

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΠΑΙΧΝΙΔΙ ΕΚΤΕΤΑΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ
ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ ΓΙΑ ΤΟ
ΜΑΘΗΜΑ ΤΗΣ ΦΥΣΙΚΗΣ Α' ΓΥΜΝΑΣΙΟΥ



Του φοιτητή
Βλαδίμηρος Σπυρίδης
Αρ. Μητρώου: 154536

Επιβλέπων
Ευκλείδης Κεραμόπουλος
Αναπληρωτής Καθηγητής

Θεσσαλονίκη 2021

Τίτλος Π.Ε. Παιχνίδι εκτεταμένης πραγματικότητας εκπαιδευτικού περιεχομένου για το μάθημα της φυσικής Α' Γυμνασίου
Κωδικός Π.Ε. 21213

Όνοματεπώνυμο φοιτητή Βλαδίμηρος Σπυρίδης
Όνοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος

Ημερομηνία ανάληψης Π.Ε. 26-3-2021

Ημερομηνία περάτωσης Π.Ε. 15-9-2021

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

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

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

Πρόλογος

Αποτελεί γεγονός, ότι η τεχνολογία γίνεται όλο και σε μεγαλύτερο βαθμό κομμάτι της καθημερινότητας, επηρεάζοντας πολλούς τομείς και κατά κύριο λόγο την επιστήμη, βοηθώντας σημαντικά στην ανάπτυξη και στην βελτίωση του τρόπου ζωής. Είναι πολύ σημαντικό το γεγονός ότι, με την βοήθεια της τεχνολογίας, η εκπαίδευση μπορεί να συνδυαστεί με την διασκέδαση και το παιχνίδι. Με την παρούσα Π.Ε. γίνεται η ένωση αυτών των δύο, της εκπαίδευσης και της διασκέδασης, με τη χρήση νέων τεχνολογιών που αρχίζουν να ενσωματώνονται στην καθημερινή ζωή του ανθρώπου, δηλαδή την επαύξηση της πραγματικότητας σε ένα σοβαρό παιχνίδι εκμάθησης.

Περίληψη

Η πτυχιακή εργασία έχει ως στόχο, έναν καλύτερο και πιο ευχάριστο τρόπο εκμάθησης του μαθήματος της φυσικής, στους μαθητές της Α' γυμνασίου, με τη χρήση της εκτεταμένης πραγματικότητας. Το παιχνίδι είναι διαθέσιμο για κινητά και tablets και περιέχει διαδραστικά πειράματα όλων των κεφαλαίων του βιβλίου της φυσικής. Οι Μαθητές πρέπει να σκανάρουν με την κάμερα του κινητού ή του tablet, ένα ή δύο QR codes, ανάλογα το πείραμα και να ολοκληρώσουν το παιχνίδι, έχοντας τις οδηγίες από το σχολικό βιβλίο. Η δημιουργία του προγράμματος έγινε με το Unity Game Engine σε συνδυασμό με το πακέτο Vuforia Engine, και όλα τα τρισδιάστατα μοντέλα κατασκευάστηκαν με το πρόγραμμα δημιουργίας τρισδιάστατων αντικειμένων, Blender.

Augmented reality educational game in the school subject of physics for the first grade of secondary education

Vladimiros Spyridis

Abstract

The purpose of the thesis is to teach the subject of physics for the first grade of secondary education, in a better and more pleasant way, with the use of augmented reality. The game is made for smartphones and tablets and has interactive experiments from every chapter of the book. The students have to scan with the camera of their smartphones or tablets, one or two QR codes, depending on the experiment, and they have to complete the game as directed from the school book. The creation of the application was accomplished with Unity Game Engine with the package Vuforia Engine and every model was formed with the 3D software Blender.

Περιεχόμενα

Πρόλογος	iii
Περίληψη	v
Abstract	vii
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xiii
Συντομογραφίες	xvii
Κεφάλαιο 1ο: Εισαγωγή	1
Κεφάλαιο 2ο: Εκτεταμένη πραγματικότητα και serious games	3
2.1 Εισαγωγή	3
2.2 Επαυξημένη πραγματικότητα	4
2.3 Εικονική πραγματικότητα	5
2.4 Μικτή πραγματικότητα	5
2.5 Χρήσεις του XR	5
2.5.1 Εκπαίδευση στα σχολεία	5
2.5.2 Ιατρική	6
2.5.3 Στρατός	7
2.5.4 Διαφήμιση και marketing	7
2.6 Χρήση της τεχνολογίας στην εκπαίδευση	8
2.7 Serious Games	8
2.8 Επίλογος	8
Κεφάλαιο 3ο: Εγκατάσταση προγραμμάτων, πακέτα και Ρυθμίσεις	9
3.1 Εισαγωγή	9
3.2 Unity	9
3.2.1 Εγκατασταση Unity Hub και Unity	9
3.2.2 Δημιουργία project	10
3.3 Blender	10
3.4 Package Manager	11
3.4.1 Πακέτο Device Simulator	11
3.4.2 Universal Render Pipeline	11
3.4.3 Visual Effect Graph	12
3.5 Vuforia	12
	iiix

3.5.1 Τρόπος εγκατάστασης Vuforia: SDK	13
3.5.2 Τρόπος εγκατάστασης Vuforia: Asset Store	13
3.5.3 Τρόπος εγκατάστασης Vuforia: Χειροκίνητος τρόπος εγκατάστασης	13
3.5.4 Vuforia camera	15
3.5.5 Δημιουργία QR codes	15
3.5.6 Δημιουργία Image Targets	16
3.5.7 Βάση εικόνων στο project	17
3.6 Ρυθμίσεις εφαρμογής	17
3.6.1 Project Settings	18
3.6.2 Build Settings	18
3.7 Επιλογος	18
Κεφάλαιο 4ο: Αρχικό μενού	21
4.1 Εισαγωγή	21
4.2 Οργάνωση φακέλων Scripts	21
4.3 Script: MainMenuController	21
4.4 Canvas και Menu Controller	23
4.5 Γραμματοσειρά	24
4.6 Scroll View	25
4.7 Main Menu Buttons	25
4.8 Φόντο, τρόπος εισαγωγής εικόνων και κουμπί εξόδου	26
4.9 Panel Κεφαλαίων	29
4.10 Panel Πειράματος	29
4.10.1 Scroll View Πειραμάτων	29
4.10.2 Έναρξη πειραμάτων και κατασκευή σκηνών	31
4.11 Επίλογος	31
Κεφάλαιο 5ο: 3D μοντέλα, σκίαση και ειδικά εφέ	33
5.1 Εισαγωγή	33
5.2 Κατασκευή 3D μοντέλων με το πρόγραμμα Blender	33
5.2.1 Δημιουργία 3D μοντέλου	33
5.2.2 UV Unwrap και χρωμάτση μοντέλου	34
5.2.3 Χαρακτήρας του παιχνιδιού	35
5.2.4 Από το Blender στο Unity	37
5.3 Universal RP στο Unity	37
5.3.1 Toon Shader	38
5.3.2 Interactable Shader	40
5.3.3 Transparent Shader	40

5.3.4 Ice Shader	40
5.3.5 Fire Shader	41
5.3.6 Thermometer Shader	42
5.3.7 Δημιουργία υλικού με τη χρήση σκίασης	43
5.4 Visual Effects στο Unity	44
5.5 Επίλογος	46
Κεφάλαιο 6ο: Κοινές λειτουργίες σκηνών	47
6.1 Εισαγωγή	47
6.2 Vuforia: target images και κάμερα	47
6.3 Scripts	49
6.4 Tags και Layers	49
6.5 Navigation system και ο τρόπος κίνησης του χαρακτήρα	49
6.5.1 Components του χαρακτήρα	50
6.5.2 Script: PlayerController	50
6.5.3 Animation του χαρακτήρα και Script: PlayerAnimationController	52
6.6 UI σκηνών των πειραμάτων	54
6.7 Επίλογος	55
Κεφάλαιο 7ο: Πείραμα μέτρησης αποστάσεων, χρόνου και βάρους	57
7.1 Εισαγωγή πειραμάτων απόστασης	57
7.2 Εικονικά αντικείμενα της σκηνής	57
7.3 Measure Tape	58
7.4 Παραλλαγή χαρακτήρα 1-1	59
7.5 Εκτέλεση της σκηνής	60
7.6 Εισαγωγή πειραμάτων χρόνου	61
7.7 Ρολόγια	61
7.8 Πείραμα 2-1: μέτρηση χρόνου-εκκρεμές	62
7.9 Πείραμα 2-2: μέτρηση χρόνου-κεκλιμένο επίπεδο	62
7.10 Εισαγωγή πειραμάτων βάρους	63
7.11 Πείραμα 3-1: μέτρηση βάρους με ζυγό	63
7.12 Πείραμα 3-2 και 3-3: μέτρηση βάρους με ελατήριο	63
7.13 Επίλογος	65
Κεφάλαιο 8ο: Πειράματα μέτρησης θερμοκρασίας και ηλεκτρομαγνητισμού	67
8.1 Εισαγωγή θερμότητας	67
8.2 Θερμόμετρο	67
8.3 Πείραμα 4-1: Μέτρηση θερμοκρασίας νερού με κενό θερμόμετρο	69
8.4 Πείραμα 5-1: Εξισορρόπηση θερμοκρασίας	69
8.5 Πείραμα 6-1: Μέτρηση θερμοκρασίας νερού με θερμόμετρο	70

8.6 Πείραμα 7-1: Πήξη υγρών σε καπάκια	70
8.7 Πείραμα 7-2: Πήξη υγρών σε μπουκάλια	70
8.8 Πείραμα 7-3: Μέτρηση θερμοκρασίας σε διαφορετικά επίπεδα	71
8.9 Πείραμα 8-1: Αλλαγή θερμοκρασίας λόγω φωτός	72
8.10 Πείραμα 8-2: Αλλαγή θερμοκρασίας λόγω σκουρόχρωμης επιφάνειας	72
8.11 Πείραμα 9-1: Φαινόμενο του θερμοκηπίου	72
8.12 Εισαγωγή ηλεκτρομαγνητισμού	73
8.13 Πειράματα του κεφαλαίου 10 του βιβλίου	73
8.14 Πείραμα 11-1: Από τον ηλεκτρισμό στον ηλεκτρομαγνητισμό	74
8.15 Πείραμα 12-1: Από τον ηλεκτρομαγνητισμό στον ηλεκτρισμό	75
8.16 Επίλογος	76
Κεφάλαιο 9ο: Τελικά συμπεράσματα και βελτιώσεις της Π.Ε.	77
ΒΙΒΛΙΟΓΡΑΦΙΑ	78
ΠΑΡΑΡΤΗΜΑ Α' :Αρχείο Mainlight.hlsl	80
ΠΑΡΑΡΤΗΜΑ Β' :Script PlayerController	80

Κατάλογος Σχημάτων

Σχήμα 2.1: Πεδία της εκτεταμένης πραγματικότητας	3
Σχήμα 2.2: Παιχνίδι AR: Pokemon GO	4
Σχήμα 2.3: Παιχνίδι VR: Beat Saber	4
Σχήμα 2.4: Χρήση του AR στην εκπαίδευση	6
Σχήμα 2.5: Χρήση του XR στην ιατρική	6
Σχήμα 2.6: Marketing της Ferrari με την βοήθεια του AR	7
Σχήμα 3.1: Unity Hub, Install Version	10
Σχήμα 3.2: Install Android Builder Support	10
Σχήμα 3.3: Create Project	11
Σχήμα 3.4: Package Manager	12
Σχήμα 3.5: Vuforia Manual Download	13
Σχήμα 3.6: Add Package From Disk	14
Σχήμα 3.7: Vuforia Behaviour	14
Σχήμα 3.8: Δημιουργία κλειδιού	14
Σχήμα 3.9: Απόκτηση κλειδιού	14
Σχήμα 3.10: Vuforia Configurations	15
Σχήμα 3.11: Target Manager	16
Σχήμα 3.12: Add Target	16
Σχήμα 3.13: Download Database	17
Σχήμα 3.14: Project Settings, Android Version	18
Σχήμα 3.15: Switch Platform to Android	19
Σχήμα 4.1: Φάκελος Scripts	22
Σχήμα 4.2: Script MainMenuController	22
Σχήμα 4.3: Ιεραρχία Καμβά	23
Σχήμα 4.4: Add Component	24
Σχήμα 4.5: Font Asset	24
Σχήμα 4.6: Ρυθμίσεις Content	25
Σχήμα 4.7: Κουμπιά του Main Menu Panel	26
Σχήμα 4.8: Αρχική Οθόνη Εφαρμογής	27
Σχήμα 4.9: Ρυθμίσεις εικόνας	27
Σχήμα 4.10: UI Icons	28

Σχήμα 4.11: Εικόνες πειραμάτων	28
Σχήμα 4.12: Panel Κεφαλαίων	28
Σχήμα 4.13: Scroll Views όλων των πειραμάτων	29
Σχήμα 4.14: Scroll View Πειράματος 2-2	30
Σχήμα 4.15: Φάκελος Σκηνών	30
Σχήμα 4.16: Εισαγωγή και Αρίθμηση Σκηνών στο Scenes In Build	31
Σχήμα 5.1: Αρχική Οθόνη του Blender	33
Σχήμα 5.2: UV Editor	34
Σχήμα 5.3: Texture Paint	35
Σχήμα 5.4: Rigging του χαρακτήρα	36
Σχήμα 5.5: Animations του χαρακτήρα	36
Σχήμα 5.6: Export αρχείο .fbx	37
Σχήμα 5.7: Εισαγωγή Renderer στις Ρυθμίσεις του Project	38
Σχήμα 5.8: Αριστερά: Αντικείμενο με Toon Shader. Δεξιά: Αντικείμενο με προεπιλεγμένο Material	38
Σχήμα 5.9: Toon Shader Graph Nodes	39
Σχήμα 5.10: Επιλογές Custom Function για τη Χρήση του Mainlight.hlsl	39
Σχήμα 5.11: Interactable Shader Graph Nodes	40
Σχήμα 5.12: Transparent Shader Graph Nodes	41
Σχήμα 5.13: Ice Shader Graph Node	41
Σχήμα 5.14: Fire Shader Graph Nodes	42
Σχήμα 5.15: Thermometer Shader Graph Nodes	42
Σχήμα 5.16: Αλλαγή της μεταβλητής Height	43
Σχήμα 5.17: Δημιουργία Material από Shader Graph	44
Σχήμα 5.18: Αλλαγή Shader ενός Material	44
Σχήμα 5.19: VFX Graph Nodes	45
Σχήμα 5.20: Bubble Particles	45
Σχήμα 6.1: Target Images	47
Σχήμα 6.2: DestroyAfterTime Script	48
Σχήμα 6.3: Interactable Script	48
Σχήμα 6.4: Player Components	51
Σχήμα 6.5: Διαχωρισμός των Animations του Χαρακτήρα	52
Σχήμα 6.6: Animator Χαρακτήρα	53

Σχήμα 6.7: PlayerAnimationController Script	53
Σχήμα 6.8: UI Canvas	54
Σχήμα 6.9: GameUIController Script	55
Σχήμα 7.1: Αντικείμενα Πειράματος 1-1	57
Σχήμα 7.2: Αντικείμενο Measure Tape και Components του	57
Σχήμα 7.3: Script MeasureTape πρώτο κομμάτι	58
Σχήμα 7.4: Script MeasureTape δεύτερο κομμάτι	59
Σχήμα 7.5: Components του αντικειμένου Player1_1	59
Σχήμα 7.6: Script Player1_1	60
Σχήμα 7.7: Εκτέλεση του Πειράματος 1-1	61
Σχήμα 7.8: Μοντέλα Ρολογιών	62
Σχήμα 7.9: Εκτέλεση του Πειράματος 2-1	62
Σχήμα 7.10: Εκτέλεση του Πειράματος 2-2	63
Σχήμα 7.11: Εκτέλεση του Πειράματος 3-1	64
Σχήμα 7.12: Εκτέλεση του Πειράματος 3-2	64
Σχήμα 7.13: Εκτέλεση του Πειράματος 3-3	64
Σχήμα 8.1: Μοντέλο Θερμομέτρου στο Μικρό QR code	67
Σχήμα 8.2: Script: Thermometer. Ένα από τα Δύο	68
Σχήμα 8.3: Script: Thermometer. Δύο από τα Δύο	68
Σχήμα 8.4: Παράδειγμα Πειράματος 4-1 Αριστερά: Το Νερό Βράζει. Δεξιά: Το Νερό Ψύχεται	69
Σχήμα 8.5: Παράδειγμα Πειράματος 5-1. Εξισορρόπηση Θερμοκρασίας	70
Σχήμα 8.6: Παράδειγμα Πειράματος 7-1. Αριστερά: Τα Καπάκια Είναι Παγωμένα. Δεξιά: Τα Παγάκια Είναι Μέσα στα Δοχεία με τα Υγρά	71
Σχήμα 8.7: Παράδειγμα Πειράματος 7-3. Πάνω: Η Θερμοκρασία είναι 0. Μέση: Η Θερμοκρασία είναι 2. Κάτω: Η Θερμοκρασία είναι 4	71
Σχήμα 8.8: Παράδειγμα Πειράματος 8-2. Φως και Θερμοκρασία	72
Σχήμα 8.9: Παράδειγμα Πειράματος 9-1. Φαινόμενο του Θερμοκηπίου	73
Σχήμα 8.10: Πειράματα Κεφαλαίου 10	74
Σχήμα 8.11: Πειράματα Κεφαλαίου 11	74
Σχήμα 8.12: Μέθοδοι CheckTouching και TurnOnLamp του Script SpinWithTouch	75

Συντομογραφίες

Π.Ε.	Πτυχιακή Εργασία
AR	Επαυξημένη Πραγματικότητα
VR	Εικονική Πραγματικότητα
MR	Μικτή Πραγματικότητα
XR	Εκτεταμένη Πραγματικότητα
UI	User Interface
3D	Τρισδιάστατο
URP	Universal Render Pipeline
VFX	Visual Effects

Κεφάλαιο 1ο: Εισαγωγή

Η πτυχιακή εργασία, με τίτλο “ΠΑΙΧΝΙΔΙ ΕΚΤΕΤΑΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΤΗΣ ΦΥΣΙΚΗΣ Α’ ΓΥΜΝΑΣΙΟΥ” παρουσιάζει ένα παιχνίδι εκτεταμένης πραγματικότητας με μαθησιακό περιεχόμενο για το μάθημα της φυσικής της Α’ γυμνασίου. Η εφαρμογή δημιουργήθηκε στο περιβάλλον Unity σε συνδυασμό με το πακέτο Vuforia, που της επιτρέπει χρησιμοποιώντας την κάμερα μιας φορητής συσκευής όπως ένα smartphone ή ένα tablet, να ανιχνεύει εικόνες στο φυσικό περιβάλλον και να ξεδιπλώνει τα πειράματα του βιβλίου, μπροστά στους μαθητές του γυμνασίου, στις οθόνες της φορητής τους συσκευής σε πραγματικό χρόνο.

Στην παρούσα Π.Ε. αναφέρονται λεπτομέρειες για την τεχνολογία της εκτεταμένης πραγματικότητας(XR), για τις υποκατηγορίες της, την εικονική πραγματικότητα(VR), την επαυξημένη πραγματικότητα(AR) και την μικτή πραγματικότητα(MR) καθώς και για την χρήση αυτών στα serious games.

Στην συνέχεια παρουσιάζεται ο τρόπος δημιουργίας της εφαρμογής με την χρήση του Unity και του Blender, δύο ισχυρών προγραμμάτων δημιουργίας ψηφιακών παιχνιδιών, και ο αναλυτικός τρόπος σχεδίασης των γραφικών και των σκηνών που εμφανίζονται στις οθόνες των φορητών συσκευών των μαθητών.

Η δομή της Π.Ε. παρουσιάζεται παρακάτω:

Στο πρώτο κεφάλαιο καλύπτονται τα βασικά στοιχεία της Π.Ε. καθώς και μία περίληψη των κεφαλαίων που ακολουθούν.

Στο δεύτερο κεφάλαιο της Π.Ε. αναφέρονται αναλυτικά οι τεχνολογίες που χρειάστηκαν για την υλοποίηση της εφαρμογής καθώς και θεωρητικά κομμάτια αυτών των τεχνολογιών. Πιο συγκεκριμένα, παρουσιάζεται η εκτεταμένη πραγματικότητα(XR) με τις υποκατηγορίες της, την εικονική πραγματικότητα(VR), την επαυξημένη πραγματικότητα(AR) και την μικτή πραγματικότητα(MR). Στην συνέχεια του κεφαλαίου υπάρχουν αναφορές για τα serious games και τα πλεονεκτήματά τους αντί της παραδοσιακής διαδικασίας εκπαίδευσης.

Στο τρίτο κεφάλαιο αναφέρονται τα προγράμματα που είναι απαραίτητα για την υλοποίηση της εφαρμογής, πληροφορίες για αυτά και οι τρόποι εγκατάστασής τους. Στην συνέχεια του κεφαλαίου υπάρχουν οι κατάλληλες οδηγίες για τη δημιουργία του project στο Unity και τις ρυθμίσεις που είναι απαραίτητες για την συγκεκριμένη εφαρμογή. Τέλος, υπάρχει η διαδικασία εγκατάστασης των πακέτων του project και πιο συγκεκριμένα πληροφορίες για το πακέτο Vuforia.

Στο τέταρτο κεφάλαιο υπάρχει η δομή του αρχικού μενού του παιχνιδιού καθώς και τρόποι υλοποίησης του UI λεπτομερώς.

Στο πέμπτο κεφάλαιο υπάρχει όλο το κομμάτι που σχετίζεται με την εμφάνιση του παιχνιδιού, δηλαδή οι τρόποι που δημιουργήθηκαν τα 3D αντικείμενα με το πρόγραμμα Blender, τα shaders με το URP και τα visual effects.

Στο έκτο κεφάλαιο αναφέρονται όλα τα κοινά scripts και λειτουργίες που υπάρχουν σε όλες τις σκηνές του project. Πιο αναλυτικά, οι βασικές ρυθμίσεις του Vuforia και ο τρόπος που λειτουργούν τα target

Κεφάλαιο 1

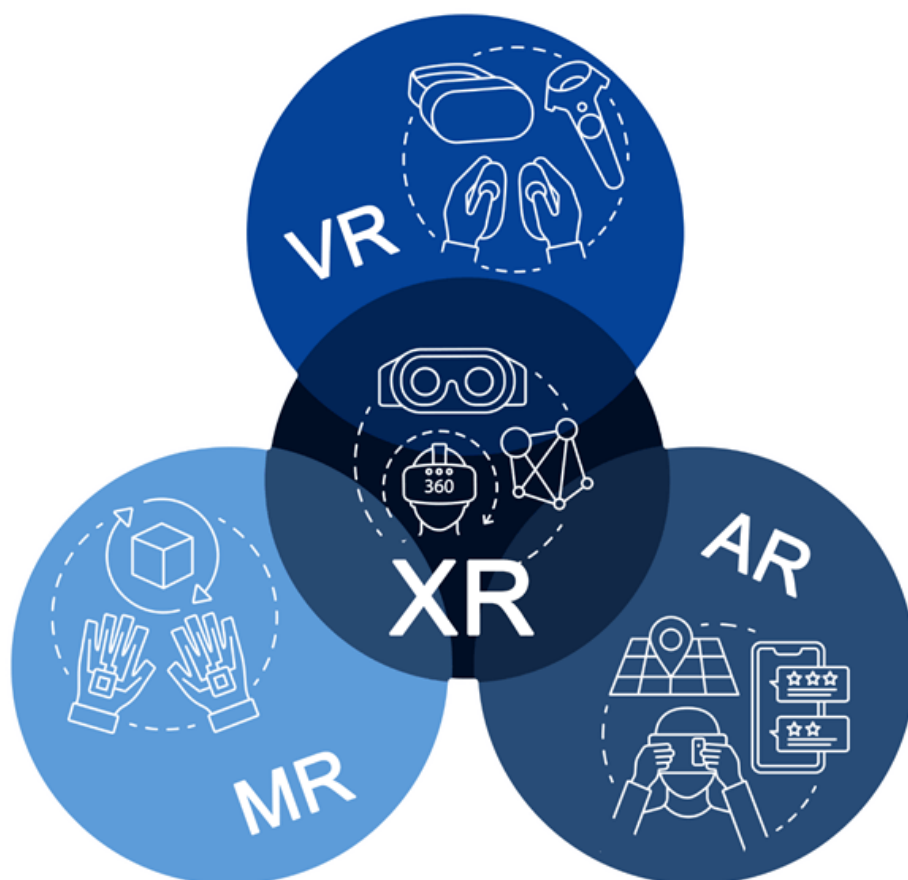
images καθώς και ο τρόπος που λειτουργεί ο κεντρικός χαρακτήρας του παιχνιδιού που αλληλεπιδρά με τα αντικείμενα των πειραμάτων.

Στο κεφάλαιο εφτά υπάρχει ο τρόπος δημιουργίας των πρώτων πειραμάτων, δηλαδή τα πειράματα του βιβλίου που σχετίζονται με την μέτρηση αποστάσεων, χρόνου και βάρους. Αυτά είναι τα πρώτα τρία κεφάλαια του σχολικού βιβλίου της φυσικής της Α' Γυμνασίου.

Στο όγδοο κεφάλαιο της Π.Ε. αναφέρεται ο τρόπος δημιουργίας των υπόλοιπων πειραμάτων. Τα κεφάλαια 4 μέχρι 9 του σχολικού βιβλίου της φυσικής της Α' Γυμνασίου σχετίζονται με την μέτρηση της θερμοκρασίας και την χρήση του θερμόμετρου. Τα τρία τελευταία κεφάλαια αφορούν τον ηλεκτρισμό και τον ηλεκτρομαγνητισμό.

Στο τελευταίο κεφάλαιο παρουσιάζονται τα τελικά συμπεράσματα που λήφθηκαν με το πέρας της εργασίας και τρόπους με τους οποίους η εφαρμογή μπορεί να βελτιωθεί.

Κεφάλαιο 2ο: Εκτεταμένη πραγματικότητα και serious games



Σχήμα 2.1: Πεδία της εκτεταμένης πραγματικότητας

2.1 Εισαγωγή

Στην εποχή που διανύουμε, ο ρυθμός ανάπτυξης της τεχνολογίας και της τεχνογνωσίας είναι ραγδαίος. Τα τελευταία χρόνια παρατηρείται εξάπλωση της εκτεταμένης πραγματικότητας (XR), ενός όρου ομπρέλας που περιέχει τρεις τεχνολογίες όπως στο σχήμα 2.1, στην καθημερινότητά μας σε διάφορους τομείς, από την διασκέδαση μέχρι την εκπαίδευση. Στην εκτεταμένη πραγματικότητα περιλαμβάνονται η επαυξημένη πραγματικότητα (AR)[1], η εικονική πραγματικότητα (VR)[2] και η μικτή πραγματικότητα (MR), τεχνολογίες που έχουν εξελιχθεί με την πάροδο του χρόνου, σε σημείο να προσφέρουν στον χρήστη ένα άκρως ρεαλιστικό αίσθημα κατά τη χρήση των εφαρμογών που χρησιμοποιούν τις παραπάνω τεχνολογίες. Καθώς τα αποτελέσματα που προσφέρει η εκτεταμένη πραγματικότητα είναι τόσο ζωντανά στα μάτια του χρήστη, είναι εύκολο να χρησιμοποιηθεί σε διάφορους τομείς με εκπαιδευτικό περιεχόμενο, για μια πιο ευχάριστη και ενόνητη εμπειρία. Στη συνέχεια του κεφαλαίου αναφέρονται αναλυτικά οι τέσσερις κατηγορίες, καθώς και η χρήση αυτών στην εκπαίδευση και σε άλλα πεδία της καθημερινότητας με την χρήση των σοβαρών παιχνιδιών (serious games).



Σχήμα 2.2: Παιχνίδι AR: Pokemon GO

2.2 Επαυξημένη πραγματικότητα

Η επαυξημένη πραγματικότητα(AR) είναι η ενίσχυση της εικόνας του πραγματικού κόσμου με τη χρήση κάποιας συσκευής, ενσωματώνοντας ήχους και ψηφιακά αντικείμενα, τρισδιάστατης μορφής και γραφικά διεπαφής χρήστη(UI) σε πραγματικό χρόνο όπως στο παιχνίδι Pokemon GO της εικόνας 2.2. Ο χρήστης μπορεί να αλληλεπιδράσει με τα αντικείμενα που δημιουργούνται στον επαυξημένο χώρο, με την βοήθεια της τεχνολογίας AR. Η συσκευή που χρειάζεται για να εκτελεστεί μια τέτοια εφαρμογή μπορεί να είναι ένα smartphone, ένα tablet ή ένας υπολογιστής με κάμερα. Απαραίτητες προϋποθέσεις για μια συσκευή επαυξημένης πραγματικότητας είναι ένας επεξεργαστής, μια κάμερα και μία οθόνη. Πάραυτα πρέπει να τηρείται το κομμάτι της ασφάλειας κατά τη χρήση μια τέτοιας εφαρμογής. Λόγω του ότι είναι απαραίτητη η χρήση της κάμερας για την καταγραφή του φυσικού χώρου, μπορεί να υπάρξει παραβίαση της ιδιωτικότητας.



Σχήμα 2.3: Παιχνίδι VR: Beat Saber

2.3 Εικονική πραγματικότητα

Ο όρος εικονική πραγματικότητα(VR) έχει εμφανιστεί σε πολλές ταινίες επιστημονικής φαντασίας, όμως η αλήθεια είναι ότι βρίσκεται πολύ κοντά στο σήμερα, καθώς συσκευές εικονικής πραγματικότητας πωλούνται για προσωπική χρήση καθημερινά, έχοντας ακόμα μεγάλο κόστος αγοράς το έτος 2021. Η εικονική πραγματικότητα ανήκει στην οικογένεια της εκτεταμένης πραγματικότητας

και είναι η δημιουργία ενός εικονικού περιβάλλοντος με την χρήση ειδικών масκών, αποκόπτοντας σχεδόν ολοκληρωτικά τον χρήστη από τον πραγματικό κόσμο. Σκοπός είναι η δημιουργία μιας προσομοίωσης στον εικονικό χώρο για την αλληλεπίδραση με αυτόν και τα 3D αντικείμενα που βρίσκονται εκεί, όπως στο σχήμα 2.3 που απεικονίζει ένα παιχνίδι εικονικής πραγματικότητας. Το υλικό hardware που είναι απαραίτητο για να δουλέψει μια VR εφαρμογή, αποτελείται από συσκευές που διαθέτουν γυροσκόπια και αισθητήρες κίνησης, για τον εντοπισμό του κεφαλιού, των χεριών και σε πιο σπάνιες περιπτώσεις, για τον εντοπισμό του σώματος. Στη μάσκα υπάρχουν 2 φακοί, υψηλής ευκρίνειας, ένας για το δεξί και ένας για το αριστερό μάτι, που έχουν ως ρόλο μικρών οθονών για να δημιουργηθεί η ψευδαίσθηση του 3D χώρου. Τα γυροσκόπια στα χέρια είναι ταυτόχρονα και χειριστήρια, για να μπορεί ο χρήστης να αλληλεπιδρά με τα αντικείμενα και να κινείται στον 3D, πατώντας τα κουμπιά που βρίσκονται στα δάχτυλά του.

2.4 Μικτή πραγματικότητα

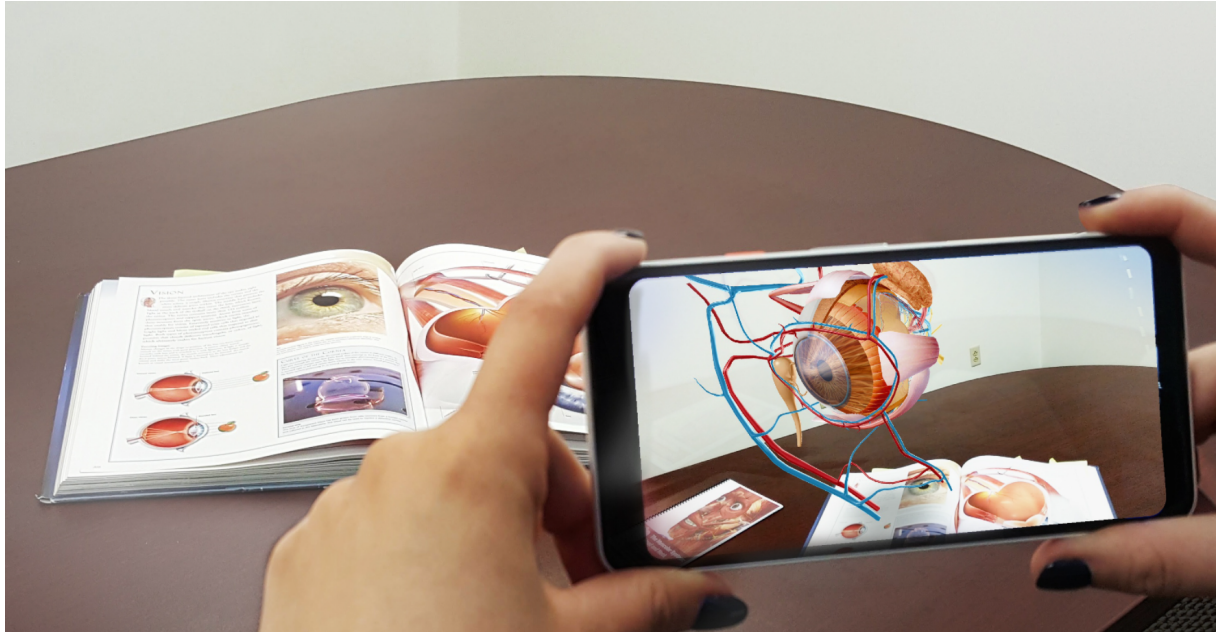
Η μικτή πραγματικότητα[3] είναι η συνδυαστική εκτέλεση των τεχνολογιών της επαυξημένης και της εικονικής πραγματικότητας, καθώς διαπλέκει το πραγματικό με το ψηφιακό περιβάλλον, αναμιγνύοντας τα φυσικά με τα πραγματικά αντικείμενα τα οποία συνυπάρχουν και αλληλεπιδρούν μεταξύ τους σε πραγματικό χρόνο. Για αυτούς τους λόγους, η εμπειρία που προσφέρει το MR είναι ρεαλιστική, διαδραστική και διασκεδαστική για τους χρήστες. Παρόλο που η τεχνολογία έχει επινοηθεί ως ιδέα πριν από περίπου 5 δεκαετίες, έχει αρχίσει να υλοποιείται τα τελευταία χρόνια στην καθημερινότητα και όχι απαραίτητα στην ακαδημαϊκή μελέτη. Η τεχνολογία είναι κατάλληλη για εκπαιδευτικά και εργασιακά περιβάλλοντα και χρησιμοποιείται κατά κύριο λόγο για την εκπαίδευση του προσωπικού και των μαθητών σε διάφορους τομείς. Ένα βασικό μειονέκτημα είναι η δημιουργία του προϊόντος και του συστήματος, καθώς και η ευχρηστία του όταν υπάρχει ευκολότερος τρόπος εκμάθησης της συγκεκριμένης εκπαιδευτικής διαδικασίας.

2.5 Χρήσεις του XR

Οι εφαρμογές που χρησιμοποιούν τις παραπάνω τεχνολογίες, δημιουργούνται για ομάδες και οργανισμούς παγκοσμίως σε ορισμένα πεδία, για μια πιο διαδραστική και αυθεντική εμπειρία.

2.5.1 Εκπαίδευση στα σχολεία

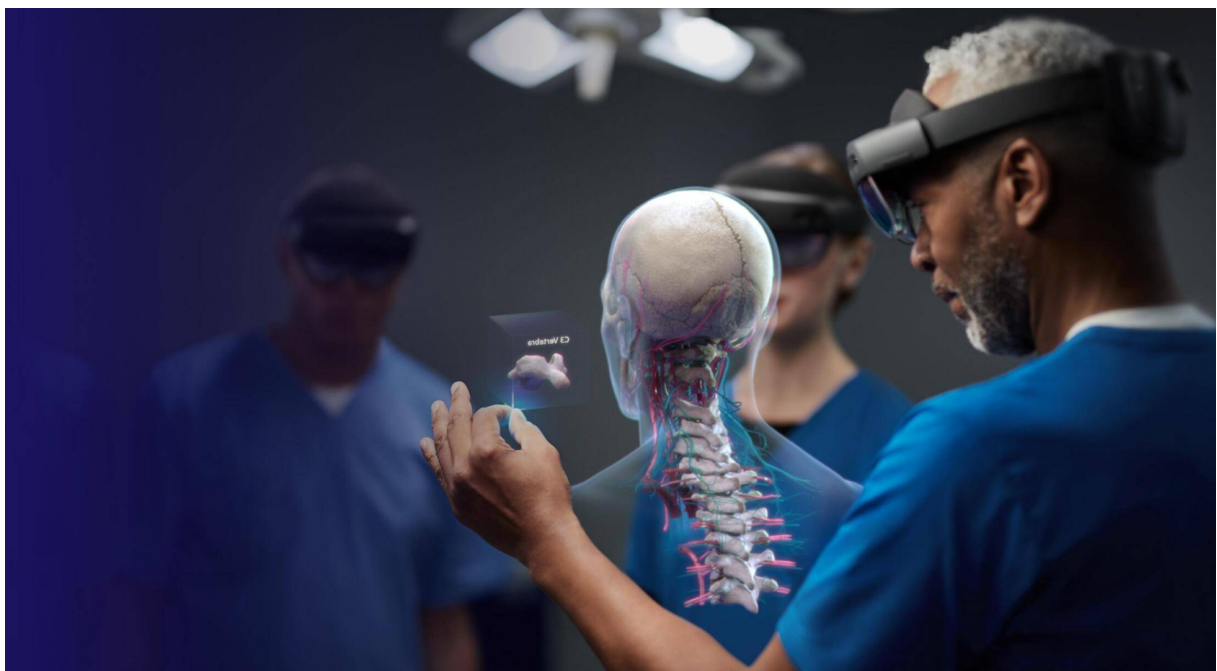
Η εκπαίδευση με την χρήση των τεχνολογιών XR και κατά κύριο λόγο την τεχνολογία AR(σχήμα 2.4), δίνει μια βελτιωμένη διαδραστική εμπειρία στους μαθητές και τους προσφέρει κίνητρο να παρακολουθήσουν με περισσότερη προσήλωση. Από την πλευρά των εκπαιδευτικών, η διαδικασία της εκμάθησης γίνεται ευκολότερη, καθώς μπορούν να φέρουν μέσα στην αίθουσα το εξωτερικό περιβάλλον, όπως για παράδειγμα πειράματα φυσικής με ασφαλέστερες συνθήκες υλοποίησης. Το γεγονός ότι τα πειράματα και η εκμάθηση γίνονται με αυτόν τον τρόπο, βελτιώνει την ομαδικότητα και την καλύτερη κατανόηση των μαθητών, λόγω του ότι δεν αποσπάται εύκολα η προσοχή τους από αυτά.



Σχήμα 2.4: Χρήση του AR στην εκπαίδευση

2.5.2 Ιατρική

Η XR τεχνολογία μπορεί εύκολα να βελτιώσει τον τρόπο που γίνονται οι χειρουργικές επεμβάσεις και οι κλινικές διαδικασίες, μειώνοντας το κόστος και αυξάνοντας την ασφάλεια της εκπαίδευσης του προσωπικού. Πέρα από αυτά μπορεί να δοκιμαστούν διάφορες νέες τεχνικές για επικίνδυνες μελλοντικές χειρουργικές επεμβάσεις. Με την τεχνολογία VR μπορούν να παρατηρούν και να μελετούν παραδείγματα από 3D ολογράμματα(σχήμα 2.5) με πολύ μικρό κόστος, βοηθώντας τους γιατρούς να κατανοούν καλύτερα την ανθρώπινη ανατομία.



Σχήμα 2.5: Χρήση του XR στην ιατρική

2.5.3 Στρατός

Σε δύσκολες εκπαιδευτικές αποστολές του στρατού υπάρχει η πιθανότητα κάποιας σοβαρής απώλειας αν η εκτέλεση δεν γίνει όπως είναι σχεδιασμένη εξ αρχής ή αν υπάρξει κάποια σοβαρή βλάβη στα όργανα ή τα οχήματα που χρησιμοποιούνται. Με την XR τεχνολογία η εκπαίδευση των στρατιωτικών μπορεί να γίνει σε ένα ασφαλές περιβάλλον με μικρότερο κόστος. Ένα παράδειγμα είναι τα εικονικά αεροπλάνα και ελικόπτερα καθώς και VR κράνη που επιτρέπουν στους χρήστες να “μεταφέρονται” σε ένα εικονικό πεδίο μάχης με 360-μοίρες όραση.

2.5.4 Διαφήμιση και marketing

Για να διαφημίσουν οι εταιρείες τα προϊόντα τους, επιθυμούν να τα κάνουν όσο πιο αξιομνημόνευτα γίνεται, για να μην ξεχαστούν από τους καταναλωτές τους. Μια τεχνική marketing που χρησιμοποιούν είναι η δημιουργία μιας διαδραστικής διαφήμισης με την τεχνολογία AR, για να ενθουσιάσουν και να κάνουν αξέχαστη την εμπειρία των καταναλωτών τους. Εμφανίζοντας 3D αντικείμενα σε smartphones με το προϊόν της εταιρείας ή σε μεγαλύτερες οθόνες για την επίδειξή τους, όπως για παράδειγμα ρούχα και αυτοκίνητα, προσφέρουν το αντικείμενο που επιθυμούν να πουλήσουν σε πραγματικό μέγεθος όπως στο σχήμα 2.6.



Σχήμα 2.6: Marketing της Ferrari με την βοήθεια του AR

2.6 Χρήση της τεχνολογίας στην εκπαίδευση

Με την εξέλιξη της τεχνολογίας, η σύγχρονη κοινωνία αναζητά πιο εύκολους και αποτελεσματικούς τρόπους εκμάθησης για την μετάδοση της γνώσης. Παγκοσμίως η αγορά ενός υπολογιστή ή smartphone με ίντερνετ είναι εφικτή από το μεγαλύτερο μέρος του πληθυσμού, και αυτό κάνει την εκπαίδευση με ηλεκτρονικά μέσα πολύ εύκολη. Έχοντας κάνει την τεχνολογία κομμάτι της ζωής μας, ο τρόπος διδασκαλίας μπορεί να γίνει απομακρυσμένα και με φυσική παρουσία, με χαμηλό κόστος για τους εκπαιδευτικούς και τους μαθητές όλων των ηλικιών.

2.7 Serious Games

Τα σοβαρά παιχνίδια[4] στις μέρες μας μπορούν να αντικαταστήσουν τους παραδοσιακούς τρόπους εκπαίδευσης, κάνοντας την διαδικασία της εκμάθησης ενδιαφέρον χωρίς να αποσπούν την προσοχή του χρήστη από το αντικείμενο. Τα serious games μπορούν να χρησιμοποιηθούν για την εκπαίδευση στους τομείς όπως ο στρατός, η σχολική εκπαίδευση, η εκπαίδευση ενός υπαλλήλου μιας εταιρείας κτλ. Η έννοια των serious games αναφέρεται κατα κύριο λόγο σε παιχνίδια ψηφιακής μορφής που έχουν ως πρωταρχικό σκοπό την εκπαίδευση. Το αποτέλεσμά τους είναι τις περισσότερες φορές πιο αποτελεσματικό από παραδοσιακές μεθόδους. Οι λόγοι είναι ότι τα σοβαρά παιχνίδια:

- βρίσκονται σε ελεγχόμενο περιβάλλον.
- είναι επαναλαμβανόμενα.
- έχουν μικρότερο κόστος.
- απαιτούν λιγότερο χρόνο για την εκπαίδευση.
- έχουν ως σκοπό την επίλυση προβλημάτων.
- προσομοιώνουν το κατάλληλο περιβάλλον ανάλογα την περίπτωση.

2.8 Επίλογος

Οι τεχνολογίες που αφορούν την εκτεταμένη πραγματικότητα αποτελούν καθοριστικό παράγοντα σε πολλούς τομείς της ζωής: δημιουργούν καινούριους τρόπους εκμάθησης, δίνουν στους χρήστες πολλές νέες δυνατότητες, κρατώντας παράλληλα χαμηλό το κόστος. Για τους λόγους αυτούς, σε πολλές περιπτώσεις προτιμώνται από τις παραδοσιακές μεθόδους, πράγμα που εξετάζει και η παρούσα Π.Ε, αξιοποιώντας τις τεχνολογίες εκτεταμένης πραγματικότητας στο χώρο της εκπαίδευσης.

Κεφάλαιο 3ο: Εγκατάσταση προγραμμάτων, πακέτα και Ρυθμίσεις

3.1 Εισαγωγή

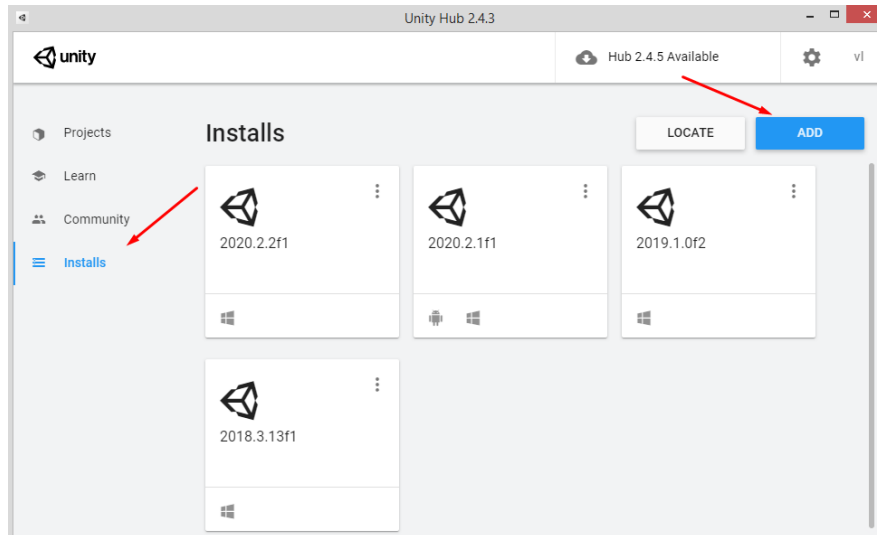
Στο παρόν κεφάλαιο υπάρχουν αναλυτικά τα βήματα για το κατέβασμα και την εγκατάσταση των προγραμμάτων, που είναι απαραίτητα για την δημιουργία του παιχνιδιού. Συγκεκριμένα τα προγράμματα που χρειάζονται για την υλοποίηση είναι το Unity[5],[6] και το Blender[7],[8]. Για να πραγματοποιηθεί οποιοδήποτε project στο Unity χρειάζεται ορισμένα βοηθητικά πακέτα, για να κάνουν ορισμένες λειτουργίες, όπως για παράδειγμα το Vuforia που χωρίς αυτό δεν μπορεί να υλοποιηθεί το AR κομμάτι του παιχνιδιού. Σε αυτό το κεφάλαιο υπάρχει αναλυτικά ο τρόπος εισαγωγής και ρύθμισης κάθε πακέτου που υπάρχει στο πρόγραμμα.

3.2 Unity

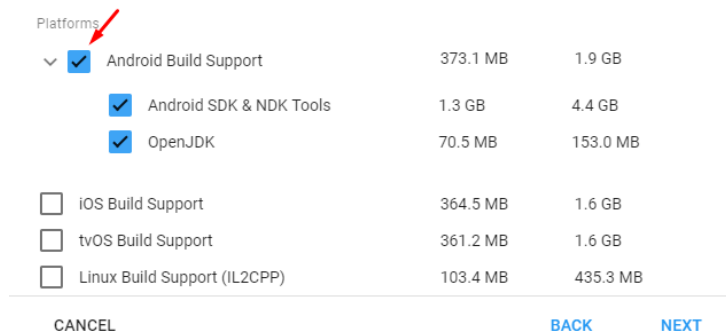
Η πλατφόρμα δημιουργίας παιχνιδιών Unity είναι το βασικό περιβάλλον για τη δημιουργία του προγράμματος. Είναι μία από τις μεγαλύτερες πλατφόρμες στον τομέα της, καθώς παρέχει το περιεχόμενό της δωρεάν στους προγραμματιστές, με συνεχείς ενημερώσεις και βελτιώσεις. Οι δυνατότητες του Unity είναι η δημιουργία 2D και 3D παιχνιδιών, όπως και VR ή AR. Στο πλαίσιο της παρούσας εργασίας χρησιμοποιήθηκε η δημιουργία 3D παιχνιδιού με δυνατότητες AR. Κατά κύριο λόγο ο κώδικας που χρησιμοποιεί το Unity είναι σε γλώσσα C#, ενώ πολλές λειτουργίες γίνονται στο UI του Unity, όπως η δημιουργία animation, σκηνών και UI. Άλλες δωρεάν πλατφόρμες είναι η Unreal Engine[9] και το Godot[10].

3.2.1 Εγκατάσταση Unity Hub και Unity

Αρχικά πρέπει να εγκατασταθεί το Unity Hub από το site του Unity [6], δημιουργώντας πρώτα έναν προσωπικό λογαριασμό και έπειτα πρέπει να επιλεγθεί η επιλογή Personal, διαβάζοντας τους όρους και τις προϋποθέσεις. Αφού εγκατασταθεί το Unity Hub version 2.4.3 ή μεγαλύτερη έκδοση, θα ανοίξει ένα παράθυρο για τη δημιουργία Project. Στη συνέχεια πρέπει να γίνει εγκατάσταση της έκδοσης του Unity όπως στο σχήμα 3.1. Το παρόν παιχνίδι είναι στην έκδοση 2020.2.1f1, αλλά μπορεί να προγραμματιστεί και σε επόμενες εκδόσεις με μικρές διαφορές, ανάλογα με τις μελλοντικές ενημερώσεις των προγραμματιστών. Απαραίτητο βήμα είναι η επιλογή του Android Build Support και των περιεχομένων του όπως στο σχήμα 3.2, χωρίς αυτό το βήμα η εφαρμογή δεν υποστηρίζεται για συσκευές android.



Σχήμα 3.1: Unity Hub, Install Version



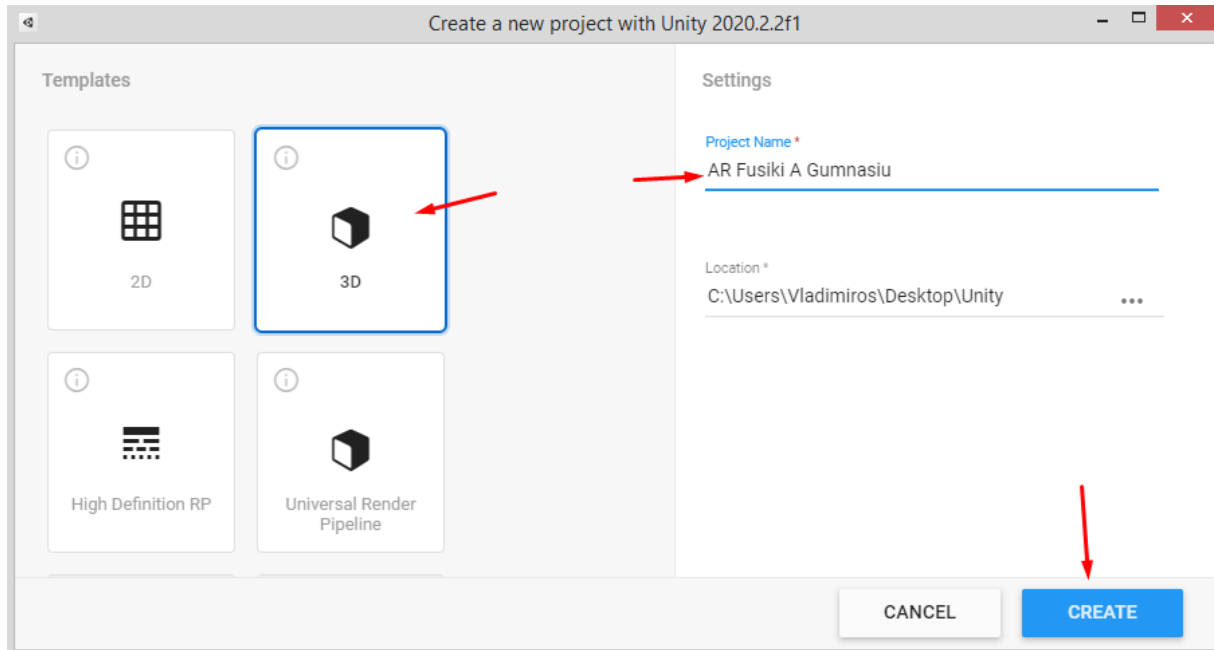
Σχήμα 3.2: Install Android Builder Support

3.2.2 Δημιουργία project

Αφού έχει εγκατασταθεί η κατάλληλη version του Unity, το επόμενο βήμα είναι η δημιουργία του project χρησιμοποιώντας τις παραπάνω επιλογές, επιλέγοντας την version 2020.2.1f1 στο βελάκι δίπλα στο κουμπί NEW. Το τελευταίο βήμα είναι η κατασκευή του project πατώντας το κουμπί NEW και έπειτα το CREATE όπως στο σχήμα 3.3

3.3 Blender

Το Blender είναι ένα δωρεάν πρόγραμμα δημιουργίας 3D αντικειμένων, που έχει ανοιχτό κώδικα (open source). Στην Π.Ε. χρησιμοποιήθηκε η έκδοση 2.80, για την δημιουργία όλων των 3D αντικειμένων και για το animation σε ένα αντικείμενο, λόγω πολυπλοκότητας. Όλα όσα θα αναφερθούν παρακάτω μπορούν να υλοποιηθούν και σε νεότερες εκδόσεις. Για την εγκατάσταση του Blender, χρειάζεται η επίσκεψη στο επίσημο site του προγράμματος [8].



Σχήμα 3.3: Create Project

3.4 Package Manager

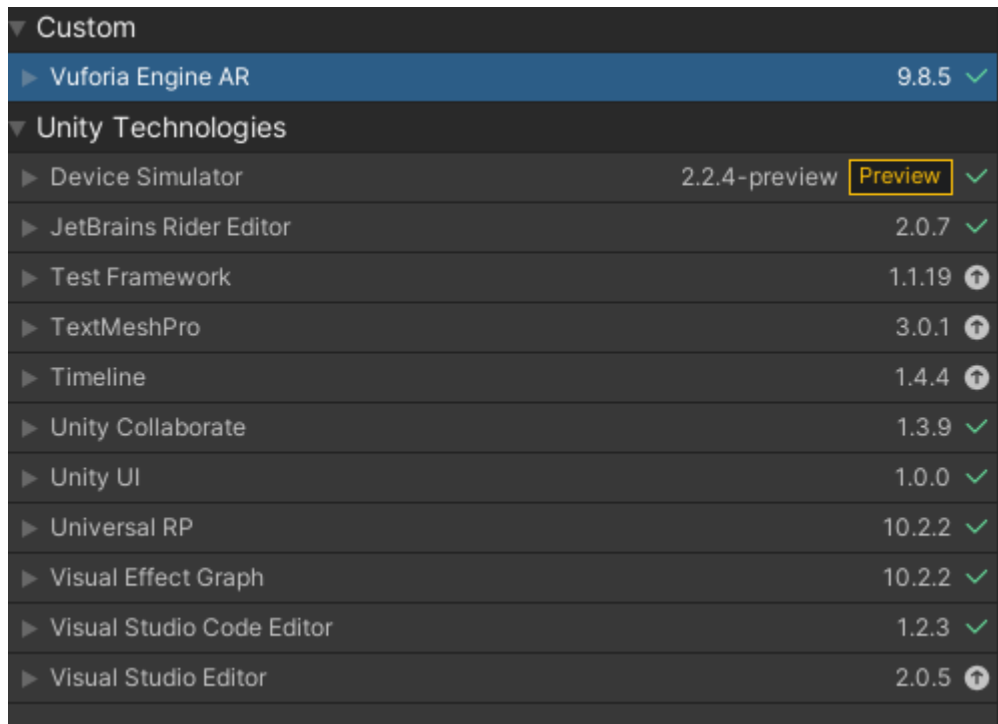
Με την εγκατάσταση του Unity και την έναρξη του project, δημιουργούνται ορισμένα προεπιλεγμένα πακέτα, βασικά για τον κορμό λειτουργίας του προγράμματος. Εκτός από τα προεπιλεγμένα, πρέπει να εγκατασταθούν κάποια χρήσιμα πακέτα με χειροκίνητο τρόπο. Για να μπουν στο project αυτά τα πακέτα πρέπει να ανοίξει το παράθυρο Package Manager από το μενού Window→Package Manager. Τα πακέτα που είναι εγκατεστημένα φαίνονται στο σχήμα 3.4.

3.4.1 Πακέτο Device Simulator

Το πακέτο Device Simulator είναι προαιρετικό για την υλοποίηση του project καθώς δεν επηρεάζει την λειτουργία του προγράμματος. Ο κύριος λόγος χρήσης του πακέτου είναι ο ευκολότερος τρόπος επίλυσης σφαλμάτων, γιατί στην προεπιλεγμένη προσομοίωση δεν υπάρχει η δυνατότητα Touch. Ένας άλλος τρόπος επίλυσης σφαλμάτων είναι η σύνδεση μιας φυσικής συσκευής με τον υπολογιστή. Λόγω του ότι το συγκεκριμένο πακέτο είναι σε πρώιμο στάδιο πρέπει να ενεργοποιηθεί μία ρύθμιση για να εμφανιστεί στη λίστα με τα πακέτα. Η ρύθμιση βρίσκεται στο Edit→Project Settings→Package Manager και πρέπει να ενεργοποιηθεί το κουτάκι Enable Preview Packages. Στο παράθυρο Package Manager στο αναδυόμενο μενού, Packages, πρέπει να επιλεγεί η επιλογή Unity Registry και στη μπάρα αναζήτησης πάνω δεξιά να πληκτρολογήσει το όνομα του πακέτου “Device Simulator”. Αφού επιλεγεί το πακέτο πρέπει να γίνει εγκατάσταση με το κουμπί Install στο κάτω δεξιά μέρος του παραθύρου.

3.4.2 Universal Render Pipeline

Το Universal Render Pipeline(URP) είναι ένας τρόπος rendering(απεικόνιση γραφικών), φτιαγμένο από την Unity, για την πιο ομαλή και βέλτιστη χρήση των γραφικών, καθώς ο χρήστης μπορεί να δημιουργεί αποκλειστικά γραφικά για κάθε ανάγκη που έχει το project του. Στο παιχνίδι της Π.Ε. χρησιμοποιήθηκε το URP για την κατασκευή ειδικών γραφικών, με σκοπό μια πιο ευχάριστη και



Σχήμα 3.4: Package Manager

ευανάγνωστη εικόνα. Η εγκατάσταση του πακέτου γίνεται στο παράθυρο Package Manager, στο αναδυόμενο μενού Packages και στην επιλογή Unity Registry. Στη μπάρα αναζήτησης, πάνω δεξιά, πρέπει να πληκτρολογηθεί το όνομα του πακέτου “Universal RP” και αφού επιλεγθεί πρέπει να εγκατασταθεί με το κουμπί Install. Στο κεφάλαιο 5 υπάρχουν περισσότερες λεπτομέρειες για το πακέτο.

3.4.3 Visual Effect Graph

Η χρήση γραφικών δεν είναι ο μόνος τρόπος για να βελτιωθεί η εικόνα ενός παιχνιδιού ή ενός project. Τα ειδικά εφέ (VFX) είναι ένα τρόπος να απογειωθεί η ποιότητα ενός εικονικού περιβάλλοντος, ζωντανεύοντας δραστικά τα αντικείμενα σε αυτόν τον χώρο, με μικρές λεπτομέρειες που δημιουργούνται και εξαφανίζονται, όπως ο καπνός, η φωτιά και άλλα. Τα λεγόμενα particles, είναι μικρές εικόνες ή σχήματα που εμφανίζονται και εξαφανίζονται για να προσφέρουν μια φυσικότητα στο περιβάλλον που εφαρμόζονται. Στην Π.Ε. η χρήση των particles, υπάρχει για να απεικονιστούν καλύτερα στον χρήστη που θα κάνει τα πειράματα, μερικές καταστάσεις όπως ο βρασμός και η πήξη του νερού. Πιο αναλυτικά τα ειδικά εφέ υπάρχουν στο κεφάλαιο 5. Η εγκατάσταση του πακέτου γίνεται στο παράθυρο Package Manager, στο αναδυόμενο μενού Packages και στην επιλογή Unity Registry. Στη μπάρα αναζήτησης πάνω δεξιά πρέπει να πληκτρολογηθεί το όνομα του πακέτου “Visual Effect Graph” και αφού επιλεγθεί πρέπει να εγκατασταθεί με το κουμπί Install.

3.5 Vuforia

Το πακέτο Vuforia είναι το κρισιμότερο εργαλείο για να λειτουργήσει το project, καθώς περιέχει την τεχνολογία για την χρήση του AR. Το Vuforia είναι η μεγαλύτερη πλατφόρμα προγραμματισμού και ανάπτυξης εφαρμογών, με υποστήριξη σε κινητές συσκευές, tablets και υπολογιστών, καθώς υποστηρίζεται από android, iOS και UWP. Έχει αρκετά μεγάλη ακρίβεια στον εντοπισμό των εικόνων στόχων (target) και μπορεί να τους ακολουθήσει με μεγάλη ευκολία, ακόμα και αν ο αριθμός των

στόχων είναι μεγάλος. Η δωρεάν έκδοση υποστηρίζει πως μπορεί να ακολουθήσει μέχρι 100 εικόνες στόχους, ενώ οι εκδόσεις επί πληρωμή δεν έχουν όριο στον αριθμό των targets. Πέρα από την απλή ανίχνευση εικόνας υπάρχουν και άλλες δυνατότητες, όπως η ανίχνευση αντικειμένων και η ανίχνευση κάποιας επίπεδης επιφάνειας, ένα τραπέζι ή πάτωμα. Οι δύο τελευταίες λειτουργίες δεν υλοποιούνται στην Π.Ε. Την στιγμή της συγγραφής το Vuforia βρίσκεται στην έκδοση 10.0 αλλά η εφαρμογή χρησιμοποιεί την έκδοση 9.8.5.

3.5.1 Τρόπος εγκατάστασης Vuforia: SDK

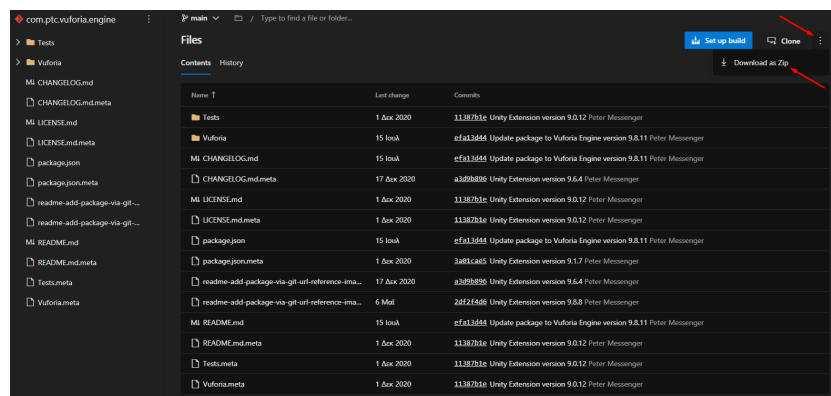
Ο πιο απλός τρόπος εγκατάστασης του Vuforia είναι κατεβάζοντας το SDK από το site του πακέτου[11]. Αφού κατέβει το αρχείο SDK για την πλατφόρμα του Unity, ανοίγοντας το εκτελέσιμο αρχείο θα δημιουργηθεί αυτόματα η πιο πρόσφατη έκδοση του Vuforia. Αν υπάρξει κάποια επιπλοκή με την εγκατάσταση του πακέτου υπάρχουν άλλοι δύο τρόποι παρακάτω.

3.5.2 Τρόπος εγκατάστασης Vuforia: Asset Store

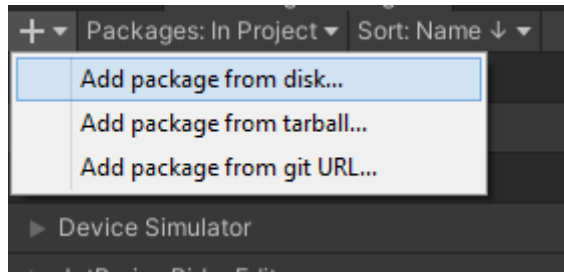
Ένας άλλος τρόπος εγκατάστασης του Vuforia είναι μέσω του Asset Store του Unity [12]. Το site χρειάζεται προσωπικό λογαριασμό στο Unity για να επιτραπεί η πρόσβαση στο πακέτο. Όταν το πακέτο συνδεθεί στον προσωπικό λογαριασμό του Unity, ο τρόπος για να γίνει εγκατάσταση του πακέτου είναι μέσω του παραθύρου Package Manager, στο αναδυόμενο μενού Packages→My Assets, εκεί υπάρχει το πακέτο Vuforia Engine. Το τελευταίο βήμα για την εγκατάσταση είναι το κουμπί Download στο κάτω δεξιά μέρος του παραθύρου, και αφού ολοκληρωθεί το κατέβασμα στο ίδιο σημείο του παραθύρου θα υπάρχει το κουμπί Import. Όταν πατηθεί το κουμπί Import θα ανοίξει ένα καινούριο παράθυρο με όλα τα αρχεία που είναι απαραίτητα, αν είναι επιλεγμένα όλα, θα πρέπει να πατηθεί το κουμπί Import που βρίσκεται στο καινούριο παράθυρο.

3.5.3 Τρόπος εγκατάστασης Vuforia: Χειροκίνητος τρόπος εγκατάστασης

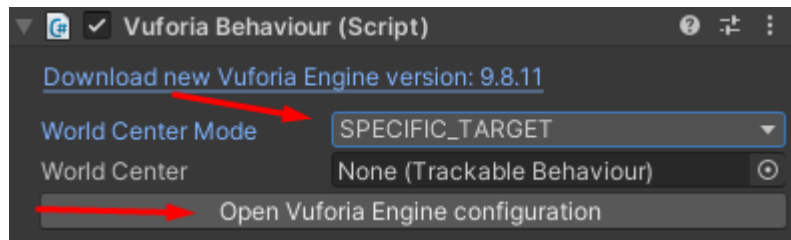
Αν οι παραπάνω τρόποι δεν είναι αρκετοί για την εγκατάσταση του πακέτου, υπάρχει ένας πιο περίπλοκος τρόπος, ο οποίος χρησιμοποιήθηκε στην παρούσα Π.Ε. Τα αρχεία του πακέτου υπάρχουν στο site [13]. Για το κατέβασμα των αρχείων τα βήματα είναι να επιλεγθούν οι 3 κουκίδες στο πάνω δεξιά μέρος της σελίδας και μετά το Download as Zip όπως στο σχήμα 3.5. Στη συνέχεια στο παράθυρο Package Manager πρέπει να ανοίξει το αναδυόμενο μενού με την εικόνα ενός σταυρού(+) και να επιλεγθεί η επιλογή “Add package from disk...” όπως στο σχήμα 3.6. Θα ανοίξει έπειτα ένα παράθυρο αναζήτησης φακέλων του υπολογιστή και μέσα στον φάκελο που κατέβηκε παραπάνω πρέπει να επιλεγθεί το αρχείο “package.json”.



Σχήμα 3.5: Vuforia Manual Download



Σχήμα 3.6: Add Package From Disk



Σχήμα 3.7: Vuforia Behaviour

License Manager

Get Development Key

Buy Deployment Key

Create a license key for your application.

Search

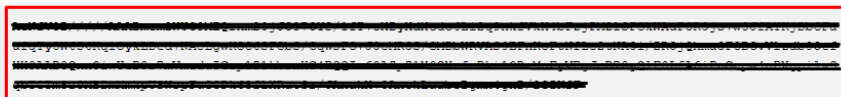
Name	Primary UUID [ⓘ]	Type	Status [▼]	Date Modified
AR test	N/A	Develop	Active	Mar 14, 2021

Σχήμα 3.8: Δημιουργία κλειδιού

AR test [Edit Name](#) [Delete License Key](#)

License Key Usage

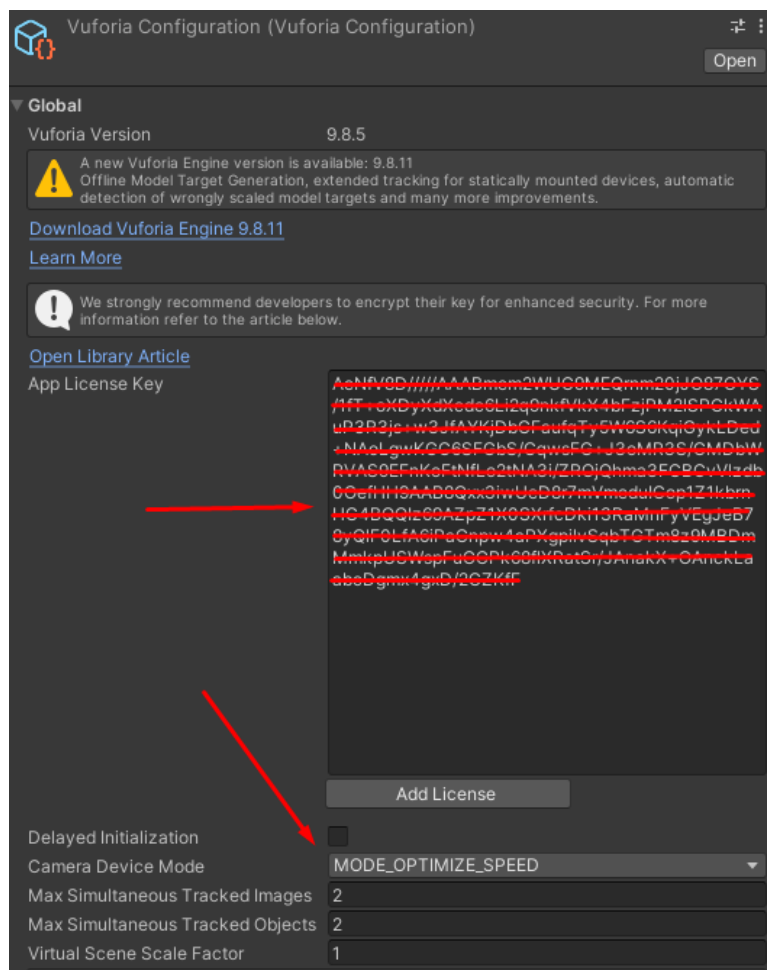
Please copy the license key below into your app



Σχήμα 3.9: Απόκτηση κλειδιού

3.5.4 Vuforia camera

Το AR χρειάζεται κατα κύριο λόγο την χρήση κάμερας για να λειτουργήσει. Στην ιεραρχία (Hierarchy) πρέπει να σβηστεί η κάμερα που έχει δημιουργηθεί αυτόματα και να φτιαχτεί μια AR κάμερα, κάνοντας δεξιά κλικ στον χώρο της ιεραρχίας→Vuforia Engine→AR Camera. Στη συνέχεια, όταν είναι επιλεγμένη η κάμερα, στο πεδίο Vuforia Behaviour πρέπει στο World Center Mode να επιλεγεί το SPECIFIC_TARGET, όπως στο σχήμα 3.7, και να πατηθεί το κουμπί Open Vuforia Engine configuration. Στο πεδίο Add License Key πρέπει να γίνει αντιγραφή από το site του Vuforia, το κλειδί για τα δικαιώματα. Ο τρόπος απόκτησης του κλειδιού γίνεται από το site, φτιάχνοντας αρχικά ένα προφίλ με το κουμπί Get Development Key(σχήμα 3.8) και στην συνέχεια το κλειδί υπάρχει μέσα στο προφίλ που δημιουργήθηκε όπως στο σχήμα 3.9. Στο σχήμα 3.10 παρουσιάζονται



Σχήμα 3.10: Vuforia Configurations

τα Configurations που πρέπει να γίνουν με σκοπό να αποκτήσει τα κατάλληλα δικαιώματα η εφαρμογή και πρέπει να θέσουμε το Camera Device Mode στην επιλογή MODE_OPTIMIZE_SPEED για πιο γρήγορη απόκριση της κάμερας, καθώς και τον αριθμό των μέγιστων εικόνων σε '2', γιατί τα περισσότερα πειράματα χρειάζονται 2 εικόνες-στόχους (image targets).

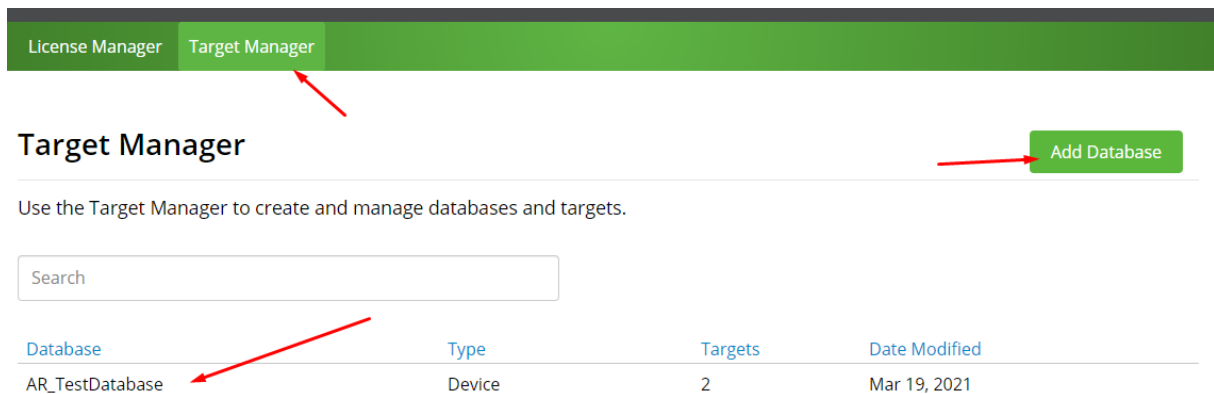
3.5.5 Δημιουργία QR codes

Ο ευκολότερος τρόπος να αναγνωρίσει και να δημιουργήσει κάτι το Vuforia, είναι με την χρήση εικόνων-στόχων (Image Targets). Μία καλή εικόνα-στόχος είναι τα QR codes, καθώς έχει πολλές

εναλλαγές χρωμάτων και είναι πιο εύκολο να εντοπιστούν με μεγαλύτερη ακρίβεια από το πρόγραμμα, λόγω του ότι έχει αρκετές λεπτομέρειες. Υπάρχουν αρκετά sites για την δημιουργία QR code στο διαδίκτυο, ένα από αυτά είναι το shawnlehner[14]. Πατώντας την επιλογή Vuforia και Download, δύο φορές, για δύο διαφορετικές εικόνες, έχουν δημιουργηθεί οι στόχοι και το μόνο που μένει είναι να κατασκευαστεί η βάση με τις εικόνες.

3.5.6 Δημιουργία Image Targets

Στο site του Vuforia, αφού έχει δημιουργηθεί το License key, πρέπει να οριστούν οι δύο εικόνες-στόχοι που αναφέρθηκαν παραπάνω. Στο σχήμα 3.11 φαίνεται ο τρόπος δημιουργίας μιας βάσης εικόνων. Πιο αναλυτικά πρέπει να εμφανιστούν αυτές οι επιλογές πατώντας το Target Manager πάνω αριστερά. Στη



Σχήμα 3.11: Target Manager

Add Target

Type:



File:

ar_marker.jpg
 jpg or .png (max file 2mb)

Width:

Enter the width of your target in scene units. The size of the target should be on the same scale as your augmented virtual content. Vuforia uses meters as the default unit scale. The target's height will be calculated when you upload your image.

Name:

Name must be unique to a database. When a target is detected in your application, this will be reported in the API.

Σχήμα 3.12: Add Target

συνέχεια η δημιουργία της βάσης γίνεται με το κουμπί Add Database και αφού δημιουργηθεί, πρέπει να πατηθεί η βάση που δημιουργήθηκε, με το όνομα που της δόθηκε. Εκεί πρέπει να προστεθούν οι δύο εικόνες με το κουμπί Add Target. Στην Π.Ε. το ένα QR code πρέπει να είναι διπλάσιο σε μέγεθος από το άλλο λόγω της δομής της εφαρμογής. Ο τρόπος για να γίνει αυτό είναι, κατά την εισαγωγή των στόχων (targets), το ένα από τα δυο QR code να έχει στο πλαίσιο Width τον αριθμό 2, με όνομα QR_big, και το δεύτερο QR code να έχει στο πλαίσιο Width τον αριθμό 1, με όνομα QRcode_small. Το Type και των δύο targets θα πρέπει να είναι απαραίτητα Single Image(σχήμα 3.12).

Download Database

2 of 2 active targets will be downloaded

Name:

AR_TestDatabase

Select a development platform:

Android Studio, Xcode or Visual Studio

Unity Editor

Cancel

Download

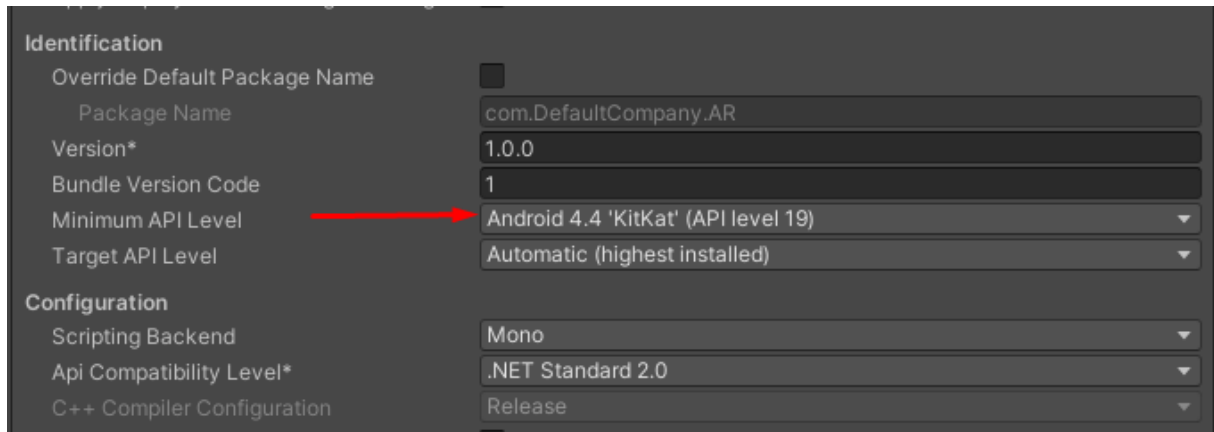
Σχήμα 3.13: Download Database

3.5.7 Βάση εικόνων στο project

Το τελευταίο βήμα για να μπορέσει το project στο Unity να χρησιμοποιήσει το Vuforia, είναι να περαστεί η βάση στην εφαρμογή. Με το κουμπί Download Database (All) κατεβαίνει ολόκληρη η βάση στον υπολογιστή. Σημαντικό είναι να επιλεγθεί το Unity Editor στο Select a development platform(σχήμα 3.13), για να μπορέσει το Unity να χρησιμοποιήσει σωστά την βάση και να κατέβει σαν πακέτο του Unity. Αφού κατέβει το πακέτο που δημιουργήθηκε, το μόνο που απομένει είναι να γίνει Import, ανοίγοντάς το την ώρα που είναι ανοιχτό και το project.

3.6 Ρυθμίσεις εφαρμογής

Για να δημιουργηθεί σωστά το project πρέπει να αλλάξουν μερικές ρυθμίσεις από τις βασικές επιλογές (Settings) του project.



Σχήμα 3.14: Project Settings, Android Version

3.6.1 Project Settings

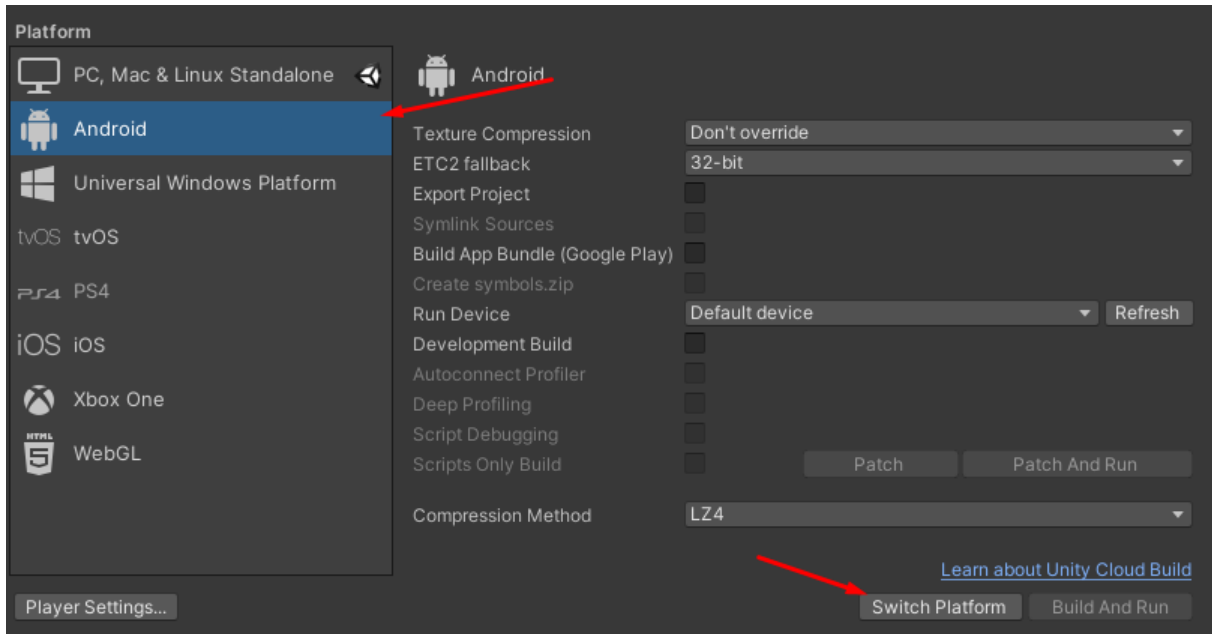
Οι ρυθμίσεις που πρέπει να αλλάξουν βρίσκονται στο Edit→Project Settings→Player. Σε σχέση με τις παλαιότερες εκδόσεις του Unity πλέον δεν χρειάζεται η αλλαγή του company name και του package name, αυτή η επιλογή γίνεται αυτόματα από την έκδοση του Unity 2020.1 και μετά. Η μόνη επιλογή που πρέπει να αλλαχτεί για τώρα είναι η ελάχιστη έκδοση android που είναι (σχήμα 3.14) η Android 4.4 'KitKat' (API level 19), η ελάχιστη έκδοση που υποστηρίζει το Vuforia

3.6.2 Build Settings

Η πιο βασική ρύθμιση που πρέπει να αλλαχτεί βρίσκεται στο File→Build Settings. Αν η εγκατάσταση της έκδοσης του Unity έγινε σωστά, τότε η επιλογή Android θα εμφανίζεται κανονικά, χωρίς την επιλογή Install. Σε αυτή τη περίπτωση θα πρέπει να γίνει εγκατάσταση της πλατφόρμας Android απο το Unity Hub. Αν έχει εγκατασταθεί σωστά, το μόνο που πρέπει να γίνει είναι να πατηθεί το κουμπί Switch Platform όπως στο σχήμα 3.15

3.7 Επίλογος

Με τα παραπάνω βήματα το project είναι έτοιμο να λειτουργήσει. Υπάρχουν μερικές ρυθμίσεις που δεν μπορούν να ολοκληρωθούν σε αυτό το στάδιο που βρίσκεται η εφαρμογή, αλλά ο τρόπος υλοποίησης υπάρχει στα επόμενα κεφάλαια.



Σχήμα 3.15: Switch Platform to Android

Κεφάλαιο 4ο: Αρχικό μενού

4.1 Εισαγωγή

Κάθε εφαρμογή ή παιχνίδι χρειάζεται ένα αρχικό μενού, με σκοπό να καθοδηγεί τους χρήστες στις επιλογές που παίρνουν, για να χειριστούν την εφαρμογή. Ένα απλό και εύκολα κατανοητό μενού, μπορεί να βοηθήσει τους χρήστες να μάθουν γρήγορα την εφαρμογή και τον τρόπο πλοήγησής της. Σε αυτό το κεφάλαιο, υπάρχουν αναλυτικά όλα τα βήματα δημιουργίας του αρχικού μενού του παιχνιδιού, καθώς και τα scripts που εφαρμόζουν την λογική στα κουμπιά και στο UI.

4.2 Οργάνωση φακέλων Scripts

Για λόγους ευκολίας και καλύτερης οργάνωσης, έχουν δημιουργηθεί φάκελοι μέσα σε έναν καινούριο υποφάκελο, στο Assets, με όνομα Scripts. Ο τρόπος δημιουργίας του φακέλου γίνεται με δεξί κλικ→Create→Folder. Όπως φαίνεται στο σχήμα 4.1, μέσα στον φάκελο Scripts, υπάρχουν εννιά υποφάκελοι και τέσσερα scripts. Τα τέσσερα scripts αναλύονται στο κεφάλαιο 6, γιατί είναι κοινές λειτουργίες που υπάρχουν στις υπόλοιπες σκηνές, ενώ στους υποφακέλους P1 μέχρι P12 υπάρχουν scripts για τα 12 κεφάλαια του βιβλίου της φυσικής. Στον υποφάκελο UI scripts εκτός από το script MainMenuController που αναφέρεται παρακάτω, άλλο ένα script έχει δημιουργηθεί με σκοπό τον έλεγχο του UI στις σκηνές που υπάρχουν τα πειράματα. Το τελευταίο αναφέρεται στο επόμενο κεφάλαιο, “Κοινές λειτουργίες σκηνών”.

4.3 Script: MainMenuController

Το αρχικό μενού χρειάζεται μόνο ένα script, που ελέγχει μερικές βασικές παραμέτρους και τα κουμπιά που υπάρχουν στο panel. Αφού κατασκευαστεί το script, το πρώτο πράγμα που πρέπει να γίνει είναι πάνω από την κλάση να προστεθεί η γραμμή “using UnityEngine.SceneManagement;”. Ο λόγος είναι για να μπορεί να γίνει έλεγχος και αλλαγή της σκηνής. Οι μέθοδοι που υπάρχουν στο script είναι (σχήμα 4.2):

- void Start(): Κάνει την διάταξη της εφαρμογής κατακόρυφη και το currentPanel το αρχικό panel.
- void Update(): Όταν πατηθεί το κουμπί “πίσω” στη συσκευή, θα καλέσει το ChangePanel και θα πάει στο αρχικό panel.
- void CloseApplication(): Κλείνει την εφαρμογή.
- void ChangePanel(GameObject nextPanel): Κλείνει το panel που είναι ανοιχτό και ανοίγει το nextPanel, κάνοντάς τό το currentPanel.
- void ChangeScene(int SceneNum): Αλλάζει την σκηνή.



Σχήμα 4.1: Φάκελος Scripts

```

public class MainMenuController : MonoBehaviour
{
    [SerializeField] GameObject mainMenuPanel;

    GameObject currentPanel;

    void Start()
    {
        //sets the screen to be portret in main menu only
        Screen.orientation = ScreenOrientation.Portrait;
        currentPanel = mainMenuPanel;
    }

    void Update()
    {
        //if phone user tuch the back button
        if(Input.GetKeyDown(KeyCode.Escape) && !currentPanel.Equals(mainMenuPanel))
        {
            ChangePanel(mainMenuPanel);
        }
    }

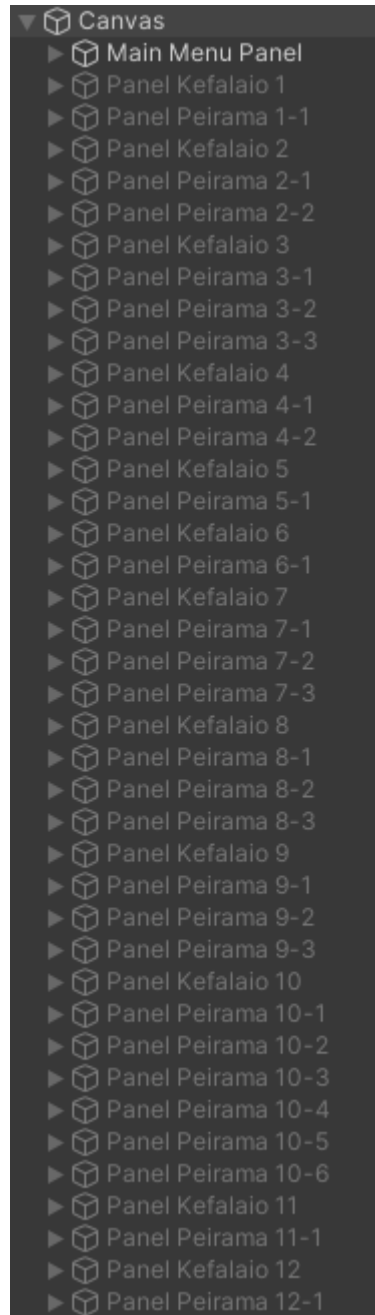
    public void CloseApplication()
    {
        Application.Quit();
    }

    public void ChangePanel(GameObject nextPanel)
    {
        currentPanel.SetActive(false);
        nextPanel.SetActive(true);
        currentPanel = nextPanel;
    }

    public void ChangeScene(int sceneNum)
    {
        SceneManager.LoadScene(sceneNum, LoadSceneMode.Single);
    }
}

```

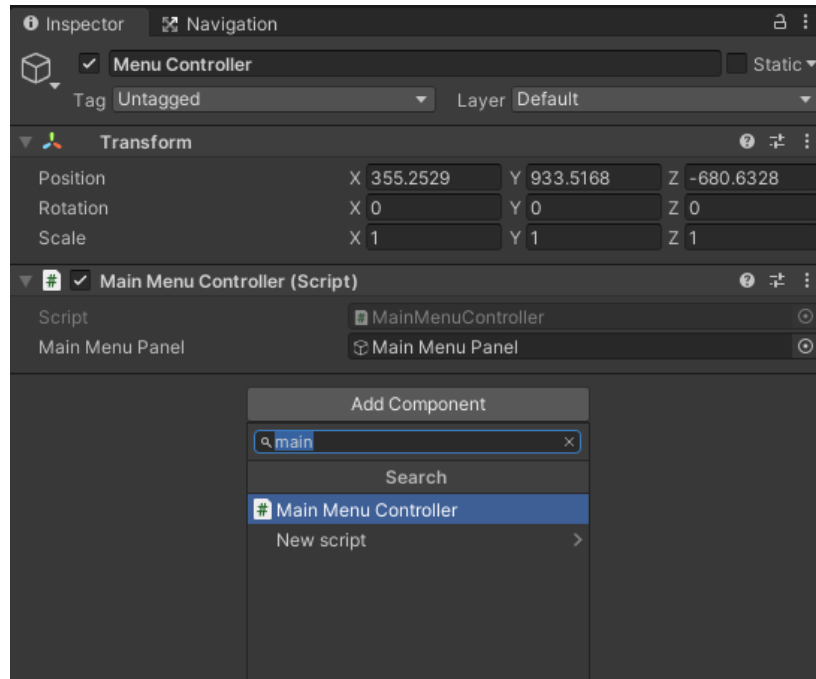
Σχήμα 4.2: Script MainMenuController



Σχήμα 4.3: Ιεραρχία Καμβά

4.4 Canvas και Menu Controller

Για να υπάρχει UI στο παιχνίδι, πρέπει να φτιαχτεί ένας καμβάς που θα περιέχει όλα τα στοιχεία, όπως panels, κουμπιά και άλλα. Στο παράθυρο ιεραρχία πρέπει να γίνει δεξί κλικ→UI→Canvas→Panel. Μαζί με το panel θα δημιουργηθεί και ο καμβάς σαν αντικείμενο γονιός. Πρέπει να φτιαχτούν τα panels όπως στο σχήμα 4.3. Το Main Menu Panel είναι το αρχικό panel που θα φαίνεται όταν ανοίξει το παιχνίδι, ενώ τα υπόλοιπα δεν θα είναι enabled, για να γίνει αυτό, το κουτάκι πάνω αριστερά στο Inspector δεν πρέπει να είναι επιλεγμένο. Αφού δημιουργήθηκε το script, πρέπει να φτιαχτεί ένα GameObject στη σκηνή, που έχει ως συστατικό το MainMenuController. Στην ιεραρχία, με δεξί κλικ και Create Empty δημιουργείται ένα κενό αντικείμενο, στο οποίο πρέπει να συρθεί το script στο Inspector ή με το κουμπί Add Component, όπως στο σχήμα 4.4. Τέλος, στο πεδίο Main

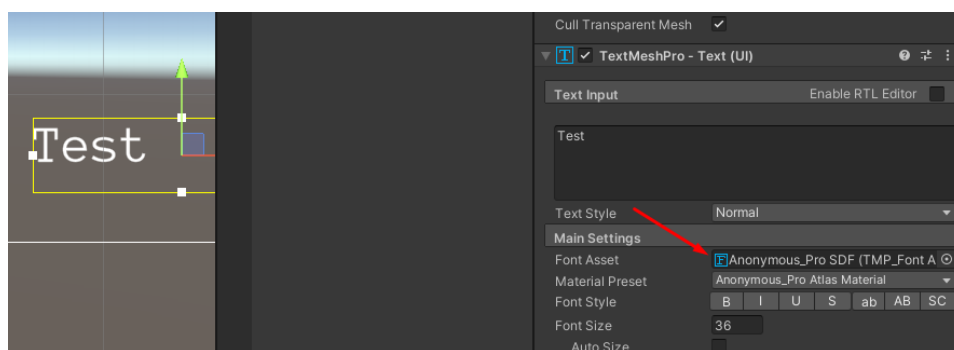


Σχήμα 4.4: Add Component

Menu Panel, πρέπει να μπει το αντικείμενο Main Menu Panel που δημιουργήθηκε παραπάνω, σέρνοντάς το ή πατώντας στην βούλα δεξιά του πλαισίου.

4.5 Γραμματοσειρά

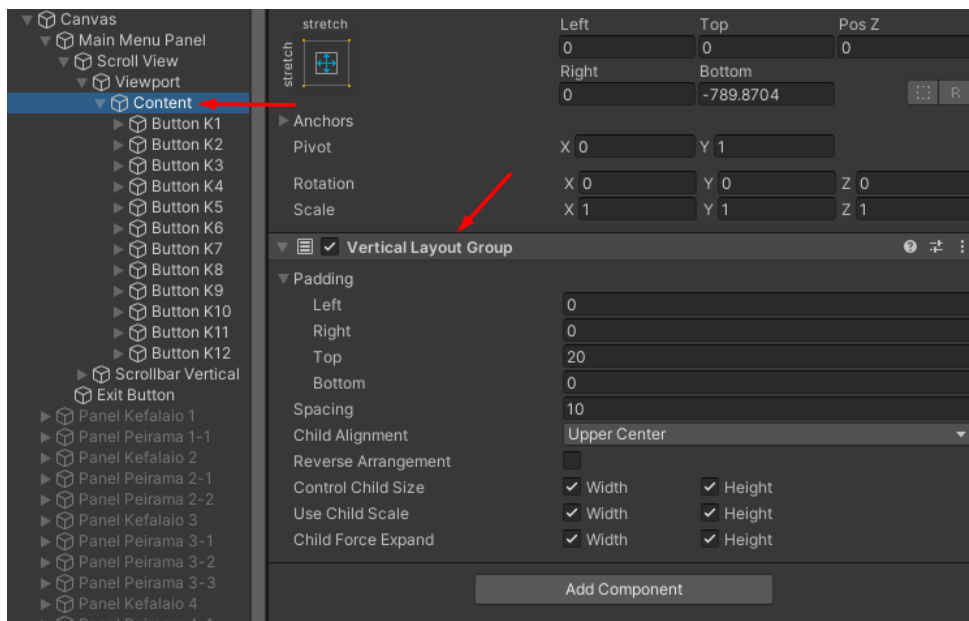
Η γραμματοσειρά που χρησιμοποιήθηκε στην Π.Ε. είναι η “Anonymous-Pro”, διότι το Unity δεν διαθέτει γραμματοσειρές με ελληνικούς χαρακτήρες, σε αντιθεση με την προαναφερθείσα, η οποία μπορεί να κατέβει από το site “Google Fonts” [15]. Αν δεν υπάρχει ο φάκελος TextMesh Pro, πρέπει πρώτα να δημιουργηθεί ένα game object, Text - TextMeshPro, κάνοντας δεξί κλικ στο Hierarchy→UI→Text - TextMeshPro. Εκεί θα εμφανιστεί ένα παράθυρο για να γίνει Import το πακέτο TextMesh Pro, μαζί με αυτό θα δημιουργηθεί ο φάκελος. Μεταφέροντας το αρχείο ttf που κατέβηκε και αφήνοντάς το στον φάκελο TextMesh Pro, μπορεί το Unity να δημιουργήσει ένα asset για να χρησιμοποιηθεί η γραμματοσειρά. Με δεξί κλικ στο αρχείο→Create→TextMeshPro→Font Asset, εμφανίζεται ένα asset το οποίο μπορεί να χρησιμοποιηθεί από οποιοδήποτε TextMesh pro game object, όπως στο σχήμα 4.5.



Σχήμα 4.5: Font Asset

4.6 Scroll View

Το Scroll View είναι ένα Game Object που επιτρέπει την εμφάνιση πολλαπλών αντικειμένων που δεν χωράνε στην οθόνη του κινητού ή του tablet. Σε διαφορετική περίπτωση, τα αντικείμενα θα ξεχειλίζουν και θα έβγαιναν από την οθόνη, αλλά με το Scroll View μπορούν να εμφανιστούν όσα αντικείμενα χρειάζεται το UI κάνοντας scroll up ή scroll down για να εμφανιστεί το υπόλοιπο περιεχόμενο. Για τη δημιουργία του, μέσα στα panels που αναφέρθηκαν παραπάνω στην ιεραρχία, πρέπει να γίνει δεξί κλικ→UI→Scroll View. Το project δεν χρειάζεται την οριζόντια μπάρα που δημιουργήθηκε ως παιδί του scroll view, οπότε με δεξί κλικ στο “Scrollbar Horizontal” και Delete, πρέπει να γίνει διαγραφή του Game Object. Στη συνέχεια, πρέπει να επιλεγθεί το δεύτερο παιδί, μέσα στο Viewport, το Content, όπως στο σχήμα 4.6. και με το κουμπί Add Component να προστεθεί το “Vertical Layout Group”, το οποίο επιτρέπει τα στοιχεία του Content να στοιχηθούν το ένα κάτω από το άλλο. Στο συστατικό που δημιουργήθηκε πρέπει το Top Padding να γίνει 20, το Child Alignment να επιλεγθεί ως Upper Center και όλα τα check boxes με όνομα Width και Height να είναι επιλεγμένα.



Σχήμα 4.6: Ρυθμίσεις Content

4.7 Main Menu Buttons

Η πλοήγηση του χρήστη στα κεφάλαια του βιβλίου γίνεται από το αρχικό μενού, με τη χρήση κουμπιών που υπάρχουν στο Content που αναφέρθηκε παραπάνω. Για τη δημιουργία κάποιου κουμπιού, πρέπει να γίνει δεξί κλικ στο Content→UI→Button-TextMeshPro. Αντιγράφοντας το κουμπί που δημιουργήθηκε, στοιχίζονται αυτόματα κάτω από το προηγούμενο, λόγω του Vertical Layout Group που αναφέρθηκε παραπάνω. Αφού δημιουργήθηκε το κουμπί, δεν εκτελεί καμία εντολή, γιατί δεν εκτελεί τίποτα στην OnClick() μέθοδο του. Επιλέγοντας το κουμπί, στο Inspector, στο συστατικό Button, πρέπει να πατηθεί το σύμβολο “+” στο κάτω μέρος του On Click (). Εκεί μπορεί να επιλεγθούν μία ή περισσότερες μέθοδοι, που θα εκτελεστούν όταν πατηθεί το κουμπί. Σε αυτή τη περίπτωση, πρέπει να μπει στο κάτω αριστερά πεδίο το game object: Menu Controller, στο πάνω δεξιά η μέθοδος ChangePanel από το script MainMenuController και τέλος σέρνοντας, στο κάτω δεξιά πλαίσιο, το αντίστοιχο panel όπως στο σχήμα 4.7. Η αντιστοίχιση γίνεται με τον εξής τρόπο:

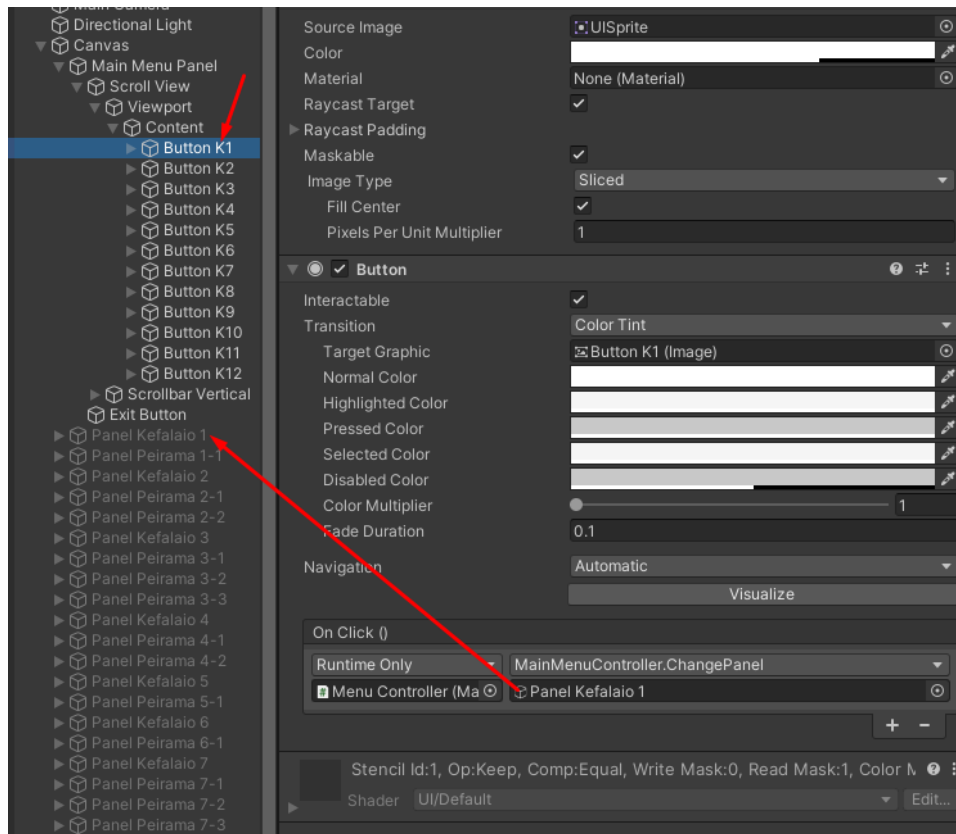
Κεφάλαιο 4

Κεφάλαιο 1→Panel Kefalaio 1

Κεφάλαιο 2→Panel Kefalaio 2

....

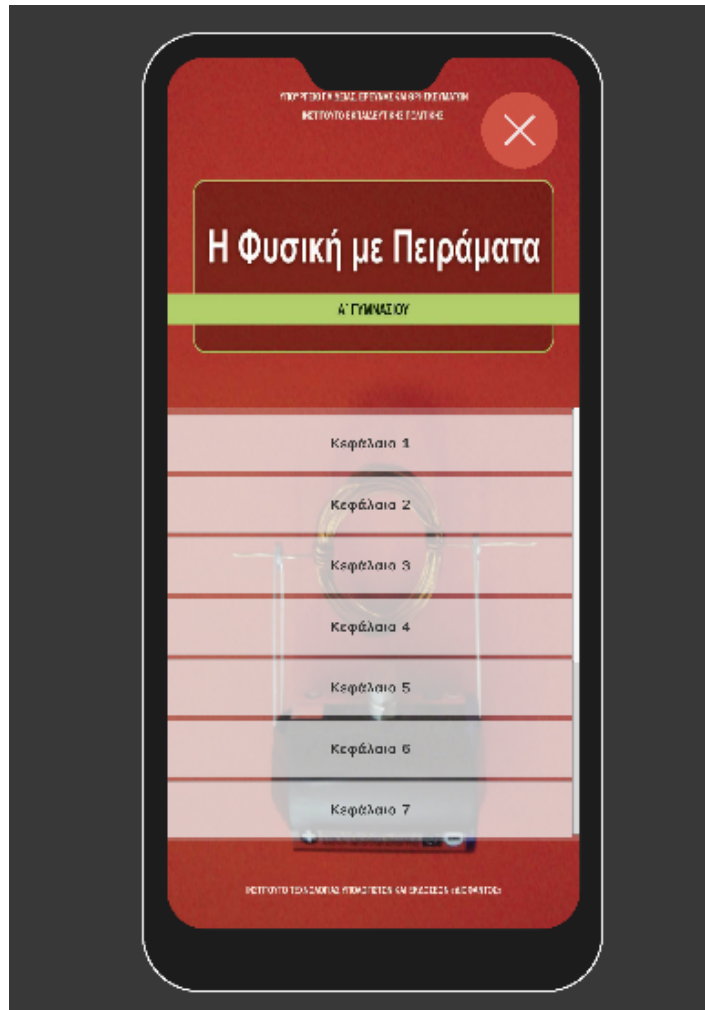
Κεφάλαιο 12→Panel Kefalaio 12



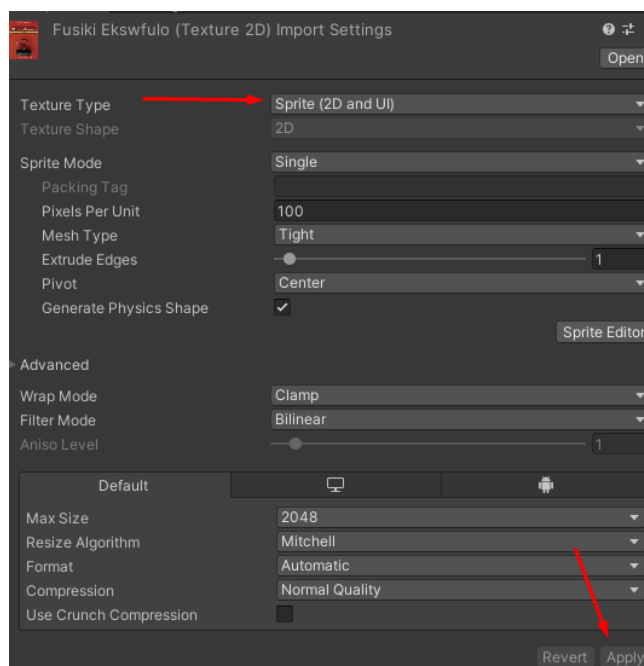
Σχήμα 4.7: Κουμπιά του Main Menu Panel

4.8 Φόντο, τρόπος εισαγωγής εικόνων και κουμπί εξόδου

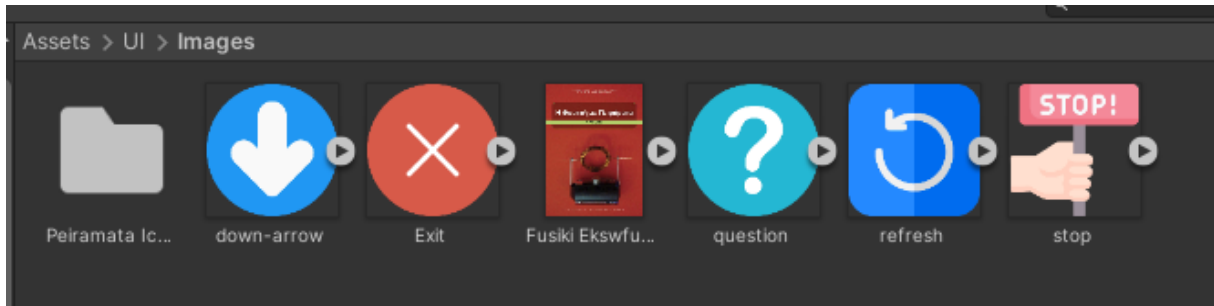
Όπως φαίνεται στην εικόνα 4.8, πίσω από τα κουμπιά υπάρχει το φόντο με το εξώφυλλο του σχολικού βιβλίου της φυσικής Α' γυμνασίου [1]. Ο τρόπος εισαγωγής της εικόνας του φόντου, όπως και κάθε άλλης εικόνας, γίνεται αρχικά σέρνοντάς την μέσα σε κάποιο φάκελο του project, στη συνέχεια με επιλεγμένη την εικόνα, στο Inspector πρέπει η επιλογή Texture Type να γίνει “Sprite (2D and UI)” και έπειτα το κουμπί Apply όπως στο σχήμα 4.9. Στην Π.Ε. χρησιμοποιήθηκαν μερικές εικόνες για UI από το site “Flat Icons”[16] και εικόνες για να παρουσιαστούν τα παραδείγματα των πειραμάτων (σχήμα 4.10 και 4.11). Για να χρησιμοποιηθεί οποιαδήποτε εικόνα στο project πρέπει να δημιουργηθεί ένα game object στον καμβά με δεξί κλικ→UI→Image και στο Source Image να επιλεγθεί η κατάλληλη εικόνα. Τέλος το κουμπί της εξόδου δημιουργείται όπως τα υπόλοιπα κουμπιά παραπάνω, με την διαφορά ότι στο πάνω δεξιά πλαίσιο του On Click (), πρέπει να γίνει επιλογή του CloseApplication του script MainMenuController.



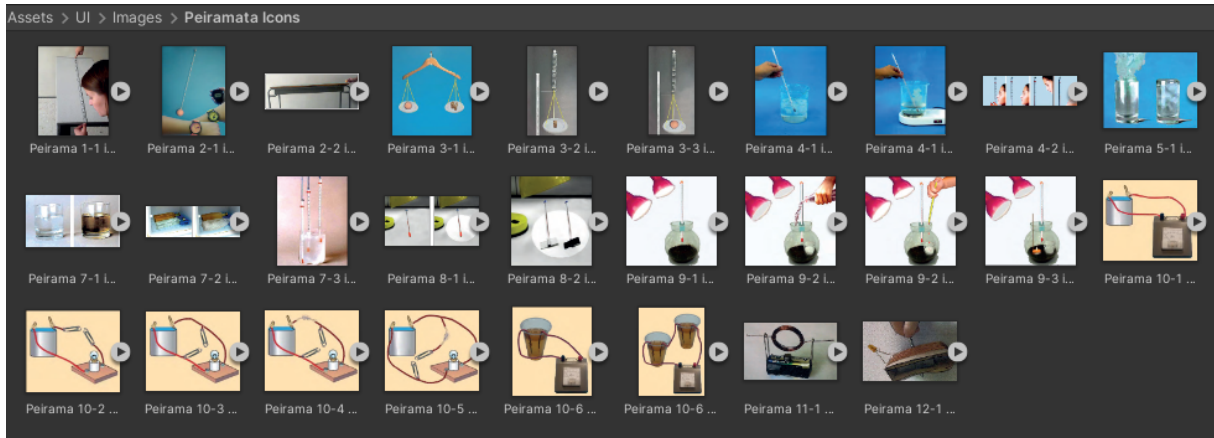
Σχήμα 4.8: Αρχική Οθόνη Εφαρμογής



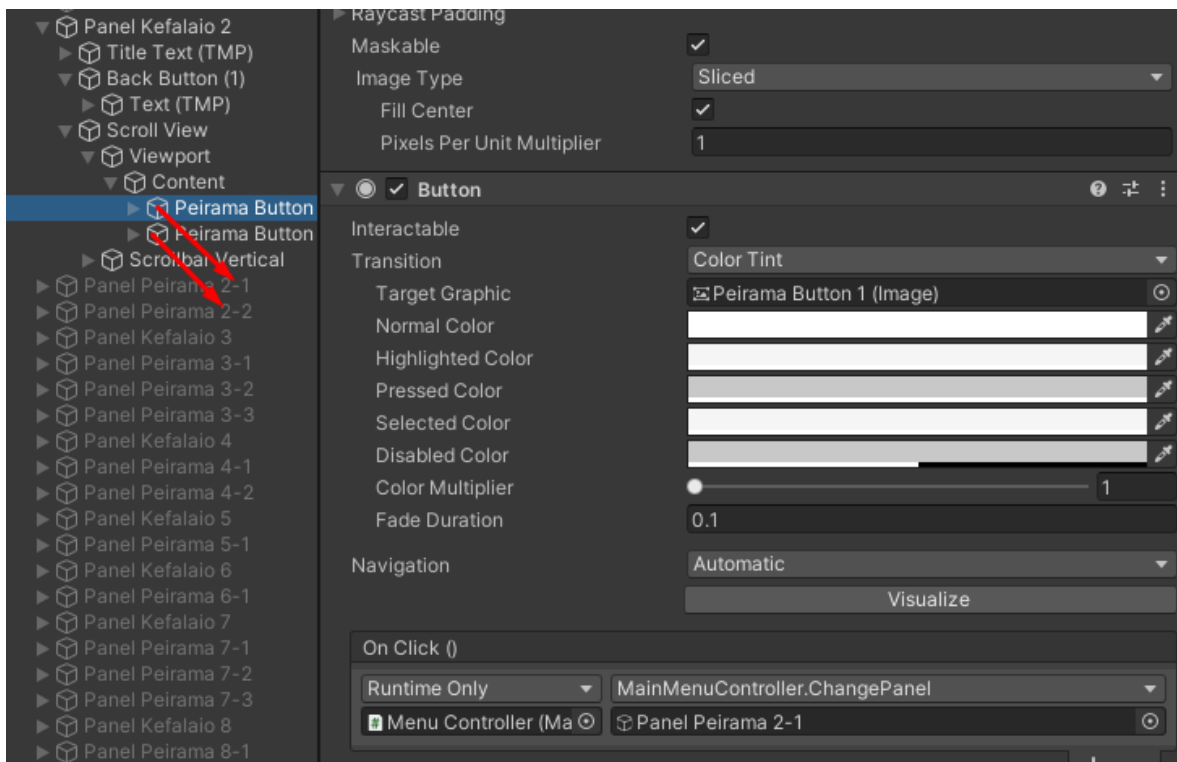
Σχήμα 4.9: Ρυθμίσεις εικόνας



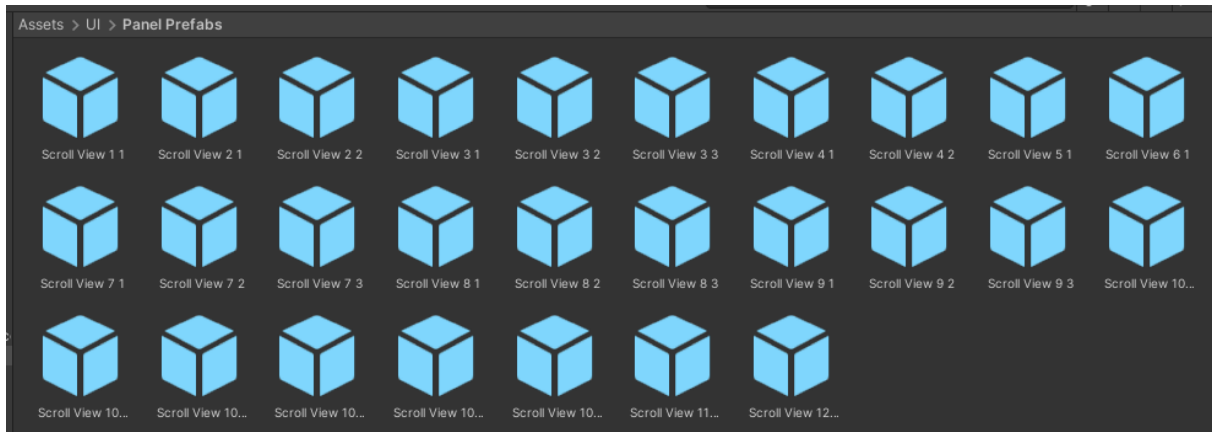
Σχήμα 4.10: UI Icons



Σχήμα 4.11: Εικόνες πειραμάτων



Σχήμα 4.12: Panel Κεφαλαίων



Σχήμα 4.13: Scroll Views όλων των πειραμάτων.

4.9 Panel Κεφαλαίων

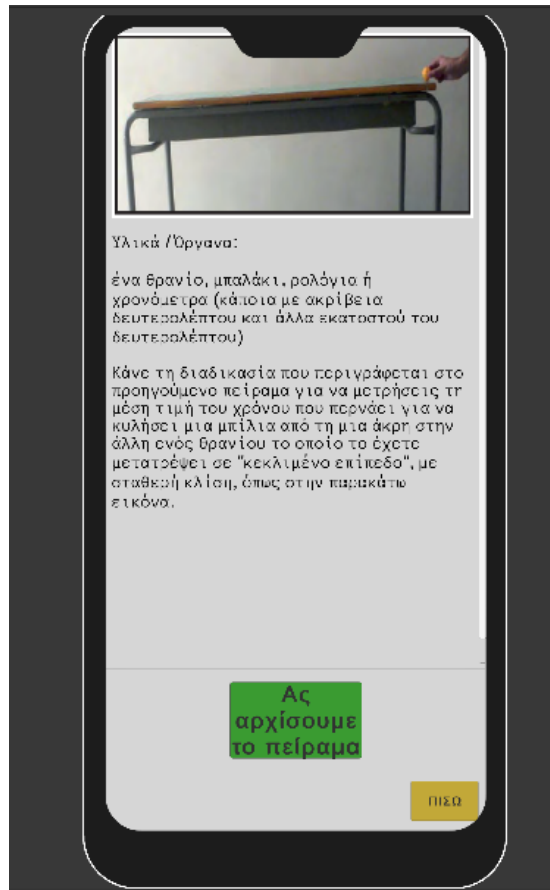
Αφού πατηθεί το κουμπί κάποιου κεφαλαίου, του αρχικού μενού, θα κλείσει το αρχικό μενού και θα ανοίξει το αντίστοιχο panel του κεφαλαίου που ζήτησε ο χρήστης. Η δομή των panel των κεφαλαίων είναι παρόμοια με την δομή του αρχικού μενού, δηλαδή έχουν ένα scroll view και κουμπιά μέσα σε αυτό, ως παιδιά του Content. Το κάθε κουμπί μεταφέρει τον χρήστη στο πείραμα που επέλεξε με την ίδια τεχνική που χρησιμοποιήθηκε προηγουμένως, με την μέθοδο ChangePanel στο On Click (). Όπως φαίνεται στο σχήμα 4.12 το πρώτο κουμπί του κεφαλαίου 2 αντιστοιχεί στο Panel Peirama 2-1, ενώ το δεύτερο κουμπί αντιστοιχεί στο Panel Peirama 2-2. Με την ίδια λογική συσχετίζονται και τα υπόλοιπα κουμπιά των άλλων κεφαλαίων με τα panels των πειραμάτων. Επιπλέον, όλα τα panels εκτός από το αρχικό μενού, έχουν το κουμπί “ΠΙΣΩ” το οποίο έχει την μέθοδο ChangePanel στο On Click () με το σωστό game object για μεταβλητή, δηλαδή τα panels των κεφαλαίων έχουν το αρχικό μενού και τα panels των πειραμάτων έχουν τα panels των κεφαλαίων τους.

4.10 Panel Πειράματος

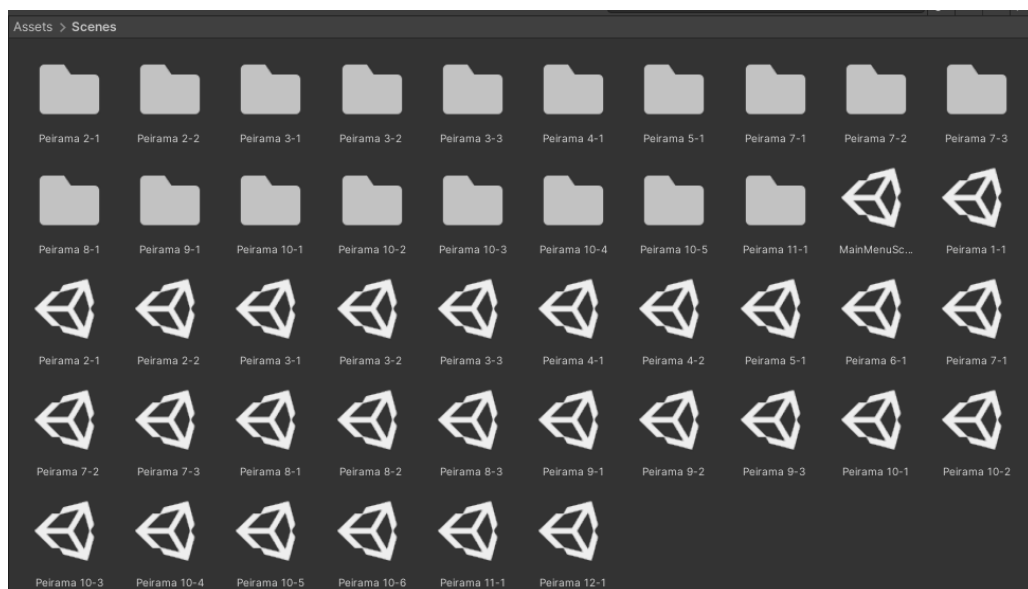
Τα Panels των πειραμάτων διαθέτουν 3 βασικά στοιχεία (σχήμα 4.14), ένα από αυτά είναι το κουμπί πίσω που αναφέρθηκε προηγουμένως. Τα υπόλοιπα δύο παρουσιάζονται παρακάτω.

4.10.1 Scroll View Πειραμάτων

Το κάθε panel των πειραμάτων έχει ένα scroll view, που περιέχει πληροφορίες για το συγκεκριμένο πείραμα και μία ή δύο εικόνες, του σχολικού βιβλίου της φυσικής[17], που δείχνουν πως μοιάζει το πείραμα σε πραγματικό περιβάλλον. Οι εικόνες και το κείμενο βρίσκονται μέσα στο Content του scroll view, στο οποίο οι μεταβλητές του padding είναι 20 στο left, right και top, ενώ το spacing είναι 40. Όταν ολοκληρωθεί το scroll view, σέρνοντάς το σε κάποιον φάκελο του project, αποθηκεύεται το αντίγραφο(prefab) του αντικειμένου. Το αντίγραφο είναι ένα επαναχρησιμοποιούμενο αντικείμενο, που αποθηκεύεται σαν αρχείο στο project, κάνοντας την αντιγραφή του από σκηνή σε σκηνή πολύ εύκολη. Στην Π.Ε. αποθηκεύτηκαν prefabs από όλα scroll view των πειραμάτων όπως φαίνεται στο σχήμα 4.13. Η χρήση των prefabs παρουσιάζεται στο επόμενο κεφάλαιο.



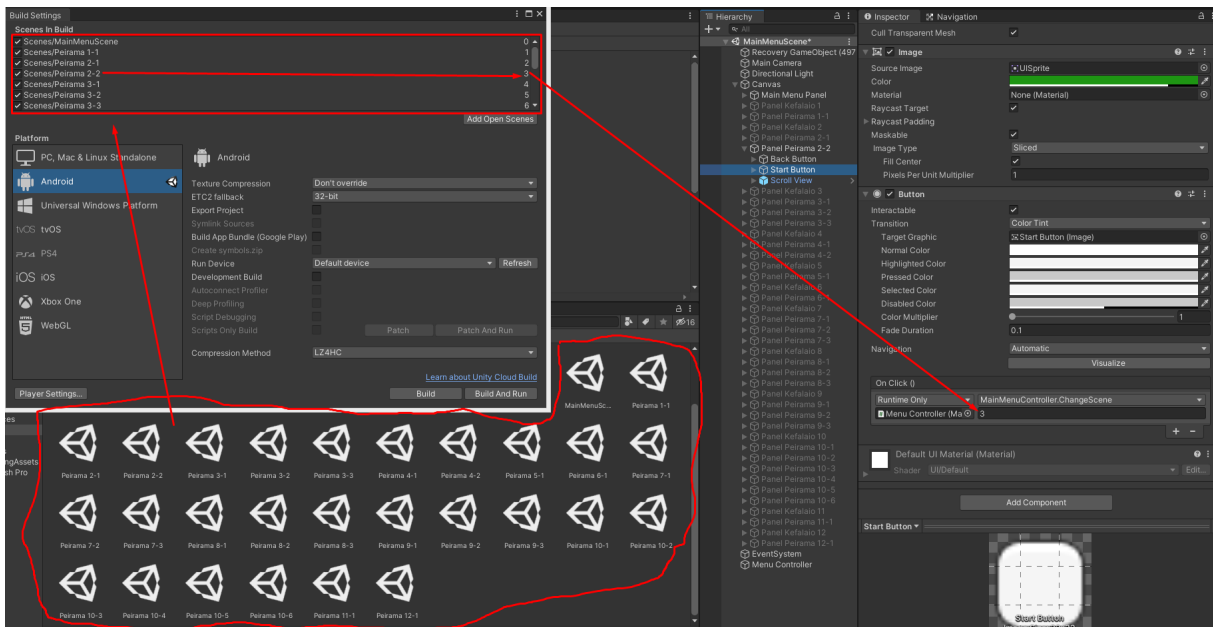
Σχήμα 4.14: Scroll View Πειράματος 2-2



Σχήμα 4.15: Φάκελος Σκηνών

4.10.2 Έναρξη πειραμάτων και κατασκευή σκηνών

Το κάθε πείραμα διαδραματίζεται σε διαφορετική σκηνή του project για την καλύτερη οργάνωση αυτού. Μέσα σε έναν καινούριο φάκελο με όνομα Scenes δημιουργούνται κενά scenes για κάθε πείραμα, με τα αντίστοιχα ονόματα όπως φαίνονται στο σχήμα 4.15, κάνοντας δεξί κλικ→Create→Scene. Όταν δημιουργηθούν όλες οι σκηνές για τα πειράματα, πρέπει να αριθμηθούν στα Build Settings (βλέπε 3.2.2 Build Settings). Επιλέγοντας όλες τις σκηνές, όπως φαίνεται στο σχήμα 4.16, και σέρνοντάς τες στο “Scenes In Build”, αριθμούνται από 0 μέχρι το 27. Μερικές από τις σκηνές δεν χρησιμοποιούνται στην εφαρμογή. Η σκηνή με αριθμό 0 είναι το αρχικό μενού και οι σκηνές από το 1 μέχρι το 27 είναι τα πειράματα. Όπως φαίνεται στο σχήμα 4.16 το κάθε panel πειράματος έχει ένα κουμπί “Start Button”, που στο UI έχει το όνομα “Ας αρχίσουμε το πείραμα”. Στο On Click () πρέπει να εφαρμοστεί η μέθοδος ChangeScene του script MainMenuController από το game object Menu Controller, και στο text box να μπει ο αριθμός που αντιστοιχεί στην σκηνή του πειράματος, για παράδειγμα το πείραμα 1-1 έχει τον αριθμό: 1, το πείραμα 2-2 έχει τον αριθμό 3, κτλ.



Σχήμα 4.16: Εισαγωγή και Αρίθμηση Σκηνών στο Scenes In Build

4.11 Επίλογος

Με την ολοκλήρωση του αρχικού μενού ο χρήστης έχει έναν ολοκληρωμένο τόπο πλοήγησης με τον οποίο θα μπορεί να οδηγηθεί στα πειράματα του παιχνιδιού.

Κεφάλαιο 5ο: 3D μοντέλα, σκίαση και ειδικά εφέ

5.1 Εισαγωγή

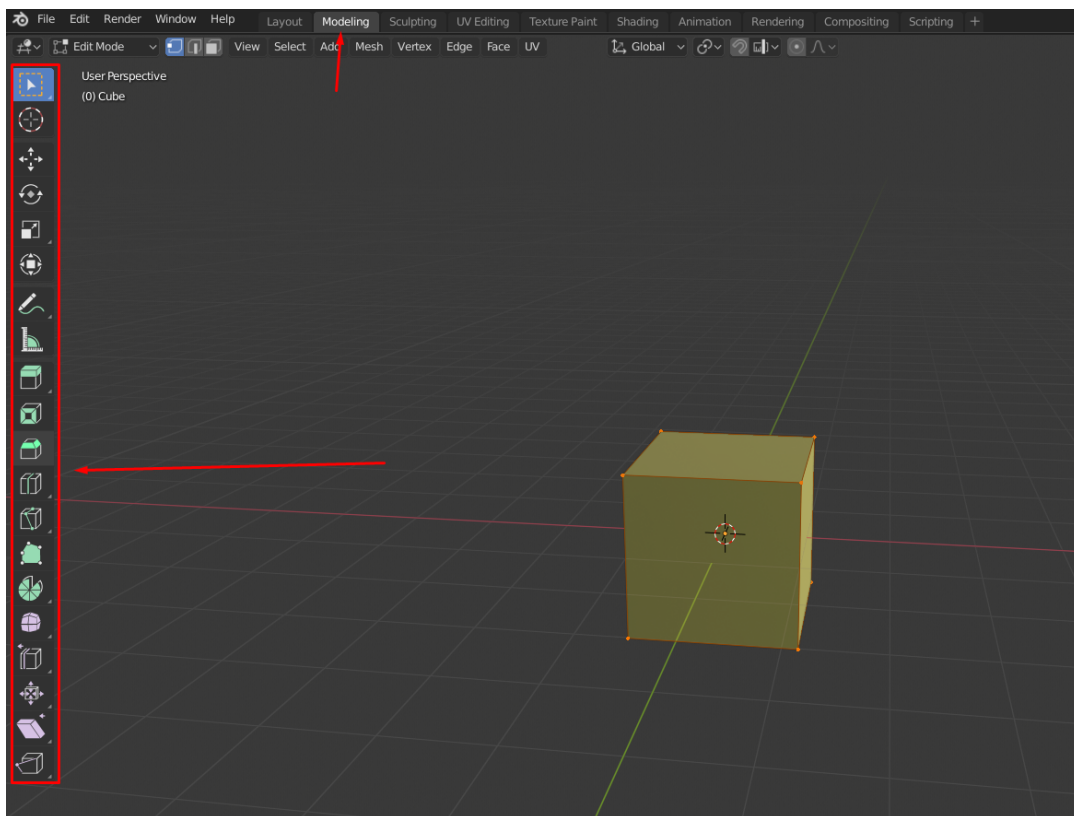
Από τα σημαντικότερα χαρακτηριστικά μιας εφαρμογής είναι τα γραφικά, καθώς προσφέρουν την βέλτιστη εικόνα με αυτό που θέλει να προσφέρει ο προγραμματιστής και ανάλογα το κοινό που έχει ως στόχο η εφαρμογή, πρέπει να εφαρμόζεται στις δικές τους ανάγκες. Σε αυτό το κεφάλαιο αναφέρονται οι τεχνικές και οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση των γραφικών της Π.Ε. Πιο συγκεκριμένα η δημιουργία 3D μοντέλων με τη χρήση του λογισμικού Blender, με και χωρίς textures, ο προγραμματισμός των σκιάσεων (shaders) με το πακέτο Universal RP και τέλος η κατασκευή ειδικών εφέ (VFX) με την χρήση του πακέτου Visual Effect Graph.

5.2 Κατασκευή 3D μοντέλων με το πρόγραμμα Blender

Το Blender, όπως αναφέρθηκε και στο κεφάλαιο 3, είναι ένα ισχυρό πρόγραμμα κατασκευής 3D γραφικών, καθώς έχει και άλλες δυνατότητες όπως η δημιουργία animations.

5.2.1 Δημιουργία 3D μοντέλου

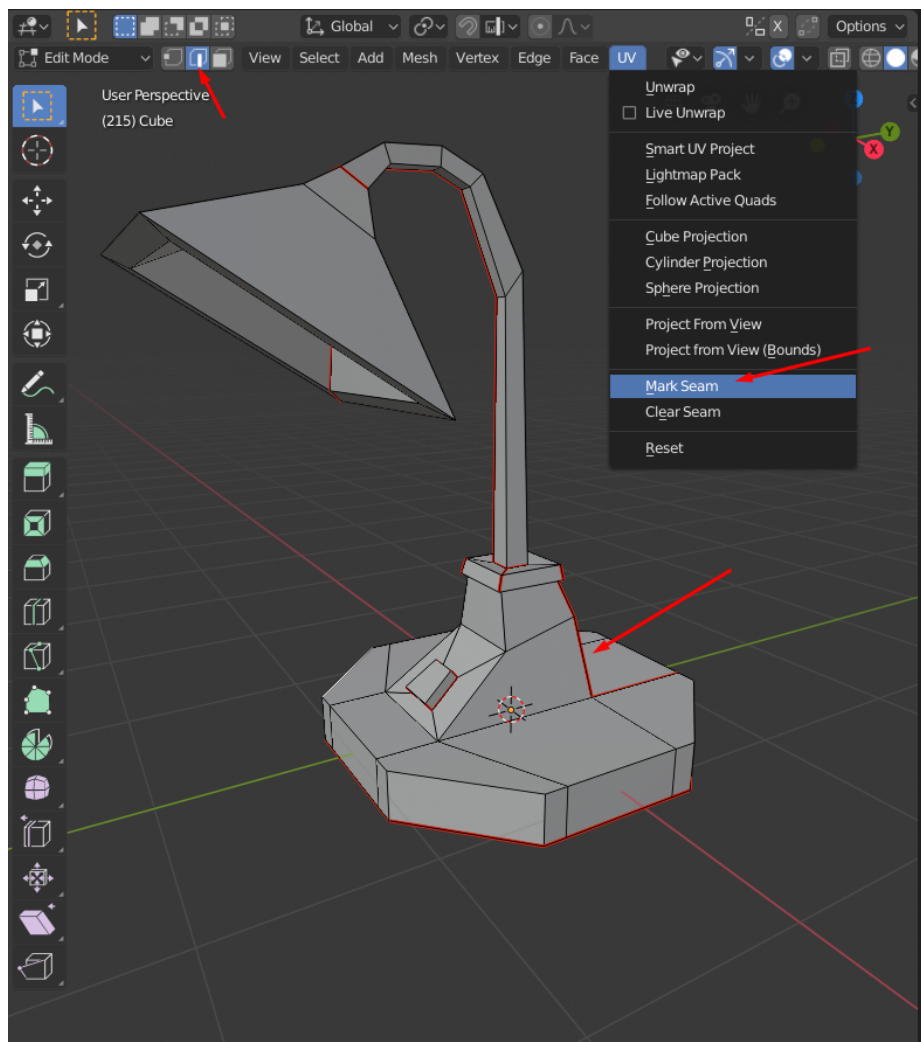
Ανοίγοντας για πρώτη φορά το πρόγραμμα Blender εμφανίζεται στον 3D χώρο, ένας κύβος, μια κάμερα και ένα σημείο φωτός, από τα οποία μόνο ο κύβος είναι απαραίτητος, ενώ τα υπόλοιπα δύο πρέπει να διαγραφούν. Στο πάνω μέρος της οθόνης πρέπει να επιλεγεί η επιλογή Modeling όπως στο σχήμα 5.1. και με τα εργαλεία στα αριστερά γίνεται η διαμόρφωση του κύβου στο επιθυμητό σχήμα.



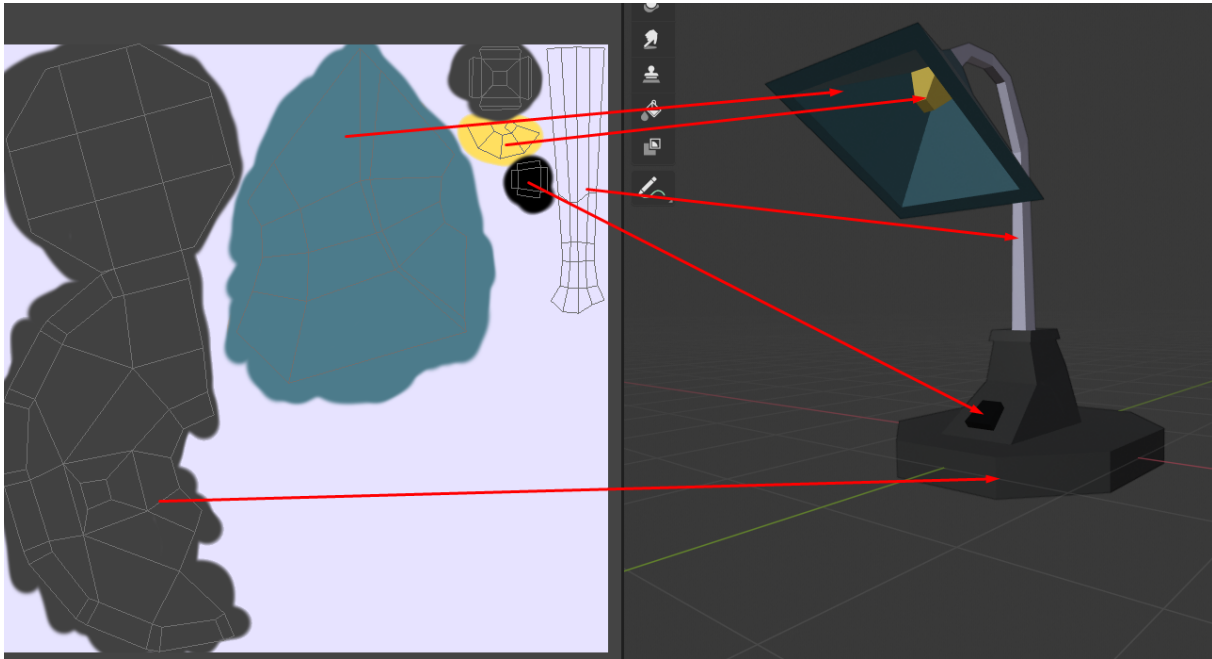
Σχήμα 5.1: Αρχική Οθόνη του Blender

5.2.2 UV Unwrap και χρωμάτωση μοντέλου

Μερικά μοντέλα, αφού δημιουργηθούν, είναι έτοιμα για χρήση από το Unity χωρίς κάποια επιπλέον μετατροπή χρωμάτων. Σε ορισμένες περιπτώσεις, που τα μοντέλα έχουν περίπλοκους συνδυασμούς χρωμάτων, πρέπει να δημιουργηθεί ένα αρχείο εικόνας PNG που ονομάζεται Texture Image και αντιπροσωπεύει κάθε pixel του μοντέλου σε έναν “χάρτη” πάνω στο αρχείο εικόνας [18]. Για να ξεχωρίσει το Blender ποιά pixels του μοντέλου αντιπροσωπεύουν τα pixels της εικόνας, πρέπει να χρησιμοποιηθεί το UV Editing στο πάνω μενού του παραθύρου. Έπειτα, επιλέγοντας το edge select όπως παρουσιάζεται στο σχήμα 5.2., πρέπει να επιλεγθούν οι πλευρές που θα “κοπούν” και στη συνέχεια πατώντας το UV→Mark Seam, διαχωρίζονται οι πλευρές με μια κόκκινη γραμμή. Για να ολοκληρωθεί ο διαχωρισμός πρέπει να πατηθεί το UV→Unwrap, όσο είναι επιλεγμένα όλα τα πολύγωνα του μοντέλου, επιλέγοντας οτιδήποτε στο 3D μοντέλο και πατώντας δυο φορές το πλήκτρο “A”. Αφού έχει γίνει η αντιστοίχιση στο UV Editor πρέπει να δημιουργηθεί η εικόνα με τα χρώματα, επιλέγοντας το Texture Paint στο πάνω μέρος του παραθύρου. Η δημιουργία της εικόνας βρίσκεται στο Image→New, και τώρα το μόνο που απομένει είναι ο χρωματισμός της όπως φαίνεται στο σχήμα 5.3.



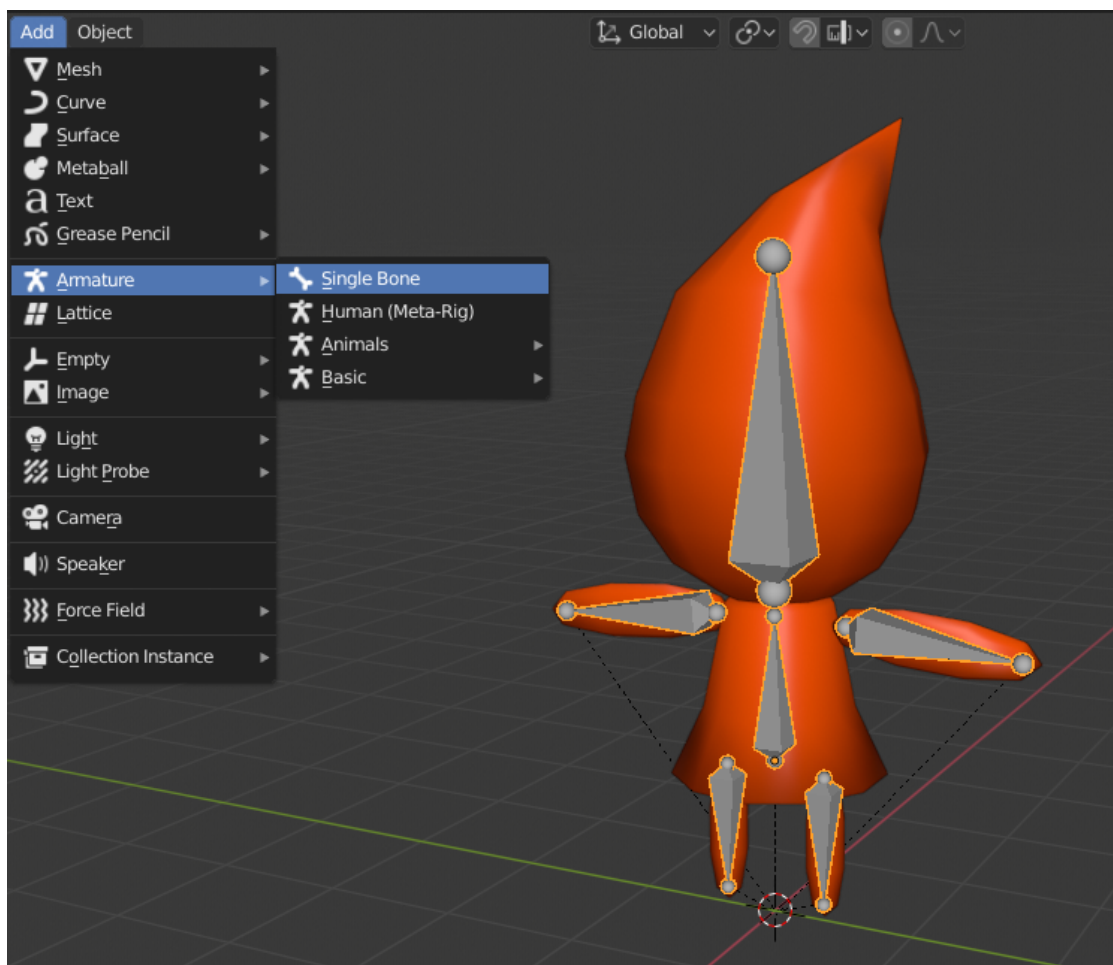
Σχήμα 5.2: UV Editor



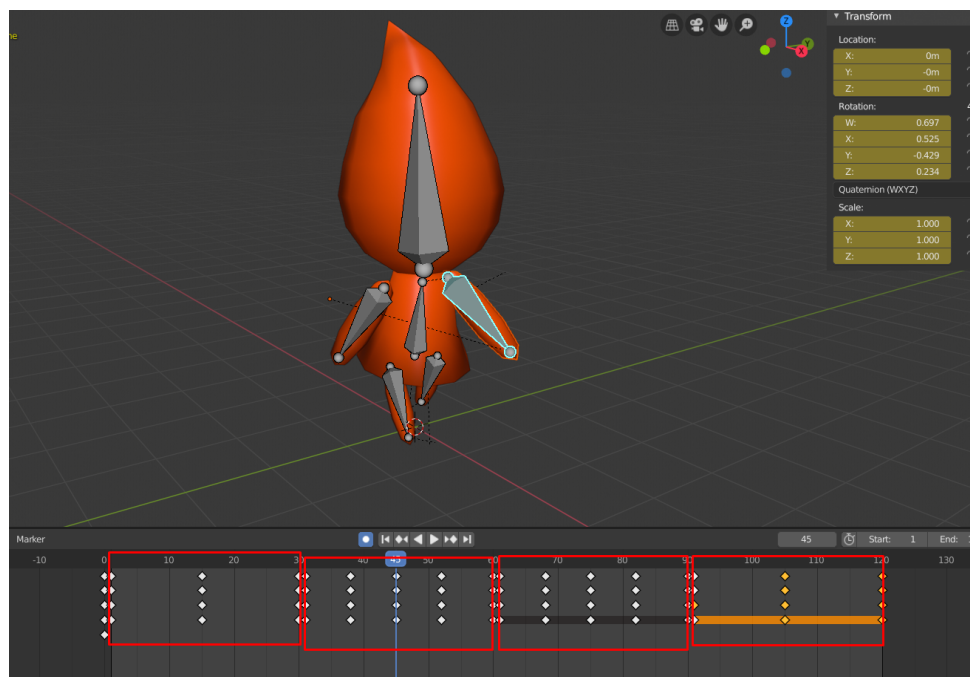
Σχήμα 5.3: Texture Paint

5.2.3 Χαρακτήρας του παιχνιδιού

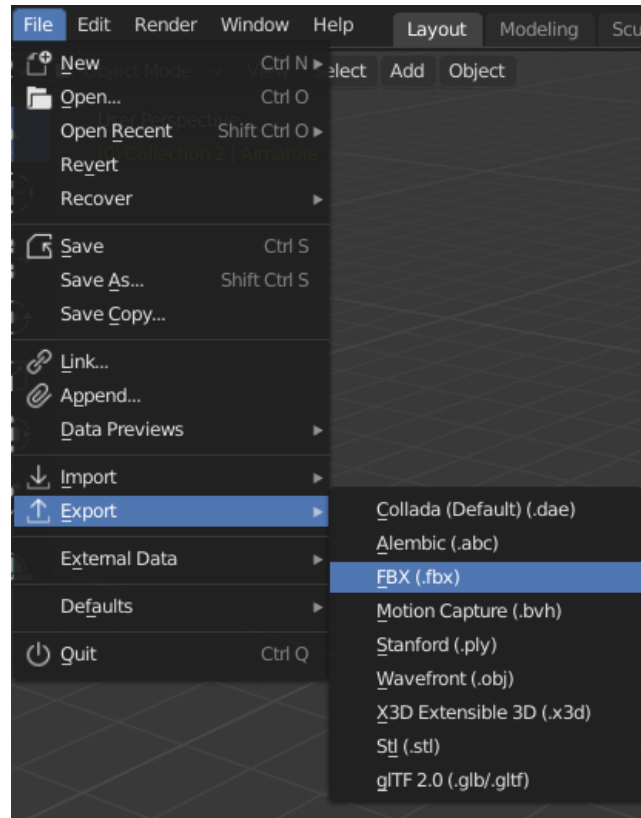
Το παιχνίδι της Π.Ε. χρησιμοποιεί έναν κεντρικό χαρακτήρα με τη μορφή μιας σταγόνας με χαρακτηριστικά ανθρώπου, που βοηθάει τους μαθητές να ολοκληρώσουν τα πειράματά τους. Η δημιουργία του έγινε εξ ολοκλήρου στο blender, δηλαδή η μοντελοποίηση, η χρωμάτιση (texturing), η τοποθέτηση εικονικών κοκκάλων στον χαρακτήρα (rigging) και το animation. Η μοντελοποίηση και το texturing έγιναν με τις τεχνικές που αναφέρθηκαν παραπάνω, ενώ στην συνέχεια παρουσιάζονται οι τρόποι που έγινε το rigging και το animation. Αφού κατασκευαστεί το βασικό μοντέλο, πρέπει να δημιουργηθούν τα κόκκαλα πατώντας Add→Armature→Single Bone(σχήμα 5.4), σχηματίζοντας έναν βασικό ανθρώπινο σκελετό με ένα κεφάλι, έναν κορμό, δύο χέρια και δύο πόδια. Στη συνέχεια κάθε κόκκαλο πρέπει να ενωθεί με το αντίστοιχο μέρος του σώματος, καθώς είναι διαφορετικά αντικείμενα και δεν ενώνονται μεταξύ τους. Επιλέγοντας πρώτα το μοντέλο και επιλέγοντας μετά το αντίστοιχο κόκκαλο με το κουμπί shift, στο Pose Mode, πρέπει στη συνέχεια να πατηθεί ο συνδυασμός κουμπιών Ctrl+P και να επιλεχθεί η επιλογή Bone, προσέχοντας να είναι επιλεγμένο το κουτάκι Keep Transform. Αφού ολοκληρώθηκε το Rigging, το μοντέλο είναι έτοιμο να “ζωντανέψει” και να αρχίσει να κινείται. Η επιλογή Animation στο πάνω μέρος του παραθύρου πρέπει να επιλεχθεί για να μπορέσει να αρχίσει η διεργασία του animation. Δημιουργώντας τα κατάλληλα στιγμιότυπα κλειδιά (Key Frames) όπως φαίνεται στο σχήμα 5.5. καταλήγουν να υπάρχουν 4 διαφορετικά animations, 4 δηλαδή τρόποι απεικόνισης ζωντανής κίνησης, οι οποίοι θα διαχωριστούν αργότερα στο Unity. Στο στιγμιότυπο 0 είναι ο χαρακτήρας κάνοντας την αρχική πόζα η οποία δεν θα υπάρχει στο παιχνίδι, από το καρέ 1 μέχρι το 30 είναι το animation του χαρακτήρα όταν είναι ακίνητος και δεν κρατάει τίποτα, από το καρέ 31 μέχρι το 60 είναι το animation του χαρακτήρα όταν περπατάει και δεν κρατάει τίποτα, από το καρέ 61 μέχρι το 90 είναι το animation του χαρακτήρα όταν περπατάει και κουβαλάει κάποιο αντικείμενο και τέλος από το καρέ 91 μέχρι το 120 είναι το animation του χαρακτήρα όταν είναι ακίνητο και κουβαλάει κάποιο αντικείμενο.



Σχήμα 5.4: Rigging του χαρακτήρα



Σχήμα 5.5: Animations του χαρακτήρα



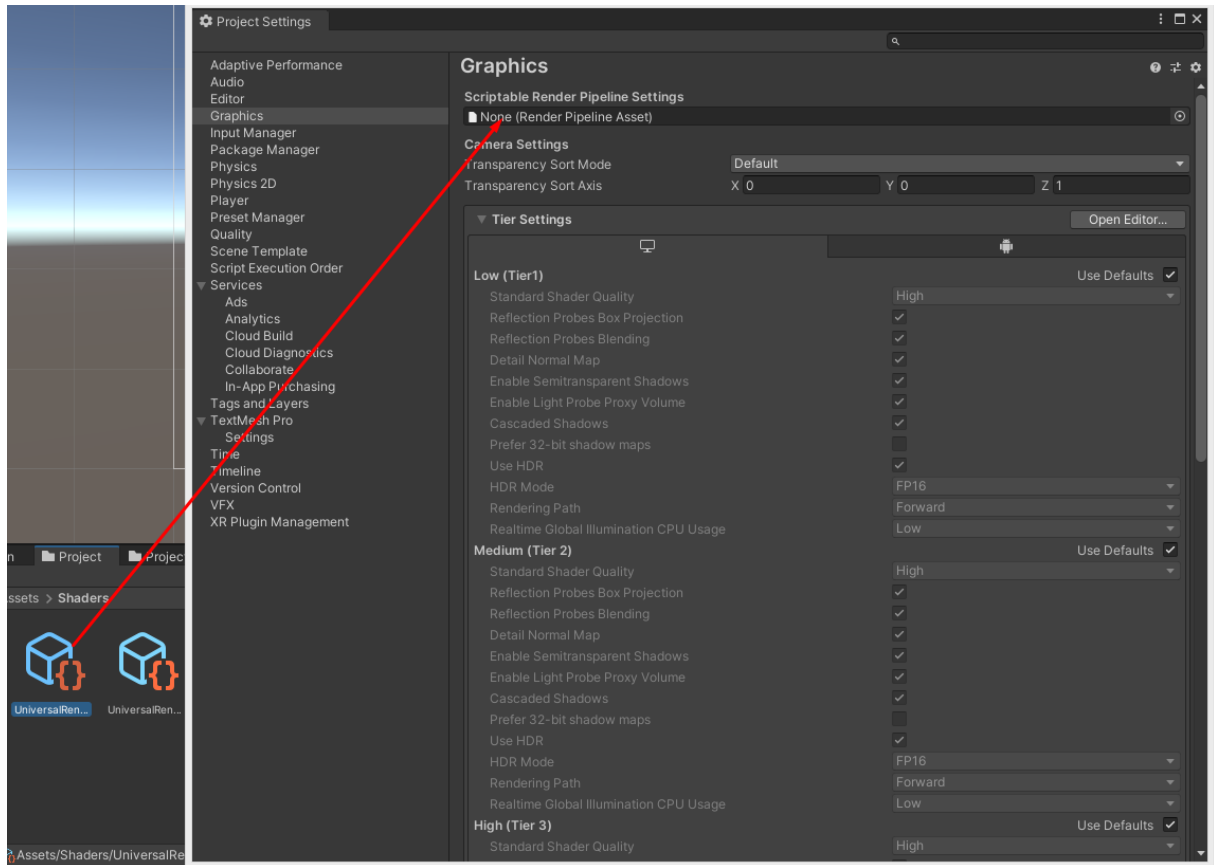
Σχήμα 5.6: Export αρχείο .fbx

5.2.4 Από το Blender στο Unity

Ένα απλό μοντέλο που δημιουργήθηκε χωρίς κάποιο ιδιαίτερο χαρακτηριστικό όπως texture ή animation, μπορεί να προστεθεί σέρνοντας το αρχείο blender που δημιουργήθηκε με την αποθήκευση του μοντέλου και αφήνοντάς το κατευθείαν σε έναν φάκελο στο project. Σε κάθε άλλη περίπτωση, ο καλύτερος τρόπος εισαγωγής ενός μοντέλου στο Unity, είναι να γίνει μετατροπή του αρχείου στην μορφή .fbx, πατώντας στο μενού File→Export→FBX(.fbx), όπως στο σχήμα 5.6., μεταφέροντας στη συνέχεια το αρχείο .fbx στο project του Unity. Αν το μοντέλο διαθέτει texture image πρέπει να συρθεί το αρχείο blender ή .fbx, ταυτόχρονα με την εικόνα χρώματος στον ίδιο φάκελο.

5.3 Universal RP στο Unity

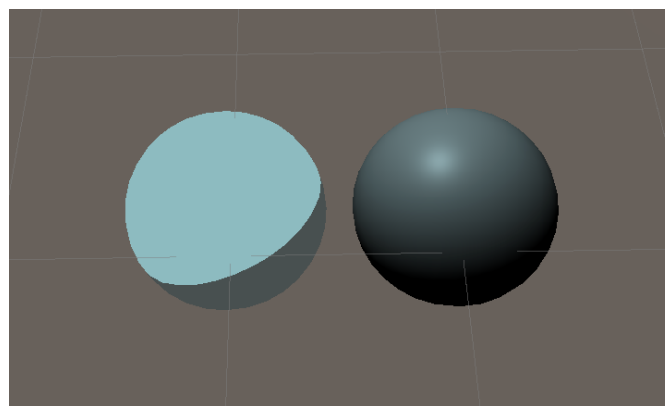
Με τη χρήση του URP είναι εφικτή η δημιουργία ειδικών υλικών (materials) στο Unity, έχοντας χαρακτηριστικά που ορίζει ο προγραμματιστής, τα οποία είναι δυνατόν να μεταβάλλονται μέσω script την ώρα που τρέχει η εφαρμογή, χωρίς να αλλάζει η γεωμετρία του μοντέλου. Αφού έχει εγκατασταθεί το πακέτο Universal RP, είναι προαιρετικό να δημιουργηθεί ένας φάκελος στο Project, για την καλύτερη οργάνωση, με όνομα Shaders. Μέσα στον φάκελο που δημιουργήθηκε, κάνοντας δεξί κλικ→Create→Rendering→Universal Render Pipeline→Pipeline Asset(Forward Renderer), και αφήνοντας το προεπιλεγμένο όνομα στο αρχείο που κατασκευάστηκε, δημιουργείται άλλο ένα αρχείο στον ίδιο φάκελο. Το επόμενο βήμα είναι να προστεθεί το αρχείο που δημιουργήθηκε στο πεδίο “Scriptable Render Pipeline Settings” στο μενού Graphics των Project Setting, όπως φαίνεται στο σχήμα 5.7. Με αυτόν τον τρόπο το Unity χρησιμοποιεί τις ρυθμίσεις που θα δημιουργηθούν και όχι τις προεπιλεγμένες.



Σχήμα 5.7: Εισαγωγή Renderer στις Ρυθμίσεις του Project

5.3.1 Toon Shader

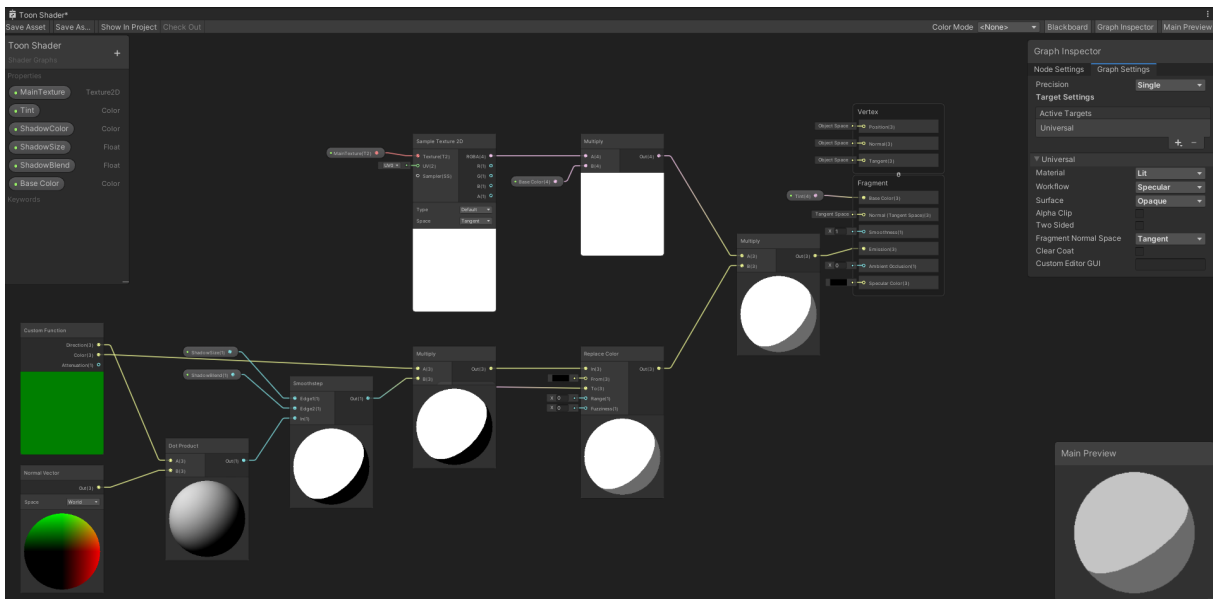
Λαμβάνοντας υπόψη ότι το παιχνίδι είναι φτιαγμένο για παιδιά ηλικίας 12 με 13 χρονών, πάρθηκε η απόφαση τα γραφικά να έχουν την όψη και υφή “καρτούν”, δηλαδή να μην υπάρχουν ομαλές αλλαγές στις σκιές, αλλά τα αντικείμενα να έχουν δύο ομάδες χρωμάτων, μια ομάδα για το φωτεινό κομμάτι και μια ομάδα για το σκοτεινό, όπως φαίνεται στο σχήμα 5.8. η αριστερή μπάλα. Η δεξιά μπάλα της εικόνας έχει τις προεπιλεγμένες ρυθμίσεις.



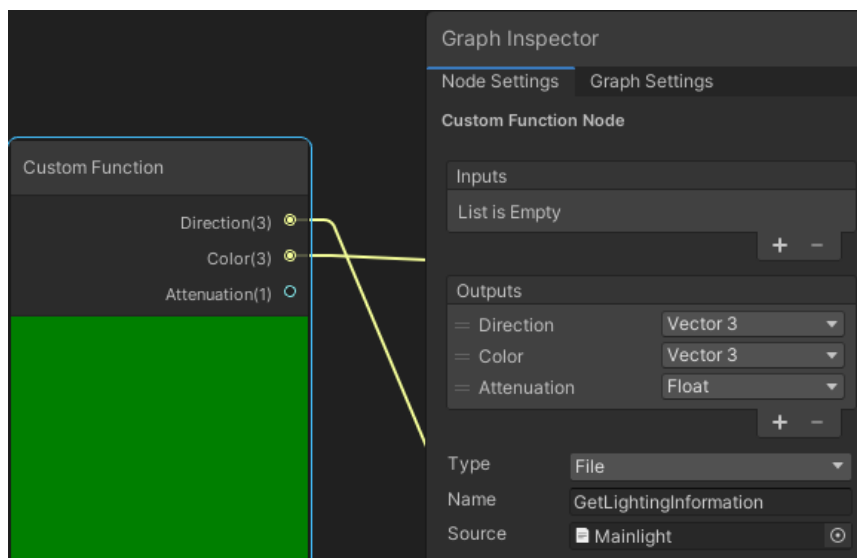
Σχήμα 5.8: Αριστερα: Αντικείμενο με Toon Shader. Δεξιά: Αντικείμενο με προεπιλεγμένο Material

Για τη δημιουργία της σκίασης (Shader) πρέπει να γίνει, στον φάκελο Shaders για την καλύτερη οργάνωση, δεξί κλικ→Create→Shader→Universal Render Pipeline→Lit Shader Graph. Κάνοντας

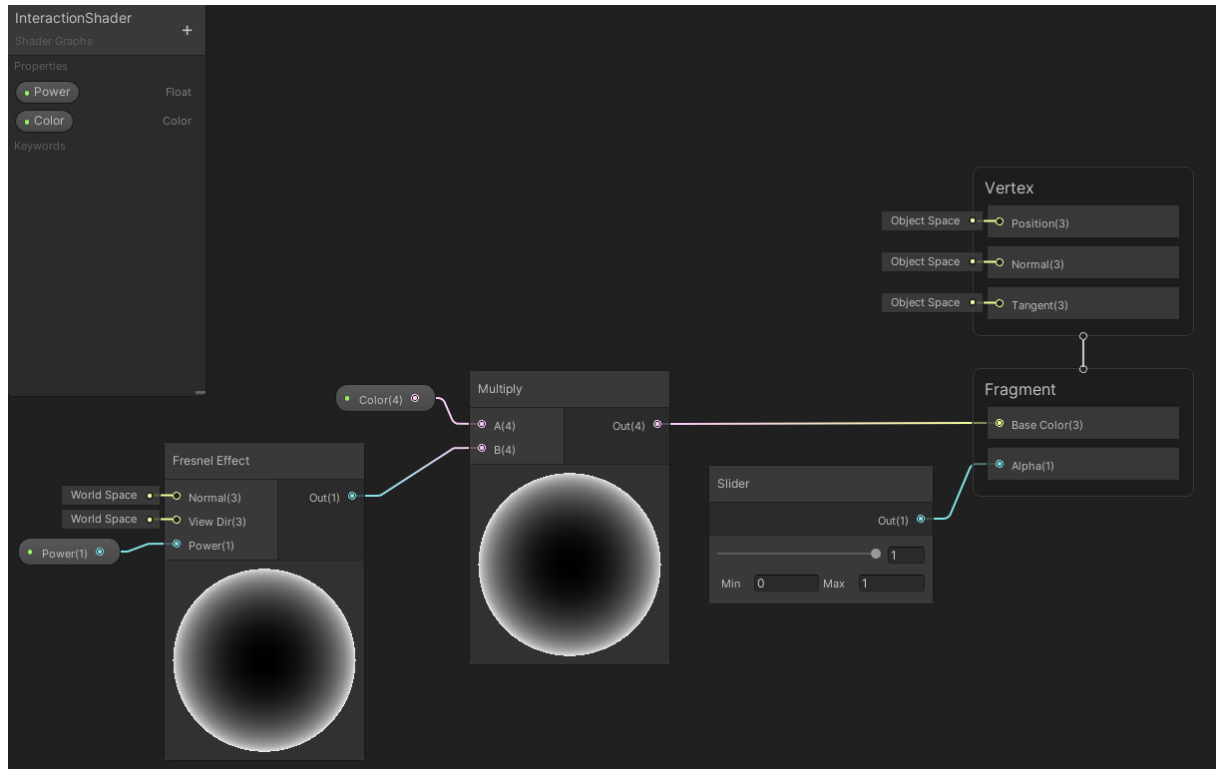
διπλό κλικ στο αρχείο που δημιουργήθηκε ανοίγει ένα νέο παράθυρο με το όνομα που δόθηκε στο αρχείο, το οποίο είναι ο χώρος δημιουργίας των shaders, με τη χρήση διαφόρων nodes και μεταβλητών. Χρησιμοποιώντας τα nodes και τις μεταβλητές που υπάρχουν στο σχήμα 5.9. μπορεί να γίνει η κατασκευή της συγκεκριμένης σκίασης (shader). Ο μόνος κόμβος (node) που χρειάζεται περισσότερες λεπτομέρειες για την κατανόησή του είναι το Custom Function, το οποίο απαιτεί το αρχείο Mainlight.hlsl (Παράρτημα Α) για να μπορεί να δεχτεί πληροφορίες για το κεντρικό φως της σκηνής. Πατώντας στον κόμβο: Custom Function πρέπει να επιλεγθεί στο Type η επιλογή File, στο πεδίο Name να πληκτρολογηθεί το όνομα της μεθόδου “GetLightingInformation” και στο Source να επιλεγθεί το αρχείο Mainlight(σχήμα 5.10). Τα nodes που χρησιμοποιήθηκαν είναι το Normal Vector, το Dot Product, το Smoothstep, το Replace Color, το Sample Texture 2D και μαθηματικές πράξεις.



Σχήμα 5.9: Toon Shader Graph Nodes



Σχήμα 5.10: Επιλογές Custom Function για τη Χρήση του Mainlight.hlsl



Σχήμα 5.11: Interactable Shader Graph Nodes

5.3.2 Interactable Shader

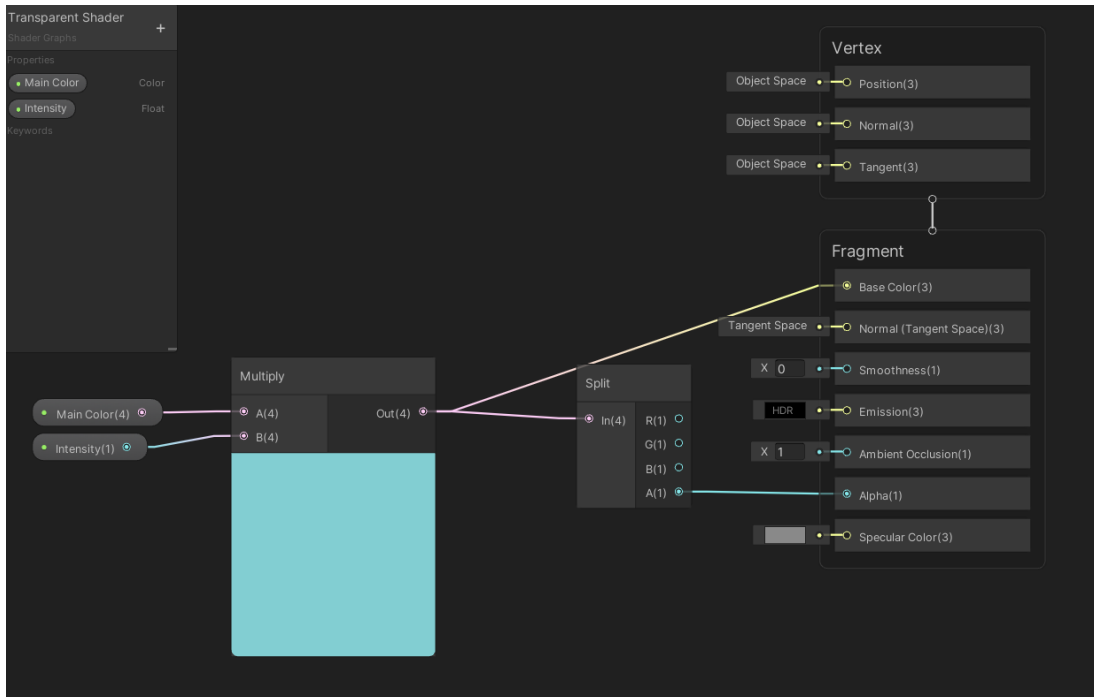
Όταν ο χρήστης θέλει να μετακινήσει τον χαρακτήρα σε κάποιο όργανο για να το χρησιμοποιήσει, πρέπει να ακουμπήσει έναν κύλινδρο που ονομάζεται σημείο αλληλεπίδρασης(Interactable). Για τη δημιουργία του shader είναι η ίδια διαδικασία με παραπάνω, με τη διαφορά ότι αντί για Lit Shader Graph, πρέπει να πατηθεί το Unlit Shader Graph, γιατί δεν χρειάζεται να αλληλεπιδρά με το φως της σκηνής. Το συγκεκριμένα γράφημα σκίασης (shader graph) είναι πιο απλό από το προηγούμενο, καθώς το μόνο που χρειάζεται είναι ένας κόμβος Fresnel Effect ενωμένο σε έναν κόμβο πολλαπλασιασμό με το χρώμα που θα δοθεί στο υλικό (material) (σχήμα 5.11).

5.3.3 Transparent Shader

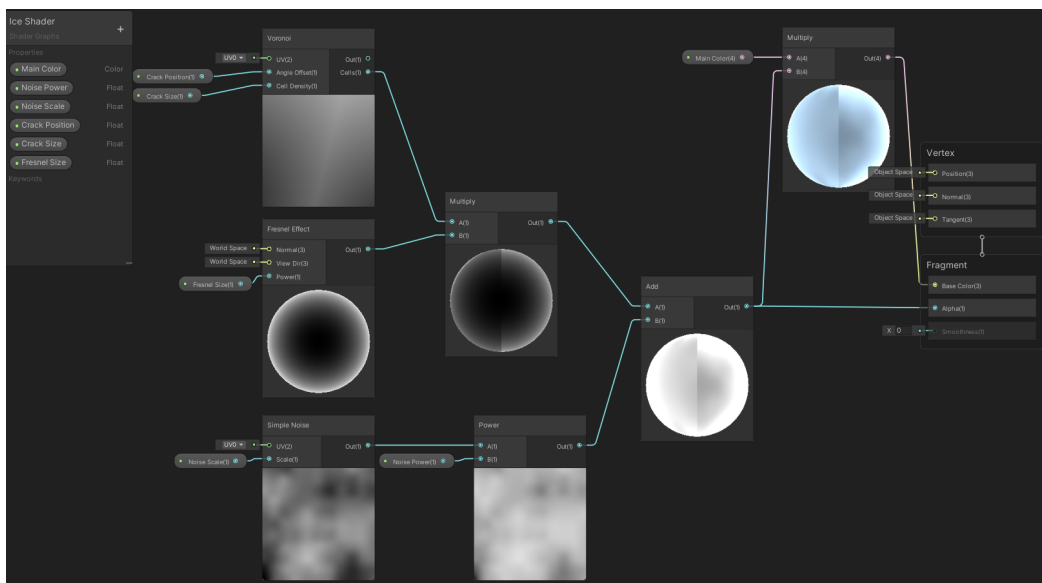
Δημιουργώντας ένα Lit Shader Graph, μπορεί να γίνει η κατασκευή του Transparent Shader, για να χρησιμοποιηθεί αργότερα σε αντικείμενα όπως γυάλινα μπουκάλια. Στο σχήμα 5.12 παρουσιάζονται οι κόμβοι και οι μεταβλητές που είναι απαραίτητα για την υλοποίηση της σκίασης.

5.3.4 Ice Shader

Η συγκεκριμένη σκίαση (shader) εμφανίζεται σε ορισμένα πειράματα που περιλαμβάνουν παγάκια, κάνοντας πιο ζωντανή την σκηνή. Είναι απαραίτητη η δημιουργία ενός Unlit Shader Graph με όνομα Ice Shader και στη συνέχεια πρέπει να κατασκευαστούν οι κόμβοι και οι μεταβλητές όπως στο σχήμα 5.13. Οι κόμβοι που χρησιμοποιούνται είναι το Voronoi, το Fresnel Effect, το Simple Noise και μαθηματικές πράξεις.



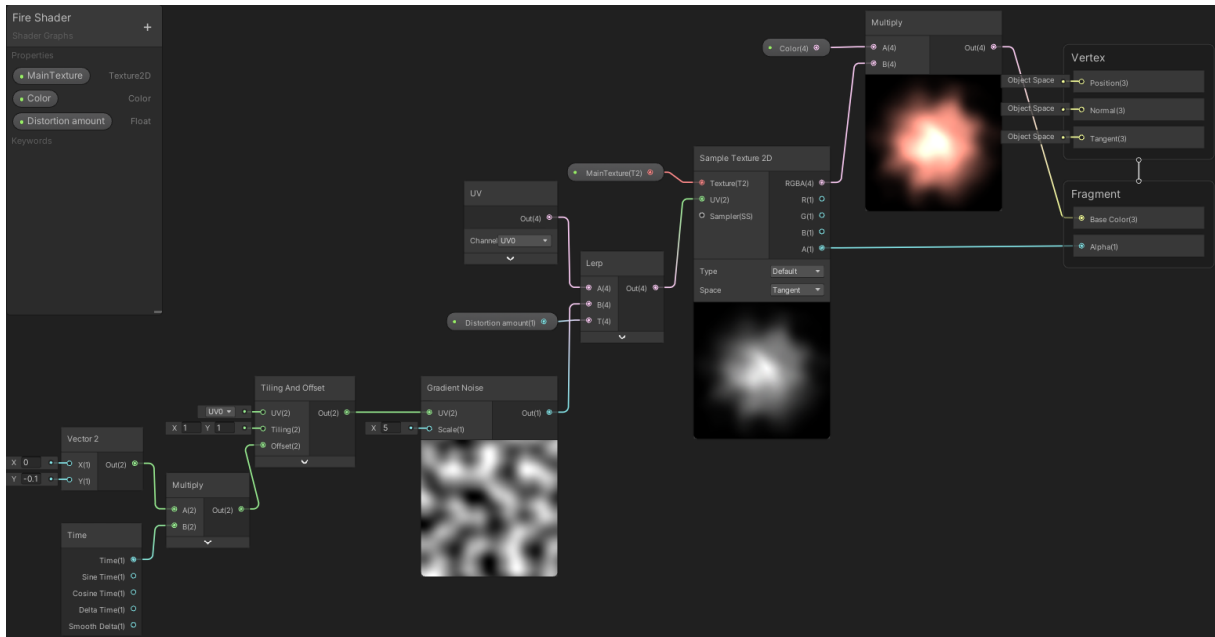
Σχήμα 5.12: Transparent Shader Graph Nodes



Σχήμα 5.13: Ice Shader Graph Node

5.3.5 Fire Shader

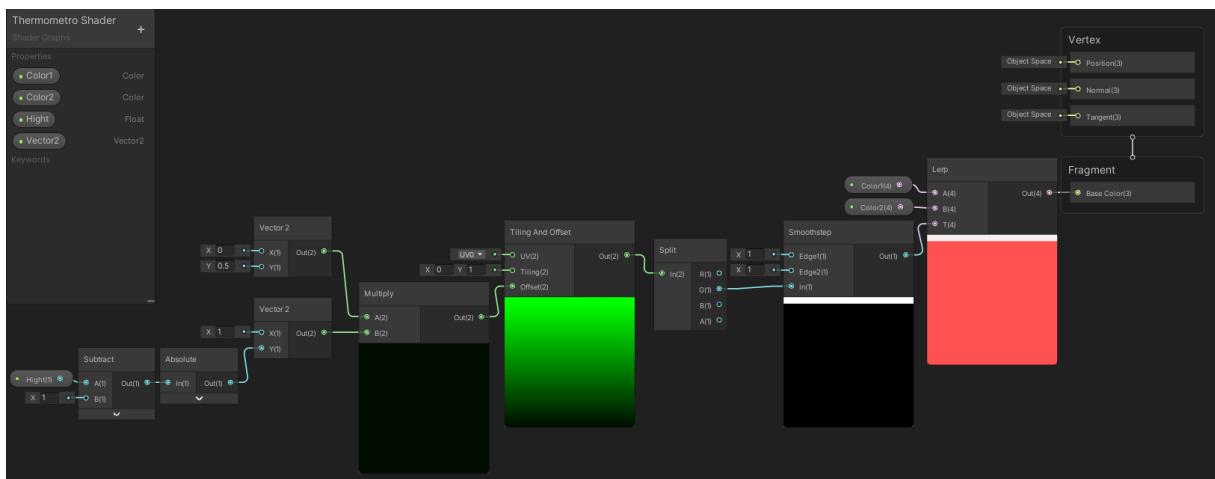
Το Fire Shader χρησιμοποιείται μόνο σε ένα αντικείμενο στο παιχνίδι, σαν φωτιά σε ένα κερί. Είναι Unlit Shader Graph και οι μεταβλητές όπως και οι κόμβοι παρουσιάζονται στο σχήμα 6.14. Πιο συγκεκριμένα οι κόμβοι που είναι απαραίτητα είναι το time, το Tiling and Offset, το Gradient Noise, το UV, το Lerp, το Simple Texture 2D και μαθηματικές πράξεις.



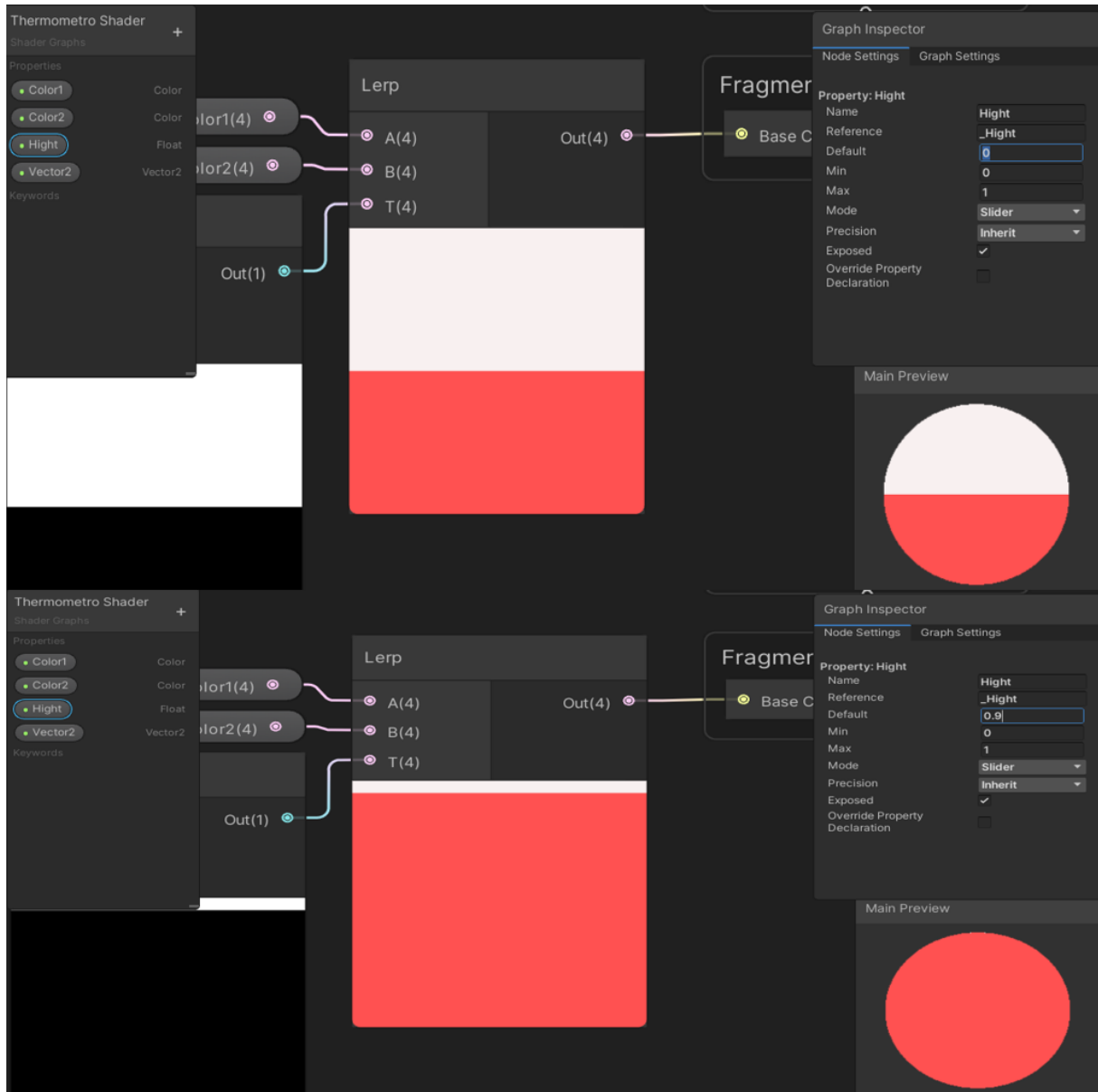
Σχήμα 5.14: Fire Shader Graph Nodes

5.3.6 Thermometer Shader

Στα περισσότερα πειράματα είναι απαραίτητη η χρήση θερμομέτρου, για αυτόν τον λόγο δημιουργήθηκε μία ειδική σκίαση (shader) για το υγρό του. Είναι ένα Unlit Shader Graph το οποίο έχει ως παραμέτρους δύο χρώματα και ένα float για το ύψος της θερμοκρασίας (η μεταβλητή Vector2 δεν χρειάζεται). Για τη δημιουργία του γραφήματος σκίασης(shader graph) δημιουργούνται οι κόμβοι Tiling and Offset, το Split, το Smoothstep, το Lerp και μαθηματικές πράξεις(σχήμα 5.15). Αλλάζοντας τον αριθμό της μεταβλητής Height, την ώρα που η εφαρμογή είναι ενεργή, εμφανίζεται δυναμικά η αλλαγή στο υλικό (material) όπως στο σχήμα 5.16.



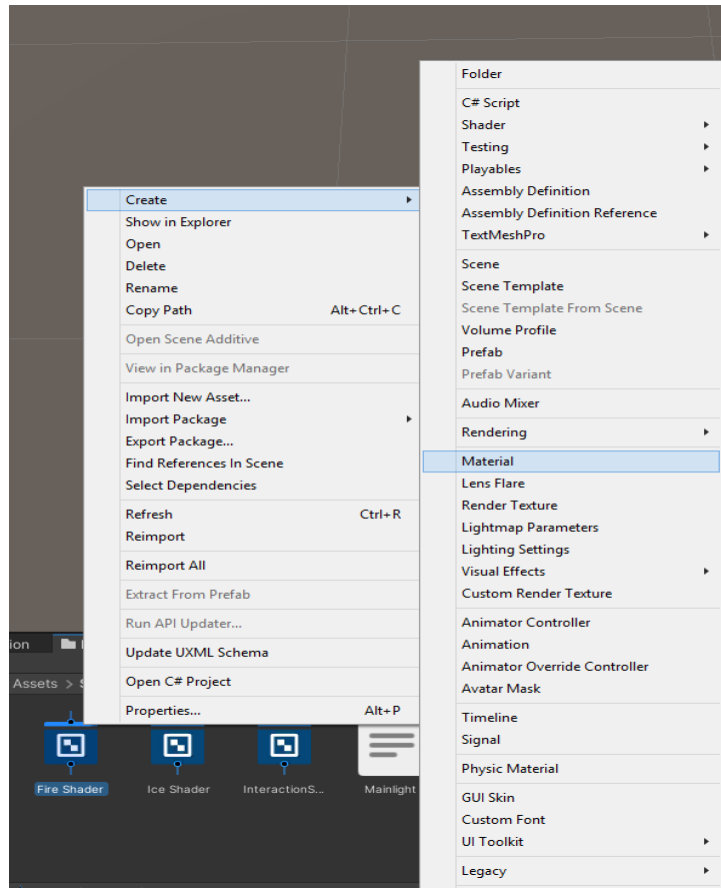
Σχήμα 5.15: Thermometer Shader Graph Nodes



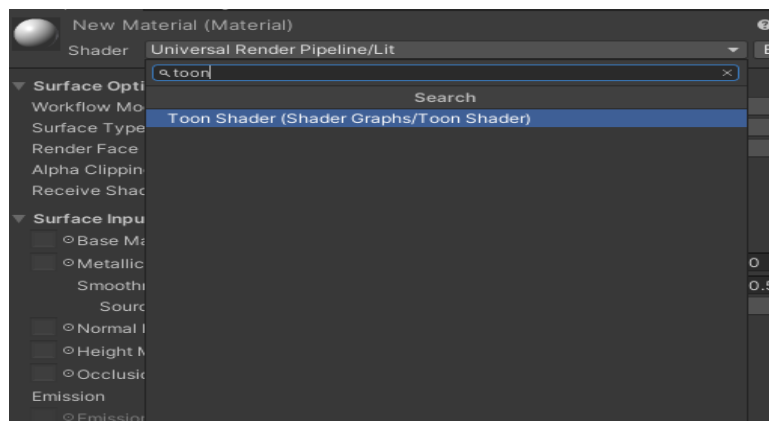
Σχήμα 5.16: Αλλαγή της μεταβλητής Height

5.3.7 Δημιουργία υλικού με τη χρήση σκίασης

Αφού έχουν κατασκευαστεί τα Shader Graphs, μπορούν να δημιουργηθούν υλικά που χρησιμοποιούν ως βάση τα πρώτα. Υπάρχουν δύο τρόποι να φτιαχτεί ένα υλικό. Ο πρώτος τρόπος είναι, κάνοντας δεξί κλικ σε ένα shader graph → Create → Material, φτιάχνοντας κατευθείαν ένα υλικό με τις ιδιότητες του συγκεκριμένου shader graph (σχήμα 5.17). Ο δεύτερος τρόπος είναι, δημιουργώντας ένα υλικό σε οποιοδήποτε φάκελο του Project και αλλάζοντας την σκίαση στο Inspector όπως φαίνεται στο σχήμα 5.18. Τέλος στο Inspector εμφανίζονται όλες οι μεταβλητές που έγιναν κατά τη κατασκευή του γραφήματος σκίασης, όπως το χρώμα, το texture image και άλλα.



Σχήμα 5.17: Δημιουργία Material από Shader Graph



Σχήμα 5.18: Αλλαγή Shader ενός Material

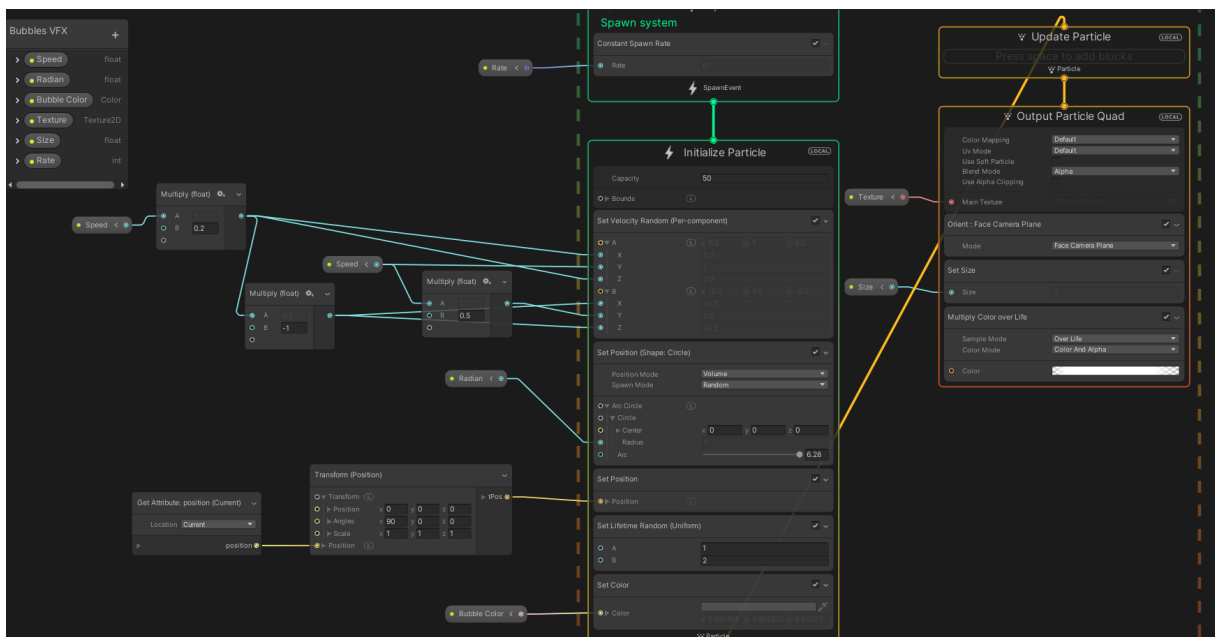
5.4 Visual Effects στο Unity

Τα VFX ή αλλιώς σωματίδια (particles), είναι εικόνες που εμφανίζονται και εξαφανίζονται και συνήθως υπάρχουν σε μεγάλες ποσότητες. Για τον πυρήνα ενός VFX πρέπει, αρχικά να δημιουργηθεί ένας φάκελος με όνομα VFX για την καλύτερη οργάνωση, και μέσα στον φάκελο να γίνει δεξί κλικ→ Create→Visual Effects→Visual Effect Graph. Κάνοντας διπλό κλικ στο αρχείο που κατασκευάστηκε ανοίγει ένα παράθυρο με κόμβους, παρόμοιο με το γράφημα σκίασης, που πρέπει ως αποτέλεσμα να δημιουργηθεί το γράφημα κόμβων του σχήματος 5.19. Τα Inputs βρίσκονται στο πάνω αριστερά μέρος

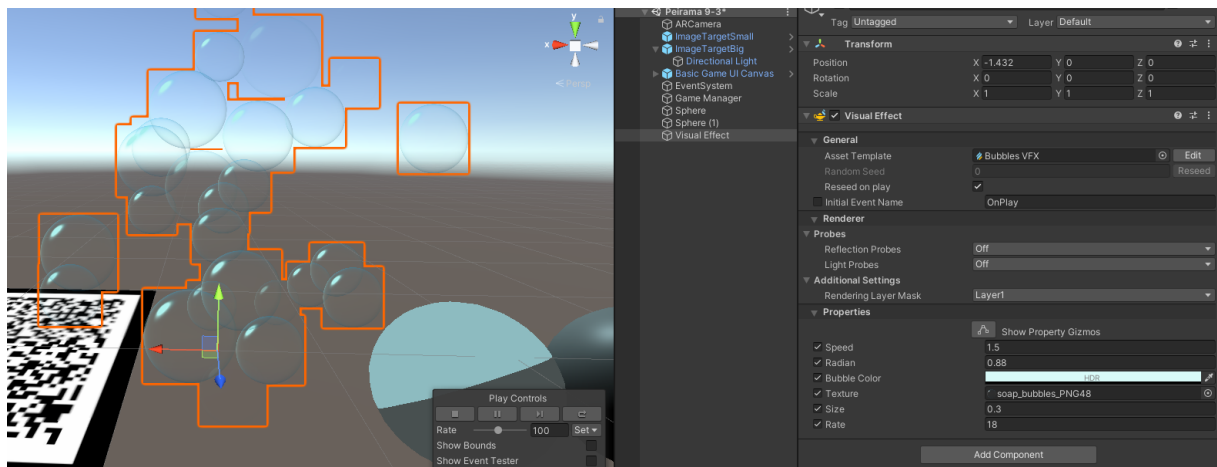
του παραθύρου, με τις μεταβλητές Speed, Radian, Bubble Color, Texture, Size, Rate, πιο αναλυτικά ο ρόλος των μεταβλητών είναι:

- Speed: πόσο γρήγορα θα κινούνται τα σωματίδια
- Radian: πόσο μεγάλος θα είναι ο κύκλος που θα εμφανίζονται τα σωματίδια
- Bubble Color: το χρώμα των σωματιδίων σε συνδυασμό με την εικόνα τους
- Texture: η εικόνα που θα έχουν τα σωματίδια
- Size: το μέγεθος των σωματιδίων
- Rate: πόσο συχνά θα εμφανίζονται τα σωματίδια

Ο τρόπος για να κατασκευαστεί μέσα στην ιεραρχία γίνεται με δεξί κλικ→Visual Effects→Visual Effect. Στο κενό VFX πρέπει εισαχθεί το asset που δημιουργήθηκε παραπάνω, σέρνοντας το asset στο πλαίσιο “Asset Template”. Έπειτα είναι απαραίτητο να αλλάξουν οι μεταβλητές των properties για να διαφοροποιηθούν κατάλληλα τα game objects(σχήμα 5.20).



Σχήμα 5.19: VFX Graph Nodes



Σχήμα 5.20: Bubble Particles

5.5 Επίλογος

Η παρούσα Π.Ε. χρησιμοποιεί τρισδιάστατα μοντέλα να την οπτικοποίηση των πειραμάτων, τα οποία δημιουργήθηκαν με το πρόγραμμα Blender. Το κυριότερο στοιχείο των γραφικών είναι οι σκιάσεις Toon Shaders, που δίνουν έναν ευχάριστο τόνο στο παιχνίδι, κάνοντας τα πειράματα πιο προσιτά για παιδιά μικρής ηλικίας. Οι υπόλοιπες σκιάσεις έγιναν για να κάνουν πιο διαδραστικό και ζωντανό το περιβάλλον των πειραμάτων, ενώ η προσθήκη ειδικών εφέ, μια μικρή λεπτομέρεια, προσφέρει έναν ρεαλιστικό τόνο.

Κεφάλαιο 6ο: Κοινές λειτουργίες σκηνών

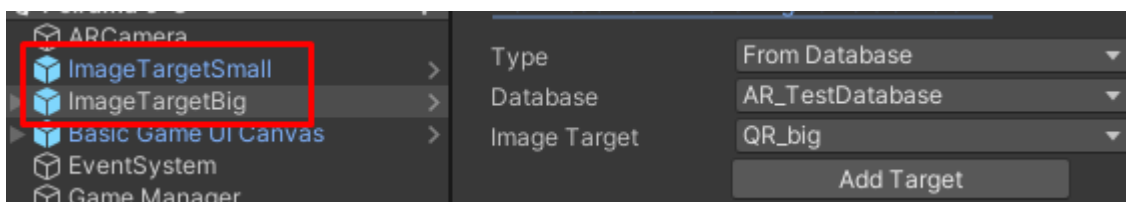
6.1 Εισαγωγή

Όλα τα πειράματα είναι διαφορετικά μεταξύ τους, έχοντας ξεχωριστά scripts και αντικείμενα στις σκηνές τους. Ο κορμός όλων των πειραμάτων είναι κοινός, καθώς μοιράζονται ορισμένα χαρακτηριστικά στην ιεραρχία των αντικειμένων. Τα παρακάτω χαρακτηριστικά αναφέρονται αναλυτικά στη συνέχεια του κεφαλαίου:

- Vuforia: target και images κάμερα.
- Scripts: ορισμένα scripts που χαρίζουν τα χαρακτηριστικά του σε άλλα scripts σε διαφορετικά πειράματα.
- Tags και Layers: ετικέτες και επίπεδα
- Navigation system: ο τρόπος που κινείται ο χαρακτήρας του παιχνιδιού.
- UI: κουμπιά.
- Interactable Items: ρύθμιση τρόπου αλληλεπίδρασης του χαρακτήρα με τα αντικείμενα.

6.2 Vuforia: target images και κάμερα

Όπως έχει προαναφερθεί, το παιχνίδι χρησιμοποιεί το πακέτο Vuforia για να εκτελέσει την τεχνολογία “AR”. Αφού έχουν γίνει τα βήματα του κεφαλαίου 4 για την εγκατάσταση και τις αρχικές ρυθμίσεις του πακέτου. Το επόμενο βήμα είναι να δημιουργηθούν τα βασικά αντικείμενα σε όλες τις σκηνές της εφαρμογής, κάνοντας στην ιεραρχία δεξί κλικ→Vuforia Engine→Image Target δύο φορές. Ονομάζοντας το ένα αντικείμενο “ImageTargetSmall” και “ImageTargetBig”, πρέπει στο Inspector η μεταβλητή Type να γίνει “From Database”, η επιλογή Database να γίνει “AR_TestDatabase” (ή το όνομα της βάσης) και τέλος το Image Target να γίνει QRcode_small στο πρώτο αντικείμενο και QR_big στο δεύτερο αντικείμενο (σχήμα 6.1). Στο αντικείμενο ImageTargetBig πρέπει να δημιουργηθεί ένα αντικείμενο φωτός, κάνοντας δεξί στο πρώτο→Light→Directional Light, για να υπάρχει πηγή φωτός στη σκηνή. Όλα τα αντικείμενα που δημιουργούνται θα έχουν ως parent το TargetImageBig, εκτός αν αναφέρεται κάτι άλλο. Τα δύο αντικείμενα μπορούν να γίνουν prefabs για την ευκολία δημιουργίας της σκηνής, σέρνοντάς τα(drag and drop) σε κάποιον φάκελο του project. Για λόγους οργάνωσης δημιουργήθηκε φάκελος με όνομα prefabs στο project. Αφού έχει δημιουργηθεί και ρυθμιστεί η ARCamera, υπάρχει ένα πεδίο που δεν έχει οριστεί, το World Center, το οποίο πρέπει να έχει ως όρισμα το αντικείμενο ImageTargetBig. Ο λόγος είναι να γίνει το κέντρο του κόσμου το μεγάλο QR code, για μεγαλύτερη σταθερότητα κατα τη λειτουργία του παιχνιδιού.



Σχήμα 6.1: Target Images

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class DestroyAfterTime : MonoBehaviour
6  {
7      public float seconds;
8
9      public void StartTimer()
10     {
11         StartCoroutine(WaitAndDestroy());
12     }
13
14     IEnumerator WaitAndDestroy()
15     {
16         yield return new WaitForSeconds(seconds);
17         Destroy(this.gameObject);
18     }
19 }

```

Σχήμα 6.2: DestroyAfterTime Script

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Interactable : MonoBehaviour
6  {
7      protected PlayerController pController;
8
9      private void OnTriggerEnter(Collider other)
10     {
11         if (other.CompareTag("Player"))
12         {
13             pController = other.GetComponent<PlayerController>();
14             Interact();
15         }
16     }
17
18     protected virtual void Interact()
19     {
20     }
21 }
22

```

Σχήμα 6.3: Interactable Script

6.3 Scripts

Υπάρχουν δύο scripts τα οποία χρησιμοποιούνται στις περισσότερες σκηνές, τα οποία είναι το “DestroyAfterTime”(σχήμα 6.2) και τα “Interactable”(σχήμα 6.3). Ο σκοπός του πρώτου είναι να εξαφανίζεται το επιλεγμένο αντικείμενο μετά από προεπιλεγμένα δευτερόλεπτα, από τη στιγμή που δημιουργηθεί στη σκηνή. Το δεύτερο είναι μια κλάση που μπορεί να κληρονομηθεί και το μόνο που κάνει είναι να ελέγχει αν ακουμπήσει, το game object που έχει το script, με κάποιο αντικείμενο που φέρει το tag “Player”. Στη συνέχεια καλεί την μέθοδο protected virtual void Interact() {}, η οποία είναι κενή, και μπορεί να κληρονομηθεί γράφοντας protected override void Interact() {(κώδικας)} στην κλάση που είναι παιδί του Interactable.

6.4 Tags και Layers

Οι ετικέτες (tags) και τα επίπεδα (layers) είναι βοηθητικοί αλφαριθμητικοί χαρακτήρες (strings) για την οργάνωση και τον έλεγχο των game objects μέσω των scripts. Για να φτιαχτεί ένα καινούριο tag ή layer, πρέπει να είναι επιλεγμένο ένα οποιοδήποτε game object και στο Inspector να ανοίξει το drop down μενού του tag και του layer, επιλέγοντας την επιλογή Add Tag και Add Layer αντίστοιχα. Η μοναδική κατηγορία που προστέθηκε στα layers είναι το Interactable, το οποίο επιτρέπει στο αντικείμενο που το έχει, να δέχεται εισαγωγή με αφή (touch input). Οι κατηγορίες και οι σκοποί των tags είναι:

- Ground: Ο χαρακτήρας πηγαίνει στο σημείο που έγινε αλληλεπίδραση με αφή.
- Pick Up: Ο χαρακτήρας πηγαίνει στο σημείο που έγινε αλληλεπίδραση με αφή και εκτελεί την μέθοδο PickUpItem(), που αναφέρεται στη υποενότητα 6.5.
- Interactable: Ο χαρακτήρας πηγαίνει στον γονέα του αντικειμένου με τον οποίο έγινε αλληλεπίδραση με αφή.

6.5 Navigation system και ο τρόπος κίνησης του χαρακτήρα

Έχοντας κάνει “import” το μοντέλο του χαρακτήρα του παιχνιδιού και αφού του έχει δοθεί ο κατάλληλος χρωματισμός(texture) και σκίαση(shader), πρέπει να δημιουργηθούν δύο scripts:

- PlayerController: έλεγχος της κίνησης του χαρακτήρα
- PlayerAnimationController: έλεγχος των animations του χαρακτήρα.

Για τον έλεγχο του χαρακτήρα πρέπει να δημιουργηθεί μια επίπεδη επιφάνεια, κάνοντας στην Ιεραρχία δεξί κλικ→3D Object→Plane με tag: Ground και Layer: Interactable, με σκοπό την πλοήγησή του στον χώρο. Το Unity έχει ένα έτοιμο πακέτο για την πλοήγηση ενός αντικειμένου, κάνοντας κλικ σε ένα σημείο, το οποίο ονομάζεται Navigation και ο τρόπος πρόσβασης του παραθύρου γίνεται από το μενού Window→AI→Navigation. Έχοντας επιλεγμένο το επίπεδο (plane), στο παράθυρο του Navigation→Object πρέπει να ενεργοποιηθεί το Navigation Static και το Navigation Area να γίνει “Walkable”. Αν χρειαστεί ένα αντικείμενο να εμποδίζει τον χαρακτήρα να “περπατάει” στον χώρο του, η επιλογή Navigation Area πρέπει να είναι επιλεγμένη ως “Not Walkable”. Στη συνέχεια, στο παράθυρο Navigation→Bake, το Angle Radius πρέπει να έχει την μικρότερη δυνατή τιμή.

6.5.1 Components του χαρακτήρα

Κάνοντας εισαγωγή του μοντέλου του χαρακτήρα στην ιεραρχία, πρέπει να δημιουργηθεί ένα κενό game object ως γονέας με όνομα “Player” και μέσα σε αυτό το αντικείμενο άλλο ένα κενό game object με όνομα “Hold Position”. Στο αντικείμενο Player, πρέπει

στο Inspector, να προστεθούν τέσσερα συστατικά με τις ρυθμίσεις που αναφέρονται στη συνέχεια και στην σχήμα 6.4:

- Box Collider: το Is Trigger πρέπει να είναι επιλεγμένο.
- Rigidbody: το Use Gravity πρέπει να μην είναι επιλεγμένο και το Is Kinematic να είναι επιλεγμένο.
- Player Controller: το script που δημιουργήθηκε παραπάνω.
- Nav Mesh Agent:
 1. Base Offset: 0
 2. Speed: 1
 3. Angular Speed: 100000
 4. Acceleration: 1
 5. Stopping Distance: 0.3
 6. Radius: 0.1
 7. Height: 0.25
 8. Quality: Medium Quality
 9. Auto Repath: unchecked

6.5.2 Script: PlayerController

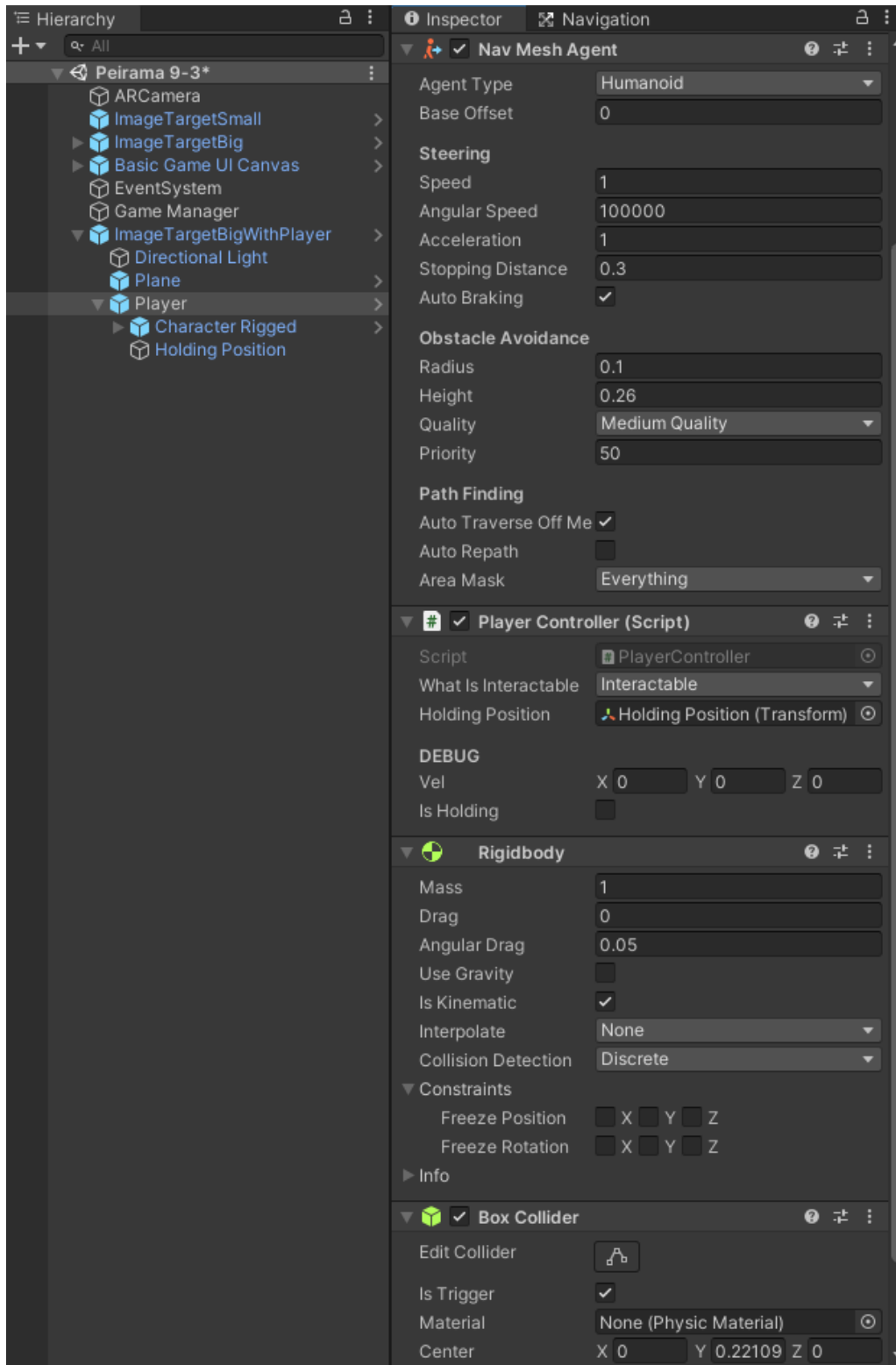
Το PlayerController(Παράρτημα Β) script είναι το βασικό συστατικό για την κίνηση του χαρακτήρα στο τρισδιάστατο περιβάλλον. Για τη λειτουργία του, είναι απαραίτητη η εισαγωγή δύο πακέτων με τον κώδικα using UnityEngine.AI; και using UnityEngine.EventSystems;. Αποτελείται από τις παρακάτω μεταβλητές και μεθόδους:

- NavMeshAgent myAgent: το συστατικό που προστέθηκε παραπάνω.
- [SerializeField] LayerMask whatIsInteractable: το layer που μπορεί να πατηθεί.
- [SerializeField] Transform holdingPosition: η θέση του game object Hold Position.
- public Vector3 vel = new Vector3(): η ταχύτητα του χαρακτήρα.
- public bool isHolding = false: έλεγχος αν ο χαρακτήρας κουβαλάει κάποιο αντικείμενο.

Στο Inspector η μεταβλητή “What Is Interactable” πρέπει να οριστεί ως Interactable και στη μεταβλητή Holding Position το game object “Holding Position”.

- void Start(): αρχικοποιεί το myAgent.
- void Update(): καλεί την μέθοδο CheckIfTouch και δηλώνει την ταχύτητα.
- void CheckIfTouch(): ελέγχει αν γίνετε touch και σε ποιο μέρος της οθόνης.
- void TtyInteract(RaycastHit hit): ελέγχει το tag του αντικειμένου που έγινε touch.
- void TryMove(Vector3 hit): μετακινεί τον χαρακτήρα στο σημείο που έγινε touch.
- public void PickupItem(Transform item): σηκώνει το αντικείμενο item και το κάνει παιδί του holdingPosition.
- public void PutDownButton(): τοποθετεί το αντικείμενο που κρατάει κάτω.
- public GameObject PutDownItem(Transform newParent, Vector3 newPosition): επιστρέφει το αντικείμενο που κρατάει και το τοποθετεί στο newPosition.
- public GameObject GetItemHolding(): επιστρέφει το αντικείμενο που κρατάει.

- private void OnTriggerEnter(Collider other): ελέγχει αν το αντικείμενο που ακουμπάει το collider του, έχει tag “Pick Up” και καλεί το PickupItem.

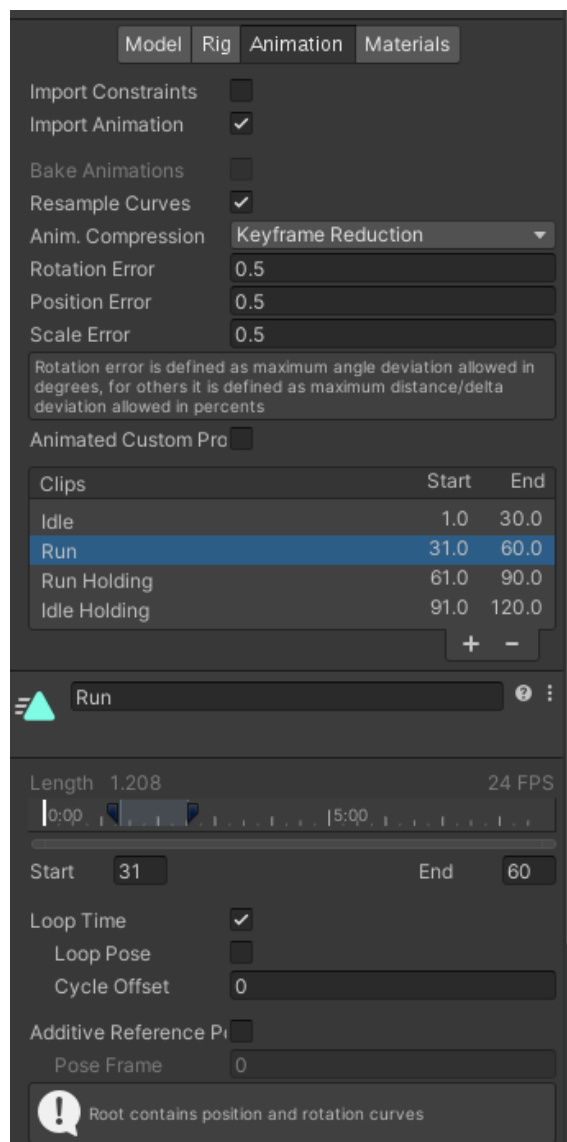


Σχήμα 6.4: Player Components

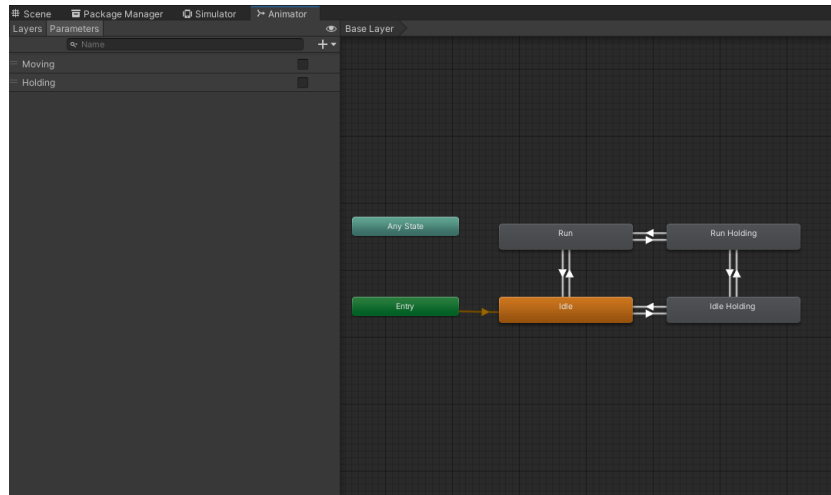
6.5.3 Animation του χαρακτήρα και Script: PlayerAnimationController

Στο μοντέλο του χαρακτήρα στο Inspector→Animation, πρέπει να διαχωριστούν οι απεικονίσεις που δημιουργήθηκαν στο Blender, όπως στο σχήμα 6.5. και στο παράθυρο Animator(ανοίγει από το Window→Animation→Animator), να δημιουργηθεί η δομή που φαίνεται στο σχήμα 6.6, με τις Trigger μεταβλητές “Moving” και “Holding”. Το child του Player, δηλαδή το μοντέλο του χαρακτήρα, χρειάζεται τα συστατικά Animator και Player Animation Controller. Το script PlayerAnimatorController(σχήμα 6.7) έχει τις παρακάτω μεταβλητές:

- private void Start(): αρχικοποιεί της μεταβλητές του pController και του anim.
- private void Update(): καλεί το ChooseAnimation.
- void ChooseAnimation(): ενεργοποιεί τα booleans “Moving” και “Holding” ανάλογα με το εάν ο χαρακτήρας κινείται και αν κρατάει αντικείμενο.



Σχήμα 6.5: Διαχωρισμός των Animations του Χαρακτήρα



Σχήμα 6.6: Animator Χαρακτήρα

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PlayerAnimationController : MonoBehaviour
6  {
7      PlayerController pController;
8      Animator anim;
9
10     private void Start()
11     {
12         pController = GetComponentInParent<PlayerController>();
13         anim = GetComponent<Animator>();
14     }
15
16     private void Update()
17     {
18         ChooseAnimation();
19     }
20
21     void ChooseAnimation()
22     {
23         if(pController.vel!=Vector3.zero)
24         {
25             anim.SetBool("Moving",true);
26         }
27         else
28         {
29             anim.SetBool("Moving", false);
30         }
31
32         if (pController.isHolding)
33         {
34             anim.SetBool("Holding", true);
35         }
36         else
37         {
38             anim.SetBool("Holding", false);
39         }
40     }
41 }

```

Σχήμα 6.7: PlayerAnimationController Script

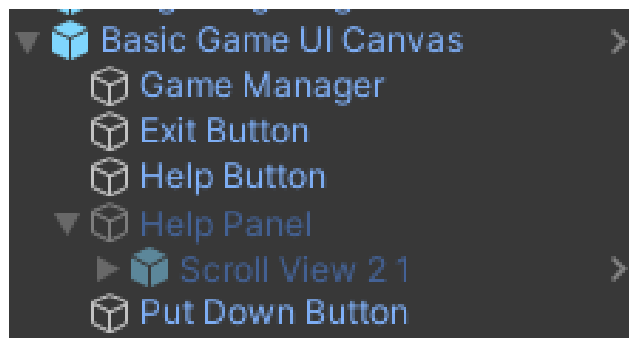
6.6 UI σκηνών των πειραμάτων

Όλες οι σκηνές διαθέτουν UI με κουμπιά για ορισμένες μεθόδους από το script `GameUIController` (σχήμα 6.8.), το οποίο βρίσκεται σε ένα κενό αντικείμενο με όνομα “Game Manager”. Το script χρειάζεται το πακέτο `using UnityEngine.SceneManagement;` και οι μέθοδοί του παρουσιάζονται παρακάτω:

- `void Start()`: κάνει τον προσανατολισμό αυτόματο.
- `void Update()`: περιμένει να πατηθεί το κουμπί πίσω για να πάει στο κεντρικό μενού.
- `public void BackToMainMenu()`: πηγαίνει στην μηδενική σκηνή, δηλαδή το κεντρικό μενού.
- `public void ManageHelpPanel(GameObject helpPanel)`: ανοίγει ή κλείνει το `helpPanel`.
- `public void RestartScene()`: κάνει επανεκκίνηση την σκηνή του πειράματος.

Στη συνέχεια πρέπει να δημιουργηθεί ένας καμβάς με τα εξής αντικείμενα (σχήμα 6.9):

- `Game Manager`: το αντικείμενο που αναφέρθηκε παραπάνω.
- `Exit Button`: κουμπί με την μέθοδο `BackToMainMenu` του script `GameUIController`, που βρίσκεται στο αντικείμενο `Game Manager`.
- `Help Panel`: panel με το `scrollview` του αντίστοιχου κεφαλαίου που δημιουργήθηκε στο κεφάλαιο 4
- `Help Button`: κουμπί με την μέθοδο `ManageHelpPanel` του script `GameUIController`, που βρίσκεται στο αντικείμενο `Game Manager`, με όρισμα το `Help Panel`.
- `Put Down Button`: κουμπί με την μέθοδο `PutDownButton` του script `PlayerController`, που βρίσκεται στο αντικείμενο `Player`.



Σχήμα 6.8: UI Canvas

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class GameUIController : MonoBehaviour
7  {
8      void Start()
9      {
10         Screen.orientation = ScreenOrientation.AutoRotation; //Sets the screen to be portret or landscape automaticly
11     }
12
13     void Update()
14     {
15         //if phone user tuch the back button
16         if (Input.GetKeyDown(KeyCode.Escape))
17         {
18             BackToMainMenu();
19         }
20     }
21
22     public void BackToMainMenu()
23     {
24         SceneManager.LoadScene(0, LoadSceneMode.Single);
25     }
26
27     public void ManageHelpPanel(GameObject helpPanel)
28     {
29         bool isActive = helpPanel.activeSelf;
30         if (!isActive)
31         {
32             Screen.orientation = ScreenOrientation.Portrait; //Sets the screen to be portret
33         }
34         else
35         {
36             Screen.orientation = ScreenOrientation.AutoRotation; //Sets the screen to be portret or landscape automaticly
37         }
38         helpPanel.SetActive(!isActive);
39     }
40
41     public void RestartScene()
42     {
43         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
44     }
45 }
46
47

```

Σχήμα 6.9: GameUIController Script

6.7 Επίλογος

Έχοντας κοινές, επαναχρησιμοποιούμενες λειτουργίες σε όλες τις σκηνές του παιχνιδιού, επιτυγχάνεται μια ομοιογένεια στην δομή του παιχνιδιού και στον τρόπο δημιουργίας του. Με όμοια χαρακτηριστικά σε κάθε σκηνή της εφαρμογής μειώνεται σημαντικά ο χρόνος κατασκευής της Π.Ε.

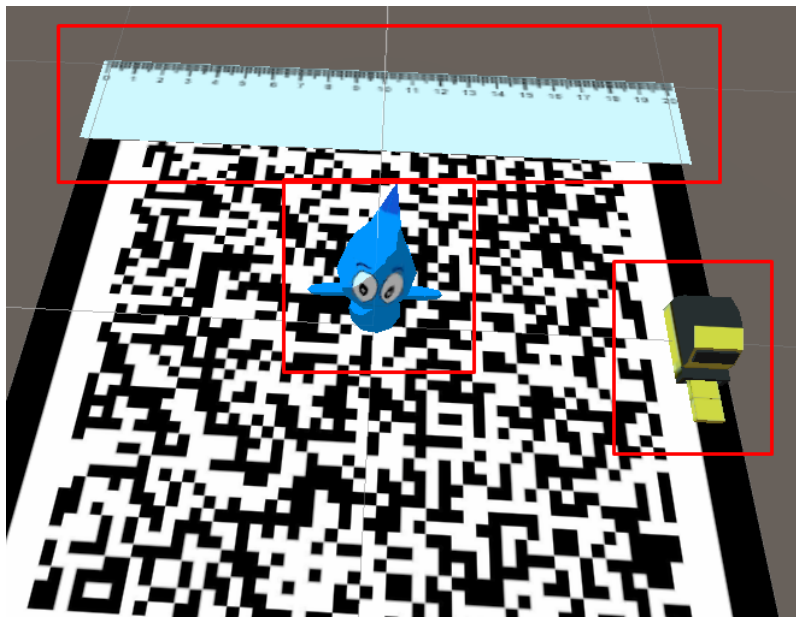
Κεφάλαιο 7ο: Πείραμα μέτρησης αποστάσεων, χρόνου και βάρους

7.1 Εισαγωγή πειραμάτων απόστασης

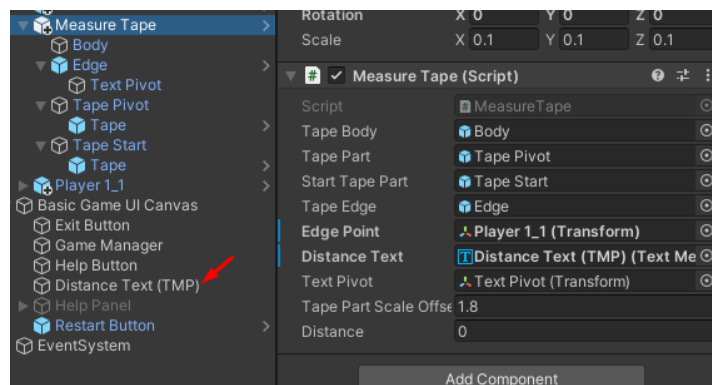
Το μοναδικό πείραμα που διαπραγματεύεται την μέτρηση της απόστασης είναι το πείραμα 1-1. Ο σκοπός του πειράματος είναι να διδάξει στους μαθητές τον τρόπο μέτρησης αντικειμένων με τη χρήση του χάρακα και της μετροταινίας. Η συγκεκριμένη σκηνή χρειάζεται ορισμένες αλλαγές στο κομμάτι της μετακίνησης του χαρακτήρα, καθώς δεν χρησιμοποιεί το πακέτο Navigation, αντ' αυτού αντικαθίσταται με το μικρό QR code έχοντας τον ρόλο του δείκτη του προορισμού.

7.2 Εικονικά αντικείμενα της σκηνής

Τα μοντέλα που είναι απαραίτητα για την υλοποίηση της σκηνής είναι, ο χαρακτήρας, ένας χάρακας και μία μετροταινία (σχήμα 7.1), τα οποία είναι αντικείμενα-παιδιά του Image Target Big, και ένα επίπεδο στο μέγεθος του Image Target Small, ως παιδί του δεύτερου. Το επίπεδο πρέπει να έχει για layer την επιλογή "Interactable".



Σχήμα 7.1: Αντικείμενα Πειράματος 1-1



Σχήμα 7.2: Αντικείμενο Measure Tape και Components του

7.3 Measure Tape

Η μετροταινία αποτελείται από το μοντέλο “Measure Tape”(σχήμα 7.2) και το script “MeasureTape” (σχήμα 7.3 και 7.4) που δημιουργείται και προστίθεται ως συστατικό του μοντέλου. Στο μοντέλο εφαρμόζονται οι κατάλληλες σκιάσεις (shaders) και χρωματισμοί (textures), όπως επίσης και μερικά κενά game objects που χρησιμεύουν ως άξονες, για τις μετρήσεις και τις αλλαγές που υλοποιεί το script. Αυτά είναι το “Text Pivot”, το “Tape Pivot” και το “Tape Start”, καθώς και η προσθήκη ενός Text-TMP με όνομα “Distance Text (TMP)” στο “Basic Game UI Canvas”. Το script χρειάζεται το πακέτο using TMPPro; και οι μέθοδοί του είναι οι:

- public void Update(): ελέγχει αν η μεζούρα είναι ενεργή στη σκηνή και καλεί τις μεθόδους GetDistance() και TextLookAtCamera().
- void GetDistance(): βρίσκει την απόσταση από την αρχή της μεζούρας μέχρι το τέλος της και την μετατρέπει σε εκατοστά.
- void TextLookAtCamera(): εμφανίζει στο UI την απόσταση.
- void ScaleMeasure(float newDistance): μεγεθύνει την ταινία ανάλογα με την απόσταση

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using TMPPro;
5
6 public class MeasureTape : MonoBehaviour
7 {
8     [SerializeField] GameObject tapeBody; //xriazete gia na dw an einai active to MeshRenderer, diladi an to blepei i kamera
9     [SerializeField] GameObject tapePart;
10    [SerializeField] GameObject startTapePart;
11    [SerializeField] GameObject tapeEdge;
12    [SerializeField] Transform edgePoint;
13
14    [SerializeField] TextMeshProUGUI distanceText;
15    [SerializeField] Transform textPivot;
16
17    [SerializeField] float tapePartScaleOffset;
18    [SerializeField] float distance;
19
20    public void Update()
21    {
22        if(edgePoint.gameObject.activeSelf)
23        {
24            gameObject.transform.LookAt(edgePoint);
25            tapeEdge.transform.position = edgePoint.position;
26            GetDistance();
27            TextLookAtCamera();
28        }
29    }
30
31    void GetDistance()
32    {
33        Vector2 edgePosition = new Vector2(tapeEdge.transform.position.x, tapeEdge.transform.position.z);
34        Vector2 startPosition = new Vector2(startTapePart.transform.position.x, startTapePart.transform.position.z);
35
36        distance = (edgePosition - startPosition).magnitude;
37        //distance = (distance-0.15f) * 100;
38        distance *= 100;
39        float newDistance = Mathf.Floor(distance) / 10f;
40        distanceText.text = newDistance + " cm";
41
42        ScaleMeasure(newDistance);
43    }
44

```

Σχήμα 7.3: Script MeasureTape πρώτο κομμάτι

```

45 void TextLookAtCamera()
46 {
47     if(tapeBody.GetComponent<MeshRenderer>().enabled)
48     {
49         Vector3 textPos = Camera.main.WorldToScreenPoint(textPivot.position);
50         distanceText.gameObject.SetActive(true);
51         distanceText.transform.position = textPos;
52     }
53     else
54     {
55         distanceText.gameObject.SetActive(false);
56     }
57 }
58
59
60 void ScaleMeasure(float newDistance)
61 {
62     //find center of start and edge of measure
63     Vector3 center = Vector3.Lerp(tapeEdge.transform.position, startTapePart.transform.position,0.5f);
64     tapePart.transform.position = center;
65
66     float newScale = tapePart.transform.localScale.x * (newDistance * tapePartScaleOffset);
67
68     tapePart.transform.localScale = new Vector3(1f, 1f, newScale);
69 }
70
71

```

Σχήμα 7.4: Script MeasureTape δεύτερο κομμάτι

7.4 Παραλλαγή χαρακτήρα 1-1

Στο πείραμα 1-1 ο χαρακτήρας δεν χρειάζεται το πακέτο Navigation, αλλά χρειάζεται έναν πιο απλό τρόπο μετακίνησης. Πρέπει να γίνει η αντικατάσταση του Component PlayerController με το script Player1_1 και το PlayerAnimationController με το PlayerAnimation1_1, καθώς και η αφαίρεση του Nav Mesh Agent (σχήμα 7.5). Το script Player1_1 (σχήμα 7.6) επιτυγχάνει την κίνηση του χαρακτήρα με το Transform του game object, τη μέθοδο Vector3.MoveTowards() και τη μέθοδο Quaternion.LookRotation().



Σχήμα 7.5: Components του αντικειμένου Player1_1

```

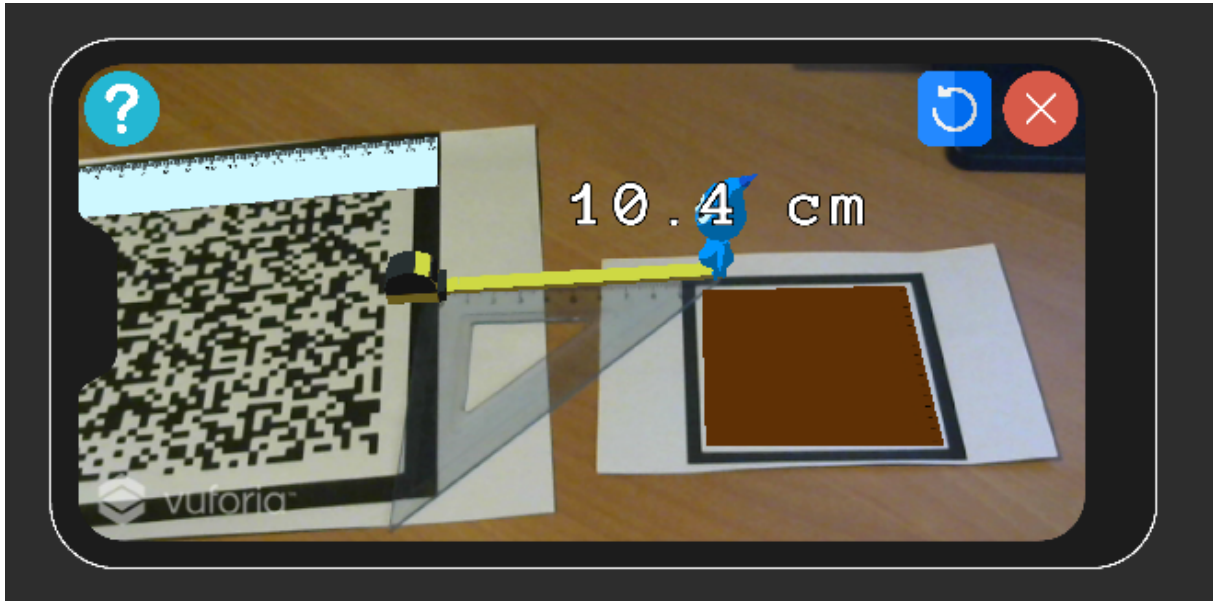
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Player1_1 : MonoBehaviour
6  {
7      [SerializeField] LayerMask whatIsInteractable;
8      [SerializeField] float speed = 0.2f;
9
10     public bool isMoving = false;
11     Vector3 hitPos;
12
13     void Update()
14     {
15         CheckIfTouch();
16     }
17
18     private void FixedUpdate()
19     {
20         if(isMoving)
21             TryMove();
22     }
23
24     void CheckIfTouch()
25     {
26         if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
27         {
28             Ray ray = Camera.main.ScreenPointToRay(Input.touches[0].position);
29             RaycastHit hit;
30             if (Physics.Raycast(ray, out hit, whatIsInteractable))
31             {
32                 Debug.Log(hit.collider.gameObject);
33                 MoveToHit(hit);
34             }
35         }
36     }
37
38     void MoveToHit(RaycastHit hit)
39     {
40         isMoving = true;
41         hitPos = hit.point;
42     }
43
44     void TryMove()
45     {
46         Vector3 targetPos = new Vector3(hitPos.x, transform.position.y, hitPos.z);
47         Vector3 relativePos = targetPos - transform.position;
48
49         transform.rotation = Quaternion.LookRotation(relativePos, Vector3.up);
50         transform.position = Vector3.MoveTowards(transform.position, targetPos, speed * Time.deltaTime);
51
52         if(transform.position == targetPos)
53         {
54             isMoving = false;
55         }
56     }
57

```

Σχήμα 7.6: Script Player1_1

7.5 Εκτέλεση της σκηνής

Ακουμπώντας με το δάχτυλο την οθόνη, στην περιοχή του καφέ τετραγώνου, ο χαρακτήρας κινείται στο συγκεκριμένο σημείο. Η σκηνή του πειράματος εμφανίζεται στο σχήμα 7.7.



Σχήμα 7.7: Εκτέλεση του Πειράματος 1-1

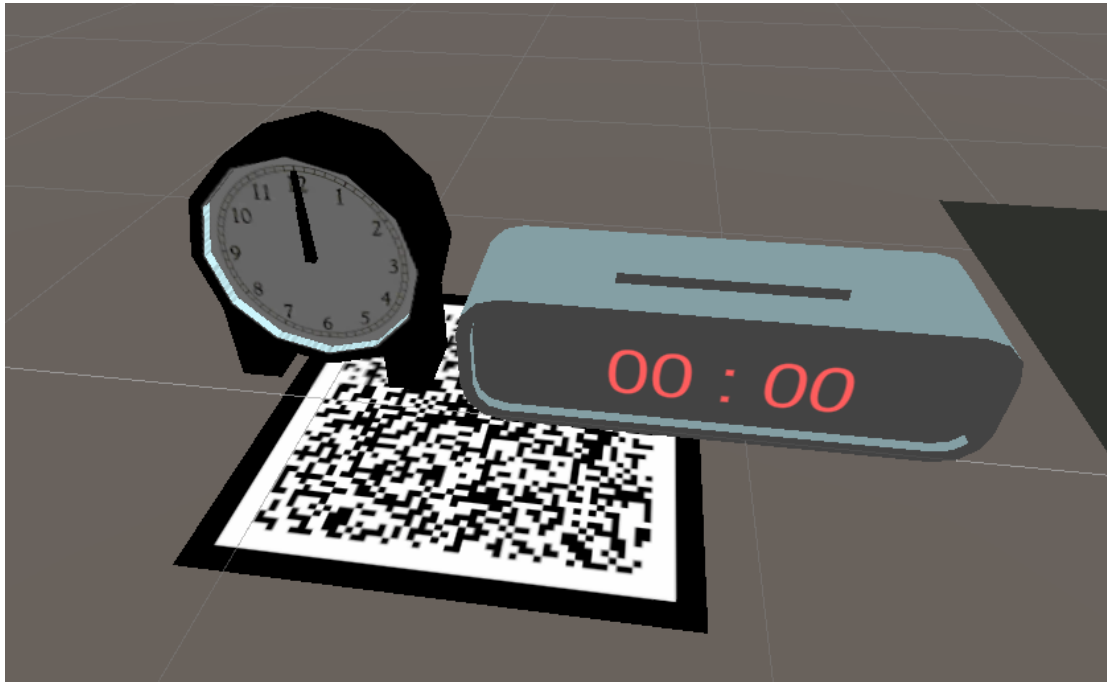
7.6 Εισαγωγή πειραμάτων χρόνου

Το δεύτερο κεφάλαιο του βιβλίου έχει δύο πειράματα που σχετίζονται με τη μέτρηση του χρόνου και τις διαφορές που έχει το ψηφιακό με το αναλογικό ρολόι. Τα δύο πειράματα διαθέτουν ένα αναλογικό και ένα ψηφιακό ρολόι, τα οποία εμφανίζονται στο μικρό QR code, ενώ στο μεγάλο QR code εμφανίζεται ο χαρακτήρας του παιχνιδιού με τα αντικείμενα του πειράματος που πρέπει να χρονομετρηθούν.

7.7 Ρολόγια

Το βασικότερο στοιχείο του κεφαλαίου είναι η μέτρηση του χρόνου και η διαφορά μεταξύ αναλογικού και ψηφιακού ρολογιού. Στην εφαρμογή δημιουργήθηκαν δύο μοντέλα (σχήμα 7.8) για αυτόν τον σκοπό, καθώς και scripts για τον έλεγχο τους, το “DigitalClock” και το “AnalogClock” που κληρονομεί στοιχεία από το πρώτο. Οι λειτουργίες τους είναι η επανεκκίνηση, η χρονομέτρηση, η παύση, η επανέναρξη και η εμφάνιση του χρόνου. Πιο αναλυτικά:

- void Update(): μετράει τον χρόνο κάθε καρέ.
- public void ResetTime(): κάνει επανεκκίνηση τον χρόνο.
- public void StopTime(): σταματάει τον χρόνο.
- public void StartTime(): αν ο χρόνος είναι σταματημένος τον κάνει επανεκκίνηση.
- protected virtual void CountTime(): χρησιμοποιείται διαφορετικά από τα δύο scripts, στο digital εμφανίζει σε ένα text(TMP) τον χρόνο και στο analog καλεί την μέθοδο RotatePointers().
- void RotatePointers(): περιστρέφει τους δείκτες του ρολογιού.
- protected void CalculateMinSec(): διαχωρίζει τα λεπτά από τα δευτερόλεπτα.



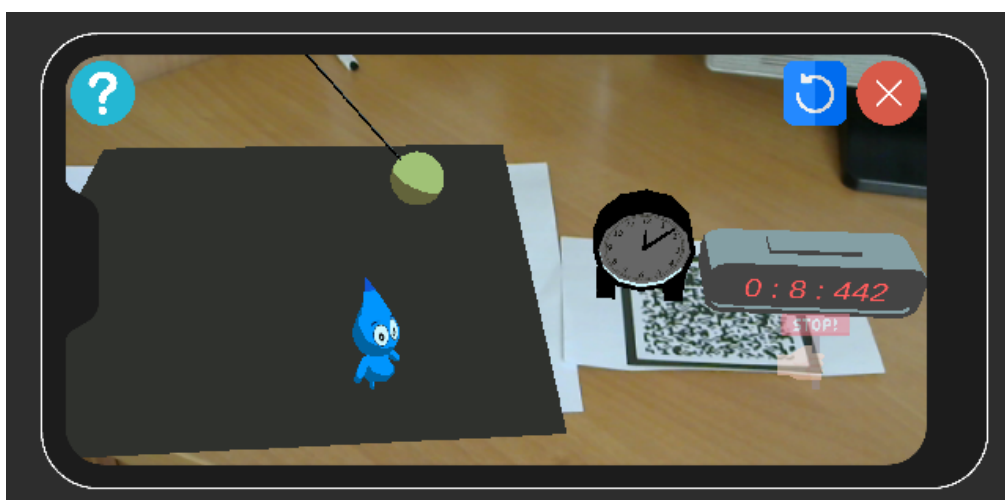
Σχήμα 7.8: Μοντέλα Ρολογιών

7.8 Πείραμα 2-1: μέτρηση χρόνου-εκκρεμές

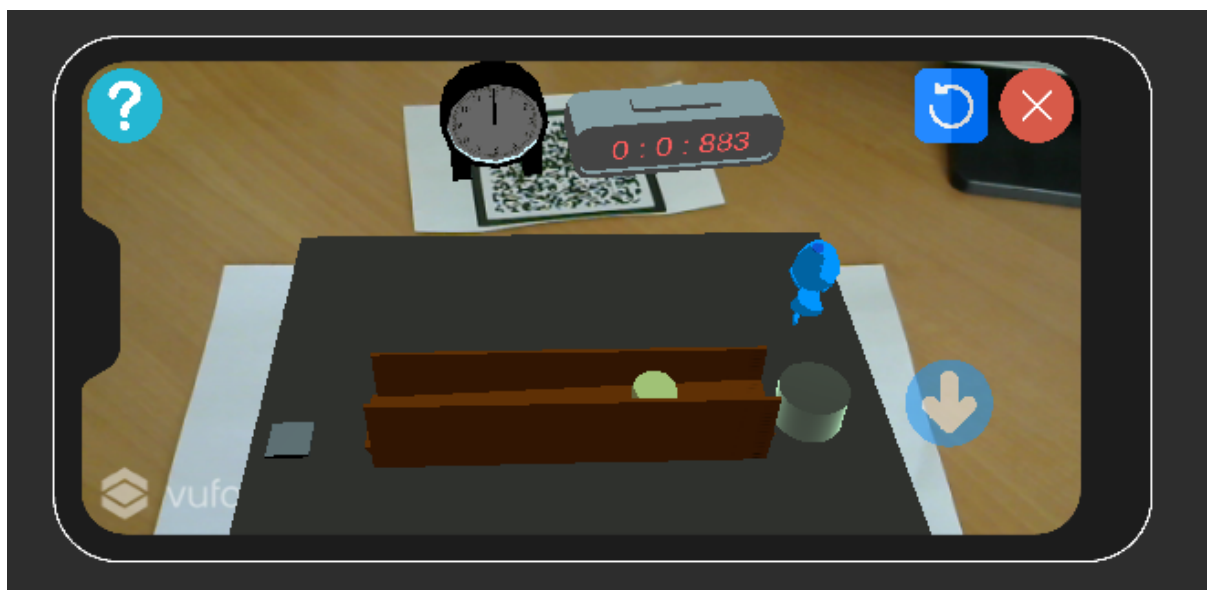
Στη σκηνή του πειράματος 2-1 (σχήμα 7.9) υπάρχει στο μεγάλο QR code ένα εκκρεμές το οποίο ταλαντώνεται όταν ο χαρακτήρας το ακουμπήσει, ξεκινώντας ταυτόχρονα τα ρολόγια. Στο UI υπάρχει το κουμπί STOP, με τη δυνατότητα να σταματάει το εκκρεμές και τον χρόνο των ρολογιών.

7.9 Πείραμα 2-2: μέτρηση χρόνου-κεκλιμένο επίπεδο

Στη σκηνή του πειράματος 2-2 (σχήμα 7.10) υπάρχει στο μεγάλο QR code μια σφαίρα, ένα κεκλιμένο επίπεδο και ένας χώρος αλληλεπίδρασης. Ο σκοπός είναι να μεταφέρει ο χρήστης την σφαίρα με την βοήθεια του χαρακτήρα στον Interactable χώρο, ξεκινώντας έτσι τα ρολόγια καθώς η σφαίρα κυλάει στο κεκλιμένο επίπεδο.



Σχήμα 7.9: Εκτέλεση του Πειράματος 2-1



Σχήμα 7.10: Εκτέλεση του Πειράματος 2-2

7.10 Εισαγωγή πειραμάτων βάρους

Το τρίτο κεφάλαιο ασχολείται με την μέτρηση του βάρους, χρησιμοποιώντας ζυγό, ελατήριο και σταθμά. Στην Π.Ε. υλοποιήθηκαν τρία πειράματα σε τρεις σκηνές χρησιμοποιώντας μόνο το μεγάλο QR code, στο οποίο υπάρχει ο χαρακτήρας, ορισμένα αντικείμενα για ζύγιση και τα όργανα για τον σκοπό αυτό. Κάθε αντικείμενο για ζύγιση, διαθέτει το script Weight το οποίο έχει μια μεταβλητή για τον έλεγχο και τη σύγκριση του βάρους του με την σκηνή.

7.11 Πείραμα 3-1: μέτρηση βάρους με ζυγό

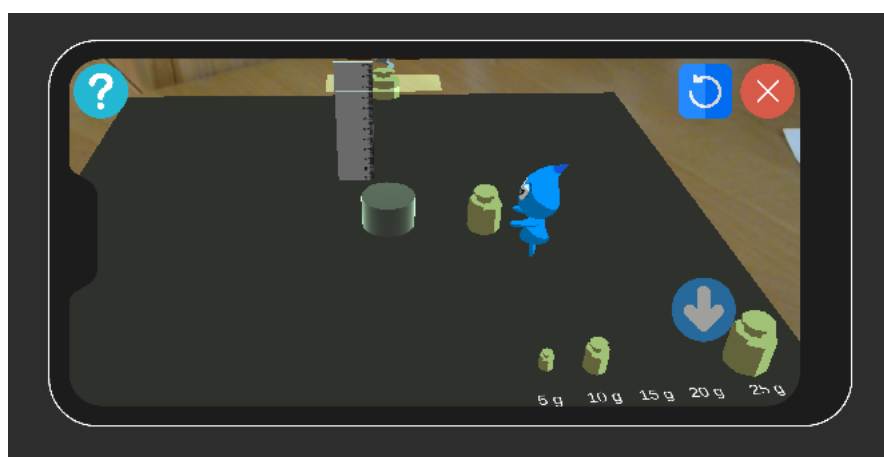
Στη σκηνή του πειράματος 3-1 υπάρχει ο ζυγός με δύο σημεία αλληλεπίδρασης, τέσσερις κατηγορίες σταθμών, βάρους 16, 24, 32 και 40 γραμμαρίων και μία σφαίρα άγνωστου βάρους για τους χρήστες. Σκοπός του πειράματος είναι οι μαθητές να λειτουργήσουν τον ζυγό και να καταφέρουν να μετρήσουμε το βάρος της σφαίρας, η οποία ζυγίζει 120 γραμμάρια. Κάθε πιάτο του ζυγού χωράει μέχρι τέσσερα αντικείμενα, τα οποία μεταφέρονται με τον χαρακτήρα από και προς τον χώρο αλληλεπίδρασης. Η κεφαλή του ζυγού τείνει προς το πιάτο με τα αντικείμενα με το μεγαλύτερο βάρος, με την χρήση του script Scale. Παράδειγμα της σκηνής υπάρχει στο σχήμα 7.11.

7.12 Πείραμα 3-2 και 3-3: μέτρηση βάρους με ελατήριο

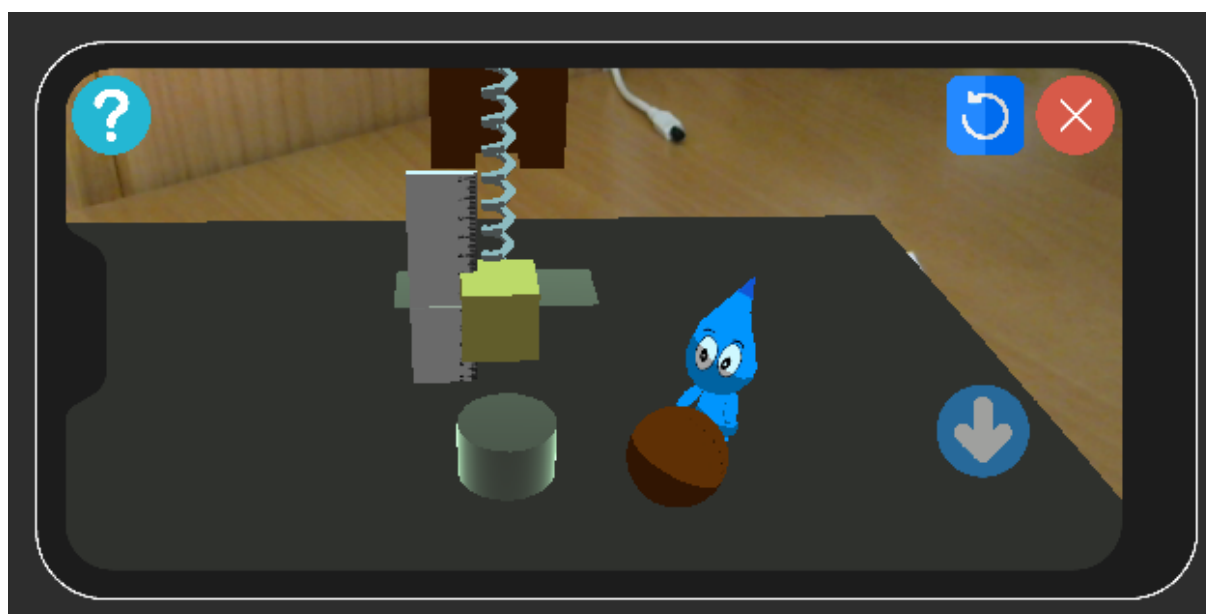
Στη σκηνή του πειράματος 3-2 υπάρχει το ελατήριο με έναν χάρακα και έναν χώρο αλληλεπίδρασης και πέντε σταθμά βάρους 5, 10, 15, 20 και 25 γραμμαρίων. Σκοπός του πειράματος είναι, να συνδυάσουν το βάρος των σταθμών με το μέγεθος την παραμόρφωσης του ελατηρίου, ελέγχοντάς το με τον χάρακα. Οι μαθητές, με τη βοήθεια του χαρακτήρα, πρέπει να μεταφέρουν τα σταθμά στον χώρο αλληλεπίδρασης και να παρατηρήσουν σε ποίο σημείο έχει φτάσει η άκρη του ελατηρίου. Τα αποτελέσματα των μετρήσεων είναι χρήσιμα στο πείραμα 3-3, καθώς η σκηνή είναι παρόμοια με του πειράματος 3-2, με τη διαφορά ότι, οι μαθητές δεν γνωρίζουν το βάρος των αντικειμένων που πρέπει να ζυγίσουν (42 ο κύβος και 55 η σφαίρα). Παραδείγματα των πειραμάτων υπάρχουν στα σχήμα 7.12 και 7.13.



Σχήμα 7.11: Εκτέλεση του Πειράματος 3-1



Σχήμα 7.12: Εκτέλεση του Πειράματος 3-2



Σχήμα 7.13: Εκτέλεση του Πειράματος 3-3

7.13 Επίλογος

Παραπάνω αναλύθηκαν συνοπτικά τα τρία πρώτα κεφάλαια του βιβλίου της φυσικής Α' Γυμνασίου, από τα συνολικά δώδεκα. Τα κεφάλαια αυτά αφορούν τη μέτρηση της απόστασης, του χρόνου και του βάρους και συνοδεύονται από αντίστοιχα πειράματα. Πιο συγκεκριμένα, το πρώτο κεφάλαιο περιλαμβάνει ένα πείραμα, το δεύτερο δύο και το τρίτο κεφάλαιο τρία πειράματα. Στα πλαίσια της Π.Ε. τα πειράματα αυτά υλοποιούνται με την χρήση του Unity.

Κεφάλαιο 8ο: Πειράματα μέτρησης θερμοκρασίας και ηλεκτρομαγνητισμού

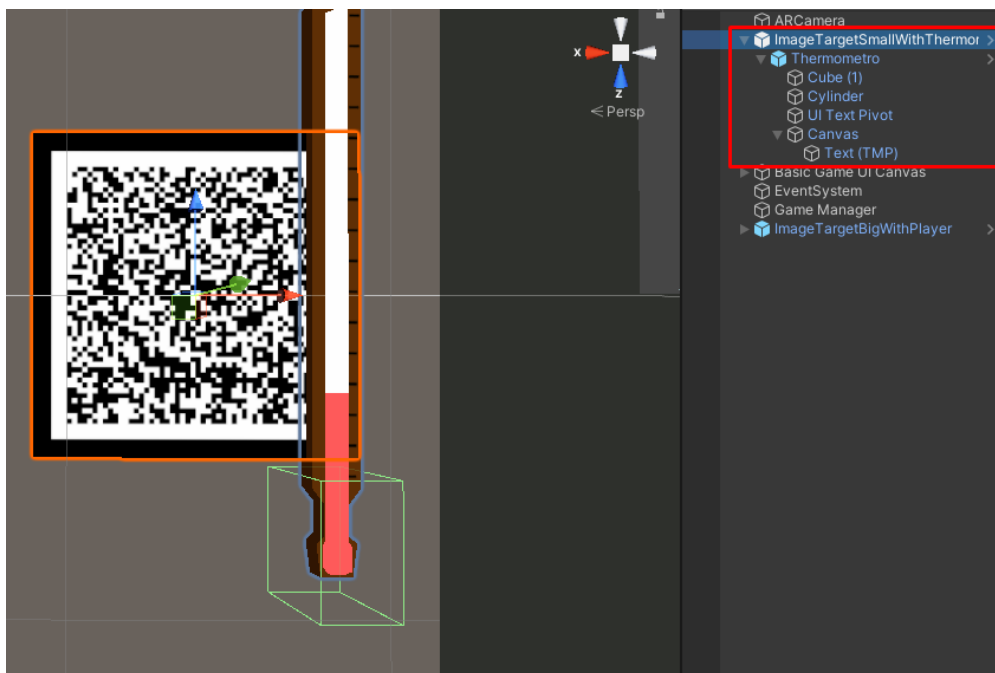
8.1 Εισαγωγή θερμότητας

Το μεγαλύτερο κομμάτι της Π.Ε. είναι η υλοποίηση των κεφαλαίων που σχετίζονται με την μέτρηση της θερμοκρασίας, καθώς έξι από τα δώδεκα κεφάλαια (κεφάλαια 4 μέχρι 9) του σχολικού βιβλίου της φυσικής, αφορούν αυτό το θέμα.

8.2 Θερμόμετρο

Στο μικρό QR code των σκηνών υπάρχει ως αντικείμενο παιδί ένα θερμόμετρο (σχήμα 8.1), το οποίο έχει ένα Box Collider στην βάση του και το script “Thermometer” (σχήμα 8.2 και 8.3). Το υγρό του θερμόμετρου έχει ως material το “Thermometro Ygro Material”, επιτρέποντας να αλλάζει δυναμικά η στάθμη του, δείχνοντας την σωστή θερμοκρασία. Οι μεταβλητές και οι μέθοδοι του script “Thermometer” παρουσιάζονται παρακάτω:

- public Renderer therMaterialRenderer: το υλικό του θερμόμετρου.
- [SerializeField] float currentTemperture = 50: τρέχουσα θερμοκρασία.
- [SerializeField] float materialTempertureHight: μεταβλητή για το ύψος του υγρού στο υλικό.
- public float maxTemperture = 130: μέγιστη θερμοκρασία.
- public float minTemperture = -40: μικρότερη θερμοκρασία.
- float tempertureRange: η διαφορά maxTemperture - minTemperture.
- [SerializeField] bool useUiText = true: boolean για το έλεγχο χρήσης του UI.
- [SerializeField] TextMeshProUGUI tempertureText: αντικείμενο κειμένου για την εμφάνιση της θερμοκρασίας στο UI.
- [SerializeField] Transform textPivot: θέση του tempertureText.



Σχήμα 8.1: Μοντέλο Θερμομέτρου στο Μικρό QR code

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using TMPro;
5
6  public class Thermometer : MonoBehaviour
7  {
8      [Header("Thermometer Variables")]
9      public Renderer therMaterialRenderer;
10     [SerializeField] float currentTemperature = 50;
11     [SerializeField] float materialTemperatureHight;
12     public float maxTemperature = 130;
13     public float minTemperature = -40;
14     float temperatureRange;
15
16     [Header("UI text")]
17     [SerializeField] bool useUiText = true;
18     [SerializeField] TextMeshProUGUI temperatureText;
19     [SerializeField] Transform textPivot;
20
21     private void Start()
22     {
23         temperatureRange = maxTemperature - minTemperature;
24         TemperatureToMaterialTemp();
25         if(!useUiText)
26         {
27             temperatureText.text = "";
28         }
29     }
30
31     void Update()
32     {
33         if(useUiText)
34         {
35             MoveUIText();
36         }
37     }
38
39     void MoveUIText()
40     {
41         Vector3 textPos = Camera.main.WorldToScreenPoint(textPivot.position);
42         temperatureText.transform.position = textPos;
43         temperatureText.text = currentTemperature.ToString("F1") + " C"; // to ToString("F1") epistrefei ton arithmo me 1 dekadiko psifio
44     }
45
46     public void SetCurrentTemperature(float newTemp)
47     {
48         currentTemperature = newTemp;
49         TemperatureToMaterialTemp();
50     }
51

```

Σχήμα 8.2: Script: Thermometer. Ένα από τα Δύο

```

52     void TemperatureToMaterialTemp()
53     {
54         materialTemperatureHight = Mathf.InverseLerp(minTemperature, maxTemperature, currentTemperature);
55         //materialTemperatureHight = (currentTemperature - minTemperature) / temperatureRange;
56         ChangeMaterialThermometerTemperature();
57     }
58
59     void ChangeMaterialThermometerTemperature()
60     {
61         therMaterialRenderer.material.SetFloat("_Hight", materialTemperatureHight);
62     }
63
64     private void OnTriggerStay(Collider other)
65     {
66         TemperatureController otherTemp;
67         if (otherTemp = other.gameObject.GetComponent<TemperatureController>())
68         {
69             SetCurrentTemperature(otherTemp.temp);
70         }
71     }
72
73
74
75

```

Σχήμα 8.3: Script: Thermometer. Δύο από τα Δύο

- private void Start(): αρχικοποίηση μεταβλητών.
- void Update(): καλεί την μέθοδο MoveUIText().
- void MoveUIText(): μετακινεί το UI text ανάλογα με την θέση του θερμόμετρου.
- public void SetCurrentTemperture(float newTemp): κάνει το currentTemperture ίσο με το newTemp και καλεί το TempertureToMaterialTemp().
- void TempertureToMaterialTemp(): ρυθμίζει το υλικό.
- void ChangeMaterialThermometerTemperture(): αλλάζει το ύψος του υγρού του υλικού.
- private void OnTriggerStay(Collider other): Αν η βάση του θερμόμετρου ακουμπάει κάποιο αντικείμενο με το script TempertureController καλεί την μέθοδο SetCurrentTemperture(otherTemp.temp).

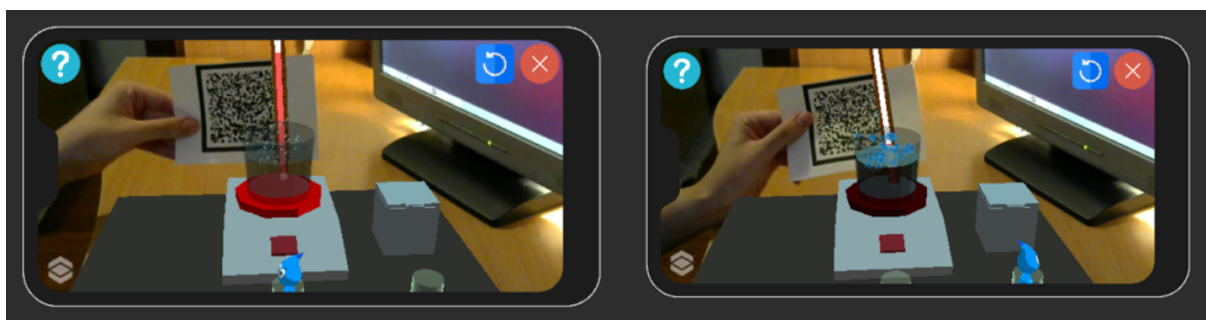
Τα αντικείμενα που πρέπει να θερμομετρηθούν έχουν το script TempertureController, το οποίο έχει μια μεταβλητή float για την θερμοκρασία του.

8.3 Πείραμα 4-1: Μέτρηση θερμοκρασίας νερού με κενό θερμόμετρο

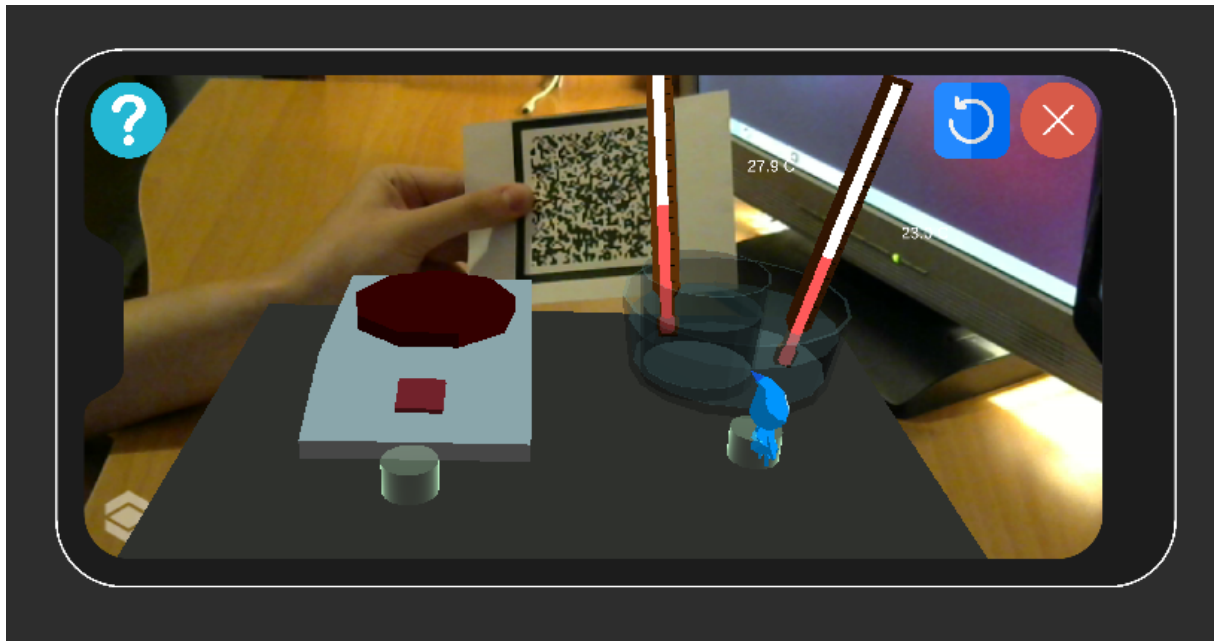
Το πείραμα ανήκει στο 4ο κεφάλαιο, το οποίο σχετίζεται με την μέτρηση της θερμοκρασίας. Σκοπός του πειράματος είναι, η τοποθέτηση του θερμομέτρου από τους μαθητές μέσα στο ποτήρι με το νερό κουνώντας το μικρό QR code με τα χέρια τους. Τα δύο σημεία αλληλεπίδρασης επιτρέπουν τον χαρακτήρα να ανοιγοκλείνει την εστία θερμότητας και το άλλο να ψυχραίνει το νερό προσθέτοντας πάγο σε αυτό. Στο συγκεκριμένο πείραμα, η μεταβλητή “Use UI Text” στο αντικείμενο του θερμομέτρου, πρέπει να είναι false, γιατί ο σκοπός του πειράματος είναι να καταλάβουν οι μαθητές σε ποιο σημείο του θερμομέτρου φτάνει το υγρό όταν η θερμοκρασία του είναι 100 και σε ποιο σημείο όταν είναι 0. Παράδειγμα του πειράματος υπάρχει στην σχήμα 8.4.

8.4 Πείραμα 5-1: Εξισορρόπηση θερμοκρασίας

Το πείραμα 5-1 έχει ως σκοπό την κατανόηση της εξισορρόπησης της θερμοκρασίας δύο υγρών, δηλαδή την αλλαγή της θερμοκρασίας ενός θερμού υγρού όταν έρχεται σε επαφή με ένα άλλο, χαμηλότερης θερμοκρασίας. Οι μαθητές πρέπει αρχικά να θερμάνουν το νερό που βρίσκεται στην εστία θέρμανσης, μετακινώντας τον χαρακτήρα στον αντίστοιχο χώρο αλληλεπίδρασης, μετρώντας την θερμοκρασία του με το θερμόμετρο και στη συνέχεια να μεταφέρουν το δοχείο από την εστία προς το μεγάλο δοχείο, που περιέχει νερό σε θερμοκρασία δωματίου και ένα ακίνητο θερμόμετρο. Οι μαθητές πρέπει να συμπεράνουν ότι κατά την επαφή τους τα δύο υγρά προσπαθούν να ισορροπήσουν τις θερμοκρασίες τους, ανεβάζοντας θερμοκρασία το ψυχρότερο από τα δύο και χάνοντας θερμοκρασία το πιο θερμό. Παράδειγμα του πειράματος υπάρχει στην σχήμα 8.5.



Σχήμα 8.4: Παράδειγμα Πειράματος 4-1 Αριστερά: Το Νερό Βράζει. Δεξιά: Το Νερό Ψύχεται



Σχήμα 8.5: Παράδειγμα Πειράματος 5-1. Εξισορρόπηση Θερμοκρασίας

8.5 Πείραμα 6-1: Μέτρηση θερμοκρασίας νερού με θερμόμετρο

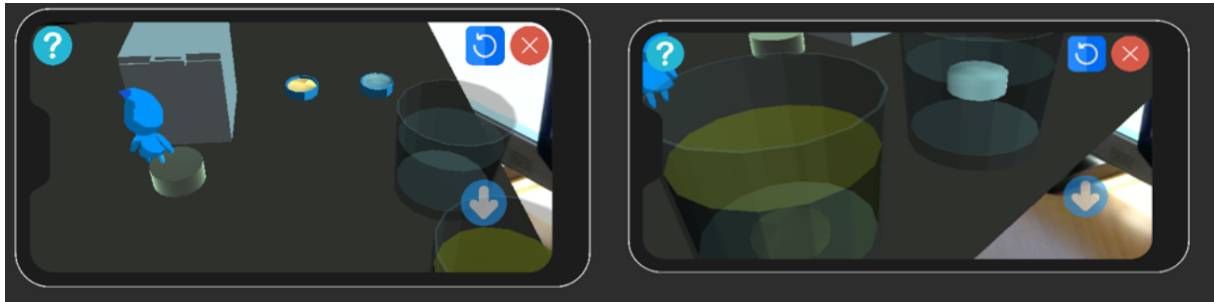
Πανομοιότυπο με το πείραμα 4-1 είναι και το πείραμα 6-1. Η δομή του είναι η ίδια με το προηγούμενο, με τη διαφορά ότι το θερμόμετρο του πειράματος 6-1, έχει την μεταβλητή “Use UI Text” = true, με σκοπό να εμφανίζεται η ακριβής θερμοκρασία του θερμομέτρου στο UI.

8.6 Πείραμα 7-1: Πήξη υγρών σε καπάκια

Σκοπός του πειράματος είναι καταλάβουν οι μαθητές την διαδικασία της πήξης των υγρών και να δουν την αντίδραση του νερού σε αυτό το φαινόμενο, σε αντίθεση με άλλα υγρά όπως το λάδι. Στην σκηνή δεν υπάρχει το μικρό QR code, λόγω του ότι δεν είναι απαραίτητη η χρήση του θερμομέτρου σε αυτό το πείραμα. Στο μεγάλο QR code υπάρχουν τα κατάλληλα αντικείμενα για την εκτέλεση του πειράματος, δηλαδή δύο καπάκια με νερό και λάδι, δύο ποτήρια με νερό και λάδι, ο χαρακτήρας και ένας καταψύκτης. Οι μαθητές πρέπει με τη βοήθεια του χαρακτήρα να μετακινήσουν τα καπάκια μέσα στον καταψύκτη και πατώντας ξανά στον χώρο αλληλεπίδρασης, εμφανίζονται παγωμένα πλέον τα δύο καπάκια. Στη συνέχεια, μεταφέροντας τα παγάκια στα αντίστοιχα ποτήρια με το σωστό υγρό, θα παρατηρήσουν ότι το παγάκι του νερού επιπλέει στην επιφάνεια του ποτηριού με το νερό, ενώ το παγάκι λαδιού βυθίζεται, λόγω της διαφοράς στην πυκνότητα. Παράδειγμα του πειράματος υπάρχει στο σχήμα 8.6.

8.7 Πείραμα 7-2: Πήξη υγρών σε μπουκάλια

Αντίστοιχο με το προηγούμενο πείραμα είναι το πείραμα 7-2, έχοντας ως βασικό στοιχείο την πήξη του νερού. Στην σκηνή δεν υπάρχει το μικρό QR code με το θερμόμετρο, ενώ στο μεγάλο QR code υπάρχει ο χαρακτήρας, ένας καταψύκτης και δύο γυάλινα μπουκάλια, το ένα με νερό και το άλλο με λάδι. Όταν οι μαθητές μεταφέρουν τα δύο μπουκάλια στον καταψύκτη, με την βοήθεια του χαρακτήρα, θα πρέπει με ένα τρίτο πάτημα του interactable να εμφανιστούν τα δύο μπουκάλια έξω από τον καταψύκτη με παγωμένο το περιεχόμενό τους. Σε αντίθεση με την πραγματική ζωή, αυτή η διαδικασία θα έπαιρνε αρκετές ώρες μέχρι να γίνει ορατό το αποτέλεσμα της πήξης των υγρών. Το

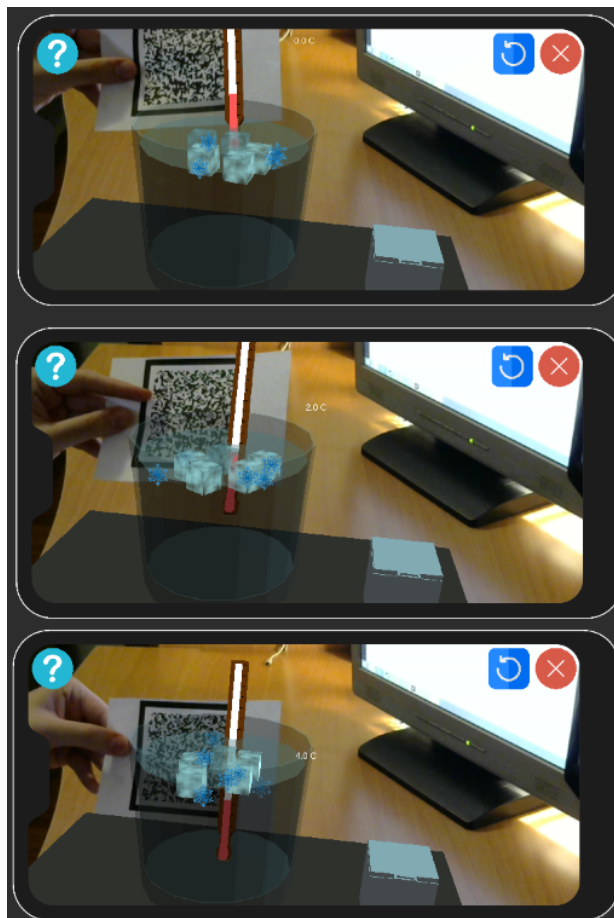


Σχήμα 8.6: Παράδειγμα Πειράματος 7-1. Αριστερά: Τα Καπάκια Είναι Παγωμένα. Δεξιά: Τα Παγάκια Είναι Μέσα στα Δοχεία με τα Υγρά

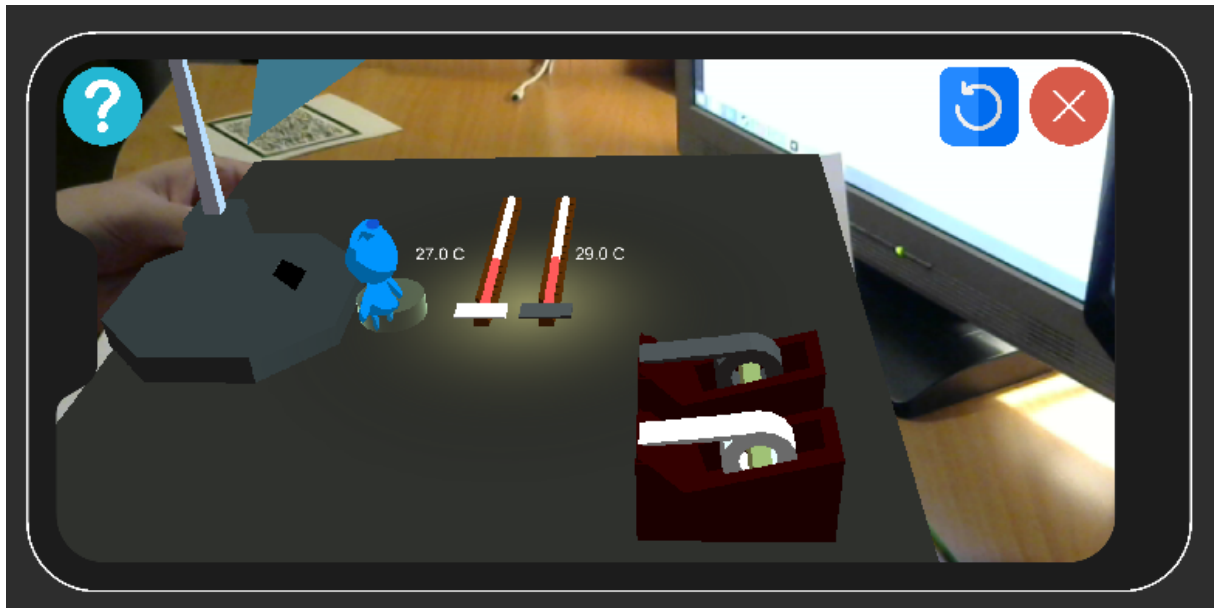
αποτέλεσμα που θα παρατηρήσουν οι μαθητές είναι ότι το γυάλινο μπουκάλι που περιείχε το νερό, είναι πλέον σπασμένο.

8.8 Πείραμα 7-3: Μέτρηση θερμοκρασίας σε διαφορετικά επίπεδα

Στο πείραμα 7-3 οι μαθητές πρέπει να θερμομετρήσουν το νερό μέσα στο δοχείο με το θερμόμετρο που διαθέτουν στο μικρό QR code. Με την βοήθεια του χαρακτήρα οι μαθητές πρέπει να γεμίσουν με παγάκια το δοχείο με το νερό και έπειτα, με το θερμόμετρο, να παρατηρήσουν τις διαφορές της θερμοκρασίας στην επιφάνεια του νερού, στον πάτο του δοχείου και ενδιάμεσα. Παράδειγμα του πειράματος υπάρχει στο σχήμα 8.7.



Σχήμα 8.7: Παράδειγμα Πειράματος 7-3. Πάνω: Η Θερμοκρασία είναι 0. Μέση: Η Θερμοκρασία είναι 2. Κάτω: Η Θερμοκρασία είναι 4



Σχήμα 8.8: Παράδειγμα Πειράματος 8-2. Φως και Θερμοκρασία

8.9 Πείραμα 8-1: Αλλαγή θερμοκρασίας λόγω φωτός

Η υλοποίηση του πειράματος 8-1 έγινε με μία πολύ απλή διαδικασία, καθώς δεν είχε μεγάλες απαιτήσεις. Ο σκοπός είναι να καταλάβουν οι μαθητές ότι ανάβοντας μία λάμπα πυρακτώσεως κοντά σε ένα θερμόμετρο, η θερμοκρασία του θα αυξηθεί. Στην σκηνή δεν γίνεται η χρήση του μικρού QR code, καθώς το θερμόμετρο βρίσκεται στατικό στο έδαφος, περιμένοντας τον λαμπτήρα να ανάψει. Το φως του λαμπτήρα δεν είναι στην πραγματικότητα κάποιο αντικείμενο με το συστατικό: Light, αλλά είναι μια επίπεδη επιφάνεια με διάφανο υλικό, που δίνει την ψευδαίσθηση του αναμμένου φωτός.

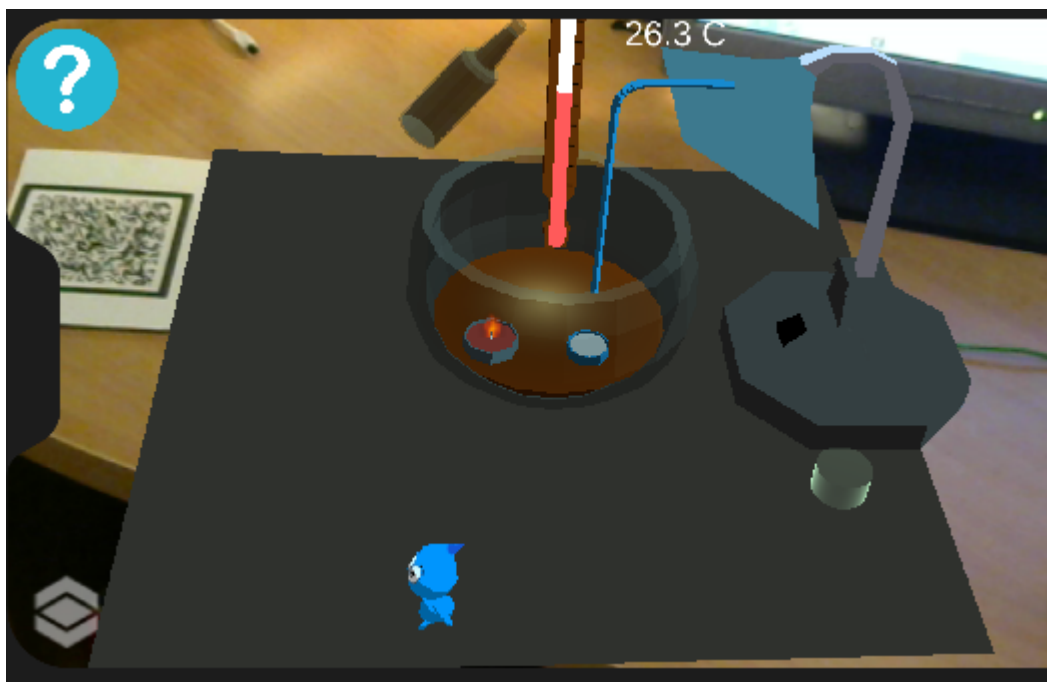
8.10 Πείραμα 8-2: Αλλαγή θερμοκρασίας λόγω σκουρόχρωμης επιφάνειας

Παρόμοιο με το προηγούμενο πείραμα είναι το πείραμα 8-2, έχοντας την ίδια λογική με το θερμόμετρο και τον λαμπτήρα. Τα επιπλέον αντικείμενα της σκηνής είναι δύο χαρτοταινίες, μια λευκή και μία μαύρη, και ένα θερμόμετρο, αυξάνοντας τον αριθμό των θερμομέτρων κατά ένα. Σκοπός του πειράματος είναι να κατανοήσουν οι μαθητές το γεγονός ότι το μαύρο χρώμα απορροφά πιο εύκολα το φως, μετατρέποντάς το σε θερμότητα. Παράδειγμα του πειράματος υπάρχει στο σχήμα 8.8.

8.11 Πείραμα 9-1: Φαινόμενο του θερμοκηπίου

Το πείραμα 9-1 είναι ο συνδυασμός τριών πειραμάτων του βιβλίου, του 9-1, του 9-2 και του 9-3, λόγω του ότι τα πειράματα πρέπει να γίνουν διαδοχικά το ένα μετά το άλλο. Οι μαθητές, με το παρόν πείραμα, πρέπει να κατανοήσουν την αύξηση της θερμοκρασίας που προκαλεί το διοξείδιο του άνθρακα, δημιουργώντας ένα μικρό περιβάλλον με αυτόν τον σκοπό. Με βάση το σχολικό βιβλίο, το περιβάλλον που πρέπει να δημιουργηθεί, διαθέτει μία σφαιρική γυάλινη φιάλη με μικρή ποσότητα χρώματος και ένα θερμόμετρο. Το πρώτο στάδιο του πειράματος καλεί τους μαθητές να ανάψουν τον λαμπτήρα πυρακτώσεως, παρατηρώντας ότι η θερμοκρασία ανεβαίνει όπως στα προηγούμενα πειράματα (πειράμα 8-1 και πείραμα 8-2). Το δεύτερο στάδιο του πειράματος είναι να τοποθετήσουν μέσα στην φιάλη λίγη ποσότητα ανθρακούχου νερού, ένα καπάκι μαγειρικής σόδας, και προσεκτικά με ένα καλαμάκι να ρίξουν μερικές σταγόνες ξυδιού στο καπάκι, με σκοπό να ανέβει και άλλο η θερμοκρασία. Φυσικά, το τελευταίο βήμα γίνεται αυτόματα με την χρήση των animations. Στο τρίτο

και τελευταίο στάδιο οι μαθητές, με την βοήθεια του χαρακτήρα, πρέπει να τοποθετήσουν μέσα στην γυάλινη σφαίρα ένα αναμμένο κεράκι, για να ανεβάσει μερικούς βαθμούς κελσίου το θερμομέτρο. Τα βήματα του πειράματος υπάρχουν ως παράδειγμα στο σχήμα 8.9.



Σχήμα 8.9: Παράδειγμα Πειράματος 9-1. Φαινόμενο του Θερμοκηπίου

8.12 Εισαγωγή ηλεκτρομαγνητισμού

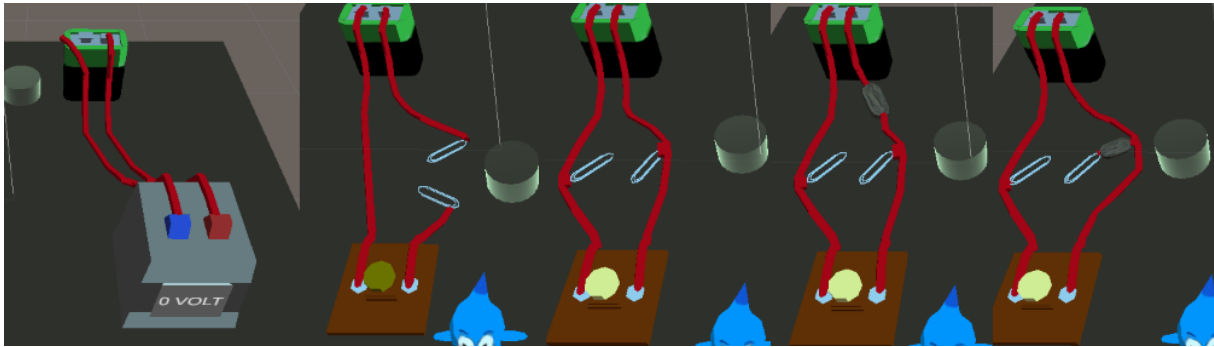
Σε αυτό το κεφάλαιο παρουσιάζονται τα τελευταία πειράματα του σχολικού βιβλίου της φυσικής, που σχετίζονται με τον ηλεκτρισμό και τον ηλεκτρομαγνητισμό. Η πρώτη κατηγορία πειραμάτων έχει ως σκοπό, να μάθει στους μαθητές τον τρόπο λειτουργίας ενός κυκλώματος. Η δεύτερη κατηγορία σχετίζεται με τον ηλεκτρομαγνητισμό, την μετατροπή ηλεκτρισμού σε μαγνητισμό και το αντίστροφο. Τα παρακάτω πειράματα χρησιμοποιούν μόνο το μεγάλο QR code, καθώς το μικρό δεν είναι απαραίτητο για την υλοποίησή τους.

8.13 Πειράματα του κεφαλαίου 10 του βιβλίου

Τα πέντε πειράματα του κεφαλαίου 10 έχουν σκοπό να διδάξουν στους μαθητές την διαφορά ενός κλειστού και ενός ανοιχτού κυκλώματος, χρησιμοποιώντας τον χαρακτήρα του παιχνιδιού για να συνδέει και να αποσυνδέει τα καλώδια των κυκλωμάτων. Στο πείραμα 10-1 τα αντικείμενα που υπάρχουν στην σκηνή είναι μια μπαταρία συνδεδεμένη σε ένα βολτόμετρο. Στα πειράματα 10-2 και 10-3 εμφανίζονται στην σκηνή μια μπαταρία συνδεδεμένη με καλώδια σε ένα λαμπάκι, επιπλέον υπάρχουν συνδετήρες για να ανοιγοκλείνουν το κύκλωμα σε διαφορετικά σημεία, ξεχωρίζοντας τα δύο πειράματα. Τα πειράματα 10-4 και 10-5 είναι παρόμοια με τα δύο προηγούμενά τους, έχοντας την διαφορά ότι τα κυκλώματα διακόπτονται από συρματένια κουβάρια, τα οποία θερμαίνονται και σπάνε όταν τα κυκλώματα είναι κλειστά, αφήνοντάς τα σε άλλη μορφή. Παραδείγματα των πειραμάτων υπάρχουν στο σχήμα 8.10.

8.14 Πείραμα 11-1: Από τον ηλεκτρισμό στον ηλεκτρομαγνητισμό

Σκοπός του πειράματος είναι να καταλάβουν οι μαθητές της πρώτης γυμνασίου, τον τρόπο μετατροπής του ηλεκτρισμού σε ηλεκτρομαγνητισμό και τελικά να παρατηρήσουν ότι μπορεί να δημιουργηθεί κίνηση. Στην σκηνή υπάρχει ο χαρακτήρας με όλα τα υλικά του πειράματος. Όταν ο χρήστης θα ακουμπήσει όλα τα υλικά, το πηνίο θα αρχίσει να περιστρέφεται στον άξονά του λόγω του μαγνήτη και του ηλεκτρικού ρεύματος που το διαπερνά. Παράδειγμα του πειράματος υπάρχει στην σήμα 8.10.



Σχήμα 8.10: Πειράματα Κεφαλαίου 10



Σχήμα 8.11: Πειράματα Κεφαλαίου 11

8.15 Πείραμα 12-1: Από τον ηλεκτρομαγνητισμό στον ηλεκτρισμό

Το τελευταίο πείραμα του βιβλίου αφορά την μετατροπή της κινητικής ενέργειας σε ηλεκτρική, με την βοήθεια του ηλεκτρομαγνητισμού. Στην σκηνή δεν υπάρχει ο χαρακτήρας του παιχνιδιού. Αντ' αυτού παίρνει την θέση του το αντικείμενο του πειράματος, δηλαδή μια κατασκευή με δύο μαγνήτες, ένα led λαμπάκι, ένα πηνίο που περιβάλλει ένα χαρτόνι, για να συγκρατεί τους μαγνήτες και ένα καρφί, όπως φαίνεται στο σχήμα 8.11. Όπως θα έπρεπε να κάνουν στην πραγματικότητα οι μαθητές, είναι απαραίτητη η περιστροφή του καρφιού στον άξονα του, για να δημιουργηθεί ρεύμα λόγω του ηλεκτρομαγνητισμού, με σκοπό να ανάψει το led λαμπάκι. Όσο πιο γρήγορη είναι η περιστροφή του καρφιού, τόσο πιο πολύ θα ανάψει το λαμπάκι του πειράματος. Η υλοποίηση αυτής της διαδικασίας γίνεται στο script SpinWithTouch, υπολογίζοντας κάθε καρέ την θέση του δακτύλου στον οθόνη και συγκρίνοντας την απόσταση που έχει με την θέση του δακτύλου που είχε στο προηγούμενο καρέ (σχήμα 8.12).

```

42 void CheckTouching()
43 {
44     if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Moved)
45     {
46         isTouching = true;
47         touchPosition = Input.touches[0].position;
48
49         rotateSpeed = (prevTouchPosition - touchPosition).magnitude;
50         TurnOnLamp();
51
52         prevTouchPosition = touchPosition;
53
54         objPos = Camera.main.WorldToScreenPoint(transform.position);
55         touchRotation = (touchPosition - objPos).normalized;
56     }
57 }
58
59 void TurnOnLamp()
60 {
61     if (rotateSpeed <= 0)
62     {
63
64     }
65     else if (rotateSpeed > 200)
66     {
67         lamp.material = lampOnFull;
68     }
69     else
70     {
71         lamp.material = lampOn2;
72     }
73 }
74

```

Σχήμα 8.12: Μέθοδοι CheckTouching και TurnOnLamp του Script SpinWithTouch

8.16 Επίλογος

Παραπάνω αναλύθηκαν τα πειράματα του τέταρτου έως και του δωδέκατου κεφαλαίου του βιβλίου της φυσικής Α' Γυμνασίου. Από το κεφάλαιο τέσσερα μέχρι το εννιά παρουσιάστηκαν τα πειράματα που σχετίζονται με την μέτρηση της θερμοκρασίας, καθώς και ο τρόπος υλοποίησης του αντικειμένου “Θερμόμετρο”. Τα τρία τελευταία κεφάλαια του βιβλίου, αφορούν πειράματα που σχετίζονται με τον ηλεκτρισμό και τον ηλεκτρομαγνητισμό και την διαδικασία δημιουργίας των σκηνών, με το περιβάλλον Unity.

Κεφάλαιο 9ο: Τελικά συμπεράσματα και βελτιώσεις της Π.Ε.

Τα τελικά συμπεράσματα που λήφθηκαν με το πέρας της Π.Ε. και της δημιουργίας της εφαρμογής είναι ότι:

- Ο τρόπος διδασκαλίας μπορεί να γίνει πιο εύκολα και πιο αποτελεσματικά από τους διδάσκοντες, δίνοντας στους μαθητές τα εργαλεία που χρειάζονται για να εκτελέσουν τα πειράματα του βιβλίου χωρίς να διαθέτουν τα εργαλεία και τον εξοπλισμό με φυσικό τρόπο. Αντ' αυτού οι διδάσκοντες είναι απαραίτητο να διαθέτουν μόνο την εφαρμογή σε μία ή περισσότερες συσκευές android, τις οποίες η πλειοψηφία των μαθητών διαθέτουν, και δύο τυπωμένες εικόνες σε χαρτί που θα έχουν τον ρόλο των target images.
- Η διδασκαλία γίνεται πιο ευχάριστη για τους μαθητές. Είναι πιο εύκολο να προσηλωθεί ένα παιδί ηλικίας 12 με 13 ετών σε μία οθόνη κινητού, θυμίζοντας την καθημερινότητά του, από το να ακούει θεωρητικά τα πειράματα χωρίς να έχει την κατάλληλη οπτικοακουστική ανταπόκριση. Το ερέθισμα που λαμβάνει ο μαθητής εκτελώντας και ολοκληρώνοντας το παιχνίδι, του δίνουν μια αίσθηση ικανοποίησης ότι κατάφερε να κάνει σωστά το πείραμα δίνοντάς του ταυτόχρονα και μία καλύτερη δυνατότητα απομνημόνευσης του πειράματος.
- Η δημιουργία ενός παιχνιδιού με τα κατάλληλα εργαλεία όπως το Unity, μπορεί να γίνει πολύ εύκολα έχοντας ένα σωστό υπόβαθρο. Μπορούν να δημιουργηθούν παιχνίδια, όχι μόνο για διασκέδαση, αλλά και για μάθηση με την χρήση του AR και άλλων τεχνολογιών του τομέα του XR.

Η εφαρμογή βρίσκεται σε μία έκδοση που μπορεί να χρησιμοποιηθεί από μαθητές και εκπαιδευτικούς, αλλά υπάρχουν πολλές βελτιώσεις που μπορούν να γίνουν για να είναι πιο εύχρηστο και πιο αποτελεσματικό στην εκμάθηση πειραμάτων. Τρόποι βελτίωσης θα μπορούσαν να είναι:

- Πιο φιλικό περιβάλλον για τους μαθητές.
- Διόρθωση και βελτίωση του κώδικα.
- Βελτίωση γραφικών από επαγγελματίες στον τομέα του 3D.
- Περισσότερα πειράματα.
- Καλύτερες οδηγίες για κάθε πείραμα(tutorials).
- Δημιουργία των πειραμάτων με την τεχνολογία VR.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] S. Chi-Yin Yuen, G. Yaoyuneyong, E. Johnson, “Augmented Reality: An Overview and Five Directions for AR in Education” 2011, pp. 119-140
- [2] F. P. Brooks, Jr, “What’s Real About Virtual Reality?” 1999
- [3] R. Che Mohd Yusoff, H. Badioze Zaman, A. Ahmad, “Evaluation of user acceptance of mixed reality technology” 2011 pp. 1369-1387
- [4] T. Susi, M. Johannesson, Per Backlund, “Serious Games – An Overview” 2007
- [5] A. Okita, *Learning C# Programming with Unity 3D*, Boca Raton, London, New York: CRC Press 2014
- [6] Unity [Online]. Available: <https://www.unity.com>
- [7] O. Villar. *Learning Blender: A Hands-On Guide to Creating 3D Animated Characters*, Upper Saddle, NJ, Boston, Indianapolis, San Francisco: Addison-Wesley 2014
- [8] Blender [Online]. Available: <https://www.blender.org>
- [9] Unreal Engine [Online]. Available: <https://www.unrealengine.com/>
- [10] Godot [Online]. Available: <https://godotengine.org>
- [11] Vuforia [Online]. Available: <https://developer.vuforia.com>
- [12] Vuforia in Unity Asset Store [Online]. Available: <https://assetstore.unity.com/packages/templates/packs/vuforia-engine-163598>
- [13] Vuforia Manual Download [Online]. Available: git-repo.developer.vuforia.com
- [14] ARMaker [Online]. Available: <https://shawnlehner.github.io/ARMaker/>
- [15] Google Fonts [Online]. Available: <https://fonts.google.com/specimen/Anonymous+Pro#standard-styles>
- [16] Flat Icons [Online]. Available: <https://www.flaticon.com>
- [17] Γ. Θ. Καλκάνης, Ο. Γκικοπούλου, Ε. Καπότης, Δ. Γουσόπουλος, Μ. Πατρινόπουλος, Π. Τσάκωνας, Π. Δημητριάδης, Λ. Παπασιμίπα, Κ. Μιτζήθρας, Α. Καπόγιαννης, Δ. Ι. Σωτηρόπουλος, Σ. Πολίτης, *Η Φυσική με Πειράματα Α΄ Γυμνασίου*, Αθήνα: Εκδόσεις Συμμετρία 2013
- [18] Θ. Θεοχάρης, Γ. Παπαϊωάννου, Ν. Πλατής, Ν. Μ. Πατρικαλάκης, *Γραφικά και Οπτικοποίηση Αρχές και Αλγόριθμοι*, Αθήνα: Ινστιτούτο Τεχνολογίας Υπολογιστών και Εκδόσεων «ΔΙΟΦΑΝΤΟΣ» 2015

ΠΑΡΑΡΤΗΜΑ Α :Αρχείο Mainlight.hlsl

```
void GetLightingInformation_float(out float3 Direction, out float3 Color, out float Attenuation)
{
#ifdef SHADERGRAPH_PREVIEW
    Direction = float3(-0.5, 0.5, -0.5);
    Color = float3(1, 1, 1);
    Attenuation = 0.4;
#else
    Light light = GetMainLight();
    Direction = light.direction;
    Attenuation = light.distanceAttenuation;
    Color = light.color;
#endif
}
```

ΠΑΡΑΡΤΗΜΑ Β :Script PlayerController

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

public class PlayerController : MonoBehaviour
{
    NavMeshAgent myAgent;
    [SerializeField] LayerMask whatIsInteractable;
    [SerializeField] Transform holdingPosition;

    [Header("DEBUG")]
    public Vector3 vel = new Vector3();
    public bool isHolding = false;
```

```

void Start()
{
    myAgent = GetComponent<NavMeshAgent>();
}

void Update()
{
    CheckIfTouch();
    vel = myAgent.velocity;
}

void CheckIfTouch()
{
    if(vel.sqrMagnitude>0)
    {
        return;
    }
    if (Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began)
    {
        if(CheckIfTouchUI()) { return; }

        Ray ray = Camera.main.ScreenPointToRay(Input.touches[0].position);
        RaycastHit hit;
        if (Physics.Raycast(ray, out hit, whatIsInteractable))
        {
            TryInteract(hit);
        }
    }
}

bool CheckIfTouchUI()
{

```

```

foreach (Touch touch in Input.touches)
{
    int id = touch.fingerId;
    if (EventSystem.current.IsPointerOverGameObject(id))
    {
        return true;
    }
}
return false;
}

```

```

void TryInteract(RaycastHit hit)
{
    if (hit.transform.CompareTag("Interactable"))
    {
        TryMove(hit.transform.parent.position);
    } else if (hit.transform.CompareTag("Ground"))
    {
        TryMove(hit.point);
    } else if (hit.transform.CompareTag("Pick up"))
    {
        TryMove(hit.point);
    }
}

```

```

void TryMove(Vector3 hit)
{
    myAgent.SetDestination(hit);
}

```

```

public void PickupItem(Transform item)
{
    if (holdingPosition.childCount > 0)

```

```

    {
        return;
    }
    item.SetParent(holdingPosition);
    item.localPosition = Vector3.zero;
    item.localRotation = Quaternion.Euler(0, 0, 0); //set rotation and position of item's parent to
holdingPosition
    isHolding = true;
}

```

```

public void PutDownButton()
{
    if (holdingPosition.childCount == 0)
        return;
    Vector3 newPosition = holdingPosition.transform.position;
    PutDownItem(transform.parent, newPosition);
}

```

```

public GameObject PutDownItem(Transform newParent, Vector3 newPosition)
{
    GameObject item=GetItemHolding();
    if (item==null)
    {
        return null;
    }
    isHolding = false;
    item.transform.SetParent(newParent);
    item.transform.position = newPosition;
    return item;
}

```

```

public GameObject GetItemHolding()
{

```

```
if (holdingPosition.childCount == 0)
{
    return null;
}
GameObject item = holdingPosition.transform.GetChild(0).gameObject;
return item;
}

private void OnTriggerEnter(Collider other)
{
    if (other.gameObject.CompareTag("Pick up"))
    {
        PickupItem(other.transform.parent);
    }
}
}
```