

Εικόνα 5.15 Results.cs script

5.3 Επίλογος

Το παιχνίδι για το κεφάλαιο «Τα είδη Χαρτών» της σελίδας 13 του σχολικού βιβλίου στο μάθημα της Γεωγραφίας της Ε' Δημοτικού, υλοποιείται με την χρήση του Vuuforia και της επαυξημένης πραγματικότητας. Ο χρήστης μέσω των ερωτήσεων που καλύπτουν την ύλη αυτού του κεφαλαίου, μαθαίνει για την χρησιμότητα κάθε είδους χάρτη, και μέσω της αλληλεπίδρασης που υπάρχει με το παιχνίδι, γίνεται πιο ευχάριστο για αυτόν.

Κεφάλαιο 6ο: Προσανατολισμός

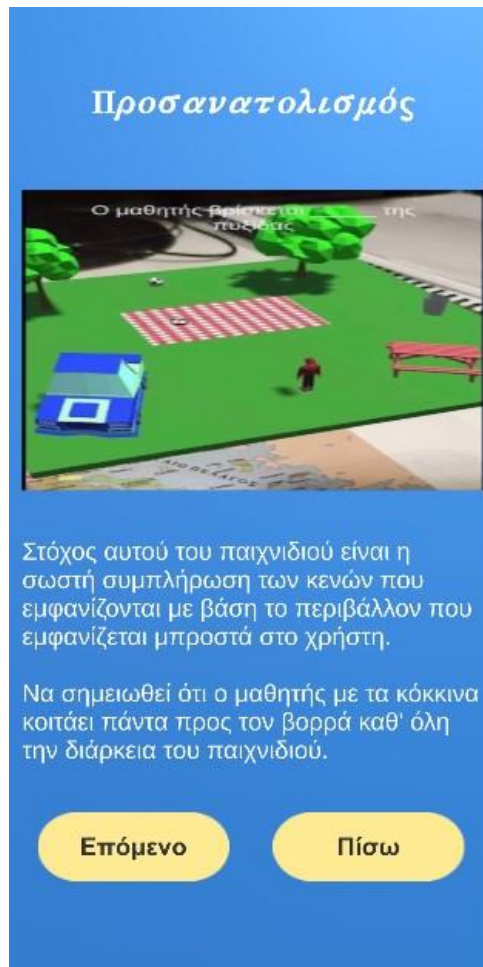
6.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται η περιγραφή του παιχνιδιού για το κεφάλαιο «Προσανατολισμός». Αποτελεί το 5^ο Κεφάλαιο του βιβλίου της Α' Ενότητας και στοχεύει στην καλύτερη κατανόηση του προσανατολισμού και της σχετικής θέσης ενός αντικειμένου πάνω στον χάρτη. Επιπλέον, αναλύονται όλα τα βήματα του παιχνιδιού καθώς και η ανάλυση του κώδικα που χρησιμοποιήθηκε σε αυτό το παιχνίδι.

6.2 Δομή Παιχνιδιού

Το συγκεκριμένο παιχνίδι αποτελεί το 5^ο Κεφάλαιο της Α' Ενότητας του σχολικού βιβλίου στις σελίδες 22-24 στο μάθημα της Γεωγραφίας της Ε' δημοτικού. Στόχος αυτού του παιχνιδιού είναι να προσφέρει στον χρήστη γνώσεις, με ευχάριστο τρόπο, σχετικά με τον προσανατολισμό και να δίνει την ευχέρεια στον μαθητή να προσδιορίζει την σχετική θέση που έχει ένα αντικείμενο. Το παιχνίδι αυτό χωρίζεται σε τρεις ενότητες, την εισαγωγή του παιχνιδιού, το παιχνίδι, και τα αποτελέσματα.

Αρχικά, στην πρώτη ενότητα με το που ξεκινήσει το παιχνίδι ο χρήστης μπαίνει στην πρώτη δισδιάστατη σκηνή (όνομα σκηνής IntroGame2) , όπου παρέχονται οι οδηγίες του συγκεκριμένου παιχνιδιού με σκοπό να καταλάβει ο χρήστης τους κανόνες του παιχνιδιού. Η σκηνή αυτή περιλαμβάνει δύο κείμενα τύπου TextMeshPro, όπου στο πρώτο εμφανίζεται ο τίτλος του κεφαλαίου ενώ στο δεύτερο κείμενο εμφανίζονται οι οδηγίες του παιχνιδιού. Ανάμεσα στα δύο κείμενα, υπάρχει μία εικόνα μέσα από το παιχνίδι, όπου σκοπός της είναι να προετοιμάζει τον χρήστη, ώστε το περιβάλλον στο οποίο μπαίνει να μοιάζει πιο οικείο. Τα δύο κουμπιά που συνοδεύουν την σκηνή καθοδηγούν τον χρήστη να συνεχίσει στο παιχνίδι πατώντας το «Επόμενο», είτε να επιστρέψει στην αρχική σελίδα πατώντας το κουμπί «Πίσω» (Εικόνα 6.1).



Εικόνα 6.1 Σκηνή IntroGame2

Με το πάτημα των κουμπιών έχει προστεθεί ένας ήχος για να είναι σίγουρος ο χρήστης ότι το κουμπί που επέλεξε έχει πατηθεί με επιτυχία. Η λειτουργία του ήχου γίνεται με την βοήθεια κώδικα μέσω του script IntroGame2Script.cs (Εικόνα 6.2). Στην αρχή, ανάλογα με το κουμπί που θα πατηθεί γίνεται η ενεργοποίηση του ClickSound τύπου GameObject το οποίο περιέχει τον ήχο, και στην συνέχεια με το SceneManager γίνεται η αλλαγή της σκηνής. Οι παραπάνω εντολές έγιναν με την βοήθεια των βιβλιοθηκών όπως φαίνονται στην Εικόνα 6.3. Τέλος, η σκηνή περιέχει ένα κομμάτι μελωδίας μέσω του αντικειμένου SoundObject, κάνοντας το περιβάλλον πιο φιλόξενο προς τον χρήστη.

```

6  public class IntroGame2Script : MonoBehaviour
7  {
8      public GameObject ClickSound;
9
10     public void DelayIntroGame2Button() {
11         ClickSound.SetActive(true);
12         Invoke("IntroGame2Button", 2f);
13     }
14
15     void IntroGame2Button() {
16         SceneManager.LoadScene("Game2");
17     }
18
19     //BackButton
20     public void DelayBackButton(){
21         ClickSound.SetActive(true);
22         Invoke("BackButton", 2f);
23     }
24
25     void BackButton() {
26         SceneManager.LoadScene("MenuScene");
27     }
28 }
29
30

```

Εικόνα 6.2 Κώδικας κουμπιών μαζί με τον ήχο

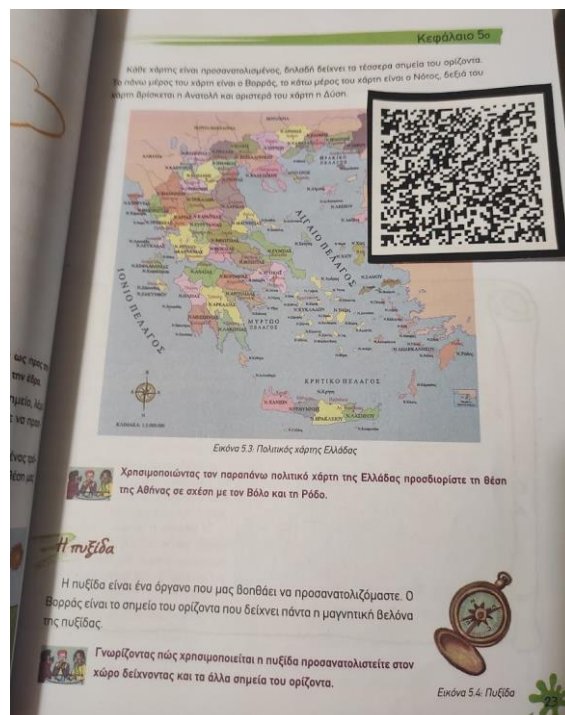
```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5

```

Εικόνα 6.3 Βιβλιοθήκες που χρησιμοποιήθηκαν στο IntroGame2Script.cs

Από την στιγμή που θα πατήσει ο χρήστης το κουμπί «Επόμενο», θα μεταφερθεί στην επόμενη σκηνή, που είναι το βασικό παιχνίδι. Σαν πρώτο βήμα, θα πρέπει ο χρήστης να πάει στην σελίδα 23 του σχολικού βιβλίου και να σκανάρει με την κάμερα του κινητού το QR code που υπάρχει επάνω στο βιβλίο (Εικόνα 6.4). Σκανάροντας με επιτυχία το QR code, ανοίγει μπροστά στο χρήστη το επανηχημένο παιχνίδι. Μετά το πέρας μερικών δευτερολέπτων, εμφανίζεται αυτόματα πάνω στην οθόνη σε δισδιάστατο κείμενο η πρώτη πρόταση που θα πρέπει να λύσει ο χρήστης. Με βάση το επανηχημένο περιβάλλον που εμφανίζεται μπροστά στον χρήστη, θα πρέπει να γίνει η σωστή συμπλήρωση των κενών στην κάθε πρόταση. Η συμπλήρωση γίνεται επιλέγοντας το κουμπί που θεωρεί ο χρήστης πως είναι το σωστό (Εικόνα 6.5). Οι συνολικές προτάσεις που θα πρέπει να συμπληρώσει ο χρήστης μέχρι το τέλος του παιχνιδιού είναι δώδεκα. Αμέσως μετά, αυτόματα, οδηγείται στην σκηνή των αποτελεσμάτων.



Εικόνα 6.4 QR code του παιχνιδιού πάνω στο βιβλίο



Εικόνα 6.5 Εμφάνιση παιχνιδιού «Προσανατολισμός»

6.2.1 Υλοποίηση σκηνής Game2

Η σκηνή αυτή, αποτελείται από δύο scripts, με βασικό script να είναι το Questions.cs, διότι περιλαμβάνει όλες τις λειτουργίες του παιχνιδιού. Το δεύτερο script, ονομάζεται Qlist.cs και περιέχει μόνο μια μεταβλητή τύπου String την Question (Εικόνα 6.6).

```
C: > GeographyAR > GeographyAR > Assets > scripts > game2 > Qlist.cs
1
2 [System.Serializable]
3 public class Qlist
4 {
5     public string Question;
6 }
```

Εικόνα 6.6 Κώδικας script Qlist.cs

Στον βασικό κώδικα, αρχικά γίνεται η αρχικοποίηση όλων των μεταβλητών και πιο συγκεκριμένα αρχικοποιείται η λίστα questions τύπου Qlist η οποία περιλαμβάνει όλες τις προτάσεις που θα εμφανιστούν στον χρήστη. Το μέγεθος της λίστας προσδιορίζεται χειροκίνητα μέσα από το Unity και οι προτάσεις εμφανίζονται πάντα σε τυχαία σειρά καθώς επίσης δεν γίνεται επανάληψη των ίδιων προτάσεων. Επιπλέον, γίνεται η αρχικοποίηση των κειμένων, τύπου Text, που έχουν τα κουμπιά και δύο μετρητών τύπου Integer που αποθηκεύουν το σύνολο των σωστών και των λανθασμένων απαντήσεων με βάση τις επιλογές του χρήστη (Εικόνα 6.7).

```
14 public class Questions : MonoBehaviour
15 {
16
17     private TextMeshProUGUI QuestionText;
18     public GameObject erotiseis;
19     public int correctAnswer,falseAnswer;
20
21     public Qlist[] questions;
22     private static List<Qlist> unansweredQuestions;
23     private Qlist currentQuestion;
24
25     [SerializeField]
26     private float timeBetweenQuestions = 2f;
27
28     public Text VorიაText; // -----Boras
29     public GameObject Vorია;
30
31     public GameObject Notia; // -----Notos
32     public Text NotiaText;
33
34     public GameObject Anatolika; // -----Anatolika
35     public Text AnatolikaText;
36
37     public GameObject Ditika; // -----Ditika
38     public Text DitikaText;
39 }
```

Εικόνα 6.7 Αρχικοποίηση μεταβλητών στο Questions.cs

Στην συνέχεια, με την ενεργοποίηση του script ξεκινάει η μέθοδος Start(), η οποία παίρνει τις τιμές των κειμένων που έχουν τα κουμπιά όπως και το κείμενο των προτάσεων. Κατόπιν, καλεί την μέθοδο showText() τύπου void, όπου εμφανίζει ένα κείμενο στον χρήστη, και μετά καλεί την μέθοδο SetRandomQuestion(), επίσης τύπου void (Εικόνα 6.8). Σε αυτή την μέθοδο, γίνεται η επιλογή μιας τυχαίας πρότασης μέσα από την λίστα που έχει δημιουργηθεί όπως προαναφέρθηκε. Η κάθε πρόταση μπορεί να επιλεγεί μόνο μια φορά μέχρι το τέλος του παιχνιδιού (Εικόνα 6.9). Στο τέλος της μεθόδου καλείται μια άλλη μέθοδος η Default(). Μέσα σε αυτή την μέθοδο γίνεται η αρχικοποίηση των κειμένων στα κουμπιά.(Εικόνα 6.10).

```

40 void Start()
41 {
42     VoriaText = Voria.GetComponent<Text>();
43     NotiaText = Notia.GetComponent<Text>();
44     AnatolikaText = Anatolika.GetComponent<Text>();
45     DitikaText = Ditika.GetComponent<Text>();
46
47     QuestionText = erotiseis.GetComponent<TextMeshProUGUI>();
48     Invoke("showText", 4.0f);
49
50     if (unansweredQuestions == null || unansweredQuestions.Count == 0){
51         unansweredQuestions = questions.ToList<Qlist>();
52     }
53     Invoke("SetRandomQuestion", 8.0f);
54 }
55

```

Εικόνα 6.8 Μέθοδος Start()

```

91 void SetRandomQuestion() {
92     int randomQuestionIndex = Random.Range (0, unansweredQuestions.Count);
93     currentQuestion = unansweredQuestions[randomQuestionIndex];
94     QuestionText.text = currentQuestion.Question;
95     Default();
96 }

```

Εικόνα 6.9 Μέθοδος SetRandomQuestion()

```

185 void Default() {
186
187     GameObject.Find("Voria").GetComponentInChildren<Text>().text = "Βόρειο";
188     GameObject.Find("Notia").GetComponentInChildren<Text>().text = "Νότιο";
189     GameObject.Find("Anatolika").GetComponentInChildren<Text>().text = "Ανατολικό";
190     GameObject.Find("Ditika").GetComponentInChildren<Text>().text = "Δυτικό";
191 }
192

```

Εικόνα 6.10 Μέθοδος Default()

Καθ' όλη την διάρκεια του script, η μέθοδος Update() χρησιμοποιεί μια συνθήκη If όπου ελέγχει την πρόταση που εμφανίζεται την δεδομένη στιγμή στον χρήστη και με βάση την πρόταση αλλάζει τα

κείμενα στα κουμπιά για να δώσει άλλες επιλογές ώστε να μπορέσει ο χρήστης να απαντήσει σωστά στην κάθε πρόταση ξεχωριστά (Εικόνα 6.11).

```

57 void Update() {
58     if(QuestionText.text == "Ο κάδος απορριμάτων βρίσκεται ____ του Πικνικ") {
59
60         GameObject.Find("Voria").GetComponentInChildren<Text>().text = "Βορειοδυτικά";
61         GameObject.Find("Notia").GetComponentInChildren<Text>().text = "Νοτιοδυτικά";
62
63     } else if (QuestionText.text == "Το αμάξι είναι ____ από τον μαθητή") {
64
65         GameObject.Find("Notia").GetComponentInChildren<Text>().text = "ΝοτιοΑνατολικά";
66
67     } else if (QuestionText.text == "Η τοποθεσία της πυξίδα είναι ____ ως προς τον κάδο") {
68
69         GameObject.Find("Anatolika").GetComponentInChildren<Text>().text = "Βορειοδυτικά";
70
71     } else if (QuestionText.text == "Το αμάξι βρίσκεται ____ της πυξίδα") {
72
73         GameObject.Find("Notia").GetComponentInChildren<Text>().text = "Νοτιοδυτικά";
74
75     } else if (QuestionText.text == "Η πυξίδα βρίσκεται ____ της μπάλας") {
76
77         GameObject.Find("Anatolika").GetComponentInChildren<Text>().text = "Βορειοανατολικά";
78     } else if (QuestionText.text == "Η πυξίδα βρίσκεται ____ του κάδου") {
79
80         GameObject.Find("Anatolika").GetComponentInChildren<Text>().text = "Βορειοδυτικά";
81     } else if (QuestionText.text == "Η μπάλα είναι ____ του κάδου") {
82
83         GameObject.Find("Ditika").GetComponentInChildren<Text>().text = "Βορειοδυτικά";
84     } else if (QuestionText.text == "Το παγκάκι βρίσκεται ____ της μπάλας") {
85
86         GameObject.Find("Notia").GetComponentInChildren<Text>().text = "ΝοτιοΑνατολικά";
87     }
88 }
89

```

Εικόνα 6.11 Μέθοδος Update() στο script Questions.cs

Στο συγκεκριμένο script έχουν δημιουργηθεί ακόμα τέσσερις διαφορετικοί μέθοδοι τύπου void, κάθε μια για ένα ξεχωριστό κουμπί επιλογής. Οι μέθοδοι αυτοί είναι οι OnClickVoria(), OnClickNotia(), OnClickAnatolika(), και η OnClickDitika(). Η ανάλυση των λειτουργιών θα γίνει σε μία από αυτές τις μεθόδους καθώς και οι τέσσερις έχουν τις ίδιες λειτουργίες και στο μόνο που διαφέρουν είναι στις ονομασίες και στις σωστές απαντήσεις (Εικόνα 6.12). Η μέθοδος αποτελείται από πολλαπλές συνθήκες If, οι οποίες ελέγχουν το κείμενο της πρότασης, και με βάση το κουμπί που πατήθηκε ελέγχεται η σωστή απάντηση. Σε περίπτωση σωστής απάντησης ο μετρητής των σωστών απαντήσεων αυξάνεται κατά μια μονάδα και αμέσως μετά καλείται η μέθοδος TransitionToNextQuestion(), η οποία λειτουργεί ακριβώς όπως στο script QuizManager.cs στο παιχνίδι «Τα είδη των χαρτών» που αναφέρθηκε στο προηγούμενο κεφάλαιο. Στην περίπτωση που η απάντηση που δίνεται από τον χρήστη δεν είναι σωστή αυξάνεται ο μετρητής των λανθασμένων απαντήσεων κατά μία μονάδα και στην συνέχεια καλείται η ίδια μέθοδος για την εμφάνιση της επόμενης πρότασης.

```

144 public void onClickAnatolika() {
145     if (QuestionText.text == "Το παγκάκι βρίσκεται ____ του μαθητή") {
146         correctAnswer++ ;
147         StartCoroutine(TransitionToNextQuestion());
148     }else if (QuestionText.text == "Το παγκάκι βρίσκεται ____ του αυτοκινήτου") {
149         correctAnswer++ ;
150         StartCoroutine(TransitionToNextQuestion());
151     }else if (QuestionText.text == "Ο κάδος απορριμάτων βρίσκεται ____ του Πικνικ") {
152         correctAnswer++ ;
153         StartCoroutine(TransitionToNextQuestion());
154     }else {
155         falseAnswer++;
156         StartCoroutine(TransitionToNextQuestion());
157     }
158 }
159

```

Εικόνα 6.12 Μέθοδος OnClickAnatolika()

Οι παραπάνω διαδικασίες ολοκληρώνονται με το που απαντηθούν και οι 12 προτάσεις. Μετά το πέρας όλων των προτάσεων καλείται αυτόματα η μέθοδος EndOfGame() η οποία με εντολή SetInt κρατάει τις τιμές και των δύο μετρητών και στέλνει τον χρήστη στην τελευταία σκηνή του παιχνιδιού όπου γίνεται και η εμφάνιση των αποτελεσμάτων (Εικόνα 6.13).



Εικόνα 6.13 Σκηνή αποτελεσμάτων

Η σκηνή αυτή, αποτελείται από μερικά κείμενα και ένα κουμπί όπου επιστρέφει τον χρήστη στην αρχική σελίδα της εφαρμογής με ένα πάτημα. Οι τιμές από τις σωστές και τις λανθασμένες απαντήσεις καλούνται από το script Results.cs. Ο συγκεκριμένος κώδικας αποτελείται από την μέθοδο Start() όπου με την έναρξη της σκηνής παίρνονται με την βοήθεια της εντολής GetInt οι τιμές των σωστών και των λανθασμένων απαντήσεων. Η εμφάνιση των τιμών γίνεται σε δύο κείμενα τύπου Text μέσα στην σκηνή (Εικόνα 6.14).

```

22      public void Start()
23      {
24          int good,wrong;
25          good = PlayerPrefs.GetInt("sostesApantisis");
26          wrong = PlayerPrefs.GetInt("lathosApantisis");
27
28          String myString = good.ToString();
29          String myString2 = wrong.ToString();
30
31          sostes.text = myString;
32          lathos.text = myString2;
33      }
34

```

Εικόνα 6.14 Μέθοδος Start() στο script Results.cs

6.3 Επίλογος

Το παιχνίδι «Προσανατολισμός» αποτελεί το 5^ο Κεφάλαιο στην σελίδα 23 του σχολικού βιβλίου στο μάθημα της Γεωγραφίας της Ε' Δημοτικού. Το συγκεκριμένο αποτελεί ένα παιχνίδι επαυξημένης πραγματικότητας και σκοπός του είναι να κατανοήσει ο χρήστης την έννοια του προσανατολισμού μέσα από τις προτάσεις που με τυχαία σειρά εμφανίζονται στην οθόνη του χρήστη.

Κεφάλαιο 7ο: Οι ακτές της Ελλάδας

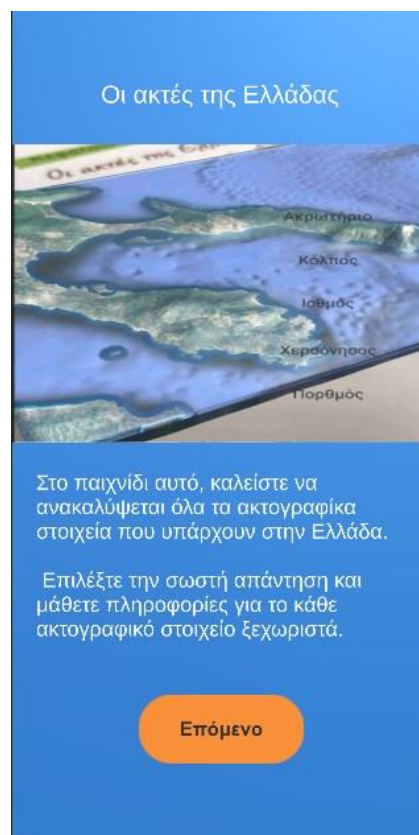
7.1 Εισαγωγή

Το κεφάλαιο αυτό, μιλάει για το παιχνίδι στο 8^ο κεφάλαιο με τίτλο «Οι ακτές της Ελλάδας». Στόχος του παιχνιδιού είναι να κάνει, πιο οικεία προς τον χρήστη, τα καταγραφικά στοιχεία που υπάρχουν. Επιπλέον, αναλύονται όλα τα βήματα που χρειάστηκαν να γίνουν για την δημιουργία αυτού του παιχνιδιού.

7.2 Η δομή του παιχνιδιού

Το παιχνίδι περιλαμβάνει την ύλη του 8^{ου} Κεφαλαίου στις σελίδες 32-34 του σχολικού βιβλίου στο μάθημα της Γεωγραφίας της Ε' δημοτικού. Με την βοήθεια της επαυξημένης πραγματικότητας που χρησιμοποιεί το παιχνίδι, ο χρήστης θα είναι σε θέση να κατανοήσει καλύτερα τα ακτογραφικά στοιχεία της Ελλάδας. Τα ακτογραφικά στοιχεία εμφανίζονται σαν επαυξημένα αντικείμενα στον χώρο και με βάση αυτά ο χρήστης θα πρέπει να επιλέξει την σωστή ονομασία μέσα από μια σειρά επιλογών.

Με το άνοιγμα του παιχνιδιού ο χρήστης μπαίνει στην πρώτη δισδιάστατη σκηνή, η οποία παρουσιάζει τις οδηγίες του παιχνιδιού. Η σκηνή αυτή αποτελείται από διάφορα κείμενα τύπου Text ένα κουμπί για να την μετακίνηση του χρήστη στην επόμενη σκηνή, και μια μελωδία που παίζει στο πίσω μέρος της σκηνής (Εικόνα 7.1).



Εικόνα 7.1 Εισαγωγική σκηνή παιχνιδιού

Η μελωδία που παίζει στο πίσω μέρος της σκηνής γίνεται με την βοήθεια ενός άδειου αντικειμένου (Empty Object). Αυτό το αντικείμενο περιέχει σαν παράμετρο ένα Audio Source που περιλαμβάνει την μελωδία της σκηνής αυτής. Όσο η σκηνή είναι στο προσκήνιο το αντικείμενο είναι ενεργοποιημένο και αναπαράγει καθ' όλη την διάρκεια την μελωδία. Το κουμπί που είναι τοποθετημένο στο κάτω μέρος της σκηνής καθοδηγεί τον χρήστη με την βοήθεια του κειμένου που περιέχει, να προχωρήσει στην επόμενη σκηνή. Αυτό επιτυγχάνεται με το script IntroButton.cs. Ο κώδικας περιέχει μια μέθοδο τύπου void την DelayOnClickButton(), η οποία με το πάτημα του κουμπιού καλείται και στην συνέχεια ενεργοποιεί τον ήχο που αναπαράγει το κουμπί. Στην συνέχεια η εφαρμογή, με την χρήση της εντολής SceneManager κατευθύνει τον χρήστη στην σκηνή του παιχνιδιού (Εικόνα 7.2).

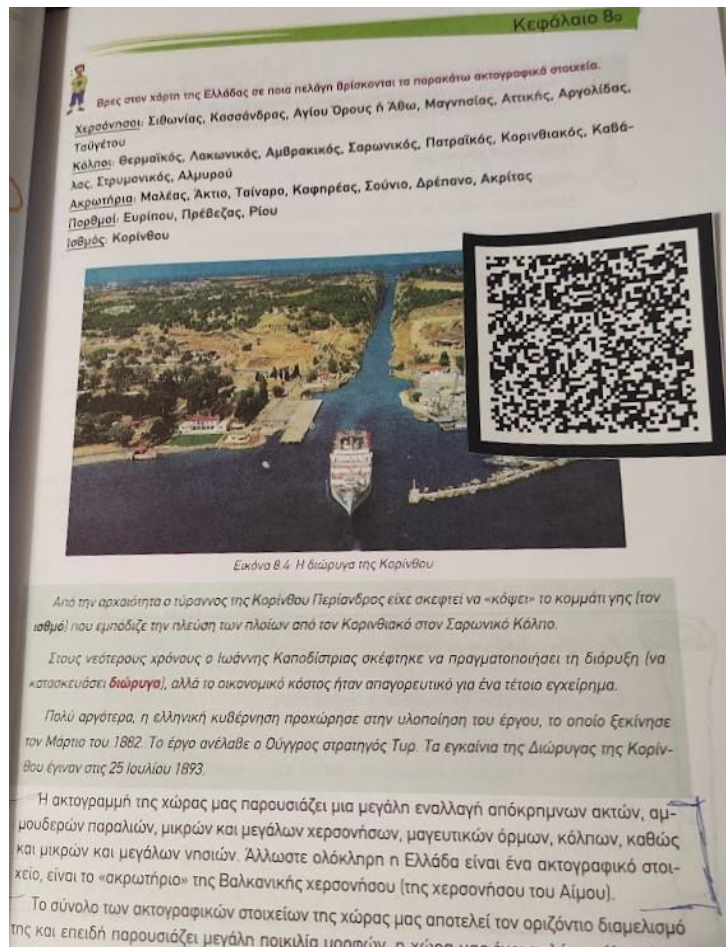
```
public class IntroButton : MonoBehaviour
{
    public GameObject clickSound;

    public void DelayOnClickButton() {
        clickSound.SetActive(true);
        Invoke("OnClickButton", 2f);
    }

    void OnClickButton() {
        SceneManager.LoadScene("ThesiGame");
    }
}
```

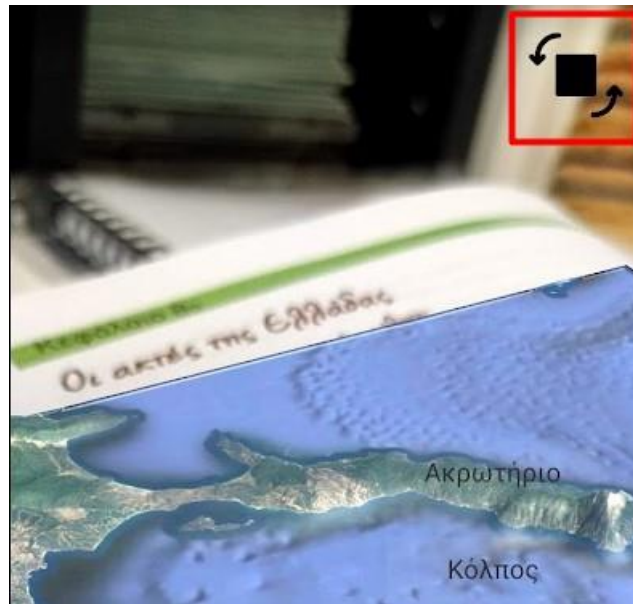
Εικόνα 7.2 Κώδικας script IntroButton.cs

Όταν ο χρήστης συνδεθεί στην σκηνή καλείται να σκανάρει, με την βοήθεια της κάμερας της συσκευής, το QR code του συγκεκριμένου παιχνιδιού, το οποίο υπάρχει στην σελίδα 33 του βιβλίου στο 8^ο Κεφάλαιο (Εικόνα 7.3).



Εικόνα 7.3 QR code από το παιχνίδι «Οι ακτές της Ελλάδας»

Με το που σκανάρει ο χρήστης το QR code, εμφανίζονται επαυξημένα αντικείμενα που αντικατοπτρίζουν διάφορες τοποθεσίες της Ελλάδας. Όπως είχε γίνει αναφορά και στο 3^ο Κεφάλαιο, τα αντικείμενα έχουν δημιουργηθεί μέσω ενός εργαλείου του Blender το οποίο παρέχει και την δυνατότητα προσθήκης ανάλογου ύψους στα αντικείμενα που δημιουργούνται. Με αυτό τον τρόπο, ο χρήστης σε αυτό το παιχνίδι έχει την δυνατότητα να παρατηρήσει και να μελετήσει τα διάφορα αντικείμενα μέσα στον χώρο. Παράλληλα στον χρήστη παρέχεται η δυνατότητα με την χρήση ενός κουμπιού να περιστρέψει το αντικείμενο έως 360 μοίρες, ώστε να είναι η μελέτη του αντικειμένου πιο εύκολη (Εικόνα 7.4).



Εικόνα 7.4 Κουμπί περιστροφής αντικειμένων

Στην συνέχεια, με βάση το κάθε αντικείμενο ο χρήστης καλείται να επιλέξει μία σωστή απάντηση από τις πέντε επιλογές που του δίνονται. Σε περίπτωση λανθασμένης απάντησης εμφανίζεται και το αντίστοιχο μήνυμα λάθους σε ένα κείμενο τύπου TextMeshPro. Στην περίπτωση που ο χρήστης επιλέξει την σωστή απάντηση, εμφανίζεται ένα αναδυόμενο διαδιάστατο Panel που παρέχει μερικές πληροφορίες για το κάθε ακτογραφικό στοιχείο. Στο πέρας όλων των απαντήσεων εμφανίζεται ένα επιπλέον Panel, το οποίο σηματοδοτεί και το τέλος του παιχνιδιού, με το συμπέρασμα του παιχνιδιού, καθώς και με ένα κουμπί που επιστρέφει τον χρήστη στην αρχική σελίδα της εφαρμογής.

Για την υλοποίηση αυτής της σκηνής χρησιμοποιούνται δύο script, ένα για την υλοποίηση του κουμπιού περιστροφής των αντικειμένων, με το όνομα ObjectRotation.cs και το δεύτερο script παρέχει όλες τις υπόλοιπες λειτουργίες της σκηνής, με το όνομα SelectObject.cs. Αρχικά, για το κουμπί περιστροφής αρχικοποιήθηκαν τρεις μεταβλητές. Η objectRotate τύπου GameObject περιέχει το αντικείμενο που περιστρέφει το κουμπί, η rotatespeed τύπου float περιέχει την τιμή της ταχύτητας περιστροφής, και τέλος η rotateStatus τύπου Boolean ελέγχει το πάτημα του κουμπιού (Εικόνα 7.5).

```

9  public class ObjectRotation : MonoBehaviour
10 {
11     public GameObject objectRotate;
12     public float rotatespeed = 50f;
13     bool rotateStatus = false;
14 }

```

Εικόνα 7.5 Αρχικοποίηση μεταβλητών του script ObjectRotation.cs

Έπειτα, με το πάτημα του κουμπιού καλείται η μέθοδος RotateObject() η οποία με την σειρά της ελέγχει το πάτημα του κουμπιού. Σε περίπτωση που πατηθεί το κουμπί η μεταβλητή rotateStatus αλλάζει και παίρνει την τιμή true. Όσο το script είναι ενεργοποιημένο η μέθοδος Update() τρέχει

συνέχεια, ελέγχοντας την μεταβλητή rotateStatus. με αποτέλεσμα όταν η τιμή γίνεται true να μπαίνει μέσα στην συνθήκη της If και να περιστρέφει το αντικείμενο με την εντολή όπως φαίνεται στην Εικόνα 7.6.

```

15     public void RotateObject() {
16         if (rotateStatus == false) {
17
18             rotateStatus = true;
19
20         }else {
21
22             rotateStatus = false;
23         }
24     }
25
26
27     void Update()
28     {
29         if (rotateStatus == true) {
30             objectRotate.transform.Rotate(Vector3.forward, rotatespeed * Time.deltaTime);
31         }
32     }

```

Εικόνα 7.6 Μέθοδος Update του script ObjectRotation.cs

Το κύριο script όπως προαναφέρθηκε, αποτελεί το βασικό κομμάτι καθώς περιλαμβάνει όλες τις υπόλοιπες λειτουργίες αυτής της σκηνής. Στην αρχή μέσα στο script γίνεται η αρχικοποίηση όλων των μεταβλητών που χρησιμοποιούνται. Οι μεταβλητές τύπου GameObject περιέχουν τα επαυξημένα αντικείμενα που εμφανίζονται ανά διαστήματα στο παιχνίδι καθώς και τα κουμπιά που αντιστοιχούν στο καθένα. Κατόπιν, γίνεται η αρχικοποίηση μερικών κειμένων τύπου Text, που εμφανίζονται μπροστά στον χρήστη, ενώ στο τέλος γίνεται η μεταβλητή τύπου GameObject που περιέχει το τελικό Panel της εφαρμογής. Τα σχόλια που υπάρχουν δίπλα στον κώδικα χρησιμεύουν στο να γίνεται κατανοητή η χρήση του κάθε αντικειμένου (Εικόνα 7.7).

```

14 public class SelectObject : MonoBehaviour
15 {
16     private TextMeshProUGUI Keimeno; //Epikefalida
17     public GameObject Text; //Epikefalida
18
19     public GameObject PanelText; //to text pou emfanizetai sto panel
20     private TextMeshProUGUI PanelTextView;
21
22     public GameObject Xersonisos; // Object tis xersonisou
23     public GameObject XersonisosButton; //Button Xersonisos
24
25     public GameObject Kolpos; //Object tou kolpou
26     public GameObject KolposButton; //Button Kolpos
27
28     public GameObject Porthmos; //Object tou Porthmos
29     public GameObject PorthmosButton; //Button Porthmos
30
31     public GameObject Isthmos; // object toy isthmou
32     public GameObject IsthmosButton; // Button isthmos
33
34     public GameObject Akrotirio; // object toy Akrotirio
35     public GameObject AkrotirioButton; // Button Akrotirio
36
37     public GameObject PliroforiesPanel; //to panel
38     public GameObject ClosePanelButton; // Button to close Panel
39
40     private String type = " ";
41     public GameObject FinalPanel;
42

```

Εικόνα 7.7 Αρχικοποίηση μεταβλητών στο script SelectObject.cs

Με την έναρξη της εφαρμογής γίνεται και η εκτέλεση της πρώτης μεθόδου Start(). Η μέθοδος αυτή κρατάει το κείμενο που εμφανίζεται μπροστά στον χρήστη και μετά από λίγα δευτερόλεπτα καλεί μια καινούργια μέθοδο την DisableText() για να απενεργοποιήσει το κείμενο ώστε να μην εμφανίζεται στον χρήστη (Εικόνα 7.8).

```

43 void Start()
44 {
45     PanelTextView = PanelText.GetComponent<TextMeshProUGUI>();
46     Keimeno = Text.GetComponent<TextMeshProUGUI>();
47     Invoke("DisableText", 4f);
48 }
49

```

Εικόνα 7.8 Μέθοδος Start() του script SelectObject.cs

Σε κάθε γύρο παιχνιδιού εμφανίζεται με την σειρά επανξημένα αντικείμενα, όπου ο χρήστης θα πρέπει να αναγνωρίσει το ακτογραφικό στοιχείο και με βάση αυτό να επιλέξει το σωστό κουμπί. Για την λειτουργικότητα των κουμπιών έχουν δημιουργηθεί πέντε διαφορετικές μέθοδοι για το κάθε κουμπί ξεχωριστά με τα ονόματα ClickObject(), ClickKolpos(), ClickIsthmos(), ClickPorthmos() και ClickAkrotirio(). Οι λειτουργίες στην κάθε μια από αυτές τις μεθόδους είναι παρόμοιες και διαφέρουν

μόνο στα επαυξημένα αντικείμενα που είναι ενεργά, και στην σωστή απάντηση κάθε κουμπιού. Για τους παραπάνω λόγους θα γίνει η ανάλυση μιας μεθόδου, και με αντίστοιχο τρόπο έχουν υλοποιηθεί και οι υπόλοιπες τέσσερις. Η μέθοδος ClickIsthmos() τύπου void χρησιμοποιείται για το πάτημα του κουμπιού «Ισθμός» και αποτελείται από μια συνθήκη If όπου ελέγχει αν το επαυξημένο αντικείμενο με το όνομα Isthmos είναι ενεργοποιημένο. Αυτός ο έλεγχος επιτυγχάνεται με την εντολή activeSelf. Στην συνέχεια εμφανίζεται ένα κείμενο όπου ενημερώνει τον χρήστη για την σωστή επιλογή του κουμπιού και έπειτα σε περίπτωση σωστής απάντησης ενεργοποιεί ένα Panel που περιέχει ένα κείμενο τύπου Text με πληροφορίες σχετικά με το επαυξημένο αντικείμενο που παρατηρεί ο χρήστης. Σε περίπτωση λανθασμένης απάντησης εμφανίζεται το αντίστοιχο κείμενο λάθους τύπου Text στον χρήστη (Εικόνα 7.9).

```

97  public void ClickIsthmos() {
98
99  if (Isthmos.activeSelf) {
100
101      type = "Isthmos";
102      Keimeno.enabled = true;
103      Keimeno.text = "Σωστό!!";
104      Invoke("DisableText",3f);
105
106      IsthmosButton.SetActive(false);
107      PliroforiesPanel.SetActive(true);
108      PanelTextView.text = "Ισθμός είναι σχετικά στενή λωρίδα εδάφους που χωρίζει δύο ακτές και συνδέει δύο μεγαλύτερες περιοχές ξηράς.";
109  } else {
110
111      Keimeno.enabled = true;
112      Keimeno.text = "Λάθος..!!";
113      Invoke("DisableText",3f);
114  }
115  }

```

Εικόνα 7.9 Μέθοδος ClickIsthmos()

Έπειτα, αφού ο χρήστης διαβάσει τις πληροφορίες που του δίνονται μέσω του Panel, μπορεί να επιλέξει το κουμπί «Τέλειο!» για να προχωρήσει στο παιχνίδι. Το κουμπί αυτό ενεργοποιεί την μέθοδο OnClickPanelButton() τύπου void η οποία κλείνει το Panel, και με την σειρά της ενεργοποιεί ξανά τα κουμπιά των απαντήσεων (Εικόνα 7.10). Επιπλέον, με το πέρας μερικών δευτερολέπτων καλεί μια καινούργια μέθοδο την NextObject(). Η συγκεκριμένη μέθοδος με μια συνθήκη If ελέγχει το επαυξημένο αντικείμενο που είναι να απενεργοποιηθεί και στην συνέχεια ενεργοποιεί το επόμενο αντικείμενο που έχει σειρά (Εικόνα 7.11).

```

159     public void OnClickPanelButton() {
160
161         XersonisosButton.SetActive(true);
162         KolposButton.SetActive(true);
163         IsthmosButton.SetActive(true);
164         PorthmosButton.SetActive(true);
165         AkrotirioButton.SetActive(true);
166
167         PliroforiesPanel.SetActive(false);
168         Invoke("NextObject", 2f);
169     }

```

Εικόνα 7.10 Μέθοδος OnClickPanelButton()

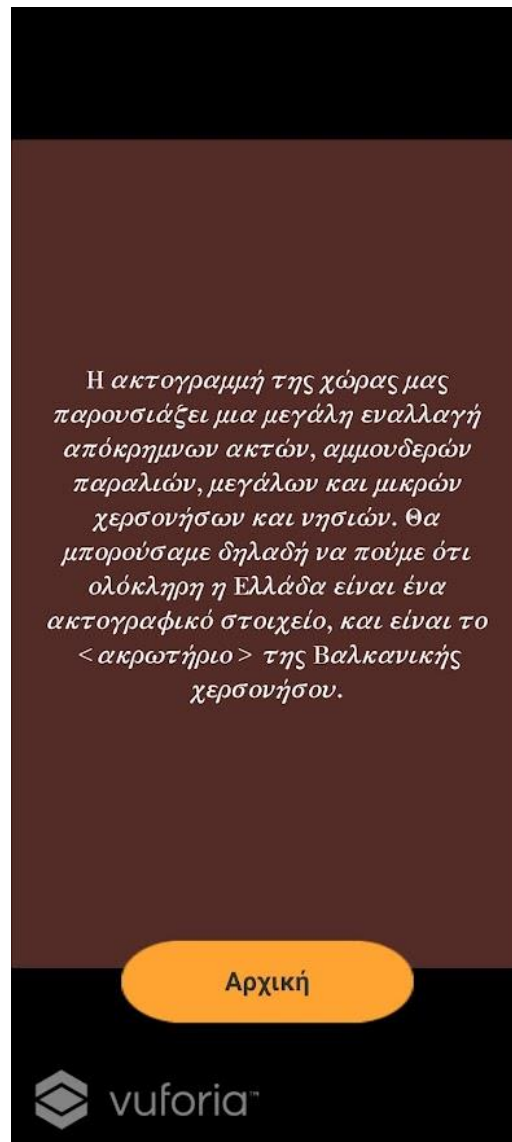
```

171     void NextObject() {
172
173         if (type == "Xersonisos") {
174
175             Xersonisos.SetActive(false);
176             Kolpos.SetActive(true);
177             Start();
178
179         } else if (type == "Kolpos") {
180
181             Kolpos.SetActive(false);
182             Isthmos.SetActive(true);
183
184         } else if (type == "Isthmos") {
185
186             Isthmos.SetActive(false);
187             Porthmos.SetActive(true);
188
189         } else if (type == "Porthmos") {
190
191             Porthmos.SetActive(false);
192             Akrotirio.SetActive(true);
193
194         } else if (type == "Akrotirio") {
195
196             Akrotirio.SetActive(false);
197             EndGame();
198
199         }

```

Εικόνα 7.11 Μέθοδος NextObject()

Με το που γίνει η εμφάνιση όλων των επαυξημένων αντικειμένων και η επιλογή των σωστών απαντήσεων καλείται η μέθοδος EndGame() η οποία αποτελεί και το τελικό στάδιο αυτού του παιχνιδιού. Σαν πρώτο βήμα ενεργοποιεί το τελικό Panel οπου εμφανίζει ένα κείμενο τύπου Text και παράλληλα απενεργοποιεί όλα τα κουμπιά αφήνοντας μόνο το κουμπί «Αρχική» για την επιστροφή του χρήστη στην αρχική σελίδα της εφαρμογής (Εικόνα 7.12).



Εικόνα 7.12 Τελικό Panel παιχνιδιού

7.3 Επίλογος

Το παιχνίδι «Οι ακτές της Ελλάδας» καλύπτει την ύλη του 8^{ου} Κεφαλαίου στις σελίδες 32-34 του σχολικού βιβλίου στο μάθημα της Γεωγραφίας της Ε' δημοτικού. Αποτελεί ένα παιχνίδι με χρήση επανξιμένης πραγματικότητας και σαν στόχο έχει να κατανοήσει ο χρήστης τα διάφορα ακτογραφικά στοιχεία της Ελλάδας παρατηρώντας επανξιμένα αντικείμενα που υπάρχουν μέσα στον χώρο.

Κεφάλαιο 8ο: Νησιά της Ελλάδας

8.1 Εισαγωγή

Το κεφάλαιο αυτό μιλάει για το παιχνίδι που έχει δημιουργηθεί για το 10^ο Κεφάλαιο του βιβλίου με τίτλο «Μεγάλα νησιωτικά συμπλέγματα και νησιά της Ελλάδας». Στόχος του παιχνιδιού είναι η εκμάθηση των ονομάτων των πιο γνωστών νησιών της Ελλάδας καθώς και τα ονόματα των νησιωτικών συμπλεγμάτων. Επιπλέον, γίνεται η ανάλυση όλου του κώδικα που υλοποιήθηκε για τις λειτουργίες αυτού του παιχνιδιού.

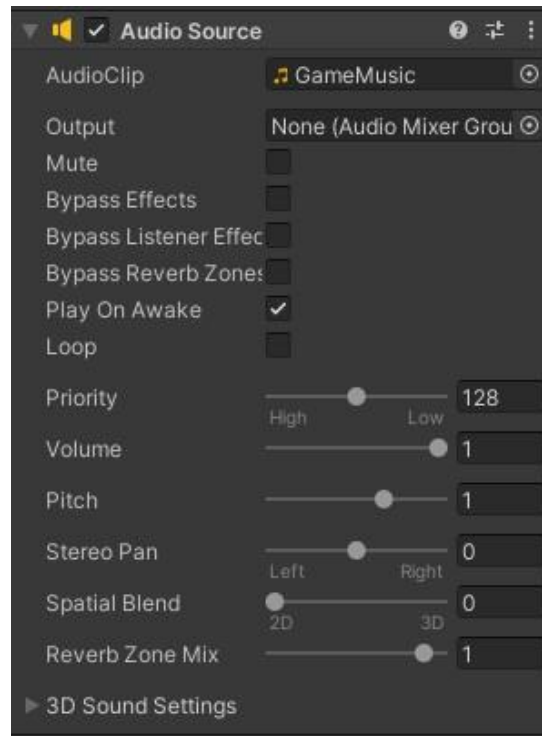
8.2 Υλοποίηση παιχνιδιού

Για να γίνει η καλύτερη ανάλυση αυτού του παιχνιδιού, θα γίνει ο διαχωρισμός του σε μερικές ενότητες. Με την έναρξη του παιχνιδιού παρουσιάζεται σε δισδιάστατη μορφή ένα Panel όπου εξηγεί στον χρήστη την διαδικασία που θα πρέπει να ακολουθήσει για τον τερματισμό του παιχνιδιού. Αυτό το Panel αποτελείται από μερικά κείμενα τύπου Text όπου σε αυτά είναι τοποθετημένες οι οδηγίες του παιχνιδιού. Στο Panel αυτό γίνεται και χρήση μιας εικόνας με τον χάρτη της Ελλάδας παρουσιάζοντας το κάθε σύμπλεγμα νησιών που θα πρέπει να βρει ο χρήστης ξεχωριστά. Με αυτόν τον τρόπο μπορεί ο χρήστης να είναι πιο ευέλικτος μέσα στο παιχνίδι, αφού ο κατατοπισμός των τοποθεσιών είναι πολύ συγκεκριμένος. Το κουμπί «Επόμενο» που υπάρχει στο κάτω μέρος της οθόνης παροτρύνει τον χρήστη να συνεχίσει στο κύριο μέρος του παιχνιδιού ξεκινώντας με την πρώτη ενότητα (Εικόνα 8.1).



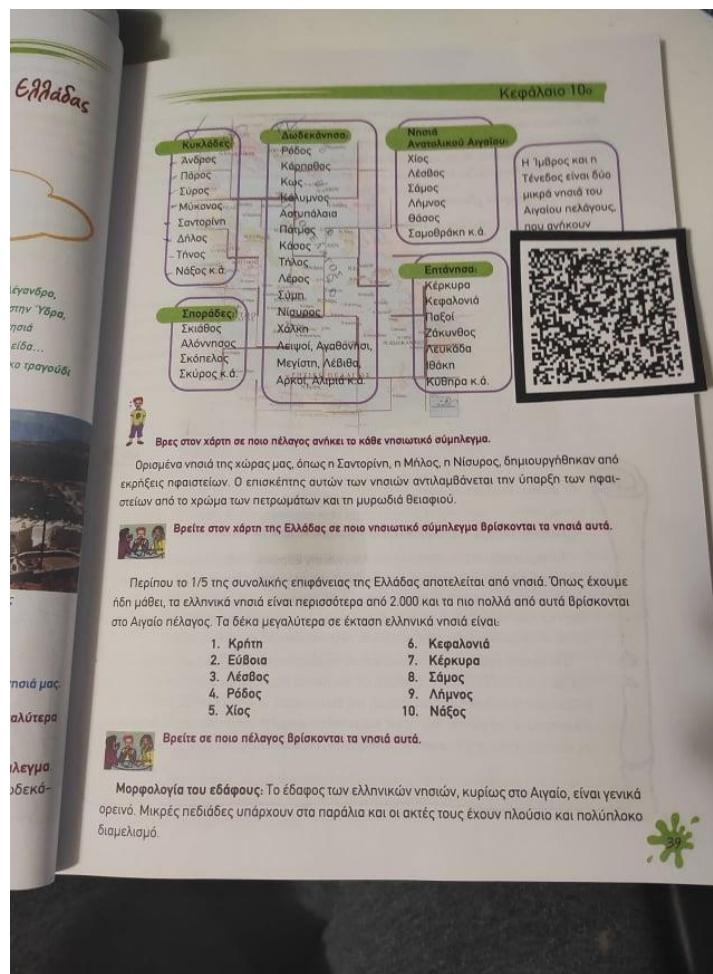
Εικόνα 8.1 Αρχικό Panel παιχνιδιού

Καθ' όλη την διάρκεια του παιχνιδιού υπάρχει και μελωδία η οποία παίζει στο πίσω μέρος της εφαρμογής, κάνοντας το περιβάλλον του παιχνιδιού πιο φιλόξενο προς τον χρήστη. Η μελωδία είναι τοποθετημένη μέσα σε ένα άδειο αντικείμενο (Empty Object) το οποίο περιέχει ένα Audio Source όπου βρίσκεται η μελωδία. Το αντικείμενο αυτό είναι ενεργοποιημένο σε όλη την διάρκεια του παιχνιδιού και αυτό έχει ως αποτέλεσμα να παίζει συνέχεια η μελωδία στο πίσω μέρος (Εικόνα 8.2).



Εικόνα 8.2 Audio Source μέσα στο SoundObject αντικείμενο

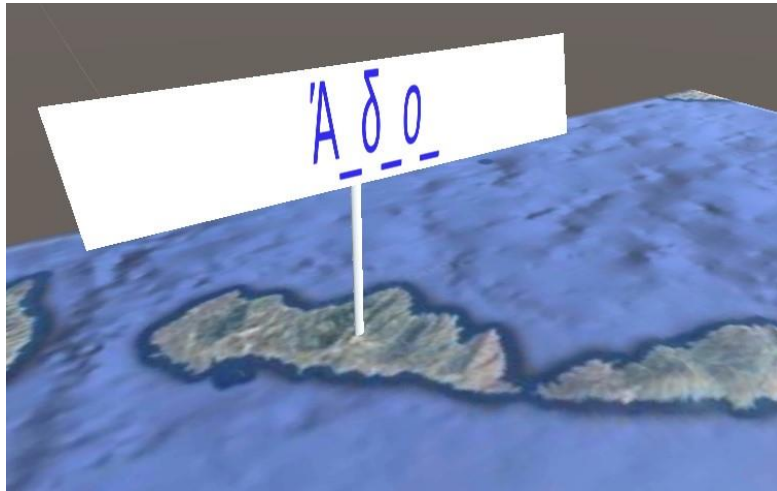
Μετά την εισαγωγή του παιχνιδιού που περιγράφηκε παραπάνω, σειρά έχει η πρώτη ενότητα του παιχνιδιού. Ο χρήστης καλείται να σκανάρει μέσω της κάμερας μιας συσκευής το QR code που υπάρχει στην σελίδα 39 στο κεφάλαιο 10 του σχολικού βιβλίου στο μάθημα της Γεωγραφίας της 5^{ης} δημοτικού (Εικόνα 8.3).



Εικόνα 8.3 Τοποθέτηση QR code πάνω στο σχολικό βιβλίο

Με το επιτυχές σκανάρισμα του QR code γίνεται η εμφάνιση των επαυξημένων αντικειμένων. Στην αρχή ο χρήστης θα πρέπει να αναγνωρίσει, μέσω των επαυξημένων αντικειμένων, τα νησιά που του εμφανίζονται, και με βάση την σειρά των πινακίδων που υπάρχουν πάνω σε κάθε νησί, θα πρέπει να γίνει η συμπλήρωση των γραμμάτων που λείπουν από τις λέξεις (Εικόνα 8.4). Η συμπλήρωση των κενών γίνεται μέσω δισδιάστατων κουμπιών που εμφανίζονται στην οθόνη του χρήστη. Κάθε φορά που γίνεται η εμφάνιση διαφορετικής πινακίδας αλλάζουν με τυχαία σειρά τα γράμματα ώστε να μπορεί να γίνει με αυτό τον τρόπο η σωστή συμπλήρωση των γραμμάτων που λείπουν (Εικόνα 8.5).

Στόχος αυτού του παιχνιδιού είναι η εκμάθηση των ονομάτων των νησιών κάθε νησιωτικού συμπλέγματος καθώς και η τοποθεσία τους. Μετά την ολοκλήρωση ενός νησιωτικού συμπλέγματος ο χρήστης έχει την δυνατότητα να εξερευνήσει μέσω επαυξημένου αντικειμένου τις τοποθεσίες πάνω στον χάρτη ώστε να εμπεδώσει καλύτερα τα ονόματα και τις τοποθεσίες των νησιών της Ελλάδας. Το παιχνίδι ολοκληρώνεται αφού ο χρήστης βρει όλα τα κενά των λέξεων από όλα τα νησιωτικά συμπλέγματα.



Εικόνα 8.4 Πινακίδα με το όνομα νησιού προς συμπλήρωση



Εικόνα 8.5 Δισδιάστατα κουμπιά για την συμπλήρωση των λέξεων

8.2.1 Υλοποίηση πρώτης ενότητας παιχνιδιού

Η πρώτη ενότητα αυτού του παιχνιδιού υλοποιείται με την βοήθεια ενός script με το όνομα `KikladesManager.cs`, το οποίο είναι γραμμένο σε γλώσσα προγραμματισμού C#. Αρχικά μέσα στο script γίνεται η αρχικοποίηση όλων των μεταβλητών που χρησιμοποιούνται. Οι αρχικές μεταβλητές τύπου `GameObject` αποτελούν τα αντικείμενα των πινακίδων που εμφανίζονται πάνω στα επαυξημένα αντικείμενα με τα ονόματα των νησιών που είναι για συμπλήρωση. Στην συνέχεια οι μεταβλητές τύπου `TMP_Text` είναι τα κείμενα που εμφανίζουν τα ονόματα των νησιών. Έπειτα οι δύο λίστες τύπου `GameObject` περιέχουν τα δισδιάστατα κουμπιά που ο χρήστης θα χρησιμοποιεί για να συμπληρώνει τα κενά των λέξεων. Τέλος, οι υπόλοιπες μεταβλητές τύπου `GameObject` περιέχουν κείμενα, σε περίπτωση που ο χρήστης συμπληρώσει λάθος γράμμα και κουμπιά που καθοδηγούν ώστε να φτάσει ο χρήστης στην ολοκλήρωση του παιχνιδιού (Εικόνα 8.6).

```

16     public GameObject Andros;
17     public GameObject Mikonos;
18     public GameObject Santorini;
19     public GameObject Naxos;
20     public GameObject Tinos;
21     public GameObject Paros;
22     public GameObject Dilos;
23     public GameObject Siros;
24
25     public TMP_Text AndrosPanel;
26     public TMP_Text MikonosPanel;
27     public TMP_Text SantoriniPanel;
28     public TMP_Text NaxosPanel;
29     public TMP_Text TinosPanel;
30     public TMP_Text ParosPanel;
31     public TMP_Text DilosPanel;
32     public TMP_Text SirosPanel;
33
34     public List<GameObject> button_list = new List<GameObject>();
35     public List<GameObject> buttons = new List<GameObject>();
36     public List<Text> buttonsName = new List<Text>();
37
38
39
40     public GameObject TextForNextQuestion;
41     public GameObject Error;
42
43     public GameObject AndrosCanvas;
44     public GameObject IslandLettersCanvas;
45     public GameObject EndKikladesCanvas;
46
47     public GameObject Kiklades;
48     public GameObject Dodekanisa;
49

```

Εικόνα 8.6 Αρχικοποίηση μεταβλητών στο script KikladesManager.cs

Μετά την αρχικοποίηση των μεταβλητών σειρά έχει η μέθοδος Start(). Αυτή η μέθοδος κρατάει τις τιμές από την λίστα των κουμπιών που περιέχουν τα γράμματα για συμπλήρωση του ονόματος του πρώτου νησιού.

```

51     void Start()
52     {
53         Button btn = button_list[0].GetComponent<Button>();
54         btn.onClick.AddListener(SelectLetterN);
55
56         Button btn1 = button_list[1].GetComponent<Button>();
57         btn1.onClick.AddListener(SelectLetterR);
58
59         Button btn2 = button_list[2].GetComponent<Button>();
60         btn2.onClick.AddListener(SelectLetterS);
61
62         Button btn3 = button_list[3].GetComponent<Button>();
63         btn3.onClick.AddListener(SelectLetter);
64

```

Εικόνα 8.7 Μέθοδος Start() από το script KikladesManager.cs

Με το που επιλέξει ο χρήστης κάποιο γράμμα για να συμπληρώσει το κατάλληλο κενό στην λέξη, καλείτε η αντίστοιχη μέθοδος όπου ενεργοποιείται με το πάτημα του κάθε κουμπιού ξεχωριστά. Η μέθοδος αυτή, με την βοήθεια της εντολής `activeSelf`, ελέγχει αν το αντικείμενο που χρειαζόμαστε είναι ενεργοποιημένο και εμφανίζεται στον χρήστη. Στην συγκεκριμένη περίπτωση, αυτό που ελέγχεται μέσα στην μέθοδο είναι αν το αντικείμενο που θέλουμε, δηλαδή η πρώτη πινακίδα με το όνομα του νησιού, εμφανίζεται στον χρήστη. Στην περίπτωση που αυτό είναι αληθές, για να γίνει ο σωστός έλεγχος της σειράς που πρέπει να τοποθετηθούν τα γράμματα, ελέγχονται τα κουμπιά. Ο έλεγχος αυτός γίνεται πάλι με την εντολή `activeSelf`. Το μοτίβο που ακολουθείτε στον έλεγχο των κουμπιών έχει να κάνει με το ποιο κουμπί είναι ενεργοποιημένο και ποιο όχι. Πιο συγκεκριμένα, όταν ο χρήστης βρει το πρώτο γράμμα από μια λέξη, το σωστό κουμπί που επιλέγεται, απενεργοποιείται. Στην συνέχεια, για το δεύτερο γράμμα γίνεται έλεγχος, αν το προηγούμενο κουμπί είναι απενεργοποιημένο και επιπλέον έχει επιλεγθεί το σωστό κουμπί για την συμπλήρωση του δεύτερου γράμματος, τότε να εμφανίζεται σωστό αποτέλεσμα. Αυτό το μοτίβο ακολουθείται για τον έλεγχο όλων των ονομάτων για τα νησιά της πρώτης ενότητας του παιχνιδιού (Εικόνα 8.8).

```

196 public void ClickButton2() {
197
198     //Mikonos
199     if (Mikonos.activeSelf) { // γ
200
201         MikonosPanel.text = "Μύκ_ο_";
202         buttons[3].SetActive(false);
203
204     } else if (Santorini.activeSelf) { //santorini
205
206         SantoriniPanel.text = "Σ_ο_ο_ο_";
207         buttons[3].SetActive(false);
208
209     } else if (Naxos.activeSelf && !buttons[2].activeSelf && !buttons[0].activeSelf) { //naxos
210
211         NaxosPanel.text = "Νά_ο_ο_";
212         TextForNextQuestion.SetActive(true);
213         Naxos.SetActive(false);
214         Invoke("ErrorText", 3f);
215         Invoke("GoToNextIslandTinos", 3f);
216
217     } else if (Tinos.activeSelf && !buttons[2].activeSelf) { //tinis
218
219         TinosPanel.text = "Τ_ο_ο_ο_";
220         buttons[3].SetActive(false);
221
222     } else if (Paros.activeSelf) { //paros
223
224         ParosPanel.text = "Π_ο_ο_ο_";
225         buttons[3].SetActive(false);
226
227     } else if (Dilos.activeSelf && !buttons[2].activeSelf) { //dilos
228
229         DilosPanel.text = "Δ_ο_ο_ο_";
230         buttons[3].SetActive(false);
231
232     } else if (Siros.activeSelf && !buttons[0].activeSelf) { //siros
233
234         SirosPanel.text = "Σ_ο_ο_ο_";
235         buttons[3].SetActive(false);
236
237     } else {
238         Error.SetActive(true);
239         Invoke("ErrorText", 2f);
240     }
241 }
242

```

Εικόνα 8.8 Μέθοδος `ClickButton2()` ελέγχει το κουμπί που επιλέχθηκε

Στην συνέχεια όταν ο χρήστης ολοκληρώσει το όνομα ενός νησιού, ανοίγει η επόμενη πινακίδα που είναι προς συμπλήρωση. Παράλληλα, γίνεται και ανακατανομή των γραμμμάτων στα κουμπιά και αντικαθιστούνται με τυχαία σειρά πάνω στα κουμπιά. Για να επιτευχτεί αυτό, μετά από κάθε ολοκλήρωση ονόματος έχει δημιουργηθεί η αντίστοιχη μέθοδος. Στο σύνολο οι μέθοδοι αυτοί είναι επτά, και η κάθε μια αλλάζει τα γράμματα ανάλογα με το όνομα του νησιού που ακολουθεί. Για την καλύτερη κατανόηση των μεθόδων αυτών θα γίνει αναφορά σε μια από αυτές τις μεθόδους, με το όνομα `GoToNextIslandNaxos()`. Η μέθοδος αυτή καλείται με το που ολοκληρωθεί η συμπλήρωση των γραμμμάτων του προηγούμενου νησιού και ακολουθεί το νησί της Νάξου. Μέσα στην μέθοδο χρησιμοποιείται η λίστα τύπου `Text` με το όνομα `buttonsName`, η οποία περιέχει τα γράμματα που τοποθετούνται πάνω στα δισδιάστατα κουμπιά. Η λίστα αυτή αλλάζει τα γράμματα με τα αντίστοιχα γράμματα για την σωστή συμπλήρωση του ονόματος, και στην συνέχεια ενεργοποιεί ξανά τα κουμπιά (Εικόνα 8.9).

```

316 void GoToNextIslandNaxos() {
317
318     Naxos.SetActive(true);
319
320     buttonsName[0].text = "Ε";
321     buttonsName[1].text = "Ν";
322     buttonsName[2].text = "Α";
323     buttonsName[3].text = "Ο";
324
325     buttons[0].SetActive(true);
326     buttons[1].SetActive(true);
327     buttons[2].SetActive(true);
328     buttons[3].SetActive(true);
329 }

```

Εικόνα 8.9 Μέθοδος `GoToNextIslandNaxos()`

Με το που συμπληρώσει ο χρήστης όλα τα ονόματα των νησιών της πρώτης ενότητας καλείται η τελευταία μέθοδος του συγκεκριμένου script, η `EndKikladesScene()`. Σε αυτή την μέθοδο ενεργοποιούνται οι πινακίδες με τα ονόματα όλων των νησιών ώστε ο χρήστης να έχει την δυνατότητα να εξερευνήσει τον επαυξημένο κόσμο ολοκληρωμένο. Παράλληλα εμφανίζεται και το κουμπί «Επόμενο» όπου κατευθύνει τον χρήστη στην επόμενη ενότητα (Εικόνα 8.10).

```

393 void EndKikladesScene() {
394
395     IslandLettersCanvas.SetActive(false);
396
397     Andros.SetActive(true);
398     Mikonos.SetActive(true);
399     Santorini.SetActive(true);
400     Naxos.SetActive(true);
401     Tinos.SetActive(true);
402     Paros.SetActive(true);
403     Dilos.SetActive(true);
404     Siros.SetActive(true);
405
406     EndKikladesCanvas.SetActive(true);
407
408 }
409

```

Εικόνα 8.10 Μέθοδος EndKikladesScene()

Η λειτουργικότητα του κουμπιού «Επόμενο» γίνεται μέσω του script ButtonManager.cs με την μέθοδο EndKikladesNextButton() (Εικόνα 8.11). Στην μέθοδο αυτή, ενεργοποιείται η εικόνα της Ελλάδας που εμφανίζει το επόμενο νησιωτικό σύμπλεγμα που θα προχωρήσει ο χρήστης και αυτό έχει ως στόχο να προσανατολίσει τον χρήστη για να γνωρίζει την τοποθεσία στην οποία πηγαίνει. Μαζί με την εικόνα αυτή ενεργοποιείται και το κουμπί «Επόμενο» όπου ανοίγει το επόμενο επανξιημένο αντικείμενο και καθοδηγεί τον χρήστη στην επόμενη ενότητα (Εικόνα 8.12).

```

72 public void EndKikladesNextButton() {
73
74     ReviewText.text = "Ας ταξιδέψουμε τώρα στα Αωδεκάνησα για να βρούμε και εκεί τα ονόματα των νησιών!";
75     Canvas.SetActive(true);
76     ImageArrow.SetActive(false);
77     ImageArrowDodekanisa.SetActive(true);
78
79     NextButton.SetActive(false);
80     StartDodekanisaBt.SetActive(true);
81
82 }

```

Εικόνα 8.11 Μέθοδος EndKikladesNextButton() στο script ButtonManager.cs



Εικόνα 8.12 Canvas επόμενης ενότητας

8.2.2 Υλοποίηση ολοκλήρωσης του παιχνιδιού

Μετά την ολοκλήρωση της πρώτης ενότητας σειρά έχει η επόμενη, η οποία έχει να κάνει με τα νησιά των Δωδεκανήσων. Το μοτίβο που ακολουθείται και σε αυτή την ενότητα είναι παρόμοιο με της προηγούμενης. Για την λειτουργικότητα αυτής της ενότητας χρησιμοποιείται το script `DodekanisaScript.cs`. Με την ενεργοποίηση του script γίνεται η αρχικοποίηση όλων των μεταβλητών που περιέχονται σε αυτό. Οι πρώτες μεταβλητές τύπου `GameObject` περιέχουν τις πινακίδες των νησιών αυτής της ενότητας, και οι μεταβλητές τύπου `TMP_Text` περιέχουν τα κείμενα με τα ονόματα των νησιών. Τέλος, μια λίστα τύπου `GameObject` περιέχει τα δισδιάστατα κουμπιά, ενώ η λίστα τύπου `Text` περιέχει τα γράμματα αυτών των δισδιάστατων κουμπιών (Εικόνα 8.13).

```

14 public class DodekanisaScript : MonoBehaviour
15 {
16
17     public GameObject Rodos;
18     public GameObject Karpathos;
19     public GameObject Patmos;
20     public GameObject Leros;
21     public GameObject Simi;
22     public GameObject Tilos;
23
24
25     public List<GameObject> buttons = new List<GameObject>();
26     public List<Text> buttonsName = new List<Text>();
27
28     public TMP_Text RodosPanel;
29     public TMP_Text KarpathosPanel;
30     public TMP_Text PatmosPanel;
31     public TMP_Text LerosPanel;
32     public TMP_Text SimiPanel;
33     public TMP_Text TilosPanel;
34
35     public GameObject IslandLetterCanvas;
36     public GameObject EndDodekanisaCanvas;
37
38     public GameObject TextForNextQuestion;
39     public GameObject Error;
40

```

Εικόνα 8.13 Αρχικοποίηση μεταβλητών στο script DodekanisaScript.cs

Με την ενεργοποίηση του script καλείται η μέθοδος Start() η οποία με την σειρά της καλεί την μέθοδο SetLetter(). Μέσα σε αυτή, ενεργοποιούνται τα δισδιάστατα κουμπιά που θα χρησιμοποιήσει ο χρήστης για να συμπληρώσει τα κενά. Και παράλληλα αρχικοποιούνται μέσω της λίστας buttonsName τα γράμματα που αντιστοιχούν στο κάθε κουμπί (Εικόνα 8.14).

```

206 void SetLetters () {
207
208     IslandLetterCanvas.SetActive(true);
209
210     buttonsName[0].text = "Α";
211     buttonsName[1].text = "Α";
212     buttonsName[2].text = "Ο";
213     buttonsName[3].text = "Ρ";
214
215 }
216

```

Εικόνα 8.14 Μέθοδος SetLetters()

Οι υπόλοιπες λειτουργίες του script είναι παρόμοιες με αυτές από την προηγούμενη ενότητα. Ο χρήστης θα πρέπει πάλι να βρει και να συμπληρώσει τα ονόματα των νησιών από τα Δωδεκάνησα και με την ολοκλήρωση των νησιών καλείται η μέθοδος EndDodekanisaScene(). Αυτή με την σειρά της εμφανίζει όλες τις πινακίδες με τα ονόματα των νησιών για να έχει την δυνατότητα και πάλι ο χρήστης να μπορεί να εξερευνήσει τον επαυξημένο κόσμο που του εμφανίζεται. Παράλληλα

ενεργοποιείται και ένα δισδιάστατο κουμπί που καθοδηγεί τον χρήστη να προχωρήσει στην επόμενη ενότητα που έχει να κάνει με το νησιωτικό σύμπλεγμα των Σποράδων (Εικόνα 8.15).

```

292 void EndDodekanisaScene() {
293
294     IslandLetterCanvas.SetActive(false);
295
296     Rodos.SetActive(true);
297     Karpathos.SetActive(true);
298     Patmos.SetActive(true);
299     Leros.SetActive(true);
300     Simi.SetActive(true);
301     Tilos.SetActive(true);
302
303     EndDodekanisaCanvas.SetActive(true);
304 }

```

Εικόνα 8.15 Μέθοδος EndDodekanisaScene() από το script DodekanisaScript.cs

Στην συνέχεια με την βοήθεια του script ButtonManager.cs και της μεθόδου StartSporades() ενεργοποιείται το καινούργιο επαυξημένο αντικείμενο και ξεκινά επίσημα η επόμενη ενότητα, η οποία περιέχει τα ονόματα από τα νησιά των Σποράδων (Εικόνα 8.16).

```

105 public void StartSporades() {
106
107     Dodekanisa.SetActive(false);
108     Sporades.SetActive(true);
109     Canvas.SetActive(false);
110     EndKikladesCanvas.SetActive(false);
111     AndrosCanvas.SetActive(false);
112     EndDodekanisaCanvas.SetActive(false);
113
114 }

```

Εικόνα 8.16 Μέθοδος StartSporades()

Η συγκεκριμένη ενότητα αποτελείται από το script SporadesScript.cs, το οποίο με την σειρά του αρχικά κάνει αρχικοποίηση όλων των μεταβλητών που χρησιμοποιεί και στην συνέχεια καλεί την μέθοδο SetLettersForSkiathos() όπου αρχικοποιεί τα δισδιάστατα κουμπιά με τα γράμματα που είναι προς συμπλήρωση για το πρώτο νησί (Εικόνα 8.17).

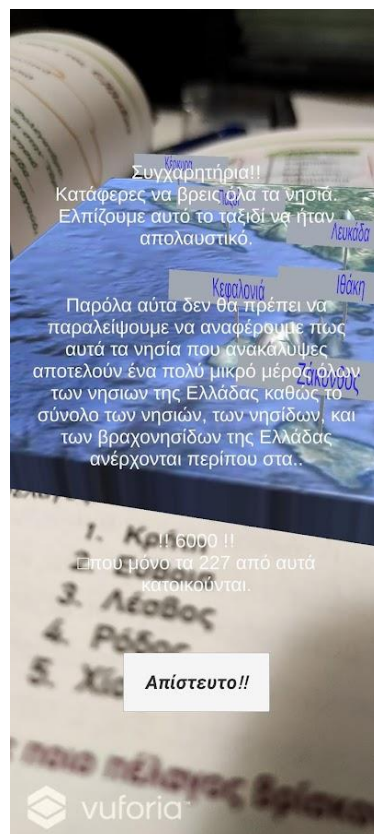
```

166 void SetLettersForSkiathos() {
167
168     IslandLetterCanvas.SetActive(true);
169
170     buttonsName[0].text = "Ι";
171     buttonsName[1].text = "Α";
172     buttonsName[2].text = "Κ";
173     buttonsName[3].text = "Θ";
174 }
175

```

Εικόνα 8.17 Μέθοδος SetLettersForSkiathos()

Αφού γίνει η συμπλήρωση όλων των ονομάτων από τον χρήστη καλείται η μέθοδος EndSporadesScene() όπου ενεργοποιεί την τελευταία ενότητα αυτού του παιχνιδιού. Η τελική ενότητα του παιχνιδιού αφορά τα Επτάνησα και την συμπλήρωση των αντίστοιχων νησιών. Με την ολοκλήρωση της συμπλήρωσης των ονομάτων των αντίστοιχων νησιών γίνεται η εμφάνιση ενός Panel με ένα κείμενο τύπου Text όπου περιγράφεται το συμπέρασμα αυτού του παιχνιδιού και ένα κουμπί όπου επιστρέφει τον χρήστη μέσω της εντολής SceneManager στην αρχική σελίδα (Εικόνα 8.18).



Εικόνα 8.18 Τελικό Panel του παιχνιδιού

8.3 Επίλογος

Το παιχνίδι «Νησιά της Ελλάδας» καλύπτει την ύλη του 10^{ου} Κεφαλαίου στις σελίδες 32-34 του σχολικού βιβλίου, στο μάθημα της Γεωγραφίας της Ε' δημοτικού. Το συγκεκριμένο παιχνίδι γίνεται με την χρήση επαυξημένης πραγματικότητας και στόχος του είναι να μάθει και να εξερευνήσει ο χρήστης τις τοποθεσίες και τα ονόματα των νησιών της Ελλάδας.

Κεφάλαιο 9ο: Τα βουνά της Ελλάδας

9.1 Εισαγωγή

Το παιχνίδι που αναλυθεί σε αυτό το κεφάλαιο έχει να κάνει με την ύλη που καλύπτεται στο 13^ο Κεφάλαιο του σχολικού βιβλίου στο μάθημα της Γεωγραφίας της Ε' δημοτικού με τίτλο «Τα βουνά της Ελλάδας». Επιπλέον περιγράφεται η δομή όλου του παιχνιδιού καθώς και όλες οι λειτουργίες του.

9.2 Η δομή του παιχνιδιού

Το παιχνίδι αυτό περιλαμβάνει την ύλη από το 13^ο κεφάλαιο του σχολικού βιβλίου στις σελίδες 47-49. Στόχος του συγκεκριμένου παιχνιδιού είναι να βοηθήσει τον χρήστη, με την βοήθεια της επαυξημένης πραγματικότητας, να εξερευνήσει και να μάθει τα βουνά της Ελλάδας. Αυτό επιτυγχάνεται απαντώντας σε ερωτήσεις σχετικά με το ύψος και την τοποθεσία του κάθε βουνού. Παράλληλα ο χρήστης έχει την δυνατότητα μέσω επαυξημένων αντικειμένων να εξερευνήσει τα βουνά που του παρουσιάζονται.

Με την έναρξη του παιχνιδιού εμφανίζονται μέσω Panel οι οδηγίες που θα πρέπει να τηρήσει ο χρήστης για να ολοκληρώσει το παιχνίδι με επιτυχία. Το Panel αυτό περιέχει κείμενα (Text) όπου είναι τοποθετημένες οι οδηγίες του παιχνιδιού, και ένα κουμπί (Button) όπου πατώντας το ο χρήστης κατευθύνεται στο επόμενο Panel με την συνέχεια της παρουσίασης των οδηγιών (Εικόνα 9.1).



Εικόνα 9.1 Πρώτο Panel με τις οδηγίες

Η λειτουργία του κουμπιού γίνεται με την βοήθεια του script CanvasManager.cs. Σε αυτό το script, αρχικά γίνεται η αρχικοποίηση όλων των μεταβλητών που χρησιμοποιούνται. Οι πρώτες μεταβλητές τύπου GameObject περιέχουν τα αντικείμενα που εμφανίζονται στο αρχικό Panel με τις οδηγίες ενώ στην συνέχεια γίνεται η αρχικοποίηση των μεταβλητών τύπου GameObject για τους Canvas που εμφανίζονται κατά την διάρκεια του παιχνιδιού. Τέλος γίνεται η αρχικοποίηση των υπόλοιπων μεταβλητών τύπου πάλι GameObject που περιέχουν όλα τα κουμπιά που καθοδηγούν τον χρήστη στην ολοκλήρωση του παιχνιδιού (Εικόνα 9.2).

```

14 public class CanvasManager : MonoBehaviour
15 {
16     //StartCanvasProp
17     public TMP_Text ReviewText;
18     public GameObject MapImage;
19     public GameObject GuideImage;
20     public GameObject GuideImage2;
21     public GameObject NextButton;
22     public GameObject ButtonStart;
23     public GameObject StartGameCanvas;
24
25     public GameObject OlimposInfoCanvas;
26     public GameObject SmolikasInfoCanvas;
27     public GameObject VorasInfoCanvas;
28     public GameObject GkionaInfoCanvas;
29     public GameObject TimfiInfoCanvas;
30
31     //OlimposInfoCanvas Prop
32     public GameObject ButtonStartOlimpos;
33     public GameObject LocationImageOlimpos;
34
35     //SmolikasInfoCanvas Prop
36     public GameObject ButtonStartSmolikas;
37     public GameObject LocationImageSmolikas;
38
39     //VorasInfoCanvas Prop
40     public GameObject ButtonStartVoras;
41     public GameObject LocationImageVoras;
42
43     //GkionaInfoCanvas Prop
44     public GameObject ButtonStartGkiona;
45     public GameObject LocationImageGkiona;
46
47     //TimfiInfoCanvas Prop
48     public GameObject ButtonStartTimfi;
49     public GameObject LocationImageTimfi;
50

```

Εικόνα 9.2 Αρχικοποίηση μεταβλητών στο script CanvasManager.cs

Η διαχείριση και η λειτουργικότητα του κουμπιού στο Panel των οδηγιών γίνεται με την βοήθεια της μεθόδου StartGuideFromCanvas(). Μέσα σε αυτή την μέθοδο γίνεται η αλλαγή του κειμένου τύπου Text και η ενεργοποίηση διαφορετικών δισδιάστατων εικόνων (Εικόνα 9.3).

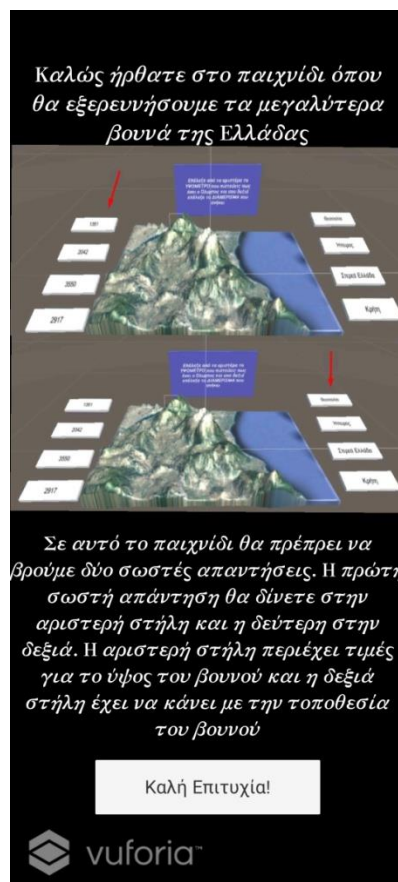
```

52 public void StartGuideFromCanvas()
53
54     ReviewText.text = "Σε αυτό το παιχνίδι θα πρέπει να βρούμε δύο σωστές απαντήσεις. " +
55     "Η πρώτη σωστή απάντηση θα δίνετε στην αριστερή στήλη και η δεύτερη στην δεξιά. Η αριστερή στήλη περιέχει τιμές για το ύψος του βουνού και η δεξιά στήλη έχει να κάνει με την τοποθεσία του βουνού";
56
57
58     GuideImage.SetActive(true);
59     GuideImage2.SetActive(true);
60     MapImage.SetActive(false);
61     ButtonStart.SetActive(true);
62

```

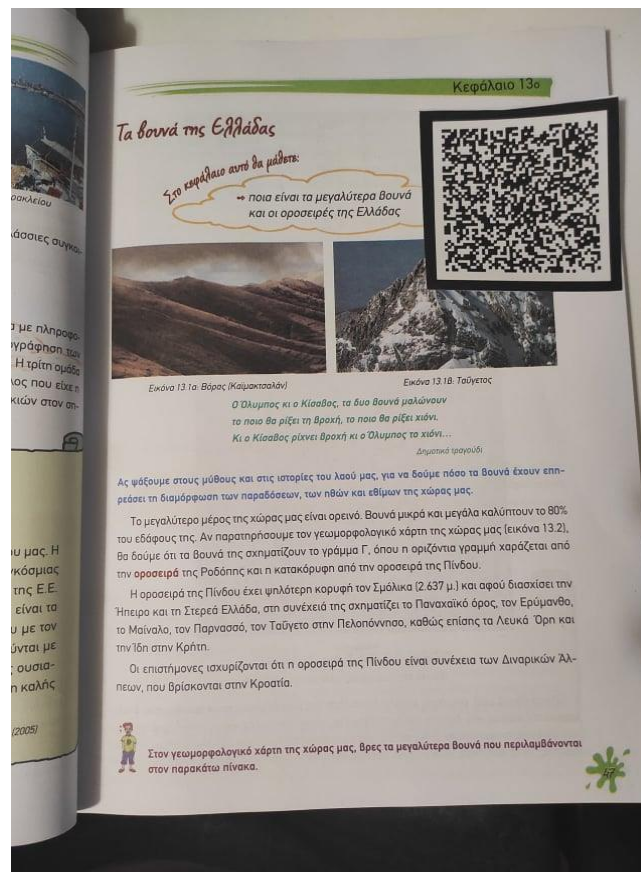
Εικόνα 9.3 Μέθοδος StartGuideFromCanvas()

Στην συνέχεια ο χρήστης θα μεταφερθεί στο επόμενο και τελευταίο εισαγωγικό Panel με το πάτημα του κουμπιού «Επόμενο» και στην συνέχεια θα γίνει η έναρξη του παιχνιδιού (Εικόνα 9.4).



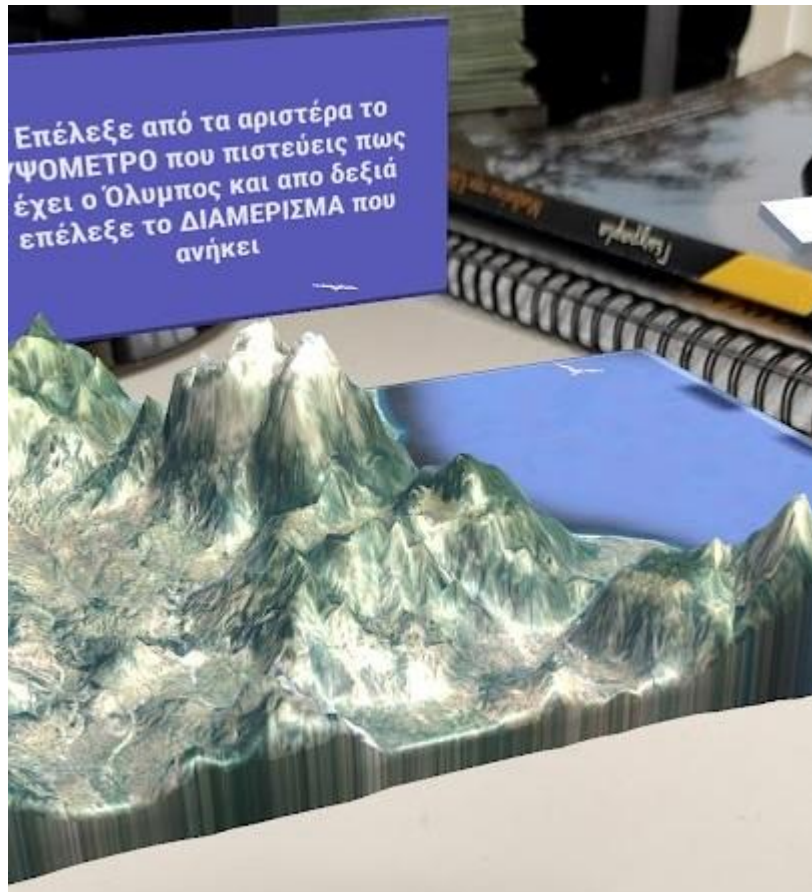
Εικόνα 9.4 Δεύτερο εισαγωγικό Panel

Ο χρήστης θα μεταφερθεί στην επόμενη σκηνή όπου είναι το κύριο παιχνίδι, και θα πρέπει στην συνέχεια να σκανάρει το QR code, το οποίο βρίσκεται στην σελίδα 47 του σχολικού βιβλίου στο μάθημα της Γεωγραφίας της Ε' δημοτικού (Εικόνα 9.5). Με το που γίνει το σκανάρισμα θα εμφανιστεί μπροστά στον χρήστη το επαυξημένο περιβάλλον του παιχνιδιού, και άμεσα θα μπορέσει να ξεκινήσει το παιχνίδι. Το παιχνίδι αυτό, αποτελείται από έναν πίνακα, στον οποίο εμφανίζονται οδηγίες και κείμενα προς τον χρήστη, αντικείμενα (Objects) και κουμπιά μέσω των οποίων θα δίνονται οι απαντήσεις.



Εικόνα 9.5 QR code τοποθετημένο πάνω στο βιβλίο

Στόχος του παιχνιδιού είναι να μάθει ο χρήστης μέσω αυτού, πληροφορίες για τα μεγαλύτερα βουνά της Ελλάδας και μέσω συγκεκριμένων ερωτήσεων να μάθουν διάφορα χαρακτηριστικά για αυτά. Αρχικά ο χρήστης θα πρέπει να επιλέξει από την αριστερή στήλη των κουμπιών το ύψος που πιστεύει πως έχει το βουνό που εμφανίζεται, και στην συνέχεια αν η απάντηση που έδωσε είναι σωστή, από την δεξιά στήλη των κουμπιών, θα πρέπει να επιλέξει την τοποθεσία που πιστεύει πως ανήκει. Με την ολοκλήρωση των απαντήσεων από τον χρήστη εμφανίζεται ένα δισδιάστατο Panel όπου δίνονται πληροφορίες για το συγκεκριμένο βουνό στο οποίο δόθηκαν οι απαντήσεις. Στην συνέχεια το παιχνίδι εμφανίζει το επόμενο βουνό στο οποίο και πάλι ο χρήστης θα πρέπει να δώσει απαντήσεις (Εικόνα 9.6).



Εικόνα 9.6 Το επαυξημένο παιχνίδι

9.2.1 Υλοποίηση της σκηνής του παιχνιδιού

Το κύριο script που χρησιμοποιήθηκε για την υλοποίηση αυτού του παιχνιδιού είναι το Buttons.cs. Το συγκεκριμένο script είναι υπεύθυνο για την διαχείριση όλων των αντικειμένων του παιχνιδιού καθώς και για την αλλαγή των χρωμάτων. Αρχικά γίνεται η αρχικοποίηση των μεταβλητών που χρησιμοποιούνται μέσα στο script. Οι πρώτες μεταβλητές τύπου GameObject περιέχουν τα αντικείμενα των βουνών που εμφανίζονται με την σειρά στον χρήστη. Η λίστα τύπου GameObject περιέχει τα κουμπιά των απαντήσεων ενώ η λίστα τύπου TextMesh περιέχει τα κείμενα των κουμπιών. Οι τελευταίες μεταβλητές περιέχουν τα Canvas που εμφανίζονται μετά το πέρας και των δυο απαντήσεων σε κάθε ένα βουνό (Εικόνα 9.7).

```

14 public class Buttons : MonoBehaviour
15 {
16
17     [SerializeField] UnityEvent anEvent;
18
19     public Text BoardText;
20
21     public GameObject OlimposMountain;
22     public GameObject SmolikasMountain;
23     public GameObject VorasMountain;
24     public GameObject TimfiMountain;
25     public GameObject GkionaMountain;
26
27     public List<GameObject> Bt= new List<GameObject>();
28     public List<TextMesh> ButtonsName = new List<TextMesh>();
29     public bool CorrectAnswer;
30     public bool CorrectLocation;
31
32     public GameObject OlimposInfoCanvas;
33     public GameObject SmolikasInfoCanvas;
34     public GameObject VorasInfoCanvas;
35     public GameObject GkionaInfoCanvas;
36     public GameObject TimfiInfoCanvas;
37
38     public GameObject EndGameCanvas;
39 }

```

Εικόνα 9.7 Αρχικοποίηση μεταβλητών στο script Buttons.cs

Στην συνέχεια έχουν δημιουργηθεί οι κατάλληλες μέθοδοι για το πάτημα του κάθε κουμπιού. Πριν γίνει η ανάλυση των μεθόδων θα γίνει πρώτα η περιγραφή των κουμπιών. Τα κουμπιά για τις απαντήσεις των ερωτήσεων είναι αντικείμενα τύπου κύβοι (cubes). Για να χρησιμοποιηθούν τα αντικείμενα σαν κουμπιά έγινε η χρήση του script MaterialSaver.cs το οποίο στην συνέχεια τοποθετήθηκε μέσα στο κάθε αντικείμενο ξεχωριστά. Το συγκεκριμένο script περιέχει μια μέθοδο η οποία με ένα UnityEvent μας δίνει την δυνατότητα να δημιουργήσουμε ένα συμβάν για οποιοδήποτε αντικείμενο (Εικόνα 9.8).

```

15 public class MaterialSaver : MonoBehaviour
16 {
17
18     [SerializeField] UnityEvent anEvent;
19
20     private void OnMouseDown()
21     {
22         anEvent.Invoke();
23     }
24
25 }

```

Εικόνα 9.8 Μέθοδος OnMouseDown() στο script MaterialSaver.cs

Μετά την τοποθέτηση ενός συμβάντος σε κάθε αντικείμενο υπάρχει η δυνατότητα χρήσης αυτών των αντικειμένων σαν κουμπιά. Κάθε κουμπί έχει μια ξεχωριστή μέθοδο για το καθένα. Για την καλύτερη περιγραφή των μεθόδων γίνεται η ανάλυση μίας μεθόδου από τις έξι καθώς η λογική των μεθόδων είναι η ίδια σε κάθε περίπτωση, απλά για διαφορετικές ερωτήσεις. Μια από τις μεθόδους είναι η `OnClickLocationbt1()`, η οποία χρησιμοποιεί μια συνθήκη `if` όπου γίνεται έλεγχος του αντικειμένου που είναι ενεργοποιημένο εκείνη την στιγμή. Ο έλεγχος αυτός πραγματοποιείται με την εντολή `activeSelf` όπου στην ουσία ελέγχεται αν το αντικείμενο που ελέγχεται εκείνη την στιγμή είναι ενεργοποιημένο, αν εμφανίζεται δηλαδή στον χρήστη (Εικόνα 9.9). Στην περίπτωση που είναι αληθής η συνθήκη, γίνεται η αλλαγή του χρώματος του αντικείμενου σε πράσινο. Στην ουσία η αλλαγή του χρώματος επιτυγχάνεται παίρνοντας το `material` του, και με την εντολή `SetColor` γίνεται η αντικατάσταση του χρώματος. Στην περίπτωση λανθασμένης συνθήκης το χρώμα του αντικειμένου αλλάζει σε κόκκινο χρώμα (Εικόνα 9.10).

```

125  public void OnClickLocationbt1() {
126
127      if (OlimposMountain.activeSelf) {
128
129          var LocationBt1Renderer = Bt[4].transform.GetComponent<Renderer>();
130          LocationBt1Renderer.material.SetColor("_Color", Color.green);
131          CorrectLocation = true;
132          BoardText.text = "Πολύ σωστά. Βρες τα όλα!!";
133          Invoke("CheckIfCorrect",1f);
134
135      }
136      else if (TimfiMountain.activeSelf) {
137
138          var LocationBt1Renderer = Bt[4].transform.GetComponent<Renderer>();
139          LocationBt1Renderer.material.SetColor("_Color", Color.green);
140          CorrectLocation = true;
141          BoardText.text = "Πολύ σωστά. Βρες τα όλα!!";
142          Invoke("CheckIfCorrect",1f);
143
144      }
145      else {
146
147          var LocationBt1Renderer = Bt[4].transform.GetComponent<Renderer>();
148          LocationBt1Renderer.material.SetColor("_Color", Color.red);
149          BoardText.text = "Λάθος απάντηση. Προσπάθησε πάλι!";
150      }
151  }

```

Εικόνα 9.9 Μέθοδος `OnClickLocationbt1()`



Εικόνα 9.10 Αλλαγή χρώματος κουμπιών

Στην συνέχεια, η κάθε μια από τις μεθόδους που αναφέρθηκαν παραπάνω καλεί μια διαφορετική μέθοδο, την `CheckIfCorrect()`. Η συγκεκριμένη μέθοδος ελέγχει αν ο χρήστης έχει απαντήσει σωστά και στις δυο στήλες. Στην περίπτωση που ο χρήστης έχει απαντήσει σωστά και στις δυο στήλες, γίνεται η αλλαγή των κειμένων που περιέχουν όλα τα κουμπιά, και στην συνέχεια γίνεται η αρχικοποίηση των χρωμάτων στα κουμπιά. Παράλληλα, απενεργοποιείται το αντικείμενο που αντικατοπτρίζει ένα βουνό και ενεργοποιείται το επόμενο για να απαντήσει ο χρήστης στις αντίστοιχες ερωτήσεις. Σε αντίθετη περίπτωση, δηλαδή στην περίπτωση όπου ο χρήστης δεν έχει βρει και τις δυο απαντήσεις σωστές, η διαδικασία του παιχνιδιού προχωρά κανονικά έως ότου βρεθούν και οι δυο σωστές απαντήσεις (Εικόνα 9.11).

```

void CheckIfCorrect() {
    if (OlimposMountain.activeSelf && CorrectAnswer && CorrectLocation) {

        CorrectAnswer = false;
        CorrectLocation = false;

        OlimposMountain.SetActive(false);
        SmolikasMountain.SetActive(true);

        ButtonsName[0].text = "1553";
        ButtonsName[1].text = "2637";
        ButtonsName[2].text = "4672";
        ButtonsName[3].text = "2150";

        //locationButtons
        ButtonsName[4].text = "Κρήτη";
        ButtonsName[5].text = "Μακεδονία";
        ButtonsName[6].text = "Θράκη";
        ButtonsName[7].text = "Ηπειρος";

        //Default Color
        var HighBt1Renderer = Bt[0].transform.GetComponent<Renderer>();
        HighBt1Renderer.material.SetColor("_Color", Color.white);
        var HighBt2Renderer = Bt[1].transform.GetComponent<Renderer>();
        HighBt2Renderer.material.SetColor("_Color", Color.white);
        var HighBt3Renderer = Bt[2].transform.GetComponent<Renderer>();
        HighBt3Renderer.material.SetColor("_Color", Color.white);
        var HighBt4Renderer = Bt[3].transform.GetComponent<Renderer>();
        HighBt4Renderer.material.SetColor("_Color", Color.white);
        var LocationBt1Renderer = Bt[4].transform.GetComponent<Renderer>();
        LocationBt1Renderer.material.SetColor("_Color", Color.white);
        var LocationBt2Renderer = Bt[5].transform.GetComponent<Renderer>();
        LocationBt2Renderer.material.SetColor("_Color", Color.white);
        var LocationBt3Renderer = Bt[6].transform.GetComponent<Renderer>();
        LocationBt3Renderer.material.SetColor("_Color", Color.white);
        var LocationBt4Renderer = Bt[7].transform.GetComponent<Renderer>();
        LocationBt4Renderer.material.SetColor("_Color", Color.white);

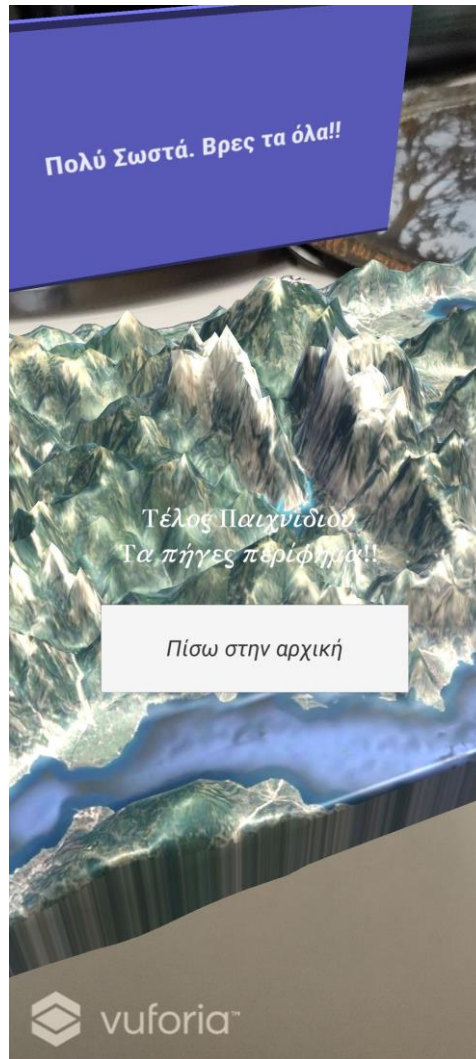
        BoardText.text = "Μπράβο!! Βρήκες και τις 2 σωστές απαντήσεις. Πάμε στα επόμενα βουνά που είναι 8 Σμόλικας";

        OlimposInfoCanvas.SetActive(true);
    }
}

```

Εικόνα 9.11 Μέθοδος CheckIfCorrect()

Με το πέρας των σωστών απαντήσεων όλων των αντικειμένων ενεργοποιείται ο τελικός Canvas, στον οποίο εμπεριέχεται ένα κουμπί «Πίσω στην αρχική», όπου και γίνεται η επιστροφή του χρήστη στην αρχική σελίδα, και ένα κείμενο τύπου Text, το οποίο προτρέπει τον χρήστη να πατήσει το συγκεκριμένο κουμπί (Εικόνα 9.12).

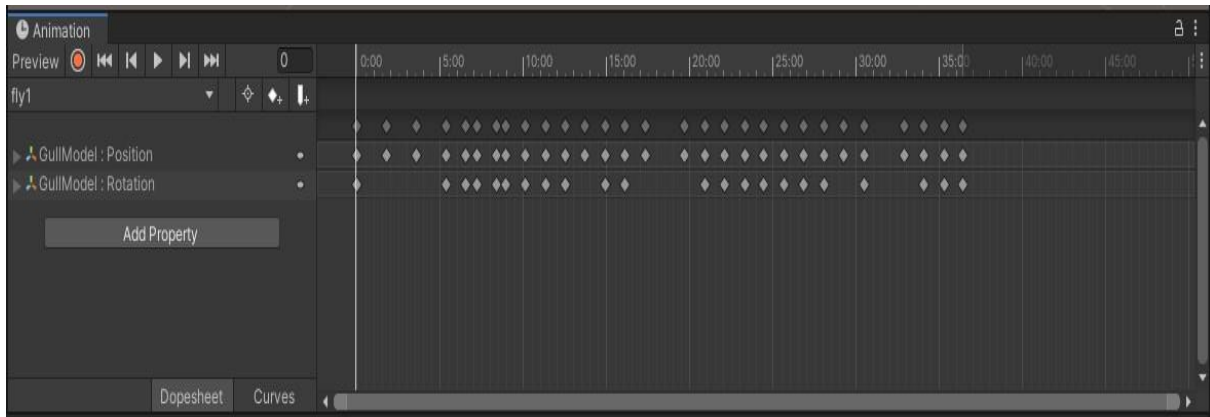


Εικόνα 9.12 Εμφάνιση κουμπιού «Πίσω στην αρχική»

9.2.2 Animations

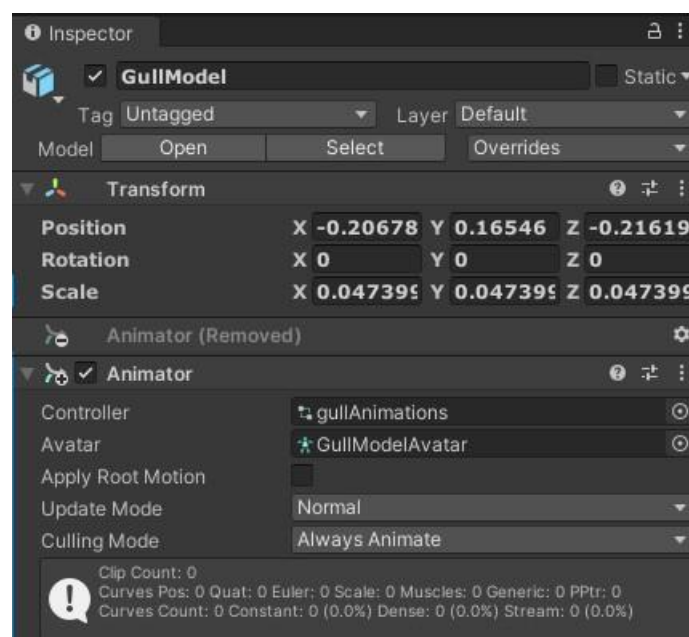
Πέρα από τις λειτουργίες που περιγράφηκαν παραπάνω, το συγκεκριμένο παιχνίδι περιέχει και μερικά κινούμενα σχέδια (Animations) όπου πραγματοποιούνται καθ όλη την διάρκεια του παιχνιδιού. Πιο συγκεκριμένα τρεις γλάροι, οι οποίοι είναι αντικείμενα πάνω στον επαυξημένο κόσμο «πετούν» σε διαφορετικά μέρη το ένα με το άλλο όσο ο χρήστης παίζει το παιχνίδι.

Η υλοποίηση για τα κινούμενα σχέδια έγινε με την βοήθεια ενός εργαλείου που παρέχεται από την Unity και ονομάζεται Animator. Μέσα σε έναν Animator μπορούν να δημιουργηθούν διάφορα Animations και στην συνέχεια να τοποθετηθούν μέσα σε αυτόν. Για την δημιουργία των Animations το Unity δίνει την δυνατότητα να καταγράψει ο χρήστης τις τοποθεσίες, τις περιστροφές και τα μεγέθη ενός αντικειμένου σε διάστημα κάποιου χρόνου, και στην συνέχεια να το αναπαράγει δημιουργώντας έτσι μια κινούμενη εικόνα (Εικόνα 9.13).



Εικόνα 9.13 Δημιουργία Animation

Στην συνέχεια, το Animations καταχωρείται μέσα στον Animator ο οποίος με την σειρά του τοποθετείται μέσα στο αντικείμενο τύπου Object που έχει επιλεγεί για να αναπαράγει μια κινούμενη εικόνα (Εικόνα 9.14).



Εικόνα 9.14 Τοποθέτηση Animator σε αντικείμενο

9.3 Επίλογος

Το παιχνίδι «Τα βουνά της Ελλάδας» καλύπτει το Κεφάλαιο 13 στις σελίδες 47-49 του σχολικού βιβλίου στο μάθημα της Γεωγραφίας της Ε΄ δημοτικού. Το συγκεκριμένο παιχνίδι έχει δημιουργηθεί με την χρήση επανξημένης πραγματικότητας και στόχος του παιχνιδιού είναι να προσφέρει γνώσεις για τα μεγαλύτερα βουνά της Ελλάδας, καθώς και πληροφορίες σχετικά με το ύψος και την τοποθεσία των βουνών.

Κεφάλαιο 10ο: Το κλίμα της Ελλάδας

10.1 Εισαγωγή

Το παιχνίδι με τίτλο «Το κλίμα της Ελλάδας» καλύπτει την ύλη που υπάρχει στο κεφάλαιο 17^ο στις σελίδες 59-62 στο μάθημα της Γεωγραφίας της Ε' δημοτικού. Σε αυτό το κεφάλαιο γίνεται η περιγραφή του συγκεκριμένου παιχνιδιού καθώς επίσης αναλύεται η δομή και οι λειτουργίες όλου του παιχνιδιού.

10.2 Η δομή του παιχνιδιού

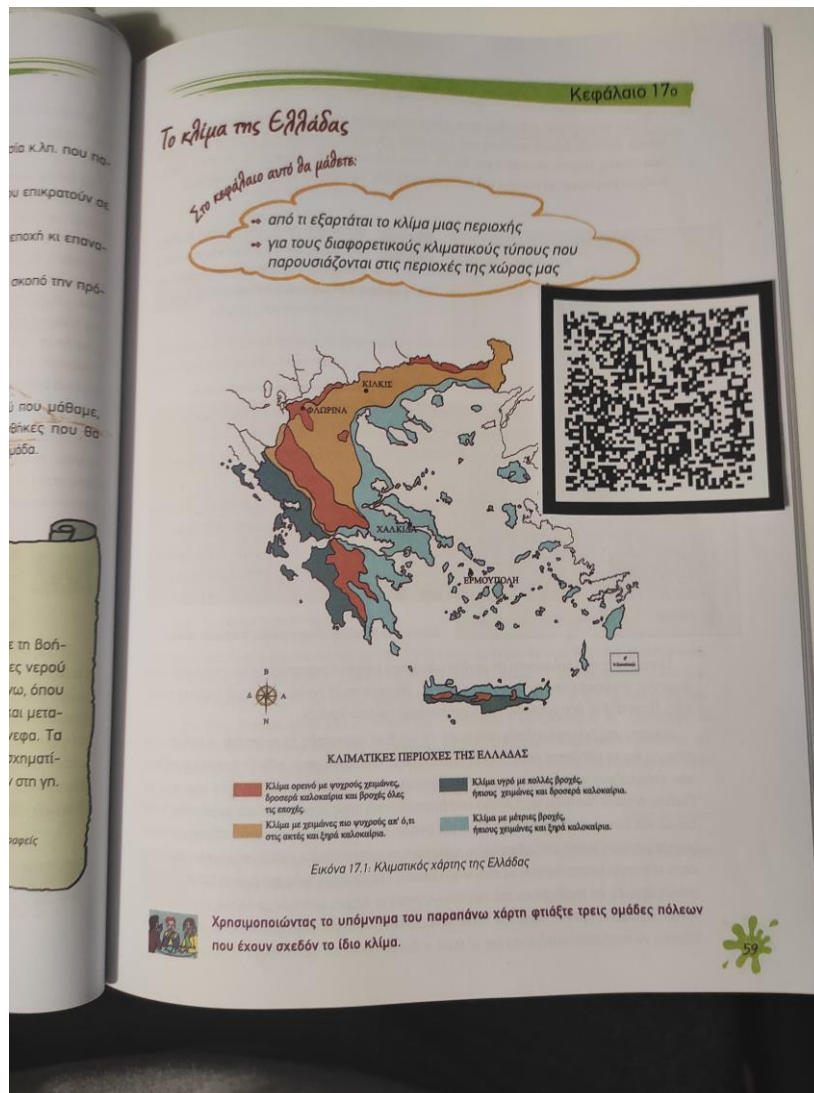
Το συγκεκριμένο παιχνίδι καλύπτει την ύλη του σχολικού βιβλίου στις σελίδες 59-62. Στόχος του είναι να βοηθήσει τον χρήστη να καταλάβει τα καιρικά φαινόμενα που υπάρχουν, καθώς και τον συμβολισμό τους. Αυτό επιτυγχάνεται με τις διάφορες ερωτήσεις που δίνονται προς απάντηση για τον χρήστη, καθώς και με την παρατήρηση των συμβολισμών που εμφανίζονται κατά την διάρκεια του παιχνιδιού. Μετά την ολοκλήρωση όλων των ερωτήσεων, εμφανίζονται στον χρήστη διάφοροι πίνακες, οι οποίοι συλλέγουν δεδομένα καιρικών φαινομένων σε πραγματικό χρόνο, με σκοπό να αναλυθούν ώστε να γίνουν κατανοητά από τον χρήστη.

Αρχικά με την έναρξη του παιχνιδιού, ο χρήστης θα πρέπει να διαβάσει και να κατανοήσει της οδηγίες που του δίνονται ώστε να μπορέσει να ολοκληρώσει με επιτυχία το παιχνίδι. Οι οδηγίες δίνονται μέσω ενός δισδιάστατου Panel που εμφανίζεται με την έναρξη του παιχνιδιού, και περιέχει ένα κείμενο τύπου Text με τις οδηγίες καθώς και ένα κουμπί «Πάμε!» το οποίο οδηγεί τον χρήστη στο βασικό παιχνίδι (Εικόνα 10.1).



Εικόνα 10.1 Αρχικό Panel παιχνιδιού

Στην συνέχεια, και αφού γίνουν κατανοητές οι οδηγίες, του δίνεται η δυνατότητα να σκανάρει με την συσκευή του το QR code που βρίσκεται στην σελίδα 59 του σχολικού βιβλίου (Εικόνα 10.2). Όταν ο χρήστης σκανάρει με επιτυχία, εμφανίζεται μπροστά του το επαυξημένο περιβάλλον το οποίο αποτελεί το συγκεκριμένο παιχνίδι.



Εικόνα 10.2 QR code παιχνιδιού πάνω στο σχολικό βιβλίο

Εφόσον ολοκληρωθεί η πρώτη φάση που περιγράφηκε παραπάνω, σειρά έχει η απάντηση των ερωτήσεων από τον χρήστη. Ουσιαστικά ο χρήστης καλείται να επιλέξει μέσω των ταμπελών που υπάρχουν επάνω στο επαυξημένο περιβάλλον, το κάθε καιρικό φαινόμενο, και να συμπληρώσει τα κενά που του εμφανίζονται σε ένα ξεχωριστό Panel για το καθένα. Η επιλογή της ταμπέλας δεν έχει συγκεκριμένη σειρά, ο χρήστης μπορεί να ξεκινήσει την συμπλήρωση των κενών από όποιο καιρικό φαινόμενο επιθυμεί (Εικόνα 10.3).



Εικόνα 10.3 Πινακίδες καιρικών φαινομένων

Η συγκεκριμένη σκηνή έχει υλοποιηθεί με το script `QuestionButtons.cs`, το οποίο αποτελεί και το βασικό script αυτού του παιχνιδιού. Αρχικά, μέσα στο script γίνεται η αρχικοποίηση όλων των μεταβλητών που χρησιμοποιούνται από την συγκεκριμένη σκηνή. Οι μεταβλητές τύπου `TMP_Text` αρχικοποιούν όλα τα κείμενα που εμπεριέχονται μέσα στα `Panels` των ερωτήσεων για το κάθε καιρικό φαινόμενο ξεχωριστά. Οι τέσσερις λίστες τύπου `Button` περιέχουν τα κουμπιά που θα δίνονται στον χρήστη σαν επιλογές των απαντήσεων του. Οι υπόλοιπες μεταβλητές τύπου `GameObject` αποτελούν τις πινακίδες του κάθε φαινομένου πάνω στον επαυξημένο κόσμο, καθώς και τα `Canvas` που εμφανίζονται ανά διαστήματα κατά την διάρκεια του παιχνιδιού (Εικόνα 10.4).

```

15 public class QuestionButton : MonoBehaviour
16 {
17
18     public TMP_Text QuestionText;
19     public TMP_Text SunnyQuestionText;
20     public TMP_Text SnowQuestionText;
21     public TMP_Text FogQuestionText;
22
23
24     public TMP_Text AnswerNumberText;
25     public TMP_Text SnowAnswerNumberText;
26     public TMP_Text FogAnswerNumberText;
27
28     public List<Button> AnswerButtons = new List<Button>();
29     public List<Button> SunnyAnswerButtons = new List<Button>();
30     public List<Button> SnowAnswerButtons = new List<Button>();
31     public List<Button> FogAnswerButtons = new List<Button>();
32
33     public GameObject RainWeatherCanvas;
34     public GameObject SunnyWeatherCanvas;
35     public GameObject SnowWeatherCanvas;
36     public GameObject FogWeatherCanvas;
37
38     public GameObject RainPanel;
39     public GameObject SunnyPanel;
40     public GameObject SnowPanel;
41     public GameObject FogPanel;
42
43     public GameObject CorrectRainPanel;
44     public GameObject CorrectSunnyPanel;
45     public GameObject CorrectSnowPanel;
46     public GameObject CorrectFogPanel;
47
48     public GameObject ButtonCloseCanvas;
49     public GameObject StartGameCanvas;
50
51     public GameObject WeatherButton; |
52

```

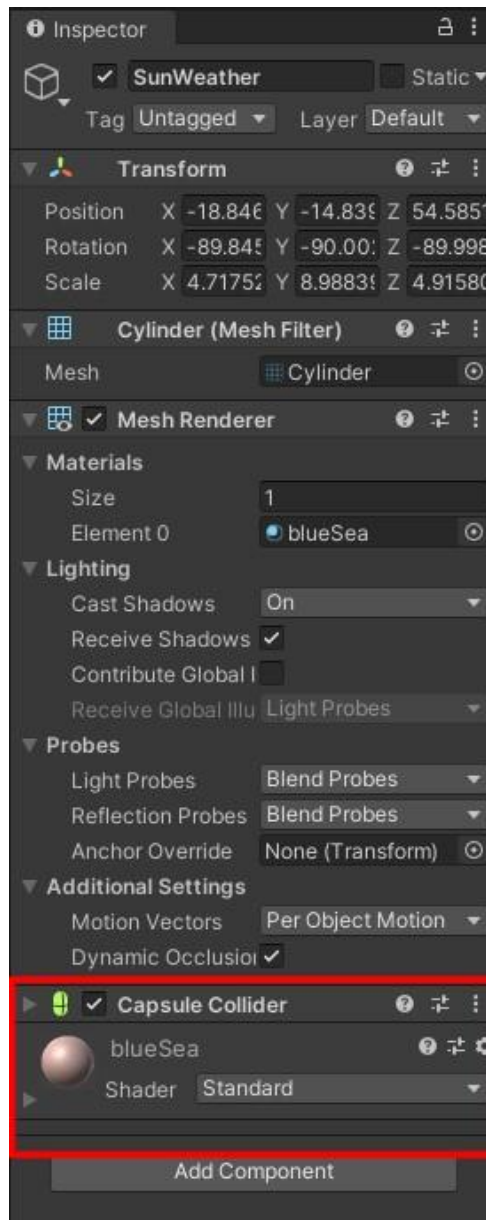
Εικόνα 10.4 Αρχικοποίηση μεταβλητών στο script QuestionButton.cs

Με το πάτημα κάθε πινακίδας ενεργοποιείται το script WeatherManager.cs. Στην συνέχεια καλείται η αντίστοιχη μέθοδος για την κάθε πινακίδα ενεργοποιώντας τον Canvas των ερωτήσεων και αλλάζοντας τα κείμενα των κουμπιών κατάλληλα για να συμπληρωθούν τα κενά (Εικόνα 10.5).

```
30 public void OnClickRainWeather() {  
31  
32     ButtonText[0].text = "Σπάνιες";  
33     ButtonText[1].text = "Συχνές";  
34     ButtonText[2].text = "Ανύπαρκτες";  
35     ButtonText[3].text = "Καθημερινές";  
36  
37     RainWeatherCanvas.SetActive(true);  
38 }  
39
```

Εικόνα 10.5 Μέθοδος OnClickRainWeather() στο script WeatherManager.cs

Παρόλο που οι πινακίδες δεν είναι τύπος κουμπιού (Button), έχουν σχεδιαστεί κατάλληλα για να μπορούν με το πάτημα τους να ενεργοποιούν μια λειτουργία. Πιο συγκεκριμένα πάνω στα αντικείμενα των πινακίδων έχουν τοποθετηθεί UnityEvents. Στην ουσία με αυτόν τον τρόπο επιτρέπουμε το αντικείμενο να εμπεριέχει ένα γεγονός (Event) και να εκτελεί μια λειτουργία ή ένα κομμάτι κώδικα. Για να λειτουργεί σωστά αυτός ο μηχανισμός στην Unity χρειάζεται η προσθήκη ενός εργαλείου σε κάθε αντικείμενο, που παρέχεται από την ίδια την Unity. Το εργαλείο αυτό ονομάζεται Capsule Collider, και χρησιμοποιείται για την ενεργοποίηση συμβάντων (Εικόνα 10.6).



Εικόνα 10.6 Capsule Collider

Στην συνέχεια, και αφού έχει γίνει επιλογή μιας πινακίδας, γίνεται η εμφάνιση ενός δισδιάστατου Panel στον χρήστη όπου δίνεται η πρόταση που είναι προς συμπλήρωση. Το κάτω μέρος του Panel περιέχει τα τέσσερα κουμπιά με τα οποία ο χρήστης θα πρέπει να συμπληρώσει τα κενά με την κατάλληλη σειρά. Κάθε κουμπί ενεργοποιεί μια αντίστοιχη μέθοδο. Η δομή των μεθόδων είναι η ίδια, με διαφορετικά στοιχεία, καθώς η κάθε μια αντιστοιχεί σε ξεχωριστό Panel. Για αυτό τον λόγο θα γίνει η περιγραφή μιας μεθόδου με το όνομα `OnClickButton3()`. Σε αυτή την μέθοδο γίνεται η χρήση μιας συνθήκης `if` όπου ελέγχει το Panel που είναι ενεργοποιημένο εκείνη την στιγμή. Αν το κουμπί που έχει πατηθεί ταιριάζει με το Panel που είναι ενεργοποιημένο τότε το χρώμα του κουμπιού αλλάζει σε πράσινο και συμπληρώνεται κατάλληλα η πρόταση με το κείμενο που έχει επιλεγεί. Σε αντίθετη περίπτωση το κουμπί παίρνει το κόκκινο χρώμα, και το παιχνίδι συνεχίζεται προσπαθώντας από τον χρήστη να συμπληρώσει κατάλληλα τα κενά (Εικόνα 10.7).

```

125 public void OnClickButton3() {
126
127     if (RainWeatherCanvas.activeSelf) {
128         AnswerButtons[2].GetComponent<Image>().color = Color.red;
129     } else if (SunnyWeatherCanvas.activeSelf) {
130         SunnyAnswerButtons[2].GetComponent<Image>().color = Color.red;
131     } else if (SnowWeatherCanvas.activeSelf) {
132         SnowAnswerButtons[2].GetComponent<Image>().color = Color.red;
133     } else if (FogWeatherCanvas.activeSelf) {
134         FogQuestionText.text = "Η ομίχλη είναι φυσικό φαινόμενο αρκετά κοντά στην επιφάνεια του εδάφους ή της θάλασσας. Τον χειμώνα παρατηρείται αρκετά συχνά στην Θεσσαλονίκη.";
135         for (int i = 0; i <=3; i++) {
136             FogAnswerButtons[i].gameObject.SetActive(false);
137         }
138         ButtonCloseCanvas.SetActive(true);
139     }
140 }
141

```

Εικόνα 10.7 Μέθοδος OnClickButton3() στο script QuestionButton.cs

Κατά την διάρκεια όλου του παιχνιδιού η μέθοδος Update() είναι ενεργοποιημένη και ελέγχει αν έχουν συμπληρωθεί σωστά όλες οι πινακίδες του παιχνιδιού (Εικόνα 10.8). Στην περίπτωση που ολοκληρωθούν σωστά όλες οι πινακίδες του παιχνιδιού, αυτόματα ενεργοποιείται ένα επανυζημένο αντικείμενο, το οποίο με το πάτημα του ενεργοποιεί κάποια API calls που τα περιγράφουν παρακάτω.

```

195 void Update() {
196     if (SunnyQuestionText.text == "Το κλίμα στην Θεσσαλονίκη είναι μεσογειακό. Τα καλοκαίρια στην Θεσσαλονίκη είναι ξηρά και ζεστά") {
197         for (int i = 0; i <=3; i++) {
198             SunnyAnswerButtons[i].gameObject.SetActive(false);
199         }
200         ButtonCloseCanvas.SetActive(true);
201     }
202
203
204
205     if(CorrectRainPanel.activeSelf && CorrectSunnyPanel.activeSelf && CorrectSnowPanel.activeSelf && CorrectFogPanel.activeSelf){
206
207         WeatherButton.SetActive(true);
208     }
209
210 }
211

```

Εικόνα 10.8 Μέθοδος Update()

10.2.1 Χρήση API calls στο παιχνίδι

Με το πάτημα του κουμπιού που αναφέρθηκε παραπάνω, ενεργοποιείται ένα διαφορετικό script με το όνομα APIWeather.cs. Το συγκεκριμένο script καλεί μέσω κάποιων API calls, δεδομένα που δείχνουν στοιχεία καιρικών φαινομένων σε πραγματικό χρόνο. Για την πραγματοποίηση αυτής της λειτουργίας χρησιμοποιήθηκε η ιστοσελίδα openweathermap.org όπου όλα τα δεδομένα και τα στοιχεία απορροφούνται από εκεί (Εικόνα 10.9). Αρχικά, χρειάστηκε η δημιουργία ενός ιδιωτικού λογαριασμού στην ιστοσελίδα για μπορέσει να γίνει η πρόσβαση στις υπηρεσίες της.



Εικόνα 10.9 Ιστοσελίδα Openweathermap.org

Στην συνέχεια γίνεται η ανάκτηση ενός μοναδικού κλειδιού, το οποίο τοποθετείται με την σειρά του μέσα στον κώδικα για να γίνει εφικτή η δυνατότητα ανάκτησης δεδομένων (Εικόνα 10.10).



Εικόνα 10.10 Ανάκτηση κλειδιού από την ιστοσελίδα

Από την στιγμή που γίνει με επιτυχία η ανάκτηση του μοναδικού κλειδιού, μπορεί να ξεκινήσει και η πρόσβαση στα δεδομένα, με API calls. Τα δεδομένα σε αυτό το παιχνίδι ανακτούνται όπως προαναφέρθηκε με το script APIWeather.cs. Αρχικά, γίνεται η αρχικοποίηση των μεταβλητών. Οι μεταβλητές τύπου TextMeshProGUI περιέχουν τα κείμενα που θα συμπληρωθούν με τα δεδομένα από τα API calls. Στην συνέχεια, με την έναρξη του script η μέθοδος Start() καλεί μια άλλη μέθοδο, την UpdateWeatherData(). Η συγκεκριμένη μέθοδος χρησιμοποιεί, τον υπέρ-σύνδεσμο (URL) της ιστοσελίδας, όπου περιέχει το γεωγραφικό πλάτος, το γεωγραφικό μήκος και το μοναδικό κλειδί που ανακτήθηκε. Στην συγκεκριμένη περίπτωση γίνεται ανάκτηση δεδομένων για την πόλη της

Θεσσαλονίκης. Επιπλέον, μαζί με την βιβλιοθήκη της Unity, η UnityEngine.Networking, πραγματοποιείται η σύνδεση για την ανάκτηση των δεδομένων που χρειάζεται το πρόγραμμα (Εικόνα 10.11). Με την βοήθεια της εντολής response γίνεται η επιλογή των πεδίων που θα χρειαστεί το παιχνίδι και εμφανίζει τα δεδομένα πάνω στον πίνακα (Εικόνα 10.12).

```

40 private void UpdateWeatherData() {
41     StartCoroutine(FetchWeatherDataFromApi(lastLocation.latitude.ToString(), lastLocation.longitude.ToString()));
42 }
43 private IEnumerator FetchWeatherDataFromApi(string latitude, string longitude)
44 {
45     string url = "http://api.openweathermap.org/data/2.5/weather?lat=40.6401&lon=22.9444&appid=7574fd289a4b3082492bc56d90b58133&units=metric&lang=el";
46     UnityWebRequest fetchWeatherRequest = UnityWebRequest.Get(url);
47     yield return fetchWeatherRequest.SendWebRequest();
48     if (fetchWeatherRequest.IsNetworkError || fetchWeatherRequest.IsHttpError)
49     {
50
51         statusText.text = fetchWeatherRequest.error;
52
53     } else {
54
55         var response = JSON.Parse(fetchWeatherRequest.downloadHandler.text);
56
57         location.text = response["name"];
58         country.text = response["country"];
59         description.text = response["weather"][0]["description"];
60         temp.text = response["main"]["temp"] + " C";
61         temp_min.text = "Μικρότερη θερμοκρασία " + response["main"]["temp_min"] + " C";
62         temp_max.text = "Μέγιστη θερμοκρασία " + response["main"]["temp_max"] + " C";
63         pressure.text = "Η πίεση είναι " + response["main"]["pressure"] + " Pa";
64         humidity.text = response["main"]["humidity"] + " % υγρασία";
65         windspeed.text = "Ταχύτητα του αέρα " + response["wind"]["speed"] + " Km/h";
66
67         Debug.Log(temp_min.text);
68         Debug.Log(temp_max.text);
69     }
70 }
71

```

Εικόνα 10.11 Μέθοδος UpdateWeatherData()



Εικόνα 10.12 Αποτελέσματα πίνακα με API calls

10.3 Επίλογος

Το παιχνίδι «Το κλίμα της Ελλάδας» καλύπτει την ύλη του παιχνιδιού που υπάρχει στο Κεφάλαιο 17^ο στις σελίδες 59-62 του σχολικού βιβλίου στο μάθημα της Γεωγραφίας της Ε' δημοτικού. Το παιχνίδι έχει δημιουργηθεί με την βοήθεια της επαυξημένης πραγματικότητας (AR) και στόχος του είναι η κατανόηση των καιρικών συνθηκών.

Κεφάλαιο 11ο: Γεωγραφική κατανομή του πληθυσμού

11.1 Εισαγωγή

Στο κεφάλαιο αυτό περιγράφεται το παιχνίδι που έχει δημιουργηθεί για το κεφάλαιο «Η γεωγραφική κατανομή του πληθυσμού της Ελλάδος» το οποίο αποτελεί το 30^ο Κεφάλαιο της Γ' Ενότητας, του σχολικού βιβλίου της Ε' δημοτικού. Στόχος το παιχνιδιού είναι να δώσει μια βασική ιδέα για τον τρόπο και τον λόγο με τον οποίο γίνεται η κατανομή του πληθυσμού σε μια χώρα. Επιπλέον γίνεται ανάλυση όλων των βημάτων του παιχνιδιού καθώς και ο κώδικας που χρησιμοποιήθηκε για την υλοποίησή του.

11.2 Υλοποίηση Παιχνιδιού

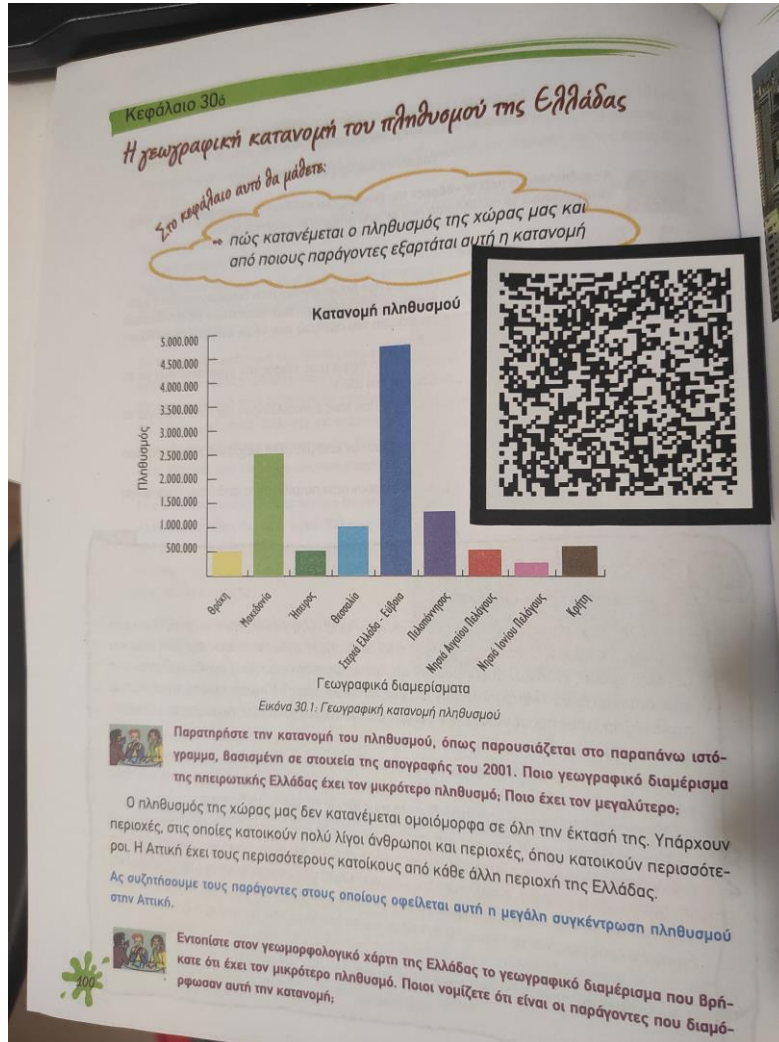
Το συγκεκριμένο παιχνίδι περιλαμβάνει την ύλη του 30^{ου} Κεφαλαίου του σχολικού βιβλίου στις σελίδες 100-102. Με την βοήθεια της επαυξημένης πραγματικότητας, ο χρήστης θα έχει την δυνατότητα να κατανοήσει καλύτερα την λογική που υπάρχει στην κατανομή ενός πληθυσμού, και συγκεκριμένα της Ελλάδας. Με την λειτουργία της μετακίνησης ενός αντικειμένου (Drag and Drop) ο χρήστης θα είναι σε θέση να τοποθετεί κατά την άποψη του τον πληθυσμό που πιστεύει πως αντιστοιχεί σε κάθε κομμάτι της Ελλάδας.

Αρχικά με το άνοιγμα του παιχνιδιού, γίνεται η εμφάνιση του πρώτου Panel με τις οδηγίες του παιχνιδιού. Στην ουσία, εξηγεί τα βήματα που θα πρέπει να ακολουθήσει ώστε να ολοκληρώσει με επιτυχία το παιχνίδι. Οι οδηγίες παρέχονται από το συγκεκριμένο Panel σε μορφή κειμένου (Text) ενώ παράλληλα υπάρχουν και δύο εικόνες (Images) για την καλύτερη κατανόηση του. Πιο αναλυτικά, το παιχνίδι ζητάει από τον χρήστη να τοποθετήσει τα νούμερα με τον πληθυσμό που του δίνονται, στην σωστή θέση. Παράλληλα με κάθε σωστή απάντηση γίνεται η εμφάνιση σχετικών πληροφοριών για κάθε περιοχή ξεχωριστά. Στην συνέχεια ο χρήστης θα πρέπει να πατήσει το κουμπί «Ξεκινάμε!» για να ξεκινήσει το κυρίως παιχνίδι (Εικόνα 11.1).



Εικόνα 11.1 Πρώτο Panel παιχνιδιού

Στην συνέχεια, ο χρήστης θα πρέπει να σκανάρει με την συσκευή του το QR code του συγκεκριμένου παιχνιδιού, το οποίο βρίσκεται στην σελίδα 100 του σχολικού βιβλίου (Εικόνα 11.2).



Εικόνα 11.2 QR Code τοποθετημένο πάνω στο σχολικό βιβλίο

Αφού γίνει επιτυχές σκανάρισμα, θα εμφανιστεί το επανυξημένο περιβάλλον του παιχνιδιού μέσω της συσκευής που χειρίζεται ο χρήστης, καθώς και ένα δισδιάστατο Panel με φιγούρες ανθρώπων που περιέχουν μια τιμή, διαφορετική η κάθε μια. Αυτές οι τιμές αντιστοιχούν σε ένα σύνολο πληθυσμού για κάθε μια από τις περιοχές που καλείται ο χρήστης να απαντήσει. Οι τιμές δεν αποτελούν αυθαίρετο αριθμό καθώς έχουν τοποθετηθεί με βάση την Ελληνική Στατιστική Αρχή από την απογραφή του πληθυσμού που ολοκληρώθηκε το 2011 (Εικόνα 11.3).



Ελληνική Στατιστική Αρχή

Εικόνα 11.3 Ελληνική Στατιστική Αρχή (ELSTAT)

Στην συνέχεια ο χρήστης θα πρέπει να επιλέξει μια πινακίδα από τις 6 που υπάρχουν στο σύνολο πάνω στον χάρτη της Ελλάδας. Αφού γίνει η επιλογή, ο χρήστης θα πρέπει να τοποθετήσει την απάντηση του μέσα σε ένα πλαίσιο για να γίνει ο έλεγχος της απάντησης. Σε περίπτωση σωστής απάντησης εμφανίζεται ένα κουμπί «Σωστό» το οποίο με το πάτημα του ανοίγει ένα καινούργιο Panel πληροφοριών σχετικά με την αντίστοιχη περιοχή που επέλεξε. Σε αντίθετη περίπτωση, η επιλογή του χρήστη επιστρέφει στην αρχική θέση, και ο χρήστης καλείται να δώσει διαφορετική απάντηση (Εικόνα 11.4).

Με την ολοκλήρωση όλων των αντιστοιχίσεων, και στις έξι πινακίδες που υπάρχουν πάνω στο χάρτη, αυτόματα εμφανίζεται μπροστά στον χρήστη ένα δισδιάστατο κουμπί «Αρχική» μαζί με μια εικόνα. Με αυτόν τον τρόπο γίνεται η ενημέρωση του χρήστη πως έχει ολοκληρώσει με επιτυχία το παιχνίδι (Εικόνα 11.5).



Εικόνα 11.4 Αποτέλεσμα σωστής απάντησης



Εικόνα 11.5 Κουμπί εξόδου παιχνιδιού

11.2.1 Υλοποίηση κώδικα σκηνής

Η υλοποίηση του συγκεκριμένου παιχνιδιού επιτυγχάνεται με τις λειτουργίες δύο scripts τα οποία είναι γραμμένα σε γλώσσα προγραμματισμού C#. Βασικό script παρόλα αυτά αποτελεί το CanvasButtonManager.cs καθώς περιέχει όλες τις βασικές λειτουργίες αυτής της σκηνής. Αρχικά, γίνεται η αρχικοποίηση όλων των μεταβλητών που χρησιμοποιούνται στο script. Οι πρώτες έξι μεταβλητές τύπου GameObject αντιπροσωπεύουν τις δισδιάστατες εικόνες με τις φιγούρες των ανθρώπων, με τις οποίες ο χρήστης θα πρέπει να σύρει και να τις τοποθετήσει στην σωστή πινακίδα που αντιστοιχούν. Οι υπόλοιπες μεταβλητές τύπου GameObject περιέχουν όλα τα δισδιάστατα Panels τα οποία περιέχουν πληροφορίες για την κάθε γεωγραφική περιοχή της Ελλάδας. Οι μεταβλητές τύπου TMP_Text περιέχουν τα κείμενα που υπάρχουν πάνω στις πινακίδες που επαυξάνονται πάνω στον εικονικό περιβάλλον (Εικόνα 11.6).

```

15 public class CanvasButtonManager : MonoBehaviour
16 {
17     public GameObject PersonAigaio;
18     public GameObject PersonAttiki;
19     public GameObject PersonIonio;
20     public GameObject PersonKriti;
21     public GameObject PersonMakedonia;
22     public GameObject PersonPeloponisos;
23
24
25     public GameObject GameButtonsCanvas;
26     public GameObject AboutInfoCanvas;
27     public GameObject AboutAigaioCanvas;
28     public GameObject AboutAttikiCanvas;
29     public GameObject AboutIonioCanvas;
30     public GameObject AboutKritiCanvas;
31     public GameObject AboutPeloponisosCanvas;
32
33     public GameObject QuestionCanvas;
34     public Text QuestionText;
35
36     //----PanelText-----//
37     public TMP_Text MakedoniaText;
38     public TMP_Text AigaioText;
39     public TMP_Text AttikiText;
40     public TMP_Text IonioText;
41     public TMP_Text KritiText;
42     public TMP_Text PeloponisosText;
43
44     public GameObject AnswerPanel;
45     public GameObject AboutInfoButton;
46
47     public GameObject EndGameCanvas;
48
49     //----StartGameCanvas-----//
50
51     public GameObject StartGameCanvas;
52

```

Εικόνα 11.6 Αρχικοποίηση μεταβλητών στο script CanvasButtonManager.cs

Στην συνέχεια ακολουθεί μία μέθοδος void η `OnClickAboutInfoBt()`. Η μέθοδος αυτή χρησιμοποιεί μία συνθήκη `if` η οποία ελέγχει το κείμενο τύπου `Text` με την ερώτηση που εμφανίζεται στον χρήστη. Με βάση την ερώτηση ανοίγει και το αντίστοιχο δισδιάστατο `Panel` που περιέχει πληροφορίες για την κάθε περιοχή αντίστοιχα. Σε περίπτωση σφάλματος έχει τοποθετηθεί η εντολή `Debug.Log` η οποία θα εμφανίσει το αντίστοιχο μήνυμα «Error» στο console της Unity (Εικόνα 11.7).

```

public void OnClickAboutInfoBt() {

    GameButtonsCanvas.SetActive(false);

    if(QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει η Μακεδονία;") {
        AboutInfoCanvas.SetActive(true);
    } else if (QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει όλο το Αιγαίο;") {
        AboutAigaioCanvas.SetActive(true);
    } else if (QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει η Αττική;") {
        AboutAttikiCanvas.SetActive(true);
    } else if (QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχουν συνολικά τα νησιά του Ιονίου;") {
        AboutIonioCanvas.SetActive(true);
    } else if (QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει τα μεγαλύτερα νησιά της Ελλάδας, η Κρήτη;") {
        AboutKritiCanvas.SetActive(true);
    } else if (QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει η Πελοπόννησος;") {
        AboutPeloponisosCanvas.SetActive(true);
    } else {
        Debug.Log("Error");
    }
}

```

Εικόνα 11.7 Μέθοδος OnClickAboutInfoBt()

Έπειτα υπάρχει η μέθοδος τύπου void, η OnClickNextBt(). Με το που πατήσει ο χρήστης το κουμπί (Button) για να κλείσει το διαδιάστατο Panel με τις πληροφορίες, ενεργοποιείται η μέθοδος. Η συγκεκριμένη μέθοδος όπως και η προηγούμενη που αναφέρθηκε παραπάνω, χρησιμοποιεί και αυτή μια συνθήκη if η οποία ελέγχει το κείμενο (Text) με την ερώτηση που εμφανίζεται στον χρήστη. Στην συνέχεια γίνεται αντίστοιχα, για την κάθε ερώτηση, η απενεργοποίηση του Panel με τις ερωτήσεις και η τοποθέτηση της απάντησης που έδωσε ο χρήστης πάνω στο κείμενο (Text) που περιέχει η κάθε πινακίδα πάνω στο επαυξημένο περιβάλλον (Εικόνα 11.8).

```

90 public void OnClickNextBt() {
91
92     if (QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει η Μακεδονία;") {
93
94         AboutInfoCanvas.SetActive(false);
95         GameButtonsCanvas.SetActive(true);
96         QuestionCanvas.SetActive(false);
97
98         AnswerPanel.SetActive(false);
99         AboutInfoButton.SetActive(false);
100         PersonMakedonia.SetActive(false);
101
102         MakedoniaText.text = "2.000.000";
103
104     } else if (QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει όλα τα Αιγαία;") {
105
106         AboutAigaioCanvas.SetActive(false);
107         GameButtonsCanvas.SetActive(true);
108         QuestionCanvas.SetActive(false);
109
110         AnswerPanel.SetActive(false);
111         AboutInfoButton.SetActive(false);
112         PersonAigaio.SetActive(false);
113
114         AigaioText.text = "500.000";
115
116     } else if (QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει η Αττική;") {
117
118         AboutAttikiCanvas.SetActive(false);
119         GameButtonsCanvas.SetActive(true);
120         QuestionCanvas.SetActive(false);
121
122         AnswerPanel.SetActive(false);
123         AboutInfoButton.SetActive(false);
124         PersonAttiki.SetActive(false);
125
126         AttikiText.text = "3.800.000";
127

```

Εικόνα 11.8 Απόσπασμα της μεθόδου OnClickNextBt()

Στην ίδια μέθοδο στο τέλος υπάρχει μια ακόμα συνθήκη if η οποία ελέγχει, αν οι διαστάτες φιγούρες των ανθρώπων είναι όλες απενεργοποιημένες. Σε περίπτωση που ισχύει η συνθήκη, γίνεται η ενεργοποίηση ενός κουμπιού (Button) με το οποίο ο χρήστης μπορεί με το πάτημα του να ολοκληρώσει το παιχνίδι και να γυρίσει στην αρχική σελίδα της εφαρμογής (Εικόνα 11.9).

```

165 if (!PersonAigaio.activeSelf && !PersonAttiki.activeSelf && !PersonIonio.activeSelf && !PersonKriti.activeSelf && !PersonMakedonia.activeSelf && !PersonPeloponisos.activeSelf) {
166
167     EndGameCanvas.SetActive(true);
168     Debug.Log("End Game");
169 }
170 }

```

Εικόνα 11.9 Έλεγχος απενεργοποίησης όλων των εικόνων

Όπως προαναφέρθηκε, ο χρήστης θα πρέπει με βάση την ερώτηση που του δίνεται να επιλέξει και να σύρει πάνω σε ένα πλαίσιο την απάντηση που πιστεύει πως είναι η σωστή. Για να επιτευχθεί αυτό, δημιουργήθηκε ένα script με το όνομα DragDropManager.cs. Το συγκεκριμένο script στην αρχή

αρχικοποιεί τις μεταβλητές που χρησιμοποιούνται. Αρχικά, οι πρώτες έξι μεταβλητές τύπου String περιέχουν τα Tags που έχουν τοποθετηθεί πάνω στην κάθε δισδιάστατη εικόνα, με τις οποίες ο χρήστης θα χρησιμοποιήσει για να δώσει την απάντηση του. Οι υπόλοιπες μεταβλητές που αρχικοποιούνται είναι τύπου GameObject και περιέχουν τα κουμπιά που εμφανίζονται ανά διαστήματα σε αυτή την σκηνή (Εικόνα 11.10).

```

15 public class DragDropManager : MonoBehaviour, IDragHandler, IEndDragHandler
16 {
17
18     [SerializeField] private string PersonAigaioTag = "PersonAigaioTag";
19     [SerializeField] private string PersonAttikiTag = "PersonAttikiTag";
20     [SerializeField] private string PersonIonioTag = "PersonIonioTag";
21     [SerializeField] private string PersonKritiTag = "PersonKritiTag";
22     [SerializeField] private string PersonMakedoniaTag = "PersonMakedoniaTag";
23     [SerializeField] private string PersonPeloponnisosTag = "PersonPeloponnisosTag";
24
25     public GameObject QuestionCanvas;
26     public Text QuestionText;
27     public GameObject AnswerPanel;
28
29     public GameObject PersonMakedonia; |
30     public GameObject PersonAigaio;
31     public GameObject PersonAttiki;
32     public GameObject PersonIonio;
33     public GameObject PersonKriti;
34     public GameObject PersonPeloponnisos;
35
36     public GameObject AboutInfoButton;
37

```

Εικόνα 11.10 Αρχικοποίηση μεταβλητών στο script DragDropManager.cs

Η πρώτη μέθοδος που ακολουθεί είναι η OnDrag() τύπου void, που επιτρέπει στον χρήστη να επιλέξει και να σύρει (Drag) μέσα στον χώρο τις εικόνες των απαντήσεων, επιτρέποντας επίσης και τον έλεγχο της θέσης στην οποία βρίσκεται η εικόνα (Εικόνα 11.11).

```

37
38     public void OnDrag(PointerEventData eventData) {
39         transform.position = Input.mousePosition;
40     }
41

```

Εικόνα 11.11 Μέθοδος OnDrag()

Στην συνέχεια ακολουθεί η μέθοδος OnEndDrag() η οποία χρησιμοποιεί μια συνθήκη if για να ελέγξει το Tag που επιλέχτηκε από τον χρήστη. Σε κάθε έλεγχο υπάρχει μία ακόμα συνθήκη για την κάθε ερώτηση που εμφανίζεται στον χρήστη. Με την βοήθεια της εντολής Vector3 μπορεί να γίνει ο έλεγχος της ακριβούς θέσης που έχει η εικόνα μέσα στον χώρο. Στην περίπτωση που η θέση της εικόνα είναι μέσα στα πλαίσια που υπάρχουν για να δώσει ο χρήστης την απάντηση, και υπάρχει η

προϋπόθεση πως είναι σωστή, η εικόνα παίρνει την ακριβή θέση στην οποία έχει τοποθετηθεί. Σε αντίθετη περίπτωση η εικόνα γυρνά στην αρχική της θέση επιτρέποντας στον χρήστη να δώσει μια διαφορετική απάντηση (Εικόνα 11.12).

```

42  public void OnEndDrag(PointerEventData eventData) {
43
44      if (gameObject.tag == "PersonAigaioTag") {
45          float DistanceAigaio = Vector3.Distance(PersonAigaio.transform.position, AnswerPanel.transform.position);
46          Debug.Log(DistanceAigaio);
47          if (DistanceAigaio < 50 && QuestionText.text == "Πόσα πληθυσμό πιστεύεις πως έχει όλα τα Αιγαία;") {
48              PersonAigaio.transform.position = AnswerPanel.transform.position;
49              AboutInfoButton.SetActive(true);
50          }
51          } else {
52              transform.localPosition = new Vector3(-218, 155, 0);
53          }
54      }
55
56      } else if (gameObject.tag == "PersonAttikiTag") {
57          float DistanceAttiki = Vector3.Distance(PersonAttiki.transform.position, AnswerPanel.transform.position);
58          if (DistanceAttiki < 50 && QuestionText.text == "Πόσα πληθυσμό πιστεύεις πως έχει η Αττική;") {
59              PersonAttiki.transform.position = AnswerPanel.transform.position;
60              AboutInfoButton.SetActive(true);
61          }
62          } else {
63              transform.localPosition = new Vector3(14, 155, 0);
64          }
65      }
66
67      } else if (gameObject.tag == "PersonIonioTag") {
68
69          float DistanceIonio = Vector3.Distance(PersonIonio.transform.position, AnswerPanel.transform.position);
70          if (DistanceIonio < 50 && QuestionText.text == "Πόσα πληθυσμό πιστεύεις πως έχουν συνολικά τα νησιά του Ιονίου;") {
71              PersonIonio.transform.position = AnswerPanel.transform.position;
72              AboutInfoButton.SetActive(true);
73          }
74          } else {
75              transform.localPosition = new Vector3(246, 156, 0);
76          }
77      }
78  }

```

Εικόνα 11.12 Απόσπασμα της μεθόδου OnEndDrag()

Για την επιλογή της κάθε πινακίδας που υπάρχει πάνω στον επαυξημένο περιβάλλον έχουν δημιουργηθεί και οι αντίστοιχες μέθοδοι, οι οποίες ενεργοποιούνται όταν ο χρήστης πατήσει πάνω σε μία από αυτές. Οι πινακίδες λειτουργούν σαν κουμπιά (Button) καθώς έχει τοποθετηθεί σε αυτές ένα Event που επιτρέπει να εκτελούν διάφορες λειτουργίες με το πάτημα τους (Εικόνα 11.13).

```

122 public void OnClickMakedoniaPanel() {
123
124     AnswerPanel.SetActive(true);
125     QuestionCanvas.SetActive(true);
126     QuestionText.text = "Πόσα πληθυσμό πιστεύεις πως έχει η Μακεδονία;";
127
128 }
129 public void OnClickAigaioPanel() {
130
131     AnswerPanel.SetActive(true);
132     QuestionCanvas.SetActive(true);
133     QuestionText.text = "Πόσα πληθυσμό πιστεύεις πως έχει όλα τα Αιγαία;";
134
135 }
136
137 public void OnClickAttikiPanel() {
138
139     AnswerPanel.SetActive(true);
140     QuestionCanvas.SetActive(true);
141     QuestionText.text = "Πόσα πληθυσμό πιστεύεις πως έχει η Αττική;";
142
143 }
144 public void OnClickIonioPanel() {
145
146     AnswerPanel.SetActive(true);
147     QuestionCanvas.SetActive(true);
148     QuestionText.text = "Πόσα πληθυσμό πιστεύεις πως έχουν συνολικά τα νησιά του Ιονίου;";
149
150 }
151 public void OnClickKritiPanel() {
152
153     AnswerPanel.SetActive(true);
154     QuestionCanvas.SetActive(true);
155     QuestionText.text = "Πόσα πληθυσμό πιστεύεις πως έχει τα μεγαλύτερα νησιά της Ελλάδας, η Κρήτη;";
156
157 }
158 public void OnClickPeloponisosPanel() {
159
160     AnswerPanel.SetActive(true);
161     QuestionCanvas.SetActive(true);
162     QuestionText.text = "Πόσα πληθυσμό πιστεύεις πως έχει η Πελοπόννησος;";
163
164 }
165

```

Εικόνα 11.13 Μέθοδοι για το πάτημα της κάθε πινακίδας

Με την ολοκλήρωση του παιχνιδιού, αφού ο χρήστης έχει δώσει την κάθε απάντηση σωστά για την κάθε πινακίδα που υπάρχει γίνεται η εμφάνιση του τελικού Panel, όπου το παιχνίδι καθοδηγεί τον χρήστη να εξέλθει από αυτό και να επιστρέψει στην αρχική σελίδα της εφαρμογής ολοκληρώνοντας με αυτό τον τρόπο το συγκεκριμένο παιχνίδι με επιτυχία.

11.3 Επίλογος

Το συγκεκριμένο παιχνίδι καλύπτει την ύλη του 30^{ου} κεφαλαίου με τίτλο «Η γεωγραφική κατανομή του πληθυσμού της Ελλάδας» στις σελίδες 100-102 του σχολικού βιβλίου της Ε΄ δημοτικού. Το παιχνίδι αυτό έχει δημιουργηθεί με την βοήθεια της τεχνολογίας της επαυξημένης πραγματικότητας και στόχος του είναι να προσφέρει μια βασική ιδέα για τον λόγο και τον τρόπο με τον οποίο γίνεται η κατανομή του πληθυσμού. Ο χρήστης επιλέγει μια πινακίδα που επιθυμεί, η οποία αντιπροσωπεύει μια περιοχή της Ελλάδας, και με βάση αυτή δίνει την απάντηση που πιστεύει πως είναι σωστή. Σε περίπτωση σωστής απάντηση εμφανίζονται διάφορες πληροφορίες. Για την ολοκλήρωση του παιχνιδιού θα πρέπει να δώσει τις σωστές απαντήσεις για όλες τις πινακίδες που υπάρχουν πάνω στον χάρτη.

Κεφάλαιο 12ο: Συμπεράσματα και προτάσεις βελτίωσης

Η επαυξημένη πραγματικότητα αποτελεί μια τεχνολογία που εξελίσσεται σε πολλούς επαγγελματικούς τομείς. Στην εκπαίδευση, έχει την δυνατότητα να αλλάξει ριζικά τον τρόπο με τον οποίο πραγματοποιούνται τα μαθήματα μέσα στις τάξεις των σχολείων. Οι μαθητές πλέον έχουν την ευχέρεια και τις βάσεις να χειρίζονται ένα κινητό ή μια υπολογιστική συσκευή στην καθημερινή τους ζωή. Επιπλέον τα παιχνίδια ανέκαθεν πρόσφεραν χαρά και ενθουσιασμό στα παιδικά χρόνια, οπότε τα εκπαιδευτικά παιχνίδια μπορούν να βελτιώσουν την διεξαγωγή των μαθημάτων. Για τους παραπάνω λόγους, η προσθήκη εκπαιδευτικών παιχνιδιών με την χρήση της επαυξημένης πραγματικότητας στα σχολεία, θα αποτελούσε μια σημαντική αναβάθμιση της εκπαίδευσης.

Συγκεκριμένα, για την εφαρμογή που βασίζεται η παρούσα διπλωματική, για το μάθημα της Γεωγραφίας της Ε' δημοτικού, υπάρχουν αρκετά θετικά χαρακτηριστικά που μπορεί κάποιος να παρατηρήσει. Αρχικά, η συγκεκριμένη εφαρμογή αποτελείται από επτά εκπαιδευτικά παιχνίδια με την χρήση επαυξημένης πραγματικότητας, που καλύπτουν το μεγαλύτερο μέρος της ύλης. Η Γεωγραφία ανέκαθεν αποτελούσε ένα πολύ σημαντικό μάθημα για την εξέλιξη των γνώσεων ενός μαθητή. Παρόλα αυτά αποτελεί και ένα μάθημα που μαθαίνεται «παπαγαλία» από τους μαθητές, καθώς η ύλη του βιβλίου είναι δομημένη με τέτοιο τρόπο που δεν αφήνει πολλά περιθώρια στον διδάσκοντα. Για αυτό τον λόγο η εφαρμογή GeographyAR έρχεται να δώσει μια λύση σε αυτό το πρόβλημα, καθώς ο μαθητής έχει την δυνατότητα να εξερευνήσει την εφαρμογή και τα παιχνίδια, μαθαίνοντας πράγματα, παρατηρώντας εικόνες και καταστάσεις μέσω αυτών. Επιπλέον ο μαθητής έχει την δυνατότητα να χρησιμοποιήσει την εφαρμογή και εκτός σχολείου πράγμα που βοηθάει στην καλύτερη μελέτη καθώς πολλές πληροφορίες δεν γίνονται αντιληπτές με την πρώτη ανάγνωση.

Θα ήταν καλό να αναφερθεί και το γεγονός πως η παρούσα εφαρμογή θα μπορούσε να βελτιωθεί σε διάφορα θέματα. Πρώτα από όλα, η προσθήκη επιπλέον παιχνιδιών στην εφαρμογή θα βοηθούσε πολύ, στην καλύτερη κάλυψη της ύλης του σχολικού βιβλίου, και στην ποικιλία των ειδών παιχνιδιού για τους μαθητές. Επιπλέον τα γραφικά στα ήδη υπάρχοντα παιχνίδια, μπορούν να βελτιωθούν και να προστεθούν περισσότερες κινούμενες εικόνες, κάνοντας τα πιο «ζωντανά» για τον χρήστη. Η χρήση περισσότερων ηχητικών εφέ στα αντικείμενα των παιχνιδιών αποτελεί ένα σημαντικό κομμάτι για την βελτίωση της εφαρμογής. Θα ήταν ιδανική η προσθήκη μιας τρισδιάστατης φιγούρας, που θα αποτελούσε τον «οδηγό» του παιχνιδιού, καθώς θα παρουσίαζε και θα εξηγούσε με πιο κατανοητό τρόπο τα παιχνίδια. Τέλος, θα ήταν χρήσιμο η εφαρμογή να είναι συμβατή και σε συσκευές με λογισμικό iOS, για χρήστες που έχουν στην κατοχή τους κινητά και tablets της Apple.

8. ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία – Επιστημονικά Άρθρα

- [1] Κωστής Κουτσόπουλος, Μαρία Σωτηράκου, Μαρία Ταστσόγλου. Βιβλίο Γεωγραφίας Ε' Δημοτικού «Μαθαίνω την Ελλάδα».
- [2] Kangdon Lee (2012). Augmented Reality in Education and Training, University of Northern Colorado.
- [3] Janne Paavilainen, Hannu Korhonen, Kati Alha, Jaakko Stenros, Elina Koskinen, Frans Mayra (2017). The Pokemon GO Experience: A Location-Based Augmented Reality Mobile Game Goes Mainstream.
- [4] John Vince (2004). Introduction to Virtual Reality, 1-4.
- [5] Tomasz Mazuryk and Michael Gervautz. Virtual Reality: History, Applications, Technology and Future. Vienna University of Technology, Austria.
- [6] Chuah, Stephanie Hui-Wen (2018). "Why and Who Will Adopt Extended Reality Technology? Literature Review, Synthesis, and Future Research Agenda".
- [7] Stapleton, C.; Hughes, C.; Moshell, M.; Micikevicius, P.; Altman, M. (December 2002). "Applying mixed reality to entertainment", 122-124.
- [8] Hui Ye, Kin Chung Kwan, Wanchao Su, Hongbo F (2020). ARAnimator: In-situ Character Animation in Mobile AR with User-defined Motion Gestures. University of Hong Kong.
- [9] Gaurav Bharaj, Thorsten Thormählen, Hans-Peter Seidel, Christian Theobalt (2012). Automatically Rigging Multi-component Characters, MPI Informatik Germany.
- [10] Ilya Baran Jovan Popovic (2007). Automatic Rigging and Animation of 3D Characters. Computer Science and Artificial Intelligence Laboratory Massachusetts Institute of Technology.
- [11] Steve Chi-Yin Yuen, Gallayanee Yaoyuneyong, Erik Johnson (2011). Augmented Reality: An Overview and Five Directions for AR in Education.
- [12] Hsion-Kai Wu, Silvia Wen-Yu Lee, Hsin-Yi Chang, Jyg-Chong Liang (2012). Current status, opportunities and challenges of augmented reality in education.

IIAPAPTHMA A : DragDropManager.cs

```
using System.Collections;

using UnityEngine;

using UnityEngine.UI;

using System.Collections.Generic;

using Vuforia;

using UnityEngine.SceneManagement;

using UnityEngine.EventSystems;


public class DragDropManager : MonoBehaviour, IDragHandler, IEndDragHandler
{
    [SerializeField] private string PersonAigaioTag = "PersonAigaioTag";

    [SerializeField] private string PersonAttikiTag = "PersonAttikiTag";

    [SerializeField] private string PersonIonioTag = "PersonIonioTag";

    [SerializeField] private string PersonKritiTag = "PersonKritiTag";

    [SerializeField] private string PersonMakedoniaTag = "PersonMakedoniaTag";

    [SerializeField] private string PersonPeloponnisosTag = "PersonPeloponnisosTag";


    public GameObject QuestionCanvas; // Canvas me text pou kanei tin erotisi

    public Text QuestionText; // To text tou QuestionCanvas gia na dinei kai sixnaritiria

    public GameObject AnswerPanel; // to panel pou topothetis ton plithismo


    public GameObject PersonMakedonia; // to icon me ton plithismo tis Makedonias

    public GameObject PersonAigaio;

    public GameObject PersonAttiki;

    public GameObject PersonIonio;
```

```

public void OnEndDrag(PointerEventData eventData) {

    if (gameObject.tag == "PersonAigaioTag") {

        float DistanceAigaio = Vector3.Distance(PersonAigaio.transform.position,
        AnswerPanel.transform.position);

        Debug.Log(DistanceAigaio);

        if(DistanceAigaio < 50 && QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει όλο το
        Αιγαίο;") {

            PersonAigaio.transform.position = AnswerPanel.transform.position;

            AboutInfoButton.SetActive(true);

        } else {

            transform.localPosition = new Vector3(-218, 155, 0);

        }

    } else if (gameObject.tag == "PersonAttikiTag") {

        float DistanceAttiki = Vector3.Distance(PersonAttiki.transform.position,
        AnswerPanel.transform.position);

        if(DistanceAttiki < 50 && QuestionText.text == "Πόσο πληθυσμό πιστεύεις πως έχει η Αττική;") {

            PersonAttiki.transform.position = AnswerPanel.transform.position;

            AboutInfoButton.SetActive(true);

        } else {

            transform.localPosition = new Vector3(14, 155, 0);

        }

    }
}

```

IIAPAPTHMA B : **APIWeather.cs**

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.Networking;

using UnityEngine.SceneManagement;

using SimpleJSON;

using UnityEngine.Android;


public class APIWeather : MonoBehaviour

{

    public TextMeshProUGUI statusText;

    public TextMeshProUGUI location;

    public TextMeshProUGUI country;

    public TextMeshProUGUI description;

    public TextMeshProUGUI temp;

    public TextMeshProUGUI temp_min;

    public TextMeshProUGUI temp_max;

    public TextMeshProUGUI pressure;

    public TextMeshProUGUI humidity;

    public TextMeshProUGUI windspeed;


    private LocationInfo lastLocation;

    public GameObject WeatherInfoPanel1;

    public GameObject WeatherInfoPanel2;
```

```

void Start() {

    UpdateWeatherData();

}

private void UpdateWeatherData() {

    StartCoroutine(FetchWeatherDataFromApi(lastLocation.latitude.ToString(),
lastLocation.longitude.ToString()));

}

private IEnumerator FetchWeatherDataFromApi(string latitude, string longitude)
{

    string url =
"http://api.openweathermap.org/data/2.5/weather?lat=40.6401&lon=22.9444&appid=7574fd289a4b3
082492bc56d90b58133&units=metric&lang=el";

    UnityWebRequest fetchWeatherRequest = UnityWebRequest.Get(url);

    yield return fetchWeatherRequest.SendWebRequest();

    if (fetchWeatherRequest.isNetworkError || fetchWeatherRequest.isHttpError)
    {

        statusText.text = fetchWeatherRequest.error;

    } else {

        var response = JSON.Parse(fetchWeatherRequest.downloadHandler.text);

        location.text = response["name"];

        country.text = response["country"];

        description.text = response["weather"][0]["description"];

        temp.text = response["main"]["temp"] + " C";

```

```

temp_min.text = "Μικρότερη θερμοκρασία " + response["main"]["temp_min"] + " C";

temp_max.text = "Μέγιστη θερμοκρασία " + response["main"]["temp_max"] + " C";

pressure.text = "Η πίεση είναι " + response["main"]["pressure"] + " Pa";

humidity.text = response["main"]["humidity"] + " % υγρασία";

windspeed.text = "Ταχύτητα του αέρα " + response["wind"]["speed"] + " Km/h";

}

}

```

```

public void OnClickWeatherButton() {

```

```

    WeatherInfoPanel1.SetActive(true);

    WeatherInfoPanel2.SetActive(true);

    WeatherButton.SetActive(false);

    EndGameButton.SetActive(true);

}

```

```

public void EndGame() {

```

```

    SceneManager.LoadScene("SelectLevel");

}

}

```

ΠΑΡΑΡΤΗΜΑ C : **ObjectRotation.cs**

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SceneManagement;


public class ObjectRotation : MonoBehaviour

{

    public GameObject objectRotate;

    public float rotatespeed = 50f;

    bool rotateStatus = false;


    public void RotateObject() {

        if (rotateStatus == false) {

            rotateStatus = true;

        }else {

            rotateStatus = false;

        }

    }

    void Update()

    {

        if (rotateStatus == true) {

            objectRotate.transform.Rotate(Vector3.forward, rotatespeed * Time.deltaTime);

        }

    }

}
```

IIAPAPTHMA D : QuizManager.cs

```
using System.Collections;

using UnityEngine;

using UnityEngine.UI;

using System.Collections.Generic;

using Vuforia;

using UnityEngine.Events;

using UnityEngine.SceneManagement;

using Random=UnityEngine.Random;

public class quizManager : MonoBehaviour

{

    public QnA[] questions;

    private static List<QnA> unansweredQuestions;

    private QnA currentQuestion;

    [SerializeField] private string GeomorfologikosTag = "GeomorfologikosTag";

    [SerializeField] private string PolitikosTag = "PolitikosTag";

    [SerializeField] private string ProiontonTag = "ProiontonTag";

    public GameObject BoardText;

    private Text onBoardText;

    private Text text;

    public int sostesApantisis, lathosApantisis;
```

[SerializeField]

private Text QuestionText;

[SerializeField]

private float timeBetweenQuestions = 2f;

void Start ()

onBoardText = BoardText.GetComponent<Text>(); //pairnei tin timi pou exei to text tou pinaka

onBoardText.text = "Ηρθε η ώρα να ξεκινήσουμε το πρώτο μας παιχνίδι";

Invoke("Update",8.0f);

if (unansweredQuestions == null || unansweredQuestions.Count == 0){

unansweredQuestions = questions.ToList<QnA>();

}

Invoke("SetRandomQuestion", 8.0f);

int totalQuestions = 15;

}

void Update()

{

if (Input.GetMouseButtonDown(0))

{

Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);

RaycastHit hit;

```

if (Physics.Raycast(ray, out hit))
{
    var selection = hit.transform;

    if (selection.CompareTag(GeomorfologikosTag)) {

        if(onBoardText.text == "Ποιος από τους χάρτες είναι ο Γεωμορφολογικός;"){

            sotesApantisis++;

            StartCoroutine(TransitionToNextQuestion());

        } else if(onBoardText.text == "Θέλω να δω ποια περιοχή είναι πεδινή και ποια είναι
ορεινή. Ποιο χάρτη θα κοιτάξω;") {

            sotesApantisis++;

            StartCoroutine(TransitionToNextQuestion());

        } else {

            lathosApantisis++;

            StartCoroutine(TransitionToNextQuestion());

        }

    }

    else if (selection.CompareTag(PolitikosTag)) {

        if(onBoardText.text == "Ποιον χάρτη μπορούμε να χρησιμοποιήσουμε για να δούμε τους
νομούς της Ελλάδας;"){

            sotesApantisis++;

            StartCoroutine(TransitionToNextQuestion());

        } else if(onBoardText.text == "Ποιος χάρτης είναι ο Πολιτικός;"){

            sotesApantisis++;

            StartCoroutine(TransitionToNextQuestion());

        } else if(onBoardText.text == "Μένω στην Θεσσαλονίκη αλλά δεν ξέρω που ακριβώς είναι τα
Ιωάννινα για να πάω το σαββατοκύριακο στο φίλο μου. Ποιον χάρτη θα πρέπει να συμβουλευτώ;"){

            sotesApantisis++;

```

```
void SetRandomQuestion () {
```

```
    int randomQuestionIndex = Random.Range (0, unansweredQuestions.Count);
```

```
    currentQuestion = unansweredQuestions[randomQuestionIndex];
```

```
    QuestionText.text = currentQuestion.Question;
```

```
}
```

```
IEnumerator TransitionToNextQuestion (){
```

```
    unansweredQuestions.Remove(currentQuestion);
```

```
    yield return new WaitForSeconds(timeBetweenQuestions);
```

```
    if(unansweredQuestions.Count <= 0 ){
```

```
        onBoardText.text = "Τέλος Παιχνιδιού!!";
```

```
        Invoke("EndGame",4.0f);
```

```
    }
```

```
    SetRandomQuestion();
```

```
}
```

```
public void EndGame () {
```

```
    PlayerPrefs.SetInt("sostesApantisis", sostesApantisis); //pername tin timi ton soston apantisewn
```

```
    PlayerPrefs.SetInt("lathosApantisis", lathosApantisis);
```

```
    SceneManager.LoadScene("ScoreGame1");
```

```
}
```

ΠΑΡΑΡΤΗΜΑ Ε : **LevelSelector.cs**

```
using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;


public class LevelSelector : MonoBehaviour

{

    public void startGame1() {

        SceneManager.LoadScene("IntroGame1")

    }

    public void startGame2() {

        SceneManager.LoadScene("IntroGame2");

    }


    public void startGame3() {

        SceneManager.LoadScene("IntroThesiGame");

    }


    public void startGame4() {

        SceneManager.LoadScene("NisiaGame");

    }


    public void startGame5() {

        SceneManager.LoadScene("MountainScene");

    }

}
```

```
public void startGame6() {  
    SceneManager.LoadScene("ClimateGame");  
}  
  
public void startGame7() {  
    SceneManager.LoadScene("PopulationGame");  
}  
}
```