

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ  
ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΟ  
ΜΑΘΗΜΑ ΒΙΟΛΟΓΙΑΣ ΤΗΣ Α' ΓΥΜΝΑΣΙΟΥ



Της φοιτήτριας  
Πασχαλίνας Γκιώση  
Αρ. Μητρώου: 164650

Επιβλέπων  
Ευκλείδης Κεραμόπουλος  
Αναπληρωτής Καθηγητής

Τίτλος Δ.Ε. Ανάπτυξη εκπαιδευτικού παιχνιδιού επαυξημένης πραγματικότητας για το μάθημα  
Βιολογίας της Α' Γυμνασίου  
Κωδικός Δ.Ε. 22188  
Όνοματεπώνυμο φοιτήτριας Πασχαλίνα Γκιώση  
Όνοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος  
Ημερομηνία ανάληψης Δ.Ε. 24-03-2022  
Ημερομηνία περάτωσης Δ.Ε. ...

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

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία της φοιτήτριας Πασχαλίνας Γκιώση 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 οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

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

## Πρόλογος

Με την εξέλιξη της τεχνολογίας δημιουργήθηκαν οι τεχνολογίες εκτεταμένης πραγματικότητας όπου εμφανίζονται σε πολλούς κλάδους και προσφέρουν πολλά οφέλη. Ένας από τους κλάδους είναι η εκπαίδευση. Η νέα γενιά μαθητών είναι εξοικειωμένη με την τεχνολογία με αποτέλεσμα να έχουν διαφορετικές εκπαιδευτικές απαιτήσεις. Έχουν συνηθίσει να λαμβάνουν τις πληροφορίες εύκολα και σε διάφορες μορφές. Η εκπαίδευση με τη χρήση της εκτεταμένης πραγματικότητας είναι ένας νέος και καινοτόμος τρόπος μάθησης. Μέσω της εκτεταμένης πραγματικότητας οι μαθητές μπορούν να μάθουν, να περιηγηθούν και να εξερευνήσουν ένα εικονικό περιβάλλον γεμάτο μαθησιακές πληροφορίες. Η εκπαίδευση μπορεί πλέον να γίνει πιο διασκεδαστική, οπτικοποιώντας τις πληροφορίες του σχολικού βιβλίου, βοηθώντας τους μαθητές να κατανοήσουν το μάθημα και να συνεργαστούν μεταξύ τους, παίζοντας παιχνίδια ή εξερευνώντας το εικονικό περιβάλλον.

## Περίληψη

Στην παρούσα διπλωματική εργασία στόχος ήταν η σχεδίαση και ανάπτυξη εκπαιδευτικής εφαρμογής για το μάθημα της Βιολογίας Α' Γυμνασίου. Με τη χρήση της επαυξημένης πραγματικότητας δημιουργήθηκαν πέντε επιμορφωτικά παιχνίδια με στόχο την ενσωμάτωσή τους στη διδασκαλία ώστε οι μαθητές να μπορούν να κατανοήσουν πολύπλοκες έννοιες με έναν διαφορετικό αλλά διασκεδαστικό τρόπο. Τα παιχνίδια έχουν δημιουργηθεί με γνώμονα τη διδακτέα ύλη του μαθήματος για το σχολικό έτος 2022-2023. Για να μπορέσουν οι μαθητές να χρησιμοποιήσουν την εφαρμογή είναι απαραίτητη μια συσκευή με την οποία σαρώνοντας τα Qr codes που είναι τοποθετημένα σε συγκεκριμένες σελίδες του σχολικού βιβλίου, εμφανίζεται το διαδραστικό περιβάλλον. Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε η πλατφόρμα Unity σε συνδυασμό με κώδικα γραμμένο σε γλώσσα C# και το πακέτο επαυξημένης πραγματικότητας Vuforia. Επιπλέον, μέσω του προγράμματος Blender έγινε η δημιουργία των τρισδιάστατων αντικειμένων.

# Development of an augmented reality educational game for the Biology high school lesson of first grade

Paschalina Gkiosi

## **Abstract**

The aim of this thesis was for the development and design of an educational application for the subject course of Biology for the first year of High School. With the use of augmented reality, five educational games were created with the aim of integrating them into the teaching process so that students can obtain an understanding of complex concepts in an alternative but entertaining way. The game have been created in consultation with the course syllabus for the school year of 2022-2023 in mind. In order for students to be able to use the app a device is needed with which to scan the Qr codes which have been placed on specific pages of the school textbooks to display the interactive interface. The Unity platform was used to implement the application in combination with a code which was written in *C#* and the augmented reality package Vuforia. In addition, the 3D objects were created using the Blender program.

## Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής εργασίας, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή κ. Ευκλείδη Κεραμόπουλο, που μου εμπιστεύθηκε την παρούσα εργασία, αναθέτοντας μου το συγκεκριμένο θέμα.

Επιπλέον, οφείλω ένα μεγάλο ευχαριστώ στην οικογένεια μου και στον άνδρα μου που με στήριξαν στην προσπάθεια μου να ολοκληρώσω τις σπουδές μου με ατελείωτη υπομονή και κατανόηση.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους μου και όλα τα άτομα που με βοήθησαν για να πραγματοποιήσω την διπλωματική εργασία.

# Περιεχόμενα

Πρόλογος.....	iii
Περίληψη.....	iv
Abstract .....	v
Ευχαριστίες .....	vi
Περιεχόμενα .....	vii
Κατάλογος Σχημάτων .....	ix
Συντομογραφίες.....	xii
Κεφάλαιο 1ο: Εισαγωγή.....	1
Κεφάλαιο 2ο: Τεχνολογίες εκτεταμένης πραγματικότητας.....	3
2.1 Εισαγωγή.....	3
2.2 Εκτεταμένη πραγματικότητα.....	3
2.2.1 Επαυξημένη πραγματικότητα.....	3
2.2.2 Μικτή πραγματικότητα .....	4
2.2.3 Εικονική πραγματικότητα .....	5
2.3 Εφαρμογή εκτεταμένης πραγματικότητας σε ορισμένους τομείς .....	5
2.3.1 Μάρκετινγκ .....	5
2.3.2 Τουρισμός.....	6
2.3.3 Εκπαίδευση.....	7
2.4 Επίλογος.....	8
Κεφάλαιο 3ο: Προγράμματα που χρησιμοποιήθηκαν .....	9
3.1 Εισαγωγή.....	9
3.2 Blender .....	9
3.3 Unity.....	9
3.3.1 Εγκατάσταση και ρυθμίσεις Unity και Unity Hub.....	10
3.4 Vuforia .....	12
3.4.1 Εγκατάσταση και ρυθμίσεις Vuforia.....	12
3.4.2 Σύνδεση Vuforia και Unity .....	13
3.5 Ρυθμίσεις κινητού.....	14
3.6 Επίλογος .....	16
Κεφάλαιο 4ο: Η εφαρμογή.....	17
4.1 Εισαγωγή.....	17
4.2 Σχολικό βιβλίο – Επιλογή παιχνιδιών .....	17

4.3	Η βάση της εφαρμογής.....	17
4.4	Γραμματοσειρές και μουσική.....	18
4.5	Αρχική σκηνή εφαρμογής.....	19
4.6	Μενού εφαρμογής.....	23
4.7	Επίλογος.....	25
Κεφάλαιο 5ο: Οργανισμοί.....		27
5.1	Εισαγωγή.....	27
5.2	Ανάλυση παιχνιδιού.....	27
5.3	Επίλογος.....	35
Κεφάλαιο 6ο: Κύτταρα.....		37
6.1	Εισαγωγή.....	37
6.2	Ανάλυση παιχνιδιού.....	37
6.3	Επίλογος.....	44
Κεφάλαιο 7ο: Τροφές.....		45
7.1	Εισαγωγή.....	45
7.2	Ανάλυση παιχνιδιού.....	45
7.3	Επίλογος.....	52
Κεφάλαιο 8ο: Σπονδυλωτά και ασπόνδυλα ζώα.....		53
8.1	Εισαγωγή.....	53
8.2	Ανάλυση παιχνιδιού.....	53
8.2.1	Animation.....	67
8.3	Επίλογος.....	70
Κεφάλαιο 9ο: Συστήματα οργάνων.....		71
9.1	Εισαγωγή.....	71
9.2	Ανάλυση παιχνιδιού.....	71
9.3	Επίλογος.....	77
Κεφάλαιο 10ο: Συμπεράσματα και βελτιώσεις.....		79
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		80
ΠΑΡΑΡΤΗΜΑ Α : PartTwoGame4.cs.....		81
ΠΑΡΑΡΤΗΜΑ Β : Game53.cs.....		85
ΠΑΡΑΡΤΗΜΑ C : TimerManager.cs.....		89

## Κατάλογος Σχημάτων

Εικόνα 2.1 Παιχνίδι επαυξημένης πραγματικότητας .....	4
Εικόνα 2.2 Παιχνίδι μικτής πραγματικότητας.....	4
Εικόνα 2.3 Παιχνίδι εικονικής πραγματικότητας.....	5
Εικόνα 2.4 Εκτεταμένη πραγματικότητα στο μάρκετινγκ .....	6
Εικόνα 2.5 Εκτεταμένη πραγματικότητα στον τουρισμό.....	7
Εικόνα 2.6 Εκτεταμένη πραγματικότητα στην εκπαίδευση .....	8
Εικόνα 3.1 Εξαγωγή αρχείων από το πρόγραμμα Blender .....	9
Εικόνα 3.2 Πρόγραμμα Unity .....	10
Εικόνα 3.3 Έκδοση Unity και Unity Hub .....	10
Εικόνα 3.4 Πρόσθετα στο Unity Hub .....	11
Εικόνα 3.5 Δημιουργία project στο Unity Hub .....	11
Εικόνα 3.6 Απόκτηση κλειδιού License Key .....	12
Εικόνα 3.7 Βάση με Qr codes .....	13
Εικόνα 3.8 Ρύθμιση AR Camera.....	13
Εικόνα 3.9 Ρύθμιση AR Camera: τοποθέτηση License Key.....	14
Εικόνα 3.10 Ρυθμίσεις Image Target .....	14
Εικόνα 3.11 Ρυθμίσεις κινητού στη Unity .....	15
Εικόνα 3.12 Εκτέλεση εφαρμογής .....	15
Εικόνα 4.1 Υλικό εφαρμογής .....	17
Εικόνα 4.2 Αλλαγή γραμματοσειράς .....	18
Εικόνα 4.3 Ρυθμίσεις ήχου.....	19
Εικόνα 4.4 Λογότυπο εφαρμογής.....	20
Εικόνα 4.5 Πρώτη σκηνή εφαρμογής.....	21
Εικόνα 4.6 Βασικές λειτουργίες του script 'IntroButtons' .....	22
Εικόνα 4.7 Κενά αντικείμενα στο Unity .....	23
Εικόνα 4.8 Επικοινωνία κουμπιών και scripts .....	23
Εικόνα 4.9 Μενού εφαρμογής.....	24
Εικόνα 4.10 Βασικές λειτουργίες του script 'MenuButton' .....	24
Εικόνα 5.1 Έναρξη παιχνιδιού 'Οργανισμοί' .....	27
Εικόνα 5.2 Εμφάνιση παιχνιδιού 'Οργανισμοί' .....	28
Εικόνα 5.3 Κουμπί βοήθειας παιχνιδιού 'Οργανισμοί' .....	29
Εικόνα 5.4 Μεταβλητές script 'Game1' .....	30
Εικόνα 5.5 Tags στα τρισδιάστατα αντικείμενα .....	30
Εικόνα 5.6 Εντολή Input.GetMouseButtonDown.....	31
Εικόνα 5.7 Τοποθέτηση ιδιότητας Box Collider.....	31
Εικόνα 5.8 Αντικείμενα με ιδιότητα Box Collider.....	32
Εικόνα 5.9 Script 'Event' .....	32
Εικόνα 5.10 Ρυθμίσεις σύνδεσης για το script 'Event' .....	33
Εικόνα 5.11 Έλεγχος σωστής ή λάθος απάντησης στο script 'Game1' .....	33
Εικόνα 5.12 Μέθοδοι για την ενεργοποίηση των cube-πλαισίων .....	33
Εικόνα 5.13 Μετρητής για το τέλος του πρώτου παιχνιδιού.....	34
Εικόνα 5.14 Σωστή τοποθέτηση αντικειμένων στο πρώτο παιχνίδι .....	34
Εικόνα 5.15 Τελευταία σκηνή πρώτου παιχνιδιού.....	35
Εικόνα 6.1 Έναρξη παιχνιδιού 'Κύτταρα'.....	37

Εικόνα 6.2 Εμφάνιση παιχνιδιού 'Κύτταρο' - Ζωικό κύτταρο .....	38
Εικόνα 6.3 Μεταβλητές Script 'Text Control' .....	39
Εικόνα 6.4 Πλαίσιο για την τοποθέτηση της απάντησης .....	40
Εικόνα 6.5 Μέθοδος onClickRibosomataBt() .....	41
Εικόνα 6.6 Καμβάς με πληροφορίες .....	41
Εικόνα 6.7 Μέθοδοι για το κλείσιμο των καμβά του script 'Information' .....	42
Εικόνα 6.8 Κώδικα για τη λάθος απάντηση .....	42
Εικόνα 6.9 Εμφάνιση παιχνιδιού 'Κύτταρο' - Φυτικό κύτταρο .....	43
Εικόνα 6.10 Τελευταία σκηνή δεύτερου παιχνιδιού .....	43
Εικόνα 7.1 Έναρξη παιχνιδιού 'Τροφές' .....	45
Εικόνα 7.2 Εμφάνιση παιχνιδιού 'Τροφές' .....	46
Εικόνα 7.3 Μεταβλητές script 'Game3' .....	47
Εικόνα 7.4 Ρυθμίσεις μπάρας προόδου στο Unity .....	47
Εικόνα 7.5 Μέθοδος 'onClickStartButton' του script 'Game3' .....	48
Εικόνα 7.6 Βασικοί έλεγχοι μεθόδου 'Update' του script 'Game3' .....	48
Εικόνα 7.7 Τοποθέτηση αντικειμένων στο καλάθι .....	49
Εικόνα 7.8 Μέθοδος ανανέωσης μπάρας προόδου .....	50
Εικόνα 7.9 Συνθήκη ολοκλήρωσης μιας κατηγορίας .....	50
Εικόνα 7.10 Καμβάς με πληροφορίες σχετικές με μια συγκεκριμένη κατηγορία .....	51
Εικόνα 7.11 Μέθοδος resetSlider() .....	51
Εικόνα 7.12 Τελευταία σκηνή τρίτου παιχνιδιού .....	52
Εικόνα 8.1 Έναρξη παιχνιδιού 'Σπονδυλωτά και ασπόνδυλα' - Μέρος πρώτο .....	54
Εικόνα 8.2 Εμφάνιση παιχνιδιού 'Σπονδυλωτά και ασπόνδυλα' - Μέρος πρώτο .....	55
Εικόνα 8.3 Μεταβλητές script 'Game4' .....	56
Εικόνα 8.4 Μέθοδος onClickStartButton() του script 'Game4' .....	56
Εικόνα 8.5 Βασικοί έλεγχοι μεθόδου Update() του script 'Game4' .....	57
Εικόνα 8.6 Συνθήκη ολοκλήρωσης πρώτου μέρους του τέταρτου παιχνιδιού .....	58
Εικόνα 8.7 Τελευταία σκηνή μέρους πρώτου τέταρτου παιχνιδιού .....	58
Εικόνα 8.8 Μεταβλητές script 'TimerManager' .....	59
Εικόνα 8.9 Μέθοδος 'Update' του script 'TimerManager' .....	60
Εικόνα 8.10 Μέθοδοι 'TimerStart' και 'TimerStop' του script 'TimerManager' .....	60
Εικόνα 8.11 Έναρξη παιχνιδιού 'Σπονδυλωτά και ασπόνδυλα' - Μέρος δεύτερο .....	61
Εικόνα 8.12 Εμφάνιση παιχνιδιού 'Σπονδυλωτά και ασπόνδυλα' - Μέρος δεύτερο .....	62
Εικόνα 8.13 Μεταβλητές script 'PartTwoGame4' .....	63
Εικόνα 8.14 Script 'Question' .....	63
Εικόνα 8.15 Δήλωση ερωτήσεων .....	64
Εικόνα 8.16 Μέθοδος 'Start' του script 'PartTwoGame4' .....	64
Εικόνα 8.17 Μέθοδος 'GetRandomQuestion' του script 'PartTwoGame4' .....	65
Εικόνα 8.18 Μέθοδος Update() του script 'PartTwoGame4' .....	65
Εικόνα 8.19 Μέθοδος 'onClickOne' και 'onClickTwo' .....	66
Εικόνα 8.20 Μέθοδος 'NextQuestion' .....	66
Εικόνα 8.21 Τελευταία σκηνή μέρους δεύτερου τέταρτου παιχνιδιού .....	67
Εικόνα 8.22 Τοποθέτηση 'bones' στα αντικείμενα του Blender για τη δημιουργία animation .....	68
Εικόνα 8.23 Ρυθμίσεις για τη κίνηση του αντικειμένου .....	68
Εικόνα 8.24 Εισαγωγή κίνησης στα αντικείμενα .....	69
Εικόνα 8.25 Ρυθμίσεις στο Unity για την πραγματοποίηση της κίνησης .....	69

Εικόνα 8.26 Ρυθμίσεις αντικειμένου για την πραγματοποίηση της κίνησης .....	70
Εικόνα 9.1 Μενού παιχνιδιού 'Συστήματα οργάνων' .....	72
Εικόνα 9.2 Έναρξη παιχνιδιών 'Συστήματα οργάνων' .....	73
Εικόνα 9.3 Εμφάνιση των τρισδιάστατων σκηνών του παιχνιδιού 'Συστήματα οργάνων' .....	73
Εικόνα 9.4 Μεταβλητές των scripts του πέμπτου παιχνιδιού.....	74
Εικόνα 9.5 Μέθοδος Start() πέμπτου παιχνιδιού.....	75
Εικόνα 9.6 Μέθοδος για τον έλεγχο επιλογής μιας ταμπέλας.....	75
Εικόνα 9.7 Ολοκλήρωση πέμπτου παιχνιδιού.....	76
Εικόνα 9.8 Μέθοδος onClickButtonEndMessage() .....	76
Εικόνα 9.9 Τελευταία σκηνή πέμπτου παιχνιδιού.....	77

## Συντομογραφίες

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

## Κεφάλαιο 1ο: Εισαγωγή

Η παρούσα διπλωματική εργασία αποτελεί ένα εκπαιδευτικό παιχνίδι επαυξημένης πραγματικότητας με τίτλο «Ανάπτυξη εκπαιδευτικού παιχνιδιού επαυξημένης πραγματικότητας για το μάθημα Βιολογίας της Α' Γυμνασίου» και βασίζεται στην ύλη του μαθήματος για το σχολικό έτος 2022-2023. Στόχος της διπλωματικής εργασίας είναι μέσω της πραγματοποίησης παιχνιδιών και εκχώρησης τους στην εκπαιδευτική διδασκαλία, η εκμάθηση της ύλης να γίνεται με έναν πιο παιγνιώδη και ευχάριστο τρόπο. Επιπλέον, τα παιχνίδια προσφέρουν μια πιο ελκυστική εμπειρία στους μαθητές, οι οποίοι δεν παρακολουθούν παθητικά αλλά συμμετέχουν ενεργά στο μάθημα.

Η εφαρμογή ARBiology βασίζεται στη τεχνολογία επαυξημένης πραγματικότητας. Για την ολοκλήρωση της εφαρμογής χρησιμοποιήθηκαν οι πλατφόρμες Unity σε συνδυασμό με το Vuforia μέσω του οποίου γίνεται η ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας. Επιπλέον, τα αντικείμενα της εφαρμογής δημιουργήθηκαν στο περιβάλλον της πλατφόρμας Blender. Οι μαθητές σαρώνοντας με την κάμερα της συσκευής τους τα κατάλληλα QR codes τα οποία είναι τοποθετημένα στο σχολικό βιβλίο, εμφανίζεται το επαυξημένο περιβάλλον.

Στο πρώτο κεφάλαιο της Διπλωματικής εργασίας με τίτλο «Εισαγωγή» παρουσιάζεται ο στόχος της εφαρμογής ARBiology και ο τρόπος που υλοποιήθηκε. Επιπλέον, πραγματοποιείται μια παρουσίαση του κάθε κεφαλαίου.

Στο δεύτερο κεφάλαιο «Τεχνολογίες εκτεταμένης πραγματικότητας» γίνεται η ανάλυση της τεχνολογίας εκτεταμένης πραγματικότητας η οποία περιλαμβάνει τις τεχνολογίες επαυξημένης πραγματικότητας, μικτής πραγματικότητας και εικονικής πραγματικότητας. Στη συνέχεια, αναφέρονται ορισμένοι κλάδοι που χρησιμοποιούνται οι παραπάνω τεχνολογίες και μερικές εφαρμογές που έχουν αποφέρει θετικά αποτελέσματα.

Στο τρίτο κεφάλαιο «Προγράμματα που χρησιμοποιήθηκαν» παρουσιάζονται τα προγράμματα που χρησιμοποιήθηκαν και αναλυτικά οι οδηγίες εγκαταστάσεων και ρυθμίσεων. Επιπλέον, αναφέρονται οι ρυθμίσεις της συσκευής για να μπορεί το πρόγραμμα να μεταφερθεί και να λειτουργεί σωστά στη συσκευή.

Στο τέταρτο κεφάλαιο «Η εφαρμογή» γίνεται μια αναφορά στην ύλη του μαθήματος για το σχολικό έτος 2022-2023 και στα παιχνίδια που δημιουργήθηκαν. Στη συνέχεια, παρουσιάζεται η βάση της εφαρμογής που περιλαμβάνει τις σκηνές, τα αντικείμενα, τις γραμματοσειρές και τη μουσική. Τέλος, αναλύονται οι πρώτες σκηνές του παιχνιδιού και ο τρόπος υλοποίησής τους.

Από το κεφάλαιο πέμπτο έως το κεφάλαιο ένατο αναλύονται τα βήματα που ακολουθήθηκαν για τη δημιουργία των παιχνιδιών της εφαρμογής. Τα βήματα περιλαμβάνουν τη λεπτομερή ανάλυση των διαστάσεων και τρισδιάστατων σκηνών και τα απαραίτητα scripts συμπεριλαμβάνοντας εικόνες για την καλύτερη περιγραφή τους. Αναλυτικότερα, το πέμπτο κεφάλαιο «Οργανισμοί», το έκτο κεφάλαιο «Κύτταρα», το έβδομο κεφάλαιο «Τροφές», το όγδοο κεφάλαιο «Σπονδυλωτά και ασπόνδυλα ζώα» και το ένατο κεφάλαιο «Συστήματα οργάνων».

Στο δέκατο κεφάλαιο «Συμπεράσματα και βελτιώσεις» αναφέρονται τα συμπεράσματα που προέκυψαν από την ολοκλήρωση της Δ.Ε και ορισμένες προτάσεις βελτίωσης για την μελλοντική εξέλιξη της εφαρμογής.



## Κεφάλαιο 2ο: Τεχνολογίες εκτεταμένης πραγματικότητας

### 2.1 Εισαγωγή

Είναι γεγονός ότι η τεχνολογία στη σύγχρονη εποχή αποτελεί αναπόσπαστο κομμάτι της καθημερινής ζωής των ανθρώπων. Η εξέλιξη της τεχνολογίας έχει οδηγήσει σε σημαντική πρόοδο των τεχνολογιών εκτεταμένης πραγματικότητας (Extended Reality). Η εκτεταμένη πραγματικότητα - η οποία αποτελεί το επίκεντρο της προσοχής - έχει αλλάξει τον τρόπο με τον οποίο οι άνθρωποι μαθαίνουν, παίζουν και επικοινωνούν, ενώνοντας τον ψηφιακό κόσμο με τον πραγματικό [2]. Ειδικότερα, ο όρος XR περιλαμβάνει τις τεχνολογίες της εικονικής πραγματικότητας (Virtual Reality), της επαυξημένης πραγματικότητας (Augmented Reality) και της μικτής πραγματικότητας (Mixed Reality) [1]. Μέσω αυτών των τεχνολογιών, έχουν αναπτυχθεί πολυάριθμες εφαρμογές για τους τομείς της υγείας, του μάρκετινγκ, της εκπαίδευσης κλπ. Στο κεφάλαιο αυτό, θα γίνει μια ανάλυση των παραπάνω τεχνολογιών και επιπλέον θα αναφερθούν ορισμένοι τομείς όπου γίνεται χρήση των τεχνολογιών εκτεταμένης πραγματικότητας.

### 2.2 Εκτεταμένη πραγματικότητα

Η εκτεταμένη πραγματικότητα προκαλεί την αίσθηση ενός διαφορετικού εικονικού κόσμου. Θολώνει τα όρια ανάμεσα στην εικονική και στην αληθινή ζωή. Αυτό πραγματοποιείται παρακολουθώντας ταινίες 360 μοίρες, παίζοντας παιχνίδια σε τρισδιάστατη μορφή, κάνοντας εικονική περιήγηση σε μουσεία, ταξιδεύοντας στο διάστημα ή οποιαδήποτε άλλη καθηλωτική εμπειρία. Η εκτεταμένη πραγματικότητα περιέχει τις τεχνολογίες της εικονικής πραγματικότητας (Virtual Reality), της επαυξημένης πραγματικότητας (Augmented Reality) και της μικτής πραγματικότητας (Mixed Reality), οι οποίες θα αναλυθούν παρακάτω [2].

#### 2.2.1 Επαυξημένη πραγματικότητα

Η επαυξημένη πραγματικότητα βελτιώνει την καθημερινότητα των ανθρώπων αφού συνδυάζει την πραγματικότητα με τις εικονικές πληροφορίες. Οι εικονικές πληροφορίες μπορούν να είναι τρισδιάστατα μοντέλα, ήχος, βίντεο, δισδιάστατα αντικείμενα κλπ. Με αυτόν τον τρόπο, ο εικονικός και ο πραγματικός κόσμος αλληλοσυμπληρώνονται ώστε να προσφέρουν στο χρήστη έναν κόσμο γεμάτο εμπειρίες και μια καινούρια οπτική για τα πράγματα που υπάρχουν γύρω του. Τα περιεχόμενα της επαυξημένης πραγματικότητας μπορούν να παρουσιαστούν με αρκετούς τρόπους. Πρώτα από όλα, θα μπορούσαν να προβληθούν μέσω συγκεκριμένων εικόνων όπου έχει ορίσει ο προγραμματιστής και όταν ο χρήστης σαρώσει με την κάμερα του κινητού του την εκάστοτε εικόνα, τότε μόνο εμφανίζονται τα εικονικά αντικείμενα. Επιπρόσθετα, χρησιμοποιώντας έξυπνα γυαλιά τα οποία όταν τα φοράει ο χρήστης τότε μόνο εμφανίζεται στον χώρο του το ψηφιακό περιεχόμενο. Αυτό σημαίνει ότι, η επαυξημένη πραγματικότητα ολοκληρώνει το πραγματικό περιβάλλον κρατώντας το στο επίκεντρο [3][4].

Αξίζει, επιπλέον, να αναφερθούμε στο πιο δημοφιλές παιχνίδι που χρησιμοποιεί επαυξημένη πραγματικότητα, το Pokémon Go. Το Pokémon Go, χρησιμοποιώντας την κάμερα των κινητών, εμφανίζει σε διάφορα σημεία του κόσμου κάποια εικονικά αντικείμενα τα Pokémon, τα οποία ο χρήστης πρέπει με ορισμένους τρόπους να τα κατακτήσει (Εικόνα 2.1).



Εικόνα 2.1 Παιχνίδι επαυξημένης πραγματικότητας

### 2.2.2 Μικτή πραγματικότητα

Η μικτή πραγματικότητα επεκτείνεται ένα βήμα παραπέρα, αφού δίνει τη δυνατότητα στα πραγματικά στοιχεία του περιβάλλοντος να αλληλεπιδρούν με τα εικονικά στοιχεία. Για παράδειγμα, σε ένα εικονικό αντικείμενο μπορούν οι αντανακλάσεις και οι σκιές του να προσαρμόζονται ανάλογα με το φυσικό φως που υπάρχει. Επίσης, τα τρισδιάστατα αντικείμενα έχουν τη δυνατότητα να εναρμονίζονται στο χώρο, προκαλώντας την αίσθηση της πραγματικότητας (Εικόνα 2.2). Με αυτόν τον τρόπο, τα εικονικά στοιχεία ενσωματώνονται επαρκώς στο πραγματικό περιβάλλον του χρήστη, σε αντίθεση με την επαυξημένη πραγματικότητα [5].

Ένα παράδειγμα παιχνιδιού όπου χρησιμοποιεί τη μικτή πραγματικότητα είναι το BlockVerse. Σε αυτό το παιχνίδι, ο χρήστης με την βοήθεια έξυπνων γυαλιών και χειριστηρίων έχει την δυνατότητα να «σκάψει» το πραγματικό έδαφος του σαλονιού του και να εξερευνήσει μέσα σε αυτό.



Εικόνα 2.2 Παιχνίδι μικτής πραγματικότητας.

### 2.2.3 Εικονική πραγματικότητα

Η εικονική πραγματικότητα εμφανίζει έντονες διαφορές σε σύγκριση με τη μικτή και την επαυξημένη πραγματικότητα. Ο χρήστης είναι πλήρως βυθισμένος σε έναν εικονικό κόσμο, ο οποίος έχει δημιουργηθεί από τον υπολογιστή και αποτελείται αποκλειστικά από ένα φανταστικό περιβάλλον. Η τεχνολογία αυτή, απομακρύνει το χρήστη από τον πραγματικό κόσμο και επηρεάζει τις βασικές αισθήσεις του όπως την όραση, αφή και τον ήχο. Συνήθως, μέσω ειδικών γυαλιών VR, ακουστικών και χειριστηρίων γίνεται η διαχείριση των κινήσεων των αντικειμένων [4][5].

Ένα γνωστό παιχνίδι εικονικής πραγματικότητας είναι το Beat Saber. Ο χρήστης φορώντας έξυπνα γυαλιά VR και χρησιμοποιώντας αισθητήρες στα χέρια του, πηγαίνει σε ένα εικονικό περιβάλλον όπου διάφορα αντικείμενα και εμπόδια κατευθύνονται κατά πάνω του και αυτός πρέπει να «χτυπήσει» κάποια από αυτά αποφεύγοντας ταυτόχρονα και τα εμπόδια που υπάρχουν (Εικόνα 2.3).



Εικόνα 2.3 Παιχνίδι εικονικής πραγματικότητας

## 2.3 Εφαρμογή εκτεταμένης πραγματικότητας σε ορισμένους τομείς

Με τη ραγδαία εξέλιξη της εκτεταμένης πραγματικότητας τα τελευταία χρόνια, έχουν δημιουργηθεί αρκετές εφαρμογές σε πολλούς τομείς όπου συναντάμε στην καθημερινότητα. Οι τομείς μπορεί να είναι η εκπαίδευση, το μάρκετινγκ, η υγεία και ο τουρισμός. Με τη διάδοση των εφαρμογών εκτεταμένης πραγματικότητας, αυξάνονται οι τομείς που μπορεί να χρησιμοποιηθεί [6].

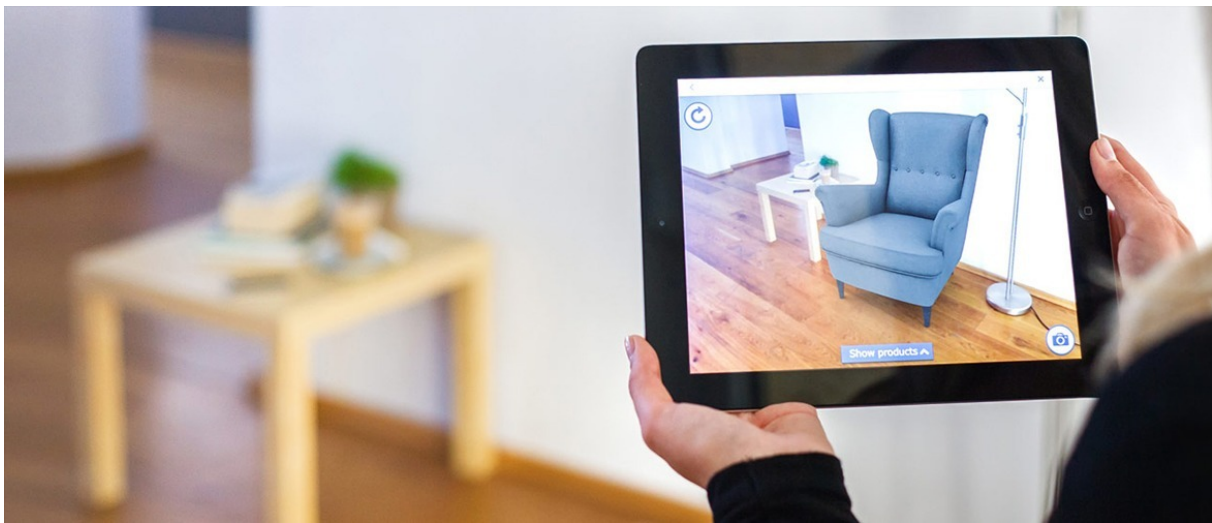
### 2.3.1 Μάρκετινγκ

Καθώς η τεχνολογία εξελίσσεται διαρκώς, το ίδιο συμβαίνει και με τις προσδοκίες και τις ανάγκες των καταναλωτών. Η εκτεταμένη πραγματικότητα είναι μια τεχνολογία που επιτρέπει στις εταιρείες να προσεγγίσουν τους καταναλωτές και να αλληλεπιδρούν μαζί τους με έναν νέο τρόπο. Μέσω του μάρκετινγκ εκτεταμένης πραγματικότητας, οι επιχειρήσεις μπορούν να δημιουργήσουν διαδραστικές εμπειρίες με στόχο την προβολή των υπηρεσιών και των προϊόντων τους. Με αυτόν τον τρόπο, το

περιεχόμενο μπορεί να γίνει πιο ελκυστικό αφήνοντας στην άκρη τις παραδοσιακές καμπάνιες μάρκετινγκ των περιοδικών και της τηλεόρασης [4].

Πολλές εταιρίες έχουν δημιουργήσει εφαρμογές για τη διακόσμηση του σπιτιού. Μέσω αυτών των εφαρμογών, οι χρήστες χρησιμοποιώντας την κάμερα της συσκευής τους μπορούν να τοποθετήσουν έπιπλα στον χώρο του σπιτιού τους πριν τα αγοράσουν. Με αυτόν τον τρόπο, αποκτούν μια πιο ακριβή εικόνα για το αντικείμενο που επιθυμούν και πως αυτό θα μπορούσε να ενσωματωθεί και να συμπληρώσει τον χώρο τους (Εικόνα 2.4).

Ένα ακόμα παράδειγμα χρήσης εκτεταμένης πραγματικότητας στο μάρκετινγκ είναι οι εφαρμογές ρούχων. Οι εταιρείες έχουν δημιουργήσει εφαρμογές όπου ο χρήστης μπορεί να δοκιμάσει ρούχα και να δει πως θα ταίριαζαν επάνω στο σώμα του, ανάλογα με το νούμερο που φοράει. Με αυτόν τον τρόπο, μειώνεται η αβεβαιότητα των ηλεκτρονικών αγορών και περιορίζονται οι επιστροφές προϊόντων.



Εικόνα 2.4 Εκτεταμένη πραγματικότητα στο μάρκετινγκ

### 2.3.2 Τουρισμός

Η τουριστική βιομηχανία είναι μια από τις ταχύτερα αναπτυσσόμενες βιομηχανίες στον κόσμο, αφού όλο και περισσότεροι άνθρωποι ταξιδεύουν καθημερινά. Η δημιουργία ενός ταξιδιού είναι μια δύσκολη διαδικασία καθώς οι πληροφορίες που υπάρχουν σε ένα καινούριο περιβάλλον είναι αρκετές. Έχουν δημιουργηθεί εφαρμογές εκτεταμένης πραγματικότητας όπου μπορούν να παρέχουν εύκολες λύσεις και πολλές πληροφορίες για το ταξίδι των τουριστών [4].

Ένα παράδειγμα χρήσης εκτεταμένης πραγματικότητας στον τουριστικό τομέας είναι εφαρμογές όπου εμφανίζουν πληροφορίες για τα αξιοθέατα. Αναλυτικότερα, χρησιμοποιώντας την κάμερα της κινητής συσκευής, ο χρήστης μπορεί να διαβάσει πληροφορίες σχετικές με το αξιοθέατο που βλέπει εκείνη τη στιγμή (Εικόνα 2.5).

Επιπλέον, όταν οι τουρίστες ταξιδεύουν σε μια ξένη χώρα δεν είναι απαραίτητο ότι γνωρίζουν την τοπική γλώσσα της εκάστοτε περιοχής. Μέσω εφαρμογών εκτεταμένης πραγματικότητας, οι χρήστες μπορούν να τοποθετήσουν την κάμερα της συσκευής τους μπροστά από μια ταμπέλα και η εφαρμογή να εμφανίσει τη μετάφραση στην γλώσσα που επιθυμούν. Με αυτόν τον τρόπο, οι τουρίστες νιώθουν μεγαλύτερη ασφάλεια και βελτιώνουν την ποιότητα της διαμονής τους σε μια διαφορετική χώρα.

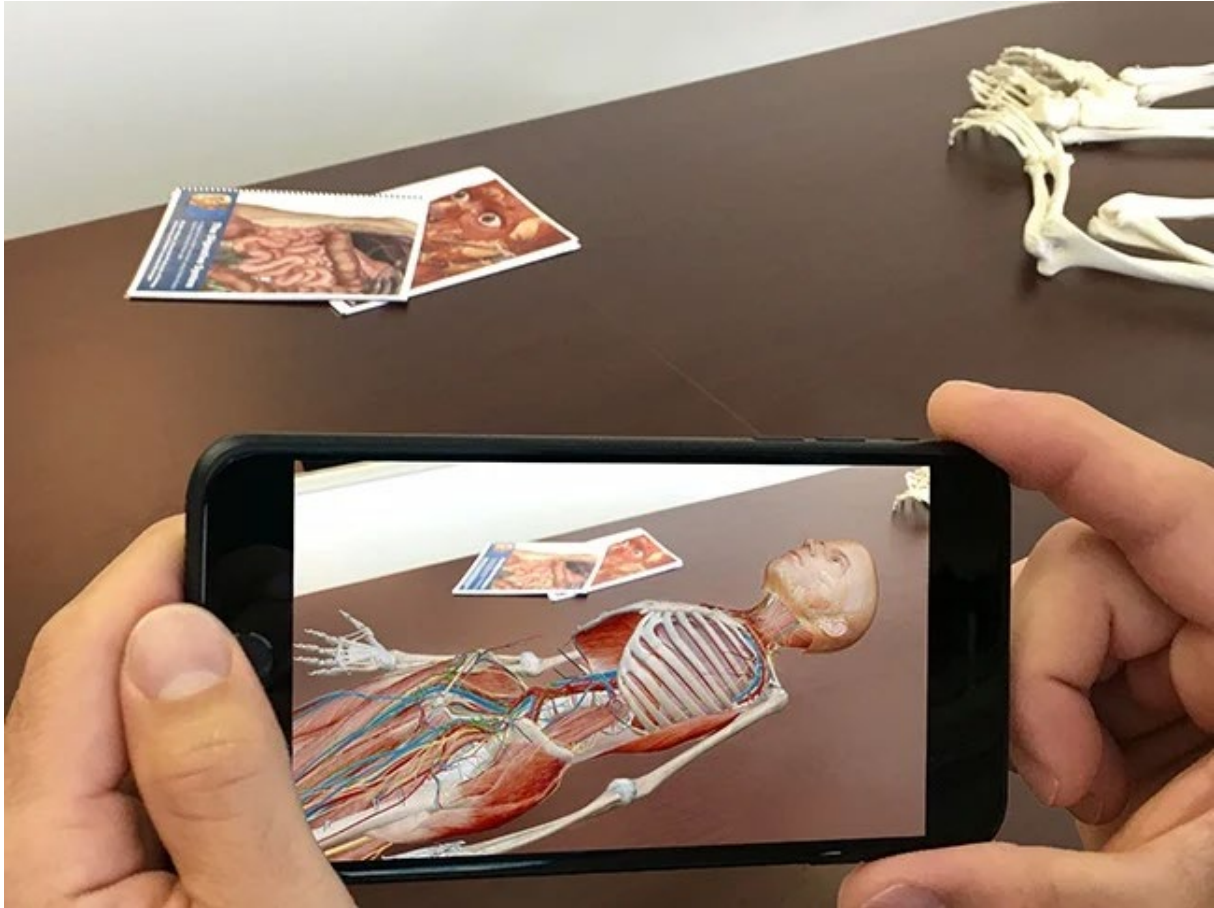


Εικόνα 2.5 Εκτεταμένη πραγματικότητα στον τουρισμό

### 2.3.3 Εκπαίδευση

Η εκτεταμένη πραγματικότητα είναι μια από τις πιο σημαντικές καινοτομίες στην σύγχρονη εκπαίδευση και τα επόμενα χρόνια αναμένεται να αλλάξει ριζικά τον τρόπο μάθησης. Ήδη σε πολλά σχολεία και πανεπιστήμια έχουν ενσωματωθεί μαθήματα όπου πραγματοποιούνται αποκλειστικά με τη βοήθεια εφαρμογών βασισμένες στην εκτεταμένη πραγματικότητα. Μέσω αυτών των εφαρμογών, οι μαθητές είναι σε θέση να κατανοήσουν καλύτερα την ύλη που διδάσκονται. Αυτό συμβαίνει γιατί οι μαθητές μπορούν να συμμετέχουν ενεργά στο μάθημα αντί να ακούν απλώς παθητικά ή να διαβάζουν από ένα σχολικό βιβλίο. Επιπλέον, μέσω παιχνιδιών οι μαθητές βελτιώνουν την παραγωγικότητα και την απόδοση τους μέσα στη τάξη αφού μέσω των εφαρμογών γίνεται οπτικοποίηση αυτό που μελετούν, δημιουργώντας ένα διαφορετικό ψηφιακό περιβάλλον. Το περιβάλλον αυτό, μπορεί να αποτελείται από τρισδιάστατα αντικείμενα όπως για παράδειγμα την ανατομία του σώματος (Εικόνα 2.6). Παρέχεται η δυνατότητα στον μαθητή να μπορεί να χειριστεί το αντικείμενο που εμφανίζεται όπως να το γυρίσει ή να το φέρει πιο κοντά του, δηλαδή όπως θα συμπεριφερόταν σε ένα πραγματικό αντικείμενο [7]. Επίσης, τέτοιου είδους εφαρμογές μπορούν να βοηθήσουν μαθητές με ειδικές ανάγκες αφού περιλαμβάνουν κείμενο, ήχο, τρισδιάστατα αντικείμενα και βίντεο [5].

Οι εφαρμογές εκτεταμένης πραγματικότητας μεταμορφώνουν την εκπαίδευση ενισχύοντας τη διδασκαλία. Υπάρχει μια ποικιλία εφαρμογών που μπορούν να χρησιμοποιηθούν στο εκπαιδευτικό σύστημα με σκοπό οι μαθητές να βελτιωθούν, να μαθαίνουν και να εξερευνούν τον κόσμο γύρω τους με έναν διαφορετικό τρόπο που δεν μπορούσαν πριν [4].



Εικόνα 2.6 Εκτεταμένη πραγματικότητα στην εκπαίδευση

## 2.4 Επίλογος

Από όλα τα παραπάνω γίνεται φανερό πως οι τεχνολογίες της εκτεταμένης πραγματικότητας έχουν ξεκινήσει να ανοίγουν τον δρόμο τους σε πολλούς κλάδους. Κάθε κλάδος παρουσιάζει διαφορετικές ανάγκες αλλά σε όλους οι τεχνολογίες μπορούν να παρουσιάσουν θετικά αποτελέσματα.

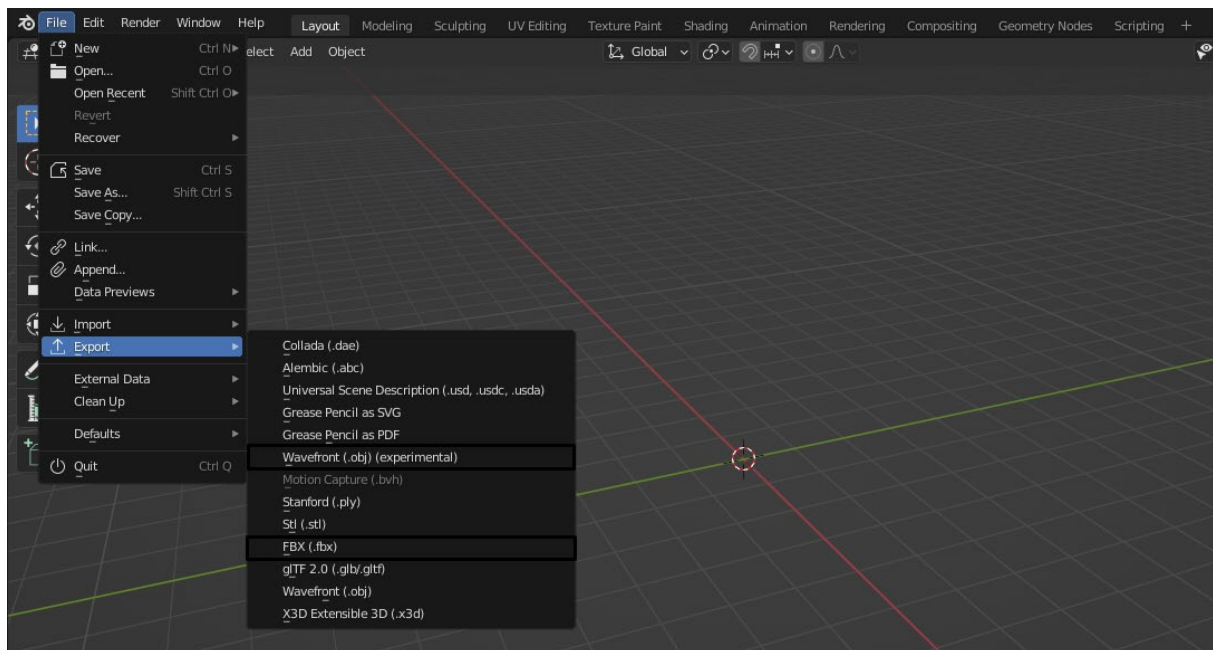
## Κεφάλαιο 3ο: Προγράμματα που χρησιμοποιήθηκαν

### 3.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται όλα τα προγράμματα που χρησιμοποιήθηκαν για την πραγματοποίηση της Δ.Ε. Αρχικά, παρουσιάζεται το πρόγραμμα Blender και ο τρόπος λήψης των τρισδιάστατων αντικειμένων που δημιουργήθηκαν μέσα σε αυτό. Επιπλέον, αναλύεται η εγκατάσταση των Unity και Unity Hub καθώς και οι ρυθμίσεις που χρειάστηκαν. Τέλος, γίνεται η ανάλυση των απαραίτητων ρυθμίσεων ώστε να μπορεί να γίνει η επικοινωνία ανάμεσα στο Unity και στο Vuforia.

### 3.2 Blender

Το Blender [9] είναι ένα από τα δημοφιλέστερα προγράμματα για τη σχεδίαση γραφικών. Κυρίως χρησιμοποιείται για τη δημιουργία τρισδιάστατων αντικειμένων, κινούμενων σχεδίων (animation) και των σχεδιασμό παιχνιδιών. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε η έκδοση 3.2.0. Όλα τα αντικείμενα της εφαρμογής έχουν δημιουργηθεί με τη βοήθεια του Blender καθώς και τα κινούμενα σχέδια, τα οποία θα αναλυθούν παρακάτω. Τα αντικείμενα έχουν εξαχθεί από το Blender στη μορφή .fbx ή .obj και στη συνέχεια, μεταφέρθηκαν στο πρόγραμμα Unity. Η εξαγωγή των αντικειμένων έγινε με τη λειτουργία Export η οποία βρίσκεται στο File → Export → Wavefront για τη μορφή .obj και File → Export → FBX για την .fbx μορφή. (Εικόνα 3.1)



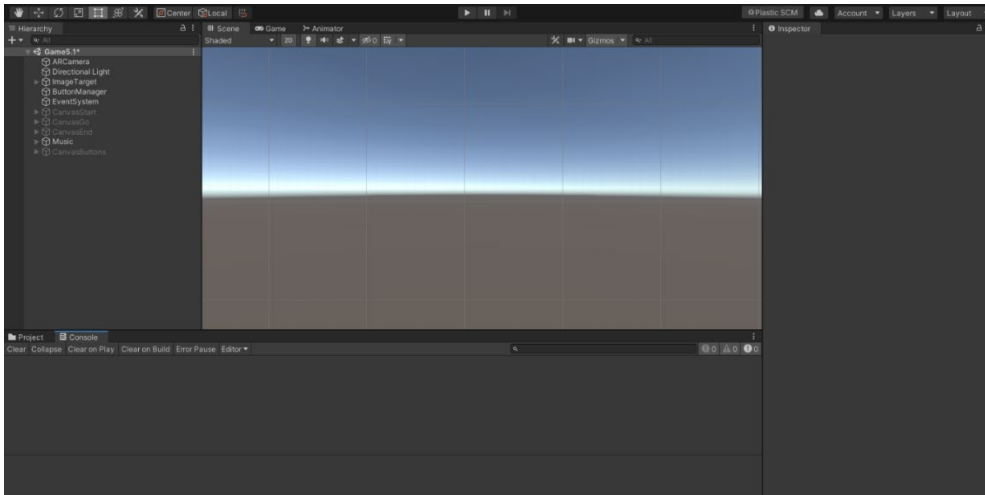
Εικόνα 3.1 Εξαγωγή αρχείων από το πρόγραμμα Blender

### 3.3 Unity

Η πλατφόρμα Unity ανακοινώθηκε για πρώτη φορά στο Apple Worldwide Developers Conference τον Ιούνιο του 2005 από την Unity Technologies ως μηχανή παιχνιδιών. Η Unity χρησιμοποιείται κυρίως

## Κεφάλαιο 3

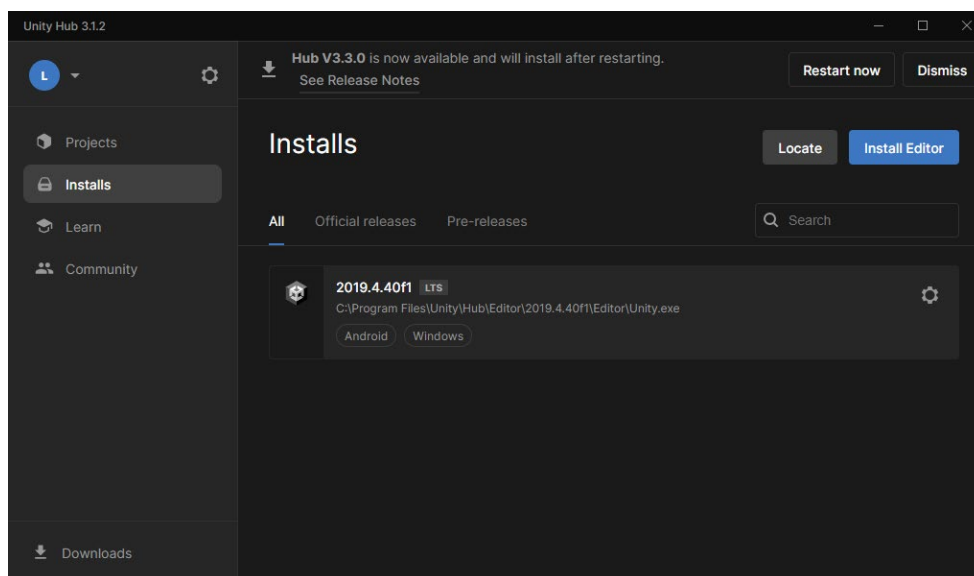
για την ανάπτυξη δισδιάστατων και τρισδιάστατων αντικειμένων και για τη δημιουργία παιχνιδιών επαυξημένης και εικονικής πραγματικότητας [8]. Είναι ένα εύκολο και φιλικό προς το χρήστη περιβάλλον και είναι το βασικό λογισμικό για την πραγματοποίηση της διπλωματικής εργασία. (Εικόνα 3.2)



Εικόνα 3.2 Πρόγραμμα Unity

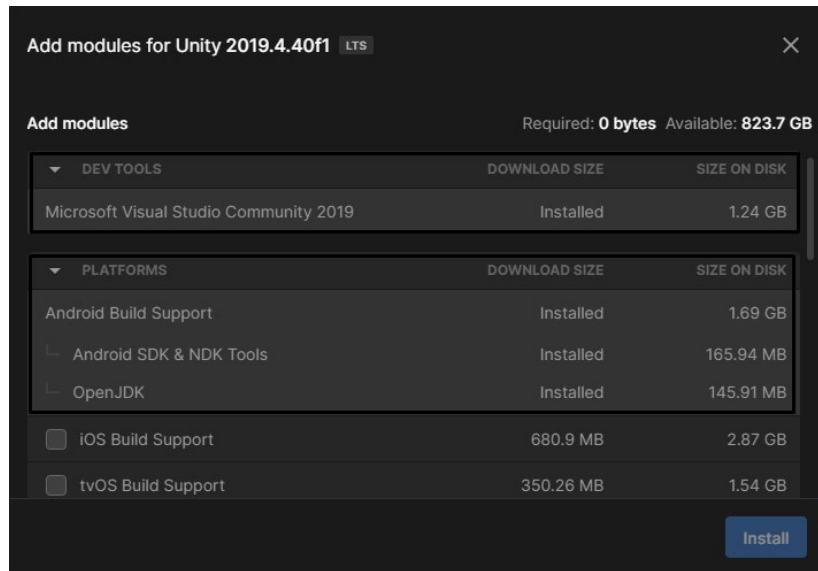
### 3.3.1 Εγκατάσταση και ρυθμίσεις Unity και Unity Hub

Στο περιβάλλον της Unity χρησιμοποιήθηκε η γλώσσα προγραμματισμού C# σε συνδυασμό με το Visual Studio Code το οποίο είναι ένα πρόγραμμα για τη σύνταξη πηγαίου κώδικα. Αρχικά, πραγματοποιήθηκε η εγκατάσταση του Unity Hub όπου είναι ένα πρόγραμμα που βελτιώνει τη διαχείριση των έργων (projects) και του Unity, από την επίσημη ιστοσελίδα της Unity [10]. Η έκδοση που χρησιμοποιήθηκε είναι η 3.1.2 για το Unity Hub και η 2019.4.40f1 για τη Unity. (Εικόνα 3.3)



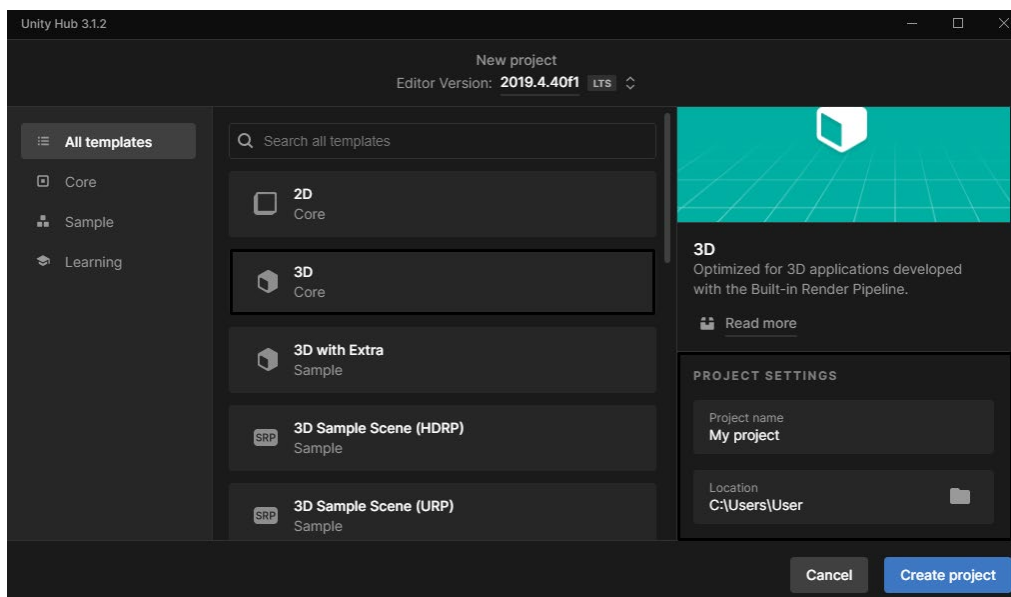
Εικόνα 3.3 Έκδοση Unity και Unity Hub

Για την εφαρμογή, έγινε εγκατάσταση των πρόσθετων Microsoft Visual Studio Community 2019 από την κατηγορία DEV TOOLS και τα Android Build Support, Android SDK & NDK Tools και OpenJDK από την κατηγορία PLATFORMS για να μπορεί να χρησιμοποιηθεί σε Android συσκευές. (Εικόνα 3.4)



Εικόνα 3.4 Πρόσθετα στο Unity Hub

Έπειτα από τα παραπάνω βήματα, δημιουργήθηκε το έργο επιλέγοντας το Projects από την αριστερή στήλη και στη συνέχεια πατώντας το κουμπί New project. Επιλέχθηκε το είδος της εφαρμογής δηλαδή το 3D core και στο project setting ορίστηκε το όνομα της εφαρμογής και ο φάκελος στον οποίο αποθηκεύτηκε.(Εικόνα 3.5)



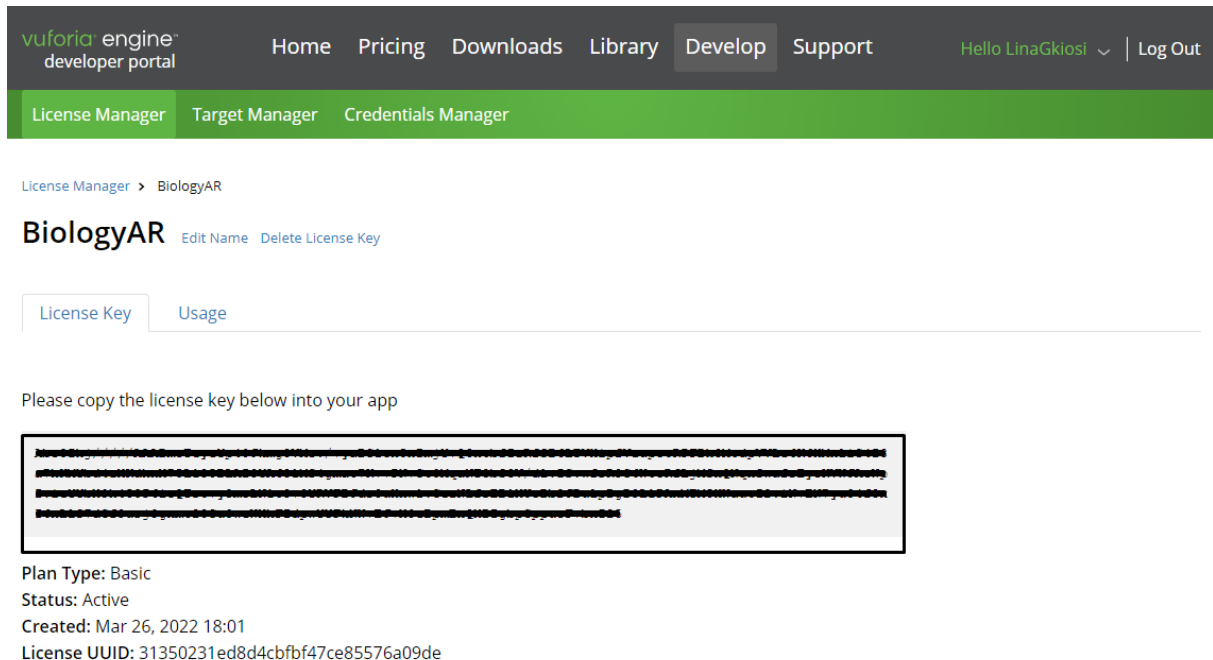
Εικόνα 3.5 Δημιουργία project στο Unity Hub

### 3.4 Vuforia

Το Vuforia χρησιμοποιείται για τη δημιουργία εφαρμογών επαυξημένης πραγματικότητας. Η κύρια λειτουργία του είναι η αναγνώριση εικόνων και αντικειμένων όταν αυτά έχουν καταγραφεί μέσω της κάμερας μιας συσκευής. Με αυτόν τον τρόπο, όταν ο χρήστης ανοίξει την κάμερα του κινητού του και η κάμερα εντοπίσει μια εικόνα ή ένα αντικείμενο, μόνο τότε θα εμφανιστούν τα επαυξημένα αντικείμενα που έχει ορίσει ο προγραμματιστής. Στην παρούσα εφαρμογή, το Vuforia χρησιμοποιήθηκε για την αναγνώριση εικόνων και συγκεκριμένα QR codes.

#### 3.4.1 Εγκατάσταση και ρυθμίσεις Vuforia

Για να γίνει η χρήση του Vuforia, έπρεπε πρώτα να δημιουργηθεί ένας λογαριασμός στην επίσημη ιστοσελίδα [11]. Στη συνέχεια, μέσω του Develop → License Manager → Get Development Key έγινε η απόκτηση του κλειδιού License Key (Εικόνα 3.6). Το License Key είναι ένας μοναδικός κωδικός ο οποίος είναι απαραίτητος για την επικοινωνία του Vuforia με τη Unity, όπου θα αναλυθεί παρακάτω.



Εικόνα 3.6 Απόκτηση κλειδιού License Key

Έπειτα, πηγαίνοντας στο Develop → Target Manager → Add Database, έγινε η δημιουργία μιας βάσης με όνομα BiologyQR. Η βάση αυτή, περιέχει όλες τις εικόνες που χρησιμοποιήθηκαν στην εφαρμογή ώστε όταν τα αναγνωρίζει η κάμερα του κινητού να εμφανίζει τα δισδιάστατα και τρισδιάστατα αντικείμενα. (Εικόνα 3.7)

Target Manager > BiologyQR

## BiologyQR [Edit Name](#)

Type: Device

Targets (8)

Add Target

Download Database (All)

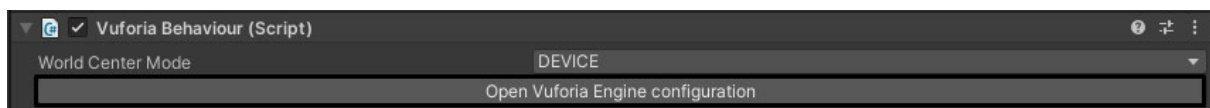
<input type="checkbox"/> Target Name	Type	Rating <sup>①</sup>	Status <sup>▼</sup>	Date Modified
<input type="checkbox"/> game5-4	Image	★★★★★	Active	Oct 06, 2022 21:00
<input type="checkbox"/> game5-3	Image	★★★★★	Active	Oct 06, 2022 20:59
<input type="checkbox"/> game5-2	Image	★★★★★	Active	Oct 06, 2022 20:59
<input type="checkbox"/> game5-1	Image	★★★★★	Active	Oct 06, 2022 20:58
<input type="checkbox"/> game4	Image	★★★★★	Active	Jul 25, 2022 19:02
<input type="checkbox"/> game3	Image	★★★★★	Active	May 30, 2022 21:36
<input type="checkbox"/> game2	Image	★★★★★	Active	Apr 28, 2022 21:21
<input type="checkbox"/> game1	Image	★★★★★	Active	Mar 26, 2022 17:34

Εικόνα 3.7 Βάση με Qr codes

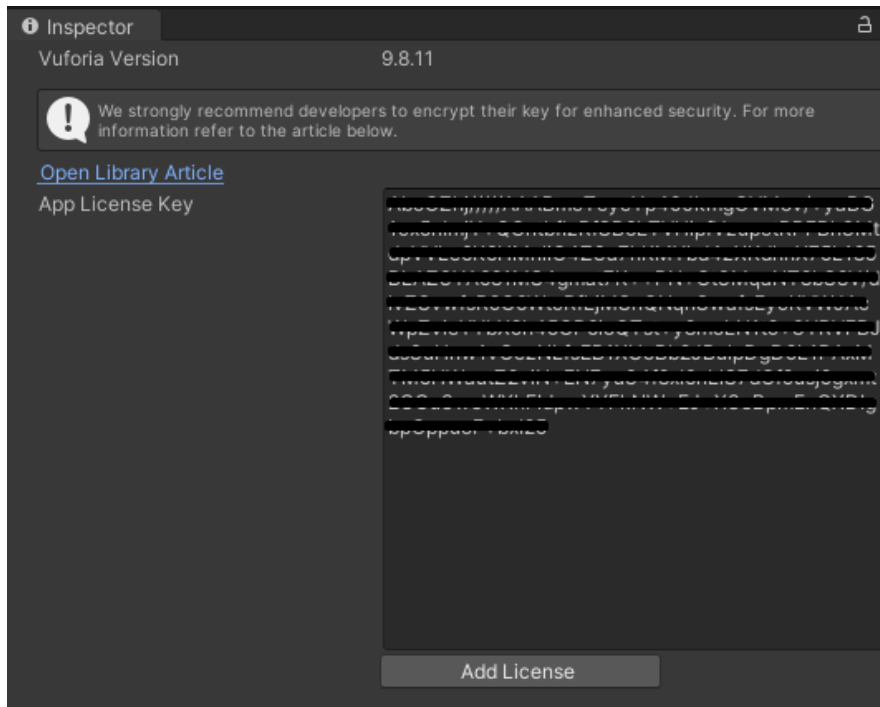
Αφού τοποθετήθηκαν οι εικόνες στη βάση και επιλέχθηκαν όλα τα Target, πατώντας το κουμπί Download Database (All) → Unity Editor → Download, πραγματοποιήθηκε η λήψη της βάσης.

### 3.4.2 Σύνδεση Vuforia και Unity

Για να γίνει η σύνδεση ανάμεσα στο Vuforia και στη Unity και να αναγνωρίζει η Unity τις εικόνες της βάσης που αναφέραμε παραπάνω, τοποθετήθηκε μια AR Camera μέσα στη σκηνή του περιβάλλοντος Unity. Η Main Camera αφαιρέθηκε και αντικαταστάθηκε από την AR Camera η οποία υπάρχει στο GameObject → Vuforia Engine → AR Camera. Επιλέγοντας την AR Camera, εμφανίστηκε στη δεξιά πλευρά το Inspector → Open Vuforia Engine configuration → App License Key (Εικόνα 3.8). Σε αυτό το σημείο, τοποθετήθηκε το License Key. (Εικόνα 3.9)

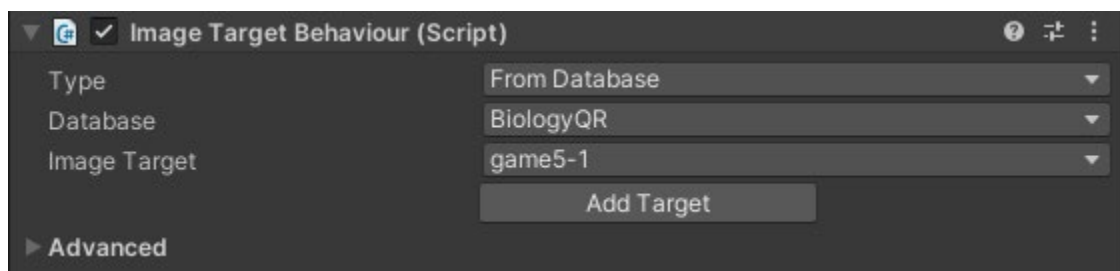


Εικόνα 3.8 Ρύθμιση AR Camera



Εικόνα 3.9 Ρύθμιση AR Camera: τοποθέτηση License Key

Στη συνέχεια, τοποθετήθηκε ένα Image Target από το GameObject → Vuforia Engine → Image Target. Το Image Target αντιπροσωπεύει μια από τις εικόνες όπου τοποθετήθηκαν νωρίτερα στη βάση του Vuforia. Από τις ρυθμίσεις στη δεξιά πλευρά στο Inspector → Image Target Behaviour, επιλέχθηκε ποια εικόνα θα αντιστοιχεί στη σκηνή (Εικόνα 3.10). Μόλις η κάμερα της συσκευής αναγνωρίσει το περιεχόμενο του Image Target, μόνο τότε θα εμφανίσει τα αντικείμενα που έχουν προγραμματιστεί.

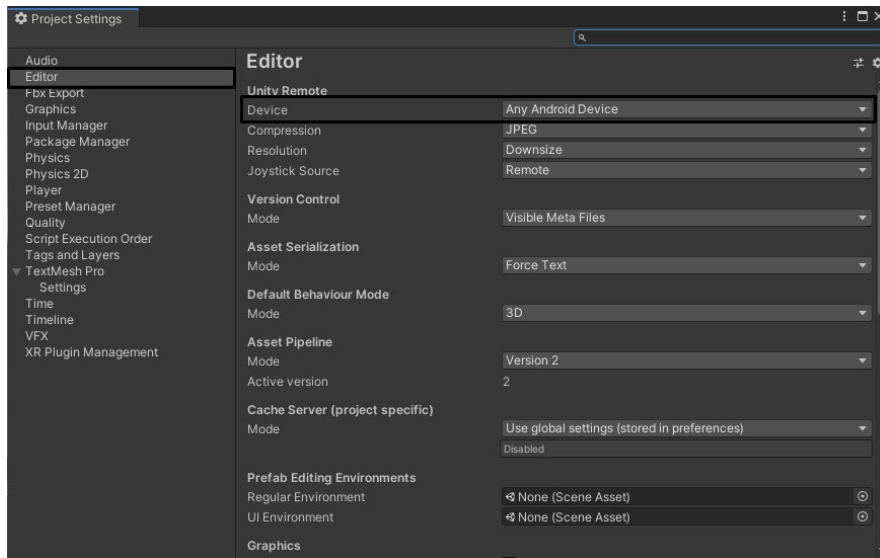


Εικόνα 3.10 Ρυθμίσεις Image Target

### 3.5 Ρυθμίσεις κινητού

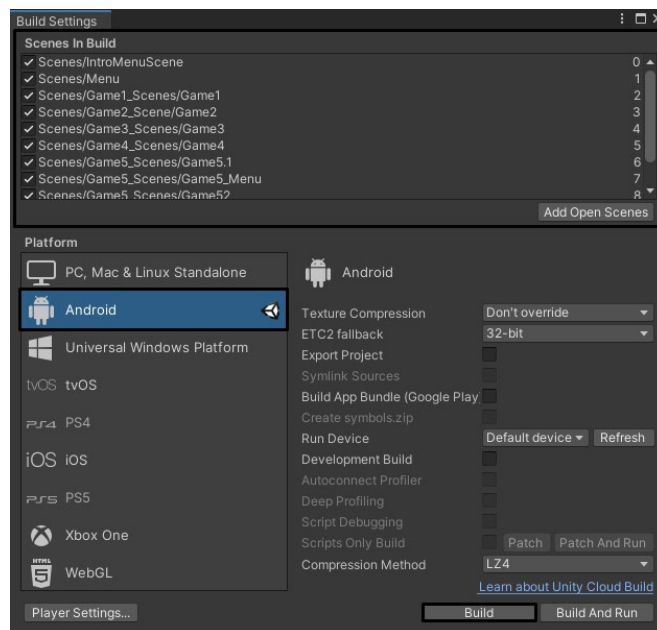
Στη συγκεκριμένη Δ.Ε. χρησιμοποιήθηκαν ρυθμίσεις κατάλληλες για Android συσκευές. Για να μπορέσει η εφαρμογή να λειτουργήσει σε κινητές συσκευές, ακολούθησαν κάποιες ρυθμίσεις στη συσκευή. Στις ρυθμίσεις της συσκευής → Σύστημα → Πληροφορίες για το τηλέφωνο → Αριθμός έκδοσης → πατώντας επάνω αρκετές φορές → Η λειτουργία προγραμματιστή έχει ενεργοποιηθεί. Στη συνέχεια, στις ρυθμίσεις της συσκευής → Σύστημα → Επιλογές για προγραμματιστές → Εντοπισμός σφαλμάτων USB → ON. Μετά την ολοκλήρωση των παραπάνω ρυθμίσεων, έπρεπε να γίνουν και

ρυθμίσεις στη Unity. Από το πάνω μέρος της Unity → Edit → Project Setting → Editor → Device → Any Android Device ( Εικόνα 3.11).



Εικόνα 3.11 Ρυθμίσεις κινητού στη Unity

Το τελευταίο βήμα για την εκτέλεση της εφαρμογής ήταν η σύνδεση της συσκευής με τον υπολογιστή και πηγαίνοντας στη Unity → File → Build Setting. Στο πάνω μέρος εμφανίζονται οι σκηνές της εφαρμογής, στην κατηγορία ‘Platform’ επιλέχθηκε το Android και τέλος, πατώντας το κουμπί Build γίνεται η μεταφορά της εφαρμογής στην Android συσκευή (Εικόνα 3.12). Για να γνωρίζει ο χρήστης ότι πραγματοποιήθηκε σωστά η σύνδεση της συσκευής και του Unity, στο κάτω μέρος της οθόνης της συσκευής, θα πρέπει να εμφανίζεται το λογότυπο ‘vuforia’.



Εικόνα 3.12 Εκτέλεση εφαρμογής

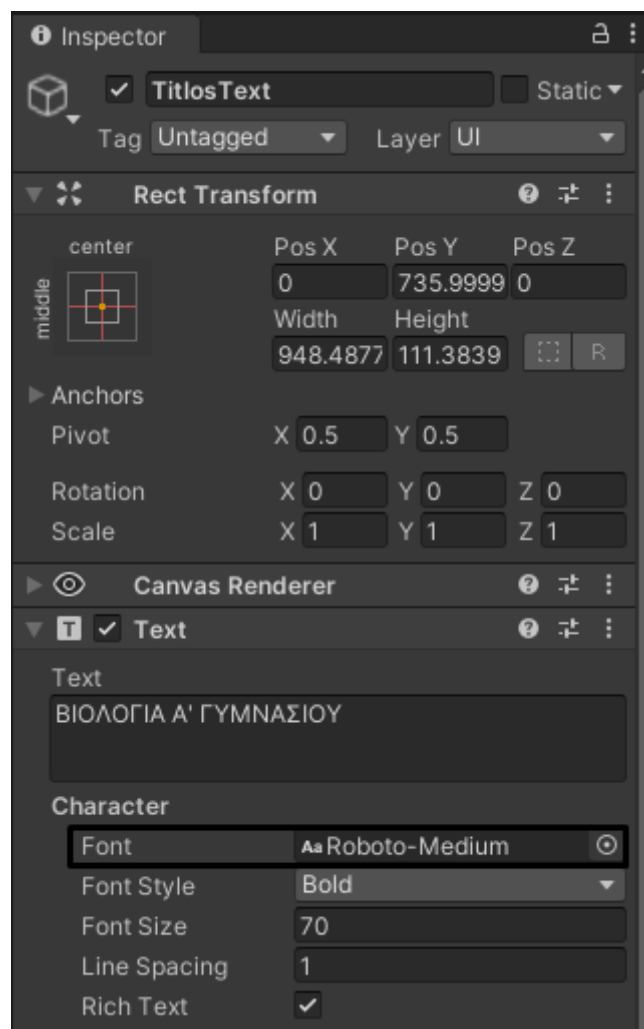
### **3.6 Επίλογος**

Πραγματοποιώντας τη σωστή εγκατάσταση των προγραμμάτων Unity, Blender και Vuforia και έχοντας ολοκληρώσει όλες τις απαραίτητες ρυθμίσεις, το πρόγραμμα είναι έτοιμο ώστε να ξεκινήσει η διαδικασία δημιουργίας των σκηνών του παιχνιδιού.



#### 4.4 Γραμματοσειρές και μουσική

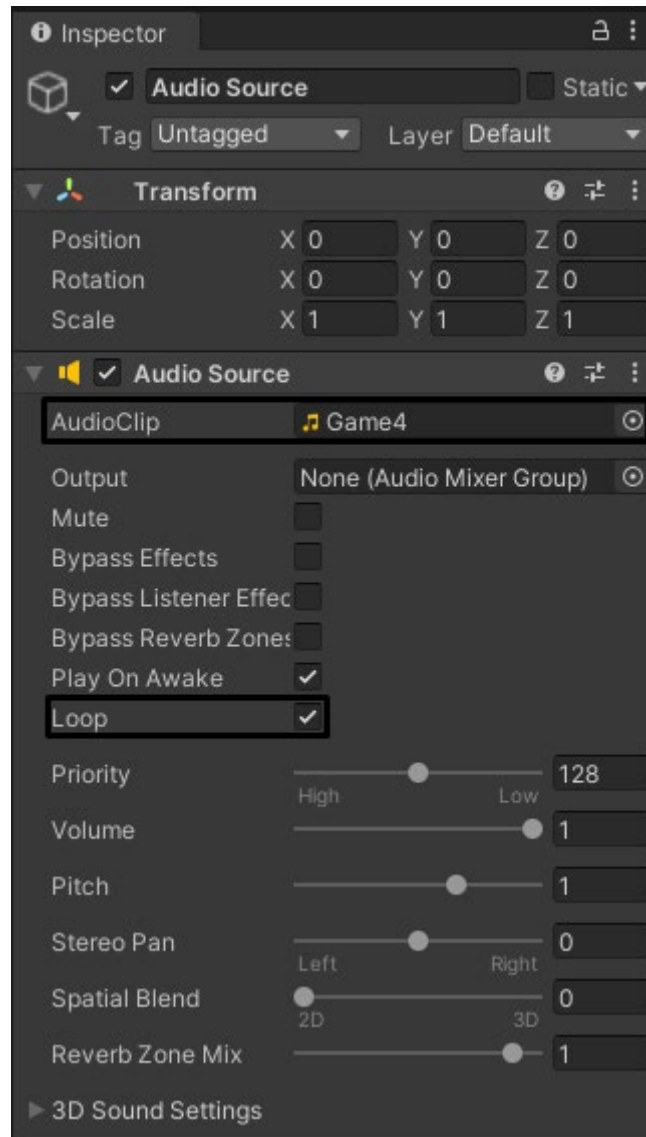
Σε όλες τις σκηνές της εφαρμογής είναι απαραίτητος ένας καμβάς. Ο καμβάς μπορεί να τοποθετηθεί κάνοντας δεξί κλικ στην αριστερή πλευρά της Unity στο Hierarchy και επιλέγοντας UI → Canvas. Μέσα στον καμβά, μπορούν να τοποθετηθούν όλα τα στοιχεία διεπαφής χρήστη. Ένα βασικό στοιχείο που τοποθετείται μέσα στον καμβά είναι τα κείμενα. Τα κείμενα τοποθετούνται από το μενού της Unity → GameObject → UI → Text. Τοποθετώντας ένα κείμενο μέσα στον καμβά και πατώντας πάνω σε αυτό, στην αριστερή πλευρά στον Inspector μπορεί να γίνει η επεξεργασία του. Στη συγκεκριμένη εφαρμογή, σε όλα τα κείμενα έγινε αλλαγή της γραμματοσειράς και χρησιμοποιήθηκε η ‘Roboto-Medium’ μέσω της ιστοσελίδας ‘Google Fonts’ [13]. Κατεβάζοντας το αρχείο – το οποίο αποθηκεύτηκε σε μορφή .ttf – τοποθετήθηκε μέσα στο φάκελο Fonts του Project. Με αυτόν τον τρόπο, έγινε η αλλαγή όλων των κειμένων της εφαρμογής, επιλέγοντας το αντικείμενο Inspector → Text → Character → Font → Roboto-Medium. (Εικόνα 4.2)



Εικόνα 4.2 Αλλαγή γραμματοσειράς

Σε όλη την εφαρμογή χρησιμοποιήθηκαν έξι αρχεία ήχου της μορφής .mp3, τα οποία έχουν τοποθετηθεί σε όλες τις σκηνές με τυχαία επιλογή. Αρχικά, δημιουργήθηκε στο Hierarchy ένα Empty Object και μέσα σε αυτό ένα Audio Source όπου βρίσκεται στο GameObject → Audio → Audio Source. Στο

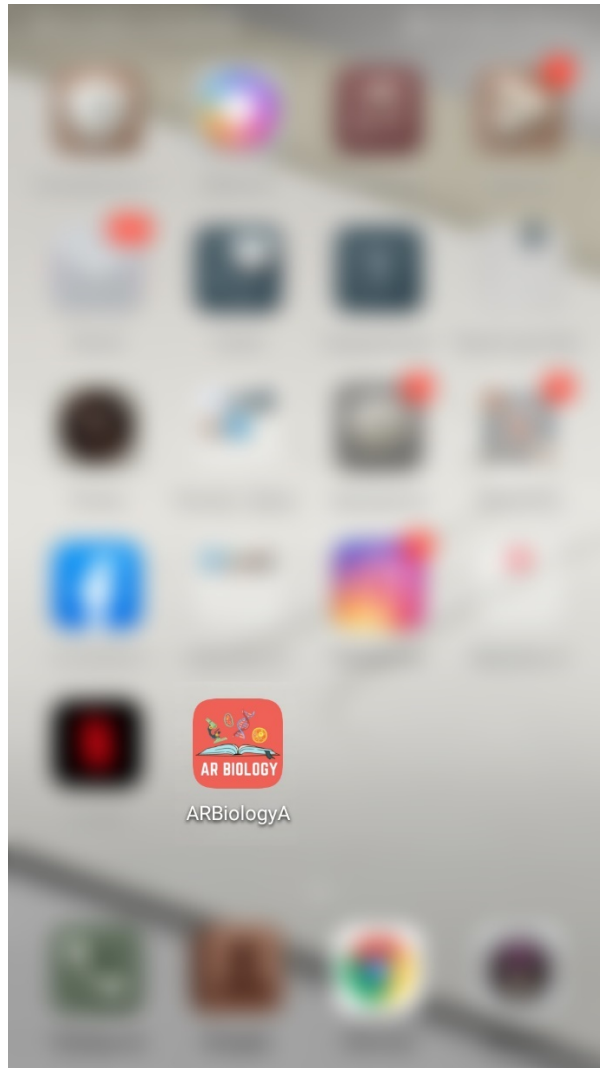
Inspector του Audio Source, τοποθετήθηκε στο Audio Source → AudioClip το αρχείο ήχου που επιλέχθηκε καθώς και το Loop ώστε ο ήχος να επαναλαμβάνεται σε περίπτωση που τελειώσει ενώ ο χρήστης βρίσκεται ακόμα μέσα στη σκηνή του παιχνιδιού (Εικόνα 4.3).



Εικόνα 4.3 Ρυθμίσεις ήχου

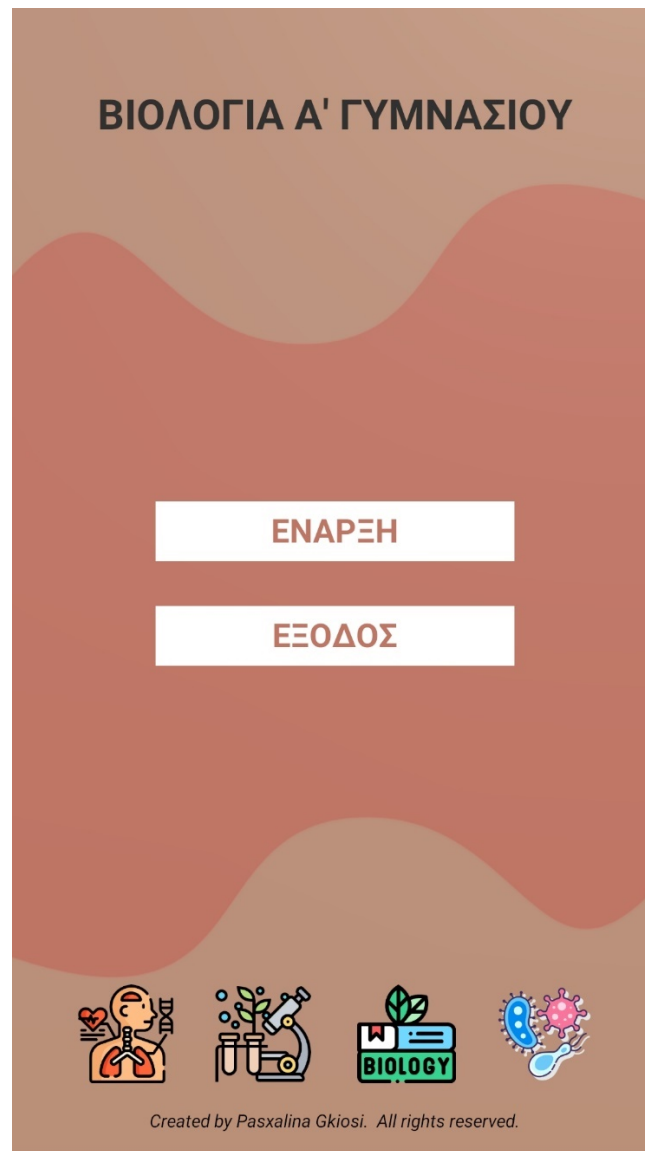
#### 4.5 Αρχική σκηνή εφαρμογής

Αφότου πραγματοποιήθηκαν με επιτυχία οι ρυθμίσεις στη συσκευή, στο μενού του κινητού εμφανίζεται το λογότυπο της εφαρμογής και η ονομασία της, ARBiologyA. (Εικόνα 4.4)



Εικόνα 4.4 Λογότυπο εφαρμογής

Το λογότυπο της εφαρμογής δημιουργήθηκε στην ιστοσελίδα 'Canva' [12] και τοποθετήθηκε μέσα από τη Unity ακολουθώντας τα βήματα από το μενού → Edit → Project Settings → Player → Icon → Adaptive. Από τη στιγμή που ο χρήστης πατήσει πάνω στο λογότυπο, ξεκινάει η εφαρμογή. Για μερικά δευτερόλεπτα εμφανίζεται το λογότυπο της Unity και στη συνέχεια η πρώτη σκηνή του παιχνιδιού. (Εικόνα 4.5)



Εικόνα 4.5 Πρώτη σκηνή εφαρμογής

Η πρώτη σκηνή αποτελείται από τον τίτλο, κουμπί 'έναρξη', κουμπί 'έξοδος' και από τέσσερα εικονίδια. Το κουμπί 'έναρξη' οδηγεί τον χρήστη στο κεντρικό μενού ενώ το κουμπί 'έξοδος' κλείνει την εφαρμογή. Για να ενεργοποιηθούν τα κουμπιά δημιουργήθηκε το script με όνομα 'IntroButton' (Εικόνα 4.6).

```

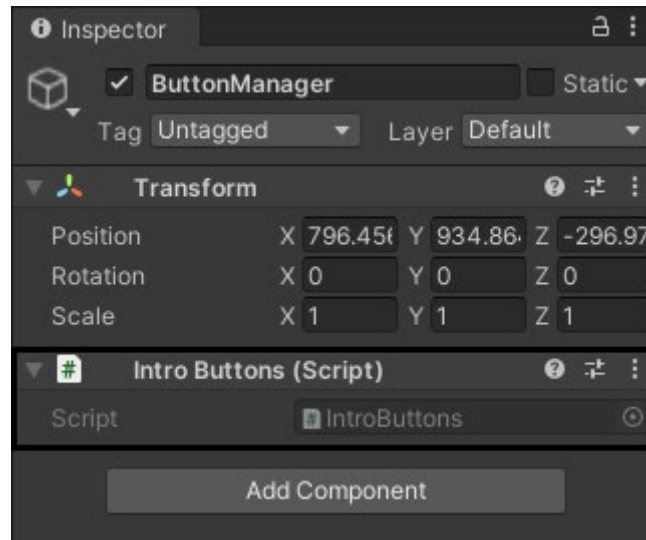
C# IntroButtons.cs X
C: > ARBiology > Assets > Scripts > C# IntroButtons.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using UnityEngine.SceneManagement;
6
7  public class IntroButtons : MonoBehaviour
8  {
9
10     public void OnClickStart() {
11         SceneManager.LoadScene("Menu");
12     }
13
14     public void OnClickExit() {
15         Application.Quit();
16     }
17 }
18
19

```

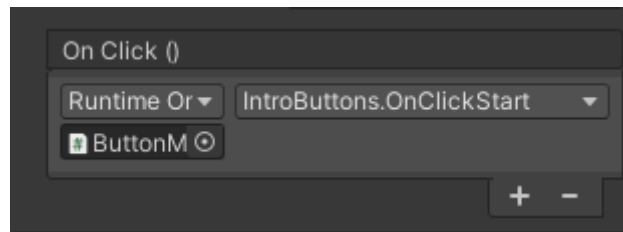
Εικόνα 4.6 Βασικές λειτουργίες του script 'IntroButtons'

Με τη βοήθεια της εντολής `SceneManager.LoadScene`, μόλις πατηθεί το κατάλληλο κουμπί, καλείται η σκηνή με το όνομα που βρίσκεται στην παρένθεση. Η μέθοδος `onClickExit` συνδέεται με το κουμπί 'έξοδος' και γίνεται ο τερματισμός της εφαρμογής με τη βοήθεια της εντολής `Application.Quit()`.

Όλα τα κουμπιά της εφαρμογής για να μπορέσουν να ενεργοποιηθούν, έπρεπε πρώτα να δημιουργηθούν μέθοδοι μέσα στα script, κατάλληλοι για το καθένα. Αναλυτικότερα, για να μπορέσουν τα scripts, οι μέθοδοι και τα κουμπιά να επικοινωνήσουν μεταξύ τους, έπρεπε να δημιουργηθούν κενά αντικείμενα. Τα κενά αντικείμενα ή αλλιώς `Empty`, τοποθετούνται από το μενού `→ GameObject → Create Empty`. Μέσα στα κενά αντικείμενα γίνεται η τοποθέτηση του κατάλληλου script (Εικόνα 4.7) και έπειτα στο Inspector ενός κουμπί στο πλαίσιο `On Click()`, τοποθετείται στο `None (object)` το κενό αντικείμενο που δημιουργήθηκε. Μόλις τοποθετηθεί, στο πλαίσιο `No Function` εμφανίζεται το όνομα του script και οι μέθοδοι που έχουν δημιουργηθεί μέσα σε αυτό (Εικόνα 4.8). Με αυτόν τον τρόπο, όταν ο χρήστης πατήσει ένα κουμπί, καλείτε η μέθοδος που επιλέχθηκε και πραγματοποιούνται οι ενέργειες της.



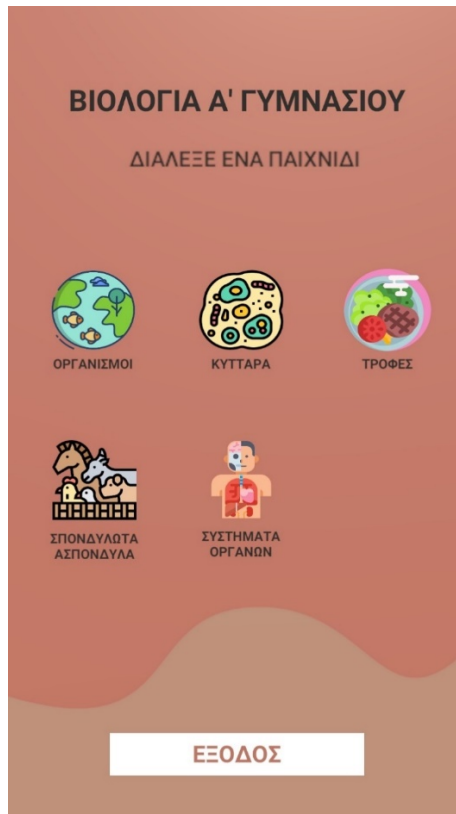
Εικόνα 4.7 Κενά αντικείμενα στο Unity



Εικόνα 4.8 Επικοινωνία κουμπιών και scripts

## 4.6 Μενού εφαρμογής

Όπως αναφέρθηκε παραπάνω, πατώντας το κουμπί ‘έναρξη’ εμφανίζεται το κεντρικό μενού της εφαρμογής (Εικόνα 4.9). Το μενού αποτελείται από τον τίτλο, το κουμπί ‘έξοδος’ και από τα πέντε παιχνίδια. Η ενεργοποίηση των κουμπιών γίνεται με τη βοήθεια του script με όνομα ‘MenuButtons’ (Εικόνα 4.10).



Εικόνα 4.9 Μενού εφαρμογής

```
MenuButton.cs X
C: > ARBiology > Assets > Scripts > MenuButton.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using UnityEngine.SceneManagement;
6
7 public class MenuButton : MonoBehaviour
8 {
9     public void OnClickStartGame1() {
10         SceneManager.LoadScene("Game1");
11     }
12
13     public void OnClickStartGame2() {
14         SceneManager.LoadScene("Game2");
15     }
16
17     public void OnClickStartGame3() {
18         SceneManager.LoadScene("Game3");
19     }
20
21     public void OnClickStartGame4() {
22         SceneManager.LoadScene("Game4");
23     }
24
25     public void OnClickStartGame5() {
26         SceneManager.LoadScene("Game5_Menu");
27     }
28
29     public void OnClickExit() {
30         Application.Quit();
31     }
32 }
33
```

Εικόνα 4.10 Βασικές λειτουργίες του script 'MenuButton'

Για κάθε παιχνίδι έχει δημιουργηθεί μια μέθοδος όπως και για το κουμπί εξόδου. Οι μέθοδοι είναι αντίστοιχοι με αυτών του script 'IntroButton', όπου αναλύθηκε παραπάνω.

#### **4.7 Επίλογος**

Μετά την ολοκλήρωση αυτού του κεφαλαίου, γίνεται κατανοητός ο τρόπος αλληλεπίδρασης του χρήστη με την εφαρμογή. Με τη βοήθεια των κενών αντικειμένων, γίνεται η επικοινωνία ανάμεσα στα scripts και στα κουμπιά και αυτός ο τρόπος χρησιμοποιήθηκε σε ολόκληρη τη Δ.Ε. Η εμφάνιση της αρχικής εισόδου και του μενού είναι πολύ σημαντική, διότι είναι οι πρώτες σκηνές που αντικρίζει ένας χρήστης ανοίγοντας την εφαρμογή.



## Κεφάλαιο 5ο: Οργανισμοί

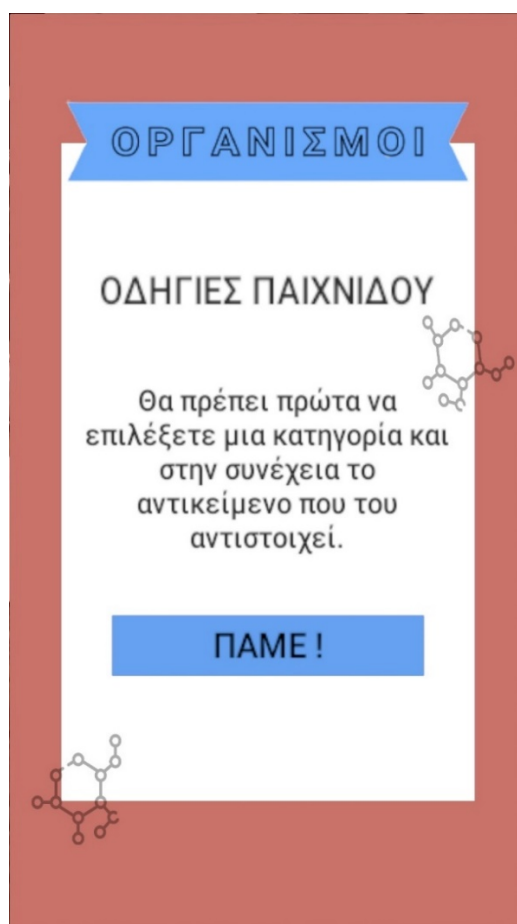
### 5.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφεται το πρώτο παιχνίδι της εφαρμογής της υπό-ενότητας ‘Τα χαρακτηριστικά των οργανισμών’. Το συγκεκριμένο παιχνίδι αποτελείται από οκτώ αντικείμενα τα οποία πρέπει να τοποθετηθούν στις σωστές κατηγορίες. Επιπλέον, παρουσιάζονται αναλυτικά όλα τα βήματα που ακολουθήθηκαν και οι λειτουργίες που χρησιμοποιήθηκαν για την υλοποίηση του παιχνιδιού.

### 5.2 Ανάλυση παιχνιδιού

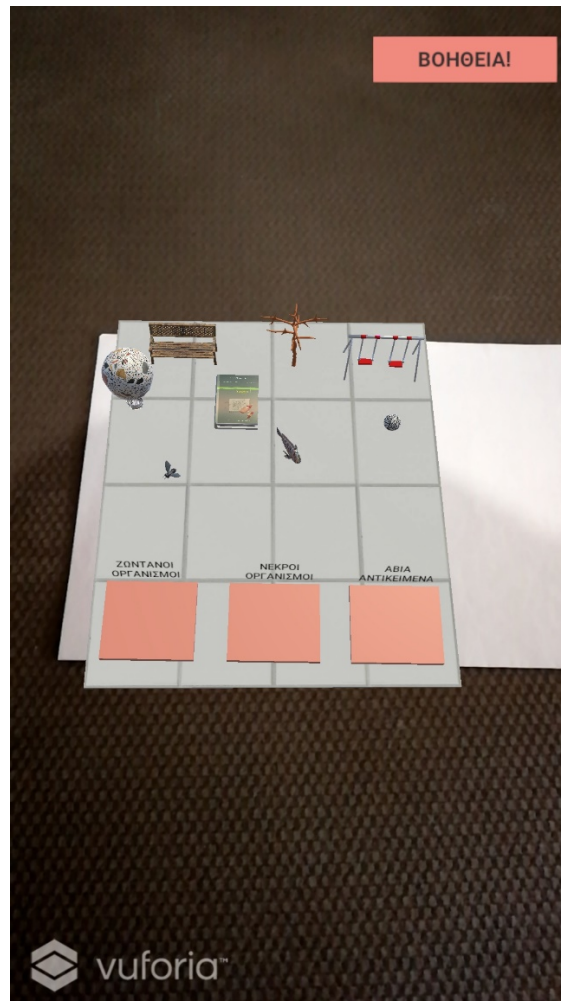
Το πρώτο παιχνίδι που δημιουργήθηκε ανήκει στην πρώτη ενότητα του σχολικού βιβλίου ‘Οργάνωση της ζωής’, στο υπό-κεφάλαιο 1.1 ‘Τα χαρακτηριστικά των οργανισμών’ και είναι η άσκηση τέσσερα της σελίδας 20-21. Σκοπός του παιχνιδιού είναι οι μαθητές να κατανοήσουν τις έννοιες των ζωντανών και νεκρών οργανισμών όπως και των αντικειμένων που δεν υπήρξαν ποτέ ζωντανά και ονομάζονται άβια. Το παιχνίδι αποτελείται από τις οδηγίες του παιχνιδιού, το παιχνίδι και το αποτέλεσμα.

Μόλις πατηθεί το εικονίδιο με όνομα ‘Οργανισμοί’ από το μενού του παιχνιδιού, εμφανίζεται μια δισδιάστατη σκηνή. Αυτή η σκηνή αποτελείται από τον τίτλο, δυο εικόνες, τις οδηγίες του παιχνιδιού και ένα κουμπί με όνομα ‘Πάμε’ (Εικόνα 5.1)



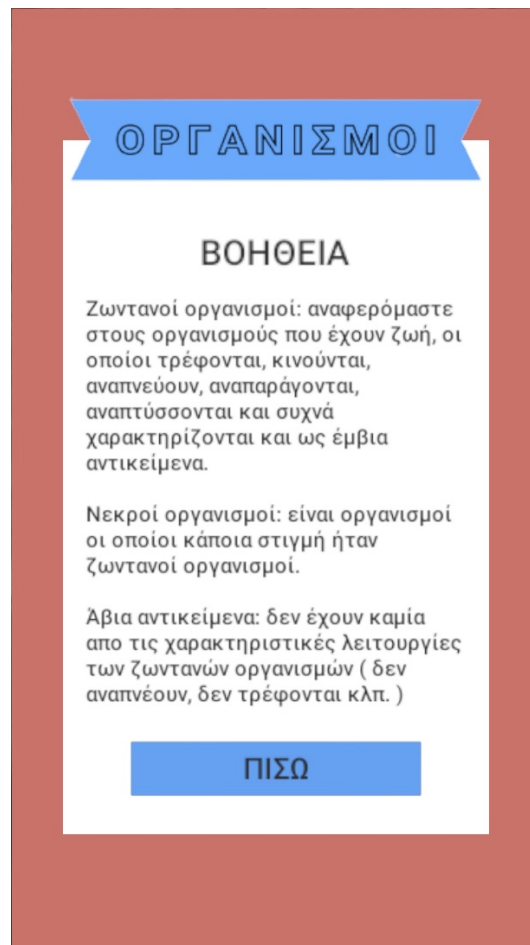
Εικόνα 5.1 Έναρξη παιχνιδιού 'Οργανισμοί'

Μόλις ο χρήστης πατήσει το κουμπί 'Πάμε' μεταφέρεται στην επόμενη και βασική σκηνή, το παιχνίδι. Αυτόματα στην οθόνη της συσκευής ενεργοποιείται η κάμερα και ο χρήστης πρέπει να μεταφερθεί στην σελίδα 21 του σχολικού βιβλίου ώστε να σκανάρει το Qr code. Σκανάροντας το Qr code, εμφανίζεται απευθείας το περιβάλλον επαυξημένης πραγματικότητας. Στο πάνω μέρος του περιβάλλοντος εμφανίζονται οκτώ αντικείμενα και στο κάτω μέρος τρία πλαίσια τοποθέτησης με τον αντίστοιχο τίτλο του καθενός (Εικόνα 5.2)



Εικόνα 5.2 Εμφάνιση παιχνιδιού 'Όργανισμοί'

Στην οθόνη, στο δεξιό πάνω μέρος είναι τοποθετημένο ένα κουμπί με όνομα 'Βοήθεια'. Αυτό το κουμπί μπορεί να πατηθεί σε όλη την διάρκεια του παιχνιδιού, σε περίπτωση που ο μαθητής θελήσει να ξαναδιαβάσει τη θεωρία του μαθήματος. Πατώντας το κουμπί 'Πίσω', εμφανίζεται ξανά το επαυξημένο περιβάλλον και μπορεί να επιστρέψει στο παιχνίδι (Εικόνα 5.3).



Εικόνα 5.3 Κουμπί βοήθειας παιχνιδιού 'Οργανισμοί'

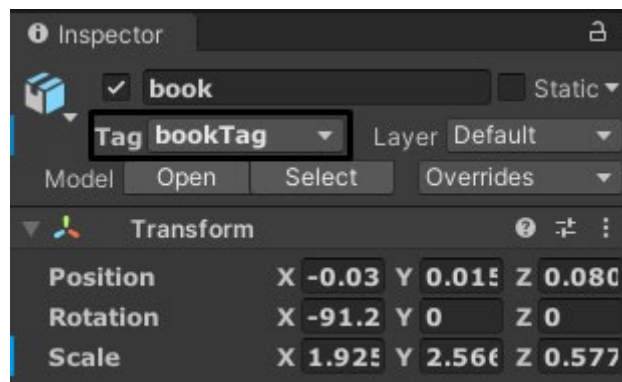
Η υλοποίηση του παιχνιδιού πραγματοποιείται με το script 'event' και 'Game1'. Το script 'Game1', είναι το βασικό script του παιχνιδιού και στο πάνω μέρος έχουν δηλωθεί όλες οι μεταβλητές που χρησιμοποιήθηκαν (Εικόνα 5.4). Οι μεταβλητές είναι τύπου GameObject δηλαδή τρισδιάστατα αντικείμενα, κείμενα, bool δηλαδή true ή false, ακέραιο για την μέτρηση των σωστών επιλογών και τύπου 'tag'. Το tag τοποθετείται πάνω στο αντικείμενο από το Inspector → Tag → Add Tag, ως αναγνωριστικό για να γίνουν μερικοί έλεγχοι που θα αναλυθούν παρακάτω (Εικόνα 5.5).

```

32 [SerializeField] private string aerostatoTag = "aerostatoTag";
33 [SerializeField] private string fishTag = "fishTag";
34 [SerializeField] private string mpalaTag = "mpalaTag";
35 [SerializeField] private string pagkakiTag = "pagkakiTag";
36 [SerializeField] private string kouniaTag = "kouniaTag";
37 [SerializeField] private string dentroTag = "dentroTag";
38 [SerializeField] private string bookTag = "bookTag";
39 [SerializeField] private string beeTag = "beeTag";
40
41 public bool AbiaCheck = false;
42 public bool NekroiCheck = false;
43 public bool ZwntanoiCheck = false;
44
45 public Text TapInfoText;
46 private int counterSostes;
47 public GameObject PanelStart;
48 public GameObject PanelHelp;
49 public GameObject InfoButton;
50 public GameObject EndGameCanvas;

```

Εικόνα 5.4 Μεταβλητές script 'Game1'



Εικόνα 5.5 Tags στα τρισδιάστατα αντικείμενα

Όλες οι λειτουργίες που πρέπει να εκτελούνται και να ελέγχονται καθ' όλη τη διάρκεια του παιχνιδιού, τοποθετούνται μέσα στην μέθοδο Update(). Ο πρώτος έλεγχος γίνεται με τη βοήθεια της εντολής GetMouseButtonDown, η οποία ελέγχει αν ο χρήστης έχει πατήσει πάνω στην οθόνη. Έπειτα, στην μεταβλητή με όνομα Ray τοποθετείται το ακριβές σημείο που πάτησε ο χρήστης. Στη συνέχεια, μια δεύτερη if ελέγχει αν το πάτημα που έγινε στην οθόνη πραγματοποιήθηκε σε κάποιο αντικείμενο στο οποίο έχει τοποθετηθεί η ιδιότητα Box Collider (Εικόνα 5.6).

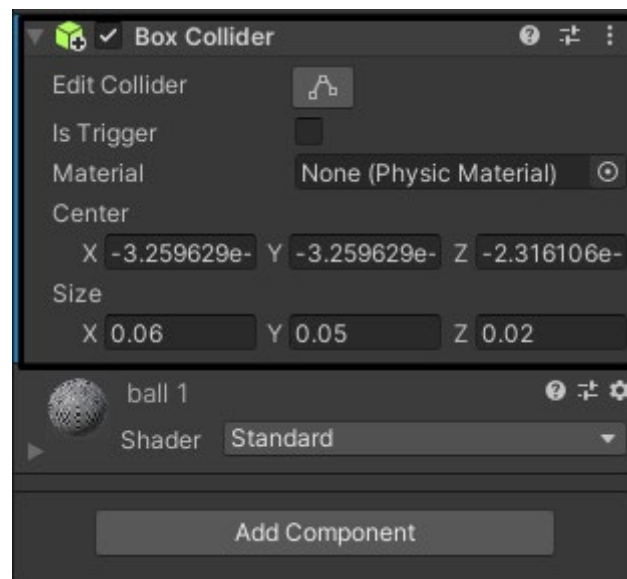
```

51 void Update()
52 {
53     if (Input.GetMouseButtonDown(0)) {
54
55         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
56         RaycastHit hit;
57
58         if (Physics.Raycast(ray, out hit))
59         {
60
61             var selection = hit.transform;
62

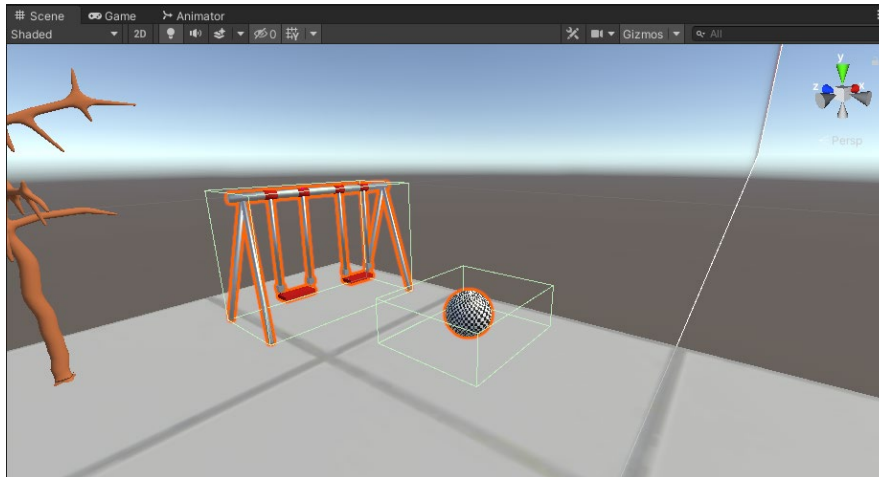
```

Εικόνα 5.6 Εντολή Input.GetMouseButtonDown

Η ιδιότητα Box Collider, τοποθετείται στα αντικείμενα από το Inspector → Add Component → Box Collider (Εικόνα 5.7) και τότε γύρω από το αντικείμενο εμφανίζεται ένα πράσινο περίγραμμα (Εικόνα 5.8). Αυτό το περίγραμμα πρέπει να διαμορφωθεί ανάλογα με τα όρια που χρειάζεται το κάθε αντικείμενο.



Εικόνα 5.7 Τοποθέτηση ιδιότητας Box Collider



Εικόνα 5.8 Αντικείμενα με ιδιότητα Box Collider

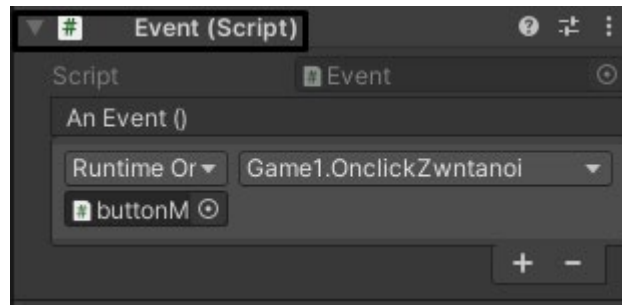
Οι επόμενοι έλεγχοι που γίνονται αφορούν την ορθή ή λανθασμένη επιλογή του χρήστη. Με άλλα λόγια, γίνεται έλεγχος αν ο χρήστης έχει επιλέξει τη σωστή κατηγορία και το σωστό αντικείμενο με βάση το tag που έχει οριστεί σε αυτό και τότε πραγματοποιούνται κάποιες ενέργειες. Οι κατηγορίες που έχουν δημιουργηθεί έχουν την μορφή cube και τέτοιου είδους αντικείμενα δεν μπορούν να αλληλεπιδρούν με τον χρήστη. Γι' αυτόν τον λόγο, δημιουργήθηκε το script με όνομα 'Event' (Εικόνα 5.9), όπου δίνει την δυνατότητα στα 'κουτιά' κατηγοριών να συμπεριφέρονται σαν κουμπιά (Εικόνα 5.10).

```

C# Game1.cs  C# Event.cs  X
C: > ARBiology > Assets > Scripts > Game1 > C# Event.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.Events;
5
6  public class Event : MonoBehaviour
7  {
8      [SerializeField] UnityEvent anEvent;
9
10     private void EventMethod()
11     {
12         anEvent.Invoke();
13     }
14 }
15

```

Εικόνα 5.9 Script 'Event'



Εικόνα 5.10 Ρυθμίσεις σύνδεσης για το script 'Event'

Αν η επιλογή του χρήστη είναι σωστή, τότε με τη βοήθεια της εντολής `transform.localPosition` το αντικείμενο τοποθετείται στο ροζ πλαίσιο. Επιπρόσθετα, με την εντολή `hit.collider.enabled` ο χρήστης δεν έχει την δυνατότητα να ξαναπατήσει το ίδιο αντικείμενο από την στιγμή που του έχει δοθεί η ιδιότητα `false`. Σε κάθε περίπτωση, εμφανίζεται το αντίστοιχο μήνυμα σωστής ή λάθος επιλογής με τη βοήθεια της μεθόδου `Invoke` η οποία καλείται μετά από τα δευτερόλεπτα που έχουν οριστεί (Εικόνα 5.11). Επιπρόσθετα, τα cube-κουμπιά που έχουν δημιουργηθεί κάθε φορά που τα επιλέγει ο χρήστης ενεργοποιούνται και είτε επιλέξει τη σωστή απάντηση είτε τη λάθος, πρέπει να απενεργοποιηθούν. Η ενεργοποίηση πραγματοποιείται με τρεις μεθόδους, μια για το κάθε cube-πλαίσιο (Εικόνα 5.12)

```

65         if (selection.CompareTag(aerostatoTag) && (AbiaCheck)) {
66
67             Aerostato.transform.localPosition = new Vector3(0.13f, 0.0648f, -0.13f);
68             AbiaCheck = false;
69             hit.collider.enabled = false;
70             TapInfoText.text = "ΣΩΣΤΗ ΕΠΙΛΟΓΗ!";
71             Invoke("EnableText",1f);
72             Invoke("DisableText",3f);
73
74             counterSostes ++;
75
76         } else if (selection.CompareTag(aerostatoTag) && ((ZwntanoiCheck) || (NekroiCheck))) {
77             TapInfoText.text = "ΛΑΘΟΣ ΕΠΙΛΟΓΗ!";
78             Invoke("EnableText",1f);
79             Invoke("DisableText",2.0f);
80             ZwntanoiCheck = false;
81             NekroiCheck = false;
82         }
83     }

```

Εικόνα 5.11 Έλεγχος σωστής ή λάθος απάντησης στο script 'Game1'

```

241     public void OnclickAbia() {
242         AbiaCheck = true;
243     }
244     public void OnclickNekroi() {
245         NekroiCheck = true;
246     }
247     public void OnclickZwntanoi() {
248         ZwntanoiCheck = true;
249     }

```

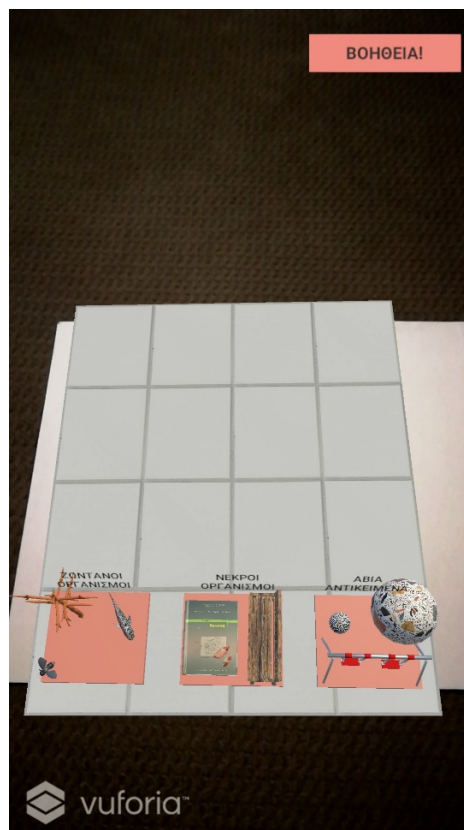
Εικόνα 5.12 Μέθοδοι για την ενεργοποίηση των cube-πλαισίων

## Κεφάλαιο 5

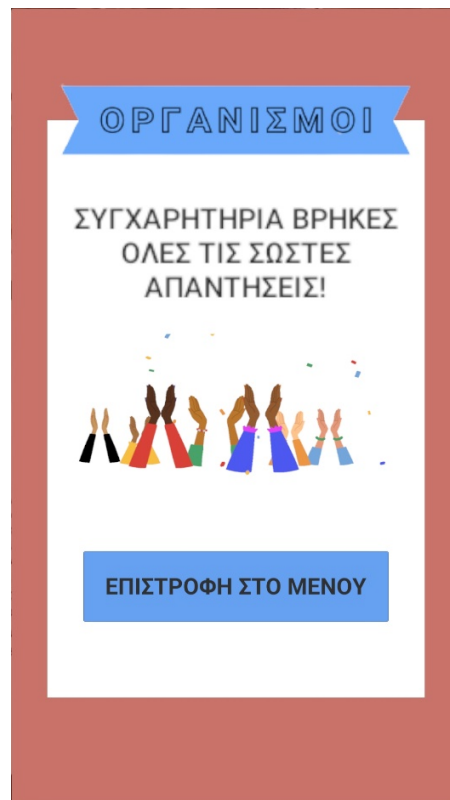
Σε κάθε σωστή απάντηση έχει οριστεί ένας μετρητής ο οποίος όταν φθάσει στον αριθμό οκτώ (Εικόνα 5.13) δηλαδή όταν όλα τα αντικείμενα έχουν τοποθετηθεί στις σωστές κατηγορίες (Εικόνα 5.14), τότε εμφανίζεται η τελευταία σκηνή του παιχνιδιού (Εικόνα 5.15). Η τελευταία σκηνή περιλαμβάνει ένα κείμενο δίνοντας συγχαρητήρια στον μαθητή για την επιτυχής ολοκλήρωση του παιχνιδιού και ένα κουμπί με όνομα 'επιστροφή στο μενού' όπου πατώντας το, ο χρήστης επιστρέφει στο κεντρικό μενού.

```
228         if (counterSostes >= 8) {  
229             counterSostes =0;  
230             TapInfoText.text = "ΤΕΛΟΣ ΠΑΙΧΝΙΔΙΟΥ";  
231             Invoke("EnableText",1f);  
232             Invoke("EndGame", 2f);  
233         }  
234     } //void update  
235
```

Εικόνα 5.13 Μετρητής για το τέλος του πρώτου παιχνιδιού



Εικόνα 5.14 Σωστή τοποθέτηση αντικειμένων στο πρώτο παιχνίδι



Εικόνα 5.15 Τελευταία σκηνή πρώτου παιχνιδιού

### 5.3 Επίλογος

Μετά την πραγματοποίηση των παραπάνω ενεργειών, ολοκληρώθηκε η υλοποίηση του πρώτου παιχνιδιού για το υπό-κεφάλαιο ‘Τα χαρακτηριστικά των οργανισμών’. Μέσω αυτού του παιχνιδιού, η μάθηση των ζωντανών, νεκρών και άβιων αντικειμένων γίνεται με έναν πιο ευχάριστο και διασκεδαστικό τρόπο.



## Κεφάλαιο 6ο: Κύτταρα

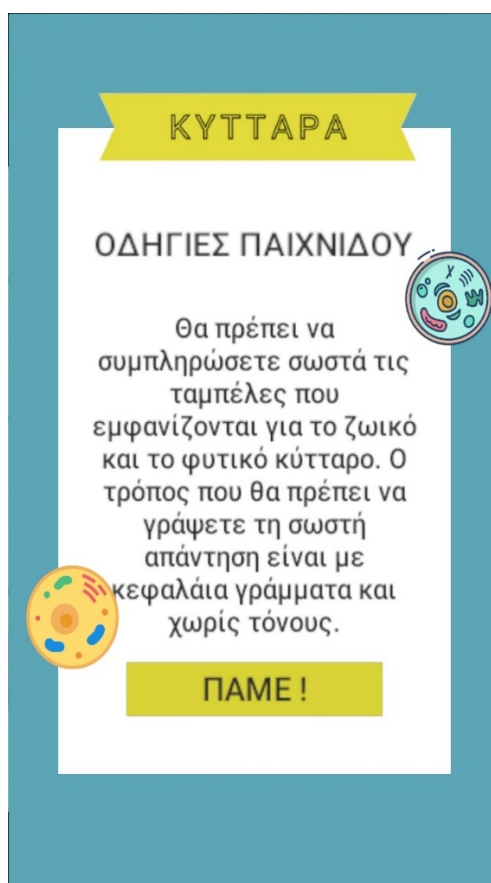
### 6.1 Εισαγωγή

Το δεύτερο παιχνίδι της εφαρμογής ανήκει στην υπό-ενότητα ‘Κύτταρα: η μονάδα της ζωής’ και θα αναλυθεί σε αυτό το κεφάλαιο. Το παιχνίδι περιλαμβάνει το ζωικό και το φυτικό κύτταρο και ο χρήστης καλείται να συμπληρώσει τις ταμπέλες με τη σωστή ονομασία. Επίσης, γίνεται η λεπτομερή ανάλυση του παιχνιδιού και παρουσιάζονται τα βήματα που ακολουθήθηκαν για την ολοκλήρωσή του.

### 6.2 Ανάλυση παιχνιδιού

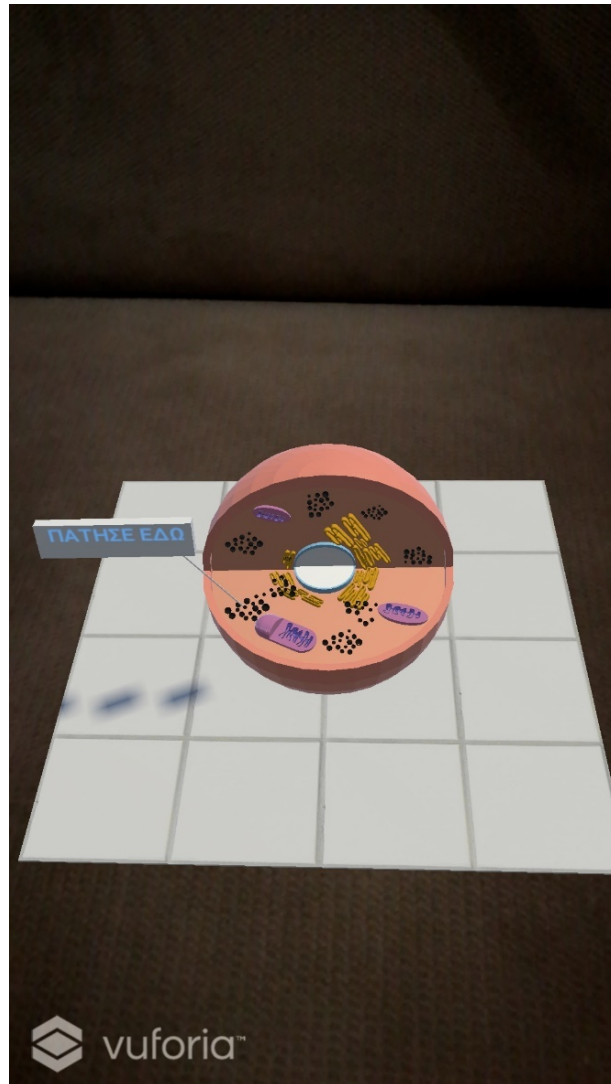
Το δεύτερο παιχνίδι καλύπτει την ύλη της ενότητας του σχολικού βιβλίου ‘Οργάνωση της ζωής’, στο υπό-κεφάλαιο 1.2 ‘Κύτταρα: η μονάδα της ζωής’ των σελίδων 22-23. Στόχος του παιχνιδιού είναι οι μαθητές να συμπληρώσουν σωστά τις ταμπέλες που εμφανίζονται σχετικά με το ζωικό και το φυτικό κύτταρο. Επιπλέον, με την επιτυχή συμπλήρωση των ταμπελών, εμφανίζονται πληροφορίες σχετικά με την εσωτερική οργάνωση των κυττάρων. Το παιχνίδι περιλαμβάνει την πρώτη σκηνή των οδηγιών, το βασικό παιχνίδι που αποτελείται από τα δυο κύτταρα και την τελευταία σκηνή των αποτελεσμάτων.

Στο αρχικό μενού του παιχνιδιού υπάρχει το κουμπί με όνομα ‘Κύτταρα’ και μόλις το πατήσει ο χρήστης εμφανίζεται μια σκηνή με δισδιάστατο περιεχόμενο. Στη σκηνή αυτή, στο πάνω μέρος υπάρχει ο τίτλος, δυο εικόνες σχετικές με το περιεχόμενο του παιχνιδιού και στη μέση οι οδηγίες. Στο κάτω μέρος της οθόνης υπάρχει το κουμπί ‘Πάμε’ το οποίο μόλις πατηθεί γίνεται η έναρξη του παιχνιδιού (Εικόνα 6.1).



Εικόνα 6.1 Έναρξη παιχνιδιού 'Κύτταρα'

Για να ξεκινήσει η έναρξη του παιχνιδιού, θα πρέπει ο χρήστης να μεταφερθεί στη σελίδα 22 του σχολικού βιβλίου και να σκανάρει το Qr code που αντιστοιχεί στο συγκεκριμένο παιχνίδι. Μόλις σκανάρει το Qr code, εμφανίζεται στην οθόνη του κινητού του το περιβάλλον επαυξημένης πραγματικότητας. Το περιβάλλον περιλαμβάνει το ζωικό κύτταρο μαζί με τις εσωτερικές λεπτομέρειες του, οι οποίες έχουν δημιουργηθεί όλες μέσα από το πρόγραμμα Blender. Επιπλέον, εμφανίζεται η πρώτη ταμπέλα με το κείμενο 'Πατήστε εδώ' (Εικόνα 6.2).



Εικόνα 6.2 Εμφάνιση παιχνιδιού 'Κύτταρο' - Ζωικό κύτταρο

Η πραγματοποίηση του παιχνιδιού γίνεται με τα script 'FutikoKuttaroTextControl', 'Information' και 'TextControl'. Το script με όνομα 'TextControl' είναι το πρώτο script που καλείται από την έναρξη του παιχνιδιού έως και την ολοκλήρωση του ζωικού κυττάρου. Στο πάνω μέρος του script έχουν δηλωθεί όλες οι μεταβλητές που χρησιμοποιήθηκαν και είναι μεταβλητές τύπου GameObject, κειμένου, tag και InputField (Εικόνα 6.3).

```

14 public GameObject ribosomataObject;
15 public GameObject pirinasObject;
16 public GameObject plasmatikiMembraniObject;
17 public GameObject kuttaroplasmaObject;
18 public GameObject mitoxondrioObject;
19 public GameObject answerPanel;
20 public GameObject PanelApotelesma;
21 public GameObject PanelStart;
22
23 public GameObject InformationPanelRivosomata;
24 public GameObject InfomationPanelPirinas;
25 public GameObject InfomationPanelPM;
26 public GameObject InfomationPanelKuttaroplasma;
27 public GameObject InfomationPanelMitoxondrio;
28
29 public InputField answerText;
30 public Text TextApotelesma;
31
32 //Tags
33 [SerializeField] private string ribosomataTag = "ribosomataTag";
34 [SerializeField] private string pirinasTag = "pirinasTag";
35 [SerializeField] private string plasmatikiMembraniTag = "plasmatikiMembraniTag";
36 [SerializeField] private string kuttaroplasmaTag = "kuttaroplasmaTag";
37 [SerializeField] private string mitoxondrioTag = "mitoxondrioTag";

```

Εικόνα 6.3 Μεταβλητές Script 'Text Control'

Ένα αντικείμενο InputField τοποθετείται από το μενού → GameObject → UI → Input Field. Είναι ένα πλαίσιο όπου ο χρήστης πατώντας πάνω σε αυτό ανοίγει το πληκτρολόγιο και μπορεί να γράψει κείμενο. Μέσα στην μέθοδο Update(), γίνεται ο έλεγχος για το ποια ταμπέλα έχει πατήσει ο χρήστης. Οι ταμπέλες μαζί με τα κείμενα είναι τρισδιάστατα αντικείμενα και έχουν τοποθετηθεί σε αυτά η ιδιότητα tag καθώς και Box Collider ώστε να μπορεί ο χρήστης να πατήσει πάνω σε αυτά και για γίνει κάποια ενέργεια, τα οποία αναλύθηκαν λεπτομερώς στο προηγούμενο κεφάλαιο. Μόλις ο χρήστης πατήσει πάνω στην ταμπέλα, εμφανίζεται ένας δισδιάστατος καμβάς. Ο καμβάς περιλαμβάνει στο πάνω μέρος μια ερώτηση, στη μέση υπάρχει το πλαίσιο InputField και στο κάτω μέρος ένα κουμπί με όνομα 'Επιλογή' (Εικόνα 6.4).



ΓΡΑΨΕ ΤΗ ΛΕΞΗ ΠΟΥ  
ΠΙΣΤΕΥΕΙΣ ΠΩΣ ΑΝΤΙΣΤΟΙΧΕΙ  
ΣΕ ΑΥΤΟ ΠΟΥ ΕΠΕΛΕΞΕΣ

Δώσε την σωστή απάντηση...

ΕΠΙΛΟΓΗ

Εικόνα 6.4 Πλαίσιο για την τοποθέτηση της απάντησης

Πατώντας το κουμπί επιλογή καλείται η μέθοδος `onClickRibosomataBt` (Εικόνα 6.5). Αρχικά, αυτή η μέθοδος ελέγχει την απάντηση που έγραψε ο χρήστης στο πλαίσιο με την εντολή `answerText.text` καθώς και ελέγχεται το αντικείμενο με την εντολή `activeSelf`. Η εντολή αυτή ελέγχει αν το αντικείμενο είναι ενεργοποιημένο εκείνη τη στιγμή. Μόλις ένα συγκεκριμένο αντικείμενο-ταμπέλα είναι ενεργοποιημένο και δοθεί μια απάντηση, τότε εμφανίζεται το κατάλληλο μήνυμα. Αν η απάντηση είναι σωστή, εμφανίζεται ένας δισδιάστατος καμβάς με αναλυτικές πληροφορίες σχετικά με τη σωστή απάντηση που έδωσε ο χρήστης και ένα κουμπί με όνομα 'Επόμενο' (Εικόνα 6.6). Το script `Information` είναι υπεύθυνο για την απενεργοποίηση όλων των δισδιάστατων καμβάδων και πατώντας το κουμπί 'Επόμενο', καλείται η αντίστοιχη μέθοδος κλείνοντας το και εμφανίζοντας το επαυξημένο περιβάλλον (Εικόνα 6.7). Ταυτόχρονα, στο πίσω μέρος κλείνει και το αντίστοιχο πλαίσιο 'Πατήστε εδώ' και εμφανίζεται το επόμενο πλαίσιο που πρέπει να επιλέξει ο χρήστης.

```

75 public void onClickRibosomataBt() {
76
77     if (answerText.text == "ΡΙΒΟΣΩΜΑΤΑ") {
78
79         TextApootelesma.text = "ΜΠΡΑΒΟ, ΣΟΣΤΗ ΑΠΑΝΤΗΣΗ !";
80         Invoke("EnableText",1f);
81         Invoke("DisableText",5.0f);
82         ribosomataObject.SetActive(false);
83         answerText.text = "";
84         InformationPanelRivosomata.SetActive(true);
85         //energopoeisi pirina
86         pirinasObject.SetActive(true);
87
88     } else if (answerText.text == "ΠΥΡΗΝΑΣ" && pirinasObject.activeSelf) {
89         TextApootelesma.text = "ΜΠΡΑΒΟ, ΣΟΣΤΗ ΑΠΑΝΤΗΣΗ !";
90         Invoke("EnableText",1f);
91         Invoke("DisableText",5.0f);
92         pirinasObject.SetActive(false);
93         answerText.text = "";
94         InfomationPanelPirinas.SetActive(true);
95         plasmatikiMembraniObject.SetActive(true);
96

```

Εικόνα 6.5 Μέθοδος onClickRibosomataBt()



Εικόνα 6.6 Καμβάς με πληροφορίες

```

33 //Zwiko Kuttaro
34 public void OnClickButtonRivosomata() {
35     InformationPanelRivosomata.SetActive(false);
36 }
37
38 public void OnClickButtonPirinas() {
39     InfomationPanelPirinas.SetActive(false);
40 }
41

```

Εικόνα 6.7 Μέθοδοι για το κλείσιμο των καμβά του script 'Information'

Αν η απάντηση είναι λάθος, εμφανίζεται μήνυμα λάθους και εξαφανίζεται η απάντηση που έγραψε ο χρήστης με στόχο να ξανά επιλέξει το πλαίσιο και να προσπαθήσει πάλι. Ο χρήστης δεν μπορεί να κλείσει αυτόν τον καμβά μέχρι να απαντήσει σωστά στη συγκεκριμένη ερώτηση (Εικόνα 6.8).

```

138     } else {
139         TextApotelesma.text = "ΛΑΘΟΣ ΑΠΑΝΤΗΣΗ, ΠΡΟΣΠΑΘΗΣΕ ΞΑΝΑ !";
140         answerText.text = "";
141         Invoke("EnableText",1f);
142         Invoke("WrongDisableText",5.0f);
143     }

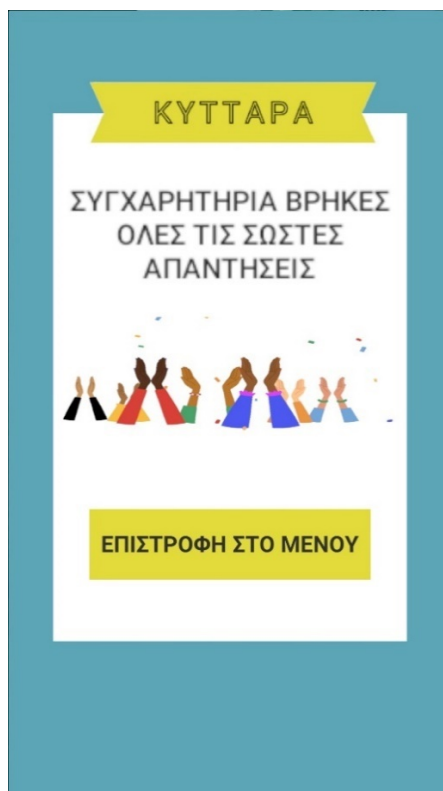
```

Εικόνα 6.8 Κώδικα για τη λάθος απάντηση

Μόλις ο χρήστης συμπληρώσει σωστά όλες τις ταμπέλες για το ζωικό κύτταρο, εμφανίζεται το φυτικό κύτταρο (Εικόνα 6.9). Και για το φυτικό κύτταρο με τη βοήθεια του script 'FutikoKuttaroTextControl', ακολουθήθηκε η ίδια διαδικασία δηλαδή ο χρήστης πατώντας τις ταμπέλες και συμπληρώνοντας σωστά την απάντηση, εμφανίζεται ένα πλαίσιο με πληροφορίες. Όταν ο χρήστης συμπληρώσει σωστά όλες τις ταμπέλες, εμφανίζεται ένας καμβάς με ένα μήνυμα που συγχαίρει τον μαθητή για την επιτυχή ολοκλήρωση του παιχνιδιού. Στο κάτω μέρος υπάρχει ένα κουμπί 'Επιστροφή στο μενού' όπου πατώντας το, ο χρήστης επιστρέφει στο κεντρικό μενού (Εικόνα 6.10). Τέλος, για το ζωικό και το φυτικό κύτταρο χρησιμοποιήθηκαν διαφορετικά script για λόγους οργάνωσης.



Εικόνα 6.9 Εμφάνιση παιχνιδιού 'Κύτταρο' - Φυτικό κύτταρο



Εικόνα 6.10 Τελευταία σκηνή δεύτερου παιχνιδιού

### 6.3 Επίλογος

Η ολοκλήρωση του δεύτερου παιχνιδιού για το υπό-κεφάλαιο ‘Κύτταρο: η μονάδα της ζωής’ ολοκληρώθηκε ακολουθώντας τα παραπάνω βήματα. Για την επιτυχή λειτουργία του παιχνιδιού δημιουργήθηκαν αντικείμενα από το πρόγραμμα Blender ώστε συνδέοντας όλα αυτά τα αντικείμενα να εμφανίζεται το φυτικό και το ζωικό κύτταρο. Υλοποιώντας το παραπάνω παιχνίδι, η μάθηση των κυττάρων γίνεται με έναν πιο ευχάριστο τρόπο.

## Κεφάλαιο 7ο: Τροφές

### 7.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται η περιγραφή του τρίτου παιχνιδιού της υπό-ενότητας ‘Πρόληψη ουσιών και πέψη στον άνθρωπο. Στο παιχνίδι παρουσιάζεται ένα σουπερμάρκετ με διάφορα τρόφιμα και ένα καλάθι και ο χρήστης πρέπει να τοποθετήσει τα τρόφιμα στο καλάθι ανάλογα με την κατηγορία που του ζητείτε. Ειδικότερα, παρουσιάζεται το παιχνίδι και αναλύονται όλα τα βήματα που ακολουθήθηκαν για τη δημιουργία του.

### 7.2 Ανάλυση παιχνιδιού

Το τρίτο παιχνίδι που δημιουργήθηκε αντιστοιχεί στη δεύτερη ενότητα του σχολικού βιβλίου ‘Πρόληψη ουσιών και πέψη’, στο υπό-κεφάλαιο 2.4 ‘Πρόσληψη ουσιών και πέψη στον άνθρωπο’ των σελίδων 45-46. Στόχος του παιχνιδιού είναι οι μαθητές να γεμίσουν το καλάθι του σουπερμάρκετ με τροφές ανάλογες με την κατηγορία που χρειάζεται. Οι μαθητές θα είναι σε θέση να κατανοήσουν ποια τρόφιμα είναι πλούσια σε πρωτεΐνες, λίπη, υδατάνθρακες, βιταμίνες και μέταλλα. Το παιχνίδι αποτελείται από την πρώτη σκηνή των οδηγιών, το παιχνίδι που περιλαμβάνει το σουπερμάρκετ και το αποτέλεσμα.

Στο κεντρικό μενού του παιχνιδιού υπάρχει ένα εικονίδιο με όνομα ‘Τροφές’. Μόλις ο χρήστης πατήσει το κουμπί, εμφανίζεται η πρώτη σκηνή των οδηγιών. Η σκηνή περιέχει τον τίτλο του παιχνιδιού, τις οδηγίες καθώς και ένα κουμπί ‘Πάμε’.



Εικόνα 7.1 Έναρξη παιχνιδιού 'Τροφές'

Όταν ο χρήστης πατήσει το κουμπί, γίνεται η έναρξη του παιχνιδιού. Επομένως, πηγαίνοντας στη σελίδα 46 του σχολικού βιβλίου και σκανάροντας το Qr code που αντιστοιχεί σε αυτό το παιχνίδι, εμφανίζεται το περιβάλλον που περιλαμβάνει επανυξημένα αντικείμενα. Τα αντικείμενα αυτά είναι ράφια, τρόφιμα, καρότσι του σουπερμάρκετ και στο πίσω μέρος υπάρχει ένα φόντο αντίστοιχο με το θέμα. Στο πάνω μέρος της οθόνης εμφανίζεται μια ταμπέλα που αναγράφεται μια κατηγορία. Ο χρήστης επιλέγει τα αντικείμενα ακολουθώντας ότι αναγράφεται στην ταμπέλα. Επιπλέον, στο κάτω μέρος υπάρχει μια μπάρα προόδου η οποία γεμίζει ανάλογα με τα πόσα σωστά αντικείμενα έχει βρει ο χρήστης. Η μπάρα προόδου είναι βοηθητική για να είναι ξεκάθαρο πόσα αντικείμενα απομένουν ώστε ο χρήστης να προχωρήσει στην επόμενη κατηγορία (Εικόνα 7.2).



Εικόνα 7.2 Εμφάνιση παιχνιδιού 'Τροφές'

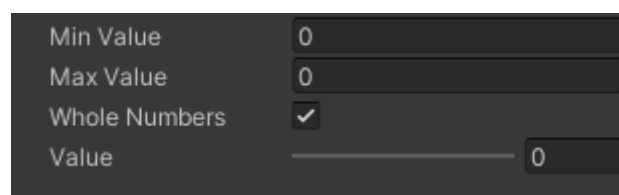
Για την υλοποίηση του παιχνιδιού δημιουργήθηκε το script 'Game3'. Στο πάνω μέρος του script έχουν δηλωθεί όλες οι απαραίτητες μεταβλητές (Εικόνα 7.3). Οι μεταβλητές είναι τύπου tag, κειμένου, GameObject, αλφαριθμητικού, ακέραιοι και Slider. Η αλφαριθμητική μεταβλητή χρησιμοποιήθηκε για τη δήλωση της κατηγορίας όπου θα αναλύσουμε παρακάτω. Η μεταβλητή με όνομα Slider είναι η μπάρα προόδου και τοποθετήθηκε από το μενού → GameObject → UI → Slider. Οι ρυθμίσεις που έγιναν στο Inspector της μπάρας προόδου είναι στο Slider → Min Value → 0, Slider → Max Value → 0 και Slider → Value → 0 (Εικόνα 7.4). Οι παραπάνω αλλαγές έγιναν διότι η διαχείριση της μπάρας προόδου γίνεται αποκλειστικά μέσα από τον κώδικα, όπου θα αναλυθεί παρακάτω.

```

15 //Tags Lipi
16 [SerializeField] private string rafiladiTag = "rafiladiTag";
17 [SerializeField] private string rafigalaTag = "rafiGalaTag";
18 [SerializeField] private string diskosCheeseTag = "diskosCheeseTag";
19 [SerializeField] private string diskosBouturoTag = "diskosBouturoTag";
20 [SerializeField] private string diskosTuriTag = "diskosTuriTag";
21
22 //Tags Idatanthrakes
23 [SerializeField] private string rafidimitriakaTag = "rafiDimitriakaTag";
24 [SerializeField] private string rafimakaroniaTag = "rafiMakaroniaTag";
25 [SerializeField] private string diskosPsoniTag = "diskosPsoniTag";
26
27 //Tags Vitamines
28 [SerializeField] private string diskosMpananesTag = "diskosMpananesTag";
29 [SerializeField] private string diskosAnanasTag = "diskosAnanasTag";
30 [SerializeField] private string diskosKerasiaTag = "diskosKerasiaTag";
31 [SerializeField] private string diskosMarouliTag = "diskosMarouliTag";
32 [SerializeField] private string diskosMiloTag = "diskosMiloTag";
33
34 //Tags Proteini
35 [SerializeField] private string diskosPsariaTag = "diskosPsariaTag";
36 [SerializeField] private string diskosXoirinoTag = "diskosXoirinoTag";
37 [SerializeField] private string diskosKotopouloTag = "diskosKotopouloTag";
38 [SerializeField] private string diskosEggsTag = "diskosEggsTag";
39
40 public Text TextCategory;
41 public GameObject panelStart;
42 public GameObject sliderCanva;
43 public GameObject panelFalse;
44 public GameObject CanvasEndGame;
45
46 public GameObject canvaInfo;
47 public Text InfoTitle;
48 public Text Odigies;
49
50 public GameObject ladi;
51 public GameObject gala;
52 public GameObject cheese;
53 public GameObject boutiro;
54 public GameObject turi;
55
56 public GameObject dimitriaka;
57 public GameObject makaronia;
58 public GameObject psoni;
59
60 public GameObject bananes;
61 public GameObject ananas;
62 public GameObject kerasia;
63 public GameObject marouli;
64 public GameObject mila;
65
66 public GameObject psaria;
67 public GameObject xoirino;
68 public GameObject kotopoulo;
69 public GameObject eggs;
70
71 private string category = "";
72 private int lipoiCounter;
73 private int idatanthrakesCounter;
74 private int vitaminesCounter;
75 private int proteinesCounter;
76
77 public Slider slider;
78 private int value;

```

Εικόνα 7.3 Μεταβλητές script 'Game3'



Εικόνα 7.4 Ρυθμίσεις μπάρας προόδου στο Unity

Η πρώτη μέθοδος που καλείται είναι η 'onClickStartButton'. Αυτή η μέθοδος αρχικά απενεργοποιεί τον καμβά με τις οδηγίες του παιχνιδιού και στη μεταβλητή 'category' γίνεται η δήλωση της πρώτης κατηγορίας, η οποία είναι η κατηγορία με τα λίπη. Έπειτα, ενεργοποιεί τον καμβά με όνομα sliderCanva που περιλαμβάνει την μπάρα προόδου και μέσω της μεταβλητής 'textCategory', εμφανίζει ένα κείμενο ενημερώνοντας τον χρήστη σε ποια κατηγορία βρίσκεται για να επιλέξει τα κατάλληλα αντικείμενα (Εικόνα 7.5).

```

275     public void onClickStartButton() {
276         panelStart.SetActive(false);
277         category = "lipoiCategory";
278         sliderCanva.SetActive(true);
279         TextCategory.text="ΤΡΟΦΕΣ ΠΛΟΥΣΙΕΣ ΣΕ ΛΙΠΗ";
280     }

```

Εικόνα 7.5 Μέθοδος 'onClickStartButton' του script 'Game3'

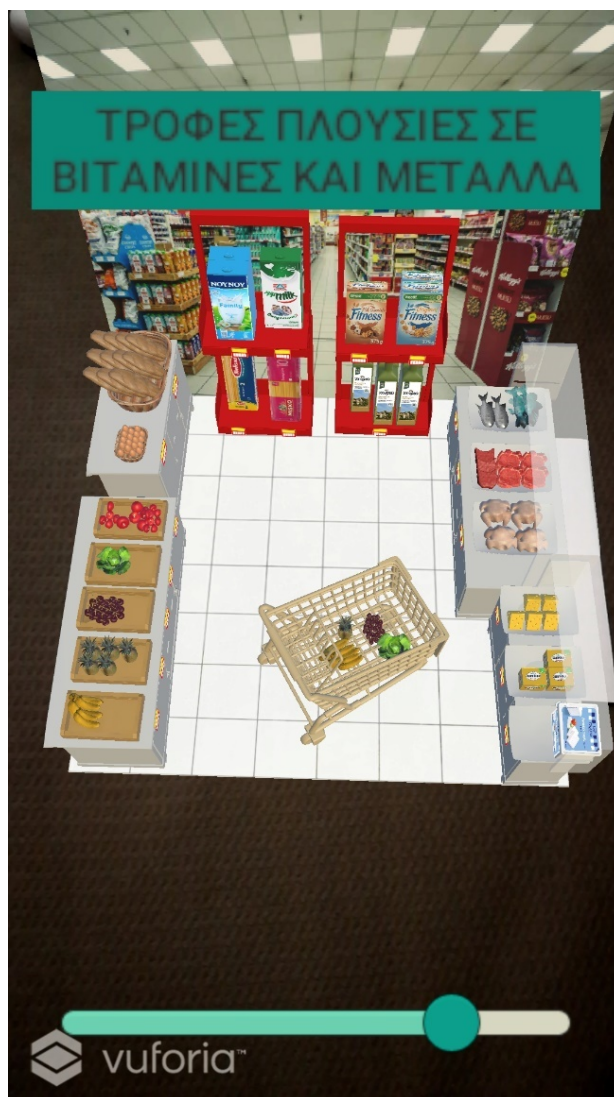
Στο βασικό μέρος του παιχνιδιού (Εικόνα 7.6) , πρώτα γίνεται έλεγχος σε ποια κατηγορία βρίσκεται ο χρήστης εκείνη τη στιγμή με την πρώτη συνθήκη if. Στη συνέχεια, ελέγχεται αν πατήθηκαν στην οθόνη της συσκευής τα αντικείμενα που αντιστοιχούν στην εκάστοτε κατηγορία με τις επόμενες συνθήκες if. Αν η επιλογή του χρήστη είναι σωστή, τότε το αντικείμενο που επέλεξε μετακινείται από το ράφι και τοποθετείται μέσα στο καλάθι του σουπερμάρκετ (Εικόνα 7.7) χρησιμοποιώντας την εντολή transform.localPosition. Δίνοντας στην εντολή hit.collider.enabled την τιμή false, ο χρήστης δεν μπορεί να ξαναπατήσει πάνω στο ίδιο αντικείμενο. Ταυτόχρονα, καλείται μια μέθοδος – αντίστοιχη για την κάθε κατηγορία – η οποία ενημερώνει την μπάρα προόδου. Για παράδειγμα, στην εικόνα 7.6 αυτή η μέθοδος είναι η UpdateValueLipi().

```

95     if (category == "lipoiCategory") {
96
97         if (selection.CompareTag(rafiLadiTag)) {
98             ladi.transform.localPosition = new Vector3(2.22f, 1.02f, 9.61f);
99             UpdateValueLipi();
100            hit.collider.enabled = false;
101
102        } else if (selection.CompareTag(rafiGalaTag)) {
103            gala.transform.localPosition = new Vector3(-5.57f, -0.94f, 5.08f);
104            UpdateValueLipi();
105            hit.collider.enabled = false;
106
107        } else if (selection.CompareTag(diskosCheeseTag)) {
108            cheese.transform.localPosition = new Vector3(29.8f, -3f, -11.4f);
109            UpdateValueLipi();
110            hit.collider.enabled = false;
111
112        } else if (selection.CompareTag(diskosBouturoTag)) {
113            boutiro.transform.localPosition = new Vector3(50.9f, 2.9f, -9.7f);
114            UpdateValueLipi();
115            hit.collider.enabled = false;
116
117        } else if (selection.CompareTag(diskosTuriTag)) {
118            turi.transform.localPosition = new Vector3(41.7f, 30.04f, -32.4f);
119            UpdateValueLipi();
120            hit.collider.enabled = false;
121
122        } else {
123            panelFalse.SetActive(true);
124            Invoke("DisableFalse",1.0f);
125        }
126
127    } else if (category == "IdatanthrakesCategory") {
128

```

Εικόνα 7.6 Βασικοί έλεγχοι μεθόδου 'Update' του script 'Game3'



Εικόνα 7.7 Τοποθέτηση αντικειμένων στο καλάθι

Σε αυτές τις μεθόδους γίνεται η αρχικοποίηση της πρώτης τιμής που έχει η μπάρα προόδου, της μεγαλύτερης τιμής που μπορεί να αποκτήσει με την εντολή `slider.maxValue` και κάθε φορά που καλείται, αυξάνεται κατά ένα. Επίσης, γίνεται η ενημέρωση της τρέχουσας τιμής ώστε κάθε φορά που ο χρήστης επιλέγει σωστή απάντηση να ανανεώνεται η μπάρα προόδου στη οθόνη της συσκευή του. Επιπλέον, γίνεται η αύξηση κατά μια μονάδα μιας μεταβλητής, η οποία χρησιμοποιείται σαν μετρητής για κάθε μια κατηγορία και είναι χρήσιμη για τον έλεγχο ολοκλήρωσης μιας κατηγορίας (Εικόνα 7.8). Όλες οι κατηγορίες έχουν διαφορετική αρχικοποίηση και ανανέωση μπάρας προόδου διότι σε κάθε κατηγορία δεν αντιστοιχεί ο ίδιος αριθμός αντικειμένων.

```

292     private void UpdateValueLipi() {
293         slider.value = 0f;
294         slider.maxValue = 5f;
295         value += 1;
296         slider.value = value;
297         lipoiCounter+=1;
298     }
299

```

Εικόνα 7.8 Μέθοδος ανανέωσης μπάρας προόδου

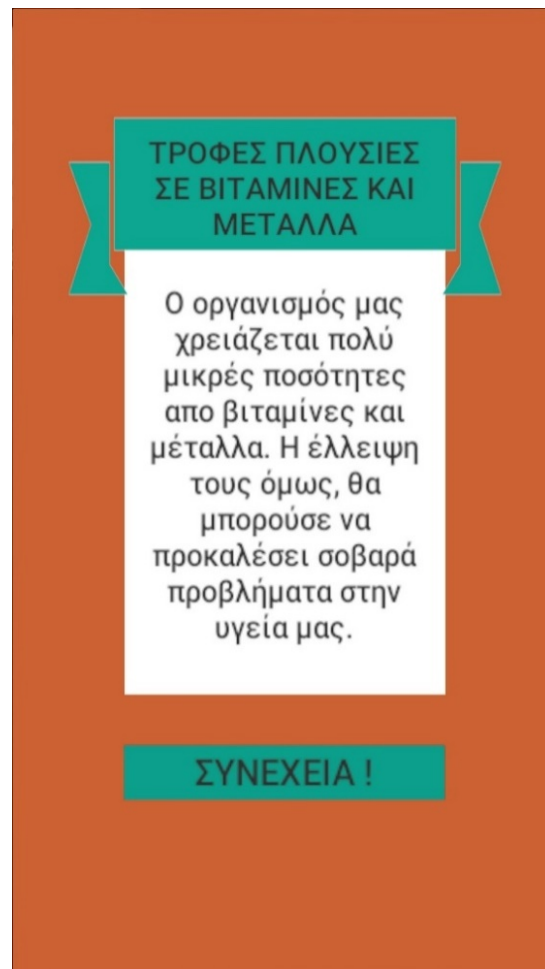
Για να ολοκληρωθεί μια κατηγορία, πρέπει ο χρήστης να έχει επιλέξει όλα τα σωστά αντικείμενα και να έχει γεμίσει ολόκληρη η μπάρα προόδου. Μέσα στην μέθοδο Update() έχουν δημιουργηθεί συνθήκες όπου ελέγχουν καθ' όλη τη διάρκεια του παιχνιδιού αν μια κατηγορία ολοκληρώθηκε (Εικόνα 7.9). Όταν συμβεί αυτό, τότε εμφανίζεται ένας καμβάς με πληροφορίες για την αντίστοιχη κατηγορία (Εικόνα 7.10) και αυτό γίνεται με την εντολή Invoke, καλώντας την μέθοδο 'InfoPanel'. Ταυτόχρονα, αφαιρούνται με την εντολή 'Destroy' τα αντικείμενα που τοποθετήθηκαν σωστά στο καρότσι και γίνεται ανανέωση της μπάρας προόδου με την μέθοδο resetSlider ώστε να ξαναγυρίσει στην αρχική της μορφή (Εικόνα 7.11). Επιπλέον, ανανεώνεται η μεταβλητή 'category' με την επόμενη κατηγορία και εμφανίζεται μήνυμα ενημερώνοντας τον χρήστη σε ποια κατηγορία βρίσκεται μέσω της μεταβλητής 'TextCategory', ώστε να τοποθετήσει στο καλάθι τα επόμενα αντικείμενα. Αν ο χρήστης επιλέξει λάθος αντικείμενο, τότε εμφανίζεται ένα μήνυμα για λίγα δευτερόλεπτα όπου του δίνει τη δυνατότητα να προσπαθήσει ξανά.

```

211     if (lipoiCounter == 5) {
212         Invoke("InfoPanel",1.0f);
213         sliderCanva.SetActive(false);
214         InfoTitle.text="ΠΡΟΦΕΣ ΠΛΟΥΣΙΕΣ ΣΕ ΛΙΠΗ";
215         Odigies.text="Τα λιπύ περιλαμβάνουν μεγάλα ποσοστά ενέργειας. Ο οργανισμός μας τα χρησιμοποιεί και αυτός ως αποθήκες ενέργειας.";
216         Destroy(ladi);
217         Destroy(cheese);
218         Destroy(boutiro);
219         Destroy(turi);
220         Destroy(gala);
221         lipoiCounter = 0;
222         resetSlider();
223         category = "IdatanthrakesCategory";
224         TextCategory.text="ΠΡΟΦΕΣ ΠΛΟΥΣΙΕΣ ΣΕ ΥΔΑΤΑΝΘΡΑΚΕΣ";
225     }
226

```

Εικόνα 7.9 Συνθήκη ολοκλήρωσης μιας κατηγορίας



Εικόνα 7.10 Καμβάς με πληροφορίες σχετικές με μια συγκεκριμένη κατηγορία

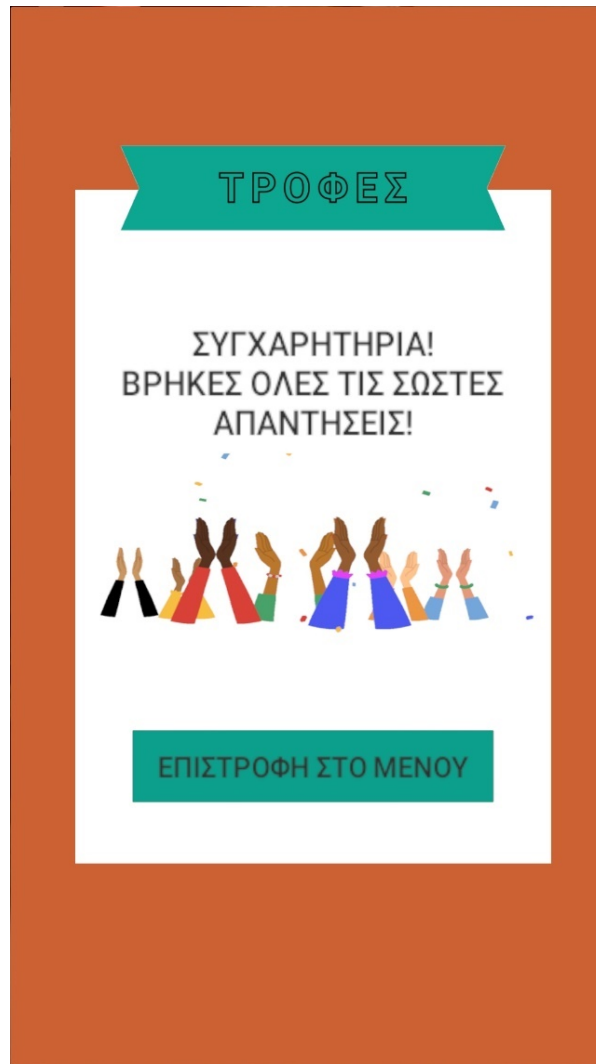
```

324     public void resetSlider() {
325         slider.maxValue = 0;
326         value = 0;
327         slider.value = value;
328     }

```

Εικόνα 7.11 Μέθοδος resetSlider()

Όλες οι παραπάνω ρυθμίσεις πραγματοποιήθηκαν για όλα τα αντικείμενα που εμφανίζονται στο επανξιμένο περιβάλλον και για όλες τις κατηγορίες. Η σειρά που εμφανίζονται οι κατηγορίες είναι πρώτα οι τροφές πλούσιες σε λίπη, υδατάνθρακες, βιταμίνες και μέταλλα και τελευταία τροφές πλούσιες σε πρωτεΐνες. Μόλις ο χρήστης ολοκληρώσει και τις τέσσερις κατηγορίες, εμφανίζεται ένας καμβάς. Ο καμβάς αυτός περιλαμβάνει τον τίτλο του παιχνιδιού, συγχαρητήρια για την επιτυχή ολοκλήρωση του παιχνιδιού και ένα κουμπί 'Επιστροφή στο μενού'. Όταν πατηθεί το κουμπί, ο χρήστης επιστρέφει το κεντρικό μενού με σκοπό να γίνει η επιλογή του επόμενου παιχνιδιού (Εικόνα 7.12).



Εικόνα 7.12 Τελευταία σκηνή τρίτου παιχνιδιού

### 7.3 Επίλογος

Ακολουθώντας τα παραπάνω βήματα, ολοκληρώθηκε το τρίτο παιχνίδι για την υπό-ενότητα 'Πρόληψη ουσιών και πέψη στον άνθρωπο'. Με τη βοήθεια αυτού του παιχνιδιού οι μαθητές είναι σε θέση να κατανοήσουν και να ξεχωρίσουν ποιες είναι οι βασικές κατηγορίες των τροφών και σε τι βοηθάει η καθεμία τροφή στον οργανισμό.

## Κεφάλαιο 8ο: Σπονδυλωτά και ασπόνδυλα ζώα

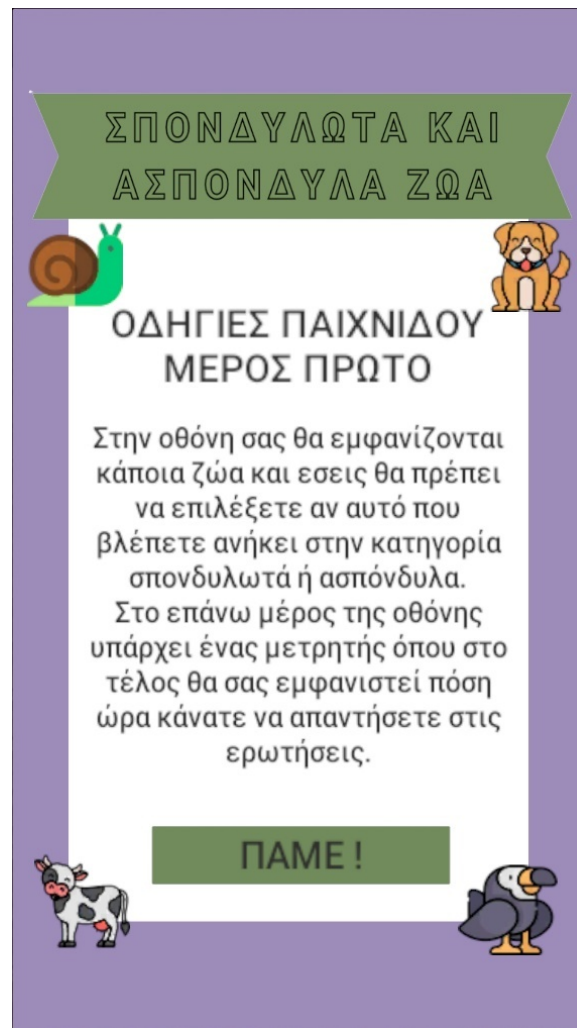
### 8.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται η περιγραφή του τέταρτου παιχνιδιού. Το τέταρτο παιχνίδι αποτελείται από τη σχολική ύλη δύο υπό-κεφαλαίων ‘Η οργάνωση των πολυκύτταρων οργανισμών’ και ‘Πρόληψη ουσιών και πέψη στους ζωικούς οργανισμούς’. Στο πρώτο μέρος του παιχνιδιού παρουσιάζονται διάφορα ζώα και ο χρήστης θα πρέπει να επιλέξει αν ανήκουν στη κατηγορία σπονδυλωτά ή ασπόνδυλα. Επιπλέον, στο δεύτερο μέρος εμφανίζονται ερωτήσεις και ο χρήστης πρέπει να επιλέξει τη σωστή απάντηση. Παρακάτω, γίνεται η ανάλυση των σκηνών, αντικειμένων και του κώδικα που χρησιμοποιήθηκε για την πραγματοποίηση αυτού του παιχνιδιού.

### 8.2 Ανάλυση παιχνιδιού

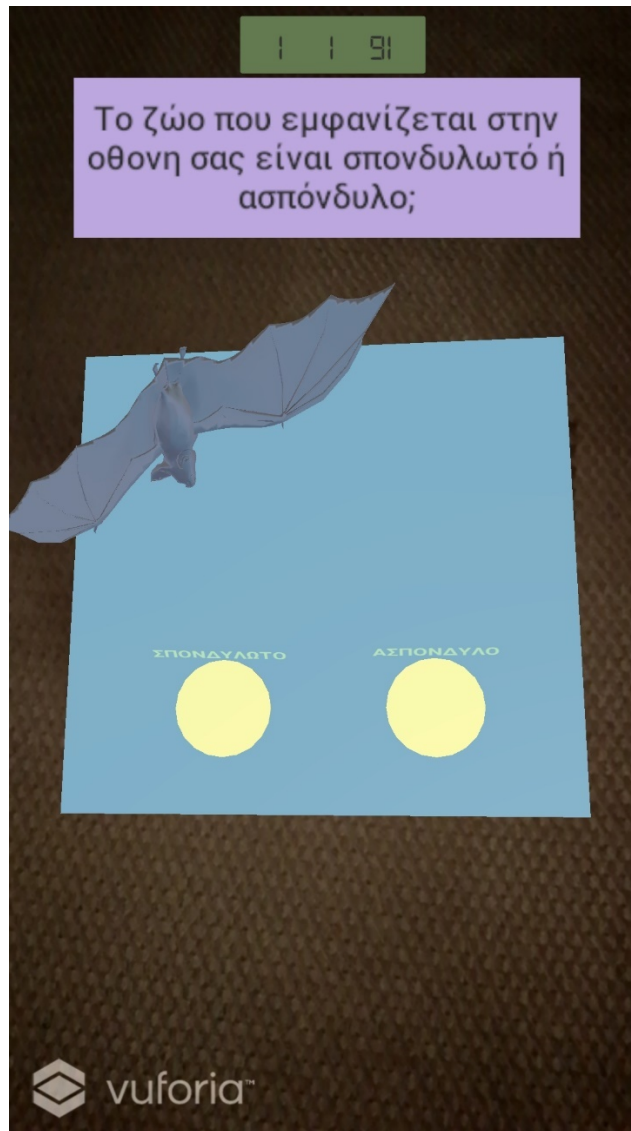
Το τέταρτο παιχνίδι χωρίζεται σε δύο μέρη. Το πρώτο μέρος αντιστοιχεί στην πρώτη ενότητα του σχολικού βιβλίου ‘Τα χαρακτηριστικά των οργανισμών’, στο υπό-κεφάλαιο 2.3 ‘Η οργάνωση των πολυκύτταρων οργανισμών’ της σελίδας 28. Το δεύτερο μέρος καλύπτει την ύλη της δεύτερης ενότητας ‘Πρόληψη ουσιών κα πέψη’, στο υπό-κεφάλαιο 2.3 ‘Πρόληψη ουσιών και πέψη στους ζωικούς οργανισμούς’, των σελίδων 42-43. Στο πρώτο μέρος στόχος του παιχνιδιού είναι ο μαθητής να επιλέξει αν το ζώο που εμφανίζεται ανήκει στην κατηγορία σπονδυλωτά ή ασπόνδυλα. Επιπλέον, στόχος του δεύτερου μέρους είναι ο μαθητής να απαντήσει σωστά στις ερωτήσεις που εμφανίζονται πάνω στο βιβλίο. Το παιχνίδι αποτελείται από τις οδηγίες, το πρώτο και το δεύτερο μέρος του παιχνιδιού και το αποτέλεσμα.

Επιλέγοντας από το κεντρικό μενού το εικονίδιο με όνομα ‘Σπονδυλωτά Ασπόνδυλα’, εμφανίζεται η πρώτη σκηνή του παιχνιδιού. Η σκηνή περιλαμβάνει τον τίτλο του παιχνιδιού, τις οδηγίες του πρώτου μέρους καθώς και το κουμπί ‘Πάμε’ (Εικόνα 8.1).



Εικόνα 8.1 Έναρξη παιχνιδιού 'Σπονδυλωτά και ασπόνδυλα' - Μέρος πρώτο

Αφού ο χρήστης πατήσει το κουμπί 'Πάμε', θα πρέπει να ανοίξει το σχολικό βιβλίο και να σκανάρει το Qr code που υπάρχει στη σελίδα 28. Μόλις σκανάρει το Qr code, εμφανίζεται το περιβάλλον με τα επαυξημένα αντικείμενα. Αρχικά, στο πάνω μέρος εμφανίζεται ένα χρονόμετρο όπου ξεκινάει να μετράει από τη στιγμή που ο χρήστης έχει εισέλθει στην σκηνή. Κάτω από το χρονόμετρο υπάρχει μια βασική ερώτηση η οποία δεν αλλάζει κατά τη διάρκεια του παιχνιδιού. Στη μέση της οθόνης εμφανίζονται διάφορα σπονδυλωτά και ασπόνδυλα ζώα και ο χρήστης πρέπει να επιλέξει τι είναι αυτό το ζώο που απεικονίζεται εκείνη τη στιγμή πατώντας τα στρόγγυλα κουμπιά (Εικόνα 8.2).



Εικόνα 8.2 Εμφάνιση παιχνιδιού 'Σπονδυλωτά και ασπόνδυλα' - Μέρος πρώτο

Η υλοποίηση του πρώτου μέρους του παιχνιδιού γίνεται με τα script 'Game4', 'EventGame4' και 'TimerManager'. Το βασικό script του πρώτου μέρους είναι στο script 'Game4' και στο πάνω μέρος έχουν δηλωθεί όλες οι απαραίτητες μεταβλητές. Οι μεταβλητές είναι τύπου GameObject, κειμένου, bool δηλαδή true ή false, ακέραιοι και η δήλωση TimerManager (Εικόνα 8.3). Η δήλωση αυτή ήταν απαραίτητη διότι χρειάστηκαν μερικές πληροφορίες από το script 'TimerManager' να μεταφερθούν στο script 'Game4', τα οποία θα αναλυθούν παρακάτω.

```

16     public GameObject panelStart;
17     public GameObject panelInfo;
18     public GameObject canvasTimer;
19     public Text textInfo;
20     public GameObject panelAnswer;
21     public GameObject canvasEndFirstGame;
22
23     public bool spondilota = false;
24     public bool aspondila = false;
25
26     public GameObject spondilotaObject;
27     public GameObject aspondilaObject;
28
29     public GameObject skilos;
30     public GameObject nixterida;
31     public GameObject gourouni;
32     public GameObject aetos;
33     public GameObject gata;
34
35     public GameObject araxni;
36     public GameObject kavouri;
37     public GameObject saligkari;
38     public GameObject axinos;
39
40     private int counterSostes;
41     public TimerManager timeManagerScript;
42
43     public Text endTimer;

```

Εικόνα 8.3 Μεταβλητές script 'Game4'

Η πρώτη μέθοδος που καλείται πατώντας το κουμπί 'Πάμε' ώστε να εμφανιστούν τα δισδιάστατα και τρισδιάστατα αντικείμενα είναι η μέθοδος `onClickStartButton()`. Αρχικά, μέσω της μεταβλητής `panelStart` κλείνει ο καμβάς με τις οδηγίες και με την εντολή `panelInfo` εμφανίζεται η σταθερή ερώτηση που υπάρχει στο πάνω μέρος. Επιπλέον, ενεργοποιείται το πρώτο ζώο, ο καμβάς που περιέχει το χρονόμετρο και καλείται και η μέθοδος `StartGame` από το script `TimerManager` για να ξεκινήσει να μετράει το χρονόμετρο (Εικόνα 8.4).

```

275     public void onClickStartButton() {
276         panelStart.SetActive(false);
277         panelInfo.SetActive(true);
278         skilos.SetActive(true);
279         canvasTimer.SetActive(true);
280         timeManagerScript.StartGame();
281     }

```

Εικόνα 8.4 Μέθοδος `onClickStartButton()` του script 'Game4'

Στον βασικό κώδικα του πρώτου μέρους (Εικόνα 8.5), στη μέθοδο `Update()` αρχικά γίνεται έλεγχος αν ο χρήστης έχει πατήσει κάποιο αντικείμενο με τη βοήθεια της `GetMouseButtonDown()`. Στη συνέχεια, για κάθε αντικείμενο που εμφανίζεται, πρώτα γίνεται έλεγχος αν είναι ενεργοποιημένο με την εντολή `activeSelf` και αν πατήθηκε η σωστή απάντηση. Αν η απάντηση που έδωσε ο χρήστης είναι σωστή, τότε

εμφανίζεται με τη μεταβλητή 'textInfo' ένα μήνυμα που ενημερώνει τον χρήστη ότι η απάντηση του είναι ορθή και ενεργοποιεί το επόμενο αντικείμενο. Ταυτόχρονα, μια μεταβλητή 'counterSostes' μετράει τις σωστές απαντήσεις ώστε όταν εμφανιστούν και απαντηθούν σωστά τα εννέα αντικείμενα να γίνουν μερικές ενέργειες που θα αναλυθούν παρακάτω. Από την άλλη πλευρά, αν ο χρήστης απαντήσει λάθος τότε εμφανίζεται ένα μήνυμα λάθος απάντησης και μπορεί να προσπαθήσει ξανά.

```

45 void Update() {
46
47     if (Input.GetMouseButtonDown(0)) {
48
49         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
50         RaycastHit hit;
51
52         if (Physics.Raycast(ray, out hit))
53         {
54             var selection = hit.transform;
55
56             //SKILOS -----
57
58             if ((skilos.activeSelf) && (spondilota)) {
59                 spondilota = false;
60
61                 textInfo.text = "ΣΩΣΤΗ ΕΠΙΛΟΓΗ!";
62                 Invoke("EnableText",1f);
63                 Invoke("DisableText",3f);
64
65                 SetActiveFalseGameObjects();
66                 Invoke("SetActiveTrueKavouri",3f);
67                 counterSostes ++;
68
69             } else if ((skilos.activeSelf) && (aspondila)) {
70                 aspondila = false;
71                 textInfo.text = "ΛΑΘΟΣ ΕΠΙΛΟΓΗ!";
72                 Invoke("EnableText",1f);
73                 Invoke("DisableText",3f);
74             }

```

Εικόνα 8.5 Βασικοί έλεγχοι μεθόδου Update() του script 'Game4'

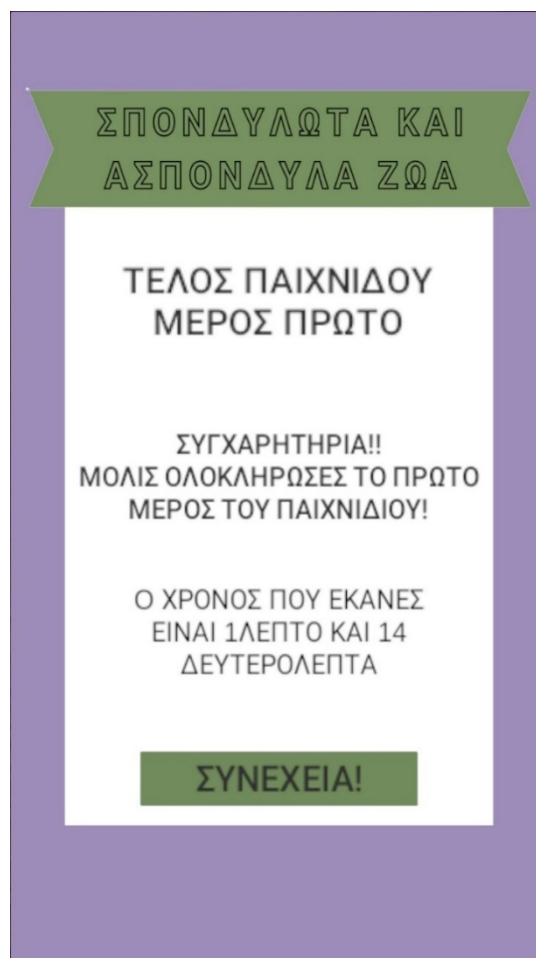
Όταν ο χρήστης απαντήσει σωστά και στα εννέα αντικείμενα τότε γίνεται είσοδος στη συνθήκη 'counterSoster >= 9' (Εικόνα 8.6). Πρώτα απ' όλα, εμφανίζεται μέσω της μεταβλητής 'textInfo' ένα μήνυμα ότι το παιχνίδι έχει τελειώσει. Έπειτα, καλείται η μέθοδος 'TimerStop' από το script 'TimerManager' ώστε να σταματήσει να μετράει ο χρόνος. Μέσω της Invoke, καλείται η μέθοδος 'EnableEndFirstGame' η οποία ενεργοποιεί τον καμβά ολοκλήρωσης του πρώτου μέρους. Ο καμβάς περιλαμβάνει τον τίτλο του παιχνιδιού, ένα μήνυμα για την επιτυχή ολοκλήρωση του πρώτου μέρους του παιχνιδιού και τον χρόνο ολοκλήρωσης (Εικόνα 8.7). Ο χρόνος ολοκλήρωσης του παιχνιδιού εμφανίζεται μέσω της μεταβλητής 'endTimer' όπως φαίνεται στην εικόνα 8.6. Με την εντολή 'GetComponent<Όνομα Script>' από το script 'TimeManager' λαμβάνονται τα λεπτά και τα δευτερόλεπτα στο script 'Game4' και τοποθετούνται στις μεταβλητές 'minuteInt' και 'secondsInt'. Οι επόμενες συνθήκες if είναι απαραίτητες ώστε να εμφανίζονται σωστά τα κείμενα στον καμβά ανάλογα με το πόσα λεπτά ή δευτερόλεπτα έκανε να ολοκληρώσει ο χρήστης το πρώτο μέρος.

```

239     if (counterSostes >= 9) {
240         counterSostes =0;
241
242         textInfo.text = "ΤΕΛΟΣ ΠΑΙΧΝΙΔΙΟΥ";
243         Invoke("EnableText",1f);
244         timeManagerScript.TimerStop();
245
246         Invoke("EnableEndFirstGame",1f);
247         //Εμφανίζει το χρόνο που έκανες για να ολοκληρώσεις το πρώτο μέρος (παίρνει την τιμή από το TimerManager Script)
248         int minutesInt = GetComponent<TimerManager> ().minutesInt;
249         int secondsInt = GetComponent<TimerManager> ().secondsInt;
250         if(minutesInt==0){
251             endTimer.text = "Ο ΧΡΟΝΟΣ ΠΟΥ ΕΚΑΝΕΣ ΕΙΝΑΙ "+secondsInt+" ΔΕΥΤΕΡΟΛΕΠΤΑ";
252         } else if (minutesInt==1) {
253             if(secondsInt==0){
254                 endTimer.text = "Ο ΧΡΟΝΟΣ ΠΟΥ ΕΚΑΝΕΣ ΕΙΝΑΙ " +minutesInt+" ΛΕΠΤΟ";
255             }else if(secondsInt==0){
256                 endTimer.text = "Ο ΧΡΟΝΟΣ ΠΟΥ ΕΚΑΝΕΣ ΕΙΝΑΙ " +minutesInt+" ΛΕΠΤΟ ΚΑΙ "+secondsInt+" ΔΕΥΤΕΡΟΛΕΠΤΟ";
257             }else{
258                 endTimer.text = "Ο ΧΡΟΝΟΣ ΠΟΥ ΕΚΑΝΕΣ ΕΙΝΑΙ " +minutesInt+" ΛΕΠΤΟ ΚΑΙ "+secondsInt+" ΔΕΥΤΕΡΟΛΕΠΤΑ";
259             }
260         } else {
261             if(secondsInt==0){
262                 endTimer.text = "Ο ΧΡΟΝΟΣ ΠΟΥ ΕΚΑΝΕΣ ΕΙΝΑΙ " +minutesInt+" ΛΕΠΤΑ";
263             }else if(secondsInt==0){
264                 endTimer.text = "Ο ΧΡΟΝΟΣ ΠΟΥ ΕΚΑΝΕΣ ΕΙΝΑΙ " +minutesInt+" ΛΕΠΤΑ ΚΑΙ "+secondsInt+" ΔΕΥΤΕΡΟΛΕΠΤΟ";
265             }else{
266                 endTimer.text = "Ο ΧΡΟΝΟΣ ΠΟΥ ΕΚΑΝΕΣ ΕΙΝΑΙ " +minutesInt+" ΛΕΠΤΑ ΚΑΙ "+secondsInt+" ΔΕΥΤΕΡΟΛΕΠΤΑ";
267             }
268         }
269     }

```

Εικόνα 8.6 Συνθήκη ολοκλήρωσης πρώτου μέρους του τέταρτου παιχνιδιού



Εικόνα 8.7 Τελευταία σκηνή μέρους πρώτου τέταρτου παιχνιδιού

Το script 'EventGame4' χρησιμοποιείται για τα δυο στρόγγυλα κουμπιά. Τα κουμπιά είναι 'cylinder' όπου τοποθετήθηκαν από το μενού → GameObject → 3d Object → Cylinder. Όπως αναλύθηκε και στο πρώτο παιχνίδι, τέτοια αντικείμενα δεν μπορούν να αλληλεπιδρούν με τον χρήστη και μόνο με αυτόν τον τρόπο συμπεριφέρονται ως κουμπιά.

Το script 'TimerManager' έχει δημιουργηθεί αποκλειστικά για τη διαχείριση του χρονομέτρου. Στο πάνω μέρος του παιχνιδιού υπάρχει το χρονόμετρο το οποίο έχει δημιουργηθεί με τρία διαφορετικά 'text'. Το πρώτο αντιστοιχεί στα λεπτά, το δεύτερο στα δευτερόλεπτα και το τρίτο στα κλάσματα του δευτερολέπτου. Στο επάνω μέρος του script έχουν δηλωθεί οι μεταβλητές (Εικόνα 8.8) οι οποίες περιλαμβάνουν τα text που αναφέρθηκαν παραπάνω, μια μεταβλητή για την αρχή του χρονομέτρου και μια για το τέλος. Επίσης, δημιουργήθηκε μια μεταβλητή με όνομα 'isRunning' όπου παίρνει τιμή true ή false ανάλογα αν το χρονόμετρο μετράει ή αν έχει σταματήσει και οι μεταβλητές τύπου int όπου είναι απαραίτητες για την υλοποίηση του χρονομέτρου.

```

16     public Text timerMinutes;
17     public Text timerSeconds;
18     public Text timerSeconds100;
19     private float startTime;
20     public float stopTime;
21     private float timerTime;
22     private bool isRunning = false;
23     public int minutesInt=0;
24     public int secondsInt=0;

```

Εικόνα 8.8 Μεταβλητές script 'TimerManager'

Στη μέθοδο Update(), στη μεταβλητή 'timerTime' έχει τοποθετηθεί ο τρέχον χρόνος και στη συνέχεια με πράξεις χωρίζεται σε λεπτά, δευτερόλεπτα και κλάσματα του δευτερολέπτου. Με μια συνθήκη if, ελέγχεται αν το χρονόμετρο έχει ξεκινήσει και βρίσκεται ακόμα σε λειτουργία και ενημερώνει τις τιμές που φαίνονται στον καμβά (Εικόνα 8.9)

Στη μέθοδος 'TimesStart()' γίνεται έλεγχος αν το χρονόμετρο δεν έχει ξεκινήσει και τότε τον ενεργοποιεί. Στη μεταβλητή 'startTime' τοποθετείται η μεταβλητή 'Time.time'. Αυτή η μεταβλητή είναι χρήσιμη διότι δείχνει την ώρα όπου ξεκίνησε το χρονόμετρο. Στη μέθοδο 'TimerStop' γίνεται έλεγχος αν το χρονόμετρο λειτουργεί και τότε απενεργοποιείται. Στη μεταβλητή 'stopTime' τοποθετείται η μεταβλητή 'Time.time', δηλαδή σε ποιον χρόνο σταμάτησε το χρονόμετρο (Εικόνα 8.10).

```

26 void Update() {
27
28     timerTime = stopTime + (Time.time - startTime);
29     minutesInt = (int)timerTime / 60;
30     secondsInt = (int)timerTime % 60;
31     int seconds100Int = (int)(Mathf.Floor((timerTime - (secondsInt + minutesInt * 60)) * 100));
32
33     if(isRunning) {
34         timerMinutes.text = minutesInt.ToString();
35         timerSeconds.text = secondsInt.ToString();
36         timerSeconds100.text = seconds100Int.ToString();
37     }
38 }
39

```

Εικόνα 8.9 Μέθοδος 'Update' του script 'TimerManager'

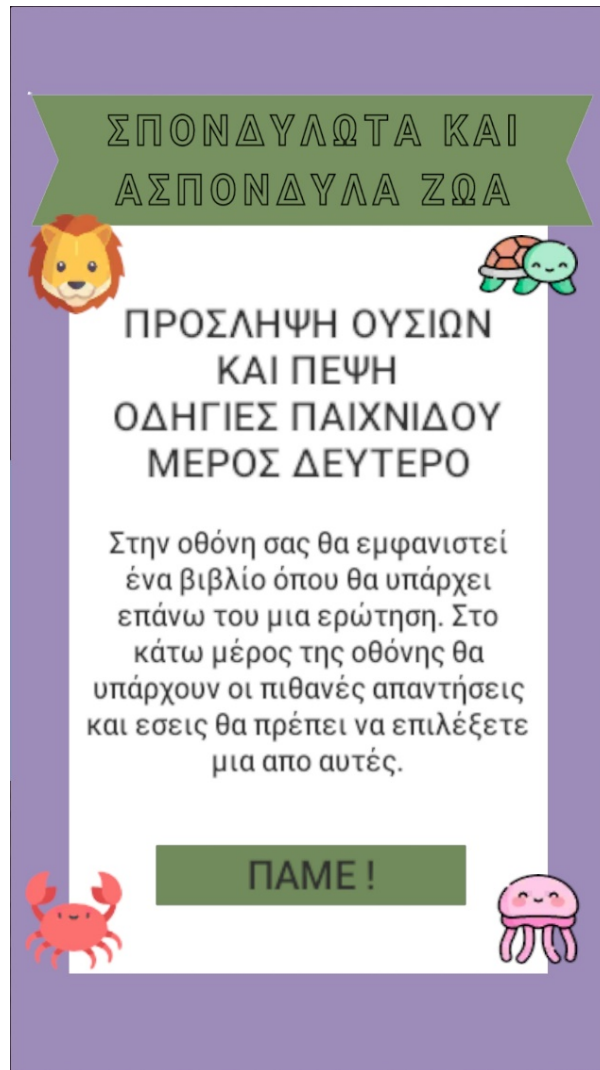
```

40 public void TimerStart() {
41
42     if(!isRunning) {
43         Debug.Log("START");
44         isRunning = true;
45         startTime = Time.time;
46     }
47 }
48
49 public void TimerStop() {
50     if(isRunning) {
51         Debug.Log("STOP");
52         isRunning = false;
53         stopTime = Time.time;
54         Debug.Log(stopTime);
55     }
56 }
57

```

Εικόνα 8.10 Μέθοδοι 'TimerStart' και 'TimerStop' του script 'TimerManager'

Όταν ο χρήστης ολοκληρώσει το πρώτο μέρος και πατήσει το κουμπί 'Συνέχεια' τότε εμφανίζεται ο καμβάς έναρξης του δεύτερου μέρους. Ο καμβάς αποτελείται από τον τίτλο του παιχνιδιού, τις οδηγίες του δεύτερου μέρους και το κουμπί 'Πάμε' (Εικόνα 8.11).



Εικόνα 8.11 Έναρξη παιχνιδιού 'Σπονδυλωτά και ασπόνδυλα' - Μέρος δεύτερο

Μόλις ο χρήστης πατήσει το κουμπί 'Πάμε' και σκανάρει το ίδιο Qr code που βρίσκεται στη σελίδα 43, τότε θα εμφανιστεί το περιβάλλον. Στο επαυξημένο περιβάλλον εμφανίζεται ένα βιβλίο όπου στην αριστερή πλευρά υπάρχουν μερικά σπονδυλωτά και ασπόνδυλα ζώα και στη δεξιά πλευρά εμφανίζεται μια ερώτηση. Στο κάτω μέρος της οθόνης εμφανίζονται δυο επιλογές. Ο χρήστης διαβάζοντας την ερώτηση που βρίσκεται πάνω στο βιβλίο θα πρέπει να επιλέξει μια από τις δυο απαντήσεις (Εικόνα 8.12).



Εικόνα 8.12 Εμφάνιση παιχνιδιού 'Σπονδυλωτά και ασπόνδυλα' - Μέρος δεύτερο

Η υλοποίηση του δεύτερου μέρους πραγματοποιήθηκε με τα scripts 'PartTwoGame4' και 'Question'. Στο βασικό script το οποίο είναι το 'PartTwoGame4', αρχικά γίνεται η δήλωση των μεταβλητών στο πάνω μέρος (Εικόνα 8.13). Οι πρώτες μεταβλητές τύπου GameObject, απενεργοποιούν όλα τα αντικείμενα που χρησιμοποιήθηκαν στο πρώτο μέρος και ενεργοποιούνται τα αντικείμενα που χρειάζονται στο δεύτερο μέρος. Οι μεταβλητές με όνομα 'ButtonOneText' και 'ButtonTwoText' είναι απαραίτητες διότι αλλάζουν αυτόματα οι απαντήσεις στο κάτω μέρος της οθόνης ανάλογα με την ερώτηση που εμφανίζεται, όπου θα αναλυθούν παρακάτω. Η μεταβλητή 'Question' είναι μια λίστα που περιέχει όλες τις απαντήσεις και τις ερωτήσεις που έχουν δηλωθεί στη Unity και γίνεται με τη βοήθεια του script 'Question' (Εικόνα 8.14). Στο παιχνίδι δημιουργήθηκαν έντεκα ερωτήσεις, αλλά το μέγεθος της λίστας δεν είναι σταθερό και μπορεί να αλλάξει (Εικόνα 8.15). Η λίστα με όνομα 'unansweredQuestions' περιέχει όλες τις ερωτήσεις που δεν έχουν απαντηθεί ακόμα.

```

16 public GameObject canvasEndFirstGame;
17 public GameObject spondilotaButton;
18 public GameObject aspondilaButton;
19 public GameObject spondilotaText;
20 public GameObject aspondilaText;
21 public GameObject panelInfo;
22
23 public GameObject canvasStartGamePartTwo;
24 public GameObject CanvasAnswers;
25 public Text QuestionBook;
26 public GameObject CanvasQuestion;
27 public GameObject book;
28
29 public GameObject ButtonOne;
30 public Text ButtonOneText;
31
32 public GameObject ButtonTwo;
33 public Text ButtonTwoText;
34
35
36 public Question[] questions;
37 private static List<Question> unansweredQuestions;
38 private Question currentQuestion;
39
40 public float timeBetweenQuestions = 1f;
41
42 public int counter;
43 public GameObject ShowSosto;
44 public GameObject ShowLathos;
45 public GameObject canvasEndSecondGame;
46 public GameObject CanvasTimer;

```

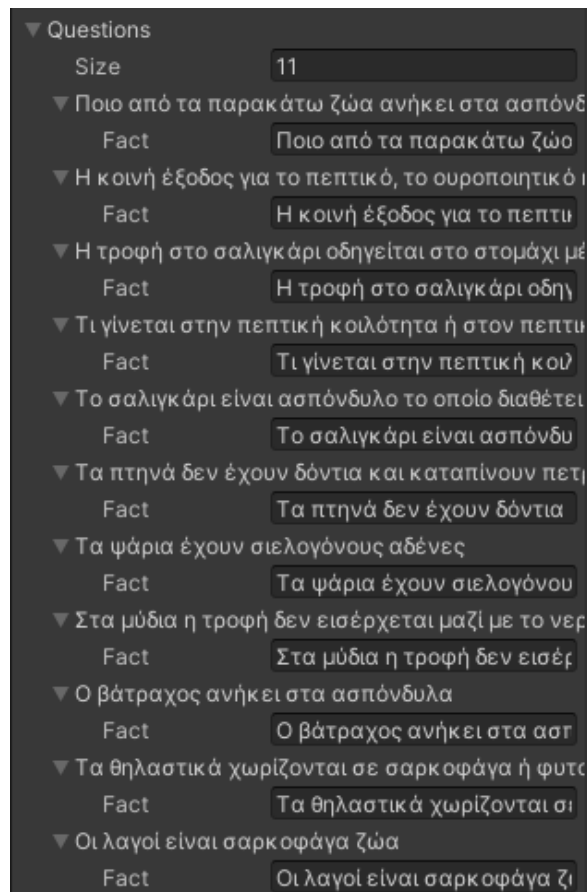
Εικόνα 8.13 Μεταβλητές script 'PartTwoGame4'

```

1
2 [System.Serializable]
3 public class Question
4 {
5     public string fact; //εδώ δηλώνω τις ερωτήσεις του παιχνιδιού
6
7 }
8

```

Εικόνα 8.14 Script 'Question'



Εικόνα 8.15 Δήλωση ερωτήσεων

Η μέθοδος `Start()` είναι η πρώτη μέθοδος που καλείται μόλις ξεκινήσει το παιχνίδι. Σε αυτήν την μέθοδο, τοποθετούνται πρώτα όλες οι ερωτήσεις στην μεταβλητή `unansweredQuestions` και στη συνέχεια με τη βοήθεια της `Invoke` καλείτε μετά από λίγα δευτερόλεπτα η μέθοδος `GetRandomQuestion` (Εικόνα 8.16). Στη μέθοδο `GetRandomQuestion` γίνεται με τυχαία σειρά η επιλογή μιας ερώτησης και εμφανίζεται επάνω στο βιβλίο μέσω της μεταβλητής `QuestionBook` (Εικόνα 8.17).

```

52 void Start()
53 {
54     if (unansweredQuestions == null || unansweredQuestions.Count == 0){
55         unansweredQuestions = questions.ToList<Question>();
56     }
57
58     Invoke("GetRandomQuestion",3f);
59 }

```

Εικόνα 8.16 Μέθοδος 'Start' του script 'PartTwoGame4'

```

177     public void GetRandomQuestion () {
178         int randomQuestionIndex = Random.Range (0, unansweredQuestions.Count);
179         currentQuestion = unansweredQuestions[randomQuestionIndex];
180
181         QuestionBook.text = currentQuestion.fact;
182     }

```

Εικόνα 8.17 Μέθοδος 'GetRandomQuestion' του script 'PartTwoGame4'

Στη μέθοδο Update() η οποία καλείτε καθ' όλη τη διάρκεια του παιχνιδιού, τοποθετούνται οι απαντήσεις που εμφανίζονται στο κάτω μέρος της οθόνης (Εικόνα 8.18). Αρχικά, γίνεται έλεγχος αν πάνω στο βιβλίο έχει τοποθετηθεί μια ερώτηση και στη συνέχεια τοποθετούνται στα 'ButtonOneText' και 'ButtonTwoText' οι δυο πιθανές απαντήσεις που θα εμφανιστούν στο κάτω μέρος της οθόνης.

```

61     void Update() {
62         if(GameObject.Find("ButtonOneText")!=null && GameObject.Find("ButtonTwoText")!=null){
63             if(QuestionBook.text == "Ποιο από τα παρακάτω ζώα ανήκει στα σπόνδυλα;") {
64                 GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΚΟΛΙΑΚΙ";
65                 GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΑΛΕΠΟΥ";
66             } else if(QuestionBook.text == "Η κοινή έξοδος για το πεπτικό, το ουροποιητικό και το αναπαραγωγικό σύστημα κάποιων σπονδυλιτών είναι:") {
67                 GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΠΕΠΤΙΚΗ ΚΟΙΛΟΤΗΤΑ";
68                 GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΑΜΑΡΑ";
69             } else if(QuestionBook.text == "Η τροφή στο σαλιγκάρι οδηγείται στο στομάχι μέσω του:") {
70                 GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΠΕΠΤΙΚΟΝ ΣΩΛΗΝΑ";
71                 GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΟΙΣΟΦΑΓΟΥ";
72             } else if(QuestionBook.text == "Τι γίνεται στην πεπτική κοιλότητα ή στον πεπτικό σωλήνα;") {
73                 GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΠΡΟΣΛΗΨΗ ΟΥΣΙΩΝ";
74                 GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΠΕΨΗ";
75             } else if(QuestionBook.text == "Το σαλιγκάρι είναι ασπόνδυλο το οποίο διαθέτει παχιά χείλη και μια οδοντωτή προεξοχή για να ροκανίζει την τροφή του;") {
76                 GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
77                 GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
78             } else if(QuestionBook.text == "Τα πτηνά δεν έχουν δόντια και καταπίνουν πετραδάκια μαζί με την τροφή για να την αλέθουν") {
79                 GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
80                 GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
81             } else if(QuestionBook.text == "Τα ψάρια έχουν στελογόνους αδένες") {
82                 GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
83                 GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
84             } else if(QuestionBook.text == "Στα μύδια η τροφή δεν εισέρχεται μαζί με το νερό") {

```

Εικόνα 8.18 Μέθοδος Update() του script 'PartTwoGame4'

Οι τοποθεσίες των απαντήσεων είναι προκαθορισμένη. Με άλλα λόγια, η μέθοδος onClickOne (Εικόνα 8.19) συνδέεται με το ButtonOne, δηλαδή με την απάντηση που υπάρχει στην αριστερή πλευρά της οθόνης. Σε τέσσερις ερωτήσεις, η σωστή απάντηση βρίσκεται στην αριστερή πλευρά των απαντήσεων. Στην μέθοδο αυτήν, γίνεται έλεγχος αν η ερώτηση που επιλέχθηκε είναι μια από τις τέσσερις και τότε εμφανίζεται μήνυμα επιτυχής επιλογής. Επίσης, έχει δημιουργηθεί ένας μετρητής που μετράει τις σωστές απαντήσεις διότι στο παιχνίδι εμφανίζονται μόνο οκτώ ερωτήσεις. Ταυτόχρονα, καλείται η μέθοδος 'NextQuestion' (Εικόνα 8.20) στην οποία σβήνετε η ερώτηση που απαντήθηκε σωστά από την λίστα ώστε να μην ξανά εμφανιστεί. Καλείτε επιπλέον η μέθοδος EndGame που ελέγχει αν οι ερωτήσεις ξεπέρασαν τις οκτώ ώστε να σταματήσει το παιχνίδι. Αν οι ερωτήσεις που εμφανίστηκαν είναι λιγότερες από οκτώ, τότε καλείται η μέθοδος 'GetRandomQuestion' για να εμφανιστεί η επόμενη ερώτηση. Αν ο χρήστης επιλέξει λάθος απάντηση, εμφανίζεται ένα μήνυμα λάθος επιλογής και του δίνεται η δυνατότητα να προσπαθήσει ξανά.

Στη μέθοδο onClickTwo, δηλαδή στο κουμπί από τη δεξιά πλευρά έχουν τοποθετηθεί επτά σωστές απαντήσεις. Η ίδια διαδικασία που αναλύθηκε παραπάνω πραγματοποιείται και σε αυτή τη μέθοδο. Το παιχνίδι ολοκληρώνεται μόλις απαντηθούν σωστά οκτώ ερωτήσεις και τότε εμφανίζεται ένας καμβάς (Εικόνα 8.21) . Ο καμβάς αυτός αποτελείται από τον τίτλο του παιχνιδιού, ένα μήνυμα συγχαρητηρίων

## Κεφάλαιο 8

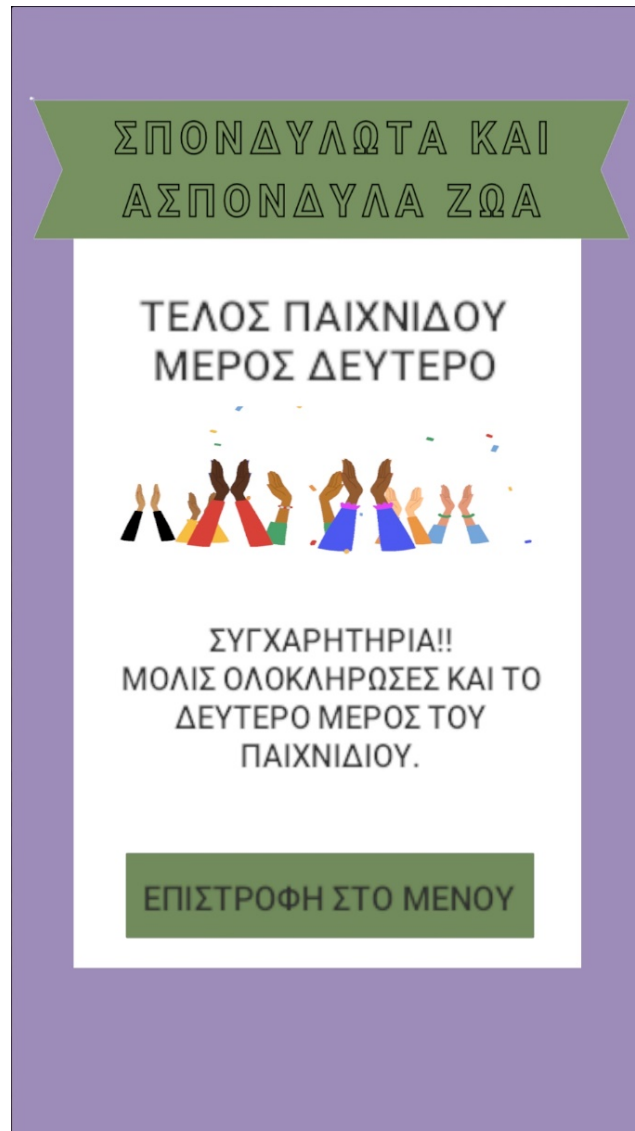
και το κουμπί ‘Επιστροφή στο μενού’. Πατώντας το κουμπί, ο χρήστης επιστρέφει στο κεντρικό μενού ώστε να επιλέξει το επόμενο παιχνίδι.

```
101 public void onClickOne () {
102     if (QuestionBook.text == "Ποιο από τα παρακάτω ζώα ανήκει στα ασπόνδυλα;") {
103         ShowSosto.SetActive(true);
104         counter++;
105         StartCoroutine(NextQuestion());
106     } else if (QuestionBook.text == "Το σαλιγκάρι είναι ασπόνδυλο το οποίο διαθέτει παχιά χείλη και μια οδοντωτή προεξοχή για να ροκανίζει την τροφή του;") {
107         ShowSosto.SetActive(true);
108         counter++;
109         StartCoroutine(NextQuestion());
110     } else if (QuestionBook.text == "Τα πτηνά δεν έχουν δόντια και καταπίνουν πετραδάκια μαζί με την τροφή για να την αλέθουν") {
111         ShowSosto.SetActive(true);
112         counter++;
113         StartCoroutine(NextQuestion());
114     } else if (QuestionBook.text == "Τα θηλαστικά χωρίζονται σε σαρκοφάγα ή φυτοφάγα") {
115         ShowSosto.SetActive(true);
116         counter++;
117         StartCoroutine(NextQuestion());
118     } else {
119         Invoke("EnableLathos",1f);
120         Invoke("DisableLathos",3f);
121     }
122 }
123 public void onClickTwo () {
124     if (QuestionBook.text == "Η τροφή στο σαλιγκάρι οδηγείται στο στομάχι μέσω του:") {
125         ShowSosto.SetActive(true);
126         counter++;
127         StartCoroutine(NextQuestion());
128     } else if (QuestionBook.text == "Η κοινή έξοδος για το πεπτικό, το ουροποιητικό και το αναπαραγωγικό σύστημα κάποιων σπονδυλιτών είναι:") {
129         ShowSosto.SetActive(true);
```

Εικόνα 8.19 Μέθοδος ‘onClickOne’ και ‘onClickTwo’

```
185 IEnumerator NextQuestion (){
186     unansweredQuestions.Remove(currentQuestion);
187
188     yield return new WaitForSeconds(timeBetweenQuestions);
189     ShowSosto.SetActive(false);
190
191     EndGame();
192     GetRandomQuestion();
193 }
```

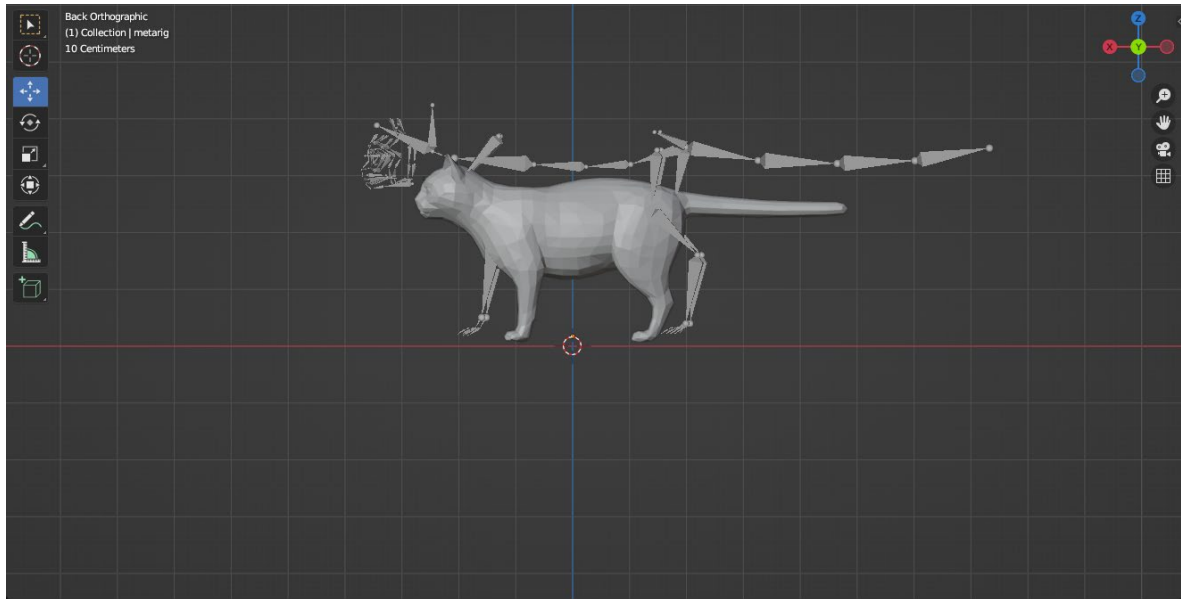
Εικόνα 8.20 Μέθοδος ‘NextQuestion’



Εικόνα 8.21 Τελευταία σκηνή μέρους δεύτερου τέταρτου παιχνιδιού

### 8.2.1 Animation

Στο πρώτο μέρος του παιχνιδιού εμφανίζονται μερικά σπονδυλωτά και ασπόνδυλα ζώα. Σε ορισμένα ζώα από αυτά, τοποθετήθηκε κίνηση μέσω του προγράμματος Blender. Ανοίγοντας το πρόγραμμα Blender και κάνοντας import το αντικείμενο ή δημιουργώντας το αντικείμενο από την αρχή, εμφανίζεται στο κέντρο της οθόνης. Στη συνέχεια, επιλέγοντας το αντικείμενο και πατώντας Add → Armature → Animals → για παράδειγμα Cat (Meta-Ring), εμφανίζονται οστά (bones) τα οποία πρέπει να προσαρμοστούν στο εκάστοτε αντικείμενο (Εικόνα 8.22).

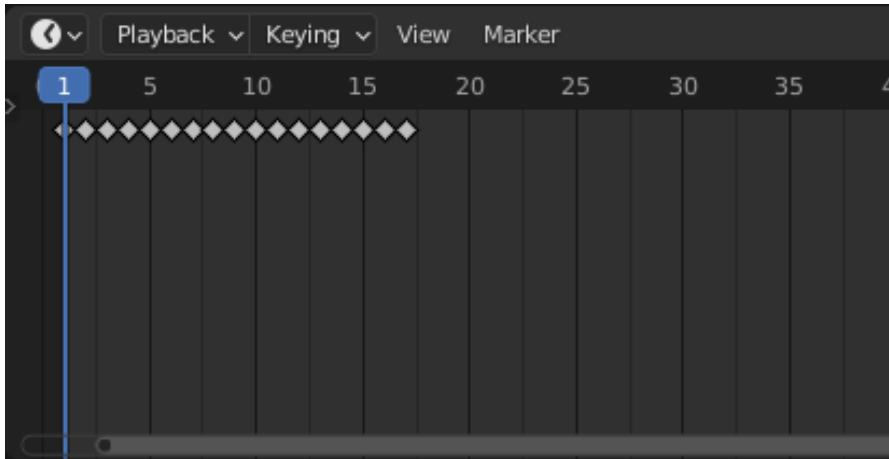


Εικόνα 8.22 Τοποθέτηση 'bones' στα αντικείμενα του Blender για τη δημιουργία animation

Έπειτα, πηγαίνοντας στις ρυθμίσεις που υπάρχουν στη δεξιά πλευρά του προγράμματος → Object Data Properties → Rigify → Generate Ring, εμφανίζεται μια μορφή στο σχήμα των οστών με όνομα rig, η οποία πρέπει να προσαρμοστεί ανάλογα με το αντικείμενο. Επιλέγοντας από το πάνω μέρος δεξιά το αντικείμενο και το 'rig' που δημιουργήθηκε, πατώντας από το μενού Object → Parent → With Automatic Weights και πηγαίνοντας σε Pose Mode τότε μπορεί το κάθε σημείο να μετακινηθεί μεμονωμένα (Εικόνα 8.23). Στη συνέχεια, κάθε φορά που γίνεται μια μετακίνηση, επιλέγοντας από το μενού Pose → Animation → Insert Keyframe και μετακινώντας τον χρόνο δημιουργείται η κίνηση του αντικειμένου (Εικόνα 8.24).

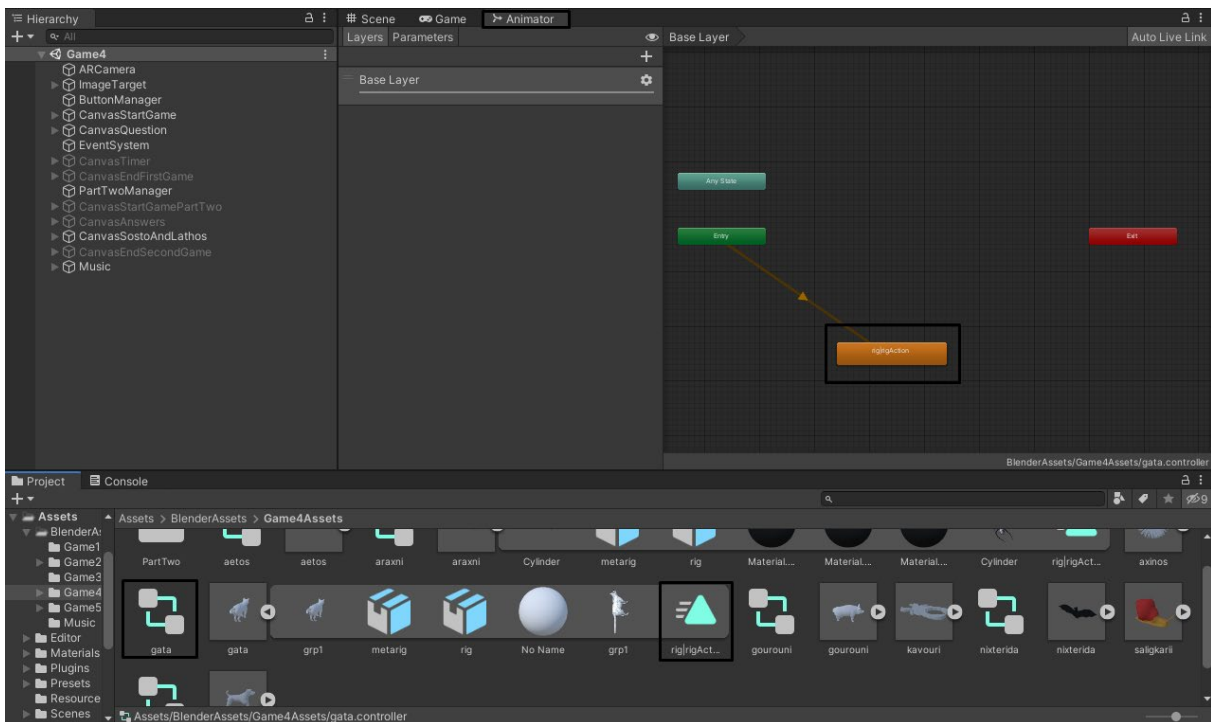


Εικόνα 8.23 Ρυθμίσεις για τη κίνηση του αντικειμένου



Εικόνα 8.24 Εισαγωγή κίνησης στα αντικείμενα

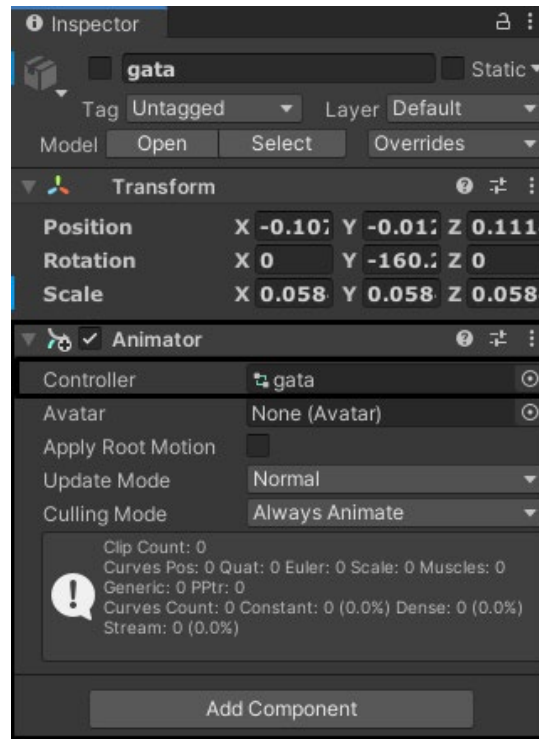
Το αντικείμενο πρέπει να γίνει export από το Blender και να τοποθετηθεί στο πρόγραμμα Unity. Στα αντικείμενα που υπάρχει κίνηση εμφανίζεται το σήμα της κίνησης με ένα όνομα και δίπλα τη λέξη Action. Πατώντας δεξί κλικ στον φάκελο με το αντικείμενο → Create → Animator Controller δημιουργείται ένα αρχείο και πατώντας πάνω του ανοίγει ένα νέο παράθυρο με όνομα Animator. Εκεί θα πρέπει να τοποθετηθεί το αντικείμενο με όνομα Action που αναφέραμε παραπάνω (Εικόνα 8.25).



Εικόνα 8.25 Ρυθμίσεις στο Unity για την πραγματοποίηση της κίνησης

Τελευταίο βήμα είναι η τοποθέτηση του αρχείου που δημιουργήθηκε μέσα στο τρισδιάστατο αντικείμενο που εμφανίζεται στην συσκευή. Επιλέγοντας το αντικείμενο → Inspector → Add

Component → Animator → Controller και τοποθετώντας το αρχείο animation, το αντικείμενο είναι έτοιμο για να αναπαράγει την κίνηση που του δόθηκε (Εικόνα 8.26).



Εικόνα 8.26 Ρυθμίσεις αντικειμένου για την πραγματοποίηση της κίνησης

### 8.3 Επίλογος

Ακολουθώντας τα παραπάνω βήματα ολοκληρώθηκε το τέταρτο παιχνίδι των υπό-κεφαλαίων ‘Η οργάνωση των πολυκύτταρων οργανισμών’ και ‘Πρόληψη ουσιών και πέψη στους ζωικούς οργανισμούς’. Ο χρήστης με τη βοήθεια του συγκεκριμένου παιχνιδιού μαθαίνει να ξεχωρίζει τα σπονδυλωτά και τα ασπόνδυλα ζώα. Επιπλέον, με τις ερωτήσεις του δεύτερου μέρους η εκμάθηση γίνεται με έναν πιο διασκεδαστικό τρόπο.

## Κεφάλαιο 9ο: Συστήματα οργάνων

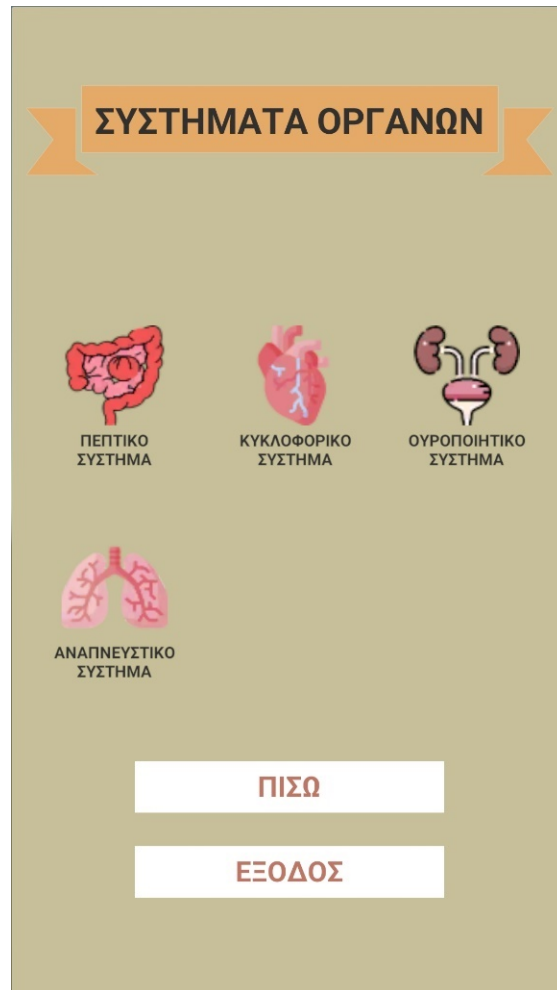
### 9.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται η περιγραφή του πέμπτου παιχνιδιού. Το πέμπτο παιχνίδι αποτελείται από τα ανθρώπινα συστήματα που βρίσκονται στις σελίδες 48,67,71,87 του σχολικού βιβλίου. Το παιχνίδι αποτελείται από ένα μενού στο οποίο είναι χωρισμένα τα τέσσερα συστήματα οργάνων. Στο κάθε παιχνίδι, ο χρήστης πρέπει να επιλέξει με τη σωστή σειρά τις ταμπέλες που εμφανίζονται στην οθόνη της συσκευής. Επιπλέον, παρουσιάζεται το παιχνίδι και όλα τα βήματα που ακολουθήθηκαν για την ολοκλήρωσή του.

### 9.2 Ανάλυση παιχνιδιού

Το πέμπτο παιχνίδι περιλαμβάνει ένα μεγάλο μέρος της ύλης του σχολικού βιβλίου. Το περιεχόμενο του παιχνιδιού ανήκει στις υπό-ενότητες ‘Πρόληψη ουσιών και πέψη στον άνθρωπο’ στη σελίδα 48, ‘Η μεταφορά και η αποβολή ουσιών στον άνθρωπο’ των σελίδων 67,71 και ‘Η αναπνοή στον άνθρωπο’ της σελίδας 87. Στόχος του παιχνιδιού είναι οι μαθητές να τοποθετήσουν τις ταμπέλες που εμφανίζονται στη σωστή σειρά. Οι μαθητές θα είναι σε θέση να κατανοήσουν την πορεία που ακολουθεί η τροφή, το αίμα, οι άχρηστες ουσίες και ο αέρας στο ανθρώπινο σώμα. Το παιχνίδι αποτελείται από ένα κεντρικό μενού, τις οδηγίες, τα παιχνίδια και τα αποτελέσματα.

Όταν πατηθεί στο κεντρικό μενού το εικονίδιο με όνομα ‘Συστήματα οργάνων’, εμφανίζεται ένα καινούριο μενού. Αυτό το μενού αποτελείται από τον τίτλο του παιχνιδιού, τέσσερα εικονίδια και δυο κουμπιά. Αν ο χρήστης επιλέξει το κουμπί με όνομα ‘Πίσω’, επιστρέφει πίσω στο κεντρικό μενού ενώ αν πατήσει το κουμπί ‘Εξοδος’ τότε τερματίζεται η εφαρμογή (Εικόνα 9.1).



Εικόνα 9.1 Μενού παιχνιδιού 'Συστήματα οργάνων'

Το καθένα από τα τέσσερα εικονίδια αντιστοιχεί σε ένα ξεχωριστό επαυξημένο περιβάλλον. Αν ο χρήστης πατήσει πάνω σε ένα εικονίδιο τότε μεταφέρεται σε μια δισδιάστατη σκηνή. Η σκηνή αποτελείται από τον τίτλο του παιχνιδιού, τις οδηγίες και ένα κουμπί 'Πάμε' (Εικόνα 9.2). Πατώντας το κουμπί, ο χρήστης θα πρέπει μεταφερθεί στο κεφάλαιο που αντιστοιχεί το σύστημα του ανθρώπου ώστε να σκανάρει το Qr code που υπάρχει στο σχολικό βιβλίο. Σκανάροντας το Qr code τότε εμφανίζεται το ανθρώπινο σώμα και ένα κουμπί 'Ας ξεκινήσουμε'. Όταν ο χρήστης πατήσει το κουμπί τότε εμφανίζεται η βασική σκηνή του παιχνιδιού. Η σκηνή αποτελείται από το ανθρώπινο σώμα και στη δεξιά πλευρά από ταμπέλες με ονόματα οργάνων (Εικόνα 9.3).



Εικόνα 9.2 Έναρξη παιχνιδιών ‘Συστήματα οργάνων’



Εικόνα 9.3 Εμφάνιση των τρισδιάστατων σκηνών του παιχνιδιού ‘Συστήματα οργάνων’

Η υλοποίηση του παιχνιδιού πραγματοποιήθηκε με τη βοήθεια τεσσάρων script με ονόματα ‘Game51’, ‘Game52’, ‘Game53’ και ‘Game54’. Όλα τα script, ακολουθούν τις ίδιες λειτουργίες και αποτελούνται από πανομοιότυπο κώδικα. Στο πάνω μέρος έχουν τοποθετηθεί όλες οι απαραίτητες μεταβλητές. Οι μεταβλητές είναι τύπου GameObject, κειμένους για τα μηνύματα σωστής και λάθος επιλογής, ακέραιοι και λίστα GameObject. Η λίστα GameObject περιλαμβάνει τις ταμπέλες που εμφανίζονται στη δεξιά πλευρά της οθόνης (Εικόνα 9.4).

```

15     public GameObject CanvasStart;
16     public GameObject PanelGo;
17
18     public GameObject stomaKaiAdenes;
19     public GameObject farigkas;
20     public GameObject oisofagos;
21     public GameObject sikoti;
22     public GameObject pagkreasKaiStomaxi;
23     public GameObject leptoEntero;
24     public GameObject paxiEntero;
25
26     public GameObject signStoma;
27     public GameObject signFarigkas;
28     public GameObject signOisofagos;
29     public GameObject signSikoti;
30     public GameObject signStomaxi;
31     public GameObject signLeptoEntero;
32     public GameObject signPaxiEntero;
33
34     public GameObject tampelaStoma;
35     public GameObject tampelaAdenes;
36     public GameObject tampelaFarigkas;
37     public GameObject tampelaOisofagos;
38     public GameObject tampelaPagkreas;
39     public GameObject tampelaSikoti;
40     public GameObject tampelaStomaxi;
41     public GameObject tampelaLeptoEntero;
42     public GameObject tampelaPaxuEntero;
43     public GameObject tampelaProktos;
44
45     public GameObject PanelSostoLathos;
46     public Text textSostoLathos;
47     public GameObject ButtonEndMessage;
48     public GameObject CanvasEnd;
49
50     public int counter=1;
51     public int lathosApantiseis=0;
52     public Text textLathosApantiseis;
53
54     public List<GameObject> buttons = new List<GameObject>();

```

Εικόνα 9.4 Μεταβλητές των scripts του πέμπτου παιχνιδιού

Στη συνέχεια, η μέθοδος Start() κρατάει τις ταμπέλες από τη λίστα με τη σειρά που έχουν τοποθετηθεί. Όλες οι ταμπέλες τοποθετούνται σε μια αντίστοιχη μεταβλητή όπου γίνεται έλεγχος αν πατήθηκε η ταμπέλα. Μόλις συμβεί αυτό, τότε καλείται η αντίστοιχη μέθοδος σε κάθε ταμπέλα (Εικόνα 9.5).

Οι ταμπέλες έχουν μια συγκεκριμένη σειρά που πρέπει να πατηθούν και η καθεμία αντιστοιχεί σε διαφορετικό αριθμό. Για παράδειγμα, η ταμπέλα που πρέπει να πατηθεί δεύτερη αντιστοιχεί στον αριθμό δύο. Με αυτόν τον τρόπο, όταν καλείται η μέθοδος που επέλεξε ο χρήστης, με μια συνθήκη if και με τη βοήθεια της μεταβλητής counter ελέγχεται αν βρίσκεται στη σωστή σειρά. Αν βρίσκεται στη σωστή σειρά τότε εξαφανίζεται η ταμπέλα από τη δεξιά πλευρά, εμφανίζεται το αντίστοιχο όργανο στο εσωτερικό του ανθρώπινου σώματος και σε μια καινούρια ταμπέλα τοποθετείται το όνομα του ανθρώπινου οργάνου. Επιπλέον, στο κάτω μέρος εμφανίζονται πληροφορίες σχετικές με το ανθρώπινο

όργανο. Ταυτόχρονα, στη μεταβλητή counter προστίθεται μια μονάδα ώστε να γνωρίζει ο counter ποια είναι η επόμενη σωστή ταμπέλα στη σειρά και εμφανίζεται ένα μήνυμα σωστής επιλογής. Αν η επιλογή του χρήστη είναι λάθος – δηλαδή αν επιλέξει την ταμπέλα αλλά με βάση τον counter δεν είναι η σωστή σειρά – τότε εμφανίζεται ένα μήνυμα λάθους και στη μεταβλητή 'lathosApantiseis' προστίθεται μια μονάδα η οποία μετράει τις λάθος απαντήσεις που έκανε ο χρήστης (Εικόνα 9.6).

```

56 void Start () {
57     Button b0 = buttons[0].GetComponent<Button>;
58     b0.onClick.AddListener(onClickButtonStoma);
59
60     Button b1 = buttons[1].GetComponent<Button>;
61     b1.onClick.AddListener(onClickButtonFariggas);
62
63     Button b2 = buttons[2].GetComponent<Button>;
64     b2.onClick.AddListener(onClickButtonOisofagos);
65
66     Button b3 = buttons[3].GetComponent<Button>;
67     b3.onClick.AddListener(onClickButtonStomaxi);
68
69     Button b4 = buttons[4].GetComponent<Button>;
70     b4.onClick.AddListener(onClickButtonSikoti);
71
72     Button b5 = buttons[5].GetComponent<Button>;
73     b5.onClick.AddListener(onClickButtonLeptoEntero);
74
75     Button b6 = buttons[6].GetComponent<Button>;
76     b6.onClick.AddListener(onClickButtonPaxiEntero);
77 }

```

Εικόνα 9.5 Μέθοδος Start() πέμπτου παιχνιδιού

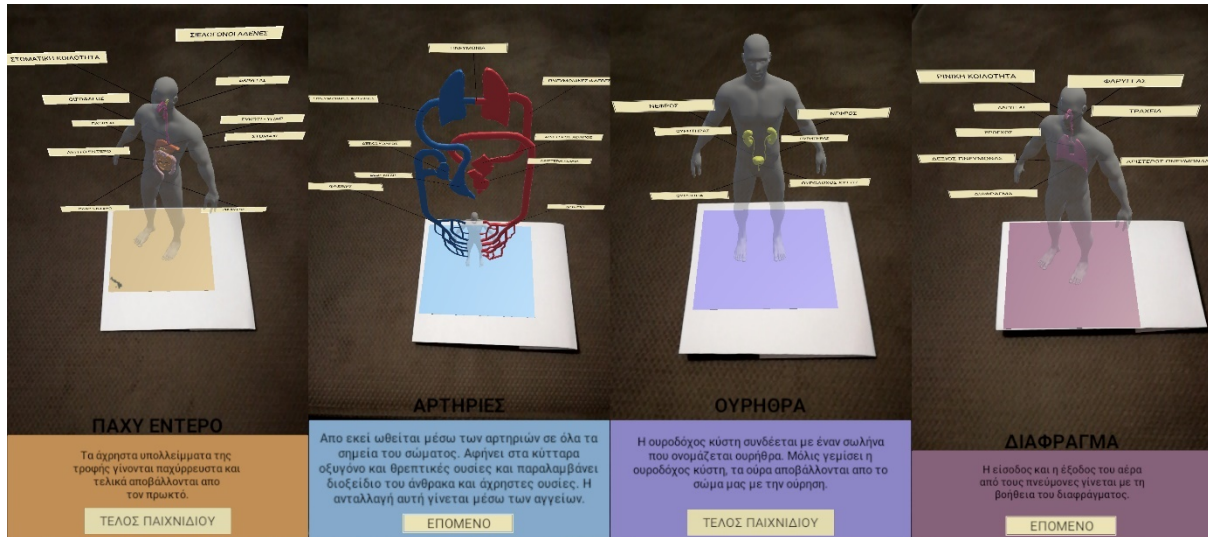
```

93 public void onClickButtonFariggas() {
94     if (counter == 2) {
95         signStoma.SetActive(false);
96         fariggas.SetActive(true);
97         signFariggas.SetActive(true);
98         tampelaFariggas.SetActive(true);
99         counter++;
100        textSostoLathos.text = "ΣΩΣΤΗ ΕΠΙΛΟΓΗ!";
101        Invoke("EnableText",1f);
102        Invoke("DisableText",3f);
103        buttons[1].SetActive(false);
104    } else {
105        textSostoLathos.text = "ΛΑΘΟΣ ΕΠΙΛΟΓΗ";
106        Invoke("EnableText",1f);
107        Invoke("DisableText",3f);
108        lathosApantiseis++;
109    }
110 }

```

Εικόνα 9.6 Μέθοδος για τον έλεγχο επιλογής μιας ταμπέλας

Μόλις ο χρήστης τοποθετήσει όλες τις ταμπέλες της δεξιάς πλευράς σωστά, τότε στο εσωτερικό του ανθρώπινου σώματος θα έχουν τοποθετηθεί όλα τα όργανα με τις αντίστοιχες ταμπέλες γύρω τους. Επιπλέον, στο κάτω μέρος της οθόνης όπου υπάρχουν οι πληροφορίες, εμφανίζεται το κουμπί ‘Επόμενο’. Ο χρήστης θα πρέπει αφού διαβάσει το περιεχόμενο, να πατήσει το κουμπί ‘Επόμενο’ μέχρι να τελειώσουν οι πληροφορίες και να εμφανιστεί το κουμπί ‘Τέλος παιχνιδιού’ (Εικόνα 9.7).



Εικόνα 9.7 Ολοκλήρωση πέμπτου παιχνιδιού

Όταν εμφανιστεί το κουμπί ‘Τέλος παιχνιδιού’ καλείται η μέθοδος `onClickButtonEndMessage()`. Στη μέθοδο αυτή εμφανίζεται μια δισδιάστατη σκηνή και τοποθετείται το κείμενο με τις λάθος απαντήσεις στο μέρος που έχει οριστεί να φαίνεται στη σκηνή (Εικόνα 9.8). Η σκηνή αυτή περιλαμβάνει τον τίτλο του παιχνιδιού, τις λάθος απαντήσεις που έκανε ο χρήστης και δυο κουμπιά. Πατώντας το κουμπί ‘Εξοδος’ εμφανίζεται το μενού του πέμπτου παιχνιδιού ενώ πατώντας το κουμπί ‘Συνέχεια’ ο χρήστης κατευθύνεται στο επόμενο παιχνίδι. Και τα τέσσερα συστήματα οργάνων έχουν τοποθετηθεί με τη σειρά που εμφανίζονται στο σχολικό βιβλίο. Πρώτα υπάρχει το πεπτικό σύστημα, το κυκλοφορικό σύστημα, το ουροποιητικό σύστημα και τέλος το αναπνευστικό σύστημα. Με αυτή τη σειρά, ο χρήστης μπορεί να περιηγηθεί από το ένα παιχνίδι στο άλλο, χωρίς να χρειαστεί να επιστρέψει στο μενού. Αυτό δεν συμβαίνει στο αναπνευστικό σύστημα το οποίο είναι το τελευταίο στη σειρά και περιλαμβάνει μόνο το κουμπί ‘Εξοδος’ (Εικόνα 9.9).

```

239     public void onClickButtonEndMessage() {
240         CanvasEnd.SetActive(true);
241         textLathosApanTiseis.text = "Συγχαρητήρια! Οι λάθος απαντήσεις σου ήταν: " + lathosApanTiseis;
242     }
    
```

Εικόνα 9.8 Μέθοδος `onClickButtonEndMessage()`



Εικόνα 9.9 Τελευταία σκηνή πέμπτου παιχνιδιού

### 9.3 Επίλογος

Η ολοκλήρωση του πέμπτου παιχνιδιού για όλα τα συστήματα οργάνων που περιλαμβάνει η σχολική ύλη του μαθήματος, ολοκληρώθηκε ακολουθώντας τα παραπάνω βήματα. Μέσω αυτού του παιχνιδιού, ο μαθητής μπορεί πιο εύκολα να μάθει τα συστήματα των ανθρώπων αφού μέσω της επαυξημένης πραγματικότητας το ανθρώπινο σώμα με τα όργανα παρουσιάζεται με έναν πιο φυσικό τρόπο.



## Κεφάλαιο 10ο: Συμπεράσματα και βελτιώσεις

Μετά την ολοκλήρωση της Δ.Ε. προέκυψαν ορισμένα αποτελέσματα και βελτιώσεις. Αρχικά, οι τεχνολογίες της εκτεταμένης πραγματικότητας μπορούν να βελτιώσουν την καθημερινότητα των χρηστών. Με το πέρασμα των χρόνων και όσο διαδίδεται αυτή η τεχνολογία, οι κλάδοι που χρησιμοποιείται γίνονται ολοένα και περισσότεροι με αποτέλεσμα οι χρήστες να ενσωματώνουν στην καθημερινότητα τους τις «βοηθητικές» εφαρμογές που έχουν δημιουργηθεί με τις τεχνολογίες XR.

Η επαυξημένη πραγματικότητα στο εκπαιδευτικό σύστημα μπορεί να παρέχει πολλά οφέλη. Η μάθηση γίνεται πιο διασκεδαστική και βελτιώνεται η παραγωγικότητα και η απόδοση των μαθητών μέσα στη τάξη. Ειδικότερα, μπορεί να χρησιμοποιηθεί για να βοηθήσει τους μαθητές με ειδικές ανάγκες αφού μέσω των εικόνων και ήχων η μάθηση και η συνεργασία των μαθητών γίνεται πιο εύκολη. Μέσω αυτών των εφαρμογών δημιουργούνται διαδραστικές εμπειρίες και όπως είπε και ο Αλμπερτ Αϊνστάιν «Η μάθηση είναι εμπειρία. Οτιδήποτε άλλο είναι απλά πληροφορία!».

Για την προσθήκη των τεχνολογιών εκτεταμένης πραγματικότητας στο σχολικό περιβάλλον θα πρέπει να υπάρχει ο κατάλληλος εξοπλισμός. Οι σχολικές τάξεις θα πρέπει να περιλαμβάνουν κινητές συσκευές ή tablet και απαραίτητος είναι ο καλός φωτισμός για να μπορεί να γίνεται σωστά η σάρωση των QR codes.

Υπάρχουν πολλές βελτιώσεις που θα μπορούσαν να γίνουν ώστε η εφαρμογή να γίνει πιο αποτελεσματική. Πρώτα απ' όλα, η εφαρμογή να μπορεί να υποστηριχθεί και από συσκευές Ios και όχι μόνο συσκευές με Android λογισμικό. Επιπλέον, η δημιουργία παραπάνω παιχνιδιών καλύπτοντας εξολοκλήρου όλη τη σχολική ύλη και η ποικιλία παιχνιδιών θα βοηθούσε τη διδασκαλία του μαθήματος. Τέλος, η βελτίωση των γραφικών, των αντικειμένων, των εικόνων και η προσθήκη περισσότερων κινούμενων εικόνων θα έκανε το περιβάλλον πιο ελκυστικό και όμορφο.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] T. Morimoto, T. Kobayashi, H. Hirata, K. Otani, M. Sugimoto, M. Tsukamoto, T. Yoshihara, M. Ueno, M. Mawatari, “XR (Extended Reality: Virtual Reality, Augmented Reality, Mixed Reality) Technology in Spine Medicine: Status Quo and Quo Vadis,” *Journal of Clinical Medicine*, 2022.
- [2] Chuah, S. H. W. (2018), ‘Why and who will adopt extended reality technology? Literature review, synthesis, and future research agenda’, working paper, University Sains Malaysia, School of Management, Department of Operation Management.
- [3] Y. Chen, Q. Wang, H. Chen, X. Song, H. Tang and M. Tian, “An overview of augmented reality technology,” *Journal of Physics: Conf. Series* 1237, 2019.
- [4] Y. Steve Chi-Yin, G. Yaoyuneyong and E. Johnson, “Augmented Reality: An Overview and Five Directions for AR in Education,” *Journal of Educational Technology Development and Exchange*, vol. 4, Iss. 1, Article 11, 2011.
- [5] Allview, “Selection of technologies related with AR,VR and MR,” 20-21, 24-25,2021.
- [6] R. Silva, J. C. Oliveira, G. A. Giraldo, “Introduction to augmented reality,” *National Laboratory for Scientific Computation*, 2003.
- [7] Mehmet Kesim, Yasin Ozarslan, “Augmented reality in education: current technologies and the potential for education,” Elsevier Ltd, 2012.
- [8] Afzal Hussain, Haad Shakeel, Faizan Hussain, Nasir Uddin, and Turab Latif Ghouri, “Unity Game Development Engine: A Technical Survey,” *University of Sindh Journal of Information and Communication Technology (USJICT)* Volume 4, Issue 2, July 2020.
- [9] “Blender” [Online]. Available: <https://blender.org>.
- [10] “Unity” [Online]. Available: <https://unity.com>.
- [11] “Vuforia Engine” [Online]. Available: <https://developer.vuforia.com>.
- [12] “Canva” [Online]. Available: <https://www.canva.com>.
- [13] “Google Fonts” [Online]. Available: <https://fonts.google.com/>.

## ΠΑΡΑΡΤΗΜΑ Α : PartTwoGame4.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
using UnityEngine.Networking;
using UnityEngine.SceneManagement;
using UnityEngine.Android;
using UnityEditor;
using System.Threading;
using System.Linq;

public class PartTwoGame4 : MonoBehaviour
{
    public GameObject canvasEndFirstGame;
    public GameObject spondilotaButton;
    public GameObject aspondilaButton;
    public GameObject spondilotaText;
    public GameObject aspondilaText;
    public GameObject panelInfo;
    public GameObject canvasStartGamePartTwo;
    public GameObject CanvasAnswers;
    public Text QuestionBook;
    public GameObject CanvasQuestion;
    public GameObject book;

    public GameObject ButtonOne;
    public Text ButtonOneText;
    public GameObject ButtonTwo;
    public Text ButtonTwoText;

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

    public float timeBetweenQuestions = 1f;
    public int counter;
    public GameObject ShowSosto;
    public GameObject ShowLathos;
    public GameObject canvasEndSecondGame;
    public GameObject CanvasTimer;

    void Start()
    {
        if (unansweredQuestions == null || unansweredQuestions.Count == 0){
            unansweredQuestions = questions.ToList<Question>();
        }

        Invoke("GetRandomQuestion",3f);
    }
}
```

```

void Update() {
    if(GameObject.Find("ButtonOneText")!=null && GameObject.Find("ButtonTwoText")!=null){
        if(QuestionBook.text == "Ποιο από τα παρακάτω ζώα ανήκει στα ασπόνδυλα;") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text=
"ΣΚΟΥΛΗΚΙ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΑΛΕΠΟΥ";
        } else if(QuestionBook.text == "Η κοινή έξοδος για το πεπτικό, το ουροποιητικό και το
αναπαραγωγικό σύστημα κάποιων σπονδυλωτών είναι:") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΠΕΠΤΙΚΗ
ΚΟΙΛΟΤΗΤΑ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΑΜΑΡΑ";
        } else if(QuestionBook.text == "Η τροφή στο σαλιγκάρι οδηγείται στο στομάχι μέσω του:") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text=
"ΠΕΠΤΙΚΟΥ ΣΩΛΗΝΑ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text=
"ΟΙΣΟΦΑΓΟΥ";
        } else if(QuestionBook.text == "Τι γίνεται στην πεπτική κοιλότητα ή στον πεπτικό σωλήνα;") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text=
"ΠΡΟΣΛΗΨΗ ΟΥΣΙΩΝ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΠΕΨΗ";
        } else if(QuestionBook.text == "Το σαλιγκάρι είναι ασπόνδυλο το οποίο διαθέτει παχιά χείλη
και μια οδοντωτή προεξοχή για να ροκανίζει την τροφή του;") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
        } else if(QuestionBook.text == "Τα πτηνά δεν έχουν δόντια και καταπίνουν πετραδάκια μαζί με
την τροφή για να την αλέθουν") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
        } else if(QuestionBook.text == "Τα ψάρια έχουν σιελογόνους αδένες") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
        } else if(QuestionBook.text == "Στα μύδια η τροφή δεν εισέρχεται μαζί με το νερό") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
        } else if(QuestionBook.text == "Ο βάτραχος ανήκει στα ασπόνδυλα") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
        } else if(QuestionBook.text == "Τα θηλαστικά χωρίζονται σε σαρκοφάγα ή φυτοφάγα") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
        } else if(QuestionBook.text == "Οι λαοί είναι σαρκοφάγα ζώα") {
            GameObject.Find("ButtonOneText").GetComponentInChildren<Text>().text= "ΣΩΣΤΟ";
            GameObject.Find("ButtonTwoText").GetComponentInChildren<Text>().text= "ΛΑΘΟΣ";
        }
    }
}

public void onClickOne () {
    if (QuestionBook.text == "Ποιο από τα παρακάτω ζώα ανήκει στα ασπόνδυλα;") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else if (QuestionBook.text == "Το σαλιγκάρι είναι ασπόνδυλο το οποίο διαθέτει παχιά χείλη και
μια οδοντωτή προεξοχή για να ροκανίζει την τροφή του;") {

```

```

        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else if (QuestionBook.text == "Τα πτηνά δεν έχουν δόντια και καταπίνουν πετραδάκια μαζί με
την τροφή για να την αλέθουν") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else if (QuestionBook.text == "Τα θηλαστικά χωρίζονται σε σαρκοφάγα ή φυτοφάγα") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else {
        Invoke("EnableLathos",1f);
        Invoke("DisableLathos",3f);
    }
}
public void onClickTwo () {
    if (QuestionBook.text == "Η τροφή στο σαλιγκάρι οδηγείται στο στομάχι μέσω του:") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else if (QuestionBook.text == "Η κοινή έξοδος για το πεπτικό, το ουροποιητικό και το
αναπαραγωγικό σύστημα κάποιων σπονδυλωτών είναι:") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else if (QuestionBook.text == "Τι γίνεται στην πεπτική κοιλότητα ή στον πεπτικό σωλήνα;") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else if (QuestionBook.text == "Τα ψάρια έχουν σιελογόνους αδένες") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else if (QuestionBook.text == "Στα μύδια η τροφή δεν εισέρχεται μαζί με το νερό") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else if (QuestionBook.text == "Ο βάτραχος ανήκει στα ασπόνδυλα") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else if (QuestionBook.text == "Οι λαγοί είναι σαρκοφάγα ζώα") {
        ShowSosto.SetActive(true);
        counter++;
        StartCoroutine(NextQuestion());
    } else {
        Invoke("EnableLathos",1f);
        Invoke("DisableLathos",3f);
    }
}
}

public void partTwoButtonGo() {

```

```

        canvasEndFirstGame.SetActive(false);
        spondilotaButton.SetActive(false);
        aspondilaButton.SetActive(false);
        spondilotaText.SetActive(false);
        aspondilaText.SetActive(false);
        panelInfo.SetActive(false);
        canvasStartGamePartTwo.SetActive(true);
    }

    public void StartGamePartTwo() {
        canvasStartGamePartTwo.SetActive(false);
        CanvasAnswers.SetActive(true);
        book.SetActive(true);
        CanvasQuestion.SetActive(true);
        CanvasTimer.SetActive(false);
    }

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

        QuestionBook.text = currentQuestion.fact;
    }

    IEnumerator NextQuestion () {
        unansweredQuestions.Remove(currentQuestion);

        yield return new WaitForSeconds(timeBetweenQuestions);
        ShowSosto.SetActive(false);

        EndGame();
        GetRandomQuestion();
    }

    public void EndGame () {
        if(counter >= 8 ){
            canvasEndSecondGame.SetActive(true);
        }
    }

    public void DisableLathos() {
        ShowLathos.SetActive(false);
    }

    public void EnableLathos() {
        ShowLathos.SetActive(true);
    }

    public void onClickEndGame() {
        SceneManager.LoadScene("Menu");
    }
}

```

## ΠΑΡΑΡΤΗΜΑ Β : Game53.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
using UnityEngine.Networking;
using UnityEngine.SceneManagement;
using UnityEngine.Android;
using UnityEditor;
using System.Threading;

public class Game53 : MonoBehaviour
{

    public GameObject CanvasStart3;
    public GameObject PanelGo3;

    public GameObject nefro1;
    public GameObject nefro2;
    public GameObject ouritiras1;
    public GameObject ouritiras2;
    public GameObject ourithra;
    public GameObject ourodoxosKisti;

    public GameObject signNefroi;
    public GameObject signOuritires;
    public GameObject signOurodoxosKisti;
    public GameObject signOurithra;

    public GameObject tampelaNefros1;
    public GameObject tampelaNefros2;
    public GameObject tampelaOuritiras1;
    public GameObject tampelaOuritiras2;
    public GameObject tampelaOurodoxosKisti;
    public GameObject tampelaOurithra;

    public GameObject EndButton;
    public GameObject CanvasEnd3;

    public int counter=1;
    public int lathosApantiseis=0;
    public Text textlathosApantiseis;
    public GameObject PanelSostoLathos;
    public Text textSostoLathos;

    public List<GameObject> buttons = new List<GameObject>();

    void Start()
    {
        Button b0 = buttons[0].GetComponent<Button>();
        b0.onClick.AddListener(onClickNefroi);
    }
}
```

```

Button b1 = buttons[1].GetComponent<Button>();
b1.onClick.AddListener(onClickOuritires);

Button b2 = buttons[2].GetComponent<Button>();
b2.onClick.AddListener(onClickOurodoxokisti);

Button b3 = buttons[3].GetComponent<Button>();
b3.onClick.AddListener(onClickOurithra);
}

public void onClickNefroi() {
    if (counter == 1) {
        nefro1.SetActive(true);
        nefro2.SetActive(true);
        signNefroi.SetActive(true);
        tampelaNefros1.SetActive(true);
        tampelaNefros2.SetActive(true);
        counter++;
        textSostoLathos.text = "ΣΩΣΤΗ ΕΠΙΛΟΓΗ!";
        Invoke("EnableText",1f);
        Invoke("DisableText",3f);
        buttons[0].SetActive(false);
    }
}

public void onClickOuritires() {
    if (counter == 2) {
        signNefroi.SetActive(false);
        ouritiras1.SetActive(true);
        ouritiras2.SetActive(true);
        signOuritires.SetActive(true);
        tampelaOuritiras1.SetActive(true);
        tampelaOuritiras2.SetActive(true);
        counter++;
        textSostoLathos.text = "ΣΩΣΤΗ ΕΠΙΛΟΓΗ!";
        Invoke("EnableText",1f);
        Invoke("DisableText",3f);
        buttons[1].SetActive(false);
    } else {
        textSostoLathos.text = "ΛΑΘΟΣ ΕΠΙΛΟΓΗ!";
        Invoke("EnableText",1f);
        Invoke("DisableText",3f);
        lathosApantiseis++;
    }
}

public void onClickOurodoxokisti() {
    if (counter == 3) {
        signOuritires.SetActive(false);
        ourodoxosKisti.SetActive(true);
        signOurodoxosKisti.SetActive(true);
        tampelaOurodoxosKisti.SetActive(true);
        counter++;
        textSostoLathos.text = "ΣΩΣΤΗ ΕΠΙΛΟΓΗ!";
        Invoke("EnableText",1f);

```

```

        Invoke("DisableText",3f);
        buttons[2].SetActive(false);
    } else {
        textSostoLathos.text = "ΛΑΘΟΣ ΕΠΙΛΟΓΗ!";
        Invoke("EnableText",1f);
        Invoke("DisableText",3f);
        lathosApantiseis++;
    }
}

```

```

public void onClickOurithra() {
    if (counter == 4) {
        signOurodoxosKisti.SetActive(false);
        ourithra.SetActive(true);
        signOurithra.SetActive(true);
        tampelaOurithra.SetActive(true);
        EndButton.SetActive(true);
        counter++;
        textSostoLathos.text = "ΣΩΣΤΗ ΕΠΙΛΟΓΗ!";
        Invoke("EnableText",1f);
        Invoke("DisableText",3f);
        buttons[3].SetActive(false);
    } else {
        textSostoLathos.text = "ΛΑΘΟΣ ΕΠΙΛΟΓΗ!";
        Invoke("EnableText",1f);
        Invoke("DisableText",3f);
        lathosApantiseis++;
    }
}

```

```

public void onClickStart3() {
    CanvasStart3.SetActive(false);
    PanelGo3.SetActive(true);
    Invoke("EnableText",1f);
    Invoke("DisableText",1f);
    lathosApantiseis=0;
}

```

```

public void onClickAsKsekinisoume3() {
    PanelGo3.SetActive(false );
    buttons[0].SetActive(true);
    buttons[1].SetActive(true);
    buttons[2].SetActive(true);
    buttons[3].SetActive(true);
}

```

```

public void DisableText() {
    PanelSostoLathos.SetActive(false);
}

```

```

public void EnableText() {
    PanelSostoLathos.SetActive(true);
}

```

```
public void onClickEndButton() {
    CanvasEnd3.SetActive(true);
    textIathosApantiseis.text = "Συγχαρητήρια! Οι λάθος απαντήσεις σου ήταν: " +IathosApantiseis;
}

public void OnClickBackToMenu2() {
    SceneManager.LoadScene("Game5_Menu");
}

public void OnClickGoNextGame3() {
    SceneManager.LoadScene("Game54");
}
}
```

## ΠΑΡΑΡΤΗΜΑ C : TimerManager.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using TMPro;
using UnityEngine.Networking;
using UnityEngine.SceneManagement;
using UnityEngine.Android;
using UnityEditor;
using System.Threading;

public class TimerManager : MonoBehaviour
{
    public Text timerMinutes;
    public Text timerSeconds;
    public Text timerSeconds100;
    private float startTime;
    public float stopTime;
    private float timerTime;
    private bool isRunning = false;
    public int minutesInt=0;
    public int secondsInt=0;

    void Update() {

        timerTime = stopTime + (Time.time - startTime);
        minutesInt = (int)timerTime / 60;
        secondsInt = (int)timerTime % 60;
        int seconds100Int = (int)(Mathf.Floor((timerTime - (secondsInt + minutesInt * 60)) * 100));

        if(isRunning) {
            timerMinutes.text = minutesInt.ToString();
            timerSeconds.text = secondsInt.ToString();
            timerSeconds100.text = seconds100Int.ToString();
        }
    }

    public void TimerStart() {

        if(!isRunning) {
            Debug.Log("START");
            isRunning = true;
            startTime = Time.time;
        }
    }

    public void TimerReset() {
        Debug.Log("RESET");
        timerMinutes.text = timerSeconds.text = timerSeconds100.text = "00";
    }
}
```

```
public void TimerStop() {
    if(isRunning) {
        Debug.Log("STOP");
        isRunning = false;
        stopTime = Time.time;
        Debug.Log(stopTime);
    }
}

public void StartGame() {
    TimerStart();
}
}
```