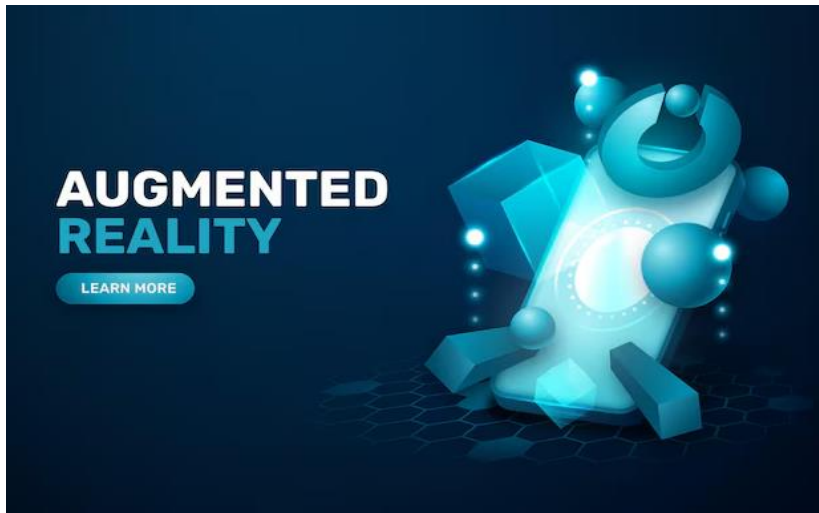


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Σχεδίαση εκπαιδευτικού παιχνιδιού επαυξημένης πραγματικότητας για τη διδασκαλία της ζωής και των ταξιδιών των Μεγάλων Εξερευνητών που αναφέρονται στην Ιστορία της ΣΤ΄ Δημοτικού»



Του φοιτητή

Γεώργιου Ελευθερούδη

Αρ. Μητρώου: 175019

Επιβλέπων

Όνοματεπώνυμο: Κεραμόπουλος Ευκλείδης

Καθηγητής

Ημερομηνία 06/05/2025

Τίτλος Δ. Ε.: Σχεδίαση εκπαιδευτικού παιχνιδιού επαυξημένης πραγματικότητας για τη διδασκαλία της ζωής και των ταξιδιών των Μεγάλων Εξερευνητών που αναφέρονται στην Ιστορία της ΣΤ΄ Δημοτικού

Κωδικός Δ.Ε.: 24187

Όνοματεπώνυμο φοιτητή: Γεώργιος Ελευθερούδης

Όνοματεπώνυμο εισηγητή: Κεραμόπουλος Ευκλείδης

Ημερομηνία ανάληψης Δ.Ε.: 01/04/2024

Ημερομηνία περάτωσης Δ.Ε.: 06/05/2025

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

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





## Πρόλογος

Η συγκεκριμένη εργασία επιλέχθηκε για ποικίλους λόγους. Αρχικά, ο προγραμματισμός με ενδιαφέρει πολύ στον τομέα της πληροφορικής και το συγκεκριμένο project απαιτεί προγραμματισμό. Επίσης, ανέκαθεν έβρισκα ενδιαφέροντα τα παιχνίδια, οπότε υπήρχε ένας επιπλέον λόγος να επιλέξω τη συγκεκριμένη εργασία, καθώς αφορά στην ανάπτυξη ενός παιχνιδιού. Η ενασχόληση με την επαυξημένη πραγματικότητα και η εκμάθηση ενός νέου εργαλείου για τον προγραμματισμό, που στο συγκεκριμένο project είναι το περιβάλλον ανάπτυξης Unity, αποτελούσε πρόκληση, καθώς δεν είχα ασχοληθεί με κάτι παρόμοιο στο παρελθόν. Πάντοτε με έλκυε η ιδέα ανάπτυξης μιας εφαρμογής, ιδιαίτερα ενός παιχνιδιού. Επιπλέον, μου φάνηκε ενδιαφέρουσα η θεματολογία της συγκεκριμένης εργασίας, που αφορά στους θαλασσοπόρους-εξερευνητές του 15ου-16ου αιώνα. Η εφαρμογή που αναπτύχθηκε βασίζεται στην επαυξημένη πραγματικότητα. Σκοπός της ένταξης της επαυξημένης πραγματικότητας στον χώρο της εκπαίδευσης είναι να εισάγει τον μαθητή σε έναν καινούριο, απλό και ευχάριστο, τρόπο εκμάθησης που διαφέρει από τον παραδοσιακό, καθώς δίνεται η δυνατότητα να χρησιμοποιηθούν ψηφιακά μέσα –υπολογιστής, κινητό τηλέφωνο, τάμπλετ– για τη διδασκαλία, η οποία με τον τρόπο αυτό ενισχύεται σημαντικά. Η παρούσα διπλωματική εργασία επικεντρώνεται στη χρήση εκτεταμένης πραγματικότητας στον τομέα της εκπαίδευσης και συγκεκριμένα στη διδασκαλία του μαθήματος της Ιστορίας.

## Περίληψη

Η παρούσα εργασία είναι η ανάπτυξη μιας εφαρμογής-παιχνιδιού επαυξημένης πραγματικότητας για εκπαιδευτικούς σκοπούς στο μάθημα της Ιστορίας της ΣΤ΄ Δημοτικού. Το θέμα της πραγματεύεται τη ζωή των θαλασσοπόρων-εξερευνητών του 15ου-16ου αιώνα, των Βαρθολομαίου Ντιάζ, Φερδινάνδου Μαγγελάνου, Χριστόφορου Κολόμβου και Βάσκο ντα Γκάμα. Συγκεκριμένα αφορά στην «Ενότητα Α. Οι εξελίξεις στην Ευρώπη κατά τους Νεότερους Χρόνους (μέσα 15<sup>ου</sup> αιώνα-αρχές 19<sup>ου</sup> αιώνα)» και το «Κεφάλαιο 2. Από τις Γεωγραφικές Ανακαλύψεις στον Διαφωτισμό», που περιέχεται στις σελίδες 16-20 του σχολικού εγχειριδίου. Μελετώντας τη συγκεκριμένη ενότητα διαπιστώσαμε πως οι εκπαιδευτικοί διδάσκουν και τα ταξίδια του Μάρκο Πόλο, του πρώτου ευρωπαίου εξερευνητή που έκανε γνωστή την Κίνα στην Ευρώπη. Έτσι, θεωρήθηκε χρήσιμο να συμπεριληφθεί στο παιχνίδι και αυτός ο εξερευνητής –αν και ανήκει στον 13<sup>ο</sup> αιώνα– για την πληρέστερη κάλυψη του θέματος. Η εφαρμογή χωρίζεται σε πέντε παιχνίδια, ένα για κάθε εξερευνητή. Κάθε παιχνίδι αποτελείται από δύο μέρη. Το πρώτο –βασικό– μέρος περιλαμβάνει την κίνηση του χαρακτήρα πάνω σε έναν χάρτη που εμφανίζεται · το δεύτερο μέρος είναι τα μίνι παιχνίδια που εμφανίζονται σε συγκεκριμένους προορισμούς και αφορούν γεγονότα που συνέβησαν ή ανακαλύψεις που έγιναν κατά τη διάρκεια των ταξιδιών των θαλασσοπόρων. Και στα πέντε παιχνίδια υπάρχει ένας χάρτης και πάνω σε αυτόν ένα πλοίο, ώστε –με τη σωστή απάντηση των ερωτήσεων που εμφανίζονται– να σχηματίζει ο παίκτης τη διαδρομή του κάθε θαλασσοπόρου. Αυτό που διαφέρει, κυρίως, είναι τα μίνι παιχνίδια, καθώς οι θαλασσοπόροι έχουν βιώσει ποικίλες και διαφορετικές εμπειρίες · ορισμένοι έχουν συμμετάσχει σε μάχες, άλλοι ανακαλύπτουν νέους θαλάσσιους δρόμους και νέα προϊόντα, άλλοι συναντούν βασιλιάδες και κλείνουν εμπορικές συμφωνίες. Όλα αυτά τα σημαντικά γεγονότα εμφανίζονται με τη μορφή μίνι παιχνιδιών/πληροφοριών. Εφόσον πρόκειται για επαυξημένη πραγματικότητα, για να ξεκινήσει το παιχνίδι, ο παίκτης πρέπει να σκανάρει το ασπρόμαυρο πορτραίτο του θαλασσοπόρου, τον οποίο έχει επιλέξει και τότε θα εμφανιστεί ο χάρτης του παιχνιδιού.

Λέξεις-κλειδιά: Επαυξημένη Πραγματικότητα, Εικονική Πραγματικότητα, Μικτή Πραγματικότητα, Εκτεταμένη Πραγματικότητα, μηχανή παιχνιδιού Unity, Ιστορία

«Design of an augmented reality educational game for teaching the life and travels of the Great Explorers mentioned in the History of the Sixth Grade»

«Georgios Eleftheroydis»

## **Abstract**

The present work is the development of an augmented reality application-game for educational purposes in the history lesson of the 6th grade. The topic deals with the lives of the 15th-16th century seafaring explorers, Bartolomeu Dias, Ferdinand Magellan, Christopher Columbus and Vasco da Gama. Specifically, it concerns “Section A. Developments in Europe in Modern Times (mid-15th century-early 19th century)”, and “Chapter 2. From Geographical Discoveries to the Enlightenment”, contained on pages 16-20 of the textbook. In studying this section, we found that teachers also teach the travels of Marco Polo, the first European explorer who introduced China to Europe. Thus, it was considered useful to include this explorer in the game –even though he belongs to the 13th century– for a more complete coverage of the material. The application is divided into five games, one for each explorer. Each game consists of two parts. The first –basic– part involves moving the character on a map that appears. The second part is the mini-games that appear in specific destinations, and involve events that happened or discoveries made during the voyages of the navigators. In all five games there is a map with a ship on it so that –by correctly answering the questions that appear– the player forms the route of each navigator. What differs, above all, are the mini-games, as the navigators have had varied and different experiences – some have been in battles, others discover new sea routes and new products, others meet kings and make trade deals. All these important events are presented in the form of mini games/informations. Since this is augmented reality, to start the game the player must scan the black and white portrait of the chosen seafarer and then the game map will appear.

Keywords: Augmented Reality, Virtual Reality, Mixed Reality, Extended Reality, Unity game engine, History

## **Ευχαριστίες**

Ευχαριστώ τον επιβλέποντα καθηγητή μου, Κ. Κεραμόπουλο Ευκλείδη, με τον οποίο είχαμε μια πολύ καλή συνεργασία και συνεννόηση, για τις πολύτιμες συμβουλές και την καθοδήγησή του, καθώς και την οικογένειά μου, που με στήριξε ηθικά στην εκπόνηση της παρούσας εργασίας.

# Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract.....	vii
Ευχαριστίες.....	viii
Περιεχόμενα.....	ix
Κατάλογος Σχημάτων.....	xi
Συντομογραφίες.....	xv
Κεφάλαιο 1ο: Εισαγωγή.....	1
Κεφάλαιο 2ο: Οι τεχνολογίες του Εικονικού Κόσμου.....	3
2.1 Εισαγωγή.....	3
2.2 Στοιχειώδεις έννοιες. VR, AR & MR.....	3
2.3 Παραδείγματα.....	3
2.4 Διαφορές μεταξύ VR, AR και MR.....	5
2.5 Πεδία εφαρμογής.....	5
2.5.1 Αρχιτεκτονική και πολεοδομικές εφαρμογές.....	6
2.5.2 Στρατιωτικές εφαρμογές.....	6
2.5.3 Ιατρική.....	6
2.5.4 Βιομηχανία και κατασκευές.....	6
2.5.5 Πολιτισμός και Ιστορία.....	6
2.5.6 Εκπαίδευση και επαγγελματική κατάρτιση.....	6
2.6 Επίλογος.....	6
Κεφάλαιο 3ο: Το περιβάλλον ανάπτυξης του παιχνιδιού.....	9
3.1 Εισαγωγή.....	9
3.2 Η έννοια «Μηχανή Παιχνιδιών» (Game Engine).....	9
3.3 Το πρόγραμμα Unity.....	9
3.4 Βασικά στοιχεία και μηχανισμοί του Unity.....	9
3.4.1 Collider component.....	9
3.4.2 Raycast.....	10
3.4.3 Canvas.....	10
3.4.4 Transform.....	11
3.4.5 Inspector.....	11
3.4.6 Animations.....	12

3.4.7	Particle System .....	12
3.5	Επίλογος.....	12
Κεφάλαιο 4ο:	Η εφαρμογή AR στο μάθημα της Ιστορίας .....	13
4.1	Εισαγωγή .....	13
4.2	Σκοπός σχεδίασης της εφαρμογής.....	13
4.3	Τα παιχνίδια .....	16
4.3.1	Το παιχνίδι του Βαρθολομαίου Ντιάζ.....	16
4.3.1.1	Βασικά στοιχεία του παιχνιδιού.....	16
4.3.1.2	Η λειτουργία των βασικών κουμπιών .....	19
4.3.1.3	Η λειτουργία του παιχνιδιού .....	24
4.3.1.4	Η κίνηση του πλοίου .....	32
4.3.2	Το παιχνίδι του Χριστόφορου Κολόμβου .....	37
4.3.2.1	Βασικά στοιχεία του παιχνιδιού.....	37
4.3.2.2	Η λειτουργία του παιχνιδιού.....	39
4.3.2.3	Το μίνι παιχνίδι drag and drop του Χριστόφορου Κολόμβου .....	44
4.3.3	Το παιχνίδι του Βάσκο ντα Γκάμα.....	47
4.3.3.1	Βασικά στοιχεία του παιχνιδιού.....	47
4.3.3.2	Η λειτουργία του παιχνιδιού .....	48
4.3.3.3	Το μίνι παιχνίδι επιλογής προϊόντων του Βάσκο ντα Γκάμα .....	56
4.3.4	Το παιχνίδι του Φερδινάνδου Μαγγελάνου .....	57
4.3.4.1	Βασικά στοιχεία του παιχνιδιού.....	57
4.3.4.2	Η λειτουργία του παιχνιδιού .....	58
4.3.4.3	Το μίνι παιχνίδι της μάχης του Μαγγελάνου στις Φιλιππίνες .....	65
4.3.5	Το παιχνίδι του Μάρκο Πόλο.....	70
4.3.5.1	Βασικά στοιχεία του παιχνιδιού.....	70
4.3.5.2	Η λειτουργία του παιχνιδιού.....	71
4.4	Επίλογος.....	76
Κεφάλαιο 5ο:	Συμπεράσματα και προτάσεις βελτίωσης.....	77
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	79

## Κατάλογος Σχημάτων

Σχήμα 2.1: Παράδειγμα εικονικής πραγματικότητας .....	4
Σχήμα 2.2: Παράδειγμα της επαυξημένης πραγματικότητας .....	4
Σχήμα 2.3: Παράδειγμα μικτής πραγματικότητας .....	5
Σχήμα 3.1: Το Collider component.....	10
Σχήμα 3.2: Παράδειγμα του αντικειμένου Canvas .....	10
Σχήμα 3.3: Παράδειγμα World Space Canvas.....	11
Σχήμα 4.1: Το αρχικό μενού.....	14
Σχήμα 4.2: Το Play Scene .....	14
Σχήμα 4.3: Το παράθυρο επιλογών.....	15
Σχήμα 4.4: Τα αρχεία του παιχνιδιού του Βαρθολομαίου Ντιάζ.....	16
Σχήμα 4.5: Εισαγωγικό πλαίσιο με το ξεκίνημα του παιχνιδιού του Βαρθολομαίου Ντιάζ .....	16
Σχήμα 4.6: Πλαίσιο οδηγιών για το πως παίζεται το παιχνίδι.....	17
Σχήμα 4.7: Ο χάρτης που αποτελεί το βασικό μέρος του παιχνιδιού του Βαρθολομαίου Ντιάζ ..	17
Σχήμα 4.8: Τα κουμπιά του παιχνιδιού και οι ζωές .....	18
Σχήμα 4.9: Σωστή απάντηση της πρώτης ερώτησης του βασικού παιχνιδιού .....	18
Σχήμα 4.10: Η κίνηση του πλοίου με το ξεκίνημα του ταξιδιού .....	19
Σχήμα 4.11: Οι βασικές μεταβλητές και η αρχικοποίηση του παιχνιδιού.....	20
Σχήμα 4.12: Τμήμα της μεθόδου createNewQuestion() .....	20
Σχήμα 4.13: Μέθοδος διαχείρισης του κουμπιού C.....	21
Σχήμα 4.14: Οι μέθοδοι σωστής και λανθασμένης απάντησης με τη σειρά .....	22
Σχήμα 4.15: Επιλογή λανθασμένης απάντησης.....	23
Σχήμα 4.16: Τέλος παιχνιδιού, όταν ο παίκτης χάνει στο παιχνίδι του Βαρθολομαίου Ντιάζ .....	23
Σχήμα 4.17: Το τέλος του παιχνιδιού, όταν ο παίκτης κερδίζει .....	24
Σχήμα 4.18: Η απάντηση της πρώτης ερώτησης .....	24
Σχήμα 4.19: Απάντηση στη δεύτερη ερώτηση .....	25
Σχήμα 4.20: Τοποθέτηση του πλοίου στη Λισαβόνα.....	25
Σχήμα 4.21: Ο τρόπος τοποθέτησης του πλοίου στη Λισαβόνα.....	26
Σχήμα 4.22: Ξεκίνημα του ταξιδιού με την αύξηση του currentDestination.....	26
Σχήμα 4.23: Το prefab της καταϊγίδας που εμφανίζεται .....	27
Σχήμα 4.24: Το μίνι παιχνίδι στον κόλπο Αλγκόα .....	27
Σχήμα 4.25: Τα Box Collider components των αντικειμένων στον κόλπο Αλγκόα .....	28

Σχήμα 4.26: Ο έλεγχος των αντικειμένων στο παιχνίδι drag and drop του Βαρθολομαίου Ντιάζ	28
Σχήμα 4.27: Η διαδικασία drag and drop του μίνι παιχνιδιού στο Algoa Bay	29
Σχήμα 4.28: Το μίνι παιχνίδι Quiz στο Ακρωτήρι της Καλής Ελπίδας	30
Σχήμα 4.29: Η τοποθέτηση του σταυρού	31
Σχήμα 4.30: Η επιστροφή του Βαρθολομαίου Ντιάζ	31
Σχήμα 4.31: Η επιτυχής ολοκλήρωση του παιχνιδιού του Βαρθολομαίου Ντιάζ	32
Σχήμα 4.32: Τα waypoints πάνω στον χάρτη	32
Σχήμα 4.33 Το στοιχείο Ship sphere	33
Σχήμα 4.34: Τα waypoints του path1	33
Σχήμα 4.35: Η μέθοδος characterMovement()	34
Σχήμα 4.36: Ο πίνακας path1	34
Σχήμα 4.37: Η μέθοδος ActivateCharacterMovement()	35
Σχήμα 4.38: Το sprite του πρώτου waypoint	35
Σχήμα 4.39: Τα sprites που χρησιμοποιούνται στο παιχνίδι	36
Σχήμα 4.40: Τμήμα της μεθόδου OnTriggerEnter() στην κίνηση του πλοίου	36
Σχήμα 4.41: Η μέθοδος infoPrefabSpawner()	37
Σχήμα 4.42: Ο φάκελος infoFolder του Βαρθολομαίου Ντιάζ	37
Σχήμα 4.43: Τα αρχεία στο παιχνίδι του Χριστόφορου Κολόμβου	38
Σχήμα 4.44: Το εισαγωγικό παράθυρο του παιχνιδιού του Χριστόφορου Κολόμβου	38
Σχήμα 4.45: Ο βασικός χάρτης του Χριστόφορου Κολόμβου	38
Σχήμα 4.46: Το πλοίο στην τοποθεσία Πάλος	39
Σχήμα 4.47: Κώδικας που τοποθετεί το πλοίο στο Πάλος στο παιχνίδι του Κολόμβου	39
Σχήμα 4.48: Ο Κολόμβος φτάνει στα Κανάρια νησιά	40
Σχήμα 4.49: Το prefab των Ινδιάνων	40
Σχήμα 4.50: Το μίνι παιχνίδι Quiz στις Μπαχάμες	41
Σχήμα 4.51: Το μίνι παιχνίδι Quiz στην Κούβα	41
Σχήμα 4.52: Το μίνι παιχνίδι Quiz στην Ισπανιόλα	42
Σχήμα 4.53: Το prefab του οχυρού στην Ισπανιόλα	42
Σχήμα 4.54: Η μέθοδος InstantiateFortress() στο παιχνίδι του Κολόμβου	43
Σχήμα 4.55: Το Transform component του βασικού χάρτη του Κολόμβου	43
Σχήμα 4.56: Το οχυρό ως Game Object «παιδί» του βασικού χάρτη στην ιεραρχία του Unity	43
Σχήμα 4.57: Το μίνι παιχνίδι drag and drop του Κολόμβου	44
Σχήμα 4.58: Δύο από τα σωστά προϊόντα έχουν ήδη μπει στο καλάθι του Κολόμβου	45

Σχήμα 4.59: Μέρος του κώδικα στο τελευταίο μίνι παιχνίδι του Κολόμβου .....	45
Σχήμα 4.60: Ο κώδικας που ολοκληρώνει το μίνι παιχνίδι drag and drop του Κολόμβου .....	46
Σχήμα 4.61: Το παράθυρο οδηγιών στο μίνι παιχνίδι drag and drop του Κολόμβου .....	46
Σχήμα 4.62: Το prefab του τελευταίου μίνι παιχνιδιού του Κολόμβου .....	47
Σχήμα 4.63: Ο φάκελος πληροφοριών του Κολόμβου .....	47
Σχήμα 4.64: Το εισαγωγικό παράθυρο του Βάσκο ντα Γκάμα .....	47
Σχήμα 4.65: Ο βασικός χάρτης του Βάσκο ντα Γκάμα.....	48
Σχήμα 4.66: Τοποθέτηση της πορτογαλικής σημαίας στον χάρτη του Βάσκο ντα Γκάμα .....	48
Σχήμα 4.67: Τοποθέτηση του πλοίου του Βάσκο ντα Γκάμα στη Λισαβόνα.....	49
Σχήμα 4.68: Το μίνι παιχνίδι της Μοζαμβίκης στο παιχνίδι του Βάσκο ντα Γκάμα .....	49
Σχήμα 4.69: Το μίνι παιχνίδι της Μοζαμβίκης σε μορφή prefab.....	50
Σχήμα 4.70: Το αρχείο V_battleScript στην καρτέλα Inspector.....	50
Σχήμα 4.71: Η μέθοδος checkForTouch() στο μίνι παιχνίδι της Μοζαμβίκης.....	51
Σχήμα 4.72: Η λειτουργία κίνησης των σφαιρών στο μίνι παιχνίδι της Μοζαμβίκης .....	51
Σχήμα 4.73: Η μέθοδος OnTriggerEnter() στο μίνι παιχνίδι της Μοζαμβίκης.....	52
Σχήμα 4.74: Τα αντικείμενα smoke στο μίνι παιχνίδι της Μοζαμβίκης .....	52
Σχήμα 4.75: Το αντικείμενο smoke πάνω στον βασικό χάρτη του Βάσκο ντα Γκάμα .....	53
Σχήμα 4.76: Το μνημείο στη μνήμη του Βάσκο ντα Γκάμα .....	53
Σχήμα 4.77: Το μίνι παιχνίδι Quiz στο Μαλίντι .....	54
Σχήμα 4.78: Ο πλοηγός που ταξιδεύει μαζί με το πλοίο μέχρι το Κοχικόδε.....	54
Σχήμα 4.79: Η πρώτη φάση του διαλόγου μεταξύ Βάσκο ντα Γκάμα και Ζαμορίν .....	55
Σχήμα 4.80: Η δεύτερη φάση του διαλόγου μεταξύ Βάσκο ντα Γκάμα και Ζαμορίν.....	55
Σχήμα 4.81: Η μέθοδος dialogueManagement() στον διάλογο του Βάσκο ντα Γκάμα .....	56
Σχήμα 4.82: Το εμπορικό κέντρο στο παιχνίδι του Βάσκο ντα Γκάμα.....	56
Σχήμα 4.83: Η μέθοδος endMiniGame(), στον θαλασσοπόρο Βάσκο ντα Γκάμα .....	57
Σχήμα 4.84: Ο βασικός χάρτης και τα αρχεία του παιχνιδιού του Μαγγελάνου .....	58
Σχήμα 4.85: Το πλοίο του Μαγγελάνου στο Ρίο ντε Τζανέιρο .....	58
Σχήμα 4.86: Το prefab με τον Μαγγελάνο και τους στασιαστές.....	59
Σχήμα 4.87: Τα Animation Controllers στο παιχνίδι του Μαγγελάνου .....	59
Σχήμα 4.88: Το AllStatesController.....	60
Σχήμα 4.89: Το tag των στασιαστών στην καρτέλα Inspector .....	60
Σχήμα 4.90: Η μέθοδος checkForTouch() στην εξουδετέρωση των στασιαστών .....	61
Σχήμα 4.91: Η μέθοδος Update() στην εξουδετέρωση των στασιαστών από τον Μαγγελάνο .....	61

Σχήμα 4.92: Η μέθοδος <code>attackMovement()</code> .....	61
Σχήμα 4.93: Η επίθεση του Μαγγελάνου στον πρώτο στασιαστή .....	62
Σχήμα 4.94: Η εξουδετέρωση του πρώτου στασιαστή .....	62
Σχήμα 4.95: Ο Μαγγελάνος, έχει εξοντώσει και τους τρεις στασιαστές .....	63
Σχήμα 4.96: Η μέθοδος <code>changeAnimation()</code> στο μίνι παιχνίδι με τους στασιαστές .....	63
Σχήμα 4.97: Ο Μαγγελάνος φτάνει στον πορθμό, στον οποίο ο ίδιος έδωσε το όνομά του .....	64
Σχήμα 4.98: Το Quiz κατά τη διάσχιση του Ειρηνικού ωκεανού από τον Μαγγελάνο .....	64
Σχήμα 4.99: Το prefab της μάχης του Μαγγελάνου στις Φιλιππίνες.....	65
Σχήμα 4.100: Η πρώτη φάση της μάχης του Μαγγελάνου στις Φιλιππίνες .....	65
Σχήμα 4.101: Η δεύτερη φάση της μάχης του Μαγγελάνου στις Φιλιππίνες.....	66
Σχήμα 4.102: Η τρίτη φάση της μάχης του Μαγγελάνου στις Φιλιππίνες.....	66
Σχήμα 4.103: Η τέταρτη φάση της μάχης του Μαγγελάνου στις Φιλιππίνες.....	67
Σχήμα 4.104: Ο στρατός του Μαγγελάνου έχει περικυκλωθεί από τους αντιπάλους του .....	67
Σχήμα 4.105: Η πέμπτη φάση της μάχης του Μαγγελάνου στις Φιλιππίνες .....	68
Σχήμα 4.106: Το τέλος της μάχης του Μαγγελάνου.....	68
Σχήμα 4.108: Οι μέθοδοι που αλλάζουν τα Animations των κόκκινων στρατιωτών .....	69
Σχήμα 4.109: Ο φάκελος με τα αρχεία των πληροφοριών του Μαγγελάνου .....	70
Σχήμα 4.110: Ο βασικός χάρτης του Μάρκο Πόλο .....	70
Σχήμα 4.111: Ο φάκελος με τις πληροφορίες του Μάρκο Πόλο.....	70
Σχήμα 4.112: Ο Μάρκο Πόλο φτάνει στην Περσία.....	71
Σχήμα 4.113: Το μίνι παιχνίδι της τέταρτης ερώτησης του Μάρκο Πόλο.....	72
Σχήμα 4.114: Ο Μάρκο Πόλο φτάνει στο Ορμούζ δια ξηράς.....	72
Σχήμα 4.115: Το Game Object CharSphere του Μάρκο Πόλο .....	73
Σχήμα 4.116: Η μέθοδος <code>activateCamel()</code> στο παιχνίδι του Μάρκο Πόλο .....	73
Σχήμα 4.117: Το μίνι παιχνίδι Quiz της ερήμου Γκόμπι.....	74
Σχήμα 4.118: Ο διάλογος μεταξύ του Μάρκο Πόλο και του Κουμπλάι Χαν.....	74
Σχήμα 4.119: Το prefab της ανεπτυγμένης πόλης της Κίνας .....	75
Σχήμα 4.120: Οι δύο στόλοι ανταλλάσσουν πυρά.....	75
Σχήμα 4.121: Ο στόλος της Βενετίας έχει ηττηθεί .....	76

## Συντομογραφίες

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

## Κεφάλαιο 1ο: Εισαγωγή

Τα τελευταία χρόνια αυξάνεται ολοένα και περισσότερο η χρήση της επαυξημένης πραγματικότητας και μέσω αυτής δημιουργούνται εφαρμογές που βελτιώνουν την καθημερινότητα του ανθρώπου. Η εκπαίδευση είναι ένας τομέας, όπου τα τελευταία χρόνια η επαυξημένη πραγματικότητα αξιοποιείται σημαντικά, στοχεύοντας στην αποτελεσματικότερη προσέγγιση της διδακτικής ύλης μέσα από μια πιο απλή και ευχάριστη διαδικασία.

Η παρούσα διπλωματική εργασία έχει τον τίτλο «Σχεδίαση εκπαιδευτικού παιχνιδιού επαυξημένης πραγματικότητας για τη διδασκαλία της ζωής και των ταξιδιών των Μεγάλων Εξερευνητών που αναφέρονται στην Ιστορία της ΣΤ΄ Δημοτικού». Στόχος της είναι η χρήση της επαυξημένης πραγματικότητας για εκπαιδευτικούς σκοπούς και, όπως φανερώνει ο τίτλος της, σχετίζεται με την ανάπτυξη ενός εκπαιδευτικού παιχνιδιού, που θα ωφελήσει τόσο τους εκπαιδευτικούς όσο και τους μαθητές.

Το μάθημα της Ιστορίας είναι απαιτητικό, καθώς περιέχει πλήθος πληροφοριών που οι μαθητές δυσκολεύονται να απομνημονεύσουν. Είναι, επομένως, σημαντικό να αναζητά κανείς μεθόδους, ώστε η διδασκαλία του να γίνεται με τρόπο ελκυστικό για τον μαθητή. Αυτός είναι ο στόχος της συγκεκριμένης εφαρμογής που αναπτύχθηκε και παρουσιάζεται στην παρούσα εργασία · να προσφέρει τη δυνατότητα στον εκπαιδευτικό να ποικίλει τη διδασκαλία με ένα χρήσιμο εκπαιδευτικό εργαλείο · να βοηθήσει τον μαθητή να προσεγγίσει τις ιστορικές γνώσεις με εναλλακτικό και κυρίως παιγνιώδη τρόπο. Μέσα από τη διάδραση η κοπιαστική εκμάθηση των ιστορικών γνώσεων μετατρέπεται σε παιχνίδι γνώσης.

Η παρούσα διπλωματική εργασία χωρίζεται σε πέντε κεφάλαια, εκ των οποίων το καθένα έχει ενότητες/υποενότητες. Συγκεκριμένα, ως προς τη δομή της η εργασία αρθρώνεται ως εξής:

Στο 1<sup>ο</sup> κεφάλαιο με τίτλο «Εισαγωγή» γίνεται μια σύντομη αναφορά στο περιεχόμενο, τον σκοπό και τη διάρθρωση της παρούσας διπλωματικής.

Στο 2<sup>ο</sup> κεφάλαιο με τίτλο «Οι τεχνολογίες του Εικονικού Κόσμου» παρουσιάζονται με σύντομο τρόπο οι τεχνολογίες της Εικονικής, Επαυξημένης και Μικτής Πραγματικότητας και οι διαφορές αυτών.

Στο 3<sup>ο</sup> κεφάλαιο με τίτλο «Το περιβάλλον ανάπτυξης του παιχνιδιού» γίνεται μια σύντομη παρουσίαση της έννοιας «Μηχανή Παιχνιδιού» και του λογισμικού ανάπτυξης Unity που χρησιμοποιήθηκε. Στη συνέχεια παρουσιάζονται ορισμένα βασικά στοιχεία και μηχανισμοί που αξιοποιήθηκαν στην παρούσα εφαρμογή.

Στο 4<sup>ο</sup> κεφάλαιο με τίτλο «Η εφαρμογή AR στο μάθημα της Ιστορίας» παρουσιάζεται η συγκεκριμένη εφαρμογή. Αρχικά αναφέρεται ο σκοπός ανάπτυξης αυτής, ενώ στη συνέχεια παρουσιάζονται τα παιχνίδια τα οποία είναι πέντε, ένα για κάθε θαλασσοπόρο-εξερευνητή. Σε κάθε παιχνίδι παρουσιάζονται τα βασικά του στοιχεία και η λειτουργία του, καθώς και ορισμένα σημαντικά τμήματα του κώδικα που αναπτύχθηκε για τον προγραμματισμό της εφαρμογής.

Στο 5<sup>ο</sup> κεφάλαιο παρουσιάζονται ορισμένα συμπεράσματα και προτείνονται βελτιώσεις που μπορούν να εξελίσσουν την εφαρμογή περισσότερο στο μέλλον.

Τέλος, παρατίθενται οι πηγές, απ' όπου αντλήθηκε χρήσιμο υλικό για την παρούσα εργασία.



## Κεφάλαιο 2ο: Οι τεχνολογίες του Εικονικού Κόσμου

### 2.1 Εισαγωγή

Οι τεχνολογίες εικονικής, επαυξημένης αλλά και μικτής πραγματικότητας είναι πλέον πάρα πολύ διαδεδομένες και ανεπτυγμένες στις μέρες μας. Προσφέρουν ένα μεγάλο πλήθος υπηρεσιών, όπως για παράδειγμα πληροφορίες, διασκέδαση, μάθηση κτλ. Μπορούν να εφαρμοστούν σχεδόν σε κάθε τομέα της καθημερινής ζωής των ανθρώπων, όπως για παράδειγμα η εκπαίδευση, το παιχνίδι, η ιατρική κτλ. Για λόγους ευκολίας, υπάρχει και ένας όρος που ονομάζεται **Εκτεταμένη πραγματικότητα (Extended Reality - XR)** ο οποίος αναφέρεται και στις τρεις προαναφερθείσες τεχνολογίες [1].

Υπόσχονται πολλά για το μέλλον, καθώς ολοένα και περισσότερο διεισδύουν στην αγορά, τις βιομηχανίες και την προσωπική ζωή των ανθρώπων, ενώ ταυτόχρονα συνδυάζονται με νέες αναπτυσσόμενες συσκευές και λογισμικά [2]. Λόγω της ταχύτατης ανάπτυξης αυτών των τεχνολογιών, προωθούνται πολλές συσκευές και πλατφόρμες που είναι ήδη παρωχημένες, ενώ την ίδια στιγμή παρουσιάζονται και νέες προκλήσεις [2]. Η εικονική, επαυξημένη και μικτή πραγματικότητα διαφέρουν μεταξύ τους, καθώς μπορούν να εφαρμοστούν και σε πολλούς διαφορετικούς μεταξύ τους τομείς.

Στο παρόν κεφάλαιο θα γίνει μια εισαγωγή για τις τεχνολογίες της εικονικής, επαυξημένης και μικτής πραγματικότητας, θα παρουσιαστούν οι κύριες διαφορές μεταξύ τους αλλά και τα οφέλη που έχουν για την καθημερινότητα των ανθρώπων.

### 2.2 Στοιχειώδεις έννοιες. VR, AR & MR

**Εικονική πραγματικότητα (Virtual Reality - VR)** είναι η προσομοίωση ενός πραγματικού ή φανταστικού περιβάλλοντος από έναν υπολογιστή [3].

**Επαυξημένη πραγματικότητα (Augmented Reality - AR)** είναι η σε πραγματικό χρόνο άμεση ή έμμεση θέαση ενός φυσικού, πραγματικού περιβάλλοντος, του οποίου τα στοιχεία επαυξάνονται από στοιχεία αναπαραγόμενα από συσκευές υπολογιστών, όπως ήχος, βίντεο, γραφικά ή δεδομένα τοποθεσίας. Ο όρος εισήχθη το 1992 από τον Τομ Κάουντελ [4].

**Μικτή πραγματικότητα (Mixed Reality - MR)** είναι όρος που χρησιμοποιείται για να περιγράψει τη συγχώνευση ενός πραγματικού περιβάλλοντος και ενός περιβάλλοντος που δημιουργείται από υπολογιστή. Φυσικά και εικονικά αντικείμενα μπορούν να συνυπάρχουν σε περιβάλλοντα μικτής πραγματικότητας και να αλληλεπιδρούν σε πραγματικό χρόνο [5].

### 2.3 Παραδείγματα

#### Εικονική πραγματικότητα

Στην παρακάτω εικόνα ο άνθρωπος, που είναι καθιστός, φοράει ειδικά γυαλιά. Αυτά τα γυαλιά απεικονίζουν τρισδιάστατα το φανταστικό περιβάλλον που δείχνει η τηλεόραση. Έτσι δίνεται η

## Κεφάλαιο 2°

αίσθηση πως ο ίδιος βρίσκεται στο φανταστικό αυτό περιβάλλον και αλληλεπιδρά μαζί του σε πραγματικό χρόνο. Το παράδειγμα φαίνεται στο Σχήμα 2.1.



Σχήμα 2.1: Παράδειγμα εικονικής πραγματικότητας [6]

### Επαυξημένη πραγματικότητα

Στο Σχήμα 2.2 φαίνεται ένα παράδειγμα, όπου ο χρήστης επαυξάνει τον πραγματικό κόσμο στη συσκευή του με φανταστικά αντικείμενα.



Σχήμα 2.2: Παράδειγμα της επαυξημένης πραγματικότητας [7]

### Μικτή πραγματικότητα

Στο Σχήμα 2.3 ο άνθρωπος κρατάει ένα αντικείμενο στα χέρια του, ενώ ταυτόχρονα φορώντας τα ειδικά γυαλιά βλέπει και το φανταστικό τρισδιάστατο μωβ αντικείμενο που εμφανίζεται. Τα βλέπει

και τα δύο ταυτόχρονα στον ίδιο χώρο, ενώ και τα δύο αυτά αντικείμενα μπορούν να αλληλεπιδρούν μεταξύ τους σε πραγματικό χρόνο.



Σχήμα 2.3: Παράδειγμα μικτής πραγματικότητας [8]

## 2.4 Διαφορές μεταξύ VR, AR και MR

Η εικονική πραγματικότητα είναι σαν να βάζει τον άνθρωπο σε ένα φανταστικό περιβάλλον και δίνει τη δυνατότητα αλληλεπίδρασης με αυτό. Έτσι δίνει την αίσθηση πως ο άνθρωπος έχει μπει στο φανταστικό περιβάλλον. Για παράδειγμα, θα μπορούσε κάποιος να «μπει» σε ένα παιχνίδι με δράκους, τους οποίους μπορεί να εξημερώσει. Αυτό δίνει την αίσθηση σαν να βρίσκεται ο ίδιος μέσα στο παιχνίδι και αλληλεπιδρά με τους δράκους.

Η επαυξημένη πραγματικότητα, όπως προαναφέρθηκε, επαυξάνει τον πραγματικό κόσμο με εικονικά αντικείμενα. Εδώ συμβαίνει το αντίστροφο με την εικονική πραγματικότητα. Δηλαδή δεν «μπαίνει» ο άνθρωπος σε ένα φανταστικό περιβάλλον, αλλά αντιθέτως, διάφορα φανταστικά αντικείμενα «εισάγονται» στον πραγματικό κόσμο. Για παράδειγμα, αν κάποιος επιθυμούσε να δει πώς θα ήταν να υπάρχει ένας καναπές (ή κάποιο άλλο αντικείμενο) μέσα σε ένα πραγματικό δωμάτιο, μπορεί με τη χρήση AR να εμφανίσει αυτόν τον καναπέ. Ταυτόχρονα, έχοντας ανοιχτή την κάμερά του, η οποία απεικονίζει το δωμάτιο, μπορεί να δει αν θα ταίριαζε ο καναπές αυτός στο δωμάτιό του.

Τέλος, η μικτή πραγματικότητα είναι ο συνδυασμός των δύο παραπάνω. Συγκεκριμένα, περιγράφει περιβάλλοντα στα οποία υπάρχουν πραγματικά και εικονικά στοιχεία, τα οποία μπορούν να αλληλεπιδρούν σε πραγματικό χρόνο και στα οποία ο χρήστης έχει τη δυνατότητα να αλληλεπιδράσει τόσο με πραγματικά όσο και με εικονικά στοιχεία.

## 2.5 Πεδία εφαρμογής

Τα πεδία εφαρμογής της εκτεταμένης πραγματικότητας είναι πολλά. Ορισμένοι πολύ βασικοί τομείς είναι οι εξής:

### **2.5.1 Αρχιτεκτονική και πολεοδομικές εφαρμογές**

Δίνεται η δυνατότητα να οπτικοποιηθεί το αρχιτεκτονικό αποτέλεσμα πριν την πραγματική του κατασκευή, καθώς και να προληφθούν ατέλειες και δυνητικές απειλές για μια κατασκευή. Έτσι, βελτιώνει τη λήψη αποφάσεων και προσδίδεται μεγάλο κέρδος σε χρόνο [9].

### **2.5.2 Στρατιωτικές εφαρμογές**

Δίνεται η δυνατότητα εκπαίδευσης των στρατιωτικών όλων των σωμάτων (αεροπορία, ναυτικό, στρατός) με εξομοιώσεις, που επιτρέπουν στους στρατιωτικούς να δοκιμάζουν διάφορες τεχνικές πολέμου ή επιβίωσης, χωρίς όμως να υπάρχει το ρίσκο της πραγματικής παρουσίας εχθρού ή άλλων επικίνδυνων καταστάσεων. Αυτό προετοιμάζει πολύ καλά τα στρατιωτικά σώματα για την περίπτωση μια μελλοντικής επικίνδυνης κατάστασης [9].

### **2.5.3 Ιατρική**

Χρησιμοποιείται κυρίως από χειρουργούς και κλινικούς γιατρούς. Μπορεί να παρουσιάσει διάφορα εικονικά αντικείμενα και ιατρικές διαδικασίες όσο πιο ταυτόσημα γίνεται με την πραγματικότητα διαδικασίες, ώστε να αποκτηθεί μεγαλύτερη εμπειρία από τους γιατρούς [9].

### **2.5.4 Βιομηχανία και κατασκευές**

Υπάρχουν εφαρμογές που προσομοιώνουν τη λειτουργία διάφορων μηχανών και διαδικασιών παραγωγής, όπως για παράδειγμα είναι η προσομοίωση χειρισμού ενός εκσκαφέα ή η ναυπήγηση πλοίων και η συντήρηση δικτύων και εγκαταστάσεων [9].

### **2.5.5 Πολιτισμός και Ιστορία**

Ένα πολύ σημαντικό παράδειγμα εδώ είναι η προσομοίωση ενός τρισδιάστατου μουσείου, που προσφέρει –μεταξύ και άλλων υπηρεσιών– το Ίδρυμα Μείζονος Ελληνισμού [9].

### **2.5.6 Εκπαίδευση και επαγγελματική κατάρτιση**

Έχουν αναπτυχθεί εφαρμογές οι οποίες υποστηρίζουν πολλά μαθήματα της σχολικής εκπαίδευσης, όπως για παράδειγμα το μάθημα της Φυσικής, της Χημείας, των Μαθηματικών, της Βιολογίας, της Ιστορίας κτλ. Όσον αφορά την επαγγελματική κατάρτιση, ένα παράδειγμα είναι η χρήση προσομοίωσης για τους αστροναύτες σε αποστολές, όπου είναι αναγκαίο εκ των προτέρων να εξοικειωθούν με τη χρήση τεχνολογιών και ποικίλων καταστάσεων [9].

## **2.6 Επίλογος**

Με βάση όσα αναλύθηκαν προηγουμένως μπορεί να γίνει αντιληπτό πως οι τεχνολογίες της XR έχουν πολλαπλά οφέλη σε πάρα πολλούς τομείς της καθημερινότητας. Όσον αφορά στην εκπαίδευση η αξιοποίηση των τεχνολογιών αυτών στη διδασκαλία μπορεί να έχει θετικά αποτελέσματα, καθώς οι

μαθητές αφοσιώνονται πιο εύκολα στις γνώσεις, εφόσον η διδασκαλία γίνεται με έναν πιο ευχάριστο και ξεκούραστο τρόπο.



## **Κεφάλαιο 3ο: Το περιβάλλον ανάπτυξης του παιχνιδιού**

### **3.1 Εισαγωγή**

Στο κεφάλαιο αυτό θα παρουσιαστεί το περιβάλλον ανάπτυξης Unity καθώς και ορισμένα στοιχεία και μηχανισμοί αυτού, που αποτελούν βασικά θεμέλια της εφαρμογής και ουσιαστικά είναι τα «μέσα», τα οποία προσδίδουν αλληλεπίδραση μεταξύ του χρήστη και της εφαρμογής. Πρόκειται για μηχανισμούς, οι οποίοι είναι υπεύθυνοι για την εμφάνιση διάφορων στοιχείων στην οθόνη, όπως κουμπιά, εικόνες, κείμενο κτλ., καθώς και μηχανισμούς με τους οποίους μπορεί ο χρήστης να αλληλεπιδρά με διάφορα αντικείμενα του κόσμου του παιχνιδιού.

### **3.2 Η έννοια «Μηχανή Παιχνιδιών» (Game Engine)**

Ως «μηχανή παιχνιδιών» ορίζεται ένα πλαίσιο λογισμικού, το οποίο είναι σχεδιασμένο με κύριο στόχο την ανάπτυξη βιντεοπαιχνιδιών. Περιλαμβάνει διάφορα προγράμματα ανάπτυξης αλλά και βιβλιοθήκες σχετιζόμενες με αυτά. Ο όρος «μηχανή παιχνιδιών» μπορεί επιπλέον να χρησιμοποιείται για να αναφερθεί στο λογισμικό ανάπτυξης που υποστηρίζει αυτό το πλαίσιο, το οποίο στις περισσότερες περιπτώσεις είναι ένα σετ από διάφορα προγραμματιστικά εργαλεία και λειτουργίες που στοχεύουν στην ανάπτυξη παιχνιδιών [10].

### **3.3 Το πρόγραμμα Unity**

Το Unity είναι μια μηχανή παιχνιδιών συμβατή για όλες τις πλατφόρμες. Αναπτύχθηκε από την Unity Technologies, η οποία έκανε την εμφάνισή της για πρώτη φορά τον Ιούνιο του 2005 στο Apple Worldwide Developers Conference ως μηχανή παιχνιδιών για το λογισμικό Mac OS X. Στη συνέχεια, όμως, εξελίχθηκε σε σημείο που μπορεί να υποστηρίζει πολλές πλατφόρμες, όπως κινητά, υπολογιστές, τάμπλετ, κονσόλες και προσφέρει δυνατότητα ανάπτυξης εφαρμογών εικονικής και επανξιημένης πραγματικότητας [11].

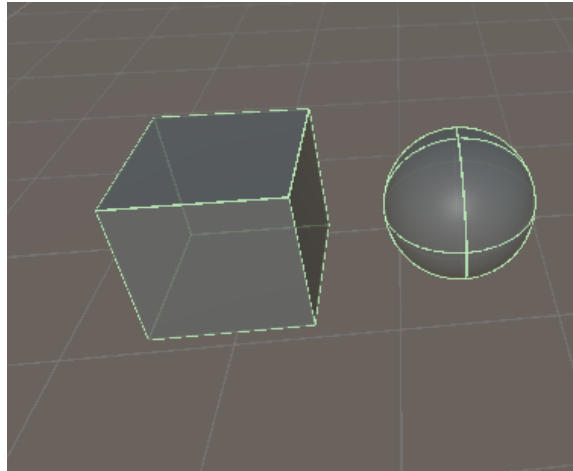
Το Unity προσφέρει δυνατότητα ανάπτυξης τρισδιάστατων (3-Dimensional, 3D), δισδιάστατων (2-Dimensional, 2D) παιχνιδιών αλλά και ποικίλων διαδραστικών προσομοιώσεων. Εκτός από τις βιομηχανίες παιχνιδιών το Unity έχει χρησιμοποιηθεί και σε πολλούς άλλους τομείς, όπως είναι ο κινηματογράφος, η αυτοκινητοβιομηχανία, η αρχιτεκτονική, η μηχανική, οι κατασκευές και οι Ένοπλες Δυνάμεις των Ηνωμένων Πολιτειών.

### **3.4 Βασικά στοιχεία και μηχανισμοί του Unity**

#### **3.4.1 Collider component**

Το Collider component είναι ένας μηχανισμός, ο οποίος δίνει υπόσταση στα αντικείμενα του κόσμου του παιχνιδιού. Για παράδειγμα, αν υπάρχουν δύο κουτιά σε ένα παιχνίδι ή δύο σφαίρες οι οποίες συγκρούονται μεταξύ τους, προκειμένου να συμβεί αλλά και να ανιχνευθεί κάποια σύγκρουση, αυτά τα κουτιά ή οι σφαίρες πρέπει να έχουν κάποια υπόσταση. Αυτήν την υπόσταση στα αντικείμενα ενός παιχνιδιού την προσδίδουν τα Colliders, που υπάρχουν σε διάφορα σχήματα, όπως Box Collider ή Sphere Collider (Σχήμα 3.1) κτλ. Επιπλέον, τα Colliders βοηθούν και στην αλληλεπίδραση του

χρήστη με τα αντικείμενα του παιχνιδιού, καθώς ανιχνεύονται και από τους μηχανισμούς Raycast, μηχανισμός που παρουσιάζεται στη συνέχεια στο παρόν κεφάλαιο. Στο παρακάτω Σχήμα 3.1 το Collider component είναι το πράσινο περίγραμμα γύρω από κάθε αντικείμενο.



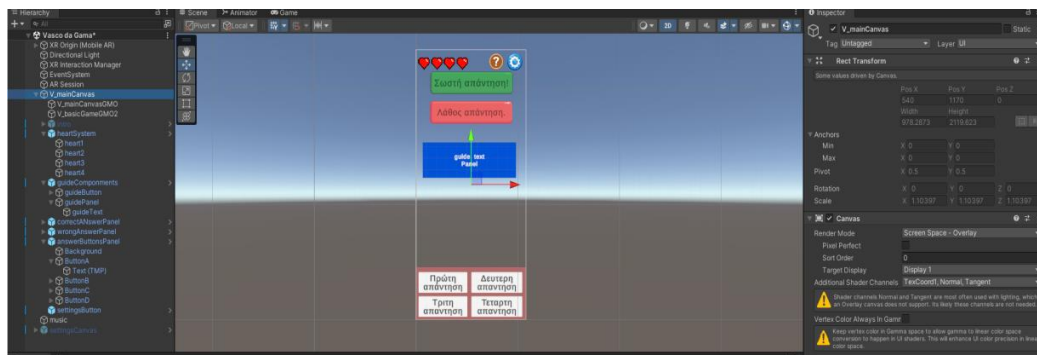
Σχήμα 3.1: Το Collider component

### 3.4.2 Raycast

Ο μηχανισμός Raycast χρησιμοποιείται για να προσφέρει αλληλεπίδραση μεταξύ διάφορων αντικειμένων μεταξύ τους ή για την αλληλεπίδραση του χρήστη με αντικείμενα του παιχνιδιού. Ουσιαστικά δημιουργεί μια νοητή γραμμή-ακτίνα (από ένα σημείο αρχής και με κατεύθυνση ένα δεύτερο σημείο), η οποία μπορεί να εκτείνεται ατελείωτα στον κόσμο του παιχνιδιού. Είναι πολύ χρήσιμος μηχανισμός σε παιχνίδια drag and drop, καθώς μπορεί να εντοπίζει διάφορα Colliders [12].

### 3.4.3 Canvas

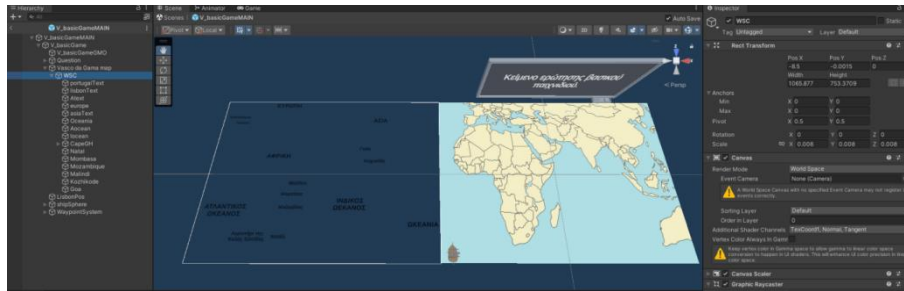
Ένα αντικείμενο Canvas αποτελεί τον χώρο, όπου υπάρχουν όλα τα αντικείμενα γραφικής διεπαφής, όπως κουμπιά, εικόνες, κείμενο κτλ., με τα οποία έχει τη δυνατότητα ο χρήστης να αλληλεπιδρά με μία εφαρμογή [13]. Στην ιεραρχία του Unity όλα τα αντικείμενα της γραφικής διεπαφής πρέπει να είναι «παιδιά» του αντικειμένου Canvas, προκειμένου να εμφανιστούν στην οθόνη. Στο Σχήμα 3.2 φαίνεται ένα παράδειγμα με διάφορα αντικείμενα γραφικής διεπαφής στην οθόνη [13].



Σχήμα 3.2: Παράδειγμα του αντικειμένου Canvas

Το συγκεκριμένο αντικείμενο Canvas, που απεικονίζεται στο Σχήμα 3.2, ανήκει στην κατηγορία Screen Space – Overlay και αυτό φαίνεται δεξιά, στην καρτέλα Inspector του Unity, όπου στο πεδίο Render Mode του Canvas component έχει επιλεγθεί το Screen Space – Overlay. Αυτό σημαίνει πως τα αντικείμενα της γραφικής διεπαφής βρίσκονται πάνω στην οθόνη και όχι στον κόσμο του παιχνιδιού, τον οποίο υπερκαλύπτουν.

Στο Render Mode υπάρχει και η επιλογή World Space Canvas, στην οποία τα αντικείμενα γραφικής διεπαφής δεν βρίσκονται πάνω στην οθόνη αλλά μέσα στον κόσμο του παιχνιδιού. Παράδειγμα για το World Space Canvas φαίνεται στο Σχήμα 3.3.



Σχήμα 3.3: Παράδειγμα World Space Canvas

Στο Σχήμα 3.3 απεικονίζεται ένα αντικείμενο Canvas κατηγορίας World Space με διάφορα ονόματα και ένας χάρτης στα δεξιά. Πρόκειται για τον χάρτη του παιχνιδιού του Βάσκο ντα Γκάμα της συγκεκριμένης εφαρμογής, που θα παρουσιαστεί στο επόμενο κεφάλαιο. Το World Space Canvas τοποθετείται πάνω στον χάρτη και έτσι δημιουργείται το τελικό αποτέλεσμα, που είναι ένας ρεαλιστικός χάρτης με ονόματα πάνω του.

### 3.4.4 Transform

Το Transform component αποτελεί τη θέση, την περιστροφή και το μέγεθος ενός αντικειμένου. Κάθε αντικείμενο έχει ένα Transform component και χρησιμοποιείται σε περίπτωση που πρέπει να αλλάξει το μέγεθος, η περιστροφή ή η θέση ενός αντικειμένου [14].

### 3.4.5 Inspector

Το Inspector είναι ένα παράθυρο, στο οποίο φαίνονται οι ιδιότητες ενός επιλεγμένου αντικειμένου, αρχείου ή component και μέσω του συγκεκριμένου παραθύρου μπορεί να γίνει η επεξεργασία των ιδιοτήτων αυτών [15]. Για παράδειγμα στο Σχήμα 3.3 φαίνεται η καρτέλα Inspector, η οποία απεικονίζει όλες τις ιδιότητες και τα components που έχει πάνω του το αντικείμενο Canvas.

### **3.4.6 Animations**

Ένα σύστημα κινούμενης εικόνας (Animation) παρέχει διάφορους μηχανισμούς και διαδικασίες για να προσδίδει κίνηση σε ιδιότητες, μοντέλα και στοιχεία [16]. Για παράδειγμα, το μοντέλο ενός ανθρώπου ή ζώου που μπορεί να τρέξει, να περπατήσει και γενικώς να συμπεριφερθεί με ρεαλιστικές κινήσεις μέσα στο Unity, μπορεί να χρησιμοποιεί ένα σύστημα κινούμενων εικόνων για να το πετύχει αυτό.

### **3.4.7 Particle System**

Ένα σύστημα σωματιδίων (Particle System) είναι μηχανισμός, ο οποίος μπορεί και προσομοιώνει οντότητες που είναι ρευστές, όπως υγρά, σύννεφα, φλόγες κλπ. Αυτό το κάνει με το να δημιουργεί και να δίνει «ζωή» σε μία πληθώρα από μικρές 2D εικόνες [17]. Για παράδειγμα, στη συγκεκριμένη εφαρμογή που παρουσιάζεται στο κεφάλαιο 4, στο παιχνίδι του Βάσκο ντα Γκάμα και του Μάρκο Πόλο, τα εφέ καπνού και εκρήξεων έχουν δημιουργηθεί με τον μηχανισμό Particle System του Unity.

## **3.5 Επίλογος**

Γενικά, υπάρχουν πάρα πολλοί μηχανισμοί και εργαλεία – στοιχεία στο περιβάλλον ανάπτυξης Unity, τα οποία είναι χρήσιμα σε όλους τους τομείς, είτε πρόκειται για ένα 2D ή 3D είτε για παιχνίδι επαγγελματικής πραγματικότητας. Αυτά τα εργαλεία καθιστούν την ανάπτυξη μίας εφαρμογής πολύ ευκολότερη. Στο παρόν κεφάλαιο παρουσιάστηκαν με σύντομο τρόπο το πρόγραμμα ανάπτυξης Unity καθώς και ορισμένα βασικά στοιχεία και μηχανισμοί αυτού, που χρησιμοποιήθηκαν στην εφαρμογή AR για το μάθημα της Ιστορίας ΣΤ΄ Δημοτικού.

## **Κεφάλαιο 4ο: Η εφαρμογή AR στο μάθημα της Ιστορίας**

### **4.1 Εισαγωγή**

Το παρόν κεφάλαιο επικεντρώνεται στη συγκεκριμένη εφαρμογή. Πιο στοχευμένα, θα παρουσιαστεί σύντομα τι είναι αυτή η εφαρμογή, με τι σχετίζεται και πώς μπορεί ο χρήστης-παίκτης να αποκομίσει κάποιο όφελος μέσω αυτής.

### **4.2 Σκοπός σχεδίασης της εφαρμογής**

Η εφαρμογή έχει σχεδιαστεί για εκπαιδευτικούς σκοπούς στα σχολεία και συγκεκριμένα, αφορά στο μάθημα της Ιστορίας της ΣΤ΄ Δημοτικού. Στόχος της είναι να περιγράψει στους μαθητές τη ζωή των πέντε θαλασσοπόρων-εξερευνητών, Φερδινάνδου Μαγγελάνου, Βαρθολομαίου Ντιάζ, Χριστόφορου Κολόμβου, Μάρκο Πόλο και Βάσκο ντα Γκάμα. Παράλληλα στοχεύει στο να κάνει τη μάθηση όσο πιο ελκυστική και ενδιαφέρουσα γίνεται, ώστε να μην είναι μονότονη και κουραστική για τους μαθητές. Για αυτόν τον λόγο ουσιαστικά πρόκειται για παιχνίδι, στο οποίο ο μαθητής-παίκτης έχει και συγκεκριμένο περιθώριο λαθών.

Η εφαρμογή απεικονίζει με ζωντανό τρόπο, με τη μορφή παιχνιδιού, τα ταξίδια των θαλασσοπόρων και περιγράφει ποικίλα γεγονότα, που έλαβαν μέρος σε διάφορους σταθμούς κατά τη διάρκεια των ταξιδιών.

Πρόκειται για μια εφαρμογή επαυξημένης πραγματικότητας, η οποία επαυξάνει τον πραγματικό χώρο με έναν χάρτη πάνω στον οποίο διαδραματίζεται η ζωή του κάθε εξερευνητή. Το περιβάλλον ανάπτυξης, το οποίο χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής, είναι το Unity.

Με το ξεκίνημα της εφαρμογής εμφανίζεται το αρχικό μενού, το οποίο έχει τις δύο επιλογές «Παίξτε» και «Εξοδος» (Σχήμα 4.1).



Σχήμα 4.1: Το αρχικό μενού

Με το πάτημα του κουμπιού «Παίξτε» εμφανίζεται το Play Scene (Σχήμα 4.2), το οποίο περιέχει τα κουμπιά με τα οποία ο παίκτης μπορεί να επιλέξει τον θαλασσοπόρο, του οποίου το παιχνίδι επιθυμεί να παίξει. Υπάρχει ένα αρχείο εντολών (Script), το Scene Loaders, το οποίο περιέχει τις εντολές μετάβασης σε κάθε σκηνή της εφαρμογής.



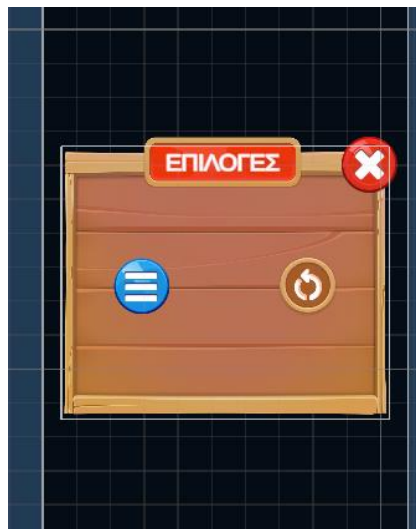
Σχήμα 4.2: Το Play Scene

Υπάρχουν πέντε παιχνίδια, ένα για κάθε θαλασσοπόρο. Τα παιχνίδια, όπως έχει προαναφερθεί, είναι παρόμοια μεταξύ τους στο βασικό μέρος. Το βασικό μέρος είναι ένας χάρτης που εμφανίζεται, πάνω στον οποίο λαμβάνει χώρα το ταξίδι κάθε θαλασσοπόρου, καθώς και διάφορα γεγονότα που συμβαίνουν κατά την πορεία του. Ο σχεδιασμός του ταξιδιού γίνεται απαντώντας κάθε φορά την ερώτηση που εμφανίζεται σε ένα γκρι-πλαίσιο. Τα κουμπιά των απαντήσεων στο βασικό παιχνίδι εμφανίζονται στο κάτω μέρος της οθόνης και είναι κάθε φορά τέσσερα, εκ των οποίων μόνο ένα αντιστοιχεί στη σωστή απάντηση.

Αυτό που διαφέρει κυρίως είναι τα μίνι παιχνίδια, τα οποία απεικονίζουν διάφορα γεγονότα. Κάποιοι θαλασσοπόροι συμμετέχουν σε μάχες, άλλοι ανακαλύπτουν πολύτιμα υλικά, άλλοι τοποθετούν συγκεκριμένα σύμβολα ως ένδειξη κυριαρχίας και πολλά άλλα. Όλα αυτά εμφανίζονται με τη μορφή μίνι παιχνιδιών/πληροφοριών σε διάφορους σταθμούς των ταξιδιών.

Στο πάνω αριστερά μέρος της οθόνης (αφού έχει ξεκινήσει το παιχνίδι κάποιου θαλασσοπόρου) εμφανίζονται τέσσερις ζωές. Αυτό σημαίνει πως ο παίκτης έχει δυνατότητα να κάνει μέχρι τρία λάθη και στο τέταρτο λάθος, αφού εξαφανιστεί και η τέταρτη ζωή, χάνει. Όμως, τα λάθη αφορούν μόνο το βασικό μέρος του παιχνιδιού και όχι το μέρος των μίνι παιχνιδιών. Αυτό σημαίνει πως, εάν ο παίκτης κάνει κάποιο λάθος σε ερώτηση που αφορά κάποιον επόμενο προορισμό για τον σχεδιασμό του ταξιδιού κάποιου θαλασσοπόρου πάνω στον χάρτη (π.χ. Ποιος ήταν ο πρώτος προορισμός του Χ θαλασσοπόρου), τότε χάνει μια ζωή, ενώ αν κάνει κάποιο λάθος σε μίνι ερώτηση Quiz (π.χ. Τι τοποθέτησε σε αυτό το σημείο ο Χ Θαλασσοπόρος), δηλαδή σε ερώτηση που δεν αφορά τον σχεδιασμό ταξιδιού, τότε δεν χάνει ζωή. Στα μίνι παιχνίδια υπάρχει η δυνατότητα απεριόριστων λαθών.

Στο πάνω δεξιά μέρος της οθόνης εμφανίζονται δύο κουμπιά. Το ένα είναι ένα αγγλικό ερωτηματικό, το οποίο, αν ο παίκτης πατήσει, εμφανίζεται η οδηγία για το τι πρέπει να κάνει σε περίπτωση που δυσκολευτεί σε κάποιο βήμα. Το δεύτερο κουμπί είναι ένα γρανάζι, το οποίο είναι το κουμπί που διαχειρίζεται το παράθυρο διάφορων επιλογών (Σχήμα 4.3). Το παράθυρο επιλογών έχει τρία κουμπιά. Το πρώτο (στα αριστερά) είναι το κουμπί που επιστρέφει τον παίκτη στο αρχικό μενού, το δεύτερο (στη μέση) είναι το κουμπί, το οποίο διακόπτει ή συνεχίζει τη μουσική και το τρίτο (στα δεξιά) είναι το κουμπί επανεκκίνησης του παιχνιδιού του θαλασσοπόρου που έχει επιλεγεί.



Σχήμα 4.3: Το παράθυρο επιλογών

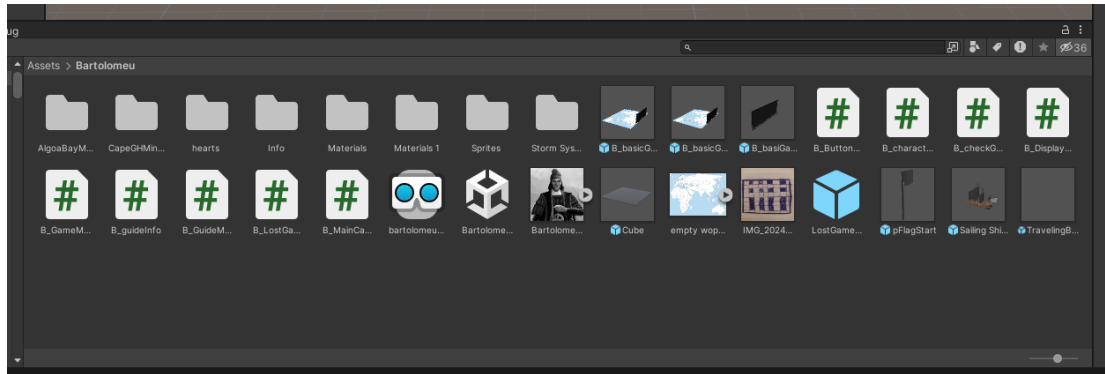
Όλα τα παραπάνω χαρακτηριστικά (βασικό μέρος παιχνιδιού, μίνι παιχνίδια, σύστημα λαθών κτλ) αναφέρονται εξ αρχής, γιατί είναι κοινά σε όλα τα παιχνίδια, τα οποία θα περιγραφούν αναλυτικά με την παρουσίασή τους.

## 4.3 Τα παιχνίδια

### 4.3.1 Το παιχνίδι του Βαρθολομαίου Ντιάζ

#### 4.3.1.1 Βασικά στοιχεία του παιχνιδιού

Αρχικά, στο Σχήμα 4.4 φαίνονται όλα τα αρχεία-assets που χρησιμοποιήθηκαν για το παιχνίδι του Βαρθολομαίου Ντιάζ.

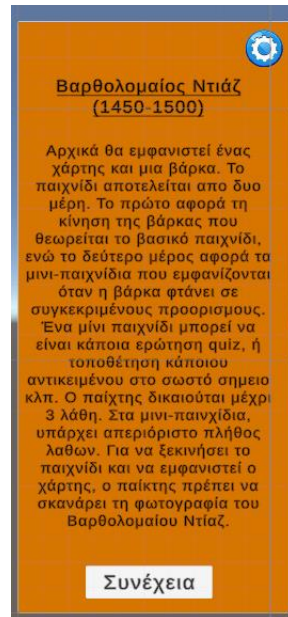


Σχήμα 4.4: Τα αρχεία του παιχνιδιού του Βαρθολομαίου Ντιάζ

Με το ξεκίνημα του παιχνιδιού εμφανίζεται ένα πλαίσιο με εισαγωγικά στοιχεία για τον θαλασσοπόρο και οδηγίες για τον τρόπο με τον οποίο παίζεται το παιχνίδι (Σχήματα 4.5 και 4.6).

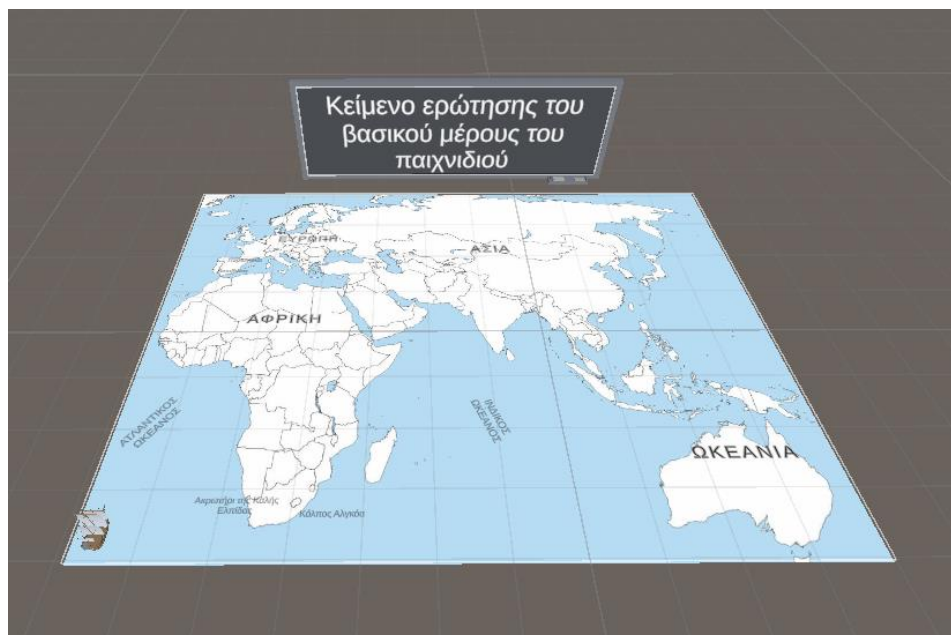


Σχήμα 4.5: Εισαγωγικό πλαίσιο με το ξεκίνημα του παιχνιδιού του Βαρθολομαίου Ντιάζ



Σχήμα 4.6: Πλαίσιο οδηγιών για το πως παίζεται το παιχνίδι

Κλείνοντας το εισαγωγικό πλαίσιο ο παίκτης πρέπει να σκανάρει με την κάμερα της συσκευής του το πορτραίτο του Βαρθολομαίου Ντιάζ, για να εμφανιστεί ο χάρτης που αποτελεί το βασικό μέρος του παιχνιδιού και να εμφανιστούν οι ζωές, τα κουμπιά απαντήσεων στις ερωτήσεις του βασικού μέρους και τα κουμπιά οδηγιών και επιλογών (Σχήματα 4.7 και 4.8).



Σχήμα 4.7: Ο χάρτης που αποτελεί το βασικό μέρος του παιχνιδιού του Βαρθολομαίου Ντιάζ



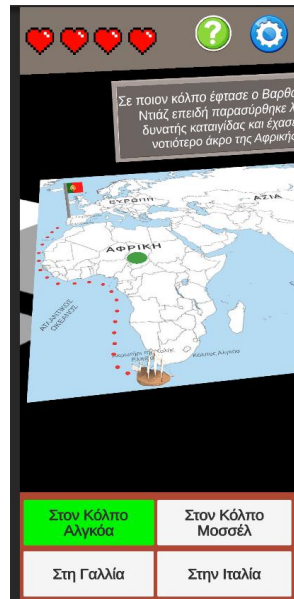
Σχήμα 4.8: Τα κουμπιά του παιχνιδιού και οι ζωές

Με την απάντηση σε κάθε ερώτηση του βασικού μέρους του παιχνιδιού, εμφανίζεται αρχικά το μήνυμα για το αν ο παίκτης έχει απαντήσει σωστά ή λανθασμένα στην ερώτηση και στη συνέχεια ορισμένες βασικές πληροφορίες που αφορούν στον θαλασσοπόρο (Σχήμα 4.9).



Σχήμα 4.9: Σωστή απάντηση της πρώτης ερώτησης του βασικού παιχνιδιού

Μετά από τρεις εισαγωγικές ερωτήσεις (από πού καταγόταν, ποια ήπειρο ανακάλυπτε και ποια ήταν η αφετηρία του ταξιδιού του Βαρθολομαίου Ντιάς) στην τέταρτη ερώτηση ξεκινά το ταξίδι και με την απάντησή της ξεκινά η κίνηση του πλοίου (Σχήμα 4.10).



Σχήμα 4.10: Η κίνηση του πλοίου με το ξεκίνημα του ταξιδιού

Το πλοίο –σε κάθε παιχνίδι, όχι μόνο στο παιχνίδι του Βαρθολομαίου Ντιάζ– αφήνει πίσω του μια διακεκομμένη κόκκινη γραμμή, η οποία χαράζει την πορεία του θαλασσοπόρου.

#### 4.3.1.2 Η λειτουργία των βασικών κουμπιών

Στη συνέχεια θα περιγραφεί ο τρόπος, με τον οποίο λειτουργεί η παραπάνω διαδικασία, μέσα από την παρουσίαση ορισμένων βασικών σημείων του κώδικα προγραμματισμού της εφαρμογής.

Αρχικά υπάρχει ένα αρχείο εντολών - Script με όνομα B\_GameManagementScript (το αρχικό B από τον Βαρθολομαίο). Αυτό είναι που δημιουργεί τις ερωτήσεις. Σε αυτό το αρχείο υπάρχει μια μεταβλητή τύπου int (ακέραιος αριθμός) questionCounter, η οποία είναι ο μετρητής ερωτήσεων. Στη μέθοδο createNewQuestion() –η οποία καλείται και στο ξεκίνημα του παιχνιδιού, για να δημιουργηθεί η πρώτη ερώτηση– δημιουργείται κάθε ερώτηση ανάλογα με τον μετρητή questionCounter. Ο μετρητής αυτός αλλάζει τιμές σε διαφορετικό αρχείο εντολών, το οποίο θα παρουσιαστεί στη συνέχεια. Η μεταβλητή questionText αποθηκεύει το κείμενο της ερώτησης του βασικού μέρους του παιχνιδιού · η μεταβλητή correctAnswer αποθηκεύει το κείμενο της σωστής απάντησης · και, επίσης, υπάρχει και ένας πίνακας answerTexts, ο οποίος περιέχει τις μεταβλητές που αποθηκεύουν τα κείμενα των τεσσάρων δυνατών απαντήσεων (Σχήματα 4.11 και 4.12).

```

public class B_GameManagementScript : MonoBehaviour
{
    // Start is called before the first frame update
    public string questionText;
    public string correctAnswer;
    private B_ButtonManagementScript b_bms;
    [SerializeField] private GameObject B_basicGameGMO;
    [SerializeField] private TextMeshProUGUI[] answerTexts;
    public int questionCounter;
    void Start()
    {
        //arxikopoihsh shmantikwn metablhtwn
        InitializeVariables();
        createNewQuestion();
    }
    private void InitializeVariables()
    {
        correctAnswer = null;
        b_bms = GameObject.Find("B_basicGameGMO2").GetComponent<B_ButtonManagementScript>();
        questionCounter = 1;
    }
}

```

Σχήμα 4.11: Οι βασικές μεταβλητές και η αρχικοποίηση του παιχνιδιού

```

public void createNewQuestion()
{
    switch (questionCounter)
    {
        case 1:
            questionText = "Από ποια χώρα ήταν καταγόντα ο Βαρθολομαίος Ντιάζ; Τοποθέτησε τη σημαία στη σωστή χώρα"; //θα επιλεγεται η Πορτογαλία
            correctAnswer = "Portugal";
            answerTexts[0].text = "Πορτογαλία"; //κουμπι Α
            answerTexts[1].text = "Αγγλία";
            answerTexts[2].text = "Γαλλία";
            answerTexts[3].text = "Ιταλία";
            break;
        case 2:
            questionText = "Σκόπος του ταξιδιού του Βαρθολομαίου Ντιάζ, ήταν να βρει το νοτιότερο άκρο μιας ηπείρου. Ποια ήπειρος ήταν αυτή;"; //
            correctAnswer = "Africa";
            answerTexts[0].text = "Ευρώπη";
            answerTexts[1].text = "Οκεανία";
            answerTexts[2].text = "Αφρική"; //κουμπι C
            answerTexts[3].text = "Ασία";
            break;
    }
}

```

Σχήμα 4.12: Τμήμα της μεθόδου createNewQuestion()

Στο σημείο αυτό κρίνεται αναγκαίο να παρουσιαστούν οι λειτουργίες της δημιουργίας μιας νέας ερώτησης και η διαχείριση των σωστών/λανθασμένων απαντήσεων. Για τη διαχείριση των σωστών και λανθασμένων απαντήσεων υπάρχει το αρχείο εντολών `B_ButtonManagementScript`. Υπάρχουν τέσσερις μέθοδοι διαχείρισης –μία για κάθε κουμπί–, οι οποίες λειτουργούν με τον ίδιο τρόπο αλλά η καθεμία διαχειρίζεται διαφορετικό κουμπί. Στο Σχήμα 4.13 φαίνεται η μέθοδος διαχείρισης του κουμπιού C.

```

public void ButtonCManagement()
{
    switch (b_gms.questionCounter)
    {
        case 2:
            if (b_gms.correctAnswer == "Africa")
            {
                correctAnswer(buttons[2]);
            }
            break;
        case 4:
            if (b_gms.correctAnswer == "Aocean")
            {
                correctAnswer(buttons[2]);
                currentDestination++;
            }
            break;
        default:
            wrongAnswer(buttons[2]);
            break;
    }
}

```

Σχήμα 4.13: Μέθοδος διαχείρισης του κουμπιού C

Αρχικά μέσα σε έναν έλεγχο switch-case εξετάζεται η τιμή της μεταβλητής questionCounter και σε έναν επιπλέον έλεγχο η τιμή της μεταβλητής correctAnswer. Ουσιαστικά, στην παραπάνω μέθοδο το κουμπί C αποτελεί τη σωστή απάντηση σε δύο περιπτώσεις. Αυτές οι δύο περιπτώσεις είναι διαδοχικά οι εξής · η πρώτη, στην οποία το questionCounter ισούται με 2, ενώ ταυτόχρονα το correctAnswer έχει την τιμή «Africa» · και η δεύτερη, στην οποία το questionCounter ισούται με 4 και ταυτόχρονα το correctAnswer έχει την τιμή «Aocean» (Atlantic Ocean). Σε αυτές τις δύο περιπτώσεις για να απαντήσει ο παίκτης σωστά, πρέπει να πατήσει το τρίτο κουμπί από τα τέσσερα που εμφανίζονται στην οθόνη. Έτσι, καλείται η μέθοδος διαχείρισης σωστής απάντησης correctAnswer(), ενώ σε κάθε άλλη περίπτωση που σημαίνει πως το τρίτο κουμπί δεν απαντάει σωστά κάποια ερώτηση, καλείται η μέθοδος διαχείρισης λανθασμένης απάντησης. Οι δύο τελευταίες μέθοδοι φαίνονται στο Σχήμα 4.14.

```

private void correctAnswer(Button button)
{
    correctAnswers++;
    b_mcm.activateCorrectPopup();
    button.GetComponent<Image>().color = Color.green;
    disableAnswerButtons();
}

private void wrongAnswer(Button button) // η μεθοδος που καλειται, οταν απαντησει λαθω
{
    wrongAnswers++;
    heartCounter--;
    b_mcm.activateWrongPopup();
    disableAnswerButtons();
    button.GetComponent<Image>().color = Color.red;
    StartCoroutine(resetAfterWrong(button, 1f));
}

public void heartManagement()

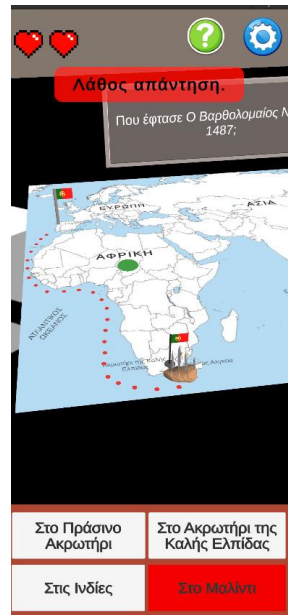
```

Σχήμα 4.14: Οι μέθοδοι σωστής και λανθασμένης απάντησης με τη σειρά

Και οι δύο μέθοδοι δέχονται ως παράμετρο ένα κουμπί, το οποίο είναι κάθε φορά το κουμπί που πάτησε ο παίκτης, για να απαντήσει σε κάποια ερώτηση.

Στην περίπτωση της σωστής απάντησης αρχικά αυξάνεται ο μετρητής σωστών απαντήσεων (correctAnswers, τύπου Int) κατά ένα. Στη συνέχεια, με την εντολή `b_mcm.activateCorrectPopup()` εμφανίζεται το μήνυμα της σωστής απάντησης σε πράσινο πλαίσιο, ενώ ταυτόχρονα γίνεται πράσινο και το ίδιο το κουμπί, όπως φαίνεται στο Σχήμα 4.9. Έπειτα, απενεργοποιούνται και τα τέσσερα κουμπιά για την αποφυγή λαθών, όσο ο παίκτης διαβάζει τις πληροφορίες που εμφανίζονται για τον θαλασσοπόρο και επανεργοποιούνται, μόλις κλείσει το παράθυρο πληροφοριών, για να μπορεί ο παίκτης να απαντήσει σε νέα ερώτηση.

Στην περίπτωση λανθασμένης απάντησης αυξάνεται ο μετρητής των λανθασμένων απαντήσεων wrongAnswers, ενώ μειώνεται ο μετρητής των ζωών του παίκτη heartCounter κατά ένα. Και οι δύο μεταβλητές είναι τύπου Int. Έπειτα εμφανίζεται το μήνυμα λανθασμένης απάντησης, χάνεται μια ζωή και ταυτόχρονα γίνεται κόκκινο το κουμπί που πάτησε ο παίκτης (Σχήμα 4.15), και τα τέσσερα κουμπιά απενεργοποιούνται για την αποφυγή πολλαπλών πατημάτων και επανεργοποιούνται μετά από ένα δευτερόλεπτο. Αν ο παίκτης χάσει και τις τέσσερις ζωές του, εμφανίζεται το παράθυρο LostGame. Είτε ο παίκτης κερδίσει είτε χάσει, του δίνονται δύο επιλογές, να επανεκκινήσει το παιχνίδι ή να μεταβεί στο αρχικό μενού, και εμφανίζεται στην οθόνη ο αριθμός των σωστών και των λανθασμένων απαντήσεων καθώς και τα αντίστοιχα μηνύματα (Σχήματα 4.16 και 4.17).



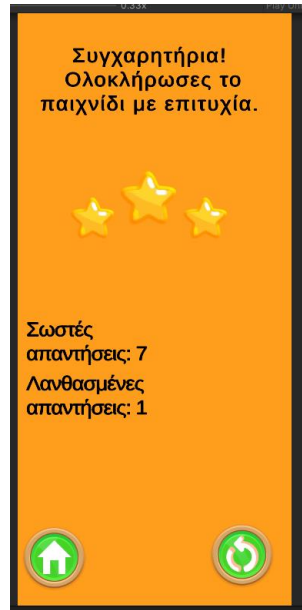
Σχήμα 4.15: Επιλογή λανθασμένης απάντησης

Μετά την επιλογή κάποιας λανθασμένης απάντησης, ο παίκτης πρέπει να περιμένει ένα δευτερόλεπτο, μέχρι να μπορεί ξανά να δώσει νέα/διαφορετική απάντηση. Αυτό γίνεται για την αποφυγή τυχαίων πολλαπλών πατημάτων, που μπορεί να συμβούν κατά λάθος.

Σε κάθε λανθασμένη απάντηση τα κουμπιά απενεργοποιούνται αμέσως και επανεργοποιούνται μετά από ένα δευτερόλεπτο. Για τον ίδιο σκοπό αυτό συμβαίνει και σε περίπτωση που ο παίκτης απαντήσει σωστά σε κάποια ερώτηση. Τα κουμπιά απενεργοποιούνται πάλι, με μόνη διαφορά πως εδώ επανεργοποιούνται, όταν έρθει η σειρά της επόμενης ερώτησης, διότι ενδιάμεσα μεσολαβούν διάφορες πληροφορίες ή μίνι παιχνίδια για κάθε θαλασσοπόρο με τη σωστή απάντηση κάθε ερώτησης. Έτσι, δεν πρέπει να ενεργοποιούνται τα κουμπιά αυτομάτως μετά από ένα δευτερόλεπτο, όταν έρθει η σειρά της επόμενης ερώτησης.



Σχήμα 4.16: Τέλος παιχνιδιού, όταν ο παίκτης χάνει στο παιχνίδι του Βαρθολομαίου Ντιάζ



Σχήμα 4.17: Το τέλος του παιχνιδιού, όταν ο παίκτης κερδίζει

Παρόμοια είναι και τα παράθυρα νίκης και ήττας σε όλα τα επόμενα παιχνίδια · το μόνο που αλλάζει είναι το χρώμα που «γεμίζει» το παράθυρο, για παράδειγμα εδώ είναι πορτοκαλί, στο παιχνίδι του Μαγγελάνου είναι μπλε, στο παιχνίδι του Βάσκο ντα Γκάμα είναι κόκκινο κλπ.

### 4.3.1.3 Η λειτουργία του παιχνιδιού

Η πρώτη ερώτηση αφορά στην καταγωγή του Βαρθολομαίου Ντιάζ. Η σωστή απάντηση είναι από την «Πορτογαλία» · όταν ο παίκτης απαντήσει σωστά, τοποθετείται η πορτογαλική σημαία πάνω στον χάρτη, στο σημείο όπου βρίσκεται η χώρα (Σχήμα 4.18).



Σχήμα 4.18: Η απάντηση της πρώτης ερώτησης

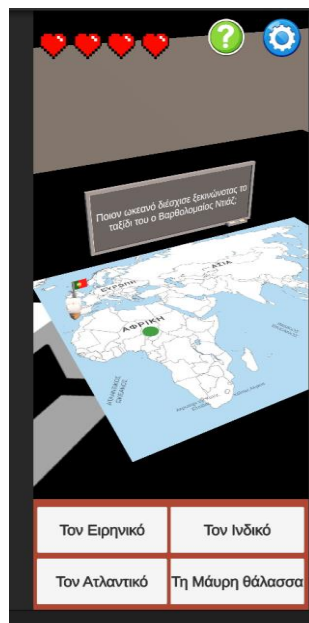
Η δεύτερη ερώτηση αφορά την ήπειρο την οποία εξερευνούσε ο Βαρθολομαίος Ντιάζ. Στο σημείο αυτό εμφανίζονται τρεις μπλε κουκκίδες-κουμπιά πάνω στον χάρτη, αντίστοιχα στην Αφρική, την Ευρώπη και την Ασία. Ο παίκτης πρέπει να πατήσει το μπλε κουμπί της σωστής ηπείρου, που στην

προκειμένη περίπτωση είναι η «Αφρική» και με την απάντηση της ερώτησης, εξαφανίζονται τα άλλα δύο κουμπιά και γίνεται πράσινο το κουμπί της Αφρικής ως ένδειξη σωστής απάντησης (Σχήμα 4.19).



Σχήμα 4.19: Απάντηση στη δεύτερη ερώτηση

Η τρίτη ερώτηση ζητά από τον παίκτη να τοποθετήσει το πλοίο στο λιμάνι, από το οποίο ξεκίνησε το ταξίδι του ο Βαρθολομαίος Ντιάζ. Η σωστή απάντηση είναι η «Λισαβόνα» και έτσι, τοποθετείται το πλοίο στη Λισαβόνα (Σχήμα 4.20). Αυτό επιτυγχάνεται με την εντολή που φαίνεται στο Σχήμα 4.21 – με παρόμοιο τρόπο τοποθετείται η πορτογαλική σημαία στην πρώτη ερώτηση.



Σχήμα 4.20: Τοποθέτηση του πλοίου στη Λισαβόνα

```

public void ButtonManagement()
{
    switch (b_gms.questionCounter)
    {
        case 3:
            if (b_gms.correctAnswer == "Lisbon")
            {
                correctAnswer(buttons[3]);
                boat.transform.position = GameObject.Find("LisbonPosition").transform.position;
                boat.transform.Rotate(0, 200, 0);
                StartCoroutine(InstantiatePrefab(infoPrefabs[2], 0f, B_mainCanvasT));
            }
    }
}

```

Σχήμα 4.21: Ο τρόπος τοποθέτησης του πλοίου στη Λισαβόνα

Η τέταρτη ερώτηση ρωτάει ποιον ωκεανό διέσχισε ο Βαρθολομαίος Ντιάζ στο ξεκίνημα του ταξιδιού του και εδώ η σωστή απάντηση είναι ο «Ατλαντικός ωκεανός». Από το σημείο αυτό ξεκινά η κίνηση του πλοίου –η λειτουργία της οποίας θα περιγραφεί κατόπιν με αναλυτικό τρόπο– αυξάνοντας μια μεταβλητή `currentDestination` (τύπου `int`) κατά ένα. Η συγκεκριμένη μεταβλητή χρησιμοποιείται, για να δηλώνεται κάθε φορά στο πρόγραμμα ποιος είναι ο τρέχων προορισμός στον οποίο πρέπει να πάει το πλοίο. Έτσι, με την απάντηση της τέταρτης ερώτησης αυξάνεται το `currentDestination` κατά ένα και από 0 που έχει ως αρχική τιμή γίνεται πλέον 1 (Σχήμα 4.22) – και αυτό σημαίνει πως το πλοίο πρέπει να κινηθεί και να πάει στον προορισμό 1.

```

case 4:
    if (b_gms.correctAnswer == "Aocean")
    {
        correctAnswer(buttons[2]);
        currentDestination++;
    }
}

```

Σχήμα 4.22: Ξεκίνημα του ταξιδιού με την αύξηση του `currentDestination`

Στο ταξίδι του Βαρθολομαίου Ντιάζ κατά τη διάσχιση του Ατλαντικού ωκεανού πιάνει καταιγίδα. Σε αυτό το σημείο εμφανίζεται ένα prefab, το οποίο είναι δύο σύννεφα που κινούνται με αντίθετη φορά και πλησιάζουν αργά μεταξύ τους και όταν ακουμπήσουν, ενεργοποιείται ένα Game Object το οποίο έχει Particle System component ενσωματωμένο πάνω του και είναι το αντικείμενο που αναπαριστά την καταιγίδα. Το prefab της καταιγίδας φαίνεται στο Σχήμα 4.23. Στη συνέχεια εμφανίζεται η πέμπτη ερώτηση, για να συνεχιστεί το παιχνίδι.



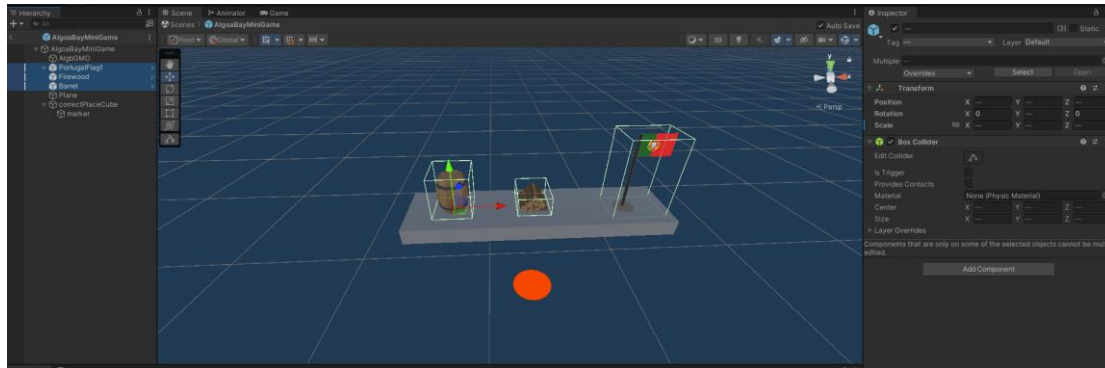
Σχήμα 4.23: Το prefab της καταιγίδας που εμφανίζεται

Η πέμπτη ερώτηση ζητάει από τον παίκτη να απαντήσει πού έφτασε ο Βαρθολομαίος Ντιάζ, όταν προσπάθησε να ξεφύγει από την κακοκαιρία και η σωστή απάντηση είναι ο «κόλπος Αλγκόα». Κατά την άφιξή του σε αυτό το σημείο ο Βαρθολομαίος τοποθέτησε μια σημαία ως αντικείμενο της πορτογαλικής κυριαρχίας. Έτσι, εμφανίζεται ως μίνι παιχνίδι ένα ταμπλό με τρία αντικείμενα και μια πορτοκαλί κουκκίδα δίπλα στο πλοίο (Σχήμα 4.24). Ο παίκτης πρέπει να σύρει τη σημαία πάνω στην πορτοκαλί κουκκίδα, για να απαντήσει στην ερώτηση του μίνι παιχνιδιού. Πρόκειται για μία drag and drop διαδικασία, που περιγράφεται παρακάτω.



Σχήμα 4.24: Το μίνι παιχνίδι στον κόλπο Αλγκόα

Τα τρία αντικείμενα που υπάρχουν πάνω στο γκρι ταμπλό που εμφανίζεται, έχουν το κάθε ένα από ένα Box Collider component (Σχήμα 4.25).



Σχήμα 4.25: Τα Box Collider components των αντικειμένων στον κόλπο Αλγκόα

Επιπροσθέτως και η πορτοκαλί κουκκίδα έχει ένα Box Collider αλλά και ένα Rigidbody component. Αυτά τα components βοηθούν στην ανίχνευση συγκρούσεων.

Αρχικά, τα τρία αντικείμενα που δίνονται προς απάντηση έχουν κάποια tags. Η πορτογαλική σημαία έχει το tag «B\_Alg\_correct», που σημαίνει πως είναι το σωστό αντικείμενο στο μίνι παιχνίδι του Algoa Bay, ενώ τα άλλα δύο έχουν το tag «B\_Alg\_wrong», που σημαίνει πως είναι τα λανθασμένα αντικείμενα στο μίνι παιχνίδι του Algoa Bay. Έτσι, όταν ο παίκτης σύρει κάποιο από τα τρία αντικείμενα στην πορτοκαλί κουκκίδα, συμβαίνει μια σύγκρουση και ελέγχεται το tag του αντικειμένου, το οποίο ο παίκτης έσυρε. Αν αυτό έχει το σωστό tag, τότε ο παίκτης απάντησε σωστά και το παιχνίδι συνεχίζεται. Διαφορετικά, σημαίνει ο παίκτης πως απάντησε λανθασμένα και πρέπει να ξαναπροσπαθήσει. Αυτή η διαδικασία επιτυγχάνεται με τον κώδικα που φαίνεται στο Σχήμα 4.26.

```

if (touch.phase == TouchPhase.Ended || touch.phase == TouchPhase.Canceled)
{
    if (selectedObject != null)
    {
        switch (triggerOn)
        {
            case false:
                selectedObject.transform.position = originalPosition;
                selectedObject = null;
                break;
            case true:
                if (selectedObject.CompareTag("B_Alg_correct"))
                {
                    correctAnswer();
                    //selectedObject = null;
                    triggerOn = false;
                }
                else
                {
                    b_mcm.activateWrongPopup();
                    selectedObject.transform.position = originalPosition;
                    selectedObject = null;
                    triggerOn = false;
                }
                break;
        }
    }
    else
    {
        Debug.Log("selected Object is null");
    }
}

```

Σχήμα 4.26: Ο έλεγχος των αντικειμένων στο παιχνίδι drag and drop του Βαρθολομαίου Ντιάζ

Η μεταβλητή triggerOn τύπου boolean που υπάρχει, είναι για να δηλωθεί στο πρόγραμμα αν ο παίκτης έχει σύρει κάποιο από τα τρία αντικείμενα πάνω στην κουκκίδα, είτε είναι το σωστό είτε το λανθασμένο αντικείμενο. Αυτή η διαδικασία υπάρχει, διότι σε περίπτωση που ο παίκτης απλά σύρει κάποιο αντικείμενο οπουδήποτε αλλού εκτός της πορτοκαλί κουκκίδας –που είναι το σημείο, όπου πρέπει να τοποθετήσει το σωστό αντικείμενο–, τότε δεν πρέπει να συμβαίνει τίποτα και απλά να επιστρέφει το αντικείμενο που επέλεξε ο παίκτης στην αρχική του θέση. Για τον λόγο αυτό

εξετάζονται τα tags, μόνο όταν συμβαίνει κάποια σύγκρουση, δηλαδή στην περίπτωση που η μεταβλητή `triggerOn` ισούται με `true`, που σημαίνει πως ο παίκτης έσυρε κάποιο αντικείμενο πάνω στην κουκκίδα. Επίσης, το αντικείμενο, το οποίο επιλέγει και μπορεί να μετακινήσει ο παίκτης, αποθηκεύεται σε μια μεταβλητή `selectedObject` τύπου `GameObject`, η οποία γίνεται `null` κάθε φορά που ο παίκτης τελειώνει τη διαδικασία μετακίνησης, ώστε να μπορεί να επιλεγθεί εκ νέου το ίδιο ή διαφορετικό αντικείμενο. Η όλη διαδικασία καλείται στη μέθοδο `Update()`.

Η όλη διαδικασία `drag and drop` γίνεται ως εξής: αρχικά ελέγχεται αν ο χρήστης έχει ακουμπήσει την οθόνη του και άρα υπάρχει διαδικασία αφής. Στη συνέχεια δημιουργείται μια νοητή ακτίνα από το σημείο που έχει αγγίξει ο παίκτης στην οθόνη και επεκτείνεται κατά βάθος σε ευθεία κατεύθυνση. Αν η γραμμή «χτυπήσει» κάποιο αντικείμενο, το οποίο έχει `Box Collider component`, σημαίνει πως ο παίκτης έχει πατήσει πάνω σε κάποιο αντικείμενο. Όμως, για αποφυγή λαθών ο παίκτης δεν πρέπει να αλληλεπιδρά με οποιοδήποτε αντικείμενο έχει `Box Collider component` αλλά μόνο με όσα φέρουν το tag `B_Alg_correct` ή `B_Alg_Wrong`, εφόσον πρόκειται για το μίνι παιχνίδι του κόλπου Αλγκόα. Έτσι, στη συνέχεια ελέγχεται το tag του αντικειμένου το οποίο επέλεξε ο παίκτης και, αν αυτό φέρει κάποια από τα δύο προαναφερθέντα tags, τότε αυτό το αντικείμενο αποθηκεύεται στη μεταβλητή `selectedObject`, για την οποία –πάλι για την αποφυγή λαθών– ελέγχεται πρώτα εάν είναι `null`, πριν αποθηκευτεί οτιδήποτε. Επίσης, αποθηκεύεται και η αρχική θέση του `selectedObject`, διότι σε περίπτωση που ο παίκτης επιλέξει λανθασμένο αντικείμενο, αυτό πρέπει να επιστρέφει στην αρχική του θέση, μόλις τελειώσει η διαδικασία `drag and drop`. Όλα αυτά γίνονται μέσα στον έλεγχο `if (touch.phase == TouchPhase.Began)`, ο οποίος καλείται στη μέθοδο `Update()`.

Στη συνέχεια ακολουθεί ο δεύτερος έλεγχος `if (touch.phase == Touch.Moved)`, όπου αρχικά αποθηκεύεται η θέση του σημείου αφής –που ο παίκτης άγγιξε την οθόνη για να επιλέξει κάποιο αντικείμενο– στη μεταβλητή `positionOfTouch`. Στη συνέχεια μετατρέπονται οι συντεταγμένες της `positionOfTouch` –ουσιαστικά πρόκειται για συντεταγμένες πάνω στην οθόνη– σε συντεταγμένες του κόσμου του παιχνιδιού με τις δύο επόμενες εντολές. Τέλος, τίθεται κάθε φορά η τιμή της θέσης του `selectedObject` στην τιμή της θέσης του `worldPosition` και αυτό επιτυγχάνεται σε κάθε `frame` του παιχνιδιού, διότι ο εξωτερικός έλεγχος καλείται στη μέθοδο `Update()`. Όλη η παραπάνω διαδικασία φαίνεται αναλυτικά στο Σχήμα 4.27.

```

if (touch.phase == TouchPhase.Began)
{
    ray = Camera.main.ScreenPointToRay(touch.screenPosition);
    if (Physics.Raycast(ray, out hit))
    {
        //Εδώ με τη μέθοδο raycast ουσιαστικά, δημιουργείται η ακτίνα ray που έχει οριστεί πιο πάνω
        //μέσω στη μέθοδο, ελέγχεται αν η ακτίνα ray έχει "χτυπήσει" κάποιο αντικείμενο το οποίο έχει collider. Η πληροφορία αυτή αποθηκεύεται και επιστ
        //ρακεται πίσω
        if (hit.collider.CompareTag("B_Alg_correct") || hit.collider.CompareTag("B_Alg_wrong")) //αν η ακτίνα έχει χτυπήσει κάποιο αντικείμενο με τα ο
        {
            if (selectedObject == null)
            {
                //Όσο αν το selfObject (το οποίο είναι) ένα από τα αντικείμενα που θέλουμε να κάνουμε drag and drop, είναι null, τότε
                //αποθηκεύουμε αυτό το αντικείμενο στο selfObject
                originalPosition = hit.collider.gameObject.transform.position; //αποθηκεύουμε και την αρχική θέση του αντικείμενου το οποίο επιλέξαμε
                selectedObject = hit.collider.gameObject; //διότι αυτό θα μας βοηθήσει να το επιστρέψουμε πίσω στην θέση του, αν
                Debug.Log("selected Object got in");
            }
        }
    }
}

if (touch.phase == TouchPhase.Moved)
{
    //Εδώ πραγματοποιείται η διαδικασία drag and drop του selfObject (το οποίο είναι) το αντικείμενο του παιχνιδιού που επιλέξαμε
    if (selectedObject != null)
    {
        //γίνεται έλεγχος αν το selfObject δεν είναι άδεια
        Vector3 positionOfTouch = touch.screenPosition;
        Vector3 position = new Vector3(positionOfTouch.x, positionOfTouch.y,
        Camera.main.WorldToScreenPoint(selectedObject.transform.position).z);
        worldPosition = Camera.main.ScreenToWorldPoint(position);
        selectedObject.transform.position = new Vector3(worldPosition.x, originalPosition.y, worldPosition.z);
    }
    else
    {
        Debug.Log("selected object is null");
    }
}

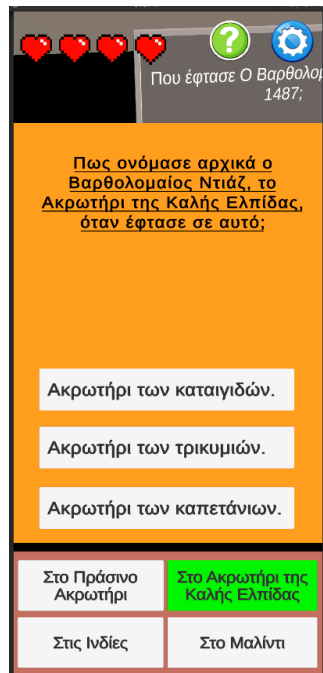
```

Σχήμα 4.27: Η διαδικασία `drag and drop` του μίνι παιχνιδιού στο `Algoa Bay`

## Κεφάλαιο 4°

Αφού ο παίκτης απαντήσει σωστά στο μίνι παιχνίδι, τοποθετείται ξανά η πορτογαλική σημαία στην πορτοκαλί κουκκίδα και το παιχνίδι συνεχίζεται (Σχήμα 4.28).

Η έκτη ερώτηση αφορά τον τόπο, όπου έφτασε ο Βαρθολομαίος Ντιάζ μετά τον κόλπο Αλγκόα, δηλαδή στο Ακρωτήριο της Καλής Ελπίδας. Όταν φτάνει σε αυτό το σημείο, εμφανίζεται ένα ακόμη μίνι παιχνίδι Quiz, το οποίο ρωτά πώς ονόμασε ο Βαρθολομαίος Ντιάζ το Ακρωτήριο της Καλής Ελπίδας (Σχήμα 4.28). Όταν απαντηθεί σωστά η ερώτηση –«Ακρωτήριο των Τρικυμιών»–, εμφανίζεται στο σημείο αυτό ένας σταυρός, τον οποίο είχε τοποθετήσει ο Βαρθολομαίος Ντιάζ, όταν έφτασε εκεί (Σχήμα 4.29).



Σχήμα 4.28: Το μίνι παιχνίδι Quiz στο Ακρωτήριο της Καλής Ελπίδας

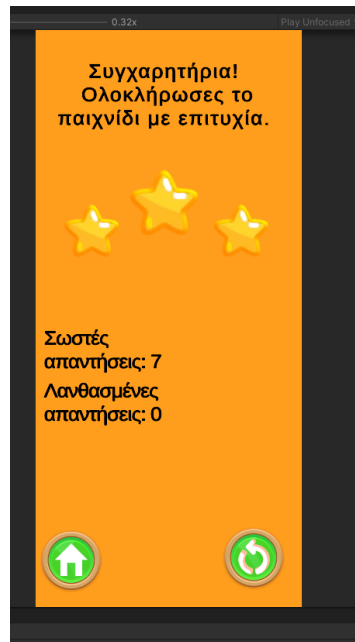


Σχήμα 4.29: Η τοποθέτηση του σταυρού

Τέλος, η έβδομη και τελευταία ερώτηση αφορά στον τόπο, όπου επέστρεψε τελικά ο Βαρθολομαίος Ντιάζ, αφού ολοκλήρωσε το ταξίδι του. Με τη σωστή απάντηση το πλοίο επιστρέφει στην Πορτογαλία. Αυτή τη φορά η διακεκομμένη γραμμή είναι πράσινη, για να σηματοδοτήσει την επιστροφή (Σχήμα 4.30). Τέλος, εμφανίζεται το μήνυμα επιτυχούς ολοκλήρωσης του παιχνιδιού και ο παίκτης έχει κερδίσει στο παιχνίδι (Σχήμα 4.31).



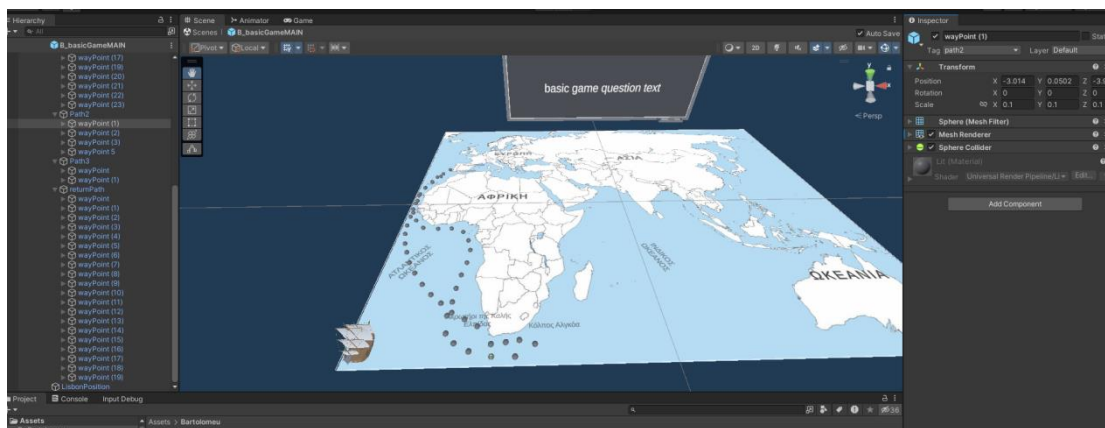
Σχήμα 4.30: Η επιστροφή του Βαρθολομαίου Ντιάζ



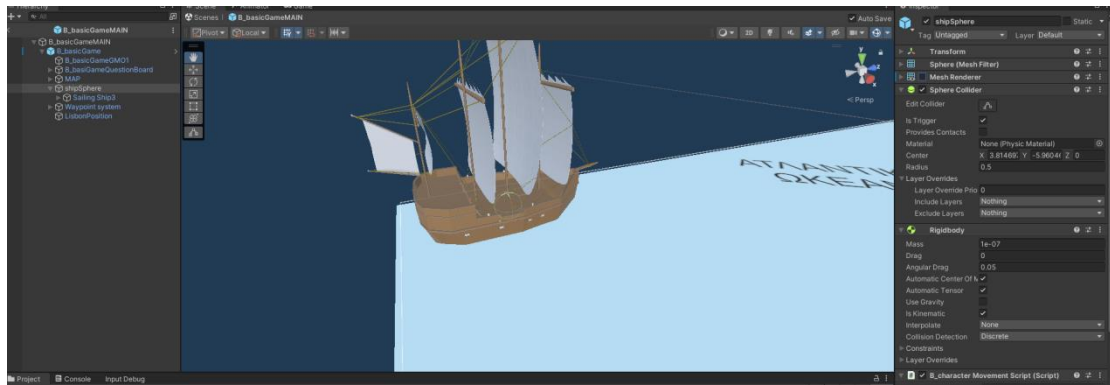
Σχήμα 4.31: Η επιτυχής ολοκλήρωση του παιχνιδιού του Βαρθολομαίου Ντιάζ

#### 4.3.1.4 Η κίνηση του πλοίου

Πάνω στον χάρτη υπάρχουν ορισμένα σημεία-σφαίρες, τα οποία ονομάζονται waypoints (Σχήμα 4.32). Τα waypoints έχουν Sphere Collider component πάνω τους, για να ανιχνεύουν συγκρούσεις. Αυτό είναι χρήσιμο, διότι το πλοίο έχει ως πατέρα (parent) στην ιεραρχία του παιχνιδιού μια σφαίρα, η οποία, επίσης, έχει ένα Sphere Collider αλλά και επιπλέον ένα Rigidbody component (Σχήμα 4.33).



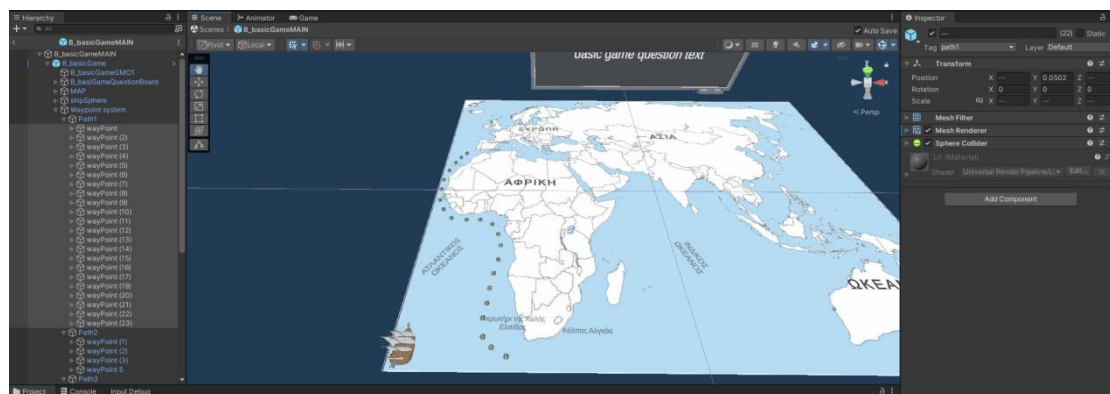
Σχήμα 4.32: Τα waypoints πάνω στον χάρτη



Σχήμα 4.33 Το στοιχείο Ship sphere

Επιπλέον, τα waypoints ανήκουν σε διαφορετικά paths και έχουν το δικό τους tag ανάλογα με το path, στο οποίο ανήκουν. Αυτό βοηθά να γίνει ξεκάθαρο σε ποιο waypoint βρίσκεται το πλοίο ανά πάσα στιγμή, γιατί ορισμένα waypoints από κάποιο path μπορεί να πέφτουν πάνω σε κάποιο waypoint διαφορετικού path και κατά συνέπεια θα υπολογίζονται διπλές συγκρούσεις μεταξύ των Colliders.

Για παράδειγμα, το path1 καθώς και τα waypoints του, που φέρουν το tag «path1», για να δηλωθεί πως ανήκουν σε αυτό, φαίνονται στο Σχήμα 4.34.



Σχήμα 4.34: Τα waypoints του path1

Οι εντολές, που διαχειρίζονται την κίνηση του πλοίου, είναι γραμμένες σε ένα αρχείο εντολών `B_characterMovementScript`, τα βασικά μέρη του οποίου παρουσιάζονται παρακάτω.

Αρχικά, στο συγκεκριμένο παιχνίδι το ταξίδι του θαλασσοπόρου ξεκινά με την απάντηση της ερώτησης 4, όπου και η τιμή της μεταβλητής `currentDestination` του `B_ButtonManagementScript` γίνεται 1, όπως προαναφέρθηκε. Υπάρχει μια μέθοδος, η `characterMovement()`, η οποία είναι υπεύθυνη για την κίνηση του πλοίου και φαίνεται στο Σχήμα 4.35.

```

//methodos kinshs ths barkas
private void characterMovement(GameObject[] currentPath, int currentWP)
{
    if (currentWP <= currentPath.Length - 1) //an o deikths einai mikroteros apo to mhkos tou pinaka (dhladh an uparxει akoma epomenos proorismos)
    {
        Quaternion lookatWP = Quaternion.LookRotation(currentPath[currentWP].transform.position - character.transform.position);
        character.transform.rotation = Quaternion.Slerp(character.transform.rotation, lookatWP, rotationSpeed * Time.deltaTime);
        character.transform.Translate(0, 0, speed * Time.deltaTime);
    }
    else if (currentWP == currentPath.Length)
    {
        character.transform.position = currentPath[currentPath.Length - 1].transform.position; //Εδώ σιγουρευομαστε, πως στan το currentWP φτάσe
        //του προορισμού, τότε για αποφυγή τυχών οφθαλμάτων, κλειδώνουμε τη θέση της βάρκας στο τελευταίο σημείο.
        informationPrefabSpawner();
    }
}

```

Σχήμα 4.35: Η μέθοδος characterMovement()

Η μέθοδος characterMovement() δέχεται δύο παραμέτρους κατά την κλήση της. Η πρώτη παράμετρος είναι ένας πίνακας με το όνομα currentPath και περιέχει όλα τα waypoints του τρέχοντος path και η δεύτερη μεταβλητή currentWP είναι ένας μετρητής τύπου int που δηλώνει κάθε στιγμή σε ποιο waypoint του συγκεκριμένου path βρίσκεται το πλοίο και έχει αρχική τιμή 0. Στο Σχήμα 4.36 φαίνεται ο πίνακας path1 που αντιστοιχεί στο path1 πάνω στον χάρτη και περιέχει ως στοιχεία του όλα τα waypoints αυτού. Αντιστοίχως υπάρχει ένας πίνακας για κάθε path και ένας μετρητής currentWP για κάθε πίνακα.



Σχήμα 4.36: Ο πίνακας path1

Στον δεύτερο έλεγχο if, όταν φτάσει το πλοίο στο τελευταίο waypoint, η θέση του παραμένει εκεί και εμφανίζεται στην οθόνη το παράθυρο πληροφοριών για τον συγκεκριμένο προορισμό του θαλασσοπόρου.

Υπάρχει μια μέθοδος ActivateCharacterMovement(), η οποία μέσα σε έναν έλεγχο switch-case εξετάζει κάθε φορά την τιμή της μεταβλητής currentDestination και αναλόγως, με τη σειρά της καλεί τη μέθοδο characterMovement(), δίνοντας κάθε φορά διαφορετικές παραμέτρους ως είσοδο, ώστε κάθε φορά το πλοίο να ακολουθεί το επιθυμητό path, για να φτάσει στον τελικό προορισμό. Η μέθοδος ActivateCharacterMovement() φαίνεται στο Σχήμα 4.37.

```

//methodos elegxou metablhtwn pou ephreazoun thn kinsh
private void ActivateCharacterMovement()
{
    switch (b_bms.currentDestination) //elegxetai o sugkekrimenos proorismos ston opoio briskomaste wste na kserei to programma ka8e poio monopati 8a akolou8hsei
    {
        case 1:
            characterMovement(path1, currentWP1);
            break;
        case 2:
            characterMovement(path2, currentWP2);
            break;
        case 3:
            characterMovement(path3, currentWP3);
            break;
        case 4:
            characterMovement(returnPath, returnWP);
            break;
    }
}

```

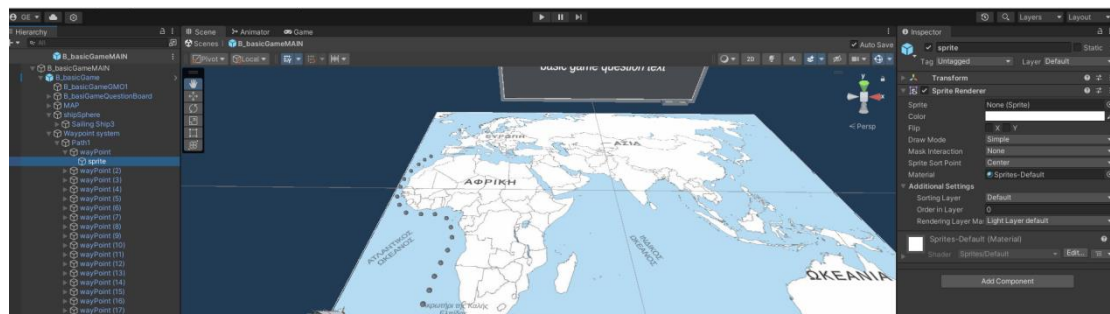
Σχήμα 4.37: Η μέθοδος ActivateCharacterMovement()

Αναλυτικότερα, όταν η μεταβλητή `currentDestination` γίνει 1, καλείται αρχικά η μέθοδος `ActivateCharacterMovement()`. Αυτή με τη σειρά της καλεί τη μέθοδο `characterMovement()`, δίνοντας ως παραμέτρους τον πίνακα `path1` και τη μεταβλητή `currentWP1`. Με τον τρόπο αυτό το πλοίο «γνωρίζει» πως για να φτάσει στον τρέχοντα προορισμό –τον προορισμό 1–, θα ακολουθήσει ένα-ένα όλα τα `waypoints` του `path1`. Αρχικά φτάνει στο πρώτο `waypoint`, σημείο από όπου αρχίζουν να παίζουν ρόλο οι συγκρούσεις μεταξύ των `colliders`. Όταν φτάσει στο πρώτο `waypoint`, ανιχνεύεται σύγκρουση λόγω των `colliders`. Εδώ ξεκινά η λειτουργία μιας άλλης μεθόδου, της μεθόδου `OnTriggerEnter()`, που ανιχνεύει συγκρούσεις.

Κάθε φορά που το πλοίο φτάνει σε ένα `waypoint` και συμβαίνει μια σύγκρουση, αυξάνεται ο τρέχων μετρητής `currentWP` κατά ένα. Έτσι, μετά την πρώτη σύγκρουση κατά το ταξίδι προς κάποιον προορισμό, ο μετρητής `currentWP` γίνεται 1. Τότε το πλοίο κινείται προς το δεύτερο `waypoint`, ο μετρητής 2 και ούτω καθεξής. Η διαδικασία επαναλαμβάνεται, έως ότου καλυφθούν όλα τα `waypoints` του συγκεκριμένου κάθε φορά `path` και μόλις συμβεί αυτό, το πλοίο έχει φτάσει σε κάποιον προορισμό.

Η μέθοδος `OnTriggerEnter()`, επίσης, είναι αυτή που ελέγχει το `tag` του κάθε `waypoint`, ώστε να αποφεύγονται οι διπλές μετρήσεις συγκρούσεων. Τμήμα της λειτουργίας αυτής φαίνεται στο Σχήμα 4.40.

Επιπροσθέτως, κάθε `waypoint` έχει στην ιεραρχία του ως «παιδί» ένα αντικείμενο `Sprite`, που αρχικά δεν έχει καμία εικόνα μέσα του (Σχήμα 4.38).

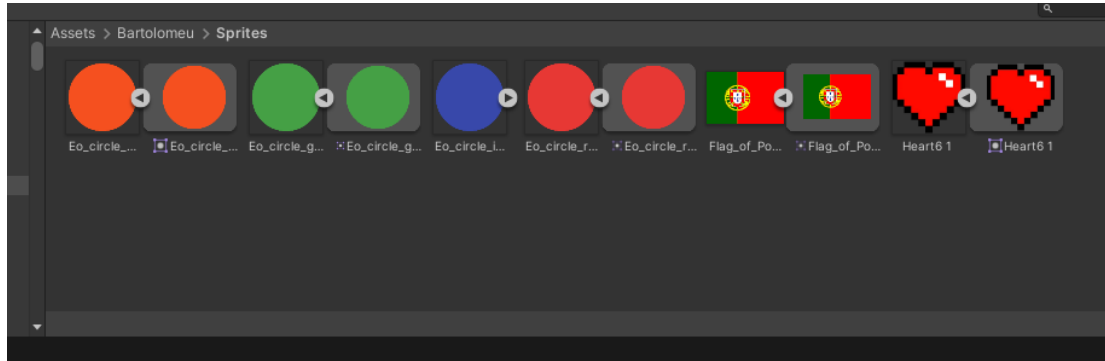


Σχήμα 4.38: Το sprite του πρώτου waypoint

Η μέθοδος `OnTriggerEnter()`, ανάλογα σε ποια διαδρομή `path` βρίσκεται το πλοίο, κάθε φορά που περνάει πάνω από ένα `waypoint`, θέτει τη μεταβλητή `Sprite` «του παιδιού» του σε `redSprite`. Με τον

## Κεφάλαιο 4°

τρόπο αυτό εμφανίζεται η διακεκομμένη κόκκινη γραμμή, που χαράζει την πορεία του πλοίου. Η μόνη διαφορά κατά την επιστροφή είναι, πως αντί για redSprite θέτει την εικόνα των waypoints της τελευταίας διαδρομής σε greenSprite που σηματοδοτεί την επιστροφή του θαλασσοπόρου. Στο Σχήμα 4.39 φαίνονται τα sprites που χρησιμοποιήθηκαν.



Σχήμα 4.39: Τα sprites που χρησιμοποιούνται στο παιχνίδι

```
void OnTriggerEnter(Collider currentWayPoint)
{
    //h methodos pou orizei to epomeno shmeio gia na metakinhei h barka/
    //an exei ginei collide, dhladh exei ftasei hdh se ena shmeio, tote arxika elegxei to currentDestination gia na dei poion metrth ba aukshsei.
    //epeita, oi metrhtes xrhsimopoiountai gia ta diaforetika paths pou uparxoun (ka8e path exei diko tou metrth). Xrhsimopoiountai ws deiktos stous pinakes tun paths
    //kai ousiastika, ka8e fora pou auksanetai enas deikths, tote to programma deixnei sto epomeno wayPoint pou prepei na metakinhei h barka.
    // me liga logia an exeis ftasei sto shmeio path[currentWP], tote metakinhse th barka sto shmeio path[currentWP+1] kai epanalabe th diadikasia mexri na telewsoun ola ta shmeia
    //gia to sugkekrimeno path sto opoio briskesai.

    switch (b_bms.currentDestination)
    {
        case 1:
            if (currentWayPoint.CompareTag("path1"))
            {
                path1[currentWP1].GetNamedChild("sprite").GetComponent<SpriteRenderer>().sprite = sprites[0];
                Debug.Log("collision with waypoint: " + currentWP1);
                currentWP1++;
            }
            break;
        case 2:
            if (currentWayPoint.CompareTag("path2"))
            {
                path2[currentWP2].GetNamedChild("sprite").GetComponent<SpriteRenderer>().sprite = sprites[0];
                currentWP2++;
            }
            break;
    }
}
```

Σχήμα 4.40: Τμήμα της μεθόδου OnTriggerEnter() στην κίνηση του πλοίου

Τέλος, όταν στη μέθοδο characterMovement(), ισχύσει η συνθήκη `if (currentWP == currentPath.length)`, σημαίνει πως το πλοίο έχει διασχίσει όλα τα waypoints της συγκεκριμένης διαδρομής path και έχει φτάσει στον προορισμό του. Τέλος, για την αποφυγή λαθών κλειδώνεται η θέση του πλοίου στο τελευταίο waypoint –που είναι και ο τελικός προορισμός– και καλείται η μέθοδος `informationPrefabSpawner()`, η οποία φαίνεται στο Σχήμα 4.41.

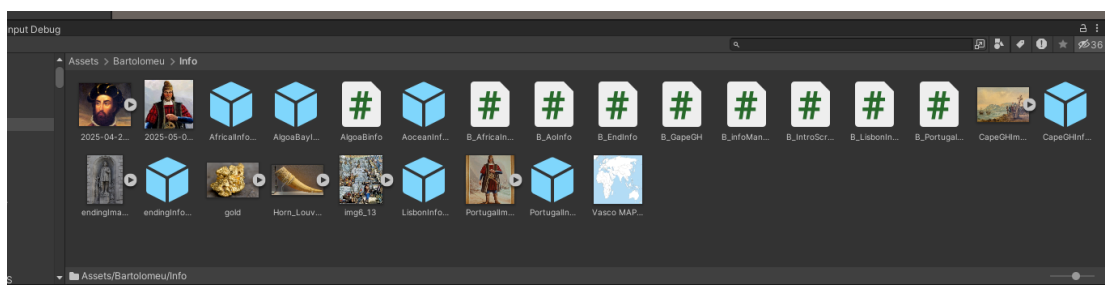
```

public void infoPrefabSpawner()
{
    switch (b_bms.currentDestination)
    {
        //Η παρακάτω διαδικασία αυτο που κάνει είναι να εμφανίζει τα infoPrefab κάθε προορισμού. Ουσιαστικά, αφού έχουμε φτάσει στο τελευταίο σημείο του προορισμού της βάρκας,
        //τότε θέλουμε να εμφανίζονται οι πληροφορίες. Έτσι ανάλογα με τον προορισμό στον οποίο έχει φτάσει η βάρκα, εμφανίζεται το αντίστοιχο παράθυρο πληροφοριών. Όμως επειδή
        //καλείται μέσω της ActivateCharacterMovement στην update, πρέπει να μπει μια μεταβλητή που θα σταματήσει την επανεμφάνιση της πληροφορίας, αλλιώς θα εκαναν spawn συνεχώς
        //Έτσι μετά από το 1ο spawn κάθε πληροφορίας, ουσιαστικά απενεργοποιούμε τη διαδικασία δημιουργίας τους, άρα δημιουργούνται μόνο μια φορά.
        case 2:
            StartCoroutine(spawnPrefab(infos[0], B_mainCanvasT, 1.5f));
            currentMP1++;
            break;
        case 3:
            StartCoroutine(spawnPrefab(infos[1], B_mainCanvasT, 0.1f));
            currentMP2++;
            break;
        case 4:
            StartCoroutine(spawnPrefab(infos[2], B_mainCanvasT, 0.1f));
            currentMP3++;
            break;
        case 5:
            b_mcm.deactivateGuideButton();
            b_mcm.deactivateAnswerButtons();
            StartCoroutine(spawnPrefab(infos[3], B_mainCanvasT, 0.1f));
            returnMP++;
            break;
    }
}

```

Σχήμα 4.41: Η μέθοδος infoPrefabSpawner()

Η μέθοδος infoPrefabSpawner() εξετάζει την τιμή της μεταβλητής currentDestination, και ανάλογα με τον τρέχοντα προορισμό στον οποίο έφτασε μόλις το πλοίο, δημιουργεί το αντίστοιχο prefab πληροφοριών. Ο φάκελος και τα αρχεία για τις πληροφορίες που χρησιμοποιήθηκαν και υπάρχουν στον φάκελο infoFolder, φαίνονται στο Σχήμα 4.42.

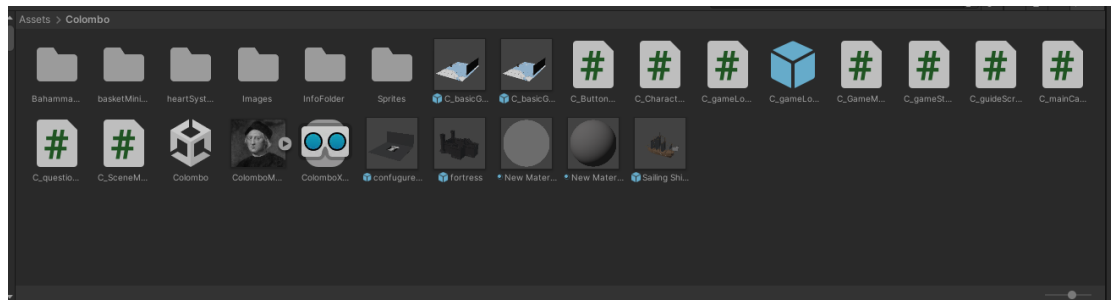


Σχήμα 4.42: Ο φάκελος infoFolder του Βαρθολομαίου Ντιάζ

## 4.3.2 Το παιχνίδι του Χριστόφορου Κολόμβου

### 4.3.2.1 Βασικά στοιχεία του παιχνιδιού

Το παιχνίδι ξεκινά με τον βασικό χάρτη και το πλοίο τοποθετημένο επάνω του. Αφού αυτός εμφανιστεί, εμφανίζονται οι ζωές, τα κουμπιά των ρυθμίσεων και των οδηγιών καθώς και τα κουμπιά των απαντήσεων του βασικού μέρους του παιχνιδιού. Οι λειτουργίες των κουμπιών των απαντήσεων του βασικού παιχνιδιού, η κίνηση του πλοίου και ο τρόπος με τον οποίο εμφανίζονται οι πληροφορίες, είναι ίδιος με το προηγούμενο παιχνίδι – καθώς και με όλα τα επόμενα παιχνίδια. Στο Σχήμα 4.43 φαίνονται τα αρχεία που χρησιμοποιήθηκαν σε αυτό το παιχνίδι και στο Σχήμα 4.44 φαίνεται το εισαγωγικό παράθυρο, που εμφανίζεται, όταν ο παίκτης επιλέξει το παιχνίδι του Χριστόφορου Κολόμβου.

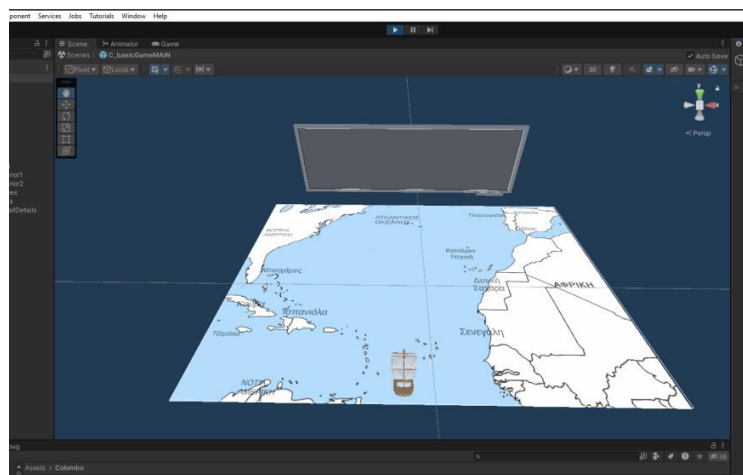


Σχήμα 4.43: Τα αρχεία στο παιχνίδι του Χριστόφορου Κολόμβου



Σχήμα 4.44: Το εισαγωγικό παράθυρο του παιχνιδιού του Χριστόφορου Κολόμβου

Αφού κλείσει το εισαγωγικό παράθυρο και ο παίκτης σκανάρει το ασπρόμαυρο πορτραίτο του θαλασσοπόρου, εμφανίζεται ο βασικός χάρτης με το πλοίο πάνω του (Σχήμα 4.45).



Σχήμα 4.45: Ο βασικός χάρτης του Χριστόφορου Κολόμβου

### 4.3.2.2 Η λειτουργία του παιχνιδιού

Η πρώτη ερώτηση αναφέρεται στην αφετηρία του ταξιδιού του Χριστόφορου Κολόμβου. Με τη σωστή απάντηση τοπλοθετείται το πλοίο στο Πάλος (Σχήμα 4.46) και αυτό γίνεται με τον κώδικα που φαίνεται στο Σχήμα 4.47.



Σχήμα 4.46: Το πλοίο στην τοποθεσία Πάλος

```
public void ButtonAManagement()
{
    switch (c_gms.questionCounter)
    {
        case 1:
            if (c_gms.correctAnswer == "Palos")
            {
                correctAnswer(buttons[0]);
                boat.transform.position = GameObject.Find("palosLocation").transform.position;
                boat.transform.Rotate(0, 220, 0);
                StartCoroutine(customInstantiate(palosInfo, C_mainCanvasT, 0.1f));
            }
            break;
    }
}
```

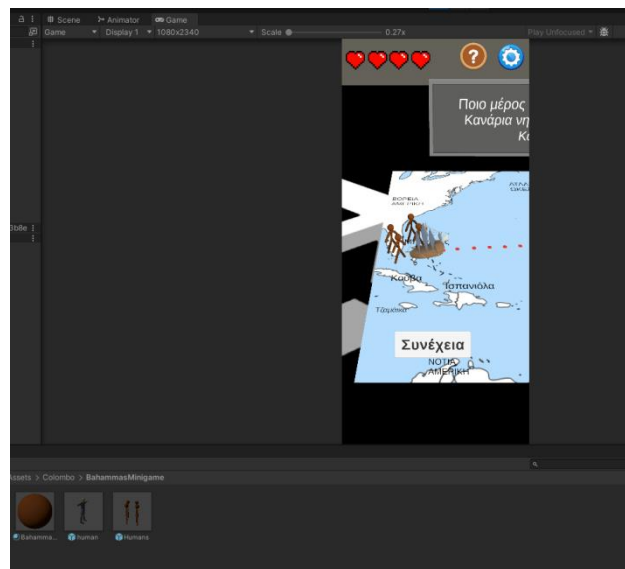
Σχήμα 4.47: Κώδικας που τοποθετεί το πλοίο στο Πάλος στο παιχνίδι του Κολόμβου

Η δεύτερη ερώτηση ζητά από τον παίκτη να απαντήσει ποιος ήταν ο πρώτος προορισμός του Κολόμβου και η σωστή απάντηση είναι τα «Κανάρια νησιά». Έτσι, ξεκινά η κίνηση του πλοίου, το οποίο φτάνει στα Κανάρια νησιά (Σχήμα 4.48).

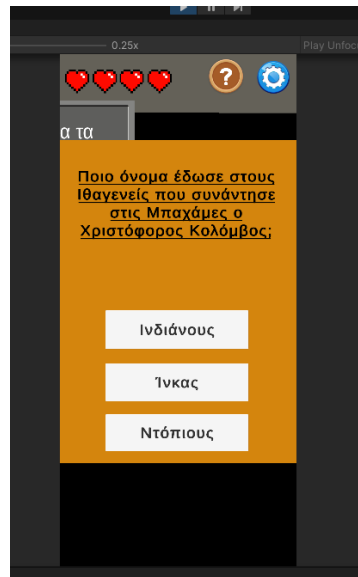


Σχήμα 4.48: Ο Κολόμβος φτάνει στα Κανάρια νησιά

Η τρίτη ερώτηση αφορά το μέρος που επισκέφθηκε ο Κολόμβος μετά τα Κανάρια νησιά και η σωστή απάντηση είναι οι «Μπαχάμες». Αν ο παίκτης απαντήσει σωστά, το πλοίο φτάνει στις Μπαχάμες. Σύμφωνα με τα ιστορικά στοιχεία εκεί ο Κολόμβος συναντά κάποιους ιθαγενείς, τους οποίους ονόμασε Ινδιάνους. Έτσι, δημιουργείται ένα prefab με ανθρώπινες φιγούρες πάνω στον χάρτη που απεικονίζουν τους Ινδιάνους (Σχήμα 4.49), και ακολουθεί ένα μίνι παιχνίδι Quiz, που ρωτά πώς ονόμασε ο Κολόμβος τους ιθαγενείς που συνάντησε (Σχήμα 4.50).



Σχήμα 4.49: Το prefab των Ινδιάνων



Σχήμα 4.50: Το μίνι παιχνίδι Quiz στις Μπαχάμες

Στη συνέχεια του ταξιδιού δίνεται η τέταρτη ερώτηση, που αφορά στην άφιξη του Κολόμβου στην Κούβα. Και εδώ εμφανίζεται ένα μίνι παιχνίδι Quiz σχετικά με τα προϊόντα που αναζητούσε ο Κολόμβος, όταν έφτασε στο νησί. Εδώ υπάρχουν τρεις εικόνες-κουμπιά, από τις οποίες η σωστή είναι «ο Χρυσός». Για να απαντήσει ο παίκτης στην ερώτηση, πρέπει να πατήσει πάνω στην εικόνα του Χρυσού (Σχήμα 4.51).



Σχήμα 4.51: Το μίνι παιχνίδι Quiz στην Κούβα

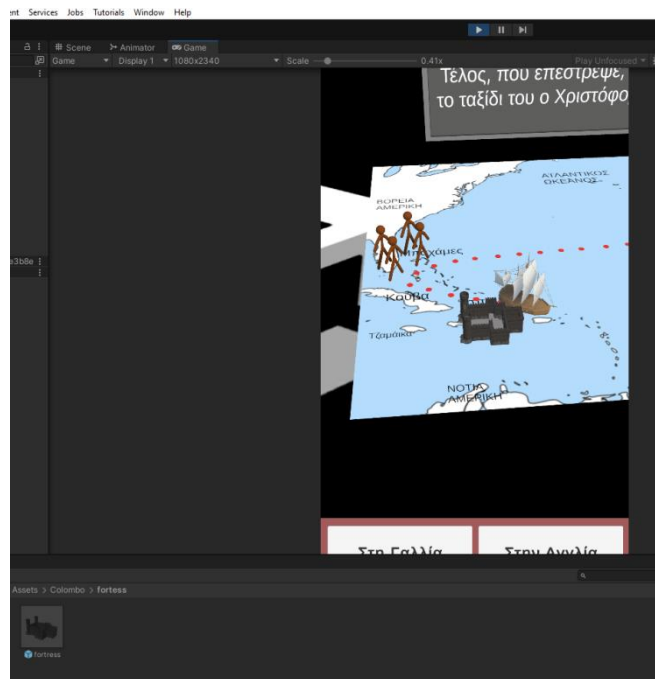
Στην επόμενη ερώτηση που αφορά στον επόμενο προορισμό του Κολόμβου, η σωστή απάντηση είναι «η Ισπανιόλα». Σε αυτό το σημείο εμφανίζεται ένα ακόμη μίνι παιχνίδι Quiz με τρεις εικόνες (Σχήμα 4.52) για το τι εγκατέστησε ο Κολόμβος στην Ισπανιόλα. Υπάρχουν τρεις εικόνες-κουμπιά και ο

## Κεφάλαιο 4°

παίχτης πρέπει να πατήσει πάνω στην εικόνα-κουμπί «Οχυρό». Στη συνέχεια εμφανίζεται ένα οχυρό πάνω στον χάρτη στην τοποθεσία της Ισπανιόλας (Σχήμα 4.53).



Σχήμα 4.52: Το μίνι παιχνίδι Quiz στην Ισπανιόλα



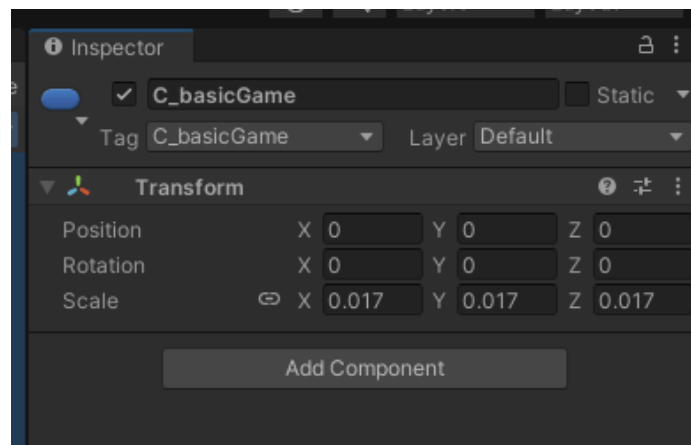
Σχήμα 4.53: Το prefab του οχυρού στην Ισπανιόλα

Η μέθοδος που δημιουργεί το οχυρό πάνω στον χάρτη είναι η `InstantiateFortress()` (Σχήμα 4.54), η οποία είναι μέθοδος-υπορουτίνα, και πρέπει να καλείται μετά από ορισμένο χρονικό διάστημα. Αυτή η μέθοδος δέχεται ως παράμετρο `delay` τύπου `float`, που είναι η μεταβλητή που ορίζει πόσο θα καθυστερήσει η μέθοδος σε δευτερόλεπτα.

```
private IEnumerator InstantiateFortress(float delay)
{
    yield return new WaitForSeconds(delay);
    Instantiate(fortress, C_basicGameT);
    Destroy(selfObject);
}
```

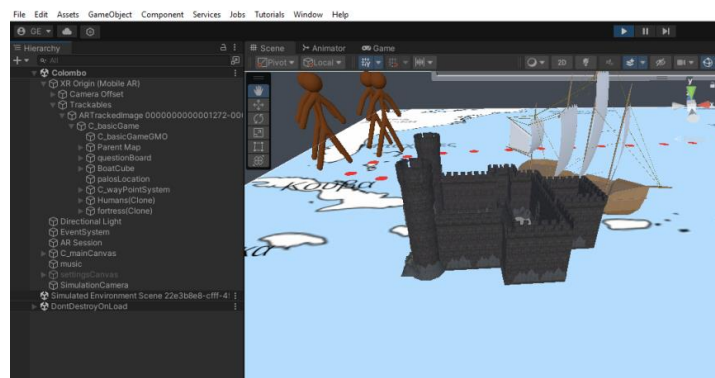
Σχήμα 4.54: Η μέθοδος InstantiateFortress() στο παιχνίδι του Κολόμβου

Στην εντολή Instantiate (fortress, C\_basicGameT) η μεταβλητή C\_basicGameT είναι μια μεταβλητή τύπου Transform και αποθηκεύει το Transform component του βασικού χάρτη του Κολόμβου, όπως φαίνεται στο Σχήμα 4.55.



Σχήμα 4.55: Το Transform component του βασικού χάρτη του Κολόμβου

Στην Instantiate(), ουσιαστικά, δημιουργείται το prefab του οχυρού, ως «παιδί» του βασικού χάρτη στην ιεραρχία του Unity και ταυτόχρονα τα τρία χαρακτηριστικά του (μέγεθος, περιστροφή, θέση) παίρνουν τιμές σε αναλογία με την τιμή του βασικού χάρτη, εφόσον στην ιεραρχία ο βασικός χάρτης είναι «πατέρας» του οχυρού στην ιεραρχία του Unity (Σχήμα 4.56).

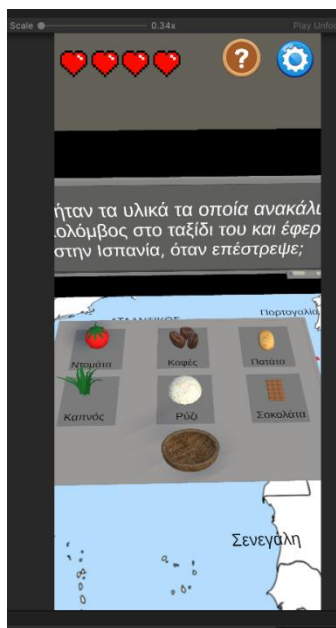


Σχήμα 4.56: Το οχυρό ως Game Object «παιδί» του βασικού χάρτη στην ιεραρχία του Unity

Τέλος, καλείται η μέθοδος Destroy(selfObject), που διαγράφει το ίδιο το Quiz από το παιχνίδι, ώστε αυτό να συνεχιστεί. Η μεταβλητή selfObject αναφέρεται στο Quiz panel της Ισπανιόλας.

### 4.3.2.3 Το μίνι παιχνίδι drag and drop του Χριστόφορου Κολόμβου

Η τελευταία ερώτηση αφορά στον τόπο, όπου επέστρεψε ο Κολόμβος με το τέλος του ταξιδιού του, και η σωστή απάντηση είναι το «Πάλος». Σύμφωνα με τα ιστορικά δεδομένα, ο γνωστός θαλασσοπόρος κατά την επιστροφή του έφερε στην πατρίδα του ορισμένα προϊόντα, που ανακάλυψε κατά τη διάρκεια του ταξιδιού του. Έτσι, πριν το πλοίο επιστρέψει, εμφανίζεται ένα μίνι παιχνίδι με έξι προϊόντα –ντομάτα, πατάτα, καφές, ρύζι, σοκολάτα και καπνός– και ένα καλάθι, το οποίο ορίζεται ως το καλάθι του Κολόμβου (Σχήμα 4.57).

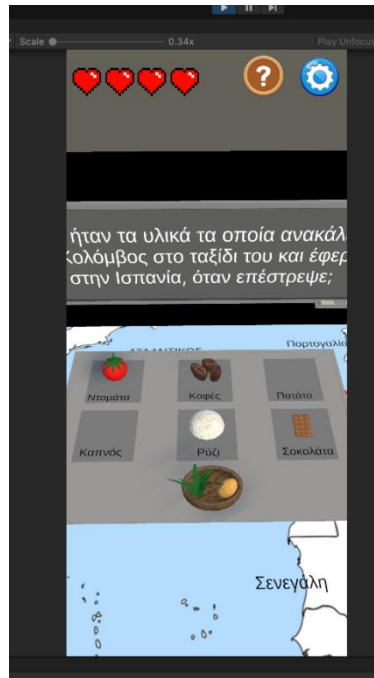


Σχήμα 4.57: Το μίνι παιχνίδι drag and drop του Κολόμβου

Πρόκειται για ένα παιχνίδι drag and drop. Ο παίκτης πρέπει να σύρει στο καλάθι τα προϊόντα που ανακάλυψε και έφερε μαζί του κατά την επιστροφή του ο Κολόμβος. Σε περίπτωση που σύρει κάποιο προϊόν –οποιοδήποτε και αν είναι αυτό– και το αφήσει έξω από το καλάθι, αυτό επιστρέφει στην αρχική του θέση. Εάν σύρει ένα από τα σωστά προϊόντα στο καλάθι, αυτό μένει εκεί –όπως φαίνεται στο Σχήμα 4.58, όπου έχουν μπει στο καλάθι ήδη δύο από τα σωστά προϊόντα– και, αν σύρει κάποιο λανθασμένο προϊόν στο καλάθι, εμφανίζεται το μήνυμα λανθασμένης απάντησης και το προϊόν επιστρέφει στην αρχική του θέση. Η διαδικασία drag and drop είναι ίδια με αυτήν του μίνι παιχνιδιού drag and drop στον κόλπο Αλγκόα στο παιχνίδι του Βαρθολομαίου Ντιάζ. Η διαφορά εδώ είναι πως υπάρχει επιπλέον ένας μετρητής υλικών/προϊόντων, που ονομάζεται ylikaCounter και είναι τύπου int.

Κάθε φορά που ο παίκτης βάζει ένα σωστό αντικείμενο μέσα στο καλάθι, ο μετρητής υλικών αυξάνεται κατά ένα (Σχήμα 4.59) και μόλις ο μετρητής γίνει ίσος με 4, σημαίνει πως ο παίκτης έχει βάλει και τα τέσσερα προϊόντα στο καλάθι και το παιχνίδι τελειώνει.

Για την αποφυγή τυχόν λαθών, σε περίπτωση που ο παίκτης επιλέξει κάποιο άλλο αντικείμενο που δεν ανήκει στο συγκεκριμένο παιχνίδι και έχει κάποιο Collider, ακολουθείται παρόμοια διαδικασία με αυτήν στο μίνι παιχνίδι (Κόλπος Αγκόα) του Βαρθολομαίου Ντιάζ.



Σχήμα 4.58: Δύο από τα σωστά προϊόντα έχουν ήδη μπει στο καλάθι του Κολόμβου

```

if ((touch.phase==TouchPhase.Ended || touch.phase == TouchPhase.Canceled) && selectedObject != null) //αν τελειώσει η διαδικασία drag και έχουμε
{
    draggingPhase = "ended"; //αρχικά δηλώνουμε ότι η διαδικασία drag έχει τελειώσει
    switch (triggerHappening) //επειτα ελεγχεται η μεταβλητη collisionHappened η οποια ενημερωνει στο προγραμμα αν έχει συμβει κάποιο OnTrigger
    {
        case false: //αν όχι, σημαίνει πως δεν έχουμε σφουρι κανένα αντικείμενο μέσα στο καλάθι, ακόμη και αν αυτο είναι το σωστό που επιλέξαμε
            //οπότε επαναφεται στην αρχική του θέση και επιστρέφει το selectedObject σε null
            selectedObject.transform.position = originalPosition;
            selectedObject = null;
            break;
        case true://αν έχει δηλωθεί κάποιο trigger event, τότε σημαίνει πως έχουμε σφουρι κάποιο αντικείμενο
            //μέσα στο καλάθι. Αρα, επετα μενει να ελεγχουμε αν είναι ένα απο τα σωστά ώστε να το αφήσουμε στο καλάθι ή να το γυρίσουμε
            //στην αρχή του θεση αναλογα
            if (selectedObject.CompareTag("C_basketWrong")) //ετσι ελεγχουμε αν φερει το tag wrongItem που σημαίνει πως είναι δεν είναι κάποιο α
            {
                //ετσι επαναφεται το αντικείμενο στην αρχική του θέση και η triggerHappened επαναφεται σε false.
                //α ελεγχος για το σωστό αντικείμενο γίνεται στο C_basketCorrectArea script
                selectedObject.transform.position = originalPosition;
                triggerHappening = false;
                selectedObject = null;
                c_mcm.activateWrongAppopup();
            }
            else if (selectedObject.CompareTag("C_basketCorrect")) //αν υπάρχει trigger event και το αντικείμενο είναι σωστό, τότε αυτο σημαίνει
            {
                //πως επιλέξαμε σωστό ένα αντικείμενο, αρα το αφήνουμε στο καλάθι (αυτο γίνεται πριν την OnMouseDown)
                //Η triggerHappening πάντα παίρνει τιμη πριν την OnMouseDown (χει σημασια αρα)
                //προκύψει κάποιο triggerEvent πρέπει να ειμαστε σε διαδικασία drag και MONO
                //τιμες, πάντα πριν την OnMouseDown()
                basketDNDCorrect();
                ylikaCounter++;
                triggerHappening = false;
                selectedObject = null;
                c_mcm.activateCorrectAppopup();
            }
            break;
    }
}

```

Σχήμα 4.59: Μέρος του κώδικα στο τελευταίο μίνι παιχνίδι του Κολόμβου

Η μέθοδος που ολοκληρώνει το παιχνίδι του Κολόμβου, είναι endGame(), η οποία καλείται μόλις το ylikaCounter γίνει ίσο με 4. Επιπλέον για την αποφυγή επαναλαμβανόμενων κλήσεων της endGame() αυξάνεται μια τελευταία φορά ο μετρητής ylikaCounter κατά ένα. Ο κώδικας που ολοκληρώνει το μίνι παιχνίδι φαίνεται στο Σχήμα 4.60.

```

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

private void checkIfGameEnded()
{
    if (ylikaCounter == 4) //σε αυτη τη μεθοδο, στον ο μετρητης υλικων γινει 4, που σημαινει πως εχουμε βαλει ολα τα σωστα αντικειμενα μεσα στο καλαθι, τοτε
    // σταματομε το παιχνιδι.
    {
        ylikaCounter++;
        StartCoroutine(endGame(1f));
    }
}

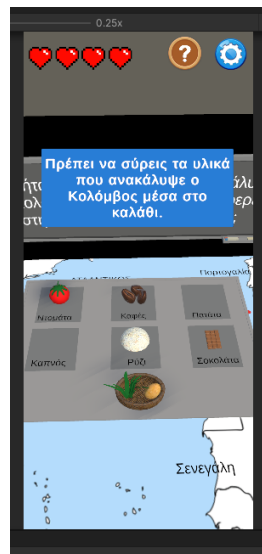
private IEnumerator endGame(float delay)
{
    yield return new WaitForSeconds(delay);
    Instantiate(basketENDinfo, C_mainCanvas);
    Destroy(selfObject);
}

```

Σχήμα 4.60: Ο κώδικας που ολοκληρώνει το μίνι παιχνίδι drag and drop του Κολόμβου

Η μεταβλητή selfObject περιέχει το μίνι παιχνίδι και η μεταβλητή basketENDinfo, περιέχει ένα παράθυρο πληροφοριών το οποίο εμφανίζεται, όταν τελειώνει το μίνι παιχνίδι.

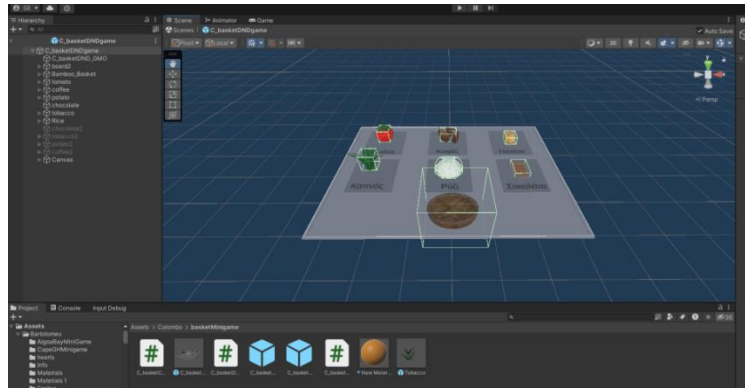
Σε περίπτωση που ο παίκτης μπερδευτεί και δυσκολευτεί σε κάποιο σημείο, μη γνωρίζοντας τι πρέπει να κάνει, πατώντας το κουμπί με το ερωτηματικό εμφανίζεται η οδηγία για το τι πρέπει να κάνει (Σχήμα 4.61).



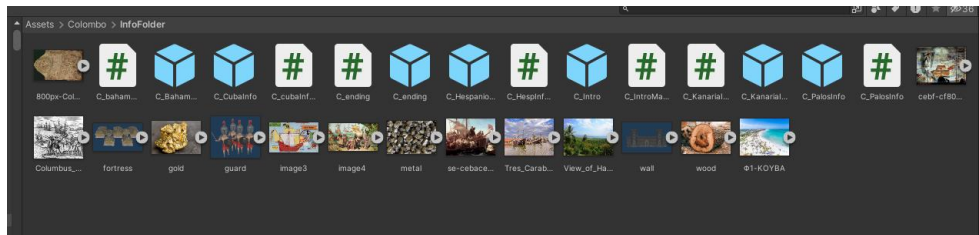
Σχήμα 4.61: Το παράθυρο οδηγιών στο μίνι παιχνίδι drag and drop του Κολόμβου

Αφού ολοκληρωθεί το μίνι παιχνίδι, το πλοίο επιστρέφει στο Πάλος. Στη συνέχεια εμφανίζεται το παράθυρο στο οποίο υπάρχουν κάποιες τελικές πληροφορίες για τη ζωή του Κολόμβου και τέλος, τα αποτελέσματα του παιχνιδιού και ο παίκτης έχει κερδίσει.

Στο Σχήμα 4.62 φαίνεται το prefab του τελευταίου μίνι παιχνιδιού του Κολόμβου στο Unity και στο Σχήμα 4.63, ο φάκελος infoFolder που περιέχει τα prefabs που εμφανίζουν τις πληροφορίες για τη ζωή του Κολόμβου που εμφανίζονται στην οθόνη, όταν το πλοίο φτάνει σε κάποιον προορισμό.



Σχήμα 4.62: Το prefab του τελευταίου μίνι παιχνιδιού του Κολόμβου

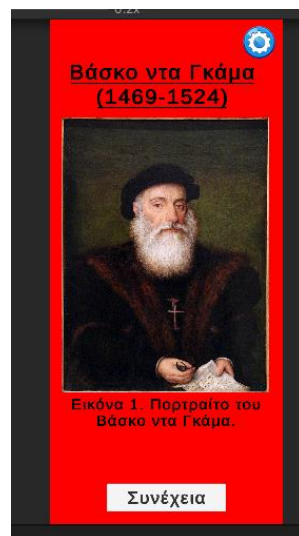


Σχήμα 4.63: Ο φάκελος πληροφοριών του Κολόμβου

### 4.3.3 Το παιχνίδι του Βάσκο ντα Γκάμα

#### 4.3.3.1 Βασικά στοιχεία του παιχνιδιού

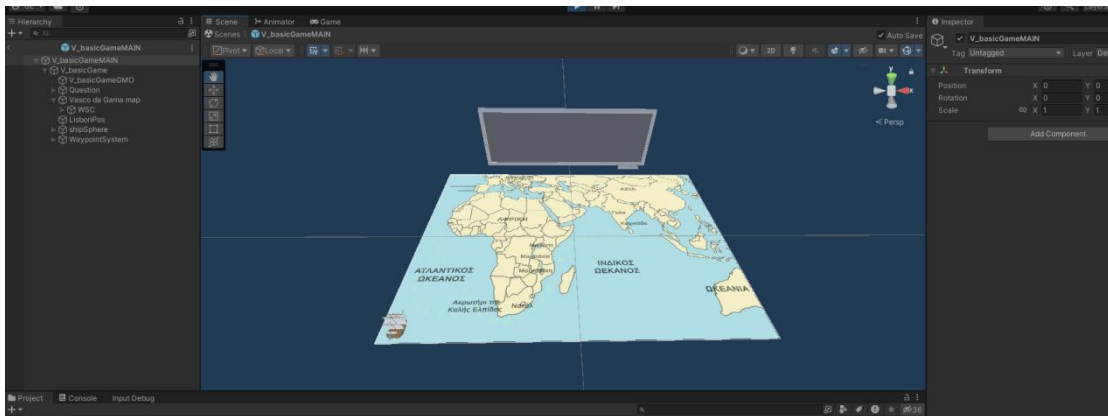
Το ξεκίνημα είναι παρόμοιο όπως και στα προηγούμενα αλλά και στα επόμενα παιχνίδια. Αρχικά εμφανίζεται ένα εισαγωγικό παράθυρο πληροφοριών με σύντομες πληροφορίες για τον Βάσκο ντα Γκάμα (Σχήμα 4.64).



Σχήμα 4.64: Το εισαγωγικό παράθυρο του Βάσκο ντα Γκάμα

## Κεφάλαιο 4°

Μόλις ο παίκτης κλείσει το εισαγωγικό παράθυρο και σκανάρει το ασπρόμαυρο πορτραίτο του Βάσκο ντα Γκάμα, εμφανίζεται στο παιχνίδι ο βασικός χάρτης με το πλοίο πάνω του (Σχήμα 4.65).



Σχήμα 4.65: Ο βασικός χάρτης του Βάσκο ντα Γκάμα

### 4.3.3.2 Η λειτουργία του παιχνιδιού

Το παιχνίδι ξεκινά με την ερώτηση που αφορά στην καταγωγή του Βάσκο ντα Γκάμα και η σωστή απάντηση είναι η «Πορτογαλία». Έτσι τοποθετείται μια πορτογαλική σημαία στην Πορτογαλία (Σχήμα 4.66).



Σχήμα 4.66: Τοποθέτηση της πορτογαλικής σημαίας στον χάρτη του Βάσκο ντα Γκάμα

Η δεύτερη ερώτηση ζητά από τον παίκτη να τοποθετήσει το πλοίο στο σημείο αφετηρίας του ταξιδιού και η απάντηση είναι η «Λισαβόνα». Έτσι, όπως προηγουμένως το πλοίο τοποθετείται στη Λισαβόνα για να ξεκινήσει το ταξίδι (Σχήμα 4.67).



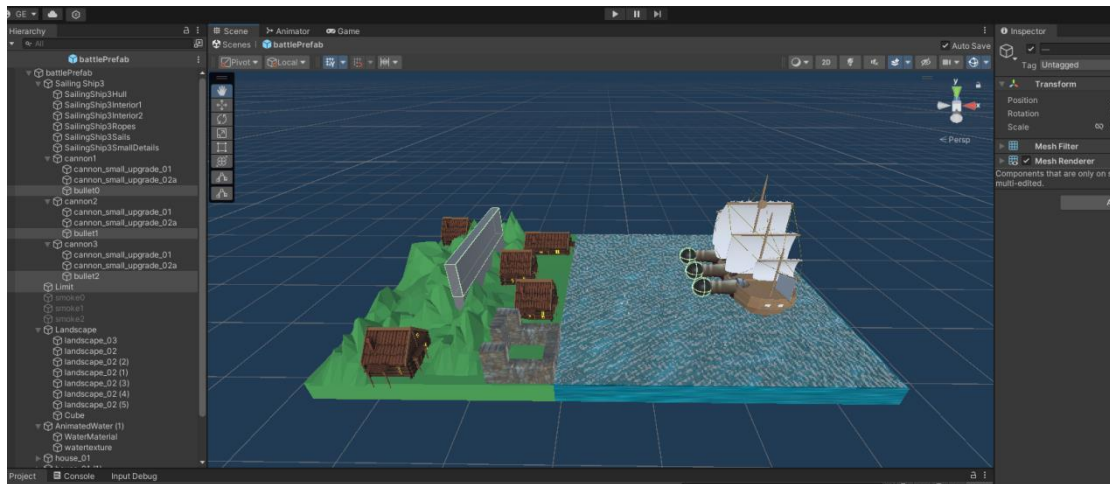
Σχήμα 4.67: Τοποθέτηση του πλοίου του Βάσκο ντα Γκάμα στη Λισαβόνα

Η τρίτη ερώτηση αφορά τον πρώτο προορισμό του ταξιδιού και η απάντηση είναι η «Μοζαμβίκη». Σύμφωνα με τα ιστορικά δεδομένα, όταν ο Βάσκο ντα Γκάμα έφτασε εκεί, βομβάρδισε τη Μοζαμβίκη. Έτσι, όταν το πλοίο φτάνει στη Μοζαμβίκη, εμφανίζεται ένα μίνι παιχνίδι με τον βομβαρδισμό της Μοζαμβίκης από τον Βάσκο ντα Γκάμα. Το μίνι παιχνίδι φαίνεται στο Σχήμα 4.68.



Σχήμα 4.68: Το μίνι παιχνίδι της Μοζαμβίκης στο παιχνίδι του Βάσκο ντα Γκάμα

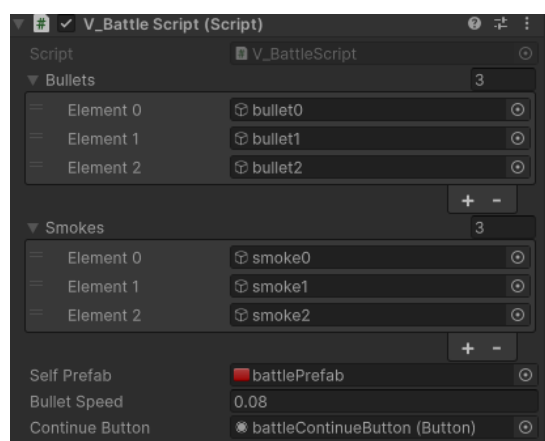
Στο Σχήμα 4.69 απεικονίζεται το μίνι παιχνίδι της Μοζαμβίκης σε μορφή prefab και στη συνέχεια θα αναλυθεί ο τρόπος λειτουργίας του.



Σχήμα 4.69: Το μίνι παιχνίδι της Μοζαμβίκης σε μορφή prefab

Υπάρχουν τρεις κανονόσφαιρες, οι bullet0, bullet1 και bullet2, οι οποίες στην αρχή είναι απενεργοποιημένες (deactivated) και ένα αντικείμενο Limit, το οποίο είναι ένας κύβος με αλλαγμένο μέγεθος – το γκρι αντικείμενο στο Σχήμα 4.69. Οι σφαίρες και το αντικείμενο Limit έχουν box collider component ενσωματωμένο πάνω τους, και επιπλέον το Limit έχει rigidbody component, ώστε να ανιχνεύονται συγκρούσεις. Αυτό θα χρειαστεί, διότι οι σφαίρες θα πρέπει να καταστρέφονται, όταν φτάσουν σε εκείνο το σημείο (για τον λόγο αυτό ονομάζεται Limit), δίνοντας την αίσθηση ότι το πλοίο βομβαρδίζει την πόλη της Μοζαμβίκης. Επίσης, για να μη φαίνεται το αντικείμενο Limit κατά τη διάρκεια του παιχνιδιού, είναι απενεργοποιημένο το component Mesh Renderer.

Στο αρχείο εντολών V\_BattleScript (το οποίο διαχειρίζεται την όλη λειτουργία της μάχης και είναι ενσωματωμένο στο αντικείμενο Limit) υπάρχει ένας πίνακας bullets, που περιέχει τις κανονόσφαιρες και επιπλέον ένας πίνακας που ονομάζεται smokes, που περιέχει τρία αντικείμενα με Particle System components ενσωματωμένα πάνω τους, τα οποία ονομάζονται smoke0, smoke1 και smoke2, όπως φαίνεται στο Σχήμα 4.70 στην καρτέλα Inspector.



Σχήμα 4.70: Το αρχείο V\_battleScript στην καρτέλα Inspector

Για να ξεκινήσει η μάχη, ο παίκτης πρέπει να πατήσει μία φορά πάνω στο κάθε κανόνι του πλοίου. Με την κίνηση αυτή ενεργοποιείται μια σφαίρα, ενώ ταυτόχρονα απενεργοποιείται το αντίστοιχο Box Collider component από το κανόνι που μόλις εκτόξευσε τη σφαίρα. Αυτό συμβαίνει, επειδή κάθε

κανόνι πρέπει να εκτοξεύει μόνο μία σφαίρα, άρα δεν πρέπει να μπορεί να «πατηθεί» από τον παίκτη παραπάνω από μία φορές. Η παραπάνω διαδικασία εκτελείται στη μέθοδο που ονομάζεται `checkForTouch()` και φαίνεται στο Σχήμα 4.71.

```
private void checkForTouch()
{
    if (Touch.activeTouches.Count != 1)
    {
        return;
    }

    Touch touch = Touch.activeTouches[0];
    if (touch.phase == TouchPhase.Began)
    {
        ray = Camera.main.ScreenPointToRay(touch.screenPosition);
        if (Physics.Raycast(ray, out hit))
        {
            switch (hit.collider.gameObject.name)
            {
                case "cannon1":
                    activateBullet0();
                    hit.collider.enabled = false;
                    break;
                case "cannon2":
                    activateBullet1();
                    hit.collider.enabled = false;
                    break;
                case "cannon3":
                    activateBullet2();
                    hit.collider.enabled = false;
                    break;
            }
        }
    }
}
```

Σχήμα 4.71: Η μέθοδος `checkForTouch()` στο μίνι παιχνίδι της Μοζαμβίκης

Η εκτόξευση κάποιας σφαίρας λειτουργεί με τον εξής τρόπο: αρχικά, όλες οι σφαίρες είναι απενεργοποιημένες, όταν όμως ο παίκτης πατήσει πάνω σε κάποιο κανόνι και ενεργοποιηθεί η αντίστοιχη σφαίρα, αυτομάτως καλείται η αντίστοιχη `moveBullet()` μέθοδος, η οποία κινεί τη σφαίρα για το χρονικό διάστημα που αυτή είναι ενεργοποιημένη στο παιχνίδι. Με τον τρόπο αυτό δημιουργείται ο μηχανισμός της εκτόξευσης. Η λειτουργία κίνησης των σφαιρών φαίνεται στο Σχήμα 4.72.

```
void Update()
{
    checkForTouch();
    moveBullet(bullets[0]);
    moveBullet(bullets[1]);
    moveBullet(bullets[2]);
    if (counter == 3)
    {
        Invoke("spawnContinueButton", 1f);
        counter++;
    }
}

private void moveBullet(GameObject bullet)
{
    if (bullet.activeSelf == true)
    {
        bullet.transform.Translate(0, 0, (bulletSpeed * Time.deltaTime), Space.Self);
    }
}
```

Σχήμα 4.72: Η λειτουργία κίνησης των σφαιρών στο μίνι παιχνίδι της Μοζαμβίκης

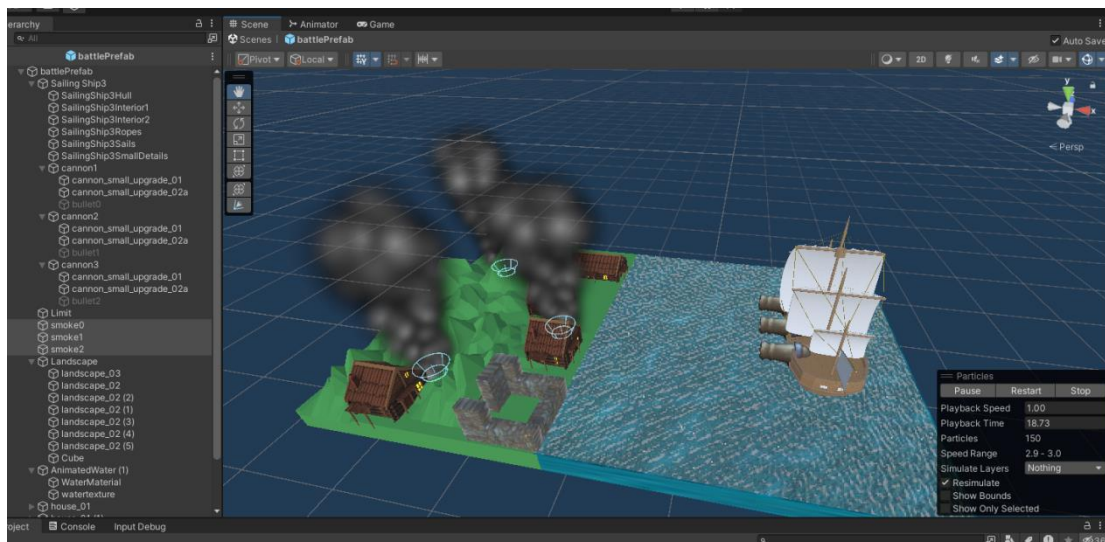
Τέλος, με τη μέθοδο `OnTriggerEnter()` (Σχήμα 4.73) ανιχνεύονται συγκρούσεις. Έτσι, όταν μια σφαίρα «συγκρούεται» με το αντικείμενο `Limit`, τότε αυτή απενεργοποιείται και ενεργοποιείται το

## Κεφάλαιο 4°

αντίστοιχο αντικείμενο smoke και καλείται η μέθοδος Play() του Particle System component, που έχει ενσωματωμένο πάνω του, ώστε να εμφανιστεί ένα εφέ καπνού μετά από τον βομβαρδισμό κάθε σφαίρας. Επιπλέον, υπάρχει ένας μετρητής counter τύπου int, ο οποίος αυξάνεται κάθε φορά που μια σφαίρα απενεργοποιείται (που σημαίνει πως η σφαίρα έχει εκτοξευθεί) και έτσι, όταν ο μετρητής counter γίνει ίσος με 3 σημαίνει πως έχουν εκτοξευθεί και οι τρεις σφαίρες και πως ο βομβαρδισμός τελείωσε. Σε αυτό το σημείο καλείται η μέθοδος spawnContinueButton(), για να εμφανιστεί το κουμπί συνέχειας παιχνιδιού στην οθόνη, όπως φαίνεται στο Σχήμα 4.72. Τα αντικείμενα smoke φαίνονται στο Σχήμα 4.74.

```
private void OnTriggerEnter(Collider other)
{
    switch (other.name)
    {
        case "bullet0":
            bullets[0].SetActive(false);
            smokes[0].SetActive(true);
            smokes[0].GetComponent<ParticleSystem>().Play();
            counter++;
            break;
        case "bullet1":
            bullets[1].SetActive(false);
            smokes[1].SetActive(true);
            smokes[1].GetComponent<ParticleSystem>().Play();
            counter++;
            break;
        case "bullet2":
            bullets[2].SetActive(false);
            smokes[2].SetActive(true);
            smokes[2].GetComponent<ParticleSystem>().Play();
            counter++;
            break;
    }
}
```

Σχήμα 4.73: Η μέθοδος OnTriggerEnter() στο μίνι παιχνίδι της Μοζαμβίκης



Σχήμα 4.74: Τα αντικείμενα smoke στο μίνι παιχνίδι της Μοζαμβίκης

Με το τέλος της μάχης εμφανίζεται ένα επιπλέον εφέ καπνού πάνω στον βασικό χάρτη στην περιοχή της Μοζαμβίκης, για να δηλώσει τον βομβαρδισμό της (Σχήμα 4.75).



Σχήμα 4.75: Το αντικείμενο smoke πάνω στον βασικό χάρτη του Βάσκο ντα Γκάμα

Η τέταρτη ερώτηση αφορά στην επίσκεψη του Βάσκο ντα Γκάμα σε μια γειτονική περιοχή, το Μαλίντι. Στο σημείο αυτό εμφανίζεται ένα μνημείο πάνω στον χάρτη, μνημείο που χτίστηκε στη μνήμη του Βάσκο ντα Γκάμα, όταν αυτός έφτασε στο Μαλίντι (Σχήμα 4.76).

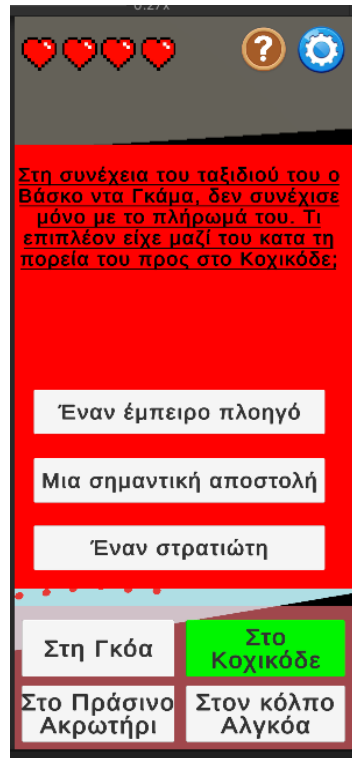


Σχήμα 4.76: Το μνημείο στη μνήμη του Βάσκο ντα Γκάμα

Η πέμπτη ερώτηση αφορά στον επόμενο προορισμό του Βάσκο ντα Γκάμα, το Κοχικόδε. Εδώ εμφανίζεται ένα μίνι παιχνίδι Quiz, που ρωτά τι πήρε μαζί του ο Βάσκο ντα Γκάμα, για να συνεχίσει το ταξίδι του (Σχήμα 4.77). Η σωστή απάντηση είναι πως πήρε μαζί του έναν έμπειρο πλοηγό. Έτσι

## Κεφάλαιο 4°

εμφανίζεται το prefab ενός πλοηγού πάνω στο πλοίο, ο οποίος ταξιδεύει με αυτό μέχρι το Κοχικόδε (Σχήμα 4.78).



Σχήμα 4.77: Το μίνι παιχνίδι Quiz στο Μαλίντι



Σχήμα 4.78: Ο πλοηγός που ταξιδεύει μαζί με το πλοίο μέχρι το Κοχικόδε

Φτάνοντας στο Κοχικόδε, ο Βάσκο ντα Γκάμα συνάντησε τον ντόπιο σουλτάνο της περιοχής Ζαμορίν και είχε έναν διάλογο μαζί του. Έτσι, εμφανίζεται ένα ταμπλό πάνω στον χάρτη, όπου υπάρχουν δύο

φιγούρες ανθρώπων να συνομιλούν μεταξύ τους, όπως φαίνεται στο Σχήμα 4.79, που είναι η πρώτη φάση του διαλόγου.



Σχήμα 4.79: Η πρώτη φάση του διαλόγου μεταξύ Βάσκο ντα Γκάμα και Ζαμορίν

Κάθε μαύρο κουτί που εμφανίζεται πάνω από κάθε φιγούρα και περιέχει ένα μικρό κείμενο διαλόγου, είναι ένα World Space Canvas. Πατώντας το κουμπί «Συνέχεια», απενεργοποιείται το canvas που είναι ενεργοποιημένο εκείνη τη στιγμή και ενεργοποιείται το άλλο canvas. Με τον τρόπο αυτό αναπαρίσταται ένας ζωντανός διάλογος, όπως φαίνεται στη δεύτερη φάση του διαλόγου στο Σχήμα 4.80.



Σχήμα 4.80: Η δεύτερη φάση του διαλόγου μεταξύ Βάσκο ντα Γκάμα και Ζαμορίν

Η διαδικασία αυτή επαναλαμβάνεται, μέχρι να τελειώσει ο διάλογος μεταξύ των δύο φιγούρων. Ο τρόπος λειτουργίας του διαλόγου φαίνεται στο Σχήμα 4.81.

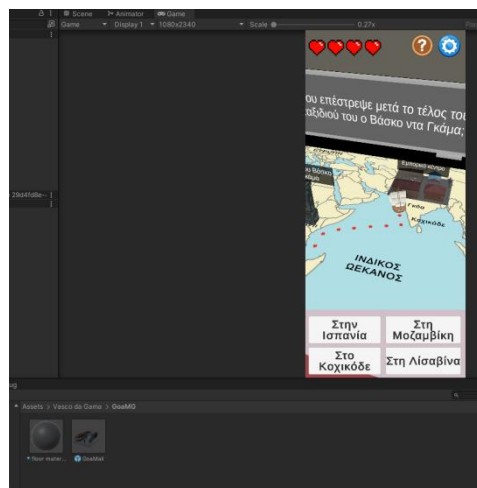
## Κεφάλαιο 4°

Ο μετρητής `dialoguePhase` είναι μια μεταβλητή, η οποία με κάθε πάτημα του κουμπιού «Συνέχεια» αυξάνεται κατά ένα. Αυτό επιτρέπει κάθε φορά στο πρόγραμμα να μπαίνει στην αντίστοιχη περίπτωση του ελέγχου `switch-case`, ώστε να εκτελέσει τις κατάλληλες εντολές. Η μέθοδος που διαχειρίζεται την παραπάνω λειτουργία, ονομάζεται `dialogueManagement()`. Στην τελευταία φάση του διαλόγου, καλείται η μέθοδος `Continue()`, για να συνεχιστεί το παιχνίδι.

```
public void dialogueManagement()
{
    switch (dialoguePhase)
    {
        case 1:
            vCanvas.SetActive(false); //αρχικά με το που πατηθεί τοκουμπι συνεχεται τη πρώτη φάση, απενεργοποιείται το dialoguebox του ζαοριν
            vCanvas.SetActive(true); //επειτα ενεργοποιείται το dialoguebox του βασκο.
            vTextbox.text = "Αναζητάς στον τόπο μας;"; //επειτα τίθεται το νέα κείμενο στο textbox του ζαοριν, για την επομενη φάση του διαλογου, που θα
            //ενα ενεργοποιηθει το dialoguebox του ζαοριν στη φάση 2.
            dialoguePhase++;
            break;
        case 2:
            vCanvas.SetActive(false); // Στη φάση 2 (αφού έχουμε πατήσει για δεύτερη φορά το κουμπι "συνεχεια"), απενεργοποιείται το κείμενο του βασκο
            vTextbox.text = "Αναζητά μπαχαρικά και χριστιανούς;"; //και τίθεται το κείμενο του βασκο με την απαντηση, για να είναι ετοιμο για τη φάση 3 που θα
            //εναενεργοποιηθει
            vCanvas.SetActive(true); //και ενεργοποιείται το κείμενο του ζαοριν, το οποίο έχει την ερωτηση που δόθηκε στη φάση 1
            dialoguePhase++;
            break;
        case 3:
            vCanvas.SetActive(false); //επειτα αφού πατηθεί για 3η φορά το κουμπι συνεχεται, απενεργοποιείται το κείμενο του ζαοριν με την ερωτηση
            vCanvas.SetActive(true); //ενεργοποιείται το κείμενο του βασκο με την απαντηση για το τι αναζητει εδω περα
            dialoguePhase++;
            break;
        case 4:
            kozCtnInfo = Instantiate(kozCtnInfoPrefab, v_mainCanvas.GetComponent<Transform>()); //εδω σε αυτη τη γραμμη, παρναμε το prefab που δημιουργουμε σε μια μεταβλητη
            //ετσι να ,μπορούμε να το διαγραφουμε μετα
            dialoguePhase++;
            break;
        case 5:
            //τελος στην φάση 4, τελειωνει ο διάλογος και συνεχίζεται το παιχνίδι κανονικα
            Continue();
            break;
    }
}
```

Σχήμα 4.81: Η μέθοδος `dialogueManagement()` στον διάλογο του Βάσκο ντα Γκάμα

Η πέμπτη ερώτηση αφορά τον τόπο αποβίβασης του Βάσκο ντα Γκάμα μετά από το Κοχικόδε, και η απάντηση είναι η «Γκόα». Σύμφωνα με τα ιστορικά γεγονότα, φτάνοντας εκεί ο Βάσκο ντα Γκάμα, έχτισε ένα εμπορικό κέντρο, οπότε αυτό εμφανίζεται πάνω στον χάρτη. Στο Σχήμα 4.82 φαίνεται το εμπορικό κέντρο πάνω στον χάρτη καθώς και ως αρχείο `prefab` στο φάκελο του Unity.



Σχήμα 4.82: Το εμπορικό κέντρο στο παιχνίδι του Βάσκο ντα Γκάμα

### 4.3.3.3 Το μίνι παιχνίδι επιλογής προϊόντων του Βάσκο ντα Γκάμα

Η τελευταία ερώτηση ρωτά πού επέστρεψε ο Βάσκο ντα Γκάμα, όταν τελείωσε το ταξίδι του και η απάντηση είναι η «Λισαβόνα». Κατά τη διάρκεια του ταξιδιού του ανακάλυψε την κανέλα και το πιπέρι. Έτσι, όταν το πλοίο επιστρέφει στη Λισαβόνα, εμφανίζεται ένα μίνι παιχνίδι με ένα ταμπλό,

πάνω στο οποίο υπάρχουν διάφορα μπαχαρικά και λαχανικά. Ο παίκτης πρέπει να πατήσει πάνω στα προϊόντα, που ανακάλυψε ο Βάσκο ντα Γκάμα. Η διαδικασία αναγνώρισης του Touch, είναι ακριβώς ίδια, όπως και στα δύο μίνι παιχνίδια drag and drop στους προηγούμενους θαλασσοπόρους, με τη διαφορά πως τα αντικείμενα εδώ απενεργοποιούνται με το πάτημά τους. Γιατί ο σκοπός εδώ δεν είναι να γεμίσει κάποιο καλάθι, άρα δεν υπάρχει κάποια διαδικασία drag ή drop στο συγκεκριμένο μίνι παιχνίδι· απλά πρέπει ο παίκτης να επιλέξει τα υλικά πατώντας πάνω τους. Σε κάθε επιλογή σωστού προϊόντος αυτό απενεργοποιείται και υπάρχει ένας μετρητής που ονομάζεται counter, ο οποίος αυξάνεται κατά ένα σε κάθε σωστή επιλογή και μόλις γίνει ίσος με 2 το παιχνίδι τελειώνει. Η μέθοδος που εκτελεί την παραπάνω διαδικασία ονομάζεται endMiniGame() και φαίνεται στο Σχήμα 4.83.

```

if (selectedObject == null) //έφόσον το επιλεγμένο αντικείμενο είναι άδειο, μπορούμε να μπει στην συγκεκριμένη if και να προχωρήσει παρακάτω
{
    if (counter <= 1) //έφόσον ο μετρητής είναι μικρότερος του 2
    {
        if (touch.phase == TouchPhase.Began)
        {
            ray = Camera.main.ScreenPointToRay(touch.screenPosition);
            if (Physics.Raycast(ray, out hit))
            {
                if (hit.collider.gameObject.CompareTag("V_endCorrect")) // αν χτυπήσαμε ανατικείμενο το οποίο έχει το σωστό tag
                {
                    selectedObject = hit.collider.gameObject; //επιλέγουμε αυτό το αντικείμενο που θα μας βοηθήσει να το απενεργοποιήσουμε μετά
                    v_sca.correctForm(); //εμφανίζεται το μήνυμα της σωστής απάντησης
                    counter++; //αυξάνεται ο μετρητής του παιχνιδιού κατά 1
                    selectedObject.SetActive(false); //απενεργοποιείται το επιλεγμένο αντικείμενο.
                }
                else if (hit.collider.gameObject.CompareTag("V_endWrong")) //διαφορετικά αν η ακτίνα έχει χτυπήσει το λανθασμένο αντικείμενο
                {
                    v_sca.wrongForm(); //δε γίνεται τίποτα άλλο παρα μόνο, εμφανίζεται το μήνυμα της λανθασμένης απάντησης.
                }
            }
        }
    }
}

if ((touch.phase == TouchPhase.Ended || touch.phase == TouchPhase.Canceled) && selectedObject != null) // εδώ πηλα στην οθόνη, αδειάζουμε τη μεταβλητή selectedObject
//επιλέγουμε νέο αντικείμενο
{
    selectedObject = null;
}

if (counter == 2)
{
    Invoke("gameEndInfo", 1f); // όταν στον εξωτερικό έλεγχο ο μετρητής γίνει πάνω από 1, σημαίνει πως έχουμε επιλέξει και τα 2 σωστά αντικείμενα
    counter++; //οποτε εμφανίζονται οι τελικές πληροφορίες του παιχνιδιού και εδω το παιχνίδι τελειώνει.
}
}

```

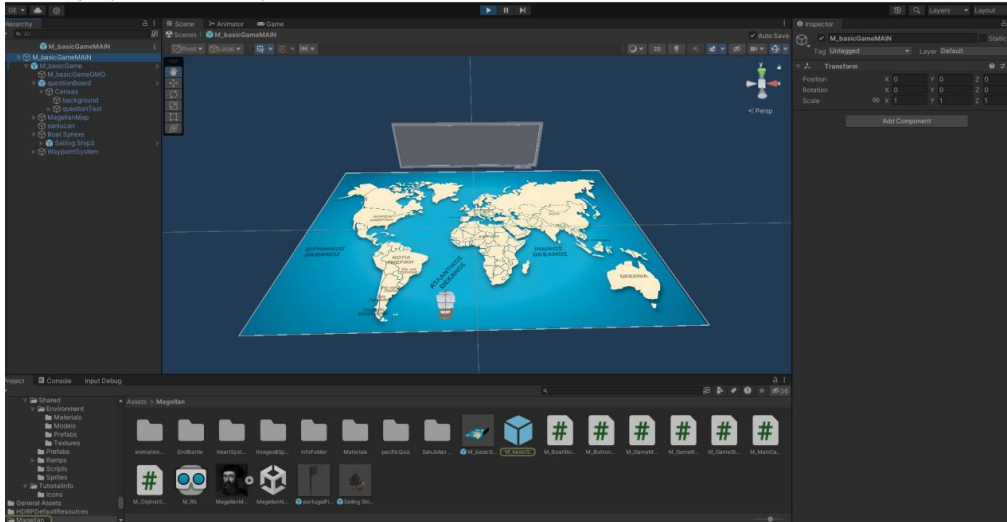
Σχήμα 4.83: Η μέθοδος endMiniGame(), στον θαλασσοπόρο Βάσκο ντα Γκάμα

Με την ολοκλήρωση του μίνι παιχνιδιού, ολοκληρώνεται το παιχνίδι του Βάσκο ντα Γκάμα, ο παίκτης κερδίζει και εμφανίζεται στην οθόνη το παράθυρο των αποτελεσμάτων.

## 4.3.4 Το παιχνίδι του Φερδινάνδου Μαγγελάνου

### 4.3.4.1 Βασικά στοιχεία του παιχνιδιού

Αρχικά, με το ξεκίνημα του παιχνιδιού, εμφανίζεται, όπως και στα προηγούμενα παιχνίδια, το εισαγωγικό παράθυρο. Με την εμφάνιση του βασικού χάρτη, εμφανίζονται και οι ζωές, τα κουμπιά των απαντήσεων του βασικού μέρους του παιχνιδιού κλπ. Στο Σχήμα 4.84, φαίνεται ο βασικός χάρτης του παιχνιδιού του Μαγγελάνου.

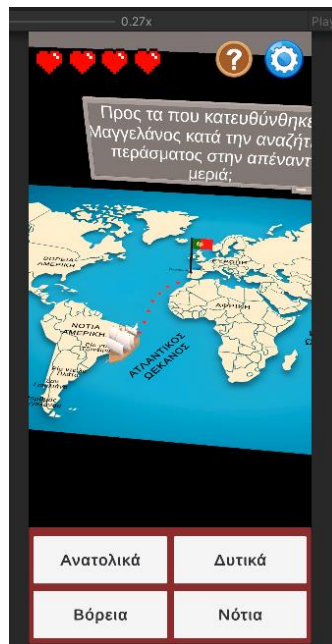


Σχήμα 4.84: Ο βασικός χάρτης και τα αρχεία του παιχνιδιού του Μαγγελάνου

#### 4.3.4.2 Η λειτουργία του παιχνιδιού

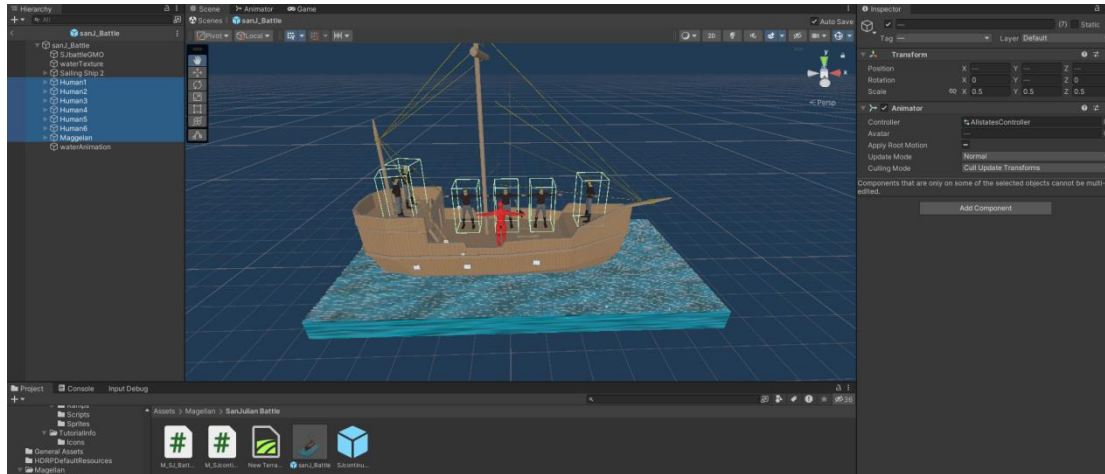
Οι πρώτες δύο ερωτήσεις αφορούν διαδοχικά την καταγωγή του Μαγγελάνου και την αφετηρία του ταξιδιού του. Με τη σωστή απάντηση και των δύο τοποθετείται αντιστοίχως μια σημαία στην Πορτογαλία και το πλοίο μεταφέρεται στην περιοχή Σανλούκαρ.

Με την τρίτη ερώτηση ξεκινά το ταξίδι του Μαγγελάνου με πρώτο προορισμό το Ρίο ντε Τζανέιρο. Έτσι το πλοίο φτάνει εκεί, όπως φαίνεται στο Σχήμα 4.85.



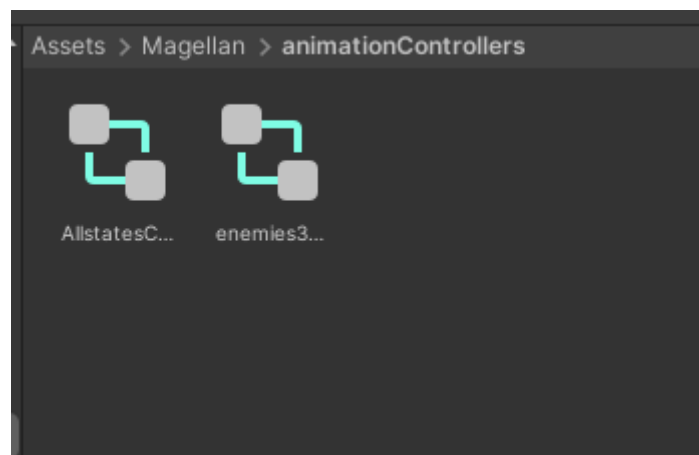
Σχήμα 4.85: Το πλοίο του Μαγγελάνου στο Ρίο ντε Τζανέιρο

Με την απάντηση της τέταρτης ερώτησης συνεχίζοντας νότια, ο Μαγγελάνος φτάνει στην περιοχή Σαν Γιουλιάνο. Σε αυτό το σημείο υπήρξαν ορισμένοι στασιαστές στο πλοίο του Μαγγελάνου με αποτέλεσμα να δημιουργηθούν συγκρούσεις στο πλήρωμα. Ο Μαγγελάνος σκότωσε σαράντα στασιαστές, ενώ σε άλλους απέδωσε χάρη. Αυτό το γεγονός απεικονίζεται στο συγκεκριμένο παιχνίδι με σύστημα κινούμενης εικόνας. Εμφανίζεται ένα μίνι παιχνίδι, στο οποίο υπάρχει ένα πλοίο με τον Μαγγελάνο και το πλήρωμά του. Ο Μαγγελάνος απεικονίζεται με την κόκκινη φιγούρα, ενώ οι υπόλοιπες φιγούρες αποτελούν το πλήρωμα, ανάμεσά τους και οι στασιαστές (Σχήμα 4.86).



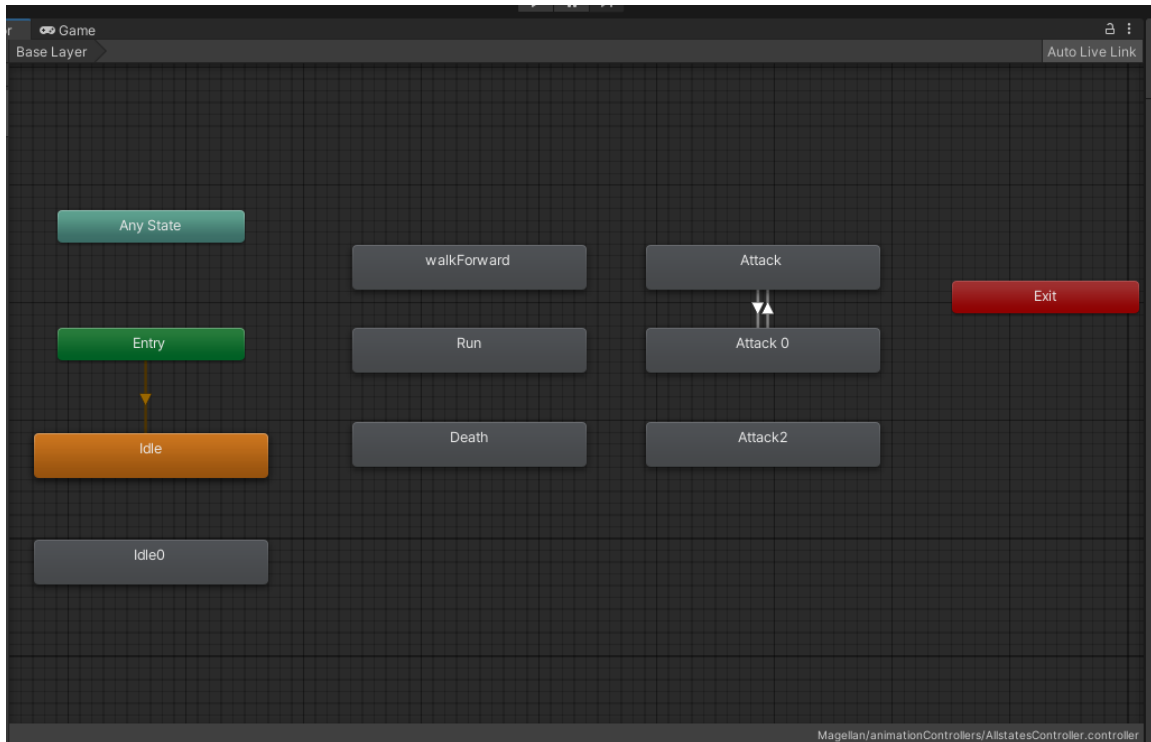
Σχήμα 4.86: Το prefab με τον Μαγγελάνο και τους στασιαστές

Όπως φαίνεται, κάθε μοντέλο έχει ένα Animator component. Το πλήρωμα και οι στασιαστές έχουν Box Collider component πάνω τους. Το Animator component έχει ένα πεδίο Controller, το οποίο δέχεται ένα Animator Controller ως είσοδο. Στο Σχήμα 4.87 φαίνονται τα Animation Controllers που χρησιμοποιούνται στο παιχνίδι του Μαγγελάνου.



Σχήμα 4.87: Τα Animation Controllers στο παιχνίδι του Μαγγελάνου

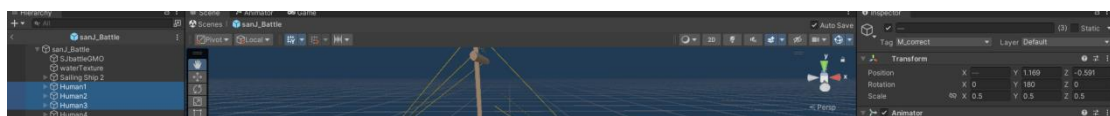
Με τη σειρά που εμφανίζονται, το πρώτο Controller ονομάζεται AllStatesController, –φαίνεται στο Σχήμα 4.88– και περιέχει όλα τα Animations: επίθεσης, θανάτου, αδράνειας, περπατήματος και τρεξίματος. Το δεύτερο Controller ονομάζεται enemies3controller και περιέχει τις ίδιες κινήσεις, οι οποίες συμβαίνουν σε διαφορετικές χρονικές στιγμές από ό, τι στο AllStatesController.



Σχήμα 4.88: Το AllStatesController

Στο μίνι παιχνίδι με τους στασιαστές χρησιμοποιείται το AllStatesController και γίνεται εναλλαγή μεταξύ των Animations του συγκεκριμένου Controller. Αρχικά, όπως φαίνεται στο Σχήμα 4.88, όλα τα μοντέλα βρίσκονται σε κατάσταση αδράνειας. Αυτό καθορίζεται από το βέλος που υπάρχει με κατεύθυνση από το πράσινο στο πορτοκαλί κουτάκι. Ουσιαστικά, κατά την είσοδο (Entry) στο παιχνίδι –δηλαδή κατά το ξεκίνημα– εφαρμόζεται το idle Animation (αδράνεια).

Για να ξεκινήσει το παιχνίδι, ο παίκτης πρέπει να πατήσει πάνω σε κάθε στασιαστή από μία φορά. Οι στασιαστές είναι οι τρεις φιγούρες που βρίσκονται απέναντι από τη φιγούρα του Μαγγελάνου και φέρουν το tag M\_correct, όπως φαίνεται στο Σχήμα 4.89, ενώ το πλήρωμα που είναι όλες οι υπόλοιπες φιγούρες φέρουν το tag M\_wrong. Το M\_correct tag δηλώνει το ποιοι από το πλήρωμα είναι οι στασιαστές και είναι οι human1, human2 και human3, ενώ το M\_wrong δηλώνει ποιοι από το πλήρωμα δεν είναι οι στασιαστές.



Σχήμα 4.89: Το tag των στασιαστών στην καρτέλα Inspector

Κάθε φορά που ο παίκτης πατάει πάνω σε κάποιον από το πλήρωμα, ελέγχεται το tag του. Αν φέρει το tag M\_correct, σημαίνει πως ο παίκτης επέλεξε έναν στασιαστή, ενώ αν φέρει το tag M\_Wrong, σημαίνει πως ο παίκτης δεν επέλεξε έναν στασιαστή. Σε κάθε περίπτωση εμφανίζεται το μήνυμα σωστής/λανθασμένης απάντησης. Στην περίπτωση σωστής απάντησης αυξάνεται ένας μετρητής counter κατά ένα, ενώ ταυτόχρονα απενεργοποιείται το Box Collider component του στασιαστή που μόλις επιλέχθηκε – αυτό συμβαίνει, διότι κάθε στασιαστής πρέπει να επιλέγεται μόνο μία φορά. Μόλις ο μετρητής γίνει ίσος με 3, σημαίνει πως ο παίκτης επέλεξε και τους τρεις στασιαστές και

Ξεκινάει η επίθεση του Μαγγελάνου. Η όλη διαδικασία φαίνεται στη μέθοδο checkForTouch() στην εξουδετέρωση των στασιαστών από τον Μαγγελάνο στο Σχήμα 4.90.

```
private void checkForTouch()
{
    if (Touch.activeTouches.Count != 1)
    {
        return;
    }

    Touch touch = Touch.activeTouches[0];

    if (touch.phase == TouchPhase.Began)
    {
        Ray ray = Camera.main.ScreenPointToRay(touch.screenPosition);
        if (Physics.Raycast(ray, out hit))
        {
            if (hit.collider.gameObject.CompareTag("M_correct"))
            {
                m_mcm.correctPopup();
                counter++;
                hit.collider.enabled = false;
            }
            else if (hit.collider.gameObject.CompareTag("M_wrong"))
            {
                m_mcm.wrongPopup();
            }
        }
    }
}
```

Σχήμα 4.90: Η μέθοδος checkForTouch() στην εξουδετέρωση των στασιαστών

Ο έλεγχος για το πότε ο μετρητής θα γίνει ίσος με 3 καθώς και το ξεκίνημα της επίθεσης, διαχειρίζονται από τη μέθοδο Update(), όπως φαίνεται στο Σχήμα 4.91.

```
void Update()
{
    checkForTouch();
    if (counter == 3) // όταν ο μετρητής γίνει 3, ξεκινάει η επίθεση
    {
        Invoke("attack1Movement", 0.2f);
        counter++;
    }

    changeAnimations(); //εδώ ελεγχεται, το ποτε μια μεταβλητη cxDied θα γίνει true.
                        //cxDied εγινε true

    if (battleEnded == true) //οταν τελειωση η μαχη, εμφανιζεται το κουμπι συνεχειας
    {
        Instantiate(continuePanel, M_mainCanvasT);
        battleEnded = false;
    }
}
```

Σχήμα 4.91: Η μέθοδος Update() στην εξουδετέρωση των στασιαστών από τον Μαγγελάνο

Αρχικά καλείται η μέθοδος attack1Movement(). Όλες οι διαδικασίες που χρησιμοποιούνται στην attack1Movement() φαίνονται στο Σχήμα 4.92. Με την κλήση της μεθόδου attack1Movement() ο Μαγγελάνος ακολουθεί μια συγκεκριμένη διαδρομή, για να φτάσει μπροστά στον πρώτο στασιαστή και ξεκινάει η επίθεση. Στο Σχήμα 4.93 ο Μαγγελάνος επιτίθεται στον πρώτο στασιαστή.

```
// οι κινήσεις των επιθέσεων
private void attack1Movement()
{
    Invoke("magellanRotateLeft", 1.1f);
    Invoke("MagellanWalkAnimation", 1.11f);
    Invoke("magellanRotateRight", 1.8f);
    Invoke("MagellanAttackAnimation", 2.1f);
    Invoke("character1Death", 2.5f);

    Invoke("attack2Movement", 3.5f); //επειτα μετα απο μισο δευτερολεπτο, καλειται η κινηση της επιθεσης2, ωστε ο να ξεκινησει η δευτερη επιθεση
    //του μαγγελανου. δουλευει με παρομοιο τροπο οπως η συγκεκριμενη μεθοδος.
}
```

Σχήμα 4.92: Η μέθοδος attack1Movement()



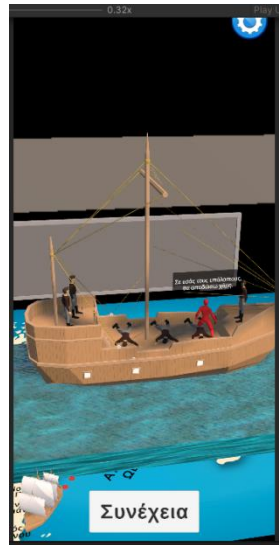
Σχήμα 4.93: Η επίθεση του Μαγγελάνου στον πρώτο στασιαστή

Κατόπιν καλείται η μέθοδος `character1Death()`, η οποία αλλάζει το `Animation` του πρώτου στασιαστή σε `Death Animation`, που τον αναπαριστά να πεθαίνει (Σχήμα 4.94).



Σχήμα 4.94: Η εξουδετέρωση του πρώτου στασιαστή

Στη συνέχεια καλούνται οι μέθοδοι `attack2Movement()` και `attack3Movement()` –οι οποίες λειτουργούν με παρόμοιο τρόπο όπως η `attack1Movement()`–, για να συνεχιστεί η εξουδετέρωση. Στο Σχήμα 4.95 ο Μαγγελάνος έχει εξουδετερώσει και τους τρεις στασιαστές.



Σχήμα 4.95: Ο Μαγγελάνος, έχει εξοντώσει και τους τρεις στασιαστές

Μόλις τελειώσει η εξουδετέρωση, εμφανίζεται ένα World Space Canvas με το κείμενο διαλόγου «Σε εσάς τους υπόλοιπους θα αποδώσω χάρη», καθώς ο Μαγγελάνος απέδωσε χάρη ζωής στο υπόλοιπο πλήρωμά του. Τέλος, εμφανίζεται το κουμπάκι «Συνέχεια», για να συνεχιστεί το ταξίδι του Μαγγελάνου. Η διαδικασία που αλλάζει τα Animations των στασιαστών σε Death Animation, φαίνεται στο Σχήμα 4.96 και περιέχεται μέσα στη μέθοδο changeAnimations().

```
//οι μεθοδοι που αλλαζουν τη περιστροφή του μαγγελανου (θέλωμε να καλούνται σε invoke γιατις υπάρχουν οι μεθοδοι)
private void magellanRotateLeft()
{
    magellan.transform.Rotate(0, -90, 0);
}
private void magellanRotateRight()
{
    magellan.transform.Rotate(0, 90, 0);
}
//οι μεθοδοι που αλλαζουν τα animators του Μαγγελανου (θέλωμε να καλούνται σε invoke)
private void MagellanAttackAnimation()
{
    magellanAnimator.Play("Attack");
}
private void MagellanWalkAnimation()
{
    magellanAnimator.Play("walkforward");
}
private void changeAnimations() //είναι οι animators των χαρακτρων που σκοτώνονται. Όταν η μεταβλητη cDied, γίνεται true, τότε αλλάζει το animation του χαρακτρα
{
    if (cDied == true)
    {
        animators[0].Play("Death");
        cDied = false;
    }
    else if (cDied == true)
    {
        animators[1].Play("Death");
        cDied = false;
    }
    else if (cDied == true)
    {
        animators[2].Play("Death");
        Invoke("activateMagellanDialogue", 2.2f);
        cDied = false;
    }
}
```

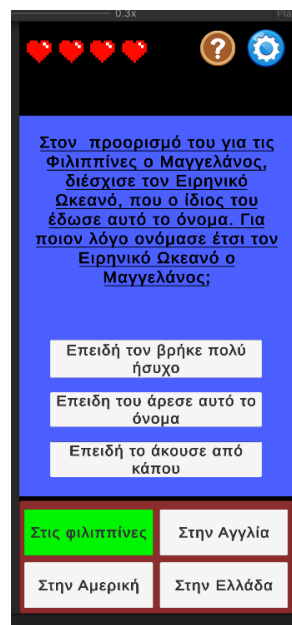
Σχήμα 4.96: Η μέθοδος changeAnimation() στο μίνι παιχνίδι με τους στασιαστές

Η πέμπτη ερώτηση ζητά από τον παίκτη να απαντήσει πού έφτασε ο Μαγγελάνος, αφού αναχώρησε από το Σαν Γιουλιάνο και η απάντηση είναι στον πορθμό του Μαγγελάνου, ένας πορθμός στον οποίον ο ίδιος έδωσε το όνομά του. Έτσι, το πλοίο φτάνει στον πορθμό του Μαγγελάνου, όπως φαίνεται στο Σχήμα 4.97.



Σχήμα 4.97: Ο Μαγγελάνος φτάνει στον πορθμό, στον οποίο ο ίδιος έδωσε το όνομά του

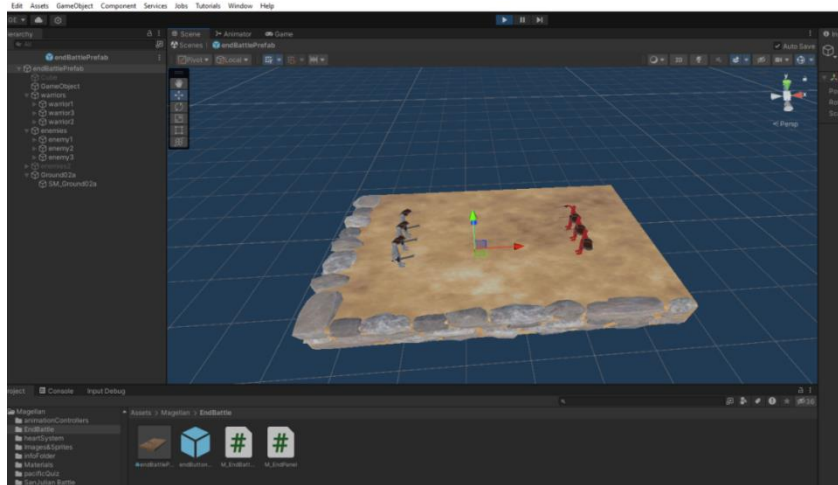
Η τελευταία ερώτηση αναφέρεται στη χώρα που αποβιβάστηκε στη συνέχεια ο Μαγγελάνος και η σωστή απάντηση είναι οι «Φιλιππίνες». Κατά το ταξίδι του προς τις Φιλιππίνες, ο Μαγγελάνος διέσχισε τον Ειρηνικό ωκεανό. Εδώ εμφανίζεται ένα μίνι παιχνίδι Quiz, το οποίο ρωτά γιατί ο Μαγγελάνος ονόμασε έτσι τον συγκεκριμένο ωκεανό, όπως φαίνεται στο Σχήμα 4.98 (ο Ειρηνικός ωκεανός πήρε το όνομά του από τον ίδιο τον Μαγγελάνο), και η απάντηση είναι «επειδή τον βρήκε πολύ ήσυχο».



Σχήμα 4.98: Το Quiz κατά τη διάσχιση του Ειρηνικού ωκεανού από τον Μαγγελάνο

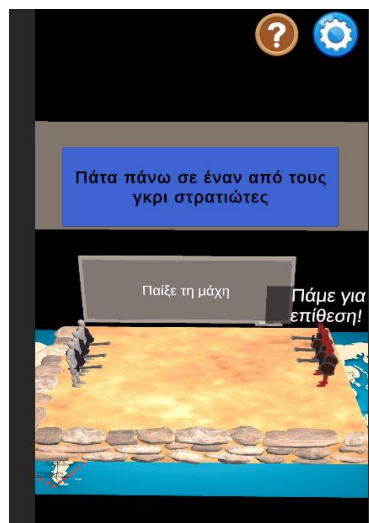
### 4.3.4.3 Το μίνι παιχνίδι της μάχης του Μαγγελάνου στις Φιλιππίνες

Τέλος, στις Φιλιππίνες ακολούθησε μια μάχη με τοπικές φυλές, στην οποία συμμετείχε και ο Μαγγελάνος, όπου και σκοτώθηκε. Έτσι, εμφανίζεται ένα πεδίο μάχης με δύο στρατιές (Σχήμα 4.99) και ο παίκτης πρέπει να παίξει τη μάχη. Οι κόκκινες φιγούρες αναπαριστούν τους στρατιώτες του Μαγγελάνου, ενώ οι γκρι τους αντιπάλους τους.



Σχήμα 4.99: Το prefab της μάχης του Μαγγελάνου στις Φιλιππίνες

Στην πρώτη φάση της μάχης εμφανίζεται ένα παράθυρο διαλόγου πάνω από την κόκκινη πλευρά, με το κείμενο «Πάμε για επίθεση». Σε αυτό το σημείο ο παίκτης πρέπει να πατήσει πάνω σε έναν από τους γκρι στρατιώτες, ώστε οι κόκκινοι να ξεκινήσουν να ορμούν κατά πάνω τους. Υπενθυμίζεται πως σε κάθε περίπτωση ο παίκτης μπορεί να πατήσει το κουμπί με το αγγλικό ερωτηματικό, για να εμφανιστεί στην οθόνη κείμενο οδηγίας για το τι πρέπει να κάνει, όπως φαίνεται στο Σχήμα 4.100 στην πρώτη φάση της μάχης.

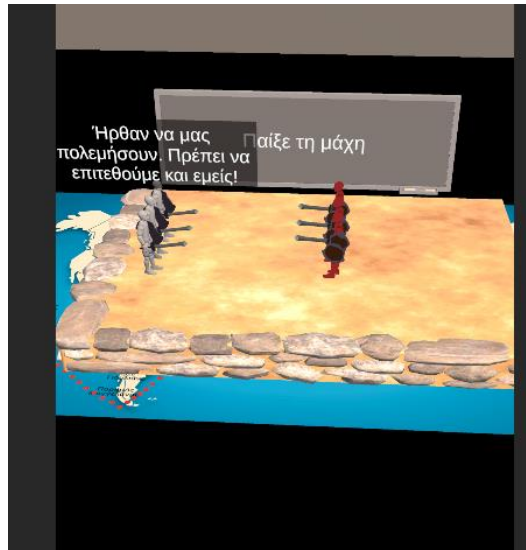


Σχήμα 4.100: Η πρώτη φάση της μάχης του Μαγγελάνου στις Φιλιππίνες

Στη συνέχεια οι κόκκινοι στρατιώτες ορμούν προς τους γκρι για να τους επιτεθούν. Εδώ γίνεται μια παύση στην κίνηση των κόκκινων στρατιωτών, καθώς είναι η σειρά των γκρι να αντεπιτεθούν. Έτσι,

## Κεφάλαιο 4°

στη δεύτερη φάση εμφανίζεται ένα παράθυρο διαλόγου πάνω από τους γκρι στρατιώτες με το μήνυμα πως οι ίδιοι πρέπει να αντεπιτεθούν (Σχήμα 4.101).



Σχήμα 4.101: Η δεύτερη φάση της μάχης του Μαγγελάνου στις Φιλιππίνες

Ο παίκτης τώρα πρέπει να πατήσει πάνω σε έναν από τους κόκκινους στρατιώτες, ώστε οι γκρι να ορμήσουν εναντίον τους και να ξεκινήσει η επίθεση και των δύο πλευρών, η οποία αποτελεί την τρίτη φάση της μάχης, και η οποία φαίνεται στο Σχήμα 4.102.



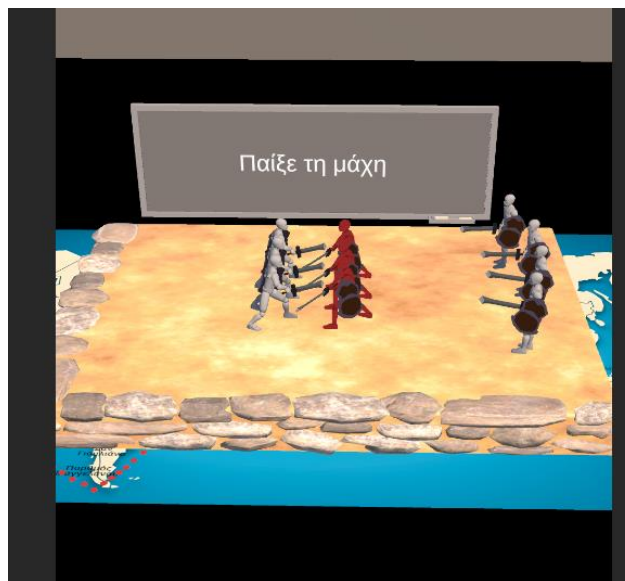
Σχήμα 4.102: Η τρίτη φάση της μάχης του Μαγγελάνου στις Φιλιππίνες

Έπειτα από ένα μικρό χρονικό διάστημα, στην τέταρτη φάση της μάχης, εμφανίζεται ένα παράθυρο διαλόγου πάνω από τους γκρι στρατιώτες με το μήνυμα πως πρέπει να περικυκλώσουν τους αντιπάλους τους. Επιπλέον, εμφανίζεται και μία ακόμη στρατιά από γκρι στρατιώτες και ένα γκρι πλαίσιο πίσω από τους κόκκινους (Σχήμα 4.103).



Σχήμα 4.103: Η τέταρτη φάση της μάχης του Μαγγελάνου στις Φιλιπίνες

Σε αυτό το σημείο ο παίκτης πρέπει να τοποθετήσει τη δεύτερη στρατιά από τους γκρι στρατιώτες στην γκρι περιοχή πίσω από τους κόκκινους. Αυτό πραγματοποιείται με διαδικασία drag and drop. Στο Σχήμα 4.104 φαίνεται η δεύτερη στρατιά που έχει περικυκλώσει τον στρατό του Μαγγελάνου.



Σχήμα 4.104: Ο στρατός του Μαγγελάνου έχει περικυκλωθεί από τους αντιπάλους του. Κατόπιν ξεκινάει η πέμπτη και τελευταία φάση της μάχης, όπου η δεύτερη στρατιά που περικύκλωσε τον Μαγγελάνο, ξεκινά να επιτίθεται (Σχήμα 4.105).



Σχήμα 4.105: Η πέμπτη φάση της μάχης του Μαγγελάνου στις Φιλιππίνες

Τέλος, ο στρατός του Μαγγελάνου ηττάται και ο ίδιος σκοτώνεται. Σε αυτό το σημείο τελειώνει η μάχη και εμφανίζεται το κουμπί συνέχειας (Σχήμα 4.106).



Σχήμα 4.106: Το τέλος της μάχης του Μαγγελάνου

Στο σημείο αυτό θα παρουσιαστούν ορισμένα από τα βασικά σημεία λειτουργίας της μάχης. Αρχικά, αφού έχει εμφανιστεί το παράθυρο με το μήνυμα επίθεσης της κόκκινης πλευράς και ο παίκτης πατήσει πάνω στους γκρι, ελέγχεται το tag τους, ώστε να γίνει σίγουρο πως ο παίκτης πάτησε πάνω στη σωστή πλευρά. Στη συνέχεια ξεκινά το Animation των στρατιωτών της κόκκινης πλευράς και ταυτόχρονα αυξάνεται ένας μετρητής battlePhase –ο οποίος έχει αρχική τιμή 1– κατά ένα και γίνεται ίσος με 2. Αυτή η μεταβλητή θα χρειαστεί για τον έλεγχο if, που υπάρχει στη μέθοδο checkForTouch(). Έπειτα, αφού η μεταβλητή battlePhase έχει γίνει ίση με 2, με το δεύτερο πάτημα του παίκτη πάνω στους κόκκινους στρατιώτες (για να ξεκινήσει η αντεπίθεση των γκρι), ξεκινάει η μάχη. Στη συνέχεια αυξάνεται ο μετρητής battlePhase κατά ένα για την επόμενη φάση της μάχης.

Μετά από μικρό χρονικό διάστημα δημιουργείται η δεύτερη ομάδα των γκρι στρατιωτών και εμφανίζεται το γκρι πλαίσιο πίσω από τους κόκκινους, στο οποίο ο παίκτης πρέπει να τοποθετήσει τη δεύτερη ομάδα γκρι στρατιωτών. Όλη αυτή η διαδικασία φαίνεται στο Σχήμα 4.107 στο οποίο φαίνεται ένα Τμήμα της μεθόδου checkForTouch().

```

if (touch.phase == TouchPhase.Began)
{
    Ray ray = Camera.main.ScreenPointToRay(touch.screenPosition);
    if (Physics.Raycast(ray, out hit))
    {
        switch (battlePhase)
        {
            case 1:
                if (hit.collider.CompareTag("enemies1"))
                {
                    canvas1.SetActive(false);
                    StartCoroutine(setAnimation(warriorAnimators, "Run", 0.1f));
                    StartCoroutine(setAnimation(warriorAnimators, "Idle", 1.50f));
                    StartCoroutine(enableObject(canvas2, true, 1.4f));
                    m_ais.guideText.text = "Πάτα σε έναν από τους κόκκινους στρατιώτες για να ξεκινήσεις τη μάχη";
                    battlePhase++;
                }
                break;
            case 2:
                if (hit.collider.CompareTag("warriors"))
                {
                    canvas2.SetActive(false);
                    text2.text = "Πρέπει να τους περικυλώσουμε!";
                    StartCoroutine(setAnimation(enemies1Animators, "Run", 0.1f));
                    StartCoroutine(setAnimation(warriorAnimators, "Attack", 1.1f));
                    StartCoroutine(setAnimation(enemies1Animators, "Attack", 1.1f));
                    StartCoroutine(enableObject(canvas2, true, 2.4f));
                    StartCoroutine(InstantiateEnemies2(4f));
                    StartCoroutine(enableObject(correctArea, true, 4f));
                    m_ais.guideText.text = "Τοποθέτησε την δεύτερη ομάδα γκρι στρατιωτών, στην περιοχή που αναβοβλίνει, πίσω από τους κόκκινους στρατιώτες";
                    battlePhase++;
                }
                break;
        }
    }
}

```

Σχήμα 4.107: Η λειτουργία της μεθόδου checkForTouch() στη μάχη του Μαγγελάνου

Στη συνέχεια, αφού η μεταβλητή battlePhase έχει γίνει ίση με 3, ο παίκτης τοποθετεί το δεύτερο κύμα των γκρι πίσω από τους κόκκινους στρατιώτες, ξεκινάει η επίθεση και της δεύτερης ομάδας των γκρι.

Τέλος, περνώντας στη τελευταία φάση, καλείται διαδοχικά η μέθοδος που αλλάζει το Animation κάθε κόκκινου στρατιώτη σε Death Animation, ώστε να σηματοδοτηθεί η ήττα των στρατιωτών του Μαγγελάνου. Τέλος, αφού τελειώσει η μάχη εμφανίζεται το τελικό παράθυρο πληροφοριών για τον Μαγγελάνο και στη συνέχεια το παιχνίδι τελειώνει. Οι μέθοδοι που αλλάζουν τα Animations των κόκκινων από σε Death Animation φαίνονται στο Σχήμα 4.108.

```

private void warrior1Death()
{
    warriorAnimators[0].Play("Death");
    Invoke("warrior2Death", 2.2f); //αφου εχει πεθανει ο warrior, μετα απο 2.2 δευτερολεπτο καλ
    // //όλα γινονται, οποτε και με τον θανατο του πρωτου warrior
}

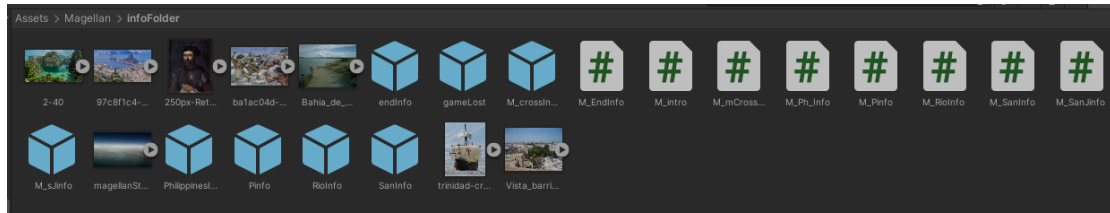
private void warrior2Death()
{
    warriorAnimators[1].Play("Death");
    Invoke("warrior3Death", 2.2f); //και αφου πεθανει και ο δευτερος warrior, τοτε καλειται η με
}

private void warrior3Death()
{
    StartCoroutine(setAnimation(enemies1Animators, "Idle0", 2.5f));
    warriorAnimators[2].Play("Death");
    Invoke("spawnEndPanel", 2.7f); //και τελος αφου πεθανει και ο 3ος warrior, τοτε οι εχθροι αν
    //δημιουργειται το κουμπι συνεχειας, το οποιο δημιουργει το ε
}

```

Σχήμα 4.108: Οι μέθοδοι που αλλάζουν τα Animations των κόκκινων στρατιωτών

Σε αυτό το σημείο ολοκληρώνεται το παιχνίδι του Μαγγελάνου και ο παίκτης έχει κερδίσει. Στο Σχήμα 4.109 φαίνεται ο φάκελος με τα αρχεία των πληροφοριών του Μαγγελάνου.

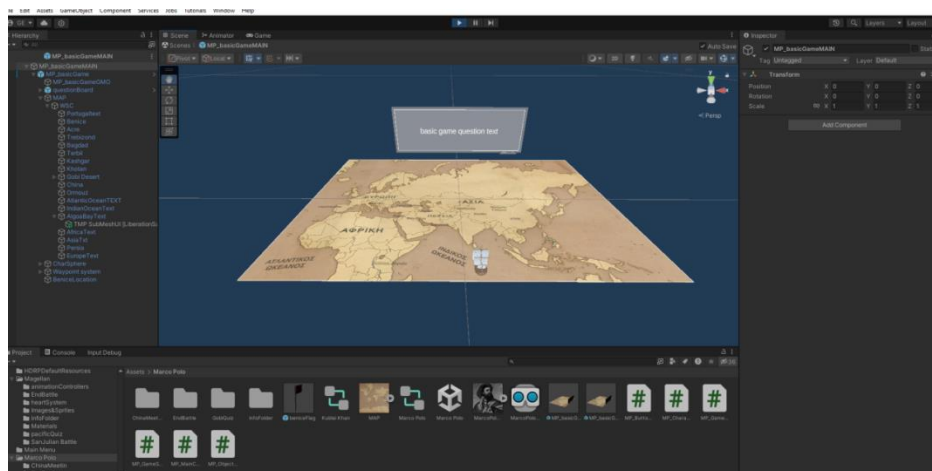


Σχήμα 4.109: Ο φάκελος με τα αρχεία των πληροφοριών του Μαγγελάνου

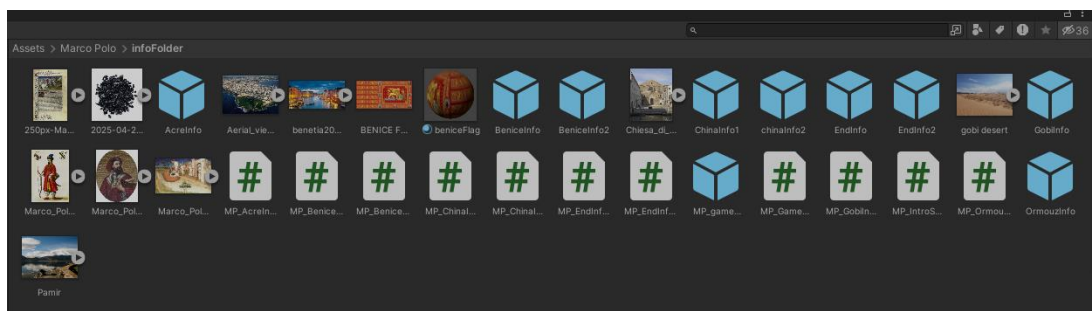
## 4.3.5 Το παιχνίδι του Μάρκο Πόλο

### 4.3.5.1 Βασικά στοιχεία του παιχνιδιού

Όπως και στα προηγούμενα παιχνίδια, έτσι και στο τελευταίο, εμφανίζονται αρχικά το εισαγωγικό παράθυρο και με το κλείσιμό του και την εμφάνιση του βασικού χάρτη, εμφανίζονται και τα διάφορα κουμπιά που θα χρειαστεί ο παίκτης για να παίξει το παιχνίδι. Στο Σχήμα 4.110 φαίνεται ο βασικός χάρτης του Μάρκο Πόλο και στο Σχήμα 4.111, ο φάκελος με τα αρχεία για τις πληροφορίες που θα εμφανίζονται στο ταξίδι του Μάρκο Πόλο.



Σχήμα 4.110: Ο βασικός χάρτης του Μάρκο Πόλο



Σχήμα 4.111: Ο φάκελος με τις πληροφορίες του Μάρκο Πόλο

### 4.3.5.2 Η λειτουργία του παιχνιδιού

Οι πρώτες δύο ερωτήσεις, αφορούν την καταγωγή του Μάρκο Πόλο και την αφετηρία του ταξιδιού του. Με τη σωστή απάντηση αυτών, αρχικά τοποθετείται μια σημαία της Βενετίας στον χάρτη και στη συνέχεια μετακινείται το πλοίο στη Βενετία, καθώς και στις δύο ερωτήσεις η σωστή απάντηση είναι η «Βενετία».

Η τρίτη ερώτηση αφορά στον πρώτο προορισμό του Μάρκο Πόλο και η σωστή απάντηση είναι η «Περσία». Έτσι, το πλοίο ξεκινάει το ταξίδι φτάνοντας στην περιοχή Άκρα της Περσίας (Σχήμα 4.112).



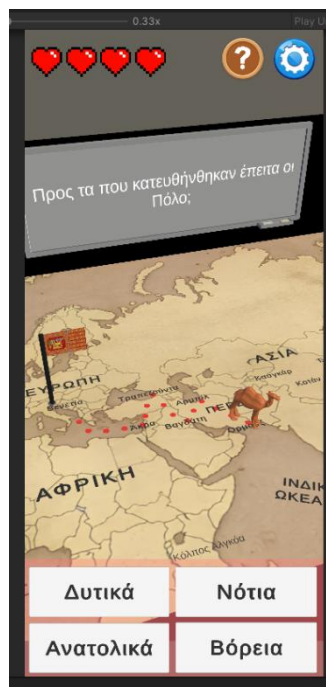
Σχήμα 4.112: Ο Μάρκο Πόλο φτάνει στην Περσία

Η τέταρτη ερώτηση αφορά στη συνέχεια του ταξιδιού που είναι ο προορισμός «Ορμούζ». Σύμφωνα με τα ιστορικά δεδομένα, επειδή δεν υπήρχε θαλάσσιος δρόμος, ο Μάρκο Πόλο διέσχισε την Περσία με καμήλες. Σε αυτό το σημείο εμφανίζεται ένα μίνι παιχνίδι, το οποίο είναι ένα ταμπλό που έχει πάνω του τρία διαφορετικά μεταφορικά μέσα της εποχής, όπως φαίνεται στο Σχήμα 4.113.



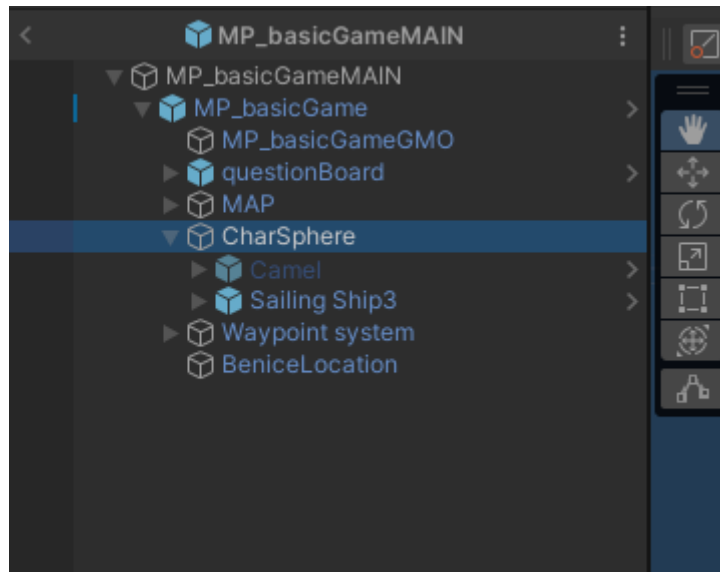
Σχήμα 4.113: Το μίνι παιχνίδι της τέταρτης ερώτησης του Μάρκο Πόλο

Ο παίκτης πρέπει να επιλέξει το σωστό μεταφορικό μέσο, με το οποίο ο Μάρκο Πόλο συνέχισε το ταξίδι του, και αφού απαντήσει σωστά, απενεργοποιείται το πλοίο και εμφανίζεται το μοντέλο μιας καμήλας, το οποίο θα κινείται πάνω στον χάρτη το διάστημα που ο Μάρκο Πόλο ταξίδευε δια ξηράς. Στο Σχήμα 4.114 φαίνεται ο Μάρκο Πόλο που έχει φτάσει στο Ορμούζ δια ξηράς.



Σχήμα 4.114: Ο Μάρκο Πόλο φτάνει στο Ορμούζ δια ξηράς

Η λειτουργία που απενεργοποιεί το μοντέλο του πλοίου και ενεργοποιεί το μοντέλο της καμήλας περιγράφεται στη συνέχεια. Εξαρχής υπάρχουν τα μοντέλα του πλοίου και της καμήλας ως «παιδιά» του charSphere στην ιεραρχία του Unity. Επειδή ο Μάρκο Πόλο ξεκινά το ταξίδι του με πλοίο μέσω θάλασσας, αρχικά είναι απενεργοποιημένο το μοντέλο της καμήλας και ενεργοποιημένο το μοντέλο του πλοίου (Σχήμα 4.115). Σε όλη τη διάρκεια του ταξιδιού και τα δύο μοντέλα βρίσκονται στην ίδια θέση, που σημαίνει πως ταξιδεύουν μαζί.



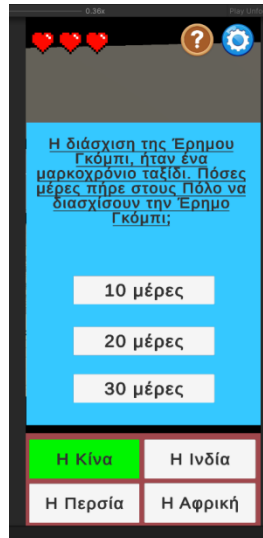
Σχήμα 4.115: Το Game Object CharSphere του Μάρκο Πόλο

Έτσι, μόλις ο παίκτης επιλέξει την καμήλα στο μίνι παιχνίδι, απενεργοποιείται το μοντέλο του πλοίου και ενεργοποιείται το μοντέλο της καμήλας. Αυτό επιτυγχάνεται με την κλήση της μεθόδου `activateCamel()`, που φαίνεται στο Σχήμα 4.116.

```
public void activateCamel()
{
    ship.SetActive(false);
    camel.SetActive(true);
}
```

Σχήμα 4.116: Η μέθοδος `activateCamel()` στο παιχνίδι του Μάρκο Πόλο

Η πέμπτη ερώτηση αφορά την κατεύθυνση προς την οποία κινήθηκε ο Μάρκο Πόλο μετά από το Ορμούζ και η σωστή απάντηση είναι πως κατευθύνθηκε ανατολικά και έφτασε στην έρημο Γκόμπι. Η έκτη ερώτηση αφορά στον τελευταίο προορισμό του ταξιδιού του Μάρκο Πόλο και η σωστή απάντηση είναι η «Κίνα». Καθώς η έρημος Γκόμπι είναι πολύ μεγάλη, χρειάστηκαν περίπου 30 μέρες για να τη διασχίσει ο Μάρκο Πόλο. Έτσι, εμφανίζεται ένα μίνι παιχνίδι Quiz που ρωτάει πόσο διάστημα διήρκεσε το πέρασμα του Μάρκο Πόλο από την έρημο Γκόμπι, όπως φαίνεται στο Σχήμα 4.117.



Σχήμα 4.117: Το μίνι παιχνίδι Quiz της ερήμου Γκόμπι

Με την απάντηση στο Quiz το πλοίο του Μάρκο Πόλο φτάνει στην Κίνα. Εκεί συναντά τον μεγάλο κινέζο αυτοκράτορα Κουμπλάι Χαν, με τον οποίο έχει έναν σύντομο διάλογο. Έτσι, εμφανίζονται οι φιγούρες δύο ανθρώπων να συνομιλούν (Σχήμα 4.118) – η λειτουργία του διαλόγου είναι όμοια με αυτήν στο παιχνίδι του Βάσκο ντα Γκάμα.



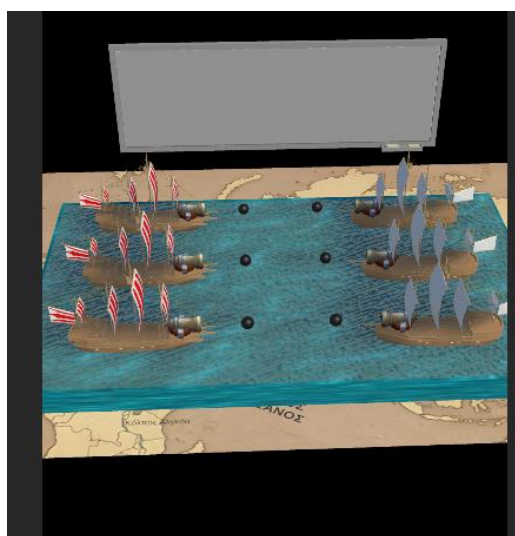
Σχήμα 4.118: Ο διάλογος μεταξύ του Μάρκο Πόλο και του Κουμπλάι Χαν

Στη συνέχεια και αφού τελειώσει ο διάλογος, ο Κουμπλάι Χαν κάνει μια ξενάγηση στον Μάρκο Πόλο και ο τελευταίος εντυπωσιάζεται από το πόσο ανεπτυγμένη ήταν η Κίνα εκείνη την εποχή. Έτσι, εμφανίζεται πάνω στον χάρτη ένα prefab μιας ανεπτυγμένης πόλης με εξελιγμένα κτίρια (Σχήμα 4.119).

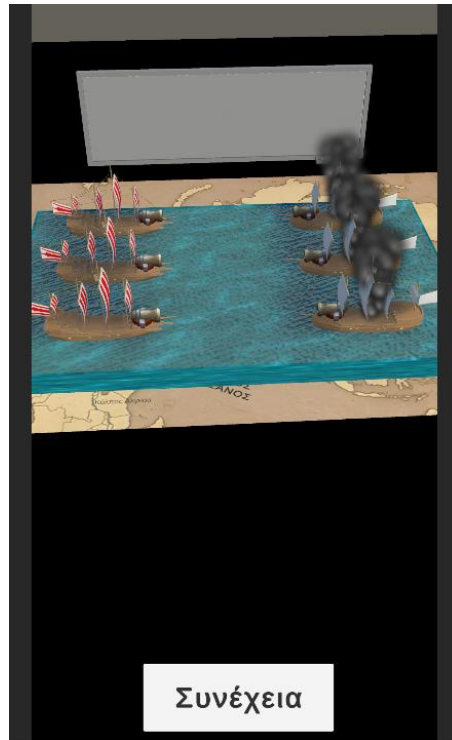


Σχήμα 4.119: Το prefab της ανεπτυγμένης πόλης της Κίνας

Τέλος, η τελευταία ερώτηση αφορά στην επιστροφή του Μάρκο Πόλο με την ολοκλήρωση του ταξιδιού του. Η σωστή απάντηση είναι η «Βενετία». Σύμφωνα με τα ιστορικά δεδομένα, όταν ο Μάρκο Πόλο επέστρεψε στη Βενετία, η πόλη βρισκόταν σε πόλεμο με τη Γένοβα. Ο Μάρκο Πόλο κλήθηκε να πολεμήσει και σε μια ναυμαχία η Βενετία ηττήθηκε · ο Μάρκο Πόλο δεν σκοτώθηκε, όμως, πιάστηκε αιχμάλωτος για μερικά χρόνια και ουσιαστικά εδώ τελειώνει και το ταξίδι του. Έτσι, εμφανίζεται πάνω στον χάρτη ένα prefab με δύο στόλους από πολεμικά πλοία να πολεμούν μεταξύ τους. Η ναυμαχία λειτουργεί με τον ίδιο τρόπο, όπως ο βομβαρδισμός της Μοζαμβίκης στο παιχνίδι του Βάσκο ντα Γκάμα. Αρχικά στην πρώτη φάση της ναυμαχίας οι δύο στόλοι ανταλλάσσουν πυρά, όπως φαίνεται στο Σχήμα 4.120 και στη δεύτερη φάση της ναυμαχίας, τα πλοία της Βενετίας –τα οποία είναι αυτά με τα λευκά πανιά– έχουν ηττηθεί και αυτό φαίνεται στο Σχήμα 4.121, καθώς ενεργοποιούνται πάλι τρία αντικείμενα smokes με Particle System components, που σηματοδοτούν την ήττα του στόλου της Βενετίας και εμφανίζεται το κουμπί της συνέχειας.



Σχήμα 4.120: Οι δύο στόλοι ανταλλάσσουν πυρά



Σχήμα 4.121: Ο στόλος της Βενετίας έχει ηττηθεί

Σε αυτό το σημείο που το παιχνίδι ολοκληρώνεται, εμφανίζεται ένα παράθυρο με τις τελικές πληροφορίες για τη ζωή του Μάρκο Πόλο και τα αποτελέσματα του παιχνιδιού και ο παίκτης έχει κερδίσει.

#### 4.4 Επίλογος

Με τη χρήση της εφαρμογής αυτής οι γνώσεις για τη ζωή και τα ταξίδια των συγκεκριμένων θαλασσοπόρων-εξερευνητών του 15<sup>ου</sup>-16<sup>ου</sup> αιώνα προσφέρονται με απλό και ευχάριστο τρόπο, καθώς επιτρέπεται στον παίκτη να συμμετέχει ενεργά στη μάθηση. Δημιουργείται ένα ευχάριστο και ελκυστικό παιχνίδι, που παρέχει τη δυνατότητα στον παίκτη να μαθαίνει παίζοντας. Η εφαρμογή, τέλος, είναι δομημένη με τρόπο τέτοιο, ώστε να αποφεύγονται όσο το δυνατόν γίνεται περισσότερα λάθη του παίκτη, πχ. διπλές απαντήσεις, τυχαίο πάτημα κουμπιού κτλ.

## Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

Στη σύγχρονη εποχή η επαυξημένη πραγματικότητα αξιοποιείται σε πάρα πολλούς τομείς της ζωής και επηρεάζει σημαντικά την καθημερινότητα του ανθρώπου. Ένας από αυτούς τους τομείς είναι και η εκπαίδευση. Μέσω της επαυξημένης πραγματικότητας, παρέχεται η δυνατότητα η γνώση να μετατραπεί σε παιχνίδι. Αυτό είναι εύκολο σήμερα, καθώς οι νέοι μαθητές έχουν πρόσβαση σε συσκευές (υπολογιστής, κινητό τηλέφωνο, τάμπλετ) και εφαρμογές που μπορούν να χρησιμοποιούν, ώστε να επωφεληθούν ποικιλοτρόπως.

Η παρούσα διπλωματική εργασία παρουσιάζει τη χρήση της εφαρμογής που σχεδιάστηκε για τη διδασκαλία του μαθήματος της Ιστορίας της ΣΤ΄ Δημοτικού. Το θέμα της πραγματεύεται τη ζωή των θαλασσοπόρων-εξερευνητών του 15ου-16ου αιώνα των Βαρθολομαίου Ντιάζ, Φερδινάνδο Μαγγελάνο, Χριστόφορο Κολόμβο και Βάσκο ντα Γκάμα. Συγκεκριμένα, πρόκειται για την «Ενότητα Α. Οι εξελίξεις στην Ευρώπη κατά τους Νεότερους Χρόνους (μέσα 15<sup>ου</sup> αιώνα – αρχές 19<sup>ου</sup> αιώνα)», και το «Κεφάλαιο 2. Από τις Γεωγραφικές Ανακαλύψεις στο Διαφωτισμό», που περιέχεται στις σελίδες 16-20 του σχολικού εγχειριδίου. Όπως προαναφέρθηκε, θεωρήθηκε χρήσιμο για την πληρέστερη κάλυψη του θέματος να συμπεριληφθούν στο παιχνίδι η ζωή και τα ταξίδια του Μάρκο Πόλο –αν και ανήκει στον 13<sup>ο</sup> αιώνα–, καθώς κατά τη μελέτη της συγκεκριμένης ενότητας διαπιστώθηκε πως διδάσκεται στη σχολική τάξη, αφού πρόκειται για τον πρώτο ευρωπαϊό εξερευνητή που έκανε γνωστή την Κίνα στην Ευρώπη.

Από τη χρήση αυτής της εφαρμογής αποκομίζονται σημαντικά οφέλη. Συγκεκριμένα, η εφαρμογή που παρουσιάζεται αποτελείται από πέντε εκπαιδευτικά παιχνίδια, όπου γίνεται χρήση επαυξημένης πραγματικότητας στο συγκεκριμένο κεφάλαιο της ύλης του σχολικού εγχειριδίου. Η Ιστορία αναμφίβολα αποτελεί ένα σπουδαίο μάθημα για την εξέλιξη των γνώσεων ενός μαθητή. Παράλληλα, όμως, αποτελεί ένα μάθημα που δυσκολεύει τα παιδιά, καθώς αναγκάζονται να αποστηθίζουν τις γνώσεις που τους παρέχονται. Τα βιβλία οπωσδήποτε ακολουθούν μια δομή που θα μπορούσε να χαρακτηριστεί με τα δεδομένα της εποχής παρωχημένη. Επομένως, είναι σημαντικό να δίνεται η δυνατότητα στον διδάσκοντα να διαφοροποιεί τη διδασκαλία στοχεύοντας στην παρουσίαση της γνώσης με όσο το δυνατόν απλούστερο τρόπο καθώς και στη μεγαλύτερη συμμετοχή των μαθητών.

Με τη συγκεκριμένη εφαρμογή δίνεται η δυνατότητα στον μαθητή να εξερευνήσει τη ζωή και τα ταξίδια των θαλασσοπόρων και εξερευνητών μεταξύ 15ου-16ου με έναν ευχάριστο τρόπο. Μέσα από τα παιχνίδια εμπλουτίζει τις γνώσεις του αποτελεσματικά, πράγμα που επιτυγχάνεται κυρίως μέσα από τη διάδραση.

Ασφαλώς και η εφαρμογή επιδέχεται βελτιώσεων και εμπλουτισμού. Αυτό θα επιτευχθεί με την προσθήκη μεγαλύτερου αριθμού παιχνιδιών που θα περιλαμβάνουν περισσότερες ερωτήσεις, εικόνες και πληροφορίες. Επιπλέον, θα μπορούσαν να βελτιωθούν τα γραφικά του παιχνιδιού και να γίνουν πιο ζωντανά, να προστεθούν και επιπλέον εφέ ζωντανής εικόνας. Τέλος, θα μπορούσε να γίνει η εφαρμογή συμβατή και για άλλες συσκευές εκτός από το Android, καθώς και να αυξηθεί ο βαθμός αλληλεπίδρασης μεταξύ του χρήστη και της εφαρμογής.



## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] «Εκτεταμένη Πραγματικότητα» [Online]. Διαθέσιμο στο: [https://en.wikipedia.org/wiki/Extended\\_reality](https://en.wikipedia.org/wiki/Extended_reality). Προσπελάστηκε: 04/03/2025
- [2] «Επαυξημένη/Εικονική πραγματικότητα στην καθημερινή μας ζωή» [Online]. Διαθέσιμο στο: <https://courses.minnlearn.com/el/courses/emerging-technologies/extended-reality-vr-ar-mr/ar-vr-in-our-daily-life>. Προσπελάστηκε: 05/03/2025
- [3] «Εικονική Πραγματικότητα» [Online]. Διαθέσιμο στο: [https://el.wikipedia.org/wiki/%CE%95%CE%B9%CE%BA%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE\\_%CF%80%CF%81%CE%B1%CE%B3%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CF%8C%CF%84%CE%B7%CF%84%CE%B1](https://el.wikipedia.org/wiki/%CE%95%CE%B9%CE%BA%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE_%CF%80%CF%81%CE%B1%CE%B3%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CF%8C%CF%84%CE%B7%CF%84%CE%B1). Προσπελάστηκε: 05/03/2025
- [4] «Επαυξημένη Πραγματικότητα» [Online]. Διαθέσιμο στο: [https://el.wikipedia.org/wiki/%CE%95%CF%80%CE%B1%CF%85%CE%BE%CE%B7%CE%BC%CE%AD%CE%BD%CE%B7\\_%CF%80%CF%81%CE%B1%CE%B3%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CF%8C%CF%84%CE%B7%CF%84%CE%B1](https://el.wikipedia.org/wiki/%CE%95%CF%80%CE%B1%CF%85%CE%BE%CE%B7%CE%BC%CE%AD%CE%BD%CE%B7_%CF%80%CF%81%CE%B1%CE%B3%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CF%8C%CF%84%CE%B7%CF%84%CE%B1). Προσπελάστηκε: 10/03/2025
- [5] «Qualcomm: Ετοιμάζει τον ανταγωνιστή του Vision Pro της Apple – Τι είναι το «mixed reality»» [Online]. Διαθέσιμο στο: <https://www.mononews.gr/business/qualcomm-etimazi-ton-antagonisti-tou-vision-pro-tis-apple-ti-ine-to-mixed-reality>. Προσπελάστηκε: 10/03/2025
- [6] «Virtual reality can help make people more copassionate compared to other media, new Stanford study finds» [Online]. Διαθέσιμο στο: <https://news.stanford.edu/stories/2018/10/virtual-reality-can-help-make-people-empathetic>. Προσπελάστηκε: 15/03/2025
- [7] «What is augmented reality?» [Online]. Διαθέσιμο στο: <https://www.livescience.com/34843-augmented-reality.html>. Προσπελάστηκε: 15/03/2025
- [8] «Mixed Reality: Technologie und Impulsgeber für eine Arbeitswelt im Wandel» [Online]. Διαθέσιμο στο: <https://news.microsoft.com/de-de/features/mixed-reality-technologien-neue-arbeitswelt>. Προσπελάστηκε: 16/03/2025
- [9] «Εικονική & Επαυξημένη πραγματικότητα» [Online]. Διαθέσιμο στο: <https://www.dit.uoi.gr/e-class/modules/document/file.php/305/Lesson1%20-%20VR.AR%20%2818.2.22%29.pdf>. Προσπελάστηκε: 20/03/2025
- [10] «Μηχανή Παιχνιδιού» [Online]. Διαθέσιμο στο: [https://en.wikipedia.org/wiki/Game\\_engine](https://en.wikipedia.org/wiki/Game_engine). Προσπελάστηκε: 20/03/2025
- [11] «Μηχανή Παιχνιδιού Unity» [Online]. Διαθέσιμο στο: [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)). Προσπελάστηκε: 20/03/2025

[12] «Unity Raycast» [Online]. Διαθέσιμο στο:  
<https://docs.unity3d.com/6000.1/Documentation/ScriptReference/Physics.Raycast.html>.  
Προσπελάστηκε: 20/05/2025

[13] «Unity Canvas» [Online]. Διαθέσιμο στο:  
<https://docs.unity3d.com/Packages/com.unity.ugui@1.0/manual/UICanvas.html>. Προσπελάστηκε:  
20/05/2025

[14] «Unity Transform» [Online]. Διαθέσιμο στο:  
<https://docs.unity3d.com/6000.1/Documentation/ScriptReference/Transform.html>. Προσπελάστηκε:  
20/05/2025

[15] «Unity Inspector» [Online]. Διαθέσιμο στο:  
<https://docs.unity3d.com/6000.1/Documentation/Manual/UsingTheInspector.html>. Προσπελάστηκε:  
20/05/2025

[16] «Unity Animations» [Online]. Διαθέσιμο στο:  
<https://docs.unity3d.com/Manual/AnimationSection.html>. Προσπελάστηκε: 20/05/2025

[17] «Unity Particle System» [Online]. Διαθέσιμο στο:  
<https://docs.unity3d.com/6000.1/Documentation/Manual/ParticleSystems.html>. Προσπελάστηκε:  
20/05/2025