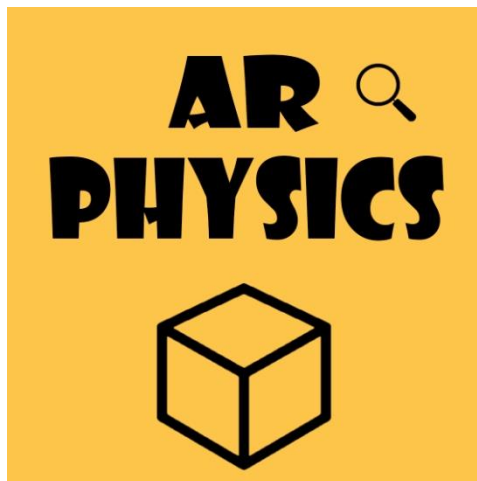




ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ  
«ΔΗΜΙΟΥΡΓΙΑ ΠΑΙΧΝΙΔΙΟΥ ΕΙΚΟΝΙΚΗΣ  
ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΠΟΥ ΑΦΟΡΑ ΤΑ  
ΠΕΙΡΑΜΑΤΑ ΤΗΣ ΦΥΣΙΚΗΣ Α' ΓΥΜΝΑΣΙΟΥ»



Του φοιτητή  
Μάριου Καμπούρη  
Αρ. Μητρώου: 164671

Επιβλέπων  
Ευκλείδης Κεραμόπουλος  
Αναπληρωτής Καθηγητής

## Θεσσαλονίκη 2022

Τίτλος Π.Ε. Δημιουργία Παιχνιδιού Εικονικής Πραγματικότητας που αφορά τα πειράματα της Φυσικής Α' Γυμνασίου.

Κωδικός Π.Ε. 21212

Όνοματεπώνυμο φοιτητή Μάριος Καμπούρης  
Όνοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος

Ημερομηνία ανάληψης Π.Ε. 26-03-2021

Ημερομηνία περάτωσης Π.Ε. ...

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

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Μάριου Καμπούρη 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 πτυχιακής 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

Η συγκεκριμένη εργασία αφορά την ανάπτυξη μίας εφαρμογής επαυξημένης πραγματικότητας (AR – Augmented Reality) για την ψηφιακή υλοποίηση των πειραμάτων του μαθήματος της Φυσικής Α' Γυμνασίου. Η AR τεχνολογία χρησιμοποιείται ευρέως σε μία πληθώρα δραστηριοτήτων μέσα στις οποίες συγκαταλέγεται και η μάθηση. Η μάθηση σε πολλές περιπτώσεις μπορεί να γίνει πιο βιωματική μέσω της ψηφιακής αναπαράστασης του AR, σε χαμηλό κόστος σε ένα μέσο άμεσα προσβάσιμο όπως μία android συσκευή [1]. Η τεχνολογία αυτή μου κέντρισε ιδιαίτερα το ενδιαφέρον, καθώς πιστεύω πως το AR μπορεί να προσφέρει στον μαθητή έναν διαδραστικό, ενδιαφέρον και παράλληλα εύκολα κατανοητό τρόπο εκμάθησης. Η Φυσική είναι ένα μάθημα που περιλαμβάνει πειράματα τα οποία απαιτούν την ύπαρξη ενός χώρου ειδικά φτιαγμένο για την υλοποίησή τους. Ο χώρος αυτός αλλά και τα υλικά τα οποία χρησιμοποιούνται δεν είναι πάντα άμεσα προσβάσιμα από τους μαθητές την στιγμή που αυτοί επιθυμούν. Επιπλέον σαν μαθητής δεν μου δόθηκε η ευκαιρία να υλοποιήσω αρκετά από αυτά τα πειράματα, λόγω έλλειψης των υλικών που είναι απαραίτητα για αυτά. Τέλος, το θέμα αυτό μου φάνηκε ιδιαίτερα ενδιαφέρον καθώς μου δόθηκε η δυνατότητα να ασχοληθώ με τεχνολογίες και τεχνικές οι οποίες προσωπικά μου αρέσουν, όπως η δημιουργία 3D γραφικών, η δημιουργία animation και η ανάπτυξη κώδικα και εφαρμογών android.

## Περίληψη

Η συγκεκριμένη εργασία παρουσιάζει την δημιουργία μιας εφαρμογής επαυξημένης πραγματικότητας (AR – Augmented Reality), η οποία βασίζεται στα πειράματα του βιβλίου της Φυσικής Α Γυμνασίου. Η εφαρμογή αυτή λειτουργεί σε κινητά τηλέφωνα και δημιουργήθηκε με την χρήση της game engine Unity. Η Unity είναι μια IDE (Integrated Development Environment) εφαρμογή, δηλαδή ένα ολοκληρωμένο περιβάλλον ανάπτυξης, η οποία χρησιμοποιείται για την δημιουργία προγραμμάτων - παιχνιδιών. Στη συγκεκριμένη εφαρμογή χρησιμοποιούνται 3D και 2D αντικείμενα. Τα αντικείμενα αυτά δημιουργήθηκαν με την χρήση των εφαρμογών Blender και Photoshop αντίστοιχα. Τα πειράματα του σχολικού βιβλίου που υλοποιήθηκαν στην εφαρμογή είναι αυτά των κεφαλαίων ένα 1 μέχρι 12. Σε όλα τα κεφάλαια της πτυχιακής εργασίας που αφορούν κατασκευές πειραμάτων (Κεφάλαιο 5 μέχρι Κεφάλαιο 15) γίνεται μία αναλυτική επεξήγηση των σκηνών των πειραμάτων, δηλαδή τα βήματα που πρέπει να ακολουθήσει ο χρήστης προκειμένου να υλοποιήσει το πείραμα. Επίσης, γίνεται μία ανάλυση των σημαντικών αντικειμένων που βρίσκονται στις σκηνές αυτές. Σε κάθε κεφάλαιο εξετάζεται ο κώδικας που ενισχύει την τρέχουσα σκηνή, κυρίως οι βασικές μέθοδοι που υλοποιούν σημαντικές και ξεχωριστές λειτουργίες. Η παρουσίαση των σκηνών και η επεξήγηση των αντικειμένων και του κώδικα εμπλουτίζονται με την βοήθεια εικόνων. Τέλος, γίνεται μία συζήτηση όπου παρουσιάζονται τα συμπεράσματα και προτείνονται μελλοντικές βελτιώσεις.

# «Creation of augmented reality based on games, regarding the Physics laboratory experiments for the First Grade of High School»

«Marios Kampouris»

## **Abstract**

This thesis presents the creation of an augmented reality application (AR – Augmented Reality), which is based on the experiments of the First Grade of High School Physics. The application is made for mobile phones and was created by using the Unity Game Engine. Unity is an IDE (Integrated Development Environment) application, which is used to design programs – video games. This specific application uses 3D and 2D objects. These objects were created by using Blender and Photoshop respectively. The school book experiments developed in the application are those of chapters 1 to 12. In all the chapters of the dissertation concerning experimental procedures (Chapter 5 to Chapter 15) there are detailed explanations of the experimental game scenes. Namely, to explain the steps that the user must follow in order to implement the experiment. An analysis of the important game objects found in these scenes is also made. Each chapter examines the code that reinforces the current scene, mainly the basic methods that implement important and distinct functions. The presentation of the scenes and the explanation of the objects and the code are enriched with the help of images. Finally, a discussion is being made where the conclusions are presented and future improvements are suggested.

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα όλα τα άτομα που με βοήθησαν και στάθηκαν δίπλα μου σε όλη αυτή την προσπάθεια να ολοκληρώσω την πτυχιακή εργασία.

Πρώτα, θα ήθελα να εκφράσω τις ευχαριστίες μου στον επιβλέπων καθηγητή κ. Ευκλείδη Κεραμόπουλο που μου επέτρεψε να ασχοληθώ με αυτή την πτυχιακή εργασία και που όποτε χρειάστηκα την βοήθεια του ήταν εκεί για να μου δώσει χρήσιμες συμβουλές και πληροφορίες.

Επίσης, ένα μεγάλο ευχαριστώ οφείλω στην κοπέλα μου για την συμπαράσταση, την βοήθεια της και την εποικοδομητική κριτική που μου προσέφερε.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους μου και την οικογένεια μου που όλο αυτό τον καιρό με στήριξαν και ήταν δίπλα μου.

# Περιεχόμενα

|  |     |
|--|-----|
| Πρόλογος.....  | iii |
| Περίληψη .....   | iv  |
| Abstract.....  | v   |
| Ευχαριστίες .....  | vi  |
| Περιεχόμενα .....  | vii |
| Κατάλογος Σχημάτων .....   | x   |
| Κατάλογος Πινάκων .....  | xv  |
| Συντομογραφίες.....  | xvi |
| Κεφάλαιο 1ο: Επαυξημένη Πραγματικότητα και Φυσική.....   | 1   |
| 1.1 Εισαγωγή .....   | 1   |
| 1.2 Στόχοι Πτυχιακής .....   | 1   |
| 1.3 Φυσική Α' Γυμνασίου .....  | 2   |
| 1.4 Σύνοψη Κεφαλαίων .....   | 2   |
| Κεφάλαιο 2ο: Εφαρμογές και Μέθοδοι.....  | 5   |
| 2.1 Εισαγωγή .....   | 5   |
| 2.2 Unity.....   | 5   |
| 2.2.1 Unity Set Up.....  | 6   |
| 2.2.2 Mobile App Set Up.....   | 10  |
| 2.3 Vuforia.....   | 12  |
| 2.3.1 Vuforia Set-Up .....   | 13  |
| 2.4 Blender.....   | 14  |
| 2.5 Photoshop.....   | 15  |
| 2.6 Audio and Music (Bosca Ceoil) .....  | 16  |
| 2.7 Τι είναι το UI και τι το UX; .....   | 16  |
| 2.7.1 Χρώματα και Χρωματικές Διαφορές .....  | 16  |
| 2.7.2 Τυπογραφία (Fonts) .....   | 18  |
| 2.7.3 Εικόνες προσκηνίου και φόντου.....   | 18  |
| 2.7.4 Η επίδραση του ήχου στην εμπειρία του χρήστη .....   | 18  |
| 2.7.5 Ηχητικά Εφέ, Μουσική και Ρυθμίσεις Ήχου Εφαρμογής (Sound Clips, Sound manager and Audio Listeners) ..... | 18  |
| 2.8 Επίλογος.....  | 19  |
| Κεφάλαιο 3ο: Παρουσίαση Εφαρμογής.....   | 21  |

|               |   |    |
|---------------|---|----|
| 3.1           | Εισαγωγή .....  | 21 |
| 3.2           | Αρχή Εφαρμογής.....   | 21 |
| 3.3           | Πρώτη Εικόνα – Αρχικό Μενού.....                              | 22 |
| 3.3.1         | Κεντρικό Μενού .....  | 22 |
| 3.3.2         | Ρυθμίσεις.....  | 25 |
| 3.3.3         | Έξοδος και Πληροφορίες .....                                  | 26 |
| 3.4           | Βασικός Κορμός Πειράματος .....                               | 27 |
| 3.5           | Τελευταία Εικόνα – Σκηνή Τέλους .....                         | 31 |
| 3.6           | Βασικός κορμός των Assets .....                               | 32 |
| 3.7           | Επίλογος.....   | 33 |
| Κεφάλαιο 4ο:  | Μετρήσεις Μήκους – Η Μέση Τιμή.....                           | 35 |
| 4.1           | Εισαγωγή .....  | 35 |
| 4.2           | Κατασκευή του πειράματος .....                                | 35 |
| 4.3           | Επίλογος.....   | 45 |
| Κεφάλαιο 5ο:  | Μετρήσεις Χρόνου – Η Ακρίβεια.....                            | 47 |
| 5.1           | Εισαγωγή .....  | 47 |
| 5.2           | Κατασκευή του πειράματος .....                                | 47 |
| 5.3           | Επίλογος.....   | 55 |
| Κεφάλαιο 6ο:  | Μετρήσεις Μάζας – Τα διαγράμματα.....                         | 57 |
| 6.1           | Εισαγωγή .....  | 57 |
| 6.2           | Κατασκευή του πειράματος .....                                | 57 |
| 6.3           | Επίλογος.....   | 64 |
| Κεφάλαιο 7ο:  | Μετρήσεις Θερμοκρασίας – Η Βαθμονόμηση.....                   | 65 |
| 7.1           | Εισαγωγή .....  | 65 |
| 7.2           | Κατασκευή του πειράματος .....                                | 65 |
| 7.3           | Επίλογος.....   | 72 |
| Κεφάλαιο 8ο:  | Από τη θερμότητα στη θερμοκρασία – Η θερμική ισορροπία.....   | 73 |
| 8.1           | Εισαγωγή .....  | 73 |
| 8.2           | Κατασκευή του πειράματος .....                                | 73 |
| 8.3           | Επίλογος.....   | 81 |
| Κεφάλαιο 9ο:  | Οι αλλαγές κατάστασης του νερού – Ο «κύκλος» του νερού .....  | 83 |
| 9.1           | Εισαγωγή .....  | 83 |
| 9.2           | Κατασκευή του πειράματος .....                                | 83 |
| 9.3           | Επίλογος.....   | 90 |
| Κεφάλαιο 10ο: | Η διαστολή και συστολή του νερού – Μια φυσική «ανωμαλία»..... | 91 |

|   |  |     |
|---|--|-----|
| 10.1  | Εισαγωγή .....   | 91  |
| 10.2  | Κατασκευή του πειράματος .....   | 91  |
| 10.3  | Επίλογος.....  | 100 |
| Κεφάλαιο 11ο:                                   | Το φως θερμαίνει – «ψυχρά» και «θερμά» χρώματα .....                   | 101 |
| 11.1  | Εισαγωγή .....   | 101 |
| 11.2  | Κατασκευή του πειράματος .....   | 101 |
| 11.3  | Επίλογος.....  | 108 |
| Κεφάλαιο 12ο:                                   | Το φαινόμενο του θερμοκηπίου υπερ-θερμαίνει .....                      | 109 |
| 12.1  | Εισαγωγή .....   | 109 |
| 12.2  | Κατασκευή του πειράματος .....   | 109 |
| 12.3  | Επίλογος.....  | 117 |
| Κεφάλαιο 13ο:                                   | Το ηλεκτρικό βραχυ-κύκλωμα – Κίνδυνοι και «ασφάλεια».....              | 119 |
| 13.1  | Εισαγωγή .....   | 119 |
| 13.2  | Κατασκευή του πειράματος .....   | 119 |
| 13.2.1  | Το ηλεκτρικό κύκλωμα .....   | 120 |
| 13.2.2  | Το ηλεκτρικό βραχυ-κύκλωμα .....                                       | 127 |
| 13.3  | Επίλογος.....  | 133 |
| Κεφάλαιο 14ο:                                   | Από το μαγνητισμό στον ηλεκτρισμό – Η ηλεκτρική (ιδιο-)γεννήτρια ..... | 135 |
| 14.1  | Εισαγωγή .....   | 135 |
| 14.2  | Κατασκευή του πειράματος .....   | 135 |
| 14.3  | Επίλογος.....  | 143 |
| Κεφάλαιο 15ο:                                   | Συμπεράσματα και προτάσεις βελτίωσης.....                              | 145 |
| ΒΙΒΛΙΟΓΡΑΦΙΑ .....                              |  | 147 |
| ΠΑΡΑΡΤΗΜΑ Α : Script “MainMenu.cs” .....        |  | 148 |
| ΠΑΡΑΡΤΗΜΑ Β : Script “SettingsScript.cs”.....   |  | 150 |
| ΠΑΡΑΡΤΗΜΑ C : Script “DemoRaycastKef8.cs” ..... |  | 151 |
| ΠΑΡΑΡΤΗΜΑ D Script “Kef8pirama1.cs”.....        |  | 153 |

## Κατάλογος Σχημάτων

|   |    |
|---|----|
| Σχήμα 1.1: Εξώφυλλο βιβλίου Φυσικής Α' Γυμνασίου .....  | 2  |
| Σχήμα 2.1: Λογότυπο Unity.....  | 5  |
| Σχήμα 2.2: Ομαδοποιημένα project στο Unity Hub.....   | 6  |
| Σχήμα 2.3: Δημιουργία νέου project και επιλογή template στο Unity.....                                    | 7  |
| Σχήμα 2.4: Project Browser window.....  | 7  |
| Σχήμα 2.5: Παράθυρο Inspector: Προβολή στοιχείων και ιδιοτήτων του κάθε υπό επεξεργασία αντικειμένου..... | 8  |
| Σχήμα 2.6: Προεπισκόπηση της εικόνας του παιχνιδιού (Game View).....                                      | 8  |
| Σχήμα 2.7: Κατασκευή αντικειμένων στο Scene View.....   | 9  |
| Σχήμα 2.8: Το Hierarchy window.....   | 10 |
| Σχήμα 2.9: Build Settings.....  | 11 |
| Σχήμα 2.10: Επιλογή ανάλυσης.....   | 11 |
| Σχήμα 2.11: Προεπισκόπηση της εικόνας του παιχνιδιού σε portrait mode (σε ανάλυση κινητού)...             | 12 |
| Σχήμα 2.12: Λογότυπο Vuforia.....   | 12 |
| Σχήμα 2.13: Στοιχεία και ιδιότητες του αντικειμένου AR Camera.....  | 13 |
| Σχήμα 2.14: Vuforia Configuration επιλογές.....   | 14 |
| Σχήμα 2.15: Λογότυπο Blender.....   | 14 |
| Σχήμα 2.16: Λογότυπο Photoshop.....   | 15 |
| Σχήμα 2.17: Button που έχει κατασκευαστεί με την χρήση του Photoshop για την συγκεκριμένη εφαρμογή.....   | 15 |
| Σχήμα 2.18: Λογότυπο Bosca Ceoil.....   | 16 |
| Σχήμα 2.19: Χρωματική αντίθεση button με background.....  | 17 |
| Σχήμα 2.20: Freeze Button.....  | 17 |
| Σχήμα 2.21: ON Button.....  | 17 |
| Σχήμα 2.22: Τύπος γραμματοσειράς της εφαρμογής.....   | 18 |
| Σχήμα 2.23: Προσθήκη Audio Listener Component.....  | 19 |
| Σχήμα 2.24: Προσθήκη Audio Source.....  | 19 |
| Σχήμα 3.1: Το App Icon της Εφαρμογής.....   | 21 |
| Σχήμα 3.2: Το Splash Screen της εφαρμογής.....  | 22 |
| Σχήμα 3.3: Το Κεντρικό Μενού.....   | 23 |
| Σχήμα 3.4: Η Σελίδα Σύνοψης Κεφαλαίων.....  | 24 |

|   |    |
|---|----|
| Σχήμα 3.5: Το MainMenu script που ελέγχει όλες τις συνδέσεις των σκηνών του προγράμματος .....                | 25 |
| Σχήμα 3.6: Η σελίδα των Ρυθμίσεων.....  | 25 |
| Σχήμα 3.7: Το script SettingScript που ελέγχει όλες τις ενέργειες των Ρυθμίσεων .....                         | 26 |
| Σχήμα 3.8: Η μέθοδος εξόδου από την εφαρμογή.....   | 26 |
| Σχήμα 3.9: Η σελίδα πληροφοριών .....   | 27 |
| Σχήμα 3.10: Η σκηνή FirstScanCodeScene .....  | 28 |
| Σχήμα 3.11: Μία τρισδιάστατη σκηνή ανίχνευσης αντικειμένων.....   | 29 |
| Σχήμα 3.12: Μία τρισδιάστατη σκηνή υλοποίησης πειράματος.....   | 30 |
| Σχήμα 3.13: Μία διδιάστατη σκηνή αποτελεσμάτων .....  | 31 |
| Σχήμα 3.14: Η διδιάστατη σκηνή τέλους .....   | 32 |
| Σχήμα 3.15: Τα Assets της εφαρμογής όπως φαίνονται στο Project Browser Window .....                           | 32 |
| Σχήμα 3.16: Τα περιεχόμενα του φακέλου «Scenes» της εφαρμογής όπως φαίνονται στο Project Browser Window ..... | 33 |
| Σχήμα 3.17 Ο βασικός κορμός – λογική του παιχνιδιού.....  | 33 |
| Σχήμα 4.1: Η διδιάστατη σκηνή της πρώτης φάσης του πειράματος του 1ου Κεφαλαίου.....                          | 36 |
| Σχήμα 4.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 1ου Κεφαλαίου.....                      | 37 |
| Σχήμα 4.3: Εισαγωγή βιβλιοθηκών στον κώδικα.....  | 37 |
| Σχήμα 4.4: Αρχικοποίηση μεταβλητών του script DemoRaycast.....  | 38 |
| Σχήμα 4.5: Η μέθοδος Awake() του script DemoRaycast.....  | 38 |
| Σχήμα 4.6: Η μέθοδος Start() του script DemoRaycast.....  | 38 |
| Σχήμα 4.7: Η μέθοδος Update() του script DemoRaycast.....   | 39 |
| Σχήμα 4.8: Το πεδίο Tag στο Inspector Window.....   | 40 |
| Σχήμα 4.9: Ο έλεγχος που οδηγεί στην φόρτωση της επόμενης σκηνής.....   | 40 |
| Σχήμα 4.10: Η σκηνή της τρίτης φάσης του πειράματος του 1ου Κεφαλαίου.....                                    | 41 |
| Σχήμα 4.11: Αρχικοποίηση μεταβλητών του script Kef1Exp1.....  | 42 |
| Σχήμα 4.12: Η μέθοδος Start() του script Kef1Exp1.....  | 43 |
| Σχήμα 4.13: Έλεγχος επιλεγμένου αντικειμένου μέσα στην μεθόδου Update() του script Kef1Exp1.....              | 43 |
| Σχήμα 4.14: Ο έλεγχος που οδηγεί στην εμφάνιση του κουμπιού τέλους.....                                       | 44 |
| Σχήμα 4.15: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 1ου Κεφαλαίου.....                     | 44 |
| Σχήμα 4.16: Η διδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 1ου Κεφαλαίου.....                        | 45 |
| Σχήμα 5.1: Η διδιάστατη σκηνή της πρώτης φάσης του πειράματος του 2ου Κεφαλαίου.....                          | 47 |
| Σχήμα 5.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 2ου Κεφαλαίου.....                      | 48 |

|  |    |
|--|----|
| Σχήμα 5.3: Η σκηνή της τρίτης φάσης του πειράματος του 2ου Κεφαλαίου.....                    | 49 |
| Σχήμα 5.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 2ου Κεφαλαίου.....     | 50 |
| Σχήμα 5.5: Αρχικοποίηση μεταβλητών του script Kef2exp1scr.....                               | 51 |
| Σχήμα 5.6: Η μέθοδος Start() του script Kef2exp1scr.....                                     | 51 |
| Σχήμα 5.7: Η μέθοδος DisplayTime() του script Kef2exp1scr.....                               | 52 |
| Σχήμα 5.8: Η μέθοδος Update() του script Kef2exp1scr.....                                    | 52 |
| Σχήμα 5.9: Οι μέθοδοι StartTimer(), StopTimer() και ResetTimer() του script Kef2exp1scr..... | 53 |
| Σχήμα 5.10: Οι μέθοδοι activateAnalog() και activateDigital() του script Kef2exp1scr.....    | 53 |
| Σχήμα 5.11: Οι μέθοδοι NextCar() και PreviousCar() του script Kef2exp1scr.....               | 54 |
| Σχήμα 5.12: Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 2ου Κεφαλαίου.....      | 55 |
| Σχήμα 6.1: Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 3ου Κεφαλαίου.....        | 57 |
| Σχήμα 6.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 3ου Κεφαλαίου.....     | 58 |
| Σχήμα 6.3: Η σκηνή της τρίτης φάσης του πειράματος του 3ου Κεφαλαίου.....                    | 59 |
| Σχήμα 6.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 3ου Κεφαλαίου.....     | 60 |
| Σχήμα 6.5: Αρχικοποίηση μεταβλητών του script Kef3pirama1script.....                         | 61 |
| Σχήμα 6.6: Η μέθοδος Start() του script Kef3pirama1script.....                               | 61 |
| Σχήμα 6.7: Η μέθοδος Update() του script Kef3pirama1script.....                              | 62 |
| Σχήμα 6.8: Ο έλεγχος που οδηγεί στην εμφάνιση του κουμπιού τέλους.....                       | 63 |
| Σχήμα 6.9: Χρήση animation μέσω script.....  | 63 |
| Σχήμα 6.10: Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 3ου Κεφαλαίου.....      | 64 |
| Σχήμα 7.1: Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 4ου Κεφαλαίου.....        | 65 |
| Σχήμα 7.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 4ου Κεφαλαίου.....     | 66 |
| Σχήμα 7.3: Η σκηνή της τρίτης φάσης του πειράματος του 4ου Κεφαλαίου.....                    | 67 |
| Σχήμα 7.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 4ου Κεφαλαίου.....     | 68 |
| Σχήμα 7.5: Αρχικοποίηση μεταβλητών του script Kef4pirama1script.....                         | 69 |
| Σχήμα 7.6: Η μέθοδος Awake() του script Kef4pirama1script.....                               | 69 |
| Σχήμα 7.7: Η μέθοδος Update() του script Kef4pirama1script.....                              | 70 |
| Σχήμα 7.8: Ο έλεγχος που οδηγεί στην εμφάνιση του κουμπιού τέλους.....                       | 71 |
| Σχήμα 7.9: Εικόνα Χρήση animation μέσω script.....   | 71 |
| Σχήμα 7.10: Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 4ου Κεφαλαίου.....      | 72 |
| Σχήμα 8.1: Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 5ου Κεφαλαίου.....        | 73 |
| Σχήμα 8.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 5ου Κεφαλαίου.....     | 74 |
| Σχήμα 8.3: Η σκηνή της τρίτης φάσης του πειράματος του 5ου Κεφαλαίου.....                    | 75 |

|   |     |
|---|-----|
| Σχήμα 8.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 5ου Κεφαλαίου.....              | 76  |
| Σχήμα 8.5: Αρχικοποίηση μεταβλητών του script Kef5pirama1script.....                                  | 77  |
| Σχήμα 8.6: Η μέθοδος Awake() του script Kef5pirama1script.....  | 78  |
| Σχήμα 8.7: Η μέθοδος Update() του script Kef5pirama1script.....                                       | 79  |
| Σχήμα 8.8: Ο έλεγχος που οδηγεί στην εμφάνιση του κουμπιού τέλους.....                                | 79  |
| Σχήμα 8.9: Χρήση animation μέσω script.....   | 79  |
| Σχήμα 8.10: Η μέθοδος Update() του script Kef5pirama1script.....                                      | 80  |
| Σχήμα 8.11: Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 5ου Κεφαλαίου.....               | 80  |
| Σχήμα 9.1: Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 6ου Κεφαλαίου.....                 | 83  |
| Σχήμα 9.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 6ου Κεφαλαίου.....              | 84  |
| Σχήμα 9.3: Η σκηνή της τρίτης φάσης του πειράματος του 6ου Κεφαλαίου.....                             | 85  |
| Σχήμα 9.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 6ου Κεφαλαίου.....              | 86  |
| Σχήμα 9.5: Αρχικοποίηση μεταβλητών του script Kef6pirama1script.....                                  | 87  |
| Σχήμα 9.6: Η μέθοδος Awake() του script Kef6pirama1script.....  | 87  |
| Σχήμα 9.7: Η μέθοδος startheating() του script Kef6pirama1script.....                                 | 88  |
| Σχήμα 9.8: Η μέθοδος getdoxeiowater() του script Kef6pirama1script.....                               | 88  |
| Σχήμα 9.9: Η μέθοδος getdoxeioBOIL() του script Kef6pirama1script.....                                | 89  |
| Σχήμα 9.10: Χρήση animation μέσω script.....  | 89  |
| Σχήμα 9.11: Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 6ου Κεφαλαίου.....               | 90  |
| Σχήμα 10.1: Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 7ου Κεφαλαίου.....                | 91  |
| Σχήμα 10.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 7ου Κεφαλαίου.....             | 92  |
| Σχήμα 10.3: Η σκηνή της τρίτης φάσης του πειράματος του 7ου Κεφαλαίου.....                            | 93  |
| Σχήμα 10.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 7ου Κεφαλαίου.....             | 94  |
| Σχήμα 10.5: Τελική φάση τρισδιάστατης σκηνής της τέταρτης φάσης του πειράματος του 7ου Κεφαλαίου..... | 95  |
| Σχήμα 10.6: Αρχικοποίηση μεταβλητών του script Kef7pirama1script.....                                 | 96  |
| Σχήμα 10.7: Η μέθοδος Awake() του script Kef7pirama1script.....                                       | 96  |
| Σχήμα 10.8: Η μέθοδος Freeze() του script Kef7pirama1script.....                                      | 97  |
| Σχήμα 10.9: Κάλεσμα της μεθόδου readytotake().....  | 97  |
| Σχήμα 10.10: Η μέθοδος readytotake() του script Kef7pirama1script.....                                | 98  |
| Σχήμα 10.11: Η μέθοδος που εμφανίζει το κουμπί τέλους.....  | 98  |
| Σχήμα 10.12: Η μέθοδος Update() του script Kef7pirama1script.....                                     | 99  |
| Σχήμα 10.13: Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 7ου Κεφαλαίου.....              | 100 |

|  |     |
|--|-----|
| Σχήμα 11.1: Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 8ου Κεφαλαίου.....                         | 101 |
| Σχήμα 11.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 8ου Κεφαλαίου.....                      | 102 |
| Σχήμα 11.3: Η σκηνή της τρίτης φάσης του πειράματος του 8ου Κεφαλαίου.....                                     | 103 |
| Σχήμα 11.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 8ου Κεφαλαίου.....                      | 104 |
| Σχήμα 11.5: Αρχικοποίηση μεταβλητών του script Kef8pirama1script.....  | 105 |
| Σχήμα 11.6: Η μέθοδος Awake() του script Kef8pirama1script.....  | 105 |
| Σχήμα 11.7: Κάλεσμα της μεθόδου gettext().....   | 106 |
| Σχήμα 11.8: Η μέθοδος Update() του script Kef8pirama1script.....   | 107 |
| Σχήμα 11.9: Κάλεσμα της μεθόδου endexperiment().....   | 107 |
| Σχήμα 11.10: Η μέθοδος που εμφανίζει το κουμπί τέλους.....   | 107 |
| Σχήμα 11.11: Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 8ου Κεφαλαίου.....                       | 108 |
| Σχήμα 12.1: Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 9ου Κεφαλαίου.....                         | 109 |
| Σχήμα 12.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 9ου Κεφαλαίου.....                      | 110 |
| Σχήμα 12.3: Η σκηνή της τρίτης φάσης του πειράματος του 9ου Κεφαλαίου.....                                     | 111 |
| Σχήμα 12.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 9ου Κεφαλαίου.....                      | 112 |
| Σχήμα 12.5: Αρχικοποίηση μεταβλητών του script Kef9pirama1script.....  | 113 |
| Σχήμα 12.6: Η μέθοδος Awake() του script Kef9pirama1script.....  | 113 |
| Σχήμα 12.7: Η μέθοδος cutfelizol() του script Kef9pirama1script.....   | 114 |
| Σχήμα 12.8: Κάλεσμα της μεθόδου switchfotistika().....   | 115 |
| Σχήμα 12.9: Η μέθοδος switchfotistika() του script Kef9pirama1script.....                                      | 115 |
| Σχήμα 12.10: Κάλεσμα της μεθόδου endexperiment().....  | 115 |
| Σχήμα 12.11: Η μέθοδος που εμφανίζει το κουμπί τέλους.....   | 116 |
| Σχήμα 12.12: Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 9ου Κεφαλαίου.....                       | 116 |
| Σχήμα 13.1: Η δισδιάστατη σκηνή της πρώτης φάσης του 1ου και 2ου πειράματος του 10ου Κεφαλαίου.....            | 119 |
| Σχήμα 13.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του 1ου και 2ου πειράματος του 10ου Κεφαλαίου.....         | 120 |
| Σχήμα 13.3: Η σκηνή της τρίτης φάσης του 1ου πειράματος του 10ου Κεφαλαίου.....                                | 121 |
| Σχήμα 13.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του 1ου πειράματος του 10ου Κεφαλαίου, με κλειστό φως..... | 122 |
| Σχήμα 13.5: Η τρισδιάστατη σκηνή της τέταρτης φάσης του 1ου πειράματος του 10ου Κεφαλαίου με ανοιχτό φως.....  | 123 |
| Σχήμα 13.6: Αρχικοποίηση μεταβλητών του script Kef10pirama1script.....   | 124 |
| Σχήμα 13.7: Η μέθοδος Awake() του script Kef10pirama1script.....   | 124 |

|   |     |
|---|-----|
| Σχήμα 13.8: Η μέθοδος Update() του script Kef10pirama1script.....                               | 125 |
| Σχήμα 13.9: Κάλεσμα μεθόδου endexperiment().....  | 126 |
| Σχήμα 13.10: Η μέθοδος που εμφανίζει το κουμπί τέλους.....                                      | 126 |
| Σχήμα 13.11: Η δισδιάστατη σκηνή της πέμπτης φάσης του 1ου πειράματος του 10ου Κεφαλαίου..      | 127 |
| Σχήμα 13.12: Η σκηνή της τρίτης φάσης του 2ου πειράματος του 10ου Κεφαλαίου.....                | 128 |
| Σχήμα 13.13: Η τρισδιάστατη σκηνή της τέταρτης φάσης του 2ου πειράματος του 10ου Κεφαλαίου..... | 129 |
| Σχήμα 13.14: Αρχικοποίηση μεταβλητών του script Kef10pirama2script.....                         | 130 |
| Σχήμα 13.15: Η μέθοδος Awake() του script Kef10pirama2script.....                               | 130 |
| Σχήμα 13.16: Η μέθοδος Update() του script Kef10pirama2script.....                              | 131 |
| Σχήμα 13.17: Κάλεσμα μεθόδου endexperiment().....   | 132 |
| Σχήμα 13.18: Η μέθοδος που εμφανίζει το κουμπί τέλους.....                                      | 132 |
| Σχήμα 13.19: Η δισδιάστατη σκηνή της πέμπτης φάσης του 2ου πειράματος του 10ου Κεφαλαίου..      | 133 |
| Σχήμα 14.1: Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 12ου Κεφαλαίου.....         | 135 |
| Σχήμα 14.2: Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 12ου Κεφαλαίου.....      | 136 |
| Σχήμα 14.3: Η σκηνή της τρίτης φάσης του πειράματος του 12ου Κεφαλαίου.....                     | 137 |
| Σχήμα 14.4: Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 12ου Κεφαλαίου.....      | 138 |
| Σχήμα 14.5: Αρχικοποίηση μεταβλητών του script Kef12pirama1script.....                          | 139 |
| Σχήμα 14.6: Η μέθοδος Awake() του script Kef12pirama1script.....                                | 139 |
| Σχήμα 14.7: Τα στοιχεία event trigger του ROTATEButton κουμπιού.....                            | 140 |
| Σχήμα 14.8: Οι μέθοδοι lightson() και stoplight() του script Kef4pirama1script.....             | 141 |
| Σχήμα 14.9: Κάλεσμα της μεθόδου endexperiment().....  | 141 |
| Σχήμα 14.10: Η μέθοδος που εμφανίζει το κουμπί τέλους. ....                                     | 142 |
| Σχήμα 14.11: Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 12ου Κεφαλαίου.....       | 142 |

## Κατάλογος Πινάκων

|  |     |
|--|-----|
| Πίνακας 1.1: Κεφάλαια του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου «Φυσική με Πειράματα» πάνω στα οποία υλοποιήθηκαν πειράματα στην παρούσα εφαρμογή..... | 3   |
| Πίνακας 15.1: Μελλοντικές επεκτάσεις της εφαρμογής.....  | 146 |

## Συντομογραφίες

|      |                                    |
|------|------------------------------------|
| Π.Ε. | Πτυχιακή Εργασία                   |
| AR   | Augmented Reality                  |
| IDE  | Integrated Development Environment |
| GUI  | Graphical User Interface           |
| QR   | Quick Response                     |
| UX   | User Experience                    |
| UI   | User Interface                     |
| IDE  | Integrated Development Environment |
| OS   | Operating System                   |

## Κεφάλαιο 1ο: Επαυξημένη Πραγματικότητα και Φυσική

### 1.1 Εισαγωγή

Η επαυξημένη πραγματικότητα (AR-Augmented Reality) είναι η επαύξηση ή αλλιώς αναπαράσταση ψηφιακών αντικειμένων στον πραγματικό κόσμο μέσω μιας υπολογιστικής συσκευής [2][3][4]. Ένα σύστημα AR τεχνολογίας συνδυάζει πραγματικά και ψηφιακά αντικείμενα στον πραγματικό χώρο, τα ενοποιεί και λειτουργεί διαδραστικά σε τρεις διαστάσεις και σε πραγματικό χρόνο [4]. Τα τελευταία χρόνια παρατηρείται μία πολύ έντονη εμφάνιση της συγκεκριμένης τεχνολογίας σε ποικίλους τομείς [3]. Ένας από τους λόγους που συμβαίνει αυτό είναι η άμεση πρόσβαση σε συσκευές που υποστηρίζουν AR τεχνολογίες. Η ιατρική, η βιομηχανία, ο στρατός και η καθημερινή μας διασκέδαση είναι κάποιοι από τους τομείς στους οποίους βρίσκει εφαρμογή η AR τεχνολογία [4]. Στην συγκεκριμένη εργασία θα ασχοληθούμε με τον τομέα της εκπαίδευσης. Μέσω της επαυξημένης πραγματικότητας τα σχολικά μαθήματα γίνονται πιο διαδραστικά καθώς η συνύπαρξη των ψηφιακών αντικειμένων με το πραγματικό χώρο επιτρέπει την οπτικοποίηση φαινομένων και εννοιών που πολλές φορές δεν είναι δυνατόν να υλοποιηθούν στον πραγματικό χώρο [5]. Με τον τρόπο αυτό τα σχολικά μαθήματα αποκτούν μεγαλύτερο ενδιαφέρον, καθώς προσφέρεται στον μαθητή η δυνατότητα μάθησης μέσω παιχνιδιού, αλλά και γίνεται η χρήση τεχνολογιών που ο μαθητής χρησιμοποιεί συχνά στην καθημερινή του ζωή, όπως είναι τα tablet και τα smartphone.

### 1.2 Στόχοι Πτυχιακής

Στην σημερινή εποχή τα παιδιά νεαρής ηλικίας έχουν ολοένα και πιο μεγάλη ευχέρεια με την τεχνολογία. Πιο συγκεκριμένα, από τα μαθησιακά τους χρόνια, τα παιδιά έρχονται πολύ συχνά σε επαφή, ή κατέχουν μία υπολογιστική συσκευή, όπως για παράδειγμα ένα smartphone ή ένα tablet. Οι συσκευές αυτές προσφέρουν στην μάθηση μία πληθώρα νέων δυνατοτήτων, όπως η δυνατότητα αναπαράστασης ή επαύξησης ψηφιακών μοντέλων μέσα στον πραγματικό κόσμο μέσω της οθόνης μίας συσκευής. Η δυνατότητα αυτή παρέχεται από την τεχνολογία του AR. Στόχος αυτής της πτυχιακής είναι η αξιοποίηση της τεχνολογίας του AR για μαθησιακούς λόγους. Στη συγκεκριμένη περίπτωση η τεχνολογία AR χρησιμοποιείται για την εκμάθηση Φυσικής με διαδραστικά ψηφιακά πειράματα. Μέσω της εφαρμογής οι μαθητές θα έχουν άμεση πρόσβαση στην υλοποίηση κάποιου πειράματος του σχολικού βιβλίου, το οποίο θα μπορεί να γίνεται οποιαδήποτε μέρα και ώρα αυτοί το επιθυμούν, αρκεί να έχουν στη διάθεση τους μία φορητή υπολογιστική συσκευή. Το σχολικό βιβλίο θα είναι σχεδιασμένο να περιέχει ένα μοναδικό QR (Quick Response – Γρήγορης Ανταπόκρισης) Code σε κάθε πείραμα, το οποίο ο μαθητής θα μπορεί να σκανάρει με την συσκευή για να υλοποιήσει το πείραμα. Με αυτό τον τρόπο, η υλοποίηση των πειραμάτων θα γίνεται ομαδοποιημένα μέσω της εφαρμογής έχοντας διδακτικό αλλά και ταυτόχρονα ψυχαγωγικό σκοπό.

### 1.3 Φυσική Α' Γυμνασίου



Εικόνα 1.1 Εξώφυλλο βιβλίου Φυσικής Α' Γυμνασίου.

Στο μάθημα της Φυσικής Α' Γυμνασίου «Η Φυσική με Πειράματα», οι μαθητές μελετούν φυσικά φαινόμενα και τεχνολογικές εφαρμογές [6]. Έρχονται σε πρώτη επαφή με την μέτρηση και την κατανόηση των τιμών φυσικών μεγεθών, κάνουν πειράματα και βγάζουν συμπεράσματα πάνω σε αυτά. Τα φυσικά μεγέθη που παρουσιάζονται είναι το μήκος, ο χρόνος, η μάζα, η θερμοκρασία, η ένταση του φωτός, ο ηλεκτρισμός και ο μαγνητισμός. Ο πειραματισμός είναι μία απαραίτητη διαδικασία για την εκμάθηση των φυσικών επιστημών. Μέσω των πειραμάτων ο μαθητής έχει την ευκαιρία να βγάζει συμπεράσματα αλλά και να τα εφαρμόζει. Στο μάθημα Φυσικής της Α' Γυμνασίου «Φυσική με Πειράματα», οι μαθητές μελετούν δώδεκα θεματικές ενότητες οι οποίες ασχολούνται με σημαντικά φυσικά φαινόμενα και τεχνολογικές εφαρμογές. Οι ενότητες αυτές εφοδιάζουν τον μαθητή με πολύ χρήσιμες γνώσεις αλλά και τον προετοιμάζουν για τα μαθήματα Φυσικής που θα συναντήσει σε επόμενες τάξεις.

### 1.4 Σύνοψη Κεφαλαίων

Η εργασία αυτή περιέχει δεκαέξι κεφάλαια. Στο πρώτο κεφάλαιο «Εισαγωγή» παρουσιάζεται ο στόχος της πτυχιακής εργασίας και γίνεται μία αναφορά στα περιεχόμενα του μαθήματος της Φυσικής Α' Γυμνασίου «Φυσική με Πειράματα». Το δεύτερο κεφάλαιο «Εφαρμογές και Μέθοδοι» αναλύει τις τεχνολογίες, και το στήσιμο των εφαρμογών που χρησιμοποιήθηκαν για την δημιουργία του AR παιχνιδιού. Στο τρίτο κεφάλαιο επεξηγείται η λογική πίσω από το UX (User Experience) και το UI (User Interface) της εφαρμογής, τις επιλογές που έγιναν πάνω σε αυτό για την αισθητική και την συνολική εμπειρία που θα προσφέρει η εφαρμογή στον χρήστη. Επιπλέον αναλύονται με την χρήση εικόνων και σχεδιαγραμμάτων οι τρόποι με τους οποίους τροποποιήθηκαν το UX και το UI. Στο τέταρτο κεφάλαιο προβάλλεται ο βασικός κορμός της εφαρμογής, από την στιγμή δηλαδή που ο χρήστης εισέρχεται στο παιχνίδι, κάνει την επιλογή του πειράματος και ολοκληρώνει την δραστηριότητα.

Στα κεφάλαια πέντε με δεκαπέντε παρουσιάζεται το πως υλοποιήθηκαν εντός της εφαρμογής τα πειράματα των κεφαλαίων του σχολικού βιβλίου. Πιο συγκεκριμένα τα ονόματα των κεφαλαίων πάνω στα οποία δημιουργήθηκαν παιχνίδια εντός της εφαρμογής είναι τα ακόλουθα: «Μετρήσεις μήκους – Η μέση τιμή», «Μετρήσεις χρόνου – Η ακρίβεια», «Μετρήσεις μάζας – Τα διαγράμματα», «Μετρήσεις θερμοκρασίας – Η βαθμονόμηση», «Από τη θερμότητα στη θερμοκρασία – Η θερμική

ισορροπία», «Οι αλλαγές κατάστασης του νερού – Ο "κύκλος" του νερού», «Η διαστολή και συστολή του νερού – Μια φυσική "ανωμαλία"», «Το φως θερμαίνει – "ψυχρά" και "θερμά" χρώματα», «Το φαινόμενο του θερμοκηπίου (υπερ-)θερμαίνει», «Το ηλεκτρικό βραχυ-κύκλωμα – Κίνδυνοι και "ασφάλεια"», «Από το μαγνητισμό στον ηλεκτρισμό – Η ηλεκτρική (ιδιο-)γεννήτρια» (Πίνακας 1.1). Τέλος, στο κεφάλαιο δεκαέξι αναλύονται τα συμπεράσματα και οι προτάσεις για μελλοντικές βελτιώσεις.

| Πειράματα του σχολικού βιβλίου |  |
|--------------------------------|--|
| 1                              | Μετρήσεις μήκους – Η μέση τιμή                                   |
| 2                              | Μετρήσεις χρόνου – Η ακρίβεια                                    |
| 3                              | Μετρήσεις μάζας – Τα διαγράμματα                                 |
| 4                              | Μετρήσεις θερμοκρασίας – Η βαθμονόμηση                           |
| 5                              | Από τη θερμότητα στη θερμοκρασία – Η θερμική ισορροπία           |
| 6                              | Οι αλλαγές κατάστασης του νερού – Ο «κύκλος» του νερού           |
| 7                              | Η διαστολή και συστολή του νερού – Μια φυσική «ανωμαλία»         |
| 8                              | Το φως θερμαίνει – «ψυχρά» και «θερμά» χρώματα                   |
| 9                              | Το φαινόμενο του θερμοκηπίου υπερ-θερμαίνει                      |
| 10                             | Το ηλεκτρικό βραχυ-κύκλωμα – Κίνδυνοι και «ασφάλεια»             |
| 12                             | Από το μαγνητισμό στον ηλεκτρισμό – Η ηλεκτρική (ιδιο-)γεννήτρια |

Πίνακας 1.1 Κεφάλαια του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου «Φυσική με Πειράματα» πάνω στα οποία υλοποιήθηκαν πειράματα στην παρούσα εφαρμογή.



## Κεφάλαιο 2ο: Εφαρμογές και Μέθοδοι

### 2.1 Εισαγωγή

Μία AR εφαρμογή σχεδιασμένη για λειτουργία σε κινητό τηλέφωνο απαιτεί την χρήση πολλαπλών εργαλείων και τεχνολογιών. Σε αυτές τις τεχνολογίες συμπεριλαμβάνονται εργαλεία δημιουργίας και μορφοποίησης εικόνας, ήχου αλλά και μηχανές ανάπτυξης παιχνιδιών με ενσωματωμένο περιβάλλον ανάπτυξης (integrated development environment - IDE). Επιπλέον, η αλληλεπίδραση του χρήστη με το περιβάλλον επαυξημένης πραγματικότητας αποτελεί ένα πολύ σημαντικό κομμάτι της σχεδίασης μιας AR εφαρμογής. Τα στοιχεία που εισάγουν τον χρήστη σε μια ολοκληρωμένη εικόνα επαυξημένης πραγματικότητας είναι το UI (User Interface ή Διεπαφή Χρήστη) και το UX (User Experience ή Εμπειρία Χρήστη). Στις έννοιες αυτές συγκαταλέγονται οπτικές λεπτομέρειες όπως οι χρωματισμοί και ο τύπος και το μέγεθος της γραμματοσειράς του κειμένου (font), αλλά και η χρήση εικόνων, ηχητικών εφέ και μουσικής.

### 2.2 Unity

Το Unity είναι μία IDE εφαρμογή (Ένα ολοκληρωμένο περιβάλλον ανάπτυξης) είναι ένα λογισμικό για τη δημιουργία εφαρμογών που συνδυάζει κοινά εργαλεία προγραμματιστή με ένα ενιαίο γραφικό περιβάλλον χρήστη (Graphical User Interface - GUI)[7] που επιτρέπει την γρήγορη και αποτελεσματική δημιουργία αντικειμένων, την εισαγωγή εξωτερικών στοιχείων και την σύνδεση όλων αυτών με έναν κώδικα [8]. Χαρακτηρίζεται ως μία τεχνολογία ιδιαίτερα χρήσιμη για την δημιουργία παιχνιδιών διότι παρέχει εργαλεία που εφαρμόζουν γραφικά, αναπαράγουν ήχο, ενσωματώνουν φυσική, αλλά και δημιουργούν αλληλεπιδράσεις και δικτύωση μεταξύ παικτών και αντικειμένων.



Εικόνα 2.1 Λογότυπο Unity.

Η εφαρμογή Unity, ή αλλιώς Unity3D, είναι γνωστή για την χρησιμότητα που παρουσιάζει σε νέους game developers λόγω των προοπτικών που παρέχει πάνω στην γρήγορη δημιουργία παιχνιδιών. Χαρακτηρίζεται ως πολύ καλή επιλογή IDE εφαρμογής για χρήση από μικρά στούντιο και ανεξάρτητους προγραμματιστές που δημιουργούν παιχνίδια μόνοι. Η μεγάλη βάση χρηστών και η εξαιρετικά ενεργή κοινότητα χρηστών (2.8 δισεκατομμύρια ενεργοί χρήστες μέχρι το 2020) επιτρέπει

σε αρχάριους αλλά και έμπειρους δημιουργούς παιχνιδιών να αποκτήσουν απαντήσεις και να μοιράσουν γρήγορα πληροφορίες [9].

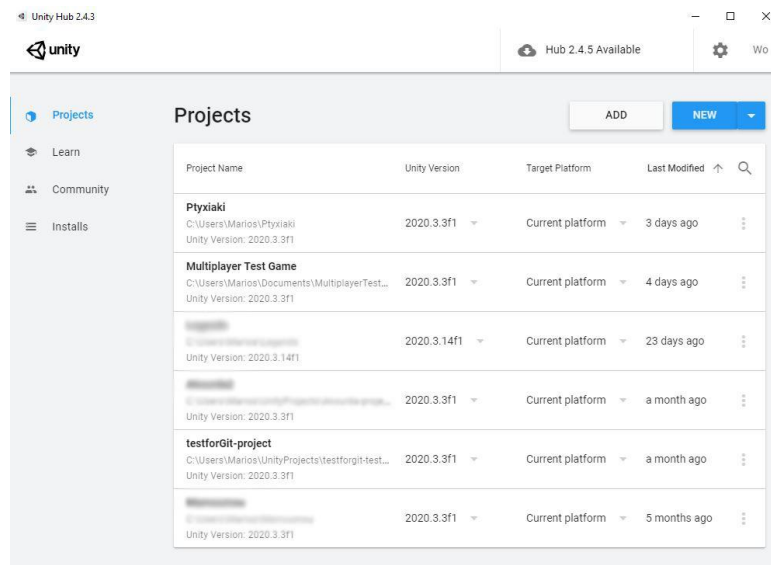
Επιπλέον, η εφαρμογή παρέχει ένα εξαιρετικό σημείο εισόδου στην ανάπτυξη παιχνιδιών, εξισορροπώντας τα χαρακτηριστικά και τη λειτουργικότητα με την τιμή κόστους χρήσης των δυνατοτήτων της. Η δωρεάν έκδοση του Unity παρέχει όλα τα απαραίτητα εργαλεία για την δημιουργία εφαρμογών – παιχνιδιών με μόνο περιορισμό τις ικανότητες του χρήστη [10].

Τέλος, ένα πολύ δυνατό χαρακτηριστικό του Unity είναι η παροχή ενός ενσωματωμένου περιβάλλοντος δέσμης ενεργειών και δυνατοτήτων δικτύωσης, αλλά και η συμβολή στην ανάπτυξη project που να είναι προσβάσιμα από πολλαπλές πλατφόρμες (PC, Console, Android, iOS).

### 2.2.1 Unity Set Up

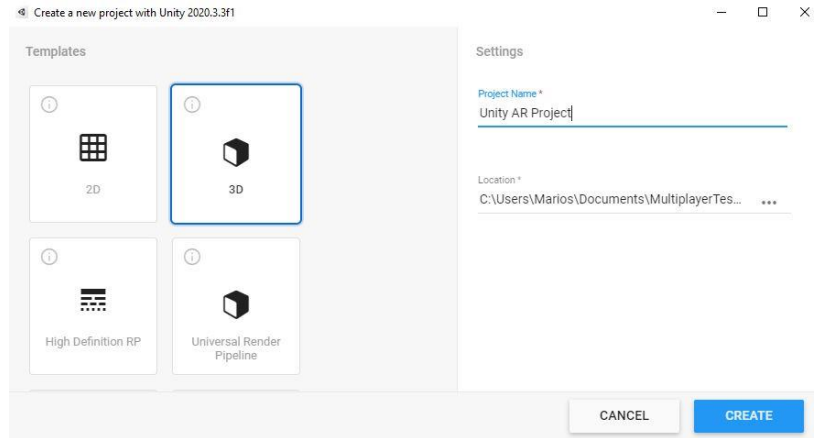
Η εγκατάσταση και το γενικό Set Up του Unity για ένα Android Project κινητού γίνεται με τρόπο απλό και γρήγορο λόγω του εύκολου και φιλικού περιβάλλοντος που προσφέρει. Αρχικά κατεβάζουμε το Unity από το site <https://unity.com/>. Από εκεί κατευθυνόμαστε στην σελίδα «download» και έπειτα πατάμε το κουμπί Download for Windows για να κατεβάσουμε την τελευταία έκδοση. Εφόσον ολοκληρωθεί το download κάνουμε την εγκατάσταση του και έπειτα τρέχουμε το executable (.exe) που δημιουργήθηκε. Είτε επιλέγοντας το Windows Start -> Programs -> Unity (C:\Program Files\Editor\Unity.exe) είτε τρέχοντας το μέσω του desktop icon εάν αυτό δημιουργήθηκε.

Το Unity έπειτα μας κατευθύνει στο Unity Hub. Σε αυτό βλέπουμε ομαδοποιημένα όλα μας τα project (Εικόνα 2.2) αλλά και μερικές άλλες επιλογές όπως τα tab του Learn , Community, Installs. Στην τελευταία επιλογή μπορούμε να δούμε όλες τις εγκατεστημένες εκδόσεις του Unity στο σύστημα μας.



Εικόνα 2.2 Ομαδοποιημένα project στο Unity Hub.

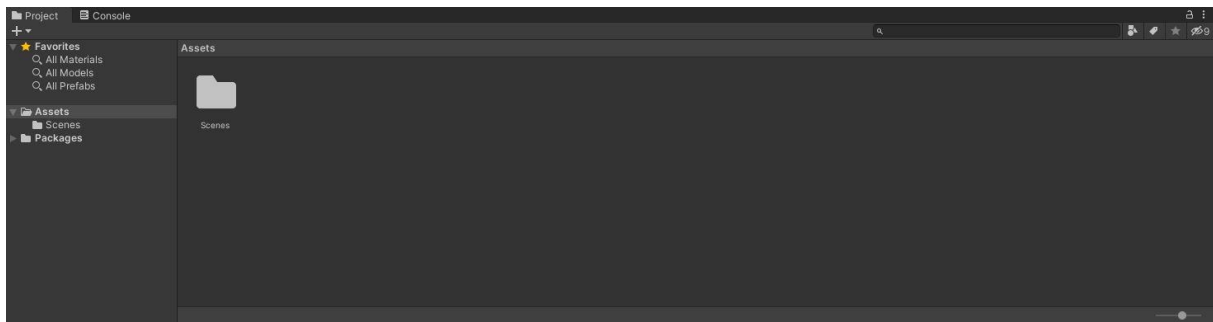
Στη συνέχεια πατάμε το κουμπί NEW για να δημιουργήσουμε ένα νέο project. Πατώντας το θα ανοίξει ένα νέο παράθυρο στο οποίο μας δίνεται η επιλογή να επιλέξουμε template για το project μας (Εικόνα 2.3). Διαλέγουμε το 3D Template , βάζουμε στο text field το Project Name της επιλογής μας (π.χ. Unity AR Project) και στο Location text field επιλέγουμε την τοποθεσία όπου θα δημιουργηθούν και θα αποθηκεύονται τοπικά όλα τα αρχεία του project μας. Τέλος πατώντας το κουμπί CREATE φτιάχνουμε το project μας και εισερχόμαστε αυτόματα σε αυτό.



Εικόνα 2.3 Δημιουργία νέου project και επιλογή template στο Unity.

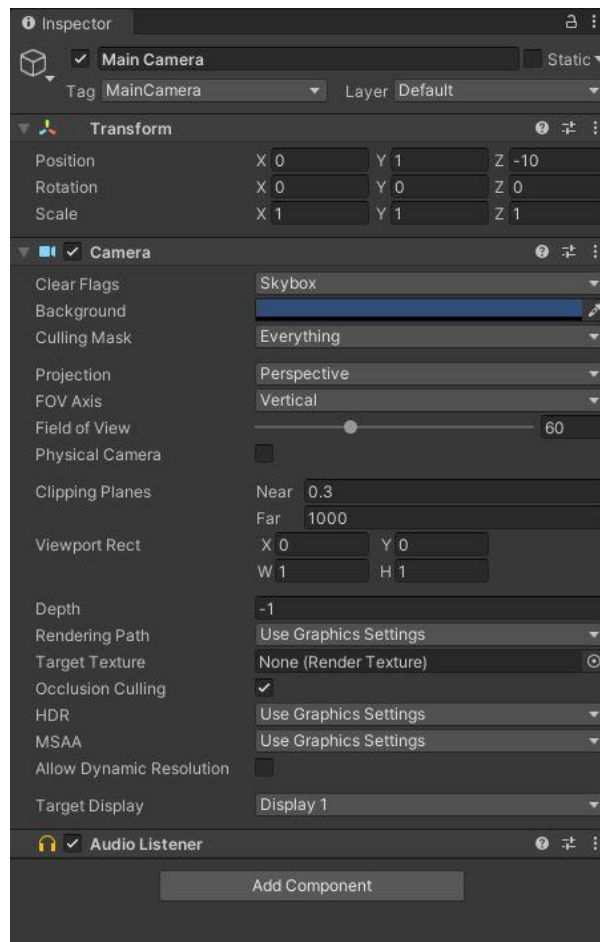
Με το που μπαίνουμε στο project μας παρατηρούμε κάποια βασικά πράγματα τα οποία αξιοποιούμε συχνά. Ο Unity Editor αποτελείται από πολλά υπο-παράθυρα [9]. Τα πιο συχνά χρησιμοποιούμενα είναι: Ο Project Browser, ο Inspector, τα παράθυρα Game View, Scene View και Hierarchy.

Ο Project Browser είναι το παράθυρο που περιέχει όλα τα στοιχεία που έχουν εισαχθεί στο Unity και είναι διαθέσιμα για χρήση (Εικόνα 2.4). Η διάταξη του είναι σχεδόν πανομοιότυπη με τον Finder στα Mac OS (Operating System) και με τον Explorer στα Windows. Αυτό κάνει τους προγραμματιστές να αισθάνονται πιο οικεία καθώς ήδη έρχονται σε επαφή με παρόμοιο περιβάλλον σε καθημερινή βάση.



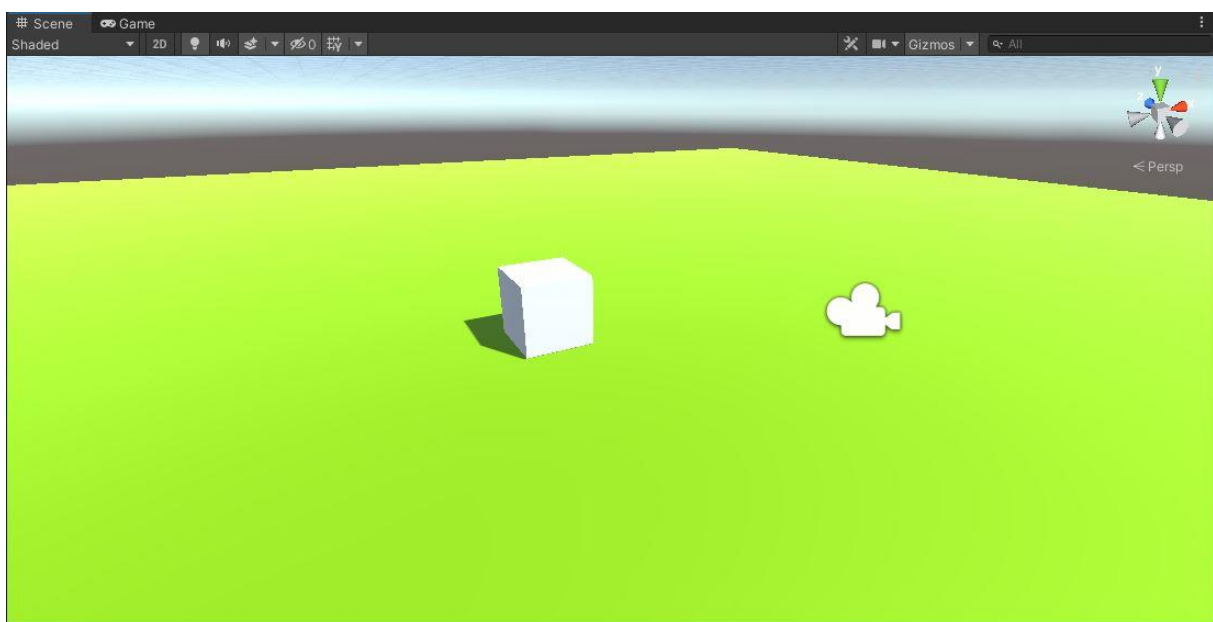
Εικόνα 2.4 Project Browser window.

Ο Inspector είναι το παράθυρο στο οποίο προβάλλονται και επεξεργάζονται τα στοιχεία (components), οι ιδιότητες (properties) και οι ιδιότητες μορφοποίησης (transform properties- τοποθεσία, rotation, scale) του κάθε αντικείμενου (Εικόνα 2.5). Εδώ μπορούμε να προσαρμόσουμε τις τιμές (values) με βάση τις προτιμήσεις σας. Τα components είναι συνδεδεμένα με ένα αντικείμενο και ο Inspector δίνει την δυνατότητα προσθήκης τους σε ένα αντικείμενο, προβολής τους και επεξεργασίας τους. Μερικά Components είναι τα Physics, Colliders και Sound. Επίσης, στο παράθυρο αυτό οι μεταβλητές που είναι εκτεθειμένες (public) από τα script της εφαρμογής, μπορούν να τροποποιηθούν.



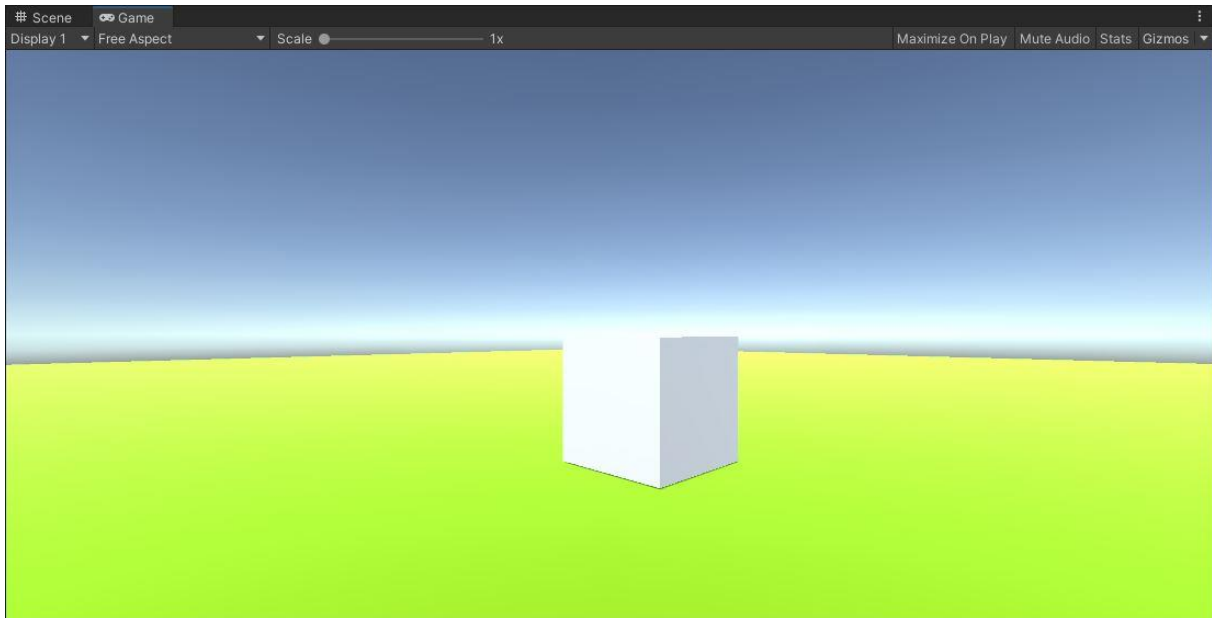
Εικόνα 2.5 Παράθυρο Inspector: Προβολή στοιχείων και ιδιοτήτων του κάθε υπό επεξεργασία αντικειμένου.

Το Game View μας παρέχει μία προεπισκόπηση της εικόνας του παιχνιδιού στον χώρο, όταν αυτό κατασκευαστεί (Εικόνα 2.6)



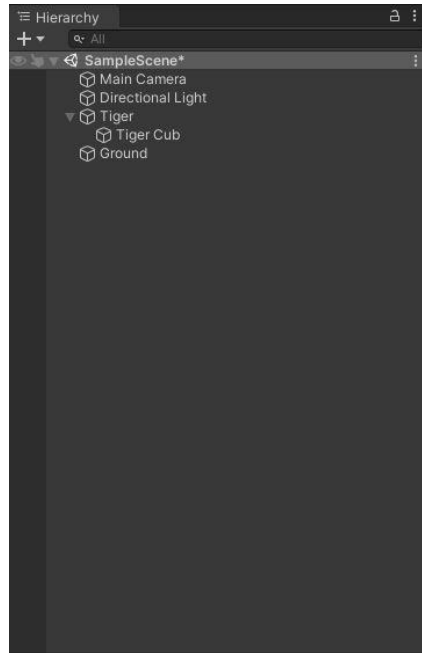
Εικόνα 2.6 Προεπισκόπηση της εικόνας του παιχνιδιού (Game View).

Το Scene View (Εικόνα 2.7) είναι το μέρος όπου κατασκευάζεται το παιχνίδι. Δίνει την δυνατότητα μεταφοράς και απόθεσης αντικειμένων (drag and drop) από το Project View. Τα τρισδιάστατα χειριστήρια που παρέχει το Scene View σας επιτρέπουν να τοποθετήσετε τα αντικείμενα σας στην τέλεια θέση, με ακρίβεια σε pixel.



Εικόνα 2.7 Κατασκευή αντικειμένων στο Scene View.

Το παράθυρο του Hierarchy (Εικόνα 2.8) περιέχει μία λίστα με όλα τα αντικείμενα που βρίσκονται στην τρέχουσα σκηνή. Αυτή η λίστα ενημερώνεται αυτόματα όταν ένα αντικείμενο εισάγεται στη σκηνή. Σε αυτό το παράθυρο, μπορούμε με drag & drop ενός αντικειμένου πάνω σε ένα άλλο να ορίσουμε μία child – parent σχέση μεταξύ των δύο αυτών αντικειμένων.

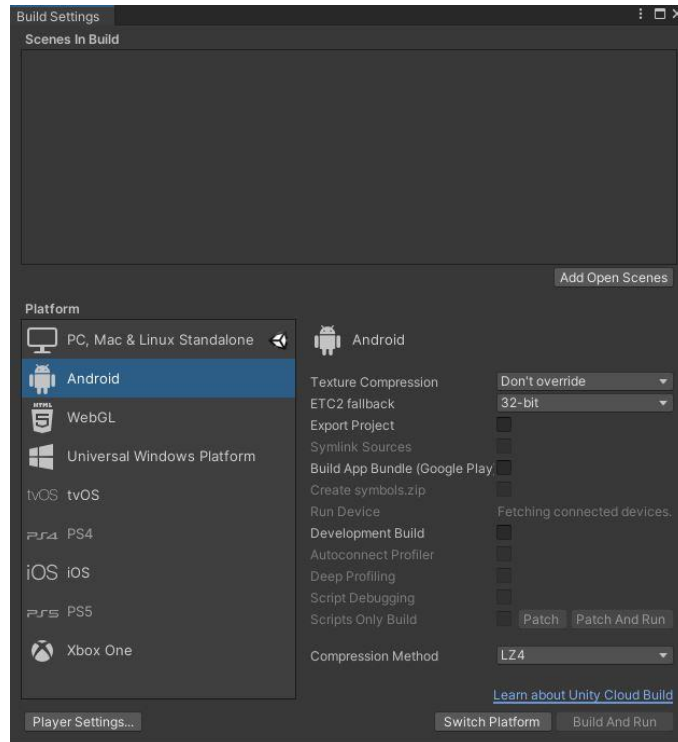


Εικόνα 2.8 Το Hierarchy window.

Όλα αυτά τα παράθυρα μπορούν να μεταφερθούν σε κάποιο άλλο μέρος του UI μέσα στο Unity με βάση τις προτιμήσεις μας.

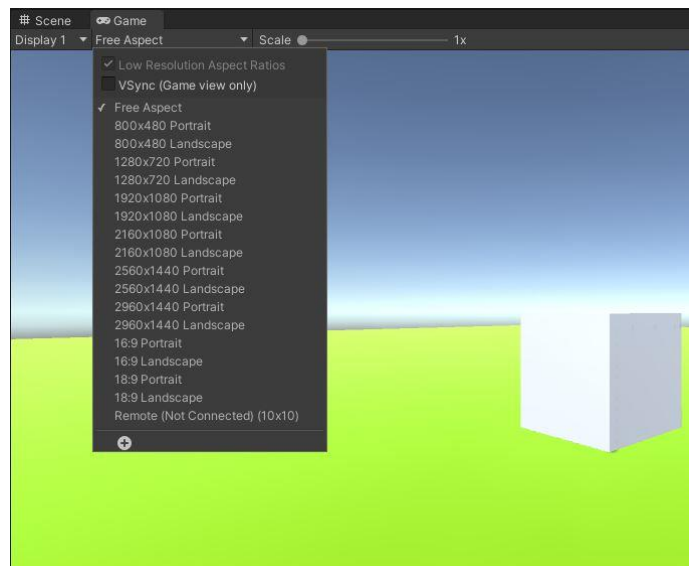
### 2.2.2 Mobile App Set Up

Προκειμένου να φτιάξουμε μία εφαρμογή για κινητά (Android) πρέπει να δημιουργήσουμε ένα βασικό Set-Up. Αρχικά πρέπει να επιλέξουμε πλατφόρμα (PC, Mac, Linux, Android, WebGL, iOS, PS5 κλπ.) στην οποία θα κάνουμε build το πρόγραμμα. Αυτό γίνεται πηγαίνοντας από το menu bar στο File, έπειτα από το drop-down menu που εμφανίζεται επιλέγουμε Build Settings (Εικόνα 2.9), στη συνέχεια επιλέγουμε Android και τέλος πατάμε το κουμπί Switch Platform.

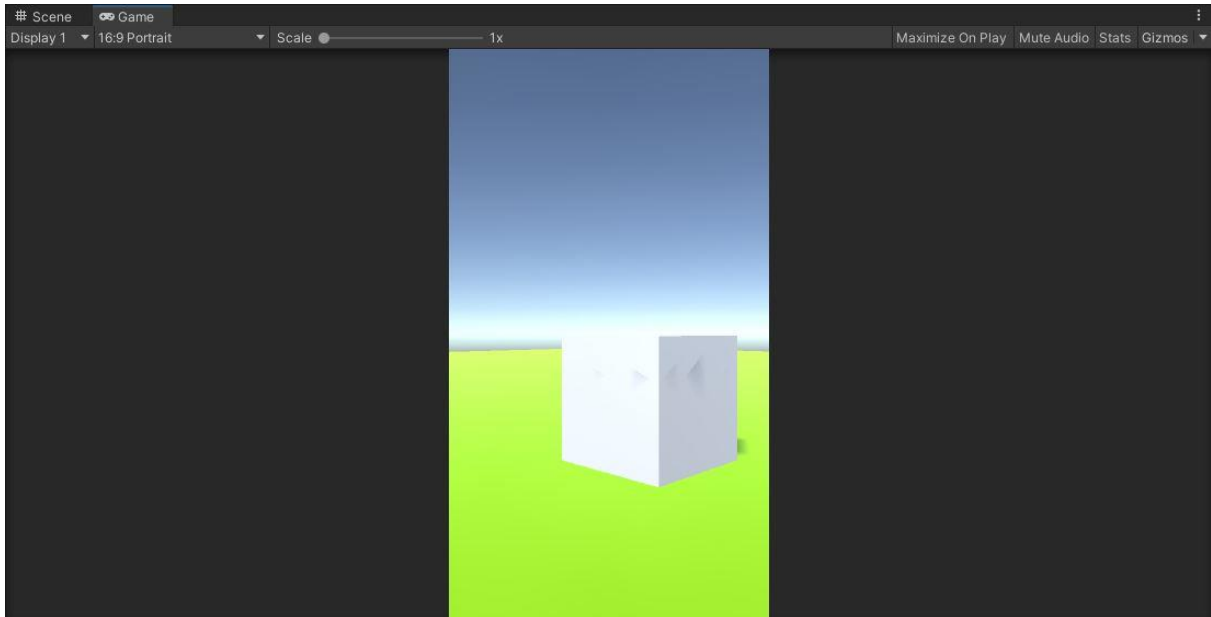


Εικόνα 2.9 Build Settings.

Επίσης μπορούμε να ορίσουμε μία ανάλυση οθόνης της επιλογής μας προκειμένου να έχουμε μία επιθυμητή αναπαράσταση με συγκεκριμένο resolution στο Game View. Η επιλογή αυτή γίνεται με βάση την ανάλυση και το aspect ratio του κινητού μας ή άλλων κινητών συσκευών (Εικόνα 2.10).



Εικόνα 2.10 Επιλογή ανάλυσης.



Εικόνα 2.11 Προεπισκόπηση της εικόνας του παιχνιδιού σε portrait mode (σε ανάλυση κινητού).

### 2.3 Vuforia

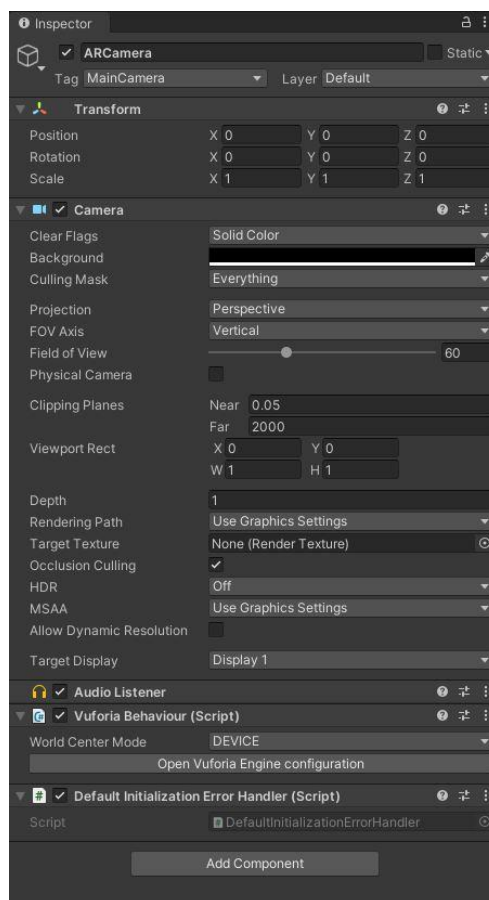


Εικόνα 2.12 Λογότυπο Vuforia.

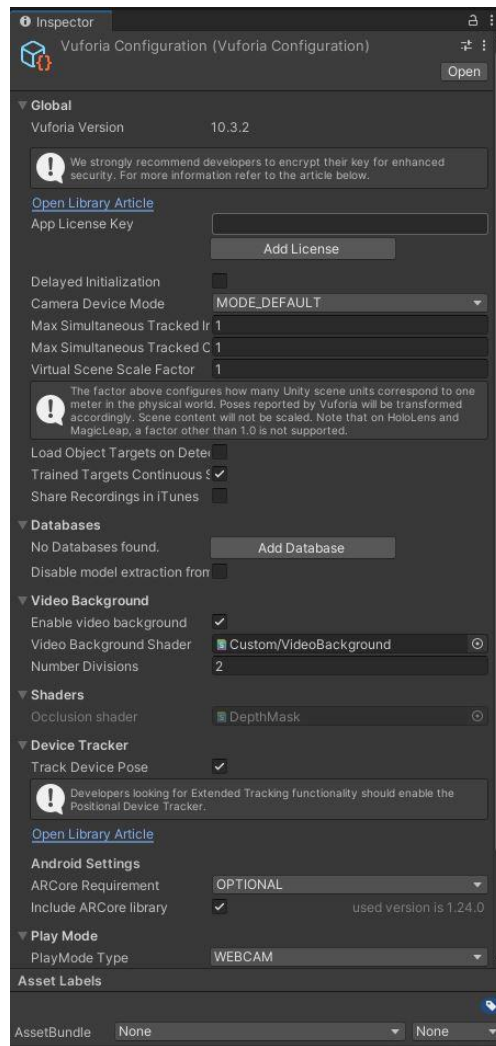
Το Vuforia AR Software Development Kit (Vuforia AR SDK) σχεδιάστηκε από την Qualcomm για εφαρμογές AR φορητών συσκευών. Η AR Virtual Camera της Vuforia αναγνωρίζει και καταγράφει επίπεδες εικόνες, μοντέλα, επιφάνειες, περιοχές ή και απλά τρισδιάστατα αντικείμενα (όπως έναν ρόμβο) έγκαιρα μέσω τεχνητής νοημοσύνης που δίνει την δυνατότητα σε έναν υπολογιστή να αναγνωρίζει συγκεκριμένα σχήματα και εικόνες. Επιπρόσθετα δίνει την δυνατότητα στους χρήστες να τοποθετούν αντικείμενα μέσα στον επαυξημένο κόσμο μέσω του φακού της κάμερας της συσκευής, προσαρμόζοντας αυτόματα τις θέσεις αυτών [11][12].

### 2.3.1 Vuforia Set-Up

Προκειμένου να χρησιμοποιήσουμε την Vuforia στο πρόγραμμά μας πρέπει πρώτα να φτιάξουμε ένα account στην ιστοσελίδα της Vuforia Engine και έπειτα να κατεβάσουμε την τελευταία έκδοση από τα Downloads. Στη συνέχεια ανοίγουμε το αρχείο που κατεβάσαμε, και έτσι πλέον έχουμε εγκαταστήσει την Vuforia μέσα στο projects μας. Δημιουργούμε μία AR Camera στη σκηνή κάνοντας δεξί click στο παράθυρο του Hierarchy και έπειτα επιλέγοντας Vuforia Engine και τέλος AR Camera (Εικόνα 2.13). Εφόσον τελειώσουμε με την προσθήκη της κάμερας στην σκηνή, κατευθυνόμαστε στην σελίδα της Vuforia και επιλέγουμε από το μενού το Develop. Από την νέα σελίδα που θα ανοίξει επιλέγουμε την επιλογή license manager και δημιουργούμε ένα νέο license (άδεια χρήσης) για την εφαρμογή μας. Αφού δημιουργήσουμε το license key (κλειδί άδειας χρήσης), το κάνουμε αντιγραφή και επικόλληση στο πρόγραμμά μας και πιο συγκεκριμένα στο text field του App License Key, το οποίο βρίσκεται στα Vuforia Configuration Options (Εικόνα 2.14) τα οποία βρίσκουμε κάτω από τον Inspector μετά την επιλογή της AR Camera της σκηνής μας.



Εικόνα 2.13 Στοιχεία και ιδιότητες του αντικειμένου AR Camera.



Εικόνα 2.14 Vuforia Configuration επιλογές.

## 2.4 Blender



Εικόνα 2.15 Λογότυπο Blender.

Το Blender είναι ένα ισχυρό εργαλείο ανοιχτού κώδικα αφιερωμένο στα τρισδιάστατα γραφικά. Τα κύρια χαρακτηριστικά του σχετίζονται με animation, φυσική (physics), σωματίδια (particles), μοντελοποίηση (modeling), απόδοση (rendering), σκίαση (shading), απεικόνιση (imaging), σύνθεση (compositing) και δημιουργία 3D παιχνιδιών σε πραγματικό χρόνο. Πολλοί επαγγελματίες και

σχεδιαστές τρισδιάστατων αντικειμένων θεωρούν πως το Blender είναι η καλύτερη λύση ανοιχτού κώδικα για την δημιουργία τρισδιάστατων γραφικών υπολογιστή [13].

Το Blender σε αυτή την πτυχιακή χρησιμοποιήθηκε για την δημιουργία όλων των τρισδιάστατων αντικειμένων των σκηνών, όπως τα αντικείμενα των πειραμάτων και εικονικού χώρου.

## 2.5 Photoshop



Εικόνα 2.16 Λογότυπο Photoshop.

Το Photoshop είναι ένα λογισμικό της Adobe για την επεξεργασία φωτογραφιών, την δημιουργία εικόνων και την σχεδίαση 3D γραφικών. Διαθέτει πάρα πολλά χρήσιμα εργαλεία τα οποία δίνουν άπειρες δυνατότητες στον χρήστη του. Κάποιες από τις πιο βασικές χρήσεις του Photoshop είναι οι εξής: Διορθώσεις φωτογραφιών, οπτικές βελτιώσεις φωτογραφιών, συνθέσεις πολλών εικόνων μαζί, προσθήκη καλλιτεχνικών εφέ και ζωγραφική ή δημιουργία εικόνων και σχεδίων από το μηδέν [14].

Το Photoshop σε αυτή την πτυχιακή χρησιμοποιήθηκε για την δημιουργία όλων των UI στοιχείων των σκηνών, όπως τα Background Images, το App Icon, το Splash Icon, τα Buttons (κουμπιά, Εικόνα 2.17) και άλλα.



Εικόνα 2.17 Button που έχει κατασκευαστεί με την χρήση του Photoshop για την συγκεκριμένη εφαρμογή.

## 2.6 Audio and Music (Bosca Ceoil)



Εικόνα 2.18 Λογότυπο Bosca Ceoil.

Το Bosca Ceoil είναι ένα δωρεάν και open source πρόγραμμα δημιουργίας μουσικής και ηχητικών εφέ, κυρίως για video games. Είναι σχεδιασμένο με ένα User Interface πολύ φιλικό προς αρχάριους αλλά και έμπειρους χρήστες. Διαθέτει ένα πολύ επεξηγηματικό tutorial το οποίο περιέχει όλες τις οδηγίες χρήσης βήμα προς βήμα. Τέλος, το Bosca έρχεται με πολλά διαφορετικά προ-εγκατεστημένα όργανα μουσικής και πολλές μελωδίες με αποτέλεσμα να διευκολύνει τον χρήστη να δημιουργήσει μουσική. Το Bosca Ceoil σε αυτή την πτυχιακή χρησιμοποιήθηκε για την δημιουργία του background music των σκηνών.

## 2.7 Τι είναι το UI και τι το UX;

Το UI είναι η αλληλεπίδραση μεταξύ ενός συστήματος και ενός χρήστη μέσω εντολών ή τεχνικών για να χειριστεί αυτός το σύστημα, να εισάγει δεδομένα και να χρησιμοποιήσει τα περιεχόμενα. Το UI μπορούμε να το συναντήσουμε σε μία πληθώρα συστημάτων, όπως υπολογιστές, κινητές συσκευές, παιχνίδια κλπ.

Το UX είναι η συνολική εμπειρία που σχετίζεται με την αντίληψη (συναίσθημα και σκέψη), αντίδραση και συμπεριφορά που ένας χρήστης αισθάνεται και λαμβάνει μέσω της άμεσης ή έμμεσης χρήσης ενός συστήματος, προϊόντος ή υπηρεσίας. Το UX είναι μία έννοια που σχετίζεται με το HCI (Human-Computer Interaction ή Επαφή Ανθρώπου-Υπολογιστή) που εφαρμόζεται ευρέως όχι μόνο στην ανάπτυξη λογισμικού και υλικού, αλλά και σε υπηρεσίες, προϊόντα, διαδικασίες, στην κοινωνία και τον πολιτισμό [15].

### 2.7.1 Χρώματα και Χρωματικές Διαφορές

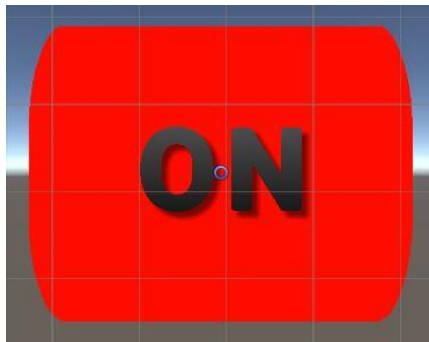
Τα χρώματα μίας εφαρμογής και οι χρωματικές διαφορές είναι ένα σημαντικό κομμάτι του UI/UX. Στην εφαρμογή της εργασίας αυτής χρησιμοποιήθηκε ένα προσαρμοσμένο θέμα της επιλογής μου, δίνοντας ένα σκοτεινό τόνο στο background και ένα πιο φωτεινό τόνο σε χρώμα πορτοκαλί στο foreground. Έτσι δημιουργείται μία χρωματική αντίθεση προκειμένου ο χρήστης να ξεχωρίζει τα αντικείμενα του background από αυτά που μπορεί να αλληλεπιδράσει στο foreground (Εικόνα 2.19). Το πορτοκαλί επιλέχθηκε ως κύριο χρώμα των button και των γραμμάτων σε σκηνές του UI, ένα χρώμα ευχάριστο στο μάτι, που δηλώνει αμέσως τον λόγο της ύπαρξής του. Τέλος, έντονα διαφορετικά χρώματα επιλέχθηκαν για διάφορα άλλα κουμπιά εντός της εφαρμογής, όπως το γαλάζιο χρώμα για το FREEZE button (ένα button που παγώνει δύο μπουκάλια που περιέχουν υγρά, Εικόνα 2.20) και το κόκκινο για το ON button (ένα button που ανάβει ένα ηλεκτρικό μάτι, Εικόνα 2.21) του πρώτου πειράματος του έβδομου κεφαλαίου όπου το γαλάζιο χρώμα είναι ένα χρώμα σχετικά ψυχρό που συνδέεται συχνά με τον πάγο σε βιντεοπαιχνίδια.



Εικόνα 2.19 Χρωματική αντίθεση button με background.



Εικόνα 2.20 Freeze Button.



Εικόνα 2.21 ON Button.

### 2.7.2 Τυπογραφία (Fonts)

Στην εφαρμογή αυτή χρησιμοποιήθηκαν δύο κύρια font styles με το βασικό να είναι το Font Style Roboto. Για την πλειοψηφία των γραμμάτων της εφαρμογής προστέθηκε επίσης ένα προσαρμοσμένο προσωπικό gradient που έφτιαξα, το οποίο δίνει μία πιο όμορφη νότα στην εφαρμογή (Εικόνα 2.22). Τα text των buttons είναι όλα σε bold προκειμένου να φαίνεται πιο καλά η λειτουργικότητα τους. Επιπρόσθετα, σε κάποια text έχει προστεθεί Underlay (σκιά πίσω από τα γράμματα) και Outline (περίγραμμα διαφορετικού χρώματος των γραμμάτων).



Εικόνα 2.22 Τύπος γραμματοσειράς της εφαρμογής.

### 2.7.3 Εικόνες προσκήνιου και φόντου

Όσον αφορά την όψη του προσκήνιου (foreground) και του φόντου (background), έγινε η εισαγωγή πολλών εικόνων για το προσκήνιο του UI και μία καθολική εικόνα για το φόντο του. Η εικόνα του φόντου και οι εικόνες που χρησιμοποιήθηκαν για τα κουμπιά φτιάχτηκαν από εμένα με την χρήση του Photoshop. Για την εικόνα του φόντου χρησιμοποιήθηκαν πολλά διαφορετικά σχήματα σχετικά με την Φυσική και έπειτα τοποθετήθηκαν με μορφή σκουρόχρωμης σκιάς σε ένα γκριζο φόντο. Για την εικόνα του φόντου ακολούθησα το χρωματικό θέμα που ορίστηκε στο UI μου, δηλαδή σκούρο στο φόντο και φωτεινό (πορτοκαλί κυρίως) στο προσκήνιο. Μερικές εικόνες του προσκήνιου που έχουν σχέση με την Φυσική τις πήρα έτοιμες από ιστοσελίδες που προσφέρουν εικόνες σε μορφές .png και .jpeg δωρεάν. Ιστοσελίδες όπως <https://pixabay.com/> και <https://unsplash.com/>.

### 2.7.4 Η επίδραση του ήχου στην εμπειρία του χρήστη

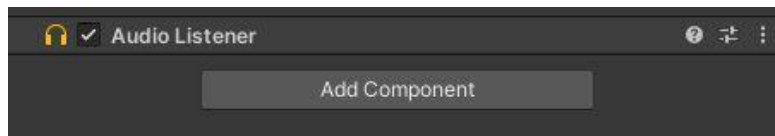
Στη σχεδίαση UX, η σχεδίαση ήχου (sound design) είναι πολύ σημαντική διότι ο ήχος ενισχύει τα οπτικά στοιχεία, παρέχοντας ανατροφοδότηση για μια ενέργεια του χρήστη ή την κατάσταση του συστήματος. Επίσης, καθιστά πιο κατανοητή μια κατάσταση, τραβάει την προσοχή του χρήστη σε σημαντικές πληροφορίες, όπως μια ηχητική προειδοποίηση, και προσδίδει προσωπικότητα στην εφαρμογή [16].

### 2.7.5 Ηχητικά Εφέ, Μουσική και Ρυθμίσεις Ήχου Εφαρμογής (Sound Clips, Sound manager and Audio Listeners)

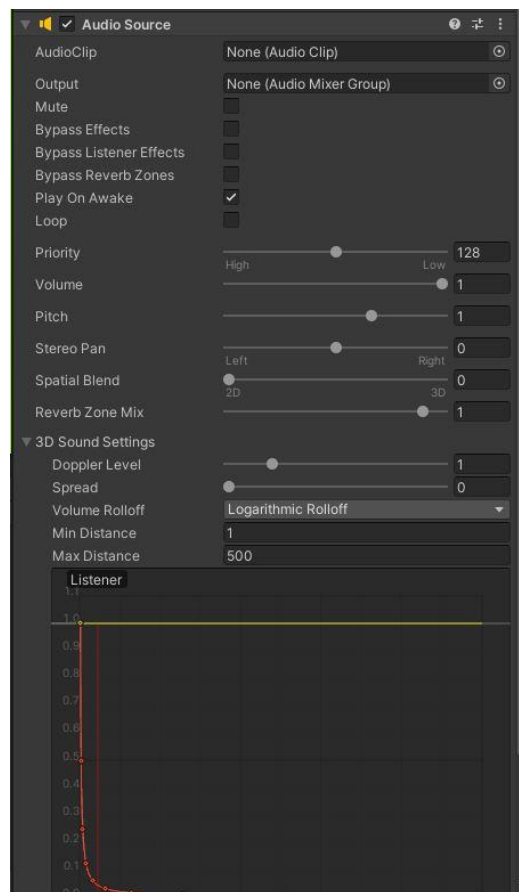
Στο πρόγραμμα της πτυχιακής εργασίας χρησιμοποιήθηκαν ηχητικά εφέ και μουσική η οποία παίζει συνεχώς στο παρασκήνιο, για την ενίσχυση της εμπειρίας που λαμβάνει ο χρήστης. Ηχητικά εφέ όπως το «κλικ» που κάνει ένα κουμπί όταν πατιέται, και μουσική όπως αυτή που παίζει στο παρασκήνιο του κεντρικού μενού.

Προκειμένου να προσθέσουμε ήχο στην εφαρμογή μας στο Unity χρειαζόμαστε κάποια βασικά στοιχεία. Αρχικά πρέπει να προσθέσουμε στην σκηνή ένα audio listener component (στοιχείο) επάνω σε ένα αντικείμενο της σκηνής μας (Εικόνα 2.23) και κατά προτίμηση σε ένα αντικείμενο το οποίο

είναι παραμένει συνεχώς ενεργό. Από το όνομα και μόνο καταλαβαίνουμε πως το στοιχείο αυτό είναι υπεύθυνο για να «ακούμε» τα audio clips που παίζουν στην σκηνή. Έπειτα, χρειάζεται να προσθέσουμε και ένα audio source component σε ένα αντικείμενο της σκηνής. Στη συνέχεια, στο πεδίο AudioClip του audio source βάζουμε το ηχητικό εφέ ή την μελωδία που θέλουμε να παίζει από αυτό το audio source (Εικόνα 2.24). Με τα στοιχεία του Audio Listener και Audio Source έτοιμα έχουμε τα βασικά εργαλεία για την παραγωγή ήχου εντός της σκηνής μας. Το κάλεσμα της ενέργειας που θα παίζει τον ήχο (audio clip) γίνεται μέσω script ή κάποιας έτοιμης μεθόδου (π.χ. OnClick Button μέθοδος).



Εικόνα 2.23 Προσθήκη Audio Listener Component.



Εικόνα 2.24 Προσθήκη Audio Source.

## 2.8 Επίλογος

Η χρήση της Unity Game Engine και της Vuforia Engine μας βοηθάει στην κατασκευή ενός λογισμικού επαυξημένης πραγματικότητας που θα αποτελέσει την βάση για την δημιουργία του παιχνιδιού. Οι εφαρμογές Blender και Photoshop εισάγουν στην βάση μας 3D και 2D αντικείμενα αντίστοιχα, τα οποία θα διαμορφώσουν το περιβάλλον με το οποίο θα αλληλεπιδράσει ο χρήστης. Το

## Κεφάλαιο 2

πρόγραμμα Bosca Ceoil προσθέτει στο περιβάλλον μουσική και ηχητικά εφέ, ολοκληρώνοντας την συνολική εμπειρία του χρήστη μαζί με επιπλέον λεπτομέρειες, όπως είναι η επιλογή γραμματοσειράς και τα χρώματα που εμφανίζονται στις 2D σκηνές.

## Κεφάλαιο 3ο: Παρουσίαση Εφαρμογής

### 3.1 Εισαγωγή

Κάθε εφαρμογή ή βιντεοπαιχνίδι έχει έναν βασικό κορμό και μια λογική πίσω από την ροή του, δύο στοιχεία τα οποία είτε γίνονται φανερά στον χρήστη απευθείας είτε τον καθοδηγούν μέσα στην εφαρμογή χωρίς να δηλώσουν την ύπαρξη τους. Ο προγραμματιστής της εφαρμογής είναι αυτός που με την χρήση σχόλιων ή πληροφοριών καθορίζει εάν αυτά τα στοιχεία θα φανερωθούν στον χρήστη ή όχι.

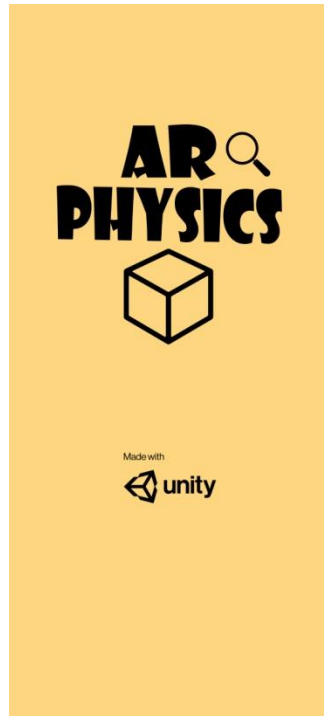
### 3.2 Αρχή Εφαρμογής

Πριν τρέξει την εφαρμογή, ο χρήστης, συναντάει τον τίτλο και το App Icon. Το App Icon φτιάχτηκε στο Photoshop σύμφωνα με τις ανάγκες της εφαρμογής. Μπορούμε να ορίσουμε ένα image ως App Icon της εφαρμογής πηγαίνοντας στα File -> Build Settings -> Player Settings -> Player -> Settings for Android -> Icon (Εικόνα 3.1).



Εικόνα 3.1 Το App Icon της Εφαρμογής.

Αφού ο χρήστης ανοίξει την εφαρμογή το πρώτο πράγμα που βλέπει είναι το Splash Screen (Εικόνα 3.2). Για Splash Screen έχουμε ορίσει ένα image το οποίο δημιουργήθηκε και αυτό στο Photoshop και αυτό. Κάτω από το Splash Screen στην δωρεάν έκδοση του Unity βλέπουμε το logo του Unity 3D.



Εικόνα 3.2 Το Splash Screen της εφαρμογής.

### 3.3 Πρώτη Εικόνα – Αρχικό Μενού

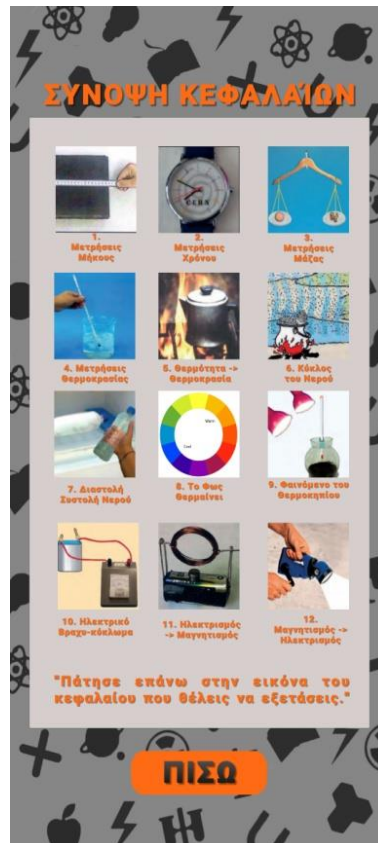
#### 3.3.1 Κεντρικό Μενού

Η πρώτη εικόνα που αντικρίζει ο χρήστης όταν εισέρχεται στην εφαρμογή είναι το Κεντρικό Μενού (Εικόνα 3.3). Το Κεντρικό Μενού (Main Menu) στην κορυφή και στο κάτω μέρος της οθόνης προσφέρει μερικές πληροφορίες όπως το όνομα της εφαρμογής, το όνομα του προγραμματιστή και την έκδοση της εφαρμογής. Επίσης περιέχει τέσσερα κουμπιά («ΠΑΙΞΕ», «ΡΥΘΜΙΣΕΙΣ», «ΕΞΟΔΟΣ», «Info») που το καθένα καθοδηγεί τον χρήστη στην ανάλογη σελίδα. Τα κουμπιά είναι μεγάλα και τα γράμματα μέσα τους ευδιάκριτα.



Εικόνα 3.3 Το Κεντρικό Μενού.

Με το κλικ του κουμπιού «ΠΑΙΞΕ» ο χρήστης καθοδηγείται στην σελίδα «Σύνοψη Κεφαλαίων», όπου διαλέγει ένα Κεφάλαιο του οποίου τα πειράματα θέλει να υλοποιήσει. Η επιλογή αυτή γίνεται με ένα κλικ σε ένα από τα δώδεκα διαφορετικά κουμπιά της σελίδας, τα οποία δηλώνουν το περιεχόμενο του κάθε κεφαλαίου με έναν τίτλο που βρίσκεται ακριβώς κάτω από το κάθε κουμπί (Εικόνα 3.4). Επιπρόσθετα, τα κουμπιά δηλώνουν το περιεχόμενό τους με την εικόνα από την οποία αποτελούνται.



Εικόνα 3.4 Η Σελίδα Σύνοψης Κεφαλαίων.

Κάνοντας κλικ πάνω σε ένα Κεφάλαιο, ο χρήστης καθοδηγείται στην σελίδα «Σύνοψη Πειραμάτων» στην οποία προβάλλονται τα διαθέσιμα πειράματα προς επίλυση. Επίσης εμφανίζεται η σελίδα του βιβλίου της Φυσικής στην οποία βρίσκεται η εικόνα που πρέπει να σκανάρει ο χρήστης προκειμένου να υλοποιήσει το αντίστοιχο πείραμα. Σχεδόν σε όλες τις σελίδες της εφαρμογής ο χρήστης έχει την επιλογή να γυρίσει στην προηγούμενη σκηνή που βρισκόταν πατώντας το κουμπί «ΠΙΣΩ». Η μεταφορά του χρήστη από σκηνή σε σκηνή επιτυγχάνεται με το script MainMenu.cs (Εικόνα 3.5). Η κάθε μέθοδος του script κατευθύνει τον χρήστη σε διαφορετική σκηνή.

```

Assembly-CSharp - MainMenu
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6
7 public class MainMenu : MonoBehaviour
8 {
9     Unity Message | 0 references
10    private void Awake()
11    {
12        QualitySettings.vSyncCount = 0;
13        Application.targetFrameRate = 30;
14    }
15    // general next scene
16    0 references
17    public void NextScene()
18    {
19        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
20    }
21    //apo FirstScan se FindHelp
22    0 references
23    public void VresTaAntikeimenaKef1pirama1()
24    {
25        SceneManager.LoadScene("Kef1pirama1FindHelp");
26    }
27    0 references
28    public void VresTaAntikeimenaKef2pirama1()
29    {
30        SceneManager.LoadScene("Kef2pirama1FindHelp");
31    }
32 }

```

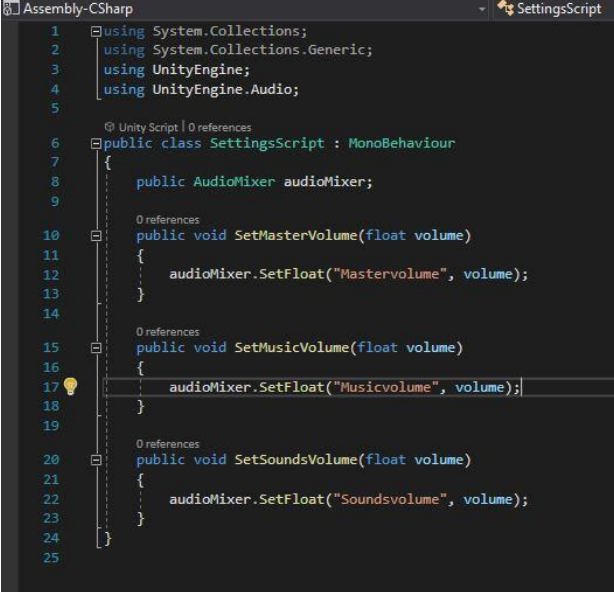
Εικόνα 3.5 Το MainMenu script που ελέγχει όλες τις συνδέσεις των σκηνών του προγράμματος.

### 3.3.2 Ρυθμίσεις

Μία άλλη επιλογή που έχει ο χρήστης από το κεντρικό μενού είναι η επιλογή των Ρυθμίσεων. Ο χρήστης μπορεί να κατευθυνθεί στην σελίδα των ρυθμίσεων πατώντας το κουμπί «ΡΥΘΜΙΣΕΙΣ». Εκεί έχει την επιλογή να αλλάξει την ένταση της μουσικής και την ένταση των ηχητικών εφέ ξεχωριστά (Εικόνα 3.6). Αυτό γίνεται εφικτό με την χρήση δύο διαφορετικών slider σε συνδυασμό με ένα script το οποίο συνδέει έναν audiomixer με τα δυο sliders της σελίδας (Εικόνα 3.7).



Εικόνα 3.6 Η σελίδα των Ρυθμίσεων.



```

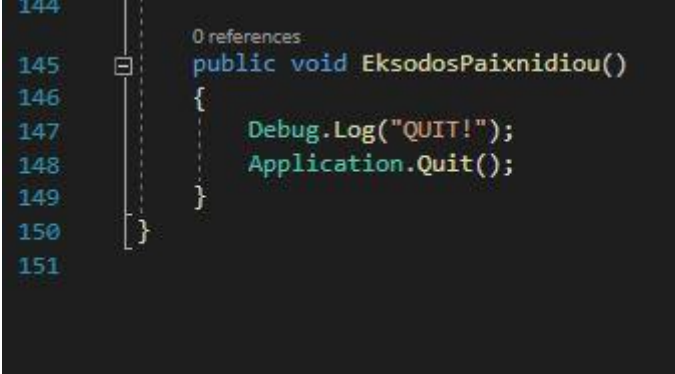
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.Audio;
5
6  public class SettingsScript : MonoBehaviour
7  {
8      public AudioMixer audioMixer;
9
10     public void SetMasterVolume(float volume)
11     {
12         audioMixer.SetFloat("Mastervolume", volume);
13     }
14
15     public void SetMusicVolume(float volume)
16     {
17         audioMixer.SetFloat("Musicvolume", volume);
18     }
19
20     public void SetSoundsVolume(float volume)
21     {
22         audioMixer.SetFloat("Soundsvolume", volume);
23     }
24 }
25

```

Εικόνα 3.7 Το script SettingScript που ελέγχει όλες τις ενέργειες των Ρυθμίσεων.

### 3.3.3 Έξοδος και Πληροφορίες

Οι τελευταίες επιλογές που δίνονται στον χρήστη μέσω του αρχικού μενού είναι τα δύο κουμπιά του «ΕΞΟΔΟΣ» και «Info» (Εικόνα 3.8). Με το κουμπί «ΕΞΟΔΟΣ» δίνεται στον χρήστη η δυνατότητα να κλείσει την εφαρμογή. Φυσικά αφού η εφαρμογή είναι σε Android ο χρήστης μπορεί να βγει από οποιαδήποτε εφαρμογή μέσω του home button του κινητού του αλλά η επιλογή του κουμπιού «ΕΞΟΔΟΣ» προσδίδει στην εφαρμογή έναν χαρακτήρα που θυμίζει βιντεοπαιχνίδι.



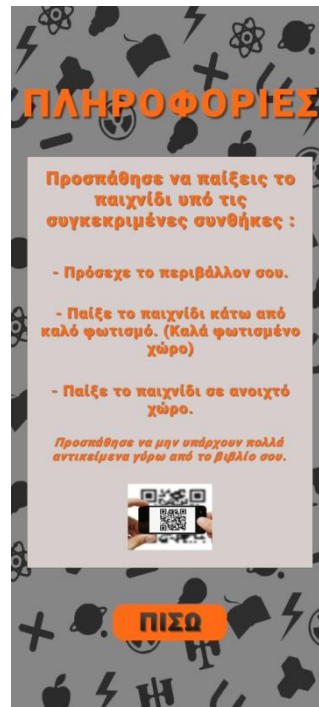
```

144
145     public void EksodosPaixnidou()
146     {
147         Debug.Log("QUIT!");
148         Application.Quit();
149     }
150 }
151

```

Εικόνα 3.8 Η μέθοδος εξόδου από την εφαρμογή.

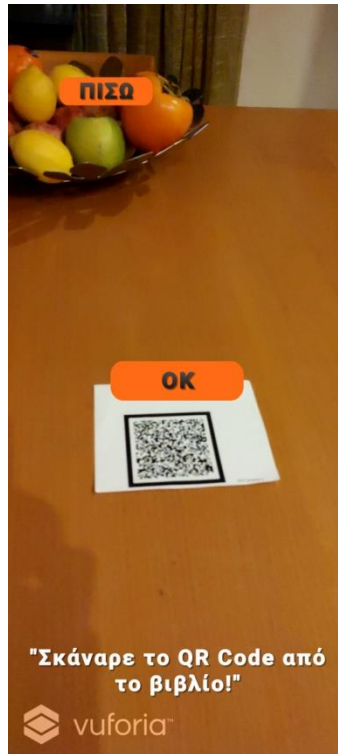
Τελευταία επιλογή που δίνεται στον χρήστη είναι το κουμπί «Info». Κάνοντας κλικ στο κουμπί αυτό ο χρήστης πηγαίνει σε μία σελίδα που περιέχει γενικές χρήσιμες πληροφορίες σχετικά με το πώς πρέπει να χρησιμοποιηθεί η εφαρμογή σωστά (Εικόνα 3.9).



Εικόνα 3.9 Η σελίδα πληροφοριών.

### 3.4 Βασικός Κορμός Πειράματος

Από την στιγμή που ο χρήστης πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ» στην σελίδα σύνοψης πειραμάτων ενός κεφαλαίου, πηγαίνει στην σκηνή FirstScanCodeScene. Στην σκηνή αυτή ενεργοποιείται η κάμερα του κινητού και η εφαρμογή ζητάει από τον χρήστη να σκανάρει μία εικόνα στόχο (target image). Αναλόγως τον στόχο που θα σκανάρει ο χρήστης, η εφαρμογή τον πηγαίνει στην αντίστοιχη σκηνή. Αφού ο χρήστης σκανάρει την εικόνα στόχο, εμφανίζεται ένα κουμπί «ΟΚ» το οποίο πατάει και πηγαίνει στην επόμενη σκηνή (Εικόνα 3.10).



Εικόνα 3.10 Η σκηνή FirstScanCodeScene.

Στη συνέχεια, ο χρήστης μεταφέρεται μεταξύ δισδιάστατων και τρισδιάστατων σκηνών με την χρήση της κάμερας. Στις δισδιάστατες σκηνές η εφαρμογή προβάλλει στον χρήστη τις απαραίτητες πληροφορίες σχετικά με την επόμενη τρισδιάστατη σκηνή.

Στην πρώτη τρισδιάστατη σκηνή που συναντάει ο χρήστης, όταν «διαβάσει» η κάμερα του κινητού την εικόνα στόχο, εμφανίζεται ένας εικονικός χώρος σε μορφή portal (πύλη) στον οποίο ο χρήστης πρέπει να παίζει ένα μικρό παιχνίδι. Το παιχνίδι είναι απλό, ο χρήστης πρέπει κουνώντας την συσκευή του, να εισέλθει στον εικονικό χώρο μέσω της πύλης και να μαζέψει τα αντικείμενα του δωματίου που ζητούνται από αυτόν (Εικόνα 3.11).



Εικόνα 3.11 Μία τρισδιάστατη σκηνή ανάγνωσης αντικειμένων.

Η δεύτερη τρισδιάστατη σκηνή που συναντάει ο χρήστης είναι η υλοποίηση του πειράματος που διάλεξε (Εικόνα 3.12). Το κάθε πείραμα είναι διαφορετικό άρα διαφορετική είναι και η υλοποίηση του. Οι υλοποιήσεις των πειραμάτων θα εξηγηθούν στα κεφάλαια πέντε με δεκαπέντε αναλυτικά. Εφόσον ο χρήστης έχει υλοποιήσει το πείραμα που διάλεξε, η εφαρμογή δίνει στον χρήστη την επιλογή να πάει στην επόμενη σκηνή μέσω του κουμπιού «ΤΕΛΟΣ» που εμφανίζεται στην οθόνη.



Εικόνα 3.12 Μία τρισδιάστατη σκηνή υλοποίησης πειράματος.

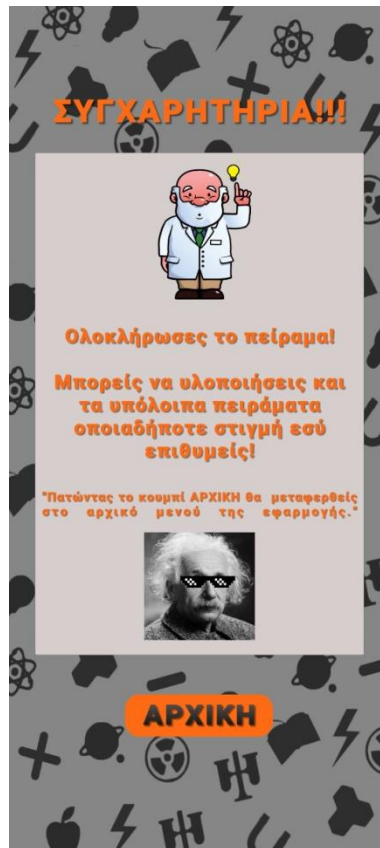
Στην διδιάστατη και τελευταία σκηνή του πειράματος ο χρήστης έρχεται αντιμέτωπος με αποτελέσματα και ερωτήσεις σχετικά με το πείραμα που υλοποίησε στον επαυξημένο κόσμο (Εικόνα 3.13).



Εικόνα 3.13 Μία δισδιάστατη σκηνή αποτελεσμάτων.

### 3.5 Τελευταία Εικόνα – Σκηνή Τέλους

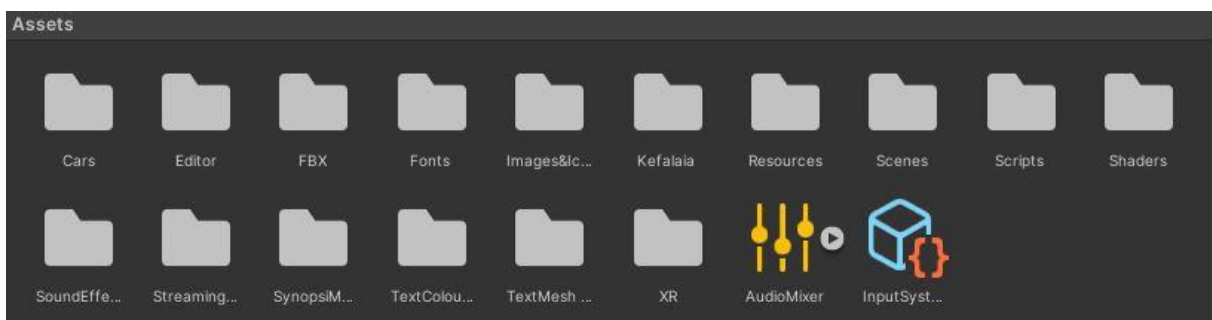
Η EndScene είναι η σκηνή που εμφανίζεται αφού ο χρήστης πατήσει το κουμπί «ΤΕΛΟΣ» σε μία σκηνή αποτελεσμάτων μετά την υλοποίηση ενός πειράματος. Στην σκηνή αυτή ο χρήστης πηγαίνει από όλα τα πειράματα και είναι το τελικό στάδιο της εφαρμογής. Από εκεί δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί «ΑΡΧΙΚΗ» και να γυρίσει πίσω στο αρχικό μενού, από το οποίο μπορεί να συνεχίσει με την υλοποίηση κάποιου άλλου πειράματος (Εικόνα 3.14).



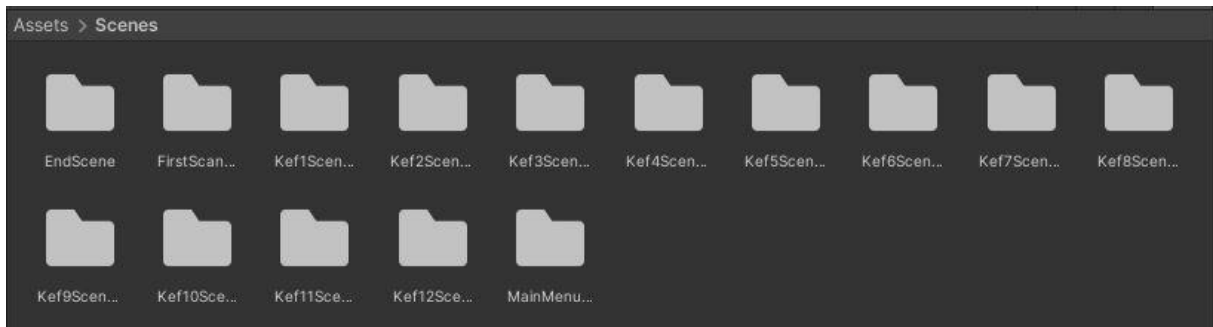
Εικόνα 3.14 Η δισδιάστατη σκηνή τέλους.

### 3.6 Βασικός κορμός των Assets

Ένα Unity Project αποτελείται από πολλά Assets (στοιχεία). Τα στοιχεία αυτά είναι οτιδήποτε περιέχεται μέσα στο project, οι σκηνές, τα τρισδιάστατα αντικείμενα, οι εικόνες, οι ήχοι, τα scripts και άλλα (Εικόνα 3.15). Στον φάκελο των assets της εφαρμογής της πτυχιακής αυτής, έχουν δημιουργηθεί πολλοί υπό-φάκελοι που περιέχουν αντικείμενα ή και άλλους υπό-φακέλους για να υπάρχει μία σωστή οργάνωση (Εικόνα 3.16). Κάποιοι από τους φακέλους αυτούς είναι οι φάκελοι των Scenes, FBX, Images&Icons, Scripts, SoundEffects. Οι φάκελοι αυτοί περιέχουν τις σκηνές, τα τρισδιάστατα αντικείμενα, τις εικόνες, τα script και τα ηχητικά εφέ της εφαρμογής αντίστοιχα.



Εικόνα 3.15 Τα Assets της εφαρμογής όπως φαίνονται στο Project Browser Window.



Εικόνα 3.16 Τα περιεχόμενα του φακέλου «Scenes» της εφαρμογής όπως φαίνονται στο Project Browser Window.

### 3.7 Επίλογος

Από την αρχή της εφαρμογής μέχρι την σκηνή του τέλους ενός πειράματος ο χρήστης πρέπει να περάσει από κάποιες συγκεκριμένες σκηνές με προκαθορισμένη σειρά. Ο χρήστης κατευθύνεται από το αρχικό μενού , στην σύνοψη κεφαλαίων, από εκεί στην σύνοψη πειραμάτων και έπειτα στην δισδιάστατη σκηνή επεξήγησης εύρεσης των αντικειμένων στον εικονικό χώρο. Στη συνέχεια περνάει στη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας εύρεσης των αντικειμένων, μετά στην δισδιάστατη σκηνή επεξήγησης υλοποίησης του πειράματος και έπειτα στη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας υλοποίησης του πειράματος. Τέλος, καταλήγει στη δισδιάστατη σκηνή αποτελεσμάτων του πειράματος και στη σκηνή τέλους και ευχαριστιών. Αυτός είναι ο βασικός κορμός που ακολουθεί ο χρήστης εντός της εφαρμογής (Εικόνα 3.17).



Εικόνα 3.17 Ο βασικός κορμός – λογική του παιχνιδιού.



## Κεφάλαιο 4ο: Μετρήσεις Μήκους – Η Μέση Τιμή.

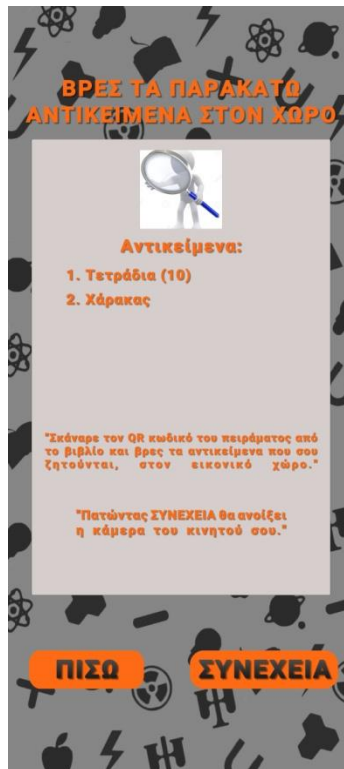
### 4.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την μέτρηση μήκους με την χρήση τετραδίων και χάρακα. Στο συγκεκριμένο πείραμα ο μαθητής χρησιμοποιεί τον χάρακα για την μέτρηση του μήκους δέκα τετραδίων και στην συνέχεια υπολογίζει την μέση τιμή του μήκους των τετραδίων αυτών.

### 4.2 Κατασκευή του πειράματος

Για λόγους ευκολίας χωρίζουμε το πείραμα σε 5 Φάσεις. Η πρώτη φάση (Φάση 1) του πειράματος είναι μία διδιάστατη σκηνή επεξήγησης, στη συνέχεια ακολουθεί μία τρισδιάστατη σκηνή επαυξημένης πραγματικότητας (Φάση 2), έπειτα μία ακόμη διδιάστατη σκηνή επεξήγησης (Φάση 3), επιπρόσθετα μία τρισδιάστατη σκηνή επαυξημένης πραγματικότητας (Φάση 4) και τέλος μία διδιάστατη σκηνή παρουσίασης αποτελεσμάτων και κατανόησης του πειράματος (Φάση 5).

Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια και πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει δέκα τετράδια και έναν χάρακα. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 4.1).



Εικόνα 4.1 Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 1<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού (Εικόνα 4.2). Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο δέκα τετράδια και έναν χάρακα. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή.



Εικόνα 4.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 1<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός script γραμμένο σε C#. Σε κάθε C# script του πειράματος πρέπει στην αρχή να κάνουμε τις απαραίτητες εισαγωγές βιβλιοθηκών προκειμένου να έχουμε όλα τα χρήσιμα εργαλεία του κώδικα στην κατοχή μας (Εικόνα 4.3). Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα DemoRaycast.cs. Πρώτα, στο script αρχικοποιούμε τις τιμές τριών μεταβλητών τύπου string που θα χρησιμοποιηθούν ως tags πάνω σε αντικείμενα, δύο μεταβλητών TextMeshProUGUI που αποθηκεύουν τα αντικείμενα των text της σκηνής και δύο integer μεταβλητών που χρησιμοποιούνται ως μετρητές (Εικόνα 4.4).

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using System;
6  using TMPro;
7  using UnityEngine.SceneManagement;
8

```

Εικόνα 4.3 Εισαγωγή βιβλιοθηκών στον κώδικα.

```

Unity Script | 0 references
public class DemoRaycast : MonoBehaviour
{
    //selection----
    private Transform _selection;

    //tags ----
    //[SerializeField] private string selectableTag = "Selectable";
    [SerializeField] private string xarakasTag = "Xarakas";
    [SerializeField] private string tetradioTag = "Tetradio";
    [SerializeField] private string tetradio4Tag = "Tetradio4";

    //counters ----
    private TextMeshProUGUI countertextxarakas;
    private TextMeshProUGUI countertexttetradio;
    public int xarakascounter = 0;
    public int tetradiocounter = 0;
}

```

Εικόνα 4.4 Αρχικοποίηση μεταβλητών του script DemoRaycast.

Χρησιμοποιώντας την μέθοδο Awake() τύπου void απενεργοποιούμε το V-Sync της εφαρμογής και ορίζουμε το μέγιστο frame rate σε 30 frames per second (Εικόνα 4.5). Αυτό το κάνουμε προκειμένου να έχουμε ένα «ταβάνι» στα frames που μπορεί να δείξει το κινητό κάθε δευτερόλεπτο, επιτυγχάνοντας έτσι χαμηλότερες θερμοκρασίες στο σύστημα – κινητό μας. Η αλλαγή αυτή στα frames και στο V-Sync, γίνεται σε όλα τα script της εφαρμογής στην αντίστοιχη Awake() μέθοδο.

```

Unity Message | 0 references
private void Awake()
{
    QualitySettings.vSyncCount = 0;
    Application.targetFrameRate = 30;
}

```

Εικόνα 4.5 Η μέθοδος Awake() του script DemoRaycast.

Έπειτα, στην μέθοδο Start() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου των αντικειμένων τύπου text της σκηνής (Εικόνα 4.6). Η μεταβλητή countertextxarakas και countertexttetradio παίρνουν τα στοιχεία του text αντικειμένου της σκηνής, με την χρήση της μεθόδου GameObject.Find(«όνομα αντικειμένου τύπου text»).GetComponent<TextMeshProUGUI>(). Στο κείμενο των αντικειμένων αυτών αναφερόμαστε με την .text κατάληξη όπως π.χ. countertextxarakas.text. Έτσι στη συγκεκριμένη περίπτωση το text του αντικειμένου xarakasText αλλάζει από την αρχική του τιμή σε “Χάρακας : 0/1”, εφόσον η αρχική τιμή της μεταβλητής xarakascounter είναι 0.

```

Unity Message | 0 references
void Start()
{
    countertextxarakas = GameObject.Find("xarakasText").GetComponent<TextMeshProUGUI>();
    countertextxarakas.text = "Χάρακας : " + xarakascounter + "/1";

    countertexttetradio = GameObject.Find("tetradioText").GetComponent<TextMeshProUGUI>();
    countertexttetradio.text = "Τετράδια : " + tetradiocounter + "/10";
}

```

Εικόνα 4.6 Η μέθοδος Start() του script DemoRaycast.

Στη συνέχεια, στην μέθοδο Update() τύπου void χρησιμοποιούμε μία if η οποία περιμένει από τον χρήστη το πάτημα του MouseButton(0), δηλαδή το αριστερό click του ποντικιού του χρήστη ή στην περίπτωση μας το πάτημα της οθόνης (Εικόνα 4.7). Ο έλεγχος αυτός γίνεται συνεχόμενα καθ'όλη την διάρκεια ζωής του script καθώς η Update function στο Unity καλείται συνεχώς με βάση το framerate της συσκευής μας. Μέσα στην if γίνεται η αρχικοποίηση μιας μεταβλητής τύπου Ray η οποία αναγνωρίζει και παίρνει την θέση του κέρσορα μας. Στη συγκεκριμένη περίπτωση, εφόσον πρόκειται για μία εφαρμογή σε Android, μιλάμε για την θέση του δάχτυλού μας πάνω στην οθόνη του κινητού. Έπειτα έχουμε μία εμφωλευμένη if που ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη εμφωλευμένη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε κλικ είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project.

Το Tag είναι μία μορφή ενός αναγνωριστικού στοιχείου αντικειμένων. Tag value είναι η τύπου string τιμή ενός tag. Tag στο Unity μπορούμε να ορίσουμε επιλέγοντας ένα αντικείμενο της σκηνής μας από το Hierarchy window. Έπειτα από το Inspector Window στο πεδίο Tag αλλάζουμε την τιμή του από “Untagged” στο αντίστοιχο Tag που επιθυμούμε (Εικόνα 4.8). Για την δημιουργία ενός νέου Tag επιλέγουμε «Add Tag...». Ανάλογα λοιπόν με το Tag που θα «διαβάσει» το script, ο κώδικας «εισέρχεται» στην ανάλογη if. Εκεί καταστρέφει το αντικείμενο στο οποίο κάναμε κλικ, με την μέθοδο GameObject.DestroyObject, ανεβάζει τον αντίστοιχο μετρητή του αντικειμένου κατά μία μονάδα και αλλάζει δυναμικά το ανάλογο text που πρέπει, προκειμένου να ξέρει ο χρήστης πόσα αντικείμενα έχει συλλέξει.

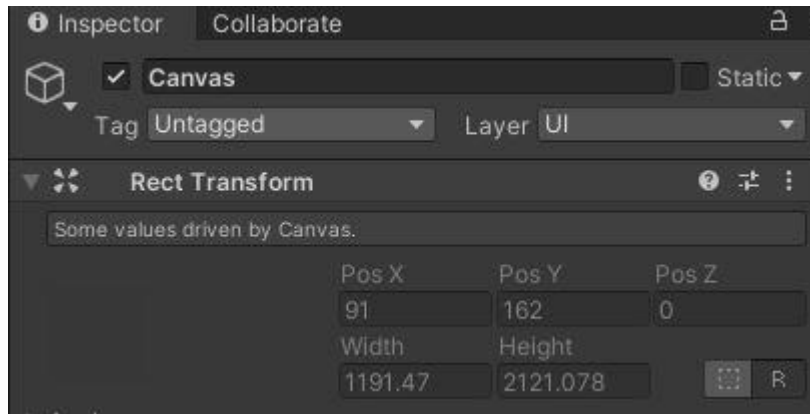
```

if (Input.GetMouseButtonDown(0))
{
    Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
    RaycastHit hit;

    if (Physics.Raycast(ray, out hit))
    {
        var selection = hit.transform;
        if (selection.CompareTag(tetradioTag)){
            GameObject.DestroyObject(hit.transform.gameObject);
            tetradiocounter = tetradiocounter + 1;
            countertexttetradio.text = "Τετράδια : " + tetradiocounter + "/10";
        }
        else if (selection.CompareTag(tetradiox4Tag))
        {
            GameObject.DestroyObject(hit.transform.gameObject);
            tetradiocounter = tetradiocounter + 4;
            countertexttetradio.text = "Τετράδια : " + tetradiocounter + "/10";
        }
        else if (selection.CompareTag(xarakasTag))
        {
            GameObject.DestroyObject(hit.transform.gameObject);
            xarakascounter = xarakascounter + 1;
            countertextxarakas.text = "Χάρακας : " + xarakascounter + "/1";
        }
    }
}

```

Εικόνα 4.7 Η μέθοδος Update() του script DemoRaycast.



Εικόνα 4.8 Το πεδίο Tag στο Inspector Window.

Τέλος, το script ελέγχει συνεχώς μέσα στην μέθοδο Update(), με την χρήση εμφωλευμένων if εάν ο χρήστης έχει επιλέξει όλα τα απαραίτητα αντικείμενα προκειμένου να προχωρήσει η εφαρμογή στην επόμενη σκηνή. Ελέγχει δηλαδή εάν οι μετρητές του script έχουν πάρει τις τιμές που πρέπει. Εφόσον αυτό έχει συμβεί, τότε με την χρήση του SceneManager η εφαρμογή περνάει στην επόμενη σκηνή (Εικόνα 4.9).

```

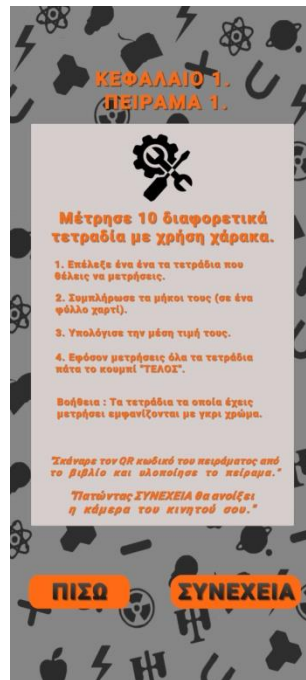
if (xarakascounter == 1)
{
    if (tetradiocounter == 10)
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }
}

```

Εικόνα 4.9 Ο έλεγχος που οδηγεί στην φόρτωση της επόμενης σκηνής.

Όλες οι ενέργειες που εκτελούνται μέσα στο DemoRaycast.cs script, συμβαίνουν και σε όλα τα υπόλοιπα παρόμοια script που υλοποιούν την δεύτερη φάση (Φάση 2) των επόμενων πειραμάτων. Η λογική των μεθόδων και οι λειτουργίες που υλοποιούν παραμένουν σε κάθε script οι ίδιες. Το μόνο που διαφέρει είναι οι ονομασίες των μεταβλητών, οι ονομασίες των αντίστοιχων text στοιχείων και το πλήθος τους.

Στην επόμενη σκηνή (Φάση 3) η εφαρμογή εξηγεί συνοπτικά στον χρήστη, τα βήματα που πρέπει να ακολουθήσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να υλοποιήσει το πείραμα. Επίσης δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 4.10). Αυτή η σκηνή επεξήγησης χρησιμοποιείται στην Φάση 3 όλων των πειραμάτων της εφαρμογής και μεταβάλλεται ανάλογα με τις ανάγκες του πειράματος.



Εικόνα 4.10 Η σκηνή της τρίτης φάσης του πειράματος του 1<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή - πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να επιλέξει ένα ένα τα τετράδια πάνω από αυτό. Με την επιλογή ενός τετραδίου, αυτό μεταφέρετε αυτόματα μπροστά στον χρήστη και ακριβώς μπροστά από τον χάρακα. Ο χρήστης σε αυτή την φάση μπορεί να μετρήσει το μήκος του τετραδίου που έχει επιλέξει και να καταγράψει το μήκος του. Επιλέγοντας ένα νέο τετράδιο, αυτό έρχεται μπροστά και το παλιό επιστρέφει στην αρχική του θέση. Τα τετράδια που έχουν ήδη επιλεγεί μία φορά αλλάζουν χρώμα κι από πορτοκαλί γίνονται γκρι. Τέλος, στο κάτω μέρος της οθόνης έχει τοποθετηθεί ένα text το οποίο αλλάζει δυναμικά και δηλώνει τον αριθμό του επιλεγμένου τετραδίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα Kef1Exp1.cs. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές μίας μεταβλητής τύπου float που θα χρησιμοποιηθεί ως μετρητής, μίας μεταβλητής integer που θα δηλώνει τον αριθμό του τετραδίου που είναι επιλεγμένο την δεδομένη χρονική στιγμή, δέκα μεταβλητών τύπου string που θα χρησιμοποιηθούν ως tags πάνω στα τετράδια και έντεκα μεταβλητές τύπου GameObject που θα κρατάνε τα δεδομένα των δέκα τετραδίων και ενός button. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου TextMeshProUGUI που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά μέσα στη σκηνή (Εικόνα 4.11).

```

public class Kef1Exp1 : MonoBehaviour
{
    //selection---
    private Transform selection;
    public float totalselected = 0f;
    public int selecteditemnumber = 0;

    //tags ---
    //[SerializeField] private string selectableTag = "Selectable";
    [SerializeField] private string tetradio1Exp1Tag = "Tetradio1Exp1";
    [SerializeField] private string tetradio2Exp1Tag = "Tetradio2Exp1";
    [SerializeField] private string tetradio3Exp1Tag = "Tetradio3Exp1";
    [SerializeField] private string tetradio4Exp1Tag = "Tetradio4Exp1";
    [SerializeField] private string tetradio5Exp1Tag = "Tetradio5Exp1";
    [SerializeField] private string tetradio6Exp1Tag = "Tetradio6Exp1";
    [SerializeField] private string tetradio7Exp1Tag = "Tetradio7Exp1";
    [SerializeField] private string tetradio8Exp1Tag = "Tetradio8Exp1";
    [SerializeField] private string tetradio9Exp1Tag = "Tetradio9Exp1";
    [SerializeField] private string tetradio10Exp1Tag = "Tetradio10Exp1";

    GameObject tetradio1;
    GameObject tetradio2;
    GameObject tetradio3;
    GameObject tetradio4;
    GameObject tetradio5;
    GameObject tetradio6;
    GameObject tetradio7;
    GameObject tetradio8;
    GameObject tetradio9;
    GameObject tetradio10;

    public GameObject TelosButton;

    private TextMeshProUGUI selecteditemnumbertext;
}

```

Εικόνα 4.11 Αρχικοποίηση μεταβλητών του script Kef1Exp1.

Στην συνέχεια, στην μέθοδο Start() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου των αντικειμένων τύπου text της σκηνής. Η μεταβλητή selecteditemnumbertext παίρνει τα στοιχεία του text αντικειμένου της σκηνής, με την χρήση της μεθόδου:

```
GameObject.Find(«όνομα αντικειμένου τύπου text»).GetComponent<TextMeshProUGUI>().
```

Στο κείμενο του αντικειμένου αυτού αναφερόμαστε με την .text κατάληξη στα αντικείμενα όπως π.χ. selecteditemnumbertext.text. Έτσι στη συγκεκριμένη περίπτωση το text του αντικειμένου selecteditemnumbertext αλλάζει από την αρχική του τιμή σε «Τετράδιο : 0», εφόσον και η αρχική τιμή της μεταβλητής selecteditemnumber είναι 0. Το text αυτό υπάρχει και αλλάζει δυναμικά προκειμένου να βοηθήσει τον χρήστη να καταλάβει ποιο βιβλίο μετράει τη δεδομένη χρονική στιγμή. Τέλος στην μέθοδο Start() απενεργοποιούμε το αντικείμενο TelosButton με την μέθοδο SetActive() με τιμή false (Εικόνα 4.12). Το αντικείμενο αυτό το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ».

```

Unity Message | 0 references
void Start()
{
    Debug.Log("Into the Script!!!");

    selecteditemnumbertext = GameObject.Find("selecteditemText").GetComponent<TextMeshProUGUI>();
    selecteditemnumbertext.text = "Τετράδιο : " + selecteditemnumber;

    //myPanel.SetActive(false);
    //TelosButton = GameObject.Find("TelosButtonCanvas");
    TelosButton.SetActive(false);
    Debug.Log("YUISS");
}

```

Εικόνα 4.12 Η μέθοδος Start() του script Kef1Exp1.

Στη συνέχεια, όπως κάναμε και στο DemoRaycast.cs script έτσι και εδώ πέρα χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Αρχικά, το αντικείμενο μεταφέρετε μπροστά μας και οριζόντια στον χάρακα και έπειτα εάν έχουμε ήδη επιλέξει μία φορά αυτό το αντικείμενο, το χρώμα του αλλάζει αυτόματα από πορτοκαλί σε γκρι. Αυτό επαναλαμβάνετε μέσα στην Update() για όλα τα τετράδια της σκηνής και όλους τους πιθανούς συνδυασμούς που μπορούν να υπάρξουν (Εικόνα 4.13).

```

if (selection.CompareTag(tetradio1Exp1Tag))
{
    if (selecteditemnumber == 1)
    {
        tetradio1 = GameObject.Find("Tetradio1cm17");
        tetradio1.transform.localPosition = new Vector3(0.7569885f, 0.9738159f, -0.111999f);
        tetradio1.transform.localRotation = Quaternion.Euler(270, 0, 261.737f);
        //to change Material Color
        var tetradio1Renderer = tetradio1.transform.GetComponent<Renderer>();
        tetradio1Renderer.material.SetColor("_Color", Color.grey);
    }
}

```

Εικόνα 4.13 Έλεγχος επιλεγμένου αντικειμένου μέσα στην μεθόδου Update() του script Kef1Exp1.

Τέλος, στην μέθοδο Update(), το script ελέγχει με την χρήση μίας if εάν ο χρήστης έχει επιλέξει δέκα τετράδια από την αρχή του πειράματος προκειμένου να πάει η εφαρμογή στην επόμενη σκηνή. Αυτό επιτυγχάνεται με την χρήση ενός μετρητή με όνομα totalselected (Εικόνα 4.14). Ελέγχει δηλαδή εάν ο μετρητής totalselected ισούται ή έχει ξεπεράσει το δέκα. Εφόσον αυτό έχει συμβεί, το κουμπί TelosButton ενεργοποιείται και εμφανίζεται στην οθόνη του χρήστη. Πατώντας το ο χρήστης θα μεταφερθεί στην επόμενη σκηνή. Το κουμπί αυτό δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω (Ε).

```
if (totalselected >= 10)
{
    TelosButton.SetActive(true);
}
```


Εικόνα 4.14 Ο έλεγχος που οδηγεί στην εμφάνιση του κουμπιού τέλους.



Εικόνα 4.15 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 1<sup>ου</sup> Κεφαλαίου.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή αποκαλύπτει στον χρήστη τα σωστά νούμερα των μετρήσεων των τετραδίων. Ο μαθητής μπορεί να συγκρίνει τα δικά του αποτελέσματα με τα αποτελέσματα που του παρουσιάζει η εφαρμογή. Εάν ο χρήστης έχει κάνει κάποιο λάθος στις μετρήσεις του, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ». Έπειτα ζητείται από τον χρήστη να υπολογίσει την μέση τιμή των μηκών των τετραδίων. Δίνεται έτοιμος ο τύπος της μέσης τιμής σε μορφή εικόνας για λόγους ευκολίας. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα (Εικόνα 4.16).

**ΑΠΟΤΕΛΕΣΜΑΤΑ  
ΠΕΙΡΑΜΑΤΟΣ 1 ΚΕΦ. 1**



**Εάν οι μετρήσεις σας ήταν σωστές τότε τα αποτελέσματα θα έπρεπε να είναι :**

|                    |                     |
|--------------------|---------------------|
| Τετράδιο 1: 17 εκ. | Τετράδιο 6: 31 εκ.  |
| Τετράδιο 2: 22 εκ. | Τετράδιο 7: 15 εκ.  |
| Τετράδιο 3: 37 εκ. | Τετράδιο 8: 35 εκ.  |
| Τετράδιο 4: 28 εκ. | Τετράδιο 9: 22 εκ.  |
| Τετράδιο 5: 26 εκ. | Τετράδιο 10: 28 εκ. |

**Ποια είναι η μέση τιμή των μηκών των τετραδίων;**

**Η μέση τιμή υπολογίζεται από τον τύπο:**

$$\bar{x} = \frac{a_1 + a_2 + \dots + a_n}{n}$$

**Όπου  $\bar{x}$  = μέση τιμή ,  $a$  = μήκοι τετραδίων ,  
 $n$  = αριθμός τετραδίων.**

*Εάν έβγαλες διαφορετικά αποτελέσματα πήγαινε πίσω και ξαναπροσπάθησε.  
"Παντώντας ΤΕΛΟΣ θα λήξει αυτό το πείραμα."*

**ΔΟΚΙΜΑΣΕ ΞΑΝΑ**      **ΤΕΛΟΣ**

Εικόνα 4.16 Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 1<sup>ου</sup> Κεφαλαίου.

### 4.3 Επίλογος

Το πείραμα του 1ου κεφαλαίου «Μετρήσεις Μήκους - Μέση Τιμή» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης συλλέγει τα 10 τετράδια που είναι τοποθετημένα πάνω στο εικονικό γραφείο στην σκηνή επαυξημένης πραγματικότητας και μετράει το μήκος της ράχης τους διαδοχικά. Η μέτρηση γίνεται με την χρήση χάρακα που παρέχεται από το παιχνίδι. Καταγράφει τις τιμές και στην συνέχεια υπολογίζει την μέση τιμή με τον μαθηματικό τύπο που αναγράφεται στην σκηνή τέλους του παιχνιδιού.



## Κεφάλαιο 5ο: Μετρήσεις Χρόνου – Η Ακρίβεια.

### 5.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την μέτρηση χρόνου με την χρήση ψηφιακού και αναλογικού ρολογιού. Στο συγκεκριμένο πείραμα ο μαθητής χρησιμοποιεί ένα ψηφιακό και έπειτα ένα αναλογικό ρολόι για την μέτρηση του χρόνου που θα κάνουν τρία διαφορετικά αυτοκινητάκια για να διανύσουν μία συγκεκριμένη απόσταση. Στη συνέχεια, ο μαθητής συγκρίνει τα αποτελέσματα του αναλογικού με αυτά του ψηφιακού ρολογιού. Τέλος, υπολογίζει την μέση τιμή των χρόνων αυτών και κάνει μία υπόθεση για το ποιο από τα δύο είναι πιο αποτελεσματικό για την ακριβή μέτρηση του χρόνου.

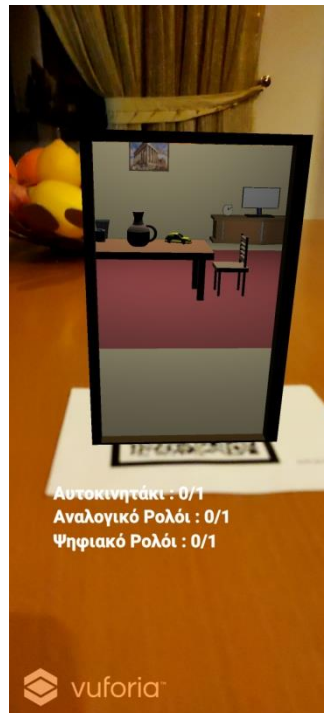
### 5.2 Κατασκευή του πειράματος

Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια και από πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει ένα παιχνίδι αυτοκινητάκι, ένα αναλογικό και ένα ψηφιακό ρολόι. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 5.1).



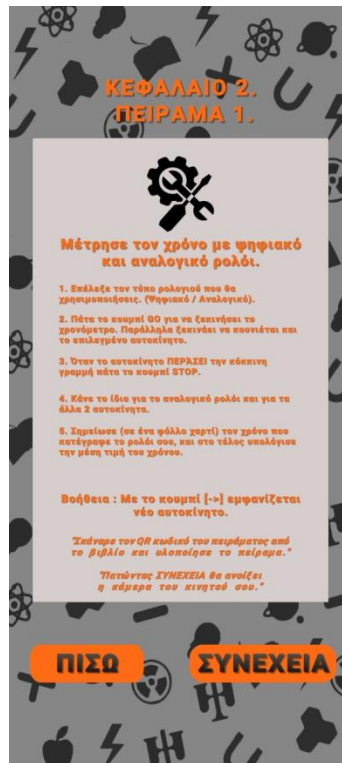
Εικόνα 5.1 Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 2<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού (Εικόνα 5.2). Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο ένα αυτοκινητάκι, ένα αναλογικό ρολόι και ένα ψηφιακό ρολόι. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή.



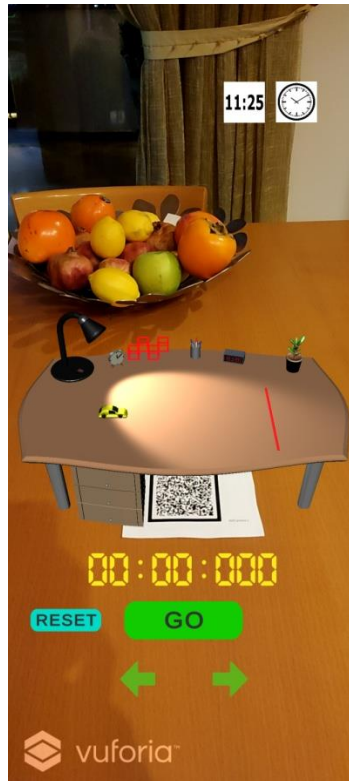
Εικόνα 5.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 2<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 3) η εφαρμογή εξηγεί συνοπτικά στον χρήστη, τα βήματα που πρέπει να ακολουθήσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να μετρήσει τον χρόνο που κάνουν κάποια αυτοκινητάκια να φτάσουν από ένα σημείο σε ένα άλλο. Η μέτρηση αυτή θα γίνεται με ένα ψηφιακό και ένα αναλογικό ρολόι.(Εικόνα 5.3).



Εικόνα 5.3 Η σκηνή της τρίτης φάσης του πειράματος του 2<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, έχει τρεις διαφορετικές επιλογές. Μπορεί είτε να αλλάξει τύπο ρολογιού πατώντας το αντίστοιχο κουμπί στο πάνω μέρος της οθόνης, είτε να αλλάξει αυτοκίνητο πατώντας τα πράσινα βελάκια για να πάει από αυτοκίνητο σε αυτοκίνητο, είτε να πατήσει το κουμπί «START» προκειμένου να ξεκινήσει να κινείται το αυτοκίνητο και να αρχίσει να μετράει το χρονόμετρο. Ο χρήστης σε όλες τις περιπτώσεις πρέπει να καταγράψει τον χρόνο που έκανε το αυτοκίνητο να περάσει την κόκκινη γραμμή. Δίνεται επίσης στο χρήστη η δυνατότητα να μηδενίσει το χρονόμετρο και να επιστρέψει το ενεργοποιημένο αυτοκίνητο στην αρχική του θέση με το πάτημα του κουμπιού «RESET». Τέλος, εφόσον ο χρήστης έχει περάσει από όλα τα αυτοκίνητα της σκηνής και έχει μετρήσει τους χρόνους και με το αναλογικό ρολόι, τότε εμφανίζεται στη σκηνή το κουμπί «ΤΕΛΟΣ» (Εικόνα 5.4).



Εικόνα 5.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 2<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα Kef2exp1scr.cs. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές τριών μεταβλητών τύπου float όπου οι δύο θα χρησιμοποιηθούν ως δείκτες και η μία με όνομα timer θα κρατάει τον χρόνο, δύο μεταβλητών integer που θα βοηθούν στον έλεγχο για το εάν το κουμπί τέλους του πειράματος πρέπει να εμφανιστεί, οκτώ μεταβλητών τύπου GameObject που θα κρατάνε τα δεδομένα των αντικειμένων της σκηνής και μίας μεταβλητής τύπου boolean που θα βοηθάει στον έλεγχο εάν το ο μετρητής είναι ενεργός ή όχι. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου TextMeshProUGUI που θα κρατάει το text του χρονομετρητή το οποίο θα αλλάζει δυναμικά μέσα στη σκηνή (Εικόνα 5.5).

```

9 public class Kef2exp1scr : MonoBehaviour
10 {
11
12     //selection---
13     private Transform selection;
14     public float ActiveScene = 0f;
15     public float carindex = 0f;
16
17
18     private int activation1 = 0;
19     private int activation2 = 0;
20
21     public GameObject yellowcar;
22     public GameObject bluecar;
23     public GameObject greencar;
24
25     public GameObject GOButton;
26     public GameObject STOPButton;
27
28     public GameObject AnalogTimerImage1;
29     public GameObject AnalogTimerImage2;
30
31     public GameObject TelosButton;
32
33
34     public float timer = 0.0f;
35     private bool isTimer = false;
36
37     public TextMeshProUGUI DigitalTimerText;
38

```

Εικόνα 5.5 Αρχικοποίηση μεταβλητών του script Kef2exp1scr.

Έπειτα, στην μέθοδο Start() τύπου void, με την ενεργοποίηση του script γίνεται απενεργοποίηση του αντικειμένου STOPButton της σκηνής και μηδενισμός του carindex (δείκτης ενεργού αμαξιού της σκηνής). Μέσα στην μέθοδο Start() απενεργοποιούμε το αντικείμενο TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο αυτό το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 5.6).

```

51 void Start()
52 {
53     TelosButton.SetActive(false);
54
55     STOPButton.SetActive(false);
56     Debug.Log("Into the Script!!!");
57
58     carindex = 0f;;
59 }

```

Εικόνα 5.6 Η μέθοδος Start() του script Kef2exp1scr.

Στη συνέχεια, στην Update() μέθοδο χρησιμοποιούμε μία if η οποία ελέγχει εάν ο χρήστης έχει πατήσει το κουμπί «START» ενώ έχει επιλεγμένο το ψηφιακό ρολόι. Εάν αυτό έχει συμβεί τότε το script καλεί την μέθοδο DisplayTime(), η οποία ξεκινάει να μετράει τον χρόνο και να τον αναπαριστά σε ακρίβεια millisecond επάνω σε ένα text (Εικόνα 5.7).

```

89 void DisplayTime()
90 {
91     int minutes = Mathf.FloorToInt(timer / 60.0f);
92     int seconds = Mathf.FloorToInt(timer % 60.0f);
93     float milliseconds = Mathf.Floor((timer * 1000f) % 1000);
94     DigitalTimerText.text = string.Format("{0:00}:{1:00}:{2:000}", minutes, seconds, milliseconds);
95 }

```

Εικόνα 5.7 Η μέθοδος DisplayTime() του script Kef2exp1scr.

Στη συνέχεια, μέσα στην μέθοδο Update(), το script ελέγχει με την χρήση μίας if εάν ο χρήστης έχει επιλέξει όλα τα αυτοκίνητα από μία φορά και εάν έχει επιλέξει τουλάχιστον μία φορά και το αναλογικό ρολόι, προκειμένου να πάει η εφαρμογή στην επόμενη σκηνή. Αυτό επιτυγχάνεται με την χρήση δύο μετρητών με ονόματα activation1 και activation2 αντίστοιχα. Ελέγχει δηλαδή εάν οι μετρητές αυτοί έχουν πάρει κάποιες συγκεκριμένες τιμές. Εφόσον αυτό έχει συμβεί, το κουμπί TelosButton ενεργοποιείται και εμφανίζεται στην οθόνη του χρήστη. Πατώντας το ο χρήστης θα μεταφερθεί στην επόμενη σκηνή. Το κουμπί αυτό δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω (Εικόνα 5.8).

```

62 void Update()
63 {
64
65     if (isTimer)
66     {
67         timer += Time.deltaTime;
68         DisplayTime();
69     }
70
71     if (activation1 >= 2 && activation2 >= 1)
72     {
73         TelosButton.SetActive(true);
74     }
75 }

```

Εικόνα 5.8 Η μέθοδος Update() του script Kef2exp1scr.

Οι μέθοδοι που ελέγχουν και υλοποιούν τις λειτουργίες των κουμπιών «START», «STOP» και «RESET» είναι οι StartTimer(), StopTimer() και ResetTimer() αντίστοιχα. Το κάλεσμα των μεθόδων γίνεται με το πάτημα των αντίστοιχων κουμπιών. Η μέθοδος StartTimer() ξεκινάει το χρονόμετρο, η μέθοδος StopTimer() σταματάει το χρονόμετρο, ενώ η ResetTimer() μηδενίζει το χρονόμετρο και καλεί την μέθοδο DisplayTime() (Εικόνα 5.9).

```

97     public void StartTimer()
98     {
99         isTimer = true;
100    }
101
102     0 references
103     public void StopTimer()
104     {
105         isTimer = false;
106    }
107
108     0 references
109     public void ResetTimer()
110     {
111         isTimer = false;
112         timer = 0.0f;
113         timer += Time.deltaTime;
114         DisplayTime();
115         DigitalTimerText.text = "00:00:000";
116    }

```

Εικόνα 5.9 Οι μέθοδοι StartTimer(), StopTimer() και ResetTimer() του script Kef2exp1scr.

Η εναλλαγή από ψηφιακό σε αναλογικό ρολόι και το ανάποδο γίνεται με τα δύο κουμπιά στο επάνω τμήμα της οθόνης του χρήστη. Τα δύο κουμπιά αυτά ονομάζονται DigitalClockButton και AnalogClockButton και με το πάτημα τους καλούν τις μεθόδους activateDigital() και activateAnalog() αντίστοιχα. Οι μέθοδοι αυτοί υλοποιούν μία ενεργοποίηση - απενεργοποίηση των ανάλογων αντικειμένων, προκειμένου να εμφανιστεί το σωστό ρολόι στην οθόνη του χρήστη (Εικόνα 5.10).

```

159     public void activateAnalog()
160     {
161         activation2 = activation2 + 1;
162         DigitalTimerText.gameObject.SetActive(false);
163         AnalogTimerImage1.SetActive(true);
164         AnalogTimerImage2.SetActive(true);
165     }
166
167     0 references
168     public void activateDigital()
169     {
170         AnalogTimerImage1.SetActive(false);
171         AnalogTimerImage2.SetActive(false);
172         DigitalTimerText.gameObject.SetActive(true);
173     }

```

Εικόνα 5.10 Οι μέθοδοι activateAnalog() και activateDigital() του script Kef2exp1scr.

Τέλος, η εναλλαγή από το ένα αυτοκίνητο στο άλλο γίνεται με τα δύο κουμπιά στο κάτω τμήμα της οθόνης του χρήστη. Τα κουμπιά αυτά έχουν το σχήμα πράσινων βέλων. Τα δύο κουμπιά αυτά ονομάζονται NextCar και PreviousCar και με το πάτημα τους καλούν τις μεθόδους NextCar() και PreviousCar() αντίστοιχα. Οι μέθοδοι αυτοί υλοποιούν μία ενεργοποίηση - απενεργοποίηση των ανάλογων αντικειμένων, προκειμένου να εμφανιστεί το σωστό αυτοκίνητο στην σκηνή. Έπειτα ο χρήστης μπορεί να μετρήσει τον χρόνο του αμαξίου αυτού (Εικόνα 5.11).

```

116 public void NextCar()
117 {
118     if (carindex == 0f)
119     {
120         activation1 = activation1 + 1;
121         yellowcar.SetActive(false);
122         bluecar.SetActive(true);
123         carindex = carindex + 1f;
124     }
125     else if (carindex == 1f)
126     {
127
128         activation1 = activation1 + 1;
129         bluecar.SetActive(false);
130         greencar.SetActive(true);
131         carindex = carindex + 1f;
132     }
133     else if (carindex == 2f)
134     {
135         Debug.Log("No more next cars!");
136     }
137 }
138
0 references
139 public void PreviousCar()
140 {
141     if (carindex == 2f)
142     {
143         greencar.SetActive(false);
144         bluecar.SetActive(true);
145         carindex = carindex - 1f;
146     }
147     else if (carindex == 1f)
148     {
149         bluecar.SetActive(false);
150         yellowcar.SetActive(true);
151         carindex = carindex - 1f;
152     }
153     else if (carindex == 0f)
154     {
155         Debug.Log("No more previous cars!");
156     }

```

Εικόνα 5.11 Οι μέθοδοι NextCar() και PreviousCar() του script Kef2exp1scr.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή αποκαλύπτει στον χρήστη τα σωστά νούμερα των μετρήσεων των χρόνων. Ο μαθητής μπορεί να συγκρίνει τα δικά του αποτελέσματα με τα αποτελέσματα που του παρουσιάζει η εφαρμογή. Εάν ο χρήστης έχει κάνει κάποιο λάθος στις μετρήσεις του, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ». Έπειτα ζητείται από τον χρήστη να υπολογίσει την μέση τιμή των χρόνων που κατέγραψε. Δίνεται έτοιμος ο τύπος της μέσης τιμής σε μορφή εικόνας για λόγους ευκολίας. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα (Εικόνα 5.12).



Εικόνα 5.12 Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 2<sup>ου</sup> Κεφαλαίου.

### 5.3 Επίλογος

Το πείραμα του 2<sup>ου</sup> κεφαλαίου «Μετρήσεις Χρόνου – Η Ακρίβεια» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης μετράει με την χρήση ενός αναλογικού και ενός ψηφιακού ρολογιού τους χρόνους που θα κάνουν τα τρία διαφορετικά αυτοκινητάκια του εικονικού χώρου για να φτάσουν στον στόχο. Καταγράφει τις τιμές και στην συνέχεια υπολογίζει την μέση τιμή των χρόνων με τον μαθηματικό τύπο που αναγράφεται στην σκηνή τέλους του παιχνιδιού.



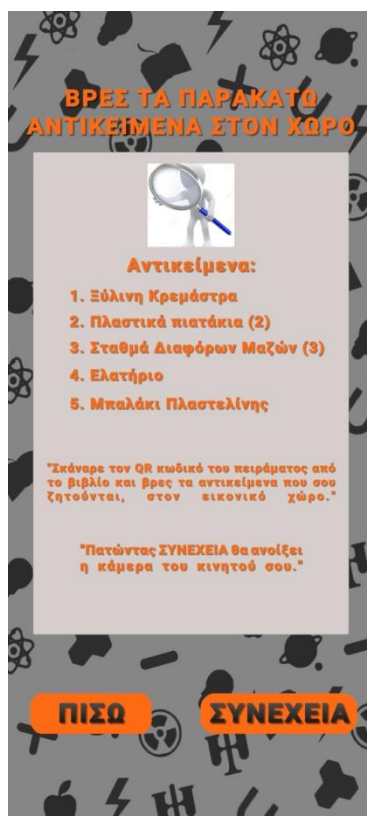
## Κεφάλαιο 6ο: Μετρήσεις Μάζας – Τα διαγράμματα.

### 6.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την μέτρηση της μάζας με την χρήση ζυγαριάς και τον υπολογισμό του βάρους μίας μπάλας. Στο συγκεκριμένο πείραμα ο μαθητής χρησιμοποιεί την ζυγαριά για την μέτρηση της μάζας τριών βαριδίων, στη συνέχεια προσπαθεί να φέρει την ζυγαριά σε ισορροπία και τέλος υπολογίζει το βάρος της μπάλας.

### 6.2 Κατασκευή του πειράματος

Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια και από πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει μία ξύλινη κρεμάστρα, δύο πλαστικά πιατάκια, τρία σταθμά διαφόρων μαζών, ένα ελατήριο και ένα μπαλάκι πλαστελίνης. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 6.1).



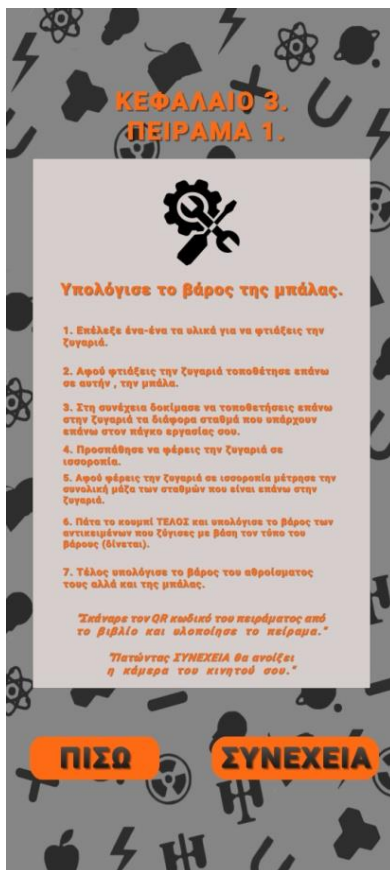
Εικόνα 6.1 Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 3<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού. Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο δύο πιάτα, μία κρεμάστρα, τρία σταθμά, ένα ελατήριο και ένα μπαλάκι. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή (Εικόνα 6.2).



Εικόνα 6.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 3<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 3) η εφαρμογή εξηγεί συνοπτικά στον χρήστη, τις ενέργειες που πρέπει να εκτελέσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να υπολογίσει το βάρος της μπάλας της σκηνής. (Εικόνα 6.3).



Εικόνα 6.3 Η σκηνή της τρίτης φάσης του πειράματος του 3<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά, ο χρήστης πρέπει με την σωστή επιλογή αντικειμένων να χτίσει την αυτοσχέδια ζυγαριά επιλέγοντας τα αντικείμενα του χώρου με την σωστή σειρά. Στη συνέχεια, ο μαθητής πρέπει να τοποθετήσει την μπάλα και τα σταθμά επάνω στην ζυγαριά. Έπειτα, ο χρήστης πρέπει να δοκιμάσει να φέρει την ζυγαριά σε ισορροπία βάζοντας και βγάζοντας από την ζυγαριά σταθμά. Τέλος, αφού η ζυγαριά έχει έρθει σε ισορροπία, ο μαθητής πρέπει να παρατηρήσει τις μάζες που αναγράφονται πάνω στα σταθμά, να τις αθροίσει και να κατανοήσει ποιος είναι ο λόγος που η ζυγαριά ισορροπεί (Εικόνα 6.4).



Εικόνα 6.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 3<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα Kef3pirama1.cs. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές πέντε μεταβλητών τύπου boolean που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές δέκα μεταβλητών τύπου string που θα χρησιμοποιηθούν ως tags πάνω στα αντικείμενα της σκηνής και δεκαεννέα μεταβλητών τύπου GameObject που θα κρατάνε τα δεδομένα των δεκαεννέα αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου TextMeshProUGUI που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 6.5).

```

9 public class Kef3pirama1script : MonoBehaviour
10 {
11     //selection---
12     private Transform selection;
13
14     public bool kremastranewisactive = false;
15     public bool piatonewisactive = false;
16     public bool piatonewisactive2 = false;
17     public bool mpalakinewisactive = false;
18
19     public bool afterbuild = false;
20
21     public float kg = 0f;
22
23     public float zygbuildindex = 0f;
24
25     //tags ---
26     // [SerializeField] private string selectableTag = "Selectable";
27     [SerializeField] private string KremastraExp3Tag = "KremastraExp3Tag";
28     [SerializeField] private string PiatoExp3Tag = "PiatoExp3Tag";
29     [SerializeField] private string Piato2Exp3Tag = "Piato2Exp3Tag";
30     [SerializeField] private string MpalakiExp3Tag = "MpalakiExp3Tag";
31     [SerializeField] private string Stathma50Exp3Tag = "Stathma50Exp3Tag";
32     [SerializeField] private string Stathma30Exp3Tag = "Stathma30Exp3Tag";
33     [SerializeField] private string Stathma20Exp3Tag = "Stathma20Exp3Tag";
34     [SerializeField] private string Stathma50Exp3TagNew = "Stathma50Exp3TagNew";
35     [SerializeField] private string Stathma30Exp3TagNew = "Stathma30Exp3TagNew";
36     [SerializeField] private string Stathma20Exp3TagNew = "Stathma20Exp3TagNew";
37
38
39     Collider mpalacollider;
40     Collider kg50collider;
41     Collider kg30collider;
42     Collider kg20collider;
43
44     public GameObject Piato2;
45     public GameObject Piato;
46     public GameObject Stathma50;
47     public GameObject Stathma30;

```

Εικόνα 6.5 Αρχικοποίηση μεταβλητών του script Kef3pirama1script.

Έπειτα, στην μέθοδο Start() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικειμένου (infotext) της σκηνης και απενεργοποίηση του αντικειμένου TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο αυτό (TelosButton) το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 6.6).

```

80 void Start()
81 {
82     Debug.Log("Into the Script!!!");
83
84     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
85     infotext.text = "Φτιάξε την ζυγαριά!";
86
87     TelosButton.SetActive(false);
88     Debug.Log("YUISS");
89 }

```

Εικόνα 6.6 Η μέθοδος Start() του script Kef3pirama1script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας

μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε τοποθετεί αυτόματα το αντικείμενο αυτό στο σημείο που πρέπει προκειμένου να δημιουργήσει την ζυγαριά. Έπειτα, όταν ο χρήστης φτιάξει την ζυγαριά και τοποθετήσει το μπαλάκι επάνω σε αυτήν είναι έτοιμος να αρχίσει να τοποθετεί σταθμά. Οι νέοι έλεγχοι που γίνονται αφήνουν τον χρήστη να επιλέξει ένα από τα βαρίδια που έχει ήδη τοποθετήσει επάνω στην ζυγαριά προκειμένου να το αφαιρέσει. Με τον τρόπο αυτό ο χρήστης αφαιρεί και προσθέτει συνεχώς βαρίδια στην ζυγαριά μέχρι να βρει τον κατάλληλο συνδυασμό που θα κάνει την ζυγαριά να ισορροπηθεί (Εικόνα 6.7). Σε αυτό το σημείο, το script ελέγχει με την χρήση μίας if εάν ο χρήστης έχει βρει τον συνδυασμό που πρέπει για να εμφανιστεί το κουμπί «ΤΕΛΟΣ» (Εικόνα 6.8). Εφόσον έχει συμβεί αυτό, δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί τέλους και να μεταφερθεί στην επόμενη σκηνή, αφού φυσικά πρώτα έχει παρατηρήσει τις τιμές των μαζών των βαριδίων. Το κουμπί αυτό δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω.

```

92 void Update()
93 {
94
95
96     if (Input.GetMouseButtonDown(0))
97     {
98         // Debug.Log("Mouse Button Down!!!");
99         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
100         RaycastHit hit;
101
102         if (Physics.Raycast(ray, out hit))
103         {
104             // Debug.Log("Hit Something!!!");
105             var selection = hit.transform;
106             if (selection.CompareTag(KremastraExp3Tag))
107             {
108                 // Kremastra.transform.localPosition = new Vector3(-0.0057f, 1.8067f, 0.076f);
109                 // Kremastra.transform.localRotation = Quaternion.Euler(0, 90, 90f);
110
111                 Debug.Log("into the kremastra IF");
112
113                 Kremastra.SetActive(false);
114                 KremastraNew.SetActive(true);
115
116                 RopesHangerLeft.SetActive(true);
117                 RopesHangerRight.SetActive(true);
118
119             }
120             else if (selection.CompareTag(PiatoExp3Tag) && kremastranewisactive == true)
121             {
122                 //tetradio1 = GameObject.Find("Tetradio1cm17");
123                 //Piato.transform.localPosition = new Vector3(-0.399f, 1.172f, 0.11f);
124
125                 Piato.SetActive(false);
126                 PiatoNew.SetActive(true);
127                 Debug.Log("piato2 was hit and isactive!!!");
128             }
129

```

Εικόνα 6.7 Η μέθοδος Update() του script Kef3pirama1script.

```

253     if (Stathma20New.activeSelf && !Stathma30New.activeSelf && Stathma50New.activeSelf && afterbuild == true)
254     {
255         //70kg!
256         KremastraNew.GetComponent<Animator>().Play("70kg");
257         LeftSideBuilt.GetComponent<Animator>().Play("70kg");
258         RightSideBuilt.GetComponent<Animator>().Play("70kg");
259
260         infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
261         infotext.text = "Η ζυγαριά ήρθε σε ισορροπία!";
262
263         TelosButton.SetActive(true);
264     }

```

Εικόνα 6.8 Ο έλεγχος που οδηγεί στην εμφάνιση του κουμπιού τέλους.

Τέλος, στο πείραμα αυτό γίνεται χρήση animation προκειμένου ο χρήστης να μπορεί να παρατηρεί άμεσα και με τρόπο παραστατικό τις αλλαγές που επέρχονται στην ζυγαριά όταν αυτός τοποθετεί ή αφαιρεί ένα βαρίδιο (Εικόνα 6.9).

```

217     if (!Stathma20New.activeSelf && !Stathma30New.activeSelf && !Stathma50New.activeSelf && afterbuild==true)
218     {
219         //ballOnly!
220         KremastraNew.GetComponent<Animator>().Play("ballOnly");
221         LeftSideBuilt.GetComponent<Animator>().Play("ballOnly");
222         RightSideBuilt.GetComponent<Animator>().Play("ballOnly");
223
224         TelosButton.SetActive(false);
225     }

```

Εικόνα 6.9 Χρήση animation μέσω script.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή αποκαλύπτει στον χρήστη τις μάζες των βαριδίων. Ο μαθητής μπορεί να συγκρίνει τα δικά του αποτελέσματα με τα αποτελέσματα που του παρουσιάζει η εφαρμογή. Εάν ο χρήστης έχει κάνει κάποιο λάθος στις μετρήσεις του, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ». Έπειτα ζητείται από τον χρήστη να υπολογίσει τα βάρη των βαριδίων, το άθροισμα τους και το βάρος της μπάλας. Δίνεται έτοιμος ο τύπος του βάρους σε μορφή εικόνας για λόγους ευκολίας. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα (Εικόνα 6.10).



Εικόνα 6.10 Η διδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 3<sup>ου</sup> Κεφαλαίου.

### 6.3 Επίλογος

Το πείραμα του 3ου κεφαλαίου «Μετρήσεις Μάζας – Τα διαγράμματα» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης κατασκευάζει την ζυγαριά στον εικονικό χώρο, τοποθετεί κατάλληλα επάνω σε αυτήν τα διάφορα αντικείμενα και προσπαθεί να φέρει την ζυγαριά σε ισορροπία. Οι μάζες των βαριδίων αναγράφονται πάνω σε αυτά. Όταν η ζυγαριά έρθει σε ισορροπία ο χρήστης καταγράφει τις τιμές και στην συνέχεια υπολογίζει το βάρος των βαριδίων και της μπάλας με τον μαθηματικό τύπο που αναγράφεται στην σκηνή τέλους του παιχνιδιού.

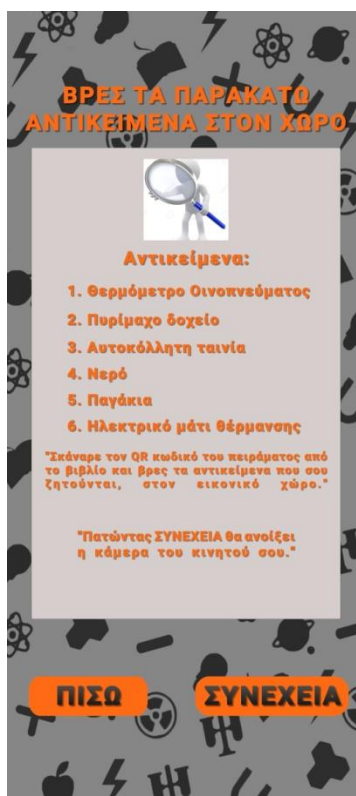
## Κεφάλαιο 7ο: Μετρήσεις Θερμοκρασίας – Η Βαθμονόμηση.

### 7.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την μέτρηση θερμοκρασίας με την χρήση θερμόμετρου. Στο συγκεκριμένο πείραμα ο μαθητής χρησιμοποιεί το θερμόμετρο για την μέτρηση της θερμοκρασίας του νερού στις διάφορες καταστάσεις του.

### 7.2 Κατασκευή του πειράματος

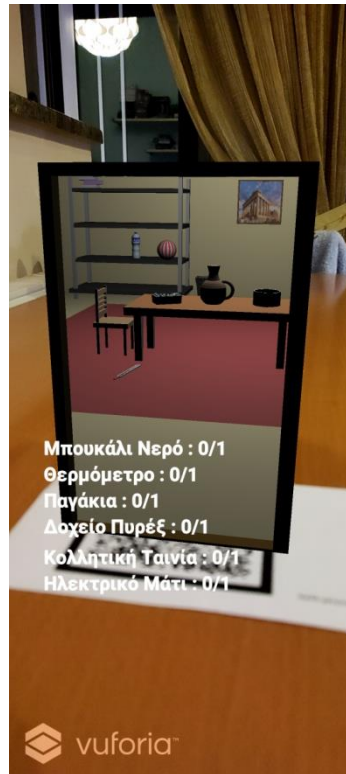
Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια και από πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει ένα θερμόμετρο οινόπνευματος, ένα πυρίμαχο δοχείο, μία αυτοκόλλητη ταινία, ένα μπουκάλι νερό, ένα δοχείο με παγάκια και ένα ηλεκτρικό μάτι θέρμανσης. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 7.1).



Εικόνα 7.1 Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 4<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα

την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού. Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο ένα μπουκάλι νερό, ένα θερμομέτρο, ένα δοχείο με παγάκια, ένα δοχείο πυρέξ, μία κολλητική ταινία και ένα ηλεκτρικό μάτι. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή (Εικόνα 7.2).



Εικόνα 7.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 4<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 3) η εφαρμογή εξηγεί συνοπτικά στον χρήστη, τα βήματα που πρέπει να ακολουθήσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να μετρήσει τις θερμοκρασίες των υγρών των δοχείων (Εικόνα 7.3).



Εικόνα 7.3 Η σκηνή της τρίτης φάσης του πειράματος του 4<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά ο χρήστης πρέπει να σηκώσει από το τραπέζι το θερμόμετρο, όταν το κάνει αυτό το θερμόμετρο προστίθεται στην οθόνη του χρήστη σε μορφή εικόνας. Στη συνέχεια, πρέπει να πάρει το μπουκάλι και να βάλει νερό στα τρία δοχεία που βρίσκονται μπροστά του. Έπειτα, ο χρήστης πρέπει να πάρει τα παγάκια και να τα βάλει στο δοχείο που του ζητείται. Τέλος, ο χρήστης πρέπει να πάρει ένα από τα τρία δοχεία, να το τοποθετήσει επάνω στο ηλεκτρικό μάτι (το ηλεκτρικό μάτι θεωρείται αναμμένο) και να μετρήσει τις θερμοκρασίες όλων των δοχείων. Την μέτρηση μπορεί να την κάνει με όποια σειρά επιθυμεί και η θερμοκρασία του υγρού του δοχείου αλλάζει δυναμικά και αναγράφεται στο θερμόμετρο που βρίσκεται σταθερό επάνω στην οθόνη του χρήστη (Εικόνα 7.4).



Εικόνα 7.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 4<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα Kef4pirama1.cs. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές δεκατεσσάρων μεταβλητών τύπου boolean που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές δεκατριών μεταβλητών τύπου string που θα χρησιμοποιηθούν ως tags πάνω στα αντικείμενα της σκηνής και δεκαεννέα μεταβλητών τύπου GameObject που θα κρατάνε τα δεδομένα των δεκαεννέα αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου TextMeshProUGUI που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 7.5).

```

9 public class Kef4pirama1script : MonoBehaviour
10 {
11
12     //selection---
13     private Transform selection;
14
15     public int piramaindex = 0;
16
17     public bool calledonce = false;
18
19     public bool thermometroisactive = false;
20     public bool mpoukaliislifted = false;
21     public bool pagakiaislifted = false;
22     public bool doxeiowater2islifted = false;
23
24
25     public bool doxeiowaterfilled = false;
26     public bool doxeiowater2filled = false;
27     public bool doxeiowater3filled = false;
28     public bool allfilled = false;
29
30     public bool doxeiowateronlyready = false;
31     public bool doxeiowaterICEReady = false;
32     public bool doxeiowaterBOILready = false;
33
34     public bool doxeiowaterICEisactive = false;
35
36     public bool readytomeasure = false;
37
38     public bool telosindex1 = false;
39     public bool telosindex2 = false;
40     public bool telosindex3 = false;
41
42
43     // [SerializeField] private string selectableTag = "Selectable";
44     [SerializeField] private string ilektrikoMatiExp4Tag = "ilektrikoMatiExp4Tag";
45     [SerializeField] private string doxeiowaterICEExp4Tag = "doxeiowaterICEExp4Tag";
46     [SerializeField] private string doxeiowaterExp4Tag = "doxeiowaterExp4Tag";
47     [SerializeField] private string doxeiowater2Exp4Tag = "doxeiowater2Exp4Tag";

```

Εικόνα 7.5 Αρχικοποίηση μεταβλητών του script Kef4pirama1script.

Έπειτα, στην μέθοδο Awake() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικειμένου (infotext) της σκηνης και απενεργοποίηση του αντικειμένου TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο αυτό (TelosButton) το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 7.6).

```

86 private void Awake()
87 {
88     Application.targetFrameRate = 30;
89     Debug.Log("Into the Script!!!");
90
91     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
92     infotext.text = "Πάρε το θερμόμετρο.";
93
94     TelosButton.SetActive(false);
95
96 }

```

Εικόνα 7.6 Η μέθοδος Awake() του script Kef4pirama1script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε ο χρήστης «κρατάει» αυτό το αντικείμενο και η εφαρμογή περιμένει από αυτόν να το χρησιμοποιήσει ή να τοποθετήσει κάπου αλλού. Αρχικά, ο χρήστης «κρατάει» το θερμόμετρο και το μπουκάλι με το νερό το οποίο έπειτα χρησιμοποιεί πατώντας πάνω στα δοχεία για να τα γεμίσει. Όταν ο χρήστης σηκώσει το θερμόμετρο καλείται μία setActive() για την ενεργοποίηση του image με μορφή θερμόμετρου στην οθόνη του χρήστη. Στη συνέχεια, ο χρήστης πρέπει με παρόμοιο τρόπο να τοποθετήσει παγάκια σε ένα δοχείο και να πάρει ένα άλλο δοχείο και να το τοποθετήσει επάνω στο ηλεκτρικό μάτι. Όλες αυτές οι διαδικασίες επιτυγχάνονται μέσα στο script με την συνεχή ενεργοποίηση και απενεργοποίηση των αντικειμένων της σκηνής ανάλογα με τις κινήσεις του χρήστη (Εικόνα 7.7). Η ενεργοποίηση και η απενεργοποίηση των αντικειμένων αυτών γίνεται με την μέθοδο setActive(false/true) με τιμή true ή false. Σε αυτό το σημείο, το script ελέγχει με την χρήση μίας if εάν ο χρήστης έχει κάνει όλες τις μετρήσεις των θερμοκρασιών που πρέπει για να εμφανιστεί το κουμπί «ΤΕΛΟΣ» (Εικόνα 7.8). Εφόσον αυτό έχει συμβεί, δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί τέλους και να μεταφερθεί στην επόμενη σκηνή αφού φυσικά έχει πρώτα παρατηρήσει και καταγράψει τις τιμές των θερμοκρασιών. Το κουμπί αυτό δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω.

```

105 void Update()
106 {
107     if (Input.GetMouseButtonDown(0))
108     {
109         // Debug.Log("Mouse Button Down!!!");
110         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
111         RaycastHit hit;
112
113         if (Physics.Raycast(ray, out hit))
114         {
115             // Debug.Log("Hit Something!!!");
116             var selection = hit.transform;
117             if (selection.CompareTag(ThermometroExp4Tag) && readytomeasure==false)
118             {
119                 // Kremastra.transform.localPosition = new Vector3(-0.0057f, 1.8067f, 0.076f);
120                 // Kremastra.transform.localRotation = Quaternion.Euler(0, 90, 90f);
121
122                 Thermometro.SetActive(false);
123                 thermometerimage.SetActive(true);
124                 thermometerindicatorimage.SetActive(true);
125
126                 infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
127                 infotext.text = "Πάρε το μπουκάλι.";
128
129                 thermometroisactive = true;
130
131                 piramaindex = piramaindex + 1;
132
133
134
135     }

```

Εικόνα 7.7 Η μέθοδος Update() του script Kef4pirama1script.

```

322     if (telosindex1==true && telosindex2 == true && telosindex3 == true)
323     {
324         TelosButton.SetActive(true);
325     }

```

Εικόνα 7.8 Ο έλεγχος που οδηγεί στην εμφάνιση του κουμπιού τέλους.

Τέλος, στο πείραμα αυτό γίνεται χρήση animation προκειμένου ο χρήστης να μπορεί να παρατηρεί άμεσα και με τρόπο παραστατικό τις αλλαγές που επέρχονται στην θερμοκρασία όταν αυτός τοποθετεί το θερμόμετρο μέσα στο υγρό κάποιου δοχείου (Εικόνα 7.9).

```

if (selection.CompareTag(doxeiowaterBOIL3Exp4Tag) && readytomeasure == true)
{
    if (!doxeiowaterwiththermometer.activeSelf && !doxeiowaterICEwiththermometer.activeSelf && !doxeiowaterBOILwiththermometer.activeSelf)
    {
        Debug.Log("should play animation");
        thermometerindicatorimage.GetComponent<Animator>().Play("20to100anim");
    }
    else if (doxeiowaterwiththermometer.activeSelf && !doxeiowaterICEwiththermometer.activeSelf && !doxeiowaterBOILwiththermometer.activeSelf)
    {
        Debug.Log("should play animation");
        thermometerindicatorimage.GetComponent<Animator>().Play("20to100anim");
    }
    else if (!doxeiowaterwiththermometer.activeSelf && doxeiowaterICEwiththermometer.activeSelf && !doxeiowaterBOILwiththermometer.activeSelf)
    {
        Debug.Log("should play animation");
        thermometerindicatorimage.GetComponent<Animator>().Play("0to100anim");
    }
    else if (!doxeiowaterwiththermometer.activeSelf && !doxeiowaterICEwiththermometer.activeSelf && doxeiowaterBOILwiththermometer.activeSelf)
    {
        Debug.Log("yes");
    }
}

```

Εικόνα 7.9 Εικόνα Χρήση animation μέσω script.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή αποκαλύπτει στον χρήστη τις θερμοκρασίες των υγρών των δοχείων. Ο μαθητής μπορεί να συγκρίνει τα δικά του αποτελέσματα με τα αποτελέσματα που του παρουσιάζει η εφαρμογή. Εάν ο χρήστης έχει κάνει κάποιο λάθος στις μετρήσεις του, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ». Έπειτα, η εφαρμογή θέτει κάποιες ερωτήσεις στον χρήστη σχετικά με το πείραμα που μόλις υλοποίησε. Για την απάντηση των ερωτήσεων αυτών ο μαθητής μπορεί να ζητήσει βοήθεια από κάποιο άλλο άτομο. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα (Εικόνα 7.10).



Εικόνα 7.10 Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 4<sup>ου</sup> Κεφαλαίου.

### 7.3 Επίλογος

Το πείραμα του 4ου κεφαλαίου “Μετρήσεις Θερμοκρασίας – Η Βαθμονόμηση” του σχολικού βιβλίου της Φυσικής Α’ Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης προσθέτει νερό μέσα στα δοχεία και τα τροποποιεί κατάλληλα ώστε να αλλάξει την θερμοκρασία του νερού στην σκηνή επαυξημένης πραγματικότητας. Έπειτα μετράει την θερμοκρασία του νερού σε κάθε δοχείο με την χρήση ενός θερμόμετρου. Καταγράφει τις τιμές της θερμοκρασία και στην συνέχεια απαντάει στις ερωτήσεις που αναγράφονται στην σκηνή τέλους του παιχνιδιού.

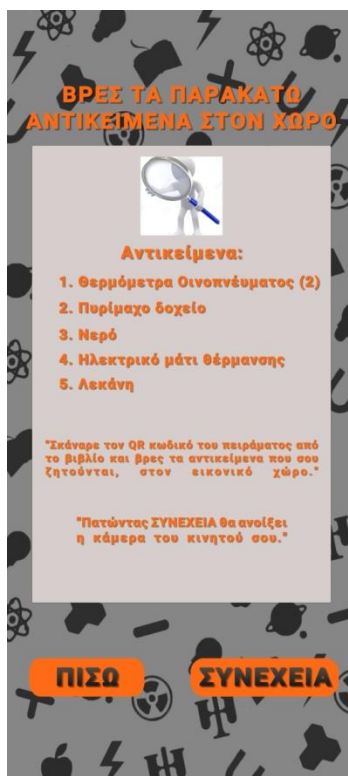
## Κεφάλαιο 8ο: Από τη θερμότητα στη θερμοκρασία – Η θερμική ισορροπία.

### 8.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την μέτρηση της θερμοκρασίας με την χρήση θερμομέτρων. Στο συγκεκριμένο πείραμα ο μαθητής χρησιμοποιεί δύο θερμομέτρα για την παράλληλη μέτρηση της θερμοκρασίας του νερού σε δύο διαφορετικά δοχεία και την παρατήρηση της ροής της θερμότητας από το ένα δοχείο στο άλλο.

### 8.2 Κατασκευή του πειράματος

Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει δύο θερμομέτρα οινόπνευματος, ένα πυρίμαχο δοχείο, ένα μπουκάλι νερό, ένα ηλεκτρικό μάτι θέρμανσης και μία λεκάνη. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 8.1).



Εικόνα 8.1 Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 5<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού. Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο δύο θερμομέτρα, ένα δοχείο πυρέξ, ένα ηλεκτρικό μάτι, ένα μπουκάλι νερό και μία λεκάνη. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή (Εικόνα 8.2).



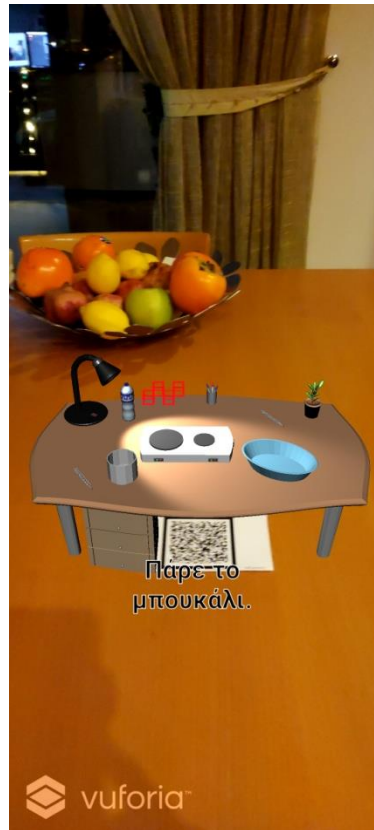
Εικόνα 8.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 5<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 3) η εφαρμογή εξηγεί συνοπτικά στον χρήστη, τα βήματα που πρέπει να ακολουθήσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να παρατηρήσει το φαινόμενο της θερμικής ισορροπίας (Εικόνα 8.3).



Εικόνα 8.3 Η σκηνή της τρίτης φάσης του πειράματος του 5<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά ο χρήστης πρέπει να σηκώσει από το τραπέζι το μπουκάλι με το νερό και να γεμίσει το δοχείο και την λεκάνη. Στη συνέχεια, πρέπει να σηκώσει το δοχείο με το νερό και να το βάλει επάνω στο ηλεκτρικό μάτι (το οποίο θεωρείται ανοιχτό εξαρχής). Έπειτα, ο μαθητής περιμένει λίγα δευτερόλεπτα να ζεσταθεί το νερό για να φτάσει σε σημείο βρασμού και μετά παίρνει το δοχείο και το τοποθετεί μέσα στη λεκάνη. Επιπρόσθετα, σηκώνει από το τραπέζι τα δύο θερμομέτρα, τοποθετεί ένα μέσα στην λεκάνη και ένα μέσα στο δοχείο και ξεκινά να μετρά και να παρατηρεί τι συμβαίνει με τις θερμοκρασίες. Οι θερμοκρασίες των υγρών του δοχείου και της λεκάνης αλλάζουν δυναμικά και αναγράφονται στα θερμομέτρα που βρίσκονται σταθερά επάνω στην οθόνη του χρήστη (Εικόνα 8.4).



Εικόνα 8.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 5<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα Kef5pirama1script.cs. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές εικοσιένα μεταβλητών τύπου boolean που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές δώδεκα μεταβλητών τύπου string που θα χρησιμοποιηθούν ως tags πάνω στα αντικείμενα της σκηνής και δεκαοκτώ μεταβλητών τύπου GameObject που θα κρατάνε τα δεδομένα των δεκαεννέα αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου TextMeshProUGUI που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 8.5).

```

9 public class Kef5pirama1script : MonoBehaviour
10 {
11     //selection----
12     private Transform selection;
13
14     public int piramaindex = 0;
15
16     public bool calledonce = false;
17
18     public bool thermometroisactive = false;
19     public bool thermometro2isactive = false;
20     public bool mpoukaliislifted = false;
21     public bool doxeiowaterislifted = false;
22
23
24     public bool doxeiofilled = false;
25     public bool lekanifilled = false;
26
27
28     public bool allfilled = false;
29
30     public bool doxeiowateronlyready = false;
31     public bool doxeiowaterICReady = false;
32     public bool doxeiowaterBOILready = false;
33
34     public bool readytomeasure1 = false;
35     public bool readytomeasure2 = false;
36
37     public bool telosindex1 = false;
38     public bool telosindex2 = false;
39     public bool telosindex3 = false;
40
41
42     public bool waterisready = false;
43
44     public bool doxeioboilislifted = false;
45
46     public int thermometracollected = 0;
47     public bool getthermometraphase = false;

```

Εικόνα 8.5 Αρχικοποίηση μεταβλητών του script Kef5pirama1script.

Έπειτα, στην μέθοδο Awake() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικειμένου (infotext) της σκηνής και απενεργοποίηση του αντικειμένου TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο αυτό (TelosButton) το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 8.6).

```

97 private void Awake()
98 {
99     Application.targetFrameRate = 30;
100    Debug.Log("Into the Script!!!");
101
102    infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
103    infotext.text = "Πάρε το μπουκάλι.";
104
105    TelosButton.SetActive(false);
106
107 }

```

Εικόνα 8.6 Η μέθοδος Awake() του script Kef5pirama1script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε ο χρήστης «κρατάει» αυτό το αντικείμενο και η εφαρμογή περιμένει από αυτόν να το χρησιμοποιήσει ή να τοποθετήσει κάπου αλλού. Αρχικά, ο χρήστης επιλέγει και «κρατάει» το μπουκάλι με το νερό το οποίο έπειτα χρησιμοποιεί πατώντας πάνω στο δοχείο και στην λεκάνη για να τα γεμίσει. Στη συνέχεια, ο χρήστης πρέπει με παρόμοιο τρόπο να πάρει και να τοποθετήσει το δοχείο επάνω στο ηλεκτρικό μάτι. Ο χρήστης περιμένει μερικά δευτερόλεπτα να ζεσταθεί το νερό, περιμένει δηλαδή να καλεστεί η μέθοδος που πρέπει. Αυτή η αναμονή επιτυγχάνεται στη συγκεκριμένη περίπτωση με την μέθοδο Invoke. Η Invoke μπορεί να καλέσει μία μέθοδο μετά από μία συγκεκριμένη ώρα που θα ορίσει ο χρήστης. Αφού το νερό είναι έτοιμο (σημείο βρασμού) ο μαθητής μπορεί να πάρει το δοχείο και να το τοποθετήσει μέσα στην λεκάνη. Τέλος ο χρήστης παίρνει τα θερμόμετρα, τα τοποθετεί μέσα στα υγρά και παρατηρεί τι συμβαίνει. Όταν ο χρήστης σηκώσει το θερμόμετρο καλείται μία setActive() για την ενεργοποίηση του image με μορφή θερμόμετρου στην οθόνη του χρήστη. Όλες αυτές οι διαδικασίες επιτυγχάνονται μέσα στο script με την συνεχή ενεργοποίηση και απενεργοποίηση των αντικειμένων της σκηνής ανάλογα με τις κινήσεις του χρήστη (Εικόνα 8.7). Η ενεργοποίηση και η απενεργοποίηση των αντικειμένων αυτών γίνεται με την μέθοδο setActive(false/true) με τιμή true ή false. Σε αυτό το σημείο, το script πάλι με την χρήση μίας Invoke μίας μεθόδου καλεί την ανάλογη μέθοδο για να εμφανιστεί το κουμπί «ΤΕΛΟΣ» (Εικόνα 8.8). Εφόσον αυτό έχει συμβεί, δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί τέλους και να μεταφερθεί στην επόμενη σκηνή αφού φυσικά έχει πρώτα παρατηρήσει και καταγράψει τις τιμές των θερμοκρασιών. Το κουμπί αυτό δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω.

```

116 void Update()
117 {
118     if (Input.GetMouseButtonDown(0))
119     {
120         // Debug.Log("Mouse Button Down!!!");
121         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
122         RaycastHit hit;
123
124         if (Physics.Raycast(ray, out hit))
125         {
126             // Debug.Log("Hit Something!!!");
127             var selection = hit.transform;
128             if (selection.CompareTag(Mpoukaliner01Exp4Tag) && readytomeasure1 == false)
129             {
130                 // Kremastra.transform.localPosition = new Vector3(-0.0057f, 1.8067f, 0.076f);
131                 // Kremastra.transform.localRotation = Quaternion.Euler(0, 90, 90f);
132
133                 Mpoukaliner01.SetActive(false);
134
135                 infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
136                 infotext.text = "Βάλε νερό στο δοχείο και στη λεκάνη.";
137
138                 mpoukaliislifted = true;
139             }
140             else if (selection.CompareTag(doxieioemptyExp4Tag) && mpoukaliislifted==true && readytomeasure1 == false)
141             {
142                 doxieioempty.SetActive(false);
143                 doxieiowater.SetActive(true);
144
145                 doxieiofilled = true;
146             }
147         }
148     }
149 }

```

Εικόνα 8.7 Η μέθοδος Update() του script Kef5pirama1script.

```

259 void endexperiment()
260 {
261     TelosButton.SetActive(true);
262 }

```

Εικόνα 8.8 Ο έλεγχος που οδηγεί στην εμφάνιση του κουμπιού τέλους.

Τέλος, στο πείραμα αυτό γίνεται χρήση animation προκειμένου ο χρήστης να μπορεί να παρατηρεί άμεσα και με τρόπο παραστατικό τις αλλαγές που επέρχονται στην θερμοκρασία όταν αυτός τοποθετεί τα θερμομέτρα μέσα στο υγρά του δοχείου και της λεκάνης (Εικόνα 8.9).

```

if (readytomeasure1==true && readytomeasure2==true)
{
    putthermometerinlekani = false;
    infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
    infotext.text = "Παρατήρησε τι συμβαίνει με τις θερμοκρασίες.";

    thermometerindicatorimage.GetComponent<Animator>().Play("100to65anim");
    thermometerindicatorimage2.GetComponent<Animator>().Play("20to65anim");
    Invoke("endexperiment", 15.0f);
}

```

Εικόνα 8.9 Χρήση animation μέσω script.

```

265 void waterboiled()
266 {
267     Debug.Log("gotinvoked");
268     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
269     infotext.text = "Πάρε το δοχείο και βάλτο μέσα στην λεκάνη.";
270     waterisready = true;
271 }

```

Εικόνα 8.10 Η μέθοδος Update() του script Kef5pirama1script.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή αποκαλύπτει στον χρήστη τις αρχικές θερμοκρασίες των θερμόμετρων και τις αρχικές θερμοκρασίες των υγρών των δοχείων πριν γίνει η οποιαδήποτε ροή θερμοκρασίας. Ο μαθητής μπορεί να συγκρίνει τα δικά του αποτελέσματα με τα αποτελέσματα που του παρουσιάζει η εφαρμογή. Εάν ο χρήστης έχει κάνει κάποιο λάθος στις μετρήσεις του, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ». Έπειτα, η εφαρμογή θέτει κάποιες ερωτήσεις στον χρήστη σχετικά με το πείραμα που μόλις υλοποίησε. Για την απάντηση των ερωτήσεων αυτών ο μαθητής μπορεί να ζητήσει βοήθεια από κάποιο άλλο άτομο. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα (Εικόνα 8.11).



Εικόνα 8.11 Η διδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 5<sup>ου</sup> Κεφαλαίου.

### 8.3 Επίλογος

Το πείραμα του 5ου κεφαλαίου «Από τη θερμότητα στη θερμοκρασία – Η θερμική ισορροπία» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης τοποθετεί νερό στην λεκάνη και στο δοχείο, τροποποιεί κατάλληλα τις θερμοκρασίες αυτών, τοποθετεί το δοχείο μέσα στην λεκάνη και μετράει τις θερμοκρασίες των περιεχομένων τους. Η μέτρηση γίνεται με την χρήση δύο θερμομέτρων που συλλέγει και τοποθετεί ο χρήστης. Τέλος, καταγράφει τις τιμές της θερμοκρασίας, παρατηρεί τι συμβαίνει και στην συνέχεια απαντάει σε ερωτήματα που αναγράφονται στην σκηνή τέλους του παιχνιδιού.



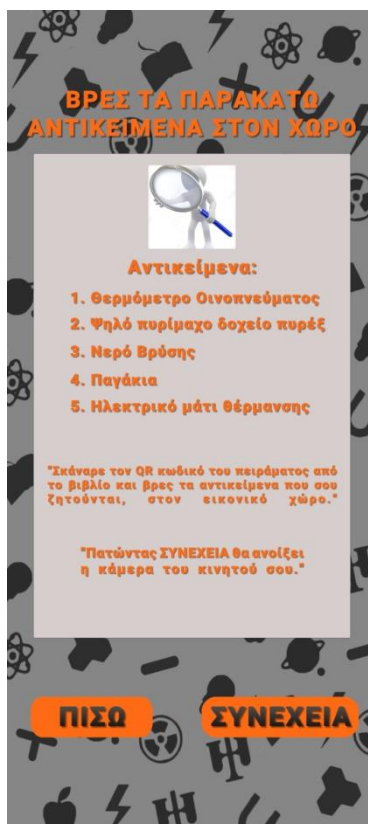
## Κεφάλαιο 9ο: Οι αλλαγές κατάστασης του νερού – Ο «κύκλος» του νερού

### 9.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την μέτρηση της θερμοκρασίας με την χρήση θερμομέτρων και παρατήρηση της κατάστασης του νερού. Στο συγκεκριμένο πείραμα ο μαθητής χρησιμοποιεί ένα θερμόμετρο για την μέτρηση της θερμοκρασίας του νερού στις διαφορετικές καταστάσεις του.

### 9.2 Κατασκευή του πειράματος

Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια και από πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει ένα θερμόμετρο οινοπνεύματος, ένα πυρίμαχο δοχείο, μία κανάτα με νερό, ένα δοχείο με παγάκια και ένα ηλεκτρικό μάτι θέρμανσης. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 9.1).



Εικόνα 9.1 Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 6<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού. Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή (Εικόνα 9.2).



Εικόνα 9.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 6<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 3) η εφαρμογή εξηγεί συνοπτικά στον χρήστη, τα βήματα που πρέπει να ακολουθήσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να παρατηρήσει την αλλαγή κατάστασης του νερού (Εικόνα 9.3).



Εικόνα 9.3 Η σκηνή της τρίτης φάσης του πειράματος του 6<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά ο χρήστης πρέπει να σηκώσει από το τραπέζι την κανάτα με το νερό και να γεμίσει το δοχείο. Στη συνέχεια, πρέπει να γεμίσει το δοχείο με παγάκια, να σηκώσει το θερμόμετρο, να το βάλει μέσα στο δοχείο και τέλος να βάλει το δοχείο επάνω στο ηλεκτρικό μάτι. Αυτή τη φορά το ηλεκτρικό μάτι δεν είναι εξαρχής αναμμένο αλλά ο χρήστης πρέπει να πατήσει το κουμπί που εμφανίζεται στην οθόνη του εκείνη τη στιγμή για να το ανάψει. Έπειτα, ο μαθητής περιμένει λίγα δευτερόλεπτα να ζεσταθεί το νερό για να φτάσει σε σημείο βρασμού. Όσο η θερμοκρασία του νερού αλλάζει, ο χρήστης πρέπει να παρακολουθεί την κατάσταση του νερού παράλληλα με τις θερμοκρασίες. Η θερμοκρασία του υγρού του δοχείου αλλάζει δυναμικά και αναγράφεται στο θερμόμετρο που βρίσκεται σταθερά επάνω στην οθόνη του χρήστη (Εικόνα 9.4).



Εικόνα 9.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 6<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα `Kefbpirama1script.cs`. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές εικοσιπέντε μεταβλητών τύπου `boolean` που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές εννέα μεταβλητών τύπου `string` που θα χρησιμοποιηθούν ως `tags` πάνω στα αντικείμενα της σκηνής και δεκαέξι μεταβλητών τύπου `GameObject` που θα κρατάνε τα δεδομένα των δεκαέξι αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου `TextMeshProUGUI` που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 9.5).

```

9 public class Kef6pirama1script : MonoBehaviour
10 {
11     //selection---
12     private Transform selection;
13
14     public int piramaindex = 0;
15
16     public bool calledonce = false;
17
18     public bool thermometroisactive = false;
19     public bool pagakiaislifted = false;
20     public bool kanataislifted = false;
21     public bool doxeiowaterislifted = false;
22
23     public bool doxeiowaterICEwiththermometerislifted = false;
24
25     public bool readytostart = false;
26
27     public bool doxeiofilled = false;
28     public bool lekanifilled = false;
29
30     public bool readytogetthermometer = false;
31
32     public bool thermometroislifted = false;
33
34     public bool allfilled = false;
35
36     public bool doxeiowateronlyready = false;
37     public bool doxeiowaterICeready = false;
38     public bool doxeiowaterBOILready = false;
39
40     public bool readytomeasure1 = false;
41     public bool readytomeasure2 = false;
42
43     public bool telosindex1 = false;
44     public bool telosindex2 = false;
45     public bool telosindex3 = false;

```

Εικόνα 9.5 Αρχικοποίηση μεταβλητών του script Kef6pirama1script.

Έπειτα, στην μέθοδο Awake() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικειμένου (infotext) της σκηνής και απενεργοποίηση του αντικειμένου TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο αυτό (TelosButton) το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 9.6).

```

99 private void Awake()
100 {
101     Application.targetFrameRate = 30;
102     Debug.Log("Into the Script!!!");
103
104     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
105     infotext.text = "Πάρε την κανάτα με το νερό.";
106
107     TelosButton.SetActive(false);
108
109 }

```

Εικόνα 9.6 Η μέθοδος Awake() του script Kef6pirama1script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του

ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε ο χρήστης «κρατάει» αυτό το αντικείμενο και η εφαρμογή περιμένει από αυτόν να το χρησιμοποιήσει ή να τοποθετήσει κάπου αλλού. Αρχικά, ο χρήστης επιλέγει και «κρατάει» την κανάτα με το νερό το οποίο έπειτα χρησιμοποιεί πατώντας πάνω στο δοχείο για να το γεμίσει. Στη συνέχεια, ο χρήστης πρέπει με παρόμοιο τρόπο να πάρει και να τοποθετήσει παγάκια και στη συνέχεια το θερμόμετρο μέσα στο δοχείο. Όταν ο χρήστης τοποθετήσει το θερμόμετρο μέσα στο δοχείο καλείται μία `SetActive()` για την ενεργοποίηση του `image` με μορφή θερμόμετρου στην οθόνη του χρήστη. Έπειτα, ο μαθητής τοποθετεί το δοχείο πάνω στο ηλεκτρικό μάτι. Κάνοντας αυτό του εμφανίζεται στην οθόνη το κουμπί «ON». Το κουμπί αυτό όταν πατηθεί καλεί την μέθοδο που του έχει οριστεί στο `OnClick()` state δηλαδή την μέθοδο `startheating()`. Τότε ξεκινάει να παίζει το ανάλογο `animation` του θερμόμετρου που μας δείχνει την θερμοκρασία να αυξάνεται και καλείτε μετά από δέκα δευτερόλεπτα λόγω της `Invoke` η νέα μέθοδος που ακολουθεί (`getdoxeiowater`). Η `Invoke` μπορεί να καλέσει μία μέθοδο μετά από μία συγκεκριμένη ώρα που θα ορίσει ο χρήστης (Εικόνα 9.7).

```

216 public void startheating()
217 {
218     SwitchOnButton.SetActive(false);
219
220     thermometerindicatorimage.GetComponent<Animator>().Play("0to100slowanim");
221
222     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
223     infotext.text = "Παρατήρησε και μέτρα τις θερμοκρασίες!";
224
225     Invoke("getdoxeiowater", 10.0f);
226 }

```

Εικόνα 9.7 Η μέθοδος `startheating()` του script `Kef6pirama1script`.

Ο χρήστης περιμένει μερικά δευτερόλεπτα να ζεσταθεί το νερό, περιμένει δηλαδή να καλεστεί η μέθοδος που πρέπει και παρατηρεί τον πάγο να λιώνει. Τότε εμφανίζεται το ανάλογο μήνυμα στην οθόνη του χρήστη και η θερμοκρασία συνεχίζει να ανεβαίνει. Μέσα στην μέθοδο `getdoxeiowater()` γίνεται άλλη μια χρήση της `Invoke()` όπου μετά από άλλα δέκα δευτερόλεπτα καλεί την μέθοδο `getdoxeioBOIL()` (Εικόνα 9.8).

```

228 void getdoxeiowater()
229 {
230     doxeiowaterICEwiththermometer2.SetActive(false);
231     doxeiowaterwiththermometer.SetActive(true);
232
233     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
234     infotext.text = "Ο πάγος έλιωσε!!!";
235
236     Invoke("getdoxeioBOIL", 10.0f);
237 }

```

Εικόνα 9.8 Η μέθοδος `getdoxeiowater()` του script `Kef6pirama1script`.

Η μέθοδος `getdoxeioBOIL()` ενεργοποιεί το `particle effect` του ατμού, ενημερώνει το `text` και ενεργοποιεί το κουμπί τέλους της σκηνής με την `SetActive()`. Αφού το νερό είναι έτοιμο, ο μαθητής παρατηρεί την τελική θερμοκρασία και κατάσταση του νερού (σημείο βρασμού, βγαίνουν ατμού).

Όλες αυτές οι διαδικασίες επιτυγχάνονται μέσα στο script με την συνεχή ενεργοποίηση και απενεργοποίηση των αντικειμένων της σκηνής ανάλογα με τις κινήσεις του χρήστη. Η ενεργοποίηση και η απενεργοποίηση των αντικειμένων αυτών γίνεται με την μέθοδο setActive(false/true) με τιμή true ή false. Σε αυτό το σημείο, εμφανίζεται το κουμπί «ΤΕΛΟΣ». Εφόσον αυτό έχει συμβεί, δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί τέλους και να μεταφερθεί στην επόμενη σκηνή αφού φυσικά έχει πρώτα παρατηρήσει και καταγράψει τις τιμές των θερμοκρασιών. Το κουμπί αυτό δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω (Εικόνα 9.9).

```

239 void getdoxeioBOIL()
240 {
241     SmokeKef6.SetActive(true);
242     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
243     infotext.text = "Το νερό άρχισε να βράζει!";
244     TelosButton.SetActive(true);
245 }
    
```

Εικόνα 9.9 Η μέθοδος getdoxeioBOIL() του script Kef6pirama1script.

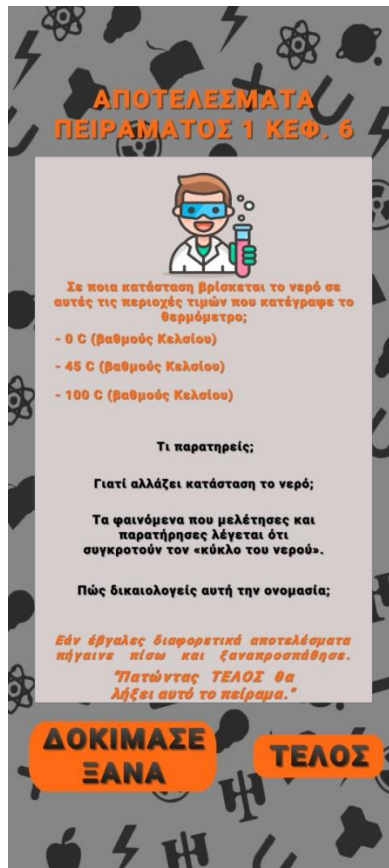
Τέλος, στο πείραμα αυτό γίνεται χρήση animation προκειμένου ο χρήστης να μπορεί να παρατηρεί άμεσα και με τρόπο παραστατικό τις αλλαγές που επέρχονται στην θερμοκρασία όταν αυτός τοποθετεί το θερμόμετρο μέσα στο υγρό του δοχείου και της λεκάνης (Εικόνα 9.10).

```

thermometerindicatorimage.GetComponent<Animator>().Play("0to100slowanim");
    
```

Εικόνα 9.10 Χρήση animation μέσω script.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή αποκαλύπτει στον χρήστη κάποιες θερμοκρασίες του υγρού του δοχείου που κατέγραψε το θερμόμετρο. Ο μαθητής μπορεί να συγκρίνει τα δικά του αποτελέσματα με τα αποτελέσματα που του παρουσιάζει η εφαρμογή. Εάν ο χρήστης έχει κάνει κάποιο λάθος στις μετρήσεις του ή θέλει να μελετήσει ξανά το πείραμα, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ». Έπειτα, η εφαρμογή θέτει κάποιες ερωτήσεις στον χρήστη σχετικά με το πείραμα που μόλις υλοποίησε και τις θερμοκρασίες που κατέγραψε. Για την απάντηση των ερωτήσεων αυτών ο μαθητής μπορεί να ζητήσει βοήθεια από κάποιο άλλο άτομο. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα (Εικόνα 9.11).



Εικόνα 9.11 Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 6<sup>ου</sup> Κεφαλαίου.

### 9.3 Επίλογος

Το πείραμα του 6ου κεφαλαίου «Οι αλλαγές κατάστασης του νερού – Ο κύκλος του νερού» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης προσθέτει νερό στο δοχείο που είναι τοποθετημένο πάνω στο εικονικό γραφείο στην σκηνή επαυξημένης πραγματικότητας και τροποποιεί κατάλληλα την θερμοκρασία του νερού. Η μέτρηση γίνεται με την χρήση ενός θερμομέτρου που ο χρήστης συλλέγει. Κατά την διάρκεια αλλαγής της θερμοκρασίας του νερού ο χρήστης καταγράφει τις διαφορετικές τιμές και παρατηρεί τις αλλαγές των καταστάσεων του νερού. Τέλος, απαντά σε ερωτήματα που αναγράφονται στην σκηνή τέλους του παιχνιδιού.

## Κεφάλαιο 10ο: Η διαστολή και συστολή του νερού – Μια φυσική «ανωμαλία»

### 10.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την κατανόηση της διαστολής και της συστολής του νερού. Στο συγκεκριμένο πείραμα ο μαθητής χρησιμοποιεί δύο μπουκάλια με υγρά και μία κατάψυξη για την ψύξη των υγρών αυτών.

### 10.2 Κατασκευή του πειράματος

Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια και από πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει δύο πλαστικά καπάκια, ένα μπουκάλι λάδι, δύο ποτήρια, ένα μπουκάλι νερό, ένα διαφανές πλαστικό δοχείο, ένα θερμόμετρο οιοπνεύματος και ένα δοχείο με παγάκια. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 10.1).



Εικόνα 10.1 Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 7<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού. Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο ένα μπουκάλι νερό, δύο ποτήρια, ένα μπουκάλι λάδι, ένα θερμόμετρο, ένα δοχείο πυρέξ, δύο καπάκια και ένα δοχείο με παγάκια. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή (Εικόνα 10.2).



Εικόνα 10.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 7<sup>ου</sup> Κεφαλαίου.

Στην σκηνή αυτή, (Φάση 3) η εφαρμογή εξηγεί συνοπτικά στον χρήστη, τις ενέργειες που πρέπει να εκτελέσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να παρατηρήσει την διαστολή και την συστολή του νερού (Εικόνα 10.3).



Εικόνα 10.3 Η σκηνή της τρίτης φάσης του πειράματος του 7<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά ο χρήστης πρέπει να σηκώσει από το τραπέζι το νερό και το λάδι και στη συνέχεια να γεμίσει τα δύο μπουκάλια με νερό και λάδι αντίστοιχα. Μετά, πρέπει να πάρει τα μπουκάλια αυτά, να τα τοποθετήσει μέσα στην κατάψυξη, να κλείσει την κατάψυξη και να πατήσει το κουμπί για να αρχίσει να δουλεύει η κατάψυξη προκειμένου να παγώσουν τα υγρά (Εικόνα 10.4). Μετά από λίγο αφού τα υγρά έχουν παγώσει ο χρήστης πρέπει να βγάλει τα μπουκάλια από εκεί μέσα και να τα τοποθετήσει μπροστά του προκειμένου να παρατηρήσει τι τους συνέβη (Εικόνα 10.5).



Εικόνα 10.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 7<sup>ου</sup> Κεφαλαίου.



Εικόνα 10.5 Τελική φάση τρισδιάστατης σκηνής της τέταρτης φάσης του πειράματος του 7<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα Kef7pirama1.cs. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές δώδεκα μεταβλητών τύπου boolean που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές δέκα μεταβλητών τύπου string που θα χρησιμοποιηθούν ως tags πάνω στα αντικείμενα της σκηνής και δεκατεσσάρων μεταβλητών τύπου GameObject που θα κρατάνε τα δεδομένα των δεκατεσσάρων αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου TextMeshProUGUI που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 10.6).

```

9 public class Kef7pirama1 : MonoBehaviour
10 {
11
12     //selection---
13     private Transform selection;
14
15     public int piramaindex = 0;
16
17     public bool calledonce = false;
18
19     public bool kanataislifted = false;
20     public bool mpoukaliemptynerofilled = false;
21     public bool mpoukaliemptyladifilled = false;
22     public bool ladiislifted = false;
23
24     public bool mpoukalinerofilledislifted = false;
25     public bool mpoukaliladifilledislifted = false;
26
27     public bool bothlifted = false;
28
29     public bool readytoclose = false;
30
31     public bool katapsiksiclosed = false;
32
33     public bool FreezerReadytoopen = false;
34
35     public bool katapsiksiopen = false;
36
37     Collider freezercollider;
38
39
40
41
42     // [SerializeField] private string selectableTag = "Selectable";
43     [SerializeField] private string kanataExp6Tag = "kanataExp6Tag";
44     [SerializeField] private string LadiExp7Tag = "LadiExp7Tag";
45     [SerializeField] private string katapsiksiTag = "katapsiksiTag";
46     [SerializeField] private string MpoukaliEmptyNew2Tag = "MpoukaliEmptyNew2Tag";
47     [SerializeField] private string MpoukaliNeroFilledTag = "MpoukaliNeroFilledTag";

```

Εικόνα 10.6 Αρχικοποίηση μεταβλητών του script Kef7pirama1script.

Έπειτα, στην μέθοδο Awake() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικειμένου (infotext) της σκηνής, αποθήκευση του collider (απαραίτητο για την επιλογή ενός αντικειμένου με «κλικ») στοιχείου της κατάψυξης και απενεργοποίηση του αντικειμένου TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο αυτό (TelosButton) το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 10.7).

```

78 private void Awake()
79 {
80     Application.targetFrameRate = 30;
81     Debug.Log("Into the Script!!!");
82
83     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
84     infotext.text = "Πάρτε την κανάτα με το νερό.";
85
86     TelosButton.SetActive(false);
87
88     freezercollider = katapsiksi.GetComponent<Collider>();
89 }

```

Εικόνα 10.7 Η μέθοδος Awake() του script Kef7pirama1script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε ο χρήστης «κρατάει» αυτό το αντικείμενο και η εφαρμογή περιμένει από αυτόν να το χρησιμοποιήσει ή να τοποθετήσει κάπου αλλού. Αρχικά, ο χρήστης επιλέγει και «κρατάει» τα μπουκάλια με το νερό και το λάδι και έπειτα τα χρησιμοποιεί πατώντας πάνω στα δύο άδεια μπουκάλια της σκηνής για να τα γεμίσει με νερό και λάδι αντίστοιχα. Στη συνέχεια, ο χρήστης πρέπει με παρόμοιο τρόπο να πάρει και να τοποθετήσει τα μπουκάλια μέσα στην κατάψυξη. Όταν ο χρήστης τοποθετήσει τα μπουκάλια μέσα στην κατάψυξη, μετά πρέπει πατώντας πάνω σε αυτήν να την κλείσει (εδώ παίζει το ανάλογο animation) και στη συνέχεια να πατήσει το κουμπί «FREEZE» που εμφανίζεται στην οθόνη του προκειμένου να αρχίσουν να παγώνουν τα υγρά. Το κουμπί αυτό όταν πατηθεί καλεί την μέθοδο που του έχει οριστεί στο Onclick() state, δηλαδή την μέθοδο Freeze(). Η μέθοδος αυτή όταν καλείτε ενημερώνει κατάλληλα το κείμενο του text, ενεργοποιεί τα αντικείμενα που πρέπει και καλεί με την Invoke() μετά από πέντε δευτερόλεπτα την μέθοδο FreezeBottles(). Η Invoke μπορεί να καλέσει μία μέθοδο μετά από μία συγκεκριμένη ώρα που θα ορίσει ο χρήστης (Εικόνα 10.8).

```

226 public void Freeze()
227 {
228     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
229     infotext.text = "Τα υγρά παγώνουν!!!";
230
231     MpoukaliNeroFilledinFreezer.SetActive(false);
232     MpoukaliExploded.SetActive(true);
233
234     FreezeButton.SetActive(false);
235     Invoke("FreezeBottles", 5.0f);
236 }

```

Εικόνα 10.8 Η μέθοδος Freeze() του script Kef7pirama1script.

Όταν η εφαρμογή ενημερώσει τον χρήστη ότι το περιεχόμενο των μπουκαλιών έχει παγώσει τότε μόνο ο χρήστης μπορεί να ανοίξει την κατάψυξη και να πάρει τα μπουκάλια από μέσα (Εικόνα 10.9) (Εικόνα 10.10).

```

199 if(selection.CompareTag(katapsiksiTag) && FreezerReadytoopen == true)
200 {
201     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
202     infotext.text = "Βγάλε τα μπουκάλια από την κατάψυξη.";
203
204     katapsiksi.GetComponent<Animator>().Play("FreezerOpen");
205
206     Invoke("readytotake", 3.0f);
207     FreezerReadytoopen = false;
208 }

```

Εικόνα 10.9 Κάλεισμα της μεθόδου readytotake().

```

250 public void readytotake()
251 {
252     katapsiksiopen = true;
253     freezercollider.enabled = !freezercollider.enabled;
254     Debug.Log("ready");
255 }
256

```

Εικόνα 10.10 Η μέθοδος readytotake() του script Kef7pirama1script.

Τέλος, όταν ο χρήστης επιλέξει τα μπουκάλια, αυτά εμφανίζονται αυτόματα μπροστά στον του για λόγους ευκολίας και μπορεί να παρατηρήσει τι έχει συμβεί στα μπουκάλια αυτά. Μετά από λίγα δευτερόλεπτα καλείτε και η μέθοδος που εμφανίζει το κουμπί τέλους (Εικόνα 10.11).

```

257 public void endexperiment()
258 {
259     TelosButton.SetActive(true);
260 }

```

Εικόνα 10.11 Η μέθοδος που εμφανίζει το κουμπί τέλους.

Όλες αυτές οι διαδικασίες επιτυγχάνονται μέσα στο script με την συνεχή ενεργοποίηση και απενεργοποίηση των αντικειμένων της σκηνής ανάλογα με τις κινήσεις του χρήστη. Η ενεργοποίηση και η απενεργοποίηση των αντικειμένων αυτών γίνεται με την μέθοδο setActive(false/true) με τιμή true ή false. Εφόσον έχει εμφανιστεί το κουμπί τέλους στη σκηνή, δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί αυτό και να μεταφερθεί στην επόμενη σκηνή αφού φυσικά έχει πρώτα παρατηρήσει τι έχει συμβεί. Το κουμπί αυτό δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω (Εικόνα 10.12).

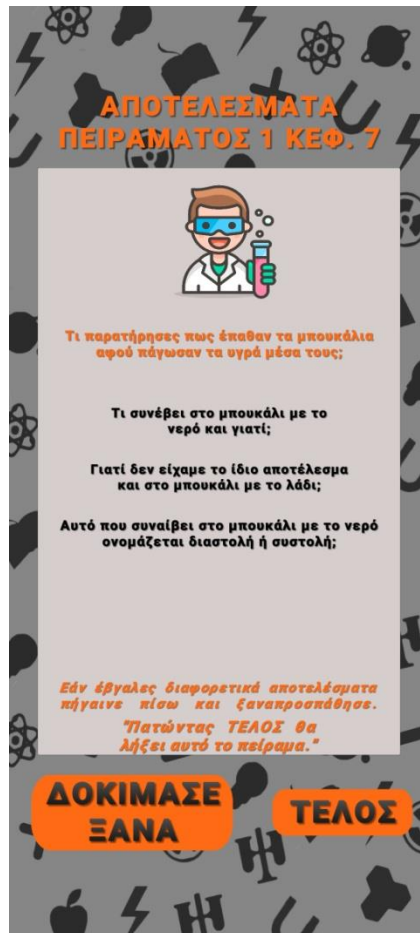
```

101 void Update()
102 {
103     if (Input.GetMouseButtonDown(0))
104     {
105         // Debug.Log("Mouse Button Down!!!");
106         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
107         RaycastHit hit;
108
109         if (Physics.Raycast(ray, out hit))
110         {
111             // Debug.Log("Hit Something!!!");
112             var selection = hit.transform;
113             if (selection.CompareTag(kanataExp6Tag))
114             {
115                 // Kremastra.transform.localPosition = new Vector3(-0.0057f, 1.8067f, 0.076f);
116                 // Kremastra.transform.localRotation = Quaternion.Euler(0, 90, 90f);
117
118                 KanataNero.SetActive(false);
119
120                 infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
121                 infotext.text = "Βάλε νερό στο αριστερό μπουκάλι.";
122                 Debug.Log("kanalifted");
123                 kanataislifted = true;
124             }
125             else if (selection.CompareTag(MpoukaliEmptyNew2Tag) && kanataislifted == true)
126             {
127                 MpoukaliEmptyNew2.SetActive(false);
128                 MpoukaliNeroFilled.SetActive(true);
129                 Debug.Log("filledwithwater");
130                 infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
131                 infotext.text = "Πάρε το λάδι.";
132
133                 mpoukaliemptynerofilled = true;
134             }
135         }
136     }
137 }

```

Εικόνα 10.12 Η μέθοδος Update() του script Kef7pirama1script.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή θέτει κάποιες ερωτήσεις στον χρήστη σχετικά με το πείραμα που μόλις υλοποίησε και τα φαινόμενα που παρατήρησε. Για την απάντηση των ερωτήσεων αυτών ο μαθητής μπορεί να ζητήσει βοήθεια από κάποιο άλλο άτομο. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα. Εάν ο χρήστης επιθυμεί να μελετήσει ξανά το πείραμα, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ» (Εικόνα 10.13).



Εικόνα 10.13 Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 7<sup>ου</sup> Κεφαλαίου.

### 10.3 Επίλογος

Το πείραμα του 7ου κεφαλαίου «Η διαστολή και συστολή του νερού – Μια φυσική ανωμαλία» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης τοποθετεί λάδι και νερό μέσα σε δύο μπουκάλια αντίστοιχα, βάζει τα μπουκάλια μέσα στην κατάψυξη, περιμένει να παγώσουν και τέλος τα βγάζει και τα τοποθετεί επάνω στο εικονικό τραπέζι. Ο χρήστης παρατηρεί τι έχει συμβεί στα υγρά των μπουκαλιών και στην συνέχεια απαντάει στα ερωτήματα που αναγράφονται στην σκηνή τέλους του παιχνιδιού.

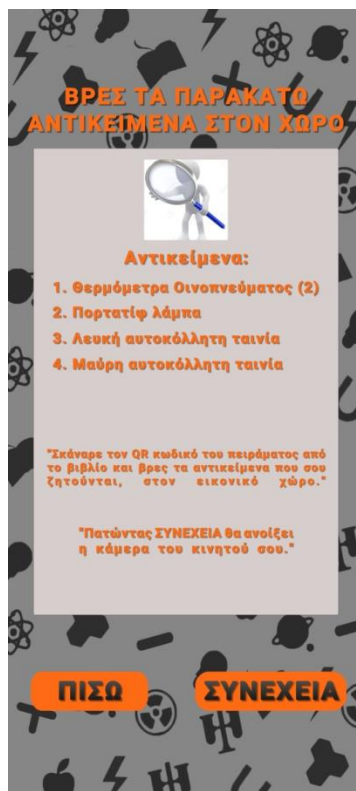
## Κεφάλαιο 11ο: Το φως θερμαίνει – «ψυχρά» και «θερμά» χρώματα

### 11.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την μέτρηση της θερμοκρασίας με την χρήση θερμομέτρου και του φωτός. Στο συγκεκριμένο πείραμα ο μαθητής χρησιμοποιεί ένα θερμοόμετρο για την μέτρηση της θερμοκρασίας που παράγεται από το φως ενός πορτατίφ.

### 11.2 Κατασκευή του πειράματος

Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια και από πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει δύο θερμοόμετρα οιοπνεύματος, ένα πορτατίφ λάμπα, μία λευκή αυτοκόλλητη ταινία και μία μαύρη αυτοκόλλητη ταινία. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνες 11.1).



Εικόνα 11.1 Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 8<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα

την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού. Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο μία κολλητική ταινία μαύρη, δύο θερμομέτρα, ένα φωτιστικό και μία κολλητική ταινία λευκή. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή (Εικόνα 11.2).



Εικόνα 11.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 8<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 3) η εφαρμογή εξηγεί συνοπτικά στον χρήστη, τα βήματα που πρέπει να εκτελέσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να παρατηρήσει την θερμότητα που παράγεται λόγω του φωτός (Εικόνα 11.3).



Εικόνα 11.3 Η σκηνή της τρίτης φάσης του πειράματος του 8<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά ο χρήστης πρέπει να σηκώσει από το τραπέζι το θερμόμετρο και στη συνέχεια να το τοποθετήσει στην θέση που του υποδεικνύει το εικονικό ολόγραμμα που εμφανίζεται στη σκηνή. Ο χρήστης, πατώντας πάνω στο εικονικό ολόγραμμα τοποθετεί το αντικείμενο που «κρατάει» στη θέση εκείνη. Σε αυτό το σημείο ο μαθητής πρέπει να παρατηρήσει και να καταγράψει την θερμοκρασία. Μετά από λίγο δίνεται στον μαθητή η δυνατότητα να ανάψει το φως του πορτατίφ πατώντας επάνω του. Ο μαθητής ανάβει το φως και παρατηρεί τι συμβαίνει στην θερμοκρασία. Όταν η θερμοκρασία σταθεροποιηθεί ο μαθητής μπορεί να πατήσει το κουμπί τέλους για να πάει στην επόμενη σκηνή (Εικόνα 11.4).



Εικόνα 11.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 8<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα `Kef8pirama1.cs`. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές τριών μεταβλητών τύπου `boolean` που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές τριών μεταβλητών τύπου `string` που θα χρησιμοποιηθούν ως tags πάνω στα αντικείμενα της σκηνής και οκτώ μεταβλητών τύπου `GameObject` που θα κρατάνε τα δεδομένα των οκτώ αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου `TextMeshProUGUI` που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 11.5).

```

9 public class Kef8pirama1 : MonoBehaviour
10 {
11     //selection----
12     private Transform selection;
13
14     public bool thermometroislifted = false;
15     public bool readytoswitchon = false;
16     public bool lighton = false;
17
18
19     // [SerializeField] private string selectableTag = "Selectable";
20     [SerializeField] private string ThermometroTag = "Thermometro";
21     [SerializeField] private string thermometrohologramTag = "thermometrohologramTag";
22     [SerializeField] private string FotistikoKef8Tag = "FotistikoKef8Tag";
23
24
25
26
27     public GameObject Thermometro;
28     public GameObject ThermometroHologram;
29     public GameObject ThermometroFinal;
30     public GameObject Fotistiko;
31     public GameObject SpotLightFotistiko;
32
33     public GameObject TelosButton;
34     private TextMeshProUGUI infotext;
35
36     public GameObject thermometerimagezoomed;
37     public GameObject thermometerindicatorimagezoomed;

```

Εικόνα 11.5 Αρχικοποίηση μεταβλητών του script Kef8pirama1script.

Έπειτα, στην μέθοδο Awake() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικειμένου (infotext) της σκηνής και απενεργοποίηση του αντικειμένου TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο αυτό (TelosButton) το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 11.6).

```

39 private void Awake()
40 {
41     Application.targetFrameRate = 30;
42     Debug.Log("Into the Script!!!");
43
44     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
45     infotext.text = "Πάρε το θερμόμετρο.";
46
47     TelosButton.SetActive(false);
48
49 }

```

Εικόνα 11.6 Η μέθοδος Awake() του script Kef8pirama1script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if,

χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε ο χρήστης «κρατάει» αυτό το αντικείμενο και η εφαρμογή περιμένει από αυτόν να το χρησιμοποιήσει ή να τοποθετήσει κάπου αλλού. Αρχικά, ο χρήστης επιλέγει και «κρατάει» το θερμόμετρο που βρίσκεται επάνω στο τραπέζι και έπειτα το τοποθετεί στην θέση του εικονικού ολογράμματος (εικονικό ολόγραμμα θερμομέτρου) που εμφανίζεται μπροστά του. Το εικονικό ολόγραμμα είναι ένα αντικείμενο που ενεργοποιείται προσωρινά στη σκηνή για να δείξει στον χρήστη το που πρέπει να αφήσει το αντικείμενο που κρατάει την δεδομένη χρονική στιγμή. Όταν ο χρήστης τοποθετήσει το θερμόμετρο καλείται μία setActive() για την ενεργοποίηση του image με μορφή θερμόμετρου στην οθόνη του χρήστη. Στη συνέχεια, ο χρήστης περιμένει μερικά δευτερόλεπτα προκειμένου το script να καλέσει την μέθοδο getText(). Η Invoke μπορεί να καλέσει μία μέθοδο μετά από μία συγκεκριμένη ώρα που θα ορίσει ο χρήστης (Εικόνα 11.7).

```

else if (selection.CompareTag(thermometrohologramTag) && thermometroislifted == true)
{
    ThermometroHologram.SetActive(false);
    ThermometroFinal.SetActive(true);

    infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
    infotext.text = "Παρατήρησε την θερμοκρασία.";

    Invoke("gettext", 5.0f);

    thermometerimagezoomed.SetActive(true);
    thermometerindicatorimagezoomed.SetActive(true);
}

```

Εικόνα 11.7 Κάλεσμα της μεθόδου getText().

Ο χρήστης την ώρα που περιμένει πρέπει να παρατηρήσει και να καταγράψει την θερμοκρασία που δείχνει το θερμόμετρο. Στη συνέχεια ο χρήστης πρέπει να ανάψει το φως πατώντας πάνω στο πορτατίφ. Όταν ανάψει το φως, ο μαθητής πρέπει να παρατηρεί συνεχώς τι συμβαίνει με την θερμοκρασία ως ότου αυτή σταθεροποιηθεί (Εικόνα 11.8). Μετά από λίγα δευτερόλεπτα, αφού η θερμοκρασία έχει σταθεροποιηθεί, καλείτε η μέθοδος που εμφανίζει το κουμπί τέλους (Εικόνα 11.9). Όλες αυτές οι διαδικασίες επιτυγχάνονται μέσα στο script με την συνεχή ενεργοποίηση και απενεργοποίηση των αντικειμένων της σκηνής ανάλογα με τις κινήσεις του χρήστη. Η ενεργοποίηση και η απενεργοποίηση των αντικειμένων αυτών γίνεται με την μέθοδο setActive(false/true) με τιμή true ή false. Εφόσον έχει εμφανιστεί το κουμπί τέλους στη σκηνή (Εικόνα 11.10), δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί αυτό και να μεταφερθεί στην επόμενη σκηνή αφού φυσικά έχει πρώτα παρατηρήσει τι έχει συμβεί με την τελική θερμοκρασία. Το κουμπί τέλους δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω.

```
60 void Update()
61 {
62     if (Input.GetMouseButtonDown(0))
63     {
64         // Debug.Log("Mouse Button Down!!!");
65         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
66         RaycastHit hit;
67
68         if (Physics.Raycast(ray, out hit))
69         {
70             // Debug.Log("Hit Something!!!");
71             var selection = hit.transform;
72             if (selection.CompareTag(ThermometroTag) )
73             {
74                 // Kremastra.transform.localPosition = new Vector3(-0.0057f, 1.8067f, 0.076f);
75                 // Kremastra.transform.localRotation = Quaternion.Euler(0, 90, 90f);
76
77                 Thermometro.SetActive(false);
78                 ThermometroHologram.SetActive(true);
79
80                 infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
81                 infotext.text = "Τοποθέτησε το θερμόμετρο κάτω από το φωτιστικό.";
82
83                 thermometroislifted = true;
84             }
85         }
86     }
87 }
```

Εικόνα 11.8 Η μέθοδος Update() του script Kef8pirama1script.

```
99 if (selection.CompareTag(FotistikoKef8Tag) && readytoswitchon == true)
100 {
101
102     if (lighton == false)
103     {
104         SpotLightFotistiko.SetActive(true);
105         thermometerindicatorimagezoomed.GetComponent<Animator>().Play("20to30slowanim");
106     }
107     else
108     {
109         SpotLightFotistiko.SetActive(false);
110         thermometerindicatorimagezoomed.GetComponent<Animator>().Play("30to20slowanim");
111     }
112
113     lighton = !lighton;
114
115     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
116     infotext.text = "Παρατήρησε την θερμοκρασία με φως (και χωρίς).";
117
118     Invoke("endexperiment", 9.0f);
119 }
```

Εικόνα 11.9 Κάλεσμα της μεθόδου endexperiment().

```
132 public void endexperiment()
133 {
134     TelosButton.SetActive(true);
135 }
```

Εικόνα 11.10 Η μέθοδος που εμφανίζει το κουμπί τέλους.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή θέτει κάποιες ερωτήσεις στον χρήστη σχετικά με το πείραμα που μόλις υλοποίησε και τα φαινόμενα που παρατήρησε. Για την

απάντηση των ερωτήσεων αυτών ο μαθητής μπορεί να ζητήσει βοήθεια από κάποιο άλλο άτομο. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα. Εάν ο χρήστης επιθυμεί να μελετήσει ξανά το πείραμα, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ» (Εικόνα 11.11).



Εικόνα 11.11 Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 8<sup>ου</sup> Κεφαλαίου.

### 11.3 Επίλογος

Το πείραμα του 8ου κεφαλαίου «Το φως θερμαίνει – ψυχρά και θερμά χρώματα» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης συλλέγει ένα θερμόμετρο που είναι τοποθετημένο πάνω στο εικονικό γραφείο στην σκηνή επαυξημένης πραγματικότητας, το βάζει κάτω από το φωτιστικό, ανάβει το φως και παρατηρεί τι συμβαίνει στην θερμοκρασία που δείχνει το θερμόμετρο. Ο χρήστης καταγράφει τις τιμές, παρατηρεί τι συμβαίνει στην θερμοκρασία και τέλος απαντάει στα ερωτήματα που αναγράφονται στην σκηνή τέλους του παιχνιδιού.

## Κεφάλαιο 12ο: Το φαινόμενο του θερμοκηπίου υπερ-θερμαίνει

### 12.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την μέτρηση της θερμοκρασίας σε ένα αυτοσχέδιο θερμοκήπιο. Στο συγκεκριμένο πείραμα ο μαθητής χρησιμοποιεί ένα θερμόμετρο για την μέτρηση της θερμοκρασίας που παράγεται από το φως. Ως πηγές φωτός χρησιμοποιούνται δύο λάμπες γραφείου.

### 12.2 Κατασκευή του πειράματος

Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια και από πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει μία σφαιρική γυάλινη φιάλη, ένα κομμάτι φελιζόλ, ένα σάκο με χώμα, ένα θερμόμετρο οινόπνευματος και δύο φωτιστικά. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 12.1).



Εικόνα 12.1 Η διδιάστατη σκηνή της πρώτης φάσης του πειράματος του 9<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού. Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο ένα σάκο χόμα, ένα δοχείο πυρέξ, ένα θερμόμετρο, δύο φωτιστικά και ένα κομμάτι φελιζόλ. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή (Εικόνα 12.2).



Εικόνα 12.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 9<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 3) παρουσιάζονται στον χρήστη, τα βήματα που πρέπει να ακολουθήσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να παρατηρήσει το φαινόμενο του θερμοκηπίου (Εικόνα 12.3).



Εικόνα 12.3 Η σκηνή της τρίτης φάσης του πειράματος του 9<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά ο χρήστης πρέπει να σηκώσει από το τραπέζι τον σάκο με το χώμα και στη συνέχεια να τοποθετήσει λίγο από αυτό το χώμα μέσα στο δοχείο που βρίσκεται μπροστά του. Στη συνέχεια, ο μαθητής πρέπει να επιλέξει το φελιζόλ, πατώντας το κουμπί που θα εμφανιστεί στην οθόνη του να κόψει ένα κομμάτι από αυτό και τέλος να το τοποθετήσει στην θέση που θα του υποδεικνύει η εφαρμογή με την χρήση ενός εικονικού ολογράμματος. Ο χρήστης, πατώντας πάνω στο εικονικό ολόγραμμα τοποθετεί το αντικείμενο που «κρατάει» στη θέση εκείνη. Έπειτα, πρέπει να πάρει το θερμόμετρο και να το τοποθετήσει και αυτό πάνω στο δοχείο. Σε αυτό το σημείο ο μαθητής πρέπει να παρατηρήσει και να καταγράψει την θερμοκρασία. Μετά από λίγο δίνεται στον μαθητή η δυνατότητα να ανάψει το φως των φωτιστικών πατώντας επάνω τους. Ο μαθητής ανάβει τα φώτα και παρατηρεί τι συμβαίνει στην θερμοκρασία. Όταν η θερμοκρασία σταθεροποιηθεί ο μαθητής μπορεί να πατήσει το κουμπί τέλους για να πάει στην επόμενη σκηνή (Εικόνα 12.4).



Εικόνα 12.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 9<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα `Kef9pirama1script.cs`. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές εννέα μεταβλητών τύπου `boolean` που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές εννέα μεταβλητών τύπου `string` που θα χρησιμοποιηθούν ως tags πάνω στα αντικείμενα της σκηνής και δεκαέξι μεταβλητών τύπου `GameObject` που θα κρατάνε τα δεδομένα των δεκαέξι αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου `TextMeshProUGUI` που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 12.5).

```

9 public class Kef9pirama1script : MonoBehaviour
10 {
11
12     //selection----
13     private Transform selection;
14
15     public bool xomaislifted = false;
16     public bool readytoclickfelizol = false;
17     public bool felizolcutislifted = false;
18     public bool thermometroeadytolift = false;
19     public bool thermometroislifted = false;
20     public bool readytoswitchon = false;
21     public bool finalphase = false;
22
23     public bool lighton = false;
24     public bool lighton2 = false;
25
26
27     // [SerializeField] private string selectableTag = "Selectable";
28     [SerializeField] private string ThermometroTag = "Thermometro";
29     [SerializeField] private string felizolTag = "felizolTag";
30     [SerializeField] private string felizolcutTag = "felizolcutTag";
31     [SerializeField] private string FotistikoKef8Tag = "FotistikoKef8Tag";
32     [SerializeField] private string Fotistiko2Kef8Tag = "Fotistiko2Kef8Tag";
33     [SerializeField] private string xomaTag = "xomaTag";
34     [SerializeField] private string doxeiostroggiloTag = "doxeiostroggiloTag";
35     [SerializeField] private string doxeiostroggiloXwmaTag = "doxeiostroggiloXwmaTag";
36     [SerializeField] private string doxeiostroggiloXwmafelizolTag = "doxeiostroggiloXwmafelizolTag";
37
38
39
40
41     public GameObject Thermometro;
42     public GameObject ThermometroFinal;
43     public GameObject Fotistiko;
44     public GameObject SpotLightFotistiko;
45     public GameObject Fotistiko2;
46     public GameObject SpotLightFotistiko2;
47     public GameObject Xoma;

```

Εικόνα 12.5 Αρχικοποίηση μεταβλητών του script Kef9pirama1script.

Έπειτα, στην μέθοδο Awake() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικειμένου (infotext) της σκηνής και απενεργοποίηση των αντικειμένων TelosButton και CutButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο TelosButton το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 12.6).

```

62 private void Awake()
63 {
64     Application.targetFrameRate = 30;
65     Debug.Log("Into the Script!!!");
66
67     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
68     infotext.text = "Πάρε το χόμα.";
69
70     TelosButton.SetActive(false);
71     CutButton.SetActive(false);
72 }

```

Εικόνα 12.6 Η μέθοδος Awake() του script Kef9pirama1script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε ο χρήστης «κρατάει» αυτό το αντικείμενο και η εφαρμογή περιμένει από αυτόν να το χρησιμοποιήσει ή να τοποθετήσει κάπου αλλού. Αρχικά, ο χρήστης επιλέγει και «κρατάει» το σακί με το χόμα που βρίσκεται επάνω στο τραπέζι και έπειτα τοποθετεί λίγο από αυτό το χόμα μέσα στο δοχείο. Στη συνέχεια ο χρήστης πρέπει να επιλέξει το φελιζόλ και αφού το κάνει αυτό εμφανίζεται στην οθόνη του το κουμπί «CUT». Το κουμπί αυτό όταν πατηθεί καλεί την μέθοδο που του έχει οριστεί στο Onclick() state, δηλαδή την μέθοδο cutfelizol(). Η μέθοδος αυτή όταν καλείτε ενημερώνει κατάλληλα το κείμενο του text και ενεργοποιεί/απενεργοποιεί τα αντικείμενα που πρέπει (Εικόνα 12.7).

```

197 public void cutfelizol()
198 {
199     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
200     infotext.text = "Πάρε το κομμάτι φελιζόλ.";
201
202     Felizol.SetActive(false);
203     felizolCUT.SetActive(true);
204     CutButton.SetActive(false);
205 }

```

Εικόνα 12.7 Η μέθοδος cutfelizol() του script Kef9pirama1script.

Αφού ο χρήστης πατήσει το κουμπί «CUT», στη συνέχεια θα εμφανιστεί μπροστά του ένα μικρό κομμάτι φελιζόλ, το οποίο πρέπει να επιλέξει και έπειτα να τοποθετήσει στην θέση του εικονικού ολογράμματος (εικονικό ολόγραμμα κομματιού φελιζόλ) που έχει εμφανιστεί στη σκηνή. Το εικονικό ολόγραμμα είναι ένα αντικείμενο που ενεργοποιείται προσωρινά στη σκηνή για να δείξει στον χρήστη το που πρέπει να αφήσει το αντικείμενο που κρατάει την δεδομένη χρονική στιγμή. Στη συνέχεια, ο μαθητής πρέπει να σηκώσει το θερμόμετρο από το τραπέζι και να το τοποθετήσει μέσα στο δοχείο, επάνω στο κομμάτι φελιζόλ. Όταν ο χρήστης τοποθετήσει το θερμόμετρο καλείται μία setActive() για την ενεργοποίηση του image με μορφή θερμόμετρου στην οθόνη του χρήστη (Εικόνα 12.8). Στη συνέχεια, ο χρήστης περιμένει μερικά δευτερόλεπτα προκειμένου το script να καλέσει την μέθοδο switchfotistika() χρησιμοποιώντας την Invoke (Εικόνα 12.9). Η Invoke μπορεί να καλέσει μία μέθοδο μετά από μία συγκεκριμένη ώρα που θα ορίσει ο χρήστης.

```

157     if (selection.CompareTag(doxeiostraggiloxwmafelizolTag) && thermometroislifted == true)
158     {
159         ThermometroFinal.SetActive(true);
160
161         thermometerimagezoomed.SetActive(true);
162         thermometerindicatorimagezoomed.SetActive(true);
163
164         infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
165         infotext.text = "Παρατήρησε και σημείωσε την θερμοκρασία.";
166
167         Invoke("switchfotistika", 8.0f);
168
169     }

```

Εικόνα 12.8 Κάλεσμα της μεθόδου switchfotistika().

```

208     public void switchfotistika()
209     {
210         infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
211         infotext.text = "Αναψε τα φωτιστικά.";
212
213         readytoswitchon = true;
214     }

```

Εικόνα 12.9 Η μέθοδος switchfotistika() του script Kef9pirama1script.

Ο χρήστης την ώρα που περιμένει πρέπει να παρατηρήσει και να καταγράψει την θερμοκρασία που δείχνει το θερμόμετρο. Στη συνέχεια ο χρήστης πρέπει να ανάψει τα φώτα πατώντας επάνω στα φωτιστικά. Όταν ανάψει τα φώτα, ο μαθητής πρέπει να παρατηρεί συνεχώς τι συμβαίνει με την θερμοκρασία ως ότου αυτή σταθεροποιηθεί. Μετά από λίγα δευτερόλεπτα, αφού η θερμοκρασία έχει σταθεροποιηθεί, καλείτε η μέθοδος endexperiment() που εμφανίζει το κουμπί τέλους και ενημερώνει το κείμενο του text αντικειμένου (Εικόνα 12.10). Όλες αυτές οι διαδικασίες επιτυγχάνονται μέσα στο script με την συνεχή ενεργοποίηση και απενεργοποίηση των αντικειμένων της σκηνής ανάλογα με τις κινήσεις του χρήστη. Η ενεργοποίηση και η απενεργοποίηση των αντικειμένων αυτών γίνεται με την μέθοδο setActive(false/true) με τιμή true ή false. Εφόσον έχει εμφανιστεί το κουμπί τέλους στη σκηνή, δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί αυτό και να μεταφερθεί στην επόμενη σκηνή αφού φυσικά έχει πρώτα παρατηρήσει τι έχει συμβεί με την τελική θερμοκρασία. Το κουμπί τέλους δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω (Εικόνα 12.11).

```

181     if (SpotLightFotistiko.activeSelf && SpotLightFotistiko2.activeSelf && finalphase == false)
182     {
183
184         infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
185         infotext.text = "Παρατήρησε την θερμοκρασία με τα φώτα ανοιχτά.";
186
187         thermometerindicatorimagezoomed.GetComponent<Animator>().Play("20to40slowanim");
188
189         Invoke("endexperiment", 14.0f);
190         finalphase = true;
191
192     }

```

Εικόνα 12.10 Κάλεσμα της μεθόδου endexperiment().

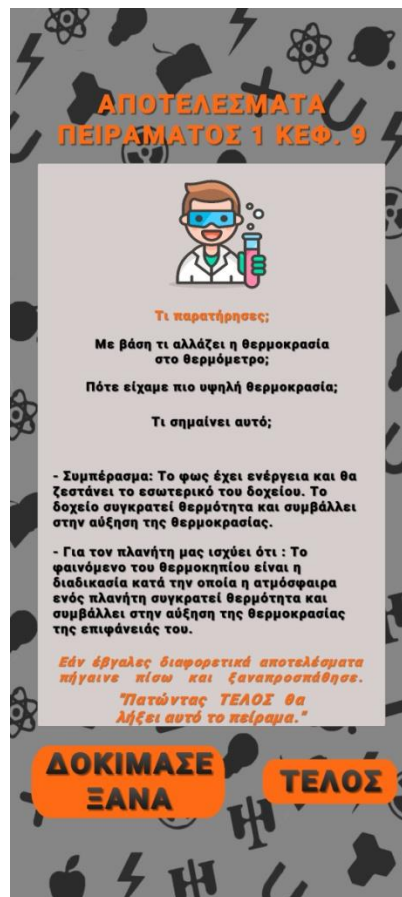
```

217     public void endexperiment()
218     {
219         TelosButton.SetActive(true);
220         infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
221         infotext.text = "Η θερμοκρασία σταθεροποιήθηκε!";
222     }

```

Εικόνα 12.11 Η μέθοδος που εμφανίζει το κουμπί τέλους.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή θέτει κάποιες ερωτήσεις στον χρήστη σχετικά με το πείραμα που μόλις υλοποίησε και τα φαινόμενα που παρατήρησε. Για την απάντηση των ερωτήσεων αυτών ο μαθητής μπορεί να ζητήσει βοήθεια από κάποιο άλλο άτομο. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα. Εάν ο χρήστης επιθυμεί να μελετήσει ξανά το πείραμα, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ» (Εικόνα 12.12).



Εικόνα 12.12 Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 9<sup>ου</sup> Κεφαλαίου [17].

### 12.3 Επίλογος

Το πείραμα του 9ου κεφαλαίου «Το φαινόμενο του θερμοκηπίου υπερ-θερμαίνει» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης δημιουργεί ένα αυτοσχέδιο θερμοκήπιο στην σκηνή επαυξημένης πραγματικότητας, τοποθετεί ένα θερμόμετρο μέσα σε αυτό, ανάβει το φως και παρατηρεί τι συμβαίνει στην θερμοκρασία. Η μέτρηση της θερμοκρασίας γίνεται με την χρήση του θερμόμετρου. Ο χρήστης καταγράφει τις τιμές της θερμοκρασίας ενώ παρατηρεί τι συμβαίνει σε αυτήν και στην συνέχεια απαντάει στα ερωτήματα που αναγράφονται στην σκηνή τέλους του παιχνιδιού.



## Κεφάλαιο 13ο: Το ηλεκτρικό βραχυ-κύκλωμα – Κίνδυνοι και «ασφάλεια»

### 13.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνονται τα πείραμα για την κατανόηση των εννοιών του κυκλώματος και του βραχυκυκλώματος. Στα πειράματα αυτά ο μαθητής δημιουργεί ένα κύκλωμα με την χρήση καλωδίων, συνδετήρων και μίας λάμπας πυρακτώσεως. Ως πηγή ενέργειας χρησιμοποιείται μία μπαταρία.

### 13.2 Κατασκευή του πειράματος

Οι σκηνές μέχρι την ουσιαστική υλοποίηση των πειραμάτων είναι κοινές και για τα δύο πειράματα.

Στη σκηνή αυτή (Φάση 1) (χρησιμοποιείται η ίδια σκηνή για τα πειράματα 1 και 2) η εφαρμογή εξηγεί στον χρήστη ποια και από πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει μία μπαταρία, μία βάση για λάμπα, μία λάμπα, ένα καλώδιο κουλούρα και ένα ζευγάρι μεταλλικούς συνδετήρες. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 13.1).



Εικόνα 13.1 Η δισδιάστατη σκηνή της πρώτης φάσης του 1<sup>ου</sup> και 2<sup>ου</sup> πειράματος του 10<sup>ου</sup> Κεφαλαίου.

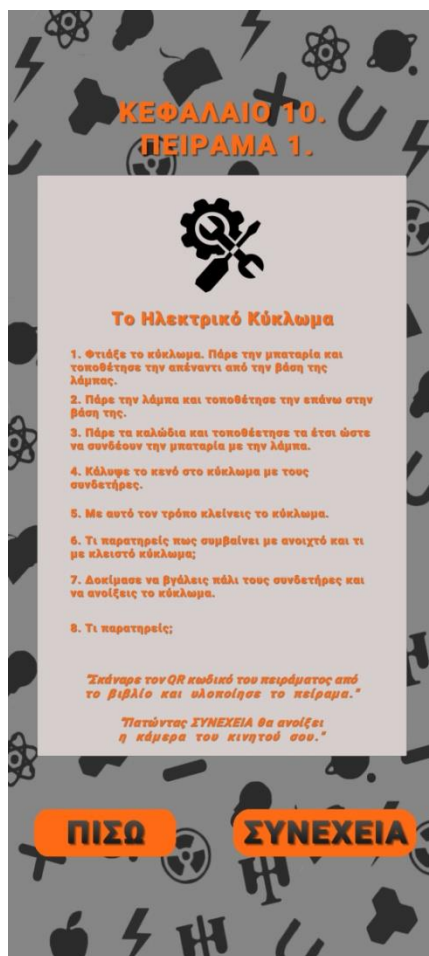
Στην επόμενη σκηνή (Φάση 2) (χρησιμοποιείται η ίδια σκηνή για τα πειράματα 1 και 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται – επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού. Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο μία βάση για λάμπα, μία λάμπα, ένας συνδετήρας, μία μπαταρία και μία κουλούρα καλώδιο. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή (Εικόνα 13.2).



Εικόνα 13.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του 1<sup>ου</sup> και 2<sup>ου</sup> πειράματος του 10<sup>ου</sup> Κεφαλαίου.

### 13.2.1 Το ηλεκτρικό κύκλωμα

Στην επόμενη σκηνή (Φάση 3) παρουσιάζονται στον χρήστη, τα βήματα που πρέπει να ακολουθήσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να δημιουργήσει ένα ηλεκτρικό κύκλωμα (Εικόνα 13.3).



Εικόνα 13.3 Η σκηνή της τρίτης φάσης του 1<sup>ου</sup> πειράματος του 10<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά ο χρήστης πρέπει να σηκώσει από το τραπέζι την μπαταρία και να την τοποθετήσει στην θέση που θα του υποδεικνύει η εφαρμογή με την χρήση ενός εικονικού ολογράμματος. Ο χρήστης, πατώντας πάνω στο εικονικό ολόγραμμα τοποθετεί το αντικείμενο που «κρατάει» στη θέση εκείνη. Στη συνέχεια, ο μαθητής πρέπει να επιλέξει την λάμπα και έπειτα επιλέγοντας την βάση της να την τοποθετήσει επάνω σε αυτή. Με παρόμοιο τρόπο, πρέπει να τοποθετήσει τα καλώδια στην σωστή θέση, έτσι ώστε να ενώνουν την μπαταρία με την βάση της λάμπας (Εικόνα 13.4). Έπειτα, ο μαθητής πρέπει να πάρει τους συνδετήρες και να τους τοποθετήσει στην θέση που του υποδεικνύει η εφαρμογή με την χρήση ενός νέου ολογράμματος. Όταν τοποθετήσει τους συνδετήρες στην σωστή θέση, το φως της λάμπας θα ανάψει και ο μαθητής θα έχει φτιάξει ένα κύκλωμα (Εικόνα 13.5). Σε αυτή την φάση ο μαθητής πρέπει να παρατηρήσει τι συμβαίνει και να προσπαθήσει να βγάλει τους συνδετήρες και να τους ξαναβάλει στην θέση τους.



Εικόνα 13.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του 1<sup>ου</sup> πειράματος του 10<sup>ου</sup> Κεφαλαίου, με κλειστό φως.



Εικόνα 13.5 Η τρισδιάστατη σκηνή της τέταρτης φάσης του 1<sup>ου</sup> πειράματος του 10<sup>ου</sup> Κεφαλαίου με ανοιχτό φως.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα `Kef10Pirama1script.cs`. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές έντεκα μεταβλητών τύπου `boolean` που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές οκτώ μεταβλητών τύπου `string` που θα χρησιμοποιηθούν ως `tags` πάνω στα αντικείμενα της σκηνής και δεκαεπτά μεταβλητών τύπου `GameObject` που θα κρατάνε τα δεδομένα των δεκαεπτά αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου `TextMeshProUGUI` που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 13.6).

```

9 public class Kef10pirama1script : MonoBehaviour
10 {
11
12
13     //selection----
14     private Transform selection;
15
16     public bool mpatariaislifted = false;
17     public bool bulbreadytolift = false;
18     public bool bulbislifted = false;
19     public bool kalodioreadytotake = false;
20     public bool kalodioislifted = false;
21     public bool sindetirasreadytotake = false;
22     public bool sindetirasislifted = false;
23     public bool kiklomaclosed = false;
24
25     public bool lastphase = false;
26
27     public bool lighton = false;
28     public bool lighton2 = false;
29
30
31     // [SerializeField] private string selectableTag = "Selectable";
32     [SerializeField] private string mpatariaTag = "mpatariaTag";
33     [SerializeField] private string kalodiokoulouraTag = "kalodiokoulouraTag";
34     [SerializeField] private string bulbTag = "bulbTag";
35     [SerializeField] private string bulbbaseTag = "bulbbaseTag";
36     [SerializeField] private string mpatariahologramTag = "mpatariahologramTag";
37     [SerializeField] private string sindetirasTag = "sindetirasTag";
38     [SerializeField] private string sindetirashologramTag = "sindetirashologramTag";
39     [SerializeField] private string sindetirasFinalTag = "sindetirasFinalTag";

```

Εικόνα 13.6 Αρχικοποίηση μεταβλητών του script Kef10pirama1script.

Έπειτα, στην μέθοδο Awake() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικείμενου (infotext) της σκηνής και απενεργοποίηση των αντικείμενων TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο TelosButton το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 13.7).

```

64 private void Awake()
65 {
66     Application.targetFrameRate = 30;
67     Debug.Log("Into the Script!!!");
68
69     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
70     infotext.text = "Πάρε την μαπαρία.";
71
72     TelosButton.SetActive(false);
73 }

```

Εικόνα 13.7 Η μέθοδος Awake() του script Kef10pirama1script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το

αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε ο χρήστης «κρατάει» αυτό το αντικείμενο και η εφαρμογή περιμένει από αυτόν να το χρησιμοποιήσει ή να τοποθετήσει κάπου αλλού. Αρχικά, ο χρήστης επιλέγει και «κρατάει» την μπαταρία που βρίσκεται επάνω στο τραπέζι και έπειτα την τοποθετεί στην θέση του εικονικού ολογράμματος (εικονικό ολόγραμμα μπαταρίας) που έχει εμφανιστεί στη σκηνή. Το εικονικό ολόγραμμα είναι ένα αντικείμενο που ενεργοποιείται προσωρινά στη σκηνή για να δείξει στον χρήστη το που πρέπει να αφήσει το αντικείμενο που κρατάει την δεδομένη χρονική στιγμή. Στη συνέχεια ο χρήστης πρέπει να επιλέξει την λάμπα και έπειτα να επιλέξει την βάση της λάμπας προκειμένου να την τοποθετήσει εκεί. Παρομοίως και για το καλώδιο, ο μαθητής το επιλέγει και έπειτα επιλέγοντας την βάση της λάμπας το τοποθετεί στην θέση που πρέπει. Στον χρήστη δίνεται τώρα η δυνατότητα να επιλέξει και να σηκώσει από το τραπέζι τους συνδετήρες και να τους τοποθετήσει στην θέση του καινούργιου εικονικού ολογράμματος που εμφανίστηκε στη σκηνή (Εικόνα 13.8).

```

83 void Update()
84 {
85     if (Input.GetMouseButtonDown(0))
86     {
87         // Debug.Log("Mouse Button Down!!!");
88         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
89         RaycastHit hit;
90
91         if (Physics.Raycast(ray, out hit))
92         {
93             // Debug.Log("Hit Something!!!");
94             var selection = hit.transform;
95             if (selection.CompareTag(mpatariaTag))
96             {
97                 // Kremastra.transform.localPosition = new Vector3(-0.0057f, 1.8067f, 0.076f);
98                 // Kremastra.transform.localRotation = Quaternion.Euler(0, 90, 90f);
99
100                Mpataria.SetActive(false);
101                mpatariahologram.SetActive(true);
102
103                infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
104                infotext.text = "Βάλε την μπαταρία απέναντι από την βάση της λάμπας.";
105
106                mpatariaislifted = true;
107            }

```

Εικόνα 13.8 Η μέθοδος Update() του script Kefl0pirama1script.

Όταν ο χρήστης τοποθετήσει τους συνδετήρες στην θέση που πρέπει τότε το script καλεί την μέθοδο endexperiment() με την χρήση της Invoke και το φως της λάμπας ανάβει. Έχουμε δηλαδή την ενεργοποίηση ενός point light αντικειμένου. Η Invoke μπορεί να καλέσει μία μέθοδο μετά από μία συγκεκριμένη ώρα που θα ορίσει ο χρήστης. Η μέθοδος endexperiment() ενεργοποιεί και εμφανίζει στην οθόνη του χρήστη το κουμπί τέλους. Όσο ο χρήστης περιμένει την Invoke() να καλέσει την μέθοδο, αυτός μπορεί να πειραματιστεί με τους συνδετήρες και να παρατηρήσει τι συμβαίνει (Εικόνα 13.9).

```

172     if (selection.CompareTag(sindetirashologramTag) && lastphase == true)
173     {
174         sindetiras1Final.SetActive(true);
175         sindetiras2Final.SetActive(true);
176         sindetirashologram1.SetActive(false);
177         sindetirashologram2.SetActive(false);
178
179         kiclomaclosed = !kiclomaclosed;
180
181         BulbLight.SetActive(true);
182         BulbParticle.SetActive(true);
183
184         infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
185         infotext.text = "Τι παρατηρείς; Βγάλε τους συνδετήρες και ξαναπροσπάθησε.";
186
187         Invoke("endexperiment", 15.0f);
188     }
189

```

Εικόνα 13.9 Κάλεσμα μεθόδου endexperiment().

Όλες αυτές οι διαδικασίες επιτυγχάνονται μέσα στο script με την συνεχή ενεργοποίηση και απενεργοποίηση των αντικειμένων της σκηνής ανάλογα με τις κινήσεις του χρήστη. Η ενεργοποίηση και η απενεργοποίηση των αντικειμένων αυτών γίνεται με την μέθοδο setActive(false/true) με τιμή true ή false. Εφόσον έχει εμφανιστεί το κουμπί τέλους στη σκηνή, δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί αυτό και να μεταφερθεί στην επόμενη σκηνή αφού φυσικά έχει πρώτα παρατηρήσει τι έχει συμβεί με την τελική θερμοκρασία. Το κουμπί τέλους δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω (Εικόνα 13.10).

```

207     void endexperiment()
208     {
209         TelosButton.SetActive(true);
210     }

```

Εικόνα 13.10 Η μέθοδος που εμφανίζει το κουμπί τέλους.

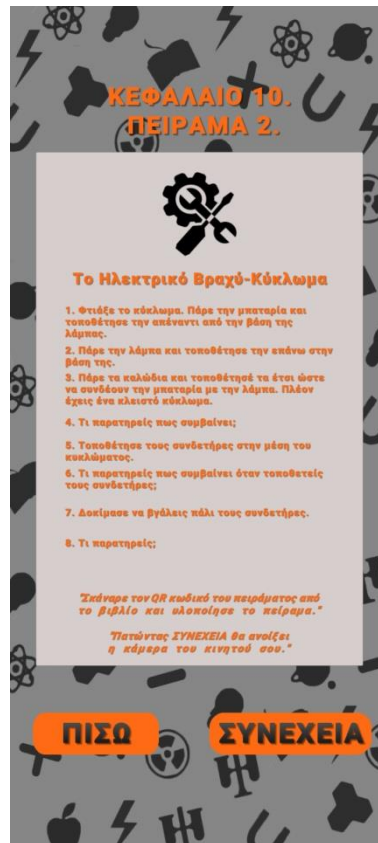
Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή θέτει κάποιες ερωτήσεις στον χρήστη σχετικά με το πείραμα που μόλις υλοποίησε και τα φαινόμενα που παρατήρησε. Για την απάντηση των ερωτήσεων αυτών ο μαθητής μπορεί να ζητήσει βοήθεια από κάποιο άλλο άτομο. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα. Εάν ο χρήστης επιθυμεί να μελετήσει ξανά το πείραμα, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ» (Εικόνα 13.11).



Εικόνα 13.11 Η δισδιάστατη σκηνή της πέμπτης φάσης του 1<sup>ου</sup> πειράματος του 10<sup>ου</sup> Κεφαλαίου.

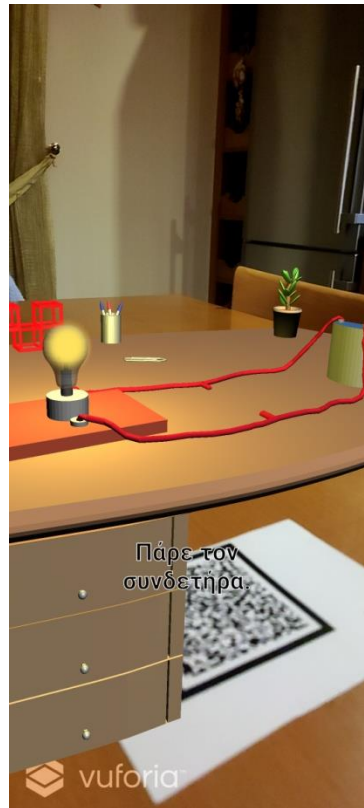
### 13.2.2 Το ηλεκτρικό βραχυ-κύκλωμα

Στην επόμενη σκηνή (Φάση 3) παρουσιάζει στον χρήστη, τα βήματα που πρέπει να ακολουθήσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να δημιουργήσει ένα ηλεκτρικό βραχυ-κύκλωμα (Εικόνα 13.12).



Εικόνα 13.12 Η σκηνή της τρίτης φάσης του 2<sup>ου</sup> πειράματος του 10<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά ο χρήστης πρέπει να σηκώσει από το τραπέζι την μπαταρία και να την τοποθετήσει στην θέση που θα του υποδεικνύει η εφαρμογή με την χρήση ενός εικονικού ολογράμματος. Ο χρήστης, πατώντας πάνω στο εικονικό ολόγραμμα τοποθετεί το αντικείμενο που «κρατάει» στη θέση εκείνη. Στη συνέχεια, ο μαθητής πρέπει να επιλέξει την λάμπα και έπειτα επιλέγοντας την βάση της να την τοποθετήσει επάνω σε αυτή. Με παρόμοιο τρόπο, πρέπει να τοποθετήσει τα καλώδια στην σωστή θέση, έτσι ώστε να ενώνουν την μπαταρία με την βάση της λάμπας. Στο σημείο αυτό ο χρήστης μπορεί να παρατηρήσει ότι η λάμπα είναι αναμμένη (άρα το κύκλωμα είναι κλειστό). Έπειτα, ο μαθητής πρέπει να πάρει τους συνδετήρες και να τους τοποθετήσει στην θέση που του υποδεικνύει η εφαρμογή με την χρήση ενός νέου εικονικού ολογράμματος. Όταν τοποθετήσει τους συνδετήρες στην σωστή θέση, το φως της λάμπας θα σβήσει και ο μαθητής θα έχει δημιουργήσει ένα βραχυ-κύκλωμα. Σε αυτή την φάση ο μαθητής πρέπει να παρατηρήσει τι συμβαίνει και να προσπαθήσει να βγάλει τους συνδετήρες και να τους ξαναβάλει στην θέση τους (Εικόνα 13.13).



Εικόνα 13.13 Η τρισδιάστατη σκηνή της τέταρτης φάσης του 2<sup>ου</sup> πειράματος του 10<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα Kef10pirama2script.cs. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές έντεκα μεταβλητών τύπου boolean που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές οκτώ μεταβλητών τύπου string που θα χρησιμοποιηθούν ως tags πάνω στα αντικείμενα της σκηνής και δεκαεπτά μεταβλητών τύπου GameObject που θα κρατάνε τα δεδομένα των δεκαεπτά αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου TextMeshProUGUI που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 13.14).

```

10 public class Kef10pirama2script : MonoBehaviour
11 {
12     //selection----
13     private Transform selection;
14
15     public bool mpatariaislifted = false;
16     public bool bulbreadytolift = false;
17     public bool bulbislifted = false;
18     public bool kalodioreadytotake = false;
19     public bool kalodioislifted = false;
20     public bool sindetirasreadytotake = false;
21     public bool sindetirasislifted = false;
22     public bool kinklomaclosed = false;
23
24     public bool lastphase = false;
25
26     public bool lighton = false;
27     public bool lighton2 = false;
28
29
30     // [SerializeField] private string selectableTag = "Selectable";
31     [SerializeField] private string mpatariaTag = "mpatariaTag";
32     [SerializeField] private string kalodiokoulouraTag = "kalodiokoulouraTag";
33     [SerializeField] private string bulbTag = "bulbTag";
34     [SerializeField] private string bulbbaseTag = "bulbbaseTag";
35     [SerializeField] private string mpatariahologramTag = "mpatariahologramTag";
36     [SerializeField] private string sindetirasTag = "sindetirasTag";
37     [SerializeField] private string sindetirashologramTag = "sindetirashologramTag";
38     [SerializeField] private string sindetirasFinalTag = "sindetirasFinalTag";
39
40
41
42     public GameObject sindetirasPre;
43     public GameObject sindetirashologram1;
44     public GameObject sindetiras1Final;
45     public GameObject sindetirashologram2;
46     public GameObject sindetiras2Final;
47     public GameObject Mpataria;

```

Εικόνα 13.14 Αρχικοποίηση μεταβλητών του script Kef10pirama2script.

Έπειτα, στην μέθοδο Awake() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικειμένου (infotext) της σκηνής και απενεργοποίηση των αντικειμένων TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο TelosButton το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 13.15).

```

63 private void Awake()
64 {
65     Application.targetFrameRate = 30;
66     Debug.Log("Into the Script!!!");
67
68     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
69     infotext.text = "Πάρε την μπαταρία.";
70
71     TelosButton.SetActive(false);
72 }

```

Εικόνα 13.15 Η μέθοδος Awake() του script Kef10pirama2script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας

μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε ο χρήστης «κρατάει» αυτό το αντικείμενο και η εφαρμογή περιμένει από αυτόν να το χρησιμοποιήσει ή να τοποθετήσει κάπου αλλού. Αρχικά, ο χρήστης επιλέγει και «κρατάει» την μπαταρία που βρίσκεται επάνω στο τραπέζι και έπειτα την τοποθετεί στην θέση του εικονικού ολογράμματος (εικονικό ολόγραμμα μπαταρίας) που έχει εμφανιστεί στη σκηνή. Το εικονικό ολόγραμμα είναι ένα αντικείμενο που ενεργοποιείται προσωρινά στη σκηνή για να δείξει στον χρήστη το που πρέπει να αφήσει το αντικείμενο που κρατάει την δεδομένη χρονική στιγμή. Στη συνέχεια ο χρήστης πρέπει να επιλέξει την λάμπα και έπειτα να επιλέξει την βάση της λάμπας προκειμένου να την τοποθετήσει εκεί. Παρομοίως και για το καλώδιο, ο μαθητής το επιλέγει και έπειτα επιλέγοντας την βάση της λάμπας το τοποθετεί στην θέση που πρέπει. Όταν τα καλώδια τοποθετηθούν τότε το φως ανάβει, δηλαδή έχουμε την ενεργοποίηση ενός point light αντικειμένου. Ο χρήστης παρατηρεί τι συμβαίνει. Στον χρήστη δίνεται τώρα η δυνατότητα να επιλέξει και να σηκώσει από το τραπέζι τους συνδετήρες και να τους τοποθετήσει στην θέση του καινούργιου εικονικού ολογράμματος που εμφανίστηκε στη σκηνή (Εικόνα 13.16).

```

82 void Update()
83 {
84     if (Input.GetMouseButtonDown(0))
85     {
86         // Debug.Log("Mouse Button Down!!!");
87         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
88         RaycastHit hit;
89
90         if (Physics.Raycast(ray, out hit))
91         {
92             // Debug.Log("Hit Something!!!");
93             var selection = hit.transform;
94             if (selection.CompareTag(mpatariaTag))
95             {
96                 // Kremastra.transform.localPosition = new Vector3(-0.0057f, 1.8067f, 0.076f);
97                 // Kremastra.transform.localRotation = Quaternion.Euler(0, 90, 90f);
98
99                 Mpataria.SetActive(false);
100                mpatariahologram.SetActive(true);
101
102                infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
103                infotext.text = "Βάλε την μπαταρία απέναντι από την βάση της λάμπας.";
104
105                mpatariaislifted = true;
106            }

```

Εικόνα 13.16 Η μέθοδος Update() του script Kef10pirama2script.

Όταν ο χρήστης τοποθετήσει τους συνδετήρες στην θέση που πρέπει τότε το φως σβήνει και το script καλεί την μέθοδο endexperiment() με την χρήση της Invoke. Σε αυτή τη φάση ο χρήστης έχει δημιουργήσει επιτυχώς το βραχυκύκλωμα. Η Invoke μπορεί να καλέσει μία μέθοδο μετά από μία συγκεκριμένη ώρα που θα ορίσει ο χρήστης. Η μέθοδος endexperiment() ενεργοποιεί και εμφανίζει στην οθόνη του χρήστη το κουμπί τέλους. Όσο ο χρήστης περιμένει την Invoke() να καλέσει την μέθοδο, αυτός μπορεί να πειραματιστεί με τους συνδετήρες και να παρατηρήσει τι συμβαίνει (Εικόνα 13.17).

```

175     if (selection.CompareTag(sindetirashologramTag) && lastphase == true)
176     {
177         sindetirashologram1.SetActive(false);
178         sindetirashologram2.SetActive(false);
179         sindetiras1Final.SetActive(true);
180         sindetiras2Final.SetActive(true);
181
182         kıklomaclosed = !kıklomaclosed;
183
184         BulbLight.SetActive(false);
185         BulbParticle.SetActive(false);
186
187         infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
188         infotext.text = "Τι παρατηρείς; Βγάλε τους συνδετήρες και ξαναπροσάθησε.";
189
190         Invoke("endexperiment", 15.0f);
191     }
192

```

Εικόνα 13.17 Κάλεσμα μεθόδου endexperiment().

Όλες αυτές οι διαδικασίες επιτυγχάνονται μέσα στο script με την συνεχή ενεργοποίηση και απενεργοποίηση των αντικειμένων της σκηνής ανάλογα με τις κινήσεις του χρήστη. Η ενεργοποίηση και η απενεργοποίηση των αντικειμένων αυτών γίνεται με την μέθοδο setActive(false/true) με τιμή true ή false. Εφόσον έχει εμφανιστεί το κουμπί τέλους στη σκηνή, δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί αυτό και να μεταφερθεί στην επόμενη σκηνή αφού φυσικά έχει πρώτα παρατηρήσει τι έχει συμβεί με την τελική θερμοκρασία. Το κουμπί τέλους δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω (Εικόνα 13.18).

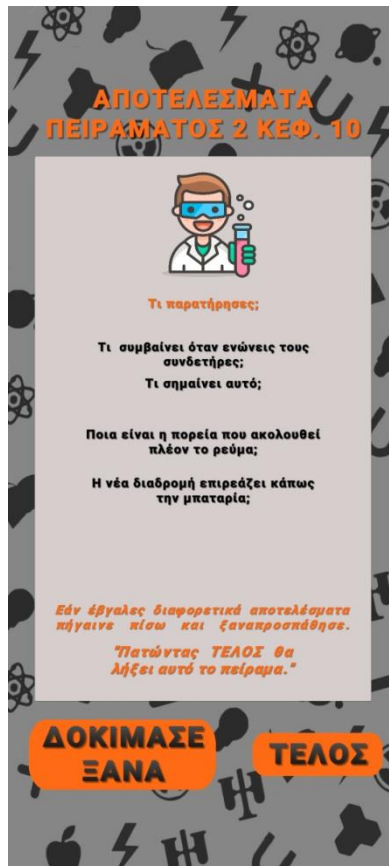
```

210     void endexperiment()
211     {
212         TelosButton.SetActive(true);
213     }
214

```

Εικόνα 13.18 Η μέθοδος που εμφανίζει το κουμπί τέλους.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή θέτει κάποιες ερωτήσεις στον χρήστη σχετικά με το πείραμα που μόλις υλοποίησε και τα φαινόμενα που παρατήρησε. Για την απάντηση των ερωτήσεων αυτών ο μαθητής μπορεί να ζητήσει βοήθεια από κάποιο άλλο άτομο. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα. Εάν ο χρήστης επιθυμεί να μελετήσει ξανά το πείραμα, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ» (Εικόνα 13.19).



Εικόνα 13.19 Η διδιάστατη σκηνή της πέμπτης φάσης του 2<sup>ου</sup> πειράματος του 10<sup>ου</sup> Κεφαλαίου.

### 13.3 Επίλογος

Τα πειράματα του 10ου κεφαλαίου «Το ηλεκτρικό βραχυ-κύκλωμα – Κίνδυνοι και ασφάλεια» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιούνται με την AR εφαρμογή που έχουμε κατασκευάσει. Στο πρώτο πείραμα ο χρήστης κατασκευάζει ένα ηλεκτρικό κύκλωμα με την χρήση μία λάμπας, μίας μπαταρίας και καλωδίων τα οποία συλλέγει από τον εικονικό χώρο. Με την χρήση δύο συνδετήρων μετατρέπει το κύκλωμα από ανοιχτό σε κλειστό. Στη συνέχεια, παρατηρεί τις αλλαγές που συμβαίνουν στην λάμπα και απαντάει τα ερωτήματα που αναγράφονται στην σκηνή τέλους του παιχνιδιού. Στο δεύτερο πείραμα συλλέγει τα ίδια αντικείμενα, αλλά τοποθετεί τους συνδετήρες με τέτοιο τρόπο ώστε το κύκλωμα να μετατραπεί από κλειστό σε ανοιχτό, δημιουργώντας έτσι ένα βραχυ-κύκλωμα. Τέλος, παρατηρεί τις αλλαγές που συμβαίνουν στην λάμπα και βγάζει τα ανάλογα συμπεράσματα.



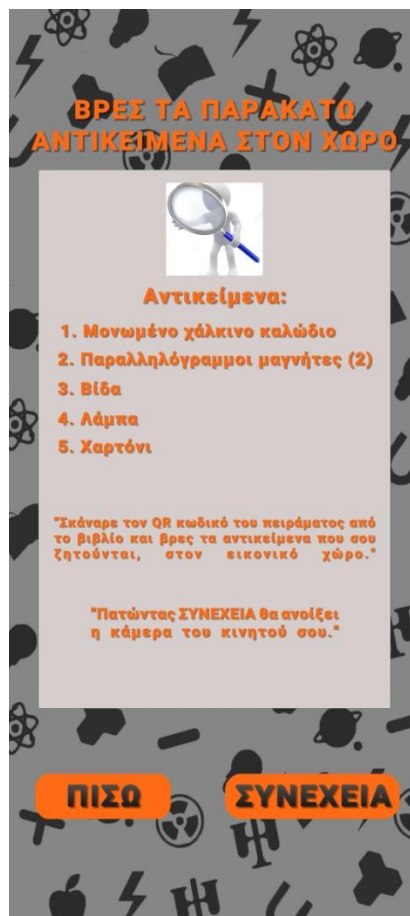
## Κεφάλαιο 14ο: Από το μαγνητισμό στον ηλεκτρισμό – Η ηλεκτρική (ιδιο-)γεννήτρια

### 14.1 Εισαγωγή

Στο κεφάλαιο αυτό γίνεται το πείραμα για την δημιουργία ενός αυτοσχέδιου ηλεκτρικού κινητήρα. Στο συγκεκριμένο πείραμα ο μαθητής με την χρήση του αυτοσχέδιου κινητήρα που δημιουργεί παράγει ηλεκτρισμό μέσω του μαγνητισμού.

### 14.2 Κατασκευή του πειράματος

Στη σκηνή αυτή (Φάση 1) η εφαρμογή εξηγεί στον χρήστη ποια και από πόσα αντικείμενα πρέπει να μαζέψει από την επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας. Επίσης, δίνει κάποιες συμβουλές στον χρήστη σε μορφή (text) σχολίων κάτω από την επεξήγηση. Στο συγκεκριμένο πείραμα η εφαρμογή ζητάει από τον χρήστη να μαζέψει ένα μονωμένο χάλκινο καλώδιο, δύο παραλληλόγραμμους μαγνήτες, μία βίδα, μία λάμπα και ένα μεγάλο χαρτόνι. Τα σχόλια από κάτω παροτρύνουν τον χρήστη αφού πατήσει το κουμπί «ΣΥΝΕΧΕΙΑ», να σκανάρει τον QR κωδικό από το βιβλίο. Προειδοποιεί επίσης τον χρήστη ότι πατώντας το κουμπί «ΣΥΝΕΧΕΙΑ» θα ανοίξει η κάμερα του κινητού του, με αποτέλεσμα να αλλάξει εντελώς η λειτουργία της εφαρμογής (Εικόνα 14.1).



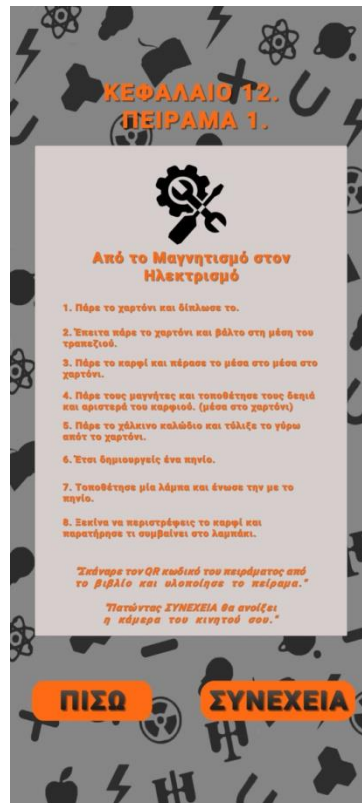
Εικόνα 14.1 Η δισδιάστατη σκηνή της πρώτης φάσης του πειράματος του 12<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 2) αφού ο χρήστης σκανάρει την εικόνα στόχο με την κάμερα του κινητού του, εμφανίζεται - επαυξάνεται μπροστά του μία πόρτα. Ο χρήστης κουνώντας κατάλληλα την συσκευή του μπορεί να μπει μέσα στην πόρτα και μέσα στον εικονικό χώρο του παιχνιδιού. Εκεί μέσα ο χρήστης πρέπει να βρει και να συλλέξει τα αντικείμενα που ζητούνται από αυτόν. Τα αντικείμενα που πρέπει να μαζέψει ο χρήστης φαίνονται και σε αυτή τη σκηνή με μορφή text επάνω στην οθόνη. Πιο συγκεκριμένα σε αυτή τη σκηνή ο χρήστης πρέπει να μαζέψει από τον εικονικό χώρο μία κουλούρα χάλκινο καλώδιο, μία λάμπα, ένα κομμάτι χαρτόνι, δύο μαγνήτες και ένα καρφί. Όταν ο χρήστης μαζεύει αντικείμενα στον εικονικό χώρο, τα text στην οθόνη του ενημερώνονται δυναμικά και αναγράφουν ακριβώς πόσα αντικείμενα έχουν συλλεχθεί. Όταν ο χρήστης μαζέψει όλα τα αντικείμενα που του ζητούνται, τότε η εφαρμογή περνάει αυτόματα στην επόμενη σκηνή (Εικόνα 14.2).



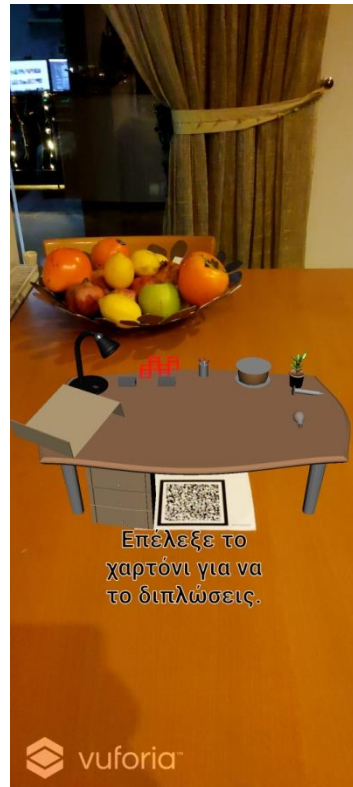
Εικόνα 14.2 Η τρισδιάστατη σκηνή της δεύτερης φάσης του πειράματος του 12<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή (Φάση 3) η εφαρμογή εξηγεί συνοπτικά στον χρήστη, τα βήματα που πρέπει να ακολουθήσει στην επόμενη τρισδιάστατη σκηνή επαυξημένης πραγματικότητας προκειμένου να δημιουργήσει ηλεκτρισμό μέσω του μαγνητισμού (Εικόνα 14.3).



Εικόνα 14.3 Η σκηνή της τρίτης φάσης του πειράματος του 12<sup>ου</sup> Κεφαλαίου.

Στην επόμενη σκηνή – πείραμα (Φάση 4) ο χρήστης, αφού σκανάρει την εικόνα στόχο και εμφανιστεί το επαυξημένο τραπέζι εργασίας, πρέπει να ακολουθήσει τις οδηγίες που του δίνει η εφαρμογή σε μορφή text. Το text αυτό έχει τοποθετηθεί στο κάτω μέρος της οθόνης και αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος. Αρχικά, ο χρήστης πρέπει να επιλέξει το χαρτόνι για να το τυλίξει και να το τοποθετήσει στην θέση που θα του υποδεικνύει η εφαρμογή με την χρήση ενός εικονικού ολογράμματος. Ο χρήστης, πατώντας πάνω στο εικονικό ολόγραμμα τοποθετεί το αντικείμενο που «κρατάει» στη θέση εκείνη. Έπειτα, πρέπει να πάρει το καρφί, τους δύο μαγνήτες και το χάλκινο καλώδιο και να τα τοποθετήσει στις θέσεις που πρέπει γύρω από το χαρτόνι και πάνω σε αυτό προκειμένου να δημιουργηθεί ένα πηνίο. Στη συνέχεια, ο μαθητής πρέπει να σηκώσει την λάμπα από το τραπέζι και να το τοποθετήσει στην νέα θέση που θα του υποδεικνύει η εφαρμογή με την χρήση ενός εικονικού ολογράμματος. Αφού ο χρήστης τοποθετήσει την λάμπα στην θέση που πρέπει, τότε εμφανίζεται στην οθόνη του ένα κουμπί. Πατώντας το κουμπί ο χρήστης περιστρέφει το καρφί. Σε αυτό το σημείο ο μαθητής πρέπει να παρατηρήσει τι συμβαίνει όταν περιστρέφει το καρφί και τι όταν δεν το περιστρέφει. Μετά από λίγο εμφανίζεται στη σκηνή το κουμπί τέλους και δίνεται στον μαθητή η δυνατότητα να το πατήσει για να πάει στην επόμενη σκηνή (Εικόνα 14.4).



Εικόνα 14.4 Η τρισδιάστατη σκηνή της τέταρτης φάσης του πειράματος του 12<sup>ου</sup> Κεφαλαίου.

Η υλοποίηση της σκηνής αυτής επιτυγχάνεται με την χρήση ενός ακόμη script γραμμένο σε C#. Το script που ελέγχει όλες τις λειτουργίες αυτής της σκηνής είναι το script με όνομα Kef12pirama1script.cs. Στην αρχή, μέσα στο script αρχικοποιούμε τις τιμές δώδεκα μεταβλητών τύπου boolean που θα χρησιμοποιηθούν για τον έλεγχο ροής του πειράματος. Θα χρησιμοποιηθούν δηλαδή για να φιλτράρουν με ποια αντικείμενα μπορεί να αλληλεπιδράσει ο χρήστης μία δεδομένη χρονική στιγμή. Έπειτα αρχικοποιούμε τις τιμές δέκα μεταβλητών τύπου string που θα χρησιμοποιηθούν ως tags πάνω στα αντικείμενα της σκηνής και δεκαεννέα μεταβλητών τύπου GameObject που θα κρατάνε τα δεδομένα των δεκαεννέα αντικειμένων του πειράματος. Τέλος, αρχικοποιούμε μια μεταβλητή τύπου TextMeshProUGUI που θα αποθηκεύει το text το οποίο θα αλλάζει δυναμικά δίνοντας οδηγίες στον χρήστη ανάλογα με την φάση του πειράματος (Εικόνα 14.5).

```

9 public class Kef12pirama1 : MonoBehaviour
10 {
11
12     //selection----
13     private Transform selection;
14
15     public bool xartoniislifted = false;
16     public bool karfireadytolift = false;
17     public bool placekarfi = false;
18     public bool magnitesreadytolift = false;
19     public bool magnitis1lifted = false;
20     public bool magnitis2lifted = false;
21     public bool placemagnites = false;
22     public bool takekalodio = false;
23     public bool placekalodio = false;
24     public bool takebulb = false;
25     public bool bulblifted = false;
26
27     public bool lighton = false;
28
29
30     // [SerializeField] private string selectableTag = "Selectable";
31     [SerializeField] private string xartoniTag = "xartoniTag";
32     [SerializeField] private string magnitisTag = "magnitisTag";
33     [SerializeField] private string magnitis2Tag = "magnitis2Tag";
34     [SerializeField] private string karfiTag = "karfiTag";
35     [SerializeField] private string kalodioxalkinoTag = "kalodioxalkinoTag";
36     [SerializeField] private string xartonifoldedTag = "xartonifoldedTag";
37     [SerializeField] private string xartonifoldedhologramTag = "xartonifoldedhologramTag";
38     [SerializeField] private string xartonifoldedFinalTag = "xartonifoldedFinalTag";
39     [SerializeField] private string bulbTag = "bulbTag";
40     [SerializeField] private string bulbhologramTag = "bulbhologramTag";
41
42
43
44     public GameObject xartoni;
45     public GameObject xartonifolded;
46     public GameObject xartonifoldedhologram;
47     public GameObject xartonifoldedFinal;

```

Εικόνα 14.5 Αρχικοποίηση μεταβλητών του script Kef12pirama1script.

Έπειτα, στην μέθοδο Awake() τύπου void, με την ενεργοποίηση του script γίνεται αλλαγή του κειμένου του text αντικειμένου (infotext) της σκηνής και απενεργοποίηση των αντικειμένων TelosButton με την μέθοδο SetActive() με τιμή false. Το αντικείμενο TelosButton το ενεργοποιούμε ξανά όταν πρέπει να δοθεί στον χρήστη η δυνατότητα να φύγει από το πείραμα και να πάει στα αποτελέσματα. Το κουμπί που εμφανίζεται στη σκηνή είναι ένα πορτοκαλί κουμπί που αναγράφει την λέξη «ΤΕΛΟΣ» (Εικόνα 14.6).

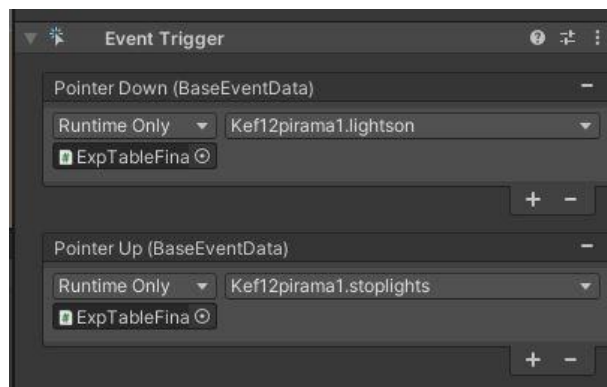
```

68 private void Awake()
69 {
70     Application.targetFrameRate = 30;
71     Debug.Log("Into the Script!!!");
72
73     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
74     infotext.text = "Επέλεξε το χαρτόνι για να το διπλώσεις.";
75
76     TelosButton.SetActive(false);
77     // ROTATEButton.SetActive(false);
78 }

```

Εικόνα 14.6 Η μέθοδος Awake() του script Kef12pirama1script.

Στη συνέχεια, μέσα στην μέθοδο Update(), όπως κάναμε και στο προηγούμενο script έτσι και εδώ χρησιμοποιούμε μία if, η οποία περιμένει το αριστερό click του χρήστη. Μέσα στην if, χρησιμοποιώντας την Raycast, η εφαρμογή ελέγχει εάν την ώρα που κάναμε το «κλικ» ο κέρσορας μας ήταν επάνω σε κάποιο αντικείμενο της σκηνής. Στη συνέχεια με μία ακόμη if ελέγχουμε εάν το αντικείμενο στο οποίο κάναμε «κλικ» είναι ένα από τα παρακάτω αντικείμενα με το tag value που του ορίσαμε στο Project. Ανάλογα με το ποιο αντικείμενο επιλέγει ο χρήστης συμβαίνουν κάποιες συγκεκριμένες λειτουργίες. Πιο συγκεκριμένα, όταν ο χρήστης επιλέξει ένα αντικείμενο της σκηνής που του επιτρέπεται να συλλέξει, τότε ο χρήστης «κρατάει» αυτό το αντικείμενο και η εφαρμογή περιμένει από αυτόν να το χρησιμοποιήσει ή να τοποθετήσει κάπου αλλού. Αρχικά, ο χρήστης επιλέγει το χαρτόνι, το οποίο έπειτα τυλίγεται αυτόματα στο σχήμα που πρέπει και εμφανίζεται το τυλιγμένο χαρτόνι στη σκηνή. Ο χρήστης επιλέγει τώρα το τυλιγμένο χαρτόνι και το τοποθετεί στην θέση που του υποδεικνύει η εφαρμογή με την χρήση ενός εικονικού ολογράμματος. Ο χρήστης, πατώντας πάνω στο εικονικό ολόγραμμα τοποθετεί το αντικείμενο που «κρατάει» στη θέση εκείνη και το εικονικό ολόγραμμα απενεργοποιείται. Έπειτα, πρέπει να πάρει το καρφί και στη συνέχεια να επιλέξει το χαρτόνι για να το τοποθετήσει επάνω του. Παρομοίως, ο μαθητής επιλέγει και τοποθετεί επάνω και γύρω από το χαρτόνι του δύο μαγνήτες και το χάλκινο καλώδιο της σκηνής. Αφού τοποθετηθούν τα αντικείμενα αυτά στις θέσεις που πρέπει, τότε θα έχει δημιουργηθεί ένα πηνίο. Στη συνέχεια, ο μαθητής πρέπει να σηκώσει την λάμπα από το τραπέζι και να το τοποθετήσει στην θέση που θα του υποδεικνύει η εφαρμογή με την χρήση ενός ακόμη εικονικού ολογράμματος. Αφού ο χρήστης τοποθετήσει την λάμπα στην θέση που πρέπει, τότε εμφανίζεται στην οθόνη του το κουμπί «ROTATE», το οποίο όταν είναι πατημένο έχει ως αποτέλεσμα να περιστρέφεται το καρφί. Όταν λοιπόν ο χρήστης κρατάει πατημένο το κουμπί αυτό καλείτε η μέθοδος του script lightson() ενώ όταν αφήνει το κουμπί καλείτε η μέθοδος stoplights(). Αυτή η συμπεριφορά του κουμπιού επιτυγχάνεται χάρη στο στοιχείο event trigger και των event «Pointer Down» και «Pointer Up» (Εικόνα 14.7) (Εικόνα 14.8).



Εικόνα 14.7 Τα στοιχεία event trigger του ROTATEButton κουμπιού.

```

230 public void lightson()
231 {
232     Debug.Log("yesyeyesy");
233     BulbLight.SetActive(true);
234     BulbParticle.SetActive(true);
235
236     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
237     infotext.text = "Τι παρατηρείς πως συμβαίνει όταν περιστρέφεις το καρφί;";
238 }
239 public void stoplights()
240 {
241     BulbLight.SetActive(false);
242     BulbParticle.SetActive(false);
243 }

```

Εικόνα 14.8 Οι μέθοδοι lightson() και stoplights() του script Kef4pirama1script.

Σε αυτό το σημείο ο μαθητής πρέπει να παρατηρήσει τι συμβαίνει όταν περιστρέφει το καρφί και τι όταν δεν το περιστρέφει, δηλαδή τι συμβαίνει κρατώντας το κουμπί πατημένο και τι όταν το αφήνει. Μετά από λίγο εμφανίζεται στη σκηνή το κουμπί τέλους και δίνεται στον μαθητή η δυνατότητα να το πατήσει για να πάει στην επόμενη σκηνή. Η καθυστέρηση στην εμφάνιση του κουμπιού ευθύνεται στην Invoke που καλεί την μέθοδο endexperiment() με καθυστέρηση δεκαοκτώ δευτερολέπτων. Η Invoke μπορεί να καλέσει μία μέθοδο μετά από μία συγκεκριμένη ώρα που θα ορίσει ο χρήστης (Εικόνα 14.9).

```

208 if (selection.CompareTag(bulbhologramTag) && bulblifted == true)
209 {
210     BulbTransparenthologram.SetActive(false);
211     BulbTransparentFinal.SetActive(true);
212
213     bulblifted = false;
214
215     infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
216     infotext.text = "Ξεκίνα να περιστρέφεις το καρφί κρατώντας πατημένο το κουμπί ROTATE.";
217
218     ROTATEButton.SetActive(true);
219     Invoke("endexperiment", 18.0f);
220 }

```

Εικόνα 14.9 Κάλεσμα της μεθόδου endexperiment().

Ο χρήστης την ώρα που περιμένει πρέπει να πειραματιστεί και να παρατηρήσει το πώς λειτουργεί ο αυτοσχέδιος ηλεκτρικός κινητήρας. Όλες αυτές οι διαδικασίες επιτυγχάνονται μέσα στο script με την συνεχή ενεργοποίηση και απενεργοποίηση των αντικειμένων της σκηνής ανάλογα με τις κινήσεις του χρήστη. Η ενεργοποίηση και η απενεργοποίηση των αντικειμένων αυτών γίνεται με την μέθοδο setActive(false/true) με τιμή true ή false. Εφόσον έχει εμφανιστεί το κουμπί τέλους στη σκηνή, δίνεται στον χρήστη η δυνατότητα να πατήσει το κουμπί αυτό και να μεταφερθεί στην επόμενη σκηνή αφού φυσικά έχει πρώτα παρατηρήσει τι έχει συμβεί με την τελική θερμοκρασία. Το κουμπί τέλους δεν πρέπει να είναι προσβάσιμο από την αρχή για να υποχρεώσει τον παίκτη να πειραματιστεί και να υλοποιήσει αυτά που του ζητούνται πριν πάει παρακάτω (Εικόνα 14.10).

```

225 void endexperiment()
226 {
227     TelosButton.SetActive(true);
228 }

```

Εικόνα 14.10 Η μέθοδος που εμφανίζει το κουμπί τέλους.

Στην επόμενη και τελευταία σκηνή του πειράματος (Φάση 5) η εφαρμογή θέτει κάποιες ερωτήσεις στον χρήστη σχετικά με το πείραμα που μόλις υλοποίησε και τα φαινόμενα που παρατήρησε. Για την απάντηση των ερωτήσεων αυτών ο μαθητής μπορεί να ζητήσει βοήθεια από κάποιο άλλο άτομο. Εφόσον ο χρήστης έχει τελειώσει με τους υπολογισμούς του, με την κατανόηση του πειράματος και των αποτελεσμάτων του, μπορεί να πατήσει το κουμπί «ΤΕΛΟΣ» για να φύγει από το πείραμα. Εάν ο χρήστης επιθυμεί να μελετήσει ξανά το πείραμα, τότε του δίνεται η ευκαιρία να επαναλάβει την υλοποίηση του πειράματος πατώντας το κουμπί «ΔΟΚΙΜΑΣΕ ΞΑΝΑ» (Εικόνα 14.11).



Εικόνα 14.11 Η δισδιάστατη σκηνή της πέμπτης φάσης του πειράματος του 12<sup>ου</sup> Κεφαλαίου.

### 14.3 Επίλογος

Το πείραμα του 12ου κεφαλαίου «Από το μαγνητισμό στον ηλεκτρισμό – Η ηλεκτρική (ιδιο-)γεννήτρια» του σχολικού βιβλίου της Φυσικής Α' Γυμνασίου, υλοποιείται με την AR εφαρμογή που έχουμε κατασκευάσει. Ο χρήστης συλλέγει μία λάμπα, ένα κομμάτι χαρτόνι, δύο μαγνήτες, ένα καρφί και ένα χάλκινο καλώδιο από τον εικονικό χώρο. Τα αντικείμενα αυτά τα χρησιμοποιεί για την κατασκευή μίας (ιδιο-)γεννήτριας. Στη συνέχεια, ο χρήστης περιστρέφει το καρφί που έχει τοποθετήσει ανάμεσα σε δύο μαγνήτες, με αποτέλεσμα να παράγεται ηλεκτρική ενέργεια η οποία τροφοδοτεί την λάμπα. Τέλος, παρατηρεί τις αλλαγές που συμβαίνουν στην λάμπα και απαντάει στα ερωτήματα που αναγράφονται στην σκηνή τέλους του παιχνιδιού.



## Κεφάλαιο 15ο: Συμπεράσματα και προτάσεις βελτίωσης

Η δημιουργία μιας AR εφαρμογής διδακτικού περιεχομένου αποδείχτηκε μια αρκετά σύνθετη διαδικασία. Η αρχική σχεδίαση βασίστηκε σε απλά και κατανοητά μοντέλα, τα οποία προτρέπουν τον χρήστη να πειραματιστεί με τα αντικείμενα που βρίσκονται στον επαυξημένο χώρο. Όπως έχουμε αναφέρει και στο κομμάτι της εισαγωγής στα παραπάνω κεφάλαια, οι AR τεχνολογίες παρουσιάζουν όλο και μεγαλύτερη χρήση σε πολλούς τομείς της καθημερινής μας ζωής. Λόγω της ραγδαίας εξέλιξης της επιστήμης της τεχνολογίας, τα άτομα σε νεαρή ηλικία είναι ιδιαίτερα εξοικιωμένα με την χρήση υπολογιστικών συσκευών και παρουσιάζουν ενδιαφέρον σε εφαρμογές-παιχνίδια που τους επιτρέπουν την χρήση τέτοιων συσκευών. Για τον λόγο αυτό παραθέτουμε ότι η υλοποίηση των καθηκόντων ενός σχολικού μαθήματος με την χρήση μιας AR εφαρμογής έχει την δυνατότητα να αποτελέσει ένα πολύτιμο εργαλείο για την μάθηση, διότι προσφέρει ένα παραπάνω ερέθισμα στον μαθητή, το οποίο είναι η χρήση και η εξερεύνηση νέων τεχνολογιών. Τέτοιου είδους γνώσεις μπορούν να φανούν χρήσιμες και σε άλλους τομείς που αξιοποιούν την τεχνολογία AR, σε μελλοντικό χρόνο.

Όσον αφορά το μάθημα της Φυσικής της Α΄ Γυμνασίου, πάνω στο οποίο δημιουργήθηκε η παρούσα εφαρμογή, διαθέτει πειράματα τα οποία μπορούν να υλοποιηθούν και στον πραγματικό αλλά και στον εικονικό χώρο. Δηλαδή, η χρήση της AR τεχνολογίας στην συγκεκριμένη περίπτωση δεν παρέχει απαραίτητα στον χρήστη την οπτικοποίηση φαινομένων τα οποία δεν θα μπορούσε να παρακολουθήσει σε ένα σχολικό εργαστήριο. Από την άλλη πλευρά όμως, δίνεται η δυνατότητα στον μαθητή να υλοποιήσει το πείραμα μόνος του, γεγονός που σε αρκετές περιπτώσεις δεν μπορεί να συμβεί σε μια τάξη με μεγάλο αριθμό ατόμων. Επιπλέον, η υλοποίηση ενός AR πειράματος δεν απαιτεί την χρήση υλικών και μηχανημάτων τα οποία αποτελούν δαπάνες για τα σχολεία. Ακόμη, ένα AR πείραμα μπορεί να υλοποιηθεί στον χώρο του σπιτιού, ιδιαίτερα σημαντικό προτέρημα για την περίοδο που διανύουμε λόγω πανδημίας, όπου τα σχολεία έμειναν και ίσως ξαναμείνουν για αρκετό καιρό κλειστά. Τέλος, τα πειράματα μιας AR εφαρμογής είναι εύκολο να επαναληφθούν εάν κάποιος από αυτά δεν έγινε κατανοητό από τον μαθητή, αλλά και εάν ο χρήστης παρουσιάσει ενδιαφέρον σε κάποιο συγκεκριμένο πείραμα και θελήσει να δοκιμάσει να το υλοποιήσει ξανά.

Για την βελτίωση της εφαρμογής, στις σκηνές υλοποίησης των πειραμάτων θα μπορούσαν να έχουν χρησιμοποιηθεί physics (βαρύτητα, συγκρούσεις) και λογικές drag and drop των αντικειμένων. Σε αυτή την περίπτωση, ο μαθητής θα είχε την δυνατότητα να επιλέξει ένα αντικείμενο και να το κουνήσει ελεύθερα στον τρισδιάστατο χώρο σε όποια κατεύθυνση αυτός ήθελε. Έτσι, όταν ο μαθητής αφήνει το αντικείμενο, αυτό πέφτει λόγω της χρήσης της βαρύτητας. Η εκδοχή όμως αυτή απορρίφθηκε διότι μία τέτοια υλοποίηση απαιτεί πολύ μεγάλη ευστοχία από τον χρήστη, αφού το πείραμα είναι σε AR. Συμπερασματικά, επειδή η εφαρμογή έχει διδακτικό σκοπό, μία πιο απλή υλοποίηση θεωρήθηκε καταλληλότερη για την κατανόηση των πειραμάτων.

Τέλος, ως μελλοντική επέκταση της εφαρμογής θα μπορούσε αρχικά να γίνει η έγκριση της από το κράτος και η προσθήκη των κωδικών (στόχων) στο βιβλίο της Φυσικής Α΄ Γυμνασίου. Ως επακόλουθο, η εφαρμογή μπορεί να προστεθεί στο Play Store της Google, από όπου οι μαθητές μπορούν να την εγκαταστήσουν δωρεάν. Επιπλέον, θα ήταν χρήσιμη η μετατροπή της εφαρμογής σε μορφή που θα υποστηρίζεται από συστήματα iOS (iPhone). Στην υπάρχουσα εφαρμογή, δίνεται η δυνατότητα να προστεθούν επιπλέον πειράματα σε κάθε κεφάλαιο, τα οποία θα προσδίδουν μία πιο ολοκληρωμένη όψη των φαινομένων που παρουσιάζονται. Επιπρόσθετα, θα ήταν καλό να προστεθούν

επιπλέον ηχητικά εφέ και μουσικές. Ακόμη, μία σημαντική βελτίωση στην εφαρμογή θα ήταν η εισαγωγή νέων συστημάτων, όπως ένα σύστημα επιβράβευσης του χρήστη και ένα σύστημα αποθήκευσης των πειραμάτων που έχουν υλοποιηθεί. Βελτιώσεις θα μπορούσαν να γίνουν επίσης, στο κομμάτι του κώδικα, στο ρυθμό ανανέωσης πλαισίου (framerate) και στην ποιότητα των γραφικών. Μαζί με την βελτίωση του κώδικα θα ήταν χρήσιμη η προσθήκη σχολίων σε αυτόν, έτσι ώστε να γίνεται πιο εύκολα κατανοητός από κάποιον πιθανό αναγνώστη (Πίνακας 15.1).

| Επεκτάσεις |   |
|------------|---|
| 1          | Προσθήκη κωδικών-εικόνων στόχων (AR image targets) στο σχολικό βιβλίο.                                      |
| 2          | Μεταφόρτωση στο Play Store.   |
| 3          | Προσθήκη πειραμάτων.  |
| 4          | Προσθήκη συστημάτων που θα ενημερώνουν τον χρήστη ποια πειράματα έχει υλοποιήσει στο παρελθόν και ποια όχι. |
| 5          | Προσθήκη ανταμοιβής του χρήστη.   |
| 6          | Προσθήκη ήχων.  |
| 7          | Βελτίωση γραφικών.  |
| 8          | Βελτίωση εκτέλεσης και αύξηση framerate.  |
| 9          | Βελτίωση κώδικα και προσθήκη σχολίων.   |
| 10         | Υποστήριξη εφαρμογής σε iOS , μεταφόρτωση στο App Store.  |

Πίνακας 15.1 Μελλοντικές επεκτάσεις της εφαρμογής.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

### Επιστημονικά Άρθρα – Βιβλία – Internet Sites

- [1] Irwansyah, F. S., Yusuf, Y. M., Farida, I., & Ramdhani, M. A. (2018). Augmented reality (AR) technology on the android operating system in chemistry learning. In IOP conference series: Materials science and engineering (Vol. 288, No. 1, p. 012068). IOP Publishing.
- [2] Carmigniani, J., & Furht, B. (2011). Augmented reality: an overview. Handbook of augmented reality, 3-46.
- [3] Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). Recent advances in augmented reality. IEEE computer graphics and applications, 21(6), 34-47.
- [4] Van Krevelen, D. W. F., & Poelman, R. (2010). A survey of augmented reality technologies, applications and limitations. International journal of virtual reality, 9(2), 1-20.
- [5] Wu, H. K., Lee, S. W. Y., Chang, H. Y., & Liang, J. C. (2013). Current status, opportunities and challenges of augmented reality in education. Computers & education, 62, 41-49.
- [6] Η Φυσική με Πειράματα Α' Γυμνασίου, Συγγραφική Ομάδα : Γεώργιος Θεοφ. Καλκάνης, Ουρανία Γκικοπούλου, Ευστράτιος Καπότης, Δημήτριος Γουσόπουλος, Ματθαίος Πατρινόπουλος, Παναγιώτης Τσάκωνας, Παναγιώτης Δημητριάδης, Λαμπρινή Παπατσιμπα, Κωνσταντίνος Μιτζήθρας, Αθανάσιος Καπόγιαννης, Δημήτριος Ι. Σωτηρόπουλος, Σάββας Πολίτης
- [7] <https://www.redhat.com/en/topics/middleware/what-is-ide>
- [8] Menard, M., & Wagstaff, B. (2012). Game development with Unity. Course Technology.
- [9] Haas, J. (2014). A history of the unity game engine. Diss. WORCESTER POLYTECHNIC INSTITUTE.
- [10] Blackman, S. (2013). Beginning 3D Game Development with Unity 4: All-in-one, multi-platform game development. Apress.
- [11] Liu11, X., Sohn, Y. H., & Park, D. W. (2018). Application development with augmented reality technique using unity 3D and Vuforia. International Journal of Applied Engineering Research, 13(21), 15068-15071.
- [12] <https://library.vuforia.com/features/objects/model-targets.html>
- [13] Alecu, F. (2010). Blender institute—the institute for open 3D projects. Open Source Science Journal, 2(1), 36-45.
- [14] Bauer, P. (2012). Photoshop CS6 For Dummies. John Wiley & Sons.
- [15] Joo, H. (2017). A study on understanding of UI and UX, and understanding of design according to user interface change. International Journal of Applied Engineering Research, 12(20), 9931-9935.
- [16] <https://www.toptal.com/designers/ux/ux-sounds-guide>
- [17] <https://slideplayer.gr/slide/12891749/>

## ΠΑΡΑΡΤΗΜΑ Α : Script “MainMenu.cs”

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

public class MainMenu : MonoBehaviour
{
    private void Awake()
    {
        QualitySettings.vSyncCount = 0;
        Application.targetFrameRate = 30;
    }
    // general next scene
    public void NextScene()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }

    //apo FirstScan se FindHelp
    public void VresTaAntikeimenaKef1pirama1()
    {
        SceneManager.LoadScene("Kef1pirama1FindHelp");
    }

    public void VresTaAntikeimenaKef2pirama1()
    {
        SceneManager.LoadScene("Kef2pirama1FindHelp");
    }

    public void VresTaAntikeimenaKef3pirama1()
    {
        SceneManager.LoadScene("Kef3pirama1FindHelp");
    }

    public void VresTaAntikeimenaKef4pirama1()
    {
        SceneManager.LoadScene("Kef4pirama1FindHelp");
    }

    public void VresTaAntikeimenaKef5pirama1()
    {
        SceneManager.LoadScene("Kef5pirama1FindHelp");
    }

    public void VresTaAntikeimenaKef6pirama1()
    {
        SceneManager.LoadScene("Kef6pirama1FindHelp");
    }

    public void VresTaAntikeimenaKef7pirama1()
    {
        SceneManager.LoadScene("Kef7pirama1FindHelp");
    }
    public void VresTaAntikeimenaKef7pirama2()
    {
        SceneManager.LoadScene("Kef7pirama2FindHelp");
    }
    public void VresTaAntikeimenaKef7pirama3()
```

```

{
    SceneManager.LoadScene("Kef7pirama3FindHelp");
}

public void VresTaAntikeimenaKef8pirama1()
{
    SceneManager.LoadScene("Kef8pirama1FindHelp");
}
public void VresTaAntikeimenaKef8pirama2()
{
    SceneManager.LoadScene("Kef8pirama2FindHelp");
}

public void VresTaAntikeimenaKef9pirama1()
{
    SceneManager.LoadScene("Kef9pirama1FindHelp");
}
public void VresTaAntikeimenaKef9pirama2()
{
    SceneManager.LoadScene("Kef9pirama2FindHelp");
}
public void VresTaAntikeimenaKef9pirama3()
{
    SceneManager.LoadScene("Kef9pirama3FindHelp");
}

public void VresTaAntikeimenaKef10pirama1()
{
    SceneManager.LoadScene("Kef10pirama1FindHelp");
}
public void VresTaAntikeimenaKef10pirama2()
{
    SceneManager.LoadScene("Kef10pirama2FindHelp");
}
public void VresTaAntikeimenaKef10pirama3()
{
    SceneManager.LoadScene("Kef10pirama3FindHelp");
}
public void VresTaAntikeimenaKef10pirama4()
{
    SceneManager.LoadScene("Kef10pirama4FindHelp");
}
}
public void VresTaAntikeimenaKef10pirama5()
{
    SceneManager.LoadScene("Kef10pirama5FindHelp");
}

public void VresTaAntikeimenaKef11pirama1()
{
    SceneManager.LoadScene("Kef11pirama1FindHelp");
}

public void VresTaAntikeimenaKef12pirama1()
{
    SceneManager.LoadScene("Kef12pirama1FindHelp");
}

//apo FindHelp skines se FindItems

public void FindItemsKef1pirama1()
{
    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
}

```

```

    }

    //general Back scene
    public void BackScene()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex - 1);
    }

    public void GoToEndScene()
    {
        SceneManager.LoadScene("EndScene");
    }

    public void GoToMainMenu()
    {
        SceneManager.LoadScene("MainMenuScene");
    }

    public void GoToFirstScanCodeScene()
    {
        SceneManager.LoadScene("FirstScanCodeScene");
    }

    public void EksodosPaixnidou()
    {
        Debug.Log("QUIT!");
        Application.Quit();
    }
}

```

## ΠΑΡΑΡΤΗΜΑ Β : Script “SettingsScript.cs”

```

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

public class SettingsScript : MonoBehaviour
{
    public AudioManager audioMixer;

    public void SetMasterVolume(float volume)
    {
        audioMixer.SetFloat("Mastervolume", volume);
    }

    public void SetMusicVolume(float volume)
    {
        audioMixer.SetFloat("Musicvolume", volume);
    }

    public void SetSoundsVolume(float volume)
    {
        audioMixer.SetFloat("Soundsvolume", volume);
    }
}

```

## ΠΑΡΑΡΤΗΜΑ C : Script “DemoRaycastKef8.cs”

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;
using UnityEngine.SceneManagement;
public class DemoRaycastKef8 : MonoBehaviour
{
    private Transform selection;

    //tags ----
    //[SerializeField] private string selectableTag = "Selectable";
    [SerializeField] private string thermometerTag = "Thermometer";
    [SerializeField] private string FotistikoKef8Tag = "FotistikoKef8Tag";
    [SerializeField] private string KollitikitainiaLeykiKef8Tag =
    "KollitikitainiaLeykiKef8Tag";
    [SerializeField] private string KollitikitainiaMayriKef8Tag =
    "KollitikitainiaMayriKef8Tag";

    //counters ----
    private TextMeshProUGUI countertextthermometro;
    private TextMeshProUGUI countertextfotistiko;
    private TextMeshProUGUI countertextkollitikitainialeyki;
    private TextMeshProUGUI countertextkollitikitainiamayri;

    public int thermometercounter = 0;
    public int fotistikocounter = 0;
    public int kollitikitainialeykicounter = 0;
    public int kollitikitainiamayricounter = 0;

    private void Awake()
    {
        QualitySettings.vSyncCount = 0;
        Application.targetFrameRate = 30;
    }

    // Start is called before the first frame update
    void Start()
    {
        countertextthermometro =
    GameObject.Find("thermometroText").GetComponent<TextMeshProUGUI>();
        countertextthermometro.text = "Θερμόμετρο : " + thermometercounter + "/2";

        countertextfotistiko =
    GameObject.Find("fotistikoText").GetComponent<TextMeshProUGUI>();
        countertextfotistiko.text = "Φωτιστικό : " + fotistikocounter + "/1";

        countertextkollitikitainialeyki =
    GameObject.Find("kollitikitainiaLeykiText").GetComponent<TextMeshProUGUI>();
        countertextkollitikitainialeyki.text = "Κολλητική Ταινία Λευκή : " +
    kollitikitainialeykicounter + "/1";

        countertextkollitikitainiamayri =
    GameObject.Find("kollitikitainiaMayriText").GetComponent<TextMeshProUGUI>();
```

```

        countertextkollitikitainiamayri.text = "Κολλητική Ταινία Μαύρη : " +
kollitikitainiamayricounter + "/1";
    }

    // Update is called once per frame
    void Update()
    {
        if (Input.GetMouseButtonDown(0))
        {
            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
            RaycastHit hit;

            if (Physics.Raycast(ray, out hit))
            {
                var selection = hit.transform;

                if (selection.CompareTag(thermometroTag))
                {
                    GameObject.DestroyObject(hit.transform.gameObject);
                    thermometrocounter = thermometrocounter + 1;
                    countertextthermometro.text = "Θερμόμετρο : " + thermometrocounter
+ "/2";
                }
                else if (selection.CompareTag(Fotistikokef8Tag))
                {
                    GameObject.DestroyObject(hit.transform.gameObject);
                    fotistikocounter = fotistikocounter + 1;
                    countertextfotistiko.text = "Φωτιστικό : " + fotistikocounter +
"/1";
                }
                else if (selection.CompareTag(KollitikitainialeykiKef8Tag))
                {
                    GameObject.DestroyObject(hit.transform.gameObject);
                    kollitikitainialeykicounter = kollitikitainialeykicounter + 1;
                    countertextkollitikitainialeyki.text = "Κολλητική Ταινία Λευκή : "
+ kollitikitainialeykicounter + "/1";
                }
                else if (selection.CompareTag(KollitikitainiamayriKef8Tag))
                {
                    GameObject.DestroyObject(hit.transform.gameObject);
                    kollitikitainiamayricounter = kollitikitainiamayricounter + 1;
                    countertextkollitikitainiamayri.text = "Κολλητική Ταινία Μαύρη : "
+ kollitikitainiamayricounter + "/1";
                }
            }
        }

        if (thermometrocounter == 2)
        {
            if (fotistikocounter == 1)
            {
                if (kollitikitainialeykicounter == 1)
                {
                    if (kollitikitainiamayricounter == 1)
                    {
                        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
                    }
                }
            }
        }
    }
}

```

```

    }
}
}

```

## ΠΑΡΑΡΤΗΜΑ D Script “Kef8pirama1.cs”

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;
using UnityEngine.SceneManagement;

public class Kef8pirama1 : MonoBehaviour
{
    //selection----
    private Transform selection;

    public bool thermometroislifted = false;
    public bool readytoswitchon = false;
    public bool lighton = false;

    // [SerializeField] private string selectableTag = "Selectable";
    [SerializeField] private string ThermometroTag = "Thermometro";
    [SerializeField] private string thermometrohologramTag = "thermometrohologramTag";
    [SerializeField] private string FotistikoKef8Tag = "FotistikoKef8Tag";

    public GameObject Thermometro;
    public GameObject ThermometroHologram;
    public GameObject ThermometroFinal;
    public GameObject Fotistiko;
    public GameObject SpotLightFotistiko;

    public GameObject TelosButton;
    private TextMeshProUGUI infotext;

    public GameObject thermometerimagezoomed;
    public GameObject thermometerindicatorimagezoomed;

    private void Awake()
    {
        Application.targetFrameRate = 30;
        Debug.Log("Into the Script!!!");

        infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
        infotext.text = "Πάρε το θερμόμετρο.";

        TelosButton.SetActive(false);
    }
}

```

```

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

}

// Update is called once per frame
void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        // Debug.Log("Mouse Button Down!!!");
        Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
        RaycastHit hit;

        if (Physics.Raycast(ray, out hit))
        {
            // Debug.Log("Hit Something!!!");
            var selection = hit.transform;
            if (selection.CompareTag(ThermometroTag) )
            {
                // Kremastra.transform.localPosition = new Vector3(-0.0057f,
1.8067f, 0.076f);
                // Kremastra.transform.localRotation = Quaternion.Euler(0, 90,
90f);

                Thermometro.SetActive(false);
                ThermometroHologram.SetActive(true);

                infotext =
GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
                infotext.text = "Τοποθέτησε το θερμόμετρο κάτω από το φωτιστικό.";

                thermometroislifted = true;
            }
            else if (selection.CompareTag(thermometrohologramTag) &&
thermometroislifted == true)
            {
                ThermometroHologram.SetActive(false);
                ThermometroFinal.SetActive(true);

                infotext =
GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
                infotext.text = "Παρατήρησε την θερμοκρασία.";

                Invoke("gettext", 5.0f);

                thermometerimagezoomed.SetActive(true);
                thermometerindicatorimagezoomed.SetActive(true);
            }
            if (selection.CompareTag(FotistikoKef8Tag) && readytoswitchon == true)
            {

                if (lighton == false)
                {
                    SpotlightFotistiko.SetActive(true);

thermometerindicatorimagezoomed.GetComponent<Animator>().Play("20to30slowanim");
                }
            }
            else
            {

```

```

        SpotlightFotistiko.SetActive(false);
thermometerindicatorimagezoomed.GetComponent<Animator>().Play("30to20slowanim");
    }

    lighton = !lighton;

    infotext =
GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
    infotext.text = "Παρατήρησε την θερμοκρασία με φως (και χωρίς).";

    Invoke("endexperiment", 9.0f);
    }
}

}

public void gettext()
{
    infotext = GameObject.Find("infotext").GetComponent<TextMeshProUGUI>();
    infotext.text = "Αναψε το φωτιστικό πατώντας επάνω του.";

    readytoswitchon = true;
}

public void endexperiment()
{
    TelosButton.SetActive(true);
}
}

```