

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
ΠΑΙΧΝΙΔΙ ΕΚΤΕΤΑΜΕΝΗΣ
ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ
ΠΕΡΙΕΧΟΜΕΝΟΥ ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΤΗΣ
ΦΥΣΙΚΗΣ ΤΗΣ Ε' ΔΗΜΟΤΙΚΟΥ



Της φοιτήτριας
Μαρίας Ζαφειροπούλου
Αρ. Μητρώου: 144215

Επιβλέπων
Ευκλείδης Κεραμόπουλος
Αναπληρωτής Καθηγητής

Τίτλος Δ.Ε. Παιχνίδι εκτεταμένης πραγματικότητας εκπαιδευτικού περιεχομένου για το μάθημα
της φυσικής της Ε' δημοτικού
Κωδικός Δ.Ε. 20110

Όνοματεπώνυμο φοιτήτριας Μαρία Ζαφειροπούλου
Όνοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος
Ημερομηνία ανάληψης Δ.Ε. 06-04-2020
Ημερομηνία περάτωσης Δ.Ε. 18-09-2020

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

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

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

Πρόλογος

Οι εξελίξεις στην τεχνολογία τα τελευταία χρόνια υπήρξαν καθοριστικές και σηματοδότησαν σημαντικές αλλαγές σε πολλούς τομείς της ανθρώπινης δημιουργίας, από την ιατρική, την τεχνολογία τροφίμων έως τη γενετική, τη διαστημική τεχνολογία, και την τεχνολογία των ανανεώσιμων πηγών ενέργειας. Όπως ήταν αναπόφευκτο, και ο τομέας της εκπαίδευσης δέχτηκε τεράστιες επιδράσεις από την εισχώρηση των σύγχρονων επιτευγμάτων της τεχνολογίας. Μια τεχνολογία η οποία έχει γνωρίσει ιδιαίτερη ανάπτυξη την τελευταία πενταετία, και κερδίζει συνεχώς έδαφος στο τομέα της εκπαίδευσης, είναι αυτή της εκτεταμένης πραγματικότητας. Η εκτεταμένη πραγματικότητα, είναι ένας όρος ο οποίος καλύπτει όλες τις νέες και αναδυόμενες τεχνολογίες που χρησιμοποιούνται για να δημιουργήσουν καθηλωτικές ψηφιακές εμπειρίες. Ειδικότερα, περιλαμβάνει τις έννοιες της εικονικής, επαυξημένης και μικτής πραγματικότητας. Ένα σύστημα επαυξημένης πραγματικότητας συνδυάζει και ενισχύει με ψηφιακά αντικείμενα του πραγματικού κόσμου, σε αληθινό χρόνο, έχοντας ως αποτέλεσμα την ψευδαίσθηση της συνύπαρξης ψηφιακού και πραγματικού κόσμου. Η εφαρμογή της εκτεταμένης πραγματικότητας βρίσκεται ακόμα σε πρώιμα στάδια. Στόχος της ένταξής της στον τομέα της εκπαίδευσης είναι η εισαγωγή ψηφιακών πληροφοριών σε φυσικά αντικείμενα, η οπτικοποίηση εννοιών, η πρακτική εξάσκηση και κατ'επέκταση η ενίσχυση του τρόπου διδασκαλίας και η βελτίωση των επιδόσεων των μαθητών. Στην παραδοσιακή διδασκαλία, και πιο συγκεκριμένα στο χώρο των φυσικών επιστημών, οι δυσκολίες κατανόησης των εννοιών και η αδυναμία κατασκευής των νοητικών αναπαραστάσεων, δημιουργεί επιτακτική την ανάγκη χρήσης εφαρμογών εκτεταμένης πραγματικότητας. Τα οφέλη της χρήσης μια τέτοιας τεχνολογίας στον τομέα της εκπαίδευσης είναι ποικίλα. Η παρούσα διπλωματική εργασία επικεντρώνεται στη χρήση εκτεταμένης πραγματικότητας στο τομέα της εκπαίδευσης και αποτελεί πολύτιμο κομμάτι των προπτυχιακών σπουδών καθώς προσφέρει ευρύ φάσμα ερευνητικών εμπειριών.

Περίληψη

Η παρακάτω διπλωματική εργασία επικεντρώνεται στη δημιουργία ενός παιχνιδιού εκπαιδευτικού περιεχομένου για το μάθημα της Φυσικής της Ε' δημοτικού, το οποίο ακολουθεί τις νέες τεχνολογικές εξελίξεις και εισάγει στους μαθητές την έννοια της εκτεταμένης πραγματικότητας. Ο στόχος του παιχνιδιού είναι επιμορφωτικός και βασίζεται στη διαδραστική εκπαίδευση και τη βελτίωση της διδασκαλίας του μαθήματος. Για κάθε ένα κεφάλαιο του βιβλίου, εντός της διδακτέας ύλης του σχολικού έτους 2019-2020, ο μαθητής/παίκτης με χρήση κινητού τηλεφώνου ή tablet, σκανάροντας ένα qrcode, ανοίγει μια διαφορετική πόρτα με χρήση επαυξημένης πραγματικότητας. Μπαίνοντας μέσα, ακολουθεί εμβύθιση σε εικονικό περιβάλλον. Σε αυτό υπάρχουν κρυμμένα εικονικά αντικείμενα τα οποία πρέπει να συλλεχθούν προκειμένου να υλοποιηθεί στη συνέχεια το εκπαιδευτικό πείραμα, με τη χρήση εκτεταμένης πραγματικότητας, στον πραγματικό χώρο.

Λέξεις κλειδιά: Φυσική, εικονική πραγματικότητα, επαυξημένη πραγματικότητα, μικτή πραγματικότητα, εκτεταμένη πραγματικότητα, Unity, VR ProBuilder, AR Foundation, AR Subsystems, Vuforia Engine AR, XR Interaction Subsystems.

Educational extended reality game for the physics course in primary school

Maria Zafeiropoulou

Abstract

The following essay, focuses on the creation of an educational game for the physics course of the 5th grade in primary school, which follows the new technological developments and introduces the concept of extended reality to the students. The purpose of the game is educational and is based on interactive education and improvement of teaching skills. For each chapter of the book, within the curriculum of the 2019-2020 school year, the student/player uses a mobile phone or tablet to scan a qr code, which opens a different door using augmented reality. Entering each door, the user is immersed into virtual environment. There are hidden virtual objects that must be collected in order to implement the educational experiment, using extended reality, in the real space.

Keywords: Physics course, virtual reality, augmented reality, mixed reality, extended reality, Unity, VR ProBuilder, AR Foundation, AR Subsystems, Vuforia Engine AR, XR Interaction Subsystems.

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους εκείνους που με βοήθησαν να διεκπεραιώσω τη διπλωματική μου εργασία.

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή Κ. Ευκλείδη Κεραμόπουλο, γιατί μου έδωσε την ευκαιρία να ασχοληθώ με ένα ιδιαίτερα ενδιαφέρον θέμα, καθώς επίσης και για την εμπιστοσύνη που μου έδειξε. Τον ευχαριστώ θερμά για τις εξαιρετικά ωφέλιμες κριτικές παρατηρήσεις του και για την ουσιαστική βοήθεια και καθοδήγηση που μου παρείχε.

Επίσης, ευχαριστώ την Κ.Χρύσα για την αμέριστη και ουσιαστική επιστημονική βοήθεια που μου παρείχε ως εκπαιδευτικός πρωτοβάθμιας εκπαίδευσης.

Ακόμη, ένα μεγάλο και εγκάρδιο ευχαριστώ στην οικογένειά μου και στις φίλες μου, για τη στήριξή τους, καθώς και για την πεποίθηση τους ότι πιστεύουν σε εμένα.

Τέλος, θα ήθελα να απευθύνω τις ευχαριστίες μου στα μέλη της εξεταστικής επιτροπής, οι οποίοι ευγενικά δέχθηκαν να αξιολογήσουν την πτυχιακή μου εργασία.

Περιεχόμενα

Πρόλογος	iii
Περίληψη	iv
Abstract.....	v
Ευχαριστίες	vi
Περιεχόμενα.....	vii
Κατάλογος Σχημάτων.....	xii
Συντομογραφίες	xvi
Κεφάλαιο 1ο: Εισαγωγή	1
Κεφάλαιο 2ο: Οι σύγχρονες τεχνολογίες στην εκπαίδευση	4
2.1 Εισαγωγή	4
2.2 Εικονική πραγματικότητα	4
2.3 Επαυξημένη πραγματικότητα	4
2.4 Μικτή πραγματικότητα	5
2.5 Εκτεταμένη πραγματικότητα	5
2.6 Σύγχρονες τεχνολογίες στην εκπαίδευση	7
2.6.1 Σύγχρονες εκπαιδευτικές θεωρίες	7
2.6.2 Κινητή μάθηση (Mobile Learning)	8
2.6.3 Βιωματική μάθηση	9
2.7 Σύγχρονες τεχνολογίες στον χώρο των φυσικών επιστημών.....	10
2.7.1 Η εφαρμογή “XRphysics” για την αναβάθμιση της διδασκαλίας του μαθήματος της φυσικής Ε’ δημοτικού.....	10
2.8 Επίλογος	11
Κεφάλαιο 3ο: Εγκατάσταση Unity και βασικές ρυθμίσεις πλατφόρμας	12
3.1 Εισαγωγή	12
3.2 Unity και Unity Hub.....	12
3.3 Ρυθμίσεις SDK – JDK.....	12
3.4 Package manager	13
3.4.1 AR Foundation	13
3.4.2 AR Subsystems	13
3.4.3 Vuforia Engine AR.....	15

3.4.4	XR Interaction subsystems	15
3.4.5	ProBuilder.....	16
3.4.6	TextMesh Pro.....	16
3.5	Project Build settings	18
3.6	Mobile settings (Android)	20
3.7	AR settings	21
3.7.1	Vuforia Engine license Key	22
3.7.2	Δημιουργία QRcodes από Generator.....	25
3.7.3	Δημιουργία βάσης από targets και εισαγωγή της στη Unity.....	26
3.8	Testing	28
3.9	Επίλογος.....	28
Κεφάλαιο 4ο: Βασικός κορμός παιχνιδιού-Panels		29
4.1	Εισαγωγή.....	29
4.2	Φυσικά Ε' δημοτικού – Επιλογή πειραμάτων	29
4.3	Αρχική σκηνή	29
4.3.1	Κείμενα και γραμματοσειρά	31
4.3.2	Εικόνες.....	32
4.3.3	Κουμπιά και scripts	33
4.4	Σκηνές “About”	34
4.5	Μουσική και script.....	35
4.6	Επίλογος.....	36
Κεφάλαιο 5ο: Εφαρμογή πειράματος για το κεφάλαιο Υλικά σώματα		37
5.1	Εισαγωγή.....	37
5.2	Σελίδα βιβλίου 24	37
5.3	Οι σκηνές του πειράματος – Υλικά σώματα.....	38
5.4	Σκηνή: Ylikabout	39
5.5	Σκηνή: Fe2-maza	40
5.6	Σκηνή: VR4	40
5.6.1	Ιεραρχία αντικειμένων VR4.....	41
5.6.2	AR πόρτα – VR4	42
5.6.3	Εικονικό περιβάλλον VR4	42
5.6.4	Εικονικά αντικείμενα VR4	43
5.6.5	Μουσική VR4	44
5.6.6	Εναλλαγή σκηνής VR4.....	45

5.7	Σκηνή: TestScene4	46
5.8	Σκηνή: 4.animation	47
5.8.1	Ιεραρχία αντικειμένων – 4.animation	48
5.8.2	Script “basicvrFOUR”	48
5.9	Σκηνή: symperasmataYlikaSwmata	51
5.10	Επίλογος	52
Κεφάλαιο 6ο: Εφαρμογή πειράματος για το κεφάλαιο Ενέργεια		53
6.1	Εισαγωγή	53
6.2	Σελίδα βιβλίου 46	53
6.3	Οι σκηνές του πειράματος – Ενέργεια	54
6.4	Σκηνή: EnergeiaAbout	55
6.5	Σκηνή: Fe3-energeia	56
6.6	Σκηνή: VR6	56
6.6.1	Εικονικό περιβάλλον και εικονικά αντικείμενα VR6	58
6.6.2	Εναλλαγή σκηνής VR6	59
6.7	Σκηνή: TestScene6	60
6.8	Σκηνή: 6.animation	61
6.8.1	Ιεραρχία αντικειμένων – 6.animation	62
6.8.2	Script “sixanim”	63
6.9	Σκηνή: symperasmataEnergeia	66
6.10	Επίλογος	67
Κεφάλαιο 7ο: Εφαρμογή πειράματος για το κεφάλαιο Πεπτικό σύστημα		68
7.1	Εισαγωγή	68
7.2	Σελίδα βιβλίου 66	68
7.3	Οι σκηνές του πειράματος – Πεπτικό σύστημα	69
7.4	Σκηνή: PeptikoAbout	70
7.5	Σκηνή: Fe3-peptiko	71
7.6	Σκηνή: VR3	71
7.6.1	Εικονικό περιβάλλον και εικονικά αντικείμενα VR3	73
7.6.2	Εναλλαγή σκηνής VR3	75
7.7	Σκηνή: TestScene3	75
7.8	Σκηνή: 3.animation	76
7.8.1	Ιεραρχία αντικειμένων – 3.animation	77
7.8.2	Η λειτουργία του πειράματος μέσα από τα scripts (Πεπτικό σύστημα)	78

7.8.3	Πείραμα-Βήμα 1.....	80
7.8.4	Πείραμα-Βήμα 2.....	81
7.8.5	Πείραμα-Βήματα 3 και 5.....	83
7.8.6	Πείραμα-Βήμα 4.....	85
7.9	Σκηνή: symperasmataPeptiko.....	86
7.10	Επίλογος.....	86
Κεφάλαιο 8ο: Εφαρμογή πειράματος για το κεφάλαιο Θερμότητα.....		87
8.1	Εισαγωγή.....	87
8.2	Σελίδα βιβλίου 74.....	87
8.3	Οι σκηνές του πειράματος – Θερμότητα.....	88
8.4	Σκηνή: ThermotitaAbout.....	89
8.5	Σκηνή: Fe2-thermotita.....	90
8.6	Σκηνή: VR2.....	90
8.6.1	Εικονικό περιβάλλον και εικονικά αντικείμενα VR2.....	92
8.6.2	Εναλλαγή σκηνής VR2.....	93
8.7	Σκηνή: TestScene2.....	94
8.8	Σκηνή: 2.animation.....	94
8.8.1	Ιεραρχία αντικειμένων – 2.animation.....	96
8.8.2	Η λειτουργία του πειράματος μέσα από τα scripts (Θερμότητα).....	97
8.9	Σκηνή: symperasmataThermotita.....	101
8.10	Επίλογος.....	101
Κεφάλαιο 9ο: Εφαρμογή πειράματος για το κεφάλαιο Ηλεκτρισμός.....		102
9.1	Εισαγωγή.....	102
9.2	Σελίδα βιβλίου 112.....	102
9.3	Οι σκηνές του πειράματος – Ηλεκτρισμός.....	103
9.4	Σκηνή: HlektrismosAbout.....	104
9.5	Σκηνή: Fe6-ilektrismos.....	105
9.6	Σκηνή: VR1.....	105
9.6.1	Εικονικό περιβάλλον και εικονικά αντικείμενα VR1.....	107
9.6.2	Εναλλαγή σκηνής VR1.....	108
9.7	Σκηνή: TestScene1.....	108
9.8	Σκηνή: 1.animation.....	109
9.8.1	Ιεραρχία αντικειμένων – 1.animation.....	110
9.8.2	Η λειτουργία του πειράματος μέσα από τα scripts (Ηλεκτρισμός).....	111

9.9	Σκηνή: symperasmataIlektrismos	114
9.10	Επίλογος	115
Κεφάλαιο 10ο:	Εφαρμογή πειράματος για το κεφάλαιο Φως	116
10.1	Εισαγωγή	116
10.2	Σελίδα βιβλίου 131.....	116
10.3	Οι σκηνές του πειράματος – Φως	117
10.4	Σκηνή: FwsAbout	118
10.5	Σκηνή: Fe2-fws	118
10.6	Σκηνή: VR5	119
10.6.1	Εικονικό περιβάλλον και εικονικά αντικείμενα VR5.....	121
10.6.2	Εναλλαγή σκηνής VR5.....	122
10.7	Σκηνή: TestScene5	123
10.8	Σκηνή: 5.animation	123
10.8.1	Ιεραρχία αντικειμένων – 5.animation.....	124
10.8.2	Τα spot lights του φακού	125
10.8.3	Η λειτουργία του πειράματος μέσα από τα scripts (Φως).....	126
10.9	Σκηνή: symperasmatafws	129
10.10	Επίλογος	129
Κεφάλαιο 11ο:	Συμπεράσματα και προτάσεις βελτίωσης	130
	ΒΙΒΛΙΟΓΡΑΦΙΑ	131
	ΠΑΡΑΡΤΗΜΑ Α : Script “Water FX.cs”	132
	ΠΑΡΑΡΤΗΜΑ Β : Script “Water Volume.cs”	141
	ΠΑΡΑΡΤΗΜΑ C : Script “FloatingObject.cs”	148
	ΠΑΡΑΡΤΗΜΑ D : Script “Straw.cs”	152
	ΠΑΡΑΡΤΗΜΑ E : Script “Timer.cs”	157

Κατάλογος Σχημάτων

Σχήμα 2.1: VR vs AR vs MR.....	5
Σχήμα 2.2: 3DoF vs 6DoF	5
Σχήμα 2.3: Μοντέλο ταξινόμησης διδακτικών στόχων (Bloom, 1956)	9
Σχήμα 3.1: Ρυθμίσεις JDK, SDK και NDK.....	13
Σχήμα 3.2: AR Foundation	14
Σχήμα 3.3: AR Subsystems	14
Σχήμα 3.4: Vuforia Engine AR.....	15
Σχήμα 3.5: XR Interaction Subsystems	16
Σχήμα 3.6: ProBuilder	17
Σχήμα 3.7: TextMesh Pro	17
Σχήμα 3.8: Player settings.....	18
Σχήμα 3.9: Οι σκηνές του project.....	18
Σχήμα 3.10: Ιεραρχία όλων των σκηνών του project	19
Σχήμα 3.11: Mobile settings Unity	20
Σχήμα 3.12: Σύνδεση συσκευής κινητού με Unity	20
Σχήμα 3.13: Γενικά components AR camera.....	21
Σχήμα 3.14: Παραμετροποίηση στοιχείων AR camera	21
Σχήμα 3.15: Vuforia Behaviour Script.....	22
Σχήμα 3.16: Vuforia Engine/Get Development Key	22
Σχήμα 3.17: Vuforia Engine/add development license key	23
Σχήμα 3.18: Vuforia Engine/ptyxiaki_finalkey	23
Σχήμα 3.19: Vuforia Engine/license key	24
Σχήμα 3.20: VuforiaConfiguration/add License	24
Σχήμα 3.21: Δημιουργία image targets	25
Σχήμα 3.22: Τα 6 image target του project.....	26
Σχήμα 3.23: Vuforia engine/add database	26
Σχήμα 3.24: Vuforia Engine/high_class database	27
Σχήμα 3.25: Components του Image target.....	27
Σχήμα 3.26: Στιγμιότυπο οθόνης για την επιβεβαίωση της ορθότητας των AR ρυθμίσεων	28
Σχήμα 4.1: Αρχική σκηνή παιχνιδιού.....	30
Σχήμα 4.2: Ιεραρχία αντικειμένων αρχικής σκηνής	30
Σχήμα 4.3: Google Fonts-Greek.....	31
Σχήμα 4.4: Ρυθμίσεις δημιουργίας γραμματοσειράς unity	31
Σχήμα 4.5: Ρυθμίσεις εικόνων	32
Σχήμα 4.6: Μετατροπή εικόνων .png σε sprite.....	32
Σχήμα 4.7: Script "menu"	33
Σχήμα 4.8: Όλες οι σκηνές about	34
Σχήμα 4.9: Audio Components.....	35
Σχήμα 4.10: BG Sound Script.....	36
Σχήμα 5.1: Σελίδα βιβλίου 24	37
Σχήμα 5.2: Σκηνές πειράματος σελίδας 24.....	38

Σχήμα 5.3: Σκηνή "YlikaAbout"	39
Σχήμα 5.4: Σκηνή fe2-maza	40
Σχήμα 5.5: Πόρτα επαυξημένης πραγματικότητας VR4	41
Σχήμα 5.6: Ιεραρχία αντικειμένων VR4	42
Σχήμα 5.7: Δωμάτιο VR4	43
Σχήμα 5.8: Υλικά πειράματος (Υλικά σώματα)	43
Σχήμα 5.9: Ρυθμίσεις ήχου VR4	44
Σχήμα 5.10: Script ήχου "audioVRFOUR"	44
Σχήμα 5.11: script "Dotvrndour"	45
Σχήμα 5.12: Components αντικειμένων VR4	45
Σχήμα 5.13: Το "canvascriptvrfour" script	46
Σχήμα 5.14: Σκηνή testscene4	46
Σχήμα 5.15: Οι οθόνες της σκηνής "4.animation"	47
Σχήμα 5.16: Το script "basicvrFOUR"	49
Σχήμα 5.17: Συμπληρωμένα πεδία component basicvrFOUR του basicscript1	50
Σχήμα 5.18: Οθόνη με την επιλογή του κουμπιού «συνδυασμός1»	50
Σχήμα 5.19: Το script "symper"	51
Σχήμα 5.20: Σκηνή-symperasmataYlikaSwmata	51
Σχήμα 5.21: Το script "ArxikiYlikaSwmatwn"	52
Σχήμα 6.1: Σελίδα βιβλίου 46	53
Σχήμα 6.2: Σκηνές πειράματος σελίδας 46	54
Σχήμα 6.3: Σκηνή "EnergeiaAbout"	55
Σχήμα 6.4: Σκηνή fe3-energeia	56
Σχήμα 6.5: Πόρτα επαυξημένης πραγματικότητας VR6 σκηνής	57
Σχήμα 6.6: Ιεραρχία αντικειμένων VR6	57
Σχήμα 6.7: Δωμάτιο VR6	58
Σχήμα 6.8: Υλικά πειράματος (Ενέργεια)	58
Σχήμα 6.9: Ρυθμίσεις ήχου VR6	59
Σχήμα 6.10: Το script "csSixx"	60
Σχήμα 6.11: Σκηνή testscene6	60
Σχήμα 6.12: Το κύκλωμα της σελίδας 46 του σχολικού βιβλίου στη σκηνή "6.animation"	61
Σχήμα 6.13: Καταστάσεις σύνδεση/αποσύνδεση	61
Σχήμα 6.14: Ιεραρχία αντικειμένων "6.animation"	62
Σχήμα 6.15: Το script "sixanim"	63
Σχήμα 6.16: inspector του empty "gameobject" της σκηνής "6.animation"	64
Σχήμα 6.17: pointlight1 (κύκλωμα σελίδα σχολικού βιβλίου 46)	64
Σχήμα 6.18: pointlight2 (κύκλωμα σελίδα σχολικού βιβλίου 46)	65
Σχήμα 6.19: paperclip1-Διακόπτης κλειστός (κύκλωμα σελίδα σχολικού βιβλίου 46)	65
Σχήμα 6.20: paperclip2-Διακόπτης ανοιχτός (κύκλωμα σελίδα σχολικού βιβλίου 46)	65
Σχήμα 6.21: Το script " tscsix"	66
Σχήμα 6.22: Σκηνή-symperasmataEnergeia	66
Σχήμα 6.23: Το script "entostar"	67
Σχήμα 7.1: Σελίδα βιβλίου 66	68

Σχήμα 7.2: Σκηνές πειράματος σελίδας 66.....	69
Σχήμα 7.3: Σκηνή “PeptikoAbout”	70
Σχήμα 7.4: Σκηνή fe3-peptiko	71
Σχήμα 7.5: Πόρτα επαυξημένης πραγματικότητας VR3 σκηνής.....	72
Σχήμα 7.6: Ιεραρχία αντικειμένων VR3	72
Σχήμα 7.7: Δωμάτιο VR3.....	73
Σχήμα 7.8: Υλικά πειράματος (Πεπτικό σύστημα)	74
Σχήμα 7.9: Ρυθμίσεις ήχου VR3	74
Σχήμα 7.10: Σκηνή testscene3	75
Σχήμα 7.11: Αρχική οθόνη σκηνής “3.animation”	76
Σχήμα 7.12: Οι οθόνες μετά από τη σωστή επιλογή του κάθε υλικού του πειράματος.....	77
Σχήμα 7.13: Η ιεραρχία των αντικειμένων της σκηνής 3.animation.....	78
Σχήμα 7.14: Το script “setactine”	79
Σχήμα 7.15: Παραμετροποίηση των πεδίων του script “setactine”	79
Σχήμα 7.16: Το script “all”	80
Σχήμα 7.17: Παραμετροποίηση των πεδίων του script “all”	80
Σχήμα 7.18: Το script “disableCollider”	80
Σχήμα 7.19: Το script “onClickEvents”	81
Σχήμα 7.20: Το script “oilclick”	82
Σχήμα 7.21: Ο inspector του pool	82
Σχήμα 7.22: Ο inspector του dynamicObjects	83
Σχήμα 7.23: Τα πεδία του script “Straw”	83
Σχήμα 7.24: Το script “soap”	85
Σχήμα 7.25: Πεδία script “soap”	85
Σχήμα 7.26: Σκηνή-symperasmataPeptiko	86
Σχήμα 8.1: Σελίδες βιβλίου 74-75	87
Σχήμα 8.2: Σκηνές πειράματος σελίδας 74.....	88
Σχήμα 8.3: Σκηνή “ThermotitaAbout”	89
Σχήμα 8.4: Σκηνή fe2-themrotita	90
Σχήμα 8.5: Πόρτα επαυξημένης πραγματικότητας VR2 σκηνής.....	91
Σχήμα 8.6: Ιεραρχία αντικειμένων VR2	91
Σχήμα 8.7: Δωμάτιο VR2.....	92
Σχήμα 8.8: Υλικά πειράματος (Θερμότητα)	93
Σχήμα 8.9: Σκηνή testscene2	94
Σχήμα 8.10: Οι οθόνες του πειράματος-2.animation	95
Σχήμα 8.11: Ιεραρχία αντικειμένων “2.animation”	96
Σχήμα 8.12: Ο inspector του Text-timer	97
Σχήμα 8.13: Το script “Txt Dis”	98
Σχήμα 8.14: Οι ρυθμίσεις του Particle System	99
Σχήμα 8.15: Το script “psoff”	99
Σχήμα 8.16: Ο inspector του button «Αρχή»	100
Σχήμα 8.17: Το script “Turn Everything On”	100
Σχήμα 9.1: Σελίδες βιβλίου 112-113	102

Σχήμα 9.2: Σκηνές πειράματος σελίδας 112	103
Σχήμα 9.3: Σκηνή "IlektrismosAbout"	104
Σχήμα 9.4: Σκηνή fe6-ilektrismos	105
Σχήμα 9.5: Πόρτα επαυξημένης πραγματικότητας VR1 σκηνής.....	106
Σχήμα 9.6: Ιεραρχία αντικειμένων VR1	106
Σχήμα 9.7: Δωμάτιο VR1.....	107
Σχήμα 9.8: Υλικά πειράματος (Ηλεκτρισμός)	107
Σχήμα 9.9: Σκηνή testscene1	108
Σχήμα 9.10: Αρχική οθόνη πειράματος σελίδα 112	109
Σχήμα 9.11: Οι οθόνες του πειράματος μετά την επιλογή όλων των υλικών	109
Σχήμα 9.12: Η ιεραρχία των αντικειμένων της σκηνής 1.animation.....	110
Σχήμα 9.13: Ο inspector του "Cube-ground"	111
Σχήμα 9.14: Το script "Rotate"	111
Σχήμα 9.15: Scroll right- scroll left (1.animation).....	112
Σχήμα 9.16: Το script "Disable"	113
Σχήμα 9.17: Τα scripts "ObjectTesting" και "NoObjectTesting"	113
Σχήμα 9.18: Ο inspector του button1 (1.animation).....	114
Σχήμα 9.19: Σκηνή-symperasmatalelektrismos	115
Σχήμα 10.1: Σελίδες βιβλίου 131-132	116
Σχήμα 10.2: Σκηνές πειράματος σελίδας 131	117
Σχήμα 10.3: Σκηνή "FwsAbout"	118
Σχήμα 10.4: Σκηνή fe2-fws.....	119
Σχήμα 10.5: Πόρτα επαυξημένης πραγματικότητας VR5 σκηνής.....	119
Σχήμα 10.6: Ιεραρχία αντικειμένων VR5	120
Σχήμα 10.7: Δωμάτιο VR5.....	121
Σχήμα 10.8: Υλικά πειράματος (Φως)	122
Σχήμα 10.9: Σκηνή testscene5	123
Σχήμα 10.10: Οι οθόνες του πειράματος (σκηνή: 5.animation)	124
Σχήμα 10.11: Η ιεραρχία των αντικειμένων της σκηνής 5.animation.....	125
Σχήμα 10.12: Το spot light "Spot Light-50 LIGO"	126
Σχήμα 10.13: Ο inspector του button1.....	127
Σχήμα 10.14: Το script "basicvrFIVE"	128
Σχήμα 10.15: Σκηνή-symperasmatafws.....	129

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΠΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
VR	Virtual Reality
AR	Augmented Reality
MR	Mixed Reality
XR	Extended Reality
UI	User Interface
π.χ.	Παραδείγματος χάρη
vs	Versus

Κεφάλαιο 1ο: Εισαγωγή

Η παρούσα διπλωματική εργασία έχει τον τίτλο «Παιχνίδι εκτεταμένης πραγματικότητας εκπαιδευτικού περιεχομένου για το μάθημα της Φυσικής της Ε' δημοτικού». Στόχος της διπλωματικής εργασίας είναι η χρήση της εκτεταμένης πραγματικότητας για τη βελτίωση της διδασκαλίας του μαθήματος της Φυσικής της Ε' Δημοτικού. Η ιδέα της συγκεκριμένης διπλωματικής εργασίας βασίζεται στην αξιοποίηση της εκτεταμένης πραγματικότητας σε πραγματικές σχολικές συνθήκες για τη διδασκαλία πολύπλοκων εννοιών και φαινομένων σχετικά με το μάθημα της φυσικής της Ε' δημοτικού. Ως εκ τούτου, δύναται να συνεισφέρει στην καλύτερη κατανόηση της επίδρασης αντίστοιχων τεχνολογιών στην κατάκτηση της γνώσης, κυρίως αυτής που άπτεται στο χώρο των φυσικών επιστημών, ενώ παράλληλα αναδεικνύει τα οφέλη που έχουν οι μαθητές κατά την αλληλεπίδραση με ένα περιβάλλον, όπου συνυπάρχουν εικονικά και πραγματικά στοιχεία.

Τα τελευταία χρόνια έχουν εμφανιστεί νέες τεχνολογίες, οι οποίες για λόγους ευκολίας, έχουν γίνει γνωστές με τα αρχικά τους γράμματα, όπως είναι η VR (Virtual reality), η AR (Augmented Reality), η MR (Mixed reality) και η XR (Extended Reality). Ενώ όλες διαφέρουν μεταξύ τους, βασίζονται στις ίδιες τεχνολογίες. Η Virtual Reality ή αλλιώς εικονική πραγματικότητα, είναι μία προσομοίωση που δημιουργείται εξολοκλήρου από ένα υπολογιστικό σύστημα. Η Augmented reality ή αλλιώς επαυξημένη πραγματικότητα, είναι μία ζωντανή, άμεση ή έμμεση προβολή ενός φυσικού αντικείμενου που βρίσκεται στον χώρο, στο οποίο γίνονται τροποποιήσεις σε πραγματικό χρόνο με την χρήση κάποιου υπολογιστικού συστήματος. Η Mixed Reality ή αλλιώς μικτή πραγματικότητα, είναι ο συνδυασμός του αληθινού και του ψηφιακού κόσμου από τους οποίους δημιουργείται κάτι ενδιάμεσο. Η Extended reality ή αλλιώς είναι και ο νεότερος όρος, και περιλαμβάνει και συνδυάζει όλες τις παραπάνω τεχνολογίες, καθώς αναφέρετε στο πως αλληλεπιδρούν τα φυσικά αντικείμενα του χώρου με τον χρήστη και τις κινήσεις του.

Οι παραπάνω τεχνολογίες, έχουν ωριμάσει στο βαθμό που πλέον εφαρμόζονται σε ένα ευρύ φάσμα πεδίων, όπως στον εκπαιδευτικό τομέα. Στη σημερινή εποχή, γίνεται σημαντική προσπάθεια ενσωμάτωσης νέων τεχνολογιών στη μάθηση, για την προσφορά πλούσιας εμπειρίας, την ενεργό συμμετοχή των μαθητών και τη βαθύτερη κατανόηση εννοιών, οι οποίες δεν μπορούν εύκολα να οπτικοποιηθούν στο πραγματικό περιβάλλον. Ιδιαίτερο ρόλο σε αυτή την προσπάθεια παίζουν οι έννοιες της βιωματικής και κινητής μάθησης, προκειμένου να ενισχυθεί η συναισθηματική εμπλοκή των μαθητών στη διδακτική διαδικασία και η γνωστική τους ικανότητα.

Η εφαρμογή «XRphysics», συνδυάζει τις μεθόδους της κινητής και βιωματικής μάθησης και χρησιμοποιεί σύγχρονες τεχνολογίες όπως αυτή της εικονικής και επαυξημένης πραγματικότητας. Μέσω της εφαρμογής «XRphysics», δίνεται στους εκπαιδευτικούς η δυνατότητα εντάξης στη διδασκαλία του μαθήματος «Φυσικά Ε' δημοτικού» μιας καινοτόμου μεθόδου διδασκαλίας, ικανής να αυξήσει το ενδιαφέρον των μαθητών και να τους υποβοηθήσει να κατανοήσουν πολύπλοκα φαινόμενα και έννοιες της φυσικής Ε' δημοτικού.

Η Διπλωματική Εργασία χωρίζεται σε έντεκα κεφάλαια. Κάθε κεφάλαιο χωρίζεται σε επιμέρους ενότητες.

Στο πρώτο κεφάλαιο «Εισαγωγή», παρουσιάζεται ο στόχος και η σημαντικότητα της διπλωματικής εργασίας, και κατ'επέκταση της εφαρμογής του παιχνιδιού. Παράλληλα, γίνεται μια σύντομη περιγραφή του κάθε κεφαλαίου.

Στο δεύτερο κεφάλαιο «Οι σύγχρονες τεχνολογίες στην εκπαίδευση», παρουσιάζεται η έννοια της εκτεταμένης πραγματικότητας η οποία περιλαμβάνει και τις έννοιες της εικονικής, επαυξημένης και μικτής πραγματικότητας. Επίσης, γίνεται εκτενής αναφορά της εφαρμογής των παραπάνω τεχνολογιών στο χώρο της εκπαίδευσης και ειδικότερα μέσω της εφαρμογής XRphysics.

Στο τρίτο κεφάλαιο «Εγκατάσταση Unity και βασικές ρυθμίσεις πλατφόρμας», αναλύονται οι αρχικές παραμετροποιήσεις των κομβικών στοιχείων της πλατφόρμας unity. Πιο συγκεκριμένα, παρουσιάζονται όλες οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής του παιχνιδιού.

Στο τέταρτο κεφάλαιο «Βασικός κορμός παιχνιδιού-Panels», περιγράφονται λεπτομερώς οι αρχικές σκηνές του project οι οποίες αποτελούν τα θεμέλια της εφαρμογής. Αυτές αποτελούνται από την αρχική σελίδα, τις σελίδες των κεφαλαίων και των διαθέσιμων πειραμάτων καθώς επίσης και της σελίδας κλεισίματος της εφαρμογής. Γίνεται μια μικρή αναφορά στα κεφάλαια του βιβλίου καθώς και στη διδακτέα ύλη για το σχολικό έτος 2019-2020. Επιπλέον, σε αυτό το κεφάλαιο περιγράφεται και η διαδικασία της εισαγωγής και εναλλαγής της μουσικής ανάμεσα στις διαφορετικές σκηνές.

Οι τίτλοι των κεφαλαίων πέντε ως δέκα αποτελούνται από τους αντίστοιχους τίτλους των κεφαλαίων του βιβλίου. Πιο συγκεκριμένα, το πέμπτο κεφάλαιο «Εφαρμογή πειράματος για το κεφάλαιο Υλικά σώματα», το έκτο κεφάλαιο «Εφαρμογή πειράματος για το κεφάλαιο Ενέργεια», το έβδομο κεφάλαιο «Εφαρμογή πειράματος για το κεφάλαιο Πεπτικό σύστημα», το όγδοο κεφάλαιο «Εφαρμογή πειράματος για το κεφάλαιο Θερμότητα», το ένατο κεφάλαιο «Εφαρμογή πειράματος για το κεφάλαιο Ηλεκτρισμός» και τέλος το δέκατο κεφάλαιο «Εφαρμογή πειράματος για το κεφάλαιο Φως». Η δομή αυτών των κεφαλαίων είναι σχεδόν ίδια, αφού κάθε ένα από αυτά αντιστοιχούν και σε ένα πείραμα. Σε αυτά τα κεφάλαια αναλύεται όλη η μεθοδολογία της δημιουργίας του κάθε πειράματος ξεχωριστά.

Τέλος, η διπλωματική εργασία ολοκληρώνεται με το ενδέκατο κεφάλαιο «Συμπεράσματα και προτάσεις βελτίωσης» στο οποίο παρουσιάζονται κάποια προσωπικά συμπεράσματα και προτάσεις για μελλοντική επέκταση και βελτίωση της εφαρμογής.

Κεφάλαιο 2ο: Οι σύγχρονες τεχνολογίες στην εκπαίδευση

2.1 Εισαγωγή

Τα τελευταία χρόνια, μια από τις πιο αναδυόμενες τεχνολογίες παγκοσμίως είναι αυτή της εκτεταμένης πραγματικότητας (Extended Reality). Ο όρος XR μπορεί να χαρακτηριστεί ως ένας όρος «ομπρέλα» καθώς περιλαμβάνει και τις τεχνολογίες της εικονικής πραγματικότητας (Virtual Reality), της επαυξημένης πραγματικότητας (Augmented Reality) και της μικτής πραγματικότητας (Mixed Reality). Κάθε μια από αυτές τις νέες τεχνολογίες, προέρχεται από διαφορετικό μέρος και επιδιώκει να κάνει διαφορετικά πράγματα. Ωστόσο, όλες χρησιμοποιούν παρόμοιες τεχνολογίες. Στο κεφάλαιο αυτό, αναφέρονται οι παραπάνω τεχνολογίες και γίνεται εκτενής αναφορά στην εφαρμογή της XR και των σύγχρονων τεχνολογιών στην εκπαίδευση, και πιο συγκεκριμένα στο χώρο των φυσικών επιστημών, αναδεικνύοντας τη σημασία της βιωματικής προσέγγισης στη μάθηση. Το κεφάλαιο ολοκληρώνεται με την παρουσίαση των σύγχρονων τεχνολογιών που χρησιμοποιεί η εφαρμογή «XRphysics» για να ενισχύσει τις νέες εκπαιδευτικές μεθόδους.

2.2 Εικονική πραγματικότητα

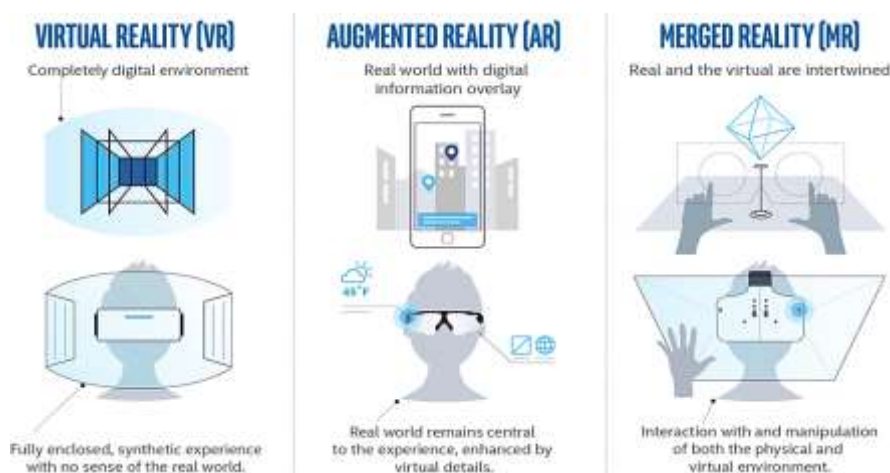
Η επιστημονική περιοχή της VR, υπάρχει και εξελίσσεται εδώ και πολλά χρόνια ως κλάδος της επιστήμης της πληροφορικής. Η VR είναι η επιστήμη που έχει ως στόχο την ανάπτυξη κατάλληλων συσκευών και συστημάτων τα οποία έχουν ως απώτερο στόχο την πλήρη αποκοπή ενός χρήστη από το πραγματικό περιβάλλον και την εμβύθισή του σε ένα εικονικό περιβάλλον[1]. Για να το επιτύχει αυτό, ένα σύστημα πρέπει να αποκόψει κάθε πληροφορία που προέρχεται από τον πραγματικό κόσμο, από τα αισθητήρια του χρήστη, και να την αντικαταστήσει με εικονική. Το πρώτο και σημαντικότερο κανάλι αλληλεπίδρασης είναι η όραση και ακολουθούν η ακοή και η αφή, ενώ πρόσφατα έχουν αναπτυχθεί και συστήματα για την παραγωγή συνθετικών ερεθισμάτων όσφρησης και γεύσης. Για να επιτευχθεί ένας ικανοποιητικός βαθμός εμβύθισης χρησιμοποιούνται εξειδικευμένες συσκευές όπως είναι οι θόνοι προσαρμογής στο κεφάλι (Head Mounted Displays), η πανκατευθυντική διοπτρική οθόνη (Boom), το σύστημα αυτόματου εικονικού περιβάλλοντος σπηλαίου (Cave) και τα τρισδιάστατα γυαλιά (LCD shutter glasses).

2.3 Επαυξημένη πραγματικότητα

Ο όρος AR αναφέρεται σε μία νέα τεχνολογία η οποία βασίζεται στην ενσωμάτωση εικονικών στοιχείων, που παράγονται από έναν υπολογιστή, σε ένα πραγματικό περιβάλλον και σε πραγματικό χρόνο, επιτρέποντας τη συνύπαρξη ψηφιακών και πραγματικών αντικειμένων. Διαφοροποιείται από την εικονική πραγματικότητα, όπου ο χρήστης είναι πλήρως εμβυθισμένος σε έναν ψηφιακό κόσμο, διότι συμπληρώνει το πραγματικό περιβάλλον, δεν το αντικαθιστά, και γίνεται αντιληπτό από τον χρήστη όχι μόνο μέσω της όρασης, αλλά και μέσω των υπόλοιπων αισθήσεων. Οι πληροφορίες οι οποίες επαυξάνονται στο πραγματικό περιβάλλον είναι διαδραστικές και τρισδιάστατες καθώς συνδυάζουν πραγματικά και εικονικά αντικείμενα. Σε αντίθεση με την εικονική πραγματικότητα, το πραγματικό περιβάλλον δεν υποκρύπτεται, αλλά έχει κυρίαρχο ρόλο. Με τη χρήση σύγχρονων τεχνολογιών χωροθέτησης, όπως το GPS (Global Positioning System), αισθητήρων κίνησης, αδράνειας και διεύθυνσης, καμερών ενσωματωμένων σε συσκευές και αισθητήρων βάθους, είναι δυνατό να επιτευχθεί ένας ικανοποιητικός βαθμός εμβύθισης σε περιβάλλον AR[2].

2.4 Μικτή πραγματικότητα

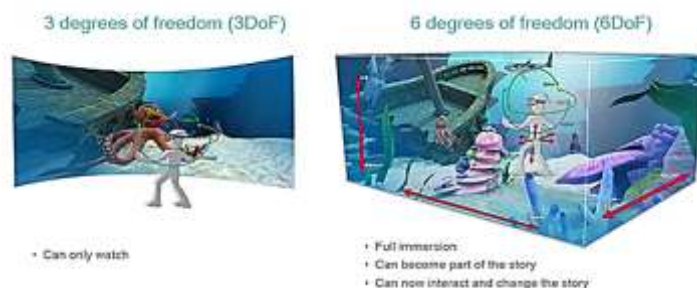
Η MR είναι η επιστημονική περιοχή που συμπεριλαμβάνει όλες τις διαβαθμίσεις πολύπλεξης (multiplex) ψηφιακής πληροφορίας στο συνεχή νοητή ευθεία που συνδέει τον πραγματικό με τον εικονικό κόσμο. Αποτελεί τον συνδυασμό των ερευνητικών περιοχών της AR και της VR, καθώς στην AR τα στοιχεία του εικονικού κόσμου συνδυάζονται διαδραστικά με το πραγματικό περιβάλλον, ενώ στην VR ο χρήστης εμβυθίζεται πλήρως στον εικονικό κόσμο. Ένα σύστημα μικτής πραγματικότητας παράγει μια σύνθετη άποψη για το χρήστη [3]. Είναι ένας συνδυασμός της πραγματικής σκηνής που βλέπει ο χρήστης και μιας εικονικής σκηνής που παράγεται από τον υπολογιστή, ο οποίος εμπλουτίζει την άποψη του χρήστη με τις πρόσθετες πληροφορίες. Οι έννοιες των τριών παραπάνω τεχνολογιών και οι λεπτές διαφορές τους γίνονται παραπάνω κατανοητές βλέποντας το σχήμα 2.1 [4].



Σχήμα 2.1: VR vs AR vs MR

2.5 Εκτεταμένη πραγματικότητα

Το XR είναι ένας όρος ομπρέλα για όλες τις βιοματικές τεχνολογίες, όπως η VR, AR, MR, καθώς και εκείνες που αναμένεται να δημιουργηθούν. Η XR αποτελείται από όλες τις νέες τεχνολογίες που είτε συνδυάζουν εικονικούς και πραγματικούς κόσμους, είτε δημιουργούν μια εντυπωσιακή εμπειρία τριών διαστάσεων και έξι βαθμών ελευθερίας (6DoF). Στις τεχνολογίες που προηγούνται της XR, ο χρήστης μπορούσε να κοιτάξει μόνο στους άξονες X/Y/Z, αποδίδοντας τρεις βαθμούς ελευθερίας (3DoF). Με την εμφάνιση της XR, ακολούθησε το ξεκλείδωμα της κίνησης στους άξονες X/Y/Z. Ο χρήστης πλέον, εκτός από το να κοιτάζει στους άξονες X/Y/Z, έχει συνολικά έξι βαθμούς ελευθερίας (6DoF) και μια πολύ πιο συναρπαστική εμπειρία εμβύθισης. Στο σχήμα 2.2 φαίνονται οι διαφορές μεταξύ τους.



Σχήμα 2.2: 3DoF vs 6DoF

Τα τελευταία πέντε χρόνια περίπου, η XR συνάντησε μεγάλο ενδιαφέρον στις επιστήμες και τη βιομηχανία, καθώς οι πρόσφατες τεχνολογικές εξελίξεις οδήγησαν σε φθηνότερες και ελαφρύτερες συσκευές (π.χ. smartphones), και σημαντικά πιο ισχυρό λογισμικό από τις προηγούμενες γενιές [5]. Προηγουμένως, η χρήση της XR παρέμεινε «στο εργαστήριο» ή μόνο σε εξειδικευμένους τομείς. Η πρόσφατη στροφή οδήγησε σε μια ευρύτερη συμμετοχή στην κοινωνία, όπως στην πολιτική άμυνα, την αεροπορία, την ετοιμότητα έκτακτης ανάγκης και τον σχεδιασμό εκκένωσης, και σχεδόν σε όλους τους εκπαιδευτικούς κλάδους καθώς και στον ιδιωτικό τομέα.

Στην Επιστήμη Γεωγραφικών Πληροφοριών (GIScience) και σε συναφείς τομείς, οι XR τεχνολογίες παρουσιάζουν μοναδικές ευκαιρίες δημιουργίας χωρικών εμπειριών, με τρόπο κατά τον οποίο οι άνθρωποι αλληλεπιδρούν με το περιβάλλον τους και αποκτούν χωρική γνώση. Με τις πρόσφατες εξελίξεις στο λογισμικό, ο χωρικός υπολογισμός έχει αναδειχθεί ως ένα ισχυρό παράδειγμα που επιτρέπει στους πολίτες να διαθέτουν smartphone για να χαρτογραφήσουν το περιβάλλον τους σε πραγματικό χρόνο, σε τρεις διαστάσεις και σε υψηλή πιστότητα.

Η XR, ωστόσο, δεν αφορά μόνο 3D αναπαραστάσεις, αλλά και συναρπαστικές εμπειρίες, βασισμένες στην οπτικοποίηση που μεταμορφώνει την αίσθηση του χώρου. Μέσα από καλά σχεδιασμένα περιβάλλοντα XR, μπορεί να επεκταθεί συλλογικά η ζωντανή εμπειρία ενός χρήστη με γεωγραφικούς χώρους (ως εκ τούτου, ο όρος εκτεταμένη πραγματικότητα). Μέσω τεχνολογιών VR, ο χρήστης μπορεί να βιώσει μια αίσθηση του τόπου συγκρίσιμη ή ακόμη και πανομοιότυπη με τον πραγματικό κόσμο. Μια τέτοια εξέλιξη έχει επηρεάσει την επιστημονική έρευνα με πρωτοφανείς τρόπους.

Εκτός από το VR, εισάγοντας πληροφορίες στο περιβάλλον με τεχνολογίες AR και MR, μπορεί να επανασχεδιαστεί ο φυσικό κόσμος με πολλούς και συναρπαστικούς τρόπους. Τέτοιοι (κείμενο, ακουστικοί, οπτικοί ή ακόμη και οσφρητικοί) τρόποι, έχουν σημαντικές δυνατότητες να επηρεάσουν την ανθρώπινη χωρική σκέψη και μάθηση καθώς και την καθημερινή ζωή των ανθρώπων.

Από τη θετική πλευρά, οι άνθρωποι μπορούν να λαμβάνουν εύκολα και γρήγορα σχετικές πληροφορίες, βοήθεια, συντροφιά και έμπνευση μέσα από εμπειρίες XR. Για παράδειγμα, ένας χρήστης μπορεί να γενικεύσει (με την έννοια της χαρτογραφικής γενίκευσης) τις πληροφορίες στον πραγματικό κόσμο επισημαίνοντας, τονίζοντας ή αποκρύβοντας συγκεκριμένα αντικείμενα ή οποιοδήποτε μέρος του οπτικού πεδίου, έτσι ώστε να επιτευχθούν διάφοροι στόχοι αναλόγως τον σκοπό που προσφέρει η κάθε XR εμπειρία. Από την ανησυχητική πλευρά, υπάρχουν σημαντικές ηθικές εκτιμήσεις όπως π.χ.: Εάν κάποιος άλλος διαχειρίζεται τη γεωγραφική πραγματικότητα ενός χρήστη, υπάρχει απειλή για την ανθρώπινη αυτονομία, πολιτικός έλεγχος, ρύπανση από πληροφορίες που προέρχονται από διαφημίσεις και βαθύτερα ζητήματα που μπορεί να επηρεάσουν το προσωπικό γνωστικό σύστημα του χρήστη σε θεμελιώδες επίπεδο. Για παράδειγμα, μπορεί να χάσει ο χρήστης τη θεμελιώδη γνωστική κατανόηση της μονιμότητας των αντικειμένων, εάν δεν καταφέρνει πλέον να ξεχωρίσει τα εικονικά αντικείμενα από τα πραγματικά. Εν ολίγοις, η αφθονία των τρισδιάστατων χωρικών δεδομένων, ο εκδημοκρατισμός των τεχνολογιών και η πρόοδος, σε γραφικά υπολογιστών που επιτρέπουν ρεαλιστικές προσομοιώσεις φυσικών φαινομένων με τη χρήση XR, απαιτούν επανεξέταση της τεχνολογίας XR και αποτελούν ερευνητικές προκλήσεις.

Συνοψίζοντας, η XR οραματίζεται μια πιο περίπλοκη σχέση, στην οποία η αληθινή πραγματικότητα πραγματοποιεί πραγματικά εφέ στο βαθμό που η αντίληψή μας για εικονικά αντικείμενα γίνεται πλήρως πραγματική. Τέλος, καλύπτει ένα ευρύ φάσμα ευκαιριών, εμπειριών, έρευνας, προκλήσεων και κινδύνων.

2.6 Σύγχρονες τεχνολογίες στην εκπαίδευση

Τα τελευταία χρόνια οι παραπάνω τεχνολογίες, έχουν ωριμάσει στο βαθμό που πλέον εφαρμόζονται σε ένα πολύ ευρύτερο φάσμα πεδίων, όπως στον εκπαιδευτικό τομέα, όπου η εφαρμογή τους είναι αξιόλογη και αποτελεσματική στην προώθηση των παιδαγωγικών σκοπών.

2.6.1 Σύγχρονες εκπαιδευτικές θεωρίες

Σημαντική προσπάθεια αναθεώρησης των παραδοσιακών μεθοδολογικών προσεγγίσεων παρατηρείται τις τελευταίες δεκαετίες, όσον αφορά στην υποστήριξη της εκπαιδευτικής διαδικασίας, καθώς και στροφή προς τις νέες θεωρίες μάθησης που προσφέρουν καινούριες δυνατότητες προσέγγισης της εκπαιδευτικής διαδικασίας[6]. Οι σύγχρονες εκπαιδευτικές θεωρίες, έρχονται να αλλάξουν το μέχρι τώρα κλασικό εκπαιδευτικό μοντέλο, παροτρύνοντας την ενεργό συμμετοχή του μαθητή στη διαδικασία μάθησης και διδακτικής πράξης, έναντι των παραδοσιακών συστημάτων εκπαίδευσης όπου ο ρόλος του ήταν παθητικός. Επίσης υποστηρίζουν νέους τρόπους μάθησης και απόκτησης της γνώσης, διατήρησης και γενίκευσής της, μέσα από μια ενεργητική διαδικασία. Οι σύγχρονες τεχνολογίες των VR, AR, MR και XR, προσφέρουν πολλές καινούριες δυνατότητες για την υποστήριξη της εκπαιδευτικής διαδικασίας. Πιο συγκεκριμένα, η αλληλεπίδραση σε πρώτο πρόσωπο με το γνωστικό αντικείμενο, η εμπειρική μάθηση μέσα από την πράξη, η ουσιαστικότερη προσέγγιση αφηρημένων εννοιών με τρόπο πολυαισθητηριακό και η δημιουργία κινήτρων, αποτελούν τα δομικά στοιχεία των σύγχρονων εκπαιδευτικών θεωριών που ενισχύουν το μοντέλο της εκπαιδευτικής διαδικασίας.

Τα περιβάλλοντα VR, AR, MR και XR, είναι περιβάλλοντα διεπαφής στα οποία ο χρήστης βιώνει τη διαδικασία της αλληλεπίδρασης περισσότερο εμπειρικά παρά νοητικά. Η χρήση τέτοιων τεχνολογιών παρέχει τη δυνατότητα εμπειρικής μάθησης και γι' αυτό μπορεί να χρησιμοποιηθεί με επιτυχία στην εκπαίδευση. Το τριδιάστατο, πολυαισθητηριακό περιβάλλον και η αντιστοίχιση ενεργειών και συμπεριφοράς του χρήστη με το απεικονιζόμενο αποτέλεσμα δημιουργούν την αίσθηση παρουσίας μέσα στο περιβάλλον αυτό. Η αίσθηση της παρουσίας θεωρείται ότι υποβοηθά την επικοινωνία της μαθησιακής εμπειρίας. Τα μαθησιακά εικονικά περιβάλλοντα υποστηρίζουν την αισθητηριακά πλούσια αλληλεπίδραση μαθητή-υπολογιστή, επιτρέποντας τον χειρισμό και την εξέταση των συνθετικών αντικειμένων, σαν να ήταν αυτά αντικείμενα του πραγματικού κόσμου.

Τέτοια περιβάλλοντα παρέχουν επίσης τη δυνατότητα στους χρήστες να δημιουργούν, να χειρίζονται και να επεξεργάζονται κάθε τύπο ψηφιακής πληροφορίας. Τα παιδιά μπορούν κατ' αυτόν τον τρόπο να δομούν ενεργά τις δικές τους σκέψεις για τον κόσμο, και συνεπώς είναι πιθανό να ενθαρρύνονται ώστε να δομούν τη δική τους γνώση, γεγονός το οποίο αποτελεί αποδεδειγμένα αποτελεσματικό τρόπο εκμάθησης. Το γεγονός πως η αλληλεπίδραση με ένα μοντέλο μπορεί να είναι ένα πολύ καλύτερο κίνητρο από εκείνο που προκαλεί η αλληλεπίδραση με ένα φυσικό υλικό αντικείμενο, έχει ως αποτέλεσμα τη δημιουργία κινήτρων για τους μαθητές.

Η χρήση αυτών των τεχνολογιών στην εκπαίδευση ενδείκνυται ιδιαίτερα στην περίπτωση της προσομοίωσης μιας δραστηριότητας, όπως π.χ. τα πειράματα στο μάθημα της φυσικής, καθώς η εμπειρία της δημιουργίας ενός προσομοιωμένου περιβάλλοντος ή μοντέλου θεωρείται πολύ σημαντική για τη γνώση του αντικειμένου της δραστηριότητας που εκτελείται στο περιβάλλον αυτό. Ο μαθητής ενθαρρύνεται να μάθει μέσα από την ενεργή συμμετοχή του. Επίσης, μέσα από την αλληλεπίδραση με το εικονικό περιβάλλον, παρέχεται καλύτερη δυνατότητα εξέτασης ενός αντικειμένου κατά τρόπο πολυαισθητηριακό, ο οποίος συμπεριλαμβάνει δεξιότητες χειρισμού και φυσικής κίνησης. Με αυτό τον τρόπο δίνεται η δυνατότητα στο μαθητή να συμμετέχει σε μια

εκπαιδευτική διαδικασία πιο ενδιαφέρουσα και διασκεδαστική, όπου κάποια χαρακτηριστικά και πτυχές αυτής μπορούν να παρουσιαστούν με πολύ πιο αντιληπτικά ολοκληρωμένο τρόπο σε σχέση με άλλα μέσα και όπου επιτρέπεται στον μαθητευόμενο να προχωρά διαμέσου ενός πειράματος με τον δικό του ρυθμό, με ανοιχτά χρονικά περιθώρια και όχι μέσα σε έναν προκαθορισμένο χρόνο.

2.6.2 Κινητή μάθηση (Mobile Learning)

Οι εφαρμογές για κινητά τηλέφωνα και tablets αποκτούν ολοένα και πιο κυρίαρχη θέση στον τομέα της εκπαιδευτικής τεχνολογίας και μάθησης [7]. Η κινητή μάθηση αποτελεί μια προσέγγιση της ηλεκτρονικής μάθησης, η οποία αναφέρεται στον τρόπο με τον οποίο οι φορητές συσκευές αξιοποιούνται στην εκπαίδευση προσφέροντας νέες μαθησιακές εμπειρίες. Πιο συγκεκριμένα, η κινητή μάθηση μπορεί να οριστεί ως η χρήση κινητών συσκευών με σκοπό την πρόσβαση σε μαθησιακό υλικό και την ανάπτυξη συνεργασίας και επικοινωνίας, ενώ σε αυτή συμπεριλαμβάνονται οι σχέσεις αλληλεπίδρασης μεταξύ των μαθητών, των εκπαιδευτικών και των μαθησιακών αντικειμένων.

Τα τελευταία χρόνια, οι εξελίξεις στις παιδαγωγικές προσεγγίσεις καθιστούν απαραίτητη την ύπαρξη νέων μέσων και εργαλείων που συμβάλλουν στην καλύτερη γνώση εννοιών και φαινομένων διαφόρων γνωστικών αντικειμένων και ειδικότερα στο χώρο των φυσικών επιστημών. Ένα τέτοιο εργαλείο αποτελούν οι φορητές συσκευές, οι οποίες έχουν ενθαρρυντικά αποτελέσματα στην εκπαίδευση, καθώς δημιουργούν ένα ελκυστικό περιβάλλον μάθησης και μια εποικοδομητική και αποτελεσματική διαδικασία μάθησης. Η καινούρια αυτή μορφή μάθησης, έρχεται να λειτουργήσει συμπληρωματικά πλαισιώνοντας τη διδασκαλία με επιπλέον δραστηριότητες, και όχι για να αντικαταστήσει τον παραδοσιακό τρόπο διδασκαλίας.

Το πιο σημαντικό χαρακτηριστικό της κινητής μάθησης είναι η ίδια του η φορητότητα η οποία επιτρέπει την επικοινωνία σε ένα πλαίσιο διαφορετικό από το συμβατικό περιβάλλον και δημιουργεί ενδιαφέρουσες ευκαιρίες για νέες μορφές μάθησης, καθώς αλλάζει τη φύση της σχέσης του εκπαιδευτικού, του μαθητή και του αντικειμένου μάθησης. Επιπλέον, με τη χρήση φορητής τεχνολογίας ενισχύεται η ομαδοσυνεργατική διδασκαλία και επιτρέπεται η αλληλεπίδραση μεταξύ των μαθητών. Οι συνεργατικές αυτές δραστηριότητες μπορούν να λάβουν χώρα σε μία σχολική τάξη. Οι μαθητές φαίνεται με αυτόν τον τρόπο να συμμετέχουν σε μεγαλύτερο βαθμό στη μαθησιακή διαδικασία. Παράλληλα, μεγιστοποιούνται τα κίνητρά τους για μάθηση, αναπτύσσεται η κριτική τους σκέψη και η δημιουργικότητα, ενώ σημειώνουν καλύτερες επιδόσεις.

Η βασική διαφορά της κινητής τεχνολογίας με τις υπόλοιπες ψηφιακές τεχνολογίες, είναι η δυνατότητα ψηφιακής αναπαράστασης των φυσικών αντικειμένων που βρίσκονται στην ίδια τοποθεσία με τον εκπαιδευόμενο στον ίδιο χρόνο. Η κινητή τεχνολογία έχει τη δυνατότητα να επαυξήσει τα αντικείμενα που βρίσκονται γύρω από το χώρο του χρήστη με οπτικές ή ακουστικές πληροφορίες οι οποίες ενεργοποιούνται αυτόματα ή κατόπιν αιτήματος από τον χρήστη.

Για την εφαρμογή της κινητής μάθησης, είναι ιδιαίτερα σημαντικό να λαμβάνεται υπόψη το μοντέλο ταξινόμησης που πρότεινε ο Bloom. Ειδικότερα, το μοντέλο του Bloom αποτελείται από πέντε επίπεδα μάθησης, κατά το οποίο ο εκπαιδευόμενος ξεκινάει από τη βάση αρχικά κατακτώντας μία επιφανειακή μάθηση και σταδιακά προχωράει σε ανώτερα επίπεδα μάθησης, στα οποία ξεκινάει σιγά σιγά η κατανόηση της γνώσης που απέκτησε. Στη συνέχεια, καλείται να κάνει εφαρμογή της γνώσης και να αναλύσει και να συνθέσει επιμέρους θέματα που σχετίζονται με την αποκτηθείσα γνώση. Έτσι, ο εκπαιδευόμενος παρακινείται ενεργά, προσελκύεται το ενδιαφέρον του και εμπλέκεται συμμετοχικά και δημιουργικά σε μια εποικοδομητική διαδικασία που θα τον οδηγήσει σε βαθιά γνώση του

αντικειμένου μέσα από μια δυναμική μάθηση. Η τελευταία βαθμίδα της αξιολόγησης, αναφέρεται κυρίως στον διαδραστικό και ενεργό ρόλο του μαθητή στην εκπαιδευτική διαδικασία με σκοπό την απόκτηση γνώσης.



Σχήμα 2.3: Μοντέλο ταξινόμησης διδακτικών στόχων (Bloom, 1956)

Σήμερα, η πιο διαδεδομένη και συνηθισμένη μορφή τεχνολογίας στην καθημερινή ζωή των ανθρώπων, και ειδικότερα στη ζωή των παιδιών, είναι οι φορητές συσκευές, είτε πρόκειται για κινητά τηλέφωνα, είτε για tablets, iPod ή φορητές πλατφόρμες παιχνιδιών. Οι φορητές συσκευές, υπερτερούν στην υποστήριξη της τεχνολογίας των VR, AR, MR, XR, καθώς όπως προαναφέρθηκε είναι τεχνολογίες οι οποίες επιτρέπουν τη μετακίνηση του χρήστη στον πραγματικό κόσμο και τη διάδρασή του με εικονικά αντικείμενα. Η φορητή επαυξημένη πραγματικότητα η οποία αναφέρεται στη χρήση φορητών συσκευών ως μέσων για την παρουσίαση εικονικών αντικειμένων σε πραγματικό χωροχρόνο παρουσιάζει ορισμένα χαρακτηριστικά τα οποία θα μπορούσαν να συντελέσουν στην εποικοδομητική χρήση τους από τους χρήστες και να ενισχύσει την αποτελεσματικότητα της διδασκαλίας. Ειδικότερα, ο χρήστης μπορεί να μεταβάλει διάφορα χαρακτηριστικά των εικονικών αντικειμένων που προβάλλονται στην οθόνη της φορητής του συσκευής, όπως το σχήμα, το χρώμα και τη θέση με σκοπό να κατανοήσει σε μεγαλύτερο βαθμό έννοιες και φαινόμενα τα οποία είναι ανέφικτο να παρατηρήσει στην πραγματική του ζωή. Στη σύγχρονη εποχή, οι φορητές συσκευές είναι ευρέως διαθέσιμες και μπορούν να υποστηρίξουν μεγάλο αριθμό εφαρμογών επαυξημένης πραγματικότητας, δίνοντας τη δυνατότητα ανάπτυξης συνεργατικών συστημάτων. Επίσης, λόγω του μικρού μεγέθους και το βάρους που έχουν, επιτρέπεται η γρήγορη και αποτελεσματική χρήση τους από τον χρήστη.

2.6.3 Βιωματική μάθηση

Η βιωματική μάθηση, είναι μια θεωρία η οποία αναφέρεται στο ρόλο που διαδραματίζει η εμπειρία στη διαδικασία μάθησης, καθώς και στους δεσμούς μεταξύ της σχολικής τάξης, της καθημερινής ζωής των μαθητών και της κοινωνικής πραγματικότητας. Το εκπαιδευτικό σύστημα της σημερινής εποχής, χαρακτηρίζεται από μειωμένη παροχή ευκαιριών στους μαθητές για καλλιέργεια της κριτικής σκέψης, δημιουργικότητας και επεξεργασίας εμπειριών ενώ συνδέονται με τα γνωστικά αντικείμενα κατά κύριο λόγο μέσω της απομνημόνευσης και της επανάληψης. Αυτό το γεγονός, δημιουργεί επιτακτική την ανάγκη να προωθηθούν βιωματικές μαθησιακές διαδικασίες οι οποίες θα είναι ικανές να ενισχύσουν την ενεργό συμμετοχή των μαθητών στη μαθησιακή διαδικασία, καθώς επίσης την αυτονομία και την ανεξαρτησία τους. Ένας ακόμα στόχος της προώθησης της βιωματικής μάθησης

ως μέσο διδασκαλίας είναι η ανάπτυξη της κριτικής σκέψης και της δημιουργικότητάς των μαθητών, κινητοποιώντας τους διανοητικά και συναισθηματικά.

Η βιωματική μάθηση αποτελεί την μάθηση μέσα από την πράξη και τη γνώση που συντελεί στη σύνδεση θεωρίας και αληθινής εμπειρίας. Ειδικότερα, στο χώρο της εκπαίδευσης η βιωματική μάθηση αναφέρεται στην οργάνωση της μαθησιακής διαδικασίας με βάση το «learning by doing» το οποίο έχει ως αποτέλεσμα την ενεργητική συμμετοχή και εμπλοκή των μαθητών στη μαθησιακή διαδικασία και σε δραστηριότητες όπως έρευνες, πειράματα, προσομοιώσεις και παρατηρήσεις. Στο πλαίσιο αυτής της θεώρησης, η βιωματική μάθηση συμπεριλαμβάνει πολλά στοιχεία διερευνητικής μάθησης. Το συγκεκριμένο μοντέλο μάθησης ξεκινάει από μία εμπειρία, ακολουθείται από μια αντίδραση, έπειτα ακολουθεί συζήτηση και τέλος η ανάλυση.

Ως βασικές εκπαιδευτικές μέθοδοι, θεωρούνται εκείνες κατά τις οποίες οι συμμετέχοντες συμμετέχουν ενεργά στη μαθησιακή διαδικασία και οδηγούνται στην πράξη εφόσον κατανοήσουν και συνειδητοποιήσουν όσα έπραξαν. Ως εκ τούτου, τόσο η ενεργητική εμπλοκή των μαθητών όσο και ο στοχασμός της εμπειρίας που βίωσαν αποτελούν αναπόσπαστο στοιχείο της βιωματικής εκπαίδευσης. Οι εκπαιδευτικές τεχνικές και μέθοδοι οι οποίες προωθούν τη βιωματική μάθηση είναι τρεις: οι εκπαιδευτικές τεχνικές που διεξάγονται μέσα στην τάξη, αυτές που διεξάγονται έξω από την τάξη και οι εκπαιδευτικές ασκήσεις. Η διπλωματική εργασία ασχολείται μόνο με την πρώτη κατηγορία όπου οι εκπαιδευτικές τεχνικές που διεξάγονται μέσα στην τάξη.

Οι εκπαιδευτικές τεχνικές της πρώτης κατηγορίας έχουν στο επίκεντρο του ενδιαφέροντός τους τον μαθητή ο οποίος αξιοποιεί τις προϋπάρχουσες γνώσεις και εμπειρίες του, προκειμένου να τις μοιραστεί με την ομάδα και να συνεργαστεί με στόχο να επιλύσει προβληματικές καταστάσεις. Με αυτό τον τρόπο, ενισχύεται η ενεργητική συμμετοχή των μαθητών στη μαθησιακή διαδικασία καθώς παράγεται άμεσα πλήθος εμπειριών. Τέτοιου είδους εκπαιδευτικές τεχνικές και μέθοδοι αποτελούν τα παιχνίδια επαυξημένης πραγματικότητας, ο καταγιγισμός ιδεών που προσφέρει το παιχνίδι, η ομαδική συζήτηση των συμπερασμάτων και οι μελέτες περίπτωσης. Σημαντικός παράγοντας για την πετυχημένη χρήση αυτών των τεχνικών είναι ο ίδιος ο εκπαιδευτικός που τις χρησιμοποιεί, καθώς πρέπει να γνωρίζει έστω σε θεωρητικό επίπεδο τη διαδικασία εφαρμογής των τεχνικών αυτών, αλλά και να τις προσαρμόζει στις ιδιαιτερότητες και τις ανάγκες των μαθητών του διαμορφώνοντας το κατάλληλο κλίμα στην τάξη.

2.7 Σύγχρονες τεχνολογίες στον χώρο των φυσικών επιστημών

Η πρόοδος του εκπαιδευτικού συστήματος συνδέεται στενά με την ένταξη της τεχνολογίας σ' αυτό. Τα τελευταία χρόνια παρατηρείται μία σταδιακή ενσωμάτωση των σύγχρονων τεχνολογιών στη διδασκαλία των Φυσικών Επιστημών σε όλες τις εκπαιδευτικές βαθμίδες. Η υποστήριξη της μάθησης μέσω των ψηφιακών τεχνολογιών και μάλιστα προηγμένων τεχνολογιών όπως αυτές της εικονικής και επαυξημένης πραγματικότητας αναδεικνύει το βιωματικό τρόπο μάθησης ως αποτελεσματικότερο συγκριτικά με αφηρημένες έννοιες και παραδοσιακές προσεγγίσεις. Αυτός ο νέος βιωματικός τρόπος προσέγγισης της μάθησης προκύπτει σε μεγάλο βαθμό από τη διαδραστικότητα που προσφέρουν οι εφαρμογές VR, AR, MR και XR.

2.7.1 Η εφαρμογή “XRphysics” για την αναβάθμιση της διδασκαλίας του μαθήματος της φυσικής Ε' δημοτικού

Είναι γεγονός ότι η Επαυξημένη Πραγματικότητα έχει καθιερωθεί ακόμα και σε χώρους που παραδοσιακά δεν χρησιμοποιούν ιδιαίτερα τεχνολογικά μέσα, όπως ο χώρος των εκδόσεων. Με την

προσθήκη qr-codes σε σελίδες του σχολικού βιβλίου της φυσικής Ε' δημοτικού, τα παραδοσιακά σχολικά βιβλία αλλάζουν μορφή και συνδυάζουν το παιχνίδι, με το διάβασμα και την τεχνολογία. Πρόκειται για ένα βιβλίο που ζωντανεύει το περιεχόμενό του τρισδιάστατα, με ήχο και κίνηση μέσα από τις οθόνες των φορητών συσκευών. Η εφαρμογή «XRphysics», συνδυάζει της μεθόδους της κινητής και βιωματικής μάθησης και χρησιμοποιεί σύγχρονες τεχνολογίες όπως αυτή της εικονικής και επαυξημένης πραγματικότητας, παρακινώντας τους μαθητές της Ε' δημοτικού να υιοθετήσουν ενεργό και συμμετοχικό ρόλο κατά τη μαθησιακή διαδικασία. Η εφαρμογή αποτελεί ένα εργαλείο για εκπαιδευτικές τεχνικές που διεξάγονται μέσα στην τάξη. Ο εκπαιδευτικός έχει τη δυνατότητα να εντάξει στη διδασκαλία του μαθήματος «Φυσικά Ε' δημοτικού» μια καινοτόμο μέθοδο διδασκαλίας η οποία θα τονώσει το ενδιαφέρον των μαθητών και θα τους υποβοηθήσει να κατανοήσουν πολύπλοκα φαινόμενα και έννοιες της φυσικής Ε' δημοτικού, που αποτελούν μια γνωστική περιοχή με πολλές ιδιαιτερότητες. Η χρήση της εφαρμογής μέσα στην τάξη γίνεται μέσω φορητών συσκευών που μπορούν να χρησιμοποιηθούν ατομικά από τους μαθητές αλλά και σε μικρές ομάδες ενισχύοντας έτσι τη συνεργασία μεταξύ των μαθητών και την ανταλλαγή απόψεων και παρέχοντας έτσι κίνητρα για μάθηση. Η χρήση φορητών συσκευών (smartphone ή tablet) σε συνδυασμό με την εφαρμογή επαυξημένης πραγματικότητας XRphysics, στοχεύει σε καλύτερα μαθησιακά αποτελέσματα σε σύγκριση με τη συμβατική διδασκαλία. Αυτό προκύπτει από το γεγονός ότι η παραστατικότητα της εφαρμογής, η οπτικοποίηση της γνώσης και η διάδραση, έχει θετικά μαθησιακά αποτελέσματα, αφού δημιουργεί ένα ελκυστικό περιβάλλον μάθησης λόγω του συνδυασμού εικόνας και ήχου. Έτσι οι μαθητές αποκτούν μεγαλύτερη αυτονομία και ελέγχουν καλύτερα τη μαθησιακή διαδικασία ακολουθώντας τους ρυθμούς τους.

2.8 Επίλογος

Τα μαθησιακά αποτελέσματα των προαναφερθεισών τεχνολογιών στην εκπαίδευση, μπορούν να ενισχυθούν εφόσον οι μαθητές κατά τη διάρκεια της διδασκαλίας βρίσκονται σε μία θετική ψυχολογική κατάσταση συμμετέχοντας σε μία ευχάριστη και ενδιαφέρουσα δραστηριότητα στην οποία εμφανίζονται απόλυτα απορροφημένοι. Απαραίτητη προϋπόθεση για την αποτελεσματική χρήση εκπαιδευτικών εφαρμογών που χρησιμοποιούν σύγχρονες τεχνολογίες, είναι η κατάλληλη διδακτική τους αξιοποίηση.

Κεφάλαιο 3ο: Εγκατάσταση Unity και βασικές ρυθμίσεις πλατφόρμας

3.1 Εισαγωγή

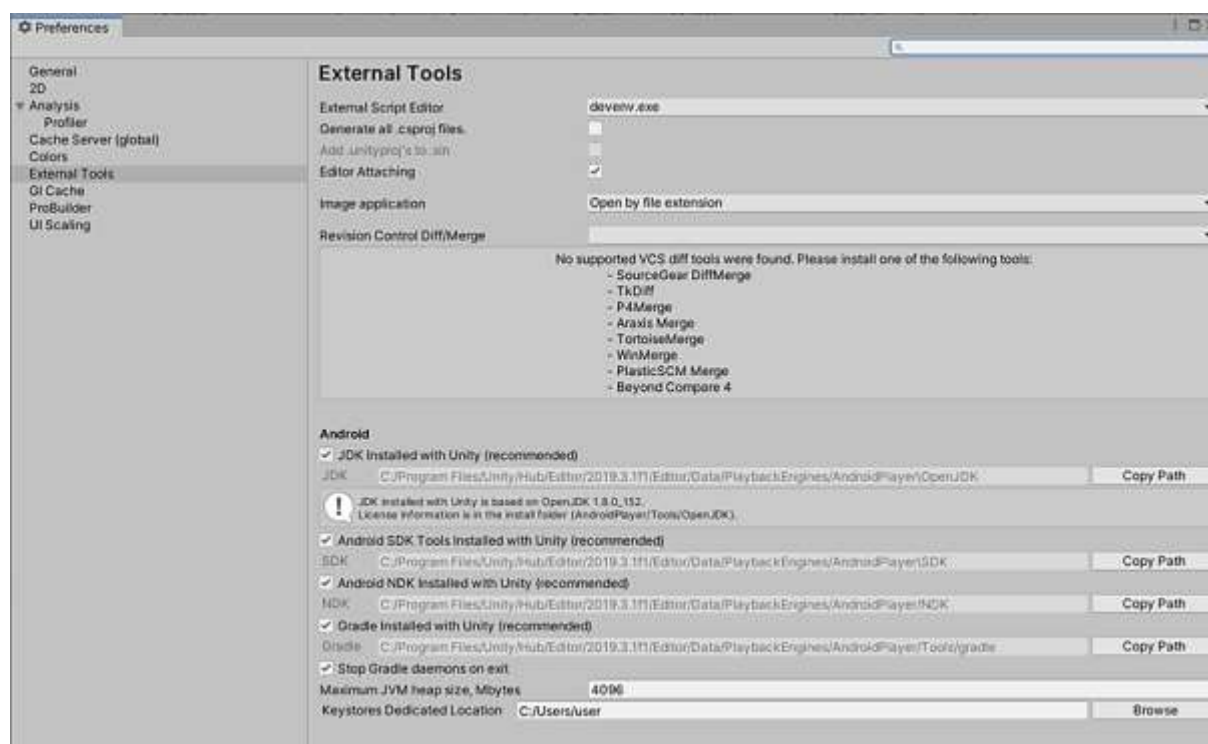
Σε αυτό το κεφάλαιο περιγράφονται όλα τα βήματα που προηγήθηκαν της δημιουργίας του παιχνιδιού, όπως η εγκατάσταση του Unity και Unity Hub, καθώς και οι βασικές ρυθμίσεις και η αρχική παραμετροποίηση των κύριων παραγόντων του προγράμματος.

3.2 Unity και Unity Hub

Η unity είναι η πιο δημοφιλής πλατφόρμα δημιουργίας παιχνιδιών στον κόσμο. Η επιλογή της πλατφόρμας για τη δημιουργία του παιχνιδιού έγινε μετά από αρκετή αναζήτηση ανάμεσα από πολλές άλλες και βασίστηκε στις ασυναγώνιστες δυνατότητες που παρέχει στη δωρεάν έκδοσή της, καθιστώντας την αρκετά ευέλικτη για κατασκευή σχεδόν οποιουδήποτε παιχνιδιού μπορεί να φανταστεί κάποιος. Ένας ακόμα λόγος που συνέβαλε στην επιλογή της είναι πως συνοδεύεται από ισχυρά εργαλεία κινούμενων σχεδίων που καθιστούν σχετικά απλή τη δημιουργία 3D σκηνών και την κατασκευή 3D animations από το μηδέν. Σχεδόν οτιδήποτε μπορεί να είναι κινούμενο στη Unity. Σε προγραμματιστικό επίπεδο, η unity υποστηρίζει C# scripting και η πλατφόρμα που χρησιμοποιήθηκε για την σύνταξη των scripts είναι η Visual studio code. Αρχικά έγινε εγκατάσταση του Unity Hub (έκδοση 2.3.0) σε υπολογιστή, το οποίο παρέχει τη δυνατότητα εγκατάστασης διάφορων εκδόσεων της unity και κατά συνέπεια τον πειραματισμό με τις δυνατότητες της κάθε μιας. Η έκδοση της unity που επιλέχθηκε είναι 2019.3.1f1, καθώς είναι η μόνη συμβατή με όλες τις τεχνολογίες που ήταν επιθυμητές να χρησιμοποιηθούν από το package manager της unity. Τέλος, αφού έγινε δημιουργία προσωπικού Unity ID, ακολούθησε έρευνα και ανάγνωση εγχειριδίων χρήσης της unity [8],[9],[10],[11].

3.3 Ρυθμίσεις SDK – JDK

Η κοινότητα της unity προτείνει τη χρήση του unity hub κυρίως για την απλούστερη παραμετροποίηση αυτών των αρχικών και κομβικών ρυθμίσεων. Για να μπορέσει ο χρήστης να δημιουργήσει και να τρέξει ένα unity project για Android, πρέπει αρχικά να εγκαταστήσει το Unity Android Build Support. Πρέπει επίσης να εγκαταστήσει το Android Software Development Kit (SDK) και το Native Development Kit (NDK) προκειμένου να μπορεί να δημιουργήσει και να εκτελέσει οποιονδήποτε κώδικα σε Android συσκευή. Από προεπιλογή, η Unity έχει εγκαταστήσει ένα Java Development Kit που βασίζεται στο OpenJDK. Σε αυτό το σημείο υπήρξαν σημαντικά προβλήματα με τις συμβατότητες των εργοστασιακά εγκατεστημένων εκδόσεων JDK, SDK και NDK. Ωστόσο, μετά από αναζήτηση, έγινε αλλαγή των εργοστασιακών ρυθμίσεων αυτών των παραμέτρων, αντικαθιστώντας τις με αυτές που φαίνονται στο σχήμα 1.1. Είναι σημαντικό να σημειωθεί πως επιλέχθηκε να δημιουργηθεί η εφαρμογή για συσκευές android, και όχι ios, διότι η συσκευή μου υποστηρίζει λειτουργικό σύστημα android και μέσω αυτής έγιναν όλες οι δοκιμές του unity project. Οι ρυθμίσεις αυτές έγιναν από unity →Edit→Preferences→External tools→Android.



Σχήμα 3.1: Ρυθμίσεις JDK, SDK και NDK

3.4 Package manager

Το Unity Package Manager, βρίσκεται στο unity→Window→Package Manager, και δίνει τη δυνατότητα στο χρήστη να δει ποια πακέτα είναι διαθέσιμα για εγκατάσταση ή έχουν ήδη εγκατασταθεί στο project του. Επιπλέον, ο χρήστης μπορεί να χρησιμοποιήσει αυτό το παράθυρο για να δει ποιες εκδόσεις είναι διαθέσιμες για εγκατάσταση, κατάργηση, απενεργοποίηση και ενημέρωση των ήδη υπαρχόντων πακέτων για κάθε project. Η επιλογή των πακέτων είναι καθοριστική για το είδος του παιχνιδιού που θα δημιουργηθεί στη συνέχεια, αφού αποτελούν τις τεχνολογίες που θα χρησιμοποιηθούν. Πέρα από τα εργοστασιακά εγκατεστημένα πακέτα του project, έγινε εγκατάσταση και των: AR Foundation 2.1.4, AR Subsystems 2.1.1, Vuforia Engine AR 8.5.9, XR Interaction Subsystems 1.0.1, ProBuilder 4.2.3 και TextMesh Pro 2.0.1.

3.4.1 AR Foundation

Το πακέτο AR Foundation (σχήμα 1.2), είναι ειδικά σχεδιασμένο για ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας και επιτρέπει αρχικά τη δημιουργία πλούσιων εμπειριών και στη συνέχεια, την ανάπτυξή τους σε διάφορα κινητά και φορητές συσκευές επαυξημένης πραγματικότητας. Το AR Foundation περιλαμβάνει βασικές δυνατότητες από ARKit, ARCore, Magic Leap και HoloLens, καθώς παρέχει μοναδικές δυνατότητες στη Unity για δημιουργία ισχυρών εφαρμογών επαυξημένης πραγματικότητας. Επιπλέον, αυτό το πακέτο επιτρέπει τη χρήση όλων αυτών των δυνατοτήτων σε μια ενοποιημένη ροή εργασίας.

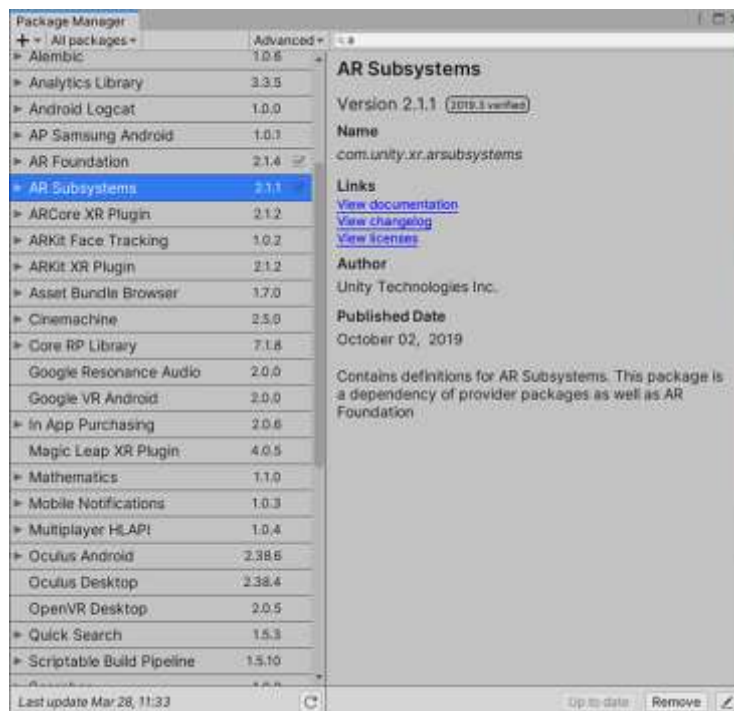
3.4.2 AR Subsystems

Το πακέτο αυτό μπορεί να θεωρηθεί αναπόσπαστο κομμάτι του AR Foundation, καθώς κανένα από τα δυο μόνο του, δεν είναι αρκετό για την ολοκλήρωση μιας εφαρμογής επαυξημένης πραγματικότητας. Πιο συγκεκριμένα, το AR Subsystems (σχήμα 1.3), είναι μια πλατφόρμα για την

εμφάνιση διαφορετικών τύπων λειτουργικότητας και δεδομένων. Τα υποσυστήματα που σχετίζονται με την επαυξημένη πραγματικότητα ορίζονται σε αυτό το πακέτο και χρησιμοποιούν το namespace UnityEngine.XR.ARSubsystems. Αυτό το πακέτο παρέχει μόνο τη διεπαφή για διάφορα υποσυστήματα. Οι υλοποιήσεις για αυτά τα υποσυστήματα ονομάζονται "πάροχοι" και μπορούν συνήθως να βρεθούν σε άλλο πακέτο. Επιπλέον, παρέχει διεπαφές για τα ακόλουθα υποσυστήματα: session, raycasting, camera, plane detection, depth, image tracking, face tracking και environment probes.



Σχήμα 3.2: AR Foundation



Σχήμα 3.3: AR Subsystems

3.4.3 Vuforia Engine AR

Το πακέτο Vuforia Engine AR (σχήμα 1.4) είναι η πιο διαδεδομένη πλατφόρμα για ανάπτυξη εφαρμογών επαυξημένης πραγματικότητας. Μέσω αυτού, οι προγραμματιστές μπορούν εύκολα να προσθέσουν προηγμένη λειτουργικότητα υπολογιστή σε εφαρμογές Android, iOS και UWP, για να δημιουργήσουν εμπειρίες AR που αλληλεπιδρούν ρεαλιστικά με αντικείμενα και το περιβάλλον. Οι δυνατότητες αυτού του πακέτου είναι εκπληκτικές και τα περισσότερα αντικείμενα του παιχνιδιού δημιουργήθηκαν από αυτό. Πιο συγκεκριμένα, δύο από τις πιο χρήσιμες δυνατότητες που παρέχει είναι η δημιουργία αντικειμένου ως κάμερα επαυξημένης πραγματικότητας και η δημιουργία αντικειμένου ως model target για την αναγνώριση των qr-codes και εμφάνιση περιεχομένου επαυξημένης πραγματικότητας. Αυτά τα δυο αντικείμενα αποτελούν σημαντικό κορμό της εφαρμογής και χωρίς το vuforia engine AR, θα ήταν αδύνατο να δημιουργηθεί μια ολοκληρωμένη εμπειρία εμπύθισης επαυξημένης πραγματικότητας. Επιπλέον, σημαντικό χαρακτηριστικό αυτού του πακέτου είναι πως δουλεύει σε συνδυασμό με την ιστοσελίδα του, η οποία δίνει τη δυνατότητα δημιουργίας βάσης από markers-qr-codes, καθώς επίσης και τη δυνατότητα δημιουργίας κλειδιού αδείας.



Σχήμα 3.4: Vuforia Engine AR

3.4.4 XR Interaction subsystems

Ο σκοπός αυτού του πακέτου είναι να παρέχει ορισμούς όλων των υποσυστημάτων που επιτρέπουν τη λειτουργικότητα αλληλεπίδρασης εκτεταμένης πραγματικότητας. Παρέχει κυρίως υποσυστήματα στα οποία ο χρήστης υποδεικνύει ότι θέλει να αλληλεπιδράσει με κάποιο αντικείμενο στον κόσμο. Αυτό μπορεί να προκληθεί πατώντας στον αέρα, τοποθετώντας το δάχτυλό του χρήστη επάνω ή αγγίζοντας μια οθόνη αφής. Βασικό μέλημα του πακέτου είναι να καθορίσει ποια χειρονομία πηγής θα πρέπει να προκαλεί αυτό (θα πρέπει να είναι η κατάλληλη για μια δεδομένη πλατφόρμα και συνεπής με τον τρόπο με τον οποίο ένας χρήστης αλληλεπιδρά διαφορετικά με UI και παγκόσμια αντικείμενα). Συνήθως αυτό θα ενεργοποιήσει ό,τι αντικείμενο είναι κεντραρισμένο στην προβολή των χρηστών. Η τεχνολογία που παρέχει το πακέτο XR Interaction subsystems (σχήμα 1.5) μπορεί να

χαρακτηριστεί ως επίκαιρη διότι κυκλοφόρησε στα τέλη του 2019 και βρίσκεται σε πρώιμο στάδιο εφαρμογής. Ακόμα και η εύρεση υλικού για βοήθεια ήταν αρκετά δύσκολη καθώς ο αριθμός πληροφοριών στο διαδίκτυο ήταν αρκετά περιορισμένος.



Σχήμα 3.5: XR Interaction Subsystems

3.4.5 ProBuilder

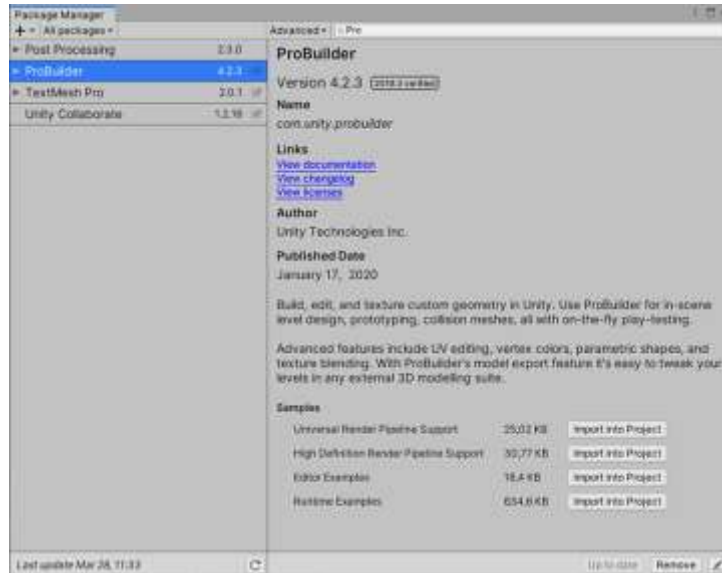
Το πακέτο του ProBuilder (σχήμα 1.6), είναι ένα μοναδικό υβριδικό εργαλείο τρισδιάστατης μοντελοποίησης και σχεδιασμού επιπέδων, βελτιστοποιημένο για την κατασκευή απλής γεωμετρίας, αλλά και ικανό για λεπτομερή επεξεργασία. Η χρήση του διευκολύνει κυρίως την κατασκευή πρωτοτύπων καθιστώντας τη γρήγορη και εύκολη σε αντίθεση με τον απλό τρόπο κατασκευής της unity. Μια ακόμα σημαντική δυνατότητα που παρέχει αυτό το πακέτο είναι τα σύνθετα χαρακτηριστικά εδάφους και η προσαρμοσμένη γεωμετρία. Το ProBuilder εκμεταλλεύεται επίσης τις απρόσκοπτες δυνατότητες της Unity με εργαλεία δημιουργίας ψηφιακού περιεχομένου (όπως η Maya), ώστε ο χρήστης να μπορεί να βάλει περισσότερες λεπτομέρειες και να γυαλίσει μοντέλα με τα εργαλεία που παρέχει. Μέσω του ProBuilder διατίθεται επίσης και το ProGrids, το οποίο προσφέρει ένα οπτικό και λειτουργικό πλέγμα, το οποίο ασφαρίζει και στους τρεις άξονες. Το ProGrids επιτρέπει την ταχύτητα και την ποιότητα, καθιστώντας την επίπεδη κατασκευή απίστευτα γρήγορη, εύκολη και ακριβή. Είναι ιδιαίτερα βολικό για αρθρωτά περιβάλλοντα ή περιβάλλοντα πλακιδίων, αλλά επίσης επιταχύνει σημαντικά τη ροή εργασίας και αυξάνει την ποιότητα για όλους τους τύπους εργασίας.

3.4.6 TextMesh Pro

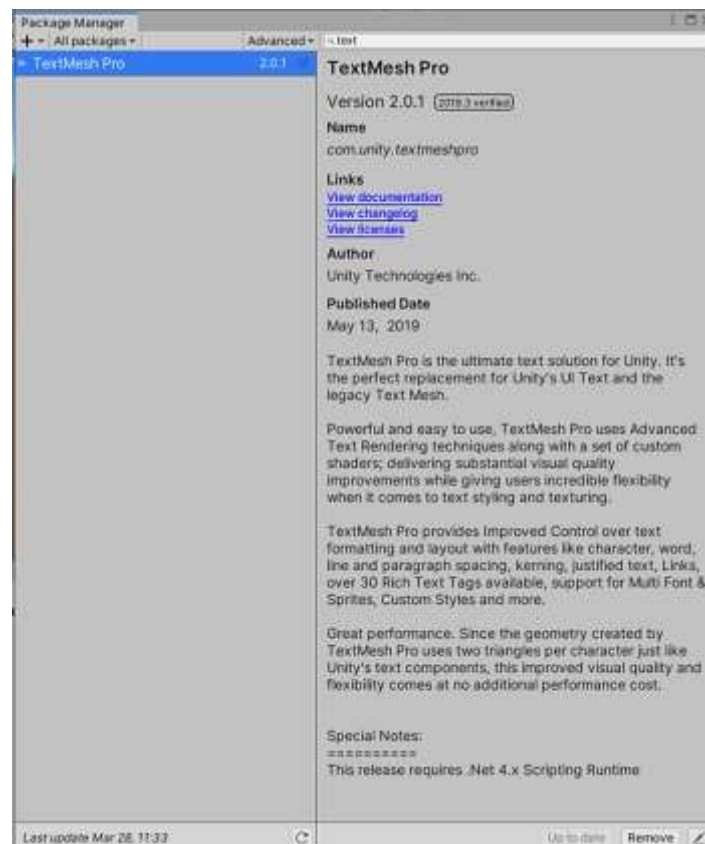
Το πακέτο TextMesh Pro (σχήμα 1.7), είναι η απόλυτη λύση κειμένου για τη Unity. Είναι η τέλεια αντικατάσταση για το Uni's UI Text και το παλαιό Text Mesh. Ισχυρό και εύκολο στη χρήση, το TextMesh Pro χρησιμοποιεί προηγμένες τεχνικές απόδοσης κειμένου μαζί με ένα σύνολο προσαρμοσμένων shaders. Παρέχει σημαντικές βελτιώσεις στην οπτική ποιότητα, ενώ παράλληλα δίνει στους χρήστες απίστευτη ευελιξία όσον αφορά το στυλ και την υφή κειμένου. Επιπλέον, παρέχει

Εγκατάσταση Unity και βασικές ρυθμίσεις πλατφόρμας

βελτιωμένο έλεγχο της μορφοποίησης και της διάταξης κειμένου με χαρακτηριστικά όπως character, word, line-paragraph spacing, kerning, justified text, links, rich text tags, υποστήριξη για multi font & sprites, custom styles και άλλα. Αυτό το πακέτο είναι ο μόνος τρόπος που χρησιμοποιήθηκε για τη σύνταξη κειμένων στη Δ.Ε.. Είναι σημαντικό να σημειωθεί πως η γεωμετρία που δημιουργήθηκε από το TextMesh Pro χρησιμοποιεί δύο τρίγωνα ανά χαρακτήρα όπως τα στοιχεία κειμένου της Unity. Αυτή η βελτιωμένη οπτική ποιότητα και ευελιξία δεν επιβαρύνεται με επιπλέον κόστος απόδοσης.



Σχήμα 3.6: ProBuilder

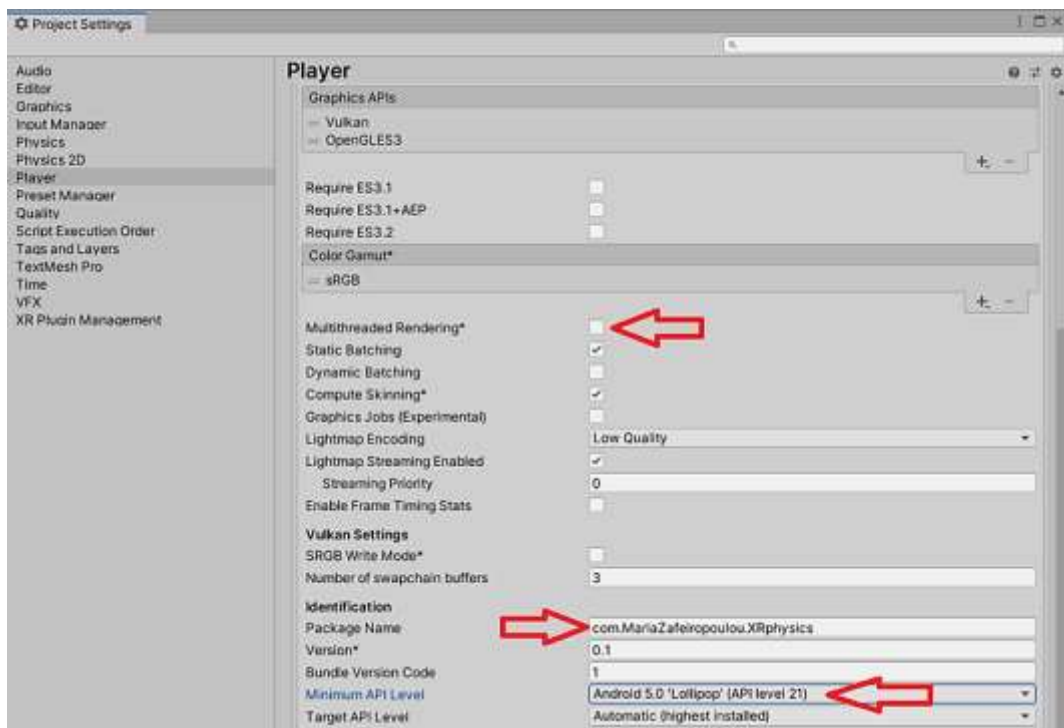


Σχήμα 3.7: TextMesh Pro

3.5 Project Build settings

Αυτές οι ρυθμίσεις αποτελούνται από δύο βασικά μέρη. Το πρώτο μέρος είναι οι ρυθμίσεις του παίκτη (player Settings) και βρίσκονται στο unity→File→Build settings→Player settings→Player→Other settings, και το δεύτερο μέρος είναι οι σκηνές του project στο unity→File→Build settings.

Αρχικά, οι ρυθμίσεις του παίκτη φαίνονται στο σχήμα 1.8. Το Multithreaded Reandering χρειάστηκε να απενεργοποιηθεί διότι παρουσιαζόταν πρόβλημα κατά το πέρασμα του αρχείου .apk στη συσκευή τηλεφώνου. Στο πεδίο indentification→Package Name, το όνομα έπρεπε να είναι της μορφής “com.CompanyName.ProductName”, ρυθμίσεις όπου προηγήθηκαν. Τέλος, στο πεδίο indentification→Minimum API Level, επιλέχθηκε το “Android 5.0 ‘Lollipop’ (API Level 21)”, διότι είναι το χαμηλότερο επιτρεπτό επίπεδο λειτουργικού συστήματος το οποίο μπορεί να υποστηρίξει το Vuforia Engine AR.



Σχήμα 3.8: Player settings

Το δεύτερο μέρος που προαναφέρθηκε αποτελείται από τις σκηνές του project όπως αυτές φαίνονται στο σχήμα 1.9.

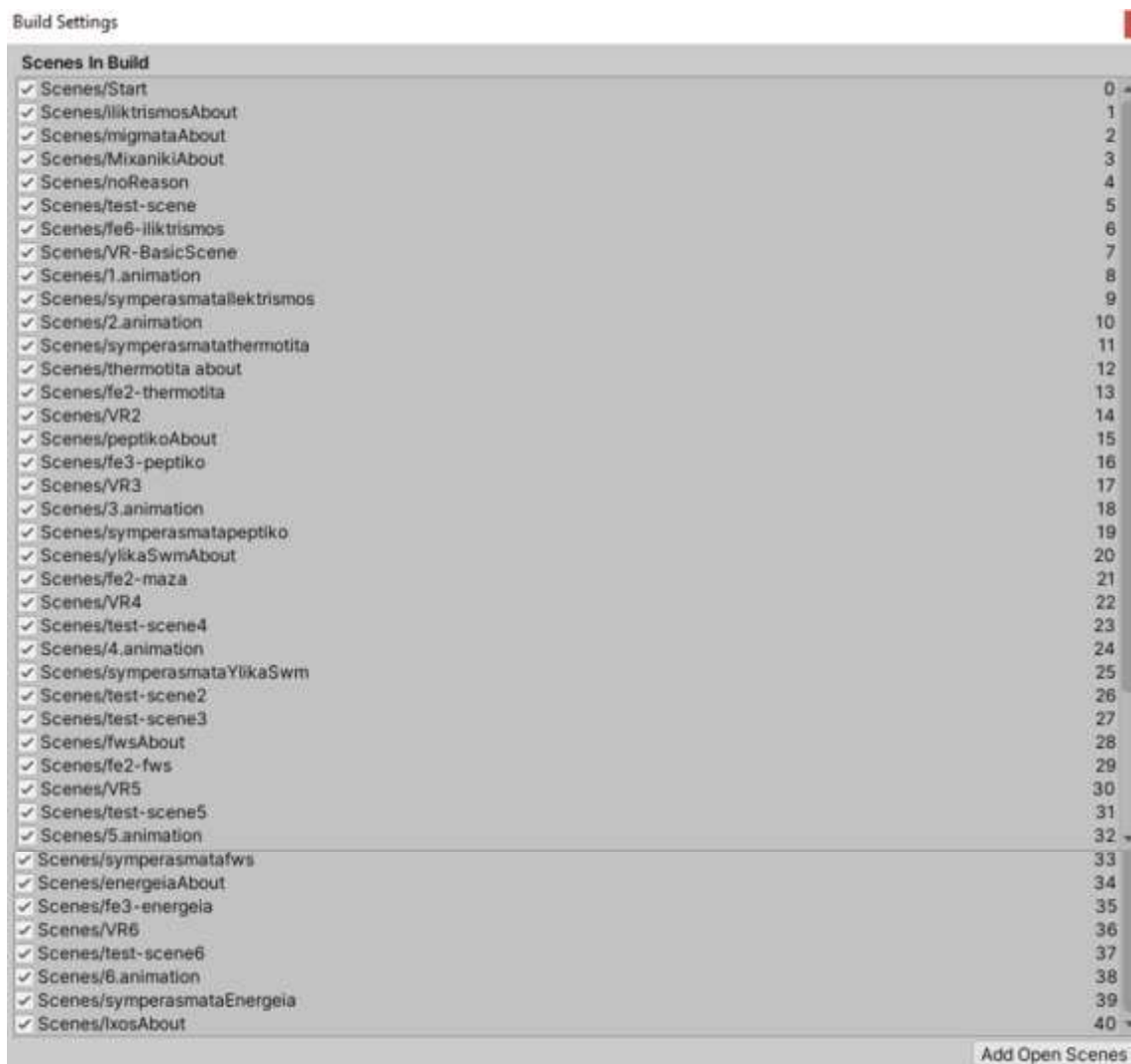


Σχήμα 3.9: Οι σκηνές του project

Όλες οι σκηνές είναι στο σύνολο σαράντα και ακολουθούν μια συγκεκριμένη μορφολογία. Πέρα από τις σκηνές του βασικού κορμού του παιχνιδιού, το κάθε πείραμα αποτελείται από τις έξι εξής σκηνές:

- Το όνομα του κεφαλαίου ακολουθούμενο από τη λέξη “About” πχ fwsAbout για το κεφάλαιο «ΦΩΣ».
- Το “fe” ακολουθούμενο από το υποκεφάλαιο του βιβλίου στο οποίο ανήκει το πείραμα και το όνομα του κεφαλαίου πχ fe2-fws για το κεφάλαιο «ΦΩΣ».
- Το “VR” ακολουθούμενο από έναν τυχαίο αριθμό από το ένα ως το έξι, όσα και τα πειράματα. Σε αυτή τη σκηνή γίνεται η εμβύθιση σε περιβάλλον εικονικής πραγματικότητας.
- Το “testscene” ακολουθούμενο από τον παραπάνω τυχαίο αριθμό.
- Το “animation” ακολουθούμενο από τον παραπάνω τυχαίο αριθμό, το οποίο αποτελεί και τη σκηνή του πειράματος.
- Το “Symperasmata” ακολουθούμενο από το όνομα του κεφαλαίου πχ SymperasmataFws για το κεφάλαιο «ΦΩΣ».

Όλες αυτές οι σαράντα σκηνές ήταν αναγκαίο να μπουν σε μια ιεραρχία προκειμένου να μπορεί να γίνει η εναλλαγή μεταξύ τους. Στο σχήμα 1.10 παρουσιάζεται αυτή η ιεραρχία.



Σχήμα 3.10: Ιεραρχία όλων των σκηνών του project

3.6 Mobile settings (Android)

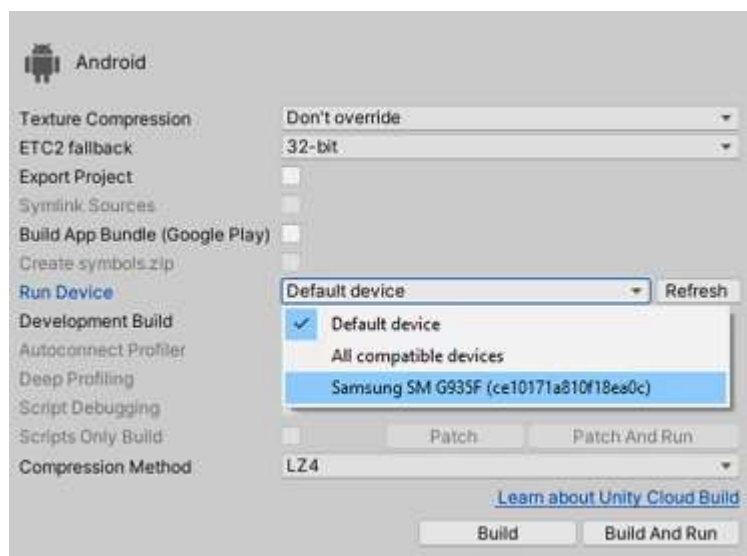
Μετά τον ολοκλήρωση των παραπάνω ρυθμίσεων, σειρά έχουν οι ρυθμίσεις του κινητού της συσκευής αλλά παράλληλα και της unity. Αρχικά έγινε η προετοιμασία της συσκευής του κινητού προκειμένου να μπορεί να εγκατασταθεί μια εφαρμογή επαυξημένης πραγματικότητας. Τα βήματα που ακολούθησαν είναι τα εξής:

1. Ρυθμίσεις κινητού → Πληροφορίες για το τηλέφωνο → Πληροφορίες για το λογισμικό → tab πολλές φορές ώσπου να εμφανιστεί ο αριθμός κατασκευής → πληκτρολόγηση PIN κινητού → Η λειτουργία προγραμματιστή για κινητά έχει ενεργοποιηθεί.
2. Ρυθμίσεις κινητού → επιλογές προγραμματιστή → εντοπισμός σφαλμάτων USB → ON.
3. Εγκατάσταση εφαρμογής από το Google play: ARCore Instant Preview.
4. Εγκατάσταση εφαρμογής από το Google play: Unity remote5.
5. Ακολούθησαν οι ρυθμίσεις στη unity: Unity → Edit → project settings → editor → unity remote → device → any android device, όπως φαίνεται στο σχήμα 1.11.



Σχήμα 3.11: Mobile settings Unity

6. Σύνδεση της συσκευής του κινητού με τον υπολογιστή μέσω USB. Για να επιβεβαιωθεί η επιτυχής σύνδεση στις ρυθμίσεις file → build settings → Android → Run device, πρέπει να εμφανίζεται η συσκευή όπως φαίνεται στο σχήμα 1.12.



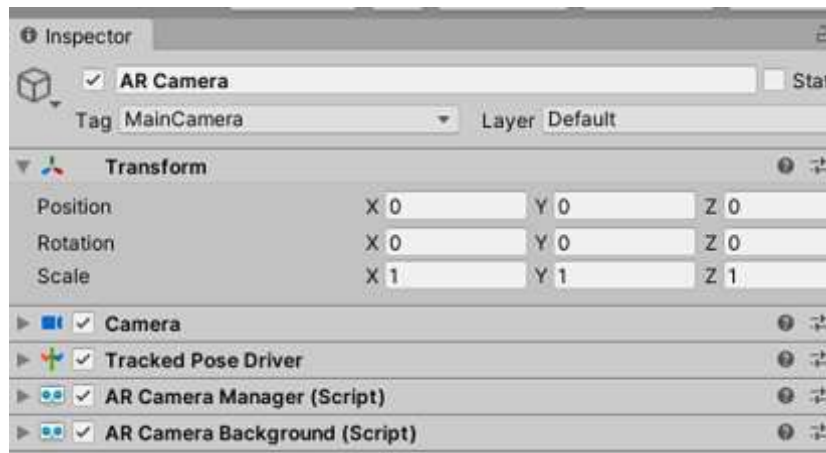
Σχήμα 3.12: Σύνδεση συσκευής κινητού με Unity

7. Τελευταίο βήμα για το πέρασμα του παιχνιδιού στη συσκευή τηλεφώνου (του αρχείου .apk) είναι: file → build settings → build system → Gradle + Build and run.

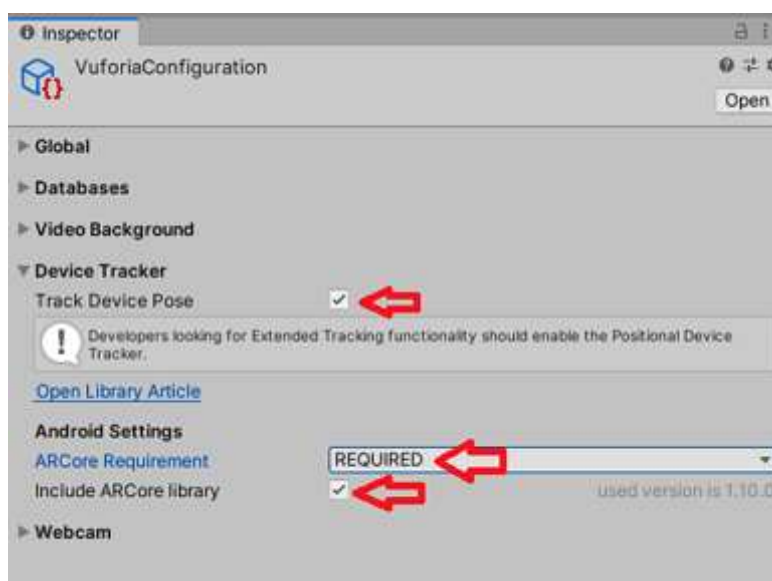
3.7 AR settings

Για την πραγματοποίηση των ρυθμίσεων αυτών είναι απαραίτητο αρχικά να δημιουργηθεί ένα αντικείμενο AR/XR κάμερας και στη συνέχεια ένα αντικείμενο image target. Για τη δημιουργία μιας κάμερας επαυξημένης πραγματικότητας στο παιχνίδι ακολουθήθηκαν τα εξής βήματα:

1. unity → GameObject → XR → AR Session.
2. unity → GameObject → XR → AR Session Origin (το οποίο συμπεριλαμβάνει την AR Camera).
3. Απενεργοποίηση της Main Camera από το tag “main camera”.
4. Ενεργοποίηση της AR Camera ως Main camera (main camera tag = ON).
5. Προσθήκη γενικών components στην AR camera (σχήμα 1.13), και παραμετροποίηση κάποιων στοιχείων (σχήμα 1.14) προκειμένου να δουλεύει η AR camera σωστά.

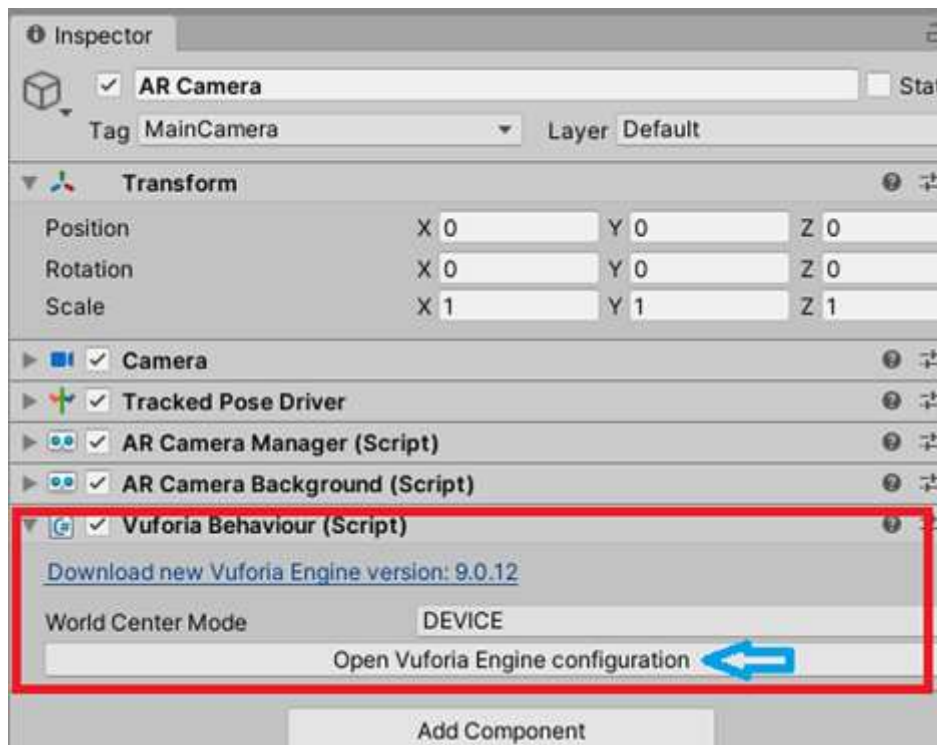


Σχήμα 3.13: Γενικά components AR camera



Σχήμα 3.14: Παραμετροποίηση στοιχείων AR camera

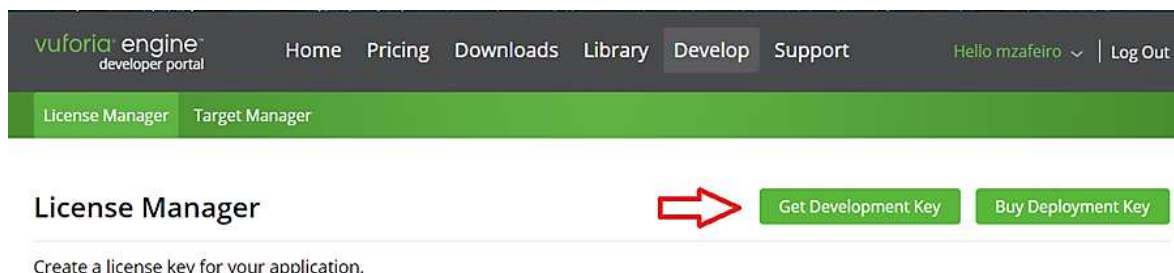
6. Προσθήκη βασικού component στην AR Camera: “Vuforia behavior script”.
7. Επιλογή του DEVICE ως World center mode για το Vuforia behavior script της AR κάμερας.
8. Άνοιγμα του “open vuforia engine configuration” (σχήμα 1.15) για την εισαγωγή κλειδιού.



Σχήμα 3.15: Vuforia Behaviour Script

3.7.1 Vuforia Engine license Key

Στο πεδίο unity → ArCamera → Inspector → vuforia Behaviour script → open vuforia engine configuration → vuforiaConfiguration → global → App License Key → Add License, χρειάζεται η εισαγωγή ενός κλειδιού προκειμένου να ενεργοποιηθεί στη unity το Vuforia Engine. Η δημιουργία αυτού του κλειδιού γίνεται μέσω του ιστότοπού της [12] δημιουργώντας έναν προσωπικό λογαριασμό και στη συνέχεια Develop → License Manager → Get Development Key (σχήμα 1.16).



Σχήμα 3.16: Vuforia Engine/Get Development Key

Στη συνέχεια, έγινε η δημιουργία ονόματος για το κλειδί, και με την αποδοχή των όρων και την επιβεβαίωση confirm (σχήμα 1.17), δημιουργήθηκε αυτόματα ένα κλειδί από το vuforia Engine με όνομα “ptyxiaki_finalkey” (σχήμα 1.18).

The screenshot shows the 'License Manager' tab in the Vuforia Engine interface. At the top, there are two tabs: 'License Manager' (selected) and 'Target Manager'. Below the tabs is a link 'Back To License Manager'. The main heading is 'Add a free Development License Key'. There is a text input field for 'License Name *' containing the text 'ptyxiaki_finalkey'. Below the input field is the text 'You can change this later'. Underneath is the section 'License Key' with the following details: 'Develop', 'Price: No Charge', 'Reco Usage: 1,000 per month', 'Cloud Targets: 1,000', 'VuMark Templates: 1 Active', and 'VuMarks: 100'. A checkbox is checked, with the text 'By checking this box, I acknowledge that this license key is subject to the terms and conditions of the Vuforia Developer Agreement.' At the bottom, there are two buttons: 'Cancel' and 'Confirm'.

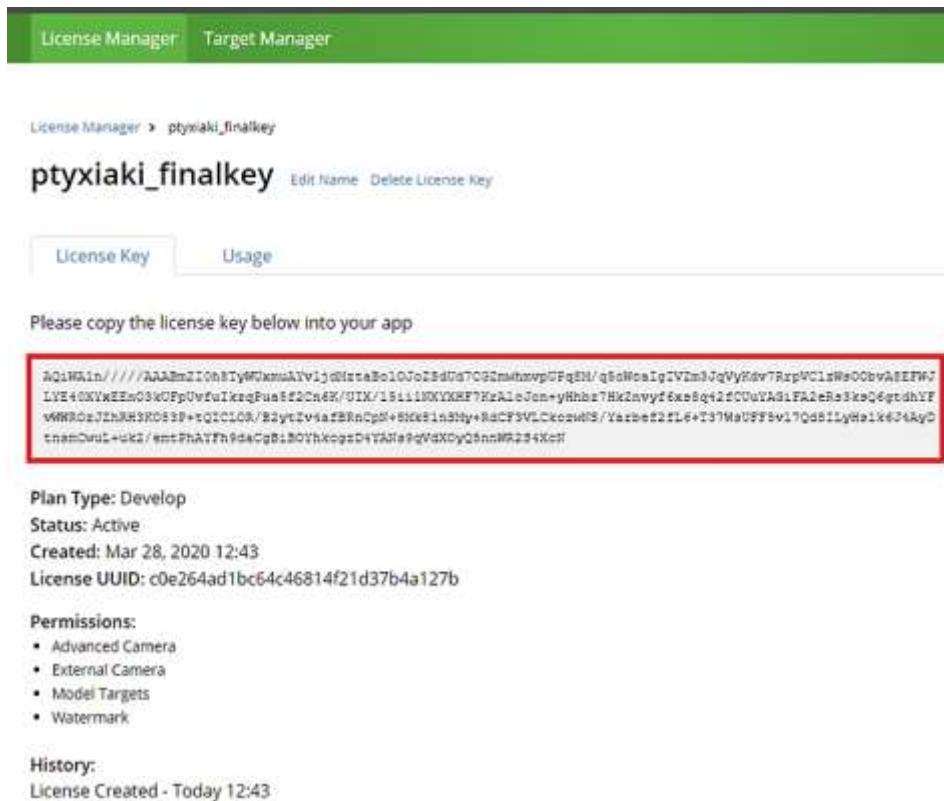
Σχήμα 3.17: Vuforia Engine/add development license key

The screenshot shows the 'License Manager' page in the Vuforia Engine developer portal. The page has a navigation bar with 'vuforia engine developer portal', 'Home', 'Pricing', 'Downloads', 'Library', 'Develop' (selected), and 'Support'. There is a user profile 'Hello mzafeiro' and a 'Log Out' link. Below the navigation bar are two tabs: 'License Manager' (selected) and 'Target Manager'. There are two buttons: 'Get Development Key' and 'Buy Deployment Key'. The main heading is 'License Manager'. Below the heading is the text 'Create a license key for your application.' and a search input field. Below the search field is a table with the following data:

Name	Primary UUID ⓘ	Type	Status ▾	Date Modified
ptyxiaki_finalkey	N/A	Develop	Active	Mar 28, 2020

Σχήμα 3.18: Vuforia Engine/ptyxiaki_finalkey

Με την ολοκλήρωση της δημιουργίας του κλειδιού και πατώντας πάνω στο όνομα του κλειδιού, η vuforia engine δείχνει τον κωδικό που χρειάζεται να περαστεί στη unity, καθώς επίσης και κάποια βασικά χαρακτηριστικά αυτού του κλειδιού όπως το Plan type, status, date, license uuid και permissions (advanced camera, external camera, model targets και watermark), όπως φαίνεται και στο σχήμα 1.19.



Σχήμα 3.19: Vuforia Engine/license key

Τέλος, έγινε αντιγραφή αυτού του κλειδιού και επικόλληση στο πεδίο unity → ArCamera → Inspector → vuforia Behaviour script → open vuforia engine configuration → vuforiaConfiguration → global → App License Key + Add License (σχήμα 1.20).



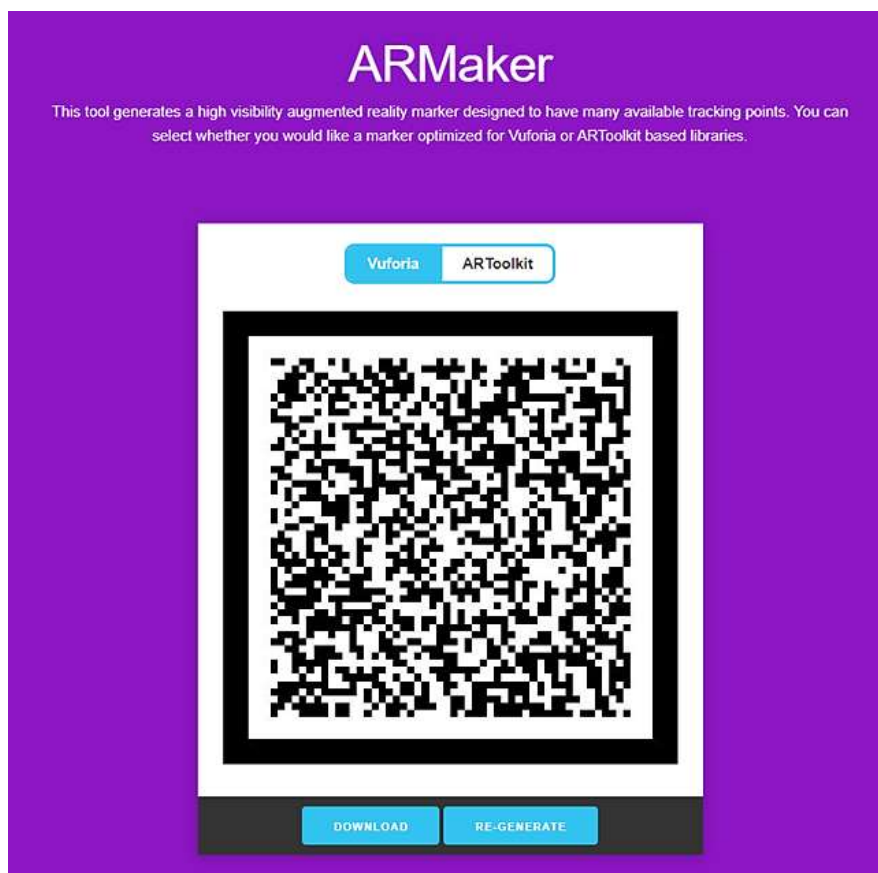
Σχήμα 3.20: VuforiaConfiguration/add License

3.7.2 Δημιουργία QRcodes από Generator

Τα κοινά σε όλους “qr-codes”, στη γλώσσα της unity χαρακτηρίζονται ως “image targets”. Τα image targets αντιπροσωπεύουν εικόνες που η Vuforia Engine μπορεί να εντοπίσει και να παρακολουθήσει. Η μηχανή ανιχνεύει και παρακολουθεί την εικόνα συγκρίνοντας τα εξαγόμενα φυσικά χαρακτηριστικά από την εικόνα της κάμερας με μια γνωστή βάση δεδομένων που αποτελείται από image targets. Μόλις εντοπιστεί το image target, η Vuforia Engine παρακολουθεί την εικόνα και αυξάνει το περιεχόμενο, χωρίς προβλήματα, χρησιμοποιώντας την καλύτερη τεχνολογία παρακολούθησης εικόνων στην αγορά.

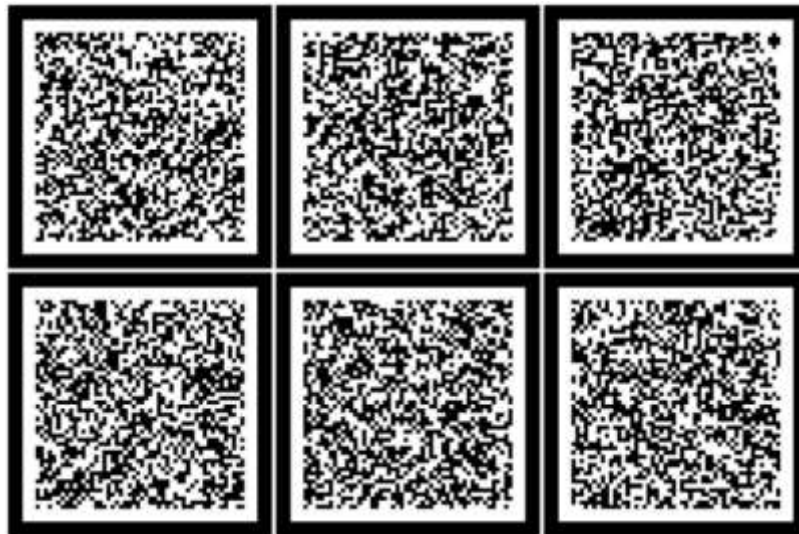
Τα image targets μπορούν να δημιουργηθούν χρησιμοποιώντας εικόνες JPG ή PNG σε RGB ή κλίμακα του γκρι. Το μέγεθος των εικόνων εισαγωγής πρέπει να είναι 2,25 MB ή λιγότερο και να έχει ελάχιστο πλάτος 320 pixel. Τα image targets, μπορούν να χρησιμοποιήσουν οποιαδήποτε επίπεδη εικόνα που παρέχει επαρκείς λεπτομέρειες για να εντοπιστεί από τη Vuforia Engine. Η δημιουργία image targets αποτελείται από δύο βήματα. Το πρώτο βήμα είναι ο σχεδιασμός των image targets και δεύτερο η διασφάλιση ότι διατηρούν τις οδηγίες στους ακόλουθους οδηγούς για το μέγεθος, τον ορισμό και τη βελτιστοποίηση των εικόνων. Οι δυνατότητες που εξάγονται από αυτές τις εικόνες αποθηκεύονται σε μια βάση δεδομένων cloud.

Η δημιουργία των έξι image targets του παιχνιδιού έγινε μέσω του εργαλείου που παρέχει η ιστοσελίδα “shawnlehner”[13]. Η επιλογή ενός generator για τη δημιουργία AR image targets έγινε για τη διασφάλιση της μέγιστης ποιότητας εικόνας και κατά συνέπεια τη διασφάλιση της μέγιστης απόδοσης των δυνατοτήτων τους. Η διαδικασία ακολούθησε τα εξής βήματα: Επιλογή “vuforia” → download → re-generate 5 φορές, όπως φαίνεται στο σχήμα 1.21.



Σχήμα 3.21: Δημιουργία image targets

Σημαντικό είναι να σημειωθεί πως τα image targets επηρεάζονται από τη στυλπνότητα και τις αντανακλάσεις από πηγές φωτισμού. Για να διατηρηθεί η καλύτερη εμπειρία, το image target πρέπει να προβάλλεται με μέτρια φωτεινό και ομοιόμορφο φωτισμό με διάχυτο φωτισμό. Η τελική μορφή των image targets φαίνεται στο σχήμα 1.22.

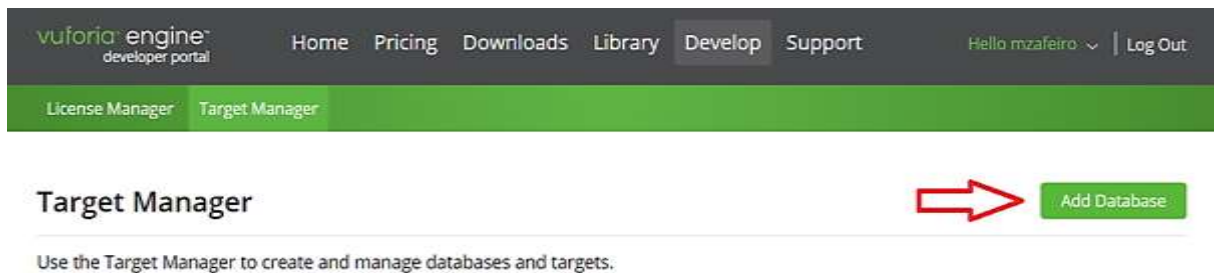


Σχήμα 3.22: Τα 6 image target του project

3.7.3 Δημιουργία βάσης από targets και εισαγωγή της στη Unity

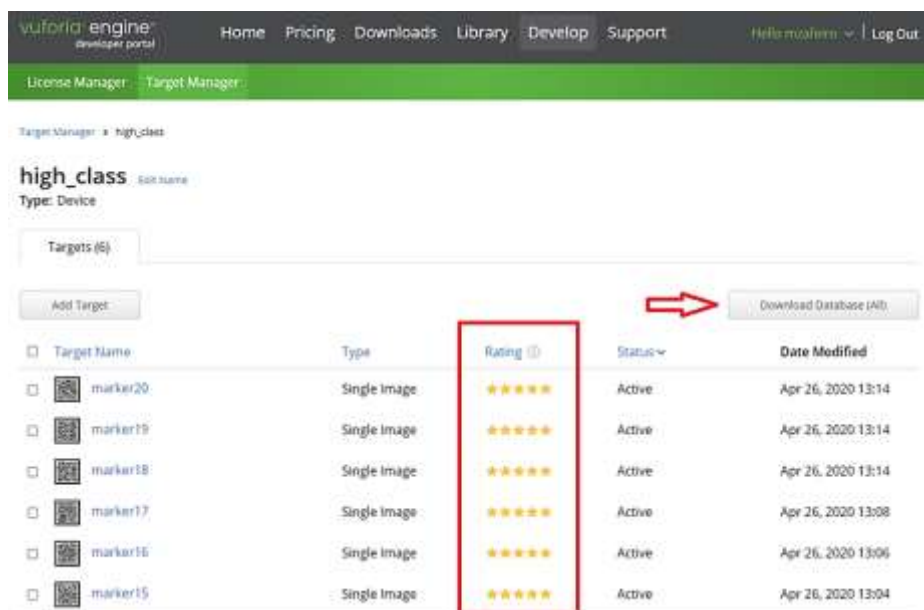
Μετά τη δημιουργία των έξι image targets έγινε η δημιουργία μιας βάσης η οποία αποτελείται από αυτά, προκειμένου να περαστεί στη unity και να χρησιμοποιηθεί. Η δημιουργία της βάσης των image targets έγινε μέσω της ιστοσελίδας του vuforia engine, επιλέγοντας:

1. vuforia engine → Develop → Target Manager → add database (σχήμα 1.23).



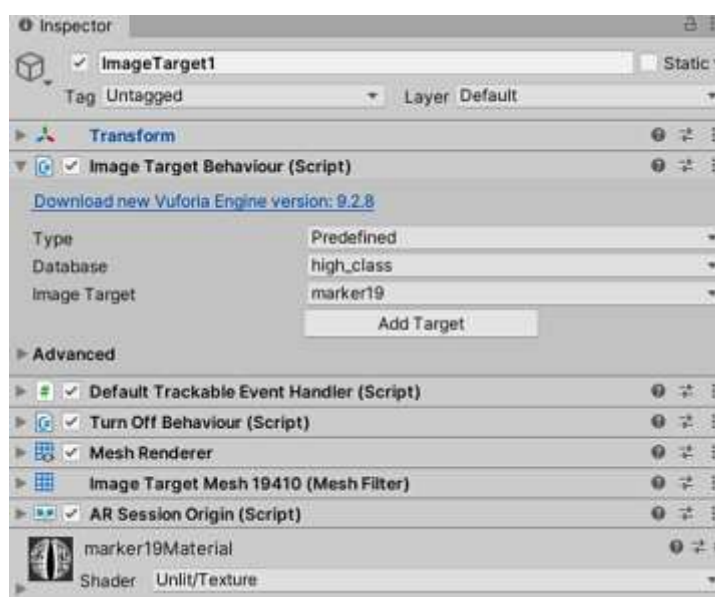
Σχήμα 3.23: Vuforia engine/add database

2. Πληκτρολογώντας database name: highclass και επιλέγοντας type = Device.
3. Κάνοντας κλικ πάνω στο όνομα της βάσης highclass → add target → file → browse (το αρχείο εικόνας του image target) → type = single image → Width=1 → add.
4. Επανάληψη του βήματος 3 αντίστοιχα για τα υπόλοιπα 5 image targets.
5. Αφού η βάση είναι έτοιμη και γίνουν αποδεκτά όλα τα image targets και η ποιότητά τους αξιολογηθεί με τη μέγιστη βαθμολογία (σχήμα 1.24), κλικ στο → download database all → select a development platform = Unity editor → download.



Σχήμα 3.24: Vuforia Engine/high_class database

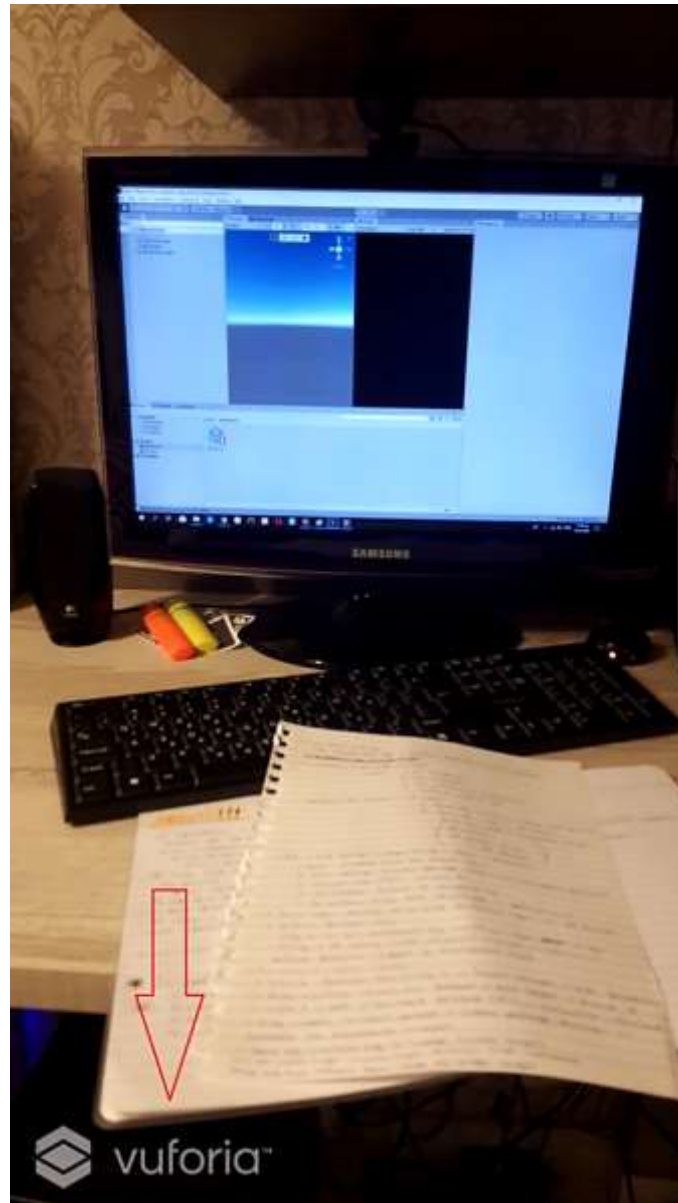
6. Το αρχείο της βάσης των image markers “highclass.unitypackage” είναι έτοιμο για εισαγωγή στη unity.
7. Για την εισαγωγή της βάσης στη unity: unity → assets → import package → custom package → browse (το αρχείο highclass.unitypackage) → import.
8. Δημιουργία αντικειμένου image target: unity → GameObject → Vuforia Engine → image target
9. Επιλογή image target από τη βάση: unity → imagetarget → Inspector → image target behavior script → Type= predefined → Database = highclass → Image marker = επιλογή ενός από τους 6.
10. Το image target είναι έτοιμο για χρήση και τα components του πρέπει να είναι της μορφής του σχήματος 1.25.



Σχήμα 3.25: Components του Image target

3.8 Testing

Για την επιβεβαίωση της ορθότητας όλων των παραπάνω ρυθμίσεων έγινε σύνδεση της συσκευής κινητού στη unity και στη συνέχεια build and run του project στο κινητό. Η ορθότητα επιβεβαιώθηκε με την εμφάνιση του λογότυπου της “vuforia” στην οθόνη του κινητού (σχήμα 1.26), με το άνοιγμα της εφαρμογής που έγινε εγκατάσταση.



Σχήμα 3.26: Στιγμιότυπο οθόνης για την επιβεβαίωση της ορθότητας των AR ρυθμίσεων

3.9 Επίλογος

Με την ολοκλήρωση των βασικών ρυθμίσεων, το πρόγραμμα είναι έτοιμο για τη δημιουργία των σκηνών του παιχνιδιού.

Κεφάλαιο 4ο: Βασικός κορμός παιχνιδιού-Panels

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφεται η δημιουργία και η λειτουργία των αντικειμένων της αρχικής σκηνής του παιχνιδιού, καθώς και των βασικών σκηνών που αναφέρονται στις ενότητες του βιβλίου.

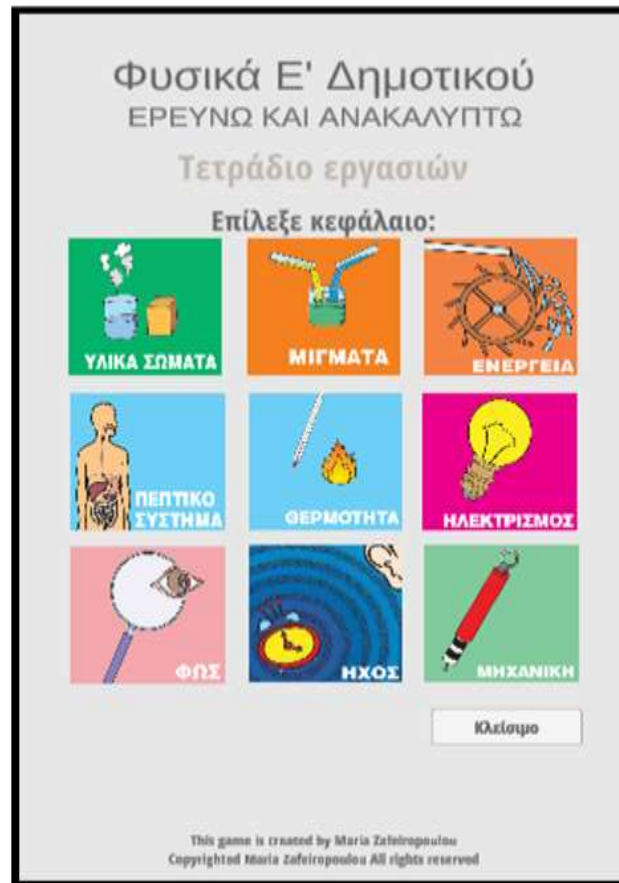
4.2 Φυσικά Ε' δημοτικού – Επιλογή πειραμάτων

Το εκπαιδευτικό υλικό που χρησιμοποιήθηκε είναι από τα Φυσικά Δημοτικού «Ερευνώ και Ανακαλύπτω», το Τετράδιο Εργασιών Ε' τάξης, Έκδοση 2015, ΙΤΥΕ "ΔΙΟΦΑΝΤΟΣ". Μετά από σχετική εισήγηση του Ινστιτούτου Εκπαιδευτικής Πολιτικής (Πράξη 26/27-06-2019 του Δ.Σ.), οι οδηγίες για τη διαχείριση της ύλης της σχολικής χρονιάς 2019-2020 για το μαθήματα των φυσικών επιστημών «Φυσικά δημοτικού» έβγαλαν εκτός διδακτέας ύλης τα κεφάλαια «Μίγματα» και «Μηχανική». Συγκεκριμένα, η ενότητα του βιβλίου «Μίγματα» έχει διδαχτεί στη Δ' τάξη στη μελέτη περιβάλλοντος, και η ενότητα «Μηχανική», μεταφέρθηκε στη ΣΤ' τάξη. Από τα υπολειπόμενα κεφάλαια εντός διδακτικής ύλης, επιλέχθηκε ένα πείραμα από το καθένα για υλοποίηση μέσω του παιχνιδιού, εκτός του κεφαλαίου «Ηχος». Η απόρριψη υλοποίησης πειράματος από το κεφάλαιο «Ηχος» βασίζεται στο γεγονός ότι χωρίς τη φυσική παρουσία ατόμου, το πείραμα θα ήταν επιρρεπές σε ασάφειες καταλληλότητας. Συνεπώς, το παιχνίδι παρέχει 6 πειράματα επιμορφωτικού χαρακτήρα για τα κεφάλαια: Υλικά σώματα, ενέργεια, πεπτικό σύστημα, θερμότητα, ηλεκτρισμός και φως.

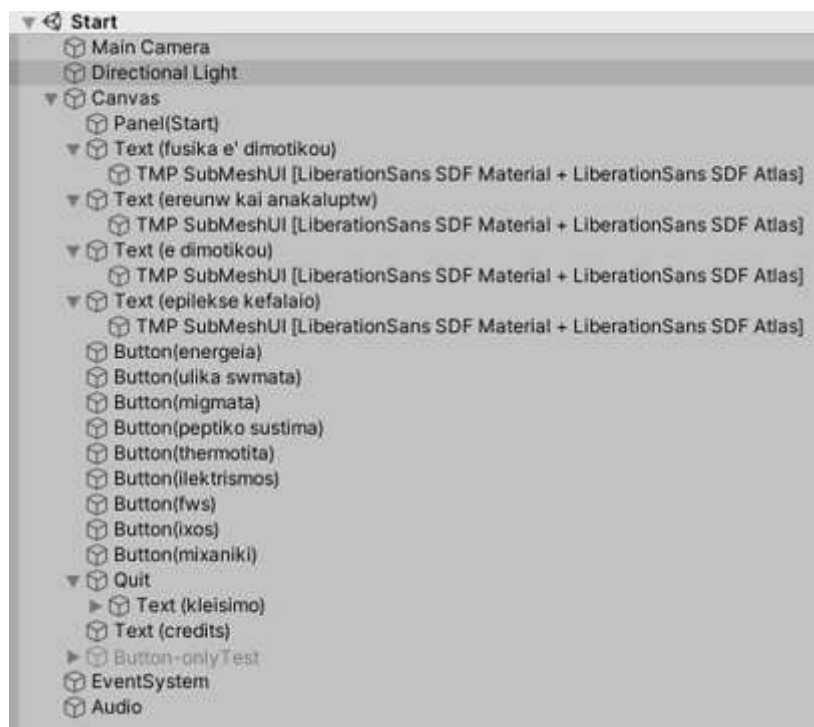
4.3 Αρχική σκηνή

Η μορφή της αρχικής σκηνής φαίνεται στο σχήμα 2.1 και η ιεραρχία των αντικειμένων της (σχήμα 2.2) αποτελείται από:

1. Μια main camera (unity → GameObject → Camera).
2. Ένα directional light (unity → GameObject → Light → Directional Light) για το σωστό φωτισμό της σελίδας.
3. Ένα event system (unity → GameObject → UI → Event system) για τη λειτουργία των κουμπιών και των εναλλαγών των σκηνών.
4. Ένα empty (unity → GameObject → Create empty) μετονομασμένο σε Audio, για τον ήχο και τις ρυθμίσεις του.
5. Ένα canva ο οποίος με τη σειρά του αποτελείται από:
 - Εννιά κουμπιά που αντιστοιχούν στα κεφάλαια του βιβλίου. Με την επιλογή τους εμφανίζονται οι βασικές σκηνές με τις ενότητες των αντίστοιχων κεφαλαίων.
 - Το κουμπί «κλείσιμο», το οποίο με την επιλογή του τερματίζεται η εφαρμογή.
 - Κείμενα τα οποία αντιπροσωπεύουν τον τίτλο και το περιεχόμενο της εφαρμογής καθώς και τα credits και τα πνευματικά δικαιώματα του δημιουργού της.



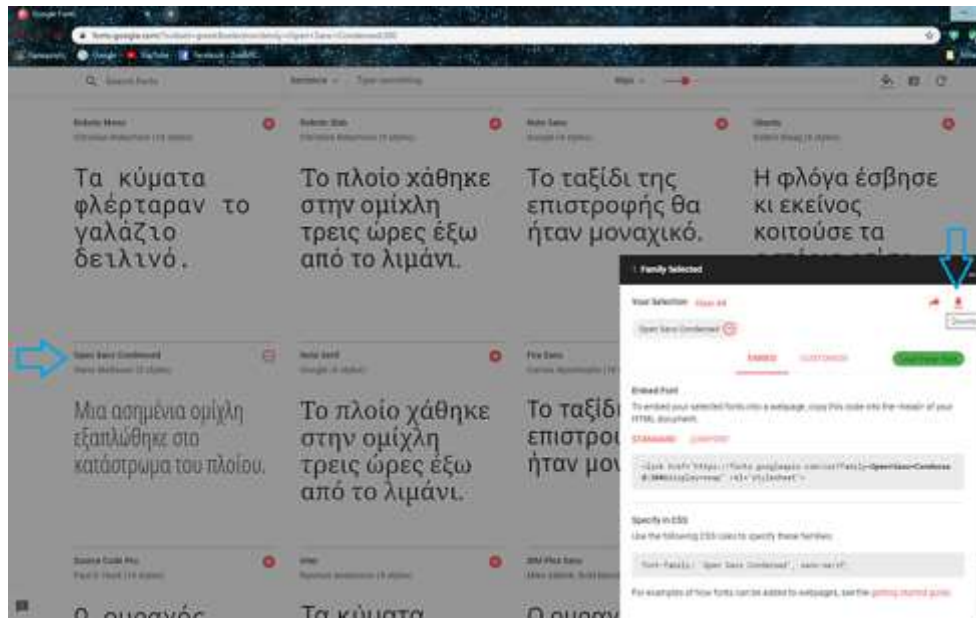
Σχήμα 4.1: Αρχική σκηνή παιχνιδιού



Σχήμα 4.2: Ιεραρχία αντικειμένων αρχικής σκηνής

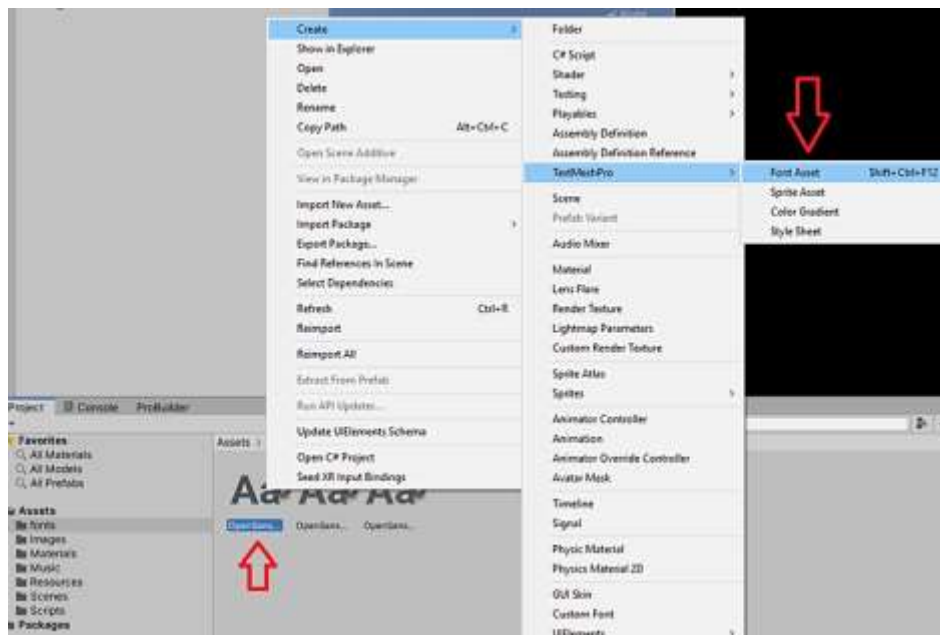
4.3.1 Κείμενα και γραμματοσειρά

Για τη δημιουργία των κειμένων της αρχικής σκηνής καθώς και όλων των κειμένων του παιχνιδιού χρησιμοποιήθηκε το πακέτο TextMesh Pro (unity → GameObject → UI → Text- TextMesh Pro). Οι υπάρχουσες γραμματοσειρές της unity δεν υποστήριζαν ελληνικά γράμματα, για αυτό χρησιμοποιήθηκε η γραμματοσειρά «Open Sans condensed» από τον ιστότοπο: “Google fonts”[14] (σχήμα 2.3).



Σχήμα 4.3: Google Fonts-Greek

Με την επιλογή download, αποθηκεύτηκε στον υπολογιστή ένα αρχείο .ttf το οποίο αποτελεί αρχείο γραμματοσειράς. Για την εισαγωγή της στη unity, έγινε drag and drop του .ttf αρχείου στα assets του project, μέσα στο φάκελο "Fonts". Για τη μετατροπή της σε αποδεκτή γραμματοσειρά unity έγιναν οι ρυθμίσεις του σχήματος 2.4.

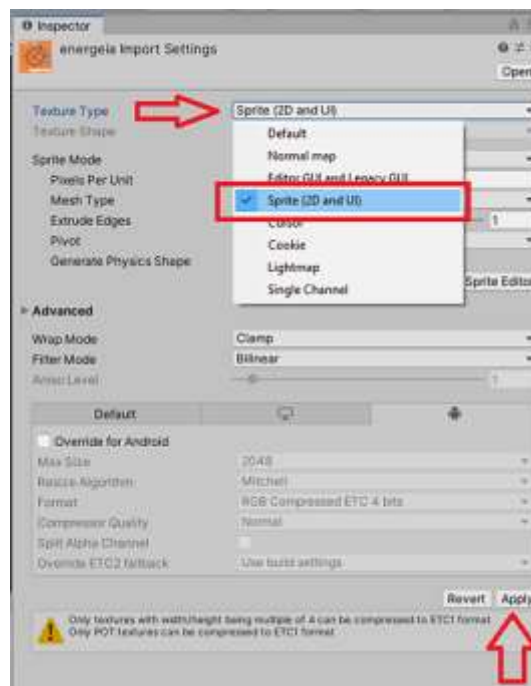


Σχήμα 4.4: Ρυθμίσεις δημιουργίας γραμματοσειράς unity

Με την ολοκλήρωση της μετατροπής της γραμματοσειράς σε έγκυρη μορφή γραμματοσειράς unity, η χρήση της μέσω TextMesh Pro γίνεται επιλέγοντάς την από τον inspector του κειμένου στο πεδίο TextMeshPro-Text(UI) → Main settings → Font Asset = OpenSansCondensed.

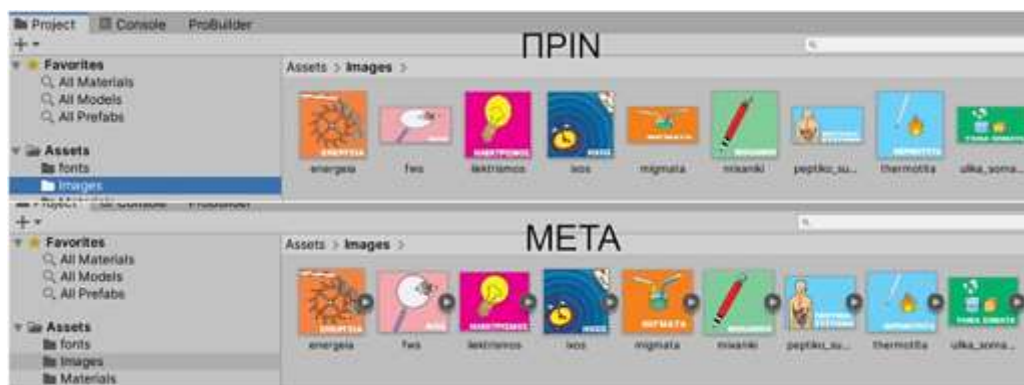
4.3.2 Εικόνες

Η μορφή των κουμπιών για την επιλογή κεφαλαίου αποτελείται από εννιά εικόνες οι οποίες αντιστοιχούν στις εικόνες του βιβλίου που αντιπροσωπεύουν το κάθε κεφάλαιο. Αρχικά έγινε η δημιουργία ενός φακέλου “images” στα assets του project, και στη συνέχεια drag and drop εκεί τα αρχεία .png μορφής. Η αρχική αυτή μορφή τους δεν είναι αποδεκτή από τη unity για χρήση τους σε TextMeshPro button. Ακολουθήθηκαν οι ρυθμίσεις που φαίνονται στο σχήμα 2.5, ξεχωριστά για τις 9 φωτογραφίες



Σχήμα 4.5: Ρυθμίσεις εικόνων

Με τις παραπάνω ρυθμίσεις οι φωτογραφίες έγιναν αποδεκτές για χρήση. Η διαφορά πριν και μετά τις ρυθμίσεις φαίνεται στο σχήμα 2.6.



Σχήμα 4.6: Μετατροπή εικόνων .png σε sprite

4.3.3 Κουμπιά και scripts

Η αρχική σελίδα αποτελείται από τα εννιά κουμπιά που αντιστοιχούν στα κεφάλαια του βιβλίου και το κουμπί “κλείσιμο” που τερματίζει την εφαρμογή. Η δημιουργία όλων των κουμπιών έγινε ως: unity → GameObject → UI → button-TextMeshPro. Πιο ειδικά, για τα εννιά κουμπιά των κεφαλαίων η εισαγωγή των εικόνων έγινε στο πεδίο inspector → Image → Source Image, με drag and drop των εικόνων. Για τη λειτουργικότητα όλων των κουμπιών είναι υπεύθυνο το script “menu” του σχήματος 2.7.

```

C:\Users\user\Desktop>physiaki>projects>final project>karantina>Assets>Scripts>Panels>Main menu>menu.cs
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3
4 public class menu : MonoBehaviour
5
6     public void quitGame()
7     {
8         Application.Quit();
9     }
10    public void startIlektrismos()
11    {
12        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +1);
13    }
14    public void startMigmata()
15    {
16        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +2);
17    }
18    public void startMixoniki()
19    {
20        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +3);
21    }
22    public void startOdigies()
23    {
24        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +4);
25    }
26    public void startThermotita()
27    {
28        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +12);
29    }
30    public void startPeptiko()
31    {
32        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +15);
33    }
34    public void startVlikaSwmata()
35    {
36        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +20);
37    }
38    public void startFws()
39    {
40        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +28);
41    }
42    public void startEnergeia()
43    {
44        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +34);
45    }
46    public void StartIxos()
47    {
48        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +48);

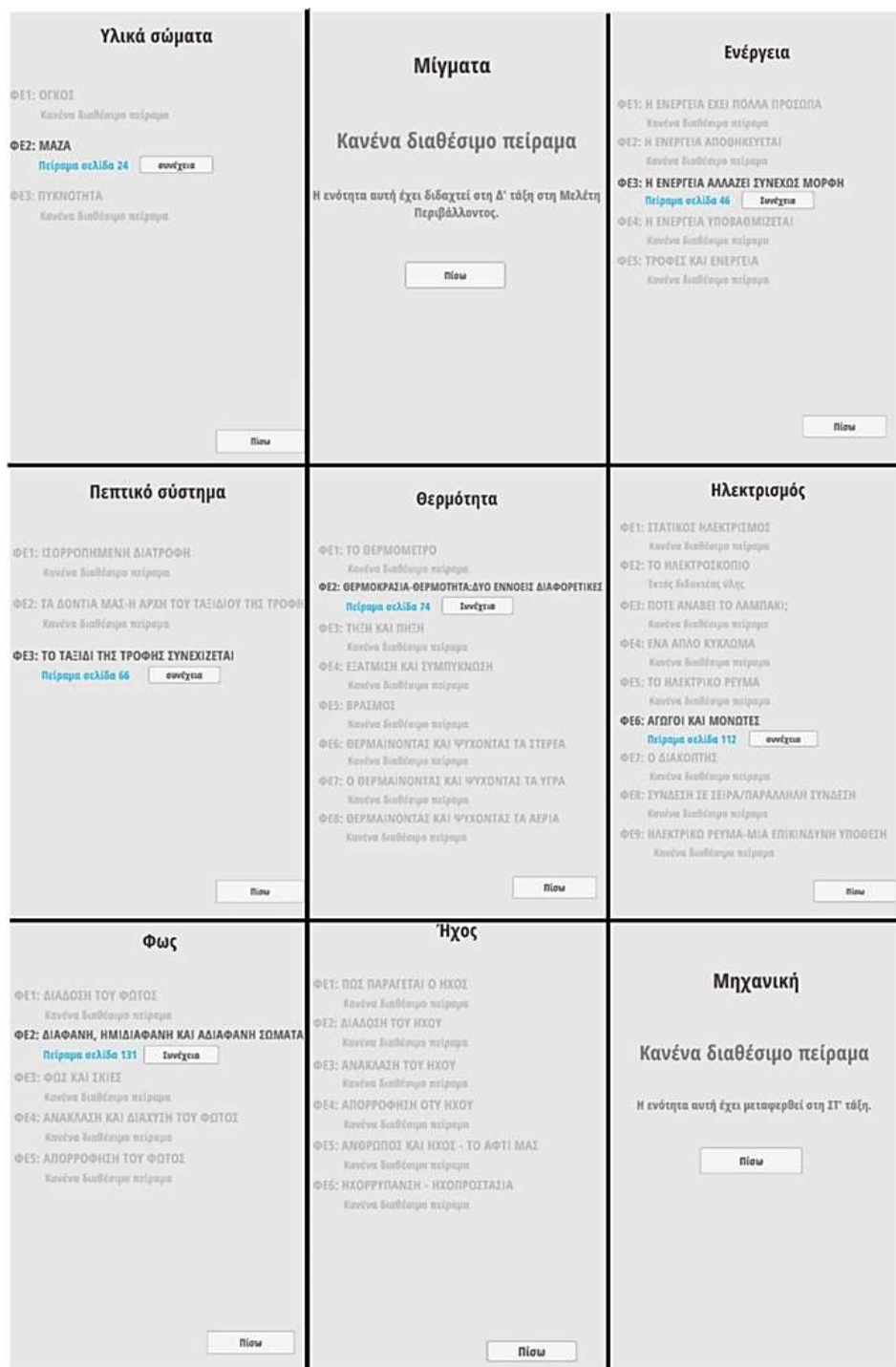
```

Σχήμα 4.7: Script "menu"

Το script “menu” αποτελείται από δέκα μεθόδους και χρησιμοποιεί τις βιβλιοθήκες UnityEngine και UnityEngine.SceneManagement. Το κουμπί “κλείσιμο” χρησιμοποιεί τη μέθοδο quitGame() η οποία τερματίζει την εφαρμογή. Τα εννιά κουμπιά των κεφαλαίων χρησιμοποιούν τις αντίστοιχες μεθόδους start() προκειμένου να αλλάζει η σκηνή στην αντίστοιχη about (βλέπε αρίθμηση σκηνών σχήμα 1.10). Το script “menu” έγινε drag and drop στο inspector του “panel(start)” της ιεραρχίας των αντικειμένων. Στη συνέχεια, στο inspector του κάθε κουμπιού, στο πεδίο button → On Click() → Select object = drag and drop το panel(start) το οποίο φέρει το script → select function = την αντίστοιχη μέθοδο κάθε κουμπιού από το script. Μετά από τις ρυθμίσεις των κουμπιών, επιλέγοντάς τα, αλλάζουν κατάλληλα οι σκηνές προχωρώντας στις about κάθε κεφαλαίου ή γυρίζοντας πίσω στην αρχική ή κλείνοντας την εφαρμογή.

4.4 Σκηνές “About”

Με το πάτημα ενός από τα εννιά κουμπιά των κεφαλαίων, ανοίγει η αντίστοιχη σκηνή about η οποία δίνει πληροφορίες για τις ενότητες του κεφαλαίου και για τα διαθέσιμα πειράματα. Κάθε σκηνή about αποτελείται από ένα ή δύο κουμπιά αναλόγως αν υπάρχει διαθέσιμο πείραμα ή όχι. Τα δύο αυτά κουμπιά αποτελούν τη «συνέχεια» στην επόμενη σκηνή με τις οδηγίες του πειράματος και το «πίσω» στην αρχική σελίδα. Το scripting αυτών των κουμπιών είναι παρόμοιο με αυτό του script “menu” των μεθόδων που χρησιμοποιούν τη βιβλιοθήκη UnityEngine.SceneManagement, για την εναλλαγή των σκηνών με βάση τον αριθμό τους στην ιεραρχία. Στο σχήμα 2.8 φαίνονται όλες οι σκηνές about.



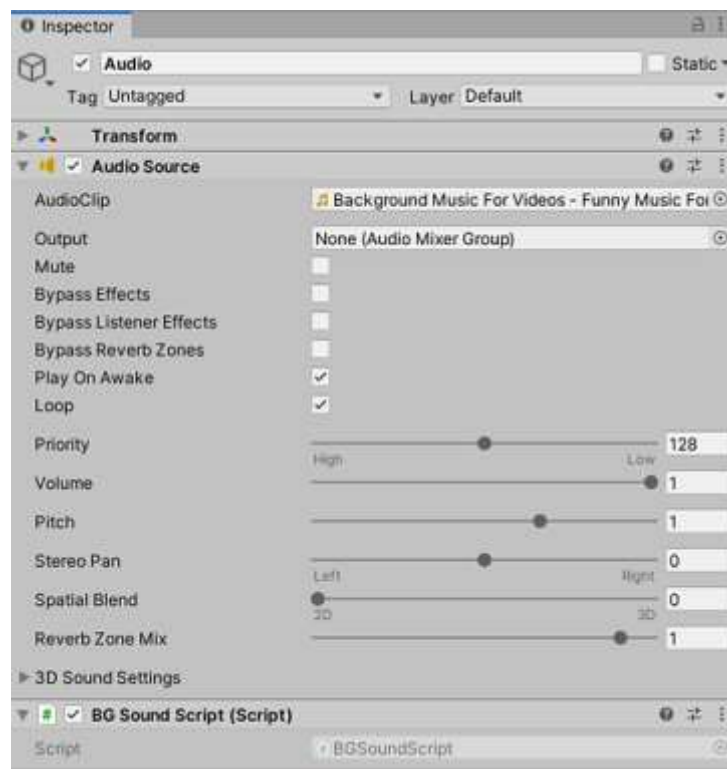
Σχήμα 4.8: Όλες οι σκηνές about

4.5 Μουσική και script

Τα αρχεία του ήχου είναι συνολικά επτά. Τα έξι αντιστοιχούν στα έξι πειράματα, ενώ το ένα στην αρχική σκηνή και στις επόμενες σκηνές πριν από τις VR-scenes των AR πυλών. Είναι της μορφής .mp3, η οποία είναι και απόδεκτη από τη unity. Όλα τα αρχεία ήχου βρίσκονται στα assets του project στο φάκελο “music”. Αρχικά έγινε η δημιουργία ενός empty με όνομα “audio” για τη διαχείριση του ήχου και των ρυθμίσεών του. Για την εισαγωγή του ήχου της αρχικής σκηνής, προστέθηκε στο audio το component “Audio source” με τις εξής ρυθμίσεις:

- Στο πεδίο “AudioClip”, drag and drop το αρχείο .mp3 του ήχου της αρχικής σκηνής.
- Play On Awake = ON, για την εκκίνηση της μουσικής με το που ανοίγει η εφαρμογή.
- Loop = ON, διότι το αρχείο ήχου έχει διάρκεια περίπου 2 λεπτά και αν ο χρήστης παραμείνει στην αρχική σκηνή πάνω από δύο λεπτά ο ήχος να συνεχίσει να παίζει.

Ένα ακόμα component που προστέθηκε στο audio είναι το script “BG Sound Script”. Το inspector του audio ολοκληρωμένο είναι της μορφής που φαίνεται στο σχήμα 2.9.



Σχήμα 4.9: Audio Components

Το script “BG Sound Script” (σχήμα 2.10), χρησιμοποιεί τις βιβλιοθήκες System.Collections, System.Collections.Generic, UnityEngine και UnityEngine.SceneManagement. Αποτελείται από:

- τη void awake(), η οποία κρατάει το αρχείο ήχου ενεργό μεταξύ των εναλλαγών των σκηνών που πρέπει να παίζει ασταμάτητα ο ήχος.
- τη void update(), η οποία μόλις γίνει εναλλαγή σκηνής σε VR-Scene, που πρέπει να αλλάξει το αρχείο ήχου, αλλά καταστρέφει το αρχείο ήχου της αρχικής σκηνής.

```

BGSoundScripts X
C:\Users\user\Desktop>physiaki>projects>final project>karantina>Assets>BGSoundScripts.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6
7  public class BGSoundScript : MonoBehaviour
8  {
9      // Start is called before the first frame update
10     void Start()
11     {
12
13     }
14     private static BGSoundScript instance = null;
15     public static BGSoundScript Instance
16     {
17         get { return instance; }
18     }
19
20     void Awake()
21     {
22         if (instance != null && instance != this)
23         {
24             Destroy(this.gameObject);
25             return;
26         }
27         else
28         {
29             instance = this;
30         }
31         DontDestroyOnLoad(this.gameObject);
32     }
33
34     void Update()
35     {
36         //for every scene i want to stop the music duplicate this if and change the name of a ascene
37         if (SceneManager.GetActiveScene().name == "VR-BasicScene")
38         {
39             Destroy(this.gameObject);
40         }
41         if (SceneManager.GetActiveScene().name == "VR2")
42         {
43             Destroy(this.gameObject);
44         }
45         if (SceneManager.GetActiveScene().name == "VR3")
46         {
47             Destroy(this.gameObject);
48         }
49         if (SceneManager.GetActiveScene().name == "VR4")
50         {
51             Destroy(this.gameObject);
52         }
53         if (SceneManager.GetActiveScene().name == "VR5")
54         {
55             Destroy(this.gameObject);
56         }
57         if (SceneManager.GetActiveScene().name == "VR6")
58         {
59             Destroy(this.gameObject);
60         }
61     }
62 }

```

Σχήμα 4.10: BG Sound Script

4.6 Επίλογος

Μετά την ολοκλήρωση της αρχικής οθόνης και των βασικών σκηνών του παιχνιδιού, ακολουθεί η δημιουργία των ειδικών σκηνών των κεφαλαίων του βιβλίου, στις οποίες υλοποιούνται τα πειράματα.

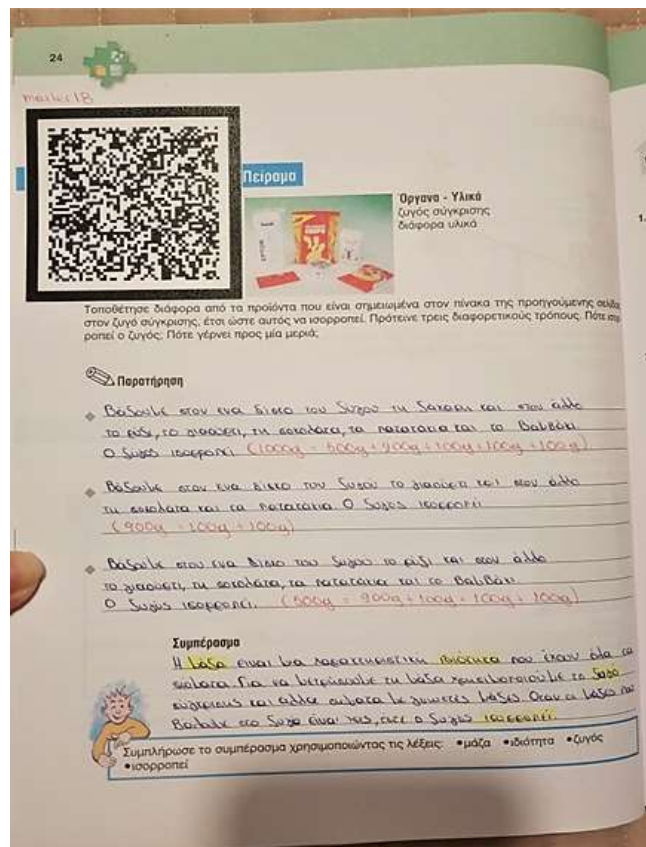
Κεφάλαιο 5ο: Εφαρμογή πειράματος για το κεφάλαιο Υλικά σώματα

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται οι μαθησιακοί στόχοι του πειράματος της σελίδας 24 του κεφαλαίου «Υλικά σώματα» του σχολικού βιβλίου. Επιπλέον, αναλύονται όλα τα βήματα της δημιουργίας των σκηνών που υλοποιούν το πείραμα, χρησιμοποιώντας τεχνολογίες εκτεταμένης πραγματικότητας.

5.2 Σελίδα βιβλίου 24

Στο κεφάλαιο «Υλικά σώματα» ανήκει η ενότητα «ΦΕ2: MAZA». Η ενότητα ξεκινάει με ένα εισαγωγικό ερέθισμα και ακολουθεί μια πειραματική αντιμετώπιση στην οποία οι μαθητές καλούνται να διαβάσουν τη μάζα κάθε προϊόντος, η οποία αναγράφεται στην ετικέτα του, διαπιστώνοντας με αυτόν τον τρόπο τη διαφορετική μάζα των προϊόντων. Συνεπώς, οι μαθητές εξοικειώνονται με τις συνήθεις μονάδες μέτρησης της μάζας. Στη συνέχεια, ακολουθεί το πείραμα της σελίδας 24 (σχήμα 3.1). Για το πείραμα αυτό οι μαθητές χρειάζεται να φέρουν στην τάξη τα προϊόντα της σελίδας 23 και το ζυγό, για να συγκρίνουν τη μάζα τους. Μετά, ζητείται από τους μαθητές να βρουν τους συνδυασμούς, ώστε ο ζυγός να ισορροπεί και να τους συμπληρώσουν στις παρατηρήσεις. Η εξαγωγή του συμπεράσματος γίνεται με κατάλληλες ερωτήσεις και συζήτηση στην τάξη, μέσα από την οποία οι μαθητές γενικεύουν τις παρατηρήσεις τους στα πειράματα που προηγήθηκαν και διατυπώνουν το συμπέρασμα.

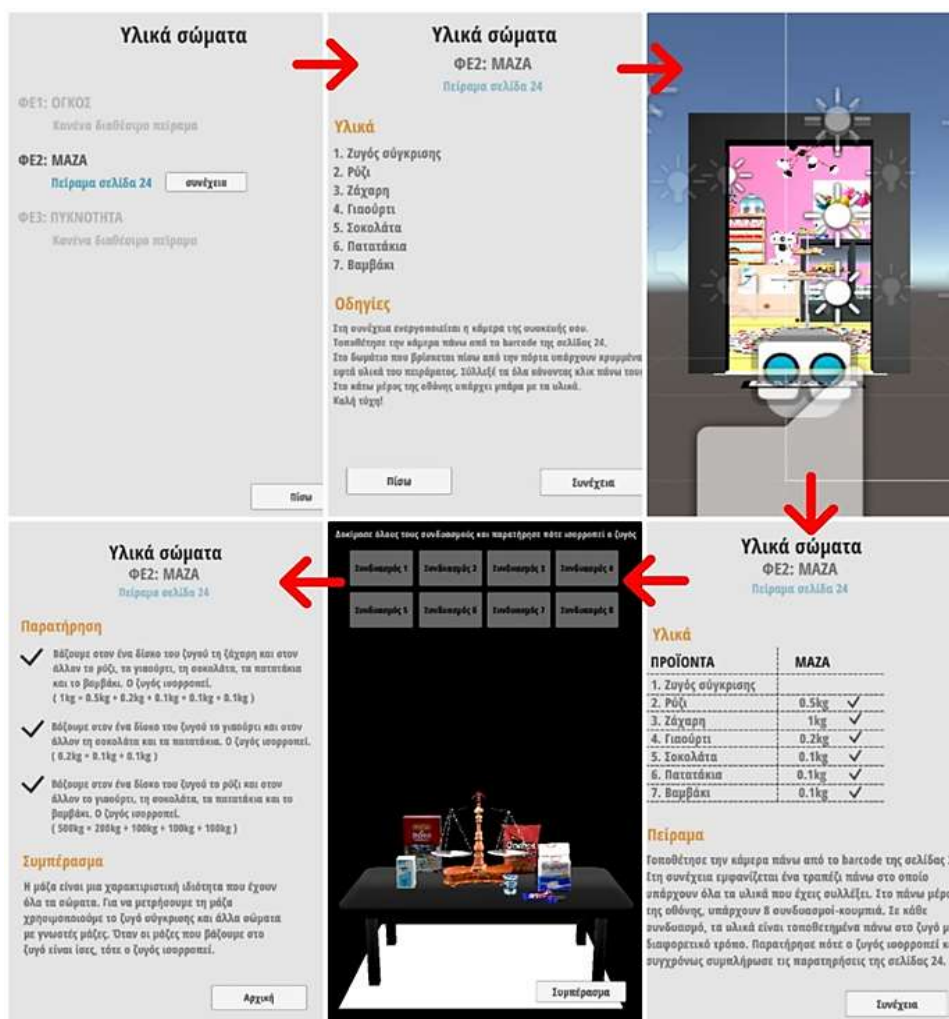


Σχήμα 5.1: Σελίδα βιβλίου 24

Επιπλέον, για την πραγματοποίηση του πειράματος εκτεταμένης πραγματικότητας, έχει προστεθεί στη σελίδα 24 εκτυπωμένο το image target18 από τη βάση high_class.

5.3 Οι σκηνές του πειράματος – Υλικά σώματα

Το πείραμα εκτεταμένης πραγματικότητας αποτελείται από έξι σκηνές όπως αυτές φαίνονται στο σχήμα 3.2. Αρχικά ο μαθητής-παίκτης, αφού επιλέξει το κεφάλαιο των υλικών σωμάτων (σκηνή: startScene), εμφανίζονται τα διαθέσιμα πειράματα των ενοτήτων του κεφαλαίου (σκηνή: Ylikabout). Αφού επιλέξει ποιο πείραμα θέλει να υλοποιηθεί, εμφανίζονται οι οδηγίες του πειράματος καθώς και τα υλικά τα οποία πρέπει να συλλέξει (σκηνή: fe2-maza). Στη συνέχεια, ενεργοποιείται η κάμερα της συσκευής του, και αφού σκανάρει το qrcode της σελίδας 24, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας και μια μπάρα με τα εναπομείναντα υλικά τα οποία βρίσκονται πίσω από αυτή, σε ένα εικονικό περιβάλλον (σκηνή: VR4). Μόλις συλλέξει όλα τα υλικά της μπάρας, εμφανίζονται οδηγίες (σκηνή: testScene4) για το πείραμα που θα ακολουθήσει. Σκανάροντας ξανά το qrcode της σελίδας 24, εμφανίζεται ένα τραπέζι επαυξημένης πραγματικότητας στο οποίο υπάρχουν πάνω όλα τα υλικά του πειράματος καθώς και οκτώ κουμπιά τα οποία αντιστοιχούν σε οκτώ συνδυασμούς προϊόντων (σκηνή: 4.animation). Στο κάτω μέρος υπάρχει το κουμπί “συμπέρασμα” το οποίο οδηγεί τον παίκτη στην επόμενη σκηνή η οποία αποτελείται από τη συμπληρωμένη παρατήρηση του βιβλίου καθώς και του συμπεράσματος (σκηνή: symperasmataYlikaSwmata).



Σχήμα 5.2: Σκηνές πειράματος σελίδας 24

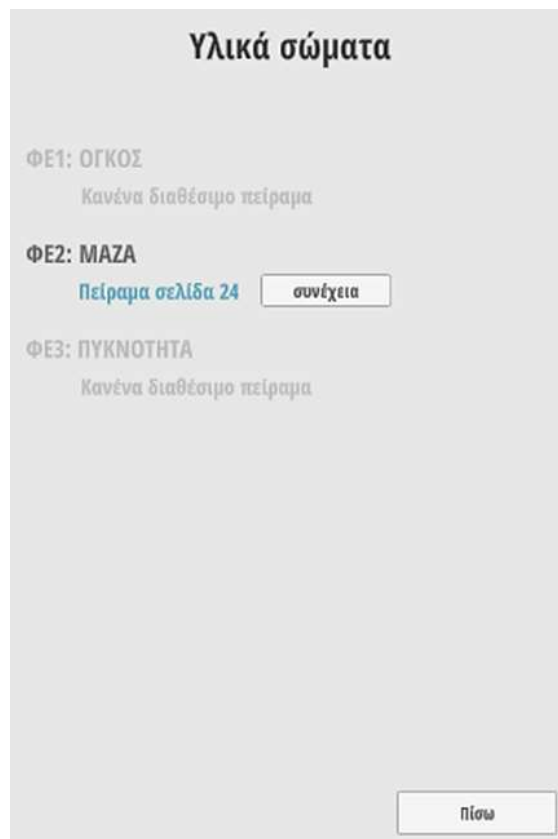
Μετά την ολοκλήρωση του παραπάνω πειράματος εκτεταμένης πραγματικότητας, οι μαθησιακοί στόχοι που είναι θεμιτό να επιτευχθούν είναι οι εξής:

- Να αναφέρουν οι μαθητές ότι η μάζα είναι μια χαρακτηριστική ιδιότητα ενός υλικού σώματος.
- Να μετρήσουν οι μαθητές τη μάζα ενός στερεού σώματος χρησιμοποιώντας ένα ζυγό σύγκρισης.
- Να αναφέρουν οι μαθητές μονάδες μέτρησης μάζας.

5.4 Σκηνή: Ylikabout

Η σκηνή αυτή φαίνεται στο σχήμα 3.3. Η ιεραρχία των αντικειμένων της αποτελείται από:

1. Μια main camera (unity → GameObject → Camera).
2. Ένα directional light (unity → GameObject → Light → Directional Light) για το σωστό φωτισμό της σελίδας.
3. Ένα event system (unity → GameObject → UI → Event system) για τη λειτουργία των κουμπιών και των εναλλαγών των σκηνών.
4. Ένα canva στον οποίο υπάρχουν πολλά textMeshPro για την εισαγωγή των κειμένων καθώς και δύο buttons. Με αυτά γίνεται η εναλλαγή των σκηνών, μέσω ενός script παρόμοιου με του script “menu”, το οποίο αποτελείται από τις μεθόδους “next()” και “back()” που χρησιμοποιούν τη βιβλιοθήκη UnityEngine.SceneManagement. Το script αυτό είναι τοποθετημένο στο inspector του canva. Για τη λειτουργία του κουμπιού της συνέχειας, στο πεδίο OnClick() → Select project = canvas → Select function = next(). Για τη λειτουργία του κουμπιού “πίσω” OnClick() → Select project = canvas → Select function = back().



Σχήμα 5.3: Σκηνή "YlikaAbout"

5.5 Σκηνή: Fe2-maza

Η σκηνή αυτή φαίνεται στο σχήμα 3.4. Η ιεραρχία των αντικειμένων της είναι ίδια με αυτή της σκηνής “ylikaAbout”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro και το περιεχόμενό τους είναι διαφορετικό.



Σχήμα 5.4: Σκηνή fe2-maza

5.6 Σκηνή: VR4

Σε αυτή τη σκηνή, ο παίκτης μετά από προτροπή των οδηγιών της προηγούμενης σκηνής, σκανάρει το qrcode που βρίσκεται στη σελίδα 24 του σχολικού βιβλίου. Μόλις η Vuforia Camera το εντοπίσει, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας πάνω από το qrcode (σχήμα 3.5). Πιο συγκεκριμένα, αυτή η πόρτα είναι ορατή από έξω αλλά το εσωτερικό της δε φαίνεται. Για να φανεί το εικονικό περιβάλλον που υπάρχει πίσω από την πόρτα, ο παίκτης πρέπει να εισέλθει με τη συσκευή του μέσα στην πόρτα. Μόλις γίνει αυτό, ακολουθεί ολική εμβύθιση του παίκτη σε ένα δωμάτιο εικονικής πραγματικότητας, στο οποίο καλείται να βρει τα επτά υλικά του πειράματος. Στο κάτω μέρος της οθόνης υπάρχει μπάρα στην οποία φαίνονται τα υπολειπόμενα υλικά. Τέλος, μόλις συλλεχθούν όλα τα υλικά, με τη χρήση δύο scripts που θα περιγραφούν σε παρακάτω κεφάλαιο, γίνεται η κατάλληλη εναλλαγή σκηνής στην επόμενη. Σε γενικές γραμμές, για κάθε χρονικό βήμα αυτής της σκηνής, η χρήση επαυξημένης πραγματικότητας πρέπει:

1. Να καθορίζει την τρέχουσα κατάσταση του φυσικού κόσμου καθώς και του εικονικού κόσμου.
2. Να εμφανίζει την εικονική πληροφορία με χωρική και χρονική συσχέτιση με τον πραγματικό κόσμο κατά τρόπο που θα επιτρέψει στο χρήστη να αντιληφθεί τα εικονικά στοιχεία ως μέρος του

φυσικού κόσμου και στη συνέχεια να επιστρέψει στο βήμα1, για να προχωρήσει στο επόμενο χρονικό βήμα.

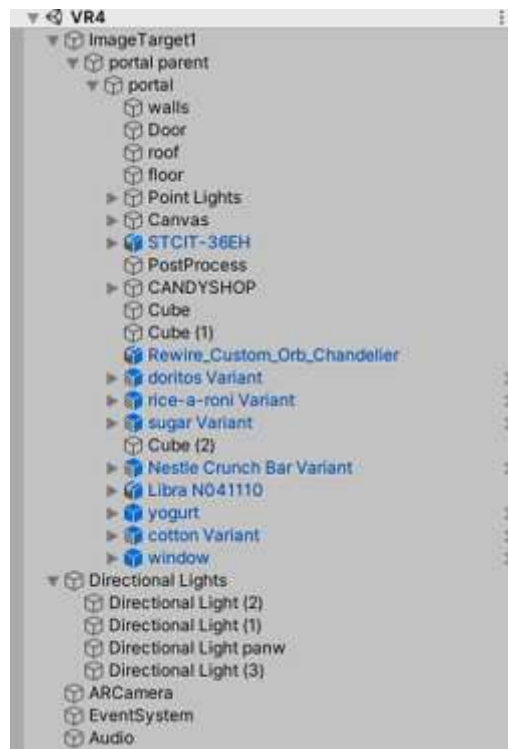


Σχήμα 5.5: Πόρτα επαυξημένης πραγματικότητας VR4 σκηνής

5.6.1 Ιεραρχία αντικειμένων VR4

Η ιεραρχία των αντικειμένων της σκηνής VR4 φαίνεται αναλυτικά στο σχήμα 3.6. Περιληπτικά, αποτελείται από:

- Πέντα *directional lights* για σωστό φωτισμό περιμετρικά της πόρτας και δεκαπέντε *point lights* εσωτερικά.
- Ένα *event system*.
- Ένα *empty* μετονομασμένο ως *audio* για τις ρυθμίσεις του ήχου.
- Μία *ARCamera*.
- Ένα *image target* το οποίο εμπεριέχει όλα τα αντικείμενα του εικονικού περιβάλλοντος όπως:
 - Τα εσωτερικά *point lights*.
 - Τη δομή του δωματίου (*walls, door, roof, floor*).
 - Έναν *canva* με όλα τα κουμπιά και τα *textmesh Pro* της μπάρας.
 - Τα εικονικά αντικείμενα που πρέπει να συλλεχθούν.
 - Τα εικονικά αντικείμενα που υπάρχουν και αποτελούν το ύφος του δωματίου.



Σχήμα 5.6: Ιεραρχία αντικειμένων VR4

5.6.2 AR πόρτα – VR4

Πριν τη δημιουργία της πόρτας, έγινε δημιουργία αντικειμένου “image target” και η επιλογή του marker18 από τη βάση highclass. Το παιδί του image target είναι ένα empty “portal parent” προκειμένου τα περιεχόμενα αυτού να εμφανιστούν εφόσον η arcamera εντοπίσει το imagemark18. Τα δομικά στοιχεία του δωματίου είναι οι τοίχοι, η πόρτα, το πάτωμα και η οροφή. Αυτά δημιουργήθηκαν με τη χρήση του Probuilder. Η πόρτα έχει ως material=black ενώ τα υπόλοιπα δομικά στοιχεία έχουν ως material του εξωτερικού τους μέρους το material9, το οποίο είναι ένα custom mask με render queue = geometry -1, το οποίο δίνει την ψευδαίσθηση του άορατου. Στις ρυθμίσεις του image target → inspector → image Target Behaviour → advanced → device tracker settings → Device tracker → Device tracker Pose = ON.

5.6.3 Εικονικό περιβάλλον VR4

Το εσωτερικό του δωματίου μοιάζει με ζαχαροπλαστείο. Αποτελείται από δεκάδες assets, .fbx και .obj που αποθηκεύτηκαν από unity asset store και από ιστοσελίδες όπως:

- free3d [15],
- sketchfab [16],
- turbosquid [17],
- poly [18], και άλλες,

Εφαρμογή πειράματος για το κεφάλαιο Υλικά σώματα

αλλά συγχρόνως και από αντικείμενα που δημιουργήθηκαν στο blender. Στο σχήμα 3.7 φαίνεται το εσωτερικό του δωματίου με λεπτομέρειες.



Σχήμα 5.7: Δωμάτιο VR4

5.6.4 Εικονικά αντικείμενα VR4

Τα επτά υλικά που ο παίκτης πρέπει να συλλέξει προκειμένου να προχωρήσει στο πείραμα φαίνονται στο σχήμα 3.8. Πέρα από το ζυγό, όλα τα υπόλοιπα είναι απλά gameobjects με materials → Legacy shaders → diffuse με επιλεγμένες φωτογραφίες πραγματικών προϊόντων. Στη συσκευασία των προϊόντων αναγράφεται η μάζα τους σύμφωνα με τον πίνακα του σχολικού βιβλίου.



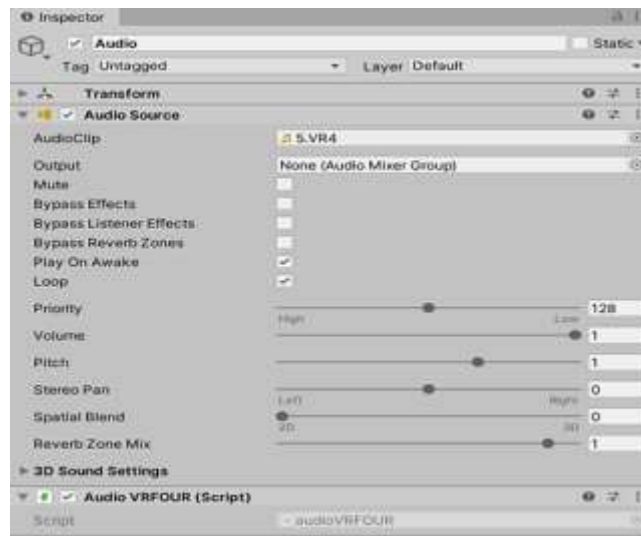
Σχήμα 5.8: Υλικά πειράματος (Υλικά σώματα)

5.6.5 Μουσική VR4

Στα components του empty “Audio” προστέθηκε ένα “Audio source” με τις εξής ρυθμίσεις:

- Στο πεδίο “AudioClip”, drag and drop το αρχείο .mp3 του ήχου της σκηνής VR4 (ένα και μοναδικό σε όλο το παιχνίδι).
- Play On Awake = ON, για την εκκίνηση της μουσικής με το που ανοίγει η σκηνή.
- Loop = ON, διότι το αρχείο ήχου έχει διάρκεια περίπου δύο λεπτά και αν ο χρήστης παραμείνει στη σκηνή πάνω από δύο λεπτά ο ήχος να συνεχίσει να παίζει.

Ένα ακόμα component που προστέθηκε στο audio είναι το script “audioVRFOUR”. Ολοκληρωμένο το inspector του audio φαίνεται στο σχήμα 3.9.



Σχήμα 5.9: Ρυθμίσεις ήχου VR4

Το script “audioVRFOUR” (σχήμα 3.10) είναι παρόμοιο με το script ήχου της αρχικής σκηνής, με τη διαφορά ότι καταστρέφει το αντικείμενο του ήχου στη σκηνή testscene4, δηλαδή την επόμενη της VR4.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class audioVRFOUR : MonoBehaviour
7  {
8      // Start is called before the first frame update
9      void Start()
10     {
11     }
12
13     private static audioVRFOUR instance = null;
14     public static audioVRFOUR Instance
15     {
16         get { return instance; }
17     }
18
19     void Awake()
20     {
21         if (Instance != null && Instance != this)
22         {
23             Destroy(this.gameObject);
24             return;
25         }
26         else
27         {
28             Instance = this;
29         }
30         DontDestroyOnLoad(this.gameObject);
31     }
32
33     void Update()
34     {
35         //for every scene I want to stop the music, duplicate this if you change the name of a scene
36         if (SceneManager.GetActiveScene().name == "test-scene4")
37         {
38             Destroy(this.gameObject);
39         }
40     }
41 }

```

Σχήμα 5.10: Script ήχου "audioVRFOUR"

5.6.6 Εναλλαγή σκηνής VR4

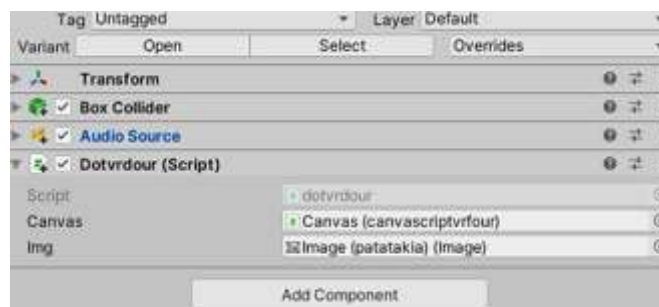
Στα επτά αντικείμενα που πρέπει να συλλέξει ο παίκτης έχουν προστεθεί τα components:

- Box collider: για να οριοθετηθούν τα αντικείμενα και να δουλεύει η συνάρτηση OnMouseDown.
- Audio source: όπου audioclip=αρχείο ήχου “win” (το ντιν που ακούγεται μόλις επιλέγεται), play on awake = off γιατί ο ήχος πρέπει να ακούγεται μόνο στη συνάρτηση OnMouseDown και loop=off.
- Το script “Dotvrndour” (σχήμα 3.11) και στα πεδία “Canva”= τον canva και “Img”=το αρχείο εικόνας tik που αντιστοιχεί στα textmesh Pro κάθε αντικειμένου στην μπάρα.

```
dotvrndour.cs X
C:\> Users > user > Desktop > ptyxiaki > projects > final project > karantina > Assets >
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class dotvrndour : MonoBehaviour
7 {
8     public canvasscriptvrfour canvas;
9     public Image img;
10
11     void Start()
12     {
13         img.enabled=false;
14     }
15
16     void OnMouseDown()
17     {
18         canvas.sum = canvas.sum +1;
19         //Debug.Log("something touched");
20         img.enabled=true;
21         gameObject.GetComponent<AudioSource>().Play();
22         Destroy(gameObject, 1);
23     }
24 }
25
```

Σχήμα 5.11: script “Dotvrndour”

Η μπάρα στο κάτω μέρος του παιχνιδιού με τα υπολειπόμενα παιχνίδια είναι ο canvas από την ιεραρχία αντικειμένων και αποτελείται από επτά textmesh Pro με τα ονόματα των αντικειμένων και αντίστοιχα επτά εικόνες (tik) που εμφανίζονται μόλις ο παίκτης επιλέξει ένα αντικείμενο από αυτά. Στη γραμμή 8 του κώδικα δημιουργείται ένα αντικείμενο τύπου canva με όνομα “canvasscriptvrfour”. Παρομοίως, στη γραμμή 7 του κώδικα δημιουργείται ένα αντικείμενο image με όνομα “img”. Με αυτό τον τρόπο ανοίγει τα πεδία “canvasscriptvrfour” και “img” στα components Dotvrndour κάθε αντικειμένου όπως φαίνεται στο σχήμα 3.12. Στο πεδίο canvasscriptvrfour γίνεται drag and drop ο canvas (ο οποίος περιέχει άλλο script που αναλύεται στην πορεία) και στο πεδίο img γίνεται drag and drop η αντίστοιχη εικόνα tik κάθε αντικειμένου.



Σχήμα 5.12: Components αντικειμένων VR4

Στη γραμμή 11 του κώδικα ξεκινάει η μέθοδος start() η οποία εκτελεί τα περιεχόμενά της με το που ανοίξει η σκηνή. Στη γραμμή 12 του κώδικα απενεργοποιείται προσωρινά η εικόνα «τικ» των αντικειμένων της μπάρας μέχρι όταν γίνει η επιλογή του αντικείμενου στη μέθοδο onMouseDown και ενεργοποιηθεί στη γραμμή 20. Στη γραμμή 21 του κώδικα ενεργοποιείται το αρχείο ήχου που είναι στα components του αντικείμενου που έγινε onMouseDown. Στη γραμμή 22 του κώδικα καταστρέφεται ολικά το αντικείμενο για να μην εμφανίζεται μετά την επιλογή του στο χώρο. Στη γραμμή 18 του κώδικα υπάρχει ο μετρητής sum ο οποίος περνάει παραμετρικά την τιμή του στο script “canvascriptvrfour” (σχήμα 3.13). Το “canvascriptvrfour” script με τη σειρά του, όταν αυτός ο μετρητής φτάσει τον αριθμό 7 (δηλαδή όταν όλα τα αντικείμενα βρεθούν), καλεί την επόμενη σκηνή μέσω της βιβλιοθήκης UnityEngine.SceneManagement.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6
7  public class canvascriptvrfour : MonoBehaviour
8  {
9      public int sum;
10     // Start is called before the first frame update
11     void Start()
12     {
13         sum = 0;
14     }
15
16     // Update is called once per frame
17     void Update()
18     {
19         if (sum == 7) {
20             SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
21         }
22     }
23 }

```

Σχήμα 5.13: Το “canvascriptvrfour” script

5.7 Σκηνή: TestScene4

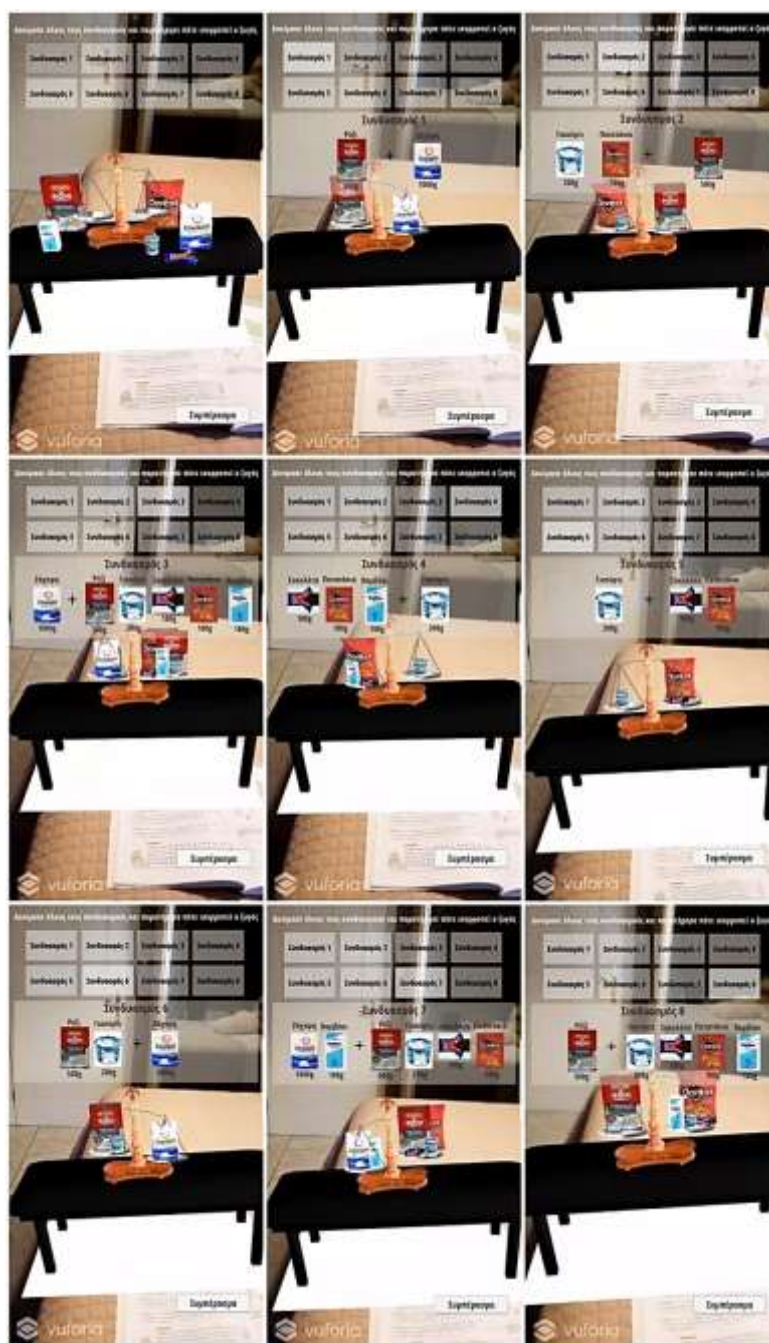
Η σκηνή αυτή φαίνεται στο σχήμα 3.14. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “ylikaAbout” και “fe2-maza”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται οδηγίες για το πείραμα που ακολουθεί.



Σχήμα 5.14: Σκηνή testscene4

5.8 Σκηνή: 4.animation

Σε αυτή τη σκηνή πραγματοποιείται το πείραμα της σελίδας με τη χρήση εκτεταμένης πραγματικότητας. Στην αρχική οθόνη υπάρχει ένα τραπέζι πάνω στο οποίο είναι τοποθετημένα τα επτά υλικά του πειράματος. Στο πάνω μέρος της οθόνης υπάρχουν οκτώ κουμπιά τα οποία αντιστοιχούν σε οκτώ συνδυασμούς προϊόντων. Ανάλογα με την αθροιστική μάζα των προϊόντων που βρίσκονται στο αριστερό και δεξί μέρος του ζυγού, αυτός είναι τοποθετημένος κατάλληλα. Ο παίκτης πρέπει να βρει ποιοι τρεις συνδυασμοί προϊόντων ισορροπούν το ζυγό και να τους συμπληρώσει στο βιβλίο. Μόλις πραγματοποιηθεί αυτό, ο παίκτης μπορεί πατώντας το κουμπί «Συμπέρασμα» να προχωρήσει στην επόμενη σκηνή. Στο σχήμα 3.15 φαίνεται η αρχική οθόνη της σκηνής καθώς και η οθόνη μετά από την επιλογή όλων των συνδυασμών.



Σχήμα 5.15: Οι οθόνες της σκηνής “4.animation”

5.8.1 Ιεραρχία αντικειμένων – 4.animation

Η ιεραρχία των αντικειμένων της σκηνής 4.animation αποτελείται από:

- Ένα event system.
- Μία ARCamera.
- Πέντε directional lights για σωστό φωτισμό περιμετρικά της πόρτας και δεκαπέντε point lights εσωτερικά.
- Ένα image target το οποίο εμπεριέχει :
 - Ground: ένα έδαφος που καλύπτει το image target.
 - Table: τα τραπέζι που πατάει πάνω στο ground και πάνω του πραγματοποιείται το πείραμα
 - Οκτώ empties (απενεργοποιημένα): τα οποία αποτελούνται από τα υλικά των συνδυασμών χωριστά.
 - Ένα empty (ενεργοποιημένο): το οποίο αποτελεί την αρχική οθόνη της σκηνής και περιέχει όλα τα υλικά.
- Ένα canvas ο οποίος εμπεριέχει :
 - Οκτώ textmesh Pro buttons (ενεργοποιημένα) τα οποία αντιστοιχούν στα οκτώ κουμπιά των συνδυασμών.
 - Οκτώ textmesh Pro buttons (απενεργοποιημένα) τα οποία αντιστοιχούν στα οκτώ κουμπιά των συνδυασμών και ενεργοποιούνται στη συνάρτηση “onClick” για να δίνουν την αίσθηση ότι είναι πατημένα.
 - Οκτώ images οι οποίες απεικονίζουν λεπτομερώς του συνδυασμούς των προϊόντων καθώς και τις μάζες τους. Αυτές ενεργοποιούνται με το πάτημα του κουμπιού του αντίστοιχου συνδυασμού.
 - Ένα button που αντιστοιχεί στο «συμπέρασμα» και προχωράει στην επόμενη σκηνή.
- Οκτώ empties μετονομασμένα ως “basic script” ακολουθούμενα από τον αριθμό 1-8, ανάλογα με τον συνδυασμό στον οποίο χρησιμοποιούνται. Στη συνέχεια αναλύεται η χρησιμότητά τους.

5.8.2 Script “basicvrFOUR”

Το script “basicvrFOUR” καθορίζει όλη τη λειτουργία του πειράματος. Ο κώδικας φαίνεται στο σχήμα 3.16. Αναλυτικά,

- Στις τέσσερις πρώτες γραμμές χρησιμοποιεί τις βιβλιοθήκες System.Collections, System.Collections.Generic, UnityEngine και UnityEngine.UI.
- Στη γραμμή 9 του κώδικα, δημιουργείται ένα πεδίο “enableEmpty” τύπου GameObject, το οποίο καθορίζεται να πάρει το αντικείμενο που αντιστοιχεί στο empty που πρέπει να ενεργοποιηθεί με το πάτημα του κουμπιού του συνδυασμού (γραμμή 42 του κώδικα) και να εμφανίσει τα κατάλληλα υλικά πάνω στο ζυγό.
- Στη γραμμή 10 του κώδικα, δημιουργείται ένα πεδίο “ disableEmptyArxiki” τύπου GameObject, το οποίο θα πάρει το empty1, δηλαδή αυτό που αντιστοιχεί στην αρχική οθόνη, προκειμένου να απενεργοποιηθεί με το πάτημα ενός κουμπιού συνδυασμού στη γραμμή 43 του κώδικα.

- Στις γραμμές 12-18 δημιουργούνται επτά πεδία “ disableEmpty1-7” τύπου GameObject, τα οποία πρέπει να πάρουν τα υπόλοιπα επτά empties τα οποία πρέπει να απενεργοποιηθούν (γραμμές 45-51 του κώδικα) με το πάτημα του κουμπιού ενός συνδυασμού.
- Στη γραμμή 20 του κώδικα, δημιουργείται ένα πεδίο “ enableCanvaBut ” τύπου GameObject, το οποίο βρίσκεται στη θέση του κουμπιού που είναι πατημένο και ενεργοποιείται (γραμμή 53 του κώδικα) δίνοντας την αίσθηση ότι είναι πατημένο.
- Στις γραμμές 22-28 δημιουργούνται επτά πεδία “ disCanvaBut 1-7” τύπου GameObject, τα οποία αντιστοιχούν στα υπόλοιπα επτά κουμπιά συνδυασμών που δεν πρέπει να φαίνονται πατημένα και για αυτό απενεργοποιούνται αντίστοιχα στις γραμμές του κώδικα 55-61.
- Στη γραμμή 30 του κώδικα, δημιουργείται ένα πεδίο “enableImg” τύπου GameObject, το οποίο πρέπει να πάρει την εικόνα του συνδυασμού ο οποίος επιλέχθηκε για να ενεργοποιηθεί στη γραμμή 63.
- Στις γραμμές 32-38 δημιουργούνται επτά πεδία “ disableImg 1-7” τύπου GameObject, τα οποία αντιστοιχούν στις υπόλοιπες επτά εικόνες συνδυασμών οι οποίες πρέπει να παραμείνουν απενεργοποιημένες (γραμμές 65-71 του κώδικα).

Στα οκτώ empties της ιεραρχίας έγινε drag and drop στα components το script. Οι εντολές που βρίσκονται έξω από την settings(), δημιουργούν πεδία τα οποία υπάρχουν στα components των empties1-8 και στη συνέχεια παραμετροποιούνται ανάλογα με τον συνδυασμό. Οι εντολές που βρίσκονται μέσα στην settings(), ενεργοποιούν ή απενεργοποιούν αυτά τα αντικείμενα που βρίσκονται στα πεδία που δημιουργήθηκαν.

```

C:\Users\user\Documents\projects\3rd project\Assets\Scripts\Animation -> basicvrFOUR.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class basicvrFOUR : MonoBehaviour
7  {
8
9      public GameObject disableEmpty1; //το πρώτο που θέλω να απενεργοώσω
10     public GameObject disableEmpty2; //το δεύτερο που
11
12     public GameObject disableEmpty3; //τα υπόλοιπα empties που από την η θέση η ενεργοώσω
13     public GameObject disableEmpty4;
14     public GameObject disableEmpty5;
15     public GameObject disableEmpty6;
16     public GameObject disableEmpty7;
17
18     public GameObject enableCanvaBut1; //enable to button in this a button position
19
20     public GameObject disCanvaBut1; //disable to button in this a button position
21     public GameObject disCanvaBut2;
22     public GameObject disCanvaBut3;
23     public GameObject disCanvaBut4;
24     public GameObject disCanvaBut5;
25     public GameObject disCanvaBut6;
26     public GameObject disCanvaBut7;
27
28     public GameObject enableImg; //image to activate a button animation
29
30     public GameObject disableImg1; //to activate a image in this a button position
31     public GameObject disableImg2;
32     public GameObject disableImg3;
33     public GameObject disableImg4;
34     public GameObject disableImg5;
35     public GameObject disableImg6;
36     public GameObject disableImg7;
37
38     public void settings()
39     {
40         enableEmpty1.SetActive(true);
41         disableEmpty1.SetActive(false);
42
43         disableEmpty2.SetActive(false);
44         disableEmpty3.SetActive(false);
45         disableEmpty4.SetActive(false);
46         disableEmpty5.SetActive(false);
47         disableEmpty6.SetActive(false);
48         disableEmpty7.SetActive(false);
49
50         enableCanvaBut1.SetActive(true);
51
52         disCanvaBut1.SetActive(false);
53         disCanvaBut2.SetActive(false);
54         disCanvaBut3.SetActive(false);
55         disCanvaBut4.SetActive(false);
56         disCanvaBut5.SetActive(false);
57         disCanvaBut6.SetActive(false);
58         disCanvaBut7.SetActive(false);
59
60         enableImg.SetActive(true);
61
62         disableImg1.SetActive(false);
63         disableImg2.SetActive(false);
64         disableImg3.SetActive(false);
65         disableImg4.SetActive(false);
66         disableImg5.SetActive(false);
67         disableImg6.SetActive(false);
68         disableImg7.SetActive(false);
69
70
71
    }

```

Σχήμα 5.16: Το script “basicvrFOUR”

Κεφάλαιο 5

Αφού το παραπάνω script έχει μπει στα components των οκτώ empties “basicscript1-8”, ακολουθεί η συμπλήρωση των πεδίων με τα κατάλληλα αντικείμενα. Στο σχήμα 3.17 φαίνονται τα πεδία του basicscript1 συμπληρωμένα. Με αντίστοιχο τρόπο συμπληρώθηκαν και τα πεδία των υπόλοιπων επτά empties-basicscripts.



Σχήμα 5.17: Συμπληρωμένα πεδία component basicvrFOUR του basicscript1

Μόλις τα basicscripts είναι έτοιμα ακολουθεί η προσθήκη τους στα αντίστοιχα κουμπιά των συνδυασμών. Σε κάθε ένα από τα οκτώ buttons των συνδυασμών έγινε drag and drop το αντίστοιχο basicscript (στο button1 του συνδυασμού 1 → drag and drop to basicscript1), στο πεδίο “Select object” του button → OnClick(). Στο πεδίο select function επιλέχθηκε για όλα τα κουμπιά η συνάρτηση basicvrFOUR.settings(). Η οθόνη μετά την επιλογή του κουμπιού «συνδυασμός1» φαίνεται στο σχήμα 3.18. Παρόμοιες είναι οι οθόνες των υπόλοιπων επτά κουμπιών-συνδυασμών.



Σχήμα 5.18: Οθόνη με την επιλογή του κουμπιού «συνδυασμός1»

Στο κουμπί «Συμπέρασμα» στα components έχει προστεθεί το script “symper” το οποίο φαίνεται στο σχήμα 3.19. Στο onclick() event: Select object = το button «Συμπέρασμα» → select function = GoToSymperasma(). Το script “symper” χρησιμοποιεί τη βιβλιοθήκη UnityEngine.SceneManagement για να πάει στην επόμενη σκηνή (γραμμή 8 του κώδικα).

```
symper.cs X
C: > Users > user > Desktop > ptychiaki > projects > final project > karantina > Assets > Scripts > 4.animation > symper.cs
1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3
4  public class symper : MonoBehaviour
5  {
6      public void GotoSymperasma()
7      {
8          SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +1);
9      }
10 }
```

Σχήμα 5.19: Το script “symper”

5.9 Σκηνή: symperasmataYlikaSwmata

Η σκηνή αυτή φαίνεται στο σχήμα 3.20. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “ylikaAbout”, “fe2-maza” και “testscene4”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται η παρατήρηση και τα συμπεράσματα που πρέπει να συμπληρώσει ο μαθητής στο βιβλίο.

Υλικά σώματα

ΦΕ2: MAZA

Πείραμα σελίδα 24

Παρατήρηση

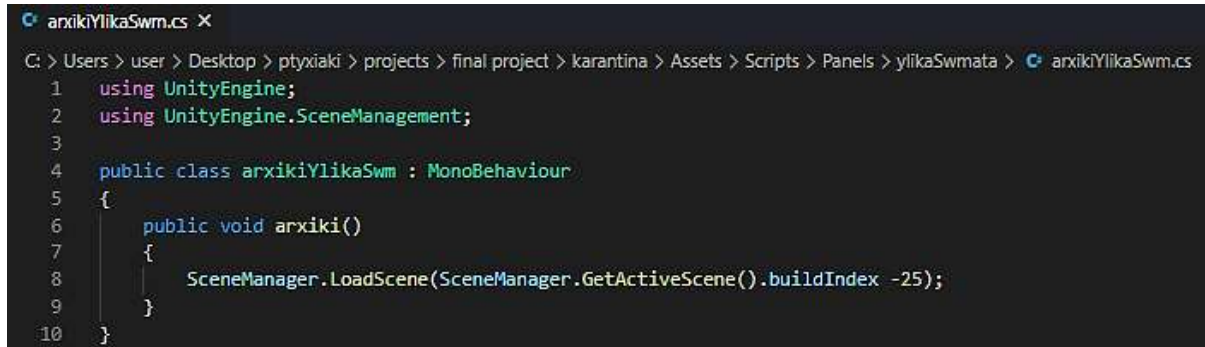
- ✓ Βάζουμε στον ένα δίσκο του ζυγού τη ζάχαρη και στον άλλον το ρύζι, το γιαούρτι, τη σοκολάτα, τα πατατάκια και το βαμβάκι. Ο ζυγός ισορροπεί.
($1\text{kg} = 0.5\text{kg} + 0.2\text{kg} + 0.1\text{kg} + 0.1\text{kg} + 0.1\text{kg}$)
- ✓ Βάζουμε στον ένα δίσκο του ζυγού το γιαούρτι και στον άλλον τη σοκολάτα και τα πατατάκια. Ο ζυγός ισορροπεί.
($0.2\text{kg} = 0.1\text{kg} + 0.1\text{kg}$)
- ✓ Βάζουμε στον ένα δίσκο του ζυγού το ρύζι και στον άλλον το γιαούρτι, τη σοκολάτα, τα πατατάκια και το βαμβάκι. Ο ζυγός ισορροπεί.
($500\text{kg} = 200\text{kg} + 100\text{kg} + 100\text{kg} + 100\text{kg}$)

Συμπέρασμα

Η μάζα είναι μια χαρακτηριστική ιδιότητα που έχουν όλα τα σώματα. Για να μετρήσουμε τη μάζα χρησιμοποιούμε το ζυγό σύγκρισης και άλλα σώματα με γνωστές μάζες. Όταν οι μάζες που βάζουμε στο ζυγό είναι ίσες, τότε ο ζυγός ισορροπεί.

Σχήμα 5.20: Σκηνή-symperasmataYlikaSwmata

Στο κουμπί «Αρχική» στα components έχει προστεθεί το script “ArxikiYlikaSwmatwn” το οποίο φαίνεται στο σχήμα 3.21. Στο onclick() event: Select object = το button «Αρχική» → select function = arxiki(). Το script “ArxikiYlikaSwmatwn” χρησιμοποιεί τη βιβλιοθήκη UnityEngine.SceneManagement για να πάει στην αρχική σκηνή (γραμμή 8 του κώδικα) η οποία βρίσκεται είκοσι πέντε σκηνές πίσω (βλέπε σχήμα 1.10).



```
ArxikiYlikaSwmatwn.cs x
C:\> Users > user > Desktop > ptyxiaki > projects > final project > karantina > Assets > Scripts > Panels > ylikaSwmata > ArxikiYlikaSwmatwn.cs
1 using UnityEngine;
2 using UnityEngine.SceneManagement;
3
4 public class ArxikiYlikaSwmatwn : MonoBehaviour
5 {
6     public void arxiki()
7     {
8         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex -25);
9     }
10 }
```

Σχήμα 5.21: Το script “ArxikiYlikaSwmatwn”

5.10 Επίλογος

Το πείραμα της σελίδας 24 του σχολικού βιβλίου, υλοποιείται με τη χρήση εκτεταμένης πραγματικότητας μέσω των παραπάνω έξι σκηνών. Οι παραπάνω έξι σκηνές είναι απαραίτητες για την ομαλή ροή του πειράματος. Μετά το πέρας της τελευταίας σκηνής του πειράματος, είναι θεμιτό να έχουν επιτευχθεί οι μαθησιακοί στόχοι που αναφέρθηκαν στο κεφάλαιο 3.3.

Κεφάλαιο 6ο: Εφαρμογή πειράματος για το κεφάλαιο Ενέργεια

6.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται οι μαθησιακοί στόχοι του πειράματος της σελίδας 46 του κεφαλαίου «Ενέργεια» του σχολικού βιβλίου. Επιπλέον, αναλύονται όλα τα βήματα της δημιουργίας των σκηνών που υλοποιούν το πείραμα, χρησιμοποιώντας τεχνολογίες εκτεταμένης πραγματικότητας.

6.2 Σελίδα βιβλίου 46

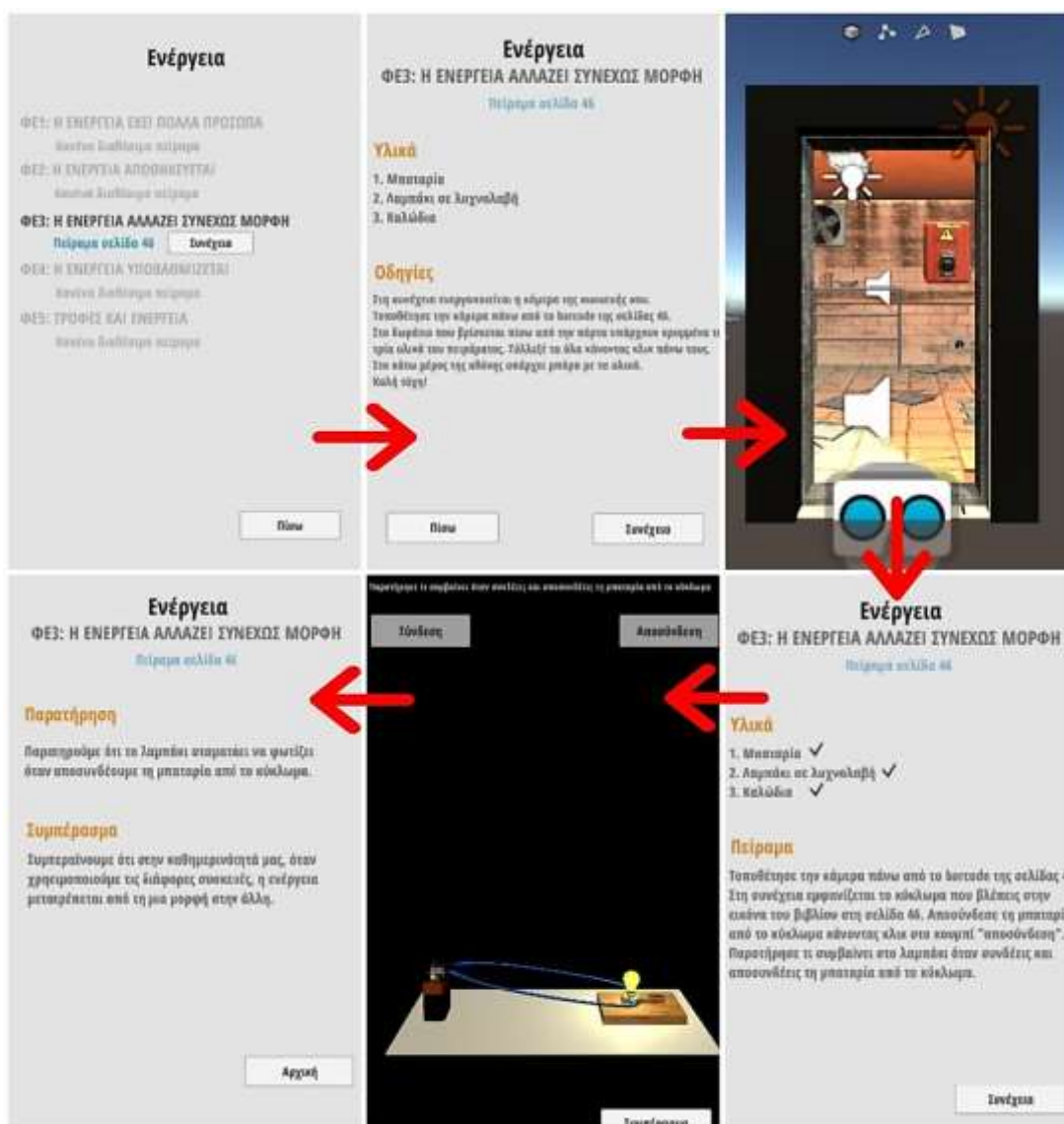
Στο κεφάλαιο «Ενέργεια» ανήκει η ενότητα «ΦΕ3: Η ΕΝΕΡΓΕΙΑ ΑΛΛΑΖΕΙ ΣΥΝΕΧΩΣ ΜΟΡΦΗ». Η ενότητα ξεκινάει με ένα εισαγωγικό ερέθισμα και ακολουθεί μια πειραματική αντιμετώπιση στην οποία οι μαθητές διαπιστώνουν τη μετατροπή της αποθηκευμένης δυναμικής ενέργειας του συμπιεσμένου ελατηρίου σε κινητική ενέργεια. Στη συνέχεια, ακολουθεί το πείραμα της σελίδας 46 (σχήμα 4.1). Στο πείραμα αυτό οι μαθητές κατασκευάζουν ένα απλό ηλεκτρικό κύκλωμα με μια μπαταρία και ένα λαμπάκι σε λυχνιολαβή. Στη συνέχεια, παρατηρούν ότι το λαμπάκι φωτίζει, όσο μετατρέπεται η αποθηκευμένη χημική ενέργεια της μπαταρίας σε ηλεκτρική. Όταν αποσυνδεθεί η μπαταρία από το κύκλωμα, το λαμπάκι παύει να φωτίζει και σταματά η μετατροπή της ενέργειας από τη μια μορφή στην άλλη. Η εξαγωγή του συμπεράσματος γίνεται με κατάλληλες ερωτήσεις και συζήτηση στην τάξη, μέσα από την οποία οι μαθητές γενικεύουν τις παρατηρήσεις τους στα πειράματα που προηγήθηκαν και διατυπώνουν το συμπέρασμα. Επιπλέον, για την πραγματοποίηση του πειράματος εκτεταμένης πραγματικότητας, έχει προστεθεί στη σελίδα 24 εκτυπωμένο το image target20 από τη βάση high_class.



Σχήμα 6.1: Σελίδα βιβλίου 46

6.3 Οι σκηνές του πειράματος – Ενέργεια

Το πείραμα εκτεταμένης πραγματικότητας αποτελείται από έξι σκηνές όπως αυτές φαίνονται στο σχήμα 4.2. Αρχικά ο μαθητής-παίκτης, αφού επιλέξει το κεφάλαιο της ενέργειας (σκηνή: startScene), εμφανίζονται τα διαθέσιμα πειράματα των ενοτήτων του κεφαλαίου (σκηνή: EnergeiaAbout). Αφού επιλέξει ποιο πείραμα θέλει να υλοποιηθεί, εμφανίζονται οι οδηγίες του πειράματος καθώς και τα υλικά τα οποία πρέπει να συλλέξει (σκηνή: fe3-energeia). Στη συνέχεια, ενεργοποιείται η κάμερα της συσκευής του, και αφού σκανάρει το qrcode της σελίδας 46, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας και μια μπάρα με τα εναπομείναντα υλικά τα οποία βρίσκονται πίσω από αυτή, σε ένα εικονικό περιβάλλον (σκηνή: VR6). Μόλις συλλέξει όλα τα υλικά της μπάρας, εμφανίζονται οδηγίες (σκηνή: testScene6) για το πείραμα που θα ακολουθήσει. Σκανάροντας ξανά το qrcode της σελίδας 46, εμφανίζεται το κύκλωμα της εικόνας του σχολικού βιβλίου καθώς και δύο κουμπιά τα οποία αντιστοιχούν σε δύο καταστάσεις “Σύνδεση” και “Αποσύνδεση” (σκηνή: 6.animation). Στο κάτω μέρος υπάρχει το κουμπί “συμπέρασμα” το οποίο οδηγεί τον παίκτη στην επόμενη σκηνή η οποία αποτελείται από τη συμπληρωμένη παρατήρηση του βιβλίου καθώς και του συμπεράσματος (σκηνή: symperasmataEnergeia).



Σχήμα 6.2: Σκηνές πειράματος σελίδας 46

Μετά την ολοκλήρωση του παραπάνω πειράματος εκτεταμένης πραγματικότητας, οι μαθησιακοί στόχοι που είναι θεμιτό να επιτευχθούν είναι οι εξής:

- Να εξηγήσουν οι μαθητές πώς η ενέργεια μετατρέπεται από μια μορφή σε μια άλλη.
- Να καταγράψουν οι μαθητές παραδείγματα ενεργειακών μετατροπών

6.4 Σκηνή: EnergeiaAbout

Η σκηνή αυτή φαίνεται στο σχήμα 4.3. Η ιεραρχία των αντικειμένων της αποτελείται από:

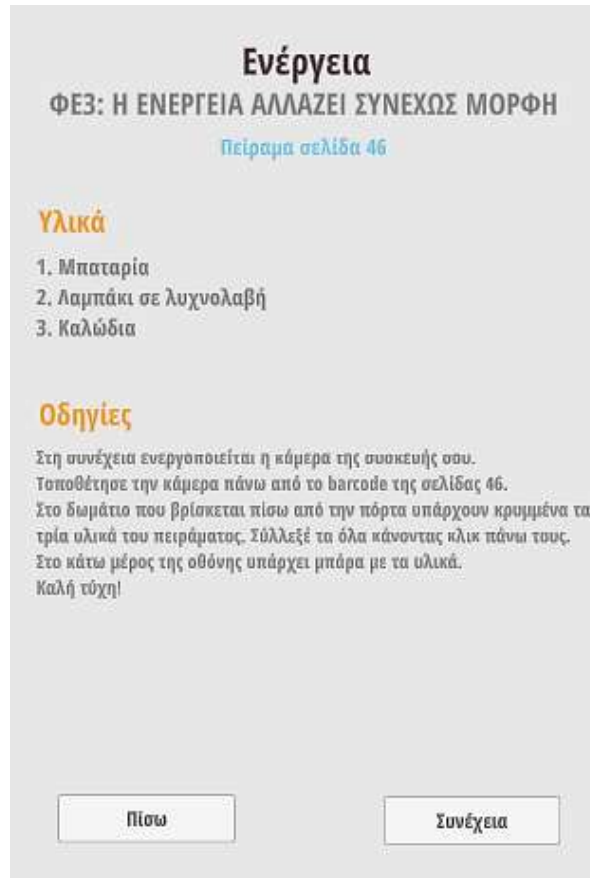
1. Μια main camera (unity → GameObject → Camera).
2. Ένα directional light (unity → GameObject → Light → Directional Light) για το σωστό φωτισμό της σελίδας.
3. Ένα event system (unity → GameObject → UI → Event system) για τη λειτουργία των κουμπιών και των εναλλαγών των σκηνών.
4. Ένα canva στον οποίο υπάρχουν πολλά textMeshPro για την εισαγωγή των κειμένων καθώς και δύο buttons. Με αυτά γίνεται η εναλλαγή των σκηνών, μέσω ενός script παρόμοιου με του script “menu”, το οποίο αποτελείται από τις μεθόδους “next()” και “back()” που χρησιμοποιούν τη βιβλιοθήκη UnityEngine.SceneManagement. Το script αυτό είναι τοποθετημένο στο inspector του canva. Για τη λειτουργία του κουμπιού της συνέχειας, στο πεδίο OnClick() → Select project = canvas → Select function = next(). Για τη λειτουργία του κουμπιού “πίσω” OnClick() → Select project = canvas → Select function = back().



Σχήμα 6.3: Σκηνή “EnergeiaAbout”

6.5 Σκηνή: Fe3-energeia

Η σκηνή αυτή φαίνεται στο σχήμα 4.4. Η ιεραρχία των αντικειμένων της είναι ίδια με αυτή της σκηνής “EnergeiaAbout”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro και το περιεχόμενό τους είναι διαφορετικό.



Σχήμα 6.4: Σκηνή fe3-energeia

6.6 Σκηνή: VR6

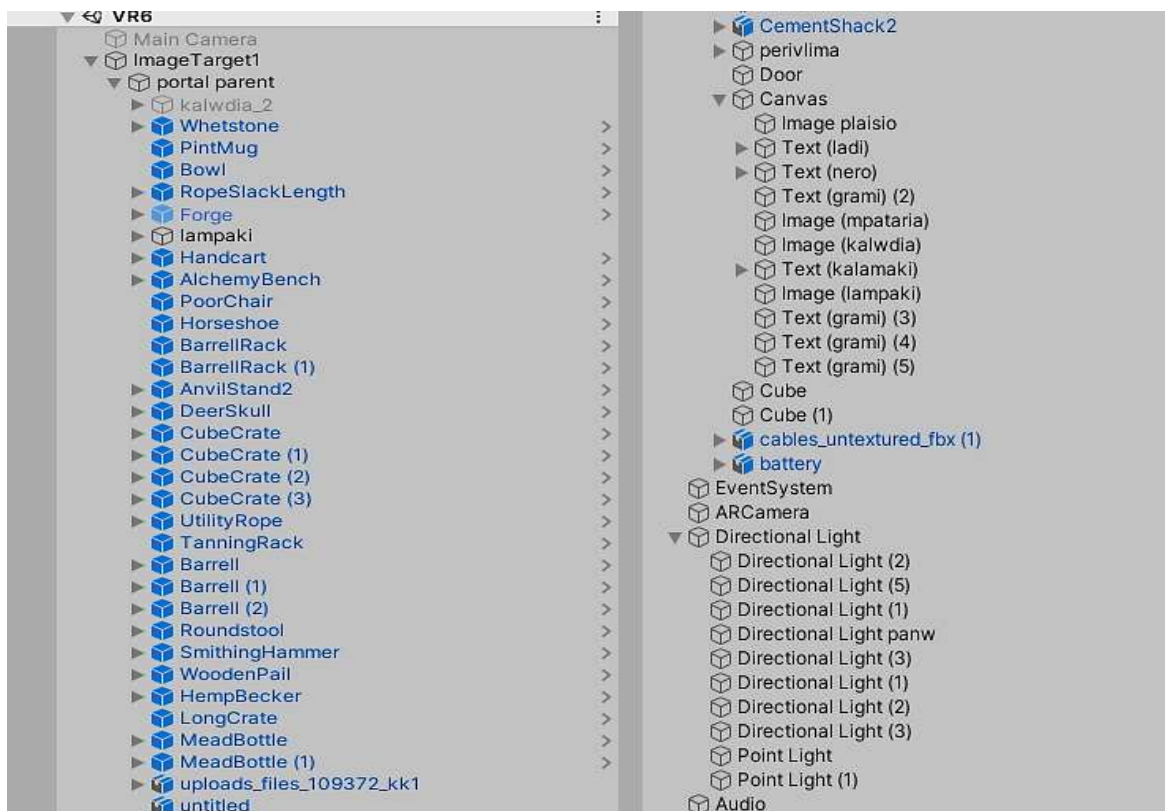
Σε αυτή τη σκηνή, ο παίκτης μετά από προτροπή των οδηγιών της προηγούμενης σκηνής, σκανάρει το qrcode που βρίσκεται στη σελίδα 46 του σχολικού βιβλίου. Μόλις η Vuforia Camera το εντοπίσει, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας πάνω από το qrcode (σχήμα 4.5). Η πόρτα επαυξημένης πραγματικότητας που εμφανίζεται, έχει την ίδια λειτουργικότητα με την πόρτα εικονικής πραγματικότητας της σκηνής “VR4”. Για να φανεί το εικονικό περιβάλλον που υπάρχει πίσω της, ο παίκτης πρέπει να εισέλθει με τη συσκευή του μέσα στην πόρτα. Μόλις γίνει αυτό, ακολουθεί ολική εμβύθιση του παίκτη σε ένα δωμάτιο εικονικής πραγματικότητας, στο οποίο καλείται να βρει τα τρία υλικά του πειράματος. Στο κάτω μέρος της οθόνης υπάρχει μπάρα στην οποία φαίνονται τα υπολειπόμενα υλικά. Η ιεραρχία των αντικειμένων της σκηνής VR6 φαίνεται αναλυτικά στο σχήμα 4.6. Περιληπτικά, αποτελείται από τα ίδια βασικά αντικείμενα της σκηνής “VR4” με τη μόνη διαφορά τα διαφορετικά εικονικά αντικείμενα που υπάρχουν και αποτελούν το ύψος του δωματίου. Πριν τη δημιουργία της πόρτας, έγινε δημιουργία αντικειμένου “image target” και η επιλογή του marker20 από τη βάση highclass. Το παιδί του image target είναι ένα empty “portal parent” προκειμένου τα περιεχόμενα αυτού να εμφανιστούν εφόσον η arcamera εντοπίσει το imagemarker20. Τα δομικά στοιχεία του δωματίου είναι οι τοίχοι, η πόρτα, το πάτωμα και η οροφή.

Εφαρμογή πειράματος για το κεφάλαιο Ενέργεια

Αυτά δημιουργήθηκαν με τη χρήση του Probuilder. Οι ειδικότερες ρυθμίσεις είναι ίδιες με αυτές της σκηνής “VR4”.



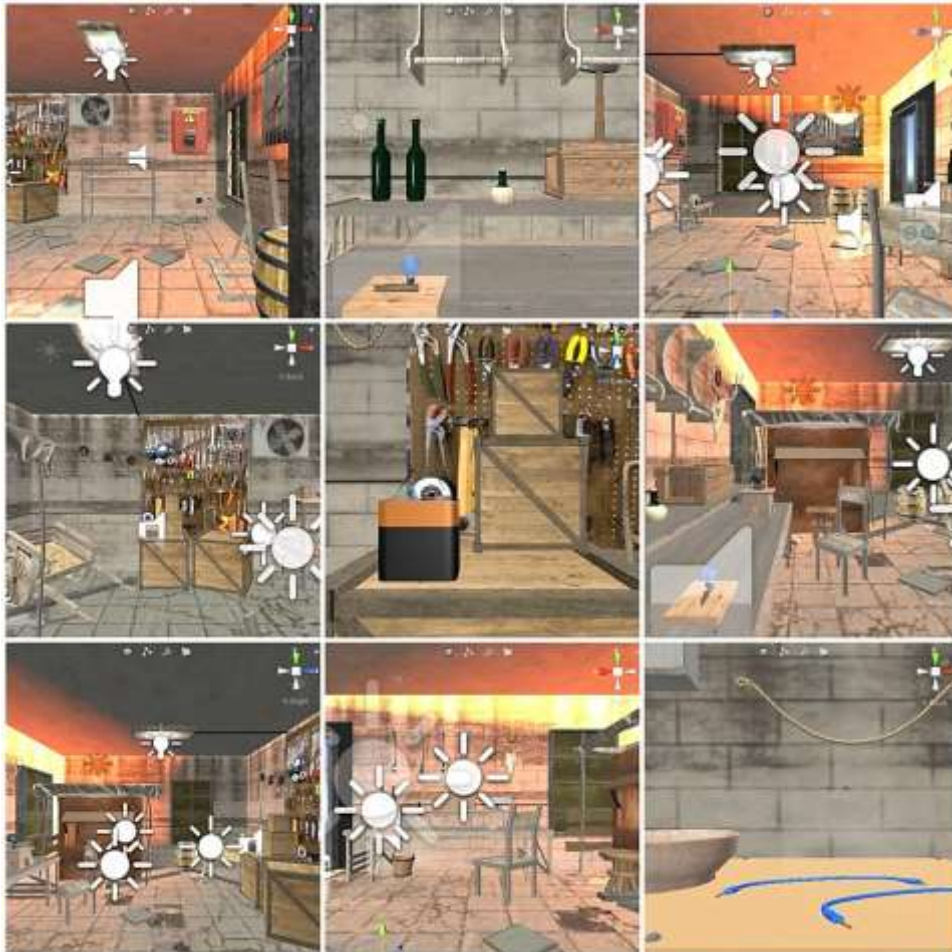
Σχήμα 6.5: Πόρτα επαυξημένης πραγματικότητας VR6 σκηνής



Σχήμα 6.6: Ιεραρχία αντικειμένων VR6

6.6.1 Εικονικό περιβάλλον και εικονικά αντικείμενα VR6

Το εσωτερικό του δωματίου μοιάζει με συνεργείο. Αποτελείται από δεκάδες assets, .fbx και .obj που αποθηκεύτηκαν από το unity asset store και από ιστοσελίδες όπως αυτές αναφέρθηκαν στο αντίστοιχο κεφάλαιο της σκηνής “VR4”. Στο σχήμα 4.7 φαίνεται το εσωτερικό του δωματίου με λεπτομέρειες.



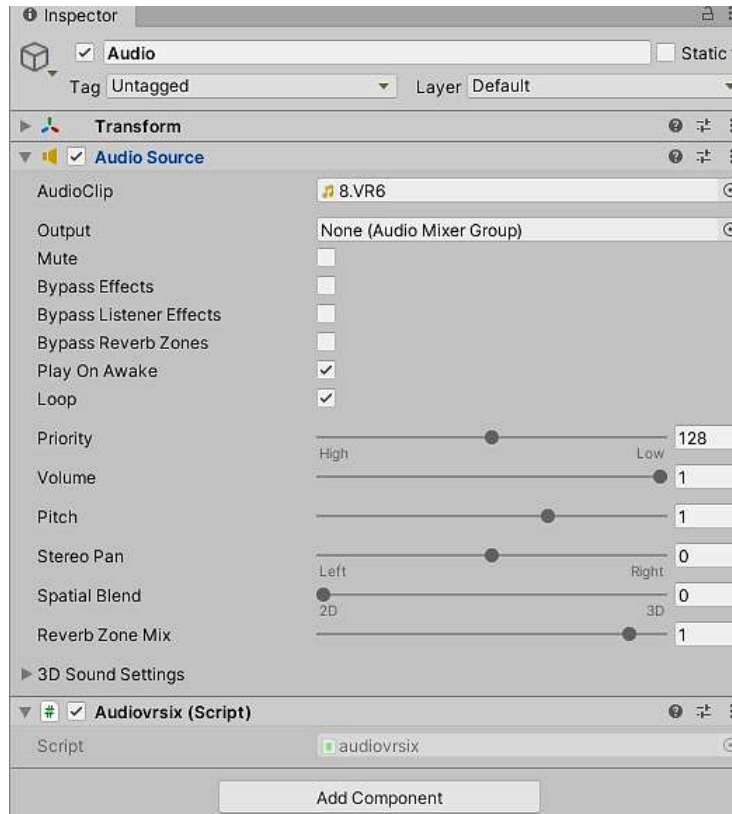
Σχήμα 6.7: Δωμάτιο VR6

Τα τρία υλικά που ο παίκτης πρέπει να συλλέξει προκειμένου να προχωρήσει στο πείραμα φαίνονται στο σχήμα 4.8. Τα υλικά που ο παίκτης πρέπει να μαζέψει προκειμένου να προχωρήσει στο πείραμα, είναι τοποθετημένα μέσα στο εικονικό περιβάλλον με τρόπο έξυπνο, τέτοιο ώστε να μην είναι φανερά εκ πρώτης όψεως αλλά συγχρόνως να είναι διακριτά.



Σχήμα 6.8: Υλικά πειράματος (Ενέργεια)

Για τον ήχο της σκηνής “VR6” χρησιμοποιήθηκε ένα gameobject empty με όνομα “Audio”. Στα components του empty “Audio” προστέθηκε ένα “Audio source” με τις ίδιες ρυθμίσεις της σκηνής “VR4” αλλάζοντας το αρχείο ήχου για να είναι μοναδικό. Ένα ακόμα component που προστέθηκε στο audio είναι το script “audiovrsix”. Το script αυτό είναι παρόμοιο με το script “audioVRFOUR” της σκηνής “VR4”, με τη διαφορά ότι καταστρέφει το αντικείμενο του ήχου στη σκηνή testscene6, δηλαδή την επόμενη της VR6. Στο σχήμα 4.9 φαίνεται ο inspector του audio.



Σχήμα 6.9: Ρυθμίσεις ήχου VR6

6.6.2 Εναλλαγή σκηνής VR6

Στα τρία αντικείμενα που πρέπει να συλλέξει ο παίκτης έχουν προστεθεί τα ίδια components όπως και στη σκηνή “VR4”:

- Box collider
- Audio source:
- Το script “Dotvrldour” (σχήμα 3.11)

Η μπάρα στο κάτω μέρος του παιχνιδιού με τα υπολειπόμενα παιχνίδια είναι ο canvas από την ιεραρχία αντικειμένων και αποτελείται από τρία textmesh Pro με τα ονόματα των αντικειμένων και αντίστοιχα τρεις εικόνες (τικ) που εμφανίζονται μόλις ο παίκτης επιλέξει ένα αντικείμενο από αυτά. Στα components του canva προστέθηκε το script “csSIXx” (σχήμα 4.10) το οποίο έχει την ίδια λειτουργικότητα με το script “canvascriptvrfour” (σχήμα 3.13). Με το script “csSIXx”, όταν ο μετρητής sum φτάσει τον αριθμό τρία στη γραμμή του κώδικα 18 (δηλαδή όταν όλα τα αντικείμενα βρεθούν), καλεί την επόμενη σκηνή μέσω της βιβλιοθήκης UnityEngine.SceneManagement στη γραμμή του κώδικα 19.

```

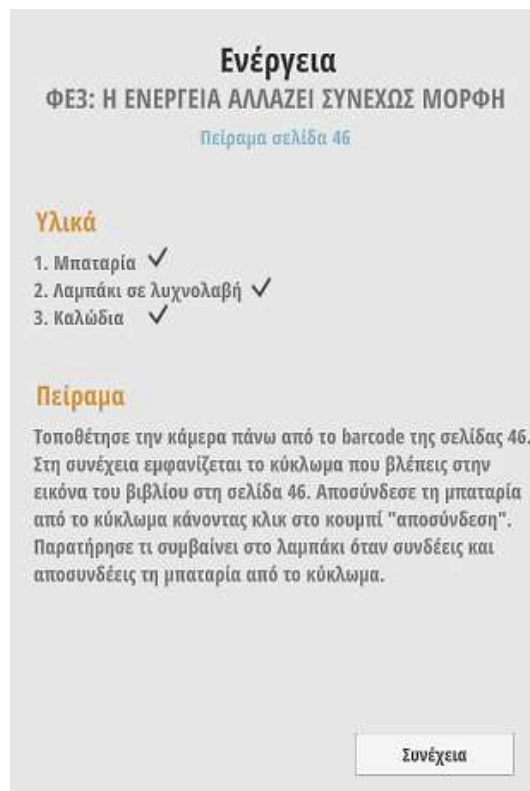
csSIXx.cs x
C:\Users\user\Desktop> pbyxiaki > projects > final project > karantina > Assets > Scripts > VR6 > csSIXx.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class csSIXx : MonoBehaviour
7 {
8     public int sum;
9     // Start is called before the first frame update
10    void Start()
11    {
12        sum = 0;
13    }
14
15    // Update is called once per frame
16    void Update()
17    {
18        if(sum == 3){
19            SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +1);
20        }
21    }
22 }

```

Σχήμα 6.10: Το script “csSIXx”

6.7 Σκηνή: TestScene6

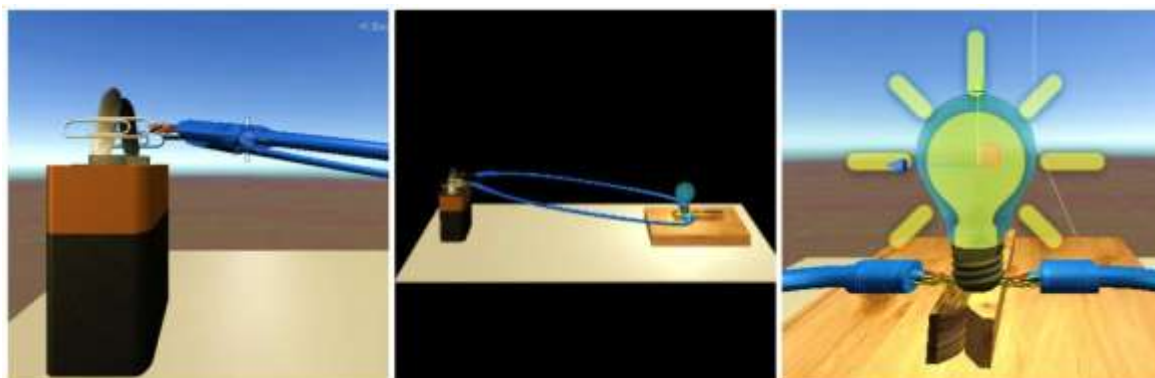
Η σκηνή αυτή φαίνεται στο σχήμα 4.11. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “energeiaAbout” και “fe3-energeia”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται οδηγίες για το πείραμα που ακολουθεί.



Σχήμα 6.11: Σκηνή testscene6

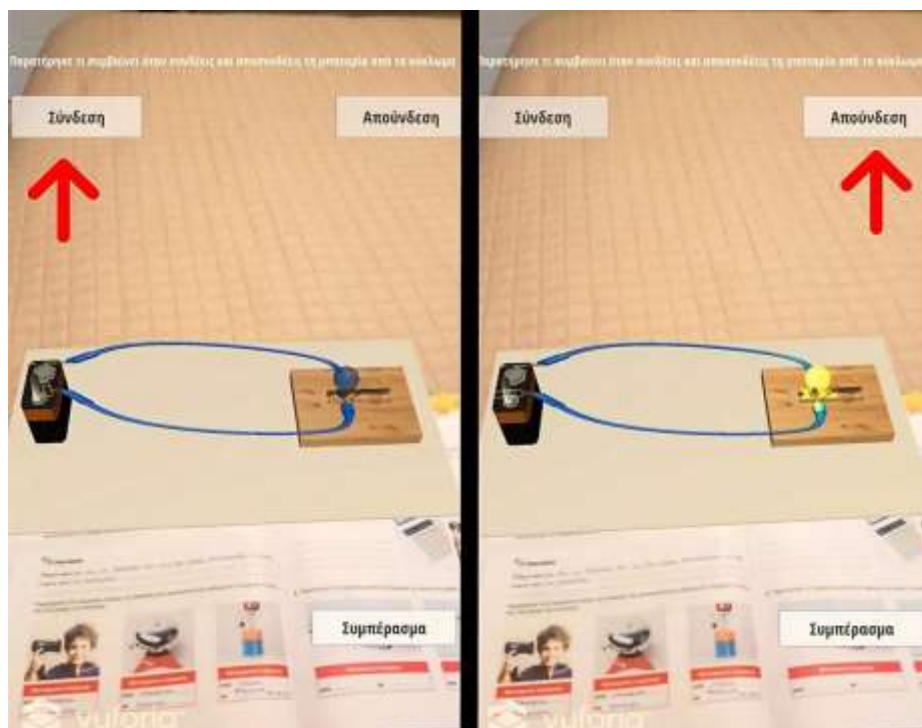
6.8 Σκηνή: 6.animation

Σε αυτή τη σκηνή πραγματοποιείται το πείραμα της σελίδας 46 με τη χρήση εκτεταμένης πραγματικότητας. Στην αρχική οθόνη υπάρχει κατασκευασμένο το κύκλωμα της εικόνας του σχολικού βιβλίου (σχήμα 4.12) το ποίο αποτελείται από τα τρία υλικά: μπαταρία, λαμπάκι σε λυχνιολαβή και τα καλώδια.



Σχήμα 6.12: Το κύκλωμα της σελίδας 46 του σχολικού βιβλίου στη σκηνή “6.animation”

Στο πάνω μέρος της οθόνης υπάρχουν δύο κουμπιά “Σύνδεση” και “Αποσύνδεση” τα οποία αναφέρονται αντίστοιχα στη σύνδεση/αποσύνδεση της μπαταρίας από το κύκλωμα, όπως αυτό ζητείται στο σχολικό βιβλίο για την ολοκλήρωση του πειράματος και την εξαγωγή της παρατήρησης και του συμπεράσματος. Μετά από προτροπή του παίκτη στο πάνω μέρος της οθόνης «Παρατήρησε τι συμβαίνει όταν συνδέεις και αποσυνδέεις την μπαταρία από το κύκλωμα», ο παίκτης πατώντας τα δύο κουμπιά βλέπει τις αλλαγές στο λαμπάκι της λυχνιολαβής. Αυτές οι δύο καταστάσεις μετά την επιλογή των δύο κουμπιών φαίνονται στο σχήμα 4.13.

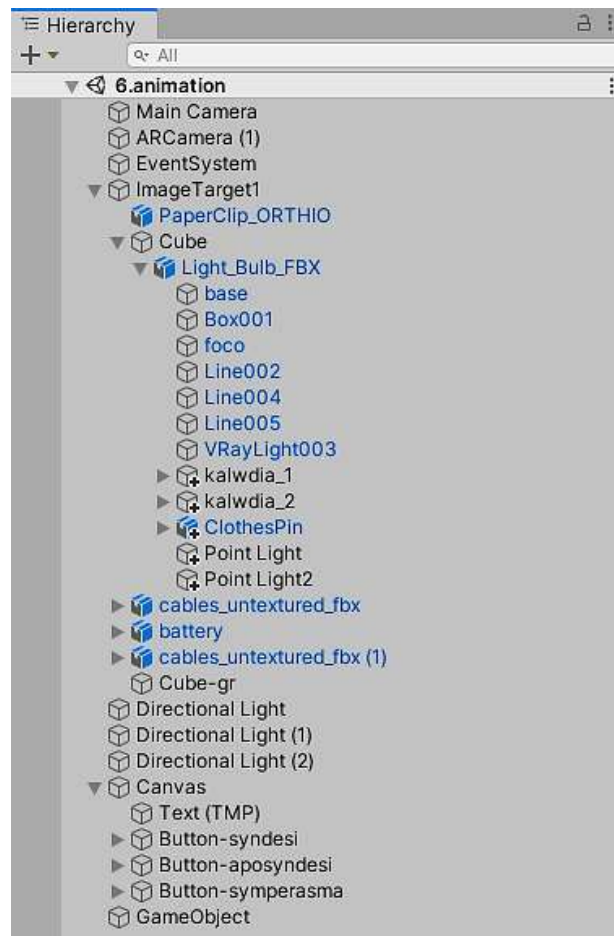


Σχήμα 6.13: Καταστάσεις σύνδεση/αποσύνδεση

6.8.1 Ιεραρχία αντικειμένων – 6.animation

Η ιεραρχία των αντικειμένων της σκηνής 6.animation (σχήμα 4.14) αποτελείται από:

- Ένα event system.
- Μία ARCamera.
- Τρία directional lights για σωστό φωτισμό περιμετρικά της πόρτας και δεκαπέντε point lights εσωτερικά.
- Το image target20 το οποίο εμπεριέχει :
 - Ground: ένα έδαφος που καλύπτει το image target.
 - Το κύκλωμα του σχολικού βιβλίου.
 - Δύο point lights για τις καταστάσεις σύνδεση και αποσύνδεση.
- Ένα canva ο οποίος εμπεριέχει :
 - Ένα TextMeshPro στο πάνω μέρος της οθόνης για οδηγίες του πειράματος.
 - Δύο buttons που αντιστοιχούν στη σύνδεση/αποσύνδεση.
 - Ένα button που αντιστοιχεί στο «συμπέρασμα» και προχωράει στην επόμενη σκηνή.
- Ένα empty μετονομασμένα ως “gameobject” στα components του οποίου προστέθηκε το script “sixanim”, που είναι υπεύθυνο για όλη τη λειτουργία του πειράματος και αναλύεται στη συνέχεια.



Σχήμα 6.14: Ιεραρχία αντικειμένων “6.animation”

6.8.2 Script “sixanim”

Το script “sixanim” φαίνεται αναλυτικά στο σχήμα 4.15. Πιο συγκεκριμένα:

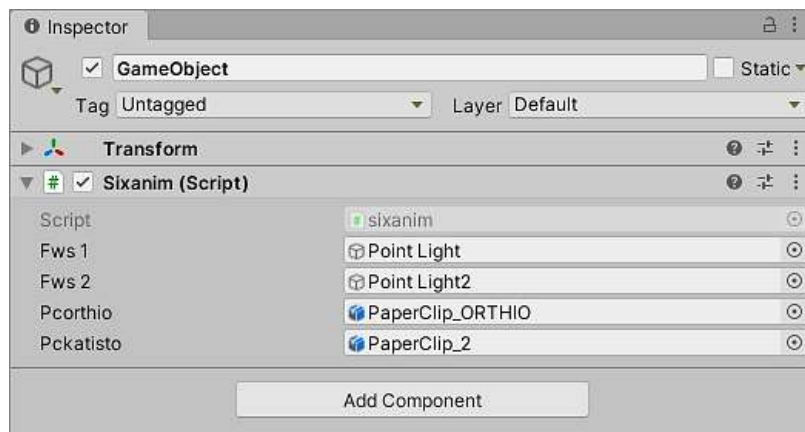
- Στις γραμμές του κώδικα 7 και 8, δημιουργούνται δύο αντικείμενα τύπου “gameobject” με τις ονομασίες “fws1” και “fws2”, τα οποία προορίζονται για τα δύο point lights που θα υλοποιούν τις καταστάσεις σύνδεση/αποσύνδεση της μπαταρίας από το κύκλωμα, δηλαδή καταστάσεις on/off του φωτός.
- Στις γραμμές του κώδικα 9 και 10, δημιουργούνται δύο αντικείμενα τύπου “gameobject” με τις ονομασίες “pcorthio” και “pckathisto”, τα οποία προορίζονται για δύο συνδετήρες (pc=paperclip), οι οποίοι ανάλογα με την κατάσταση σύνδεση/αποσύνδεση είναι όρθιοι ή καθιστοί αντίστοιχα. Ουσιαστικά αποτελούν τους διακόπτες του κυκλώματος.
- Στη συνάρτηση Start() απενεργοποιούνται τα δύο point lights (γραμμές του κώδικα 15 και 16) γιατί στην αρχή το φως του κυκλώματος πρέπει να είναι σβηστό. Στη γραμμή του κώδικα 17 απενεργοποιείται ο συνδετήρας ο καθιστός διότι το κύκλωμα είναι σβηστό. Αυτό σημαίνει πως το μόνο gameobject που είναι ενεργοποιημένο στην αρχική οθόνη είναι ο συνδετήρας ο όρθιος για να δίνει την εικόνα ενός αποσυνδεδεμένου κυκλώματος από την μπαταρία.
- Στη συνάρτηση Syndesi() προκειμένου να δοθεί η αίσθηση ενός συνδεδεμένου κυκλώματος με την μπαταρία είναι ενεργοποιημένα τα δύο point lights (γραμμές του κώδικα 22 και 23) και ο συνδετήρας ο καθιστός (γραμμή του κώδικα 25). Ο συνδετήρας ο όρθιος απενεργοποιείται (γραμμή του κώδικα 24).
- Στη συνάρτηση Aposyndesi() οι καταστάσεις false/true είναι όλες αντίθετες με τις αντίστοιχες της συνάρτησης Syndesi() προκειμένου να δοθεί η αίσθηση ενός μη συνδεδεμένου κυκλώματος με την μπαταρία.

```
sixanim.cs X
C:\Users\user\Desktop\ptyxiaki\projects\final project\karantina\Assets\Scripts\6.animation\sixanim.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class sixanim : MonoBehaviour
6  {
7      public GameObject fws1;
8      public GameObject fws2;
9      public GameObject pcorthio;
10     public GameObject pckatisto;
11
12
13     void Start()
14     {
15         fws1.SetActive(false);
16         fws2.SetActive(false);
17         pckatisto.SetActive(false);
18     }
19
20     public void Syndesi()
21     {
22         fws1.SetActive(true);
23         fws2.SetActive(true);
24         pcorthio.SetActive(false);
25         pckatisto.SetActive(true);
26     }
27
28     public void Aposyndesi()
29     {
30         fws1.SetActive(false);
31         fws2.SetActive(false);
32         pcorthio.SetActive(true);
33         pckatisto.SetActive(false);
34     }
35 }
```

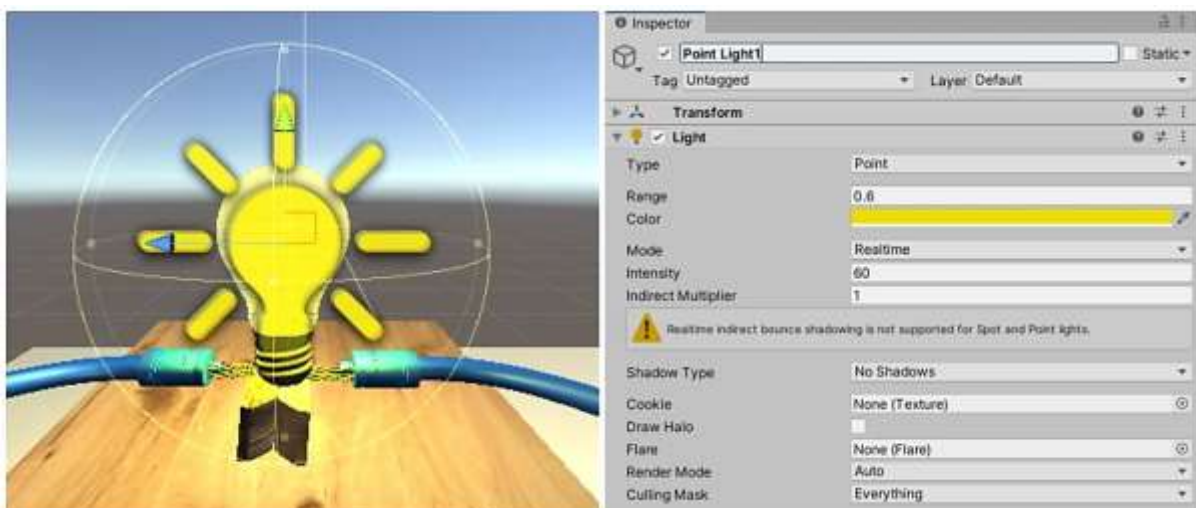
Σχήμα 6.15: Το script “sixanim”

Στο inspector του empty “gameobject” της ιεραρχίας των αντικειμένων της σκηνής “6.animation” προστέθηκε το script “sixanim”. Όπως φαίνεται στο σχήμα 4.16, στα πεδία:

- “fws1” έγινε drag and drop το pointlight1 (σχήμα 4.17) από την ιεραρχία αντικειμένων.
- “fws2” έγινε drag and drop το pointlight2 (σχήμα 4.18) από την ιεραρχία αντικειμένων. Χρησιμοποιούνται δύο point lights για την κατάσταση «σύνδεση», για μια πιο ρεαλιστική κατάσταση του φωτός.
- “Pcorthio” έγινε drag and drop το paperclip1 (σχήμα 4.19), δηλαδή το όρθιο, από την ιεραρχία αντικειμένων.
- “Pckathisto” έγινε drag and drop το paperclip2 (σχήμα 4.20), δηλαδή το καθιστό, από την ιεραρχία αντικειμένων.



Σχήμα 6.16: inspector του empty “gameobject” της σκηνής “6.animation”

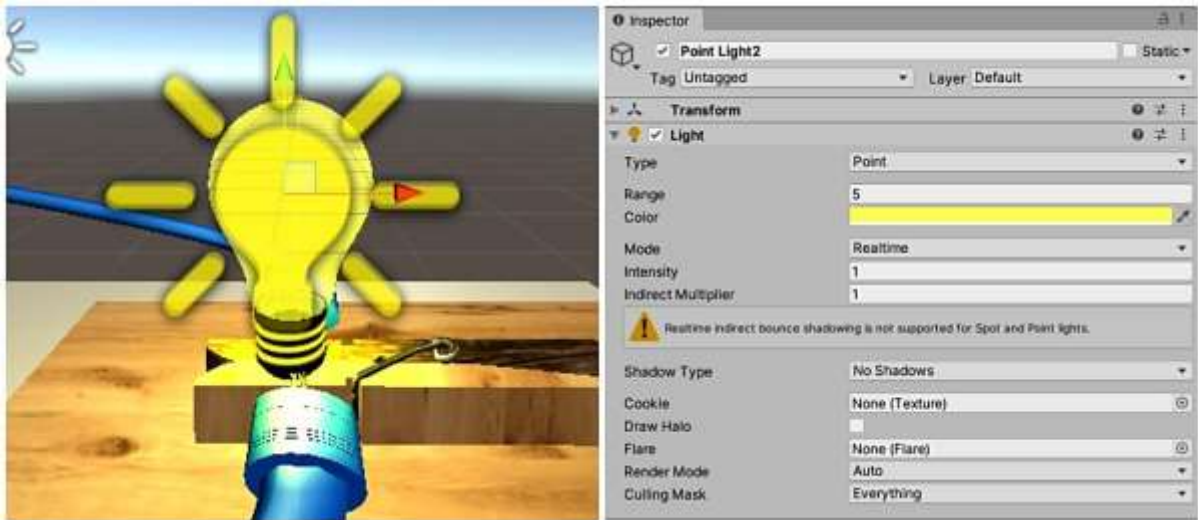


Σχήμα 6.17: pointlight1 (κύκλωμα σελίδα σχολικού βιβλίου 46)

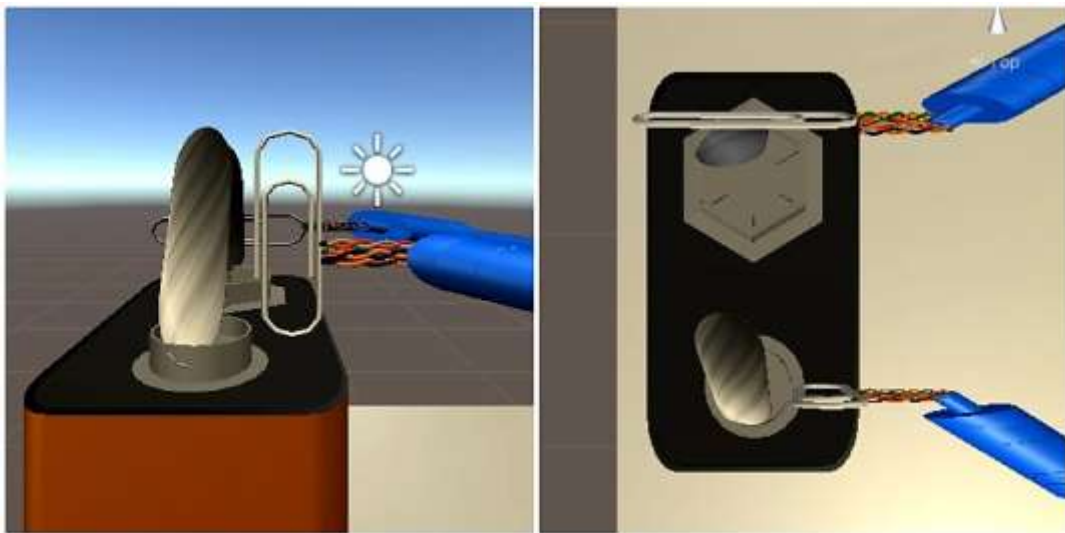
Για τη λειτουργικότητα των κουμπιών σύνδεση/αποσύνδεση:

- Στο canva, στο inspector του button “syndesi”: onClick() event → select object = gameobject → select function = sixanim.Syndesi.
- Στο canva, στο inspector του button “aposyndesi”: onClick() event → select object = gameobject → select function = sixanim.Aposyndesi.

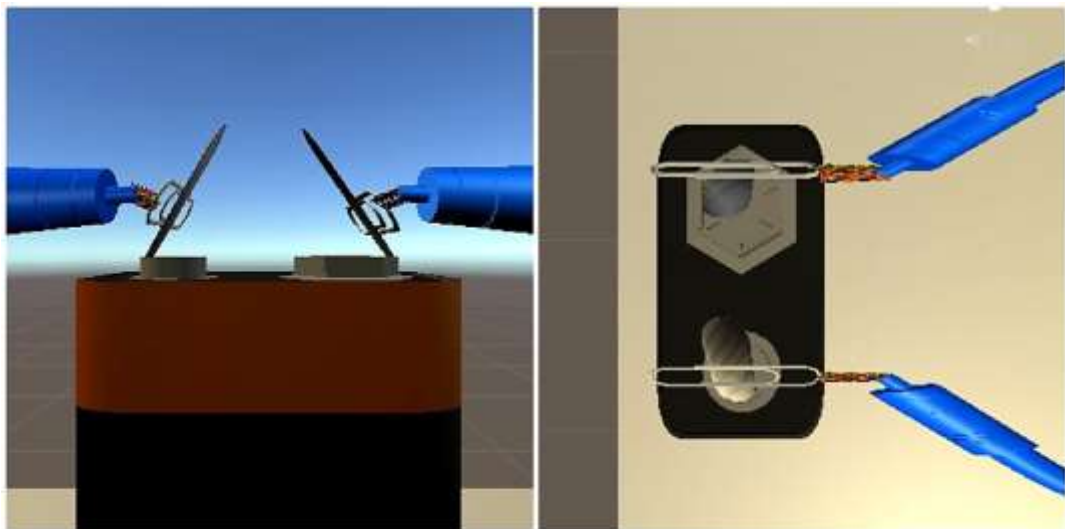
Εφαρμογή πειράματος για το κεφάλαιο Ενέργεια



Σχήμα 6.18: pointlight2 (κύκλωμα σελίδα σχολικού βιβλίου 46)



Σχήμα 6.19: paracclip1-Διακόπτης κλειστός (κύκλωμα σελίδα σχολικού βιβλίου 46)



Σχήμα 6.20: paracclip2-Διακόπτης ανοιχτός (κύκλωμα σελίδα σχολικού βιβλίου 46)

Για τη λειτουργικότητα του κουμπιού «Συμπέρασμα» έγινε drag and drop το script “ tscsix” (σχήμα 4.21) στα components του button “symperasma”. Το script “ tscsix” χρησιμοποιώντας τη βιβλιοθήκη UnityEngine.SceneManagement, προχωράει το παιχνίδι στην επόμενη σκηνή δηλαδή στο συμπέρασμα. Στο inspector του button “symperasma”: onClick() event → select object = button-symperasma → select function = tscsix.synexeia.

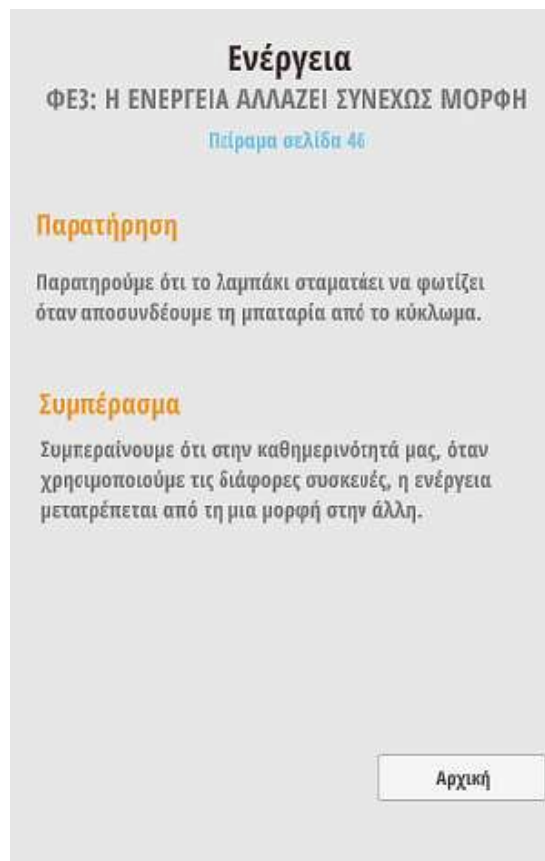
```

tscsix.cs
C:\Users\user\Desktop\ptyxiaki\projects\final project\karantina\Assets\Scripts\Panels\energeia> tscsix.cs
1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3
4  public class tscsix : MonoBehaviour
5  {
6      public void synexeia()
7      {
8          SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +1);
9      }
10 }
    
```

Σχήμα 6.21: Το script “ tscsix”

6.9 Σκηνή: symperasmataEnergeia

Η σκηνή αυτή φαίνεται στο σχήμα 4.22. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “EnergeiaAbout”, “fe3-energeia” και “testscene6”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται η παρατήρηση και τα συμπεράσματα που πρέπει να συμπληρώσει ο μαθητής στο βιβλίο.



Σχήμα 6.22: Σκηνή-symperasmataEnergeia

Στο κουμπί «Αρχική» στα components έχει προστεθεί το script “entostar” το οποίο φαίνεται στο σχήμα 4.23. Στο onclick() event: Select object = το button «Αρχική» → select function = arxiki(). Το script “entostar” χρησιμοποιεί τη βιβλιοθήκη UnityEngine.SceneManagement για να πάει στην αρχική σκηνή (γραμμή 8 του κώδικα) η οποία βρίσκεται τριάντα εννιά σκηνές πίσω (βλέπε σχήμα 1.10).

```
entostar.cs
C:\Users\user\Desktop\ptyxiaki\projects\final project\karantina\Assets\Scripts\Panels\energeia\entostar.cs
1  using UnityEngine;
2  using UnityEngine.SceneManagement;
3
4  public class entostar : MonoBehaviour
5  {
6      public void arxiki()
7      {
8          SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex -39);
9      }
10 }
```

Σχήμα 6.23: Το script “entostar”

6.10 Επίλογος

Το πείραμα της σελίδας 46 του σχολικού βιβλίου, υλοποιείται με τη χρήση εκτεταμένης πραγματικότητας μέσω των παραπάνω 6 σκηνών. Οι παραπάνω έξι σκηνές είναι απαραίτητες για την ομαλή ροή του πειράματος. Μετά το πέρας της τελευταίας σκηνής του πειράματος, είναι θεμιτό να έχουν επιτευχθεί οι μαθησιακοί στόχοι που αναφέρθηκαν στο κεφάλαιο 4.3.

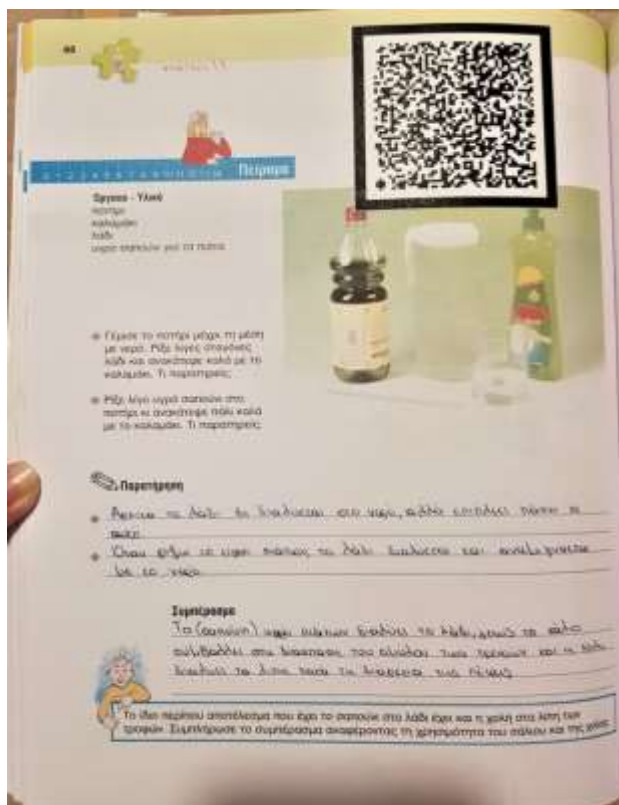
Κεφάλαιο 7ο: Εφαρμογή πειράματος για το κεφάλαιο Πεπτικό σύστημα

7.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται οι μαθησιακοί στόχοι του πειράματος της σελίδας 66 του κεφαλαίου «Πεπτικό σύστημα» του σχολικού βιβλίου. Επιπλέον, αναλύονται όλα τα βήματα της δημιουργίας των σκηνών που υλοποιούν το πείραμα, χρησιμοποιώντας τεχνολογίες εκτεταμένης πραγματικότητας.

7.2 Σελίδα βιβλίου 66

Στο κεφάλαιο «Πεπτικό σύστημα» ανήκει η ενότητα «ΦΕ3: ΤΟ ΤΑΞΙΔΙ ΤΗΣ ΤΡΟΦΗΣ ΣΥΝΕΧΙΖΕΤΑΙ». Η ενότητα ξεκινάει με ένα εισαγωγικό ερέθισμα και ακολουθεί μια πειραματική αντιμετώπιση στην οποία οι μαθητές καλούνται να μασήσουν το ψωμί περίπου τέσσερα με πέντε λεπτά, χωρίς να το καταπιούν. Ακολουθεί συζήτηση στην τάξη σχετικά με τη λειτουργία των αδένων. Στη συνέχεια γίνεται εξήγηση στους μαθητές ότι το ψωμί περιέχει μια ουσία, το άμυλο, και αναφέρεται ότι η αλλαγή χρώματος του βάμματος ιωδίου αποτελεί ένδειξη για την ύπαρξη αμύλου. Με το πείραμα αυτό οι μαθητές διαπιστώνουν ότι στο ψωμί που μάσησαν το βάμμα ιωδίου δεν αλλάζει χρώμα, επειδή το σάλιο διασπά το άμυλο των τροφών. Μετά, ακολουθεί το πείραμα της σελίδας 66 (σχήμα 5.1). Για το πείραμα αυτό οι μαθητές χρειάζεται να φέρουν στην τάξη τα προϊόντα της σελίδας 66. Με το πείραμα αυτό διαπιστώνουν ότι το σαπούνι πιάτων διαλύει το λάδι. Οι μαθητές διαβάζουν προσεκτικά τις οδηγίες, εκτελούν το πείραμα και σημειώνουν την παρατήρησή τους στον προβλεπόμενο χώρο στο βιβλίο τους.

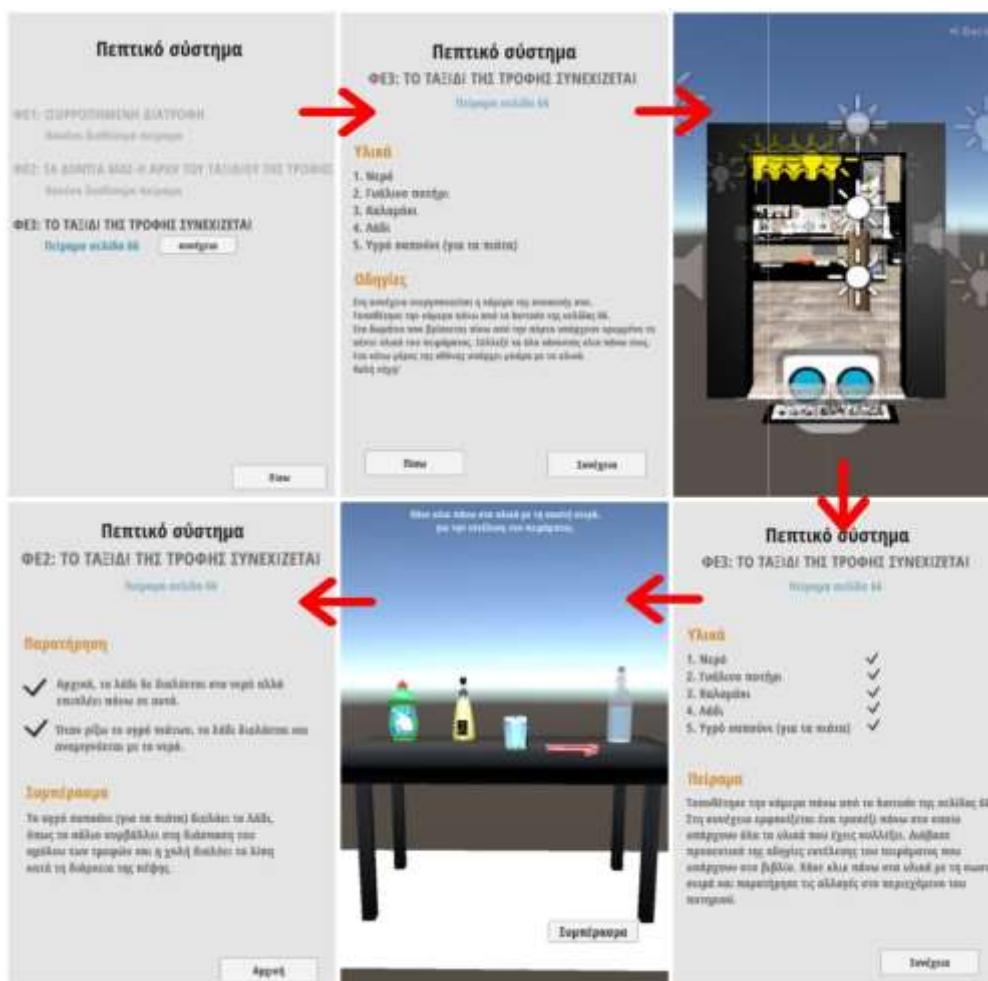


Σχήμα 7.1: Σελίδα βιβλίου 66

Η εξαγωγή του συμπεράσματος γίνεται με κατάλληλες ερωτήσεις και συζήτηση στην τάξη, μέσα από την οποία οι μαθητές γενικεύουν τις παρατηρήσεις τους από τα προηγούμενα πειράματα και διατυπώνουν το συμπέρασμα. Καθώς το συμπέρασμα από το πρώτο πείραμα είναι δύσκολο, δίνεται η απαραίτητη βοήθεια για τη διατύπωσή του. Εισάγεται ο όρος «διάσπαση» και ακολουθεί η εξήγησή του. Στο τέλος οι μαθητές πρέπει να εμπεδώσουν ότι η επίδραση που έχει το υγρό σαπουνί για τα πιάτα στο λάδι έχει και η χολή στα λίπη των τροφών. Έπειτα καλούνται να συνδυάσουν τα επιμέρους συμπεράσματα των παραπάνω πειραμάτων και να διατυπώσουν ένα τελικό συμπέρασμα. Επιπλέον, για την πραγματοποίηση του πειράματος εκτεταμένης πραγματικότητας, έχει προστεθεί στη σελίδα 24 εκτυπωμένο το image target17 από τη βάση high_class.

7.3 Οι σκηνές του πειράματος – Πεπτικό σύστημα

Το πείραμα εκτεταμένης πραγματικότητας αποτελείται από έξι σκηνές όπως αυτές φαίνονται στο σχήμα 5.2. Αρχικά ο μαθητής-παίκτης, αφού επιλέξει το κεφάλαιο του πεπτικού συστήματος (σκηνή: startScene), εμφανίζονται τα διαθέσιμα πειράματα των ενοτήτων του κεφαλαίου (σκηνή: PeptikoAbout). Αφού επιλέξει ποιο πείραμα θέλει να υλοποιηθεί, εμφανίζονται οι οδηγίες του πειράματος καθώς και τα υλικά τα οποία πρέπει να συλλέξει (σκηνή: fe3-peptiko). Στη συνέχεια, ενεργοποιείται η κάμερα της συσκευής του, και αφού σκανάρει το qrcode της σελίδας 66, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας και μια μπάρα με τα εναπομείναντα υλικά τα οποία βρίσκονται πίσω από αυτή, σε ένα εικονικό περιβάλλον (σκηνή: VR3).



Σχήμα 7.2: Σκηνές πειράματος σελίδας 66

Μόλις συλλέξει όλα τα υλικά της μπάρας, εμφανίζονται οδηγίες (σκηνή: testScene3) για το πείραμα που θα ακολουθήσει. Σκανάροντας ξανά το qrcode της σελίδας 66, εμφανίζεται ένα τραπέζι πάνω στο οποίο βρίσκονται όλα τα υλικά του πειράματος (σκηνή: 3.animation). Κάνοντας κλικ με τη σωστή σειρά πάνω στα υλικά, εξελίσσεται το πείραμα. Μόλις ολοκληρωθεί, εμφανίζεται στο κάτω μέρος της οθόνης το κουμπί “συμπέρασμα” το οποίο οδηγεί τον παίκτη στην επόμενη σκηνή η οποία αποτελείται από τη συμπληρωμένη παρατήρηση του βιβλίου καθώς και του συμπεράσματος (σκηνή: symperasmataPeptiko). Μετά την ολοκλήρωση του παραπάνω πειράματος εκτεταμένης πραγματικότητας, οι μαθησιακοί στόχοι που είναι θεμιτό να επιτευχθούν είναι οι εξής:

- Να αναφέρουν οι μαθητές τα όργανα του πεπτικού συστήματος και να εξηγήσουν τη λειτουργία καθενός από αυτά.
- Να εντοπίσουν οι μαθητές σε τομή του ανθρώπινου σώματος τη θέση των οργάνων του πεπτικού συστήματος.
- Να διαπιστώσουν οι μαθητές πειραματικά τη χρησιμότητα του σάλιου για τη διάσπαση του αμύλου των τροφών.
- Να διαπιστώσουν οι μαθητές πειραματικά τη χρησιμότητα της χολής στη λειτουργία της πέψης.

7.4 Σκηνή: PeptikoAbout

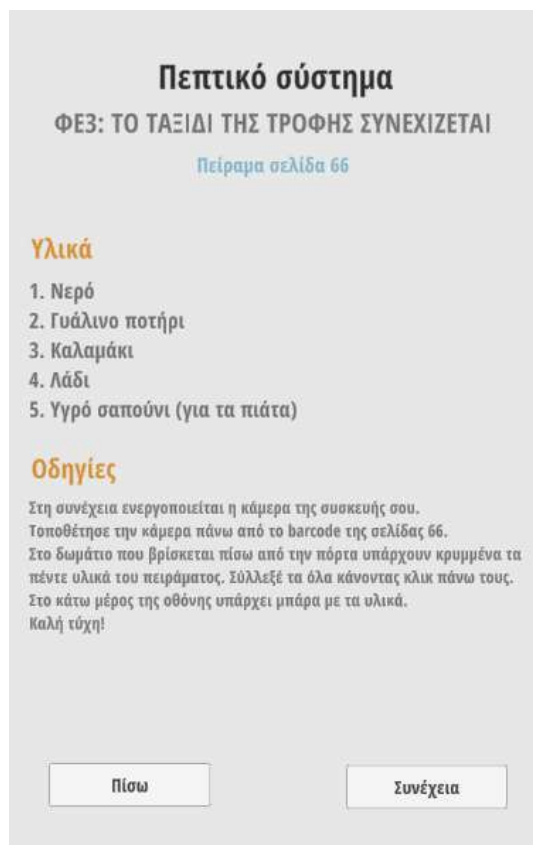
Η σκηνή αυτή φαίνεται στο σχήμα 5.3. Η ιεραρχία των αντικειμένων της αποτελείται από: μια main camera, ένα directional light για το σωστό φωτισμό της σελίδας, ένα event system για τη λειτουργία των κουμπιών και των εναλλαγών των σκηνών και ένα canva στον οποίο υπάρχουν πολλά textMeshPro για την εισαγωγή των κειμένων καθώς και 2 buttons. Με αυτά γίνεται η εναλλαγή των σκηνών, μέσω ενός script παρόμοιου με του script “menu”, το οποίο αποτελείται από τις μεθόδους “next()” και “back()” που χρησιμοποιούν τη βιβλιοθήκη UnityEngine.SceneManagement.



Σχήμα 7.3: Σκηνή “PeptikoAbout”

7.5 Σκηνή: Fe3-peptiko

Η σκηνή αυτή φαίνεται στο σχήμα 5.4. Η ιεραρχία των αντικειμένων της είναι ίδια με αυτή της σκηνής “PeptikoAbout”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro και το περιεχόμενό τους είναι διαφορετικό.



Σχήμα 7.4: Σκηνή fe3-peptiko

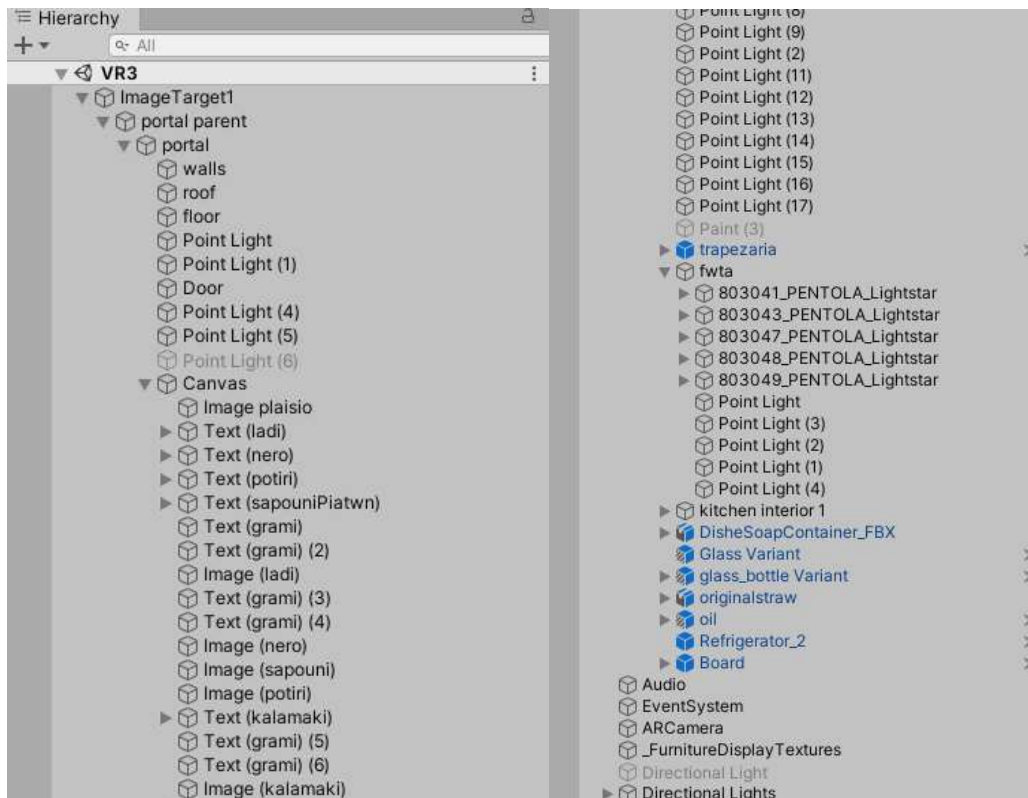
7.6 Σκηνή: VR3

Σε αυτή τη σκηνή, ο παίκτης μετά από προτροπή των οδηγιών της προηγούμενης σκηνής, σκανάρει το qrcode που βρίσκεται στη σελίδα 66 του σχολικού βιβλίου. Μόλις η Vuforia Camera το εντοπίσει, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας πάνω από το qrcode (σχήμα 5.5). Η πόρτα επαυξημένης πραγματικότητας που εμφανίζεται, έχει την ίδια λειτουργικότητα με την πόρτα εικονικής πραγματικότητας των προηγούμενων σκηνών VR. Για να φανεί το εικονικό περιβάλλον που υπάρχει πίσω της, ο παίκτης πρέπει να εισέλθει με τη συσκευή του μέσα στην πόρτα. Μόλις γίνει αυτό, ακολουθεί ολική εμβύθιση του παίκτη σε ένα δωμάτιο εικονικής πραγματικότητας, στο οποίο καλείται να βρει τα 5 υλικά του πειράματος. Στο κάτω μέρος της οθόνης υπάρχει μπάρα στην οποία φαίνονται τα υπολειπόμενα υλικά. Η ιεραρχία των αντικειμένων της σκηνής VR3 φαίνεται αναλυτικά στο σχήμα 5.6. Περιληπτικά, αποτελείται από τα ίδια βασικά αντικείμενα των προηγούμενων σκηνών VR, με τη μόνη διαφορά τα διαφορετικά εικονικά αντικείμενα που υπάρχουν και αποτελούν το ύψος του δωματίου. Πριν τη δημιουργία της πόρτας, έγινε δημιουργία αντικειμένου “image target” και η επιλογή του marker17 από τη βάση highclass. Το παιδί του image target είναι ένα empty “portal parent” προκειμένου τα περιεχόμενα αυτού να εμφανιστούν εφόσον η arcamera εντοπίσει το imagemarker17. Τα δομικά στοιχεία του δωματίου είναι οι τοίχοι, η πόρτα, το πάτωμα και η οροφή.

Αυτά δημιουργήθηκαν με τη χρήση του Probuilder. Οι ειδικότερες ρυθμίσεις είναι ίδιες με αυτές των προηγούμενων σκηνών VR.



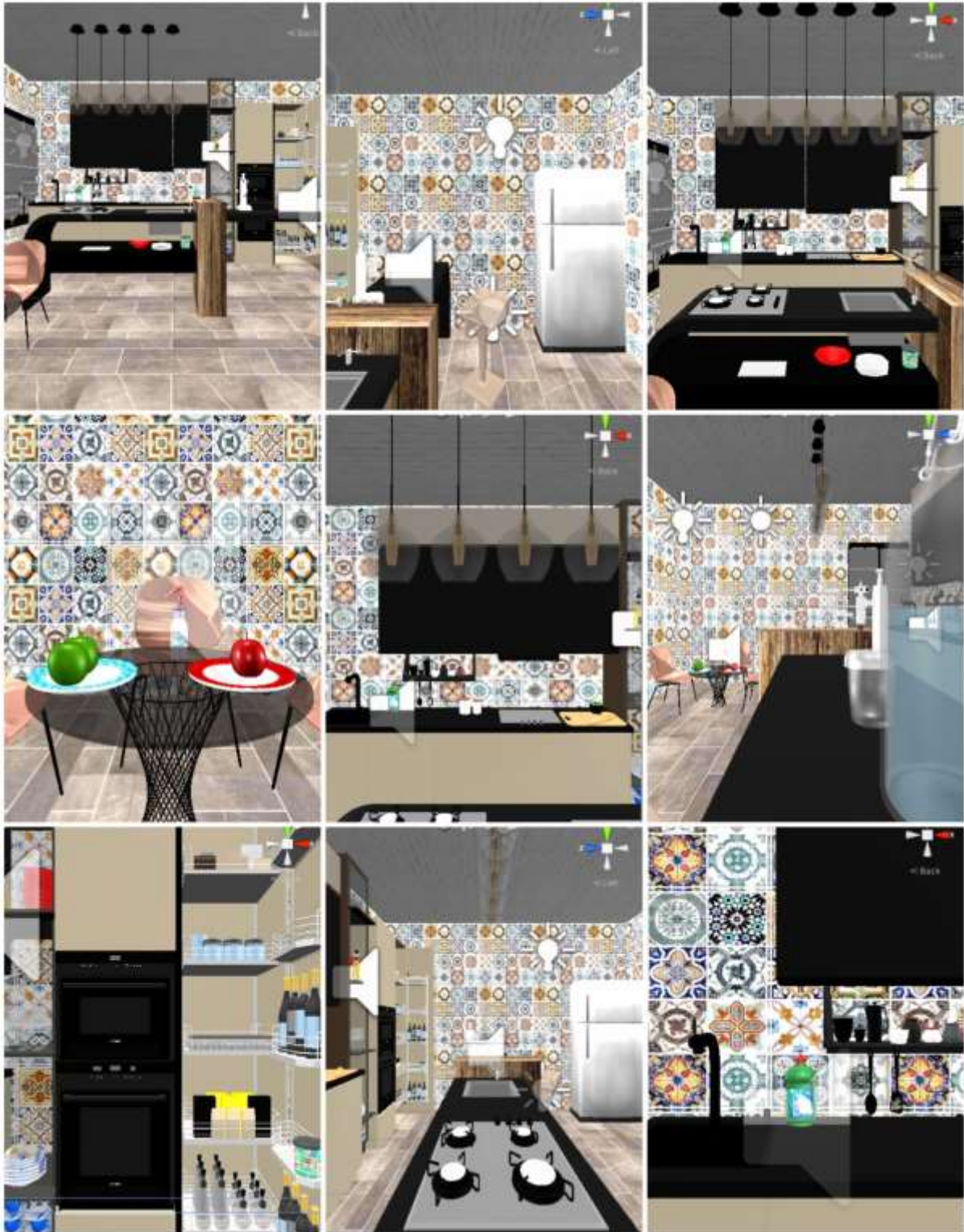
Σχήμα 7.5: Πόρτα επαυξημένης πραγματικότητας VR3 σκηνής



Σχήμα 7.6: Ιεραρχία αντικειμένων VR3

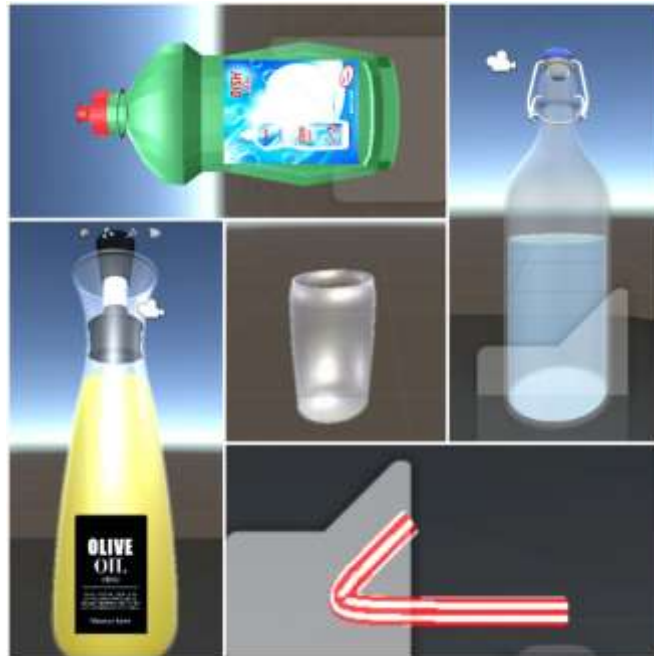
7.6.1 Εικονικό περιβάλλον και εικονικά αντικείμενα VR3

Το εσωτερικό του δωματίου μοιάζει με κουζίνα με μπαρ. Αποτελείται από δεκάδες assets, .fbx και .obj που αποθηκεύτηκαν από το unity asset store και από ιστοσελίδες όπως αυτές αναφέρθηκαν στο αντίστοιχο κεφάλαιο της σκηνής “VR4”. Στο σχήμα 5.7 φαίνεται το εσωτερικό του δωματίου με λεπτομέρειες.



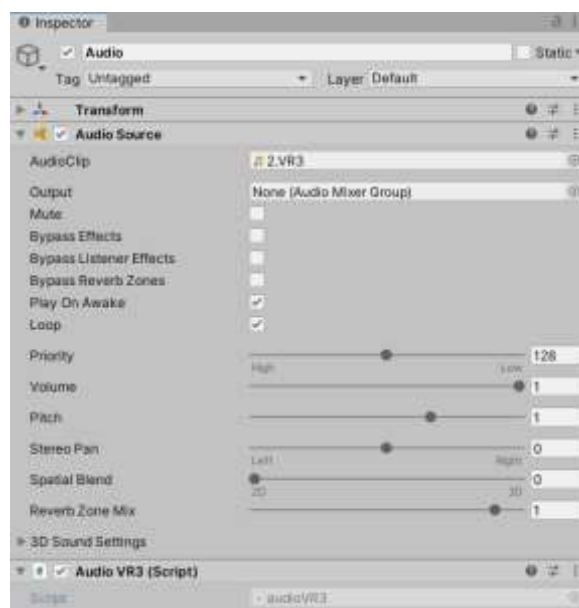
Σχήμα 7.7: Δωμάτιο VR3

Τα τρία υλικά που ο παίκτης πρέπει να συλλέξει προκειμένου να προχωρήσει στο πείραμα φαίνονται στο σχήμα 5.8. Τα υλικά που ο παίκτης πρέπει να μαζέψει προκειμένου να προχωρήσει στο πείραμα, είναι τοποθετημένα μέσα στο εικονικό περιβάλλον με τρόπο έξυπνο, τέτοιο ώστε να μην είναι φανερά εκ πρώτης όψεως αλλά συγχρόνως να είναι διακριτά. Το καλαμάκι είναι εξ ολοκλήρου φτιαγμένο στο blender και τα υπόλοιπα είναι κατεβασμένα και τροποποιημένα κατάλληλα για τις ανάγκες του πειράματος.



Σχήμα 7.8: Υλικά πειράματος (Πεπτικό σύστημα)

Για τον ήχο της σκηνής “VR3” χρησιμοποιήθηκε ένα gameobject empty με όνομα “Audio” με τις ίδιες ρυθμίσεις των προηγούμενων σκηνών “VR4” αλλάζοντας το αρχείο ήχου για να είναι μοναδικό. Το script “ audioVR3” που προστέθηκε στα componens καταστρέφει το αντικείμενο του ήχου στη σκηνή testscene3, δηλαδή την επόμενη της VR3. Στο σχήμα 5.9 φαίνεται ο inspector του audio.



Σχήμα 7.9: Ρυθμίσεις ήχου VR3

7.6.2 Εναλλαγή σκηνής VR3

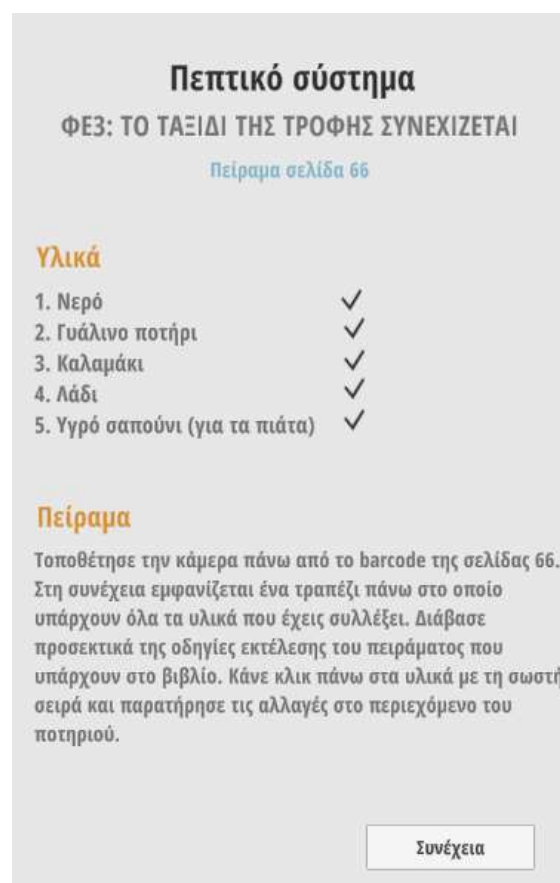
Στα τρία αντικείμενα που πρέπει να συλλέξει ο παίκτης έχουν προστεθεί τα ίδια components όπως και στις προηγούμενες σκηνές VR:

- Box collider
- Audio source:
- Το script “Dotvrndour” (σχήμα 3.11)

Η μπάρα στο κάτω μέρος του παιχνιδιού με τα υπολειπόμενα παιχνίδια είναι ο canvas από την ιεραρχία αντικειμένων και αποτελείται από πέντε textmesh Pro με τα ονόματα των αντικειμένων και αντίστοιχα πέντε εικόνες (τικ) που εμφανίζονται μόλις ο παίκτης επιλέξει ένα αντικείμενο από αυτά. Στα components του canva προστέθηκε το script “CanvasScriptVR3”, το οποίο έχει την ίδια λειτουργικότητα με το script “canvascriptvrfour” (σχήμα 3.13) και το script “csSIXx” (σχήμα 4.10). Με το script “CanvasScriptVR3”, όταν ο μετρητής sum φτάσει τον αριθμό πέντε (δηλαδή όταν όλα τα αντικείμενα βρεθούν), καλεί την επόμενη σκηνή μέσω της βιβλιοθήκης UnityEngine.SceneManagement.

7.7 Σκηνή: TestScene3

Η σκηνή αυτή φαίνεται στο σχήμα 5.10. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “PepitikoAbout” και “fe3-pepitiko”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται οδηγίες για το πείραμα που ακολουθεί.



Σχήμα 7.10: Σκηνή testscene3

7.8 Σκηνή: 3.animation

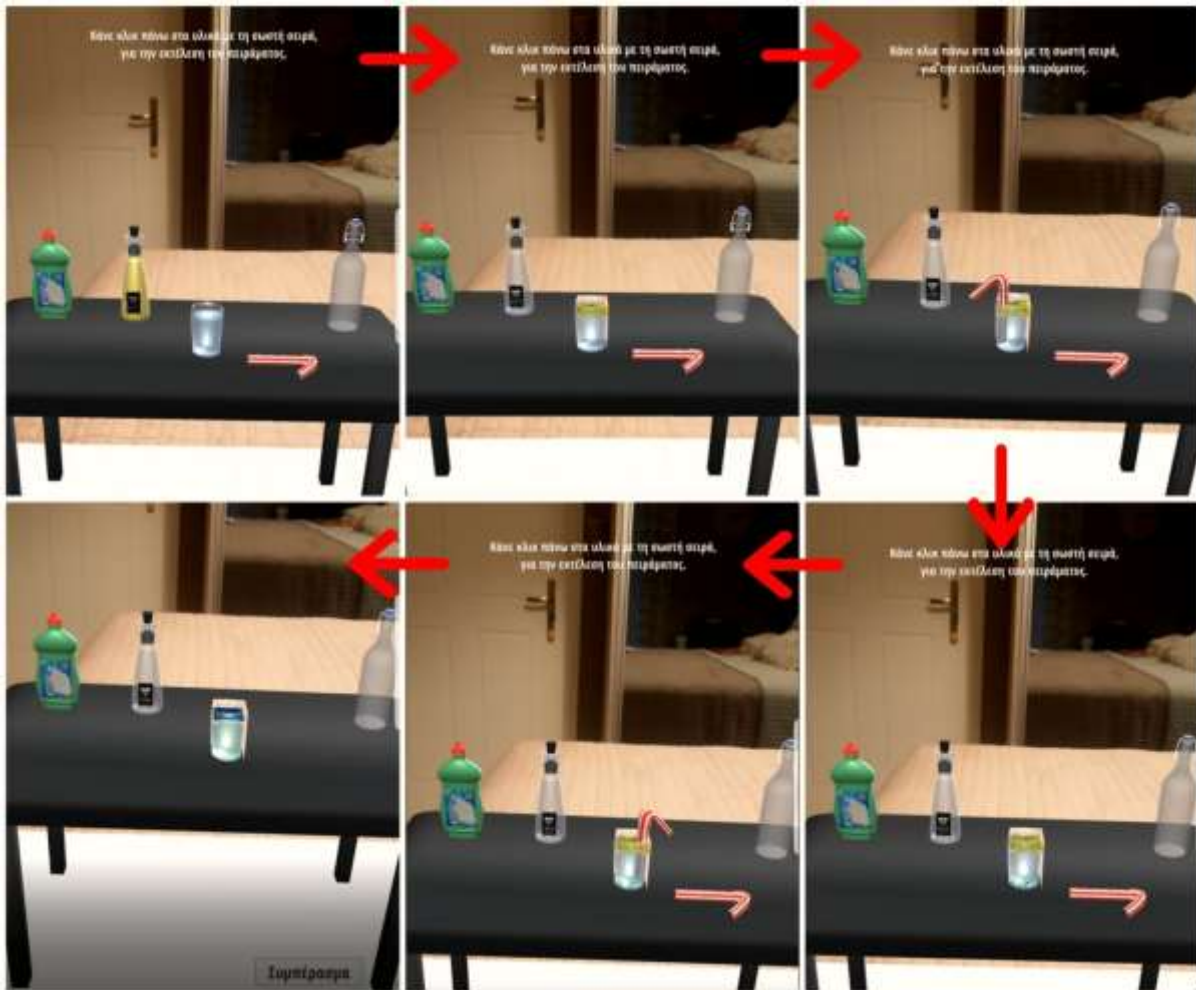
Σε αυτή τη σκηνή πραγματοποιείται το πείραμα της σελίδας 66 με τη χρήση εκτεταμένης πραγματικότητας. Στην αρχική οθόνη (σχήμα 5.11) υπάρχει ένα τραπέζι στο οποίο είναι τοποθετημένα τα πέντε υλικά του πειράματος. Στο πάνω μέρος της οθόνης υπάρχει κείμενο-προτροπή για τον παίκτη «Κάνε κλικ πάνω στα υλικά με τη σωστή σειρά, για την εκτέλεση του πειράματος. Η σωστή σειρά σύμφωνα με την εκφώνηση του σχολικού βιβλίου είναι: Μπουκάλι νερό → Λάδι → Καλαμάκι → Υγρό σαπούνι για τα πιάτα → Καλαμάκι. Το πείραμα αποτελείται από τα εξής βήματα:

1. Κλικ στο μπουκάλι με νερό → εισαγωγή νερού στο ποτήρι (βήμα 1).
2. Κλικ στο λάδι → εισαγωγή λαδιού στο ποτήρι (βήμα 2).
3. Κλικ στο καλαμάκι → ανακάτεμα μίγματος νερού (βήμα 3).
4. Κλικ στο υγρό σαπούνι για τα πιάτα → εισαγωγή του υγρού στο ποτήρι (βήμα 4).
5. Κλικ στο καλαμάκι → ανακάτεμα μίγματος νερού (βήμα 5).
6. Συμπέρασμα.

Ο παίκτης αρχικά πρέπει να παρατηρήσει ότι το λάδι δε διαλύεται στο νερό αλλά επιπλέει. Στη συνέχεια με την προσθήκη του υγρού πιάτων, πρέπει να παρατηρήσει ότι το λάδι διαλύεται και αναμιγνύεται με το νερό. Μόλις ολοκληρωθούν τα βήματα του πειράματος, στο κάτω μέρος της οθόνης εμφανίζεται το κουμπί «Συμπέρασμα» το οποίο οδηγεί στην επόμενη σκηνή. Στο σχήμα 5.12 φαίνονται οι οθόνες μετά από τη σωστή επιλογή του κάθε υλικού του πειράματος.



Σχήμα 7.11: Αρχική οθόνη σκηνής “3.animation”



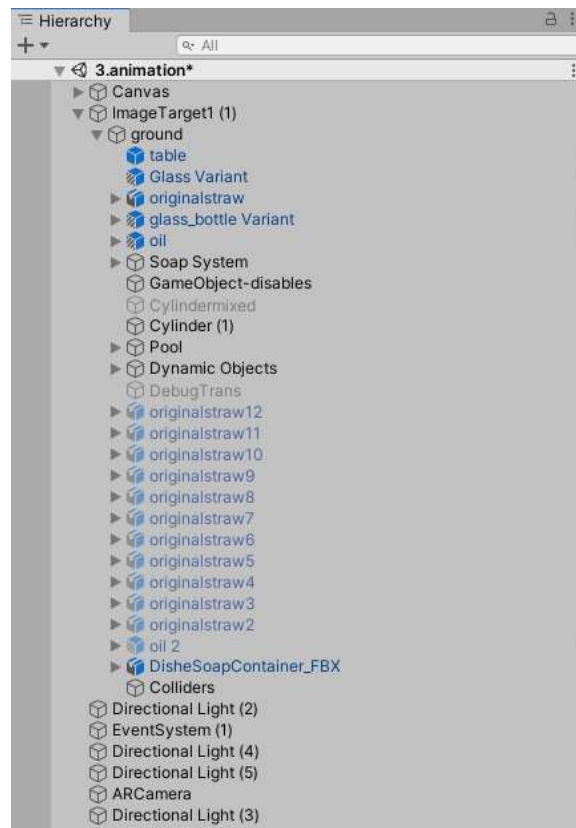
Σχήμα 7.12: Οι θρόνες μετά από τη σωστή επιλογή του κάθε υλικού του πειράματος

7.8.1 Ιεραρχία αντικειμένων – 3.animation

Η ιεραρχία των αντικειμένων της σκηνής 3.animation (σχήμα 5.13) αποτελείται από:

- Ένα event system.
- Μία ARCamera.
- Έξι directional lights για σωστό φωτισμό περιμετρικά της πόρτας και δεκαπέντε point lights εσωτερικά.
- Το image target17 το οποίο εμπεριέχει :
 - Ένα ground με το τραπέζι, τα οποία αποτελούν αντικείμενα «γονείς» για όλα τα παρακάτω.
 - Τα πέντε gameobjects-υλικά του πειράματος
 - Τρία gameobject → cylinder τα οποία αποτελούν το περιεχόμενο (δηλαδή νερό) του μπουκαλιού και του ποτηριού πριν και μετά από την ανάμιξη των υγρών.
 - Ένα δεύτερο λάδι χωρίς το εσωτερικό του (για όταν αδειάσει το περιεχόμενο του).
 - Ένα soapSystem από gameobjects → spheres το οποίο αποτελεί το υγρό των πιάτων μέσα στο ποτήρι.

- Το dynamicObjects το οποίο αποτελείται από δεκάδες buoyancy Spheres, το οποίο θα αναλυθεί παρακάτω λεπτομερώς.
- Μια επιφάνεια pool η οποία αποτελεί την επιφάνεια του νερού στο ποτήρι και θα αναλυθεί παρακάτω.
- Δύο empties “GameObject-disables” και “Colliders” αποκλειστικά για τη λειτουργία scripts.
- Έντεκα κλώνοι του καλαμακιού.
- Ένα canvas ο οποίος εμπεριέχει :
 - Ένα TextMeshPro στο πάνω μέρος της οθόνης για οδηγίες του πειράματος.
 - Ένα button που αντιστοιχεί στο «συμπέρασμα» και προχωράει στην επόμενη σκηνή.



Σχήμα 7.13: Η ιεραρχία των αντικειμένων της σκηνής 3.animation

7.8.2 Η λειτουργία του πειράματος μέσα από τα scripts (Πεπτικό σύστημα)

Αρχικά έγινε η εισαγωγή του component “box Collider” σε τέσσερα υλικά του πειράματος: καλαμάκι, ποτήρι, λάδι και υγρό σαπούνι για τα πιάτα. Σε κάθε ένα από αυτά τα υλικά το boxCollider τοποθετήθηκε προκειμένου η περιοχή του χώρου που πιάνει να οριοθετεί το άγγιγμα του παίκτη για τη συνάρτηση onMouseDown90. Στο ποτήρι δεν έγινε εισαγωγή boxCollider διότι ο παίκτης δε χρειάζεται να το επιλέξει, είναι μόνο για παρατήρηση. Στα παραπάνω τέσσερα υλικά προστέθηκε στα components ένα AudioSource το οποίο είναι ανάλογο με το άκουσμα της λειτουργίας τους. Ακολούθησε η δημιουργία ενός gameObject → empty με όνομα “GameObject-disables” στο οποίο έγινε η επισύναψη του script “setactine” (σχήμα 5.14). Το script “setactine” δημιουργεί τέσσερα πεδία τύπου game object με ονομασίες: cylinderPotiri, pool, synamicObjects και oil2. Στη συνάρτηση

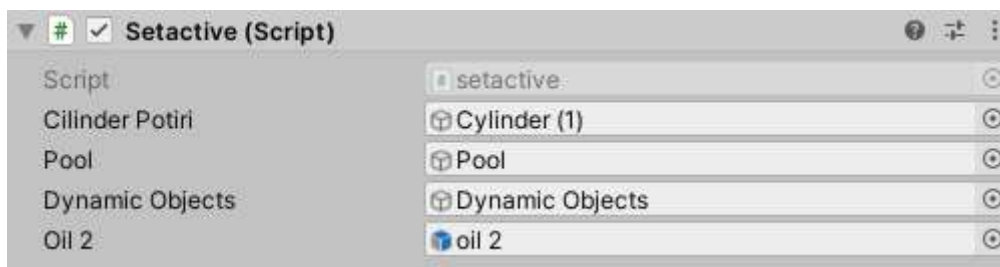
Εφαρμογή πειράματος για το κεφάλαιο Πεπτικό σύστημα

Start() τα απενεργοποιεί. Αυτά τα τέσσερα αντικείμενα είναι εξ αρχής απενεργοποιημένα γιατί χρειάζονται στη συνέχεια.

```
setactive.cs X
C:\Users\user\Desktop\ptyxiaki\projects\final project\karantina\Assets\Scripts\3.animation\setactive.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class setactive : MonoBehaviour
6 {
7     public GameObject cilinderPotiri;
8     public GameObject pool;
9     public GameObject dynamicObjects;
10    public GameObject oil2;
11
12    void Start()
13    {
14        cilinderPotiri.SetActive(false);
15        pool.SetActive(false);
16        dynamicObjects.SetActive(false);
17        oil2.SetActive(false);
18    }
19 }
```

Σχήμα 7.14: Το script “setactine”

Τα τέσσερα πεδία που δημιουργήθηκαν παραμετροποιήθηκαν κατάλληλα όπως φαίνεται στο σχήμα 5.15.



Σχήμα 7.15: Παραμετροποίηση των πεδίων του script “setactine”

Επιπλέον, έγινε η δημιουργία ενός ακόμα gameObject → empty με όνομα “Colliders” στο οποίο έγινε η επισύναψη του script “all” (σχήμα 5.16). Το script “all”:

- Στις γραμμές 12-14 του κώδικα δημιουργούνται τρία πεδία τύπου game object με ονομασίες: gm1, gm2 και gm3.
- Στις γραμμές 8-10 του κώδικα δημιουργούνται τρία αντικείμενα τύπου collider.
- Στις γραμμές 18-19 αφού δέχεται τον collider του gameObject gm1, τον απενεργοποιεί. Αντίστοιχα στις επόμενες γραμμές για τους δύο επόμενους colliders των gm2 και gm3.

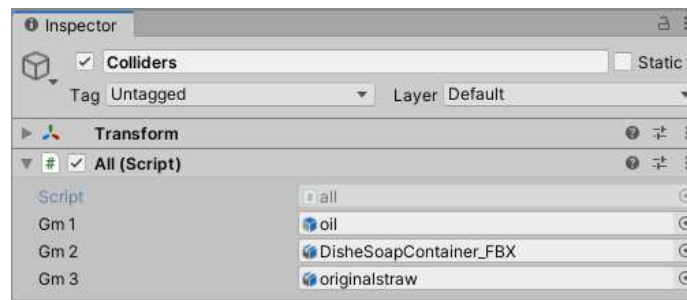
Το script “all” γενικά απενεργοποιεί τη δυνατότητα του λάθους κλικ σε αντικείμενο. Αφήνει δηλαδή εξ αρχής ως πρώτη επιλογή κλικ μόνο στο αντικείμενο του μπουκαλιού (για να γεμίσει νερό στο ποτήρι).

```

all.cs
C:\Users\user\Desktop> physiki\projects> final project> karantina> Assets> Scripts> 3.animation> all.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class all : MonoBehaviour
7 {
8     Collider m_Collider1;
9     Collider m_Collider2;
10    Collider m_Collider3;
11
12    public GameObject gm1;
13    public GameObject gm2;
14    public GameObject gm3;
15
16    void Start()
17    {
18        m_Collider1=gm1.GetComponent<Collider>();
19        m_Collider1.enabled = false;
20
21        m_Collider2=gm2.GetComponent<Collider>();
22        m_Collider2.enabled = false;
23
24        m_Collider3=gm3.GetComponent<Collider>();
25        m_Collider3.enabled = false;
26    }
27 }
    
```

Σχήμα 7.16: Το script “all”

Τα τρία πεδία που δημιουργήθηκαν παραμετροποιήθηκαν κατάλληλα όπως φαίνεται στο σχήμα 5.17.



Σχήμα 7.17: Παραμετροποίηση των πεδίων του script “all”

7.8.3 Πείραμα-Βήμα 1

Στο βήμα 1, μετά από κλικ του παίκτη πάνω στο μπουκάλι με το νερό, γίνεται η εισαγωγή του νερού στο ποτήρι. Στο μπουκάλι (στην ιεραρχία των αντικειμένων) έγινε επισύναψη του script “disableCollider” (σχήμα 5.18).

```

DisableCollider.cs
C:\Users\user\Desktop> physiki\projects> final project> karantina> Assets> Scripts> 3.animation> DisableCollider.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class DisableCollider : MonoBehaviour
7 {
8     Collider m_Collider1; // collider you the disable
9     Collider m_Collider2;
10    public GameObject gm;
11
12
13
14    void OnMouseDown()
15    {
16        m_Collider1=gm.GetComponent<Collider>();
17        m_Collider1.enabled = true;
18
19        m_Collider2=GetComponent<Collider>();
20        m_Collider2.enabled = false;
21    }
    }
    
```

Σχήμα 7.18: Το script “disableCollider”

Στη γραμμή 10 του κώδικα δημιουργείται ένα πεδίο τύπου `gameObject` με όνομα “gm”. Στο πεδίο αυτό έγινε drag and drop το `gameObject` του λαδιού. Στη γραμμή 13 του κώδικα ξεκινάει η συνάρτηση `onMouseDown()` της οποίας το περιεχόμενο εκτελείται όταν γίνει το πρώτο άγγιγμα στο αντικείμενο που έχει επισημανθεί το `script`, δηλαδή στο μπουκάλι. Συνεπώς, μόλις γίνει κλικ στο μπουκάλι, παύει να έχει `collider` για να μην μπορεί να επιλεγθεί λανθασμένα (γραμμές 18-19), και ενεργοποιείται ο `collider` του επόμενου αντικειμένου που πρέπει να επιλέξει ο παίκτης δηλαδή το λάδι. Επίσης, στο μπουκάλι (στην ιεραρχία των αντικειμένων) έγινε επισύναψη και άλλου `script`, του “`onClickEvents`” (σχήμα 5.19).



```
onclিকেvents.cs
C:\Users\user\Desktop\ptyxiaki\projects\final project\karantina\Assets\Scripts\3.animation\onclিকেvents.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class onclিকেvents : MonoBehaviour
6 {
7     public GameObject neroMroukali;
8     public GameObject neroPotiri;
9
10    void OnMouseDown()
11    {
12        neroMroukali.SetActive(false);
13        neroPotiri.SetActive(true);
14        gameObject.GetComponent<AudioSource>().Play();
15    }
16 }
```

Σχήμα 7.19: Το script “onClickEvents”

Σε αυτό το `script`, στις γραμμές 7 και 8, δημιουργούνται δύο πεδία τύπου `gameObject` με ονομασίες “`neroMroukali`” και “`neroPotiri`”. Στο πεδίο “`neroMroukali`” έγινε drag and drop το `gameObject` → `cylinder` του μπουκαλιού, δηλαδή το εσωτερικό του μπουκαλιού (νερό), ενώ στο πεδίο “`neroPotiri`” έγινε drag and drop το `gameObject` → `cylinder` του ποτηριού, δηλαδή το εσωτερικό του ποτηριού (νερό). Στη συνέχεια, μόλις γίνει κλικ στο μπουκάλι, ενεργοποιείται η συνάρτηση στη γραμμή 10 η οποία με τη σειρά της ενεργοποιεί το νερό στο ποτήρι και συγχρόνως απενεργοποιεί το νερό στο μπουκάλι, δίνοντας την αίσθηση ότι το μπουκάλι γέμισε το ποτήρι με το νερό του. Στη γραμμή του κώδικα 14, καλείται να παίξει το `audiosource`, το οποίο αντιστοιχεί σε ήχο «γεμίσματος νερού».

7.8.4 Πείραμα-Βήμα 2

Στο βήμα 2, μετά από κλικ του παίκτη πάνω στο λάδι, γίνεται η εισαγωγή του λαδιού στο νερό του ποτηριού. Στο λάδι (στην ιεραρχία των αντικειμένων) έγινε επισύναψη του `script` “`disableCollider`” (σχήμα 5.18). Στο πεδίο “`gm`” έγινε drag and drop το `gameObject` καλαμάκι, προκειμένου να ενεργοποιηθεί ο `collider` του, αφού είναι το επόμενο υλικό για κλικ. Επιπλέον, Στο λάδι (στην ιεραρχία των αντικειμένων) έγινε επισύναψη και του `script` “`oilclick`” (σχήμα 5.20). Αυτό το `script` δημιουργεί τρία πεδία τύπου `gameObject`:

- Oil2 (γραμμή 7): σε αυτό το πεδίο έγινε drag and drop το αντικείμενο `oil2` το οποίο είναι η συσκευασία του λαδιού χωρίς το περιεχόμενό της. Μόλις γίνει κλικ στο λάδι (`oil1`), στη γραμμή 15 καταστρέφεται το αντικείμενο του γεμάτου λαδιού και τη θέση του παίρνει το άδειο (γραμμή 14).
- Pool (γραμμή 8): σε αυτό το αντικείμενο έγινε drag and drop το αντικείμενο `pool`, το οποίο αποτελεί την επιφάνεια του νερού στο ποτήρι. Είναι ένα αρκετά περίπλοκο αντικείμενο καθώς έχει πολλές φυσικές ιδιότητες προκειμένου να κάνει την αίσθηση πως επιπλέει το λάδι

πάνω στο νερό. Ο inspector του pool φαίνεται στο σχήμα 5.21 και αποτελείται κατά βάση από δύο script “Water FX” (Παράρτημα Α) και “Water Volume” (Παράρτημα Β).

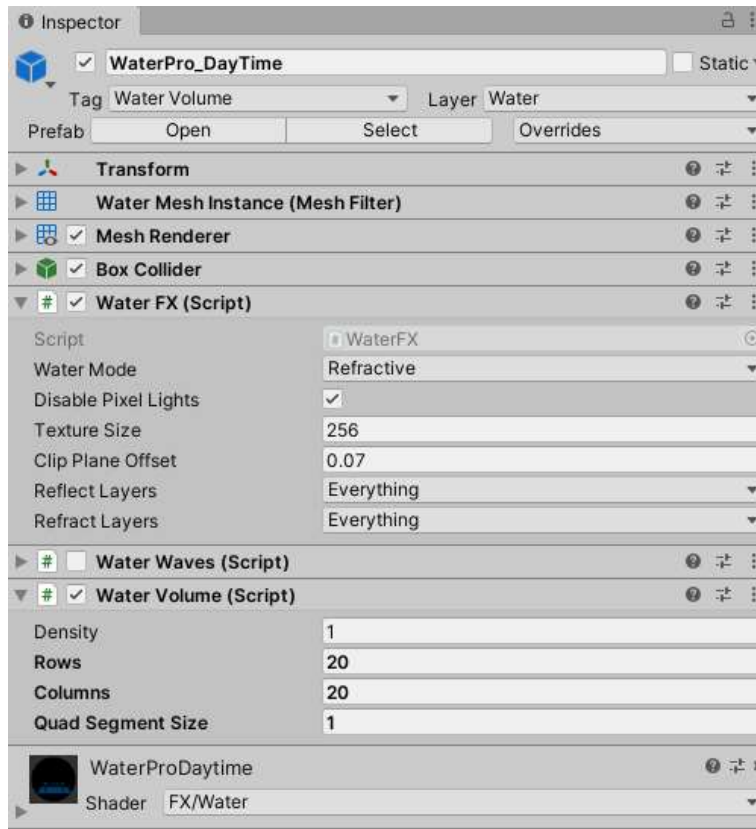
- ο DynamicObjects (γραμμή 9): Είναι ένα σύνολο αντικειμένων από buoyancy spheres, οι οποίες αποτελούν το υγρό του λαδιού. Ο inspector του dynamicObjects φαίνεται στο σχήμα 5.22 και αποτελείται από το script “Floating Object” (Παράρτημα C) το οποίο δίνει στη δυνατότητα στις buoyancy spheres να επιπλέουν στην επιφάνεια pool του νερού χάρη στον συνδυασμό των παραπάνω τριών scripts.

```

oilclick.cs
C:\Users\user\Desktop\ptyxiaki\projects\final project\karantina\Assets\Scripts\3.animation> oilclick.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class oilclick : MonoBehaviour
6 {
7     public GameObject oil2;
8     public GameObject pool;
9     public GameObject dynamicObjects;
10
11     void OnMouseDown()
12     {
13         gameObject.GetComponent<AudioSource>().Play();
14         oil2.SetActive(true);
15         Destroy(gameObject, 1);
16         dynamicObjects.SetActive(true);
17         pool.SetActive(true);
18     }
19 }

```

Σχήμα 7.20: Το script “oilclick”



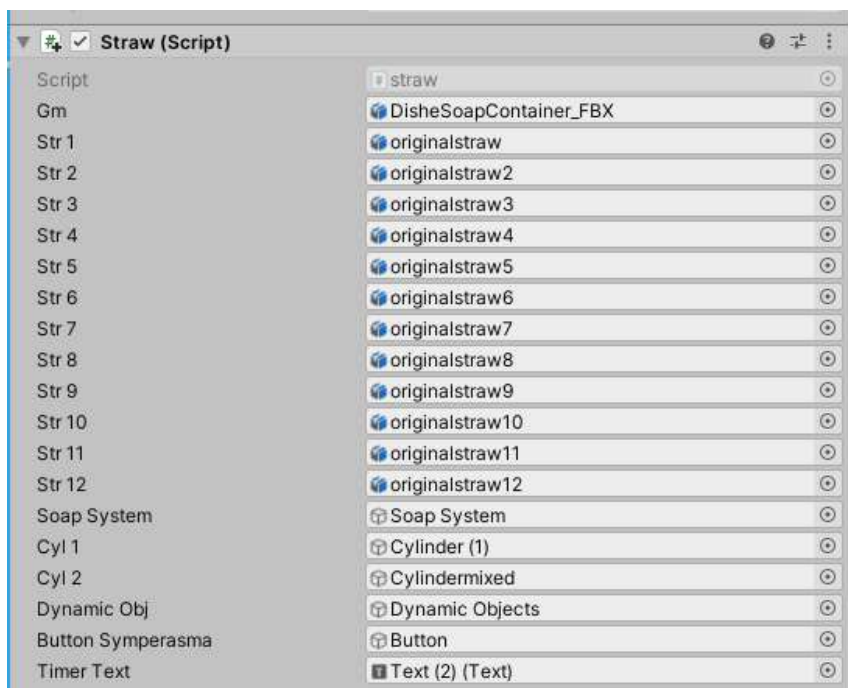
Σχήμα 7.21: Ο inspector του pool



Σχήμα 7.22: Ο inspector του dynamicObjects

7.8.5 Πείραμα-Βήματα 3 και 5

Στο βήμα 3, μετά από κλικ του παίκτη πάνω στο καλαμάκι, γίνεται η εισαγωγή του στο μίγμα του νερού. Το καλαμάκι κάνει ένα μικρό κύκλο μέσα στο νερό ανακατεύοντας το μίγμα και στη συνέχεια τοποθετείται ξανά πίσω στη θέση του στο τραπέζι. Στο καλαμάκι (στην ιεραρχία των αντικειμένων) έγινε επισύναψη του script “Straw” (Παράρτημα D). Στο script δημιουργούνται δεκαεννιά πεδία διαφορετικών τύπων τα οποία φαίνονται συμπληρωμένα στο σχήμα 5.23 και αναλύονται παρακάτω:



Σχήμα 7.23: Τα πεδία του script “Straw”

- Πεδίο Gm: Δημιουργείται στη γραμμή 10 του κώδικα του script “Straw” και είναι τύπου gameobject. Παίρνει το αντικείμενο του υγρού των πιάτων “DisheSoapContainer_FBX”. Στη γραμμή 55-56, στη συνάρτηση onMouseDown() του καλαμακιού, ενεργοποιείται ο collider του καθώς είναι το επόμενο αντικείμενο για κλικ. Στη γραμμή 61 ο μετρητής sum παίρνει την τιμή 1 που σημαίνει πως έγινε κλικ στο καλαμάκι μία φορά. Όταν ο sum πάρει την τιμή 2 σημαίνει πως ολοκληρώθηκε το βήμα πέντε του πειράματος. Στη γραμμή 62 ξεκινάει ο timer τη μέτρηση. Όταν λοιπόν ο sum=2 (γραμμή 126) και έχουν γίνει όλα τα βήματα, απενεργοποιείται ο collider από το καλαμάκι καθώς είναι το τελευταίο αντικείμενο και ο παίκτης δεν πρέπει να κάνει κλικ σε κανένα αντικείμενο.
- Πεδία Str1-12: Δημιουργούνται στις γραμμές 12-23 και είναι τύπου gameobject. Το κάθε str παίρνει το αντίστοιχο σε αριθμό αντικείμενο από τα δώδεκα καλαμάκια. Τα δώδεκα καλαμάκια είναι τοποθετημένα μέσα στο ποτήρι με το νερό σε διαφορετικές θέσεις, ικανά να δώσουν την ψευδαίσθηση (ενεργοποιώντας το ένα και απενεργοποιώντας το άλλο) του ανακατέματος. Αρχικά αυτά τα καλαμάκια είναι απενεργοποιημένα στη συνάρτηση Start() στις γραμμές 39-49. Και στις δύο περιπτώσεις που γίνεται κλικ στο καλαμάκι (sum=1 και sum=2), ξεκινάει μια μέτρηση και ανα δύο κλάσματα του δευτερολέπτου ενεργοποιείται ένα καλαμάκι σε διαφορετική θέση και απενεργοποιείται το προηγούμενό του στις γραμμές 76-175. Στη γραμμή 127, αφού έχει γίνει κλικ για δεύτερη φορά στο καλαμάκι μετά από δύο δευτερόλεπτα, δηλαδή αφού ολοκληρωθεί το δεύτερο ανακάτεμα, καταστρέφεται το αντικείμενο του και τελειώνει το πείραμα.
- Πεδίο soapSystem: Δημιουργείται στη γραμμή 25 του κώδικα και είναι τύπου gameobject. Παίρνει το αντικείμενο “soapSystem” το οποίο περιέχει πολλούς gameobject → sphere και αποτελεί το υγρό του σαπουνιού στον βυθό του ποτηριού μετά την εισαγωγή του υγρού πιάτων στο ποτήρι. Συνεπώς, στη γραμμή 177 απενεργοποιείται προκειμένου να ενεργοποιηθεί στο επόμενο βήμα.
- Πεδία Cyl1 και Cyl2: Δημιουργούνται στις γραμμές 26-27 και είναι τύπου gameobject. Μόλις γίνει το δεύτερο κλικ στο καλαμάκι και τελειώσει ο timer, στη γραμμή 178 απενεργοποιείται ο κύλινδρος1 δηλαδή το γαλάζιο εσωτερικό του νερού, καθώς πλέον μετά την εισαγωγή του υγρού πιάτων το λάδι διαλύθηκε και το χρώμα του νερού αλλάζει (ενεργοποίηση κύλινδρου2 με πιο πράσινο χρώμα γραμμή 179).
- Πεδίο DynamicObj: Δημιουργείται στη γραμμή 28 και είναι τύπου gameobject. Παίρνει το αντικείμενο “Dynamic Objects” το οποίο είναι το σύνολο αντικειμένων από buoyancy spheres, οι οποίες αποτελούν το υγρό του λαδιού. Μόλις γίνει το δεύτερο κλικ στο καλαμάκι και τελειώσει ο timer, στη γραμμή 180 απενεργοποιείται το dynamicObj αφού το μίγμα διαλύεται και παύει το λάδι (dynamicObj) να επιπλέει.
- Πεδίο button symperasma: Δημιουργείται στη γραμμή 29 και είναι τύπου gameobject. Παίρνει το αντικείμενο του κουμπιού του συμπεράσματος. Αρχικά είναι απενεργοποιημένο στη συνάρτηση Start() στη γραμμή 50, ενώ μόλις ολοκληρωθούν τα πάντα στο πείραμα ενεργοποιείται το κουμπί στη γραμμή 181. Το buttonSymperasma χρησιμοποιώντας τη βιβλιοθήκη UnityEngine.SceneManagement προχωράει στην επόμενη σκηνή δηλαδή στο συμπέρασμα.
- Πεδίο TimerText: Υπάρχει μόνο για να βοηθάει τον timer. Δε φαίνεται πουθενά στο παιχνίδι γιατί είναι απενεργοποιημένο.

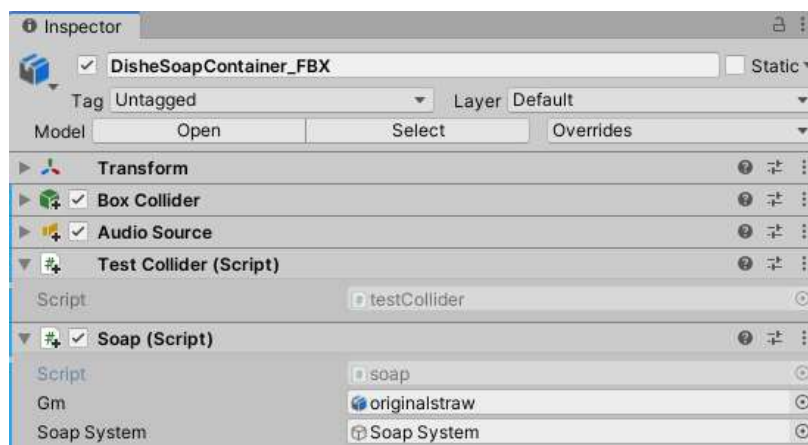
7.8.6 Πείραμα-Βήμα 4

Στο βήμα 4, μετά από κλικ του παίκτη πάνω στο υγρό σαπούνι για τα πιάτα, γίνεται η εισαγωγή του υγρού στο νερό του ποτηριού. Στο υγρό σαπούνι για τα πιάτα (στην ιεραρχία των αντικειμένων) έγινε επισύναψη του script “soap” (σχήμα 5.24).

```
soap.cs
C:\Users\user\Desktop>ptyxiaki>projects>final project>karantina>Assets>Scripts>3animation>soap.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class soap : MonoBehaviour
7 {
8     Collider m_Collider1; //o collider pou tha disable
9     Collider m_Collider2;
10    public GameObject gm;
11
12    public GameObject soapSystem;
13
14    void OnMouseDown()
15    {
16        m_Collider1=gm.GetComponent<Collider>();
17        m_Collider1.enabled = true;
18
19        m_Collider2=GetComponent<Collider>();
20        m_Collider2.enabled = false;
21
22        soapSystem.SetActive(true);
23    }
24 }
```

Σχήμα 7.24: Το script “soap”

Στο πεδίο “gm” έγινε drag and drop το gameobject καλαμάκι και στο πεδίο “soapSystem” το αντικείμενο του soapSystem όπως φαίνεται στο σχήμα 5.25.



Σχήμα 7.25: Πεδία script “soap”

Στο script “soap” στη συνάρτηση onMouseDown() μόλις γίνει κλικ στο υγρό σαπούνι των πιάτων:

- Στις γραμμές 16-17 ενεργοποιείται ο collider από το καλαμάκι καθώς είναι το επόμενο αντικείμενο για κλικ.
- Στις γραμμές 18-19 απενεργοποιείται ο collider του υγρού σαπουνιού των πιάτων προκειμένου να μη γίνει λανθασμένη επιλογή του στη συνέχεια.
- Στη γραμμή 22 ενεργοποιείται το soapSystem δίνοντας την αίσθηση ότι το υγρό σαπούνι για τα πιάτα, μετά την εισαγωγή του στο μίγμα, έκατσε στο πυθμένα του ποτηριού.

7.9 Σκηνή: symperasmataPeptiko

Η σκηνή αυτή φαίνεται στο σχήμα 5.26. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “PeptikoAbout”, “fe3-peptiko” και “testscene3”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται η παρατήρηση και τα συμπεράσματα που πρέπει να συμπληρώσει ο μαθητής στο βιβλίο.



Σχήμα 7.26: Σκηνή-symperasmataPeptiko

Στο κουμπί «Αρχική» στα components έχει προστεθεί script το οποίο χρησιμοποιώντας τη βιβλιοθήκη UnityEngine.SceneManagement πάει στην αρχική σκηνή η οποία βρίσκεται δεκαεννιά σκηνές πίσω (βλέπε σχήμα 1.10).

7.10 Επίλογος

Το πείραμα της σελίδας 66 του σχολικού βιβλίου, υλοποιείται με τη χρήση εκτεταμένης πραγματικότητας μέσω των παραπάνω έξι σκηνών. Μετά το πέρας της τελευταίας σκηνής του πειράματος, είναι θεμιτό να έχουν επιτευχθεί οι μαθησιακοί στόχοι που αναφέρθηκαν στο κεφάλαιο 5.3.

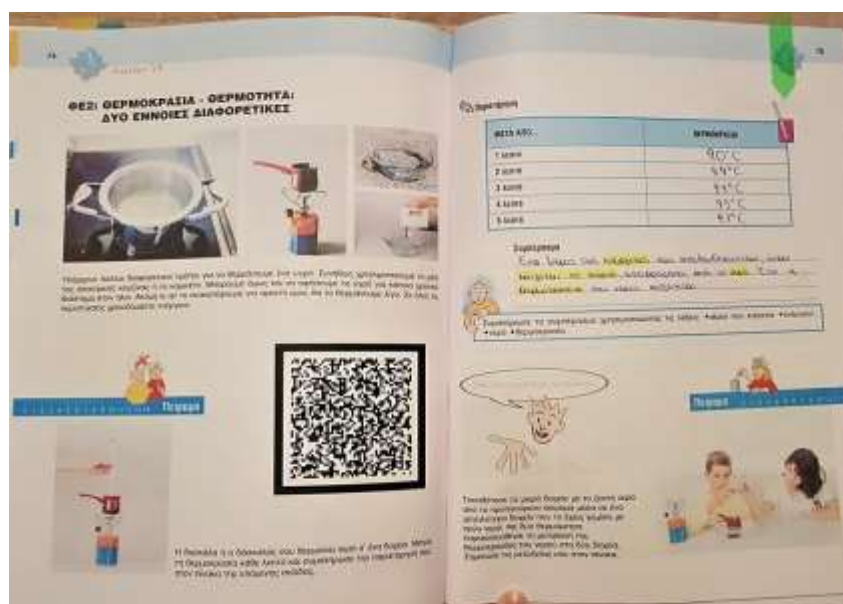
Κεφάλαιο 8ο: Εφαρμογή πειράματος για το κεφάλαιο Θερμότητα

8.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται οι μαθησιακοί στόχοι του πειράματος της σελίδας 74 του κεφαλαίου «Θερμότητα» του σχολικού βιβλίου. Επιπλέον, αναλύονται όλα τα βήματα της δημιουργίας των σκηνών που υλοποιούν το πείραμα, χρησιμοποιώντας τεχνολογίες εκτεταμένης πραγματικότητας.

8.2 Σελίδα βιβλίου 74

Στο κεφάλαιο «Θερμότητα» ανήκει η ενότητα «ΦΕ2: ΘΕΡΜΟΚΡΑΣΙΑ-ΘΕΡΜΟΤΗΤΑ: ΔΥΟ ΕΝΝΟΙΕΣ ΔΙΑΦΟΡΕΤΙΚΕΣ». Η ενότητα ξεκινάει με ένα εισαγωγικό ερέθισμα και ακολουθεί η πειραματική αντιμετώπιση στην οποία οι μαθητές καλούνται να παρατηρήσουν το μπρίκι με νερό, το οποίο είναι τοποθετημένο στο αναμμένο καμινέτο. Ζητείται από ένα μαθητή να μετρά τη θερμοκρασία κάθε λεπτό και να ανακοινώνει τη μέτρησή του στους συμμαθητές του. Οι υπόλοιποι μαθητές σημειώνουν τις μετρήσεις στον προβλεπόμενο χώρο στο βιβλίο τους. Σημειώνεται στους μαθητές να γράφουν δίπλα σε κάθε τιμή της θερμοκρασίας και τη μονάδα μέτρησης ($^{\circ}\text{C}$). Με το πείραμα αυτό οι μαθητές πρέπει να κατανοήσουν ότι ένα μέρος της ενέργειας, που απελευθερώνεται με την καύση του αερίου, απορροφάται από το νερό με αποτέλεσμα την αύξηση της θερμοκρασίας του. Οι μαθητές έχουν μελετήσει τις ενεργειακές μεταβολές, γι' αυτό γίνεται αναλυτική αναφορά στην ενέργεια, για να κατανοήσουν οι μαθητές ότι η προσφορά ενέργειας έχει ως αποτέλεσμα την αύξηση της θερμοκρασίας. Δε γίνεται ωστόσο ακόμη αναφορά στην έννοια «θερμότητα». Οι μαθητές πρέπει, για να μπορέσουν να διακρίνουν τις έννοιες «θερμοκρασία» και «θερμότητα», να κατανοήσουν σταδιακά ότι η «θερμότητα» είναι μια μορφή ενέργειας. Γίνεται αναφορά λοιπόν στη γενικότερη έννοια «ενέργεια», μέχρι οι μαθητές να εμπεδώσουν ότι η προσφορά ενέργειας σε ένα σώμα συνεπάγεται την αύξηση της θερμοκρασίας του. Αφού αυτό γίνει σαφές, στο επόμενο κεφάλαιο γίνεται ξεκάθαρο πότε η «ενέργεια» ονομάζεται «θερμότητα».

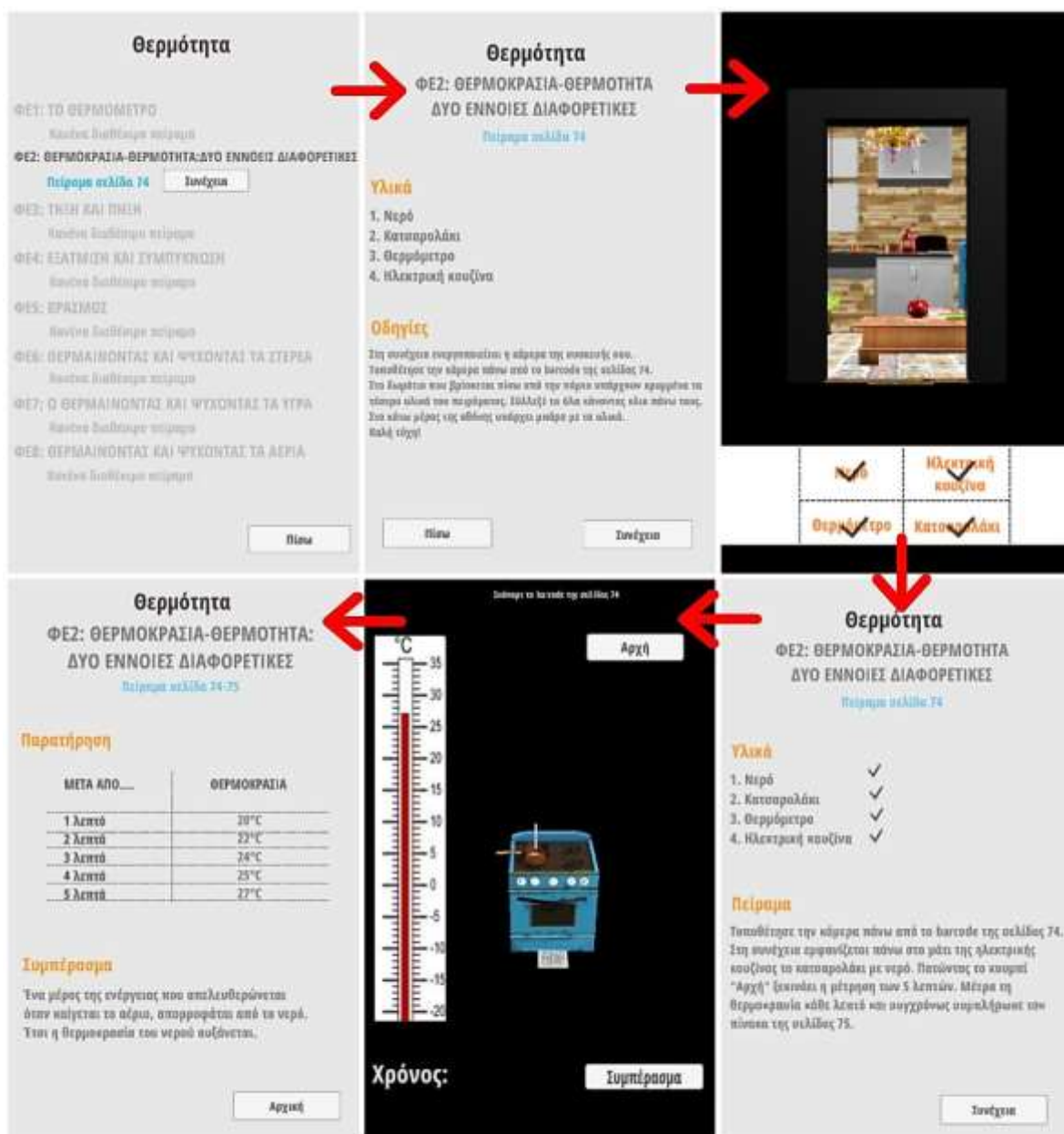


Σχήμα 8.1: Σελίδες βιβλίου 74-75

Επιπλέον, για την πραγματοποίηση του πειράματος εκτεταμένης πραγματικότητας, έχει προστεθεί στη σελίδα 74 εκτυπωμένο το image target16 από τη βάση high_class.

8.3 Οι σκηνές του πειράματος – Θερμότητα

Το πείραμα εκτεταμένης πραγματικότητας αποτελείται από έξι σκηνές όπως αυτές φαίνονται στο σχήμα 6.2. Αρχικά ο μαθητής-παίκτης, αφού επιλέξει το κεφάλαιο της θερμότητας (σκηνή: startScene), εμφανίζονται τα διαθέσιμα πειράματα των ενοτήτων του κεφαλαίου (σκηνή: ThermtitaAbout). Αφού επιλέξει ποιο πείραμα θέλει να υλοποιηθεί, εμφανίζονται οι οδηγίες του πειράματος καθώς και τα υλικά τα οποία πρέπει να συλλέξει (σκηνή: fe2-thermotita). Στη συνέχεια, ενεργοποιείται η κάμερα της συσκευής του, και αφού σκανάρει το qrcode της σελίδας 74, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας και μια μπάρα με τα εναπομείναντα υλικά τα οποία βρίσκονται πίσω από αυτή, σε ένα εικονικό περιβάλλον (σκηνή: VR2).



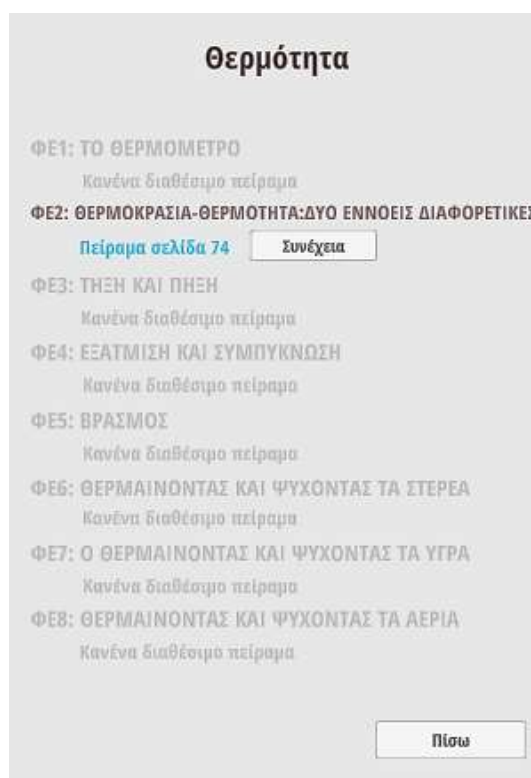
Σχήμα 8.2: Σκηνές πειράματος σελίδας 74

Μόλις συλλέξει όλα τα υλικά της μπάρας, εμφανίζονται οδηγίες (σκηνή: testScene2) για το πείραμα που θα ακολουθήσει. Σκανάροντας ξανά το qrcode της σελίδας 74, εμφανίζεται η ηλεκτρική κουζίνα πάνω στην οποία βρίσκεται τοποθετημένο το κατσαρολάκι με το νερό και το θερμόμετρο (σκηνή: 2.animation). Πατώντας το κουμπί «Αρχή» ξεκινάει να μετράει ο χρόνος για πέντε λεπτά και ανάβει η ηλεκτρική κουζίνα. Η θερμοκρασία στο θερμόμετρο αλλάζει αναλόγως. Μόλις περάσουν τα πέντε λεπτά, εμφανίζεται στο κάτω μέρος της οθόνης το κουμπί “συμπεράσμα” το οποίο οδηγεί τον παίκτη στην επόμενη σκηνή η οποία αποτελείται από τη συμπληρωμένη παρατήρηση του βιβλίου καθώς και του συμπεράσματος (σκηνή: symperasmataThermotita). Μετά την ολοκλήρωση του παραπάνω πειράματος εκτεταμένης πραγματικότητας, οι μαθησιακοί στόχοι που είναι θεμιτό να επιτευχθούν είναι οι εξής:

- Να αναφέρουν οι μαθητές διάφορους τρόπους με τους οποίους μπορούμε να αυξήσουμε τη θερμοκρασία ενός σώματος.
- Να διαπιστώσουν οι μαθητές πειραματικά ότι, όταν ένα σώμα απορροφά θερμότητα, η θερμοκρασία του αυξάνεται.
- Να διαπιστώσουν οι μαθητές πειραματικά ότι η θερμότητα ρέει από τα θερμά στα ψυχρά σώματα.

8.4 Σκηνή: ThermotitaAbout

Η σκηνή αυτή φαίνεται στο σχήμα 6.3. Η ιεραρχία των αντικειμένων της αποτελείται από: μια main camera, ένα directional light για το σωστό φωτισμό της σελίδας, ένα event system για τη λειτουργία των κουμπιών και των εναλλαγών των σκηνών και ένα canva στον οποίο υπάρχουν πολλά textMeshPro για την εισαγωγή των κειμένων καθώς και δύο buttons. Με αυτά γίνεται η εναλλαγή των σκηνών, μέσω ενός script παρόμοιου με του script “menu”, το οποίο αποτελείται από τις μεθόδους “next()” και “back()” που χρησιμοποιούν τη βιβλιοθήκη UnityEngine.SceneManagement.



Σχήμα 8.3: Σκηνή “ThermotitaAbout”

8.5 Σκηνή: Fe2-thermotita

Η σκηνή αυτή φαίνεται στο σχήμα 6.4. Η ιεραρχία των αντικειμένων της είναι ίδια με αυτή της σκηνής “PeritikoAbout”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro και το περιεχόμενό τους είναι διαφορετικό.



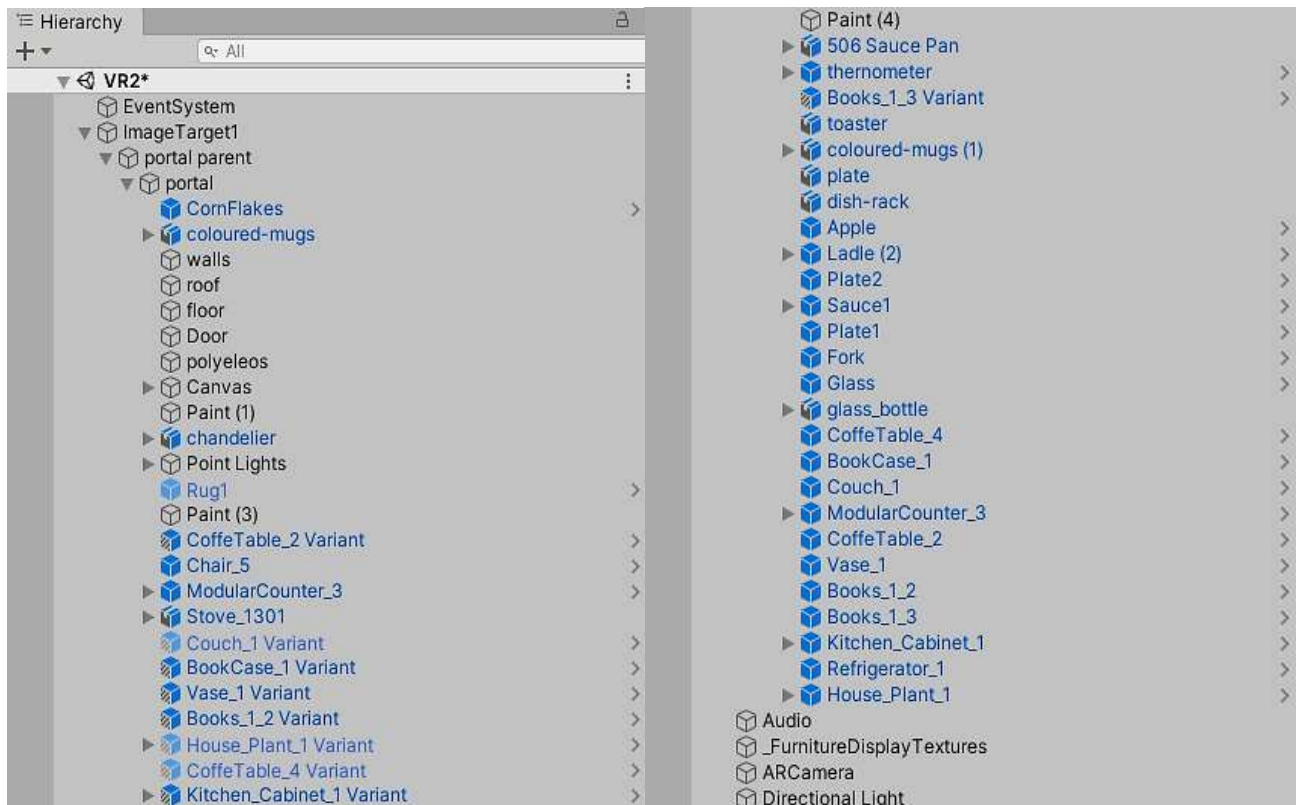
Σχήμα 8.4: Σκηνή fe2-thermotita

8.6 Σκηνή: VR2

Σε αυτή τη σκηνή, ο παίκτης μετά από προτροπή των οδηγιών της προηγούμενης σκηνής, σκανάρει το qrcode που βρίσκεται στη σελίδα 74 του σχολικού βιβλίου. Μόλις η Vuforia Camera το εντοπίσει, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας πάνω από το qrcode (σχήμα 6.5). Η πόρτα επαυξημένης πραγματικότητας που εμφανίζεται, έχει την ίδια λειτουργικότητα με τις πόρτες εικονικής πραγματικότητας των προηγούμενων σκηνών “VR”. Για να φανεί το εικονικό περιβάλλον που υπάρχει πίσω της, ο παίκτης πρέπει να εισέλθει με τη συσκευή του μέσα στην πόρτα. Μόλις γίνει αυτό, ακολουθεί ολική εμβύθιση του παίκτη σε ένα δωμάτιο εικονικής πραγματικότητας, στο οποίο καλείται να βρει τα τρία υλικά του πειράματος. Στο κάτω μέρος της οθόνης υπάρχει μπάρα στην οποία φαίνονται τα υπολειπόμενα υλικά. Η ιεραρχία των αντικειμένων της σκηνής VR2 φαίνεται αναλυτικά στο σχήμα 6.6. Περιληπτικά, αποτελείται από τα ίδια βασικά αντικείμενα των σκηνών “VR” με τη μόνη διαφορά τα διαφορετικά εικονικά αντικείμενα που υπάρχουν και αποτελούν το ύψος του δωματίου. Πριν τη δημιουργία της πόρτας, έγινε δημιουργία αντικειμένου “image target” και η επιλογή του marker20 από τη βάση highclass. Το παιδί του image target είναι ένα empty “portal parent” προκειμένου τα περιεχόμενα αυτού να εμφανιστούν εφόσον η arcamera εντοπίσει το imagemarker16. Τα δομικά στοιχεία του δωματίου είναι οι τοίχοι, η πόρτα, το πάτωμα και η οροφή.



Σχήμα 8.5: Πόρτα επαυξημένης πραγματικότητας VR2 σκηνής



Σχήμα 8.6: Ιεραρχία αντικειμένων VR2

8.6.1 Εικονικό περιβάλλον και εικονικά αντικείμενα VR2

Το εσωτερικό του δωματίου μοιάζει με σαλοκουζίνα. Αποτελείται από δεκάδες assets, .fbx και .obj που αποθηκεύτηκαν από το unity asset store και από ιστοσελίδες όπως αυτές αναφέρθηκαν στο αντίστοιχο κεφάλαιο της σκηνής “VR4”. Στο σχήμα 6.7 φαίνεται το εσωτερικό του δωματίου με λεπτομέρειες.



Σχήμα 8.7: Δωμάτιο VR2

Τα τέσσερα υλικά που ο παίκτης πρέπει να συλλέξει προκειμένου να προχωρήσει στο πείραμα φαίνονται στο σχήμα 6.8. Τα υλικά που ο παίκτης πρέπει να μαζέψει προκειμένου να προχωρήσει στο πείραμα, είναι τοποθετημένα μέσα στο εικονικό περιβάλλον με τρόπο έξυπνο, τέτοιο ώστε να μην είναι φανερά εκ πρώτης όψεως αλλά συγχρόνως να είναι διακριτά.



Σχήμα 8.8: Υλικά πειράματος (Θερμότητα)

Για τον ήχο της σκηνής “VR2” χρησιμοποιήθηκε ένα `gameobject empty` με όνομα “Audio”. Στα `components` του `empty “Audio”` προστέθηκε ένα “Audio source” με τις ίδιες ρυθμίσεις των προηγούμενων σκηνών “VR5” αλλάζοντας το αρχείο ήχου για να είναι μοναδικό. Ένα ακόμα `component` που προστέθηκε στο `audio` είναι ένα `script` για να καταστρέφει το αντικείμενο του ήχου στη σκηνή `testscene2`, δηλαδή την επόμενη της VR2.

8.6.2 Εναλλαγή σκηνής VR2

Στα τέσσερα αντικείμενα που πρέπει να συλλέξει ο παίκτης έχουν προστεθεί τα ίδια `components` όπως και στις προηγούμενες σκηνές VR:

- Box collider
- Audio source:
- Το `script “Dotvrdoor”` (σχήμα 3.11)

Η μπάρα στο κάτω μέρος του παιχνιδιού με τα υπολειπόμενα παιχνίδια είναι ο `canvas` από την ιεραρχία αντικειμένων και αποτελείται από τέσσερα `textmesh Pro` με τα ονόματα των αντικειμένων και αντίστοιχα τέσσερις εικόνες (τικ) που εμφανίζονται μόλις ο παίκτης επιλέξει ένα αντικείμενο από αυτά. Στα `components` του `canva` προστέθηκε το `script “CanvasScriptVR2”`, το οποίο έχει την ίδια λειτουργικότητα με το `script “canvascriptvrfour”` (σχήμα 3.13) και το `script “csSIXx”` (σχήμα 4.10). Με το `script “CanvasScriptVR2”`, όταν ο μετρητής `sum` φτάσει των αριθμό τέσσερα (δηλαδή όταν

όλα τα αντικείμενα βρεθούν), καλεί την επόμενη σκηνή μέσω της βιβλιοθήκης UnityEngine.SceneManagement.

8.7 Σκηνή: TestScene2

Η σκηνή αυτή φαίνεται στο σχήμα 6.9. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “ThermotitaAbout” και “fe2-thermotita”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται οδηγίες για το πείραμα που ακολουθεί.



Σχήμα 8.9: Σκηνή testscene2

8.8 Σκηνή: 2.animation

Σε αυτή τη σκηνή πραγματοποιείται το πείραμα της σελίδας 74 με τη χρήση εκτεταμένης πραγματικότητας. Στην αρχική οθόνη υπάρχει η ηλεκτρική κουζίνα, πάνω στο μάτι της οποίας είναι τοποθετημένο το κατσαρολάκι. Στο κατσαρολάκι υπάρχει νερό, και μέσα στο νερό το θερμόμετρο που πρόκειται να μετράει τη θερμοκρασία καθώς περνούν τα λεπτά. Στο πάνω μέρος της οθόνης υπάρχει το κουμπί «Αρχή». Μόλις ο παίκτης κάνει κλικ στο κουμπί «Αρχή», ξεκινάει να μετράει ο χρόνος στο κάτω αριστερό μέρος, καθώς ανάβει η φωτιά του ματιού της ηλεκτρικής κουζίνας που βρίσκεται το κατσαρολάκι με το νερό και το θερμόμετρο. Στο Αριστερό μέρος της οθόνης υπάρχει σε μεγέθυνση το θερμόμετρο διότι το θερμόμετρο μέσα στο νερό είναι αρκετά μικρό και γυρίζοντας το κινητό από άλλη οπτική γωνία, η θερμοκρασία φαίνεται διαφορετική από ότι είναι στην πραγματικότητα. Καθώς ο χρόνος κυλάει, το θερμόμετρο δείχνει κάθε λεπτό τις θερμοκρασίες: 1 λεπτό→20°C, 2 λεπτά→22°C, 3 λεπτά→24°C, 4 λεπτά→25°C και 5 λεπτά→27°C (σύμφωνα με το βιβλίο δασκάλου). Όταν ο χρόνος ξεπεράσει τα πέντε λεπτά, τότε εμφανίζεται στο κάτω μέρος της οθόνης το κουμπί «Συμπέρασμα» το οποίο οδηγεί στην επόμενη σκηνή. Αυτή αποτελείται από το

Εφαρμογή πειράματος για το κεφάλαιο Θερμότητα

συμπέρασμα του πειράματος καθώς και τον πίνακα της σελίδας 75 του σχολικού βιβλίου συμπληρωμένο. Στο σχήμα 6.10 φαίνονται οι οθόνες του πειράματος κατά τη διάρκεια των πέντε λεπτών.

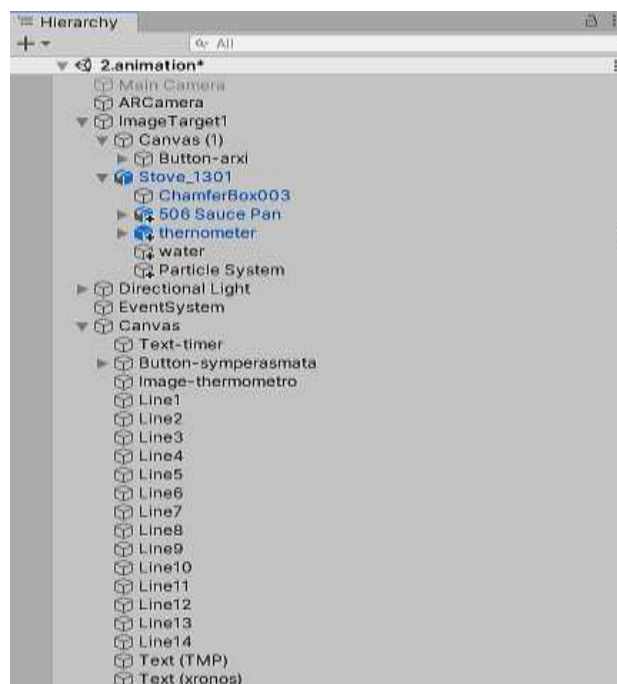


Σχήμα 8.10: Οι οθόνες του πειράματος-2.animation

8.8.1 Ιεραρχία αντικειμένων – 2.animation

Η ιεραρχία των αντικειμένων της σκηνης 2.animation (σχήμα 6.11) αποτελείται από:

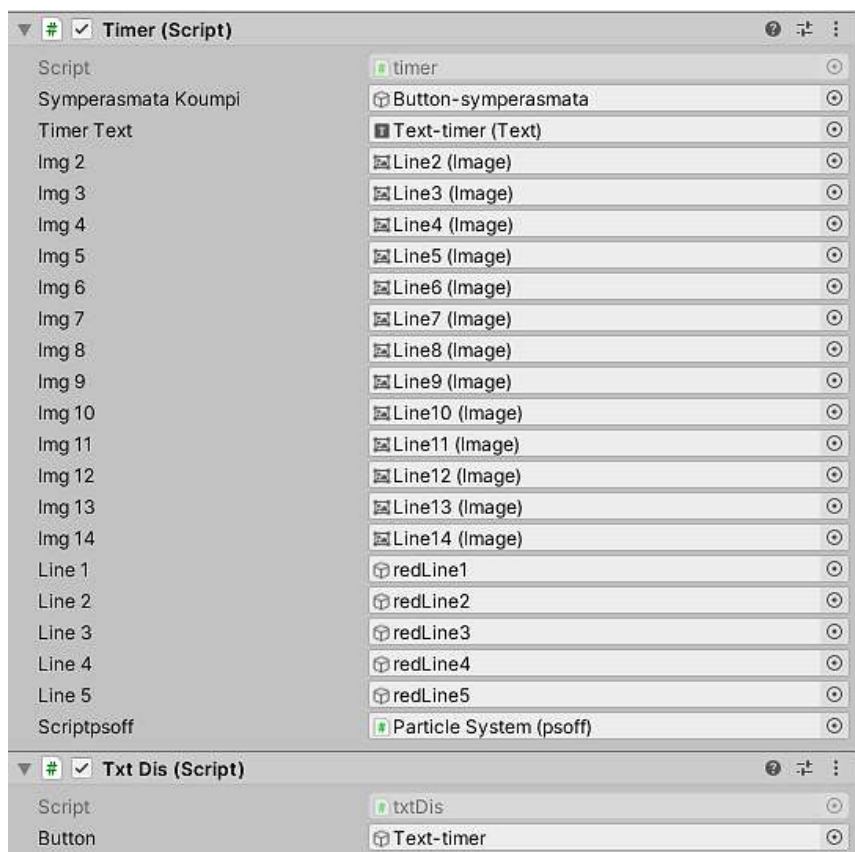
- Ένα event system.
- Μία ARCamera.
- Τρία directional lights για σωστό φωτισμό περιμετρικά της πόρτας και δεκαπέντε point lights εσωτερικά.
- Το image target16 το οποίο εμπεριέχει :
 - Την ηλεκτρική κουζίνα.
 - Το κατσαρολάκι.
 - Το θερμόμετρο το οποίο με τη σειρά του αποτελείται από έξι red lines οι οποίες είναι τοποθετημένες κατάλληλα για να δείχνουν συγκεκριμένες τιμές θερμοκρασίας.
 - Έναν cylinder “water” ο οποίος είναι το νερό μέσα στο κατσαρολάκι.
 - Ένα Particle system το οποίο αποτελεί τη φωτιά στο μάτι της κουζίνας.
 - Ένα canva με το button «Αρχή».
- Ένα canva ο οποίος εμπεριέχει :
 - Ένα textMeshPro για timer.
 - Ένα button «Συμπέρασμα».
 - Μια image η οποία είναι το θερμόμετρο σε μεγέθυνση.
 - Δεκατέσσερις red lines (images) οι οποίες είναι τοποθετημένες κατάλληλα μέσα στα θερμόμετρα να δείχνουν θερμοκρασίες.
 - Ένα textMeshPro για την προτροπή του χρήστη «Σκάνανε το qrcode της σελίδας 74» στο πάνω μέρος της οθόνης.
 - Ένα textMeshPro για την εμφάνιση του χρόνου.



Σχήμα 8.11: Ιεραρχία αντικειμένων “2.animation”

8.8.2 Η λειτουργία του πειράματος μέσα από τα scripts (Θερμότητα)

Αρχικά στο Text-timer από την ιεραρχία των αντικειμένων της σκηνής έγινε προσθήκη στα components δύο scripts, το “Timer” και το “Txt Dis”. Στο σχήμα 6.12 φαίνεται ο inspector του Text-timer με συμπληρωμένα όλα τα πεδία που δημιουργούνται στα παραπάνω δύο script.



Σχήμα 8.12: Ο inspector του Text-timer

Στο script “Timer” (Παράρτημα Ε) δημιουργούνται δώδεκα πεδία διαφορετικών τύπων τα οποία φαίνονται συμπληρωμένα στο σχήμα 6.12 και αναλύονται παρακάτω:

- Αρχικά δημιουργεί τα δώδεκα πεδία του σχήματος 6.12. Στη συνάρτηση void Startt() (γραμμή 38) ξεκινάει η μέτρηση (γραμμή 40). Αυτή η συνάρτηση δεν είναι ίδια με τις void Start() διότι δεν πρέπει να ξεκινήσει η συνάρτηση μόλις ξεκινήσει η σκηνή, αλλά μόλις πατηθεί το κουμπί «Αρχή». Για να ξεκινήσει η λειτουργία του script πρέπει να κληθεί η συνάρτηση void Startt()).
- Πεδίο Timer text: δημιουργείται το πεδίο “timerText” τύπου Text (γραμμή 10) προκειμένου να εμφανίζεται σε αυτό το κείμενο η τιμή του χρόνου. Αυτό πραγματοποιείται μέσω της συνάρτησης void UpdateTime() (γραμμή 113-116).
- Πεδία img2-14: δημιουργούνται τα πεδία “img2-14” τύπου Image (γραμμές 13-25). Τα πεδία αυτά συμπληρώνονται με τις εικόνες που αποτελούν μια κόκκινη γραμμή στο θερμόμετρο της οθόνης και είναι ανάλογα τοποθετημένες να δείχνουν τη σωστή θερμοκρασία. Η κάθε γραμμή ενεργοποιείται ανάλογα τη χρονική στιγμή στις γραμμές 50-104. Παραδείγματος χάρη, στη γραμμή 70, μόλις ο χρόνος φτάσει τα εξήντα δευτερόλεπτα ενεργοποιείται η εικόνα7 (γραμμή 72) η οποία αντιστοιχεί σε θερμοκρασία 20°C. Χρησιμοποιούνται παραπάνω από πέντε εικόνες (πέντε λεπτά→πέντε θερμοκρασίες) για περισσότερη

λεπτομέρεια στο θερμόμετρο (λεπτομέρεια δευτερολέπτων) και για να ανεβαίνει σταδιακά η θερμοκρασία.

- Πεδία Line1-5: δημιουργούνται τα πεδία “Line1-5” τύπου GameObject (γραμμές 27-31). Τα πεδία αυτά συμπληρώνονται με τις redlines που αποτελούν τις κόκκινες γραμμές στο θερμόμετρο που βρίσκεται μέσα στο νερό. Η κάθε γραμμή ενεργοποιείται ανάλογα τη χρονική στιγμή στις γραμμές 70-105.
- Πεδίο symperasmataKoumpri: δημιουργείται στη γραμμή 9 και είναι τύπου GameObject. Αποτελεί το κουμπί των συμπερασμάτων που πρέπει να ενεργοποιηθεί (γραμμή 106) εφόσον περάσουν τα πέντε λεπτά (300 δευτερόλεπτα) στη γραμμή 102.
- Πεδίο scriptpsoff: δημιουργείται στη γραμμή 36 και είναι τύπου psuff (script που θα αναλυθεί παρακάτω). Στη γραμμή 48 απενεργοποιείται, όταν δηλαδή έχει τελειώσει ο χρόνος στα 301 δευτερόλεπτα (γραμμή 44). Το scriptpsoff αποτελείται από το ParticleSystem δηλαδή από τη φωτιά.

Στο script “Txt Dis” (σχήμα 6.13) δημιουργείται το πεδίο “button” τύπου GameObject και παίρνει την τιμή του αντικειμένου TimerText. Στη γραμμή 10 το απενεργοποιεί προκειμένου να ενεργοποιηθεί η λειτουργία του μόνο όταν πατηθεί το κουμπί «Αρχή».

```

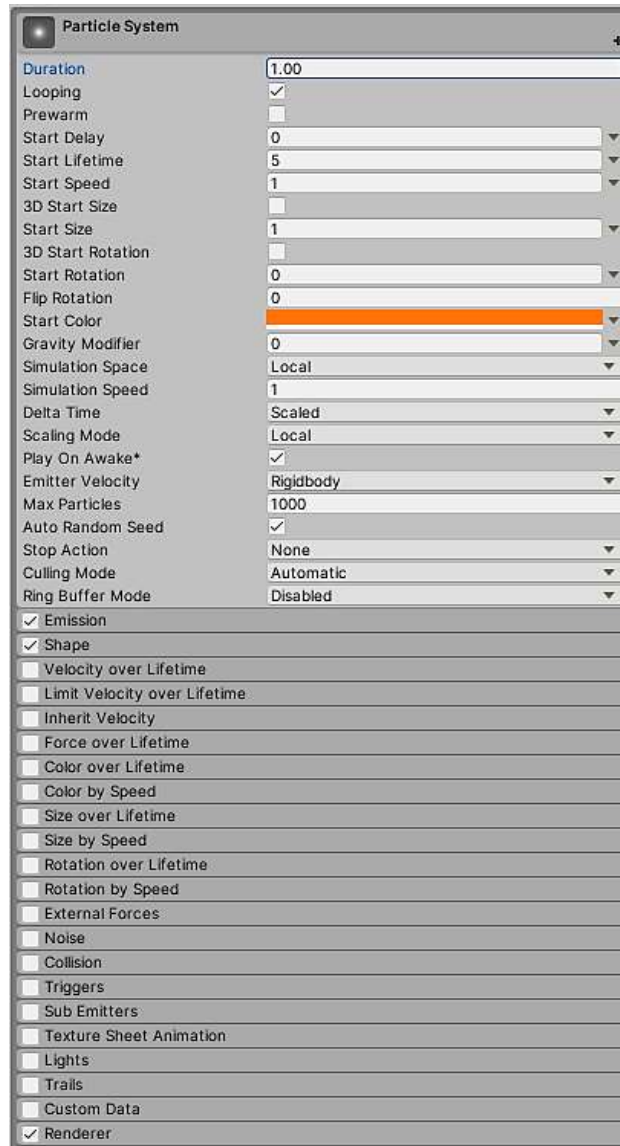
txtDis.cs
C:\Users\user\Desktop\ptyxiaki\projects\final project\karantina\Assets\Scripts\2.animation> txtDis.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class txtDis : MonoBehaviour
6 {
7     public GameObject button;
8     void Start()
9     {
10        button.SetActive(false);
11    }
12
13 }
    
```

Σχήμα 8.13: Το script “Txt Dis”

Το ParticleSystem από την ιεραρχία αντικειμένων αποτελείται από τα εξής components:

- Ένα Particle System του οποίου οι ρυθμίσεις φαίνονται στο σχήμα 6.13.
- Ένα Audio Source το οποίο έχει ως audioClip ένα αρχείο ήχου «βρασμού νερού» και είναι απενεργοποιημένο το Play On Awake για να ενεργοποιηθεί ο ήχος όταν πατηθεί το κουμπί «Αρχή» και ανάψει η φωτιά.
- Το script “psuff” σχήμα 6.14. Αυτό αποτελείται από τρεις συναρτήσεις.
 - Την void Start() η οποία μόλις ξεκινάει η σκηνή απενεργοποιεί το particleSystem (γραμμή 10).
 - Την public void enabletheps() η οποία καλείται στην onClickEvent του κουμπιού «Αρχή» και ενεργοποιεί το ParticleSystem (γραμμή 14) και το αρχείο ήχου που είναι επισημασμένο (γραμμή 15).
 - Την void disabletheps() η οποία καλείται όπως είδαμε παραπάνω από το script “Timer” στη γραμμή 48 για να απενεργοποιηθεί το particleSystem μόλις περάσουν τα πέντε λεπτά, δηλαδή να σβήσει η φωτιά.

Εφαρμογή πειράματος για το κεφάλαιο Θερμότητα

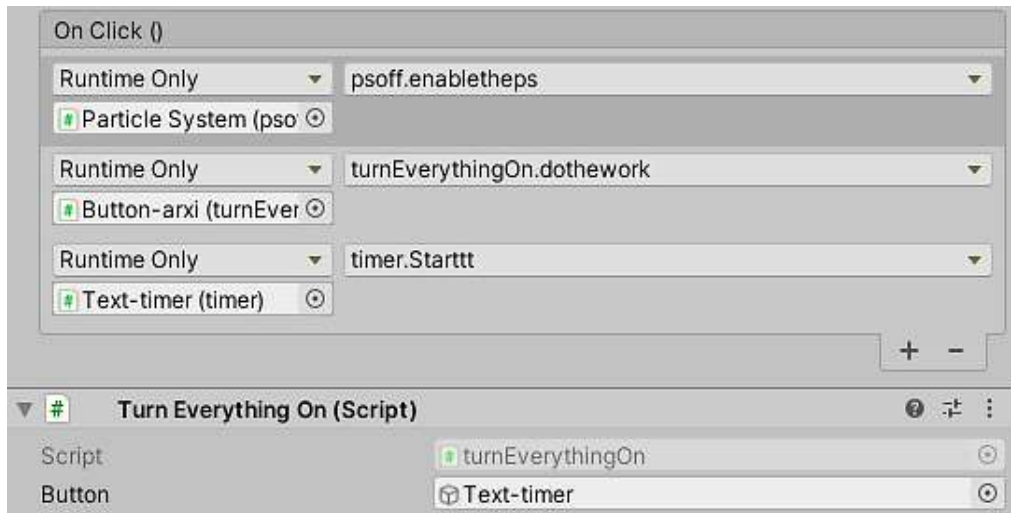


Σχήμα 8.14: Οι ρυθμίσεις του Particle System

```
psoff.cs
C:\Users\user\Desktop\physikiaki\projects\final project\karantina\Assets\Scripts\2.animation\psoff.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class psoff : MonoBehaviour
6 {
7
8     void Start()
9     {
10         gameObject.GetComponent<ParticleSystem>().enableEmission = false;
11     }
12     public void enablethepts()
13     {
14         gameObject.GetComponent<ParticleSystem>().enableEmission = true;
15         gameObject.GetComponent<AudioSource>().Play();
16     }
17     public void disablethepts()
18     {
19         gameObject.GetComponent<ParticleSystem>().enableEmission = false;
20     }
21
22 }
```

Σχήμα 8.15: Το script “psoff”

Όλα τα παραπάνω δεν είναι αρκετά χωρίς την παραμετροποίηση του κουμπιού «Αρχή». Ο inspector του button «Αρχή» φαίνεται στο σχήμα 6.16.



Σχήμα 8.16: Ο inspector του button «Αρχή»

Στα components του κουμπιού «Αρχή» έχει προστεθεί το script “Turn Everything On” (σχήμα 6.17). Αυτό αποτελείται από τη συνάρτηση void dothework(). Η συνάρτηση αυτή δέχεται στο πεδίο button το αντικείμενο της ιεραρχίας “Text-timer” και στη γραμμή 11 το ενεργοποιεί. Στη γραμμή 12 καταστρέφει μετά από δύο δευτερόλεπτα το κουμπί «Αρχή» διότι μετά την επιλογή του δεν έχει λόγο ύπαρξης. Η συνάρτηση καλείται στο onClickEvent του button «Αρχή». Επίσης, στο onClickEvent του button «Αρχή» καλείται και η συνάρτηση Starttt() του script “timer” για να ξεκινήσει τη μέτρηση του χρόνου καθώς και η συνάρτηση enablethepps() του script “psoff” η οποία ενεργοποιεί το ParticleSystem και το αρχείο ήχου.

```

turnEverythingOn.cs X
C:\Users\user\Desktop\ptyxiaki\projects\final project\karantina\Assets\Scripts\2.animation\turnEverythingOn.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class turnEverythingOn : MonoBehaviour
6  {
7      public GameObject button;
8
9      public void dothework()
10     {
11         button.SetActive(true);
12         Destroy(gameObject, 2);
13     }
14
15 }

```

Σχήμα 8.17: Το script “Turn Everything On”

Τέλος, στα components του κουμπιού συμπεράσμα, έχει προστεθεί ένα script το οποίο χρησιμοποιώντας τη βιβλιοθήκη UnityEngine.SceneManagement προχωράει στην επόμενη σκηνή, η οποία βρίσκεται μία θέση μετά (βλέπε σχήμα 1.10), και αποτελεί το συμπέρασμα του πειράματος.

8.9 Σκηνή: symperasmataThermotita

Η σκηνή αυτή φαίνεται στο σχήμα 6.17. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “ThermotitaAbout”, “fe2-thermotita” και “testscene2”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται τα συμπεράσματα καθώς και ο πίνακας της σελίδας 75 του σχολικού βιβλίου, που πρέπει να συμπληρώσει ο μαθητής σε αυτό.

Θερμότητα
ΦΕ2: ΘΕΡΜΟΚΡΑΣΙΑ-ΘΕΡΜΟΤΗΤΑ:
ΔΥΟ ΕΝΝΟΙΕΣ ΔΙΑΦΟΡΕΤΙΚΕΣ
Πείραμα σελίδα 74-75

Παρατήρηση

ΜΕΤΑ ΑΠΟ....	ΘΕΡΜΟΚΡΑΣΙΑ
1 λεπτό	20°C
2 λεπτά	22°C
3 λεπτά	24°C
4 λεπτά	25°C
5 λεπτά	27°C

Συμπέρασμα
Ένα μέρος της ενέργειας που απελευθερώνεται όταν καίγεται το αέριο, απορροφάται από το νερό. Έτσι η θερμοκρασία του νερού αυξάνεται.

Αρχική

Σχήμα 6.17: Σκηνή-symperasmataThermotita

Στο κουμπί «Αρχική» στα components έχει προστεθεί script το οποίο χρησιμοποιώντας τη βιβλιοθήκη UnityEngine.SceneManagement πάει στην αρχική σκηνή η οποία βρίσκεται έντεκα σκηνές πίσω (βλέπε σχήμα 1.10).

8.10 Επίλογος

Το πείραμα της σελίδας 74 του σχολικού βιβλίου, υλοποιείται με τη χρήση εκτεταμένης πραγματικότητας μέσω των παραπάνω έξι σκηνών. Μετά το πέρας της τελευταίας σκηνής του πειράματος, είναι θεμιτό να έχουν επιτευχθεί οι μαθησιακοί στόχοι που αναφέρθηκαν στο κεφάλαιο 6.3.

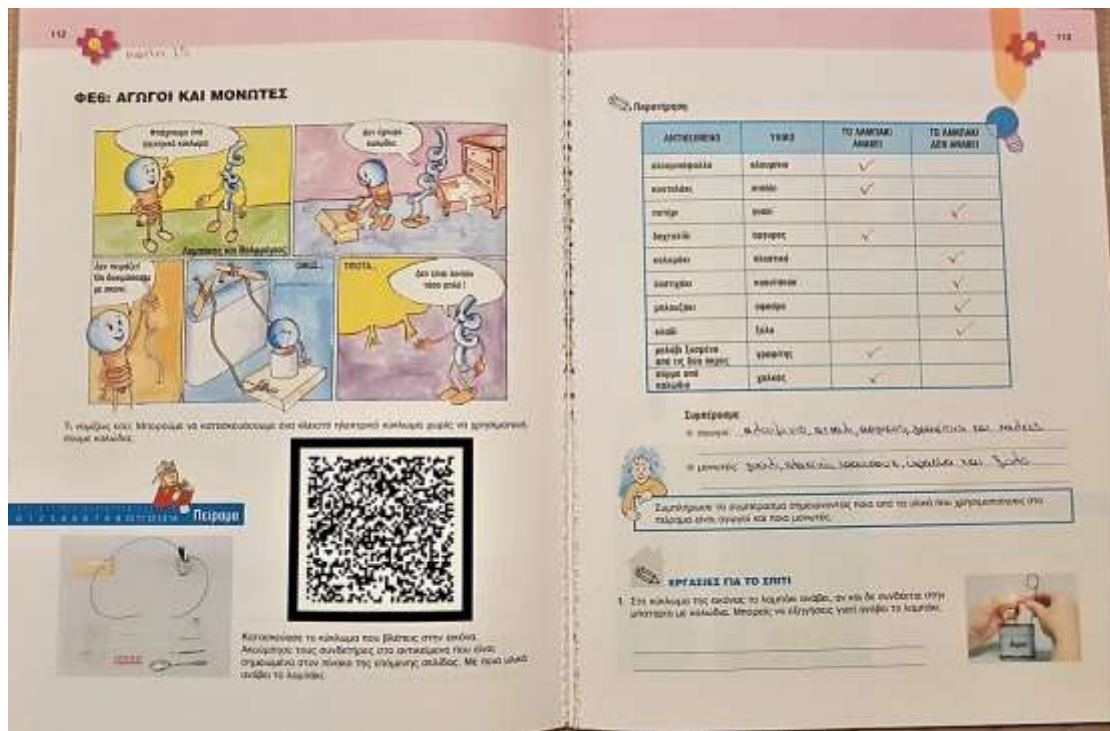
Κεφάλαιο 9ο: Εφαρμογή πειράματος για το κεφάλαιο Ηλεκτρισμός

9.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται οι μαθησιακοί στόχοι του πειράματος της σελίδας 112 του κεφαλαίου «Ηλεκτρισμός» του σχολικού βιβλίου. Επιπλέον, αναλύονται όλα τα βήματα της δημιουργίας των σκηνών που υλοποιούν το πείραμα, χρησιμοποιώντας τεχνολογίες εκτεταμένης πραγματικότητας.

9.2 Σελίδα βιβλίου 112

Στο κεφάλαιο «Ηλεκτρισμός» ανήκει η ενότητα «ΦΕ6: ΑΓΩΓΟΙ ΚΑΙ ΜΟΝΩΤΕΣ». Η ενότητα ξεκινάει με ένα εισαγωγικό ερέθισμα και ακολουθεί η πειραματική αντιμετώπιση. Το εισαγωγικό ερέθισμα στην ενότητα αυτή δίνεται μέσα από ένα κόμικ με ήρωες το Λαμπάκη και το Βολφράμιο. Ανάλογα κόμικ παρουσιάζονται στην εισαγωγή αρκετών Φύλλων Εργασίας της Ενότητας «Ηλεκτρισμός» τόσο στην Ε΄ όσο και στη Στ΄ Δημοτικού. Στην ενότητα 3 του κεφαλαίου, δόθηκε στους μαθητές η πληροφορία ότι το υλικό από το οποίο είναι κατασκευασμένο το συρματάκι στους λαμπτήρες πυρακτώσεως ονομάζεται βολφράμιο. Στη συνέχεια ζητείτε από τους μαθητές να σχολιάσουν τη μορφή του Βολφράμιου στο κόμικ και να διαβάσουν τους διαλόγους προκειμένου να περιγράψουν το πρόβλημα που αντιμετωπίζουν ο Λαμπάκης και ο Βολφράμιος. Στο τέλος του εισαγωγικού ερεθίσματος, οι μαθητές καλούνται να διαβάσουν το ερώτημα που είναι σημειωμένο κάτω από το κόμικ και ο δάσκαλος προκαλεί τη διατύπωση υποθέσεων σχετικά με υλικά τα οποία μπορούν να χρησιμοποιηθούν σε ένα κλειστό ηλεκτρικό κύκλωμα. Ακολουθεί η πειραματική αντιμετώπιση της σελίδας 112 (σχήμα 71.), με την οποία οι μαθητές διαπιστώνουν ότι άλλα υλικά επιτρέπουν τη ροή του ηλεκτρικού ρεύματος, ενώ άλλα όχι.

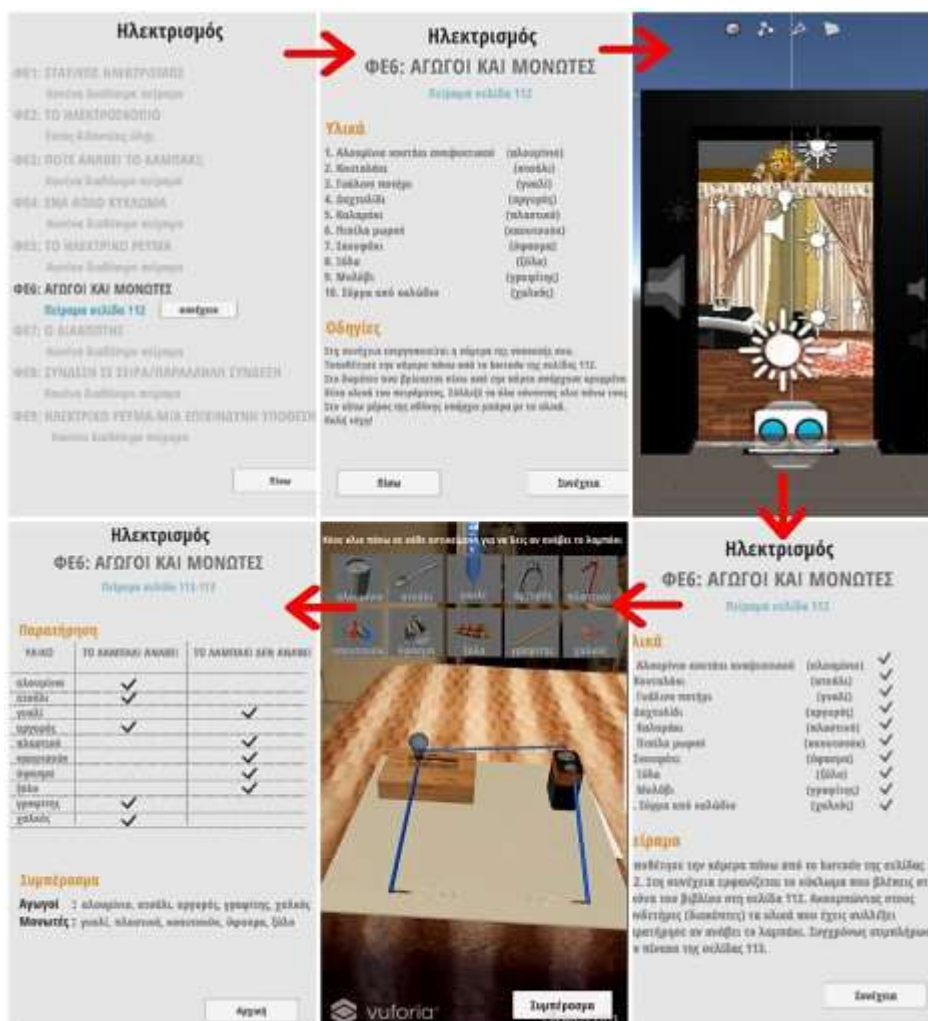


Σχήμα 9.1: Σελίδες βιβλίου 112-113

Μετά, οι μαθητές καλούνται να συμπληρώσουν τον πίνακα σύμφωνα με την παρατήρησή τους. Ταξινομούν τα υλικά σε δύο κατηγορίες, ανάλογα με το αν το λαμπάκι ανάβει ή όχι. Ο δάσκαλος, προκαλεί συζήτηση στην τάξη, μέσα από την οποία οι μαθητές διατυπώνουν το συμπέρασμα. Εισάγει επιπλέον τους όρους «αγωγός» και «μονωτής» και τους εξηγεί στους μαθητές. Στη συνέχεια τους προτρέπει να ταξινομήσουν τα υλικά που μελέτησαν στο παραπάνω πείραμα σε αγωγούς και μονωτές. Οι αναφορές γίνονται αυστηρά στα υλικά και όχι στα αντικείμενα. Επιπλέον, για την πραγματοποίηση του πειράματος εκτεταμένης πραγματικότητας, έχει προστεθεί στη σελίδα 112 εκτυπωμένο το image target15 από τη βάση high_class.

9.3 Οι σκηνές του πειράματος – Ηλεκτρισμός

Το πείραμα εκτεταμένης πραγματικότητας αποτελείται από έξι σκηνές όπως αυτές φαίνονται στο σχήμα 7.2. Αρχικά ο μαθητής-παίκτης, αφού επιλέξει το κεφάλαιο του ηλεκτρισμού (σκηνή: startScene), εμφανίζονται τα διαθέσιμα πειράματα των ενοτήτων του κεφαλαίου (σκηνή: PektρισmosAbout). Αφού επιλέξει ποιο πείραμα θέλει να υλοποιηθεί, εμφανίζονται οι οδηγίες του πειράματος καθώς και τα υλικά τα οποία πρέπει να συλλέξει (σκηνή: fe6-ilektrismos). Στη συνέχεια, ενεργοποιείται η κάμερα της συσκευής του, και αφού σκανάρει το qrcode της σελίδας 112, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας και μια μπάρα με τα εναπομείναντα υλικά τα οποία βρίσκονται πίσω από αυτή, σε ένα εικονικό περιβάλλον (σκηνή: VR1).



Σχήμα 9.2: Σκηνές πειράματος σελίδας 112

Μόλις συλλέξει όλα τα υλικά της μπάρας, εμφανίζονται οδηγίες (σκηνή: testScene1) για το πείραμα που θα ακολουθήσει. Σκανάροντας ξανά το qrcode της σελίδας 112, εμφανίζεται το κύκλωμα της εικόνας του σχολικού βιβλίου. Ο παίκτης επιλέγοντας τα κουμπιά των υλικών, τα αντίστοιχα αντικείμενα μπαίνουν ανάμεσα στους συνδετήρες. Στο κάτω μέρος της οθόνης υπάρχει το κουμπί “συμπέρασμα”, το οποίο οδηγεί τον παίκτη στην επόμενη σκηνή η οποία αποτελείται από το συμπέρασμα (σκηνή: symperasmaPlektrismos). Μετά την ολοκλήρωση του παραπάνω πειράματος εκτεταμένης πραγματικότητας, οι μαθησιακοί στόχοι που είναι θεμιτό να επιτευχθούν είναι οι εξής:

- Να διαπιστώσουν οι μαθητές πειραματικά την ύπαρξη υλικών που άγουν το ηλεκτρικό ρεύμα και υλικών που δεν άγουν το ηλεκτρικό ρεύμα.
- Να αναγνωρίσουν οι μαθητές το μέρος των καλωδίων που είναι κατασκευασμένο από αγωγούς και αυτό που είναι κατασκευασμένο από μονωτές.

9.4 Σκηνή: PlektrismosAbout

Η σκηνή αυτή φαίνεται στο σχήμα 7.3. Η ιεραρχία των αντικειμένων της αποτελείται από: Μία main camera, ένα directional light για το σωστό φωτισμό της σελίδας, ένα event system για τη λειτουργία των κουμπιών και των εναλλαγών των σκηνών και ένα canva στον οποίο υπάρχουν πολλά textMeshPro για την εισαγωγή των κειμένων καθώς και δύο buttons. Με αυτά γίνεται η εναλλαγή των σκηνών, μέσω ενός script παρόμοιου με του script “menu”, το οποίο αποτελείται από τις μεθόδους “next()” και “back()” που χρησιμοποιούν τη βιβλιοθήκη UnityEngine.SceneManagement.



Σχήμα 9.3: Σκηνή “PlektrismosAbout”

9.5 Σκηνή: Fe6-ilektrismos

Η σκηνή αυτή φαίνεται στο σχήμα 7.4. Η ιεραρχία των αντικειμένων της είναι ίδια με αυτή της σκηνής “PlektrismosAbout”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro και το περιεχόμενό τους είναι διαφορετικό.

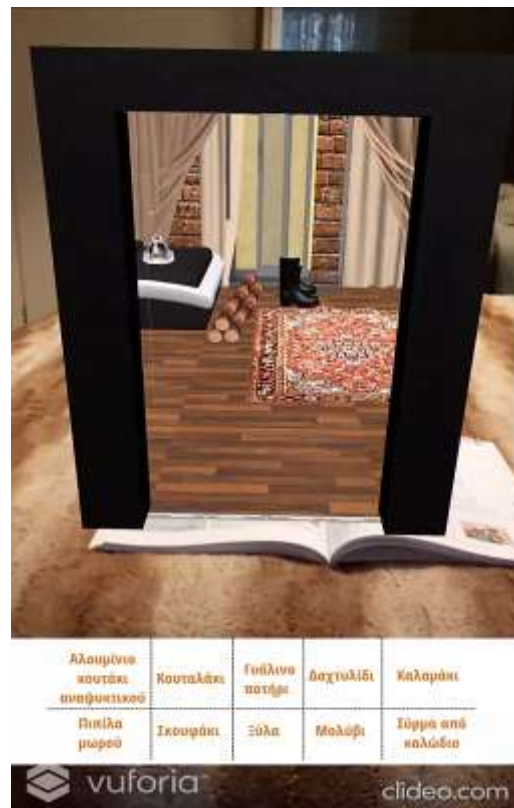


Σχήμα 9.4: Σκηνή fe6-ilektrismos

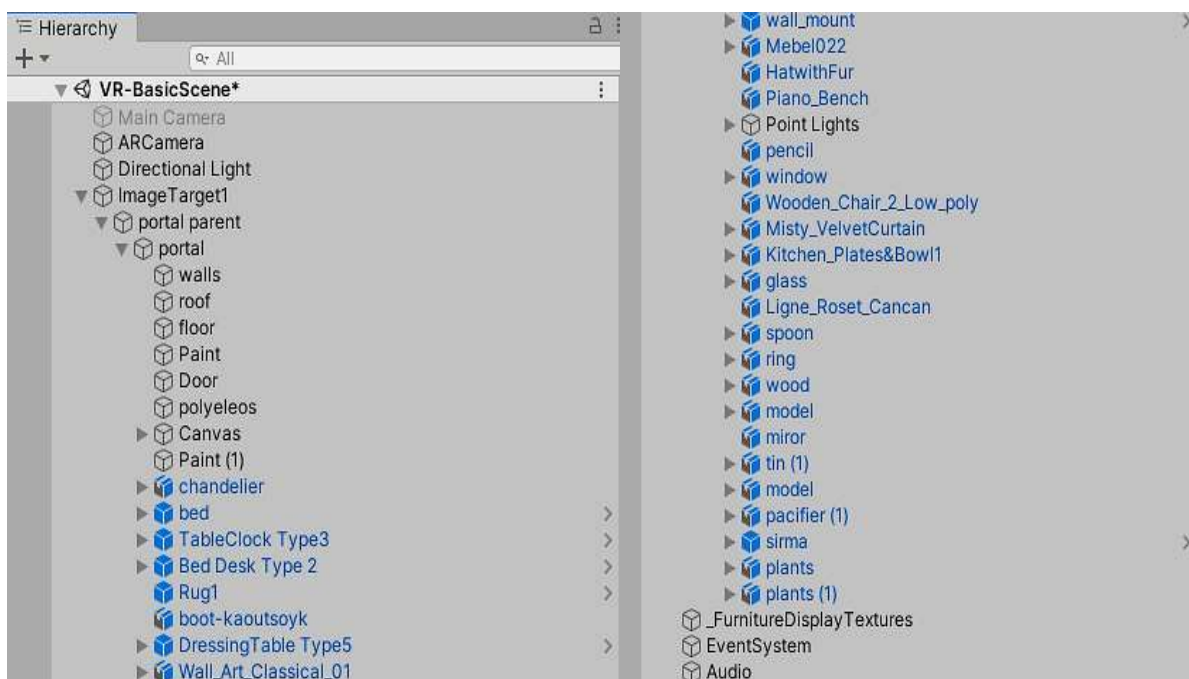
9.6 Σκηνή: VR1

Σε αυτή τη σκηνή, ο παίκτης μετά από προτροπή των οδηγιών της προηγούμενης σκηνής, σκανάρει το qrcode που βρίσκεται στη σελίδα 112 του σχολικού βιβλίου. Μόλις η Vuforia Camera το εντοπίσει, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας πάνω από το qrcode (σχήμα 7.5). Η πόρτα επαυξημένης πραγματικότητας που εμφανίζεται, έχει την ίδια λειτουργικότητα με την πόρτα εικονικής πραγματικότητας των προηγούμενων σκηνών VR. Για να φανεί το εικονικό περιβάλλον που υπάρχει πίσω της, ο παίκτης πρέπει να εισέλθει με τη συσκευή του μέσα στην πόρτα. Μόλις γίνει αυτό, ακολουθεί ολική εμβύθιση του παίκτη σε ένα δωμάτιο εικονικής πραγματικότητας, στο οποίο καλείται να βρει τα δέκα υλικά του πειράματος. Στο κάτω μέρος της οθόνης υπάρχει μπάρα στην οποία φαίνονται τα υπολειπόμενα υλικά. Η ιεραρχία των αντικειμένων της σκηνής VR1 φαίνεται αναλυτικά στο σχήμα 7.6. Περιληπτικά, αποτελείται από τα ίδια βασικά αντικείμενα των προηγούμενων σκηνών VR, με τη μόνη διαφορά τα διαφορετικά εικονικά αντικείμενα που υπάρχουν και αποτελούν το ύψος του δωματίου. Πριν τη δημιουργία της πόρτας, έγινε δημιουργία αντικειμένου “image target” και η επιλογή του marker15 από τη βάση highclass. Το παιδί του image target είναι ένα empty “portal parent” προκειμένου τα περιεχόμενα αυτού να εμφανιστούν εφόσον η arcamera

εντοπίσει το imagemarker15. Τα δομικά στοιχεία του δωματίου είναι οι τοίχοι, η πόρτα, το πάτωμα και η οροφή. Αυτά δημιουργήθηκαν με τη χρήση του Probuilder. Οι ειδικότερες ρυθμίσεις είναι ίδιες με αυτές των προηγούμενων σκηνών VR.



Σχήμα 9.5: Πόρτα επαυξημένης πραγματικότητας VR1 σκηνής



Σχήμα 9.6: Ιεραρχία αντικειμένων VR1

9.6.1 Εικονικό περιβάλλον και εικονικά αντικείμενα VR1

Το εσωτερικό του δωματίου μοιάζει με υπνοδωμάτιο. Αποτελείται από δεκάδες assets, .fbx και .obj που αποθηκεύτηκαν από το unity asset store και από ιστοσελίδες όπως αυτές αναφέρθηκαν στο αντίστοιχο κεφάλαιο της σκηνής “VR4”. Στο σχήμα 7.7 φαίνεται το εσωτερικό του δωματίου με λεπτομέρειες.



Σχήμα 9.7: Δωμάτιο VR1

Τα 10 υλικά που ο παίκτης πρέπει να συλλέξει προκειμένου να προχωρήσει στο πείραμα φαίνονται στο σχήμα 7.8.



Σχήμα 9.8: Υλικά πειράματος (Ηλεκτρισμός)

Για τον ήχο της σκηνής “VR1” χρησιμοποιήθηκε ένα `gameobject empty` με όνομα “Audio”. Στα `components` του `empty` “Audio” προστέθηκε ένα “Audio source” με τις ίδιες ρυθμίσεις των προηγούμενων σκηνών “VR” αλλάζοντας το αρχείο ήχου για να είναι μοναδικό. Ένα ακόμα `component` που προστέθηκε στο `audio` είναι ένα `script` για να καταστρέφει το αντικείμενο του ήχου στη σκηνή `testscene1`, δηλαδή την επόμενη της `VR2`.

9.6.2 Εναλλαγή σκηνής VR1

Στα δέκα αντικείμενα που πρέπει να συλλέξει ο παίκτης έχουν προστεθεί τα ίδια `components` όπως και στις προηγούμενες σκηνές `VR`:

- `Box collider`
- `Audio source`:
- Το `script` “Dotvrndour” (σχήμα 3.11)

Η μπάρα στο κάτω μέρος του παιχνιδιού με τα υπολειπόμενα παιχνίδια είναι ο `canvas` από την ιεραρχία αντικειμένων και αποτελείται από δέκα `textmesh Pro` με τα ονόματα των αντικειμένων και αντίστοιχα δέκα εικόνες (τικ) που εμφανίζονται μόλις ο παίκτης επιλέξει ένα αντικείμενο από αυτά. Στα `components` του `canva` προστέθηκε το `script` “CanvasScriptVR1”, το οποίο έχει την ίδια λειτουργικότητα με το `script` “`canvascriptvrfour`” (σχήμα 3.13) και το `script` “`csSIXx`” (σχήμα 4.10). Με το `script` “`CanvasScriptVR1`”, όταν ο μετρητής `sum` φτάσει των αριθμό δέκα (δηλαδή όταν όλα τα αντικείμενα βρεθούν), καλεί την επόμενη σκηνή μέσω της βιβλιοθήκης `UnityEngine.SceneManagement`.

9.7 Σκηνή: TestScene1

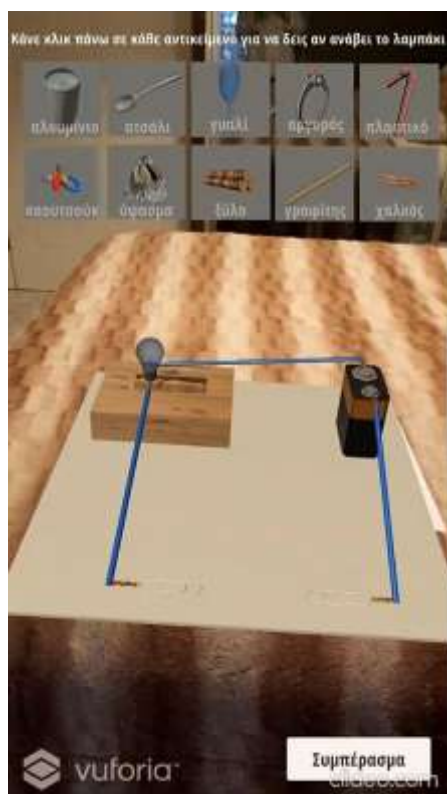
Η σκηνή αυτή φαίνεται στο σχήμα 7.9. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “`PlektrismosAbout`” και “`feb-ilektrismos`”, με τη μόνη διαφορά ότι περιέχει περισσότερα `textMesh Pro` με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται οδηγίες για το πείραμα που ακολουθεί.



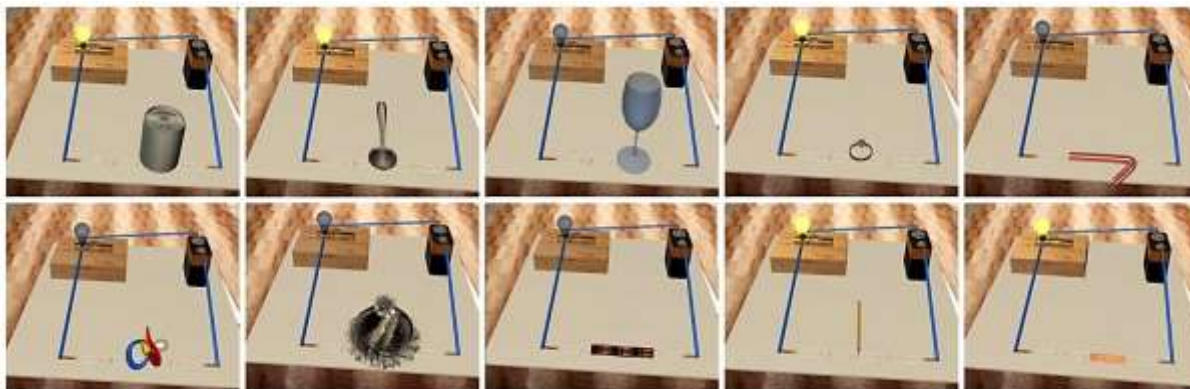
Σχήμα 9.9: Σκηνή `testscene1`

9.8 Σκηνή: 1.animation

Σε αυτή τη σκηνή πραγματοποιείται το πείραμα της σελίδας 112 με τη χρήση εκτεταμένης πραγματικότητας. Στην αρχική οθόνη (σχήμα 7.10) υπάρχει κατασκευασμένο ένα κλειστό ηλεκτρικό κύκλωμα. Στο πάνω μέρος της οθόνης υπάρχουν 10 κουμπιά τα οποία αντιστοιχούν στα δέκα υλικά του πειράματος. Ο παίκτης με την προτροπή «Κάνε κλικ πάνω σε κάθε αντικείμενο για να δεις αν ανάβει το λαμπάκι», μπορεί να επιλέξει όποιο υλικό θέλει για να τοποθετηθεί ανάμεσα στους συνδετήρες. Το λαμπάκι αναλόγως θα είναι ή σβηστό ή ανοιχτό. Μόλις ο παίκτης βγάλει τα συμπεράσματά του και συμπληρώσει τον πίνακα του σχολικού βιβλίου στη σελίδα 113, μπορεί πατώντας το κουμπί «Συμπέρασμα» να προχωρήσει στην επόμενη σκηνή η οποία είναι το συμπέρασμα και ο πίνακας συμπληρωμένος. Στο σχήμα 7.11, φαίνονται οι οθόνες του πειράματος μετά την επιλογή όλων των υλικών (και των 10 κουμπιών).



Σχήμα 9.10: Αρχική οθόνη πειράματος σελίδα 112

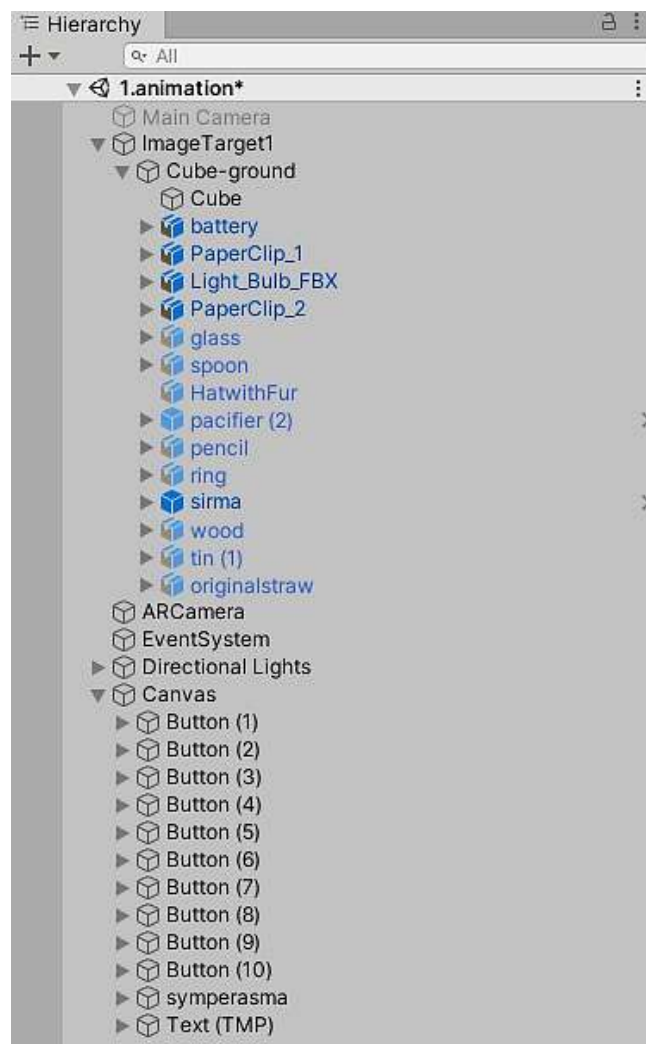


Σχήμα 9.11: Οι οθόνες του πειράματος μετά την επιλογή όλων των υλικών

9.8.1 Ιεραρχία αντικειμένων – 1.animation

Η ιεραρχία των αντικειμένων της σκηνής 1.animation (σχήμα 7.12) αποτελείται από:

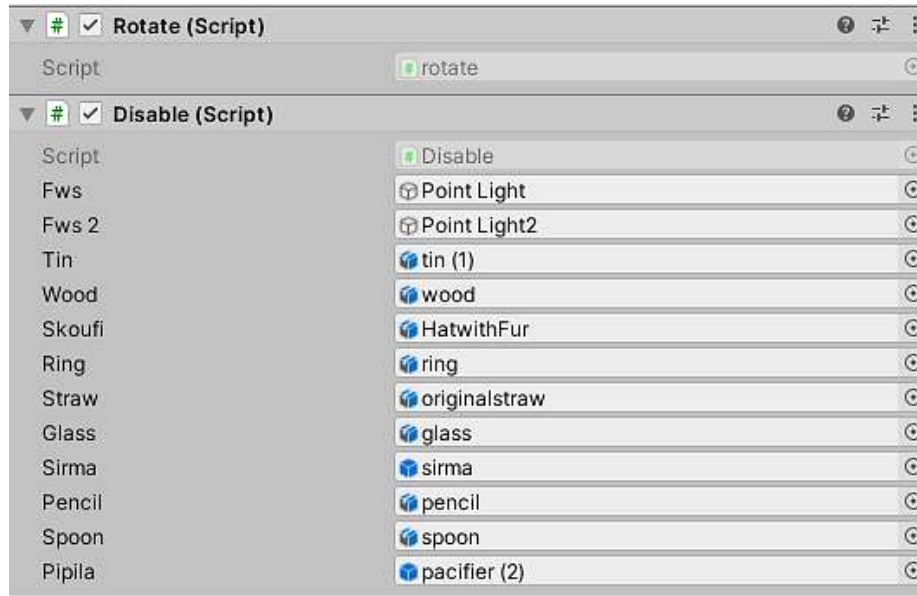
- Ένα event system.
- Μία ARCamera.
- Έξι directional lights για σωστό φωτισμό περιμετρικά της πόρτας και δεκαπέντε point lights εσωτερικά.
- Το image target15 το οποίο εμπεριέχει :
 - Ένα cube-ground το οποίο αποτελεί τη βάση για το κύκλωμα.
 - Τα δέκα υλικά του πειράματος απενεργοποιημένα.
 - Το ηλεκτρικό κύκλωμα.
- Ένα canvas ο οποίος εμπεριέχει :
 - Τα δέκα buttons που αντιστοιχούν στα υλικά του πειράματος.
 - Ένα button «Συμπέρασμα».
 - Ένα TextMeshPro για την προτροπή στο πάνω μέρος της οθόνης.



Σχήμα 9.12: Η ιεραρχία των αντικειμένων της σκηνής 1.animation

9.8.2 Η λειτουργία του πειράματος μέσα από τα scripts (Ηλεκτρισμός)

Αρχικά στο “Cube-ground” από την ιεραρχία των αντικειμένων, έχουν προστεθεί στα components δύο script, το “Rotate” και το “Disable” όπως φαίνεται στο σχήμα 7.13.



Σχήμα 9.13: Ο inspector του “Cube-ground”

Το script “Rotate” (σχήμα 7.14), είναι υπεύθυνο για την περιστροφή του Cube-ground γύρω από τον εαυτό του. Με αυτό το script, ο παίκτης έχει τη δυνατότητα, με ένα άγγιγμα (scroll right- scroll left) να περιστρέψει το Cube-ground (σχήμα 7.15).

```

rotate.cs x
C:\> Users > user > Desktop > ptyxiaki > projects > final project > karantina > Assets > Scripts > 1animationScene > rotate.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class rotate : MonoBehaviour
6  {
7      private Touch touch;
8      private Vector2 touchPosition;
9      private Quaternion rotationY;
10     private float rotateSpeedModifier = 0.1f;
11
12     void Update()
13     {
14         if(Input.touchCount > 0)
15         {
16             touch=Input.GetTouch(0);
17             if(touch.phase == TouchPhase.Moved)
18             {
19                 rotationY=Quaternion.Euler(0f, -touch.deltaPosition.x*rotateSpeedModifier, 0f);
20                 //-touch.deltaPosition.x --> to "x" dilwnei tin kinisi tou daxtulou ston aksona x deksia aristera
21                 // anti gia "x" mporw na valw "y" gia kinisi daxtilou panw katw
22                 // an thelw na kounietai o canvas panw katw:
23                 //rotationX=Quaternion.Euler(touch.deltaPosition.y*rotateSpeedModifier, 0f, 0f);
24                 transform.rotation=rotationY*transform.rotation;
25             }
26         }
27     }
28 }
    
```

Σχήμα 9.14: Το script “Rotate”



Σχήμα 9.15: Scroll right- scroll left (1.animation)

Το script “Disable” (σχήμα 7.16), δημιουργεί δέκα πεδία τύπου gameObject στις γραμμές 7-18, τα οποία αντιστοιχούν στα δέκα υλικά του πειράματος (βλέπε σχήμα 7.13). Στη συνάρτηση void Start(), μόλις ξεκινάει η σκηνή, απενεργοποιούνται και τα δέκα υλικά (γραμμές 21-32), αφού στην αρχή δεν έχει γίνει η επιλογή κανενός κουμπιού. Στη συνέχεια, στα δέκα buttons της ιεραρχίας των αντικειμένων, έχει προστεθεί ξεχωριστά για κάθε ένα, τα script “ObjectTesting” ή “NoObjectTesting” αναλόγως αν το λαμπάκι πρέπει να ανάβει με το πάτημα του κουμπιού (“ObjectTesting”) ή όχι (“NoObjectTesting”). Συνεπώς, τα buttons: αλουμίνιο, ατσάλι, αργυρός, γραφίτης και χαλκός έχουν το script “ObjectTesting”, ενώ τα buttons: γυαλί, πλαστικό, καουτσούκ, ύφασμα και ξύλο έχουν το script “NoObjectTesting”. Τα παραπάνω δύο scripts έχουν σχεδόν την ίδια λειτουργικότητα. Στις γραμμές 9-10 δημιουργούνται δύο πεδία τύπου gameObject για τα δύο pointlights (για το λαμπάκι του κυκλώματος). Στη γραμμή 11 δημιουργείται ένα πεδίο “go” τύπου gameObject το οποίο αποτελεί το αντικείμενο που πρέπει να μπει ανάμεσα στους συνδετήρες (ενεργοποιείται στη γραμμή 25). Στις

Εφαρμογή πειράματος για το κεφάλαιο Ηλεκτρισμός

γραμμές 13-21 δημιουργούνται εννιά πεδία τύπου gameObject τα οποία αποτελούν τα υπόλοιπα εννιά αντικείμενα που πρέπει να μείνουν απενεργοποιημένα (γραμμές 30-38) όσο το “go” βρίσκεται ανάμεσα από τους συνδετήρες. Η διαφορά των 2 scripts βρίσκεται στις γραμμές 26-27, όπου στο “ObjectTesting” που αφορά τα υλικά που το λαμπάκι ανάβει, τα pointlights ενεργοποιούνται ενώ στο “NoObjectTesting” που αφορά τα υλικά που το λαμπάκι δεν ανάβει, τα pointlights απενεργοποιούνται. Τα παραπάνω δύο scripts φαίνονται στο σχήμα 7.17.

```
Disable.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Disable : MonoBehaviour
6 {
7     public GameObject fws;
8     public GameObject fws2;
9     public GameObject tin;
10    public GameObject wood;
11    public GameObject skoufi;
12    public GameObject ring;
13    public GameObject straw;
14    public GameObject glass;
15    public GameObject sirma;
16    public GameObject pencil;
17    public GameObject spoon;
18    public GameObject pipila;
19    void Start()
20    {
21        fws.SetActive(false);
22        fws2.SetActive(false);
23        tin.SetActive(false);
24        wood.SetActive(false);
25        skoufi.SetActive(false);
26        ring.SetActive(false);
27        straw.SetActive(false);
28        glass.SetActive(false);
29        sirma.SetActive(false);
30        pencil.SetActive(false);
31        spoon.SetActive(false);
32        pipila.SetActive(false);
33    }
34 }
35
```

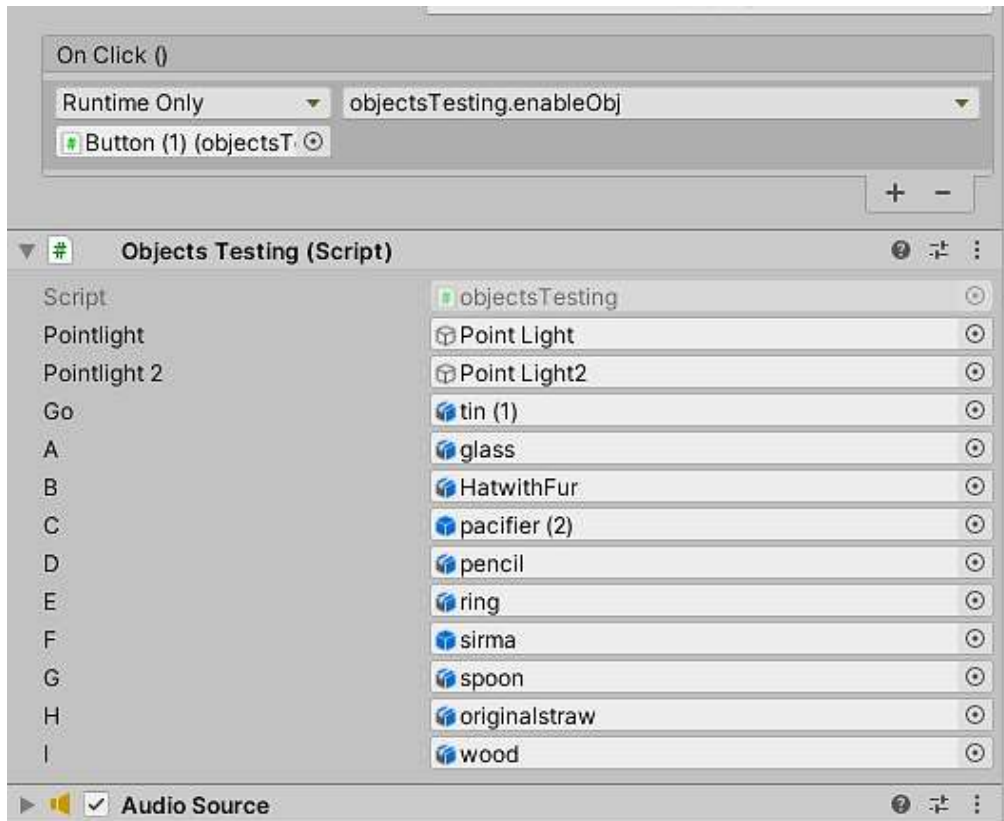
Σχήμα 9.16: Το script “Disable”

```
NoObjectTesting.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class NoObjectTesting : MonoBehaviour
6 {
7     //script gia ta antikimena pou DEN anavai to fws
8
9     public GameObject pointlight;
10    public GameObject pointlight2;
11    public GameObject go; // ta antikimena pou thelw na einai active
12
13    public GameObject a;
14    public GameObject b;
15    public GameObject c;
16    public GameObject d;
17    public GameObject e;
18    public GameObject f;
19    public GameObject g;
20    public GameObject h;
21    public GameObject i;
22
23    public void enableObj()
24    {
25        go.SetActive(true);
26        pointlight.SetActive(false);
27        pointlight2.SetActive(false);
28
29        //ta upoloipta 9 antikimena p thelw n einai disactive
30        a.SetActive(false);
31        b.SetActive(false);
32        c.SetActive(false);
33        d.SetActive(false);
34        e.SetActive(false);
35        f.SetActive(false);
36        g.SetActive(false);
37        h.SetActive(false);
38        i.SetActive(false);
39    }
40 }
41
```

```
objectTesting.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class objectTesting : MonoBehaviour
6 {
7     //script gia ta antikimena pou anavai to fws
8
9     public GameObject pointlight;
10    public GameObject pointlight2;
11    public GameObject go; // ta antikimena pou thelw na einai active
12
13    public GameObject a;
14    public GameObject b;
15    public GameObject c;
16    public GameObject d;
17    public GameObject e;
18    public GameObject f;
19    public GameObject g;
20    public GameObject h;
21    public GameObject i;
22
23    public void enableObj()
24    {
25        go.SetActive(true);
26        pointlight.SetActive(true);
27        pointlight2.SetActive(true);
28
29        //ta upoloipta 9 antikimena p thelw n einai disactive
30        a.SetActive(false);
31        b.SetActive(false);
32        c.SetActive(false);
33        d.SetActive(false);
34        e.SetActive(false);
35        f.SetActive(false);
36        g.SetActive(false);
37        h.SetActive(false);
38        i.SetActive(false);
39        gameObject.GetComponent().Play();
40    }
41 }
```

Σχήμα 9.17: Τα scripts “ObjectTesting” και “NoObjectTesting”

Για παράδειγμα, στο button1 το οποίο αντιστοιχεί στο αλουμίνιο, ο inspector του έχει τη μορφή του σχήματος 7.18. Στο πεδίο “Go” προστέθηκε το αντικείμενο της ιεραρχίας “tin” και στα πεδία A-I τα υπόλοιπα αντικείμενα. Επιπλέον, στο onClick() event του button: select object → button1 και select function → objectsTesting.enableObj. Στα υλικά όπου το λαμπάκι ανάβει, έχει προστεθεί επίσης ένα audio source το οποίο είναι ένα αρχείο ήχου που μοιάζει με «άναμμα λάμπας» (στη γραμμή 39 του “ObjectTesting” ενεργοποιείται το αρχείο ήχου).



Σχήμα 9.18: Ο inspector του button1 (1.animation)

Τέλος, στα components του κουμπιού συμπεράσμα, έχει προστεθεί ένα script το οποίο χρησιμοποιώντας τη βιβλιοθήκη UnityEngine.SceneManagement προχωράει στην επόμενη σκηνή, η οποία βρίσκεται μία θέση μετά (βλέπε σχήμα 1.10), και αποτελεί το συμπέρασμα του πειράματος.

9.9 Σκηνή: symperasmataIlektrismos

Η σκηνή αυτή φαίνεται στο σχήμα 7.19. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “IlektrismosAbout”, “fe6-ilektrismos” και “testscene1”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται τα συμπεράσματα καθώς και ο πίνακας της σελίδας 113 του σχολικού βιβλίου, που πρέπει να συμπληρώσει ο μαθητής. Στο κουμπί «Αρχική» στα components έχει προστεθεί script το οποίο χρησιμοποιώντας τη βιβλιοθήκη UnityEngine.SceneManagement πάει στην αρχική σκηνή η οποία βρίσκεται 9 σκηνές πίσω (βλέπε σχήμα 1.10).

Ηλεκτρισμός		
ΦΕ6: ΑΓΩΓΟΙ ΚΑΙ ΜΟΝΩΤΕΣ		
Πείραμα σελίδα 112-113		
Παρατήρηση		
ΥΛΙΚΟ	ΤΟ ΛΑΜΠΑΚΙ ΑΝΑΒΕΙ	ΤΟ ΛΑΜΠΑΚΙ ΔΕΝ ΑΝΑΒΕΙ
αλουμίνιο	✓	
ατσάλι	✓	
γυαλί		✓
αργυρός	✓	
πλαστικό		✓
καουτσούκ		✓
ύφασμα		✓
ξύλο		✓
γραφίτης	✓	
χαλκός	✓	

Συμπέρασμα

Αγωγοί : αλουμίνιο, ατσάλι, αργυρός, γραφίτης, χαλκός
Μονωτές : γυαλί, πλαστικό, καουτσούκ, ύφασμα, ξύλο

Αρχική

Σχήμα 9.19: Σκηνή-symperasmataHlektrismos

9.10 Επίλογος

Το πείραμα της σελίδας 112 του σχολικού βιβλίου, υλοποιείται με τη χρήση εκτεταμένης πραγματικότητας μέσω των παραπάνω έξι σκηνών. Μετά το πέρας της τελευταίας σκηνής του πειράματος, είναι θεμιτό να έχουν επιτευχθεί οι μαθησιακοί στόχοι που αναφέρθηκαν στο κεφάλαιο 7.3.

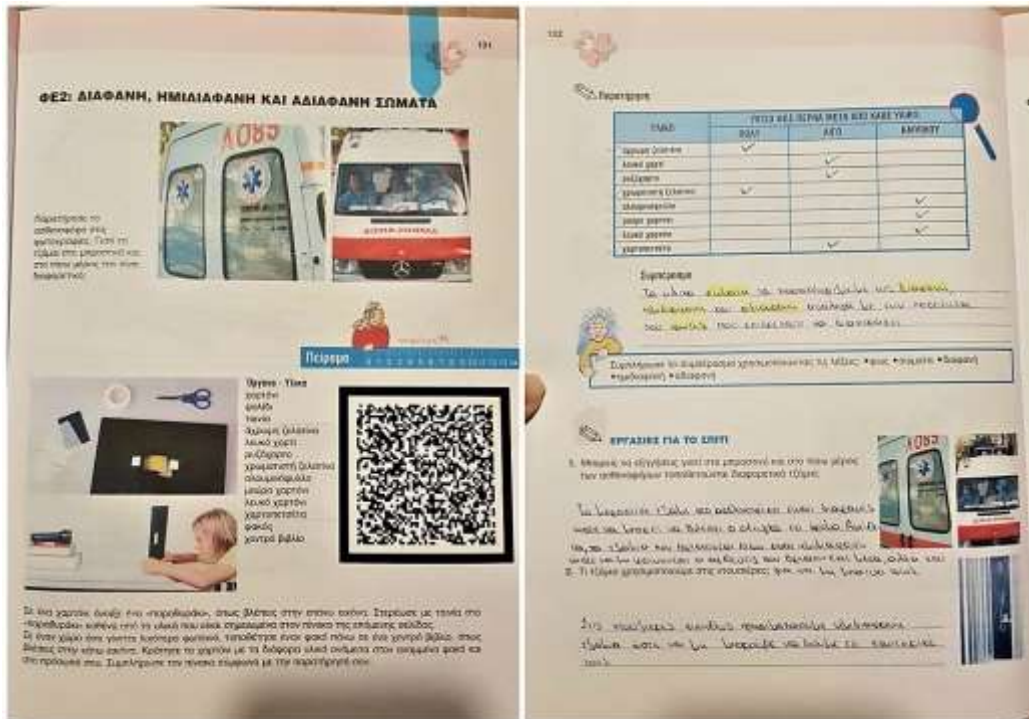
Κεφάλαιο 10ο: Εφαρμογή πειράματος για το κεφάλαιο Φως

10.1 Εισαγωγή

Σε αυτό το κεφάλαιο περιγράφονται οι μαθησιακοί στόχοι του πειράματος της σελίδας 131 του κεφαλαίου «Φως» του σχολικού βιβλίου. Επιπλέον, αναλύονται όλα τα βήματα της δημιουργίας των σκηνών που υλοποιούν το πείραμα, χρησιμοποιώντας τεχνολογίες εκτεταμένης πραγματικότητας.

10.2 Σελίδα βιβλίου 131

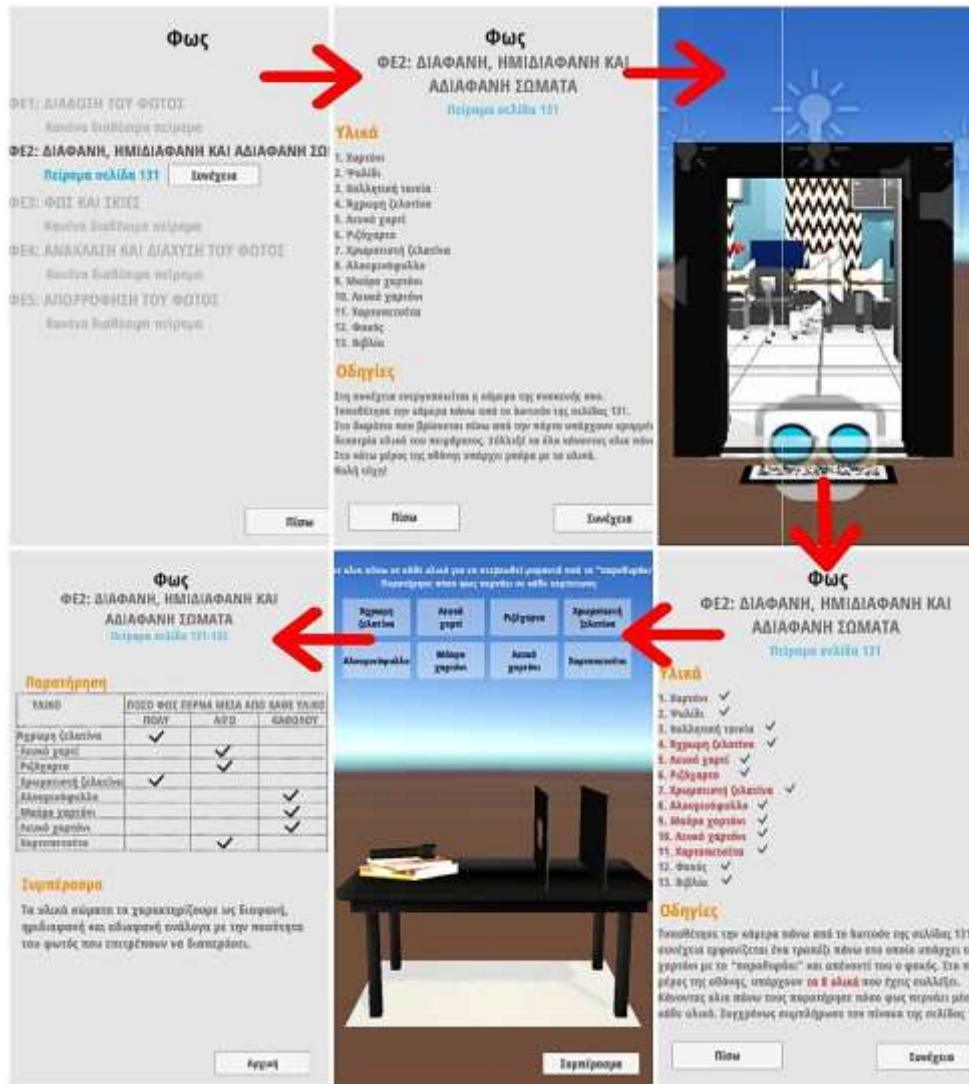
Στο κεφάλαιο «Φως» ανήκει η ενότητα «ΦΕ2: ΔΙΑΦΑΝΗ, ΗΜΙΔΙΑΦΑΝΗ ΚΑΙ ΑΔΙΑΦΑΝΗ ΣΩΜΑΤΑ». Η ενότητα ξεκινάει με ένα εισαγωγικό ερέθισμα και ακολουθεί μια πειραματική αντιμετώπιση. Στο εισαγωγικό ερέθισμα ζητείται από τους μαθητές να παρατηρήσουν τις εικόνες και να σχολιάσουν το διαφορετικό είδος τζαμιών στο μπροστινό και στο πίσω μέρος του ασθενοφόρου. Στο πείραμα που ακολουθεί (σχήμα 8.1), οι μαθητές εξετάζουν πόσο φως περνά μέσα από διάφορα υλικά. Οι μαθητές στερεώνουν τον φακό πάνω σε μερικά βιβλία, όπως φαίνεται στην εικόνα του σχολικού βιβλίου, και κοιτούν προς αυτόν μέσα από το χαρτόνι με τα διάφορα υλικά. Το χαρτόνι πρέπει να έχει αρκετά μεγάλο μέγεθος, ώστε οι μαθητές να μη βλέπουν τον φακό παρά μόνο μέσα από το «παραθυράκι». Οι μαθητές συμπληρώνουν την παρατήρησή τους, σημειώνοντας για κάθε υλικό ένα tick στην αντίστοιχη στήλη του πίνακα της σελίδας 132. Για την εξαγωγή του συμπεράσματος, προκαλείται συζήτηση στην τάξη, μέσα από την οποία οι μαθητές γενικεύουν τις παρατηρήσεις τους στο πείραμα που προηγήθηκε και διατυπώνουν το συμπέρασμα. Εισάγονται οι έννοιες «διαφανές», «ημιδιαφανές» και «αδιαφανές σώμα» από το δάσκαλο ο οποίος βοηθάει τους μαθητές να συνδέσουν αυτούς τους χαρακτηρισμούς των υλικών με τις παρατηρήσεις που έκαναν στα πειράματα. Επιπλέον, για την πραγματοποίηση του πειράματος εκτεταμένης πραγματικότητας, έχει προστεθεί στη σελίδα 131 εκτυπωμένο το image target19 από τη βάση high_class.



Σχήμα 10.1: Σελίδες βιβλίου 131-132

10.3 Οι σκηνές του πειράματος – Φως

Το πείραμα εκτεταμένης πραγματικότητας αποτελείται από έξι σκηνές όπως αυτές φαίνονται στο σχήμα 8.2. Αρχικά ο μαθητής-παίκτης, αφού επιλέξει το κεφάλαιο του φωτός (σκηνή: startScene), εμφανίζονται τα διαθέσιμα πειράματα των ενοτήτων του κεφαλαίου (σκηνή: fwsAbout). Αφού επιλέξει ποιο πείραμα θέλει να υλοποιηθεί, εμφανίζονται οι οδηγίες του πειράματος καθώς και τα υλικά τα οποία πρέπει να συλλέξει (σκηνή: fe2-fws). Στη συνέχεια, ενεργοποιείται η κάμερα της συσκευής του, και αφού σκανάρει το qrcode της σελίδας 131, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας και μια μπάρα με τα εναπομείναντα υλικά τα οποία βρίσκονται πίσω από αυτή, σε ένα εικονικό περιβάλλον (σκηνή: VR5). Μόλις συλλέξει όλα τα υλικά της μπάρας, εμφανίζονται οδηγίες (σκηνή: testScene5) για το πείραμα που θα ακολουθήσει. Σκανάροντας ξανά το qrcode της σελίδας 131, εμφανίζεται ένα τραπέζι πάνω στο οποίο βρίσκονται ο φακός στερεωμένος στα βιβλίο και το χαρτόνι με το παραθυράκι. Στο πάνω μέρος της οθόνης υπάρχουν οκτώ κουμπιά που αντιστοιχούν στα υλικά που πρέπει να στερεωθούν στο παραθυράκι του χαρτονιού. Στο κάτω μέρος της οθόνης υπάρχει το κουμπί “συμπέρασμα” το οποίο οδηγεί τον παίκτη στην επόμενη σκηνή η οποία αποτελείται από το συμπληρωμένο πίνακα της παρατήρησης του βιβλίου καθώς και του συμπεράσματος (σκηνή: symperasmataFws).



Σχήμα 10.2: Σκηνές πειράματος σελίδας 131

Μετά την ολοκλήρωση του παραπάνω πειράματος εκτεταμένης πραγματικότητας, οι μαθησιακοί στόχοι που είναι θεμιτό να επιτευχθούν είναι οι εξής:

- Να αναφέρουν οι μαθητές ότι ένα σώμα χαρακτηρίζεται διαφανές, ημιδιαφανές ή αδιαφανές ανάλογα με το πόσο φως περνά μέσα από αυτό.
- Να ταξινομήσουν οι μαθητές διάφορα σώματα σε διαφανή, ημιδιαφανή και αδιαφανή.

10.4 Σκηνή: FwsAbout

Η σκηνή αυτή φαίνεται στο σχήμα 8.3. Η ιεραρχία των αντικειμένων της αποτελείται από: μία main camera, ένα directional light για το σωστό φωτισμό της σελίδας, ένα event system για τη λειτουργία των κουμπιών και των εναλλαγών των σκηνών και ένα canva στον οποίο υπάρχουν πολλά textMeshPro για την εισαγωγή των κειμένων καθώς και δύο buttons. Με αυτά γίνεται η εναλλαγή των σκηνών, μέσω ενός script παρόμοιου με του script “menu”, το οποίο αποτελείται από τις μεθόδους “next()” και “back()” που χρησιμοποιούν τη βιβλιοθήκη UnityEngine.SceneManagement.



Σχήμα 10.3: Σκηνή “FwsAbout”

10.5 Σκηνή: Fe2-fws

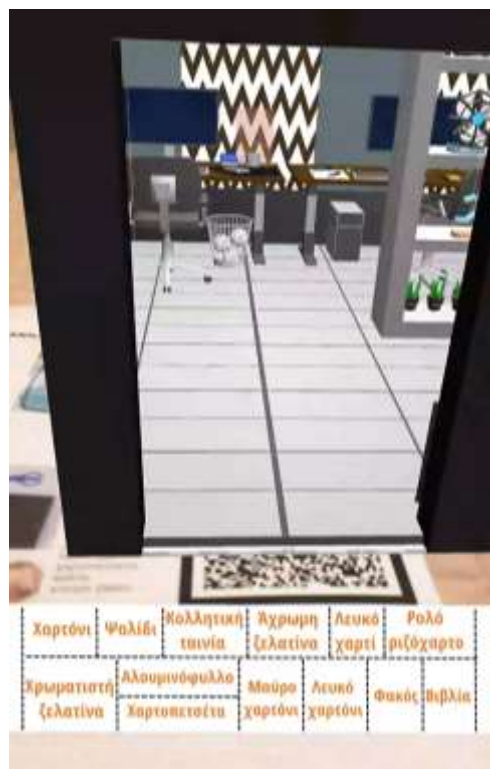
Η σκηνή αυτή φαίνεται στο σχήμα 6.4. Η ιεραρχία των αντικειμένων της είναι ίδια με αυτή της σκηνής “PeptikoAbout”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro και το περιεχόμενό τους είναι διαφορετικό.



Σχήμα 10.4: Σκηνή fe2-fws

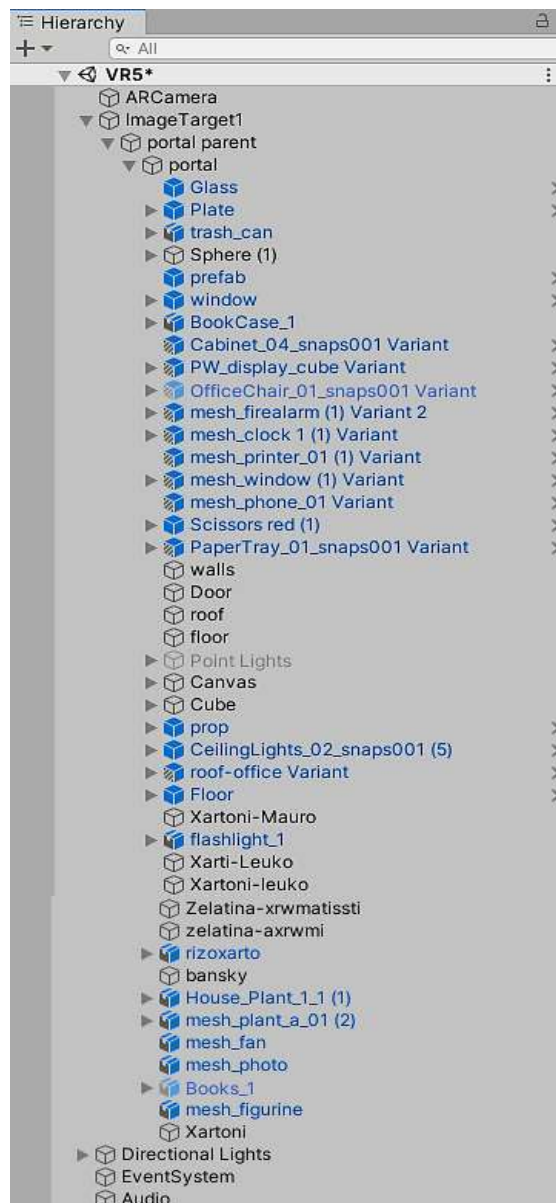
10.6 Σκηνή: VR5

Σε αυτή τη σκηνή, ο παίκτης μετά από προτροπή των οδηγιών της προηγούμενης σκηνής, σκανάρει το qrcode που βρίσκεται στη σελίδα 131 του σχολικού βιβλίου. Μόλις η Vuforia Camera το εντοπίσει, εμφανίζεται μια πόρτα επαυξημένης πραγματικότητας πάνω από το qrcode (σχήμα 8.5).



Σχήμα 10.5: Πόρτα επαυξημένης πραγματικότητας VR5 σκηνής

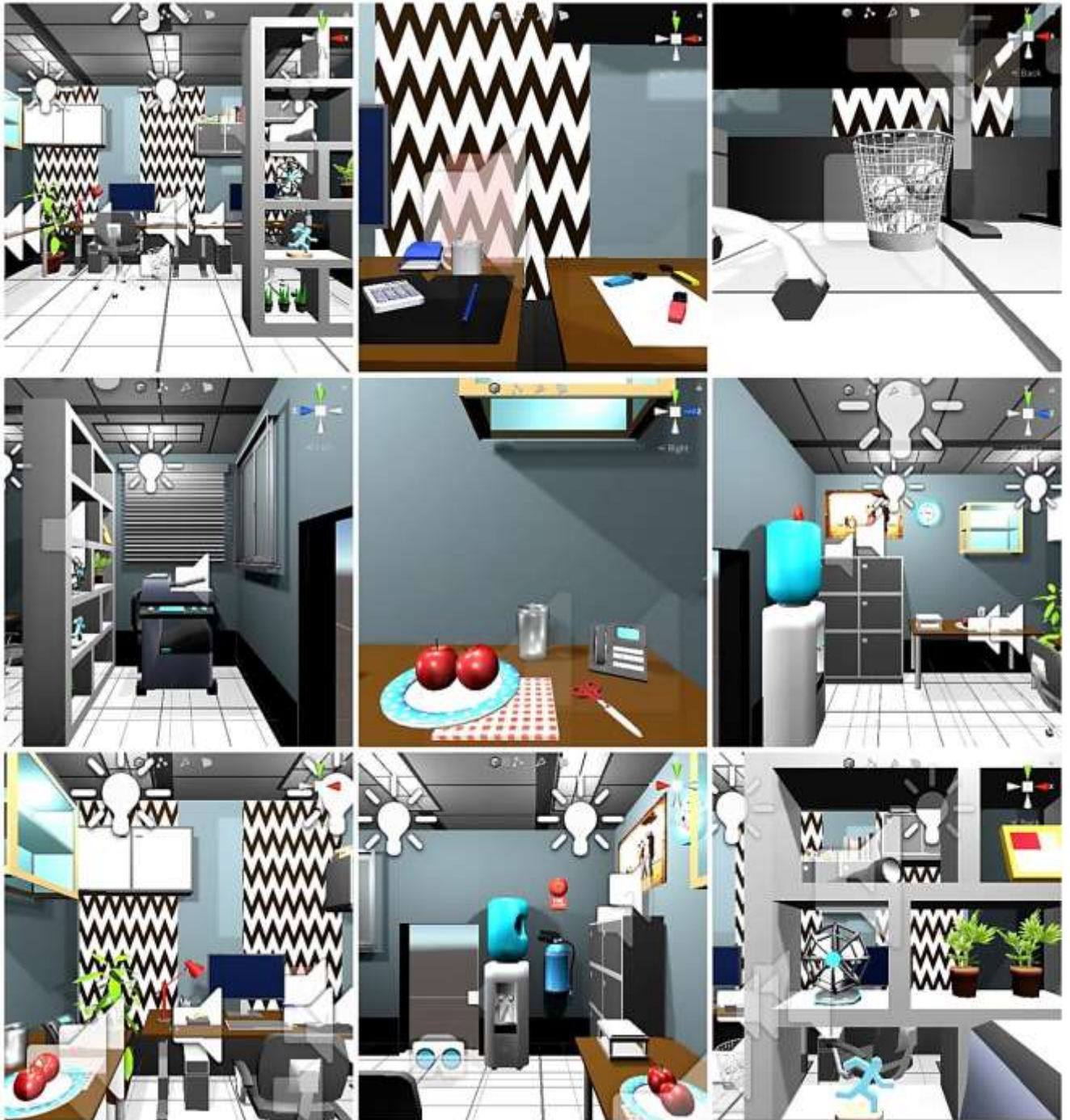
Η πόρτα επανξημένης πραγματικότητας που εμφανίζεται, έχει την ίδια λειτουργικότητα με την πόρτα εικονικής πραγματικότητας των προηγούμενων σκηνών VR. Για να φανεί το εικονικό περιβάλλον που υπάρχει πίσω της, ο παίκτης πρέπει να εισέλθει με τη συσκευή του μέσα στην πόρτα. Μόλις γίνει αυτό, ακολουθεί ολική εμβύθιση του παίκτη σε ένα δωμάτιο εικονικής πραγματικότητας, στο οποίο καλείται να βρει τα δεκατρία υλικά του πειράματος. Στο κάτω μέρος της οθόνης υπάρχει μπάρα στην οποία φαίνονται τα υπολειπόμενα υλικά. Η ιεραρχία των αντικειμένων της σκηνής VR5 φαίνεται αναλυτικά στο σχήμα 8.6. Περιληπτικά, αποτελείται από τα ίδια βασικά αντικείμενα των προηγούμενων σκηνών VR, με τη μόνη διαφορά τα διαφορετικά εικονικά αντικείμενα που υπάρχουν και αποτελούν το ύφος του δωματίου. Πριν τη δημιουργία της πόρτας, έγινε δημιουργία αντικειμένου “image target” και η επιλογή του marker19 από τη βάση highclass. Το παιδί του image target είναι ένα empty “portal parent” προκειμένου τα περιεχόμενα αυτού να εμφανιστούν εφόσον η arcamera εντοπίσει το imagemark19. Τα δομικά στοιχεία του δωματίου είναι οι τοίχοι, η πόρτα, το πάτωμα και η οροφή. Αυτά δημιουργήθηκαν με τη χρήση του Probuilder. Οι ειδικότερες ρυθμίσεις είναι ίδιες με αυτές των προηγούμενων σκηνών VR.



Σχήμα 10.6: Ιεραρχία αντικειμένων VR5

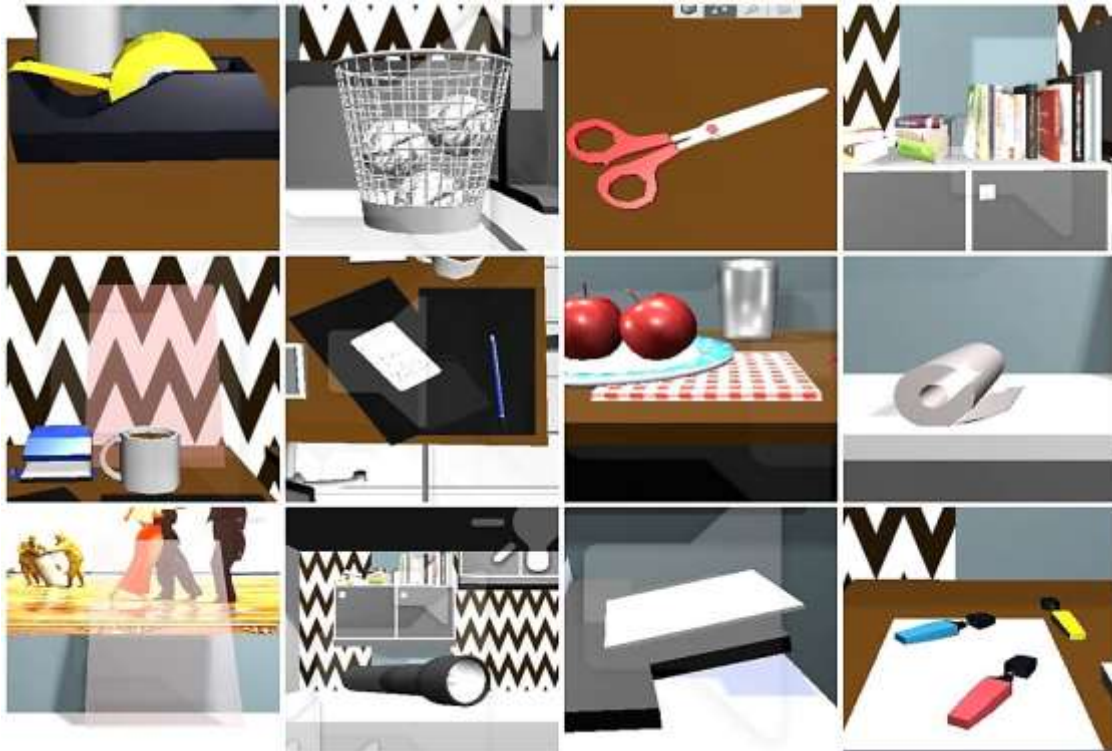
10.6.1 Εικονικό περιβάλλον και εικονικά αντικείμενα VR5

Το εσωτερικό του δωματίου μοιάζει με εταιρικό γραφείο. Αποτελείται από δεκάδες assets, .fbx και .obj που αποθηκεύτηκαν από το unity asset store και από ιστοσελίδες όπως αυτές αναφέρθηκαν στο αντίστοιχο κεφάλαιο της σκηνής “VR4”. Στο σχήμα 8.7 φαίνεται το εσωτερικό του δωματίου με λεπτομέρειες.



Σχήμα 10.7: Δωμάτιο VR5

Τα δεκατρία υλικά που ο παίκτης πρέπει να συλλέξει προκειμένου να προχωρήσει στο πείραμα φαίνονται στο σχήμα 8.8.



Σχήμα 10.8: Υλικά πειράματος (Φως)

Για τον ήχο της σκηνής “VR5” χρησιμοποιήθηκε ένα `gameobject empty` με όνομα “Audio”. Στα `components` του `empty “Audio”` προστέθηκε ένα “Audio source” με τις ίδιες ρυθμίσεις των προηγούμενων σκηνών “VR” αλλάζοντας το αρχείο ήχου για να είναι μοναδικό. Ένα ακόμα `component` που προστέθηκε στο `audio` είναι ένα `script` για να καταστρέφει το αντικείμενο του ήχου στη σκηνή `testscene5`, δηλαδή την επόμενη της `VR5`.

10.6.2 Εναλλαγή σκηνής VR5

Στα δεκατρία αντικείμενα που πρέπει να συλλέξει ο παίκτης έχουν προστεθεί τα ίδια `components` όπως και στις προηγούμενες σκηνές `VR`:

- `Box collider`
- `Audio source`:
- Το `script “Dotvrldour”` (σχήμα 3.11)

Η μπάρα στο κάτω μέρος του παιχνιδιού με τα υπολειπόμενα παιχνίδια είναι ο `canvas` από την ιεραρχία αντικειμένων και αποτελείται από δεκατρία `textmesh Pro` με τα ονόματα των αντικειμένων και αντίστοιχα δεκατρείς εικόνες (`τικ`) που εμφανίζονται μόλις ο παίκτης επιλέξει ένα αντικείμενο από αυτά. Στα `components` του `canva` προστέθηκε το `script “CanvasScriptVR5”`, το οποίο έχει την ίδια λειτουργικότητα με το `script “canvascriptvrfour”` (σχήμα 3.13) και το `script “csSIXx”` (σχήμα 4.10). Με το `script “CanvasScriptVR5”`, όταν ο μετρητής `sum` φτάσει τον αριθμό δεκατρία (δηλαδή όταν όλα τα αντικείμενα βρεθούν), καλεί την επόμενη σκηνή μέσω της βιβλιοθήκης `UnityEngine.SceneManagement`.

10.7 Σκηνή: TestScene5

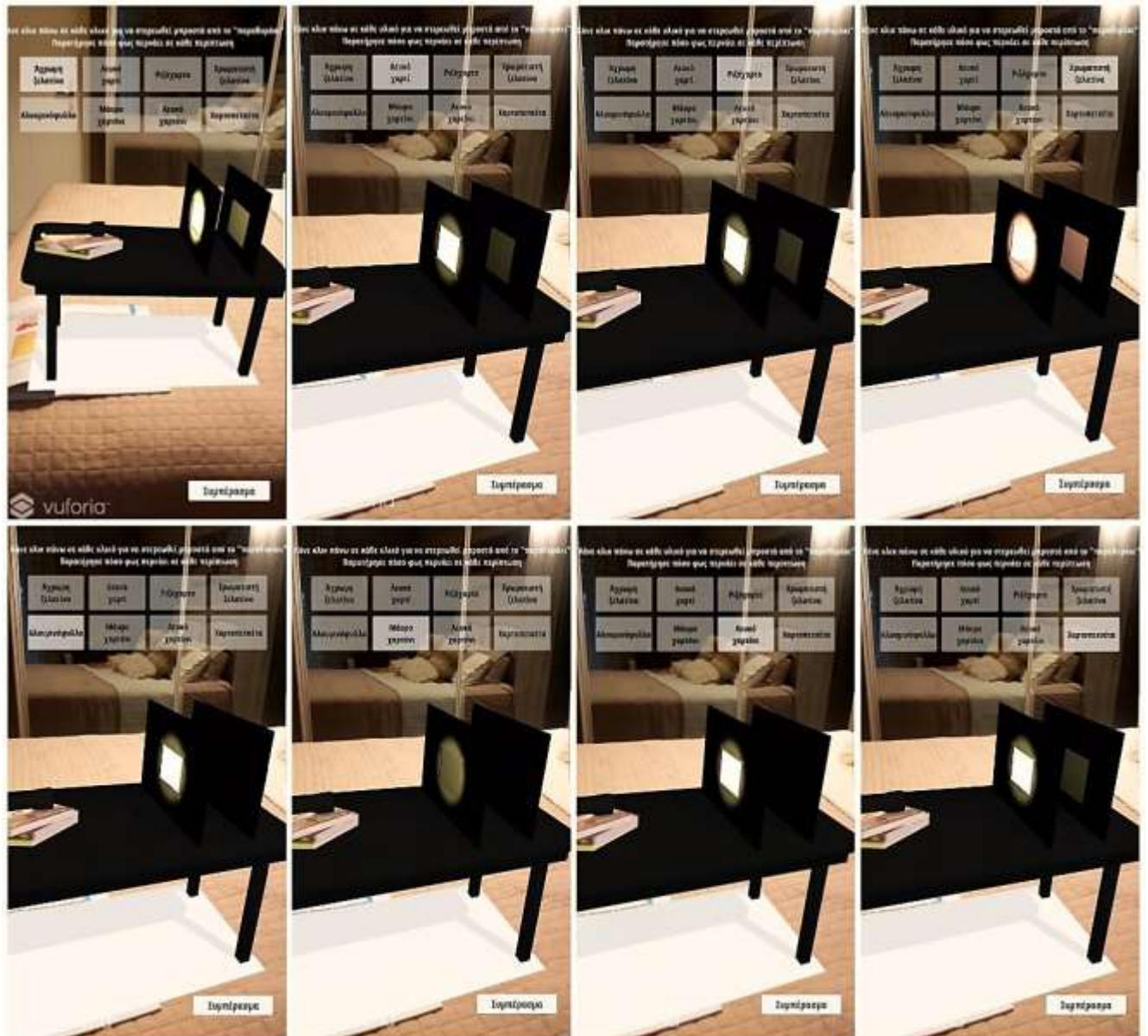
Η σκηνή αυτή φαίνεται στο σχήμα 8.9. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “fwsAbout” και “fe2-fws”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται οδηγίες για το πείραμα που ακολουθεί.



Σχήμα 10.9: Σκηνή testscene5

10.8 Σκηνή: 5.animation

Σε αυτή τη σκηνή πραγματοποιείται το πείραμα της σελίδας 131 με τη χρήση εκτεταμένης πραγματικότητας. Στην αρχική οθόνη υπάρχει ένα τραπέζι πάνω στο οποίο είναι τοποθετημένα τα βιβλία, ο φακός, το χαρτόνι με το παραθυράκι και ένα μαύρο χαρτόνι πίσω του για να φαίνεται πάνω του πόσο φως περνάει από το παραθυράκι. Στο πάνω μέρος της οθόνης υπάρχουν οκτώ κουμπιά τα οποία αντιστοιχούν στα οκτώ υλικά που πρέπει να τοποθετηθούν στο παραθυράκι. Πάνω από τα οκτώ κουμπιά, υπάρχει προτροπή για τον παίκτη «Κάνε κλικ πάνω σε κάθε υλικό για να στερεωθεί μπροστά από το "παραθυράκι". Παρατήρησε πόσο φως περνάει σε κάθε περίπτωση». Στο κάτω μέρος της οθόνης υπάρχει το κουμπί «Συμπέρασμα» με την επιλογή του οποίου προχωράει το πείραμα στην επόμενη σκηνή δηλαδή στο συμπέρασμα. Στο σχήμα 8.10 φαίνονται οι οθόνες του πειράματος μετά την επιλογή και των οκτώ κουμπιών. Η κλίμακα του φωτός που περνάει από το παραθυράκι χωρίζεται σε καθόλου-λίγο-πολύ και στο σχήμα 8.10 δε φαίνονται ξεκάθαρα οι διαφορές στις κλίμακες από ότι στην εφαρμογή.



Σχήμα 10.10: Οι οθόνες του πειράματος (σκηνή: 5.animation)

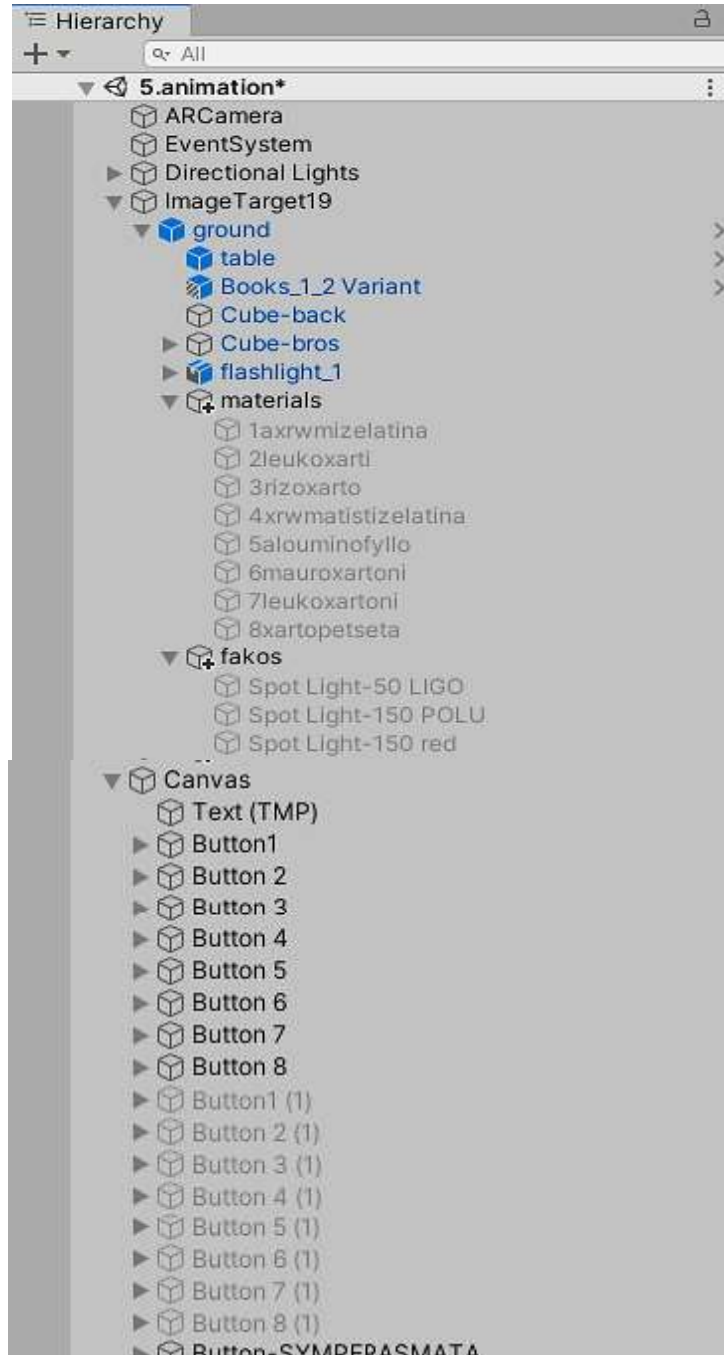
10.8.1 Ιεραρχία αντικειμένων – 5.animation

Η ιεραρχία των αντικειμένων της σκηνής 5.animation (σχήμα 8.11) αποτελείται από:

- Ένα event system.
- Μία ARCamera.
- Έξι directional lights για σωστό φωτισμό.
- Το image target19 το οποίο εμπεριέχει :
 - Μια βάση, το τραπέζι, τα βιβλία, τον φακό, τα οκτώ υλικά, τρία spott lights για το κλιμακωτό φως του φακού, το μαύρο χαρτόνι με το παραθυράκι και ένα μαύρο χαρτόνι πίσω του.
 - Τα οκτώ υλικά, που τοποθετούνται στο παραθυράκι, απενεργοποιημένα.
- Ένα canva ο οποίος εμπεριέχει :

Εφαρμογή πειράματος για το κεφάλαιο Φως

- Τα οκτώ buttons που αντιστοιχούν στα υλικά του πειράματος.
- Άλλα οκτώ buttons που αντιστοιχούν στα υλικά του πειράματος, για να δίνουν την αίσθηση ότι το κουμπί που επιλέχθηκε είναι πατημένο.
- Ένα button «Συμπέρασμα».
- Ένα TextMeshPro για την προτροπή στο πάνω μέρος της οθόνης.

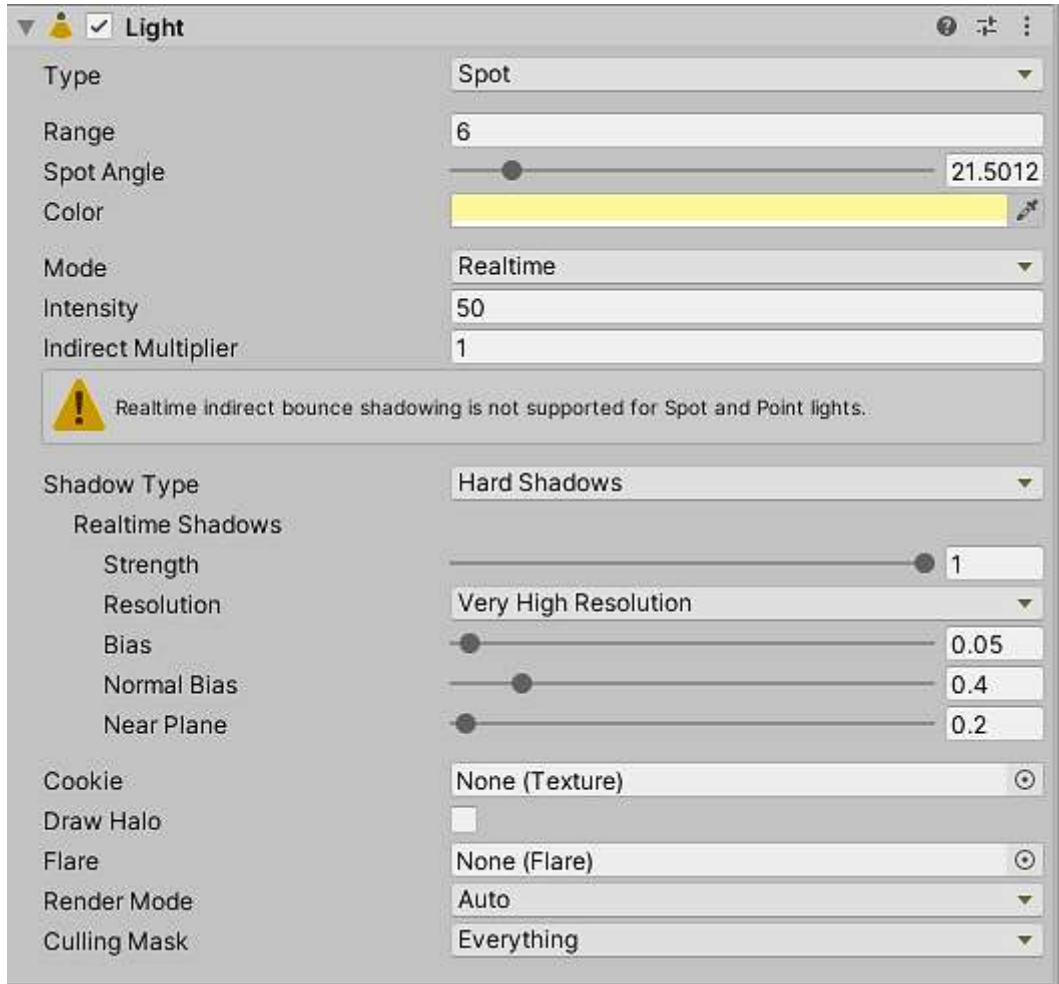


Σχήμα 10.11: Η ιεραρχία των αντικειμένων της σκηνής 5.animation

10.8.2 Τα spot lights του φακού

Στην ιεραρχία των αντικειμένων, ο φακός αποτελείται από τρία φώτα-spot lights. Πιο συγκεκριμένα, το spot light “Spot Light-50 LIGO” (σχήμα 8.12) αντιστοιχεί στο φως του φακού έχουν τα υλικά που

περνάει λίγο φως μέσα από το παραθυράκι. Στο σχήμα 8.12 έχει επιλεγθεί $intensity=50$ για τη χαμηλή ένταση του φωτός. Επίσης, κομβικό μέρος του πειράματος είναι η επιλογή του Shadow type = Hard shadows, προκειμένου ο παίκτης να βλέπει στο πίσω χαρτόνι τη σκιά και να καταλαβαίνει αν περνάει πολύ, λίγο ή καθόλου φως από τα παραθυράκι. Στο color έχει επιλεγθεί ένα βασικό χρώμα φωτός, σχεδόν λευκό προς παστέλ κίτρινο.

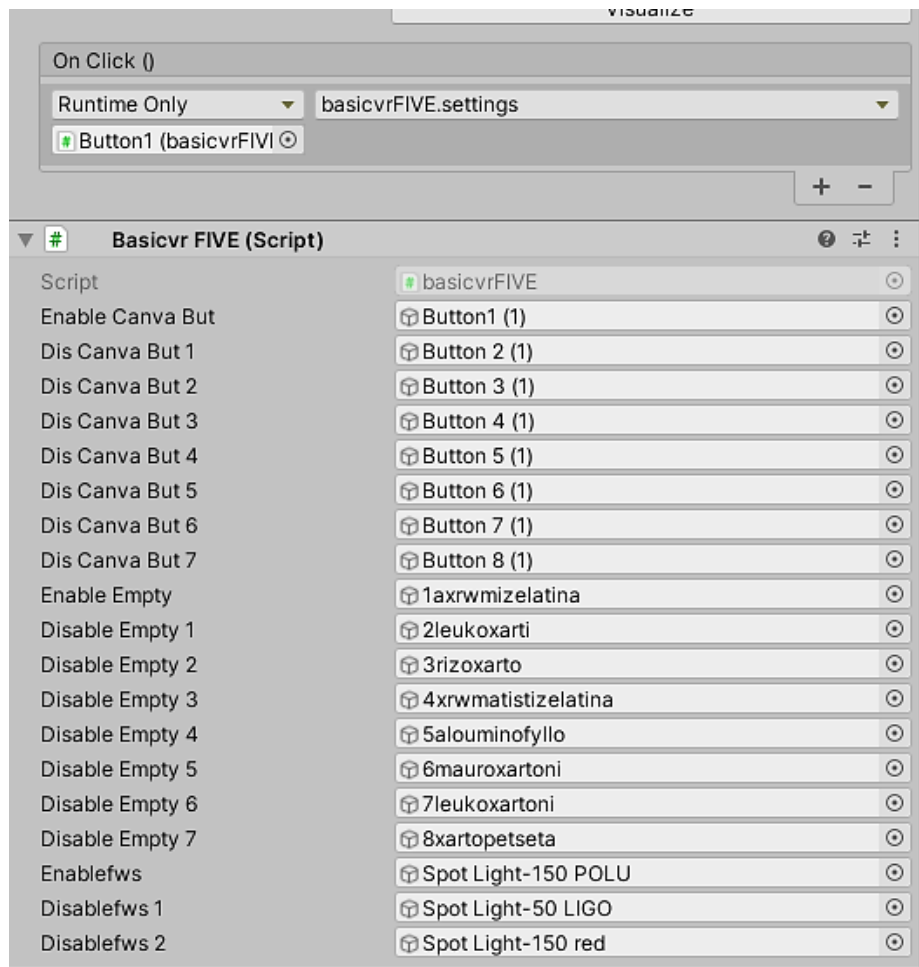


Σχήμα 10.12: Το spot light “Spot Light-50 LIGO”

Επίσης, το spot light “Spot Light-150 POLU” είναι παρόμοιο του “Spot Light-50 LIGO” με τη μόνη διαφορά στο $intensity=150$, για τα υλικά στα οποία περνάει πολύ φως από το παραθυράκι. Επιπλέον, για το υλικό «χρωματιστή ζελατίνα» έπρεπε να δημιουργηθεί ένα spot light ειδικά για αυτό διότι το χρώμα της είναι μοναδικό. Συνεπώς, το “Spot Light-150 red” είναι παρόμοιο του “Spot Light-150 POLU” με τη μόνη διαφορά στο color στο οποίο επιλέχθηκε ένα διαφανές κόκκινο χρώμα αφού η χρωματιστή ζελατίνα είναι διαφανής κόκκινη.

10.8.3 Η λειτουργία του πειράματος μέσα από τα scripts (Φως)

Αρχικά στα οκτώ κουμπιά των υλικών, έχει προστεθεί το script “basicvrFIVE”. Στο onclick() event κάθε κουμπιού έχει επιλεγθεί select function → basicvrFIVE.settings. Η τελική μορφή των inspector των κουμπιών έχει τη μορφή του σχήματος 8.13, με τη διαφορά ότι τα πεδία του basicvrFIVE συμπληρώνονται κατάλληλα για τη λειτουργία του κάθε κουμπιού. Στο σχήμα 8.13 φαίνεται ο inspector του button1, δηλαδή το κουμπί της «άχρωμης ζελατίνας».



Σχήμα 10.13: Ο inspector του button1

Το script “basicvrFIVE” (σχήμα 8.14) δημιουργεί δεκαεννιά πεδία. Αναλυτικά:

- Πεδίο enableCanvaBut: δημιουργεί ένα πεδίο τύπου gameObject στη γραμμή 8, για το κουμπί το οποίο πρέπει να ενεργοποιηθεί (γραμμή 35) για να δίνει την αίσθηση του πατημένου. Αυτό το κουμπί αρχικά είναι απενεργοποιημένο και έχει τοποθετηθεί ακριβώς πάνω στο κουμπί.
- Πεδία disCanvaBut1-7: δημιουργούνται επτά πεδία τύπου gameObject στις γραμμές 10-16, για τα κουμπιά τα οποία πρέπει να φαίνονται σβηστά και απενεργοποιούνται στις γραμμές 37-43.
- Πεδίο enableEmpty: δημιουργεί ένα πεδίο τύπου gameObject στη γραμμή 18, για το υλικό που τοποθετείται στο παραθυράκι. Αυτό ενεργοποιείται με το πάτημα του κουμπιού στη γραμμή 45.
- Πεδία disableEmpty1-7: δημιουργούνται επτά πεδία τύπου gameObject στις γραμμές 20-26, για τα υπόλοιπα επτά υλικά που δεν πρέπει να βρίσκονται στο παραθυράκι όταν βρίσκεται το “enableEmpty”, για αυτό και απενεργοποιούνται στις γραμμές 47-53.
- Πεδίο enablefws: δημιουργεί ένα πεδίο τύπου gameObject στη γραμμή 28, για το φως που πρέπει να ενεργοποιείται με το πάτημα του κουμπιού στη γραμμή 55.
- Πεδία disablefws1-2: δημιουργούνται δύο πεδία τύπου gameObject στις γραμμές 30 και 31, για τα υπόλοιπα δύο φώτα τα οποία πρέπει να μείνουν απενεργοποιημένα (γραμμές 56 και 57).

```

C basicvrFIVEcs X
C > Users > user > Desktop > ptyziaki > projects > final project > karantina > Assets > Scripts > 5.animation > basicvrFIVEcs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class basicvrFIVE : MonoBehaviour
7  {
8      public GameObject enableCanvaBut; //enable to koympi p thelw n fainetai patimeno
9
10     public GameObject disCanvaBut1; //disable ola ta oipolipa koympia p thelw n fnt svista
11     public GameObject disCanvaBut2;
12     public GameObject disCanvaBut3;
13     public GameObject disCanvaBut4;
14     public GameObject disCanvaBut5;
15     public GameObject disCanvaBut6;
16     public GameObject disCanvaBut7;
17
18     public GameObject enableEmpty; //to uliko pou thelw na energoopiisw
19
20     public GameObject disableEmpty1; //ta upoloipa ulika pera apo ayto p thekw n energooihsv
21     public GameObject disableEmpty2;
22     public GameObject disableEmpty3;
23     public GameObject disableEmpty4;
24     public GameObject disableEmpty5;
25     public GameObject disableEmpty6;
26     public GameObject disableEmpty7;
27
28     public GameObject enablefws; //to fws tou fakou pou thelw n enngopoiisw
29
30     public GameObject disablefws1; //ta 2 fwta p thelw n apenngopoiisw
31     public GameObject disablefws2;
32
33     public void settings()
34     {
35         enableCanvaBut.SetActive(true);
36
37         disCanvaBut1.SetActive(false);
38         disCanvaBut2.SetActive(false);
39         disCanvaBut3.SetActive(false);
40         disCanvaBut4.SetActive(false);
41         disCanvaBut5.SetActive(false);
42         disCanvaBut6.SetActive(false);
43         disCanvaBut7.SetActive(false);
44
45         enableEmpty.SetActive(true);
46
47         disableEmpty1.SetActive(false);
48         disableEmpty2.SetActive(false);
49         disableEmpty3.SetActive(false);
50         disableEmpty4.SetActive(false);
51         disableEmpty5.SetActive(false);
52         disableEmpty6.SetActive(false);
53         disableEmpty7.SetActive(false);
54
55         enablefws.SetActive(true);
56         disablefws1.SetActive(false);
57         disablefws2.SetActive(false);
58     }
59 }

```

Σχήμα 10.14: Το script “basicvrFIVE”

Τέλος, στα components του κουμπιού συμπέρασμα, έχει προστεθεί ένα script το οποίο χρησιμοποιώντας τη βιβλιοθήκη UnityEngine.SceneManagement προχωράει στην επόμενη σκηνή, η οποία βρίσκεται μία θέση μετά (βλέπε σχήμα 1.10), και αποτελεί το συμπέρασμα του πειράματος.

10.9 Σκηνή: symperasmatafws

Η σκηνή αυτή φαίνεται στο σχήμα 8.15. Η ιεραρχία των αντικειμένων της είναι παρόμοια με αυτή της σκηνής “fwsAbout”, “fe2-fws” και “testscene5”, με τη μόνη διαφορά ότι περιέχει περισσότερα textMesh Pro με το περιεχόμενό τους να είναι διαφορετικό. Σε αυτή τη σκηνή δίνονται τα συμπεράσματα καθώς και ο πίνακας της σελίδας 132 του σχολικού βιβλίου, που πρέπει να συμπληρώσει ο μαθητής. Στο κουμπί «Αρχική» στα components έχει προστεθεί script το οποίο χρησιμοποιώντας τη βιβλιοθήκη UnityEngine.SceneManagement πάει στην αρχική σκηνή η οποία βρίσκεται 33 σκηνές πίσω (βλέπε σχήμα 1.10).

Φως

ΦΕ2: ΔΙΑΦΑΝΗ, ΗΜΙΔΙΑΦΑΝΗ ΚΑΙ
ΑΔΙΑΦΑΝΗ ΣΩΜΑΤΑ

Πείραμα σελίδα 131-132

Παρατήρηση

ΥΛΙΚΟ	ΠΟΣΟ ΦΩΣ ΠΕΡΝΑ ΜΕΣΑ ΑΠΟ ΚΑΘΕ ΥΛΙΚΟ		
	ΠΟΛΥ	ΛΙΓΟ	ΚΑΘΟΛΟΥ
Αχρωμη ζελατίνα	✓		
Λευκό χαρτί		✓	
Ριζόχαρτο		✓	
Χρωματιστή ζελατίνα	✓		
Αλουμινοφυλλο			✓
Μαύρο χαρτόνι			✓
Λευκό χαρτόνι			✓
Χαρτοπετσέτα		✓	

Συμπέρασμα

Τα υλικά σώματα τα χαρακτηρίζουμε ως διαφανή, ημιδιαφανή και αδιαφανή ανάλογα με την ποσότητα του φωτός που επιτρέπουν να διαπεράσει.

Σχήμα 10.15: Σκηνή-symperasmatafws

10.10 Επίλογος

Το πείραμα της σελίδας 131 του σχολικού βιβλίου, υλοποιείται με τη χρήση εκτεταμένης πραγματικότητας μέσω των παραπάνω έξι σκηνών. Μετά το πέρας της τελευταίας σκηνής του πειράματος, είναι θεμιτό να έχουν επιτευχθεί οι μαθησιακοί στόχοι που αναφέρθηκαν στο κεφάλαιο 8.3.

Κεφάλαιο 11ο: Συμπεράσματα και προτάσεις βελτίωσης

Στην παρούσα Δ.Ε. προτάθηκε η χρήση της εκτεταμένης πραγματικότητας για εκπαιδευτικούς σκοπούς. Η ολοκλήρωσή της έδωσε τα παρακάτω συμπεράσματα:

1. Βασικό συμπέρασμα είναι ότι οι διαδραστικές προσομοιώσεις και οι οπτικοποιήσεις με τη χρήση εκτεταμένης πραγματικότητας συμβάλλουν στην καλύτερη πλαισίωση της γνώσης, ιδιαίτερα στις περιπτώσεις όπου υπάρχει έλλειψη εξοπλισμού και εργαστηριακών πόρων. Ένα από τα σημαντικότερα πλεονεκτήματα στο οποίο βασίζεται η Δ.Ε. είναι ότι το παιχνίδι προσφέρει τη μοναδική δυνατότητα στους εκπαιδευτικούς να ενσωματώσουν στη διδασκαλία τους ένα μέσο που συνδυάζει τη μάθηση με τη διασκέδαση και συγχρόνως τη μάθηση μέσα από την εμπειρία. Η δυναμική μάθηση απορρέει μέσα από βυθιζόμενα, τρισδιάστατα, διαδραστικά και εικονικά περιβάλλοντα. Οι εικονικοί κόσμοι, υποστηρίζουν τη μάθηση με τρόπους τους οποίους η παραδοσιακή διδασκαλία δεν μπορεί, και ταυτόχρονα δίνουν μεγάλη ελευθερία στον τρόπο με τον οποίο οι εκπαιδευτικοί θα τους ενσωματώσουν στη διδασκαλία, κάνοντάς την περισσότερο διασκεδαστική και αποτελεσματική.
2. Κάθε μαθητής είναι ξεχωριστός και ο τρόπος με τον οποίο μαθαίνει καλύτερα και πιο εύκολα διαφέρει από τον έναν στον άλλον. Για παράδειγμα, κάποιοι μαθητές μαθαίνουν καλύτερα ακούγοντας το μαθησιακό υλικό. Άλλοι μαθαίνουν ευκολότερα παρακολουθώντας οπτικοποιημένο το μαθησιακό περιεχόμενο και άλλοι κάνοντας πρακτική εφαρμογή του μαθησιακού περιεχομένου. Το παιχνίδι, χρησιμοποιώντας περιβάλλον εκτεταμένης πραγματικότητας, έχει δημιουργήσει ένα μικτό μαθησιακό περιεχόμενο και συγχρόνως ένα συνδυασμό δραστηριοτήτων που ανταποκρίνεται στις ανάγκες όλων των τύπων μαθητών. Το περιεχόμενο του παιχνιδιού υποστηρίζει τα διαφορετικά μαθησιακά μοντέλα και προσφέρει στους μαθητές δυνατότητες για να ανακαλύψουν, να εξερευνήσουν και να εκφράσουν τη δική τους κατανόηση στο θέμα.
3. Το unity αποτελεί ένα πανίσχυρο εργαλείο για όποιον επιθυμεί να δημιουργήσει μια εφαρμογή εκτεταμένης πραγματικότητας. Με την αξιοποίηση των δυνατοτήτων του, μπορούν να δημιουργηθούν εξαιρετικές εμπειρίες εκτεταμένης πραγματικότητας οι οποίες βασίζονται στην επέκταση των ανθρώπινων εμπειριών που σχετίζονται με τη συναίσθηση (που εκπροσωπούνται από την εικονική πραγματικότητα) και την απόκτηση της γνώσης (που αντιπροσωπεύεται από την επαυξημένη πραγματικότητα).

Οι προτάσεις για μελλοντικές βελτιώσεις συνοψίζονται στις εξής:

1. Σε επόμενη έκδοση της εφαρμογής, τα qrcode θα αντικατασταθούν από αναγνώριση κατάλληλης εικόνας του βιβλίου.
2. Ενσωμάτωση περισσότερων πειραμάτων.
3. Προσθήκη περισσότερων εικονικών και ηχητικών effects και animations.
4. Βελτίωση των γραφικών του παιχνιδιού.
5. Βελτίωση του κώδικα.
6. Εκκαθάριση και συμπίκνωση των assets της εφαρμογής προκειμένου να καταλαμβάνει λιγότερο χώρο στη μνήμη της έξυπνης συσκευής.
7. Πειραματική χρήση της εφαρμογής σε τάξη της ε' δημοτικού.
8. Προσθήκη της εφαρμογής στο playstore.
9. Δημιουργία της εφαρμογής σε ios λογισμικό και προσθήκη της στο app store.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία - ebooks

- [1] Βοσινάκης, Σ., 2015. Εικονικοί κόσμοι. [ηλεκτρ. βιβλ.] Αθήνα:Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. Διαθέσιμο στο: <http://hdl.handle.net/11419/3187>
- [2] Μουστάκας Κ., Παλιόκας Ι., Τσακίρης Α., Τζοβάρας Δ., 2015. Γραφικά και εικονική πραγματικότητα.
- [3] Ευσταθίου Φίλιππος, Συστήματα επικοινωνίας μικτής πραγματικότητας σε πραγματικό χρόνο. 2007.
- [4] Appliedart [Online]. Available: <http://www.appliedart.com/blog/vr-ar-or-mr-what-s-the-difference-why-should-i-care>
- [5] Çöltekin A., Lochhead I., Madden M., Christophe S., Devaux A., Pettit C., Lock O., Shukl S., Herman L., Stacho ņ Z., Kubíček P., Snopková D., Bernardes S., Hedley N, Extended Reality in Spatial Sciences: A Review of Research Challenges and Future Directions, 2020.
- [6] Λέπουρας Γ., Αντωνίου Α., Πλατής Ν., Χαρίτος Δ., Ανάπτυξη συστημάτων εικονικής πραγματικότητας. Ελληνικά ακαδημαϊκά ηλεκτρονικά συγγράμματα και βοηθηματα, 2015.
- [7] Μπάκα Μ., Επαυξημένη πραγματικότητα και εκπαίδευση: Διερεύνηση των μαθησιακών αποτελεσμάτων σε μαθητές της πρωτοβάθμιας εκπαίδευσης στο πλαίσιο διδασκαλίας. 2018.
για το Ηλιακό Σύστημα
- [8] Matt Smith, Unity 2018 Cookbook: Over 160 recipes to take yous 2D and 3D game development to the next level. Packt Publishing Ltd, 2018.
- [9] John P.Doran, Building an FPS game with Unity. Packt Publishing Ltd, 2015.
- [10] Jonathan Linowes and Krystian Babilinski, Augmented reality for developers: Build practical augmented reality applications with Unity, ARCore, ARKit and Vuforia. Packt Publishing Ltd, 2017.
- [11] Dawid Borycki, Programming for Mixed reality with windows 10, Unity, Vuforia and UrhoSharp. Pearsons Education Inc, 2019.

Internet Site

- [12] Vuforia Engine [Online]. Available: <https://engine.vuforia.com/engine>
- [13] Shawn Lehner [Online]. Available: <https://shawnlehner.github.io/ARMaker/>
- [14] Google Fonts [Online]. Available: <https://fonts.google.com/?subset=greek>
- [15] free3D [Online]. Available: <https://free3d.com/3d-models/fbx>
- [16] Sketchfab [Online]. Available: <https://sketchfab.com/feed>
- [17] Turbosquid [Online]. Available: <https://www.turbosquid.com/>
- [18] Poly [Online]. Available: <https://poly.google.com/>

ΠΑΡΑΡΤΗΜΑ Α : Script “Water FX.cs”

```
1. using System;
2. using System.Collections.Generic;
3. using UnityEngine;
4.
5. namespace WaterBuoyancy
6. {
7.     [ExecuteInEditMode] // Make water live-update even when not in play mode
8.     public class WaterFX : MonoBehaviour
9.     {
10.         public enum WaterMode
11.         {
12.             Simple = 0,
13.             Reflective = 1,
14.             Refractive = 2,
15.         };
16.
17.         [SerializeField]
18.         private WaterMode waterMode = WaterMode.Refractive;
19.
20.         [SerializeField]
21.         private bool disablePixelLights = true;
22.
23.         [SerializeField]
24.         private int textureSize = 256;
25.
26.         [SerializeField]
27.         private float clipPlaneOffset = 0.07f;
28.
29.         [SerializeField]
30.         private LayerMask reflectLayers = -1;
31.
32.         [SerializeField]
33.         private LayerMask refractLayers = -1;
34.
35.         private Dictionary<Camera, Camera> m_ReflectionCameras = new Dictionary<Camera,
            Camera>(); // Camera -> Camera table
36.         private Dictionary<Camera, Camera> m_RefractionCameras = new Dictionary<Camera,
            Camera>(); // Camera -> Camera table
37.         private RenderTexture m_ReflectionTexture;
38.         private RenderTexture m_RefractionTexture;
39.         private WaterMode m_HardwareWaterSupport = WaterMode.Refractive;
```

```

40. private int m_OldReflectionTextureSize;
41. private int m_OldRefractionTextureSize;
42. private static bool s_InsideWater;
43.
44. // This is called when it's known that the object will be rendered by some
45. // camera. We render reflections / refractions and do other updates here.
46. // Because the script executes in edit mode, reflections for the scene view
47. // camera will just work!
48. public void OnWillRenderObject()
49. {
50. if (!enabled || !GetComponent<Renderer>() || !GetComponent<Renderer>().sharedMaterial ||
51. !GetComponent<Renderer>().enabled)
52. {
53. return;
54. }
55.
56. Camera cam = Camera.current;
57. if (!cam)
58. {
59. return;
60. }
61.
62. // Safeguard from recursive water reflections.
63. if (s_InsideWater)
64. {
65. return;
66. }
67. s_InsideWater = true;
68.
69. // Actual water rendering mode depends on both the current setting AND
70. // the hardware support. There's no point in rendering refraction textures
71. // if they won't be visible in the end.
72. m_HardwareWaterSupport = FindHardwareWaterSupport();
73. WaterMode mode = GetWaterMode();
74.
75. Camera reflectionCamera, refractionCamera;
76. CreateWaterObjects(cam, out reflectionCamera, out refractionCamera);
77.
78. // find out the reflection plane: position and normal in world space
79. Vector3 pos = transform.position;
80. Vector3 normal = transform.up;
81.
82. // Optionally disable pixel lights for reflection/refraction
83. int oldPixelLightCount = QualitySettings.pixelLightCount;
84. if (disablePixelLights)
85. {

```

```

86. QualitySettings.pixelLightCount = 0;
87. }
88.
89. UpdateCameraModes(cam, reflectionCamera);
90. UpdateCameraModes(cam, refractionCamera);
91.
92. // Render reflection if needed
93. if (mode >= WaterMode.Reflective)
94. {
95. // Reflect camera around reflection plane
96. float d = -Vector3.Dot(normal, pos) - clipPlaneOffset;
97. Vector4 reflectionPlane = new Vector4(normal.x, normal.y, normal.z, d);
98.
99. Matrix4x4 reflection = Matrix4x4.zero;
100. CalculateReflectionMatrix(ref reflection, reflectionPlane);
101. Vector3 oldpos = cam.transform.position;
102. Vector3 newpos = reflection.MultiplyPoint(oldpos);
103. reflectionCamera.worldToCameraMatrix = cam.worldToCameraMatrix * reflection;
104.
105. // Setup oblique projection matrix so that near plane is our reflection
106. // plane. This way we clip everything below/above it for free.
107. Vector4 clipPlane = CameraSpacePlane(reflectionCamera, pos, normal, 1.0f);
108. reflectionCamera.projectionMatrix = cam.CalculateObliqueMatrix(clipPlane);
109.
110. reflectionCamera.cullingMask = ~(1 << 4) & reflectLayers.value; // never render water layer
111. reflectionCamera.targetTexture = m_ReflectionTexture;
112. GL.invertCulling = true;
113. reflectionCamera.transform.position = newpos;
114. Vector3 euler = cam.transform.eulerAngles;
115. reflectionCamera.transform.eulerAngles = new Vector3(-euler.x, euler.y, euler.z);
116. reflectionCamera.Render();
117. reflectionCamera.transform.position = oldpos;
118. GL.invertCulling = false;
119. GetComponent<Renderer>().sharedMaterial.SetTexture("_ReflectionTex",
    m_ReflectionTexture);
120. }
121.
122. // Render refraction
123. if (mode >= WaterMode.Refractive)
124. {
125. refractionCamera.worldToCameraMatrix = cam.worldToCameraMatrix;
126.
127. // Setup oblique projection matrix so that near plane is our reflection
128. // plane. This way we clip everything below/above it for free.
129. Vector4 clipPlane = CameraSpacePlane(refractionCamera, pos, normal, -1.0f);
130. refractionCamera.projectionMatrix = cam.CalculateObliqueMatrix(clipPlane);

```

```

131.   refractionCamera.cullingMask = ~(1 << 4) & refractLayers.value; // never render water layer
132.   refractionCamera.targetTexture = m_RefractionTexture;
133.   refractionCamera.transform.position = cam.transform.position;
134.   refractionCamera.transform.rotation = cam.transform.rotation;
135.   refractionCamera.Render();
136.   GetComponent<Renderer>().sharedMaterial.SetTexture("_RefractionTex",
    m_RefractionTexture);
137.   }
138.
139.   // Restore pixel light count
140.   if (disablePixelLights)
141.   {
142.   QualitySettings.pixelLightCount = oldPixelLightCount;
143.   }
144.
145.   // Setup shader keywords based on water mode
146.   switch (mode)
147.   {
148.   case WaterMode.Simple:
149.   Shader.EnableKeyword("WATER_SIMPLE");
150.   Shader.DisableKeyword("WATER_REFLECTIVE");
151.   Shader.DisableKeyword("WATER_REFRACTIVE");
152.   break;
153.   case WaterMode.Reflective:
154.   Shader.DisableKeyword("WATER_SIMPLE");
155.   Shader.EnableKeyword("WATER_REFLECTIVE");
156.   Shader.DisableKeyword("WATER_REFRACTIVE");
157.   break;
158.   case WaterMode.Refractive:
159.   Shader.DisableKeyword("WATER_SIMPLE");
160.   Shader.DisableKeyword("WATER_REFLECTIVE");
161.   Shader.EnableKeyword("WATER_REFRACTIVE");
162.   break;
163.   }
164.
165.   s_InsideWater = false;
166.   }
167.
168.   // Cleanup all the objects we possibly have created
169.   void OnDisable()
170.   {
171.   if (m_ReflectionTexture)
172.   {
173.   DestroyImmediate(m_ReflectionTexture);

```

```

174.  m_ReflectionTexture = null;
175.  }
176.  if (m_RefractionTexture)
177.  {
178.  DestroyImmediate(m_RefractionTexture);
179.  m_RefractionTexture = null;
180.  }
181.  foreach (var kvp in m_ReflectionCameras)
182.  {
183.  DestroyImmediate((kvp.Value).gameObject);
184.  }
185.  m_ReflectionCameras.Clear();
186.  foreach (var kvp in m_RefractionCameras)
187.  {
188.  DestroyImmediate((kvp.Value).gameObject);
189.  }
190.  m_RefractionCameras.Clear();
191.  }
192.
193.  // This just sets up some matrices in the material; for really
194.  // old cards to make water texture scroll.
195.  void Update()
196.  {
197.  if (!GetComponent<Renderer>())
198.  {
199.  return;
200.  }
201.  Material mat = GetComponent<Renderer>.sharedMaterial;
202.  if (!mat)
203.  {
204.  return;
205.  }
206.
207.  Vector4 waveSpeed = mat.GetVector("WaveSpeed");
208.  float waveScale = mat.GetFloat("_WaveScale");
209.  Vector4 waveScale4 = new Vector4(waveScale, waveScale, waveScale * 0.4f, waveScale *
    0.45f);
210.
211.  // Time since level load, and do intermediate calculations with doubles
212.  double t = Time.timeSinceLevelLoad / 20.0;
213.  Vector4 offsetClamped = new Vector4(
214.  (float)Math.IEEEERemainder(waveSpeed.x * waveScale4.x * t, 1.0),
215.  (float)Math.IEEEERemainder(waveSpeed.y * waveScale4.y * t, 1.0),
216.  (float)Math.IEEEERemainder(waveSpeed.z * waveScale4.z * t, 1.0),
217.  (float)Math.IEEEERemainder(waveSpeed.w * waveScale4.w * t, 1.0)
218.  );

```

```

219.
220.     mat.SetVector("_WaveOffset", offsetClamped);
221.     mat.SetVector("_WaveScale4", waveScale4);
222.     }
223.
224.     void UpdateCameraModes(Camera src, Camera dest)
225.     {
226.         if (dest == null)
227.         {
228.             return;
229.         }
230.         // set water camera to clear the same way as current camera
231.         dest.clearFlags = src.clearFlags;
232.         dest.backgroundColor = src.backgroundColor;
233.         if (src.clearFlags == CameraClearFlags.Skybox)
234.         {
235.             Skybox sky = src.GetComponent<Skybox>();
236.             Skybox mysky = dest.GetComponent<Skybox>();
237.             if (!sky || !sky.material)
238.             {
239.                 mysky.enabled = false;
240.             }
241.             else
242.             {
243.                 mysky.enabled = true;
244.                 mysky.material = sky.material;
245.             }
246.         }
247.         // update other values to match current camera.
248.         // even if we are supplying custom camera&projection matrices,
249.         // some of values are used elsewhere (e.g. skybox uses far plane)
250.         dest.farClipPlane = src.farClipPlane;
251.         dest.nearClipPlane = src.nearClipPlane;
252.         dest.orthographic = src.orthographic;
253.         dest.fieldOfView = src.fieldOfView;
254.         dest.aspect = src.aspect;
255.         dest.orthographicSize = src.orthographicSize;
256.     }
257.
258.     // On-demand create any objects we need for water
259.     void CreateWaterObjects(Camera currentCamera, out Camera reflectionCamera, out Camera
        refractionCamera)
260.     {
261.         WaterMode mode = GetWaterMode();
262.
263.         reflectionCamera = null;

```

```

264.   refractionCamera = null;
265.
266.   if (mode >= WaterMode.Reflective)
267.   {
268.       // Reflection render texture
269.       if (!m_ReflectionTexture || m_OldReflectionTextureSize != textureSize)
270.       {
271.           if (m_ReflectionTexture)
272.           {
273.               DestroyImmediate(m_ReflectionTexture);
274.           }
275.           m_ReflectionTexture = new RenderTexture(textureSize, textureSize, 16);
276.           m_ReflectionTexture.name = "__WaterReflection" + GetInstanceID();
277.           m_ReflectionTexture.isPowerOfTwo = true;
278.           m_ReflectionTexture.hideFlags = HideFlags.DontSave;
279.           m_OldReflectionTextureSize = textureSize;
280.       }
281.
282.       // Camera for reflection
283.       m_ReflectionCameras.TryGetValue(currentCamera, out reflectionCamera);
284.       if (!reflectionCamera) // catch both not-in-dictionary and in-dictionary-but-deleted-GO
285.       {
286.           GameObject go = new GameObject("Water Refl Camera id" + GetInstanceID() + " for " +
                currentCamera.GetInstanceID(), typeof(Camera), typeof(Skybox));
287.           reflectionCamera = go.GetComponent<Camera>();
288.           reflectionCamera.enabled = false;
289.           reflectionCamera.transform.position = transform.position;
290.           reflectionCamera.transform.rotation = transform.rotation;
291.           reflectionCamera.gameObject.AddComponent<FlareLayer>();
292.           go.hideFlags = HideFlags.HideAndDontSave;
293.           m_ReflectionCameras[currentCamera] = reflectionCamera;
294.       }
295.   }
296.
297.   if (mode >= WaterMode.Refractive)
298.   {
299.       // Refraction render texture
300.       if (!m_RefractionTexture || m_OldRefractionTextureSize != textureSize)
301.       {
302.           if (m_RefractionTexture)
303.           {
304.               DestroyImmediate(m_RefractionTexture);
305.           }
306.           m_RefractionTexture = new RenderTexture(textureSize, textureSize, 16);
307.           m_RefractionTexture.name = "__WaterRefraction" + GetInstanceID();
308.           m_RefractionTexture.isPowerOfTwo = true;

```

```

309.     m_RefractionTexture.hideFlags = HideFlags.DontSave;
310.     m_OldRefractionTextureSize = textureSize;
311.     }
312.
313.     // Camera for refraction
314.     m_RefractionCameras.TryGetValue(currentCamera, out refractionCamera);
315.     if (!refractionCamera) // catch both not-in-dictionary and in-dictionary-but-deleted-GO
316.     {
317.         GameObject go =
318.             new GameObject("Water Refr Camera id" + GetInstanceID() + " for " +
                currentCamera.GetInstanceID(),
319.                 typeof(Camera), typeof(Skybox));
320.         refractionCamera = go.GetComponent<Camera>();
321.         refractionCamera.enabled = false;
322.         refractionCamera.transform.position = transform.position;
323.         refractionCamera.transform.rotation = transform.rotation;
324.         refractionCamera.gameObject.AddComponent<FlareLayer>();
325.         go.hideFlags = HideFlags.HideAndDontSave;
326.         m_RefractionCameras[currentCamera] = refractionCamera;
327.     }
328.     }
329.     }
330.
331.     WaterMode GetWaterMode()
332.     {
333.         if (m_HardwareWaterSupport < waterMode)
334.         {
335.             return m_HardwareWaterSupport;
336.         }
337.         return waterMode;
338.     }
339.
340.     WaterMode FindHardwareWaterSupport()
341.     {
342.         if (!GetComponent<Renderer>())
343.         {
344.             return WaterMode.Simple;
345.         }
346.
347.         Material mat = GetComponent<Renderer>.sharedMaterial;
348.         if (!mat)
349.         {
350.             return WaterMode.Simple;
351.         }
352.
353.         string mode = mat.GetTag("WATERMODE", false);

```

```

354.   if (mode == "Refractive")
355.   {
356.   return WaterMode.Refractive;
357.   }
358.   if (mode == "Reflective")
359.   {
360.   return WaterMode.Reflective;
361.   }
362.
363.   return WaterMode.Simple;
364.   }
365.
366.   // Given position/normal of the plane, calculates plane in camera space.
367.   Vector4 CameraSpacePlane(Camera cam, Vector3 pos, Vector3 normal, float sideSign)
368.   {
369.   Vector3 offsetPos = pos + normal * clipPlaneOffset;
370.   Matrix4x4 m = cam.worldToCameraMatrix;
371.   Vector3 cpos = m.MultiplyPoint(offsetPos);
372.   Vector3 cnormal = m.MultiplyVector(normal).normalized * sideSign;
373.   return new Vector4(cnormal.x, cnormal.y, cnormal.z, -Vector3.Dot(cpos, cnormal));
374.   }
375.
376.   // Calculates reflection matrix around the given plane
377.   static void CalculateReflectionMatrix(ref Matrix4x4 reflectionMat, Vector4 plane)
378.   {
379.   reflectionMat.m00 = (1F - 2F * plane[0] * plane[0]);
380.   reflectionMat.m01 = (-2F * plane[0] * plane[1]);
381.   reflectionMat.m02 = (-2F * plane[0] * plane[2]);
382.   reflectionMat.m03 = (-2F * plane[3] * plane[0]);
383.
384.   reflectionMat.m10 = (-2F * plane[1] * plane[0]);
385.   reflectionMat.m11 = (1F - 2F * plane[1] * plane[1]);
386.   reflectionMat.m12 = (-2F * plane[1] * plane[2]);
387.   reflectionMat.m13 = (-2F * plane[3] * plane[1]);
388.
389.   reflectionMat.m20 = (-2F * plane[2] * plane[0]);
390.   reflectionMat.m21 = (-2F * plane[2] * plane[1]);
391.   reflectionMat.m22 = (1F - 2F * plane[2] * plane[2]);
392.   reflectionMat.m23 = (-2F * plane[3] * plane[2]);
393.
394.   reflectionMat.m30 = 0F;
395.   reflectionMat.m31 = 0F;
396.   reflectionMat.m32 = 0F;
397.   reflectionMat.m33 = 1F;
398.   }
399.   }

```

400. }

ΠΑΡΑΡΤΗΜΑ Β : Script “Water Volume.cs”

```
1. using UnityEngine;
2. using System.Collections.Generic;
3. using WaterBuoyancy.Collections;
4.
5. namespace WaterBuoyancy
6. {
7. [RequireComponent(typeof(BoxCollider))]
8. [RequireComponent(typeof(MeshFilter))]
9. public class WaterVolume : MonoBehaviour
10. {
11. public const string TAG = "Water Volume";
12.
13. [SerializeField]
14. private float density = 1f;
15.
16. [SerializeField]
17. private int rows = 10;
18.
19. [SerializeField]
20. private int columns = 10;
21.
22. [SerializeField]
23. private float quadSegmentSize = 1f;
24.
25. //[SerializeField]
26. //private Transform debugTrans; // Only for debugging
27.
28. private Mesh mesh;
29. private Vector3[] meshLocalVertices;
30. private Vector3[] meshWorldVertices;
31.
32. public float Density
33. {
34. get
35. {
36. return this.density;
37. }
38. }
39.
40. public int Rows
41. {
```

```
42. get
43. {
44. return this.rows;
45. }
46. }
47.
48. public int Columns
49. {
50. get
51. {
52. return this.columns;
53. }
54. }
55.
56. public float QuadSegmentSize
57. {
58. get
59. {
60. return this.quadSegmentSize;
61. }
62. }
63.
64. public Mesh Mesh
65. {
66. get
67. {
68. if (this.mesh == null)
69. {
70. this.mesh = this.GetComponent<MeshFilter>().mesh;
71. }
72.
73. return this.mesh;
74. }
75. }
76.
77. protected virtual void Awake()
78. {
79. this.CacheMeshVertices();
80. }
81.
82. protected virtual void Update()
83. {
84. this.CacheMeshVertices();
85. }
86.
87. protected virtual void OnDrawGizmosSelected()
```

```

88. {
89. Gizmos.color = Color.green;
90. Gizmos.matrix = this.transform.localToWorldMatrix;
91.
92. Gizmos.DrawWireCube(this.GetComponent<BoxCollider>().center,
    this.GetComponent<BoxCollider>().size);
93. }
94.
95. protected virtual void OnDrawGizmos()
96. {
97. if (!Application.isPlaying)
98. {
99. Gizmos.color = Color.cyan - new Color(0f, 0f, 0f, 0.75f);
100.     Gizmos.matrix = this.transform.localToWorldMatrix;
101.
102.     Gizmos.DrawCube(this.GetComponent<BoxCollider>().center - Vector3.up * 0.01f,
        this.GetComponent<BoxCollider>().size);
103.
104.     Gizmos.color = Color.cyan - new Color(0f, 0f, 0f, 0.5f);
105.     Gizmos.DrawWireCube(this.GetComponent<BoxCollider>().center,
        this.GetComponent<BoxCollider>().size);
106.
107.     Gizmos.matrix = Matrix4x4.identity;
108.     }
109.     else
110.     {
111.     Draw surface normal
112.     ar vertices = this.meshWorldVertices;
113.     ar triangles = this.Mesh.triangles;
114.     or (int i = 0; i < triangles.Length; i += 3)
115.
116.     Gizmos.color = Color.white;
117.     Gizmos.DrawLine(vertices[triangles[i + 0]], vertices[triangles[i + 1]]);
118.     Gizmos.DrawLine(vertices[triangles[i + 1]], vertices[triangles[i + 2]]);
119.     Gizmos.DrawLine(vertices[triangles[i + 2]], vertices[triangles[i + 0]]);
120.     Vector3 center = MathfUtils.GetAveragePoint(vertices[triangles[i + 0]], vertices[triangles[i +
        1]], vertices[triangles[i + 2]]);
121.     Vector3 normal = this.GetSurfaceNormal(center);
122.     Gizmos.color = Color.green;
123.     Gizmos.DrawLine(center, center + normal);
124.
125.     Draw mesh vertices
126.     f (this.meshWorldVertices != null)
127.
128.     for (int i = 0; i < this.meshWorldVertices.Length; i++)
129.     {

```

```

130. DebugUtils.DrawPoint(this.meshWorldVertices[i], Color.red);
131. }

132. Test GetSurroundingTrianglePolygon(Vector3 worldPoint);
133. f (debugTrans != null)
134.
135. Gizmos.color = Color.blue;
136. Gizmos.DrawSphere(debugTrans.position, 0.1f);
137. var point = debugTrans.position;
138. var triangle = this.GetSurroundingTrianglePolygon(point);
139. if (triangle != null)
140. {
141. Gizmos.color = Color.red;
142. Gizmos.DrawLine(triangle[0], triangle[1]);
143. Gizmos.DrawLine(triangle[1], triangle[2]);
144. Gizmos.DrawLine(triangle[2], triangle[0]);
145. }
146.
147. }
148. }
149.
150. public Vector3[] GetSurroundingTrianglePolygon(Vector3 worldPoint)
151. {
152. Vector3 localPoint = this.transform.InverseTransformPoint(worldPoint);
153. int x = Mathf.CeilToInt(localPoint.x / this.QuadSegmentSize);
154. int z = Mathf.CeilToInt(localPoint.z / this.QuadSegmentSize);
155. if (x <= 0 || z <= 0 || x >= (this.Columns + 1) || z >= (this.Rows + 1))
156. {
157. return null;
158. }
159.
160. Vector3[] trianglePolygon = new Vector3[3];
161. if ((worldPoint - this.meshWorldVertices[this.GetIndex(z, x)]).sqrMagnitude <
162. ((worldPoint - this.meshWorldVertices[this.GetIndex(z - 1, x - 1)]).sqrMagnitude))
163. {
164. trianglePolygon[0] = this.meshWorldVertices[this.GetIndex(z, x)];
165. }
166. else
167. {
168. trianglePolygon[0] = this.meshWorldVertices[this.GetIndex(z - 1, x - 1)];
169. }
170.
171. trianglePolygon[1] = this.meshWorldVertices[this.GetIndex(z - 1, x)];
172. trianglePolygon[2] = this.meshWorldVertices[this.GetIndex(z, x - 1)];
173.

```

```

174. return trianglePolygon;
175. }

176. public Vector3[] GetClosestPointsOnWaterSurface(Vector3 worldPoint, int pointsCount)
177. {
178.     MinHeap<Vector3> allPoints = new MinHeap<Vector3>(new
        Vector3HorizontalDistanceComparer(worldPoint));
179.     for (int i = 0; i < this.meshWorldVertices.Length; i++)
180.     {
181.         allPoints.Add(this.meshWorldVertices[i]);
182.     }
183.
184.     Vector3[] closestPoints = new Vector3[pointsCount];
185.     for (int i = 0; i < closestPoints.Length; i++)
186.     {
187.         closestPoints[i] = allPoints.Remove();
188.     }
189.
190.     return closestPoints;
191. }
192.
193. public Vector3 GetSurfaceNormal(Vector3 worldPoint)
194. {
195.     Vector3[] meshPolygon = this.GetSurroundingTrianglePolygon(worldPoint);
196.     if (meshPolygon != null)
197.     {
198.         Vector3 planeV1 = meshPolygon[1] - meshPolygon[0];
199.         Vector3 planeV2 = meshPolygon[2] - meshPolygon[0];
200.         Vector3 planeNormal = Vector3.Cross(planeV1, planeV2).normalized;
201.         if (planeNormal.y < 0f)
202.         {
203.             planeNormal *= -1f;
204.         }
205.
206.         return planeNormal;
207.     }
208.
209.     return this.transform.up;
210. }
211.
212. public float GetWaterLevel(Vector3 worldPoint)
213. {
214.     Vector3[] meshPolygon = this.GetSurroundingTrianglePolygon(worldPoint);
215.     if (meshPolygon != null)
216.     {

```

```

217. Vector3 planeV1 = meshPolygon[1] - meshPolygon[0];
218. Vector3 planeV2 = meshPolygon[2] - meshPolygon[0];
219. Vector3 planeNormal = Vector3.Cross(planeV1, planeV2).normalized;
220. if (planeNormal.y < 0f)
221. {
222. planeNormal *= -1f;
223. }
224.
225. // Plane equation
226. float yOnWaterSurface = -(worldPoint.x * planeNormal.x) - (worldPoint.z * planeNormal.z)
+ Vector3.Dot(meshPolygon[0], planeNormal) / planeNormal.y;
227. //Vector3 pointOnWaterSurface = new Vector3(point.x, yOnWaterSurface, point.z);
228. //DebugUtils.DrawPoint(pointOnWaterSurface, Color.magenta);
229.
230. return yOnWaterSurface;
231. }
232.
233. return this.transform.position.y;
234. }
235.
236. public bool IsPointUnderWater(Vector3 worldPoint)
237. {
238. return this.GetWaterLevel(worldPoint) - worldPoint.y > 0f;
239. }
240.
241. private int GetIndex(int row, int column)
242. {
243. return row * (this.Columns + 1) + column;
244. }
245.
246. private void CacheMeshVertices()
247. {
248. this.meshLocalVertices = this.Mesh.vertices;
249. this.meshWorldVertices = this.ConvertPointsToWorldSpace(meshLocalVertices);
250. }
251.
252. private Vector3[] ConvertPointsToWorldSpace(Vector3[] localPoints)
253. {
254. Vector3[] worldPoints = new Vector3[localPoints.Length];
255. for (int i = 0; i < localPoints.Length; i++)
256. {
257. worldPoints[i] = this.transform.TransformPoint(localPoints[i]);
258. }
259.
260. return worldPoints;
261. }

```

```
262. private class Vector3HorizontalDistanceComparer : IComparer<Vector3>
263. {
264.     private Vector3 distanceToVector;
265.
266.     public Vector3HorizontalDistanceComparer(Vector3 distanceTo)
267.     {
268.         this.distanceToVector = distanceTo;
269.     }
270.
271.     public int Compare(Vector3 v1, Vector3 v2)
272.     {
273.         v1.y = 0;
274.         v2.y = 0;
275.         float v1Distance = (v1 - distanceToVector).sqrMagnitude;
276.         float v2Distance = (v2 - distanceToVector).sqrMagnitude;
277.
278.         if (v1Distance < v2Distance)
279.         {
280.             return -1;
281.         }
282.         else if (v1Distance > v2Distance)
283.         {
284.             return 1;
285.         }
286.         else
287.         {
288.             return 0;
289.         }
290.     }
291. }
292. }
293. }
```

ΠΑΡΑΡΤΗΜΑ C : Script “FloatingObject.cs”

```
1. using System.Collections.Generic;
2. using UnityEngine;
3.
4. namespace WaterBuoyancy
5. {
6. [RequireComponent(typeof(Collider))]
7. [RequireComponent(typeof(Rigidbody))]
8. [RequireComponent(typeof(MeshFilter))]
9. public class FloatingObject : MonoBehaviour
10. {
11. [SerializeField]
12. private bool calculateDensity = false;
13.
14. [SerializeField]
15. private float density = 0.75f;
16.
17. [SerializeField]
18. [Range(0f, 1f)]
19. private float normalizedVoxelSize = 0.5f;
20.
21. [SerializeField]
22. private float dragInWater = 1f;
23.
24. [SerializeField]
25. private float angularDragInWater = 1f;
26.
27. private WaterVolume water;
28. private new Collider collider;
29. private new Rigidbody rigidbody;
30. private float initialDrag;
31. private float initialAngularDrag;
32. private Vector3 voxelSize;
33. private Vector3[] voxels;
34.
35. protected virtual void Awake()
36. {
37. this.collider = this.GetComponent<Collider>();
38. this.rigidbody = this.GetComponent<Rigidbody>();
39.
40. this.initialDrag = this.rigidbody.drag;
41. this.initialAngularDrag = this.rigidbody.angularDrag;
42.
43. if (this.calculateDensity)
```

```

44. {
45. float objectVolume =
    MathfUtils.CalculateVolume_Mesh(this.GetComponent<MeshFilter>().mesh, this.transform);
46. this.density = this.rigidbody.mass / objectVolume;
47. }
48. }
49.
50. protected virtual void FixedUpdate()
51. {
52. if (this.water != null && this.voxels.Length > 0)
53. {
54. Vector3 forceAtSingleVoxel = this.CalculateMaxBuoyancyForce() / this.voxels.Length;
55. Bounds bounds = this.collider.bounds;
56. float voxelHeight = bounds.size.y * this.normalizedVoxelSize;
57.
58. float submergedVolume = 0f;
59. for (int i = 0; i < this.voxels.Length; i++)
60. {
61. Vector3 worldPoint = this.transform.TransformPoint(this.voxels[i]);
62.
63. float waterLevel = this.water.GetWaterLevel(worldPoint);
64. float deepLevel = waterLevel - worldPoint.y + (voxelHeight / 2f); // How deep is the voxel
65. float submergedFactor = Mathf.Clamp(deepLevel / voxelHeight, 0f, 1f); // 0 - voxel is fully out of
    the water, 1 - voxel is fully submerged
66. submergedVolume += submergedFactor;
67.
68. Vector3 surfaceNormal = this.water.GetSurfaceNormal(worldPoint);
69. Quaternion surfaceRotation = Quaternion.FromToRotation(this.water.transform.up,
    surfaceNormal);
70. surfaceRotation = Quaternion.Slerp(surfaceRotation, Quaternion.identity, submergedFactor);
71.
72. Vector3 finalVoxelForce = surfaceRotation * (forceAtSingleVoxel * submergedFactor);
73. this.rigidbody.AddForceAtPosition(finalVoxelForce, worldPoint);
74.
75. Debug.DrawLine(worldPoint, worldPoint + finalVoxelForce.normalized, Color.blue);
76. }
77.
78. submergedVolume /= this.voxels.Length; // 0 - object is fully out of the water, 1 - object is fully
    submerged
79.
80. this.rigidbody.drag = Mathf.Lerp(this.initialDrag, this.dragInWater, submergedVolume);
81. this.rigidbody.angularDrag = Mathf.Lerp(this.initialAngularDrag, this.angularDragInWater,
    submergedVolume);
82. }
83. }
84. protected virtual void OnTriggerEnter(Collider other)

```

```

85. {
86. if (other.CompareTag(WaterVolume.TAG))
87. {
88. this.water = other.GetComponent<WaterVolume>();
89. if (this.voxels == null)
90. {
91. this.voxels = this.CutIntoVoxels();
92. }
93. }
94. }
95.
96. protected virtual void OnTriggerExit(Collider other)
97. {
98. if (other.CompareTag(WaterVolume.TAG))
99. {
100.     this.water = null;
101.     }
102.     }
103.
104. protected virtual void OnDrawGizmos()
105. {
106.     if (this.voxels != null)
107.     {
108.         for (int i = 0; i < this.voxels.Length; i++)
109.         {
110.             Gizmos.color = Color.magenta - new Color(0f, 0f, 0f, 0.75f);
111.             Gizmos.DrawCube(this.transform.TransformPoint(this.voxels[i]), this.voxelSize * 0.8f);
112.         }
113.     }
114. }
115.
116. private Vector3 CalculateMaxBuoyancyForce()
117. {
118.     float objectVolume = this.rigidbody.mass / this.density;
119.     Vector3 maxBuoyancyForce = this.water.Density * objectVolume * -Physics.gravity;
120.
121.     return maxBuoyancyForce;
122. }
123.
124. private Vector3[] CutIntoVoxels()
125. {
126.     Quaternion initialRotation = this.transform.rotation;
127.     this.transform.rotation = Quaternion.identity;
128.
129.     Bounds bounds = this.collider.bounds;
130.     this.voxelSize.x = bounds.size.x * this.normalizedVoxelSize;

```

```

131.   this.voxelSize.y = bounds.size.y * this.normalizedVoxelSize;
132.   this.voxelSize.z = bounds.size.z * this.normalizedVoxelSize;
133.   int voxelsCountForEachAxis = Mathf.RoundToInt(1f / this.normalizedVoxelSize);
134.   List<Vector3> voxels = new List<Vector3>(voxelsCountForEachAxis *
      voxelsCountForEachAxis * voxelsCountForEachAxis);
135.
136.   for (int i = 0; i < voxelsCountForEachAxis; i++)
137.   {
138.     for (int j = 0; j < voxelsCountForEachAxis; j++)
139.     {
140.       for (int k = 0; k < voxelsCountForEachAxis; k++)
141.       {
142.         float pX = bounds.min.x + this.voxelSize.x * (0.5f + i);
143.         float pY = bounds.min.y + this.voxelSize.y * (0.5f + j);
144.         float pZ = bounds.min.z + this.voxelSize.z * (0.5f + k);
145.
146.         Vector3 point = new Vector3(pX, pY, pZ);
147.         if (ColliderUtils.IsPointInsideCollider(point, this.collider, ref bounds))
148.         {
149.           voxels.Add(this.transform.InverseTransformPoint(point));
150.         }
151.       }
152.     }
153.   }
154.
155.   this.transform.rotation = initialRotation;
156.
157.   return voxels.ToArray();
158. }
159. }
160. }

```

ΠΑΡΑΡΤΗΜΑ D : Script “Straw.cs”

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5.
6. public class straw : MonoBehaviour
7. {
8.     Collider m_Collider1; //o collider pou tha disable
9.     Collider m_Collider2;
10. public GameObject gm;
11.
12. public GameObject str1;
13. public GameObject str2;
14. public GameObject str3;
15. public GameObject str4;
16. public GameObject str5;
17. public GameObject str6;
18. public GameObject str7;
19. public GameObject str8;
20. public GameObject str9;
21. public GameObject str10;
22. public GameObject str11;
23. public GameObject str12;
24.
25. public GameObject soapSystem;
26. public GameObject cyl1;
27. public GameObject cyl2;
28. public GameObject dynamicObj;
29. public GameObject buttonSymperasma;
30.
31. public Text timerText; //gia ton timer
32. private float startTime; //gia ton timer
33. bool keepTiming; //gia ton timer
34. float Timer; //gia ton timer
35. int sum;
36.
37. void Start()
38. {
39.     str2.SetActive(false);
40.     str3.SetActive(false);
41.     str4.SetActive(false);
42.     str5.SetActive(false);
43.     str6.SetActive(false);
```

```

44. str7.SetActive(false);
45. str8.SetActive(false);
46. str9.SetActive(false);
47. str10.SetActive(false);
48. str11.SetActive(false);
49. str12.SetActive(false);
50. buttonSymperasma.SetActive(false);
51. sum=0;
52. }
53. void OnMouseDown()
54. {
55. m_Collider1=gm.GetComponent<Collider>();
56. m_Collider1.enabled = true;
57.
58. m_Collider2=GetComponent<Collider>();
59. m_Collider2.enabled = false;
60.
61. sum=sum+1;
62. StartTimer(); //ksekinaei i metrisi
63. }
64.
65. void Update () {
66. /*if(Time.time - startTime > 10) //stop timer se 7 deuterolepta
67. {
68. Debug.Log("Timer stopped at " + TimeToString(StopTimer()));
69. }*/
70. /*if(Time.time - startTime >1)
71. {
72. str1.SetActive(false);
73. str2.SetActive(true);
74. }*/
75. if (sum==1){ //(sum>0 && sum<3)
76. if(Time.time - startTime > 0.2)
77. {
78. str2.SetActive(false);
79. str3.SetActive(true);
80. }
81. if(Time.time - startTime >0.4)
82. {
83. str3.SetActive(false);
84. str4.SetActive(true);
85. }
86. if(Time.time - startTime >0.6)
87. {
88. str4.SetActive(false);
89. str5.SetActive(true);

```

```
90. }
91. if(Time.time - startTime >0.8)
92. {
93. str5.SetActive(false);
94. str6.SetActive(true);
95. }
96. if(Time.time - startTime >1)
97. {
98. str6.SetActive(false);
99. str7.SetActive(true);
100. }
101. if(Time.time - startTime >1.2)
102. {
103. str7.SetActive(false);
104. str8.SetActive(true);
105. }
106. if(Time.time - startTime >1.4)
107. {
108. str8.SetActive(false);
109. str9.SetActive(true);
110. }
111. if(Time.time - startTime >1.6)
112. {
113. str9.SetActive(false);
114. str10.SetActive(true);
115. }
116. if(Time.time - startTime >1.8)
117. {
118. str10.SetActive(false);
119. str11.SetActive(true);
120. }
121. if(Time.time - startTime >1.9)
122. {
123. str11.SetActive(false);
124. }
125. }
126. if (sum==2){
127. Destroy(gameObject, 2);
128. if(Time.time - startTime > 0.2)
129. {
130. str2.SetActive(false);
131. str3.SetActive(true);
132. }
133. if(Time.time - startTime >0.4)
134. {
135. str3.SetActive(false);
```

```
136. str4.SetActive(true);
137. }
138. if(Time.time - startTime >0.6)
139. {
140. str4.SetActive(false);
141. str5.SetActive(true);
142. }
143. if(Time.time - startTime >0.8)
144. {
145. str5.SetActive(false);
146. str6.SetActive(true);
147. }
148. if(Time.time - startTime >1)
149. {
150. str6.SetActive(false);
151. str7.SetActive(true);
152. }
153. if(Time.time - startTime >1.2)
154. {
155. str7.SetActive(false);
156. str8.SetActive(true);
157. }
158. if(Time.time - startTime >1.4)
159. {
160. str8.SetActive(false);
161. str9.SetActive(true);
162. }
163. if(Time.time - startTime >1.6)
164. {
165. str9.SetActive(false);
166. str10.SetActive(true);
167. }
168. if(Time.time - startTime >1.8)
169. {
170. str10.SetActive(false);
171. str11.SetActive(true);
172. }
173. if(Time.time - startTime >1.9)
174. {
175. str11.SetActive(false);
176.
177. soapSystem.SetActive(false);
178. cyl1.SetActive(false);
179. cyl2.SetActive(true);
180. dynamicObj.SetActive(false);
181. buttonSymperasma.SetActive(true);
```

```

182.
183.  m_Collider1=gm.GetComponent<Collider>();
184.  m_Collider1.enabled = false;
185.  }
186.  }
187.
188.  if(keepTiming){
189.  UpdateTime();
190.  }
191.  }
192.
193.  void UpdateTime(){
194.  Timer = Time.time - startTime;
195.  timerText.text = TimeToString(Timer);
196.  }
197.
198.  float StopTimer(){
199.  keepTiming = false;
200.  return Timer;
201.  }
202.
203.  void ResumeTimer(){
204.  keepTiming = true;
205.  startTime = Time.time-Timer;
206.  }
207.
208.  void StartTimer(){
209.  keepTiming = true;
210.  startTime = Time.time; //Time.time;
211.  }
212.
213.  string TimeToString(float t){
214.  string minutes = ((int) t / 60).ToString();
215.  string seconds = (t % 60 ).ToString("00");
216.  return minutes + ":" + seconds;
217.  }
218.  }

```

ΠΑΡΑΡΤΗΜΑ Ε : Script “Timer.cs”

```
1. using System.Collections;
2. using System.Collections.Generic;
3. using UnityEngine;
4. using UnityEngine.UI;
5. using UnityEngine.UI;
6.
7. public class timer : MonoBehaviour
8. {
9.     public GameObject symperasmataKoumpi;
10.    public Text timerText;
11.    private float startTime;
12.
13.    public Image img2;
14.    public Image img3;
15.    public Image img4;
16.    public Image img5;
17.    public Image img6;
18.    public Image img7;
19.    public Image img8;
20.    public Image img9;
21.    public Image img10;
22.    public Image img11;
23.    public Image img12;
24.    public Image img13;
25.    public Image img14;
26.
27.    public GameObject line1;
28.    public GameObject line2;
29.    public GameObject line3;
30.    public GameObject line4;
31.    public GameObject line5;
32.
33.    bool keepTiming;
34.    float Timer;
35.
36.    public psoff scriptpsoff;
37.
38.    public void Starttt () {
39.
40.        StartTimer();
41.    }
42.
43.    void Update () {
```

```

44.     if(Time.time - startTime > 301) //stop timer se deuterolepta (5 lepta --> 300 deuterolepta )
45.     {
46.         Debug.Log("Timer stopped at " + TimeToString(StopTimer()));
47.         //symperasmataKoumpi.SetActive(true);
48.         scriptpsoff.disablethepts(); //gia disable tou particle system
49.     }
50.     if(Time.time - startTime > 10)
51.     {
52.         img2.enabled=true;
53.     }
54.     if(Time.time - startTime > 20)
55.     {
56.         img3.enabled=true;
57.     }
58.     if(Time.time - startTime > 30)
59.     {
60.         img4.enabled=true;
61.     }
62.     if(Time.time - startTime > 40)
63.     {
64.         img5.enabled=true;
65.     }
66.     if(Time.time - startTime > 50)
67.     {
68.         img6.enabled=true;
69.     }
70.     if(Time.time - startTime > 60)
71.     {
72.         img7.enabled=true;
73.         line1.SetActive(true);
74.     }
75.     if(Time.time - startTime > 90)
76.     {
77.         img8.enabled=true;
78.     }
79.     if(Time.time - startTime > 120)
80.     {
81.         img9.enabled=true;
82.         line2.SetActive(true);
83.     }
84.     if(Time.time - startTime > 150)
85.     {
86.         img10.enabled=true;
87.     }
88.     if(Time.time - startTime > 180)
89.     {

```

```

90.     img11.enabled=true;
91.     line3.SetActive(true);
92. }
93. if(Time.time - startTime > 240)
94. {
95.     img12.enabled=true;
96.     line4.SetActive(true);
97. }
98. if(Time.time - startTime > 270)
99. {
100.     img13.enabled=true;
101. }
102. if(Time.time - startTime > 300)
103. {
104.     img14.enabled=true;
105.     line5.SetActive(true);
106.     symperasmataKoumpi.SetActive(true);
107. }
108. if(keepTiming){
109.     UpdateTime();
110. }
111. }
112.
113. void UpdateTime(){
114.     Timer = Time.time - startTime;
115.     timerText.text = TimeToString(Timer);
116. }
117.
118. float StopTimer(){
119.     keepTiming = false;
120.     return Timer;
121. }
122.
123. void ResumeTimer(){
124.     keepTiming = true;
125.     startTime = Time.time-Timer;
126. }
127.
128. void StartTimer(){
129.     keepTiming = true;
130.     startTime = Time.time; //Time.time;
131. }
132.
133. string TimeToString(float t){
134.     string minutes = ((int) t / 60).ToString();
135.     string seconds = (t % 60 ).ToString("00");

```

```
136.         return minutes + ":" + seconds;
137.     }
138.
139. }
```