

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Εκπαιδευτική εφαρμογή αφήγησης με χρήση
επαυξημένης πραγματικότητας για το μάθημα της
Ιστορίας της Δ' Δημοτικού»



Του φοιτητή
Καμπάνη Παναγιώτη
Αρ. Μητρώου: AM164670

Επιβλέπων
Ευκλείδης Κεραμόπουλος
Καθηγητής

Ημερομηνία 25-01-2025

Τίτλος Δ.Ε. «Εκπαιδευτική εφαρμογή αφήγησης με χρήση επαυξημένης πραγματικότητας για το μάθημα της Ιστορίας της Δ' Δημοτικού»

Κωδικός Δ.Ε. 23180

Ονοματεπώνυμο φοιτητή Καμπάνης Παναγιώτης

Ονοματεπώνυμο εισηγητή Κεραμόπουλος Ευκλείδης

Ημερομηνία ανάληψης Δ.Ε. 29-03-2023

Ημερομηνία περάτωσης Δ.Ε. 25-01-2025

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 Καμπάνη Παναγιώτη 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 απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

« Αφιερώνω την διατριβή μου στην σύντροφο μου Αιμιλιανή Χριστίδου, που με ενέπνευσε και με στήριξε στις πιο δύσκολες στιγμές υλοποίησης της παρούσας πτυχιακής εργασίας. Η υπομονή και η ενθάρρυνσή της με βοήθησαν να φτάσω στο σημείο που βρίσκομαι σήμερα»

Πρόλογος

Η επιλογή της συγκεκριμένης πτυχιακής εργασίας με θέμα την "Εκπαιδευτική εφαρμογή αφήγησης με χρήση επαυξημένης πραγματικότητας για το μάθημα της Ιστορίας της Δ' Δημοτικού" προέκυψε από την ανάγκη για έναν πιο διαδραστικό και ελκυστικό τρόπο διδασκαλίας. Πάντα πίστευα ότι η Ιστορία είναι κάτι περισσότερο από ημερομηνίες και γεγονότα, είναι ιστορίες που πρέπει να ζωντανεύουν, αλλά δυστυχώς στην εποχή μου, όταν δηλαδή πήγαινα εγώ Τετάρτη δημοτικού, δεν υπήρχε τόσο εύκολα αυτή η δυνατότητα. Πλέον μέσω της επαυξημένης πραγματικότητας (AR), μπορούμε να αναβιώσουμε σημαντικά ιστορικά γεγονότα και μάχες του παρελθόντος, μετατρέποντας τη διαδικασία της μάθησης σε μια βιωματική εμπειρία.

Αυτό το θέμα συνδύασε την αγάπη μου για την εκπαιδευτική τεχνολογία με τη δημιουργικότητα και την καινοτομία, ενώ ταυτόχρονα απαντά σε ένα πραγματικό ζήτημα. Τη δυσκολία που συχνά έχουν οι μαθητές να συνδεθούν με την Ιστορία.

Κατά τη διάρκεια της υλοποίησης της παρούσας πτυχιακής εργασίας, απέκτησα μια μικρή εμπειρία στην ανάπτυξη εφαρμογών με τεχνολογίες AR, όπως επίσης και στη προσαρμογή τους στις ανάγκες των μαθητών.

Ωστόσο, ο μεγαλύτερος στόχος αποτελεί τη σκέψη ότι η εργασία μου μπορεί να προσφέρει μια διασκεδαστική προσέγγιση στη μάθηση, ενδεχομένως εμπνέοντας περισσότερους μαθητές να αγαπήσουν την Ιστορία αλλά και την επιστήμη της πληροφορικής.

Περίληψη

Η πτυχιακή εργασία οδήγησε στη δημιουργία μιας ολοκληρωμένης εκπαιδευτικής εφαρμογής, που προσφέρει μια διαδραστική εμπειρία μάθησης για το μάθημα της Ιστορίας Δ' Δημοτικού. Η εφαρμογή, η οποία υλοποιήθηκε μέσω της πλατφόρμας Unity, της Vuforia Engine και του προγράμματος σχεδίασης 3D γραφικών Blender αξιοποιεί την τεχνολογία της AR για την αναβίωση ιστορικών γεγονότων και την αφήγηση σημαντικών μαχών για την ιστορία.

Η εφαρμογή εμπεριέχει τρεις βασικές μάχες από το σχολικό βιβλίο, οι οποίες επελέγησαν με βάση τη σημασία τους αλλά και την καταλληλότητά τους για παρουσίαση μέσω AR. Μέσω της εφαρμογής, οι μαθητές μπορούν να αλληλοεπιδράσουν με τους στόλους ή τα τάγματα Ελλήνων και Περσών, να ακούσουν αφήγηση της μάχης αλλά και να συμμετέχουν σε κουίζ γνώσεων και άλλες διαδράσεις που ενισχύουν την κατανόηση του μαθήματος.

Ως αποτέλεσμα από την χρήση της εφαρμογής είναι οι μαθητές να βρουν τη διαδικασία της μάθησης πιο ενδιαφέρουσα και διασκεδαστική, και με αυτο τον τρόπο να κατανοήσουν καλύτερα τα ιστορικά γεγονότα.

«Creation of an augmented reality educational app for the History course of the 4th year of elementary school»

«Panagiotis Kampanis»

Abstract

The thesis led to the creation of a comprehensive educational application, which offers an interactive learning experience for the History lesson of 4th grade. The game, which was implemented through the Unity platform, the Vuforia Engine and the 3D graphics design program Blender, utilizes the technology of the Greek Academy of Sciences to revive historical events and narrate important battles for history. The application contains three key battles from the textbook, which were selected based on their importance and their suitability for presentation through AR.

Through the application, students can interact with the fleets or battalions of Greeks and Persians, listen to the narration of the battle and participate in knowledge quizzes and other interactions that enhance understanding of the lesson.

As a result of using the application, the students should find the learning process more interesting and fun, and in this way better understand historical events.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή Ευκλείδη Κεραμόπουλο, για την καθοδήγηση του, την υπομονή και την υποστήριξη που μου πρόσφερε καθ' όλη τη διάρκεια εκπόνησης της παρούσας πτυχιακής εργασίας αλλά και επίσης για την ευκαιρία που μου έδωσε να συνεργαστούμε. Ευχαριστώ την οικογένειά μου για την αγάπη, την υποστήριξη και την πίστη στις ικανότητές μου, ακόμα και τις στιγμές που δεν πίστευα σε αυτές. Τους ευχαριστώ που πάντα με στήριζαν και μου έδιναν τη δύναμη και το χρόνο να συνεχίσω.

Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xii
Συντομογραφίες.....	xiv
Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Στόχος.....	1
1.2 Δομή	1
1.3 Επίλογος.....	2
Κεφάλαιο 2ο: Το Συνεχές Εικονικότητας-Πραγματικότητας.....	3
2.1 Εισαγωγή.....	3
2.2 Επαυξημένη Πραγματικότητα (AR).....	4
2.2.1 Κατανόηση της Επαυξημένης Πραγματικότητας.....	4
2.3 Εικονική Πραγματικότητα (VR)	5
2.4 Μικτή πραγματικότητα	6
2.5 Εκτεταμένη Πραγματικότητα (XR).....	6
2.6 Εφαρμογές της Εκτεταμένης Πραγματικότητας	6
2.6.1 Αγορά	7
2.6.2 Υγεία.....	8
2.6.3 Εκπαίδευση.....	10
2.6.4 Ψυχαγωγία	11
2.7 Επίλογος.....	11
Κεφάλαιο 3ο: Πλατφόρμες & εργαλεία υλοποίησης.....	13
3.1 Εισαγωγή.....	13
3.2 Πλατφόρμα & εργαλεία Unity	13
3.2.1 TextMeshPro	13
3.2.2 Terrain Sample Asset Pack.....	14
3.2.3 Dreamteck Splines.....	15
3.2.4 Animation & Timeline Director	15
3.2.5 Colliders & Triggers.....	16

3.2.6	AudioSource	16
3.2.7	Render Pipelines & Shader Graph.....	16
3.2.8	Αναβάθμιση των Materials σε URP.....	17
3.3	Vuforia Engine	18
3.3.1	Εισαγωγή της Vuforia Engine στο Unity	18
3.3.2	Vuforia Image Target Database.....	19
3.3.3	Image Target στο Unity.....	20
3.4	Blender & FBX	21
3.5	Επίλογος.....	21
Κεφάλαιο 4ο:	Υλοποίηση της εφαρμογής	23
4.1	Εισαγωγή.....	23
4.2	Βιβλίο Μαθητή & Τετράδιο εργασιών.....	23
4.3	Εικονίδιο Εφαρμογής	23
4.4	Κοινές Διεπαφές χρήστη	24
4.4.1	Βασικά κουμπιά.....	24
4.4.2	WrongAnswerImage.....	25
4.4.3	InBattle	27
4.4.4	Win	28
4.5	Κοινά scripts.....	29
4.5.1	QuestionController scripts.....	29
4.5.2	SceneTransitions script.....	30
4.6	Audio Source.....	31
4.7	Δημιουργία 3D μοντέλων μέσω Blender.....	31
4.7.1	3D Στρατιώτες.....	31
4.7.2	3D Πλοία	32
4.7.3	Υφές & Εξαγωγή 3D μοντέλων.....	34
4.8	Αρχικό Μενού (StartScene)	34
4.8.1	WelcomeUI.....	34
4.8.2	MainMenuUI	35
4.8.3	InstructionsUI.....	36
4.9	Μάχη του Μαραθώνα.....	36
4.9.1	Δομή της άσκησης.....	37
4.9.2	Επίλογος της άσκησης «Μάχη του Μαραθώνα».....	45
4.10	Μάχη των Θερμοπυλών	45
4.10.1	Δομή της άσκησης «Μάχη των Θερμοπυλών»	45

4.10.2	Επίλογος της άσκησης «Μάχη των Θερμοπυλών»	52
4.11	Ναυμαχία της Σαλαμίνας	52
4.11.1	Δομή της άσκησης.....	53
4.11.2	Επίλογος της άσκησης «Ναυμαχία της Σαλαμίνας»	62
Κεφάλαιο 5ο:	Συμπεράσματα & Μελλοντικές βελτιώσεις.....	63
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	64
	ΠΑΡΑΡΤΗΜΑ Α : QuestionHandlerStart.cs C# Script.....	68
	ΠΑΡΑΡΤΗΜΑ Β : InBattle.cs C# Script.....	69
	ΠΑΡΑΡΤΗΜΑ C : SceneTransitions.cs C# Script	70
	ΠΑΡΑΡΤΗΜΑ D : QuestionHandler.cs C# Script	71
	ΠΑΡΑΡΤΗΜΑ E : QuestionHandlerThermopylae.cs C# Script.....	80
	ΠΑΡΑΡΤΗΜΑ F : SelectionOfPersians.cs C# Script.....	95
	ΠΑΡΑΡΤΗΜΑ G : SelectionOfGreeks.cs C# Script	97
	ΠΑΡΑΡΤΗΜΑ H : DragUIelement.cs C# Script.....	98
	ΠΑΡΑΡΤΗΜΑ I : WrongAnimationSalamina.cs C# Script	102
	ΠΑΡΑΡΤΗΜΑ J : QuestionHandlerSalamina.cs C# Script.....	103
	ΠΑΡΑΡΤΗΜΑ K : SplineAnimationController.cs C# Script.....	111

Κατάλογος Σχημάτων

Σχήμα 2.1: Το Συνεχές Εικονικότητας-Πραγματικότητας.....	3
Σχήμα 2.2: Επαυξημένη Πραγματικότητα (AR)	4
Σχήμα 2.3: Πως λειτουργεί η Ε.Π.	5
Σχήμα 2.4: Εικονική Πραγματικότητα(VR).....	5
Σχήμα 2.5: Microsoft Hololens	6
Σχήμα 2.6: PokemonGO.....	6
Σχήμα 2.7: Virtual Fitting Room της εταιρίας Lily.....	7
Σχήμα 2.8: Εφαρμογή AR της Hyundai	8
Σχήμα 2.9: Επέμβαση σπονδυλικής στήλης με AR στο πανεπιστήμιο John Hopkins	9
Σχήμα 2.10: Ολόγραμμα AR στα φάρμακα της Bayer Pharmaceuticals	10
Σχήμα 2.11: Google Expeditions στο Έβερεστ	10
Σχήμα 3.1: TextMeshPro στα Ελληνικά στρατεύματα	14
Σχήμα 3.2: Στενό των Θερμοπυλών	14
Σχήμα 3.3: Animation και Timeline Director	15
Σχήμα 3.4: Πριν την αναβάθμιση Material σε URP.....	17
Σχήμα 3.5: Αναβάθμιση των Material σε URP	17
Σχήμα 3.6: Vuforia Engine AR Package.....	18
Σχήμα 3.7: Vuforia Configuration.....	19
Σχήμα 3.8: Δημιουργία βάσης με Image Targets	20
Σχήμα 3.9: Ρυθμίσεις Image Target Behaviour (script)	20
Σχήμα 3.10: 3d μοντέλο Πέρση στρατιώτη στο Blender	21
Σχήμα 4.1: Εικονίδιο Εφαρμογής.....	23
Σχήμα 4.2: Παράδειγμα UI εφαρμογής.....	24
Σχήμα 4.3: Βασικά κουμπιά των UI.....	25
Σχήμα 4.4: WrongAnswerImageUI.....	25
Σχήμα 4.5: Animator controller WrongAnswerImage	26
Σχήμα 4.6: Μέθοδος χρήσης του UI WrongAnswerImage	26
Σχήμα 4.7: Script WrongAnswerImage	27
Σχήμα 4.8: InBattleUI animation	27
Σχήμα 4.9: InBattleUI inspector.....	28
Σχήμα 4.10: WinUI	28
Σχήμα 4.11: Τοποθετημένο script μέσα σε Empty GameObject.....	29
Σχήμα 4.12: Ενσωμάτωση script SceneTransitions στο onClick() κουμπιού	30
Σχήμα 4.13: Script αλλαγής σκηνών SceneTransitions	31
Σχήμα 4.14: Δημιουργία Έλληνα στρατιώτη στο Blender	32
Σχήμα 4.15: Κατάρτι της βάρκας	32
Σχήμα 4.16: Πανί της βάρκας	33
Σχήμα 4.17: Πλώρη και η πρύμνη του πλοίου	33
Σχήμα 4.18: Τελική όψη του πλοίου	34
Σχήμα 4.19: WelcomeUI.....	35
Σχήμα 4.20: InstructionsUI	36
Σχήμα 4.21: Άσκηση μάχης του Μαραθώνα (σελ.23 Τετράδιο εργασιών μαθητή)	37
Σχήμα 4.22: ArxiAsk1	37
Σχήμα 4.23: Hierarchy της μάχης του Μαραθώνα.....	38

Σχήμα 4.24: Μέθοδος onButtonClickArxi()	38
Σχήμα 4.25: Κουμπιά Question1 GameObject.....	38
Σχήμα 4.26: μέθοδος CorrectBattleground.....	39
Σχήμα 4.27: Τοποθέτηση ταγμάτων Μάχης του Μαραθώνα & αόρατα Cubes	39
Σχήμα 4.28: Εργαλείο Signal Receiver με τους κατάλληλους Signal Emmiters	40
Σχήμα 4.29: Ερώτηση Phase1_2	41
Σχήμα 4.30: Επιλογή κουμπιού λάθος απάντησης.....	41
Σχήμα 4.31: Μέθοδοι που καλούνται σε περίπτωση λάθους από QuestionHandler	42
Σχήμα 4.32: Μέθοδοι που καλούνται όταν ο χρήστης απαντήσει σωστά.....	42
Σχήμα 4.33: Phase1_3Question.....	43
Σχήμα 4.34: Phase2Question.....	44
Σχήμα 4.35: Τέλος πρώτης άσκησης.....	44
Σχήμα 4.36: Image Target άσκησης της Μάχης των Θερμοπυλών.....	45
Σχήμα 4.37: SelectionOfTroopsUI.....	46
Σχήμα 4.38: DragUIElement script	46
Σχήμα 4.39: Μέθοδοι onBeginDrag() και OnDrag()	47
Σχήμα 4.40: Μέθοδοι onEndDrag() και instantiate του prefab	47
Σχήμα 4.41: Περσικός collider και τάγμα Περσών.....	48
Σχήμα 4.42: Μέθοδος ελέγχου των booleans του SelectionUI	49
Σχήμα 4.43: Question1UI Μάχη των Θερμοπυλών	50
Σχήμα 4.44: Μέθοδοι που καλούνται με την επιλογή σωστής στρατηγικής στη μάχη των Θερμοπυλών	51
Σχήμα 4.45: 300 & Υδάρνης.....	51
Σχήμα 4.46: Μέθοδος καταστροφής τάγματος Υδάρνη.....	52
Σχήμα 4.47: Image Target άσκησης της Ναυμαχίας της Σαλαμίνας.....	52
Σχήμα 4.48: ArxiUI σε στυλ κόμικ	53
Σχήμα 4.49: Arxi2 UI με την επιλογή συμπλήρωσε το κενό	54
Σχήμα 4.50: Σωστή απάντηση από το χρήστη στη Ναυμαχία της Σαλαμίνας.....	54
Σχήμα 4.51: Arguing UI Speech Bubbles	55
Σχήμα 4.52: Arguing UI επιλογή αρχηγού.....	55
Σχήμα 4.53: ReadInput script.....	56
Σχήμα 4.54: UI που προετοιμάζει το χρήστη για τη ναυμαχία της Σαλαμίνας.....	57
Σχήμα 4.55: Δημιουργία χάρτη της Ναυμαχίας της Σαλαμίνας.....	57
Σχήμα 4.56: Δημιουργία μονοπατιού μέσω Dreamtech Spline.....	58
Σχήμα 4.57: Ρυθμίσεις Spline Follower	59
Σχήμα 4.58: Χειρισμός κινήσεων Dreamteck splines	60
Σχήμα 4.59: Trigger πάνω σε spline και κάλεσμα μεθόδων	60
Σχήμα 4.60: Phase2 UI , ναυμαχία της Σαλαμίνας	61
Σχήμα 4.61: Question2 spline trigger και κάλεσμα μεθόδων.....	61
Σχήμα 4.62: Question2 UI, ναυμαχία της Σαλαμίνας	62

Συντομογραφίες

Π.Ε.	Πτυχιακή Εργασία
AR	Augmented Reality (Επαυξημένη Πραγματικότητα)
VR	Virtual Reality (Εικονική Πραγματικότητα)
MR	Mixed Reality (Μικτή Πραγματικότητα)
ER	Extended Reality (Εκτεταμένη Πραγματικότητα)
URP	Universal Render Pipeline
UI	User Interface (διεπαφή χρήστη)
π.χ	Παραδείγματος χάριν
κ.α	Και άλλα

Κεφάλαιο 1ο: Εισαγωγή

1.1 Στόχος

Η εφαρμογή αποσκοπεί κατά κύριο λόγο στην διευκόλυνση της κατανόησης του μαθήματος της ιστορίας Δ' Δημοτικού με έναν τρόπο σύγχρονο, που να είναι γνώριμος στους μαθητές αλλά πάνω απ' όλα διασκεδαστικός.

Αυτό επιτυγχάνεται μέσω της ανάπτυξης της εφαρμογής επαυξημένης πραγματικότητας που περιγράφεται στα παρακάτω κεφάλαια. Η εφαρμογή σχεδιάστηκε με στόχο να λειτουργεί ως συμπληρωματικό και βοηθητικό εργαλείο για την διδασκαλία του μαθήματος της Ιστορίας της Δ' Δημοτικού. Επίσης, σχεδιάστηκε και με το στόχο να επιτρέπει στους μαθητές να κατανοήσουν καλύτερα τις πληροφορίες, συνδέοντας τα γεγονότα με οπτικές αναπαραστάσεις και διαδράσεις.

Η εφαρμογή έχει σχεδιαστεί με τέτοιο τρόπο ώστε να καθοδηγεί τον μαθητή διαδραστικά στην εξερεύνηση ιστορικών μαχών, ενώ παράλληλα ενσωματώνει εκπαιδευτικά στοιχεία συνδέοντας το περιεχόμενο της εφαρμογής με σχετικό υλικό από το βιβλίο του μαθητή και το τετράδιο εργασιών της ιστορίας της τέταρτης τάξης του δημοτικού σχολείου. Τα γραφικά σχεδιάστηκαν με σκοπό να βοηθούν στην κατανόηση στρατηγικών κινήσεων για τρεις σημαντικές μάχες στην ιστορία των αρχαίων χρόνων. Τη μάχη του Μαραθώνα, τη μάχη των Θερμοπυλών και τη ναυμαχία της Σαλαμίνας.

Επιπρόσθετα, το παιχνίδι στοχεύει στην καλλιέργεια μιας θετικής στάσης προς το μάθημα της Ιστορίας, παρουσιάζοντας την ως μια συναρπαστική και ενδιαφέρουσα, αντί για μια απλή αποστήθιση γεγονότων.

Ως τελικός στόχος της παρούσας πτυχιακής λογίζεται η δημιουργία ενός εργαλείου που θα μπορεί να χρησιμοποιηθεί στην τάξη ή και στο σπίτι, εμπλουτίζοντας τη διδασκαλία με την μέθοδο της τεχνολογίας.

1.2 Δομή

Η δομή του κειμένου της παρούσας πτυχιακής εργασίας αποτελείται από πέντε κεφάλαια, τα οποία καλύπτουν την ανάπτυξη και την παρουσίαση της εκπαιδευτικής εφαρμογής αφήγησης με χρήση AR. Αρχικά στο πρώτο κεφάλαιο γίνεται αναφορά στον στόχο και στην δομή της πτυχιακής εργασίας με τίτλο «Εκπαιδευτική εφαρμογή αφήγησης με χρήση επαυξημένης πραγματικότητας για το μάθημα της Ιστορίας της Δ' Δημοτικού».

Στο δεύτερο κεφάλαιο αναλύεται το θεωρητικό υπόβαθρο της επαυξημένης πραγματικότητας μέσα από το συνεχές της Εικονικότητας-Πραγματικότητας. Το κεφάλαιο αυτό περιλαμβάνει ορισμούς και εφαρμογές της εκτεταμένης πραγματικότητας και εφαρμογές της, καθώς και τη σημασία της σε πολλούς βασικούς τομείς της ζωής μας.

Στην συνέχεια στο τρίτο κεφάλαιο, γίνεται ανάλυση και παρουσίαση των βασικών τεχνολογιών που χρησιμοποιήθηκαν κατά την ανάπτυξη της εκπαιδευτικής εφαρμογής. Οι τεχνολογίες αυτές στη βάση τους είναι το Unity, το Blender και η Vuforia Engine.

Στο τέταρτο κεφάλαιο αναλύονται αρχικά οι βασικές λειτουργίες που επαναχρησιμοποιήθηκαν κατά την υλοποίηση της εφαρμογής, καθώς παρουσιάζεται και η λειτουργικότητα της εφαρμογής, μέσω της ανάλυσης των χαρακτηριστικών και της δομής των βασικών ασκήσεων. Περιλαμβάνει λεπτομέρειες

Κεφάλαιο 1

για τη Μάχη του Μαραθώνα, τη Μάχη των Θερμοπυλών και τη Ναυμαχία της Σαλαμίνας, όπως επίσης και για το κεντρικό μενού.

Τέλος, στο πέμπτο και τελευταίο κεφάλαιο, εξάγουμε συμπεράσματα σχετικά με τα ευρήματα της πτυχιακής εργασία της εφαρμογής και προτείνονται βελτιώσεις για την επέκταση και τη βελτίωση της εφαρμογής.

1.3 Επίλογος

Στο πρώτο αυτό κεφάλαιο γίνεται συνοπτική αναφορά στη δομή και τους στόχους της Π.Ε σχετικά με την εκπαιδευτική εφαρμογή της αφήγησης επαυξημένης πραγματικότητας για το μάθημα της ιστορίας της τέταρτης δημοτικού.

Κεφάλαιο 2ο: Το Συνεχές Εικονικότητας-Πραγματικότητας

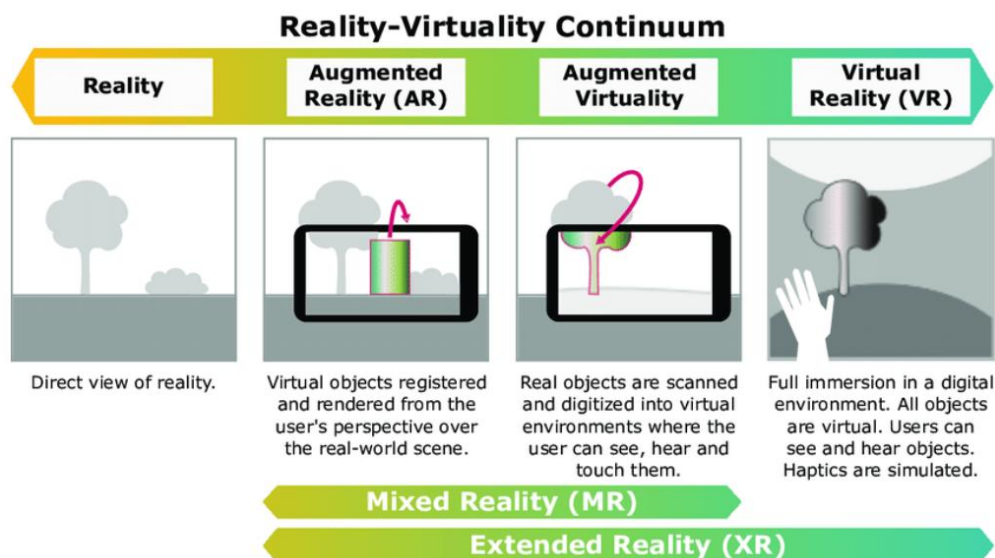
2.1 Εισαγωγή

Ενώ η πλειονότητα των ανθρώπων έχει εστιάσει στο AR και το VR για να περιγράψει τις διάφορες τεχνολογίες, αυτοί οι όροι δεν επαρκούν για την πλήρη κατανόηση του βάθους αυτών των τεχνολογιών και την αξιοποίηση των δυνατοτήτων τους.

Η επαυξημένη πραγματικότητα είναι μία τεχνολογία που ανήκει στο φάσμα της εικονικότητας-πραγματικότητας (Σχήμα 2.1), η οποία είναι μια συνεχής κλίμακα που κυμαίνεται μεταξύ της εντελώς εικονικής και της εντελώς πραγματικής πραγματικότητας και βοηθά στην οπτικοποίηση και την κατανόηση των διαφόρων τεχνολογιών που υπάρχουν μέσα σε αυτό το φάσμα [1].

Το συνεχές πραγματικότητας-εικονικότητας επομένως περιλαμβάνει όλες τις πιθανές παραλλαγές και συνδυασμούς πραγματικών και εικονικών αντικειμένων. Η ιδέα προτάθηκε για πρώτη φορά από τους Paul Milgram και Fumio Kishino το 1994 [2-3]

Οι τεχνολογίες του φάσματος είναι αναφορικά η επαυξημένη πραγματικότητα (Augmented Reality), η εικονική πραγματικότητα (Virtual Reality), και από κει και πέρα ο συνδυασμός αυτών μας δίνει τη μικτή πραγματικότητα (Mixed Reality) και την εκτεταμένη πραγματικότητα (Extended Reality), που αποτελείται από όλες τις τεχνολογίες που επεκτείνουν την πραγματικότητα που βιώνουμε, είτε συνδυάζοντας τον εικονικό με τον πραγματικό κόσμο, είτε δημιουργώντας μια πλήρως εικονική εμπειρία [1-3].



Σχήμα 2.1: Το Συνεχές Εικονικότητας-Πραγματικότητας

2.2 Επαυξημένη Πραγματικότητα (AR)

Η επαυξημένη πραγματικότητα αναφέρεται σε μια τεχνολογία που προσθέτει ψηφιακές πληροφορίες σε αντικείμενα ή τοποθεσίες του πραγματικού κόσμου, με στόχο να εμπλουτίσει την εμπειρία του χρήστη. Εικονικές πληροφορίες που παράγονται από υπολογιστή, όπως κείμενο, εικόνες, μοντέλα 3D, μουσική και βίντεο προσομοιώνονται και εφαρμόζονται στον πραγματικό κόσμο μέσω της AR. Με την επαυξημένη πραγματικότητα, μπορούμε να βλέπουμε μια σύνθεση φυσικών και ψηφιακών στοιχείων. Ωστόσο δεν μπορεί να υπάρξει αλληλεπίδραση μεταξύ των ψηφιακών στοιχείων και των στοιχείων του φυσικού κόσμου [4]. Το σχήμα 2.2 αποτελεί ένα παράδειγμα επαυξημένης πραγματικότητας. Τα εικονικά μοντέλα που εμφανίζονται στην οθόνη ενός smartphone επαυξάνονται με πληροφορίες σε πραγματικό χρόνο που παρέχονται από την κάμερα ενός κινητού τηλεφώνου. Η υλοποίηση αυτής της πτυχιακής εργασίας χρησιμοποιεί τεχνολογία επαυξημένης πραγματικότητας μέσω της χρήσης της πλατφόρμας Unity και των σχετικών εργαλείων της.



Σχήμα 2.2: Επαυξημένη Πραγματικότητα (AR)

2.2.1 Κατανόηση της Επαυξημένης Πραγματικότητας

Μια συσκευή AR, όπως προαναφέραμε, λαμβάνει μια ροή περιεχομένου βίντεο από το οπτικό πεδίο του χρήστη, ανιχνεύοντας το περιβάλλον και παρακολουθώντας φυσικά αντικείμενα προσφέροντας στους χρήστες μια εμπειρία που μπορεί να είναι ενημερωτική ή διασκεδαστική με βάση την εκάστοτε εφαρμογή. Αυτό, μπορεί να περιλαμβάνει τη συλλογή δεδομένων από διάφορα τεχνολογικά προϊόντα όπως smartphone, tablet, γυροσκόπια, GPS κ.α. [5-6].

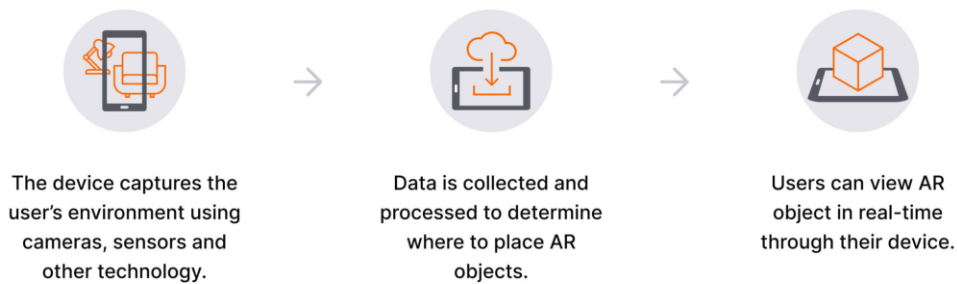
Η AR χρειάζεται συγκεκριμένα στοιχεία και ακολουθεί μια διαδικασία για να δημιουργήσει μια πραγματικότητα όσο το δυνατόν πιο κοντά στον ρεαλισμό. Αρχικά, οι συσκευές εισόδου αποτελούνται από κάμερες και αισθητήρες που καταγράφουν πληροφορίες από τον πραγματικό κόσμο, μεταδίδοντας στη συνέχεια αυτά τα δεδομένα για επεξεργασία [6].

Στη συνέχεια, χρησιμοποιούνται αλγόριθμοι για την ανάλυση των δεδομένων που λαμβάνονται από τις συσκευές εισόδου, επιτρέποντας την παρακολούθηση της θέσης, του προσανατολισμού και των αντικειμένων που επαυξάνονται στο πραγματικό περιβάλλον. Οι πληροφορίες AR παρουσιάζονται σε μια τρισδιάστατη «εμπειρία» πάνω στο αντικείμενο και όχι σαν μια ιστοσελίδα στο διαδίκτυο που

παρουσιάζει 2D τα δεδομένα σε μια οθόνη. Αυτό που βλέπει ο χρήστης, λοιπόν, είναι εν μέρει πραγματικό και εν μέρει ψηφιακό [6,7].

Οι συσκευές εξόδου, συμπεριλαμβανομένων οθονών, και διάφορων μέσων προβολής γενικότερα, συνδυάζουν το ψηφιακό και το φυσικό πεδίο για να παρέχουν μια ομαλή εμπειρία επαυξημένης πραγματικότητας.

Οι συσκευές που συνδέονται συχνότερα με την επαυξημένη πραγματικότητα είναι τα κινητά τηλέφωνα. Εξοπλισμένα με όλα τα απαραίτητα στοιχεία για τη δημιουργία εμπειριών AR, τα κινητά τηλέφωνα μπορεί να μην προσφέρουν τη μέγιστη δυνατή υπολογιστική δύναμη, ωστόσο παραμένουν αποτελεσματικά και, κυρίως, εξαιρετικά προσιτά [7]. Το Σχήμα 2.3 παρουσιάζει μια απλοποιημένη αναπαράσταση της διαδικασίας που προαναφέρθηκε



Σχήμα 2.3: Πως λειτουργεί η Ε.Π.

2.3 Εικονική Πραγματικότητα (VR)

Μέσα από την Εικονική πραγματικότητα γίνεται χρήση της τεχνολογίας των υπολογιστών, smartphone κ.α για να δημιουργηθεί ένας προσομοιωμένο, τρισδιάστατος κόσμος που ο χρήστης μπορεί να χειριστεί και να εξερευνήσει νιώθοντας σαν να ήταν στον πραγματικό κόσμο. Στις μέρες μας έχουν σχεδιαστεί πολλές συσκευές και εφαρμογές για την επίτευξη αυτού του στόχου είτε για μάθηση, έρευνα, είτε ακόμα και για ψυχαγωγία-διασκέδαση (Σχήμα 2.4).



Σχήμα 2.4: Εικονική Πραγματικότητα(VR)

Οι απόψεις δίστανται για το τι ακριβώς συνιστά μια πραγματική εμπειρία VR, αλλά γενικά τα βασικά στοιχεία που θα πρέπει να περιλαμβάνει είναι δύο. Αρχικά, οι τρισδιάστατες εικόνες, που παρουσιάζονται στον χρήστη κατά την διάρκεια μιας εμπειρίας VR, θα πρέπει να φαίνονται να έχουν φυσικό μέγεθος από την οπτική του γωνία του χρήστη. Κατά δεύτερον, να είναι δυνατή η παρακολούθηση των κινήσεων ενός χρήστη, ιδιαίτερα των κινήσεων του κεφαλιού και των ματιών του, και αντίστοιχα να υπάρχει προσαρμογή των εικόνων στην οθόνη που βλέπει ο χρήστης, ώστε να αντικατοπτρίζουν την αλλαγή της προοπτικής [8].

2.4 Μικτή πραγματικότητα

Οι ειδικοί έχουν αντιμετωπίσει δυσκολία στον ακριβή προσδιορισμό της Μικτής Πραγματικότητας. Έχουν περιγράψει διάφορα χαρακτηριστικά, όπως ότι το γεγονός ότι η Μικτή Πραγματικότητα είναι μία μίξη πραγματικής και εικονικής πραγματικότητας, ή μια ενισχυμένη εκδοχή της Επαυξημένης Πραγματικότητας, ή ακόμα και ένας συνδυασμός Εικονικής και Επαυξημένης πραγματικότητας σε μια εφαρμογή ή συσκευή.

Κάποια παραδείγματα της Μικτής Πραγματικότητας, που χαρακτηρίζονται από κάποια ή και από όλα τα χαρακτηριστικά της που αναφέρθηκαν παραπάνω, είναι το Microsoft HoloLens (Σχήμα 2.5) και εφαρμογές όπως το PokemonGO (Σχήμα 2.6). Ωστόσο, είναι σημαντικό να σημειώσουμε ότι στην κατηγορία της Μ.Π, ανήκουν εφαρμογές ή περιβάλλοντα που λαμβάνουν υπόψιν τους το φυσικό περιβάλλον και δεν είναι μια απλή επικάλυψη πληροφοριών [9].



Σχήμα 2.5, 2.6: Microsoft HoloLens (αριστερά) και PokemonGO(δεξιά)

2.5 Εκτεταμένη Πραγματικότητα (XR)

Η Εκτεταμένη Πραγματικότητα είναι ένας γενικός όρος που περιλαμβάνει όλες τις τεχνολογίες που προαναφέρθηκαν. Περιλαμβάνει τεχνολογίες όπως η επαυξημένη πραγματικότητα, η εικονική πραγματικότητα και η μικτή πραγματικότητα, καθώς και εκείνες που ενδέχεται να δημιουργηθούν στο μέλλον. Όλες αυτές οι τεχνολογίες επεκτείνουν την πραγματικότητα που βιώνει ο κάθε άνθρωπος, είτε συνδυάζοντας τον εικονικό με τον πραγματικό κόσμο δημιουργώντας μια πλήρως καθηλωτική εμπειρία. Σύμφωνα με πρόσφατες έρευνες, πάνω από το 60% των ερωτηθέντων πιστεύουν ότι η XR θα γίνει ευρέως διαδεδομένη μέσα στα επόμενα χρόνια [10].

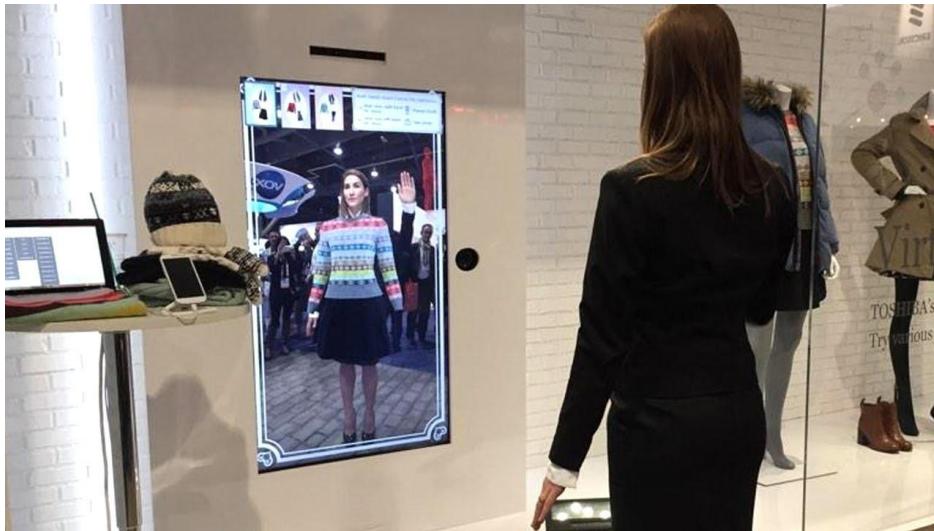
2.6 Εφαρμογές της Εκτεταμένης Πραγματικότητας

Τα τελευταία χρόνια, οι νέες τεχνολογικές εφαρμογές επεκτείνονται προς την εκτεταμένη πραγματικότητα, επιτυγχάνοντας εκπληκτικά αποτελέσματα και δημιουργώντας προσδοκίες για την

καθημερινή ζωή και τις επιχειρήσεις. Σε αυτή την υπο-ενότητα θα αναφερθούν ορισμένοι τομείς της καθημερινότητας όπου έχουν εμφανιστεί τέτοιου είδους εφαρμογές.

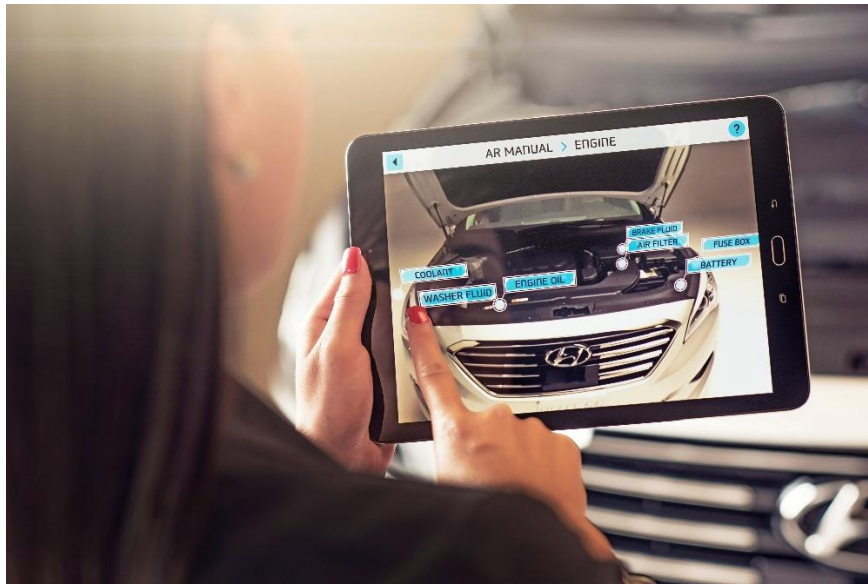
2.6.1 Αγορά

Η εκτεταμένη πραγματικότητα φέρνει την επανάσταση στον τομέα του λιανεμπορίου, παρέχοντας καινοτόμες λύσεις που βελτιώνουν την εμπειρία των πελατών και ενισχύουν την αλληλεπίδραση τους με τα προϊόντα. Μια από τις κύριες χρήσεις της XR είναι η εικονική δοκιμή προϊόντων (virtual try-on), όπου η επαυξημένη πραγματικότητα επιτρέπει στους καταναλωτές να εξετάζουν, να εξατομικεύουν και να ενημερώνονται για τα προϊόντα από την αγορά τους. Αυτό οδηγεί σε πιο σίγουρες επιλογές και αυξημένη ικανοποίηση του καταναλωτή. Χαρακτηριστικό παράδειγμα είναι η κινεζική εταιρεία ρούχων Lily, που εγκατέστησε εικονικό δοκιμαστήριο σε σταθμό μετρό στη Σαγκάη, επιτρέποντας στους πελάτες να δοκιμάζουν ρούχα μέσω διαδραστικών οθονών (Σχήμα 2.7).



Σχήμα 2.7: Virtual Fitting Room της εταιρίας Lily

Επιπλέον, η εκτεταμένη πραγματικότητα βελτιώνει την εμπειρία των πελατών αντικαθιστώντας τα παραδοσιακά έντυπα εγχειρίδια οδηγιών χρήσεως με εγχειρίδια επαυξημένης πραγματικότητας. Αυτά παρέχουν βοηθητικές οπτικές εμπειρίες μέσω κινητών συσκευών για την εισαγωγή, εγκατάσταση και επίλυση προβλημάτων που μπορεί να εμφανίσουν τα προϊόντα τους. Για παράδειγμα, η Hyundai έχει αξιοποιήσει την επαυξημένη πραγματικότητα για να παρέχει σαφείς οδηγίες σχετικά με τα προϊόντα τις, καθιστώντας τη διαδικασία πιο φιλική για τους χρήστες (Σχήμα 2.8).



Σχήμα 2.8: Εφαρμογή AR της Hyundai

Τέλος, η εκτεταμένη πραγματικότητα ενισχύει τις εμπειρίες στο φυσικό κατάστημα μετατρέποντας τα κινητά τηλέφωνα σε ψηφιακούς βοηθούς αγορών. Με τη χρήση της κάμερας του τηλεφώνου, οι πελάτες μπορούν να λαμβάνουν διαδραστικές πληροφορίες για τα προϊόντα που τους ενδιαφέρουν [11].

Αυτές οι εφαρμογές δείχνουν πως η εκτεταμένη πραγματικότητα αλλάζει ριζικά το λιανεμπόριο, ενισχύοντας την αλληλεπίδραση με τους πελάτες και προσφέροντας μοναδικές εμπειρίες που ενισχύουν τη σύνδεση με το brand.

2.6.2 Υγεία

Η εκτεταμένη πραγματικότητα φέρνει την επανάσταση στον τομέα της υγείας, προσφέροντας καινοτόμες λύσεις που βελτιώνουν τη φροντίδα των ασθενών, την εκπαίδευση των επαγγελματιών υγείας και την αποτελεσματικότητα θεραπειών.

Μία από τις βασικές εφαρμογές της εκτεταμένης πραγματικότητας είναι η απομακρυσμένη φροντίδα (Remote Care), όπου οι τεχνολογίες επαυξημένης πραγματικότητας σε συνδυασμό με αυτές της εικονικής πραγματικότητας ενισχύουν την προσβασιμότητα και την αποτελεσματικότητα της τηλεϊατρικής και της απομακρυσμένης παρακολούθησης ασθενών. Μέσα από πλήρως εικονικά περιβάλλοντα, οι ασθενείς και οι γιατροί μπορούν να αλληλοεπιδρούν σαν να βρίσκονται στον ίδιο χώρο, εξαλείφοντας τα εμπόδια που αντιμετωπίζουν κοινότητες σε απομακρυσμένες ή υπο-εξυπηρετούμενες περιοχές [12].

Η εκτεταμένη πραγματικότητα έχει επίσης εφαρμογές στην ιατρική απεικόνιση, βοηθώντας τους επαγγελματίες υγείας να οπτικοποιούν δεδομένα με ακρίβεια και να σχεδιάζουν πιο αποτελεσματικές θεραπείες. Τεχνολογίες επαυξημένης πραγματικότητας και εικονικής πραγματικότητας επιτρέπουν την προβολή τρισδιάστατων ανατομικών δεδομένων απευθείας πάνω στους ασθενείς διευκολύνοντας τη διάγνωση και τις επεμβατικές διαδικασίες. Χαρακτηριστικό παράδειγμα αποτελεί η πρώτη χειρουργική επέμβαση σπονδυλικής στήλης με εκτεταμένη πραγματικότητα που πραγματοποιήθηκε στο πανεπιστήμιο John Hopkins, όπου οι χειρουργοί έβλεπαν τρισδιάστατες εικόνες ενώ πραγματοποίησαν χειρουργική επέμβαση σε ζωντανό ασθενή [13]. Κατά τη διάρκεια της διαδικασίας τοποθέτησαν έξι βίδες στη σπονδυλική στήλη του ασθενούς οι οποίες θα βοηθούσαν στην ανακούφιση από τον χρόνιο

πόνο που είχε. Η τεχνολογία που χρησιμοποιήθηκε αποτελούνταν από ένα headset με see through οπτική οθόνη που πρόβαλε εικόνες της εσωτερικής ανατομίας του ασθενούς, όπως οστά αιμοφόρα αγγεία κ.α. (Σχήμα 2.9).



Σχήμα 2.9: Επέμβαση σπονδυλικής στήλης με AR στο πανεπιστήμιο John Hopkins

Η εκτεταμένη πραγματικότητα, επίσης, επαναπροσδιορίζει την εκπαίδευση επαγγελματιών υγείας παρέχοντας τη δυνατότητα εκπαίδευσης από οποιονδήποτε και οποιαδήποτε στιγμή. Μέσα από προσομιώσεις XR, οι εκπαιδευόμενοι μπορούν να εξασκούνται με ασφάλεια και αποτελεσματικότητα ακόμα χωρίς την ανάγκη ακριβού εξοπλισμού ή ρίσκου που σχετίζεται με πραγματικές διαδικασίες.

Επιπλέον, η εκτεταμένη πραγματικότητα συμβάλλει στη διαχείριση του πόνου και του άγχους. Η θεραπεία μέσω εικονικής πραγματικότητας βοηθά τους ασθενείς να αποσπαστούν από τον πόνο ή τις ανησυχίες τους μέσω προβολής εικονικών θεραπευτικών κόσμων. Μελέτες δείχνουν ότι η εικονική πραγματικότητα μπορεί να μειώσει σημαντικά τον πόνο ενώ πάνω από 100 νοσοκομεία στις Ηνωμένες Πολιτείες την εφαρμόζουν ήδη για τη διαχείριση του πόνου και του άγχους [14].

Τέλος η εκτεταμένη πραγματικότητα χρησιμοποιείται στη φαρμακευτική βιομηχανία για τη δημιουργία παρουσιάσεων που εξηγούν πώς λειτουργούν τα φάρμακα στο σώμα. Για παράδειγμα, η Bayer Pharmaceuticals αξιοποιεί εικονική πραγματικότητα και επαυξημένη πραγματικότητα για να δημιουργεί διαδραστικές παρουσιάσεις που διευκολύνουν την κατανόηση των εφαρμογών θεραπειών από γιατρούς [15] (Σχήμα 2.10).



Σχήμα 2.10: Ολόγραμμα AR στα φάρμακα της Bayer Pharmaceuticals

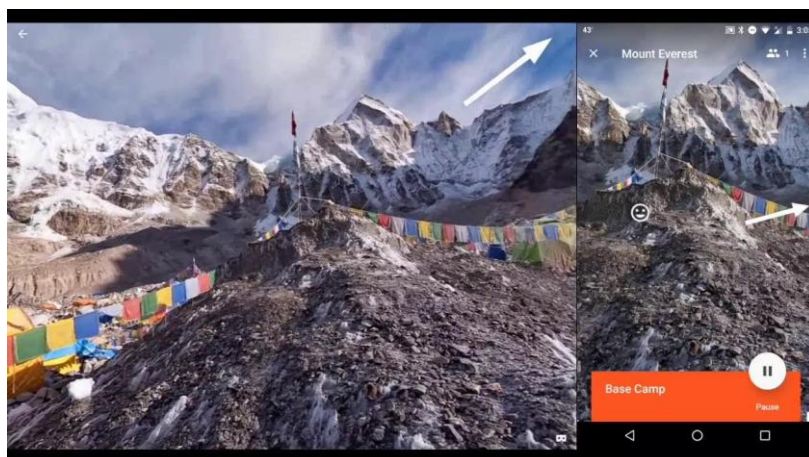
Αυτές οι χρήσεις καταδεικνύουν τις δυνατότητες της εκτεταμένης πραγματικότητας στην υγειονομική περιθαλψη.

2.6.3 Εκπαίδευση

Στην εκπαίδευση, η εκτεταμένη πραγματικότητα φέρνει επανάσταση στον τρόπο που οι μαθητές και επαγγελματίες εκπαιδεύονται, προσφέροντας εμπειρίες εκτεταμένης πραγματικότητας που κάνουν τη μάθηση πιο διαδραστική και αποτελεσματική.

Στο πλαίσιο της ενίσχυσης της εκπαιδευτικής διαδικασίας στην εκπαίδευση μαθητών, ένα παράδειγμα είναι η χρήση της επαυξημένης πραγματικότητας για να φέρουν επιστημονικές έννοιες και εικόνες να ζωντανεύουν μέσα στην τάξη. Μία τέτοια εφαρμογή, είναι το SkyView, όπου επιτρέπει στους μαθητές να εξερευνούν το σύμπαν και να μαθαίνουν για ουράνια αντικείμενα στον νυχτερινό ουρανό, ενώ εφαρμογές όπως το Froggipedia δίνουν τη δυνατότητα να εξετάσουν τα εσωτερικά όργανα ενός βατράχου χωρίς να απαιτείται η φυσική ανατομία [16].

Η εκτεταμένη πραγματικότητα προσφέρει επίσης εικονικές σχολικές εκδρομές, όπως επισκέψεις στο Base Camp του Έβερεστ ή στο μουσείο του Λούβρου μέσω της εφαρμογής Google Expeditions (Σχήμα 2.11).



Σχήμα 2.11: Google Expeditions στο Έβερεστ

Στην ιατρική εκπαίδευση, το Microsoft HoloLens προσφέρει εμπειρίες μεικτής πραγματικότητας, όπου οι φοιτητές μπορούν να περιηγηθούν στο ανθρώπινο σώμα ακόμα να εξερευνήσουν το κυκλοφοριακό σύστημα και να κατανοήσουν πώς να αντιμετωπίζουν ιατρικές καταστάσεις [17].

Εκτός από την εκπαίδευση μαθητών και φοιτητών η εκτεταμένη πραγματικότητα εφαρμόζεται και στο πεδίο της ποινικής δικαιοσύνης. Για παράδειγμα, χρησιμοποιείται για την εκπαίδευση και την κατάρτιση αστυνομικών, δικαστών και άλλων επαγγελματιών του χώρου, επιτρέποντάς τους να εξασκούνται σε προσομοιωμένα σενάρια χωρίς τους κινδύνους του πραγματικού κόσμου. Επιπλέον, συμβάλλει στην αναπαράσταση εγκληματικών σκηνών, δίνοντας τη δυνατότητα στους ερευνητές να δημιουργούν εικονικές αναπαραστάσεις τόπων εγκλήματος, γεγονός που διευκολύνει την ανάλυση και την κατανόηση των γεγονότων. Επιπρόσθετα, η εκτεταμένη πραγματικότητα βρίσκει εφαρμογή στην αποκατάσταση κρατουμένων, βοηθώντας τους να αναπτύξουν κοινωνικές δεξιότητες και να προετοιμαστούν για την επανένταξή τους στην κοινωνία [18].

2.6.4 Ψυχαγωγία

Η ψυχαγωγία υπήρξε ο πρώτος τομέας μέσω του οποίου το ευρύ κοινό ήρθε σε επαφή με την εκτεταμένη πραγματικότητα. Τα τελευταία χρόνια, με την ταχεία ανάπτυξη της τεχνολογίας, η εκτεταμένη πραγματικότητα έχει γίνει ακόμη πιο προσιτή, προσθέτοντας νέες δυνατότητες και εμπειρίες που αλλάζουν τον τρόπο που αντιλαμβανόμαστε την ψυχαγωγία.

Για παράδειγμα, ας πάρουμε το παιχνίδι PokemonGo, όπου πολλοί άνθρωποι είναι εξοικειωμένοι. Κυκλοφόρησε το 2015, χρησιμοποιεί τεχνολογία επαυξημένης πραγματικότητας smartphone που βασίζεται στην τοποθεσία για να δημιουργήσει μια προσομοιωμένη εμπειρία παρακολούθησης χαρακτήρων Pokemon στον πραγματικό κόσμο. Με περισσότερα από 500 εκατομμύρια χρήσεις τον πρώτο χρόνο της κυκλοφορίας της, εξακολουθεί να είναι ο κορυφαίος υποστηρικτής των παιχνιδιών επαυξημένης πραγματικότητας για κινητά παγκοσμίως [19].

Σήμερα η εκτεταμένη πραγματικότητα δεν περιορίζεται μόνο σε παιχνίδια. Εφαρμογές όπως το SnapChat και το Instagram χρησιμοποιούν την εκτεταμένη πραγματικότητα για τη δημιουργία φίλτρων που εμπλουτίζουν την έκφραση, την εμφάνιση και τη διασκέδαση. Παράλληλα οι πλατφόρμες streaming ενσωματώνουν τεχνολογίες εκτεταμένης πραγματικότητας σε ζωντανές μεταδόσεις, από συναυλίες ή αθλητικά γεγονότα για να δημιουργήσουν ρεαλιστικά εικονικά περιβάλλοντα.

2.7 Επίλογος

Η εκτεταμένη πραγματικότητα και οι τεχνολογίες που την απαρτίζουν, αντιπροσωπεύουν μια σημαντική πρόοδο στην συγχώνευση των κόσμων του πραγματικού και του εικονικού. Η εκτεταμένη πραγματικότητα εξελίχθηκε σε ένα εργαλείο με πολλαπλές εφαρμογές, επηρεάζοντας όλους τους τομείς της ανθρώπινης ζωής. Η δυναμική της εκτεταμένης πραγματικότητας δεν περιορίζεται μόνο στο παρόν. Οι τεχνολογίες συνεχίζουν να εξελίσσονται, προσφέροντας μεγαλύτερη ακρίβεια και βελτιωμένη διαδραστικότητα. Αυτό το κεφάλαιο ανέδειξε τον ορισμό και σημασία τέτοιου είδους τεχνολογιών, καθώς και τις δυνατότητες που παρέχουν μέσα από παραδείγματα.

Κεφάλαιο 3ο: Πλατφόρμες & εργαλεία υλοποίησης

3.1 Εισαγωγή

Στο παρόν κεφάλαιο θα παρουσιαστούν οι πλατφόρμες, τα πακέτα και τα εργαλεία που χρησιμοποιήθηκαν για την δημιουργία της AR εφαρμογής. Οι βασικές πλατφόρμες που χρησιμοποιήθηκαν, ήταν αρχικά, το πρόγραμμα σχεδίασης 3D γραφικών Blender, για την δημιουργία των χαρακτήρων και των πλοίων που εμφανίζονται στις ασκήσεις. Επιπλέον, για την ανάπτυξη της AR εφαρμογής το Unity ήταν η βασική πλατφόρμα υλοποίησης της εφαρμογής, σε συνδυασμό με την τεχνολογία της Vuforia Engine για την ενσωμάτωση περισσότερων χαρακτηριστικών AR. Τέλος, για την συγγραφή του κώδικα σε C#, έγινε χρήση του Visual Studio Code.

3.2 Πλατφόρμα & εργαλεία Unity

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκαν βασικές λειτουργίες και εργαλεία της μηχανή παιχνιδιού Unity (Animation, prefabs κ.α.), όπως επίσης και του Unity Package Manager. Η πλατφόρμα Unity είναι μια πλατφόρμα ανάπτυξης λογισμικού που χρησιμοποιείται κυρίως για τη δημιουργία διαδραστικών εμπειριών, όπως παιχνίδια, εφαρμογές AR και VR. Είναι γνωστή για την ευελιξία και την ευκολία χρήσης της, καθιστώντας την ιδανική επιλογή για προγραμματιστές όλων των επιπέδων, από αρχάριους έως επαγγελματίες.

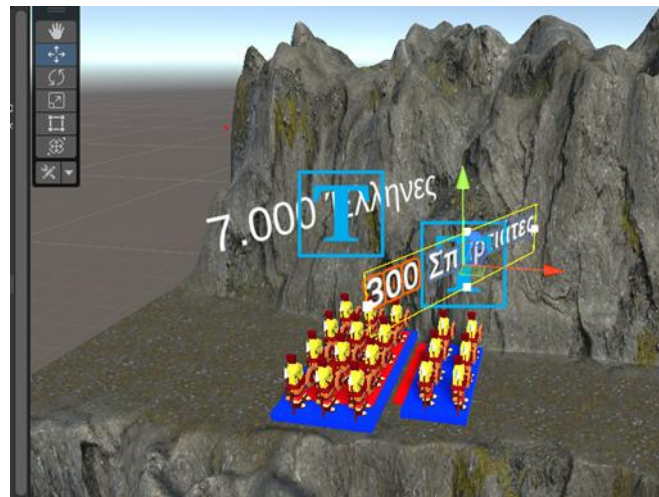
Το Unity Παρέχει ένα ισχυρό περιβάλλον ανάπτυξης, το Unity Editor, όπου οι χρήστες μπορούν να σχεδιάζουν σκηνές, να προσθέτουν GameObjects και να δημιουργούν scripts με τη χρήση C#. Το Unity διαθέτει επίσης ισχυρή μηχανή γραφικών για τη δημιουργία 2D και 3D περιεχομένου, ενώ υποστηρίζει την προσομοίωση φυσικής στα GameObjects, φωτισμού και animation.

Μια από τις μεγάλες δυνατότητες του Unity είναι η ενσωμάτωσή του με το Asset Store, μια πλατφόρμα όπου οι προγραμματιστές μπορούν να βρουν προ-δημιουργημένα στοιχεία, όπως 3D μοντέλα, ήχους και scripts. Επιπλέον, το Unity είναι ιδανικό για την ανάπτυξη εφαρμογών AR, για αυτό το λόγο επιλέχθηκε για την ανάπτυξη της παρούσας πτυχιακής.

Το Unity Package Manager είναι ένα ενσωματωμένο εργαλείο στο Unity που επιτρέπει τη διαχείριση και εγκατάσταση εργαλείων επεξεργασίας και βιβλιοθηκών, όπως προγράμματα επεξεργασίας κειμένου, λειτουργίες animation κ.α. Ακολουθούν τα πιο σημαντικά πακέτα που χρησιμοποιήθηκαν για την ανάπτυξη της παρούσας πτυχιακής εργασίας μέσα από το Unity Package Manager [20].

3.2.1 TextMeshPro

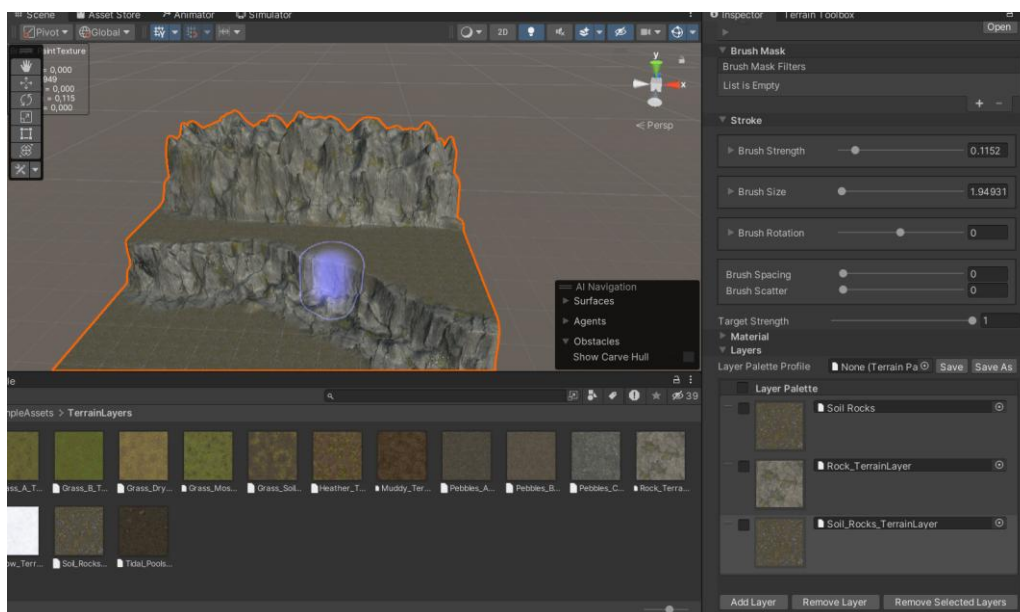
Το συγκεκριμένο πακέτο χρησιμοποιήθηκε για την εισαγωγή κειμένου μέσα στην εφαρμογή. Είναι ένα εργαλείο το οποίο προσφέρει υψηλή ανάλυση κειμένου, σε μικρές και μεγάλες κλίμακες και σε υψηλά επίπεδα ζουμ, ένα χαρακτηριστικό που φάνηκε χρήσιμο για την ένταξη πληροφορίας πάνω από τα στρατεύματα της δεύτερης μάχης του παιχνιδιού καθώς αυτά θα έπρεπε να είναι ευανάγνωστα στον χρήστη ανεξάρτητα από την κλίμακα τους (Σχήμα 3.1). Επίσης, η ευελιξία στην μορφοποίηση, στο στυλ και στις επιλογές της γραμματοσειράς, έδωσε την δυνατότητα να υποστηρίζονται μικρά και κεφαλαία γράμματα, όπως και ειδικόι χαρακτήρες της ελληνικής γλώσσας στην εφαρμογή.



Σχήμα 3.1: TextMeshPro στα Ελληνικά στρατεύματα

3.2.2 Terrain Sample Asset Pack

Το Terrain Sample Asset Pack είναι ένα πακέτο συμβατό με το σύστημα δημιουργίας Terrain του Unity, καθιστώντας εύκολη τη δημιουργία και την προσαρμογή εδάφους. Παρέχει όλα τα απαραίτητα εργαλεία και πόρους για την δημιουργία περιβαλλόντων που θέλουμε να έχουν ρεαλιστικά εδάφη. Αυτό το πακέτο στάθηκε ιδιαίτερα χρήσιμο, γιατί μας δίνει τη δυνατότητα να έχουμε περισσότερες επιλογές στο sculpting του εδάφους με διαφορετικές υφές και βούρτσες. Τα εδάφη που δημιουργούνται από το συγκεκριμένο εργαλείο μπορούν να έχουν από την αρχή συγκεκριμένα settings για το scale και το position τους. Τέλος, η μεγάλη συλλογή από υφές υψηλής ανάλυσης που παρείχε το εργαλείο έκαναν το έδαφος να είναι πιο κοντά στην πραγματικότητα. για να φτιαχτεί το στενό των Θερμοπυλών (Σχήμα 3.2).



Σχήμα 3.2: Στενό των Θερμοπυλών

3.2.3 Dreamteck Splines

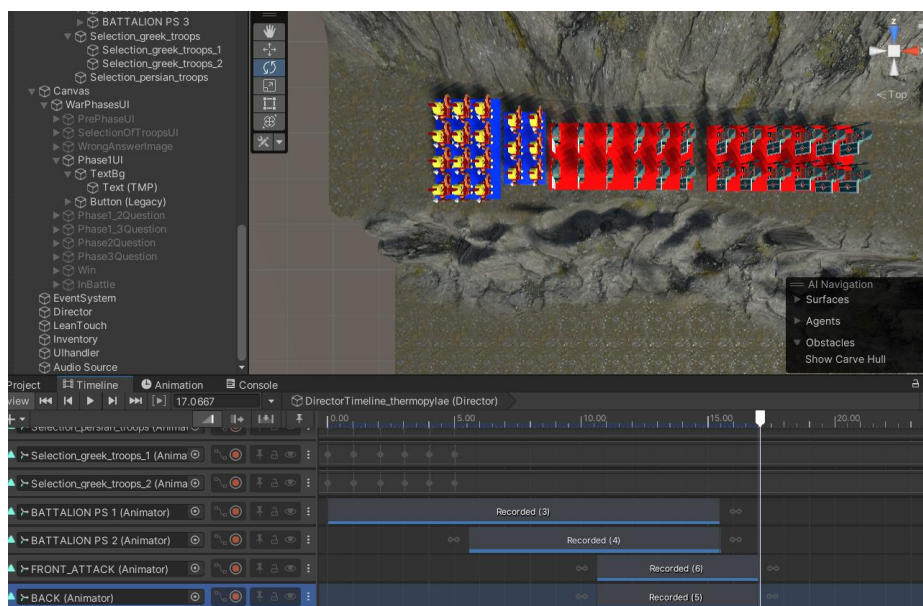
Ένα ακόμα πακέτο που χρησιμοποιήθηκε κατά την ανάπτυξη της εφαρμογής είναι το πακέτο Dreamteck Splines, το οποίο είναι ένα ισχυρό σύστημα διαχείρισης καμπυλών (splines). Τα splines μας βοηθούν στην δημιουργία διαδρομών που μπορούν να ακολουθηθούν μέσα από τα Game Objects της εφαρμογής. Το συγκεκριμένο εργαλείο αποδείχθηκε εξαιρετικά χρήσιμο για τη δημιουργία μονοπατιών πάνω στον διαδραστικό χάρτη του παιχνιδιού της Ναυμαχίας της Σαλαμίνας. Αυτά τα μονοπάτια χρησιμοποιήθηκαν για να ορίζουν τις διαδρομές που ακολουθούν τα πλοία, στις διαφορετικές φάσεις της αφήγησης και κατά συνέπεια της μάχης. Με τη χρήση των εργαλείων της τεχνολογίας όπως οι καμπύλες Bezier, ήταν δυνατό να σχεδιάσουμε ομαλές διαδρομές οι οποίες προσομοιώνουν τη κίνηση των πλοίων στη θάλασσα. Η δυνατότητα προσθήκης ελέγχου ταχύτητας και αλλαγής κατεύθυνσης κατά μήκος του μονοπατιού επέτρεψε την διάδραση του χρήστη με την εφαρμογή, καθώς ανάλογα με τις απαντήσεις του, τα πλοία μένουν στάσιμα, ή συνεχίζουν την πορεία τους [21].

3.2.4 Animation & Timeline Director

Το εργαλείο animation του Unity είναι ένα ισχυρό εργαλείο για τη δημιουργία κινήσεων και αλληλουχιών κινουμένων σχεδίων. Μέσα από το Timeline Animation Window (Σχήμα 3.3), δημιουργήθηκαν keyframes, δηλαδή βασικά σημεία κίνησης μέσα στον χρόνο, για να δημιουργηθούν οι ρεαλιστικές κινήσεις των στρατευμάτων και των στόλων.

Επίσης μια λειτουργία που χρησιμοποιήθηκε κατά την ανάπτυξη της εφαρμογής είναι το Timeline σε συνδυασμό με το Timeline Director, για τη δημιουργία και τη διαχείριση σκηνών που περιλαμβάνουν τις κινήσεις των Game Objects κατά τη διάρκεια της αφήγησης [22].

Τα εργαλεία του animation και του timeline, βοήθησαν στην δημιουργία διαδραστικής αφήγησης. Για παράδειγμα, ήταν εφικτή η κίνηση των στρατευμάτων στο έδαφος της μάχης των Θερμοπυλών.



Σχήμα 3.3: Animation και Timeline Director

3.2.5 Colliders & Triggers

Η χρήση των δύο αυτών εργαλείων επέτρεψε τα αντικείμενα της εφαρμογής να επηρεάζονται από την φυσική, να αλληλεπιδρούν μεταξύ τους αλλά και με το περιβάλλον τους.

Οι Colliders ορίζουν τα όρια ενός αντικειμένου για τον εντοπισμό συγκρούσεων (collisions) με άλλα αντικείμενα. Αρχικά, έγινε χρήση των Colliders για να διασφαλίσουμε ότι τα κινούμενα αντικείμενα (π.χ στρατεύματα) που ήταν τοποθετημένα πάνω στα εδάφη της εφαρμογής θα αλληλεπιδρούν σωστά με το περιβάλλον, δεν θα διαπεράσουν το έδαφος και θα μένουν σταθερά πάνω σε αυτό. Επίσης, οι colliders εφαρμόστηκαν στα στρατεύματα του παιχνιδιού για να περιορίζονται οι περιττές κινήσεις και οι μη ρεαλιστικές συμπεριφορές, όπως για παράδειγμα ένας στρατιώτης να διαπεράσει το βουνό στο στενό των Θερμοπυλών.

Οι Triggers είναι στην ουσία μια ειδική κατηγορία από colliders που, αντί να προκαλούν φυσικές συγκρούσεις, ενεργοποιούν γεγονότα μέσω scripts όταν ένα αντικείμενο εισέλθει σε έναν άλλο collider [23].

3.2.6 AudioSource

Το εργαλείο AudioSource μας βοηθά να συνδυάσουμε και να χειριστούμε ηχητικά εφέ στην εφαρμογή. Με το AudioSource, μπορούμε να αναπαράγουμε εφέ ήχου και επίπεδα μουσικής.

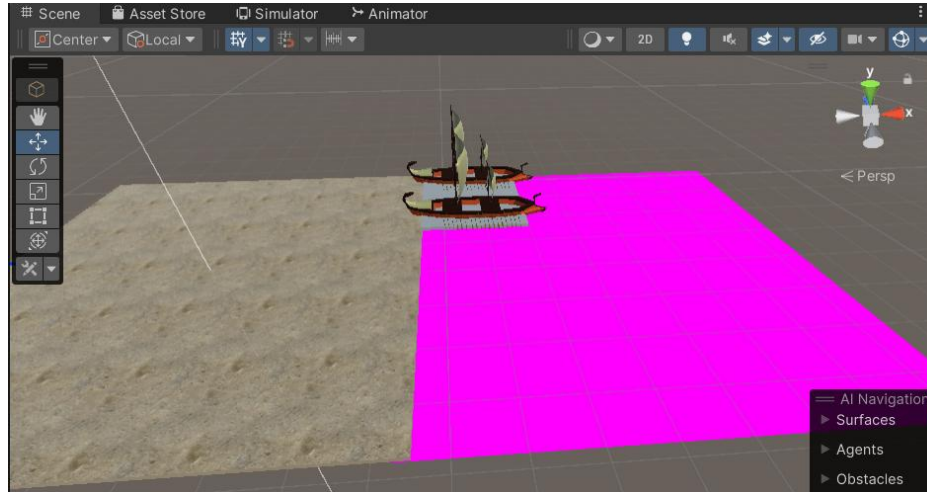
Το συγκεκριμένο εργαλείο χρησιμοποιήθηκε για να μπορεί το Unity να αναγνωρίζει τις ηχογραφήσεις που πραγματοποιήθηκαν για την αφήγηση των μαχών, οι οποίες δημιουργήθηκαν με σκοπό να βοηθήσουν ακόμα περισσότερο τους χρήστες μέσω της ακρόασης. Η διαχείριση του ήχου πραγματοποιήθηκε ενεργά, με το AudioSource αφού έκανε δυνατή την εξατομίκευση των ηχητικών εφέ σε κάθε σκηνή, ρυθμίζοντας κατάλληλα την ένταση και τις ποιότητες ήχου ανάλογα με το αν είναι background μουσική ή αν είναι στο ηχητικό εφέ του σωστού ή του λάθους.

3.2.7 Render Pipelines & Shader Graph

Τα Render Pipelines είναι συστήματα στο Unity που διαχειρίζονται τη διαδικασία απόδοσης των γραφικών. Προσδιορίζουν πώς η μηχανή του Unity θα χειριστεί φωτισμούς, σκιές, υλικά και άλλα οπτικά εφέ για να αποδώσει τις τελικές σκηνές στην οθόνη. Το Unity διαθέτει 3 Pipelines. Το Built-in Render Pipeline, είναι και το βασικό σύστημα που παρέχει μια βασική συμβατότητα αλλά σχετικά μειωμένη προσαρμοστικότητα. Το Universal Render Pipeline (URP), το οποίο είναι ελαφρύ και ευέλικτο και παρέχει βελτιστοποιημένα γραφικά και τέλος το High Definition Render Pipeline (HDRP), το οποίο είναι σχεδιασμένο για έργα που απαιτούν γραφικά πολύ υψηλής ποιότητας [24]. Στην εφαρμογή επιλέχθηκε το URP, το οποίο επέτρεψε την εφαρμογή να λειτουργεί ομαλά σε smartphone, χωρίς να υποβαθμίζει την ποιότητα αλλά επίσης, μέσω του URP αξιοποιήθηκε και ένας custom Shader Graph, που ήταν υπεύθυνος της αναπαράστασης μιας ρεαλιστικής απεικόνισης της θάλασσάς με δυναμική αντανάκλαση και κυματισμούς στο έδαφος στο παιχνίδι της “Μάχης του Μαραθώνα”.

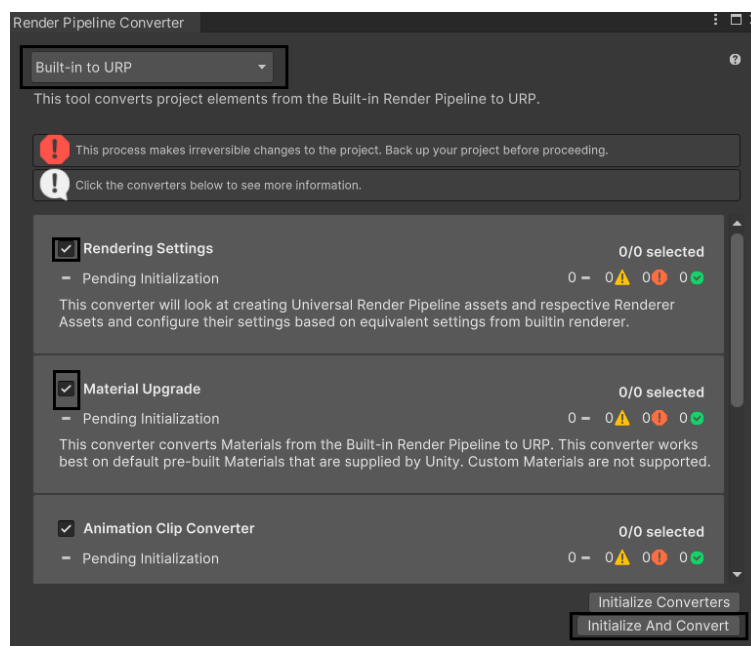
3.2.8 Αναβάθμιση των Materials σε URP

Η μετάβαση των υλικών της εφαρμογής σε URP απαιτούσε τη προσαρμογή τους ούτως ώστε να είναι συμβατά με το νέο pipeline. Ο σκοπός της αναβάθμισης ήταν να διασφαλιστεί η ορθή εμφάνιση των αντικειμένων στον κόσμο της εφαρμογής μας (Σχήμα 3.4).



Σχήμα 3.4: Πριν την αναβάθμιση Material σε URP

Για την πραγματοποίηση της ορθής μετάβασης των υλικών, έπρεπε να έχει γίνει η εγκατάσταση του URP και να έχει αλλάξει το Rendering από τα Project Settings. Όπως επίσης μέσα από το Render Pipeline Converter (Window-> Rendering-> Render Pipeline Converter) να πατηθεί η επιλογή Upgrade project to UniversalRP Materials που ελέγχει όλα τα materials εφαρμογής τα αναβαθμίζει σε URP materials (Σχήμα 3.5).



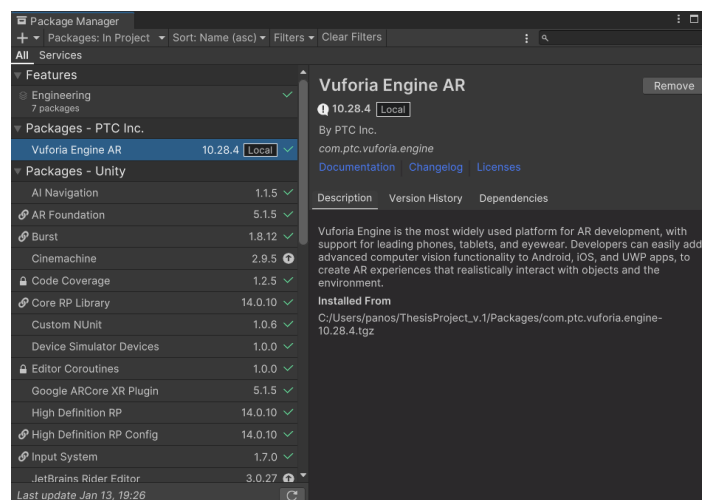
Σχήμα 3.5: Αναβάθμιση των Material σε URP

3.3 Vuforia Engine

Για την ανάπτυξη της εφαρμογής μια από τις πιο σημαντικές βιβλιοθήκες υλοποίησης αποτέλεσε η Vuforia Engine, η οποία και ενσωματώθηκε με το Unity. Η Vuforia Engine είναι μια πλατφόρμα ανάπτυξης επαυξημένης πραγματικότητας εφαρμογών, που συνεργάζεται με το Unity. Επιτρέπει την αναγνώριση αντικειμένων, εικόνων και αντικειμένων ώστε να προβάλλονται οι επαυξημένες πληροφορίες σωστά στη συσκευή εξόδου του χρήστη [25].

3.3.1 Εισαγωγή της Vuforia Engine στο Unity

Η εισαγωγή της Vuforia Engine στο Unity, είναι από τις σημαντικότερες διαδικασίες που πραγματοποιήθηκαν κατά την ανάπτυξη της εφαρμογής. Αρχικά μέσα από το Unity Asset Store κατεβάζουμε το πακέτο της Vuforia Engine, το Vuforia Engine AR για να μπορέσει στην συνέχεια να ενσωματωθεί η έκδοση 10.28.4 στο Unity (Σχήμα 3.6).



Σχήμα 3.6: Vuforia Engine AR Package

Στην συνέχεια πρέπει να γίνει λήψη άδειας χρήσης από το site της Vuforia Engine. Αφού δημιουργήσουμε λογαριασμό στο site, στο tab Plan & Licences πατάμε το κουμπί Generate Basic License, και δίνουμε το κατάλληλο όνομα.

Στην συνέχεια, πραγματοποιούμε λήψη της έκδοσης 10.28.4 για την πλατφόρμα Unity από το tab Downloads->Download for Unity. Αφού το κατεβάσουμε, προσθέτουμε το add-vuforia-package-10-28-4.unitypackage στο Unity, μέσα από το μενού Assets -> Import Package -> Custom Package...

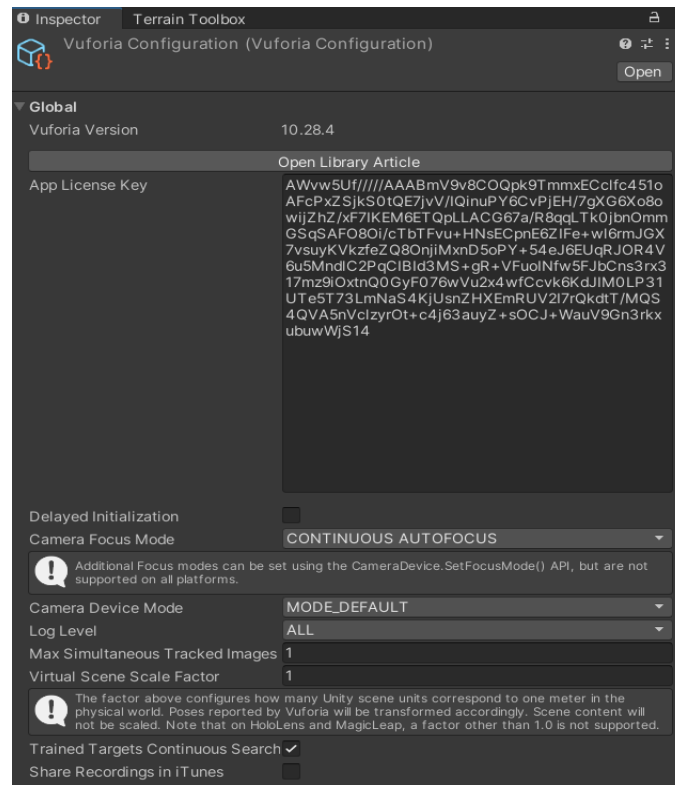
Με την εισαγωγή του πακέτου, το Unity «τρέχει» αυτόματα ένα script για την αυτόματη εκτέλεση των απαιτούμενων τροποποιήσεων στο project Unity με σκοπό την εγκατάσταση ή την αναβάθμιση στο πιο πρόσφατο Vuforia Engine SDK.

Αφού ολοκληρώσουμε την εισαγωγή του Vuforia Engine, έγινε προσβάσιμο από το μενού GameObject στο Unity.

Ένα GameObject ARCamera είναι απαραίτητο για κάθε σκηνή που θέλει να κάνει χρήση και να παράγει επαυξημένο περιεχόμενο. Αυτό το στοιχείο είναι συμβατό με εφαρμογές σχεδιασμένες για φορητές συσκευές.

Μέσα από την Hierarchy κάνοντας δεξί κλικ, επιλέγουμε Vuforia Engine-> ARCamera. Αυτό προσθέτει την ARCamera και μέσα από τον Inspector της ARCamera στο script Vuforia Behavior (script) πατάμε το κουμπί Open Vuforia Engine Configuration.

Στο πεδίο Application License Key, κάνουμε επικόλληση το κλειδί άδειας χρήσης που δημιουργήθηκε προηγουμένως (Σχήμα 3.7).



Σχήμα 3.7: Vuforia Configuration

3.3.2 Vuforia Image Target Database

Η λειτουργία επαύξησης μοντέλων στο χώρο μέσω του Vuforia επιτυγχάνεται αναγνωρίζοντας κάποιο αντικείμενο στο χώρο που βλέπει η κάμερα. Για να συμβεί αυτό θα πρέπει να υπάρχει κάποια βάση δεδομένων που θα περιέχει τις πληροφορίες των στόχων, ώστε να μπορούν να συγκριθούν και να ταυτοποιηθούν [26].

Αρχικά, ξεκινάμε τη διαδικασία δημιουργώντας μια βάση δεδομένων συσκευής στο tab Target Manager. Έπειτα μέσα από την πύλη προγραμματιστών Vuforia Engine, επιλέγουμε τον κατάλληλο τύπο συσκευής σε Device και επιλέγοντας ένα ευδιάκριτο όνομα μπορούμε να δημιουργήσουμε τη βάση με την επιλογή Generate Database. (Σχήμα 3.8)

Generate Database

Database Name *
image|

Type:

Device
 Cloud
 VuMark

Cancel Generate

Σχήμα 3.8: Δημιουργία Βάσης με Image Targets

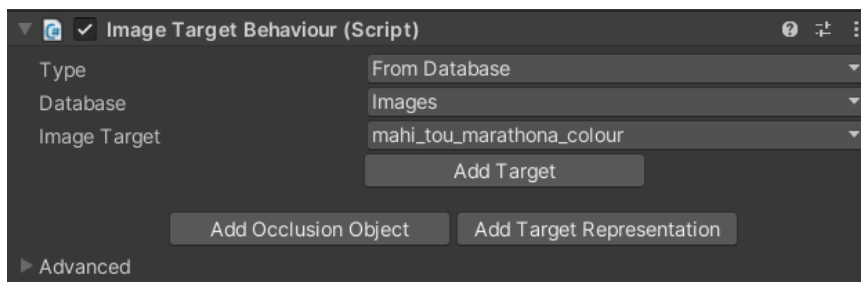
Για την εισαγωγή εικόνων που θα λειτουργούν ως στόχοι, κάνουμε κλικ στη βάση που δημιουργήσαμε και στην συνέχεια επιλέγουμε το Add Target. Από εκεί ανοίγει ένα αναδυόμενο παράθυρο με επιλογές για προσθήκη εικόνας, κυλίνδρου ή πολλαπλών στόχων στη βάση δεδομένων. Πολύ σημαντική επιλογή είναι οι διαστάσεις που παρέχουμε σε έναν στόχο που βασίζεται σε εικόνα, να είναι πραγματικές.

Έπειτα, Διαλέγουμε την εικόνα που θέλουμε να εισάγουμε σαν στόχο και βάζοντας το κατάλληλο όνομα και την αληθινή διάσταση της φωτογραφίας την προσθέτουμε σαν στόχο στην βάση δεδομένων μας. Αυτή η διαδικασία επαναλαμβάνεται κάθε φορά που θέλουμε να εισάγουμε έναν νέο στόχο στην εφαρμογή μας.

3.3.3 Image Target στο Unity

Αφού δημιουργήσαμε τη βάση δεδομένων και τη γεμίσαμε με τις εικόνες στόχους, επόμενο βήμα είναι η προσθήκη Image target στο Unity για την αναγνώριση της βάσης.

Δημιουργούμε ένα νέο GameObject κάνοντας δεξί κλικ στο Hierarchy -> Vuforia Engine -> ImageTarget και μέσα από το inspector αλλάζουμε τις ρυθμίσεις του Image Target Behavior (script). Το Type πρέπει να είναι from database, που αναφέρεται στην βάση που κάναμε import από το Vuforia. Έπειτα, επιλέγουμε την εικόνα που θέλουμε να ορίσουμε ως στόχο ανάλογα με το παιχνίδι που υλοποιούμε (Σχήμα 3.9).

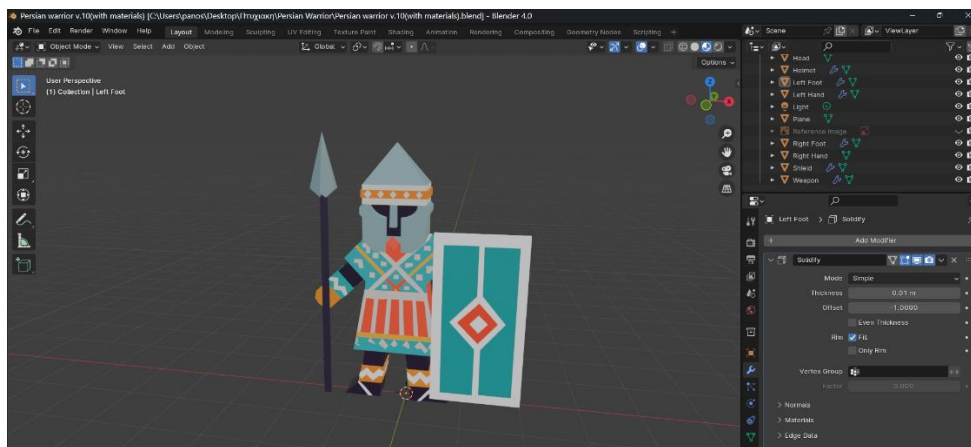


Σχήμα 3.9: Ρυθμίσεις Image Target Behaviour (script)

3.4 Blender & FBX

Κατά τη διάρκεια της ανάπτυξης της εφαρμογής, το Blender ήταν ένα βασικό εργαλείο για την κατασκευή των τρισδιάστατων γραφικών και συγκεκριμένα, στην ανάπτυξη των μοντέλων των στρατιωτών και των πλοίων που χρησιμοποιούνται στο παιχνίδι. Το Blender επειδή είναι μια πλατφόρμα τρισδιάστατης μοντελοποίησης ανοιχτού κώδικα, μας δίνει την δυνατότητα και τον πλήρη έλεγχο στην σχεδίαση 3d γραφικών.

Μέσω του Blender, σχεδιάστηκαν μοντέλα low-poly (Σχήμα 3.10), δηλαδή μοντέλα με σχετικά λιγότερα πολύγωνα, έτσι ώστε να μην παρεμποδίζεται πολύ η απόδοση της εφαρμογής και αλλά ταυτόχρονα να διατηρεί όλες τις ιστορικές λεπτομέρειες στον χαρακτήρα. Με αυτό τον τρόπο, τα γραφικά του παιχνιδιού θα βασίζονται σε έγκυρα ιστορικά στοιχεία και ταυτόχρονα δεν θα επιβαρυνθεί το πρόγραμμα ανεξάρτητα με το πόσα 3D μοντέλα κι αν εισάγουμε.



Σχήμα 3.10: 3d μοντέλο Πέρση στρατιώτη στο Blender

Στη συνέχεια, οι στρατιώτες εφοδιάστηκαν με αρθρώσεις μέσω του συστήματος armature του Blender. Το σύστημα armature επέτρεψε τη δημιουργία χαρακτήρων που μπορούν να κρατούν αντικείμενα.

Τα τελικά μοντέλα εξήχθησαν από το Blender στην μορφή που υποστηρίζει το Unity, την τεχνολογία FBX, επιλέγοντας τις κατάλληλες ρυθμίσεις, ώστε να είναι έτοιμα για χρήση χωρίς περαιτέρω τροποποιήσεις. Αυτή η μεταφορά ενσωματώνει επίσης την αλλαγή μεγέθους για να επιβάλλει καλή απόδοση και να διασφαλίζει ότι η εφαρμογή εκτελείται ομαλά σε σχεδόν οποιαδήποτε συσκευή Android.

3.5 Επίλογος

Συμπερασματικά, σε αυτό το κεφάλαιο αναπτύχθηκαν οι τεχνολογίες και οι πλατφόρμες υλοποίησης, που χρησιμοποιήθηκαν καθώς και κάποιες αρχικές διαδικασίες που εφαρμόστηκαν κατά την υλοποίηση της εφαρμογής στο Unity, το Vuforia Engine και το Blender.

Κεφάλαιο 4ο: Υλοποίηση της εφαρμογής

4.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο θα γίνει περιγραφή των βασικών λειτουργιών της εφαρμογής που επαναχρησιμοποιήθηκαν κατά την υλοποίηση όλων των ασκήσεων-μαχών, όπως επίσης και θα παρουσιαστεί η εφαρμογή στο σύνολο της.

4.2 Βιβλίο Μαθητή & Τετράδιο εργασιών

Το πλαίσιο του εκπαιδευτικού υλικού αναπτύχθηκε με τη χρήση του Βιβλίου της ιστορίας της Δ' τάξης του δημοτικού και του τετραδίου εργασιών, τα οποία και βρέθηκαν σε ηλεκτρονική μορφή από την ιστοσελίδα του Υπουργείου Θρησκευμάτων και Αθλητισμού [27-28]. Επιπλέον, χρησιμοποιήθηκε η ιστοσελίδα Φωτόδεντρο, η οποία είναι η κύρια διαδικτυακή πηγή του Υπουργείου Παιδείας για την εύρεση και την παράδοση ψηφιακού εκπαιδευτικού περιεχομένου στα σχολεία και είναι προσβάσιμο σε όλους [29]. Οι μάχες που υλοποιήθηκαν στην παρούσα Π.Ε επιλέχθηκαν σε πρώτη φάση με βάση τη σημασία τους και την ευκολία στην κατανόηση από την πλευρά του μαθητή. Κατά δεύτερον, επιλέχθηκαν με βάση το διαθέσιμο υλικό που υπήρχε στο βιβλίο σε συνδυασμό με το υλικό που βρέθηκε στον ιστότοπο Φωτόδεντρο αλλά και γενικά στο διαδίκτυο. Κάθε άσκηση-μάχη είναι και μια σκηνή στο Unity. Υλοποιήθηκαν συνολικά τέσσερις σκηνές, η Μάχη του Μαραθώνα, η Μάχη των Θερμοπυλών και η ναυμαχία της Σαλαμίνας και μία σκηνή ακόμα είναι το κεντρικό μενού.

Οι βασικότερες πληροφορίες από το βιβλίο και του τετραδίου εργασιών συνοψίστηκαν στις ερωτήσεις που γίνονται στον χρήστη μέσα από την απεικόνιση UI, που εμφανίζονται κατά την διάρκεια της αφήγησης μιας μάχης. Επιπλέον, η επιλογή και η ανάπτυξη των τρισδιάστατων μοντέλων του εδάφους και των χαρτών έχουν σχεδιαστεί με βάση περιγραφών που γίνονται στο βιβλίο του μαθητή και του οπτικού υλικού από το Φωτόδεντρο [30-33]. Αυτή η προσέγγιση στοχεύει στο να παρέχει στον μαθητή που ασχολείται με την εφαρμογή μια ολοκληρωμένη εμπειρία μάθησης, επιτρέποντας την οπτικοποίηση των γεγονότων καθώς αυτά εκτυλίσσονται και αφηγούνται.

4.3 Εικονίδιο Εφαρμογής

Σε πρώτη φάση έγινε η δημιουργία του εικονιδίου. Με τη βοήθεια της χρήσης AI βρέθηκε ένα επιθυμητό εικονίδιο το οποίο τροποποιήθηκε με τη χρήση της ζωγραφικής της Microsoft, για τη χρήση του συγκεκριμένου εικονιδίου στην εφαρμογή. Για να μας το εμφανίζει η εφαρμογή πρέπει μέσα από το μενού του Build, να μεταβούμε στο File -> BuildSettings-> PlayerSettings -> Player -> DefaultIcon και επιλέγουμε την εικόνα που δημιουργήσαμε. Το εικονίδιο φαίνεται στο Σχήμα 4.1

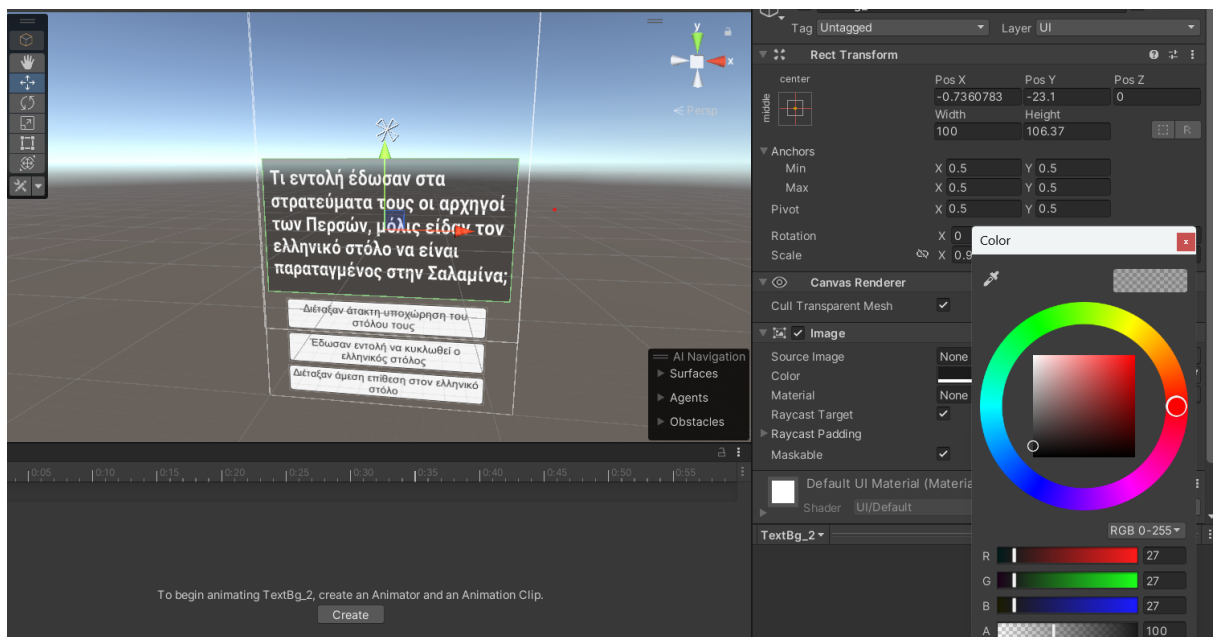


Σχήμα 4.1: Εικονίδιο Εφαρμογής

4.4 Κοινές Διεπαφές χρήστη

Τα UI της εφαρμογής παρουσιάζουν κοινά χαρακτηριστικά όσο το δυνατόν περισσότερο για να υπάρχει ένα αίσθημα συνέπειας και φιλικότητας προς το χρήστη κατά την διάρκεια της χρήσης της εφαρμογής. Τα περισσότερα UI περιλαμβάνουν ένα κείμενο με την χρήση TextMeshPro, το οποίο είναι τοποθετημένο σε ένα background ημιδιαφανές. Για την δημιουργία του background αυτού χρησιμοποιήθηκε ένα αντικείμενο rawImage (UI -> rawImage) και τροποποιήθηκε η ιδιότητα Color του αντικειμένου χρησιμοποιώντας RGBA τιμές με Alpha value 60-200.

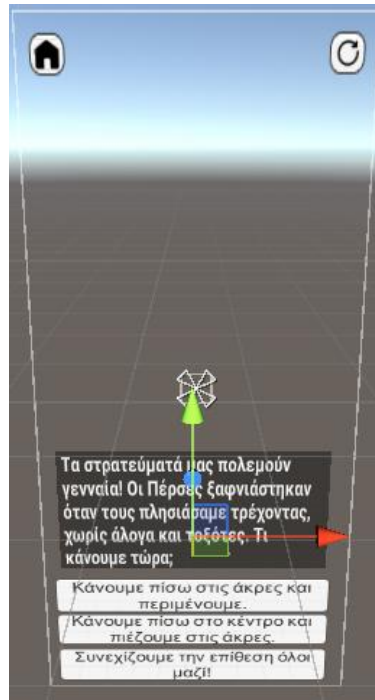
Τα UI που περιέχουν ερωτήσεις περιέχουν κείμενο μαζί με δύο ή τρεις επιλογές πολλαπλής επιλογής μέσω κουμπιών. Το κείμενο που εμφανίζεται πάνω από τα κουμπιά κάνει μια σύντομη περιγραφή στο χρήστη και του παραθέτει την ερώτηση για να επιλέξει, και έπειτα να γίνει έλεγχος της ορθότητας της απάντησης του χρήστη μέσω script (Σχήμα 4.2).



Σχήμα 4.2: Παράδειγμα UI εφαρμογής

4.4.1 Βασικά κουμπιά

Στην επάνω δεξιά πλευρά σε όλα τα UI που παρουσιάζονται στο χρήστη, υπάρχουν δύο κουμπιά. Το κουμπί που πραγματοποιεί επανάληψη της άσκησης σε περίπτωση που ο χρήστης επιθυμεί να ξεκινήσει μία άσκηση από την αρχή. Αριστερά, υπάρχει το κουμπί που ο χρήστης μπορεί να επιστρέψει στην αρχική σκηνή του παιχνιδιού, ενδεχομένως, για να επιλέξει άλλη μάχη (Σχήμα 4.3).

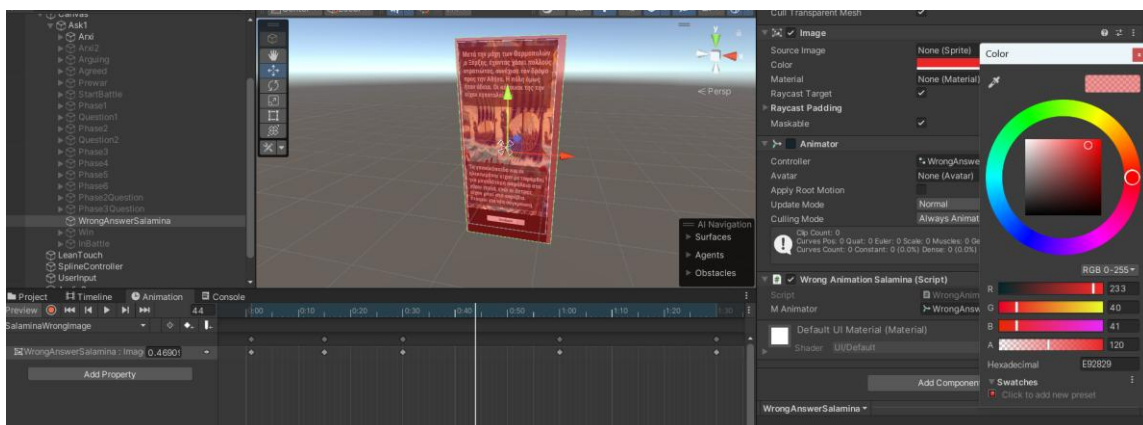


Σχήμα 4.3: Βασικά κουμπιά των UI

Αξίζει να σημειωθεί ότι έξοδο από την εφαρμογή ο χρήστης μπορεί να πραγματοποιήσει από την αρχική σκηνή μέσα από το σύμβολο «X» που υπάρχει στην πάνω δεξιά γωνία της StartScene που θα αναλύσουμε και παρακάτω.

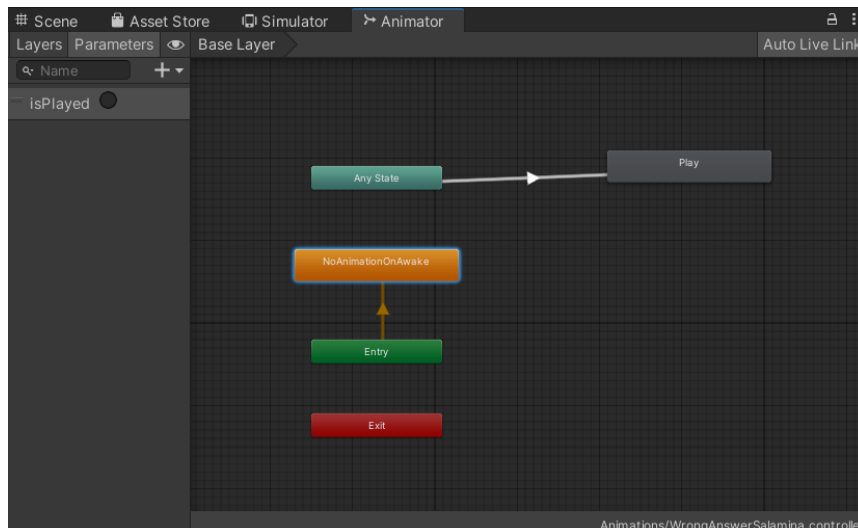
4.4.2 WrongAnswerImage

Είναι σημαντικό ο χρήστης να λαμβάνει ένα μήνυμα από την εφαρμογή, στην περίπτωση που απαντήσει σε κάποια ερώτηση λάθος. Για αυτό το λόγο δημιουργήθηκε το συγκεκριμένο UI που μέσω από ένα animation η οθόνη αναβοσβήνει κόκκινη και παίζεται ταυτόχρονα ένα ηχητικό εφέ. Αρχικά πρέπει να δημιουργηθεί ένα αντικείμενο UI->Image μέσα από το Canvas στο οποίο και βάζουμε κόκκινο χρώμα στην εικόνα. Στην συνέχεια, δημιουργούμε το animation που ρυθμίζουν την ιδιότητα Alpha της κόκκινης εικόνας μέσα από 5 Keyframes. Από αυτά τα keyframes, το πρώτο και το τελευταίο ρυθμίστηκαν σε Alpha με τιμή 0 και τα ενδιάμεσα Keyframe αλλάζουν εναλλάξ το Alpha από 255 σε 0 (Σχήμα 4.4).



Σχήμα 4.4: WrongAnswerImageUI

Επιπρόσθετα, στον animation controller του WrongAnswerImage (Σχήμα 4.5), προστίθεται μια νέα κενή state, η οποία ορίζεται ως default . Αυτό εξασφαλίζει ότι το animation δεν ενεργοποιείται αυτόματα κατά την εκκίνηση (on awake). Στη συνέχεια, δημιουργείται μια διαδρομή (transition) από το Any State στο βασικό animation, ενώ προστίθεται μια παράμετρος τύπου trigger που συνδέεται με αυτή τη διαδρομή. Είναι σημαντικό να σημειωθεί ότι αν οριστεί τιμή -1 στο speed του transition, τότε το animation θα παίζει αντίστροφα.



Σχήμα 4.5: Animator controller WrongAnswerImage

Για τον έλεγχο του animation χρειάζεται να δημιουργηθεί μία μέθοδος στο QuestionHandler script της εκάστοτε άσκησης, το script δηλαδή της βασικής λειτουργίας μίας άσκησης (θα αναλυθεί παρακάτω). Επίσης χρειάστηκε ακόμα ένα script που είναι τοποθετημένο μέσα στο αντικείμενο UI. Η μέθοδος (Σχήμα 4.6) περιλαμβάνει τη λογική για την ενεργοποίηση και την απενεργοποίηση του UI μετά από 2 δευτερόλεπτα. Επιπλέον, μέσω της Invoke("HideWrongAnswerUI", 2f). Το script (Σχήμα 4.7) είναι υπεύθυνο για το reset του trigger του animator, η οποία κάθε φορά που καλείται να παίζει το animation, κάνει true τον trigger και μετά από 2 δευτερόλεπτα τον ξανακάνει false μέσω της μεθόδου Invoke("ResetTrigger", 2f)

```

107 public void ShowWrongAnswerUI()
108 {
109     {
110         wrongAnswer.SetActive(true);
111         Invoke("HideWrongAnswerUI", 2f);
112     }
113 }

```

Σχήμα 4.6: Μέθοδος χρήσης του UI WrongAnswerImage

```

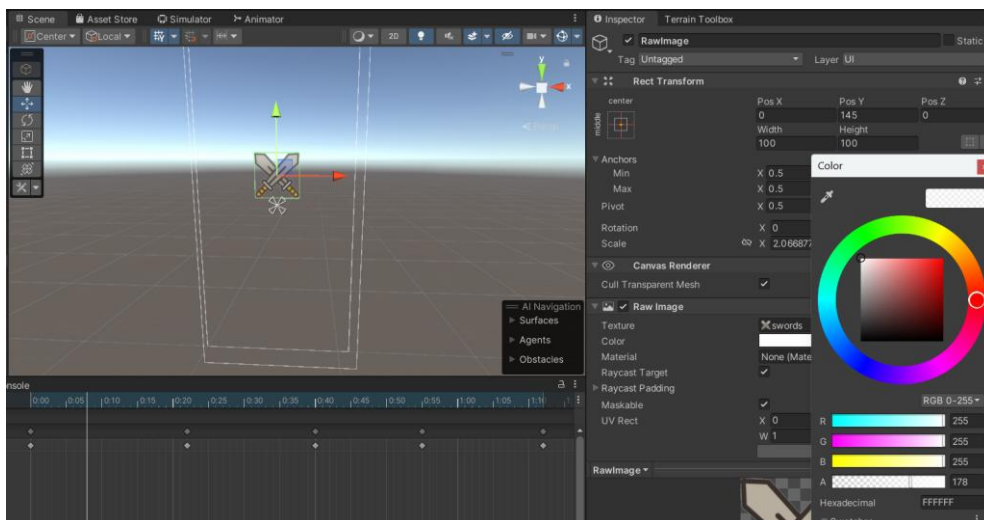
Assembly-CSharp
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  Unity Script (1 asset reference) | 0 references
6  public class WrongAnimationSalamina : MonoBehaviour
7  {
8      public Animator mAnimator;
9      // Start is called before the first frame update
10     void Start()
11     {
12         mAnimator = GetComponent<Animator>();
13     }
14
15     0 references
16     public void ResetTrigger()
17     {
18         mAnimator.ResetTrigger("isPlayed");
19     }
20
21     0 references
22     public void PlayWrongAnimation()
23     {
24         mAnimator.SetTrigger("isPlayed");
25         Invoke("ResetTrigger",2f);
26     }
27
28 }

```

Σχήμα 4.7: Script WrongAnswerImage

4.4.3 InBattle

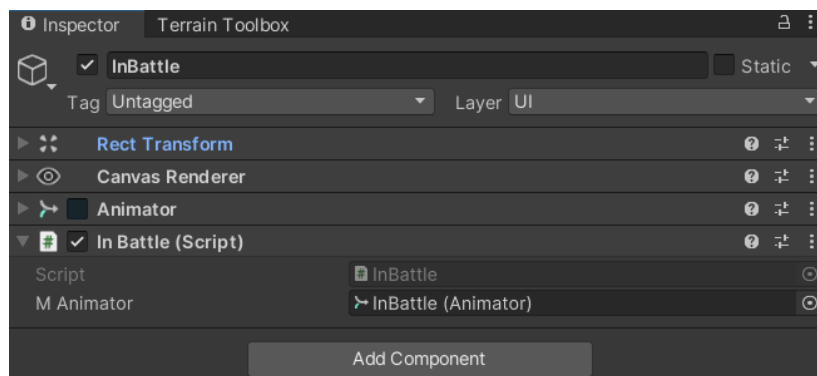
Ένα ακόμα UI που επαναχρησιμοποιείται είναι και το InBattle. Σε παρόμοια λογική με το WrongAnswerImage, δημιουργήθηκε ένα animation, το οποίο περιλαμβάνει μια εικόνα από 2 σπαθιά. Η εικόνα αυτή εμφανίζεται στην οθόνη του χρήστη και αναβοσβήνει όταν τα στρατεύματα μιας άσκησης πραγματοποιούν σύγκρουση. Αρχικά δημιουργήθηκε ένα αντικείμενο rawImage. Έπειτα στο property Texture έγινε εισαγωγή της εικόνας των σπαθιών, που βρέθηκε δωρεάν από το ιντερνετ και τοποθετήθηκε στις κατάλληλες διαστάσεις μέσα στο GameObject UI InBattle. Στην συνέχεια, δημιουργήθηκε το animation που ρυθμίζει την ιδιότητα Alpha της εικόνας με τα σπαθιά βάσει των Keyframes (Σχήμα 4.8).



Σχήμα 4.8: InBattleUI animation

Επιπρόσθετα, στον animator controller του InBattle, προστίθεται μια νέα κενή state προεπιλεγμένη, η οποία ορίζεται ως default . Αυτό εξασφαλίζει ότι το animation δεν ενεργοποιείται αυτόματα κατά την εκκίνηση (on awake). Στη συνέχεια, δημιουργήθηκε μια διαδρομή (transition) από το Any State στο βασικό animation, ενώ προστίθεται μια παράμετρος τύπου trigger που συνδέεται με αυτή τη διαδρομή.

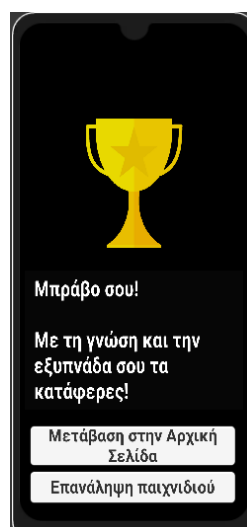
Για τον έλεγχο του animation αυτή τη φορά αναφερόμαστε στη μέθοδο με ονόμα ShowInBattleUI(), που είναι υπεύθυνη για την ενεργοποίηση και την απενεργοποίηση του UI μετά από 2 δευτερόλεπτα που βρίσκεται στο Question Handler της εκάστοτε άσκησης. Το script InBattle που είναι τοποθετημένο μέσα στο InBattleUI GameObject (Σχήμα 4.9), κάνει reset τον Trigger IsPlayed, για να μπορεί να παίζει το animation του UI κάθε φορά που το καλούμε.



Σχήμα 4.9: InBattleUI inspector

4.4.4 Win

Είναι, επίσης, σημαντικό ο χρήστης να λαμβάνει ένα μήνυμα από την εφαρμογή, στην περίπτωση που ολοκληρώνει με επιτυχία κάθε άσκηση-μάχη. Για αυτό το λόγο δημιουργήθηκε το συγκεκριμένο UI (Win) παίζει ένα ηχητικό εφέ και δίνει τη δυνατότητα στο χρήστη να μεταβεί στην αρχική σελίδα ή να επαναλάβει την άσκηση (Σχήμα 4.10).

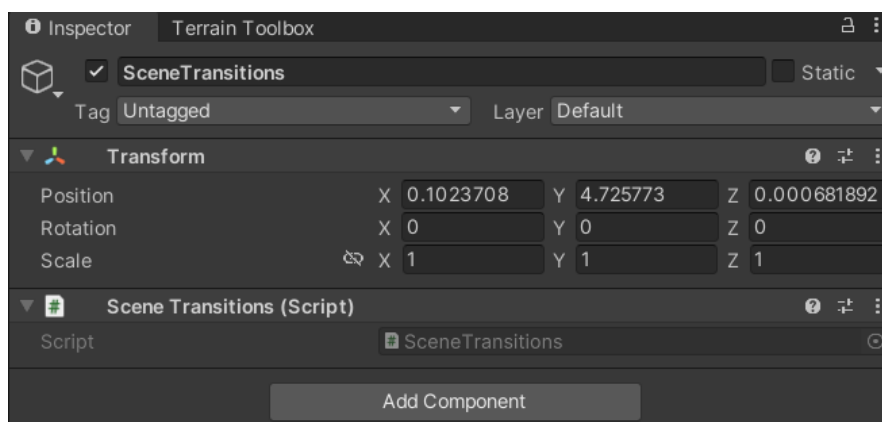


Σχήμα 4.10: WinUI

Οι λειτουργίες των κουμπιών για μετάβαση στην αρχική σελίδα και επανάληψη του παιχνιδιού, αξιοποιούν τις μεθόδους του script SceneTransitions, το οποίο και θα αναλυθεί παρακάτω.

4.5 Κοινά scripts

Κατά τη διάρκεια της υλοποίησης της εφαρμογής δημιουργήθηκαν και κάποια script που στη βάση τους έχουν παρόμοιες λειτουργίες ανάμεσα στις ασκήσεις. Να σημειωθεί ότι, τα script μέσα στο Unity δεν μπορούν να λειτουργήσουν αποτελεσματικά από μόνα τους. Πρέπει να τοποθετηθούν μέσα σε Empty GameObjects (Εικόνα 4.11), δηλαδή άορα αντικείμενα που, όταν τοποθετούνται σε μια σκηνή, παραμένουν μη ανιχνεύσιμα στον χρήστη. Ωστόσο, αποτελούν αναπόσπαστο κομμάτι της υλοποίησης των script. Το script που λειτουργεί μέσω ενός Empty GameObject, ενεργοποιείται όταν ο χρήστης εισέρχεται στην σκηνή και εκτελεί κάποιες βασικές λειτουργίες που θέλουμε στην σκηνή.



Σχήμα 4.11: Τοποθετημένο script μέσα σε Empty GameObject

4.5.1 QuestionController scripts

Τα Script QuestionController σαν ιδέα χρησιμοποιούνται από όλες τις ασκήσεις, απλά με διαφορετικό όνομα και με κάποιες μικρές διαφορές ανά άσκηση.

Όπως υποδεικνύεται από το όνομά τους, λειτουργούν ως διαχειριστές που βοηθούν κατά βάση, στην εμφάνιση και εξαφάνιση των UI του Canvas, τη ροή της μάχης και την εμπλοκή του χρήστη με το περιβάλλον.

Σαν βάση, αρχικοποιούνται οι μεταβλητές περιλαμβάνουν τύπου GameObject που περιλαμβάνουν τα UI του Canvas που θέλουμε να χειριστούμε ανάλογα με την άσκηση. Αυτές οι μεταβλητές περιλαμβάνουν τα UI που αφορούν ερωτήσεις, αυτά που παραθέτουν πληροφορία μέσα από το κείμενο και τη λειτουργία των UI που επαναχρησιμοποιούνται, που αναφέρθηκαν παραπάνω.

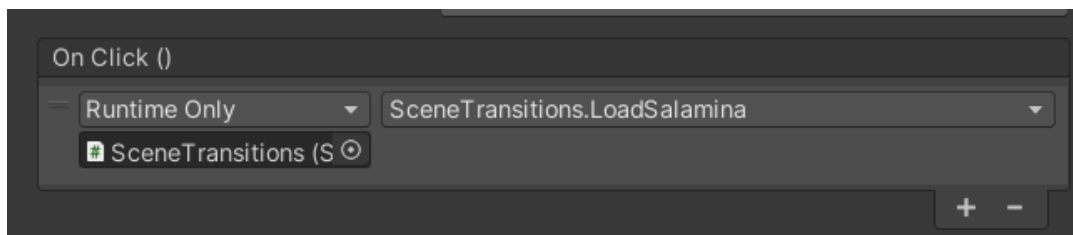
Επίσης, αρχικοποιούνται μεταβλητές τύπου AudioClip για ηχητικά εφέ που ενεργοποιούνται για σωστές ή λάθος απαντήσεις, για ηχητικά εφέ που παίζουν καθ' όλη τη διάρκεια των ασκήσεων και οι φωνητικές αφηγήσεις. Όλες οι μέθοδοι είναι Void, που σημαίνει ότι δεν επιστρέφουν καμία τιμή κατά την εκτέλεση και οι περισσότερες είναι δημόσιες για να επιτρέπουν την πρόσβαση εκτός του Script. Στη μέθοδο Έναρξη, αρχικοποιούνται οι τιμές που καλούνται κατά την εκκίνηση μίας άσκησης. Όλες οι μεταβλητές έχουν οριστεί ως δημόσιες αφού αφορούν αντικείμενα εντός της σκηνής.

Όσον αφορά τις μεθόδους, οι περισσότερες αφορούν ενεργοποιήσεις (setActive(true)) και απενεργοποιήσεις (setActive(false)) των UI του Canvas που αρχικοποιήθηκαν.

Τέλος, για τη διαχείριση του Timeline υπάρχουν οι μέθοδοι ResumeTimeline(), SlowMotionTimeline(), playTimeline() και PauseTimeline(). Εκτός από τις μεθόδους, στις μάχες των Θερμοπυλών και του Μαραθώνα χρησιμοποιείται η βιβλιοθήκη UnityEngine.Playables, η οποία επιτρέπει τη διαχείριση των PlayableDirectors ή στοιχείων που σχετίζονται με το Timeline.. Συγκεκριμένα, η μεταβλητή του τύπου PlayableDirector χρησιμεύει για την διαχείριση του Timeline, από τις ασκήσεις που το χρησιμοποιούν. Χαρακτηριστικά και λειτουργίες που είναι συγκεκριμένες για μία άσκηση, αναφέρονται παρακάτω.

4.5.2 SceneTransitions script

Για να ενεργοποιηθεί η λειτουργικότητα των κουμπιών και να διευκολυνθούν οι μεταβάσεις σκηνής, ήταν απαραίτητο να αναπτυχθεί ένα script με το όνομα SceneTransitions. Αυτό το script, έχει τοποθετηθεί στο φάκελο Scripts των Assets του Unity για μελλοντική συμπερίληψη και μετέπειτα χρήση. Το προβλεπόμενο script ενσωματώνεται ως στοιχείο σε ένα Empty GameObject που δημιουργήσαμε με το ίδιο όνομα και το «περνάμε» ως μεταβλητή στις προϋπάρχουσες μεθόδους OnClick() των κουμπιών της σκηνής μας που θέλουμε να διαχειριστούν σκηνές (Σχήμα 4.12).



Σχήμα 4.12: Ενσωμάτωση script SceneTransitions στο onClick() κουμπιού

Το script SceneTransitions (Σχήμα 4.13) διαχειρίζεται τις μεταβάσεις μεταξύ σκηνών. Κάθε μία άσκηση είναι και μία σκηνή, επομένως και το script μας περιέχει τις μεθόδους φόρτωσης των μαχών. Οι μέθοδοι LoadStartScene(), LoadBattleOfMarathon(), LoadBattleOfThermopylae(), LoadSalamina(), χρησιμοποιούνται στα κατάλληλα κουμπιά του αρχικού μας μενού MainMenuUI για να επιλέγεται η κατάλληλη σκηνή ανάλογα με την επιλογή του χρήστη. Οι μέθοδοι RestartGame() και QuitGame() χρησιμοποιούνται για να υπάρχει η δυνατότητα επανεκκίνησης και εξόδου από την άσκηση-μαχη. Με αυτόν τον τρόπο δημιουργήσαμε έναν κώδικα που παρέχει έναν δομημένο τρόπο πλοήγησης και παρουσίασης όχι μόνο για την παρούσα σκηνή αλλά και για τις υπόλοιπες.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5  public class SceneTransitions : MonoBehaviour
6  {
7      public void LoadBattleOfMarathon()
8      {
9          SceneManager.LoadScene("Marathonas");
10     }
11
12     public void LoadBattleOfThermopylae()
13     {
14         SceneManager.LoadScene("Thermopylae battle");
15     }
16
17     public void LoadSalamina()
18     {
19         SceneManager.LoadScene("Salamina");
20     }
21
22     public void LoadStartScene()
23     {
24         SceneManager.LoadScene("StartScene");
25     }
26
27     public void RestartGame()
28     {
29         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
30     }
31
32     public void QuitGame()
33     {
34         Application.Quit();
35     }
36 }

```

Σχήμα 4.13: Script αλλαγής σκηνών SceneTransitions

4.6 Audio Source

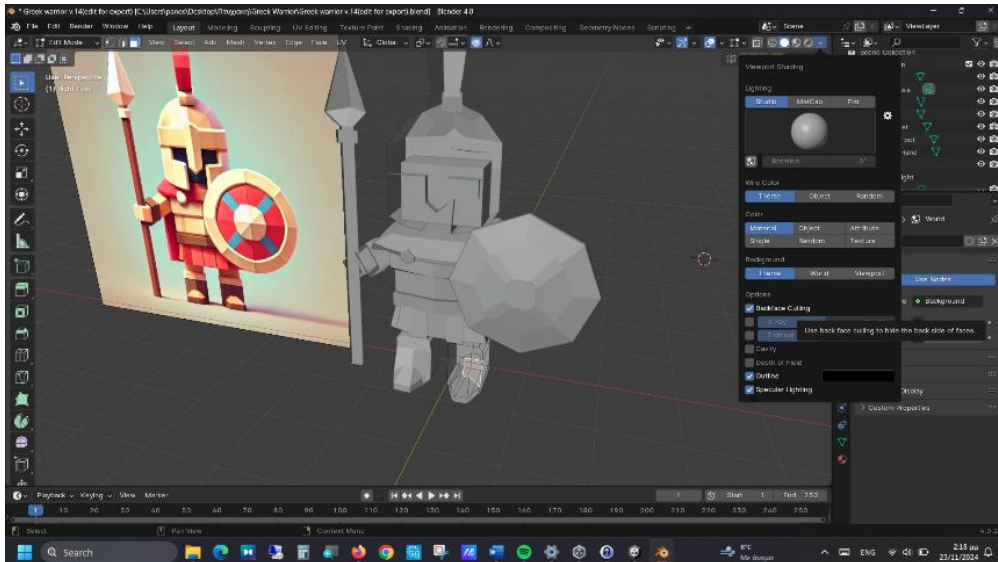
Όλες οι ασκήσεις έχουν ένα μουσικό υπόβαθρο είτε σαν background είτε σαν ένα απλό ηχητικό εφέ (π.χ. σωστή-λάθος απάντηση, κ.α.). Η ένταξη της μουσικής σε κάθε σκηνή γίνεται με τη βοήθεια της τεχνολογίας Audio Source και πάλι πάνω σε ένα empty GameObject. Μέσα από αυτήν τοποθετούμε το ηχητικό εφέ ή τη μουσική υπόκρουση της εφαρμογής.

4.7 Δημιουργία 3D μοντέλων μέσω Blender

Το Blender χρησιμοποιήθηκε, για τη δημιουργία των τρισδιάστατων μοντέλων που αναπαριστούν τα πλοία και τους στρατιώτες των ασκήσεων, όπως επίσης και για την αναπαράσταση του χάρτη της ναυμαχίας της Σαλαμίνας.

4.7.1 3D Στρατιώτες

Η δημιουργία των στρατιωτών δημιουργήθηκε με τη χρήση εικόνας αναφοράς, που δημιουργήθηκε με AI. Ας πάρουμε για παράδειγμα τον Έλληνα πολεμιστή. Με τη χρήση ενός single vertex, που προήλθε από ένα plane αντικείμενο, ξεκίνησε η διαμόρφωση του χαρακτήρα με βάση την εικόνα αναφοράς. Ειδική προσοχή δόθηκε στην ευθυγράμμιση των σημείων (align) με το σκελετό του χαρακτήρα. Έπειτα, σε συνδυασμό των εργαλείων extrude, δημιουργήσαμε το σκελετό του στρατιώτη και στη συνέχεια με τον κατάλληλο solidify modifier και με τη χρήση του εργαλείου Fill του Blender, δώσαμε σώμα στο στρατιώτη μας (Σχήμα 4.14). Με τον ίδιο τρόπο δημιουργήθηκε και ο Πέρσης στρατιώτης.

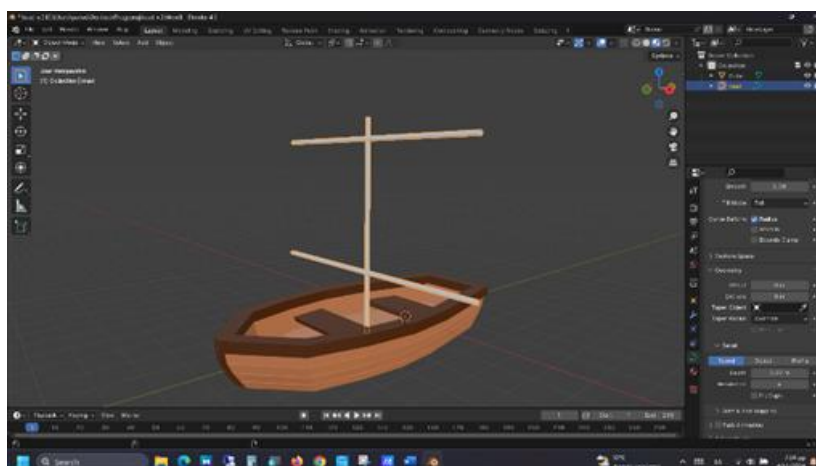


Σχήμα 4.14: Δημιουργία Έλληνα στρατιώτη στο Blender

4.7.2 3D Πλοία

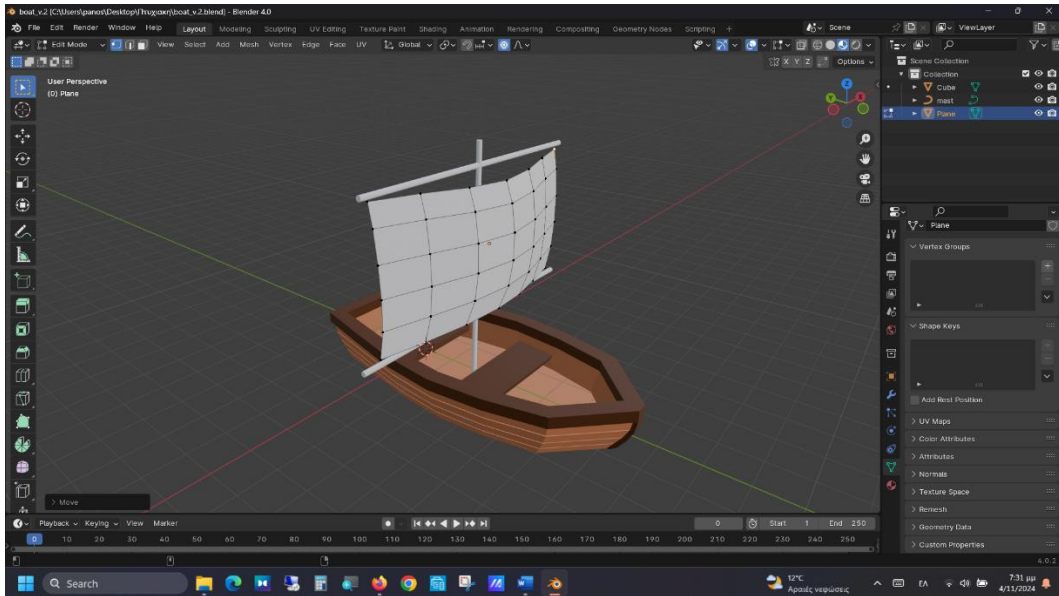
Η διαδικασία δημιουργίας του πλοίου στο Blender ξεκινά με τη δημιουργία μιας βάρκας μέσα από τη χρήση του βασικού κύβου του Blender. Ο κύβος τροποποιήθηκε μέσω των εργαλείων scale και extrude για να πάρει τη μορφή της βάρκας. Στην συνέχεια προστέθηκαν λεπτομέρειες στο πλοίο με το εργαλείο loop cut (Ctrl+R), για να διαμορφώσουν τις καμπύλες της βάρκας.

Η διαδικασία για τη δημιουργία των καταρτιών ξεκίνησε με την αντιγραφή ενός vertex (Shift+D) και την τοποθέτησή του στη βάση της βάρκας, ενώ χρησιμοποιήθηκε το extrude στον άξονα z για να σχηματιστούν τα άκρα του βασικού άξονα του καταρτιού. Τα κάθετα μέρη του καταρτιού δημιουργήθηκαν με επανάληψη της διαδικασίας duplication και extrude. Αφού ολοκληρώθηκε το σχήμα, το αντικείμενο μετατράπηκε σε καμπύλη (curve) μέσω του object mode και η επιλογή fill caps εξασφάλισε την πλήρη κάλυψη του σχήματος (Σχήμα 4.15).



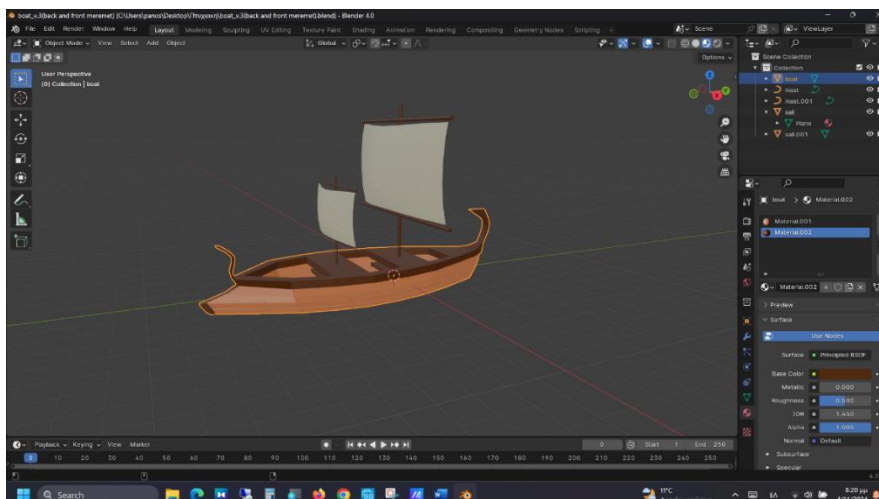
Σχήμα 4.15: Κατάρτι της βάρκας

Για τη δημιουργία των πανιών, προστέθηκε ένα επίπεδο (plane) σε object mode και τοποθετήθηκε μπροστά από το κατάρτι. Έπειτα, χρησιμοποιήθηκε το proportional editing στο edit mode για να δημιουργηθεί καμπύλη, προσαρμόζοντας τα κεντρικά vertices (Σχήμα 4.16).



Σχήμα 4.16: Πανί της βάρκας

Έπειτα, για να μετατρέψουμε τη βάρκα μας σε πλοίο προσθέσαμε πλώρη και πρύμνη. Η πλώρη και η πρύμνη του πλοίου δημιουργήθηκαν με μετακινήσεις των faces και τη χρήση εργαλείων όπως το loop cut, proportional editing, extrude, και scale. Επίσης προστέθηκε ακόμα ένα κατάρτι με πανί, ώστε να ανταποκρίνονται στα ιστορικά χαρακτηριστικά των πλοίων εκείνης της εποχής (Σχήμα 4.17).



Σχήμα 4.17: Πλώρη και η πρύμνη του πλοίου

Τέλος το πλοίο ολοκληρώθηκε με τη δημιουργία των κουπιών. Τα κουπιά δημιουργήθηκαν εισάγοντας έναν κύβο, τον οποίο διαμορφώσαμε με *scale*, *extrude* και *move*. Μετά, έγινε πολλαπλασιασμός (*duplicate*) και ομαδοποίηση (*join*) για τη σωστή διάταξη σε δεξιά και αριστερή πλευρά (Σχήμα 4.18).



Σχήμα 4.18: Τελική όψη του πλοίου

4.7.3 Υφές & Εξαγωγή 3D μοντέλων

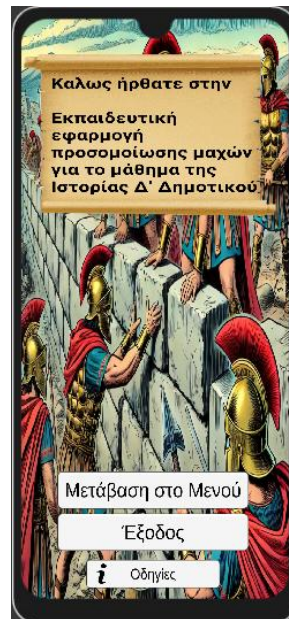
Η διαδικασία δημιουργίας των στρατιωτών και των πλοίων ολοκληρώνεται με την προσθήκη των κατάλληλων υλικών με τη χρήση των *material properties* του Blender. Εφαρμόσαμε απλά χρώματα, χωρίς περίπλοκες υφές και πραγματοποιήθηκε η εξαγωγή των μοντέλων σε μορφή FBX για την εισαγωγή τους και τη χρήση τους στο Unity σαν *prefabs*.

4.8 Αρχικό Μενού (StartScene)

Η εφαρμογή διαθέτει ένα κεντρικό μενού από το οποίο οι χρήστες καλωσορίζονται και μπορούν να επιλέξουν τη άσκηση-μάχη που επιθυμούν μέσω των επιλογών-κουμπιών που εμφανίζουν τόσο το όνομα της μάχης όσο και την αντίστοιχη εικόνα που απαιτείται για την έναρξη της μάχης.

4.8.1 WelcomeUI

Μόλις ο χρήστης ανοίξει την εφαρμογή θα παρουσιαστεί στο χρήστη το Welcome UI, το οποίο διαθέτει ένα «μπάνερ» σε στυλ πάπυρου που επεξηγεί το θέμα που έχει η συγκεκριμένη εφαρμογή και καλωσορίζει τον χρήστη. Σαν φόντο έχει επιλεγθεί μια εικόνα που χρησιμοποιείται και στην ναυμαχία της Σαλαμίνας, συνοδευόμενο από το UI και 2 κουμπιά, Μετάβαση στο Μενού, Έξοδος και Οδηγίες (Σχήμα 4.19).



Σχήμα 4.19: WelcomeUI

Για να ξεκινήσουμε την δημιουργία, μετονομάζουμε την προεπιλεγμένη σκηνή σε StartScene. Στη συνέχεια, πρέπει να δημιουργήσουμε ένα θυγατρικό αντικείμενο Canvas για να μπορέσει το Unity να εμφανίζει στον χρήστη τις διεπαφές UI που θα δημιουργήσουμε. Επιλέγουμε λοιπόν στο παράθυρο Hierarchy, με δεξί κλικ UI -> Canvas. Έπειτα, σαν child του Canvas, για την προσθήκη απλού φόντου προσθέτουμε ένα rawImage , κάνοντας δεξί κλικ πάνω στο Canvas και UI-> RawImage. Στον inspector, του RawImage σαν StartUprImage επιλέγουμε την εικόνα που δημιουργήσαμε σαν φόντο του WelcomeUI.

Συμπερασματικά, η διεπαφή χρήστη (UI) βελτιώθηκε με την ενσωμάτωση τριών κουμπιών, καθένα από τα οποία εξυπηρετεί ξεχωριστές λειτουργίες: έξοδο από την εφαρμογή, εμφάνιση πληροφοριών και μετάβαση στο Μενού επιλογής μαχών. Το κείμενο σε αυτά τα κουμπιά αποδόθηκε χρησιμοποιώντας το TextMeshPro στην ελληνική γλώσσα, χρησιμοποιώντας τη γραμματοσειρά RobotoBold. Αυτή η ίδια γραμματοσειρά εφαρμόστηκε με συνέπεια στο κείμενο σε όλα τα κουμπιά. Η εισαγωγή κουμπιών γίνεται πάλι μέσα από το Canvas ->UI-> Button.

4.8.2 MainMenuUI

Αυτή η διεπαφή χρήστη (UI) είναι το μενού επιλογής μάχης που δημιουργήθηκε για να μπορεί ο εκάστοτε χρήστης να πλοηγηθεί στις μάχες-ασκήσεις που επιθυμεί. Στην κορυφή του μενού, είναι γραμμένος ο τίτλος «ΜΕΝΟΥ ΕΠΙΛΟΓΗΣ ΜΑΧΗΣ», τοποθετημένος σε φόντο σαν πάπυρο, ενισχύοντας έτσι την ιστορική φύση της εφαρμογής. Κάτω από τον τίτλο, υπάρχει το κουμπί με το "Οδηγίες" το οποίο ενεργοποιεί το InstructionsUI και θα το αναλύσουμε παρακάτω.

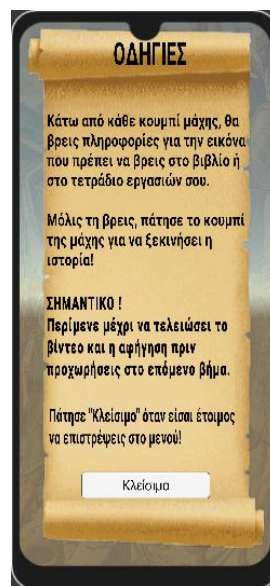
Το UI αποτελείται από τις τρεις κύριες επιλογές μάχης: τη μάχη του Μαραθώνα, τη μάχη των Θερμοπυλών και τη Ναυμαχία της Σαλαμίνας. Κάθε επιλογή συνοδεύεται από μια εικόνα που αντιπροσωπεύει το ImageTarget που πρέπει να σαρώσει ο χρήστης, με διαθέσιμο υποστηρικτικό υλικό αναφοράς για να ξέρει που μπορεί να το βρει ακριβώς στα βιβλία του. Για παράδειγμα, η Μάχη του Μαραθώνα βρίσκεται στο Τετράδιο Εργασιών (Κεφάλαιο 16, σελ. 23), η Μάχη των Θερμοπυλών στο

βιβλίο του μαθητή(Κεφάλαιο 17, σελ. 53) και η Ναυμαχία της Σαλαμίνας επίσης βρίσκεται στο βιβλίο του μαθητή (Κεφάλαιο 18 , σελ. 56).

Τέλος, στο επάνω μέρος της οθόνης, εμφανίζονται δύο εικονίδια. Αυτό που βρίσκεται στα αριστερά μεταφέρει τον χρήστη στην αρχική σελίδα ενώ κάνοντας κλικ στο δεξί (σημειωμένο με το σύμβολο X) πραγματοποιείται έξοδος της εφαρμογής. Τα δύο αυτά κουμπιά καλούν και τις κατάλληλες μεθόδους από το SceneTransitions script που αναφέρθηκαν παραπάνω.

4.8.3 InstructionsUI

Το συγκεκριμένο UI (Σχήμα 4.20) αναφέρεται καθοδήγηση του χρήστη μέσω των οδηγιών. Οι φράσεις επιλέχθηκαν για να παρέχουν όσο το δυνατόν περισσότερο σαφείς οδηγίες για τη χρήση της εφαρμογής πριν γίνει η επιλογή κάποιου παιχνιδιού. Επεξηγεί στο χρήστη ότι και τα τρία κουμπιά επιλογής μάχης έχουν το καθένα πληροφορίες που σχετίζονται με μια εικόνα που πρέπει να βρίσκεται είτε στο σχολικό βιβλίο είτε στο βιβλίο εργασίας για να ξεκινήσει η αφήγηση. Επομένως, ο χρήστης κατευθύνεται να πατήσει το κατάλληλο κουμπί μάχης και στη συνέχεια να ακολουθήσει τις οδηγίες που εμφανίζονται κατά την εκκίνηση της εκάστοτε μάχης. Στο κάτω μέρος υπάρχει μια σημαντική σημείωση, που ενημερώνει το χρήστη πως τα κινούμενα σχέδια και η αφήγηση πρέπει να ολοκληρωθούν πριν συνεχιστεί η αφήγηση. Το κουμπί "Κλείσιμο" στο κάτω μέρος της οθόνης επιτρέπει τελικά στον χρήστη να επιστρέψει στο προηγούμενο μενού που βρισκόταν κάνοντας disable το InstructionsUI.



Σχήμα 4.20: InstructionsUI

4.9 Μάχη του Μαραθώνα

Η άσκηση «Μάχη Του Μαραθώνα» αντλεί έμπνευση από το κεφάλαιο 16 του σχολικού βιβλίου με τίτλο «Ο περσικός κίνδυνος» σε συνδυασμό με το αντίστοιχο κεφάλαιο στο τετράδιο εργασιών. Συγκεκριμένα η συγκεκριμένη μάχη επαυξάνει την άσκηση 2 (Σχήμα 4.21), η οποία ζητάει από τους μαθητές να τοποθετήσουν σωστά τα στρατεύματα και να δώσουν μια εξήγηση για το πώς εξελίχθηκε η μάχη.

2. Ο Μιλτιάδης εφάρμοσε ένα έξυπνο σχέδιο επίθεσης των Ελλήνων κατά των Περσών, γνωστό ως «λαβίδα». Θα μπορούσες, με τη βοήθεια του δασκάλου σου, στις παρακάτω εικόνες να τοποθετήσεις τους δύο στρατούς και να εξηγήσεις πώς εξελίχθηκε η μάχη;

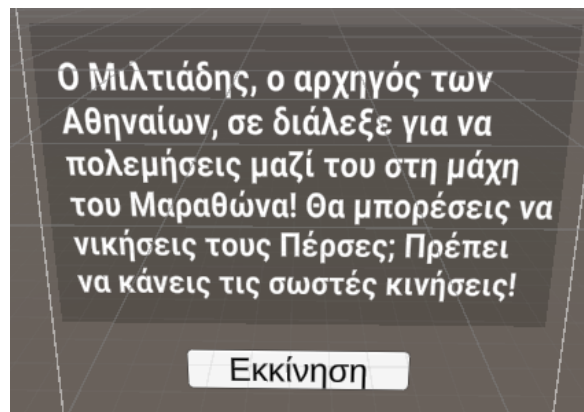


Σχήμα 4.21: Άσκηση μάχης του Μαραθώνα (σελ.23 Τετράδιο εργασιών μαθητή)

Η άσκηση εφαρμόζει το AR με τέτοιο τρόπο ώστε να επιτρέπει στους μαθητές να αλληλεπιδρούν απευθείας με τα στρατεύματα μέσω τρισδιάστατων γραφικών και να βλέπουν εικόνες της στρατηγικής μάχης σε πραγματικό χρόνο και όχι απλά να την «ζωγραφίσουν» μέσα από την άσκηση.

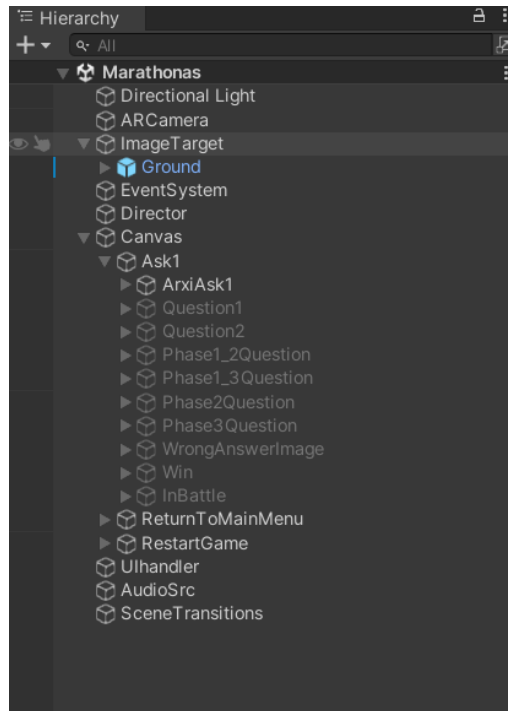
4.9.1 Δομή της άσκησης

Αρχικά, κατά τη σάρωση της φωτογραφίας, μέσω της default μεθοδου On Target Found() του ImageTarget καλείται η μέθοδος DisplayArxikiQuestion() η οποία κάνει enable το UI ArxiAsk1. Η μέθοδος OnTargetFound() καλείται αυτόματα από τον Vuforia Engine όταν ένα Image Target γίνεται ορατό στην κάμερα [34]. Έτσι λοιπόν καλείται και το αρχικό μας UI. Το ArxiAskUI, εισάγει τον χρήστη στην άσκηση με ένα απλό κείμενο σε δεύτερο πρόσωπο για να δημιουργήσει άμεση αλληλεπίδραση, και να εντυπωσιάσει τον χρήστη (Σχήμα 4.22).



Σχήμα 4.22: ArxiAsk1

Στο κάτω μέρος της οθόνης βρίσκεται το κουμπί με την ένδειξη "Εκκίνηση", το οποίο χρησιμεύει στο να απενεργοποιεί το παρόν αρχικό UI ArxiAsk1 του Canvas, όπως φαίνεται στο Hierarchy στο Σχήμα 4.23 και να κάνει enable την πρώτη ερώτηση Question1 μέσω της μεθόδου onClickArxi() του script QuestionHandler (Σχήμα 4.24).



Σχήμα 4.23: Hierarchy της μάχης του Μαραθώνα

```

148 public void OnButtonClickArxi()
149 {
150     HideArxikiQuestion();
151     ShowQuestion1();
152 }
153

```

Σχήμα 4.24: Μέθοδος onButtonClickArxi()

Στη συνέχεια της εμφάνισης της Question1, το κείμενο ζητά από τον χρήστη σε πρώτη φάση να επιλέξει την κατάλληλη τοποθέτηση των στρατευμάτων των Ελλήνων σε σχέση με αυτά των Περσών. Κάτω από το κείμενο υπάρχουν δύο διαθέσιμες τοποθετήσεις των στρατευμάτων, στην μορφή κουμπιών, καθεμία από τις οποίες συνοδεύεται από μια εικόνα που απεικονίζει την συγκεκριμένη επιλογή για να είναι κατανοητό στον χρήστη (Σχήμα 4.25).



Σχήμα 4.25: Κουμπιά Question1 GameObject

Εφόσον ο χρήστης επιλέξει την σωστή τοποθέτηση «Δυνατά στις άκρες, με αποδυνάμωση στο κέντρο», τότε γίνεται και η εμφάνιση των στρατευμάτων και των πλοίων σε τρισδιάστατη μορφή, μέσω της μεθόδου `CorrectBattleground` (Σχήμα 4.26) η οποία κάνει active το έδαφος της μάχης, το οποίο έχει τοποθετημένα πάνω τα στρατεύματα και ενεργοποιεί το επόμενο UI το οποίο με το πάτημα του κουμπιού «Επίθεση!» ξεκινάει την αφήγηση μέσω του Timeline.

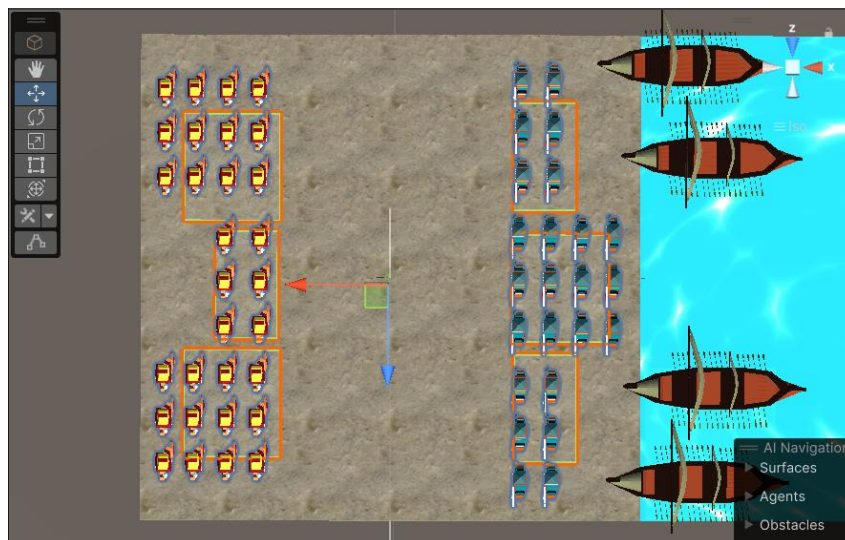
```

234 public void CorrectBattleground()
235 {
236
237     battleground.SetActive(true);
238     ShowQuestion2();
239     question1.SetActive(false);
240
241 }

```

Σχήμα 4.26: μέθοδος `CorrectBattleground`

Όταν ο χρήστης πατήσει το κουμπί τότε, το Timeline ενεργοποιείται, συνοδευόμενο από μια μουσική υπόκρουση που παίζει σαν background κατά τη διάρκεια όλης της μάχης. Για να πετύχουμε αυτά τα δύο αποτελέσματα, αρχικά δημιουργήσαμε τα τάγματα. Τα τάγματα είναι έξι, τρία για τους Έλληνες και 3 Για τους Πέρσες. Για κάθε ένα από τα τάγματα, τοποθετήθηκαν τα prefabs των 3D μοντελων που δημιουργήσαμε στο blender, και αυτά με τη σειρά τους τοποθετήθηκαν σαν child πάνω σε ένα GameObject τύπου Cube, το οποίο και φτιάξαμε σε ορθογώνιο για να θυμίζουν την στρατηγική που ο μαθητής καλείται να ζωγραφίσει στην άσκηση του τετραδίου εργασιών. Σε αυτά τα cubes περάσαμε ένα αόρατο material για να μην είναι ορατά στην χρήση της εφαρμογής (Σχήμα 4.27)



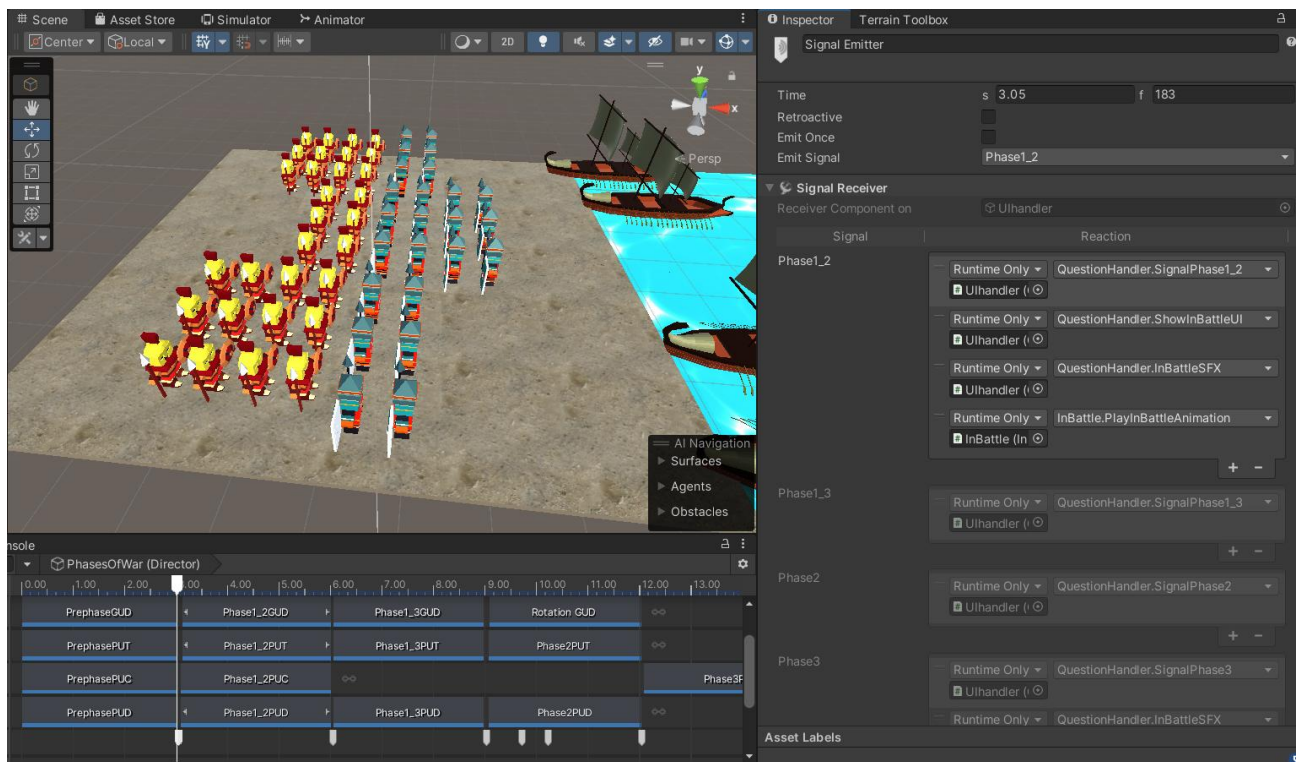
Σχήμα 4.27 Τοποθέτηση ταγμάτων Μάχης του Μαραθώνα & αόρατα Cubes

Οι κύβοι δημιουργήθηκαν ώστε ένα τάγμα να μπορεί να κινηθεί σαν ομάδα, καθώς με βάση τη στρατηγική του Μιλτιάδη, τα τάγματα θα χρειαστεί να κάνουν διαφορετικές κινήσεις στο χρόνο που τρέχει το animation.

Όσον αφορά το συνοδευόμενο ηχητικό εφέ – μουσική υπόκρουση, έχουμε προσθέσει το εργαλείο AudioSource στο empty GameObject με όνομα AudioSrc, το οποίο το αρχικοποιούμε και αυτό σαν μεταβλητή στο script μας και το καλούμε μέσω της μεθόδου BattlePreparations. Για να μπορούμε να διαχειριστούμε το Timeline χρειάστηκε να προσδεθεί η βιβλιοθήκη using UnityEngine.Playables στο script του QuestionHandler

Για τη συνέχεια της εξιστόρησης της μάχης έχουν δημιουργηθεί τρία UI (Phase1_2Question, Phase1_3Question και Phase2) κάθε ένα αφορά και μια διαφορετική φάση της μάχης. Σαν φάση θεωρούμε τις ερωτήσεις που γίνονται προς το χρήστη μέσα από τα UI κατά την διάρκεια τη μάχης.

Ο χειρισμός του Timeline γίνεται με τη χρήση του εργαλείου Signal Receiver το οποίο μας επιτρέπει μέσω των Signal Emitter σε συγκεκριμένες χρονικές στιγμές του Timeline να καλούμε τις κατάλληλες μεθόδους signal από το script. Αυτές ενεργοποιούν το κατάλληλο UI ανάλογα με την ερώτηση που θέλουμε να εμφανίσουμε. (Σχήμα 4.28)



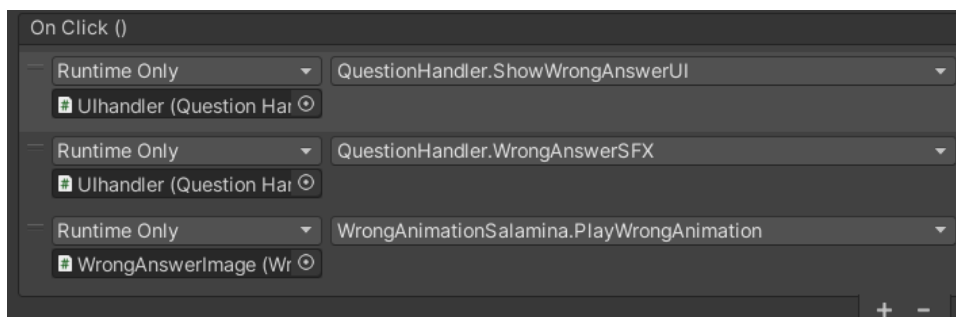
Σχήμα 4.28: Εργαλείο Signal Receiver με τους κατάλληλους Signal Emitters

Με το που πατήσει το κουμπί «Επίθεση!» ο χρήστης θα παρατηρήσει τα 2 στρατεύματα να συγκρούονται. Μόλις συγκρουστούν μέσω του πρώτου Signal emitter με όνομα Phase1_2 ο emitter θα καλέσει τη μέθοδο SignalPhase1_2(), η οποία κάνει pause το timeline και εμφανίζει την πρώτη ερώτηση στον χρήστη (Σχήμα 4.29). Επίσης καλούνται και οι κατάλληλοι μέθοδοι για να γίνει χρήση του InBattle UI με το κατάλληλο ηχητικό εφέ διότι πραγματοποιήθηκε η σύγκρουση των στρατευμάτων



Σχήμα 4.29: Ερώτηση Phase1_2

Η επιλογή του χρήστη εδώ μπορεί να έχει δύο αποτελέσματα, ή να επιλέξει σωστό ή λάθος. Αν επιλέξει κάποια από τις λάθος απαντήσεις (“Κάνουμε πίσω στις άκρες και περιμένουμε”, «Συνεχίζουμε την επίθεση όλοι μαζί») οι λειτουργίες που γίνονται είναι τρεις (Σχήμα 4.30). Αρχικά, μέσω του On Click της λάθος επιλογής (πάτημα λάθος κουμπιού) γίνεται κάλεσμα των μεθόδων WrongAnswerSFX που παίζει το ηχητικό εφέ του λάθους και ShowWrongAnswerUI που κάνει active το UI του Λάθους, που είναι μέρος του QuestionHandler script (Σχήμα 4.31). Κατά δεύτερον, ενεργοποιείται το Animation του WrongAnswer μέσα από τη μέθοδο PlayWrongAnimation του WrongImage script.



Σχήμα 4.30: Επιλογή κουμπιού λάθος απάντησης

```

61 //START----- manipulate wrong answer-----
62 public void ShowWrongAnswerUI()
63 {
64     wrongImageAnimation.SetActive(true);
65     Invoke("HideWrongAnswerUI", 2f);
66 }
67
68 0 references
69 public void HideWrongAnswerUI()
70 {
71     wrongImageAnimation.SetActive(false);
72 }
73
74 0 references
75 public void WrongAnswerSFX()
76 {
77     src.clip = sfw_wrong;
78     src.Play();
79 }
80 //END-----manipulate wrong answer-----

```

Σχήμα 4.31: Μέθοδοι που καλούνται σε περίπτωση λάθους από QuestionHandler

Αν παρατηρήσουμε τον κώδικα, η ShowWrongAnswerUI πραγματοποιεί και μια ακόμα λειτουργία. Μόλις καλεστεί η μέθοδος, τότε αυτή με τη σειρά της μέσω της Invoke (“HideWrongAnswerUI”, 2f), καλεί τη μέθοδο HideWrongAnswerUI η οποία μετά από 2 δευτερόλεπτα απενεργοποιεί το UI. Αυτό το κάνουμε για να μπορεί να πατηθεί η λάθος επιλογή παραπάνω από μία φορές από τον χρήστη. Με έναν τρόπο γίνεται reset το UI της λάθος απάντησης κάθε φορά που ο χρήστης πατάει λάθος.

Εφόσον ο χρήστης πατήσει τη σωστή απάντηση «Κάνουμε πίσω στο κέντρο και πιέζουμε στις άκρες», τότε παίζει το κατάλληλο ηχητικό εφέ της σωστής ερώτησης μέσω της μεθόδου CorrectChoiceSFX και αποκρύπτονται όλα τα UI μέσω της μεθόδου HideAllQuestions (Σχήμα 4.32) . Επίσης με τη σωστή απάντηση του χρήστη καλείται η μέθοδος ResumeTimeline η οποία συνεχίζει το animation των GameObject των στρατευμάτων , και κατά συνέπεια ο χρήστης παρατηρεί το κεντρικό στράτευμα των Ελλήνων να οπισθοχωρεί, ενώ ταυτόχρονα πιέζονται τα ακριανά στρατεύματα των Περσών.

```

186 public void CorrectChoiceSFX()
187 {
188     src.clip = sfx_correct;
189     src.Play();
190 }
191
192 0 references
193 public void ResumeTimeline()
194 {
195     if (timeline != null)
196         timeline.playableGraph.GetRootPlayable(0).SetSpeed(0.7);
197 }
198
199 1 reference
200 public void HideAllQuestions()
201 {
202     if (question1 != null) question1.SetActive(false);
203     if (question2 != null) question2.SetActive(false);
204     if (phase1_2question != null) phase1_2question.SetActive(false);
205     if (phase1_3question != null) phase1_3question.SetActive(false);
206     if (phase2question != null) phase2question.SetActive(false);
207     if (phase3question != null) phase3question.SetActive(false);
208     if (wrongImageAnimation != null) wrongImageAnimation.SetActive(false);
209     if (inBattle != null) inBattle.SetActive(false);
210     if (win_ui != null) win_ui.SetActive(false);
211 }

```

Σχήμα 4.32: Μέθοδοι που καλούνται όταν ο χρήστης απαντήσει σωστά

Οι επόμενες 2 φάσεις Phase1_3 και Phase2 ακολουθούν παρόμοια λογική με την προηγούμενη φάση . Δηλαδή σταματούν το animation μέσω της μεθόδου PauseTimeline και ενεργοποιούν τις αντίστοιχες ερωτήσεις σε μορφή UI την Phase1_3Question (Σχήμα 4.33) και Phase2Question.



Σχήμα 4.33: Phase1_3Question

Οι λειτουργίες των κουμπιών είναι ίδιες με αυτές που αναφέρθηκαν παραπάνω σε περίπτωση σωστού και λάθους. Για να συνεχιστεί η μάχη η σωστή επιλογή στη Phase1_3Question είναι η επιλογή «Αφήνουμε τους Πέρσες στα πλάγια να φύγουν προς τη θάλασσα», τότε ο χρήστης θα παρατηρήσει τα ελληνικά ακριανά στρατεύματα να πιέζονται ακόμα παραπάνω και να φτάνουν στα πλοία τους όπου και ενεργοποιείται το Phase2Question (Σχήμα 4.34)



Σχήμα 4.34: Phase2Question

Η διαφοροποίηση που υπάρχει μεταξύ της Phase2 & της Phase3 είναι το γεγονός ότι βλέπει ο χρήστης τα στρατεύματα των περσών να μπαίνουν στα πλοία και να φεύγουν. Αυτό πραγματοποιείται μέσα από τους signal Emitters First και second row board ship οι οποίοι Emitters, στην ουσία ενεργοποιούν το animation των αντικειμένων των πλοίων που έχουν αρχικοποιηθεί στο script Question Handler. Απαντώντας στην ερώτηση σωστά, ο χρήστης παρατηρεί του Έλληνες να επιτίθενται και να νικούν τους πέρσες που είχαν απομείνει στη μάχη και τέλος εμφανίζεται το Win UI που τον ενημερώνει για το τέλος της άσκησης-μάχης (Σχήμα 4.35).



Σχήμα 4.35: Τέλος πρώτης άσκησης

4.9.2 Επίλογος της άσκησης «Μάχη του Μαραθώνα»

Συμπερασματικά, η άσκηση της Μάχης του Μαραθώνα συνδυάζει τρισδιάστατα γραφικά και τεχνολογίες του Unity και ηχητικά εφέ για να δώσει στους μαθητές μια πιο ολοκληρωμένη εμπειρία για τη στρατηγική που ακολούθησαν οι Έλληνες στη μάχη του Μαραθώνα. Στις φάσεις της μάχης, ο χρήστης πρέπει να κάνει επιλογές που θα επηρεάσουν τα αποτελέσματα της μάχης του Μαραθώνα. Η δομή της άσκησης περιλαμβάνει πολλαπλές φάσεις, ερωτήσεις και δυναμικές σκηνές μάχης, μέσω της χρήσης της τεχνολογίας animation Timeline σε συνδυασμό με τα Signal Emitters. Ταυτόχρονα, προσφέρει τη βάση για τις επόμενες δύο ασκήσεις που ακολουθούν.

4.10 Μάχη των Θερμοπυλών

Η άσκηση «Μάχη των Θερμοπυλών» αντλεί έμπνευση από το κεφάλαιο 17 του σχολικού βιβλίου με ομώνυμο τίτλο. Συγκεκριμένα επαυξάνει την εικόνα του βιβλίου « 1. Η εκστρατεία του Ξέρξη » (Σχήμα 4.36).

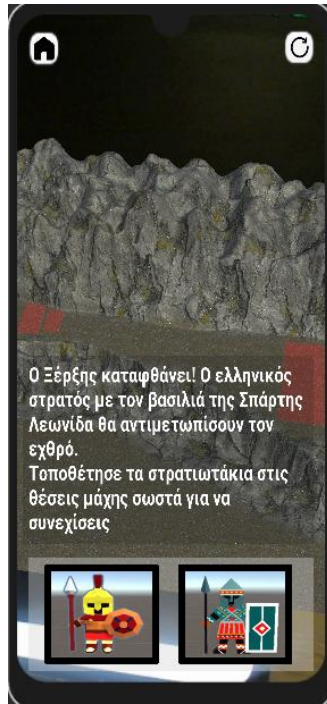


Σχήμα 4.36: Image Target άσκησης της Μάχης των Θερμοπυλών

4.10.1 Δομή της άσκησης «Μάχη των Θερμοπυλών»

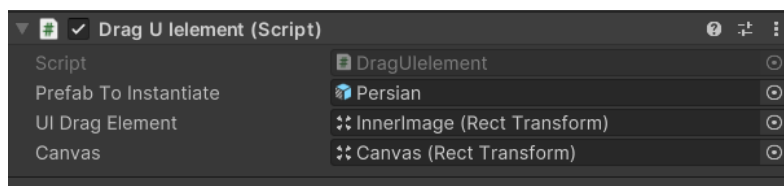
Αρχικά, κατά τη σάρωση της φωτογραφίας, μέσω της default μεθόδου On Target Found() του ImageTarget καλείται η μέθοδος ShowPrePhaseUI η οποία κάνει enable το αρχικό UI το οποίο κάνει μια εισαγωγή στο χρήστη για τη μάχη των Θερμοπυλών. Στο κάτω μέρος της οθόνης βρίσκεται το κουμπί με την ένδειξη "Πάμε για μάχη", το οποίο χρησιμεύει στο να απενεργοποιεί το παρόν αρχικό UI (PrePhaseUI), και να κάνει enable την πρώτη ερώτηση SelectionOfTroopsUI μέσω της μεθόδου OnButtonClickArchi() του script QuestionHandler.

Το SelectionOfTroopsUI εμφανίζει στον χρήστη το έδαφος της άσκησης, το οποίο δημιουργήθηκε με τη βοήθεια του εργαλείου Terrain Sample Asset Pack, και επίσης εμφανίζει δύο περιοχές στο έδαφος που αναβοσβήνουν. Αυτό που ζητάει από το χρήστη είναι να τοποθετήσει στη σωστή μεριά του στενού τον Έλληνα και τον Πέρση στρατιώτη (Σχήμα 4.37).



Σχήμα 4.37: SelectionOfTroopsUI

Οι στρατιώτες εμφανίζονται στο χρήστη με τη βοήθεια ενός Πάνελ του SelectionPanel , που σαν Inner Images έχει τους δύο στρατιώτες. Μέσα σε κάθε InnerImage έχει τοποθετηθεί το script DragUIElement. (Σχήμα 4.38)



Σχήμα 4.38: DragUIElement script

Το συγκεκριμένο script υλοποιεί το μηχανισμό drag-and-drop από το UI στο έδαφος, ανάλογα με τη θέση που τελειώνει το drag για τους στρατιώτες-prefabs που εμφανίζονται στο UI. Αρχικά, γίνεται αρχικοποίηση κάποιων βασικών μεταβλητών, όπως το prefab που θα δημιουργηθεί και η αρχική θέση του UI στοιχείου, του στρατιώτη. Η λειτουργία ξεκινά με τη μέθοδο OnBeginDrag, η οποία καταγράφει την αρχική θέση του στρατιώτη που σύρεται, καθώς και τη θέση του pointer στην οθόνη, δηλαδή που βρίσκεται το δάχτυλο του χρήστη στην οθόνη. Κατά τη διάρκεια του drag, η μέθοδος OnDrag υπολογίζει τη νέα θέση του pointer και μετακινεί το UI στοιχείο ανάλογα. Χρησιμοποιείται η διαφορά

της τρέχουσας θέσης του pointer από την αρχική του θέση για την αναπροσαρμογή της θέσης του στοιχείου (Σχήμα 4.39).

```

void Start()
{
    //we store the initial position of the draggable ui Item
    mOriginalPosition = UIDragElement.localPosition;
}

0 references
public void OnBeginDrag(PointerEventData data)
{
    mOriginalPanelLocalPosition = UIDragElement.localPosition;
    RectTransformUtility.ScreenPointToLocalPointInRectangle(
        Canvas,
        data.position,
        data.pressEventCamera,
        out mOriginalLocalPointerPosition);
}

0 references
public void OnDrag(PointerEventData data)
{
    Vector2 localPointerPosition;
    if (RectTransformUtility.ScreenPointToLocalPointInRectangle(
        Canvas,
        data.position,
        data.pressEventCamera,
        out localPointerPosition))
    {
        Vector3 offsetToOriginal =
            localPointerPosition -
            mOriginalLocalPointerPosition;

        UIDragElement.localPosition =
            mOriginalPanelLocalPosition +
            offsetToOriginal;
    }
}

```

Σχήμα 4.39: Μέθοδοι onBeginDrag() και OnDrag()

Όταν τελειώνει το drag, η μέθοδος OnEndDrag επαναφέρει σταδιακά το UI στοιχείο, το rawImage που δείχνει τον στρατιώτη, στην αρχική του θέση. Παράλληλα, εκτελείται ένα raycast για να εντοπιστεί αν υπάρχει κάποιο αντικείμενο στον κόσμο κάτω από τη θέση του pointer, δηλαδή αν υπάρχει ένα αντικείμενο με collider για να τοποθετηθεί ο στρατιώτης πάνω. Αν εντοπιστεί σημείο πρόσκρουσης, δημιουργείται με τη χρήση της μεθόδου Instantiate ένα νέο αντικείμενο (το prefab στρατιώτης) σε αυτή τη θέση του pointer (Σχήμα 4.40).

```

63 public void OnEndDrag(PointerEventData data)
64 {
65     StartCoroutine(
66         Coroutine_MoveUIElement(
67             UIDragElement,
68             mOriginalPosition,
69             0.5f));
70     RaycastHit hit;
71     Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
72     if (Physics.Raycast(ray, out hit, 1000.0f))
73     {
74         Vector3 worldpoint = hit.point;
75         CreateObject(worldpoint);
76     }
77 }
78 }
79 }
80 }
81 }
82 1 reference
83 void CreateObject(Vector3 pos)
84 {
85     //here we instantiate the object
86     if (prefabToInstantiate==null)
87     {
88         Debug.Log("NO prefab to instantiate");
89         return;
90     }
91     GameObject obj = Instantiate(
92         prefabToInstantiate,
93         pos,
94         Quaternion.Identity);
95 }
96 }

```

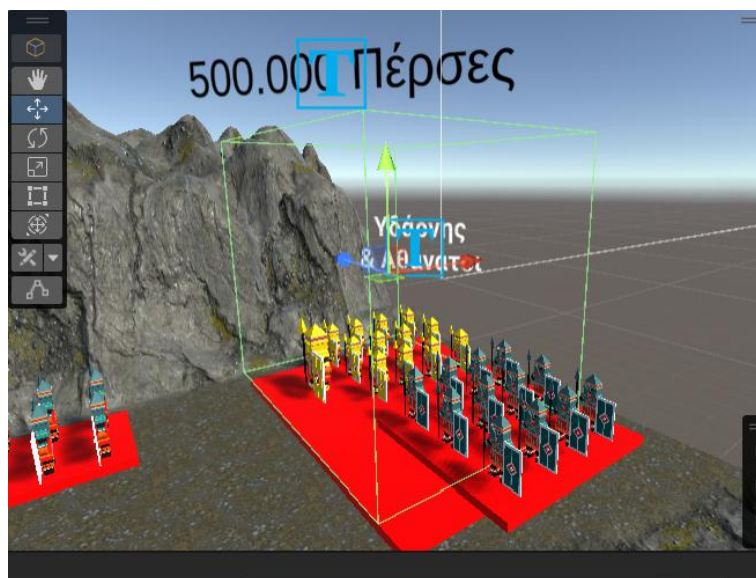
Σχήμα 4.40: Μέθοδοι onEndDrag() και instantiate του prefab

Το script ενσωματώνει επίσης μια βοηθητική μέθοδο για την επαναφορά θέσης του στοιχείου (Coroutine_MoveUIElement), η οποία μετακινεί σταδιακά το στοιχείο πίσω στην αρχική του θέση στο UI.

Το script που μόλις περιγράψαμε πραγματοποιεί απλά τη λειτουργία drag and drop του στρατιώτη χωρίς να ελέγχει αν έχει τοποθετηθεί στο κατάλληλο σημείο πάνω στο έδαφος, δηλαδή στο σημείο που αναβοσβήνει στο έδαφος. Για να το πετύχουμε αυτό χρειαστήκαμε πάνω από τα σημεία αυτά να βάλουμε από έναν collider. Ο collider αυτός τόσο για τους Πέρσες όσο και για τους Έλληνες περιέχουν από ένα script με όνομα SelectionOfPersians και SelectionOfGreeks αντίστοιχα.

Ας πάρουμε σας παράδειγμα το script SelectionOfGreeks.cs το οποίο έχει ως σκοπό τη διαχείριση της επιλογής και της εμφάνισης του τάγματος των Ελλήνων στο στενό των Θερμοπυλών, στην περίπτωση που ο χρήστης τοποθετήσει τον κατάλληλο στρατιώτη στον κατάλληλο collider. Στο Σχήμα 4.41 παρατηρούμε με πράσινο πλαίσιο τον collider και από κάτω είναι το τάγμα των Περσών.

Αρχικά η μέθοδος Start αρχικοποιεί τα αντικείμενα. Το greekSoldiers αντικείμενο, που αντιπροσωπεύει τους στρατιώτες των Ελλήνων που είναι ήδη τοποθετημένοι με τη σωστή παράταξη στο έδαφος, ξεκινά ως ανενεργό, ενώ το greekCollider ενεργοποιείται, επιτρέποντας την ανίχνευση των συγκρούσεων. Αυτή η διαμόρφωση εξασφαλίζει ότι οι στρατιώτες δεν είναι ορατοί έως ότου ανιχνευθεί ένα trigger.



Σχήμα 4.41: Περσικός collider και τάγμα Περσών

Η μέθοδος OnTriggerEnter ενεργοποιείται όταν κάποιο αντικείμενο εισέλθει στον χώρο του collider που σχετίζεται με το script. Κάθε αντικείμενο που γίνεται instantiate έχει και από ένα συγκεκριμένο tag όνομα. Αν ο χρήστης τοποθέτησε σωστά το ελληνικό στράτευμα πάνω από το σωστό collider εκτελείται μια σειρά ενεργειών.

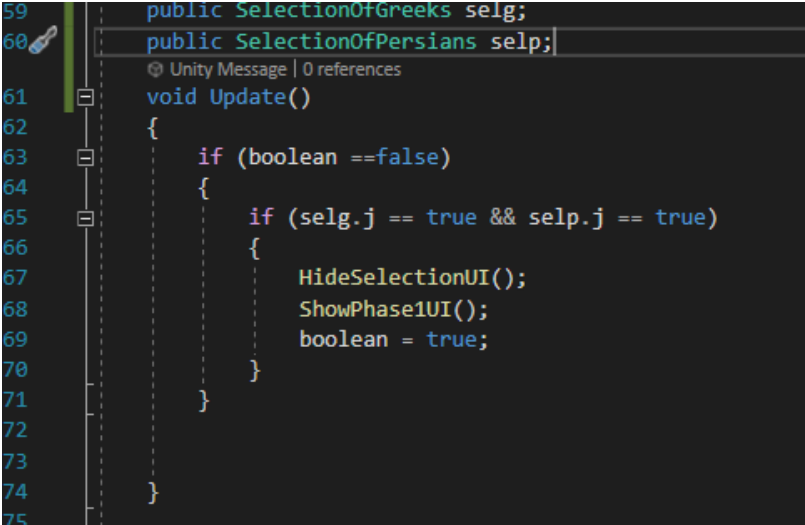
Σε πρώτη φάση εκτελείται μια foreach που αναφέρει ότι για κάθε αντικείμενο με το tag "GreekWarrior" που βρίσκεται πάνω στο έδαφος πρέπει με κάποιο τρόπο να απενεργοποιηθεί. Εννοούμε τα prefabs από το drag and drop που είχε κάνει ο χρήστης πάνω στο έδαφος και δεν είναι τοποθετημένα πάνω στο σωστό collider άρα και παραμένουν στο έδαφος μας χωρίς να έχουν λόγο ύπαρξης. Αυτή τη λειτουργία

την πετυχαίνουμε με τη μέθοδο `GameObject.FindGameObjectsWithTag` όπου εντοπίζει όλα τα αντικείμενα με το tag "GreekWarrior" και τα απενεργοποιεί.

Έπειτα εκτελούνται οι παρακάτω λειτουργίες:

- Το `greekArmy`, που αντιπροσωπεύει την εικόνα του selection από το πάνελ του `SelectionOfTroopsUI` απενεργοποιείται.
- Το `greekSoldiers` ενεργοποιείται, κάνοντας ορατούς τους στρατιώτες τοποθετημένους σε παράταξη.
- Το `greekCollider` απενεργοποιείται, αποτρέποντας περαιτέρω ανιχνεύσεις trigger.
- Το `AudioSource` φορτώνει το ηχητικό clip `sfx_correct` και παίζει το ηχητικό εφέ της σωστής επιλογής.

Η boolean μεταβλητή `j` τίθεται σε `true`, για να σηματοδοτήσει την ολοκλήρωση της διαδικασίας αλλά και για να περαστεί για έλεγχο στο script `QuestionHandlerThermopylae`. Μόλις λοιπόν ο χρήστης τοποθετήσει και τα δύο prefabs στα σωστά colliders τότε μπορεί να συνεχιστεί η άσκηση. Ο έλεγχος αυτός πραγματοποιείται μέσα στη μέθοδο `update` του script `QuestionHandlerThermopylae`. Αυτή η μέθοδος ελέγχει συνέχεια αν έχουν γίνει `true` οι δύο boolean μεταβλητές των script `SelectionOfPersians` και `SelectionOfGreeks`. Εφόσον έχουν γίνει, τότε αυτό σημαίνει ότι τα στρατεύματα έχουν τοποθετηθεί σωστά (Σχήμα 4.42).



```

59 public SelectionOfGreeks selg;
60 public SelectionOfPersians selp;
61
62 void Update()
63 {
64     if (boolean == false)
65     {
66         if (selg.j == true && selp.j == true)
67         {
68             HideSelectionUI();
69             ShowPhase1UI();
70             boolean = true;
71         }
72     }
73 }
74
75

```

Σχήμα 4.42: Μέθοδος ελέγχου των booleans του SelectionUI

Μόλις λοιπόν ο χρήστης τοποθετήσει τα στρατεύματα στα σωστά σημεία τότε εμφανίζεται το Phase1UI. Στο κάτω μέρος της οθόνης βρίσκεται το κουμπί με την ένδειξη "Επίθεση!", το οποίο χρησιμεύει στο να απενεργοποιεί το παρόν αρχικό UI `ArchiAsk1` του Canvas, και να ξεκινήσει η αφήγηση μέσω του timeline.

Η διαφοροποίηση της συγκεκριμένης άσκησης με την προηγούμενη είναι ότι η συγκεκριμένη άσκηση έχει και φωνητική αφήγηση εκτός από τα UI που παρέχουν πληροφορίες και Ερωτήσεις. Η αφήγηση, λοιπόν, ξεκινά, δείχνοντας τα στρατεύματα του Ξέρξη να πλησιάζουν τα στρατεύματα των Ελλήνων μέσα από το στενό πέρασμα των Θερμοπυλών. Στη συνέχεια, γίνεται και η πρώτη παύση του timeline, με την ίδια λογική που γινόταν και στο πρώτο παιχνίδι. Η πρώτη ερώτηση προς το χρήστη είναι το `Question1UI` όπου, ο χρήστης ερωτάται τι έκαναν τα στρατεύματα των Ελλήνων στη συνέχεια, με δύο

πιθανές απαντήσεις: "Προσποίηση υποχώρησης και ξαφνική αντεπίθεση" ή "Παραδόθηκαν στους Πέρσες χωρίς να δώσουν μάχη." Ο στόχος είναι να αναγνωριστεί η σωστή στρατηγική των Ελλήνων που τους οδήγησε σε επιτυχή άμυνα (Σχήμα 4.43).

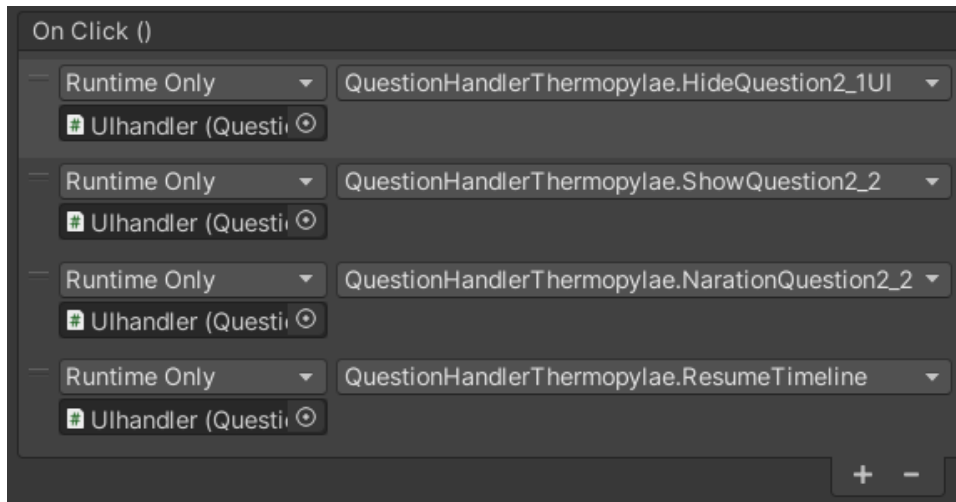


Σχήμα 4.43: Question1UI Μάχη των Θερμοπυλών

Με τη σωστή επιλογή του χρήστη, αποδεικνύεται μέσα από το animation αλλά και του αφηγητή η στρατηγική της προσποίησης υποχώρησης από τους Έλληνες και της ξαφνικής αντεπίθεσης.

Εφόσον ο χρήστης έχει δει και ακούσει τη στρατηγική των Ελλήνων η εφαρμογή του ζητάει μέσα από μια σειρά ερωτήσεων να επαναλάβει την έκβαση της στρατηγικής με τη σωστή σειρά. Οι επιλογές είναι τρείς, «Μετωπική σύγκρουση», «Προσποίηση υποχώρησης Ελλήνων» και «Ελληνική αντεπίθεση» και πρέπει να πατήσει τα κουμπιά με αυτή τη σειρά για να συνεχίζει κάθε φορά την αφήγηση και κατά συνέπεια την έκβαση της μάχης.

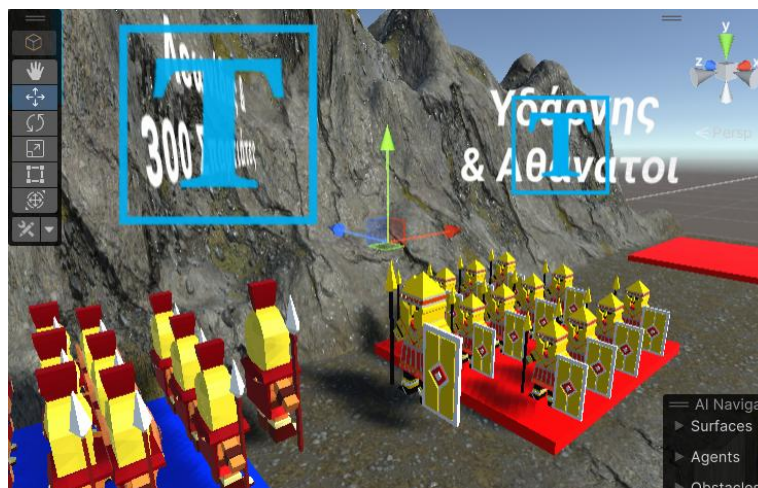
Η λειτουργία που πραγματοποιείται από το QuestionHandlerThermopylae σε σωστή απάντηση του χρήστη είναι τρείς. Αρχικά, απενεργοποιείται το παρον UI και εμφανίζεται το νέο με τις ενημερωμένες επιλογές και στη συνέχεια γίνεται Resume το Timeline και ξεκινάει το κατάλληλο ηχητικό εφέ που στην περίπτωση μας αφορά τις ηχογραφημένες αφηγήσεις (Σχήμα 4.44). Οι μέθοδοι Hide και Show αναφέρονται σε απόκρυψη και εμφάνιση των κατάλληλων UI που έχουν περαστεί σαν μεταβλητές στο script QuestionHandlerThermopylae.



Σχήμα 4.44: Μέθοδοι που καλούνται με την επιλογή σωστής στρατηγικής στη μάχη των Θερμοπυλών

Η τελική σκηνή, μετά από όλες τις σωστές επιλογές που έκανε ο χρήστης, παρουσιάζει την επιτυχή εφαρμογή της στρατηγικής των Ελλήνων κατά των Περσών στο στενό των Θερμοπυλών.

Η αφήγηση στην συνέχεια της άσκησης επικεντρώνεται στις στρατιές των 300 του Λεωνίδα και στην ελίτ μονάδα των Περσών, τους Αθανάτους με τον στρατηγό Υδάρνη. Για τον διαχωρισμό των 2 αυτών ομάδων έχουν τοποθετηθεί τα TextMeshPro από πάνω τους και περιγράφουν ποιοι είναι κατά τη διάρκεια της αφήγησης. Επίσης όσο ο Λεωνίδας τόσο και ο Υδάρνης είναι μεγαλύτεροι σε μέγεθος από τους υπόλοιπους στρατιώτες και η ελίτ ομάδα των Περσών είναι διαφορετικό χρώμα από τους βασικούς Πέρσες στρατιώτες (Σχήμα 4.45).



Σχήμα 4.45: 300 & Υδάρνης

Η σκηνή υπογραμμίζει την κλιμάκωση της έντασης με το ελίτ σώμα των Περσών να κατατροπώνεται από τους Έλληνες. Όλες τις φορές, που ο χρήστης παρατηρεί ένα τάγμα να κατατροπώνεται από τους Έλληνες, αυτό σημαίνει ότι ο συγκεκριμένο τάγμα έχει αρχικοποιηθεί στο script μας σαν αντικείμενο και κατά συνέπεια την κατάλληλη στιγμή που θέλουμε εμείς το κάνουμε disable μέσα από την κληση της κατάλληλης μεθόδου (Σχήμα 4.46).

```

0 references
117 public void YdarnisDestruction()
118 {
119     battalionPS3.SetActive(false);
120 }
121

```

Σχήμα 4.46: Μέθοδος καταστροφής τάγματος Υδάρνη

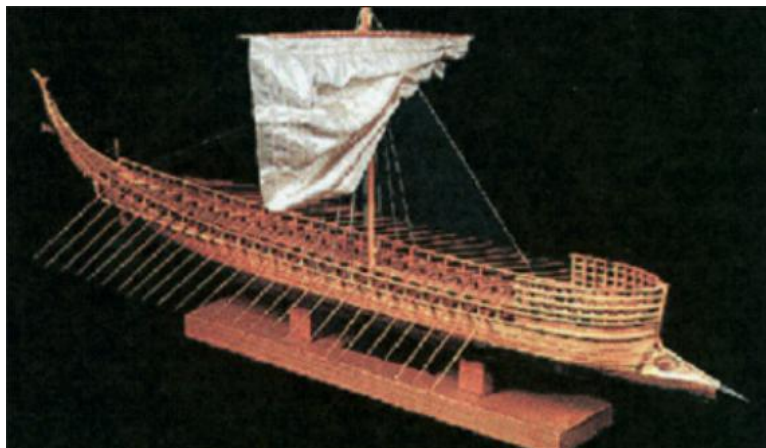
Το τελευταίο σκέλος της μάχης που βλέπει ο χρήστης αφορά την προδοσία των Ελλήνων από τον Εφιάλτη. Η τελευταία ερώτηση, απεικονίζει το UI ερώτηση EfialtisQuestion και ρωτάει το χρήστη «Ποιο ήταν το όνομα του προδότη; με δύο πιθανές απαντήσεις: «Εφιάλτης» ή «Αρταφέρνης». Εφόσον ο χρήστης πατήσει τη σωστή επιλογή του κουμπιού «Εφιάλτης» τότε παρατηρεί το τελευταίο σκέλος της αφήγησης με τους πέρσες να νικούν τους Έλληνες και να εμφανίζεται το WinUI.

4.10.2 Επίλογος της άσκησης «Μάχη των Θερμοπυλών»

Συμπερασματικά, η άσκηση της Μάχης των Θερμοπυλών συνδυάζοντας τις τεχνολογίες του Unity, της ακουστικής αφήγησης όσο και μηχανισμούς drag-and-drop μέσω custom script, οι χρήστες έχουν την ευκαιρία να κατανοήσουν τις στρατηγικές κινήσεις των Ελλήνων στη Μάχη των Θερμοπυλών. Επιπρόσθετα, η άσκηση προσφέρει συνδυασμό ηχητικών εφέ και animation, καθιστώντας την μάθηση ευχάριστη. Το σύστημα ελέγχου σωστών επιλογών, οι ρυθμίσεις του timeline, και οι μέθοδοι που ενεργοποιούν ή απενεργοποιούν αντικείμενα, εξασφαλίζουν τη συνοχή της αφήγησης.

4.11 Ναυμαχία της Σαλαμίνας

Η άσκηση «Ναυμαχία της Σαλαμίνας» αντλεί έμπνευση από το κεφάλαιο 18 του σχολικού βιβλίου με ομώνυμο τίτλο. Συγκεκριμένα επαυξάνει την εικόνα του βιβλίου « 1. Αθηναϊκή τριήρης σε αναπαράσταση (Πειραιάς, Ναυτικό Μουσείο) » (Σχήμα 4.47).



Σχήμα 4.47: Image Target άσκησης της Ναυμαχίας της Σαλαμίνας

Η παρούσα άσκηση σαν βάση χρησιμοποιεί παρόμοιες τεχνολογίες με τις δύο προηγούμενες, θα επικεντρωθούμε κυρίως στα χαρακτηριστικά που τη διαφοροποιούν από τις υπόλοιπες.

4.11.1 Δομή της άσκησης

Αρχικά το πρώτο σκέλος της άσκησης που βλέπει ο χρήστης αφορά μια σειρά από UI σε μορφή κόμικ.. Το πρώτο UI κόμικ Αρχι περιγράφει τη συνέχιση της εκστρατείας του Ξέρξη μετά τη μάχη των Θερμοπυλών. Το σκηνικό περιλαμβάνει μια εικόνα απο τα πλοία των Αθηναίων, δημιουργώντας έτσι μια αίσθηση κινητικότητας. Η εικόνα με τα πλοία ενσωματώνεται στο Canvas μέσω ενός rawImage component στο GameObject του Canvas με όνομα Αρχι για να παρέχει το οπτικό υπόβαθρο της σκηνής.

Το κείμενο που περιγράφει την κατάσταση (εκκένωση της Αθήνας, προετοιμασία για ναυμαχία) υλοποιείται με το γνωστό Text component με το κατάλληλο background για να είναι ορατό το κείμενο. Το κουμπί "Συνέχεια" δίνει στον χρήστη τη δυνατότητα να προχωρήσει στην επόμενη σκηνή (Σχήμα 4.48).

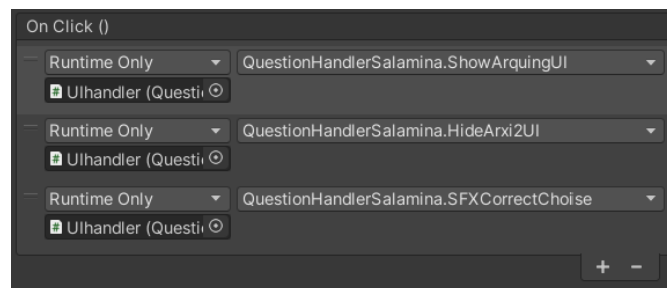


Σχήμα 4.48: ΑρχιUI σε στυλ κόμικ

Όταν ο χρήστης πατήσει την επιλογή συνέχεια τότε, η αφήγηση προχωρά και δείχνει, το επόμενο UI με τους Πελοποννήσιους να υψώνουν τείχος για να εμποδίσουν τους Πέρσες να εισέλθουν στην Πελοπόννησο (Σχήμα 4.49). Ο χρήστης καλείται να συμπληρώσει τη σωστή λέξη (Ισθμός) για να ολοκληρώσει το κενό στην αφήγηση. Η επιλογή γίνεται με τη μορφή κουμπιού. Όλα τα κουμπιά σε αυτό το πρώτο σκέλος είναι υπεύθυνα να απενεργοποιούν το παρόν UI με χρήση της μεθόδου `SetActive(false)` και να ενεργοποιούν το επόμενο UI με τη χρήση της μεθόδου `SetActive(true)`. Επίσης εφόσον ο χρήστης απαντάει σωστά σε ερώτηση σε αυτό το UI κόμικ section ακούγεται και το κατάλληλο ηχητικό εφέ της σωστής επιλογής (Σχήμα 4.50).



Σχήμα 4.49: Arxi2 UI με την επιλογή συμπλήρωσε το κενό



Σχήμα 4.50: Σωστή απάντηση από το χρήστη στη Ναυμαχία της Σαλαμίνας

Τα επόμενα 2 UI (Arquing και Agreed) αφιερώθηκαν στη διαμάχη που είχαν οι δύο ηγέτες, ο Θεμιστοκλής και ο Ευρυβιάδης. Ο Θεμιστοκλής προτείνει να αντιμετωπίσουν τους Πέρσες στη Σαλαμίνα, ενώ ο Ευρυβιάδης θεωρεί ότι πρέπει να εμποδιστεί η πρόσβαση στην Πελοπόννησο. Ο χρήστης καλείται να επιλέξει ποιος από τους δύο πιστεύει ότι έχει δίκιο.

Στο UI που αντικρίζει ο χρήστης (Arquing) μετά την σωστή επιλογή του Ισθμου παρουσιάζεται η εικόνα των δύο ηγετών (Θεμιστοκλής (Αθηναίος) και Ευρυβιάδης (Σπαρτιάτης)). Οι "ομιλίες" τους υλοποιούνται με τη χρήση Speech Bubbles για να θυμίζει περισσότερο κόμικ. Οι Speech Bubbles είναι εικόνες που βρέθηκαν δωρεάν από το διαδίκτυο, και εισάχθηκαν σαν rawImage component, ώστε να προσφέρουν οπτική αναπαράσταση των χαρακτήρων (Σχήμα 4.51)



Σχήμα 4.51: Aruing UI Speech Bubbles

Η επιλογή ανάμεσα στους δύο χαρακτήρες γίνεται με τη χρήση κουμπιού, επιτρέποντας στον χρήστη να επιλέξει ποιον θεωρεί σωστό με βάση τη θεωρία του βιβλίου (σωστή επιλογή: Θεμιστοκλής) (Σχήμα 4.52)



Σχήμα 4.52: Arguing UI επιλογή αρχηγού

Μετά την επιλογή, η αφήγηση τονίζει τη στρατηγική υπεροχή της Σαλαμίνας και απεικονίζει τους δύο αρχηγούς να συμφωνούν «δίνοντας τα χέρια». Σαν τελευταία απαίτηση από τον μαθητή και για να προστεθεί μια επιπλέον διάδραση, ο χρήστης καλείται να εισάγει μέσω πληκτρολογίου την ονομασία των ελληνικών πλοίων. Η σωστή ονομασία των ελληνικών πλοίων "Τριήρεις" συμπληρώνεται από τον χρήστη μέσω ενός Input Field. Για τη λειτουργία και τον έλεγχο του Input Field δημιουργήθηκε το script ReadInput.

Το script χρησιμοποιεί το TMProInputField για την ανάκτηση του κειμένου που εισάγεται από τον χρήστη. Η κύρια μέθοδος CheckIfCorrect() ελέγχει εάν η απάντηση που εισάγει ένας χρήστης ταιριάζει με κάποια από τις αποδεκτές παραλλαγές της λέξης "Τριήρεις" λαμβάνοντας υπόψη ορθογραφία, κεφαλαία, πεζά, ακόμη και μία εναλλακτική επιλογή για λατινικούς χαρακτήρες. Εάν η απάντηση είναι σωστή, καλούνται μέθοδοι από το QuestionHandlerSalamina, για να ενεργοποιηθεί ένα ηχητικό εφέ σωστής επιλογής και να γίνουν οι κατάλληλες αλλαγές στα UIs.

Κεφάλαιο 4

Σε περίπτωση λανθασμένης απάντησης, αυτό ενεργοποιεί το γνωστό πλέον WrongImage UI σε συνδυασμό με το ηχητικό εφέ του σφάλματος. Για να είναι εφικτή η χρήση του TMPProInputField εισάγαμε τη βιβλιοθήκη TMPPro και για να μπορέσουμε να καλέσουμε μεθόδους του script QuestionHandlerSalamina μέσα από το ReadInput, αρχικοποιήσαμε μια μεταβλητή τύπου QuestionHandlerSalamina q μέσα στο ReadInput (Σχήμα 4.53)

```
ReadInput.cs  X QuestionHandlerSalamina.cs
Assembly-CSharp  ReadInput
4  using UnityEngine.UI;
5  using TMPPro;
6
7  public class ReadInput : MonoBehaviour
8  {
9      public GameObject TMPProInputField;
10     public QuestionHandlerSalamina q;
11
12
13     public void CheckIfCorrect()
14     {
15
16         if (TMPProInputField.GetComponent<TMP_InputField>().text=="ΤΡΙΗΡΕΙΣ" ||
17             TMPProInputField.GetComponent<TMP_InputField>().text == "Τριήρεις" ||
18             TMPProInputField.GetComponent<TMP_InputField>().text == "Τριηρεις" ||
19             TMPProInputField.GetComponent<TMP_InputField>().text == "τριήρεις" ||
20             TMPProInputField.GetComponent<TMP_InputField>().text == "τριηρεις" ||
21             TMPProInputField.GetComponent<TMP_InputField>().text == "τριήρεις" ||
22             TMPProInputField.GetComponent<TMP_InputField>().text == "trihreis" ||
23             TMPProInputField.GetComponent<TMP_InputField>().text == "Trihreis")
24         {
25
26             q.SFXCorrectChoise();
27             q.ShowPrewarUI();
28             q.HideAgreedUI();
29
30         }
31     else
32     {
33         q.SFXWrongChoise();
34         q.ShowWrongAnswerUI();
35     }
36 }
37
38 }
```

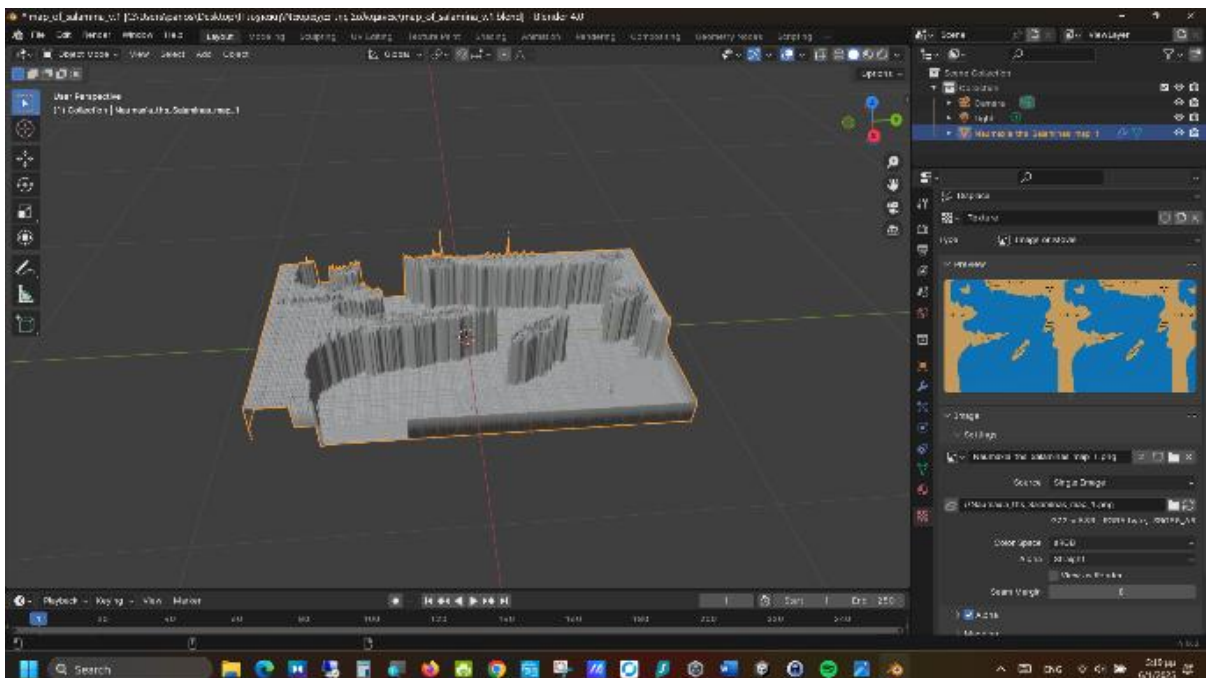
Σχήμα 4.53: ReadInput script

Ο χρήστης λοιπόν εισάγει μέσω του πληκτρολογίου του κινητού του την κατάλληλη απάντηση και έπειτα πατάει το κουμπί "Ελεγχος", το οποίο και καλεί τη μέθοδο CheckIfCorrect του ReadInput script. Μόλις ο χρήστης εισαγάγει τη σωστή λέξη "Τριήρεις" εμφανίζεται το UI Prewar, το οποίο προτρέπει το χρήστη με ένα εισαγωγικό text να σαρώσει τη φωτογραφία του βιβλίου που λειτουργεί σαν Image Target. Τότε λοιπόν θα εμφανιστεί, στον χρήστη το UI με το κουμπί «Εκκίνηση» και ο χάρτης της μάχης (Σχήμα 4.54).



Σχήμα 4.54: UI που προετοιμάζει το χρήστη για τη ναυμαχία της Σαλαμίνας

Ο χάρτης που εμφανίζεται στο χρήστη δημιουργήθηκε από 2D φωτογραφία της τοποθεσίας της Σαλαμίνας που βρέθηκε στον ιστότοπο Φωτόδεντρο. Για τη τροποποίηση του χάρτη σε 3D, έπρεπε σε πρώτη φάση να τον επεξεργαστούμε μέσω Blender. Αρχικά, εισάγουμε την εικόνα του χάρτη μου plane, προσθέτουμε έναν Subdivision Modifier και επιλέγουμε το Simple Mode (Σχήμα 4.55). Για να αφαιρέσουμε τις κορυφές που έχουν δημιουργηθεί από τον modifier ορίζουμε την τιμή Levels σε 10 και κάνουμε apply στον modifier.



Σχήμα 4.55: Δημιουργία χάρτη της Ναυμαχίας της Σαλαμίνας

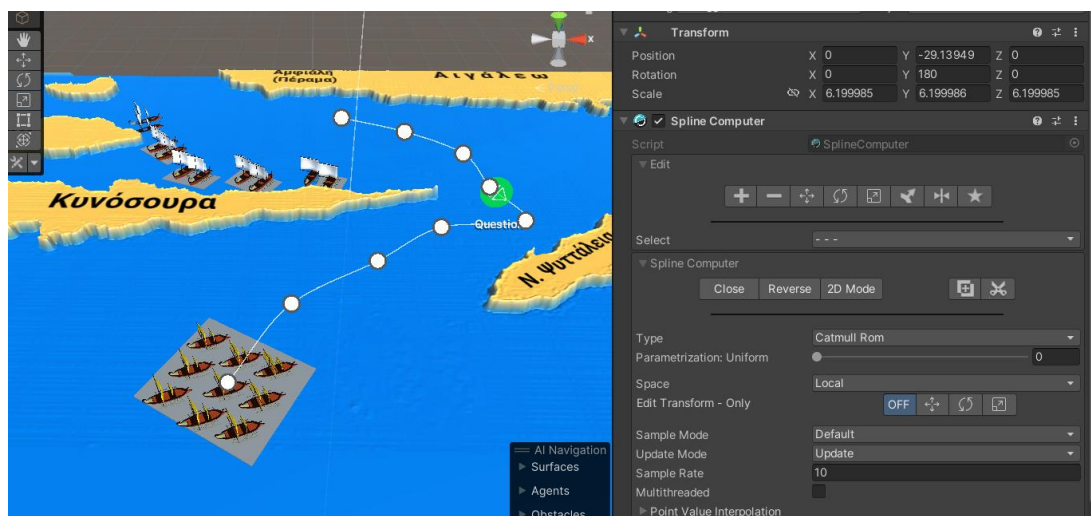
Τέλος, για να έχει ο χάρτης μας τα χρώματα της εικόνας reference, προσθέτουμε ένα Displace Modifier και στην συνέχεια, πρέπει δημιουργούμε ένα νέο texture (New). Στις ρυθμίσεις του Displace Modifier,

επιλέγουμε UV Coordinates. Επίσης, στην καρτέλα Texture, ορίζουμε τον τύπο του texture σε Image or Movie και επιλέγουμε την εικόνα του χάρτη μου από το σημείο που την έχουμε αποθηκεύσει στον υπολογιστή μας, μέσω της επιλογής Browse to be linked image. Καταληκτικά, ρυθμίζουμε το Strength του Displace Modifier για να διαμορφώσω τον χάρτη μου στο επιθυμητό ανάγλυφο αποτέλεσμα. Με αυτόν τον τρόπο λοιπόν δημιουργήσαμε ένα έδαφος, από μια εικόνα του χάρτη της Σαλαμίνας.

Για να συνεχιστεί η αφήγηση, λοιπόν, πρέπει ο χρήστης να πατήσει το κουμπί «Εκκίνηση» στο UI StartBattle.

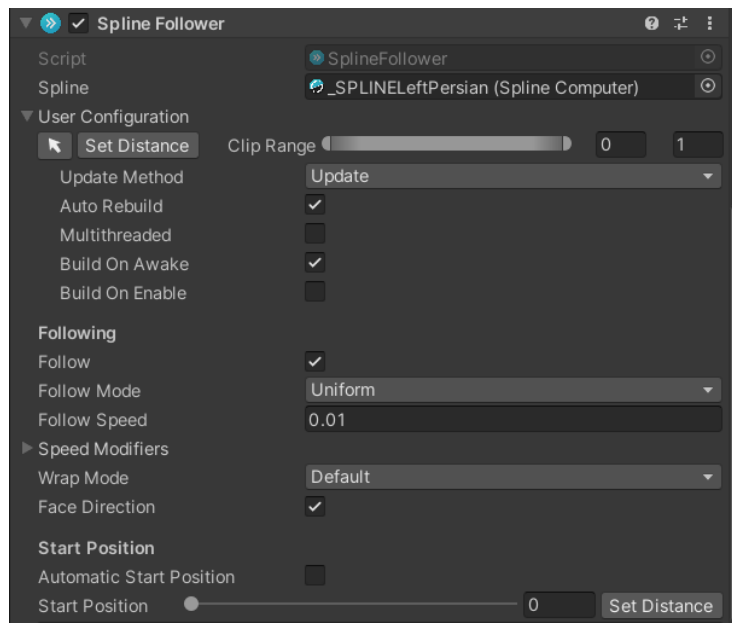
Όταν ο χρήστης πατήσει το κουμπί «Εκκίνηση» τότε, η αφήγηση ξεκινά και ο χρήστης βλέπει τους στόλους των Περσών να κινούνται και να καταλαμβάνουν την Ψυττάλεια. Τα 3D μοντέλα των πλοίων είναι τοποθετημένα σε στόλους, με παρόμοιο τρόπο που δημιουργήθηκαν και τα τάγματα στις προηγούμενες δύο ασκήσεις, για να είναι εφικτή η κίνηση τους μέσω του animation σαν ομάδα.

Οι κινήσεις των πλοίων που παρατηρεί ο χρήστης, σε αυτή την άσκηση δημιουργήθηκαν με τη βοήθεια του εργαλείου Dreamteck splines. Για να δημιουργήσουμε κίνηση για ένα αντικείμενο, πρέπει αρχικά να δημιουργηθούν splines. Τα splines αφορούν τα μονοπάτια που θα ακολουθήσουν οι στόλοι στο χάρτη. Για να δημιουργήσουμε τα μονοπάτια, κάνουμε δεξιά κλικ στο Hierarchy και μέσα από την επιλογή 3D Object -> spline computer εισάγω το αντικείμενο spline στο Hierarchy. Με αυτό τον τρόπο «ζωγραφίζουμε» το μονοπάτι που θέλουμε να ακολουθήσει ο εκάστοτε στόλος μέσα πάνω στον χάρτη μας (Σχήμα 4.56).



Σχήμα 4.56: Δημιουργία μονοπατιού μέσω Dreamtech Spline

Για να μπορέσει ο κάθε στόλος να ακολουθήσει ένα μονοπάτι θα πρέπει να εισάγουμε το component Spline Follower στο αντικείμενο του κάθε στόλου που θέλουμε να κινείται πάνω σε ένα συγκεκριμένο μονοπάτι. (Σχήμα 4.57).



Σχήμα 4.57: Ρυθμίσεις Spline Follower

Για το χειρισμό της κίνησης του των στόλων μέσω του spline follower χρειάστηκε η δημιουργία του script SplineAnimationController.

Το script είναι υπεύθυνο για την έναρξη και το σταμάτημα της κίνησης των στόλων και για την ενεργοποίηση και απενεργοποίηση ορισμένων στόλων. Σαν πρώτη φάση θεωρείται, η κίνηση των στόλων των περσών, μέχρι και να παρατασθούν για μάχη στο στενό της Σαλαμίνας.

Στη μέθοδο Start, αναφέρεται ότι όλες οι κινήσεις των στόλων θα είναι σταματημένες on-awake, μέσω της μεθόδου StopALLAnimation. Οι υπόλοιπες μέθοδοι του script αφορούν την ενεργοποίηση ή απενεργοποίηση των στόλων ανάλογα με το αν είναι Έλληνες ή Πέρσες. Συγκεκριμένα, η μέθοδος StartAnimation ενεργοποιούν την κίνηση των δύο αντικειμένων SplineFollower που αντιπροσωπεύουν τον περσικό στόλο που βλέπει ο χρήστης στην πρώτη φάση της μάχης.

Υπάρχουν και μέθοδοι για το σταμάτημα αυτών των κινήσεων. Η StopAnimation σταματά τους περσικούς στόλους του της πρώτης φάσης της μάχης, ενώ η StopROUND2AnimationPERSIANS και η StopGREEKSAnimation σταματούν τους στόλους της δεύτερης φάσης της μάχης των Περσών και των Ελλήνων, αντίστοιχα.

Η ενεργοποίηση και απενεργοποίηση στόλων (GameObjects) γίνεται μέσω των μεθόδων EnableROUND1, EnableROUND2, DisableROUND1 και DisableROUND2. Επιπλέον, οι μέθοδοι DestructPersianBattalion1, DestructPersianBattalion2 και DestructPersianBattalion3 απενεργοποιούν συγκεκριμένα περσικά στρατεύματα του δεύτερου γύρου, για να προσομοιώσουν την καταστροφή τους κατά τη διάρκεια της μάχης. Στο Σχήμα 4.58, παραθέτετε ο τρόπος με τον οποίο μια κίνηση ξεκινά πάνω στο spline.

```

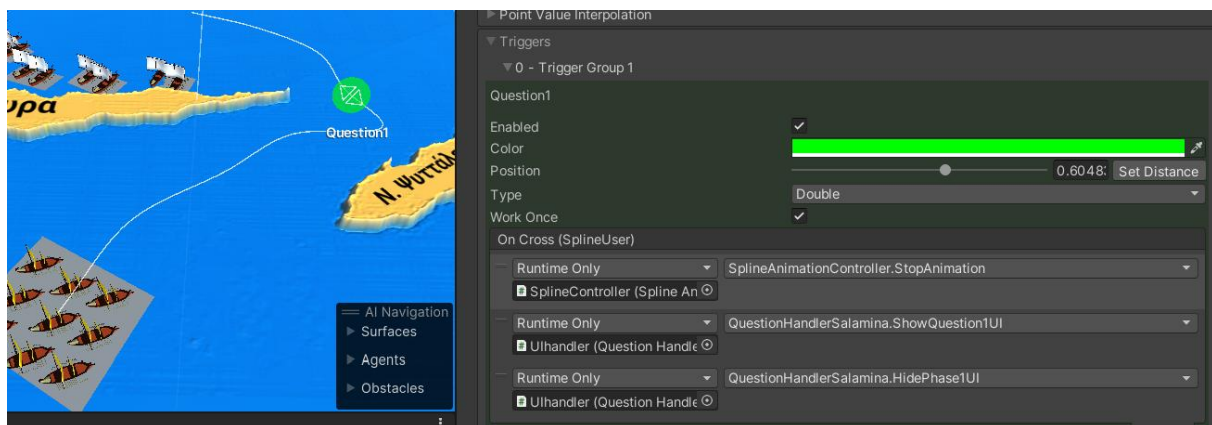
4  using Dreamteck.Splines;
5
6  © Unity Script (1 asset reference) | 0 references
7  public class SplineAnimationController : MonoBehaviour
8  {
9      0 references
10     public void StartAnimation()
11     {
12         round1leftPersianBattallions.enabled = true;
13         round1rightPersianBattallions.enabled = true;
14     }

```

Σχήμα 4.58: Χειρισμός κινήσεων Dreamteck splines

Αφού λοιπόν αναλύσαμε και το πως κινούνται οι στόλοι κατά τη διάρκεια της αφήγησης, το μόνο που μένει είναι να συνεχίσουμε την αφήγηση.

Στη συνέχεια της αφήγησης της μάχης, ο χρήστης παρακολουθεί τους στόλους των Περσών να φθάνουν στην Ψυττάλεια και να σταματούν. Το σταμάτημα του animation πραγματοποιείται σε παρόμοια λογική με τις προηγούμενες δύο ασκήσεις, δηλαδή με τη χρήση triggers. Η διαφορά στη συγκεκριμένη περίπτωση είναι ότι οι triggers βρίσκονται πάνω στα μονοπάτια splines. Όπως βλέπουμε και στο Σχήμα 4.59, τη στιγμή που θα φτάσουν οι στόλοι των Περσών πάνω στον trigger με όνομα Question1, θα γίνει η κλήση της μεθόδου StopAnimation του SplineAnimationController script, όπως επίσης και η εμφάνιση της πρώτης ερώτησής Question1 UI.



Σχήμα 4.59: Trigger πάνω σε spline και κάλεσμα μεθόδων

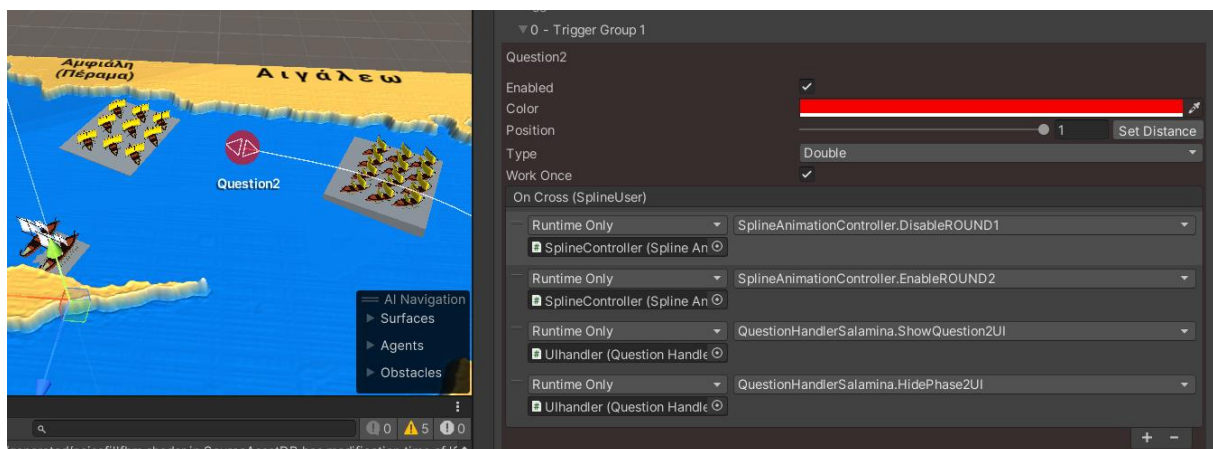
Οι στόλοι σταματούν και εμφανίζεται η πρώτη ερώτηση στο χρήστη. Στην ερώτηση αυτή, παρουσιάζεται το ερώτημα: "Τι εντολή έδωσαν στα στρατεύματά τους οι αρχηγοί των Περσών, μόλις είδαν τον ελληνικό στόλο να είναι παραταγμένος στη Σαλαμίνα;". Ο χρήστης καλείται να επιλέξει μία από τις τρεις επιλογές. Οι επιλογές είναι σε μορφή κουμπιών και ο χρήστης πρέπει να πατήσει τη σωστή επιλογή «Έδωσαν εντολή να κυκλωθεί ο ελληνικός στόλος».

Η σωστή απάντηση ενεργοποιεί την αφήγηση, δηλαδή κάνει enable τους κατάλληλους spline followers στόλους και ενεργοποιεί το επόμενο UI Phase2 που εξηγεί στον χρήστη μέσω του Phase2 UI ότι οι Πέρσες εισήλθαν στο στενό με σκοπό να κυκλώσουν τους Έλληνες (Σχήμα 4.60).

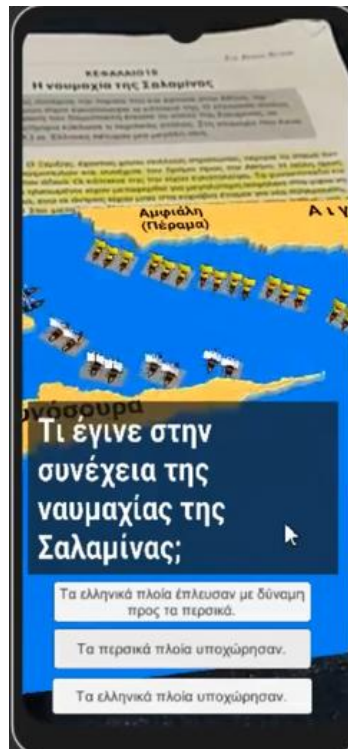


Σχήμα 4.60: Phase2 UI, ναυμαχία της Σαλαμίνας

Μόλις λοιπόν ο δεύτερος στόλος των περσών από την πρώτη φάση φτάσει στο trigger Question2, τότε ξεκινάει και η δεύτερη φάση Round2 που αναφέρθηκε παραπάνω. Θα απενεργοποιηθούν οι συγκεκριμένοι στόλοι της ROUND1 (μέθοδος DisableROUND1) των περσών και θα εμφανιστούν οι στόλοι τις round2 μέσω της μεθόδου EnableROUND2, από το script SplineAnimationController. Η ROUND2 δείχνει τους στόλους των περσών παραταγμένους απέναντι από αυτούς των Ελλήνων και ταυτόχρονα ενεργοποιείται και η Question2 (Σχήμα 4.61), με το κάλεσμα της κατάλληλης μεθόδου από το QuestionControllerSalamina script. Η ερώτηση «Τι έγινε στη συνέχεια της ναυμαχίας της Σαλαμίνας;» (Σχήμα 4.62).



Σχήμα 4.61: Question2 spline trigger και κάλεσμα μεθόδων



Σχήμα 4.62: Question2 UI, ναυμαχία της Σαλαμίνας

Με τη σωστή επιλογή από την μεριά του χρήστη «Τα ελληνικά πλοία έπλευσαν με δύναμη προς τα περσικά», τα πλοία απεικονίζονται να κινούνται δυναμικά προς τη δεξιά πλευρά του χάρτη, το οποίο και επιτευχθεί με τη δημιουργία των κατάλληλων Dreamteck Splines για κάθε ελληνικό στόλο.

Τέλος, η δράση που βλέπει ο χρήστης συνεχίζεται, με την Αθηναϊκή μοίρα να εκτελεί τον στρατηγικό ελιγμό της και να διασπά τη δεξιά πτέρυγα των Περσών και ταυτόχρονα βλέπει την Περσική μοίρα να καταδιώκεται, μέχρι και την νίκη των Ελλήνων, η οποία δηλώνεται με τη χρήση του Win UI

4.11.2 Επίλογος της άσκησης «Ναυμαχία της Σαλαμίνας»

Συμπερασματικά, η άσκηση της Ναυμαχίας της Σαλαμίνας συνδυάζει τεχνολογίες του Unity και τρισδιάστατα γραφικά του χάρτη και των πλοίων που αναπτύχθηκαν στο Blender, για να δώσει στους χρήστες μια πιο ολοκληρωμένη εμπειρία για τη στρατηγική που ακολούθησαν οι Έλληνες στη Ναυμαχία της Σαλαμίνας. Ο χρήστης αρχικά πρέπει να απαντήσει βασικές ερωτήσεις που του παρουσιάζονται σε μορφή κομικ πριν μεταβεί στην διαδραστική αφήγηση. Οι στόλοι κινήθηκαν με τη χρήση της τεχνολογίας των Dreamteck Splines και ο έλεγχος της ροής της αφήγησης, των κινήσεων του στόλου και των απαντήσεων των χρηστών πραγματοποιήθηκε μέσω των script SplineAnimationController, ReadInput και QuestionControllerSalamina. Η εφαρμογή συγχωνεύει την τεχνολογία με την αφήγηση. Ως εκ τούτου, προσφέρει στον χρήστη μια εμπλουτισμένη εκπαιδευτική εμπειρία, επειδή κάνει τη μάθηση διαδραστική και επίσης δίνει μια οπτική αναπαράσταση.

Κεφάλαιο 5ο: Συμπεράσματα & Μελλοντικές βελτιώσεις

Συμπερασματικά, η εφαρμογή που δημιουργήσαμε για να βοηθήσει το μαθητή στη κατανόηση του μαθημα της Ιστορίας της Δ' Δημοτικού, χρησιμοποίησε σύγχρονα εργαλεία όπως το Unity, το Blender και το Vuforia Engine. Οι μαθητές έχουν τη δυνατότητα μέσα από την εφαρμογή να μάθουν και να κατανοήσουν ιστορικές μάχες με τρισδιάστατες εικόνες, φωνητική αφήγηση της ιστορίας και ερωτήσεις, καθιστώντας την εφαρμογή ένα βοηθητικό εργαλείο για μελέτη.

Εννοείται πως υπάρχουν δυνατότητες για τη βελτίωση της εφαρμογής. Αρχικά, αν θέλουμε να αναφερθούμε σε μεγαλύτερες τάξεις του δημοτικού με εφαρμογή παρόμοιας λογικής, οι ερωτήσεις θα χρειαστούν ενίσχυση στις εξηγήσεις τους. Οι λανθασμένες απαντήσεις, μπορούν να παρέχουν περισσότερες πληροφορίες για να εξηγήσουν τις στρατηγικές και τα ιστορικά γεγονότα. Μια ακόμα βελτίωση είναι η εισαγωγή περισσότερων οπτικών και ηχητικών εφέ για να γίνει η εφαρμογή πιο ελκυστική και «αληθινή», επομένως και η μάθηση να γίνει πιο διασκεδαστική.

Μια σημαντική βελτίωση, αφορά και την εισαγωγή συγκεκριμένων animation που θα πραγματοποιούν οι στρατιώτες ή και τα πλοία κατά τη διάρκεια μίας μετωπικής σύγκρουσής για παράδειγμα.

Ένα άλλο θα ήταν να συμπεριληφθεί ένα σύστημα ανατροφοδότησης με βάση τη χρήση της εφαρμογής, το οποίο θα παρείχε πληροφορίες στους εκπαιδευτικούς για να αξιολογήσουν πόσο καλά τα πηγαίνει ένας μαθητής και ενδεχομένως να βοηθούσε στην προσαρμογή των οδηγιών που παρατίθενται στους μαθητές, όπως επίσης και θα να διευρυνθεί η χρήση της εφαρμογής σε θέματα και εκτός της ιστορίας.

Τέλος, όπως έχει αναφερθεί, η τεχνητή νοημοσύνη χρησιμοποιήθηκε για την παραγωγή πολλών φωτογραφιών αλλά και reference image της παρούσας εφαρμογής. Η χρήση τέτοιων τεχνολογιών όπως η τεχνητή νοημοσύνη, θα μπορούσε στο μέλλον να χρησιμοποιηθεί για την δυναμική προσαρμογή του περιεχομένου του κειμένου αλλά και της χρήσης της εφαρμογής με βάση τις ανάγκες κάθε μαθητή. Για παράδειγμα, να καθορίζει ο εκπαιδευτικός μια συγκεκριμένη μάχη που θέλει να εξιστορηθεί στους μαθητές, ή να τροποποιεί το περιεχόμενο της εφαρμογής, με τη βοήθεια της τεχνητής νοημοσύνης, για να είναι συμβατό και για μαθητές που παρουσιάζουν μαθησιακές δυσκολίες.

Η εφαρμογή μπορεί να γίνει καλύτερη, με πάρα πολλούς τρόπους, και σε πολλά μαθήματα βρίσκοντας τη θέση της ως χρήσιμο εργαλείο για τη σύγχρονη εκπαίδευση, καθώς η μάθηση δεν σταματά ποτέ!

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Donna Berryman and Matthew B. Hoy, *Augmented Reality: A Review*. University of Rochester Medical Center , Rochester , New York , 2012
- [2] Interaction Desing foundation, “Beyond AR vs. VR: What is the Difference between AR vs. MR vs. VR vs. XR?” *Laia Tremosa*, [Online]. Available: <https://www.interaction-design.org/literature/article/beyond-ar-vs-vr-what-is-the-difference-between-ar-vs-mr-vs-vr-vs-xr?srsId=AfmBOoqkBpny4lQRNrKJ5q-yp7l4OkJScbzuzqWkfJxibmyXPuu-RuDil>.
- [3] Richard Skarbez , Missie Smith and Mary C. Whitton , *Revisiting Milgram and Kishino's Reality-Virtuality Continuum.*, Australia., United States, 2021
- [4] Yunqiang Chen, “An overview of augmented reality technology,” *Journal of Physics: Conf. Series* 1237, 2019.
- [5] Molly Hayes, Amanda Downie, “What is augmented reality?”, *IBM*, 2024. [Online]. Available: <https://www.ibm.com/think/topics/augmented-reality>.
- [6] “How Does Augmented Reality Work? (with examples)”, REYDAR,. [Online]. Available: <https://www.reydar.com/how-does-augmented-reality-work/>.
- [7] Michael E. Porter and James E. Heppelmann, “What is augmented reality?”, Spotlight Series/A Manager’s Guide to Augmented Reality, 2017. [Online]. Available: <https://hbr.org/2017/11/how-does-augmented-reality-work>.
- [8] J Strickland, “How Stuff **Works**”, 2007: <https://hbr.org/2017/11/how-does-augmented-reality-work>.
- [9] Speicher, M., Hall, B. D., & Nebeling, M. (2019). What is Mixed Reality? Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19. Glasgow, Scotland, UK, 2019.
- [10] Bernard Marr, “What Is Extended Reality Technology? A Simple Explanation For Anyone?”, *Forbes*, 2021. [Online]. Available: <https://www.forbes.com/sites/bernardmarr/2019/08/12/what-is-extended-reality-technology-a-simple-explanation-for-anyone/>.
- [11] Cem Dilmegani , “XR/AR in Retail: Benefits, Use Cases & Examples”, AI Multiple Research,. [Online]. Available: <https://research.aimultiple.com/ar-in-retail/>.
- [12] European Lung Foundation, “Remote care: looking after your health using digital tools”, ELF ,2024 last updated. [Online]. Available: <https://europeanlung.org/en/information-hub/living-with-a-lung-condition/remote-care-looking-after-your-health-using-digital-tools-ipsuim/>.
- [13] Josh Sandberg, “ First Augmented Reality Spine Surgery Using FDA-Cleared Augmedics xvision™ Spine System Completed in U.S.”, 2020. [Online]. Available: <https://orthospinenews.com/2020/06/11/first-augmented-reality-spine-surgery-using-fda-cleared-augmedics-xvision-spine-system-completed-in-u-s/>.
- [14] Meeri Kim, “Virtual reality is being used by hospitals to help people cope with pain”, *The Washington Post*,2018. [Online]. Available: https://www.washingtonpost.com/national/health-science/virtual-reality-is-being-used-by-hospitals-to-help-people-cope-with-pain/2018/06/08/07e8b39e-6424-11e8-a768-ed043e33f1dc_story.html.

- [15] Robin Robinson, “The Realities of VR and AR in the Industry”, PharmaVoice, 2019. [Online]. Available: <https://www.pharmavoices.com/news/2019-06-ar-vr/612434/>.
- [16] Bernard Marr, “10 Best Examples Of VR And AR In Education”, *Forbes*, 2021. [Online]. Available: <https://www.forbes.com/sites/bernardmarr/2021/07/23/10-best-examples-of-vr-and-ar-in-education/>.
- [17] Andy Trotman, “Using holograms to train nurses: Pearson and Microsoft launch mixed-reality curriculum”, *Microsoft*, 2018. [Online]. Available: <https://ukstories.microsoft.com/features/using-holograms-train-nurses-pearson-microsoft-launch-mixed-reality-curriculum/>.
- [18] Γεώργιος Μπορμπαντωνάκης, “Εφαρμογές της εικονικής πραγματικότητας στο πεδίο της ποινικής δικαιοσύνης”, *The Art of Crime*, 2023. [Online]. Available: <https://theartofcrime.gr/%CE%B5%CF%86%CE%B1%CF%81%CE%BC%CE%BF%CE%B3%CE%AD%CF%82-%CF%84%CE%B7%CF%82-%CE%B5%CE%B9%CE%BA%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE%CF%82-%CF%80%CF%81%CE%B1%CE%B3%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CF%8C%CF%84/>.
- [19] “Επαυξημένη/Εικονική πραγματικότητα στην καθημερινή μας ζωή”, *MinnaLearn*. [Online]. Available: <https://courses.minnalearn.com/el/courses/emerging-technologies/extended-reality-vr-ar-mr/ar-vr-in-our-daily-life/>.
- [20] Unity’s Package Manager, *Unity Documentation*. [Online]. Available: <https://docs.unity3d.com/Manual/Packages.html>.
- [21] Dreamteck Splines, *Unity Asset Store*. [Online]. Available: <https://assetstore.unity.com/packages/tools/utilities/dreamteck-splines-61926>.
- [22] Playable Director component, *Unity Documentation*. [Online]. Available: <https://docs.unity.cn/2019.2/Documentation/Manual/class-PlayableDirector.html>.
- [23] Collider with both collision and triggering, choosing at runtime, *Unity Community Forum*. [Online]. Available: <https://discussions.unity.com/t/collider-with-both-collision-and-triggering-choosing-at-runtime/571572>.
- [24] Introduction to render pipelines, *Unity Documentation*. [Online]. Available: <https://docs.unity3d.com/6000.0/Documentation/Manual/render-pipelines-overview.html>
- [25] Getting Started with Vuforia Engine in Unity, *Vuforia Engine Developer Portal*. [Online]. Available: <https://developer.vuforia.com/library/getting-started/getting-started-vuforia-engine-unity>.
- [26] Working with Device Databases, *Vuforia Engine Developer Portal*. [Online]. Available: <https://developer.vuforia.com/library/objects/device-databases>.
- [27] Ιστορία (Δ΄ Δημοτικού) - Βιβλίο Μαθητή (Εμπλουτισμένο), *ΔΙΑΔΡΑΣΤΙΚΑ ΣΧΟΛΙΚΑ ΒΙΒΛΙΑ*. [Online]. Available: http://ebooks.edu.gr/ebooks/v/html/8547/2174/Istoria_D-Dimotikou_html-empl/.
- [28] ΙΣΤΟΡΙΑ (ΤΕΤΡΑΔΙΟ ΕΡΓΑΣΙΩΝ), *ΔΙΑΔΡΑΣΤΙΚΑ ΣΧΟΛΙΚΑ ΒΙΒΛΙΑ*. [Online]. Available: <http://ebooks.edu.gr/ebooks/handle/8547/46>.
- [29] ΦΩΤΟΔΕΝΤΡΟ - ΕΘΝΙΚΟΣ ΣΥΣΣΩΡΕΥΤΗΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ, [Online]. Available: <http://photodentro.edu.gr/aggregator/>

- [30] Η Ναυμαχία της Σαλαμίνας: Η μάχη, *ΦΩΤΟΔΕΝΤΡΟ*. [Online]. Available: <http://photodentro.edu.gr/aggregator/lo/photodentro-educationalvideo-8522-473>.
- [31] Ελληνικός Πολιτισμός, Μάχες, Η ναυμαχία της Σαλαμίνας 480 π.Χ. [Online]. Available: <https://users.sch.gr/ipap/Ellinikos%20Politismos/maxes/Salamina.htm/>
- [32] Ελληνικός Πολιτισμός, Χάρτες: Χάρτης της μάχης των Θερμοπυλών 480 π.Χ. με κίνηση. [Online]. Available: <http://users.sch.gr/ipap/Ellinikos%20Politismos/xartes/Thermopylae.htm>.
- [33] Ελληνικός Πολιτισμός, Χάρτες: Χάρτες της μάχης του Μαραθώνα το 490 π.Χ. [Online]. Available: <https://users.sch.gr/ipap/Ellinikos%20Politismos/xartes/Marathonas2.htm>.
- [34] Vuforia tips, troubleshooting and resources , *Unity Documentation* [Online]. Available: https://docs.unity3d.com/es/2018.4/Manual/vuforia_tips.html

ΠΑΡΑΡΤΗΜΑ Α : QuestionHandlerStart.cs C# Script

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

public class QuestionHandlerStart : MonoBehaviour
{
    public GameObject mainMenu;
    public GameObject welcomeImage;
    public GameObject instructions;
    // Start is called before the first frame update
    void Start()
    {
        welcomeImage.SetActive(true);
        mainMenu.SetActive(false);
        instructions.SetActive(false);
    }
    public void MainMenu()
    {
        mainMenu.SetActive(true);
    }
    public void ShowInstructions()
    {
        instructions.SetActive(true);
    }

    public void HideWelcomeImage()
    {
        welcomeImage.SetActive(false);
    }
    public void HideInstructions()
```

```

    {
        instructions.SetActive(false);
    }
}.

```

ΠΑΡΑΡΤΗΜΑ Β : InBattle.cs C# Script

```

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

public class InBattle : MonoBehaviour
{

    public Animator mAnimator;
    // Start is called before the first frame update
    void Start()
    {
        mAnimator = GetComponent<Animator>();
    }
    public void ResetTrigger()
    {
        mAnimator.ResetTrigger("isPlayed");
    }

    public void PlayInBattleAnimation()
    {
        mAnimator.SetTrigger("isPlayed");
        Invoke("ResetTrigger", 2f);
    }
}

```

ΠΑΡΑΡΤΗΜΑ C : SceneTransitions.cs C# Script

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class SceneTransitions : MonoBehaviour
{
    public void LoadBattleOfMarathon()
    {
        SceneManager.LoadScene("Marathonas");
    }

    public void LoadBattleOfThermopylae()
    {
        SceneManager.LoadScene("Thermopylae");
    }
    public void LoadSalamina()
    {
        SceneManager.LoadScene("Salamina_battle");
    }
    public void LoadStartScene()
    {
        SceneManager.LoadScene("StartScene");
    }
    public void RestartGame()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }
    public void QuitGame()
    {
        Application.Quit();
    }
}}
```

ΠΑΡΑΡΤΗΜΑ Δ : QuestionHandler.cs C# Script

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

public class QuestionHandler : MonoBehaviour
{
    public GameObject firstRowBottom;
    public GameObject secondRowBottom;

    public GameObject firstRowTop;
    public GameObject secondRowTop;

    public GameObject PesrianUnitTop;
    public GameObject PesrianUnitCentre;
    public GameObject PesrianUnitBottom;

    public GameObject battleground;
    public GameObject restartButton;
    public GameObject homeButton;

    public GameObject arxikiQuestion;
    public GameObject question1;
    public GameObject question2;
    public GameObject phase1_2question;
    public GameObject phase1_3question;
    public GameObject phase2question;
    public GameObject phase3question;
    public GameObject win_ui;
    public GameObject wrongImageAnimation;
    public GameObject inBattle;
```

```

public PlayableDirector timeline;

public AudioSource src;
public AudioSource src2;
public AudioClip sfx_correct, sfx_wrong, sfx_clash,sfx_win;

void Start()
{

    //win_ui.SetActive(false);
    battleground.SetActive(false);
    //HideArxikiQuestion();
    HideAllQuestions();
}

//START----- manipulate wrong answer-----
public void ShowWrongAnswerUI()
{

    wrongImageAnimation.SetActive(true);
    Invoke("HideWrongAnswerUI", 2f);

}
public void HideWrongAnswerUI()
{
    wrongImageAnimation.SetActive(false);
}

```

```

public void WrongAnswerSFX()
{
    src.clip = sfw_wrong;
    src.Play();

}
//END-----manipulate wrong answer-----

// START----manipulate inBattle UI-----
public void ShowInBattleUI()
{

    inBattle.SetActive(true);
    Invoke("HideInBattleUI", 2f);

}
public void HideInBattleUI()
{
    inBattle.SetActive(false);
}

//END-----manipulate inBattle UI-----

//START-----Signal Emmiters-----
public void SignalPhase1_2()
{
    PauseTimeline();
    Showphase1_2question();

}

```

```
public void SignalPhase1_3()
{
    PauseTimeline();
    Showphase1_3question();
}
```

```
public void SignalPhase2()
{
    PauseTimeline();
    Showphase2question();
}
```

```
public void SignalPhase3()
{
    phase2question.SetActive(false);
    SlowMotionTimeline();
    Showphase3question();
}
```

```
public void FirstrowSignalBoardOnShip()
{
    firstRowBottom.SetActive(false);
    firstRowTop.SetActive(false);
}
```

```
public void SecondRowSignalBoardOnShip()
{
    secondRowBottom.SetActive(false);
    secondRowTop.SetActive(false);
}
```

```

public void SignalEndOfBattle()
{
    PesrianUnitCentre.SetActive(false);
    PesrianUnitTop.SetActive(false);
    PesrianUnitBottom.SetActive(false);
    Win();
}

//END-----Signal Emmiters-----

//Start-----Timeline maninulation-----

public void playTimeline()
{
    timeline.Play();
    //generals_timeline.Play();
}

public void PauseTimeline()
{
    if (timeline != null)
        timeline.playableGraph.GetRootPlayable(0).SetSpeed(0);
}

public void ResumeTimeline()
{
    if (timeline != null)
        timeline.playableGraph.GetRootPlayable(0).SetSpeed(0.7);
}

public void SlowMotionTimeline()

```

```
{
    if (timeline != null)
        timeline.playableGraph.GetRootPlayable(0).SetSpeed(0.7);
}
```

```
//END-----Timeline manipulation-----
```

```
//START-----SFX-----
```

```
public void BattlePreparations()
```

```
{
```

```
    src2.Play();
```

```
}
```

```
public void InBattleSFX()
```

```
{
```

```
    src.clip = sfx_clash;
```

```
    src.Play();
```

```
}
```

```
public void CorrectChoiseSFX()
```

```
{
```

```
    src.clip = sfx_correct;
```

```
    src.Play();
```

```

}

public void WinSFX()
{
    src.clip = sfx_win;
    src.Play();

}
//END-----SFX-----

```

```

public void Win()
{
    win_ui.SetActive(true);
    WinSFX();
    restartButton.SetActive(false);
    homeButton.SetActive(false);
}

```

```

public void HideAllQuestions()
{
    if (question1 != null) question1.SetActive(false);
    if (question2 != null) question2.SetActive(false);
    if (phase1_2question != null) phase1_2question.SetActive(false);
    if (phase1_3question != null) phase1_3question.SetActive(false);
    if (phase2question != null) phase2question.SetActive(false);
    if (phase3question != null) phase3question.SetActive(false);
    if (wrongImageAnimation != null) wrongImageAnimation.SetActive(false);
    if (inBattle != null) inBattle.SetActive(false);
    if (win_ui != null) win_ui.SetActive(false);

}

```

```
public void OnButtonClickArxi()
{
    HideArxikiQuestion();
    ShowQuestion1();
}
```

```
public void CorrectBattleground()
{

    battleground.SetActive(true);
    ShowQuestion2();
    question1.SetActive(false);

}
```

```
//START-----UI manipulation-----
```

```
public void HideArxikiQuestion()
{
    arxikiQuestion.SetActive(false);
}
```

```
public void HidePhase3Question()
{
    phase3question.SetActive(false);
}
```

```
public void ShowQuestion1()
{
    question1.SetActive(true);
}
```

```
public void ShowQuestion2()
{
```

```

        question2.SetActive(true);
    }

    public void Showphase1_2question()
    {
        phase1_2question.SetActive(true);
    }
    public void Showphase1_3question()
    {
        phase1_3question.SetActive(true);
    }
    public void Showphase2question()
    {
        phase2question.SetActive(true);
    }
    public void Showphase3question()
    {
        phase3question.SetActive(true);
    }
    public void DipslayArxikiQuestion()
    {
        arxikiQuestion.SetActive(true);
    }

    //END-----UI manipulation-----
}

```

ΠΑΡΑΡΤΗΜΑ Ε : QuestionHandlerThermopylae.cs C# Script

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

public class QuestionHandlerThermopylae : MonoBehaviour
{
    //START--VARIABLES--UIs
    public GameObject prephaseUI;
    public GameObject Narration1UI;
    public GameObject Narration2UI;
    public GameObject Narration3UI;
    public GameObject question1UI;
    public GameObject question2UI;
    public GameObject question2_1UI;
    public GameObject question2_2UI;
    public GameObject question2_3UI;
    public GameObject efialtisQuestionUI;
    public GameObject efialtisUI;
    public GameObject ydarnisUI;
    public GameObject endOfDay2UI;
    public GameObject endOfDay1UI;
    public GameObject selectionOfTroopsUI;
    public GameObject phase1UI;
    public GameObject win_ui;
    public GameObject wrongImageAnimation;
    public GameObject inBattle;
    //END--VARIABLES--UIs

    //START--VARIABLES--OBJECTS
    public GameObject battalionPS1;
```

```
public GameObject battalionPS2;
public GameObject battalionPS3;
public GameObject battalionPS5;
public GameObject thespieis;
public GameObject front;
public GameObject back;
public GameObject row1;
public GameObject row2;
public GameObject row3;
public GameObject row1b2;
public GameObject row2b2;
public GameObject row3b2;
//END--VARIABLES--OBJECTS
```

```
//START--VARIABLES--SFXs
public AudioSource src;
public AudioSource src2;
public AudioSource src3;
public AudioSource src4;
public AudioClip sfx_correct, sfx_wrong, sfx_pre_war_narration, sfx_steps,sfx_war_drum,
sfx_narration2,sfx_after_destruction,sfx_march, sfx_win, sfx_clash;
public AudioClip sfx_question2UI, sfx_question2_1UI, sfx_question2_2UI, sfx_question2_3UI,
sfx_YdarnisUI,sfx_endofday2UI, sfx_endofday1UI, sfx_EfialtisUI;
//END--VARIABLES--SFXs
```

```
//START--VARIABLES--SELECTION OF TROOPS VARIABLES
bool boolean = false;
public SelectionOfGreeks selg;
public SelectionOfPersians selp;

//END--VARIABLES--SELECTION OF TROOPS
```

```
public GameObject battleground;  
public GameObject restartButton;  
public GameObject homeButton;  
public PlayableDirector timeline;
```

```
void Start()
```

```
{  
    battalionPS5.SetActive(false);  
    thespieis.SetActive(false);  
    battleground.SetActive(false);  
    PauseTimeline();  
    HideAllQuestions();  
}
```

```
void Update()
```

```
{  
    //CHECK IF BOTH TROOPS ARE SELECTED  
    if (boolean ==false)  
    {  
        if (selg.j == true && selp.j == true)  
        {  
            HideSelectionUI();  
            ShowPhase1UI();  
            boolean = true;  
        }  
    }  
}
```

```
public void Row1Destruction(){
    row1.SetActive(false);
}
```

```
public void Row2Destruction(){
    row2.SetActive(false);
}
```

```
public void Row3Destruction(){
    row3.SetActive(false);
    battalionPS1.SetActive(false);
}
```

```
public void Row1Destructionb2()
{
    row1b2.SetActive(false);
}
```

```
public void Row2Destructionb2()
{
    row2b2.SetActive(false);
}
```

```
public void Row3Destructionb2()
{
    row3b2.SetActive(false);
}
```

```

    battalionPS2.SetActive(false);
}
public void YdarnisDestruction()
{
    battalionPS3.SetActive(false);
}

//START-----Narations-----

public void PreWarNaration(){
    src.clip = sfx_pre_war_narration;
    src.Play();
    Narration1UI.SetActive(true);
}

public void NarationQuestion2()
{
    src.clip = sfx_question2UI;
    src.Play();
}

public void NarationQuestion2_1()
{
    src.clip = sfx_question2_1UI;
    src.Play();
}

public void NarationQuestion2_2()
{
    src.clip = sfx_question2_2UI;
    src.Play();
}
}

```

```

public void NarationQuestion2_3()
{
    src.clip = sfx_question2_3UI;
    src.Play();

}

public void NarationYdarnis()
{
    src.clip = sfx_YdarnisUI;
    src.Play();
    ydarnisUI.SetActive(true);

}

public void NarationEndOfDay2()
{
    src.clip = sfx_endofday2UI;
    src.Play();
    endOfDay2UI.SetActive(true);
    HideBACK();
    ShowThespieis();

}

public void NarationEndOfDay1()
{
    src.clip = sfx_endofday1UI;
    src.Play();
    endOfDay1UI.SetActive(true);

}

public void AfterDestructionOf1stBattalionNaration()
{
    src.clip = sfx_after_destruction;

```

```

src.Play();
Narration3UI.SetActive(true);

}

public void NarationEfialtis()
{
    src.clip = sfx_EfialtisUI;
    src.Play();
    efialtisUI.SetActive(true);

}

//END-----Narations-----

```

```

public void DefeatOfGreeks()
{
    thespieis.SetActive(false);
    front.SetActive(false);

}

```

```

public void CorrectSelectionQuestion1()
{
    src.clip = sfx_narration2;
    src.Play();
    Narration2UI.SetActive(true);

}

```

```

//START-----SFX-----

```

```
public void PlayMarchSFX()
{
    src3.clip = sfx_march;
    src3.Play();

}
public void StopMarchSFX()
{
    src3.Stop();

}

public void BackgroundMusic()
{
    src2.Play();
}
public void WinSFX()
{
    src.clip = sfx_win;
    src.Play();

}
public void InBattleSFX()
{

    src4.clip = sfx_clash;
    src4.Play();

}
//END-----SFX-----
```

```
//START-----CORRECT ANSWER-----  
public void CorrectChoise()  
{  
  
    // HideAllQuestions();  
    src4.clip = sfx_correct;  
    src4.Play();  
  
}  
//END-----CORRECT ANSWER-----
```

```
// START----manipulate inBattle UI-----  
public void ShowInBattleUI()  
{  
  
    inBattle.SetActive(true);  
    Invoke("HideInBattleUI", 2f);  
  
}  
public void HideInBattleUI()  
{  
    inBattle.SetActive(false);  
}  
  
//END-----manipulate inBattle UI-----
```

```
//START----- manipulate wrong answer-----
```

```

public void ShowWrongAnswerUI()
{

    wrongImageAnimation.SetActive(true);
    Invoke("HideWrongAnswerUI", 2f);

}
public void HideWrongAnswerUI()
{
    wrongImageAnimation.SetActive(false);
}

public void WrongAnswerSFX()
{
    src.clip = sfw_wrong;
    src.Play();

}

//END-----manipulate wrong answer-----

//START-----Questions & UIs manipulation-----

public void OnButtonClickArxi()
{
    battleground.SetActive(true);
    HidePrephaseUIQuestion();
    ShowSelectionUI();
}
public void ShowPrephaseUI()
{
    prephaseUI.SetActive(true);
}

```

```
public void ShowQuestion1()
{
    question1UI.SetActive(true);
}
public void ShowQuestion2()
{
    question2UI.SetActive(true);
}
public void ShowQuestion2_1()
{
    question2_1UI.SetActive(true);
}
public void ShowQuestion2_2()
{
    question2_2UI.SetActive(true);
}
public void ShowQuestion2_3()
{
    question2_3UI.SetActive(true);
}

public void showEfialtisQuestionUI()
{
    efialtisQuestionUI.SetActive(true);
}

public void hideNarration1UI()
{
    Narration1UI.SetActive(false);
}

public void hideNarration2UI()
{
```

```

    Narration2UI.SetActive(false);
}

public void hideNarration3UI()
{
    Narration3UI.SetActive(false);
}

public void Win()
{
    win_ui.SetActive(true);
    //WinSFX();
    restartButton.SetActive(false);
    homeButton.SetActive(false);
}

public void HideAllQuestions()
{
    if (efialtisUI != null) efialtisUI.SetActive(false);
    if (efialtisQuestionUI != null) efialtisQuestionUI.SetActive(false);
    if (question1UI != null) question1UI.SetActive(false);
    if (question2UI != null) question2UI.SetActive(false);
    if (question2_1UI != null) question2_1UI.SetActive(false);
    if (question2_2UI != null) question2_2UI.SetActive(false);
    if (question2_3UI != null) question2_3UI.SetActive(false);
    if (ydarnisUI != null) ydarnisUI.SetActive(false);
    if (endOfDay1UI != null) endOfDay1UI.SetActive(false);
    if (endOfDay2UI != null) endOfDay2UI.SetActive(false);
    if (selectionOfTroopsUI != null) selectionOfTroopsUI.SetActive(false);
    if (wrongImageAnimation != null) wrongImageAnimation.SetActive(false);
    if (phase1UI != null) phase1UI.SetActive(false);
    if (prephaseUI != null) prephaseUI.SetActive(false);
    if (win_ui != null) win_ui.SetActive(false); }

```

```
public void ShowPhase1UI()
{
    phase1UI.SetActive(true);
}
```

```
public void HidePrephaseUIQuestion()
{
    prephaseUI.SetActive(false);
}
```

```
public void DipslayPrephaseUI()
{
    prephaseUI.SetActive(true);
}
```

```
public void ShowSelectionUI()
{

    selectionOfTroopsUI.SetActive(true);

}
```

```
public void HideSelectionUI()
{

    selectionOfTroopsUI.SetActive(false);

}
```

```
public void HideQuestion2()
{
```

```

        question2UI.SetActive(false);
    }
    public void HideQuestion2_1UI()
    {

        question2_1UI.SetActive(false);

    }
    public void HideQuestion2_2UI()
    {

        question2_2UI.SetActive(false);

    }
    public void HideQuestion2_3UI()
    {

        question2_3UI.SetActive(false);

    }
    public void HideYdarnisUI()
    {

        ydarnisUI.SetActive(false);

    }
    public void HideQEndOfDay1UI()
    {

        endOfDay1UI.SetActive(false);

    }
    public void HideQEndOfDay2UI()

```

```

{

    endOfDay2UI.SetActive(false);

}

public void HideEfialtisUI()
{

    efialtisUI.SetActive(false);

}

public void HideEfialtisQuestionUI()
{

    efialtisQuestionUI.SetActive(false);

}

public void ShowBatallionPS5()
{
    battalionPS5.SetActive(true);
}

public void ShowThespieis()
{
    thespieis.SetActive(true);
}

public void HideBACK()
{
    back.SetActive(false);
}

//END-----Questions & UIs manipulation-----

```

```

//START-----Timeline manipulation-----
public void playTimeline()
{
    timeline.Play();
    //generals_timeline.Play();
}
public void PauseTimeline()
{

    if (timeline != null)
        timeline.playableGraph.GetRootPlayable(0).SetSpeed(0);

}

public void ResumeTimeline()
{
    if (timeline != null)
        timeline.playableGraph.GetRootPlayable(0).SetSpeed(0.7);
}
public void SlowMotionTimeline()
{
    if (timeline != null)
        timeline.playableGraph.GetRootPlayable(0).SetSpeed(0.5);

}
//END-----Timeline manipulation-----
}

```

IIAPAPTHMA F : SelectionOfPersians.cs C# Script

```

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

```

```

public class SelectionOfPersians : MonoBehaviour
{

    public GameObject persianCollider;
    public GameObject persianSoldiers;
    public GameObject persianArmy;
    public AudioSource src;
    public AudioClip sfx_correct;
    public bool j;

    // Start is called before the first frame update

    void Start()
    {

        persianSoldiers.SetActive(false);
        persianCollider.SetActive(true);
    }

    private void OnTriggerEnter(Collider other)
    {

        if (other.gameObject.tag == "PersianWarrior")
        {
            GameObject[] persians = GameObject.FindGameObjectsWithTag("PersianWarrior");
            foreach (GameObject obj2 in persians)
                obj2.SetActive(false);
            persianArmy.SetActive(false);
            persianSoldiers.SetActive(true);
            persianCollider.SetActive(false);
        }
    }
}

```

```

        src.clip = sfx_correct;
        src.Play();
        j = true;
    }
}
}

```

ΠΑΡΑΡΤΗΜΑ Γ : SelectionOfGreeks.cs C# Script

```

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

public class SelectionOfGreeks : MonoBehaviour
{
    public GameObject greekCollider;
    public GameObject greekSoldiers;
    public GameObject greekArmy;
    public AudioSource src;
    public AudioClip sfx_correct;
    public bool j;
    // Start is called before the first frame update

    void Start()
    {

        greekSoldiers.SetActive(false);
        greekCollider.SetActive(true);

    }

    private void OnTriggerEnter(Collider other)

```

```

{

if (other.gameObject.tag=="GreekWarrior")
{
    GameObject[] greeks = GameObject.FindGameObjectsWithTag("GreekWarrior");
    foreach (GameObject obj in greeks)
        obj.SetActive(false);
    Debug.Log("TO BLEPW greek");
    greekArmy.SetActive(false);
    greekSoldiers.SetActive(true);
    greekCollider.SetActive(false);
    src.clip = sfx_correct;
    src.Play();
    j = true;
}
}
}

```

ΠΑΡΑΡΤΗΜΑ Η : DragUIelement.cs C# Script

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;

public class DragUIelement : MonoBehaviour,
    IBeginDragHandler,
    IDragHandler,
    IEndDragHandler

{
    //this is the prefab i want to create on drag and drop from the UI ELEMENT
    [SerializeField]
    GameObject prefabToInstantiate;

```

```

[SerializeField]
RectTransform UIDragElement;

//cache the reference to the canvas
[SerializeField]
RectTransform Canvas;

//private data to store pointer positions
private Vector2 mOriginalLocalPointerPosition;
private Vector3 mOriginalPanelLocalPosition;
private Vector3 mOriginalPosition;

void Start()
{
    //we store the initial position of the draggable ui Item
    mOriginalPosition = UIDragElement.localPosition;
}

public void OnBeginDrag(PointerEventData data)
{
    mOriginalPanelLocalPosition = UIDragElement.localPosition;
    RectTransformUtility.ScreenPointToLocalPointInRectangle(
        Canvas,
        data.position,
        data.pressEventCamera,
        out mOriginalLocalPointerPosition);
}

public void OnDrag(PointerEventData data)
{
    Vector2 localPointerPosition;
    if (RectTransformUtility.ScreenPointToLocalPointInRectangle(

```

```

Canvas,
data.position,
data.pressEventCamera,
out localPointerPosition))
{
Vector3 offsetToOriginal =
    localPointerPosition -
    mOriginalLocalPointerPosition;

UIDragElement.localPosition =
    mOriginalPanelLocalPosition +
    offsetToOriginal;
}

}

public void OnEndDrag(PointerEventData data)
{
StartCoroutine(
    Coroutine_MoveUIElement(
        UIDragElement,
        mOriginalPosition,
        0.5f));

RaycastHit hit;
Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
if(Physics.Raycast(ray,out hit,1000.0f))
{
Vector3 worldpoint = hit.point;
CreateObject(worldpoint);

}

}

```

```

void CreateObject(Vector3 pos)
{
    //here we instantiate the object
    if (prefabToInstantiate==null)
    {
        Debug.Log("NO prefab to instantiate");
        return;
    }
    GameObject obj = Instantiate(
        prefabToInstantiate,
        pos,
        Quaternion.identity);
}

```

```

IEnumerator Coroutine_MoveUIElement(
    RectTransform r,
    Vector2 targetPosition,
    float duration = 0.1f)
{
    float elapsedTime = 0;
    Vector2 startingPos = r.localPosition;
    while (elapsedTime < duration)
    {
        r.localPosition=Vector2.Lerp(
            startingPos,
            targetPosition,
            (elapsedTime/duration));
        elapsedTime += Time.deltaTime;
        yield return new WaitForEndOfFrame();
    }
}

```

```

        r.localPosition = targetPosition;
    }

}

```

ΠΑΡΑΡΤΗΜΑ Ι : WrongAnimationSalamina.cs C# Script

```

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

public class WrongAnimationSalamina : MonoBehaviour
{
    public Animator mAnimator;
    // Start is called before the first frame update
    void Start()
    {
        mAnimator = GetComponent<Animator>();
    }
    public void ResetTrigger()
    {
        mAnimator.ResetTrigger("isPlayed");
    }

    public void PlayWrongAnimation()
    {
        mAnimator.SetTrigger("isPlayed");
        Invoke("ResetTrigger",2f);
    }
}

```

ΠΑΡΑΡΤΗΜΑ J : QuestionHandlerSalamina.cs C# Script

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

public class QuestionHandlerSalamina : MonoBehaviour
{

    public GameObject arxi;
    public GameObject arxi2;
    public GameObject arqing;
    public GameObject agreed;
    public GameObject prewar;
    public GameObject startBattle;
    public GameObject phase1;
    public GameObject question1;
    public GameObject phase2;
    public GameObject question2;
    public GameObject phase3;
    public GameObject question3;
    public GameObject phase4;
    public GameObject phase5;
    public GameObject inBattle;
    public GameObject win;

    public GameObject wrongAnswer;

    public GameObject buttonSTART;

    public AudioSource src;
    public AudioSource src3;
    public AudioClip sfx_correct, sfx_wrong,sfx_win , sfx_clash;
```

```

public GameObject restartButton;
public GameObject homeButton;

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

    ShowArxiUI();
    HideButtonSTART();
}

//----- energopoihsh UI
public void ShowArxiUI()
{
    arxi.SetActive(true);
}
public void ShowArxi2UI()
{
    arxi2.SetActive(true);
}
public void ShowArquingUI()
{
    arquing.SetActive(true);
}
public void ShowAgreedUI()
{
    agreed.SetActive(true);
}
public void ShowPrewarUI()
{
    prewar.SetActive(true);
}

```

```
}

public void ShowStartBattleUI()
{
    startBattle.SetActive(true);
}

public void ShowPhase1UI()
{
    phase1.SetActive(true);
}

public void ShowQuestion1UI()
{
    question1.SetActive(true);
}

public void ShowPhase2UI()
{
    phase2.SetActive(true);
}

public void ShowQuestion2UI()
{
    question2.SetActive(true);
}

public void ShowPhase3UI()
{
    phase3.SetActive(true);
}

public void ShowQuestion3()
{
    question3.SetActive(true);
}

public void ShowPhase4UI()
{
    phase4.SetActive(true);
}
```

```

}
public void ShowPhase5UI()
{
    phase5.SetActive(true);
}

public void ShowButtonSTART()
{
    buttonSTART.SetActive(true);
}

public void ShowWrongAnswerUI()
{
    wrongAnswer.SetActive(true);
    Invoke("HideWrongAnswerUI", 2f);
}

public void Win()
{
    win.SetActive(true);
    WinSFX();
    restartButton.SetActive(false);
    homeButton.SetActive(false);
}

//-----

// START----manipulate inBattle UI-----
public void ShowInBattleUI()
{

```

```

    inBattle.SetActive(true);
    Invoke("HideInBattleUI", 2f);

}
public void HideInBattleUI()
{
    inBattle.SetActive(false);
}
public void InBattleSFX()
{

    src.clip = sfx_clash;
    src.Play();

}

//END-----manipulate inBattle UI-----

//START-----SFX-----

public void BackgroundWaves()
{
    src3.Play();
}

public void WinSFX()
{
    src.clip = sfx_win;
    src.Play();
}

```

```
}
```

```
//END-----SFX-----
```

```
//-----apenergopoihsh UI
```

```
public void HideArxiUI()
```

```
{
```

```
    arxi.SetActive(false);
```

```
}
```

```
public void HideArxi2UI()
```

```
{
```

```
    arxi2.SetActive(false);
```

```
}
```

```
public void HideArquingUI()
```

```
{
```

```
    arquing.SetActive(false);
```

```
}
```

```
public void HideAgreedUI()
```

```
{
```

```
    agreed.SetActive(false);
```

```
}
```

```
public void HidePrewarUI()
```

```
{
```

```
    prewar.SetActive(false);
```

```
}
```

```
public void HideStartBattleUI()
```

```
{
```

```
    startBattle.SetActive(false);
```

```
}
```

```
public void HidePhase1UI()
```

```
{
```

```
    phase1.SetActive(false);
}
public void HideQuestion1UI()
{
    question1.SetActive(false);
}
public void HidePhase2UI()
{
    phase2.SetActive(false);
}
public void HideQuestion2UI()
{
    question2.SetActive(false);
}
public void HidePhase3UI()
{
    phase3.SetActive(false);
}
public void HideQuestion3()
{
    question3.SetActive(false);
}
public void HidePhase4UI()
{
    phase4.SetActive(false);
}
public void HidePhase5UI()
{
    phase5.SetActive(false);
}
```

```

public void HideButtonSTART()
{
    buttonSTART.SetActive(false);
}

public void HideWrongAnswerUI()
{
    wrongAnswer.SetActive(false);
}
//-----

//sound effect if correct
public void SFXCorrectChoise()
{

    src.clip = sfx_correct;
    src.Play();
}
//sound effect if wrong
public void SFXWrongChoise()
{

    src.clip = sfx_wrong;
    src.Play();
    HideWrongAnswerUI();

}
}

```

ΠΑΡΑΡΤΗΜΑ Κ : SplineAnimationController.cs C# Script

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

public class SplineAnimationController : MonoBehaviour
{

    [SerializeField] SplineFollower round1leftPersianBattallions;
    [SerializeField] SplineFollower round1rightPersianBattallions;
    [SerializeField] SplineFollower greekBattallionFrontLeft1;
    [SerializeField] SplineFollower greekBattallionFrontLeft2;
    [SerializeField] SplineFollower greekBattallionFrontLeft3;
    [SerializeField] SplineFollower greekBattallionBackRight4;
    [SerializeField] SplineFollower greekBattallionBackRight5;
    [SerializeField] SplineFollower round2PersianBattallion1;
    [SerializeField] SplineFollower round2PersianBattallion2;
    [SerializeField] SplineFollower round2PersianBattallion3;
    [SerializeField] SplineFollower round2PersianBattallion4;
    [SerializeField] SplineFollower round2PersianBattallion5;
    public GameObject round1PersianBattalions;
    public GameObject round2PersianBattalions;
    public GameObject round2PersianBattalion1;
    public GameObject round2PersianBattalion2;
    public GameObject round2PersianBattalion3;

    // Start is called before the first frame update
    void Start()
    {
        StopALLAnimation();
        DisableROUND2();
    }
}
```

```

    EnableROUND1();
}

public void StartAnimation()
{
    round1leftPersianBattallions.enabled = true;
    round1rightPersianBattallions.enabled = true;
}

public void StartROUND2AnimationPERSIANS()
{

    round2PersianBattalion1.enabled = true;
    round2PersianBattalion2.enabled = true;
    round2PersianBattalion3.enabled = true;
    round2PersianBattalion4.enabled = true;
    round2PersianBattalion5.enabled = true;
}

public void StartGREEKSAnimation()
{
    greekBattalionFrontLeft1.enabled = true;
    greekBattalionFrontLeft2.enabled = true;
    greekBattalionFrontLeft3.enabled = true;
    greekBattalionBackRight4.enabled = true;
    greekBattalionBackRight5.enabled = true;
}

public void StopAnimation()
{

    round1leftPersianBattallions.enabled = false;
    round1rightPersianBattallions.enabled = false;
}

```

```
public void StopROUND2AnimationPERSIANS()
{

    round2PersianBattalion1.enabled = false;
    round2PersianBattalion2.enabled = false;
    round2PersianBattalion3.enabled = false;
    round2PersianBattalion4.enabled = false;
    round2PersianBattalion5.enabled = false;
}
```

```
public void StopGREEKSAnimation()
{

    greekBattalionFrontLeft1.enabled = false;
    greekBattalionFrontLeft2.enabled = false;
    greekBattalionFrontLeft3.enabled = false;
    greekBattalionBackRight4.enabled = false;
    greekBattalionBackRight5.enabled = false;
}
```

```
public void StopALLAnimation()
{

    round1leftPersianBattalions.enabled = false;
    round1rightPersianBattalions.enabled = false;
    greekBattalionFrontLeft1.enabled = false;
    greekBattalionFrontLeft2.enabled = false;
    greekBattalionFrontLeft3.enabled = false;
    greekBattalionBackRight4.enabled = false;
    greekBattalionBackRight5.enabled = false;
    round2PersianBattalion1.enabled = false;
    round2PersianBattalion2.enabled = false;
    round2PersianBattalion3.enabled = false;
}
```

```

    round2PersianBattalion4.enabled = false;
    round2PersianBattalion5.enabled = false;
}
public void Start12Animation()
{

    greekBattalionFrontLeft1.enabled = true;
    greekBattalionFrontLeft2.enabled = true;
    round2PersianBattalion1.enabled = true;
    round2PersianBattalion2.enabled = true;

}
public void Stop12Animation()
{

    greekBattalionFrontLeft1.enabled = false;
    greekBattalionFrontLeft2.enabled = false;
    round2PersianBattalion1.enabled = false;
    round2PersianBattalion2.enabled = false;
}

public void Start345Animation()
{

    greekBattalionFrontLeft3.enabled = true;
    greekBattalionBackRight4.enabled = true;
    greekBattalionBackRight5.enabled = true;
    round2PersianBattalion3.enabled = true;
    round2PersianBattalion4.enabled = true;
    round2PersianBattalion5.enabled = true;
}
public void Stop345Animation()
{

```

```
greekBattalionFrontLeft3.enabled = false;
greekBattalionBackRight4.enabled = false;
greekBattalionBackRight5.enabled = false;
round2PersianBattalion3.enabled = false;
round2PersianBattalion4.enabled = false;
round2PersianBattalion5.enabled = false;
}
```

```
public void EnableROUND1()
{
    round1PersianBattalions.SetActive(true);
}
```

```
public void EnableROUND2()
{
    round2PersianBattalions.SetActive(true);
}
```

```
public void DisableROUND1()
{
    round1PersianBattalions.SetActive(false);
}
```

```
public void DisableROUND2()
{
    round2PersianBattalions.SetActive(false);
}
```

```
public void DestructPersianBattalion1()
{
    round2PersianBattalion1.SetActive(false);
}
```

```
public void DestructPersianBattalion2()
```

```
{  
    round2PersianBattalion2.SetActive(false);  
}  
public void DestructPersianBattalion3()  
{  
    round2PersianBattalion3.SetActive(false);  
}  
}
```