

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Εκπαιδευτική εφαρμογή επαυξημένης  
πραγματικότητας για το μάθημα Αισθητικής Αγωγής  
Α' και Β' Δημοτικού»



Του φοιτητή  
Σκορδά Φώτη  
Αρ. Μητρώου: 164743

Επιβλέπων  
Κεραμόπουλος Ευκλείδης  
Βαθμίδα Καθηγητής

“Εκπαιδευτική εφαρμογή επαυξημένης πραγματικότητας για το μάθημα Αισθητικής Αγωγής Α’ και Β’ Δημοτικού”

23244

Ονοματεπώνυμο φοιτητή Σκορδάς Φώτιος

Ονοματεπώνυμο εισηγητή Κεραμόπουλος Ευκλείδης

Ημερομηνία ανάληψης Π.Ε 19 Οκτωβρίου 2023

Ημερομηνία περάτωσης Π.Ε. 2024

*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 Σκορδά Φώτη 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 απαραίτητα και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

## Πρόλογος

Με την ραγδαία εξέλιξη της τεχνολογίας στον τομέα της Πληροφορικής εμφανίστηκαν οι τεχνολογίες της επαυξημένης(AR), της εικονικής(VR) και της μικτής(MR) πραγματικότητας, που είναι ο συνδυασμός αυτών των δύο. Η επαυξημένη πραγματικότητα αποτελεί μια από τις σημαντικότερες αναδυόμενες εκπαιδευτικές τεχνολογίες, καθώς ο συνδυασμός του πραγματικού περιβάλλοντος με την ψηφιακή πληροφορία θεωρείται ότι παρέχει ενισχυμένες δυνατότητες για διδασκαλία και μάθηση, συνδυάζοντας μαζί την ψυχαγωγία και την αλληλεπίδραση των μαθητών. Η AR τεχνολογία έχει βοηθήσει εξίσου σημαντικά στους επαγγελματικούς τομείς, όπως είναι οι μηχανικοί, οι προγραμματιστές ή και ακόμα στον ιατρικό τομέα.

## Περίληψη

Η παρούσα πτυχιακή εργασία έχει ως στόχο την εκμάθηση με έναν ψυχαγωγικό τρόπο του τομέα της αισθητικής αγωγής, και συγκεκριμένα τα μαθήματα των “Εικαστικών” και της “Μουσικής” της Α’ και Β’ τάξης του Δημοτικού σχολείου. Για τον σκοπό αυτό δημιουργήθηκε μία εφαρμογή, που περιέχει διάφορα AR παιχνίδια και μη βασισμένα στα αντίστοιχα σχολικά βιβλία. Η εφαρμογή διατίθεται για κινητά τηλέφωνα (smart phones) τύπου android ή και tablet. Σε κάποια παιχνίδια οι μαθητές για να ξεκινήσουν το παιχνίδι θα πρέπει να σκανάρουν με την κάμερα του κινητού ή του tablet τους κάποιες συγκεκριμένες εικόνες που θα τις βρουν στα σχολικά τους βιβλία. Κάποια από αυτά τα παιχνίδια έχουν την ιδιότητα της αλληλεπίδρασης μεταξύ των μαθητών. Η εφαρμογή αυτή αναπτύχθηκε με το πρόγραμμα Unity Game Engine. Για την επίτευξη της επαυξημένης πραγματικότητας, που έπαιξε σημαντικό ρόλο, χρησιμοποιήθηκε η βιβλιοθήκη Vuforia Engine και τέλος, χρησιμοποιήθηκε το πρόγραμμα δημιουργίας τρισδιάστατων αντικειμένων Blender, όπου υπήρχε αλληλεπίδραση με 3D αντικείμενα ή το photoshop και illustrator, όπου είχαμε 2D για κύρια λειτουργικότητα.

# Augmented reality educational application for the A' and B' grade in the school subject of Aesthetic Education course

Skordas Fotis

## **Abstract**

The present thesis aims to learn in a recreational way the field of aesthetic education and specifically the lessons of visual arts and music in primary school grades A and B. For this purpose, an application was created that contains various AR games and non based on the respective textbooks. The app is available for smart phones (android or tablet). In some games, in order to start the game, the students have to scan with the camera of their mobile or tablet some specific images that they will find in their textbooks. This application was developed with the Unity Game Engine. To achieve the augmented reality that played an important role the Vuforia Engine library was used and finally the 3D object creation program Blender was used, where there was interaction with 3D objects or photoshop and illustrator where we had 2D for main functionality.

## Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τους φίλους, συμφοιτητές, συναδέλφους και οικογένεια για την αμέριστη στήριξη και συμπαράσταση καθ' όλη την διάρκεια των σπουδών μου, καθώς ο καθένας βοήθησε διαφορετικά με τον τρόπο του. Θα ήθελα ακόμη να ευχαριστήσω τον επιβλέποντα καθηγητή Κ. Κεραμόπουλο Ευκλείδη για την άψογη συνεργασία που είχαμε κατά την εκπόνηση της πτυχιακής εργασίας και για την δυνατότητα που μου παρείχε, ώστε να κατορθώσω να εκπονήσω μια τόσο ενδιαφέρουσα εργασία, ώστε να αναπτύξω τις γνώσεις μου στον συγκεκριμένο κλάδο. Τέλος, θα ήθελα να ευχαριστήσω την κοπέλα μου τόσο για την συμπαράσταση και την στήριξη που μου προσέφερε όσο και για την αντικειμενική κριτική της, ώστε να διεξαχθεί ένα ικανοποιητικό αποτέλεσμα.

# Περιεχόμενα

Πρόλογος.....	iii
Περίληψη.....	iv
Abstract.....	v
Ευχαριστίες.....	vi
Περιεχόμενα.....	vii
Κατάλογος Σχημάτων.....	x
Συντομογραφίες.....	xii
<b>Κεφάλαιο 1ο: Εισαγωγή.....</b>	<b>1</b>
<b>Κεφάλαιο 2ο: Τεχνολογίες απεικόνισης και χρήσεις στην Εκπαίδευση.....</b>	<b>3</b>
2.1 Εισαγωγή.....	3
2.2 Εκτεταμένη Πραγματικότητα.....	3
2.2.1 Εικονική πραγματικότητα.....	3
2.2.2 Επαυξημένη πραγματικότητα.....	4
2.2.3 Μικτή πραγματικότητα.....	5
2.3 Χρήσεις XR.....	5
2.3.1 Εκπαίδευση .....	5
2.3.2 Μάρκετινγκ.....	7
2.3.3 Υγεία.....	8
2.3.4 Μηχανική.....	9
2.3.5 Τουρισμός.....	10
2.4 Επίλογος.....	11
<b>Κεφάλαιο 3ο: Προγράμματα και εφαρμογές.....</b>	<b>13</b>
3.1 Εισαγωγή.....	13
3.2 Unity.....	13
3.2.1 Εγκατάσταση Unity Hub και Unity.....	13
3.2.2 Δημιουργία νέου Project.....	15
3.3 Vuforia Engine.....	16
3.3.1 Εισαγωγή.....	16
3.3.2 Εγκατάσταση Vuforia Engine package.....	16
3.3.3 Βασικές ρυθμίσεις Vuforia Engine.....	17
3.3.4 Δημιουργία Βάσης.....	19

3.4	Unity Set Up.....	23
3.5	Build ρυθμίσεις.....	25
3.6	Photoshop.....	26
3.7	Blender.....	27
3.8	Επίλογος.....	27
<b>Κεφάλαιο 4ο: Αρχική Οθόνη.....</b>		<b>29</b>
4.1	Εισαγωγή.....	29
4.2	Πρώτο μενού.....	29
4.3	Scripts.....	30
4.4	Δεύτερο μενού.....	32
4.5	Scenes.....	33
4.6	Fonts.....	34
4.7	Ηχοι.....	34
4.8	Επίλογος.....	35
<b>Κεφάλαιο 5ο: Μαθαίνοντας τα μουσικά όργανα.....</b>		<b>37</b>
5.1	Εισαγωγή.....	37
5.2	Λειτουργία drag and drop.....	38
5.2.1	Έλεγχος σωστής κατηγορίας.....	39
5.3	Spawn οργάνου.....	41
5.4	Οθόνη νίκης.....	42
5.5	Επίλογος.....	43
<b>Κεφάλαιο 6ο: Μαθαίνοντας τις νότες.....</b>		<b>45</b>
6.1	Εισαγωγή.....	45
6.2	Πιάνο.....	45
6.3	Script.....	46
6.4	Επίλογος.....	49
<b>Κεφάλαιο 7ο: Μαθαίνοντας τα σχήματα.....</b>		<b>51</b>
7.1	Εισαγωγή.....	51
7.2	UI.....	51
7.3	Script.....	53
7.3.1	Έλεγχος σωστής κατηγορίας.....	54
7.3.2	Spawn τυχαίου σχήματος.....	55
7.4	Επίλογος.....	57
<b>Κεφάλαιο 8ο: Πίνακας ζωγραφικής.....</b>		<b>59</b>
8.1	Εισαγωγή.....	59

8.2	Πίνακας ζωγραφικής.....	59
8.2.1	UI.....	59
8.2.2	Script.....	60
8.2.3	Τυχαία λέξη.....	61
8.2.4	Οθόνη νίκης.....	62
8.3	Επίλογος.....	63
<b>Κεφάλαιο 9ο: QUIZ.....</b>		<b>65</b>
9.1	Εισαγωγή.....	65
9.2	UI.....	65
9.3	Script.....	66
8.3.1	Έλεγχος απάντησης.....	66
8.3.2	Μέθοδος nextQuestion().....	67
9.4	Οθόνη νίκης.....	67
9.5	Επίλογος.....	67
<b>Κεφάλαιο 10ο: Παιχνίδι μνήμης.....</b>		<b>69</b>
10.1	Εισαγωγή.....	69
10.2	UI.....	69
10.3	Script.....	70
10.3.1	Έλεγχος δύο καρτών.....	71
10.3.2	Μέθοδος CheckIfTheGameIsFinished() και οθόνη νίκης.....	72
10.4	Επίλογος.....	73
<b>Κεφάλαιο 11ο: Κρυπτόλεξο.....</b>		<b>75</b>
11.1	Εισαγωγή.....	75
11.2	UI.....	75
11.3	Script.....	76
11.3.1	Spawn grid.....	77
11.3.2	Μέθοδος Drug.....	78
11.3.3	Μέθοδος checkWord.....	79
11.3.4	Μέθοδος CheckBoardCompleted.....	80
11.3.5	Timer.....	80
11.4	Οθόνη νίκης.....	81
11.5	Επίλογος.....	81
<b>Κεφάλαιο 12ο: Κρεμάλα.....</b>		<b>83</b>
12.1	Εισαγωγή.....	83
12.2	UI.....	83

12.3	Script.....	84
12.3.1	Τυχαία λέξη.....	85
12.3.2	Μέθοδος GenerateWord().....	86
12.3.3	Εισαγωγή λέξης.....	86
12.3.4	Έλεγχος γραμμάτων.....	87
12.3.5	Έλεγχος λέξης.....	87
12.4	Οθόνη νίκης και οθόνη ήττας.....	88
12.5	Επίλογος.....	89
<b>Κεφάλαιο 13ο:</b>	<b>Συμπεράσματα και προτάσεις βελτίωσης.....</b>	<b>91</b>
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>		<b>92</b>
ΠΑΡΑΡΤΗΜΑ Α :	Αρχείο GameManager.cs αρχικής οθόνης.....	93
ΠΑΡΑΡΤΗΜΑ Β :	Αρχείο CustomEditor.cs έβδομου παιχνιδιού.....	94

## Κατάλογος Σχημάτων

Σχήμα 2.1	VR headset.....	3
Σχήμα 2.2	Παιχνίδι Pokemon GO.....	4
Σχήμα 2.3	Apple vision Pro.....	6
Σχήμα 2.4	AR Educational.....	6
Σχήμα 2.5	Rolex AR app.....	8
Σχήμα 2.6	AR στον τομέα της υγείας.....	9
Σχήμα 2.7	AR στον τομέα της μηχανικής.....	10
Σχήμα 2.8	AR στον τομέα του τουρισμού.....	11
Σχήμα 3.1	Unity Hub.....	13
Σχήμα 3.2	Unity Hub install έκδοσης.....	14
Σχήμα 3.3	Unity Hub install libraries.....	14
Σχήμα 3.4	Δημιουργία νέου project.....	15
Σχήμα 3.5	Ρυθμίσεις νέου project.....	15
Σχήμα 3.6	Import Vuforia engine package.....	16
Σχήμα 3.7	Δημιουργία τις AR Camera.....	17
Σχήμα 3.8	Vuforia Engine configuration.....	17
Σχήμα 3.9	Έκδοση License Key.....	18
Σχήμα 3.10	Σελίδα License Key.....	18
Σχήμα 3.11	Επικόλληση License Key στο configuration menu.....	19
Σχήμα 3.12	Δημιουργία Βάσης Δεδομένων.....	19
Σχήμα 3.13	Επιλογή Add Target.....	20
Σχήμα 3.14	Εμφανίσει του image database.....	21
Σχήμα 3.15	Rating τις αναγνωρίσιμες εικόνας.....	21
Σχήμα 3.16	Δημιουργία image target.....	22
Σχήμα 3.17	Component Image Target Behaviour.....	22
Σχήμα 3.18	Επιλογή διαστάσεων οθόνης.....	23
Σχήμα 3.19	Εισαγωγή διαστάσεων στον Canva.....	23
Σχήμα 3.20	Switch platform σε android.....	24
Σχήμα 3.21	Ρυθμίσεις αρχιτεκτονικής και λογισμικού.....	24
Σχήμα 3.22	Build του παιχνιδιού στο κινητό μας.....	25
Σχήμα 3.23	Photoshop logo.....	26
Σχήμα 3.24	Button που δημιουργήθηκε στο photoshop.....	26
Σχήμα 3.25	Δημιουργία μοντέλου στο blender.....	27
Σχήμα 4.1	Αρχική οθόνη.....	29
Σχήμα 4.2	Script για εναλλαγή μενού.....	30
Σχήμα 4.3	Δημιουργία on click event και εισαγωγή function.....	31
Σχήμα 4.4	Δεύτερο μενού.....	32
Σχήμα 4.5	SceneSwitcherManager function.....	32
Σχήμα 4.6	Όλες οι σκηνές της εφαρμογής.....	33
Σχήμα 4.7	Function loadScene().....	33

Σχήμα 4.8	Επιλογή Font.....	34
Σχήμα 4.9	Function sound.play().....	34
Σχήμα 5.1	Εκκίνηση πρώτου παιχνιδιού.....	37
Σχήμα 5.2	Function Input.GetTouch().....	38
Σχήμα 5.3	Collider μουσικού οργάνου.....	39
Σχήμα 5.4	Script σωστού ελέγχου.....	39
Σχήμα 5.5	Scripts για διαφορετικά tags.....	40
Σχήμα 5.6	function SpawnObjectsOnDestroy().....	41
Σχήμα 5.8	Οθόνη νίκης.....	42
Σχήμα 6.1	AR πιάνο.....	45
Σχήμα 6.2	Εικόνα για scanning.....	45
Σχήμα 6.3	function open-close του παράθυρου.....	46
Σχήμα 6.4	Νότες μέσα στο script.....	46
Σχήμα 6.5	Function note.play().....	47
Σχήμα 6.6	Εισαγωγή μεθόδου στο κουμπί.....	47
Σχήμα 6.7	Μέθοδοι open-close των δυο frames.....	48
Σχήμα 6.8	Παράθυρα τραγουδιών.....	48
Σχήμα 7.1	Εικόνα για scanning.....	51
Σχήμα 7.2	Function open-close του παράθυρου.....	51
Σχήμα 7.3	Πλάκες με σχήματα.....	52
Σχήμα 7.4	Function Input.GetTouch().....	53
Σχήμα 7.5	Collider σχήματος.....	54
Σχήμα 7.6	Tag σχήματος.....	54
Σχήμα 7.7	RigidBody σχήματος.....	54
Σχήμα 7.8	Script σωστού ελέγχου.....	55
Σχήμα 7.9	Function SpawnObjectsOnDestroy.....	55
Σχήμα 7.10	Οθόνη νίκης.....	56
Σχήμα 8.1	Λευκός πίνακας.....	59
Σχήμα 8.2	Update μέθοδος.....	60
Σχήμα 8.3	Μετατροπή pixels σε συντεταγμένες.....	60
Σχήμα 8.4	Δημιουργία λίστας και επιλογή τυχαία λέξης.....	61
Σχήμα 8.5	Function changeTitleValue.....	61
Σχήμα 8.6	Οθόνη νίκης.....	62
Σχήμα 9.1	UI Quiz.....	65
Σχήμα 9.2	Εισαγωγή ερωτήσεων.....	66
Σχήμα 9.3	Function CheckAnswer ().....	66
Σχήμα 9.4	Function nextQuestion().....	67
Σχήμα 9.5	Οθόνη νίκης.....	67
Σχήμα 10.1	UI παιχνίδι μνήμης.....	69
Σχήμα 10.2	Εικόνα σκαναρισματούς.....	70
Σχήμα 10.3	Δημιουργία καρτών.....	70
Σχήμα 10.4	Function getButtons().....	70
Σχήμα 10.5	Event pickAPuzzle().....	71

Σχήμα 10.6	Μέθοδος για Έλεγχο δυο καρτών.....	71
Σχήμα 10.7	Function CheckIfThePuzzlesMatch().....	72
Σχήμα 10.8	Οθόνη νίκης.....	72
Σχήμα 11.1	UI κρυπτόλεξου.....	75
Σχήμα 11.2	Custom editor.....	76
Σχήμα 11.3	Function spawnGridSquares().....	77
Σχήμα 11.4	Function selectRay().....	78
Σχήμα 11.5	Function squareSelected().....	79
Σχήμα 11.6	Function checkWord().....	79
Σχήμα 11.7	Function CheckBoardCompleted().....	80
Σχήμα 11.8	Function Timer().....	80
Σχήμα 11.9	Οθόνη νίκης.....	81
Σχήμα 11.10	Οθόνη ήττας.....	81
Σχήμα 12.1	Οθονή σκαναρίσματος.....	83
Σχήμα 12.2	UI κρεμάλας.....	84
Σχήμα 12.3	Επιλογή mode παιχνιδιού.....	84
Σχήμα 12.4	Εισαγωγή λέξης.....	85
Σχήμα 12.5	Function InitialiseGame().....	85
Σχήμα 12.6	Script WordList.....	86
Σχήμα 12.7	Function TeacherMode().....	86
Σχήμα 12.8	Function CheckLetter().....	87
Σχήμα 12.9	Function CheckOutCome().....	87
Σχήμα 12.10	Οθόνη νίκης.....	88
Σχήμα 12.11	Οθόνη ήττας.....	88

## Συντομογραφίες

Π.Ε.	Πτυχιακή Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
AR	Επαυξημένη Πραγματικότητα
XR	Εκτεταμένη Πραγματικότητα
VR	Εικονική Πραγματικότητα
MR	Μικτή Πραγματικότητα
AI	Τεχνητή Νοημοσύνη
UX	User Experience
UI	User Interface

## Κεφάλαιο 1ο: Εισαγωγή

Η παρούσα πτυχιακή εργασία με όνομα «Εκπαιδευτική εφαρμογή επαυξημένης πραγματικότητας για το μάθημα της Αισθητικής Αγωγής της Α' και Β' Δημοτικού» αποτελείται από μια ομάδα παιχνιδιών επαυξημένης πραγματικότητας και μη. Ο κύριος σκοπός της είναι η βοήθεια και η διευκόλυνση προς τους μαθητές για την εκμάθηση στον συγκεκριμένο τομέα, με έναν εύκολο και ψυχαγωγικό τρόπο. Η πτυχιακή εργασία βασίζεται στα σχολικά βιβλία των αντίστοιχων μαθημάτων.

Η εφαρμογή χρησιμοποιεί σε μεγάλο βαθμό την τεχνολογία της επαυξημένης πραγματικότητας. Δημιουργήθηκε στο game engine unity και με την βοήθεια της βιβλιοθήκης vuforia Engine στο κομμάτι του AR διεξήχθη ένα πετυχημένο αποτέλεσμα. Η παρούσα πτυχιακή εργασία αποτελείται από 13 συνολικά κεφάλαια.

Στο δεύτερο κεφάλαιο με τίτλο «Τεχνολογίες απεικόνισης και χρήσεις στην Εκπαίδευση» παρουσιάζονται με κάποιες επιπρόσθετες και σημαντικές πληροφορίες, οι τεχνολογίες που έχουν χρησιμοποιηθεί στην πτυχιακή εργασία. Έπειτα, γίνεται μια αναφορά για την αξιοποίηση αυτών στον επαγγελματικό τομέα, καθώς επίσης και των θετικών αποτελεσμάτων που μπορούν να επιφέρουν.

Στο τρίτο κεφάλαιο με τίτλο «Προγράμματα και εφαρμογές» αναφέρονται τα προγράμματα και οι εφαρμογές που έχουν χρησιμοποιηθεί και δίνονται αναλυτικές οδηγίες για την εγκατάσταση και το setup των ρυθμίσεων. Επιπλέον, στο παρόν κεφάλαιο παρουσιάζεται βήμα - βήμα η δημιουργία του unity project και οι απαραίτητες ρυθμίσεις, που χρειάζονται για να “τρέξει” σε μια συσκευή τύπου android.

Στο τέταρτο κεφάλαιο με τίτλο «Αρχική οθόνη» παρουσιάζεται η αρχική οθόνη της εφαρμογής και γίνεται μια αναλυτική περιγραφή για το πως δημιουργήθηκε.

Από το πέμπτο κεφάλαιο μέχρι και το δωδέκατο κεφάλαιο αναφέρονται λεπτομερώς όλα τα βήματα που ακολουθήθηκαν για την δημιουργία των παιχνιδιών. Συμπεριλαμβάνονται εικόνες και αρχεία scripts για ευκολία προς την ανάγνωση. Πιο συγκεκριμένα, το κεφάλαιο πέντε «*Μαθαίνοντας τα μουσικά όργανα*», το κεφάλαιο έξι «*Μαθαίνοντας τις νότες*», το κεφάλαιο επτά «*Μαθαίνοντας τα σχήματα*», το κεφάλαιο οχτώ «*Μαθαίνοντας τα χρώματα*», το κεφάλαιο εννέα «*QUIZ*», το κεφάλαιο δέκα «*Παιχνίδι μνήμης*», το κεφάλαιο ένδεκα «*Κρυπτόλεξο*», το κεφάλαιο δώδεκα «*Κρεμάλα*».

Στο δέκατο τρίτο κεφάλαιο με τίτλο «*Συμπεράσματα και προτάσεις βελτίωσης*» παρουσιάζονται τα κύρια συμπεράσματα που προέκυψαν από την εκπόνηση της πτυχιακής εργασίας και κάποιες προτάσεις για μελλοντικά σχέδια, ώστε να βελτιωθεί η εφαρμογή.

Τέλος, προστέθηκαν δυο παραρτήματα με περιεχόμενο κώδικα που χρησιμοποιήθηκε στην εφαρμογή, τα οποία δεν συμπεριλήφθηκαν στα ανάλογα κεφάλαια.



## Κεφάλαιο 2ο: Τεχνολογίες απεικόνισης και χρήσεις στην εκπαίδευση.

### 2.1 Εισαγωγή

Στο κεφάλαιο αυτό θα γίνει μια αναφορά στις τεχνολογίες, όπως είναι η εκτεταμένη πραγματικότητα (XR), η επαυξημένη πραγματικότητα (AR), η μικτή πραγματικότητα (MR) καθώς και η εικονική πραγματικότητα (VR). Οι ραγδαίες τεχνολογικές εξελίξεις κατέχουν τη σπουδαιότερη θέση στη σύγχρονη κοινωνία χωρίς να αφήνουν ανεπηρέαστη και την εκπαίδευση. Οι τεχνολογίες της επαυξημένης πραγματικότητας και της εικονικής πραγματικότητας έχουν φέρει επανάσταση στις μαθησιακές προσεγγίσεις μέσω της καθηγητικής, ψηφιακής εμπειρίας του διαδραστικού περιβάλλοντος, της προσομοίωσης και της εμπλοκής. Ωστόσο, οι τεχνολογίες αυτές βρίσκονται σε στάδιο ανάπτυξης και απαιτούν μαζικές επενδύσεις και μαζική προσαρμογή για να καλύψουν την υψηλή ζήτηση στην εκπαίδευση.

### 2.2 Εκτεταμένη Πραγματικότητα

Η εκτεταμένη πραγματικότητα (XR) είναι μια πρωτοποριακή τεχνολογία που συνδυάζει τα πραγματικά και τα εικονικά περιβάλλοντα, φέρνοντας επανάσταση σε διάφορους κλάδους συμπεριλαμβανομένης και της εκπαίδευσης. Η XR περιλαμβάνει την Επαυξημένη Πραγματικότητα (AR), τη Μικτή Πραγματικότητα (MR) και την Εικονική Πραγματικότητα (VR), προσφέροντας μοναδικές καθηλωτικές εμπειρίες στους χρήστες. Στο πλαίσιο της διδασκαλίας, η XR έχει τεράστιες δυνατότητες να μετατρέψει τις παραδοσιακές αίθουσες διδασκαλίας σε δια-δραστικούς και ελκυστικούς χώρους μάθησης [4].

#### 2.2.1 Εικονική πραγματικότητα

Η τεχνολογία εικονικής πραγματικότητας (VR) εξακολουθεί να είναι ένας ταχέως εξελισσόμενος και διαφοροποιούμενος τομέας. Η εξέλιξή της έχει υποστεί σχεδόν δύο δεκαετίες έρευνας και ανάπτυξης και έχει φθάσει αφενός σε μια αξιοπρεπή ωριμότητα, αφετέρου όμως δεν έχει φθάσει ακόμη στην ευρεία διάδοση ούτε στον επαγγελματικό τομέα ούτε στον οικιακό ή στον τομέα της ψυχαγωγίας. Ένα παράδειγμα ενός VR headset φαίνεται στο σχήμα 2.1. Τα περιβάλλοντα της εικονικής πραγματικότητας διαφέρουν από τα συμβατικά επιτραπέζια περιβάλλοντα με την έννοια ότι ενσωματώνουν τον χρήστη σε ένα περιβάλλον δεδομένων που δημιουργείται από τον υπολογιστή.



Σχήμα 2.1 VR headset

### 2.2.2 Επαυξημένη πραγματικότητα

Η επαυξημένη πραγματικότητα (AR) είναι μια βελτιωμένη εκδοχή του πραγματικού κόσμου, η οποία επιτυγχάνεται με τη χρήση ψηφιακών πληροφοριών, που παράγονται από τον υπολογιστή. Αυτές περιλαμβάνουν οπτικά, ηχητικά και άλλα αισθητηριακά στοιχεία [3]. Η AR χρησιμοποιεί υλικό και λογισμικό υπολογιστή, όπως εφαρμογές, κονσόλες, οθόνες ή προβολές, προκειμένου να συνδυάσει ψηφιακές πληροφορίες με το περιβάλλον του πραγματικού κόσμου [1].

Η επαυξημένη πραγματικότητα συνεχίζει να αναπτύσσεται και να γίνεται όλο και πιο διαδεδομένη σε ένα ευρύ φάσμα εφαρμογών. Από το ξεκίνημά της, οι έμποροι και οι εταιρείες τεχνολογίας έπρεπε να καταπολεμήσουν την αντίληψη ότι η επαυξημένη πραγματικότητα είναι κάτι περισσότερο από ένα εργαλείο μάρκετινγκ.

Το Pokémon GO ήταν ένα από τα πρώτα mainstream παιχνίδια κινητής τηλεφωνίας που ενσωμάτωσαν με επιτυχία την επαυξημένη πραγματικότητα (AR). Οι παίκτες χρησιμοποιούν την κάμερα του smartphone τους για να τραβήξουν φωτογραφίες εικονικών Pokémon σε φυσικές τοποθεσίες. Για παράδειγμα, παίζοντας στην αυλή τους, στεκόμενοι στο τραπέζι της κουζίνας τους, ποζάροντας δίπλα στον εαυτό τους ή ακόμη δίπλα και σε άλλους ανθρώπους. Αυτό το χαρακτηριστικό, που χρησιμοποιήθηκε στο διαφημιστικό υλικό του παιχνιδιού, αποτέλεσε μια επιτυχημένη τακτική διαδικτυακού μάρκετινγκ [2].



Σχήμα 2.2 Παιχνίδι Pokemon GO.

### 2.2.3 Μικτή πραγματικότητα

Η μικτή πραγματικότητα (MR) είναι μια υβριδική τεχνολογία, όπου ο πραγματικός κόσμος συγχωνεύεται με εικονικά ερεθίσματα. Ο όρος 'Μικτή Πραγματικότητα' επινοήθηκε από τους Milgram και Kishino στην εργασία "The Taxonomy of Mixed Reality Virtual Devices" το 1994. Περιλαμβάνει τεχνολογίες AR και VR που επικαλύπτουν εικονικά αντικείμενα πάνω από φυσικές οντότητες. Η τεχνολογία MR είναι το αποτέλεσμα της συγχώνευσης του φυσικού κόσμου. Η MR μπορεί να γίνει βίωμα με τη χρήση ειδικών συσκευών π.χ. Microsoft Hololens ή εφαρμογές κινητών τηλεφώνων.

Επιπρόσθετα, αξίζει να αναφερθεί ότι η εταιρία Apple κυκλοφόρησε το 2023 το vision pro headset σχήμα, το οποίο φαίνεται και στο σχήμα 2.3. Η ισχυρή κάμερα και οι δυνατότητες AI/AR/VR του Vision Pro θα μπορούσαν να χρησιμοποιηθούν για τη δημιουργία πιο ελκυστικών και διαδραστικών εμπειριών μάθησης. Για παράδειγμα, οι φοιτητές θα μπορούσαν να χρησιμοποιήσουν το Apple Vision Pro στην τριτοβάθμια εκπαίδευση για να κάνουν τρισδιάστατες σαρώσεις αντικειμένων και στη συνέχεια, να χρησιμοποιήσουν αυτές τις σαρώσεις για να δημιουργήσουν εικονικά μοντέλα. Αυτό θα μπορούσε να αξιοποιηθεί για την εκμάθηση της ανατομίας, της φυσικής, της αστρολογίας, της μηχανικής ή άλλων θεμάτων.

## 2.3 Χρήσεις XR

Η τεχνολογία της εκτεταμένης πραγματικότητας (XR) χρησιμοποιείται ήδη σε διάφορους τομείς που συναντάμε καθημερινά, όπως είναι η εκπαίδευση, το μάρκετινγκ, η υγεία, η μηχανική, ακόμα και ο τουρισμός [3].

### 2.3.1 Εκπαίδευση

Τα εκπαιδευτικά ιδρύματα στοχεύουν όλο και περισσότερο στη βελτίωση των διαδικασιών διδασκαλίας και μάθησης με την ενσωμάτωση της τεχνολογίας. Ο μετασχηματισμός της διδασκαλίας και της μάθησης, που προκαλείται από την πρόοδο της τεχνολογίας, παρέχει την ευκαιρία να σχεδιαστούν μαθησιακά περιβάλλοντα, τα οποία είναι ρεαλιστικά, αυθεντικά, ελκυστικά και διασκεδαστικά. Μία τέτοια εφαρμογή φαίνεται στο σχήμα 2.4. Η AR στην εκπαίδευση, προσφέρει τη δυνατότητα να μετασχηματίσει την αλληλεπίδραση ανθρώπου-υπολογιστή και κερδίζει ολοένα και μεγαλύτερη δημοτικότητα στον τομέα αυτό.

Ενώ οι έρευνες αναδεικνύουν τη δυνατότητα της AR να βελτιώσει την ακαδημαϊκή αυτό-αποτελεσματικότητα των μαθητών, ελάχιστες έρευνες διερευνούν τους παράγοντες του πλαισίου (αξία του έργου, χαρακτηριστικά της τεχνολογίας και χώρος μάθησης) στον οποίο χρησιμοποιείται η AR σε εκπαιδευτικά περιβάλλοντα, καθώς επίσης και τον αντίκτυπο που έχει στην ακαδημαϊκή αυτό-αποτελεσματικότητα των μαθητών. Παρά ταύτα, είναι σημαντικό να πραγματοποιηθεί η συγκεκριμένη διερεύνηση, καθώς η χρήση της AR στο εκπαιδευτικό πλαίσιο συνεχώς και αυξάνεται. Υποστηρίζεται περαιτέρω, ότι η ενσωμάτωση της τεχνολογίας στις διδακτικές πρακτικές έχει παρατηρήσει μια αύξηση των τροποποιήσεων στο περιβάλλον της τάξης, όπου οι παραδοσιακές σειρές θρανίων και καρεκλών αντικαθίστανται από μια σειρά επίπλων που μπορούν να διαμορφωθούν με διάφορους τρόπους για να διευκολύνουν τη διδασκαλία και τη μάθηση.



Σχήμα 2.3 Apple vision Pro.



Σχήμα 2.4 AR Educational.

### 2.3.2 Μάρκετινγκ

Καθώς η τεχνολογία εξελίσσεται, το ίδιο συμβαίνει και με τον τρόπο που οι έμποροι προσεγγίζουν τον κοινό στόχο τους. Η επαυξημένη πραγματικότητα (AR), η εικονική πραγματικότητα (VR) και η μεικτή πραγματικότητα (MR) είναι μερικές μόνο από τις καθηλωτικές τεχνολογίες που μεταμορφώνουν τον τρόπο με τον οποίο οι μάρκες εμπλέκουν τους καταναλωτές. Γνωστές και ως διευρυμένη πραγματικότητα ή XR, αυτά τα εργαλεία αιχμής επιτρέπουν στους χρήστες να αλληλεπιδρούν με το ψηφιακό περιεχόμενο πιο ρεαλιστικά παρέχοντας έτσι, μια αξέχαστη και αποτελεσματική εμπειρία [5].

Όταν χρησιμοποιείται σωστά, η XR μπορεί να βοηθήσει τους εμπόρους να επιτύχουν διάφορους στόχους. Οι στόχοι αυτοί ενδεχομένως να αφορούν την αύξηση της αναγνωρισιμότητας της μάρκας ή ακόμη και την προώθηση των πωλήσεων. Ακολουθούν μερικά μόνο από τα οφέλη που μπορεί να προσφέρει η XR:

- **Λιανική πώληση:** Το XR δίνει στους πελάτες τη δυνατότητα να δοκιμάσουν, πριν αγοράσουν. Ο κατασκευαστής ρολογιών Rolex διαθέτει μια εφαρμογή AR που επιτρέπει στους καταναλωτές να δοκιμάσουν τα ρολόγια στον πραγματικό τους καρπό, όπως παρουσιάζεται στο ακόλουθο σχήμα 2.5, όπως επίσης και η εταιρεία επίπλων ΙΚΕΑ προσφέρει στους πελάτες τη δυνατότητα να τοποθετούν αντικείμενα επίπλων στο σπίτι τους μέσω του smartphone τους.
- **Ενίσχυση της δέσμευσης:** Το XR μπορεί να τραβήξει την προσοχή όπως κανένα άλλο μέσο. Βυθίζοντας τους χρήστες σε ένα εντελώς νέο περιβάλλον, οι μάρκες έχουν την ευκαιρία να κάνουν μια μόνιμη εντύπωση.
- **Βελτίωση της εμπειρίας του πελάτη:** Το XR μπορεί να χρησιμοποιηθεί για τη δημιουργία μιας διαδραστικής εμπειρίας πελατών που είναι τόσο ενημερωτική όσο και διασκεδαστική. Οι μάρκες μπορούν να χρησιμοποιήσουν το XR για να παραχωρήσουν στους πελάτες μια γεύση του τι μπορούν να περιμένουν από τα προϊόντα ή τις υπηρεσίες τους.
- **Αύξηση των πωλήσεων:** Σε τελική ανάλυση, όλοι οι έμποροι επιθυμούν να αυξήσουν τις πωλήσεις τους. Και οι XR μπορούν σίγουρα να βοηθήσουν σε αυτό, δίνοντας στους πελάτες μια πιο προσωπική ματιά στα προϊόντα πριν τα αγοράσουν.



Σχήμα 2.5 Rolex AR app.

### 2.3.3 Υγεία

Οι τομείς της ιατρικής, οι οποίοι ωφελούνται από την τεχνολογία XR, είναι πολλοί. Αρχικά, οι εκπαιδευόμενοι και οι φοιτητές της ιατρικής έχουν πλέον τη δυνατότητα να πραγματοποιούν χειρουργικές επεμβάσεις σε ένα περιβάλλον ελεγχόμενο, επιτηρούμενο και πολύ πιο ασφαλές για τους ίδιους, αλλά κυρίως για τους ασθενείς όπως φαίνεται στο σχήμα 2.6. Αυτό συμβαίνει, διότι η AR επιτρέπει τη βελτιωμένη αντίληψη τόσο του προβλήματος, όσο και της εικόνας του ασθενούς. Έτσι, είτε αναφερόμαστε στην εκπαίδευση πάνω σε τομείς, όπως η ανατομία, είτε αναφερόμαστε στη λειτουργία και τον εξοπλισμό ενός μαγνητικού τομογράφου ή στην πραγματοποίηση μιας πολύπλοκης χειρουργικής επέμβασης, η AR μπορεί να ενισχύσει σημαντικά τα αποτελέσματα τόσο της εκπαίδευσης, όσο και της διάγνωσης, αλλά και της θεραπείας [6].

Οι γιατροί μπορούν να χρησιμοποιούν την AR για να προετοιμαστούν για τη χειρουργική επέμβαση και να ενισχύσουν την εμπιστοσύνη με τους ασθενείς. Αυτό πραγματοποιείται, διότι η επαυξημένη πραγματικότητα μπορεί να βοηθήσει τους ασθενείς να οπτικοποιήσουν το πρόβλημα και τη θεραπεία. Ένα γεγονός που καθιστά τη διαδικασία της ενημέρωσης πληρέστερη και την συναίνεση του ασθενούς ισχυρότερη, ταχύτερη και βασισμένη στη γνώση. Επιπλέον, οι γιατροί μπορούν να χρησιμοποιήσουν εργαλεία, όπως το Microsoft HoloLens Headset για να μεταφέρουν πληροφορίες σχετικές με την θεραπεία στους ασθενείς καθιστώντας σαφή τα οφέλη, αλλά και τους κινδύνους που συνδέονται με την θεραπεία.



Σχήμα 2.6 AR στον τομέα της υγείας.

### 2.3.4 Μηχανική

Στον τομέα της μηχανικής, η χρήση της AR αποτελεί βασικό μέρος της έννοιας της βιομηχανίας 4.0 που επικεντρώνεται στις προηγμένες τεχνολογίες στα συστήματα παραγωγής και στα εργοστάσια. Η χρήση της AR έχει αξιοποιηθεί με πολλούς τρόπους από τη βιομηχανία, συμπεριλαμβανομένης της οπτικοποίησης πολύπλοκων συναρμολογήσεων, της διευκόλυνσης προγραμμάτων κατάρτισης και της ανάπτυξης προγραμμάτων συντήρησης και γραμμών παραγωγής [7].

Για παράδειγμα, το Augmented Reality Engineering Space "ARES" της εταιρείας AR Holo-Light, επιτρέπει στους μηχανικούς να απεικονίζουν, να χειρίζονται και να συνεργάζονται σε δεδομένα 3D CAD σε ένα πραγματικό περιβάλλον όπως παρουσιάζεται στο σχήμα 2.7. Το λογισμικό AR προορίζεται, ώστε να προσφέρει στους μηχανικούς ένα νέο εργαλείο για τη βελτιστοποίηση των ροών εργασίας στον σχεδιασμό, την ανάπτυξη και την παραγωγή. Τα παραπάνω παραδείγματα φανερώνουν τα πλεονεκτήματα που προσφέρει η Επαυξημένη Πραγματικότητα στον τομέα ακόμη και σήμερα, καθώς επίσης αποδεικνύουν και ποιες προκλήσεις πρέπει να αντιμετωπίσει κανείς για να επαναπροσδιορίσει τις ροές εργασίας των μηχανικών με καθηλωτικές τεχνολογίες [7].



Σχήμα 2.7 AR στον τομέα της μηχανικής.

### 2.3.5 Τουρισμός

Η τουριστική βιομηχανία είναι ένας από τους σημαντικότερους τομείς της παγκόσμιας οικονομίας. Κατά τη διάρκεια της πανδημίας, ήταν ένας από τους τομείς που υπέστη τις μεγαλύτερες επιπτώσεις, αλλά σήμερα ανακτά και πάλι την ανάπτυξη και την εξέλιξή του [8].

Καθημερινά, οι νέες τεχνολογίες προσφέρουν στους τουρίστες μια πολύ πιο ικανοποιητική και συμφέρουσα εμπειρία. Με αυτή ακριβώς την έννοια, η επαυξημένη πραγματικότητα αρχίζει να αποκτά σημασία στον τομέα του τουρισμού, μεταφέροντας την εμπειρία σε άλλο επίπεδο (σχήμα 2.8).

Η εμπειρία της επαυξημένης πραγματικότητας στον τουριστικό κλάδο ευνοήθηκε, κυρίως, από τον ψηφιακό μετασχηματισμό του τομέα και, γενικότερα, της κοινωνίας. Τα έξυπνα κινητά τηλέφωνα, η μαζική και εύκολη πρόσβαση στο διαδίκτυο και ο γεωγραφικός εντοπισμός είναι οι κύριοι παράγοντες που έχουν δώσει ώθηση στην AR στα ταξίδια.

Μερικά παραδείγματα είναι:

- Περιήγηση AR
- Ξεναγός AR
- Μουσειακή εμπειρία επαυξημένης πραγματικότητας



Σχήμα 2.8 AR στον τομέα του τουρισμού.

## 2.4 Επίλογος

Με όσα αναφέρθηκαν παραπάνω, μπορεί να ειπωθεί ότι οι συγκεκριμένες τεχνολογίες θα παίξουν σημαντικό ρόλο στο μέλλον σε διάφορους τομείς. Υπάρχουν πολλές προκλήσεις που πρέπει να αντιμετωπισθούν, αλλά το αποτέλεσμα θα είναι αρκετά θετικό. Ο κάθε τομέας παρουσιάζει διαφορετικές ανάγκες. Όσο οι τεχνολογίες όμως εξελίσσονται, θα βρίσκονται και οι λύσεις τους.



## Κεφάλαιο 3ο: Προγράμματα και εφαρμογές

### 3.1 Εισαγωγή

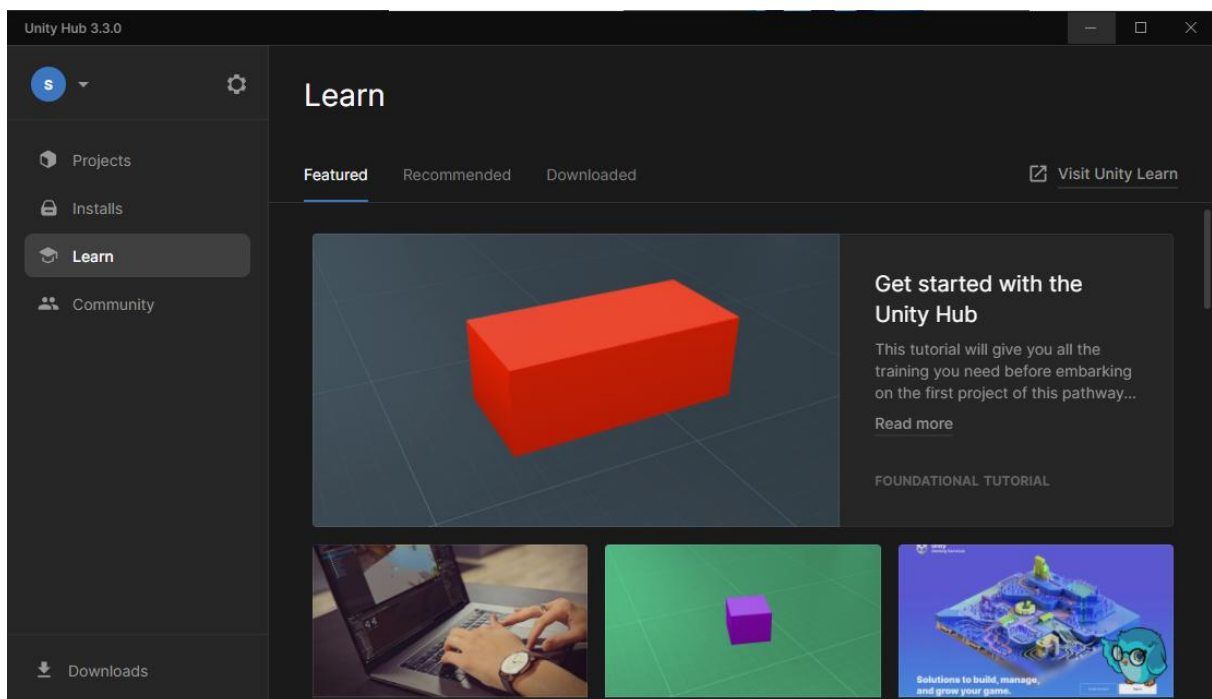
Στο κεφάλαιο αυτό θα αναλυθούν τα προγράμματα και οι εφαρμογές που χρησιμοποιήθηκαν, όπως είναι η Unity και το Vuforia, η οποία είναι μια σημαντική βιβλιοθήκη για την επίτευξη της επαυξημένης πραγματικότητας. Επίσης, θα γίνει αναφορά στον τρόπο λήψης, εγκατάστασης και χρήσης. Τέλος, θα αναλυθούν και οι ρυθμίσεις αυτών των προγραμμάτων.

### 3.2 Unity

Η Unity είναι μια μηχανή παιχνιδιών πολλαπλών πλατφορμών που αναπτύχθηκε από την Unity Technologies, η οποία ανακοινώθηκε και κυκλοφόρησε για πρώτη φορά τον Ιούνιο του 2005 στο Apple Worldwide Developers Conference ως μηχανή παιχνιδιών για Mac OS X. Έκτοτε, η μηχανή επεκτάθηκε σταδιακά, προκειμένου να υποστηρίξει διάφορες πλατφόρμες επιτραπέζιων υπολογιστών, κινητών, κονσόλας, επαυξημένης καθώς επίσης και εικονικής πραγματικότητας. Είναι ιδιαίτερα δημοφιλής για την ανάπτυξη παιχνιδιών για κινητά iOS και Android. Θεωρείται εύκολη στη χρήση για αρχάριους προγραμματιστές και είναι δημοφιλής για την ανάπτυξη ανεξάρτητων παιχνιδιών [10].

#### 3.2.1 Εγκατάσταση Unity Hub και Unity

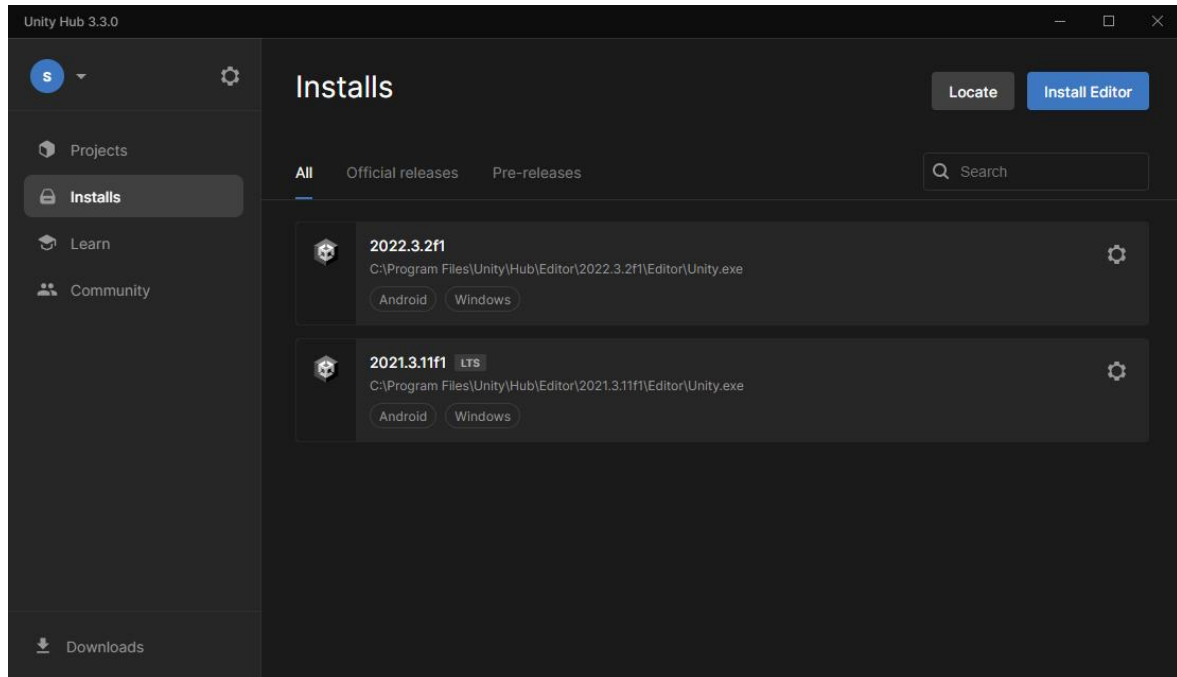
Αρχικά, για να δημιουργηθεί κάποιο project πρέπει να γίνει η εγκατάσταση του Unity Hub από την ιστοσελίδα της Unity. Το Unity Hub είναι ένα εργαλείο, το οποίο εξυπηρετεί στην οργάνωση και τη διαχείριση των έργων (project) του χρήστη στο Unity, καθώς και τις εγκαταστάσεις των εκδόσεων [10]. Η εγκατάσταση του Unity Hub είναι πολύ απλή και εύκολη. Αφού επιλεγεί ο φάκελος που θα γίνει η εγκατάσταση και θα “τρέξει” η εφαρμογή, θα εμφανιστεί το παρακάτω παράθυρο που φαίνεται στο σχήμα 3.1 του Unity Hub:



Σχήμα 3.1 Unity Hub.

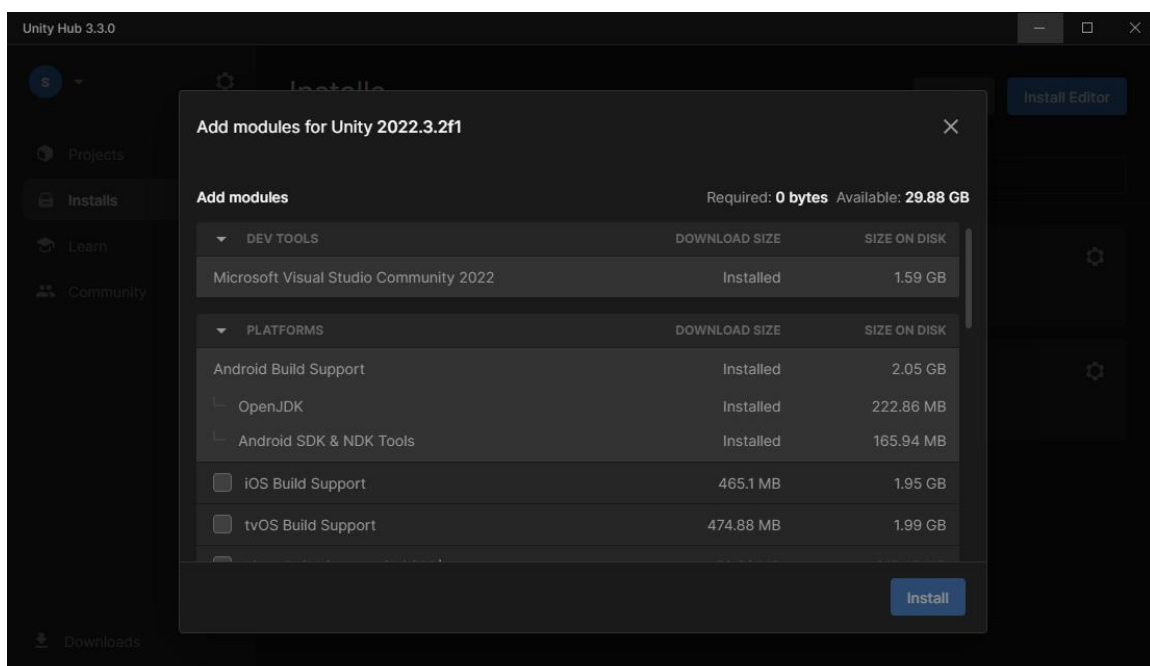
## Κεφάλαιο 3

Το επόμενο βήμα είναι να γίνει install πηγαίνοντας στην αριστερή στήλη και επιλέγοντας την επιθυμητή έκδοση. Συνήθως, επιλέγονται εκδόσεις που είναι LTS, δηλαδή long time support για τους ευνότερους λόγους. Στην παρούσα πτυχιακή εργασία χρησιμοποιείται η έκδοση 2021.3.11f1, όπως φαίνεται στο σχήμα 3.2.



Σχήμα 3.2 Unity Hub install έκδοσης.

Μαζί με την έκδοση, που θα γίνει install, πρέπει να επιλεγθούν και κάποιες βιβλιοθήκες, οι οποίες θα συμπεριληφθούν μαζί. Οι βιβλιοθήκες, που θα γίνουν install, επιλέγονται με βάση τις ανάγκες του παιχνιδιού. Αφού επιλεγθούν αυτά που χρειάζονται, πρέπει να πατηθεί το κουμπί “Install” κάτω δεξιά που φαίνεται αναφέρεται το σχήμα 3.3, και με αυτήν την ενέργεια ολοκληρώνεται η εγκατάσταση του Editor / έκδοση.

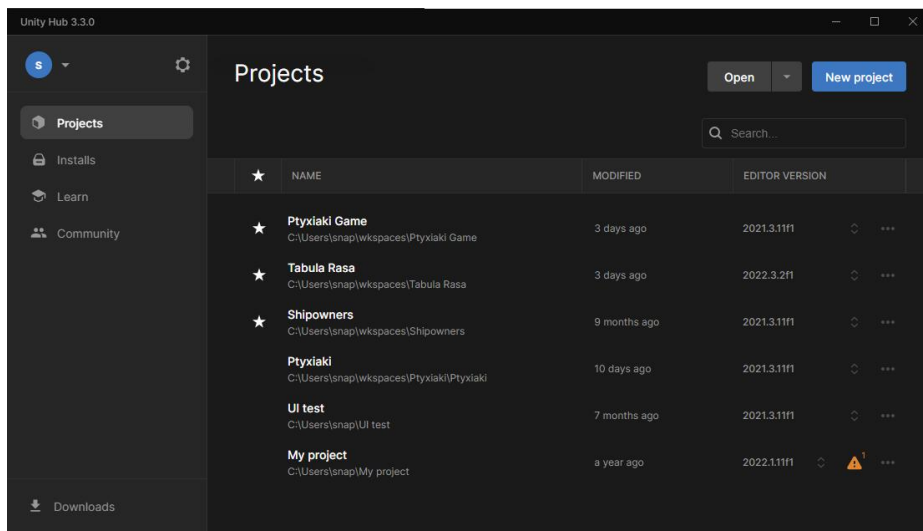


Σχήμα 3.3 Unity Hub install libraries.

Σε αυτήν την πτυχιακή εργασία από external libraries έχουν επιλεγεί και χρησιμοποιούνται από την κατηγορία DEV TOOLS, το Microsoft Visual Studio Community 2019, ενώ από την κατηγορία PLATFORMS, το Android Build Support, το OpenJDK και το android SDK & NDK Tools.

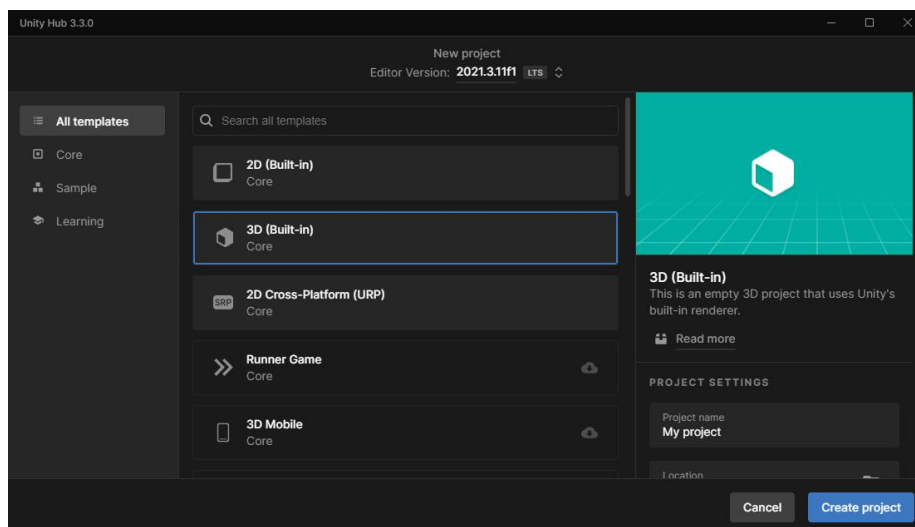
### 3.2.2 Δημιουργία νέου Project

Το επόμενο βήμα είναι να δημιουργηθεί το καινούργιο project. Πηγαίνοντας στην αριστερή στήλη “Projects” και επιλέγοντας το κουμπί “New Project” που αναφέρεται και το σχήμα 3.4, θα ανοίξει το παρακάτω παράθυρο. Όπως φαίνεται και στο σχήμα 3.5, θα πρέπει να επιλεγθούν κάποιες βασικές ρυθμίσεις για το καινούργιο project.



Σχήμα 3.4 Δημιουργία νέου project.

Σε αυτό το παράθυρο, όπως αναφέρθηκε και παραπάνω, πρέπει να επιλεγθούν κάποιες βασικές ρυθμίσεις του project. Πρώτα, θα επιλεγθεί η επιθυμητή έκδοση και αν το παιχνίδι προορίζεται για 2D ή 3D. Στην παρούσα περίπτωση έχει επιλεγθεί 3D. Τέλος, πρέπει να δοθεί ένα όνομα για το project και να επιλεγθεί το location που θα αποθηκευτεί το έργο. Κάποιες από αυτές τις ρυθμίσεις δεν μπορούν να αλλάξουν μετέπειτα.



Σχήμα 3.5 Ρυθμίσεις νέου project.

## 3.3 Vuforia Engine

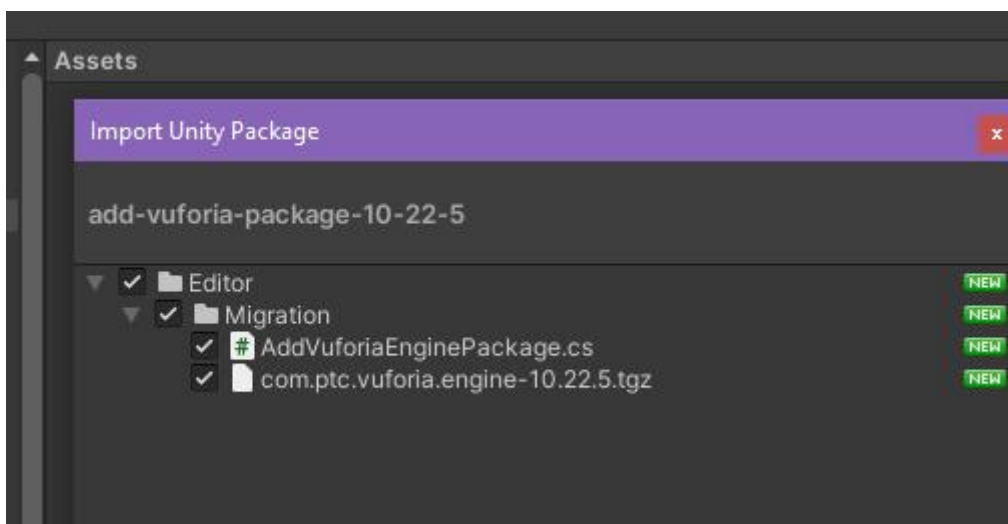
### 3.3.1 Εισαγωγή

Το Vuforia είναι ένα kit ανάπτυξης λογισμικού επαυξημένης πραγματικότητας (SDK) για κινητές συσκευές που επιτρέπει τη δημιουργία εφαρμογών επαυξημένης πραγματικότητας. Χρησιμοποιεί τεχνολογία όρασης υπολογιστών για την αναγνώριση και παρακολούθηση επίπεδων εικόνων και τρισδιάστατων αντικειμένων σε πραγματικό χρόνο. Αυτή η δυνατότητα καταγραφής εικόνων επιτρέπει στους προγραμματιστές να τοποθετούν και να προσανατολίζουν εικονικά αντικείμενα, όπως τρισδιάστατα μοντέλα και άλλα μέσα, σε σχέση με αντικείμενα του πραγματικού κόσμου, όταν αυτά προβάλλονται μέσω της κάμερας μιας κινητής συσκευής [11].

### 3.3.2 Εγκατάσταση Vuforia Engine package

Ένας από τους τρόπους που μπορεί να λειτουργήσει το vuforia engine στο project είναι να γίνει import το vuforia package με mime type .unitypackage, αφού πρώτα γίνει εγκατάσταση από την ιστοσελίδα τους. Για να καταστεί αυτό εφικτό, πρέπει να δημιουργηθεί ένας λογαριασμός και στη συνέχεια πρέπει να γίνει το download [11].

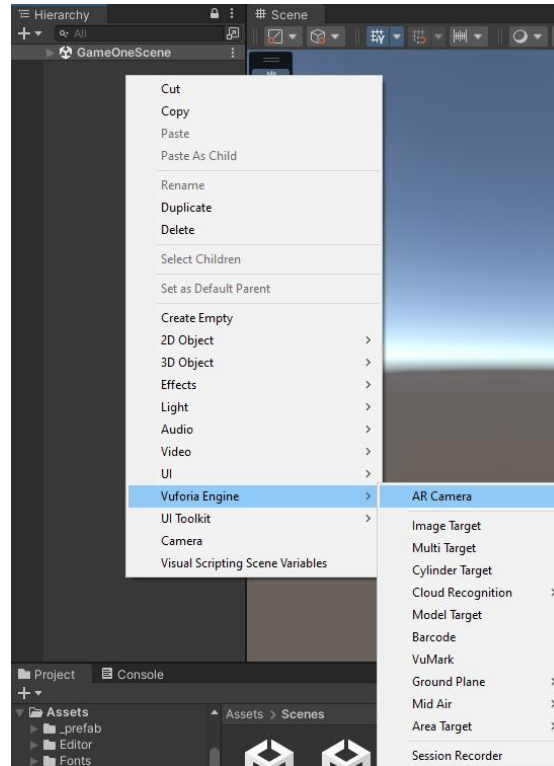
Αφού εκτελεστεί η διαδικασία αυτή, το μόνο που απομένει είναι να γίνει import (εισαγωγή) στην unity. Αυτό μπορεί να γίνει εύκολα με drag and drop μέσα στο φάκελο με τα assets που βρίσκεται στο project. Τέλος, με την ολοκλήρωση και αυτού του βήματος θα εμφανιστεί η παρακάτω εικόνα του σχήματος 3.6., κατά την οποία πρέπει να πατηθεί το import και το vuforia package θα εγκατασταθεί με επιτυχία.



Σχήμα 3.6 Import Vuforia engine package.

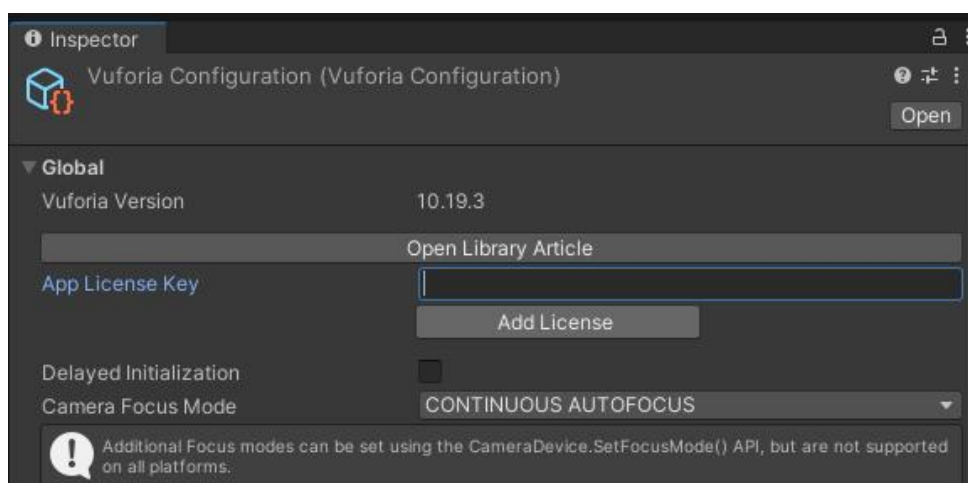
### 3.3.3 Βασικές ρυθμίσεις Vuforia Engine.

Το πρώτο βήμα για να υπάρχει ένα σωστό configuration είναι, αφού έχει ανοιχτεί το project, αριστερά στην στήλη hierarchy κάνοντας δεξί κλικ και σέρνοντας το ποντίκι πάνω στο vuforia engine, να δημιουργηθεί μια AR Camera, όπως φαίνεται στην παρακάτω εικόνα 3.7.



Σχήμα 3.7 Δημιουργία τις AR Camera.

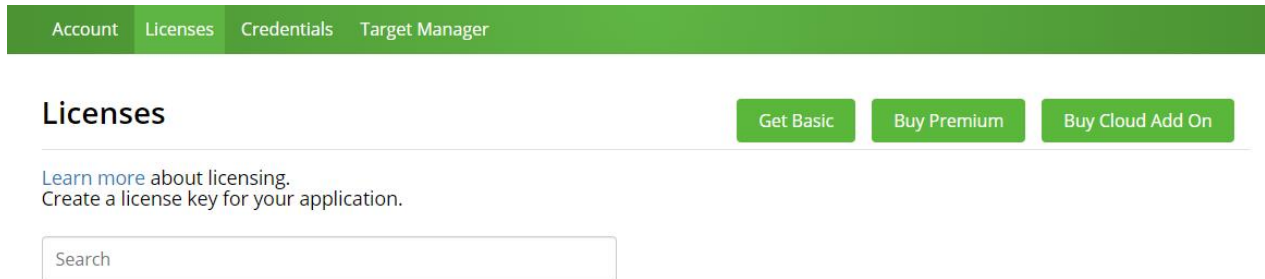
Αφού εκτελεστεί αυτή η διαδικασία, στο hierarchy πρέπει να εμφανίζεται ένα καινούργιο Game Object με όνομα ARCamera. Έπειτα, κλικάροντας πάνω στην κάμερα, δεξιά στον inspector ανοίγουν οι ρυθμίσεις της κάμερας. Στο σημείο αυτό πρέπει να επιλεγθεί το “Open Vuforia Engine configuration”. Αφού γίνει αυτό, θα πρέπει να εμφανίζεται το παρακάτω παράθυρο, όπως φαίνεται στο σχήμα 3.8.



Σχήμα 3.8 Vuforia Engine configuration.

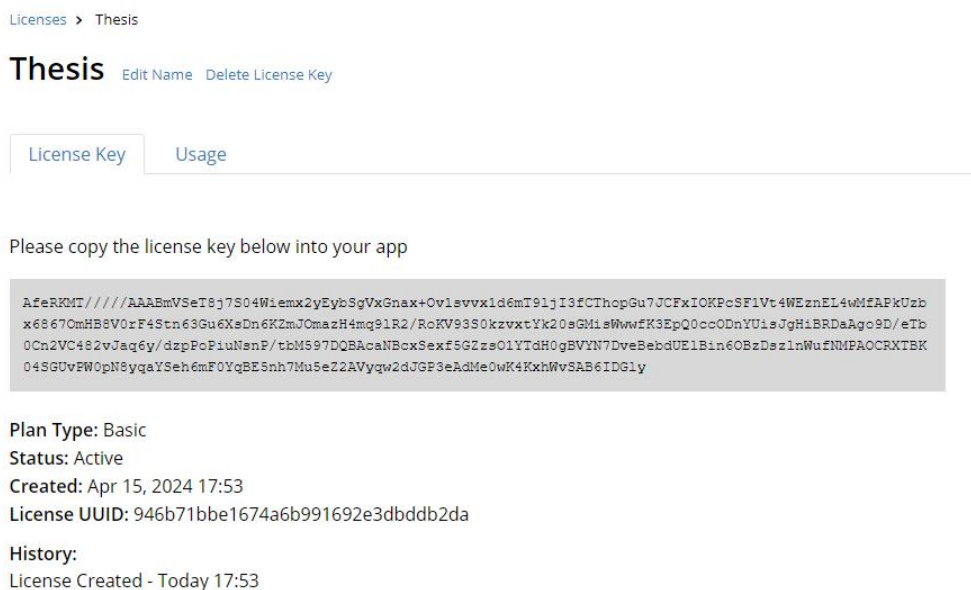
## Κεφάλαιο 3

Στο νέο παράθυρο που εμφανίστηκε, πρέπει να εισαχθεί ένα license key για το configuration της vuforia. Χωρίς αυτό δεν είναι δυνατόν να προχωρήσει η διαδικασία, διότι με αυτόν το τρόπο συνδέεται με το account που έχει δημιουργηθεί προ ολίγου και έτσι, έχει πρόσβαση στην βάση που φαίνεται παρακάτω. Το license key θα πρέπει να παρθεί από την ιστοσελίδα της vuforia. Αρχικά, πρέπει γίνει σύνδεση και έπειτα η επιλογή των licenses από την πάνω πράσινη μπάρα, όπως φαίνεται στο σχήμα 3.9.



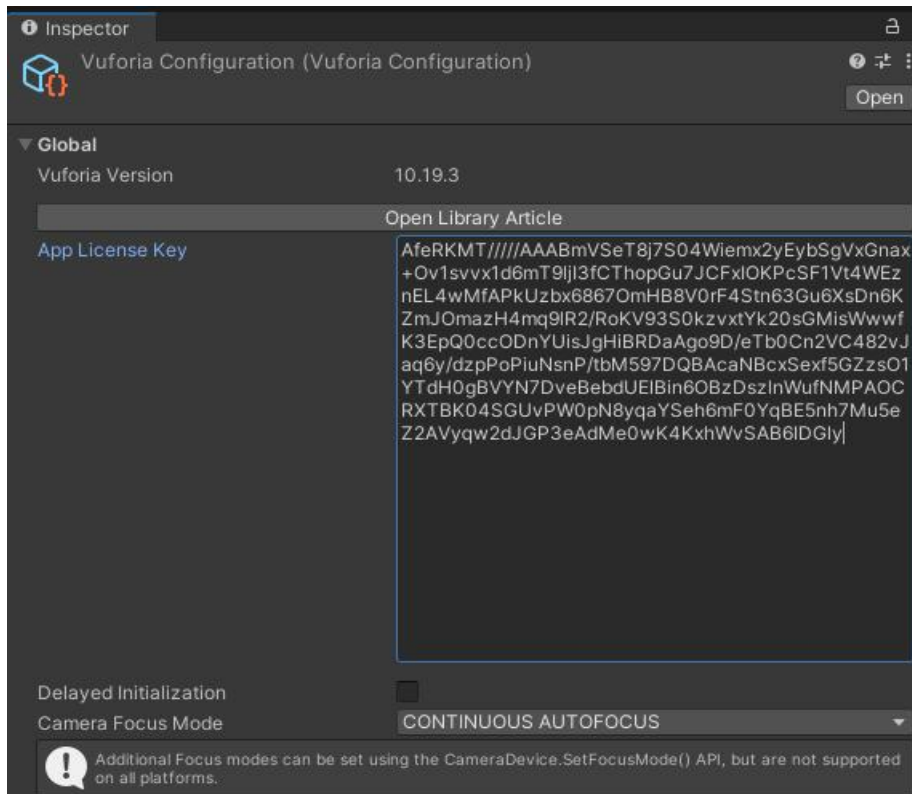
Σχήμα 3.9 Έκδοση License Key.

Έπειτα, πρέπει να γίνει κλικ στο κουμπί “Get Basic”, το οποίο θα εμφανίσει ένα καινούργιο παράθυρο. Εκεί πρέπει να εισαχθεί ένα όνομα για το license name και, αφού γίνει αυτό, πρέπει να πατηθεί το confirm κουμπί. Τέλος, γίνεται επιστροφή στην προηγούμενη σελίδα, αλλά πλέον φαίνεται να έχει δημιουργηθεί ένα καινούργιο αρχείο με το όνομα που δόθηκε, το οποίο αφού ανοιχτεί, εμφανίζεται μέσα ότι έχει δημιουργηθεί το license key που χρειάζεται. Το key παρουσιάζεται στο σχήμα 3.10.



Σχήμα 3.10 Σελίδα License Key.

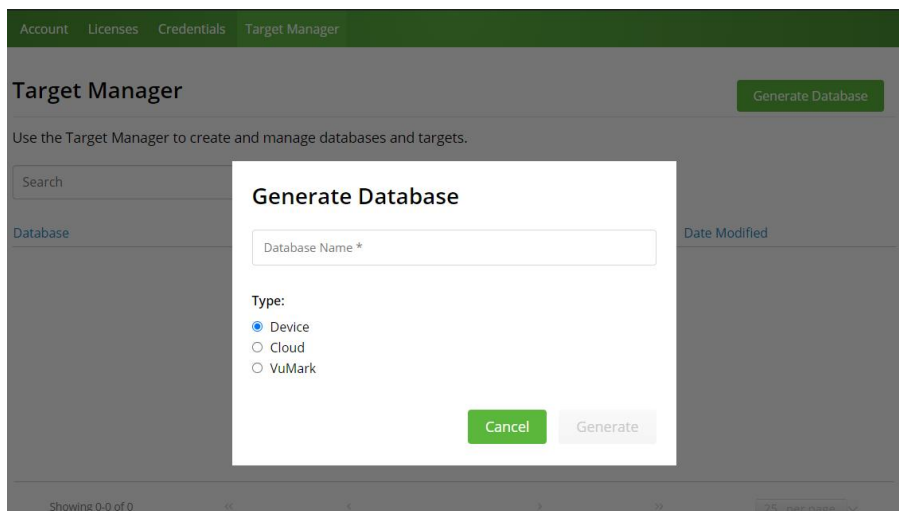
Τέλος, πρέπει να γίνει αντιγραφή το key. Πηγαίνοντας πίσω στην unity χρειάζεται να γίνει η επικόλληση στο configuration menu της AR Camera, που δημιουργήθηκε στα προηγούμενα βήματα, όπως φαίνεται στο σχήμα 3.11.



Σχήμα 3.11 Επικόλληση License Key στο configuration menu..

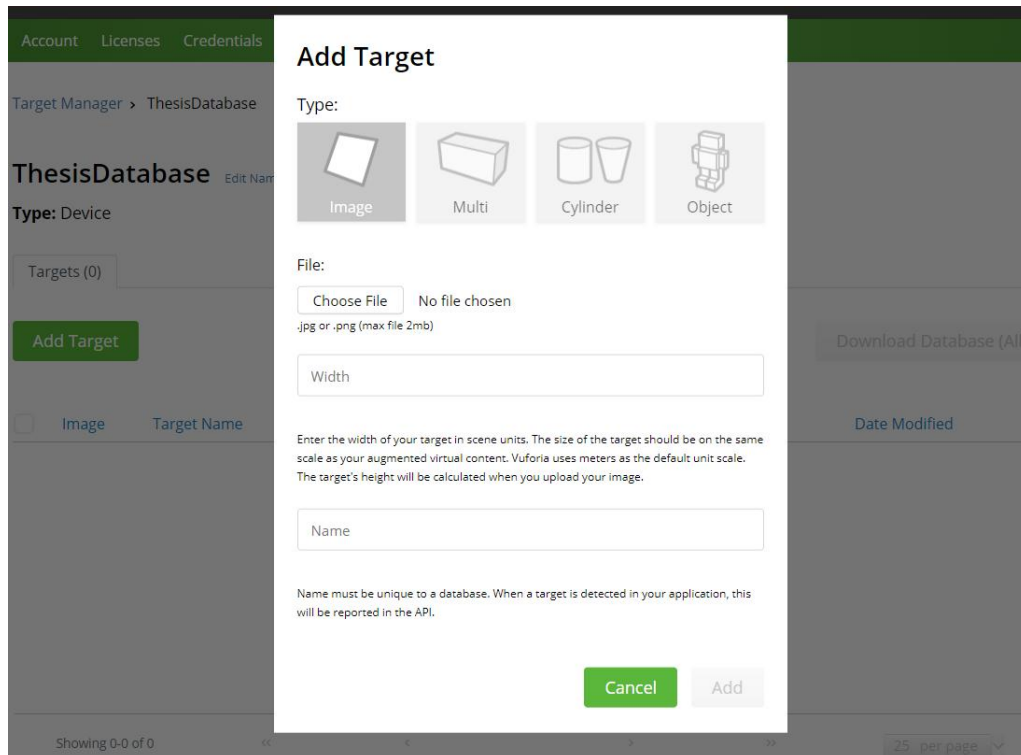
### 3.3.4 Δημιουργία Βάσης

Τελευταίο, αλλά εξίσου σημαντικό βήμα που πρέπει να εκτελεστεί, για να υπάρχει ένα σωστό set up του vuforia είναι να δημιουργηθεί η βάση δεδομένων. Οπότε, πηγαίνοντας στο Target Manager από την πάνω πράσινη μπάρα και κλικάροντας στο Generate Database εμφανίζεται το παρακάτω παράθυρο, όπου πρέπει να πραγματοποιηθεί η επιλογή “Device”, όπως γίνεται φανερό στο σχήμα 3.12.



Σχήμα 3.12 Δημιουργία βάσης δεδομένων

Το επόμενο βήμα είναι να εισαχθεί ένα όνομα για την βάση και η τιμή για το width πρέπει να είναι 1. Αφού πατηθεί το κουμπί Add κλείνει το παράθυρο, που φαίνεται στο σχήμα 3.13, και επιστρέφει στην αρχική οθόνη ενώ πλέον, εμφανίζεται με δημιουργημένο το Database με το όνομα που δόθηκε.



Σχήμα 3.13 Επιλογή Add Target.

Το image target στην ουσία αντιπροσωπεύει εικόνες που μπορεί να ανιχνεύσει και να παρακολουθήσει το Vuforia Engine. Το Vuforia Engine ανιχνεύει και παρακολουθεί την εικόνα συγκρίνοντας τα εξαγόμενα φυσικά χαρακτηριστικά από την εικόνα της κάμερας με μια γνωστή βάση δεδομένων πόρων στόχων. Μόλις εντοπιστεί ο στόχος εικόνας, η μηχανή Vuforia Engine θα παρακολουθήσει την εικόνα [12].

Οι συνήθεις χρήσεις των Image Targets περιλαμβάνουν την αναγνώριση και την επαύξηση των έντυπων μέσων και των συσκευασιών προϊόντων για εκστρατείες μάρκετινγκ, παιχνίδια και την οπτικοποίηση προϊόντων στο περιβάλλον όπου το προϊόν προοριζόταν να χρησιμοποιηθεί. Το Image target μπορεί να χρησιμοποιηθεί σε οποιαδήποτε επίπεδη εικόνα που παρέχει επαρκή λεπτομέρεια για να ανιχνευθεί από τη μηχανή Vuforia Engine [12].

Στην παρούσα πτυχιακή εργασία θα χρησιμοποιηθεί το image target. Ο λόγος συνίσταται στο γεγονός πως η χρήση του είναι βολική, προκειμένου να σκαναριστούν οι εικόνες από τα σχολικά βιβλία.


Έπειτα από την διαδικασία αυτή, η ιστοσελίδα ξεκινάει να κάνει ένα processing για την εικόνα που δόθηκε και στη συνέχεια (Σχήματα 3.14 και 3.15), θα βγάλει ένα σχετικό rating για το πόσο καλά είναι αναγνωρίσιμη η εικόνα.

**ThesisDatabase** [Edit Name](#)

**Type:** Device

Targets (1)

[Add Target](#) [Download Database \(All\)](#)

<input type="checkbox"/>	Image	Target Name	Type	Rating <sup>①</sup>	Status <sup>▼</sup>	Date Modified
<input type="checkbox"/>		image1	Image	★★★★☆	Active	Apr 15, 2024

Σχήμα 3.14 Εμφάνιση του image database.

**image1**

[Edit Name](#) [Remove](#)



**Type:** Image

**Status:** Active

**Target ID:** 791842d6f482427a82ce8a631bd60a02

**Augmentable:** ★★★★★

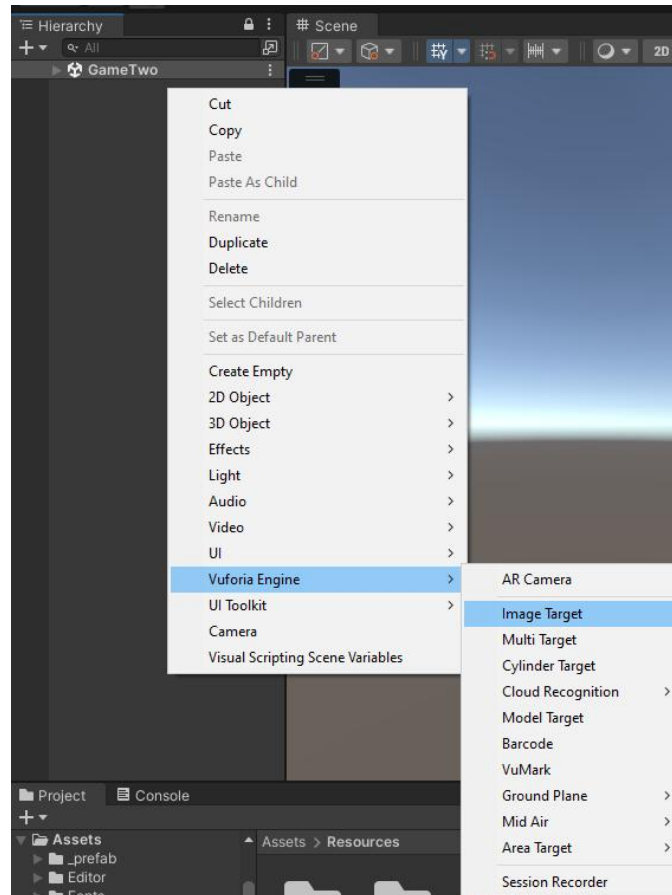
**Added:** Apr 15, 2024

**Modified:** Apr 15, 2024

[Update Target](#) [Show Features](#)

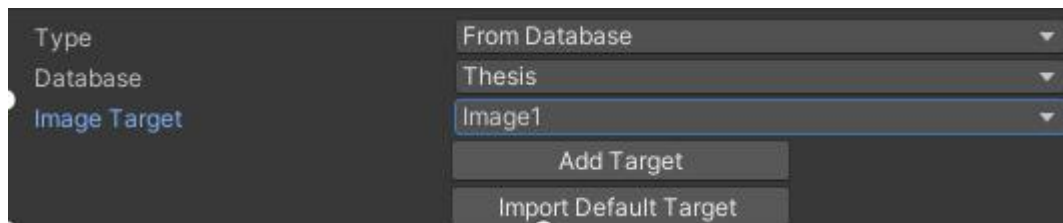
Σχήμα 3.15 Rating τις αναγνωρίσιμης εικόνας.

Τέλος, εφόσον έχει δημιουργηθεί το image target, πρέπει να πατηθεί το κουμπί "Download Database" και πραγματοποιώντας την επιλογή "Unity Editor", κατεβαίνει ένα πακέτο με το όνομα της βάσης που δόθηκε με την κατάληξη .packagemanager, το οποίο με drag and drop πρέπει να συρθεί και αυτό στο project. Για να γίνει χρήση του image target εντός Unity, πρέπει στο "Hierarchy" να δημιουργηθεί ένα "image target" κάνοντας δεξί κλικ, επιλέγοντας Vuforia Engine και τέλος, "Image Target", όπως γίνεται ορατό στο κάτωθι σχήμα 3.16.



Σχήμα 3.16 Δημιουργία Image Target.

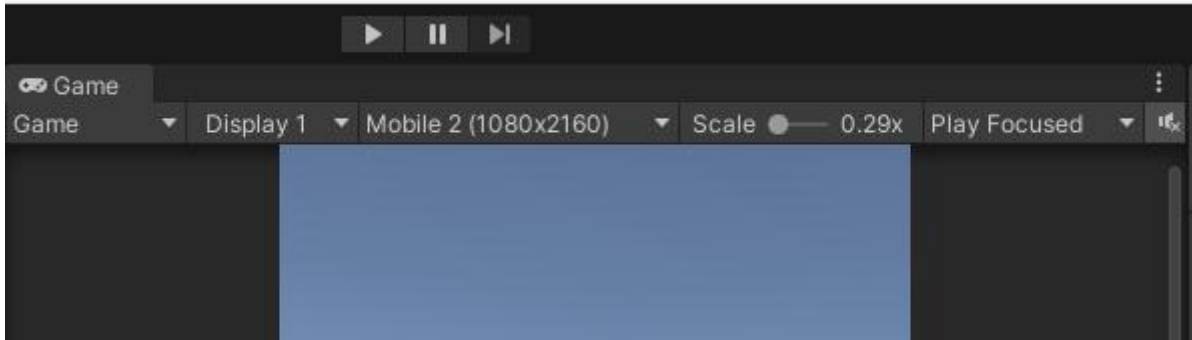
Αφού επιλεγθεί το Image Target στον Inspector, πρέπει να επιλεγθεί από δεξιά το component "Image Target Behaviour" και επιλέγοντας το type να είναι "From Database", στο Database επιλέγοντας το όνομα που δόθηκε και τέλος στο image target επιλέγεται η εικόνα με το όνομα που έγινε Add, όπως φαίνεται και στο σχήμα 3.17.



Σχήμα 3.17 Component Image Target Behaviour.

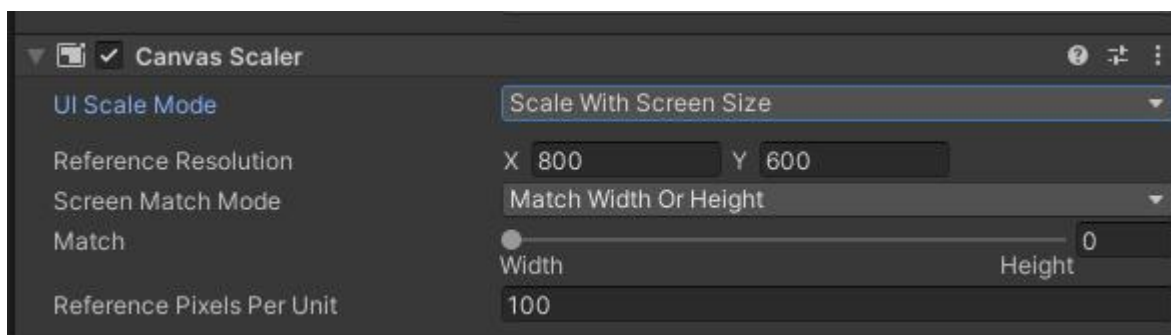
### 3.4 Unity Set Up

Πριν ξεκινήσει να γίνεται κάποια υλοποίηση στην unity, προτείνεται να αλλαχθούν κάποιες ρυθμίσεις που θα βοηθήσουν στην πορεία του project. Αρχικά, είναι ωφέλιμο στο game viewport να ορισθούν οι διαστάσεις που θα εμφανίζονται στη οθόνη. Στην παρούσα πτυχιακή εργασία χρησιμοποιούνται διαστάσεις 1080 width x 2160 height, όπως φαίνεται στο 3.18.



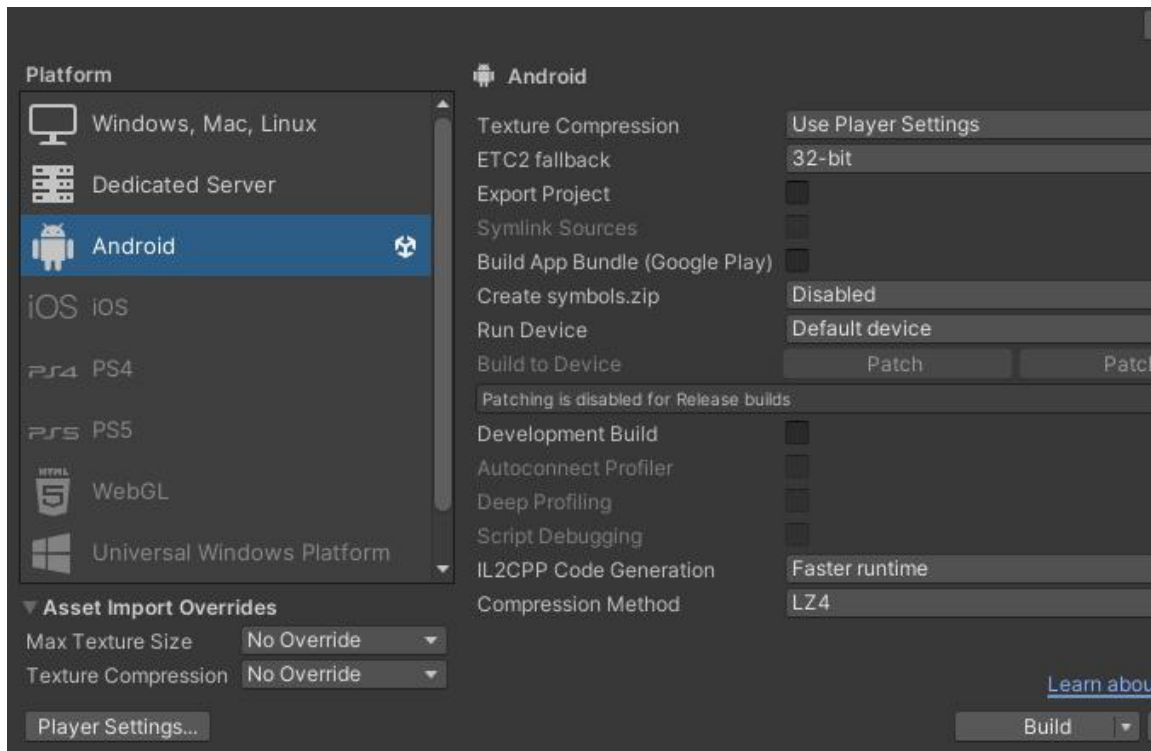
Σχήμα 3.18 Επιλογή διαστάσεων οθόνης.

Έπειτα, πρέπει να ορισθούν κάποια properties στον canva που εκεί μέσα θα κρατηθεί ό,τι χρειαστεί από UI. Θα γίνει κλικ στον Canva στην αριστερή στήλη “hierarchy” και θα ανοίξει το παράθυρο δεξιά του inspector με τα settings. Στο component “Canvas Scaler” και κλικάροντας στο “UI Scale Mode” και θα πρέπει να πραγματοποιηθεί η επιλογή Scale with screen size. Τέλος, αφού γίνει η παραπάνω διαδικασία, θα πρέπει να εμφανιστεί ακριβώς πιο κάτω το “Reference Resolution” , σημείο στο οποίο θα πρέπει να εισαχθούν οι διαστάσεις, οι οποίες έχουν επιλεγεί πιο πάνω. Άρα, στο X θα εισαχθεί 1080 και στο Y θα εισαχθεί 2160. Με αυτόν τον τρόπο πλέον, ό,τι εμφανίζεται στο game viewport θα εμφανίζεται και το αντίστοιχο στην οθόνη του κινητού. Αφού εκτελεστεί η παραπάνω διαδικασία, το αποτέλεσμα θα πρέπει να συμφωνεί με την κάτωθι φωτογραφία, όπως φαίνεται στο σχήμα 3.19.



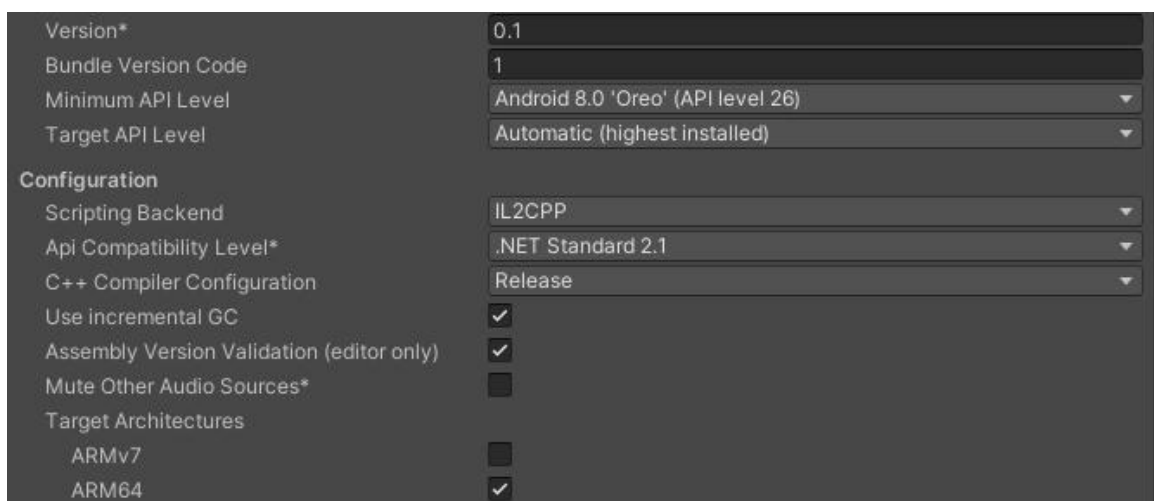
Σχήμα 3.19 Εισαγωγή διαστάσεων στον Canva.

Πρέπει, επίσης, να πραγματοποιηθούν ορισμένα σημαντικά βήματα στις ρυθμίσεις της unity, ώστε να καταφέρει να λειτουργήσει το AR. Πρώτα, στα build settings της unity και στην στήλη platform πρέπει να επιλεγθεί το Android, και στη συνέχεια πρέπει να πατηθεί το κουμπί switch platform, όπως ακριβώς υποδεικνύεται και στο σχήμα 3.20.



Σχήμα 3.20 Switch platform σε android.

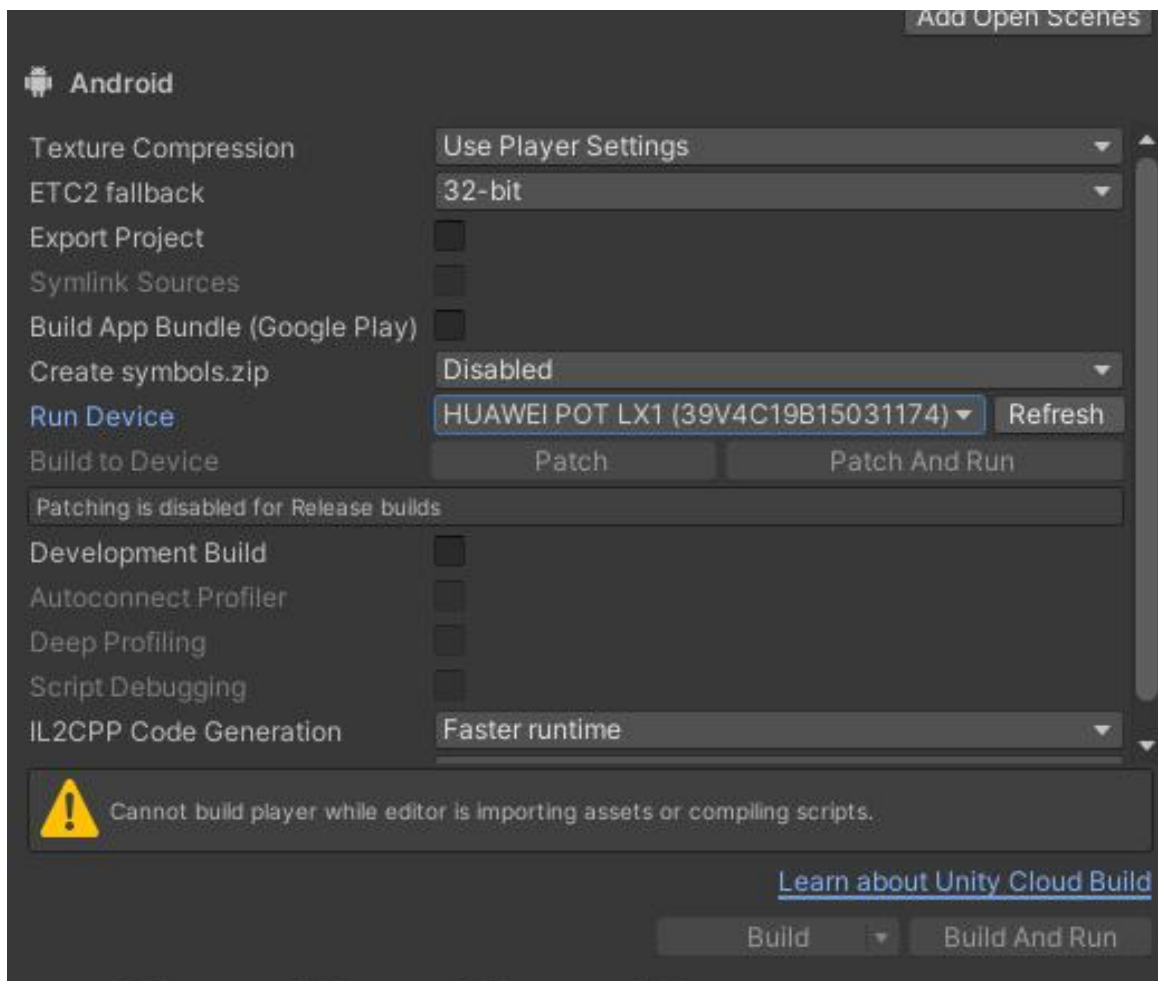
Ύστερα, πηγαίνοντας στα player settings από την αριστερή στήλη επιλέγεται το setting με όνομα “Target API Level” . Εκεί πρέπει να επιλεγθεί “Android 8.0 ‘oreo’(API Level 26)”. Αυτό συνιστά ένα σημαντικό βήμα, επειδή στο vuforia engine είναι το minimum λογισμικού που υποστηρίζεται. Τέλος, πρέπει να βρεθεί το “Target Architectures” και να μπει τσεκ την αρχιτεκτονική ARM64 και να αφαιρεθεί το τσεκ από το ARMv7. Αν εκτελεστούν σωστά τα παραπάνω βήματα, τα setting πρέπει να είναι ίδια, όπως ακριβώς στο σχήμα 3.21.



Σχήμα 3.21 Ρυθμίσεις αρχιτεκτονικής και λογισμικού.

### 3.5 Build ρυθμίσεις

Προκειμένου να γίνει build το παιχνίδι στο κινητό, πρέπει να “σεταριστούν” κάποια συγκεκριμένα settings. Πρώτα, πρέπει να βρεθεί το Run Device και εκεί να επιλεγεί η συσκευή. Σημαντικό είναι να γίνει αντιληπτό ότι φαίνεται η συσκευή στις επιλογές. Το κινητό θα πρέπει να είναι συνδεδεμένο στον υπολογιστή / laptop με usb. Επίσης, οι ρυθμίσεις του κινητού θα πρέπει να είναι σε debugging mode. Η επιλογή αυτή θα εμφανιστεί μόλις το κινητό συνδεθεί στον υπολογιστή. Αφού πατηθεί το κουμπί build and run, θα πρέπει να επιλεγεί ένας φάκελος που θα αποθηκεύεται το apk αρχείο στον υπολογιστή. Τέλος, επιλέγεται ο φάκελος και το build ξεκινάει. Όλα τα προαναφερθέντα βήματα παρουσιάζονται στο κάτωθι σχήμα 3.22.



Σχήμα 3.22 Build του παιχνιδιού στο κινητό μας.

### 3.6 Photoshop

Το Photoshop είναι ένα λογισμικό επεξεργασίας φωτογραφιών και σχεδιασμού γραφικών που επιτρέπει στους χρήστες να δημιουργούν, να επεξεργάζονται και να χειρίζονται διάφορα γραφικά καθώς και ψηφιακή τέχνη. Επιτρέπει, επίσης, τη δημιουργία και την επεξεργασία εικόνων ράστερ με πολλαπλά επίπεδα και την εισαγωγή των εικόνων σε διάφορες μορφές αρχείων. Το Photoshop αναπτύσσεται από την Adobe Systems τόσο για Windows όσο και για MacOS [13]. Το logo του Photoshop εμφανίζεται στο σχήμα 3.23.



Σχήμα 3.23 Photoshop logo.

Το Photoshop στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε για την σχεδίαση 2D γραφικών, όπως είναι οι εικόνες background, σχήματα και κουμπιά. Στο photoshop δεν χρειάζεται να οριστούν κάποια settings, αρκεί οι διαστάσεις που γίνονται export τα png αρχεία να είναι ίδιες με αυτές που χρειάζονται μέσα στην unity, ώστε να μην αλλοιώνεται η ανάλυση από αυτό που έχει ήδη φτιαχτεί. Ένα παράδειγμα από ένα button, το οποίο δημιουργήθηκε στο photoshop, φαίνεται στο κάτωθι σχήμα 3.24.

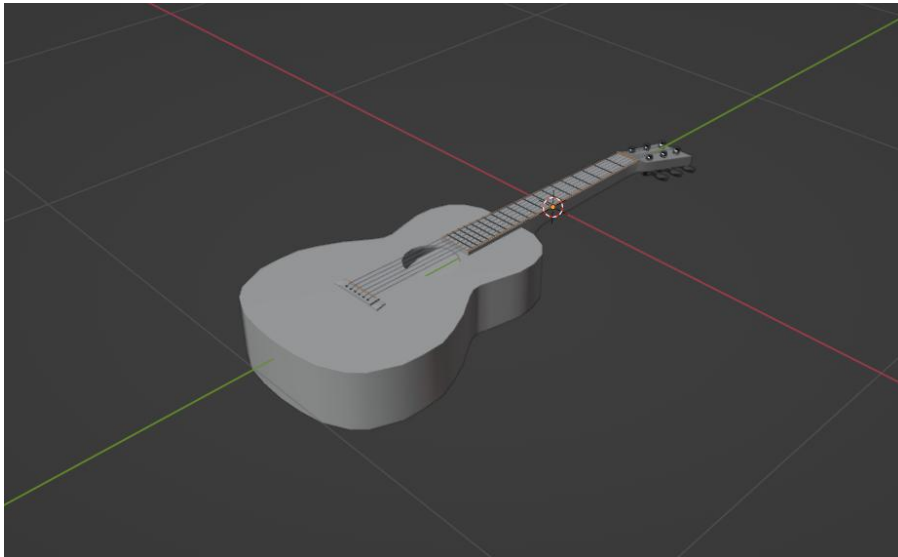


Σχήμα 3.24 Button που δημιουργήθηκε στο photoshop.

### 3.7 Blender

Το Blender είναι ένα free και open source λογισμικό που χρησιμοποιείται από πολλούς καλλιτέχνες γραφικών και VFX για να μεταφέρουν τη φαντασία τους στον οπτικά ρεαλιστικό κόσμο στις οθόνες σε 2D και 3D γραφικά υπολογιστών. Το λογισμικό αυτό αναπτύσσεται και συντηρείται από το Blender Foundation από το 1998. Πολλοί επαγγελματίες που εργάζονται ως modelers, texturing, skinning, skeleton, rendering, rigging, προσομοίωση, motion graphics, video game [14].

Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε το blender για να δημιουργηθεί οποιοδήποτε τρισδιάστατο αντικείμενο χρειαζόταν για τις σκηνές. Επίσης, στο blender πάλι δεν χρειάζεται να έχει κάποια συγκεκριμένη ρύθμιση πάντα σταθερή, αρκεί και πάλι οι διαστάσεις του αντικειμένου, που θα γίνει export με κατάληξη .blend να είναι σύμφωνα με αυτά που χρειάζεται το project. Ένα παράδειγμα για μοντέλο που δημιουργήθηκε στο blender αντικατοπτρίζεται στο σχήμα 3.25. Πρέπει να σημειωθεί πως τα textures εισήλθισαν στην συνέχεια μέσα στην unity.



Σχήμα 3.25 Δημιουργία μοντέλου στο blender.

### 3.8 Επίλογος

Εκτελώντας σωστά όλα τα παραπάνω βήματα και πραγματοποιώντας τη σωστή εγκατάσταση των βιβλιοθηκών που θα χρειαστούν, πλέον όλα είναι έτοιμα προκειμένου να ξεκινήσει η δημιουργία του παιχνιδιού.



## Κεφάλαιο 4ο: Αρχική οθόνη

### 4.1 Εισαγωγή

Αυτό το κεφάλαιο αναφέρεται στην αρχική οθόνη του παιχνιδιού. Συνήθως σε κάθε παιχνίδι υπάρχει μία αρχική οθόνη. Κάτι τέτοιο κρίνεται ως απαραίτητο, ώστε ο χρήστης να μπορέσει να πλοηγηθεί μέσα στη εφαρμογή. Στην παρούσα πτυχιακή εργασία υπάρχουν δύο διαφορετικά μενού. Το ένα μενού θα εμφανιστεί μπροστά, μόλις ανοίξει το παιχνίδι, το οποίο θα περιλαμβάνει τρία κουμπιά, και ένα δεύτερο μενού αφορά την επιλογή του παιχνιδιού.

### 4.2 Πρώτο μενού:

Αρχικά, για την δημιουργία των UI θα πρέπει να δημιουργηθεί ένας canvas από την αριστερή στήλη του inspector, στον οποίο θα εισχωρήσουν όλα τα elements. Στην παρούσα αρχική οθόνη έχουν εισαχθεί images και buttons. Στο πρώτο μενού της αρχικής οθόνης εμφανίζονται τρία κουμπιά: Εκκίνηση, Σχετικά και Έξοδος. Το κουμπί “Εκκίνηση” πηγαίνει στο δεύτερο μενού, που είναι η επιλογή του παιχνιδιού. Το κουμπί “Σχετικά” αναφέρει ορισμένες πληροφορίες για τον δημιουργό της εφαρμογής (στην παρούσα περίπτωση εμένα), καθώς επίσης και ορισμένες πληροφορίες σχετικά με την πτυχιακή εργασία. Τέλος, απομένει το κουμπί “Έξοδος”, το οποίο τη στιγμή που πατηθεί, τερματίζει την εφαρμογή. Το πρώτο μενού φαίνεται στο σχήμα 4.1.



Σχήμα 4.1 Αρχικό μενού.

### 4.3 Scripts

Για να είναι εφικτή η πλοήγηση μέσα στην εφαρμογή, απαιτείται να δημιουργηθεί ο κώδικας που θα κάνει την λειτουργία αυτή. Οπότε, σε πρώτο στάδιο, πρέπει να δημιουργηθεί ένα script πηγαίνοντας στην καρτέλα με τα “assets”. Εκεί θα δημιουργηθεί ένας φάκελος με όνομα “scripts”. Αφού γίνει αυτό, πρέπει να γίνει είσοδος στον φάκελο και κάνοντας δεξί κλικ γίνεται η επιλογή “New script → Create και C# script”. Έπειτα, ανοίγοντας το “script” με το “visual studio”, πρέπει να γραφτεί το “function” που χρειάζεται για το κουμπί “Εκκίνηση”. Η λειτουργικότητα που θα περιλαμβάνει το συγκεκριμένο “script” είναι να ανοιγοκλείνει αυτά τα δύο αρχικά μενού. Ένα “function” δηλαδή, για να πηγαίνει στη επιλογή του παιχνιδιού, ένα “function” για το κουμπί “Σχετικά” για να δείχνει τις πληροφορίες και ένα “function” για το κουμπί “Εξοδος”, το οποίο θα κλείνει την εφαρμογή. Τέλος, θα χρειαστεί ένα κουμπί για να επιστρέφει τον χρήστη στο αρχικό μενού, αν είναι επιθυμητό. Το “Script” θα πρέπει να μοιάζει όπως στο σχήμα 4.2.

```

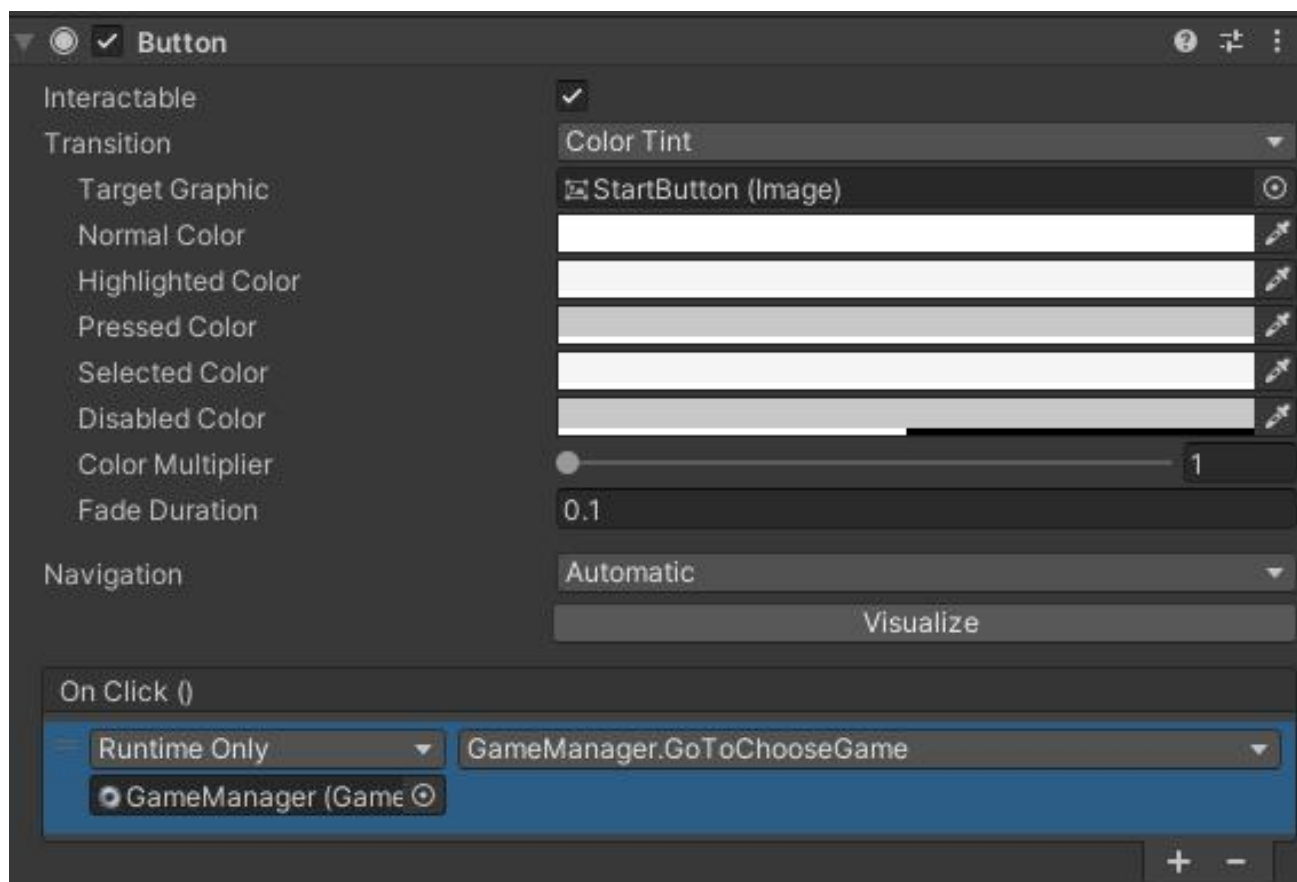
7
8     public static GameManager Instance;
9     public GameObject starBtn;
10    public GameObject mainMenuUi;
11    public GameObject AboutUi;
12    public GameObject ChooseGameUi;
13    public GameObject goBackBtn;
14    public GameObject quitBtn;
15
16
17    @ Unity Message | 0 references
18    private void Awake()
19    {
20        if(Instance == null)
21        {
22            Instance = this;
23        }
24        else
25        {
26            Destroy(gameObject);
27        }
28    }
29
30    0 references
31    public void GoToChooseGame()
32    {
33        mainMenuUi.SetActive(false);
34        ChooseGameUi.SetActive(true);
35    }
36
37    0 references
38    public void GoToMainMenu()
39    {
40        mainMenuUi.SetActive(true);
41        ChooseGameUi.SetActive(false);
42    }
43
44    0 references
45    public void GoToAboutMenu()
46    {
47        AboutUi.SetActive(true);
48        mainMenuUi.SetActive(false);
49    }
50
51    0 references
52    public void ExitGame()
53    {
54        Application.Quit();
55    }

```

Σχήμα 4.2 Script για εναλλαγή μενού.

Τα μενού, που δημιουργήθηκε, είναι τύπου game object. Εξαφανίζεται και εμφανίζεται με την μέθοδο setActive(). Αφού δημιουργηθούν σωστά τα functions, πρέπει να ορισθούν στα αντίστοιχα buttons. Αυτό γίνεται με την χρήση on click events μέσω του inspector. Αρχικά, κλικάροντας στο hierarchy, στο κουμπί δεξιά στον inspector το “component button” πρέπει να πατηθεί στο εικονίδιο “+” και έτσι, δημιουργείται ένα click event.

Έπειτα, πρέπει να συρθεί το script μέσα στο event και τέλος, πρέπει να επιλεγθεί το ανάλογο function για το κουμπί. Ένα παράδειγμα παρουσιάζεται στο σχήμα 4.3.

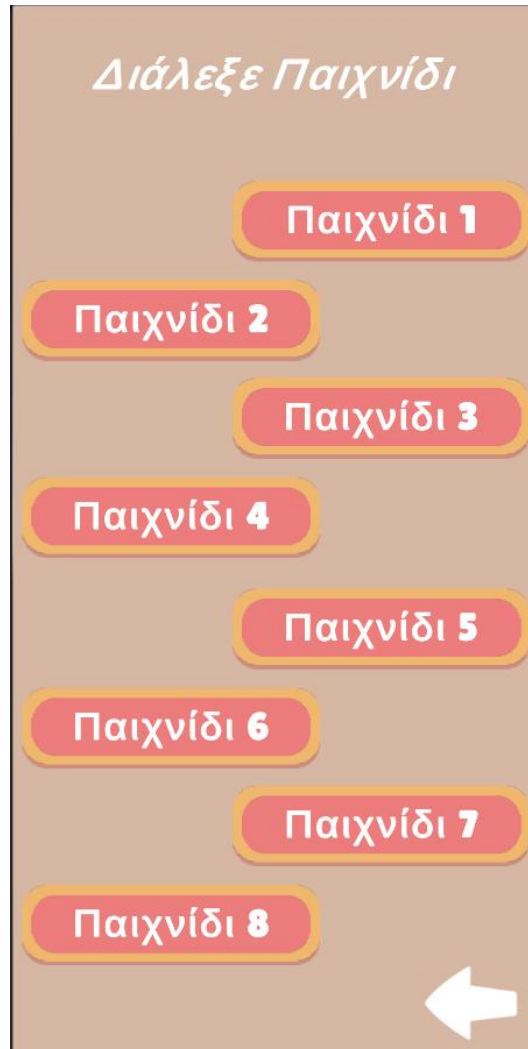


Σχήμα 4.3 Δημιουργία on click event και εισαγωγή function.

Αφού υλοποιηθούν σωστά τα παραπάνω βήματα, το παιχνίδι είναι έτοιμο για play. “Τρέχοντάς” το και πατώντας το κουμπί “Εκκίνηση” θα αλλάξει την οθόνη και θα οδηγήσει την σκηνή στο δεύτερο μενού, που είναι η επιλογή του παιχνιδιού.

#### 4.4 Δεύτερο μενού

Στην δεύτερο μενού της αρχικής οθόνης εμφανίζονται οχτώ κουμπιά για οχτώ διαφορετικά παιχνίδια και ένα κουμπί που μπορεί να επιστρέψει στην αρχική. Το κάθε παιχνίδι οδηγεί σε διαφορετικό κεφάλαιο από τα σχολικά βιβλία. Το δεύτερο μενού φαίνεται στο σχήμα 4.4.



Σχήμα 4.4 Δεύτερο μενού.

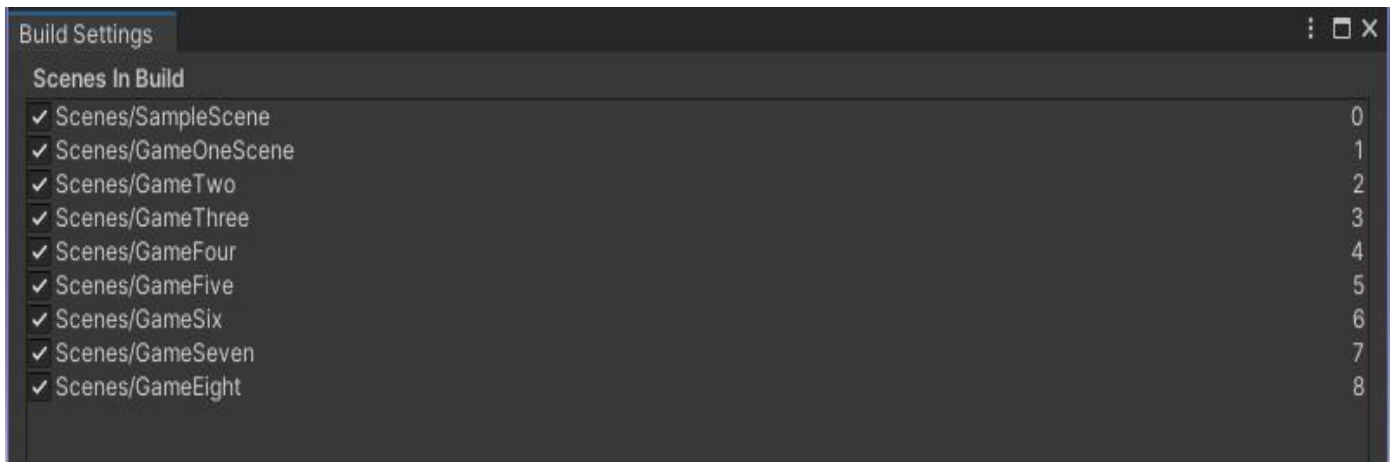
Το κάθε παιχνίδι ανήκει σε διαφορετικό scene, οπότε θα χρειαστούν και τα ανάλογα functions σε ένα καινούργιο script με όνομα SceneSwitcherManager, που θα υλοποιηθούν μέσα του. Ένα παράδειγμα ενός τέτοιου function υποδεικνύεται στο σχήμα 4.5.

```
9 | public void GoToGameOne()  
10 | {  
11 |     SceneManager.LoadScene(1);  
12 | }
```

Σχήμα 4.5 SceneSwitcherManager function.

## 4.5 Scenes

Όπως αναφέρθηκε και ανωτέρω, το κάθε παιχνίδι μέσα στην εφαρμογή ανήκει σε καινούργιο scene. Αυτό βοηθά να κρατηθεί καθαρός ο inspector και τα αρχεία να μην γίνονται πολύ μεγάλα. Στην ουσία τα scenes περιέχουν τα αντικείμενα του παιχνιδιού. Μπορούν να χρησιμοποιηθούν για τη δημιουργία ενός κύριου μενού, μεμονωμένων επιπέδων και οτιδήποτε άλλο. Σκεφτείτε κάθε μοναδικό αρχείο σκηνής ως ένα μοναδικό επίπεδο. Σε κάθε Σκηνή, θα τοποθετήσετε τα περιβάλλοντα, τα εμπόδια και τις διακοσμήσεις σας, ουσιαστικά σχεδιάζοντας και χτίζοντας το παιχνίδι σας σε κομμάτια [15]. Στην παρούσα πτυχιακή εργασία υπάρχουν στο σύνολο εννέα σκηνές. Η μία σκηνή αφορά το αρχικό μενού, και οι υπόλοιπες σκηνές είναι για τα οχτώ παιχνίδια που υπάρχουν στο σύνολο. Τα προαναφερόμενα καταδεικνύονται και στο παρακάτω σχήμα 4.6.



Σχήμα 4.6 Όλες οι σκηνές της εφαρμογής.

Επίσης, σημαντικό είναι να αναφερθεί ότι όταν γίνεται εναλλαγή μεταξύ σκηνών, εκείνη την στιγμή ο κώδικας επιστρέφει στην αρχική του κατάσταση. Δηλαδή, αν στην προηγούμενη σκηνή, όταν “έτρεχε” το παιχνίδι γεμίζαμε κάποιον πίνακα, μόλις η σκηνή αλλάξει, ο πίνακας θα επανέλθει στην αρχική του μορφή. Αυτό διευκολύνει αρκετά, ώστε να μην ξεχαστεί κάτι “βρόμικο” πίσω. Επίσης αν είναι επιθυμητό, μπορεί να πραγματοποιηθεί restart στο παιχνίδι καλώντας την μέθοδο `loadScene(index)`. Στην περίπτωση που επιλεγθεί το restart στο παιχνίδι είναι μια κίνηση, η οποία πραγματοποιείται αρκετά εύκολα, όπως γίνεται αντιληπτό στο σχήμα 4.7.

```

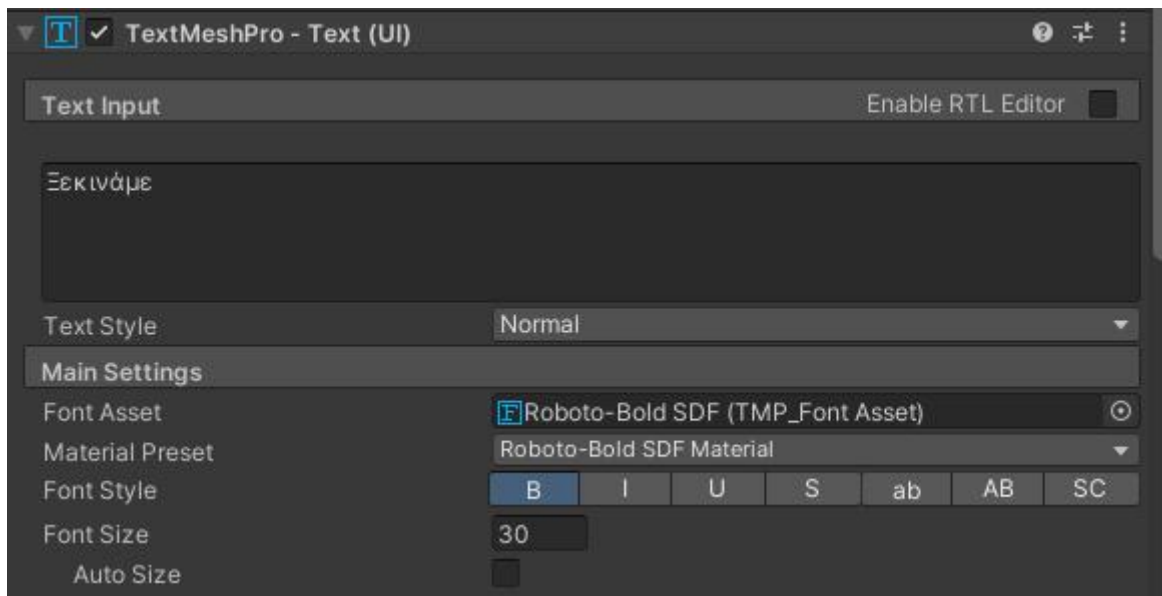
0 references
public void GoToGameOne()
{
    SceneManager.LoadScene(1);
}

```

Σχήμα 4.7 Function `loadScene()`.

## 4.6 Fonts

Τα fonts που θα χρησιμοποιηθούν στην παρούσα πτυχιακή εργασία δεν θα είναι κάποια από τα defaults της unity, διότι δεν καλύπτουν την ελληνική αλφάβητο. Ένα από τα κύρια fonts που χρησιμοποιούνται είναι το “Roboto-Bold SDF”. Δυστυχώς, δεν μπορεί να ορισθεί σαν προεπιλογή, οπότε κάθε φορά που δημιουργείται ένα text, στο οποίο μπαίνει κάποιο κείμενο, θα πρέπει να αλλάξει χειροκίνητα. Αυτό γίνεται κάνοντας κλικ από την αριστερή στήλη στο hierarchy το text και πηγαίνοντας δεξιά στα settings, βρίσκοντας το component “TextMeshPro-Text” και στο font asset επιλέγεται το επιθυμητό font. Κάτι τέτοιο καταδεικνύεται στο σχήμα 4.8.



Σχήμα 4.8 Επιλογή Font.

## 4.7 Ήχοι

Οι ήχοι που χρησιμοποιούνται στην συγκεκριμένη πτυχιακή εργασία δεν έχουν κάποια ιδιαιτερότητα. Γίνονται import μέσα στο project με mime type .mp4 και κάθε φορά που χρησιμοποιούνται πρέπει να καλείται η function `sound.Play()`, όπου το `sound` είναι ο ήχος που έχει γίνει prefab. Η function `sound.Play()` παρουσιάζεται στο σχήμα 4.9.

```

}
0 references
public void A_Note_Play()
{
    ANote.Play();
}

```

Σχήμα 4.9 Function `sound.play()`.

## 4.8 Επίλογος

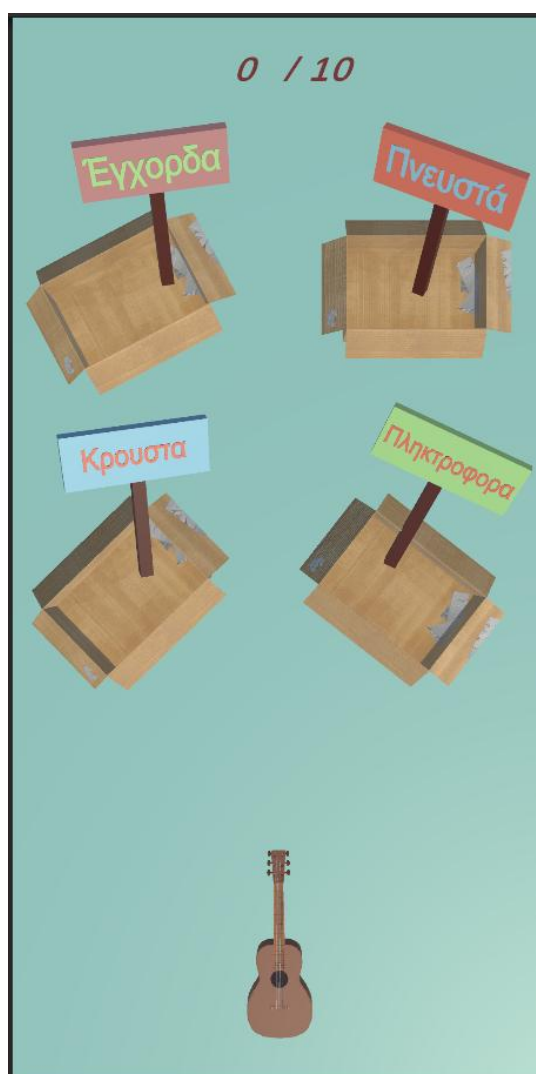
Σε αυτό το κεφάλαιο αναφερθήκαμε στην αρχική οθόνη και στα μενού που περιέχει αυτή. Έπειτα, αναλύσαμε κάποια σημαντικά functions. Τέλος, αναφερθήκαμε στα fonts και στους ήχους. Ο στόχος ήταν το αποτέλεσμα να κινείται σε ευδιάθετα επίπεδα για το κοινό που προορίζεται. Εκτός αυτού όμως, στόχος αποτελεί και η καλαισθησία.. Επίσης, σημαντικό είναι να σημειωθεί πως πλέον, έχουμε έναν ολοκληρωμένο τρόπο πλοήγησης και είμαστε έτοιμοι να παρουσιάσουμε το πρώτο μας παιχνίδι στο επόμενο κεφάλαιο.



## Κεφάλαιο 5ο: Μαθαίνοντας τα μουσικά όργανα

### 5.1 Εισαγωγή

Σε αυτό το κεφάλαιο δημιουργήθηκε το πρώτο παιχνίδι, το οποίο αφορά το όγδοο κεφάλαιο του βιβλίου της “Μουσικής” της Α’τάξης του Δημοτικού. Το πρώτο παιχνίδι είναι ένα παιχνίδι τύπου αντιστοίχισης. Όταν ξεκινάει το παιχνίδι, εμφανίζονται κάποια μουσικά όργανα τυχαία και ο χρήστης πρέπει να τα ταιριάζει με την κατηγορία τους, πχ. Κιθάρα - έγχορδα. Επίσης, κρατείται ένα score για το πόσες σωστές επιλογές πραγματοποιήσε. Η μορφή του παιχνιδιού φαίνεται στο σχήμα 5.1.



Σχήμα 5.1 Εκκίνηση του πρώτου παιχνιδιού.

Ο χρήστης μπορεί να αγγίξει το μουσικό όργανο και να το σύρει σε όποιο σημείο θέλει μέσα στην οθόνη. Ο έλεγχος, αν το έχει ταιριάζει με την σωστή κατηγορία του οργάνου, γίνεται μόλις αφήσει το δάχτυλό του. Στην περίπτωση που το αφήσει σε κενό σημείο, δεν γίνεται κάτι.

## 5.2 Λειτουργία drag and drop

Η λειτουργία όπου ο χρήστης μπορεί να πιάσει κάποιο αντικείμενο και να το σύρει γίνεται μέσω script, και πιο συγκεκριμένα με την μέθοδο `Input.GetTouch()`. Η μέθοδος αυτή πρέπει να “τρέχει” αναγκαστικά μέσα σε ένα function update, το οποίο δεν είναι αρκετά ωφέλιμο για την ταχύτητα και για την μνήμη του παιχνιδιού, επειδή το συγκεκριμένο update function “τρέχει” κάθε frame. Το Script, σύμφωνα με αυτά που αναφέρθηκαν παραπάνω, πρέπει να είναι όπως στο σχήμα 5.2.

```
Unity Message | 0 references
void Update()
{
    transform.Rotate(_rotation * speedrot * Time.deltaTime);

    if(Input.touchCount > 0)
    {
        touch = Input.GetTouch(0);

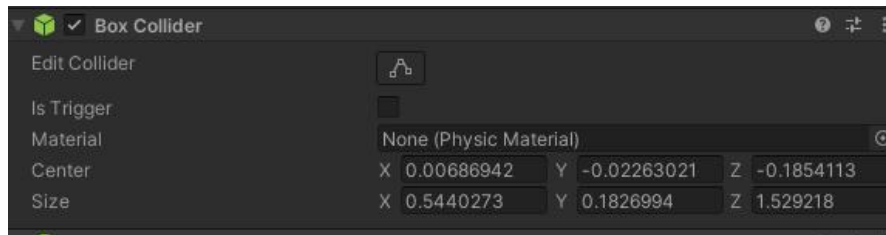
        if(touch.phase == TouchPhase.Moved) {
            transform.position = new Vector3(
                transform.position.x + touch.deltaPosition.x * speed,
                transform.position.y,
                transform.position.z + touch.deltaPosition.y * speed);
            speedrot = 0;
        }
        else
        {
            speedrot = 10;
        }
    }
}
```

Σχήμα 5.2 Function `Input.GetTouch()`

Το δημιουργηθέν αρχείο script πρέπει να “ζει” σε κάθε μουσικό όργανο που εμφανίζεται στην οθόνη. Για να γίνει αυτό, πρέπει το μουσικό όργανο να είναι prefab και να συρθεί το script μέσα στο μουσικό όργανο σαν component.

### 5.2.1 Έλεγχος σωστής κατηγορίας

Ο έλεγχος, όπως ειπώθηκε ανωτέρω, γίνεται κάθε φορά όταν ο χρήστης τραβήξει το δάχτυλό του από το αντικείμενο. Για να γίνει αυτό αντιληπτό, αν στο σημείο όπου άφησε το δάχτυλό του ο χρήστης, έχει κάποιο κουτί κατηγορίας, από κάτω του υπάρχουν colliders και ελέγχεται αν το tag του collider είναι ίδιο με αυτό που ψάχνουμε. Για να προστεθεί tag στο μουσικό όργανο, πρέπει να γίνει από δεξιά στον inspector κάνοντας add tag. Το collider, επίσης, πρέπει να μπει σαν component στο prefab του μουσικού οργάνου, όπως φαίνεται στο σχήμα 5.3.



Σχήμα 5.3 Collider μουσικού οργάνου.

Τέλος, για τον έλεγχο, πρέπει να διακριθεί αν το `input.GetTouch()` είναι ίσο με 0, δηλαδή αν ο χρήστης έχει σηκώσει το δάχτυλό του, μετά ελέγχει τα collider tags, και αν ο έλεγχος είναι αληθής, ο κώδικας προχωρά σε ορισμένες ακόμη λειτουργίες. Τη στιγμή που κάνει destroy το μουσικό όργανο, αλλάζει το score σε +1 και κάνει spawn το επόμενο τυχαίο μουσικό όργανο. Το script γίνεται φανερό στο παρακάτω σχήμα 5.4.

```

Unity Message | 0 references
void OnTriggerStay(Collider collider)
{
    if (Input.touchCount == 0)
    {
        if (collider.gameObject.tag == "drum")
        {
            Destroy(collider.gameObject);
            int scoretemp = int.Parse(scoreText.GetComponent<TextMeshProUGUI>().text) + 1;
            scoreText.GetComponent<TextMeshProUGUI>().text = scoretemp.ToString();
            gameManager.spawnObjectsOnDestroy();
        }
    }
}

```

Σχήμα 5.4 Script σωστού ελέγχου.

Επίσης, αυτό το script πρέπει να δημιουργηθεί αρκετές φορές. Για την ακρίβεια όσα είναι και τα μουσικά όργανα, διότι κάθε φορά πρέπει να ελέγχεται κάθε διαφορετικό tag και το μουσικό όργανο πρέπει να έχει ένα script μαζί του σαν component που θα ελέγχει το tag του. Στο συγκεκριμένο

## Κεφάλαιο 5

παιχνίδι υπάρχουν 10 μουσικά όργανα με τέσσερις διαφορετικές κατηγορίες, όπως διακρίνεται στο σχήμα 5.5.



Σχήμα 5.5 Scripts για διαφορετικά tags.

### 5.3 Spawn οργάνου

Αρχικά, έχει φτιαχτεί μια λίστα με τα μουσικά όργανα και κάθε φορά που το μουσικό όργανο τελειώνει την υλοποίησή του αφαιρείται από την λίστα. Το spawn του μουσικού οργάνου κάθε φορά είναι τυχαίο, το οποίο ελέγχεται από έναν game manager. Κάθε φορά που ο χρήστης αφήνει ένα αντικείμενο στην κατηγορία του καλεί τον game manager και εκείνος “τρέχει” ένα function με όνομα “SpawnObjectsOnDestroy()”. Πρώτα, το function ελέγχει, αν η λίστα με τα μουσικά όργανα είναι μεγαλύτερα από 0 και έπειτα, κάνει ένα random.range() για να διαλέξει ένα τυχαίο αριθμό από την λίστα. Τέλος, με μια switch case και ανάλογα με τον τυχαίο αριθμό εμφανίζεται το αντίστοιχο μουσικό όργανο από την λίστα. Το function παρουσιάζεται στο σχήμα 5.6.

```
public void spawnObjectsOnDestroy()
{
    if (gameObjectsList.Count > 0)
    {
        randomInt = Random.Range(0, gameObjectsList.Count);
        switch (randomInt)
        {
            case 0:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
            case 1:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
            case 2:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
            case 3:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
            case 4:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
            case 5:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
            case 6:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
            case 7:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
            case 8:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
            case 9:
                Instantiate(gameObjectsList[randomInt], spawnPos.transform.position, spawnPos.transform.rotation);
                gameObjectsList.Remove(gameObjectsList[randomInt]);
                break;
        }
    }
}
```

Σχήμα 5.6 function SpawnObjectsOnDestroy().

## 5.4 Οθόνη νίκης

Τέλος, αν ο έλεγχος της λίστας είναι ίσος με 0, σημαίνει ότι το πρώτο παιχνίδι έχει τελειώσει και στο σημείο αυτό πρέπει να εμφανιστεί ένα winning screen, προκειμένου να ενημερωθεί ο χρήστης για το score και για το πως θα πλοηγηθεί στην συνέχεια. Η οθόνη νίκης φαίνεται στο σχήμα 5.8.



Σχήμα 5.8 Οθόνη νίκης.

## 5.5 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάστηκε το πρώτο παιχνίδι που αναφέρεται στο όγδοο κεφάλαιο του βιβλίου. Αυτό ήταν ένα από τα οχτώ διαφορετικά παιχνίδια που υπάρχουν, με τίτλο “Μαθαίνοντας τα μουσικά όργανα”. Μέσω αυτού του παιχνιδιού, η μάθηση των κατηγοριών των μουσικών οργάνων πραγματοποιείται με έναν πιο ευχάριστο και ψυχαγωγικό τρόπο.



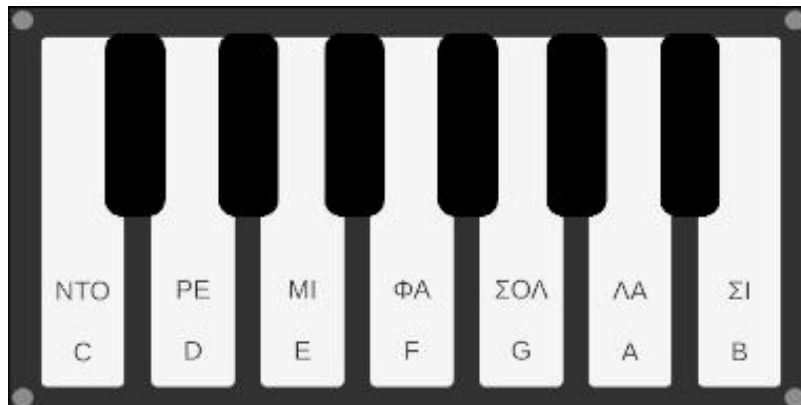
## Κεφάλαιο 6ο: Μαθαίνοντας τις νότες

### 6.1 Εισαγωγή

Στο παρόν κεφάλαιο δημιουργήθηκε το δεύτερο παιχνίδι της εφαρμογής, το οποίο αφορά την τέταρτη ενότητα του βιβλίου της “Μουσικής” της Β’ Δημοτικού τάξης. Η συγκεκριμένη σκηνή είναι η πρώτη που κάνει χρήση της AR επαυξημένης πραγματικότητας. Πιο συγκεκριμένα, ο χρήστης “σκανάρει” μια εικόνα και εμφανίζεται ένα λειτουργικό πιάνο στο οποίο και μπορεί να παίζει τις βασικές νότες.

### 6.2 Πιάνο

Το πιάνο δημιουργήθηκε μέσα στην unity, χρησιμοποιώντας 2D εικόνες, κάποιες από τις οποίες δημιουργήθηκαν στο photoshop και στη συνέχεια έγινε η μετατροπή στην unity. Το κάθε πλήκτρο αντιπροσωπεύει και μια διαφορετική νότα, το οποίο πραγματοποιείται με την χρήση event onclick(). Το UI του παιχνιδιού παρουσιάζεται στο σχήμα 6.1.



Σχήμα 6.1 AR πιάνο.

Το πιάνο θα εμφανιστεί στην οθόνη, αφού ο χρήστης “σκανάρει” μια συγκεκριμένη εικόνα, η οποία θα του εμφανίζεται στην έναρξη του παιχνιδιού, όπως γίνεται φανερό στο σχήμα 6.2.



Σχήμα 6.2 Εικόνα για scanning

Για να ανοιγοκλείνει αυτό το παράθυρο και να μην είναι συνεχώς μέσα στην οθόνη έχει δημιουργηθεί ένα function, το οποίο υλοποιεί τον σκοπό αυτόν. Η μέθοδος αυτή φαίνεται στο παρακάτω σχήμα 6.3.

```
0 references
public void openTheFrame()
{
    OpenTheImageSpawnBtn.SetActive(false);
    OpenTheImageSpawnFrame.SetActive(true);
    CloseTheImageSpawnBtn.SetActive(true);
}
0 references
public void CloseTheFrame()
{
    OpenTheImageSpawnBtn.SetActive(true);
    OpenTheImageSpawnFrame.SetActive(false);
    CloseTheImageSpawnBtn.SetActive(false);
}
```

Σχήμα 6.3 function open-close του παράθυρου.

### 6.3 Script

Το script για να ακουστεί ο ήχος από κάθε νότα με το αντίστοιχο κουμπί του είναι αρκετά απλό. Πρέπει να δημιουργηθεί ένας musicManager, που θα δέχεται σαν ορίσματα όλες τις νότες, οι οποίες θα είναι τύπου AudioSource, όπως γίνεται φανερό στο σχήμα 6.4.

```
public AudioSource ANote;
public AudioSource BNote;
public AudioSource CNote;
public AudioSource DNote;
public AudioSource ENote;
public AudioSource FNote;
public AudioSource GNote;
```

Σχήμα 6.4 Νότες μέσα στο script

Έπειτα, μέσα στον musicManager βρίσκονται και οι μέθοδοι που θα υπάρχουν στο αντίστοιχο κουμπί για να ακουστεί ο ήχος της νότας. Η λειτουργία αυτή γίνεται με την μέθοδο note.play(). Σε αυτό το script υπάρχουν επτά διαφορετικά τέτοια functions, όσες ακριβώς είναι και οι νότες. Τα functions γνωστοποιούνται στο κάτωθι σχήμα 6.5.

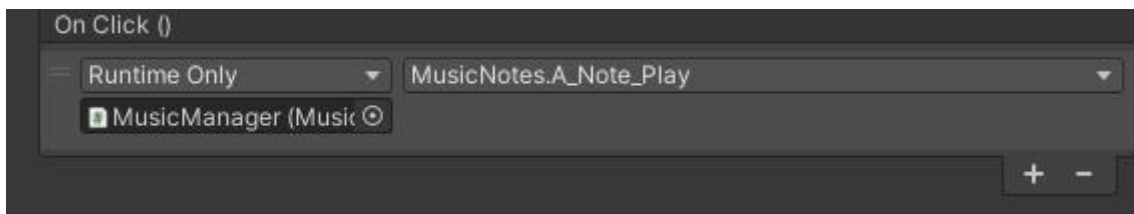
```

0 references
public void A_Note_Play()
{
    ANote.Play();
}
0 references
public void B_Note_Play()
{
    BNote.Play();
}
0 references
public void C_Note_Play()
{
    CNote.Play();
}
0 references
public void D_Note_Play()
{
    DNote.Play();
}
0 references
public void E_Note_Play()
{
    ENote.Play();
}
0 references
public void F_Note_Play()
{
    FNote.Play();
}
0 references
public void G_Note_Play()
{
    GNote.Play();
}

```

Σχήμα 6.5 Function note.play().

Το κάθε function ανατίθεται μετά στο αντίστοιχο κουμπί του. Αυτό γίνεται από δεξιά μέσω του inspector, αφού έχει γίνει κλικ πρώτα πάνω στο κουμπί, όπως καταδεικνύεται στο σχήμα 6.6.



Σχήμα 6.6 Εισαγωγή μεθόδου στο κουμπί.

Τέλος, στην σκηνή υπάρχουν 2 frames μέσα στα οποία υπάρχουν κάποια γνωστά τραγούδια πιάνου που αναγράφονται με τις νότες τους. Επιπρόσθετα, και σε αυτήν την λειτουργία υπάρχουν οι αντίστοιχες μέθοδοι για να ανοιγοκλείνουν τα frames, ώστε να μην καταλαμβάνουν ολόκληρη την οθόνη. Οι μέθοδοι, οι οποίες υλοποιούν την λειτουργία αυτή, φαίνονται στο σχήμα 6.7.

```

0 references
public void triggerOpenSongOne()
{
    songOneButtonOpen.transform.GetChild(1).GetComponent<Image>().color = new Color32(240, 184, 110, 255);
    songOneNotes.SetActive(true);
    songOneButtonClose.SetActive(true);
    songOneButtonOpen.GetComponent<Button>().interactable = false;
}

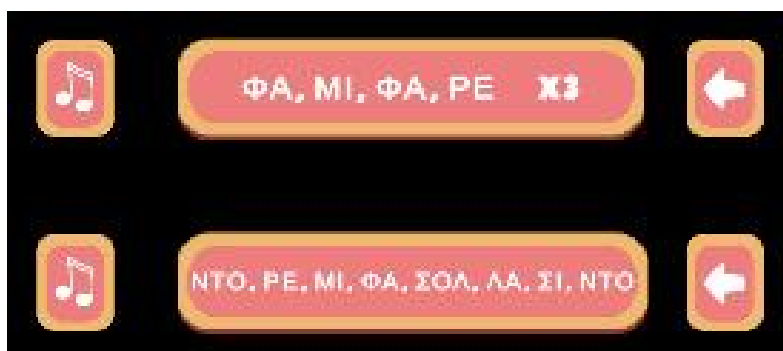
0 references
public void triggerCloseSongOne()
{
    songOneButtonOpen.transform.GetChild(1).GetComponent<Image>().color = new Color32(255, 255, 255, 255);
    songOneNotes.SetActive(false);
    songOneButtonClose.SetActive(false);
    songOneButtonOpen.GetComponent<Button>().interactable = true;
}

0 references
public void triggerOpenSongTwo()
{
    songTwoButtonOpen.transform.GetChild(1).GetComponent<Image>().color = new Color32(240, 184, 110, 255);
    songTwoNotes.SetActive(true);
    songTwoButtonClose.SetActive(true);
    songTwoButtonOpen.GetComponent<Button>().interactable = false;
}

0 references
public void triggerCloseSongTwo()
{
    songTwoButtonOpen.transform.GetChild(1).GetComponent<Image>().color = new Color32(255, 255, 255, 255);
    songTwoNotes.SetActive(false);
    songTwoButtonClose.SetActive(false);
    songTwoButtonOpen.GetComponent<Button>().interactable = true;
}
    
```

Σχήμα 6.7 Μέθοδοι open-close των δυο frames.

Για να ανοίξουν τα frames, πρέπει να πατηθεί το αριστερό κουμπί με την νότα, και να για να κλείσουν πρέπει να πατηθεί το κουμπί με το εικονίδιο πίσω. Τα κουμπιά αναδεικνύονται στο παρακάτω σχήμα 6.8.



Σχήμα 6.8 Παράθυρα τραγουδιών.

## 6.4 Επίλογος

Το παρόν κεφάλαιο αναφέρεται στην τέταρτη ενότητα του σχολικού βιβλίου της “Μουσικής” της Β’ Δημοτικού τάξης, που ασχολείται με τις νότες. Με την συγκεκριμένη υλοποίηση επίσης, κάναμε χρήση της επαυξημένης πραγματικότητας, η οποία μας προσφέρει βοήθεια στο παιχνίδι, ώστε να εκμεταλλευτούμε με έναν διαδραστικό τρόπο και να επέλθει η εκμάθηση της νότας με έναν ακόμη πιο ευχάριστο και ψυχαγωγικό τρόπο.



## Κεφάλαιο 7ο: Μαθαίνοντας τα σχήματα

### 7.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιαστεί το τρίτο παιχνίδι της εφαρμογής, το οποίο αναφέρεται στην Α' ενότητα του σχολικού βιβλίου "Εικαστικά" της Α' & της Β' τάξης του Δημοτικού με τίτλο "Ο κόσμος όλος με σχήματα". Το παιχνίδι αυτό πραγματοποιεί εξίσου τη χρήση της επαυξημένης πραγματικότητας.

### 7.2 UI

Στο UI, που παρουσιάζεται στο σχήμα 7.1, η υλοποίηση του περιλαμβάνει ένα frame για την εικόνα σκαναρίσματος, που ο χρήστης μπορεί να το ανοιγοκλείνει και διάφορα 3D μοντέλα, τα οποία έχουν δημιουργηθεί στο blender.



Σχήμα 7.1 Εικόνα για scanning

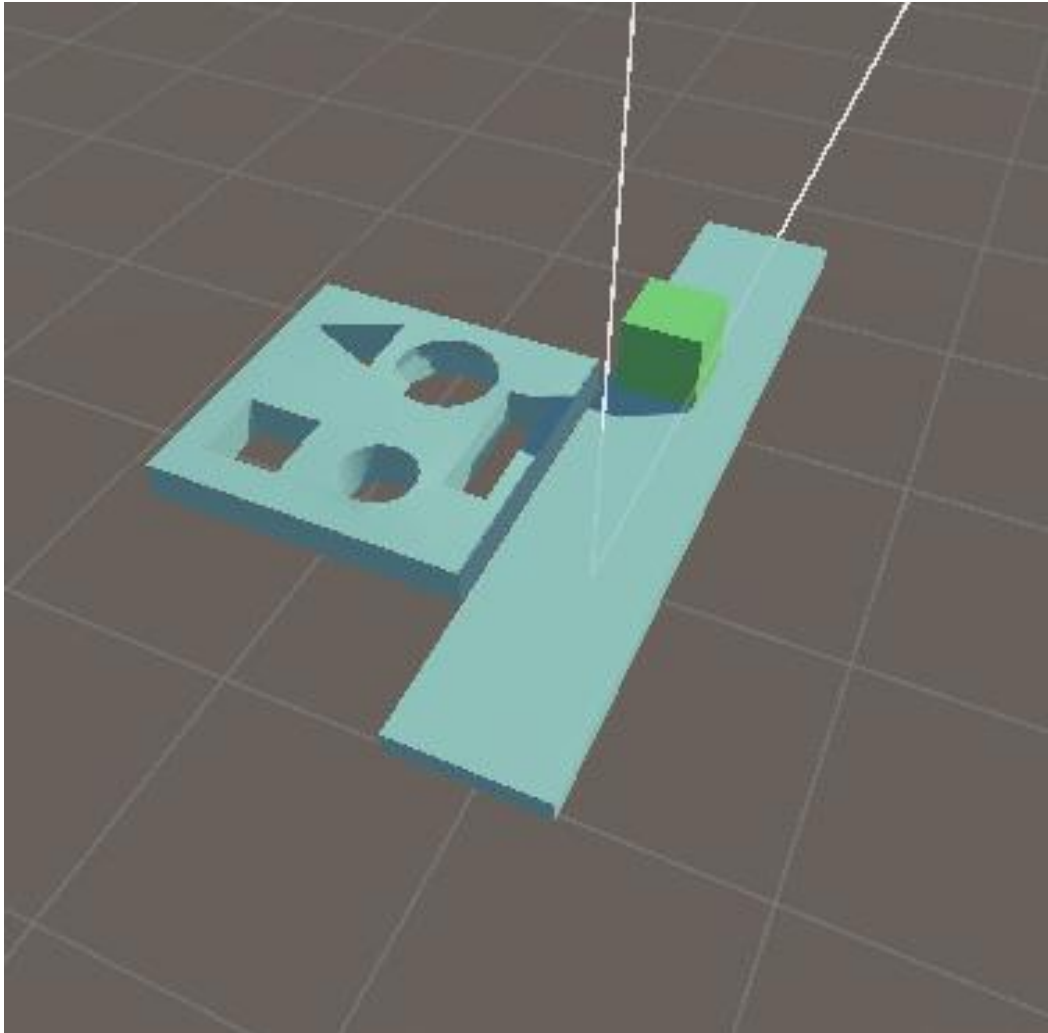
Ο χρήστης μπορεί να ανοιγοκλείσει το frame πατώντας ένα κουμπί στα δεξιά του. Η λειτουργία αυτή γίνεται μέσω script με ένα απλό function, που χρησιμοποιεί την μέθοδο `frame.SetActive(true)`. Οι μέθοδοι αυτές υποδεικνύονται στο σχήμα 7.2.

```
0 references
public void showImage()
{
    openIcon.SetActive(false);
    imageFrame.SetActive(true);
    closeIcon.SetActive(true);
}

0 references
public void closeImage()
{
    closeIcon.SetActive(false);
    imageFrame.SetActive(false);
    openIcon.SetActive(true);
}
```

Σχήμα 7.2 function open-close του παράθυρου.

Στο παιχνίδι αυτό, μόλις ξεκινήσει και αφού σκαναριστεί η εικόνα, εμφανίζεται μια πλάκα με κομμένα κάποια σχήματα πάνω του. Μαζί με αυτό εμφανίζεται και ένα σχήμα σε μια άλλη πλάκα πιο κάτω. Στο σημείο αυτό, ο χρήστης πρέπει να σύρει το αντικείμενο και να το ταιριάξει με κόψιμο πάνω στην πρώτη πλάκα, όπως γίνεται φανερό στο σχήμα 7.3.



Σχήμα 7.3 Πλάκες με σχήματα.

Επίσης, αφού ο χρήστης αφήσει το αντικείμενο να πέσει στην κόψη της πλάκας, γίνεται έλεγχος για το αν ταιριάζει το σχήμα με την χρήση colliders.

### 7.3 Script

Για να έχει ο χρήστης τη δυνατότητα να σύρει κάποιο αντικείμενο, ο στόχος αυτός επιτυγχάνεται μέσω κάποιου script. Πιο συγκεκριμένα, με την μέθοδο `Input.GetTouch()`. Αυτή η μέθοδος πρέπει να “τρέχει” αναγκαστικά μέσα σε ένα function `update`. Βέβαια, αξίζει να αναφερθεί πως αυτό δεν είναι αρκετά ωφέλιμο για τον τρόπο ταχύτητας και για την μνήμη του παιχνιδιού μας, επειδή το `update` function “τρέχει” κάθε frame. Το Script, σύμφωνα μεόσα αναφέραμε παραπάνω, πρέπει να είναι όπως στο σχήμα 7.3.

```
Unity Message | 0 references
void Update()
{
    transform.Rotate(_rotation * speedrot * Time.deltaTime);

    if (Input.touchCount > 0)
    {
        touch = Input.GetTouch(0);

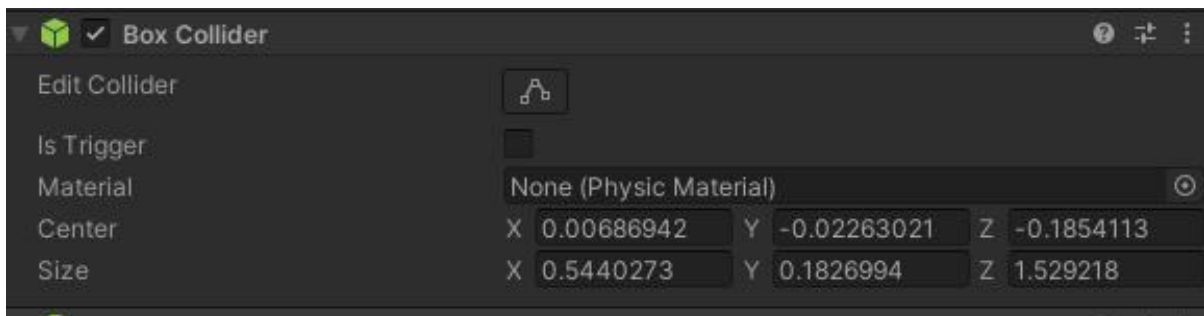
        if (touch.phase == TouchPhase.Moved)
        {
            transform.position = new Vector3(
                transform.position.x + touch.deltaPosition.x * speed,
                transform.position.y,
                transform.position.z + touch.deltaPosition.y * speed);
            speedrot = 0;
        }
        else
        {
            speedrot = 10;
        }
    }
}
```

Σχήμα 7.4 Function `Input.GetTouch()`.

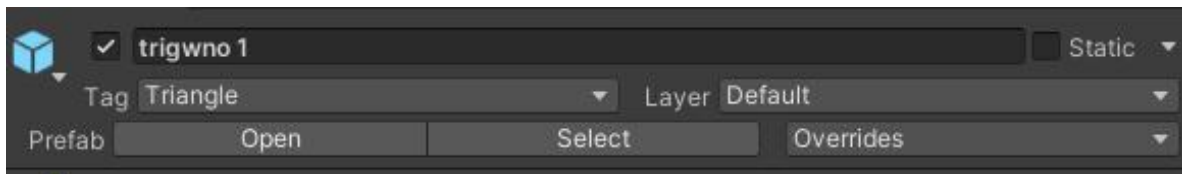
Αυτό το αρχείο script, το οποίο δημιουργήθηκε, πρέπει να “ζει” σε κάθε σχήμα που εμφανίζεται στην οθόνη. Για να καταστεί αυτό δυνατόν, πρέπει το σχήμα να είναι prefab και σέρνοντάς το μέσα σαν component του script.

### 7.3.1 Έλεγχος σωστής κατηγορίας

Ο έλεγχος γίνεται κάθε φορά, όταν ο χρήστης τραβήξει το δάχτυλό του από το αντικείμενο, και όταν το αντικείμενο πέσει στο έδαφος και περάσει ανάμεσα από τις κόψεις. Για τον έλεγχο χρησιμοποιούνται colliders και ελέγχεται αν το tag του collider είναι ίδιο με αυτό που χρειάζεται. Για να προστεθεί το tag στο κάθε σχήμα γίνεται από δεξιά στο inspector κάνοντας add tag. Το collider επίσης, πρέπει να μπει σαν component στο prefab του σχήματος. Το σχήμα 7.5 αναφέρεται στα colliders ενώ το σχήμα 7.6 στα tags.

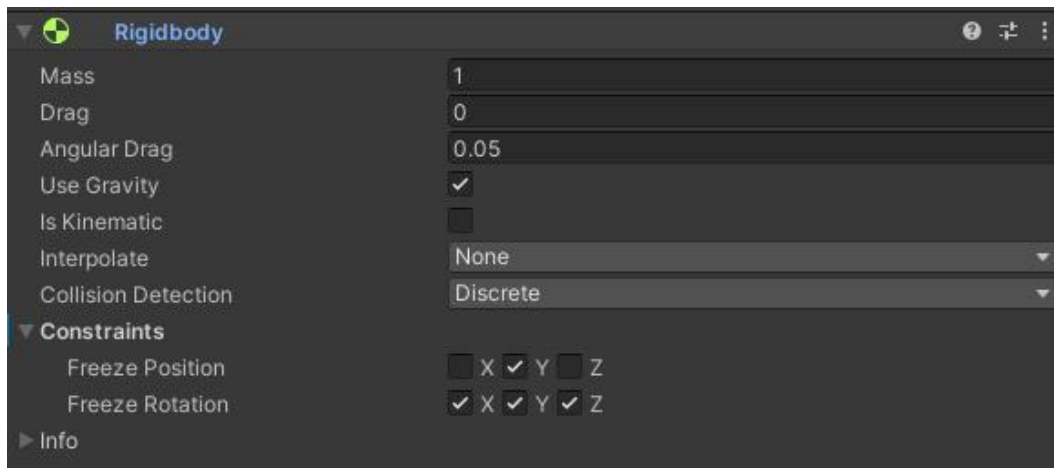


Σχήμα 7.5 Collider σχήματος.



Σχήμα 7.6 Tag σχήματος.

Επιπλέον, τα σχήματα αυτά για να αρχίζουν να πέφτουν προς το έδαφος, όταν ο χρήστης αφήσει το δάχτυλό του, πρέπει να προσθέσει στο κάθε σχήμα ένα component, που ονομάζεται Rigidbody και να ανοίξει την επιλογή use gravity, όπως ακριβώς φαίνεται και στο παρακάτω σχήμα 7.7.



Σχήμα 7.7 Rigidbody σχήματος.

Τέλος, ελέγχει τα collider tags και αν ο έλεγχος είναι αληθής κάνει κάποιες ακόμα λειτουργίες, όπως το να κάνει destroy το σχήμα, να εμφανίσει κάποια διαφορετικά εφέ (particle system) και να κάνει spawn το επόμενο τυχαίο σχήμα. Το script φαίνεται στο παρακάτω σχήμα 7.8.

```

5 references
void OnTriggerEnter(Collider collider)
{
    if (collider.gameObject.tag == "Square")
    {
        Destroy(collider.gameObject);
        GameManager.spawnObjectsOnDestroy();
        GameObject particleSystemObject = Instantiate(particlePrefab, squareSpawnPrefab.transform.position, Quaternion.identity);
        ParticleSystem particleSystem = particleSystemObject.GetComponent<ParticleSystem>();
        particleSystem.Play();
        Destroy(particleSystem, 2f);
    }
}

```

Σχήμα 7.8 Script σωστού ελέγχου.

### 7.3.2 Spawn τυχαίου σχήματος

Αρχικά, έχει φτιαχτεί μία λίστα με τα σχήματα και κάθε φορά που το σχήμα καταστρέφεται, αφαιρείται από την λίστα. Το spawn του σχήματος κάθε φορά είναι τυχαίο, κάτι το οποίο ελέγχεται και εκτελείται από έναν game manager. Κάθε φορά που ο χρήστης αφήνει ένα αντικείμενο στην κατηγορία του, καλεί τον game manager και εκείνος “τρέχει” ένα function με το όνομα “SpawnObjectsOnDestroy()”. Πρώτα, το function ελέγχει αν η λίστα με τα σχήματα είναι μεγαλύτερα από 0 και έπειτα, κάνει ένα random.range() για να διαλέξει έναν τυχαίο αριθμό από την λίστα. Τέλος, εκτελείται μια switch case και ανάλογα με τον τυχαίο αριθμό εμφανίζεται το αντίστοιχο σχήμα από την λίστα. Το SpawnObjectsOnDestroy() παρουσιάζεται στο σχήμα 7.9.

```

5 references
public void spawnObjectsOnDestroy()
{
    if (gameObjectList.Count > 0)
    {
        randomInt = Random.Range(0, gameObjectList.Count);
        switch (randomInt)
        {
            case 0:
                Instantiate(gameObjectList[randomInt], spawnPos2.transform.position, spawnPos2.transform.rotation);
                gameObjectList.Remove(gameObjectList[randomInt]);
                break;
            case 1:
                Instantiate(gameObjectList[randomInt], spawnPos2.transform.position, spawnPos2.transform.rotation);
                gameObjectList.Remove(gameObjectList[randomInt]);
                break;
            case 2:
                Instantiate(gameObjectList[randomInt], spawnPos2.transform.position, spawnPos2.transform.rotation);
                gameObjectList.Remove(gameObjectList[randomInt]);
                break;
            case 3:
                Instantiate(gameObjectList[randomInt], spawnPos2.transform.position, spawnPos2.transform.rotation);
                gameObjectList.Remove(gameObjectList[randomInt]);
                break;
        }
    }
    else if (gameObjectList.Count <= 0)
    {
        winPanel.SetActive(true);
    }
}

```

Σχήμα 7.9 Function SpawnObjectsOnDestroy.

Εν τέλει, αν ο έλεγχος της λίστας είναι ίσος με 0, αυτό σημαίνει ότι το παιχνίδι έχει τελειώσει και εκεί πρέπει να εμφανιστεί ένα winning screen για να μπορέσει ο χρήστης να πλοηγηθεί στην συνέχεια.



Σχήμα 7.10 Οθόνη νίκης.

## 7.4 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάστηκε ένα παιχνίδι, το οποίο αφορά την ενότητα 7 του σχολικού βιβλίου “Εικαστικά” της Α΄ & Β΄ τάξης του Δημοτικού με τίτλο “Ο κόσμος όλος με σχήματα”. Το παρόν παιχνίδι έκανε χρήση της επαυξημένης πραγματικότητας και περιείχε τρισδιάστατα μοντέλα. Για την επιτυχή λειτουργία του παιχνιδιού δημιουργήθηκαν αντικείμενα από το πρόγραμμα Blender, ώστε να επιτευχθεί η διαδραστικότητα με όλα αυτά τα αντικείμενα στο παιχνίδι. Είναι άξιο επισήμανσης πως υλοποιώντας το παραπάνω παιχνίδι, η μάθηση των σχημάτων γίνεται με έναν πιο ευχάριστο τρόπο.



## Κεφάλαιο 8ο: Μαθαίνοντας τα χρώματα

### 8.1 Εισαγωγή

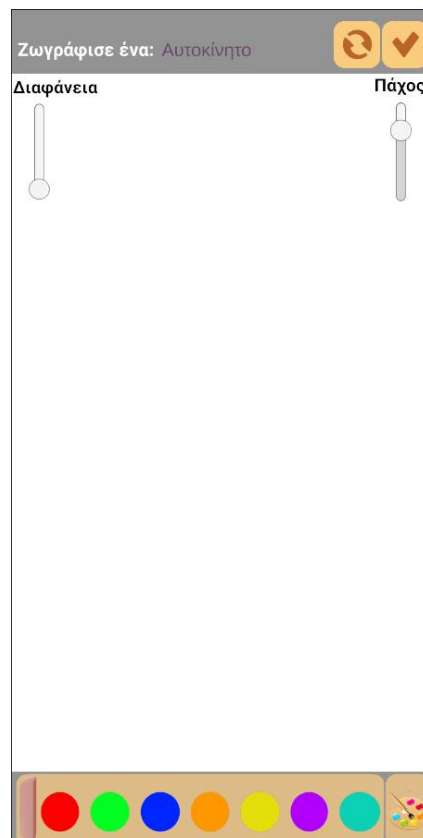
Στο παρόν κεφάλαιο θα δημιουργηθεί ένας λειτουργικός λευκός πίνακας. Το παιχνίδι αυτό κάνει αναφορά στην Α' ενότητα του σχολικού βιβλίου των "Εικαστικών" της Α' & Β' τάξης του Δημοτικού με τίτλο "Μορφές από γραμμές σχήματα και χρώματα". Ο χρήστης θα μπορεί να ζωγραφίσει στον λευκό πίνακα διαλέγοντας χρώματα από μια παλέτα χρωμάτων.

### 8.2 Πίνακας ζωγραφικής

Στο παρόν παιχνίδι, που έχει δημιουργηθεί στην εφαρμογή, ο χρήστης έχει τη δυνατότητα μέσω μίας παλέτας χρωμάτων να επιλέξει ένα χρώμα και να ζωγραφίσει ό,τι θέλει στον πίνακα. Υπάρχει, επίσης, και μία γόμα, όπου μπορεί κανονικά να σβήσει. Επιπλέον, έχουν δοθεί στον χρήστη δυο sliders: ένα που μπορεί να ελέγξει το πάχος από το πινέλο του, και ένα για να ελέγξει την διαφάνειά του. Τέλος, παραχωρούνται στον χρήστη με ένα text ορισμένες ιδέες, τις οποίες και μπορεί να αξιοποιήσει στη ζωγραφική του.

#### 8.2.1 UI

Το παιχνίδι αυτό χρησιμοποιεί σαν πίνακα ένα δισδιάστατο image με κάποια επεξεργασία στο photoshop. Η παλέτα χρωμάτων έχει, επίσης, υλοποιηθεί στο photoshop. Στην πάνω δεξιά γωνία του πίνακα έχουν δημιουργηθεί και δύο κουμπιά. Το ένα για να διαλέξει άλλη λέξη, που θα ήθελε να ζωγραφίσει, και το δεύτερο κουμπί σχετίζεται με μία τύπου "ολοκλήρωση", ώστε να ολοκληρωθεί το παιχνίδι. Τα sliders είναι από την unity σαν 2D elements και δεν έχει πραγματοποιηθεί κάποια επεξεργασία πάνω τους. Το τελικό UI παρουσιάζεται στο κάτωθι σχήμα 8.1.



Σχήμα 8.1 Λευκός πίνακας.

## 8.2.2 Script

Για να γίνει εφικτή η ζωγραφική στον πίνακα χρησιμοποιήθηκαν διάφορες τεχνικές. Αρχικά, πρέπει να γίνει κατανοητό ότι ο χρήστης ακουμπάει το δάχτυλό του στον πίνακα, κάτι το οποίο υλοποιήθηκε με μία update μέθοδο και με τη χρήση collider2D, όπου κατανοήθηκε ακριβώς σε ποιο σημείο ήταν το δάχτυλό του, έπειτα περνιέται το πινέλο και το χρώμα που είχε διαλέξει και μετατράπηκαν τα pixels σε συντεταγμένες. Οι τρεις αυτές μέθοδοι φαίνονται στα σχήματα 8.2 και 8.3.

```

Unity Message | 0 references
void Update()
{
    bool touch_held_down = Input.touchCount > 0;

    if (touch_held_down && !no_drawing_on_current_drag)
    {
        Vector2 touch_position = Camera.main.ScreenToWorldPoint(Input.GetTouch(0).position);

        Collider2D hit = Physics2D.OverlapPoint(touch_position, Drawing_Layers.value);

        if (hit != null && hit.transform != null)
        {
            current_brush(touch_position);
        }
        else
        {
            previous_drag_position = Vector2.zero;
            if (!mouse_was_previously_held_down)
            {
                no_drawing_on_current_drag = true;
            }
        }
    }
    else if (!touch_held_down)
    {
        previous_drag_position = Vector2.zero;
        no_drawing_on_current_drag = false;
    }
    mouse_was_previously_held_down = touch_held_down;
}

```

Σχήμα 8.2 Update μέθοδος.

```

0 references
public void ColourPixels(Vector2 center_pixel, int pen_thickness, Color color_of_pen)
{
    int center_x = (int)center_pixel.x;
    int center_y = (int)center_pixel.y;
    for (int x = center_x - pen_thickness; x <= center_x + pen_thickness; x++)
    {
        for (int y = center_y - pen_thickness; y <= center_y + pen_thickness; y++)
        {
            drawable_texture.SetPixel(x, y, color_of_pen);
        }
    }

    drawable_texture.Apply();
}

2 references
public Vector2 WorldToPixelCoordinates(Vector2 world_position)
{
    Vector3 local_pos = transform.InverseTransformPoint(world_position);

    float pixelWidth = drawable_sprite.rect.width;
    float pixelHeight = drawable_sprite.rect.height;
    float unitsToPixels = pixelWidth / drawable_sprite.bounds.size.x * transform.localScale.x;

    float centered_x = local_pos.x * unitsToPixels + pixelWidth / 2;
    float centered_y = local_pos.y * unitsToPixels + pixelHeight / 2;

    Vector2 pixel_pos = new Vector2(Mathf.RoundToInt(centered_x), Mathf.RoundToInt(centered_y));

    return pixel_pos;
}

```

Σχήμα 8.3 Μετατροπή pixels σε συντεταγμένες.

### 8.2.3 Τυχαία λέξη

Αρχικά, έχει δημιουργηθεί μια λίστα με τυχαίες λέξεις. Κάθε φορά που επιλέγεται μια λέξη, όσες φορές και να κάνει refresh ο χρήστης, δεν θα εμφανιστεί ξανά η ίδια. Η επιλογή την πρώτης λέξης γίνεται στην start μέθοδο, ενώ η δημιουργία της λίστας γίνεται στην awake μέθοδο, όπως υποδεικνύει το σχήμα 8.4.

```
Unity Message | 0 references
private void Awake()
{
    gameNames.Add("Σπίτι");
    gameNames.Add("Ποδήλατο");
    gameNames.Add("Αυτοκίνητο");
    gameNames.Add("Μηχανάκι");
    gameNames.Add("Δένδρο");
    gameNames.Add("Τετράδιο");
    gameNames.Add("Στυλό");
    gameNames.Add("Σκύλο");
    gameNames.Add("Γάτο");
    gameNames.Add("Υπολογιστή");
    gameNames.Add("Δράκο");
    gameNames.Add("Φαγητό");
    gameNames.Add("Κινητό");
    gameNames.Add("Τραπέζι");
    gameNames.Add("Μαρκαδόρο");
    gameNames.Add("Μολύβι");
    gameNames.Add("Φυτό");
    gameNames.Add("Ζώο");
    gameNames.Add("Φρούτο");
    gameNames.Add("Παιχνίδι");
    gameNames.Add("Τετράγωνο");
    gameNames.Add("Κύκλο");
    gameNames.Add("Τρίγωνο");
}

Unity Message | 0 references
private void Start()
{
    Color randomColor = new Color(Random.value, Random.value, Random.value, 1f);
    titleValue.GetComponent<TextMeshProUGUI>().color = randomColor;
    int randomIndex = Random.Range(0, gameNames.Count);
    titleValue.GetComponent<TextMeshProUGUI>().text = gameNames[randomIndex].ToString();
    gameNames.RemoveAt(randomIndex);
    particalPrefab.SetActive(false);
}
```

Σχήμα 8.4 Δημιουργία λίστας και επιλογή τυχαία λέξης.

Έπειτα, καλείται η changeTitleValue() κάθε φορά που ο χρήστης κάνει refresh. Αυτή η μέθοδος ελέγχει σε πρώτο στάδιο αν έχουν απομείνει λέξεις στην λίστα και σε δεύτερο στάδιο εμφανίζει την επόμενη λέξη με τυχαίο τρόπο, ενώ απευθείας την διαγράφει και από την λίστα. Η μέθοδος changeTitleValue() παρουσιάζεται στο σχήμα 8.5.

```
0 references
public void changeTitleValue()
{
    if(gameNames.Count > 0)
    {
        Color randomColor = new Color(Random.value, Random.value, Random.value, 1f);
        titleValue.GetComponent<TextMeshProUGUI>().color = randomColor;
        int randomIndex = Random.Range(0, gameNames.Count);
        titleValue.GetComponent<TextMeshProUGUI>().text = gameNames[randomIndex].ToString();
        gameNames.RemoveAt(randomIndex);
    }
    else
    {
        gameOverTitle.SetActive(true);
        title.SetActive(false);
        titleValue.SetActive(false);
    }
}
```

Σχήμα 8.5 Function changeTitleValue.

#### 8.2.4 Οθόνη νίκης

Τέλος, αν ο έλεγχος της λίστας είναι ίσος με 0, αυτό σημαίνει ότι το παιχνίδι έχει τελειώσει, σημείο στο οποίο πρέπει να εμφανιστεί ένα winning screen για να μπορέσει ο χρήστης να πλοηγηθεί στην συνέχεια. Η οθόνη νίκης φαίνεται στο σχήμα 8.6.



Σχήμα 8.6 Οθόνη νίκης.

### 8.3 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάστηκε ένα παιχνίδι, το οποίο αφορά την ενότητα του σχολικού βιβλίου των “Εικαστικών” της Α΄ & Β΄ τάξης του Δημοτικού με τίτλο “Μορφές από γραμμές, σχήματα και χρώματα”. Το παρόν παιχνίδι έκανε χρήση δισδιάστατων στοιχείων. Για την επιτυχή λειτουργία του παιχνιδιού τα στοιχεία αυτά δημιουργήθηκαν από το πρόγραμμα Photoshop. Υλοποιώντας το παραπάνω παιχνίδι, ο μαθητής μπορεί να έχει στη διάθεσή του έναν λευκό πίνακα, να μπορεί να ζωγραφίσει και να μάθει τα χρώματα με έναν πιο διασκεδαστικό τρόπο.



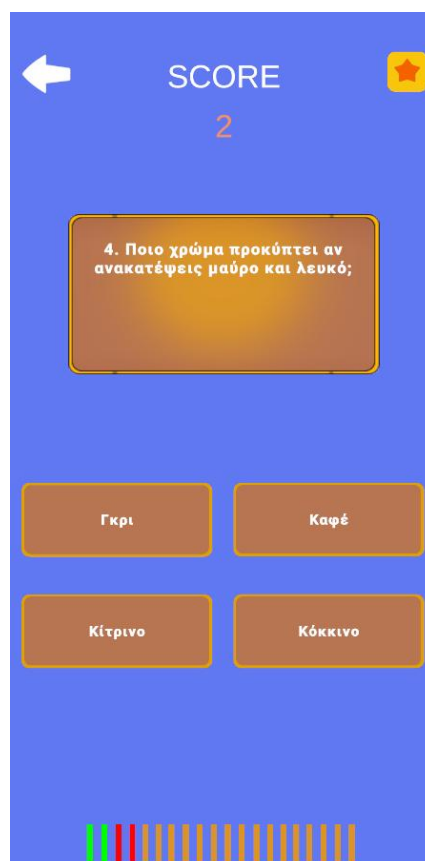
## Κεφάλαιο 9ο: QUIZ

### 9.1 Εισαγωγή

Σε αυτό το κεφάλαιο το παιχνίδι που θα δημιουργηθεί είναι το πρώτο παιχνίδι, το οποίο αφορά και τα δυο μαθήματα της Α΄ και Β΄ τάξης του Δημοτικού. Με το quiz δίνεται η δυνατότητα στον χρήστη με ένα γρήγορο και ενδιαφέρον τρόπο να αξιολογήσει τις γνώσεις του. Το παρόν παιχνίδι περιέχει 20 ερωτήσεις και 4 απαντήσεις σε κάθε ερώτηση. Για κάθε σωστή απάντηση κερδίζει έναν πόντο και ανανεώνεται το score του στην οθόνη. Στο τέλος του παιχνιδιού εμφανίζεται μια οθόνη νίκης για την επίδοση του χρήστη. Τέλος, σημαντικό είναι να αναφερθεί ότι οι ερωτήσεις είναι ή με κείμενο ή ο χρήστης πρέπει να ακούσει κάτι και να απαντήσει, όπως επίσης και να έρθει σε επαφή με κάποια εικόνα και στη συνέχεια να απαντήσει.

### 9.2 UI

Όλο το UI του παιχνιδιού έχει δημιουργηθεί στο photoshop. Υπάρχει ένα μεγάλος πίνακας, στον οποίο θα εμφανίζεται η κάθε ερώτηση μαζί με τον αριθμό της. Κάτω από τον πίνακα υπάρχουν 4 πιο μικρά images με τις απαντήσεις. Η παλέτα χρωμάτων έχει υλοποιηθεί και εκείνη με τη σειρά της στο photoshop. Πάνω υπάρχει το score του, το οποίο συνιστά απλώς ένα textMeshPro. Αριστερά πάνω βρίσκεται ένα κουμπί, το οποίο οδηγεί τον χρήστη στην αρχική οθόνη και κάτω κάτω υπάρχουν 20 διαφορετικά μικρά παραλληλόγραμμα images, που υποδεικνύουν στον χρήστη σε ποια ερώτηση είναι, και αν οι προηγούμενες απαντήσεις του ήταν σωστές ή λανθασμένες. Το τελικό UI γίνεται φανερό στο σχήμα 9.1.



Σχήμα 9.1 UI Quiz.

### 9.3 Script

Αρχικά, στην start μέθοδο πρέπει να δημιουργηθούν οι ερωτήσεις και να τοποθετηθούν σε μια λίστα. Οι ερωτήσεις αυτές είναι γραμμένες στο χέρι, όπως και επίσης οι απαντήσεις τους. Όλος ο έλεγχος γίνεται μέσω ενός script, που ονομάζεται “GameFiveManager”. Στο παρακάτω σχήμα 9.2 φαίνεται η εισαγωγή των ερωτήσεων.

```
listOfQuestions.Add("Ποιο χρώμα προκύπτει αν ανακατέψεις Κόκκινο και Κίτρινο;");
listOfQuestions.Add("Ποιο χρώμα προκύπτει αν ανακατέψεις Κόκκινο και Μπλε;");
listOfQuestions.Add("Πως ονομάζεται αυτό το Σχήμα;");
listOfQuestions.Add("Από ποιο μουσικό όργανο είναι αυτός ο ήχος;");
listOfQuestions.Add("Ποιο χρώμα προκύπτει αν ανακατέψεις μαύρο και λευκό;");
listOfQuestions.Add("Ποιο υλικό χρησιμοποιείς όταν ζωγραφίζεις και θέλεις να το καθαρίσεις εύκολα;");
listOfQuestions.Add("Σε ποια κατηγορία οργάνων ανήκει το πιάνο;");
listOfQuestions.Add("Σε ποια κατηγορία οργάνων ανήκει η κιθάρα;");
listOfQuestions.Add("Πως ονομάζεται αυτό το Σχήμα;");
listOfQuestions.Add("Αν ανακατέψεις πράσινο και κόκκινο, ποιο χρώμα θα προκύψει;");
listOfQuestions.Add("Τι χρώμα προκύπτει αν ανακατέψεις κόκκινο και μπλε;");
listOfQuestions.Add("Από ποιο μουσικό όργανο είναι αυτός ο ήχος;");
listOfQuestions.Add("Ποιά είναι αυτή η νότα;");
listOfQuestions.Add("Τι υλικό χρησιμοποιείς για να δημιουργήσεις σχέδια με άσπρο και μαύρο;");
listOfQuestions.Add("Τι είδος ζωγραφικής κάνεις όταν βάζεις χρώματα σε έναν μεγάλο χαρτί;");
listOfQuestions.Add("Από ποιο μουσικό όργανο είναι αυτός ο ήχος;");
listOfQuestions.Add("Πως ονομάζεται αυτό το Σχήμα;");
listOfQuestions.Add("Ποιά είναι αυτή η νότα;");
listOfQuestions.Add("Από ποιο μουσικό όργανο είναι αυτός ο ήχος;");
listOfQuestions.Add("Σε ποια κατηγορία οργάνων ανήκουν τα Ντραμς;");
```

Σχήμα 9.2 Εισαγωγή ερωτήσεων.

#### 9.3.1 Έλεγχος απάντησης

Για τον έλεγχο της απάντησης καλείται η μέθοδος CheckAnswer() που, αρχικά, βρίσκει ποια ερώτηση είναι, και έπειτα ελέγχει, αν το κουμπί που πατήθηκε, είναι αυτό που έχουμε ορίσει σαν σωστή απάντηση για την συγκεκριμένη ερώτηση. Αν είναι σωστή, αλλάζει το χρώμα του πλαισίου της απάντησης σε πράσινο, κάνει το score +1 και στο τέλος αυτής της λειτουργίας καλεί την μέθοδο nextQuestion() που ξεκινάει την διαδικασία για την επόμενη ερώτηση, όπως φαίνεται στο σχήμα 9.3.

```
case 19:
    if (btn == btnThree.GetComponent<UnityEngine.UI.Button>())
    {
        questionIcon20.GetComponent<UnityEngine.UI.Image>().color = Color.green;
        btn.GetComponent<UnityEngine.UI.Image>().color = new Color(0f, 1f, 0f, 1f);
        scoreValue++;
        scoreText.GetComponent<TextMeshProUGUI>().text = "";
        scoreText.GetComponent<TextMeshProUGUI>().text += scoreValue.ToString();
    }
    else
    {
        questionIcon20.GetComponent<UnityEngine.UI.Image>().color = Color.red;
        btn.GetComponent<UnityEngine.UI.Image>().color = new Color(1f, 0f, 0f, 1f);
    }
    break;
}
Invoke("NextQuestion", 1f);
}
```

Σχήμα 9.3 Function CheckAnswer ().

### 9.3.2 Μέθοδος nextQuestion()

Η μέθοδος nextQuestion() είναι αυτή που αναλαμβάνει να εμφανίσει την επόμενη ερώτηση. Σε πρώτο επίπεδο, ελέγχει με την παράμετρο questionCounter αν η ερώτηση είναι μικρότερη ή ίση του 19, δηλαδή από το μέγεθος της λίστας. Έπειτα, κάνει reset τα χρώματα του προηγούμενου πλαισίου και το βάζει στο default color. Επίσης, το μεγάλο πλαίσιο εμφανίζει το κείμενο της επόμενης ερώτησης. Και τέλος, τοποθετεί στα κουμπιά των απαντήσεων τα αντίστοιχα κείμενα. Το σχήμα 9.4 αναφέρεται στη μέθοδο nextQuestion().

```
if (questionCounter <= 19)
{
    //reset colors στα buttons
    btnOne.GetComponent<UnityEngine.UI.Image>.color = new Color(224f / 255f, 160f / 255f, 0f / 255f, 1f);
    btnTwo.GetComponent<UnityEngine.UI.Image>.color = new Color(224f / 255f, 160f / 255f, 0f / 255f, 1f);
    btnThree.GetComponent<UnityEngine.UI.Image>.color = new Color(224f / 255f, 160f / 255f, 0f / 255f, 1f);
    btnFour.GetComponent<UnityEngine.UI.Image>.color = new Color(224f / 255f, 160f / 255f, 0f / 255f, 1f);

    //αρχικοποίηση του title και τα buttons
    questionTitle.GetComponent<TextMeshProUGUI>.text = questionCounter.ToString() + " " + " " + listOfQuestions[questionCounter].ToString();
    btnOne.transform.GetChild(1).GetComponent<TextMeshProUGUI>.text = btnOneAnswers[questionCounter].ToString();
    btnTwo.transform.GetChild(1).GetComponent<TextMeshProUGUI>.text = btnTwoAnswers[questionCounter].ToString();
    btnThree.transform.GetChild(1).GetComponent<TextMeshProUGUI>.text = btnThreeAnswers[questionCounter].ToString();
    btnFour.transform.GetChild(1).GetComponent<TextMeshProUGUI>.text = btnFourAnswers[questionCounter].ToString();
}
```

Σχήμα 9.4 Function nextQuestion().

## 9.4 Οθόνη νίκης

Εν τέλει, αν ο έλεγχος της λίστας είναι ίσος με 0, αυτό σημαίνει ότι το παιχνίδι έχει οδηγηθεί στο τέλος του και εκεί πρέπει να εμφανιστεί μια οθόνη νίκης. Ακόμη, πρέπει να εμφανιστεί το score και να μπορέσει ο χρήστης να πλοηγηθεί στην συνέχεια. Το παρακάτω σχήμα 9.5 παρουσιάζει την οθόνη νίκης.



Σχήμα 9.5 Οθόνη νίκης.

## 9.5 Επίλογος

Το κεφάλαιο αυτό ήταν διαφορετικό από τα προηγούμενα,. Αυτό έγκειται στο γεγονός πως δημιουργήθηκε ένα διαφορετικό είδος παιχνιδιού από τα προηγούμενα. Αρχικά, ήταν ένα παιχνίδι μίξης και των δυο μαθημάτων, ώστε ο μαθητής να μπορεί να ελέγξει τις γνώσεις που έχει αποκομίσει. Επιπλέον, η διαφορετικότητά του στηρίζεται στο γεγονός πως υπήρχε ποικιλία στις ερωτήσεις είτε γραπτά, είτε ηχητικά, είτε και οπτικά. Σαν αποτέλεσμα, η διαδραστικότητα μεταξύ των μαθητών θα μπορούσε να είναι μεγάλη.



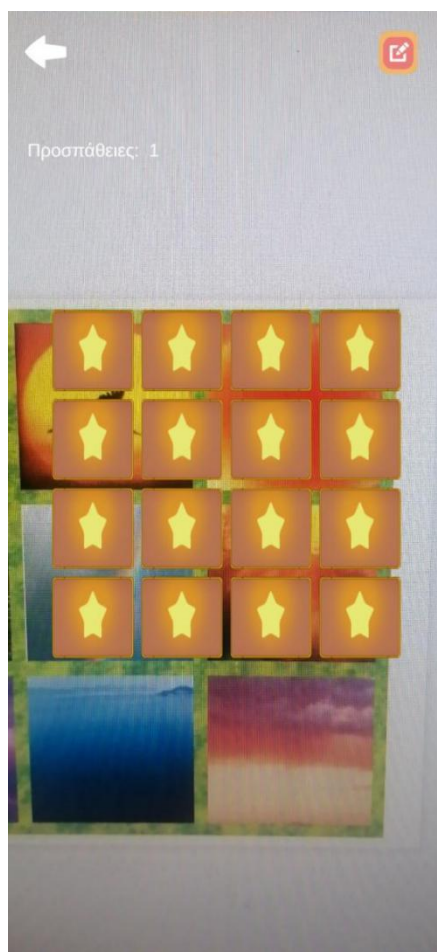
## Κεφάλαιο 10ο: Παιχνίδι μνήμης

### 10.1 Εισαγωγή

Σε αυτό το κεφάλαιο το παιχνίδι που θα δημιουργηθεί είναι ένα ακόμα παιχνίδι από εκείνα, τα οποία αφορούν και τα δυο μαθήματα της Α΄ και Β΄ τάξης του Δημοτικού. Στο παιχνίδι αυτό ο μαθητής θα παρακολουθήσει διάφορα σχήματα, χρώματα, και νότες, τα οποία θα προσπαθήσει να τα ταιριάξει. Το παρόν παιχνίδι περιέχει 18 κάρτες και από πίσω του 9 διαφορετικές εικόνες. Για κάθε λανθασμένη απάντηση υπάρχει ένας μετρητής, ο οποίος υπολογίζει πόσες προσπάθειες έχει κάνει ο μαθητής. Η συγκεκριμένη σκηνή κάνει χρήση της AR επαυξημένης πραγματικότητας. Έτσι, ο χρήστης σκανάρει μια εικόνα και στη συνέχεια του εμφανίζονται οι κάρτες.

### 10.2 UI

Όλο το UI του παιχνιδιού έχει δημιουργηθεί στο photoshop. Υπάρχει μια εικόνα που χρησιμοποιείται σαν background και στις 18 κάρτες, όπως επίσης και διάφορα images σαν βασικές εικόνες μέσα. Πάνω αριστερά υπάρχει το score τύπου, που είναι απλώς ένα textMeshPro. Έπειτα, πιο πάνω αριστερά υπάρχει ένα κουμπί, το οποίο σε μεταφέρει στην αρχική οθόνη και πάνω δεξιά βρίσκεται ένα κουμπί, στο οποίο τη στιγμή που πραγματοποιείται το κλικ πάνω του, ανοίγει ένα frame, το οποίο υποδεικνύει στον μαθητή ποια εικόνα πρέπει να σκανάρει, όπως παρουσιάζει το σχήμα 10.1.



Σχήμα 10.1 UI παιχνίδι μνήμης



Σχήμα 10.2 Εικόνα σκαναρίσματος.

### 10.3 Script

Αρχικά, στην awake μέθοδο πρέπει να γίνουν instantiate οι κάρτες σε ένα game object που θα έχει component με όνομα grid, για να δημιουργηθούν οι κάρτες πρέπει να είναι prefab. Η μέθοδος awake παρουσιάζεται στο σχήμα 10.3.

```
[SerializeField]
private Transform puzzleField;
[SerializeField]
private GameObject puzzleBtn;

@ Unity Message | 0 references
private void Awake()
{
    for (int i = 0; i < 16; i++)
    {
        GameObject btn = Instantiate(puzzleBtn);
        btn.name = "" + i;
        btn.transform.SetParent(puzzleField, false);
    }
}
```

Σχήμα 10.3 Δημιουργία καρτών.

Έπειτα, σε μια μέθοδο με όνομα getButtons() πρέπει να βρεθούν τα buttons και να εισαχθούν τα κατάλληλα images, όπως υποδεικνύεται στο σχήμα 10.4.

```
1 reference
void getButtons()
{
    GameObject[] objects = GameObject.FindGameObjectsWithTag("PuzzleButton");

    for(int i=0; i< objects.Length; i++)
    {
        btns.Add(objects[i].GetComponent<Button>());
        btns[i].image.sprite = bgImage;
    }
}
```

Σχήμα 10.4 Function getButtons().

Ύστερα, πρέπει να εισαχθεί στην κάθε κάρτα ένα event τύπου on click. Εκεί θα γίνεται κατανοητό, αν ο χρήστης έχει πατήσει την πρώτη εικόνα ή την δεύτερη και θα περνάει την απάντηση σε μια λίστα, ώστε στο τέλος αυτής της μεθόδου να καλείται η μέθοδος CheckIfThePuzzlesMatch(), η οποία θα κάνει τον έλεγχο για το αν ταιριάζουν. Τέλος, αυξάνει το countGuesses για να δείξει στον χρήστη στο τέλος του παιχνιδιού πόσες προσπάθειες πραγματοποιήσε, όπως παρουσιάζει το σχήμα 10.5.

```
1 reference
public void pickAPuzzle()
{
    if (!firstGuess)
    {
        firstGuess = true;

        firstGuessIndex = int.Parse(UnityEngine.EventSystems.EventSystem.current.currentSelectedGameObject.name);
        firstGuessPuzzle = gamePuzzles[firstGuessIndex].name;

        btns[firstGuessIndex].image.sprite = gamePuzzles[firstGuessIndex];
    }
    else if (!secondGuess)
    {
        secondGuess = true;

        secondGuessIndex = int.Parse(UnityEngine.EventSystems.EventSystem.current.currentSelectedGameObject.name);
        secondGuessPuzzle = gamePuzzles[secondGuessIndex].name;

        btns[secondGuessIndex].image.sprite = gamePuzzles[secondGuessIndex];

        countGuesses++;

        titleValue.GetComponent<TextMeshProUGUI>().text = countGuesses.ToString();
        StartCoroutine(CheckIfThePuzzlesMatch());
    }
}
```

Σχήμα 10.5 Event pickAPuzzle().

### 10.3.1 Έλεγχος δυο καρτών

Βλέποντας στο τέλος της μεθόδου pickAPuzzle καλείται η μέθοδος CheckIfThePuzzlesMatch(). Η μέθοδος αυτή είναι επιφορτισμένη με τον έλεγχο μεταξύ των δυο καρτών. Πρώτα, ελέγχει αν η πρώτη επιλογή ταιριάζει με την δεύτερη. Αν ο έλεγχος είναι αληθής, κάνει τα δύο κουμπιά να μην μπορούν να πατηθούν ξανά και τα εξαφανίζει. Τέλος, καλεί την μέθοδο CheckIfTheGameIsFinished(). Αν ο πρώτος έλεγχος δεν είναι αληθής, τότε επιστρέφει τις κάρτες στο αρχικό τους state, όπως γίνεται φανερό στο σχήμα 10.6.

```
1 reference
IEnumerator CheckIfThePuzzlesMatch()
{
    yield return new WaitForSeconds(0.5f);

    if (firstGuessPuzzle == secondGuessPuzzle)
    {
        yield return new WaitForSeconds(.2f);

        btns[firstGuessIndex].interactable = false;
        btns[secondGuessIndex].interactable = false;

        btns[firstGuessIndex].image.color = new Color(0,0,0,0);
        btns[secondGuessIndex].image.color = new Color(0,0,0,0);

        CheckIfTheGameIsFinished();
    }
    else
    {
        yield return new WaitForSeconds(.2f);

        btns[firstGuessIndex].image.sprite = bgImage;
        btns[secondGuessIndex].image.sprite = bgImage;
    }

    yield return new WaitForSeconds(.2f);

    firstGuess = secondGuess = false;
}
```

Σχήμα 10.6 Μέθοδος για έλεγχο δύο καρτών.

### 10.3.2 Μέθοδος CheckIfTheGameIsFinished() και οθόνη νίκης

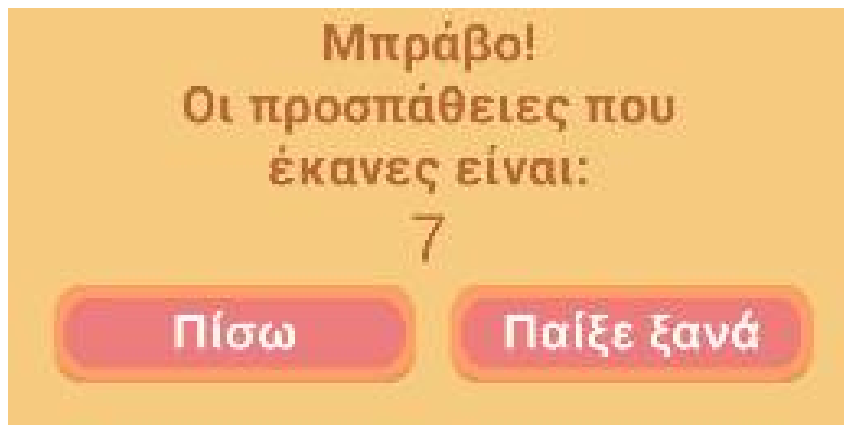
Στο τέλος της μεθόδου CheckIfThePuzzlesMatch(), όπως ήδη ειπώθηκε, καλείται η μέθοδος CheckIfTheGameIsFinished η οποία ελέγχει αν το παιχνίδι τελείωσε. Αυτό πραγματοποιείται ελέγχοντας αν η παράμετρος CorrectCountGuesses είναι ίση με την gameGuesses. Στην περίπτωση που είναι αληθής, ενεργοποιείται η οθόνη νίκης και δείχνουμε τον αριθμό των προσπαθειών. Η μέθοδος CheckIfThePuzzlesMatch() παρουσιάζεται στο σχήμα 10.7.

```
1 reference
void CheckIfTheGameIsFinished()
{
    CorrectCountGuesses++;

    if (CorrectCountGuesses == gameGuesses)
    {
        winningPanel.SetActive(true);
        winningScore.GetComponent<TextMeshProUGUI>().text = countGuesses.ToString();
    }
}
```

Σχήμα 10.7 Function CheckIfThePuzzlesMatch().

Στην οθόνη νίκης ο χρήστης μπορεί να παρακολουθήσει το score του και μπορεί να επιλέξει αν θέλει να παίξει ξανά ή αν θέλει να επιστρέψει στην αρχική οθόνη και να επιλέξει κάποιο άλλο παιχνίδι. Το σχήμα 10.8. αναδεικνύει την οθόνη νίκης.



Σχήμα 10.8 Οθόνη νίκης.

## 10.4 Επίλογος

Αυτό το κεφάλαιο αναφέρθηκε σε ένα παιχνίδι μίξης μεταξύ των δυο μαθημάτων, εμπνευσμένο από το τετράδιο εργασιών των "Εικαστικών" της Α' & Β' τάξης του Δημοτικού στην ενότητα Γ3 και με τίτλο "Αιωρούμενα παιχνίδια". Το παιχνίδι μνήμης είναι ένα ψυχαγωγικό παιχνίδι, όπου ο μαθητής πρέπει να ταιριάξει τις κρυφές εικόνες μεταξύ τους με στόχο να πραγματοποιήσει όσο λίγες προσπάθειες μπορεί. Τέλος, αξίζει να αναφερθεί πως έγινε χρήση της επαυξημένης πραγματικότητας, ώστε το παιχνίδι να οδηγηθεί σε πιο διασκεδαστικά αποτελέσματα.



## Κεφάλαιο 11ο: Κρυπτόλεξο

### 11.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα δημιουργηθεί ένα παιχνίδι - κρυπτόλεξο. Αυτό το παιχνίδι αφορά και τα δύο μαθήματα, αφού ο χρήστης θα προσπαθήσει να ανακαλύψει τις κρυμμένες λέξεις μέσα από ένα σύνολο γραμμάτων. Οι λέξεις, που θα είναι κρυμμένες, αφορούν και το μάθημα των “Εικαστικών” αλλά και το μάθημα της “Μουσικής”. Το παρόν παιχνίδι περιέχει 9 στήλες οριζόντια και 9 στήλες κάθετα, το οποίο σημαίνει ότι θα έχουμε 81 γράμματα στο σύνολο. Ο στόχος του χρήστη είναι να εντοπίσει όλες τις λέξεις πριν λήξει ο χρόνος.

### 11.2 UI

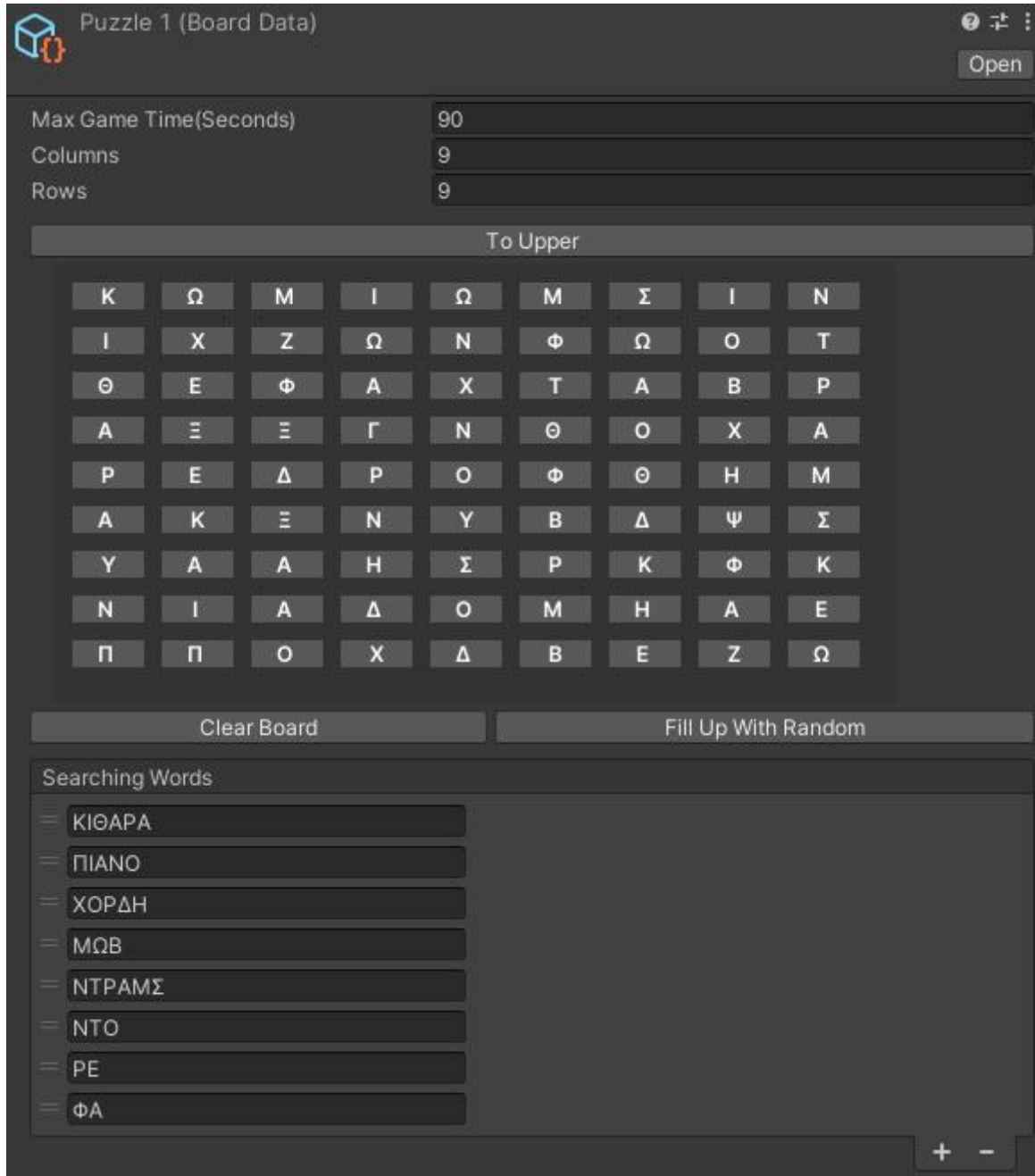
Όλο το UI του παιχνιδιού έχει δημιουργηθεί στο photoshop. Έχουν δημιουργηθεί τρία διαφορετικά images για τα γράμματα για το ανάλογο state που βρίσκονται (normal-correct-incorrect). Τα γράμματα και ο έλεγχος των λέξεων βασίζονται στην ελληνική αλφάβητο. Επίσης, πάνω στα αριστερά υπάρχει ένα κουμπί με το οποίο ο χρήστης μπορεί να επιστρέψει στην αρχική οθόνη. Πάνω στην μέση υπάρχει ένα πλαίσιο στο οποίο θα εμφανίζεται ο χρόνος που απομένει στον χρήστη. Τέλος, κάτω από το κρυπτόλεξο εμφανίζονται τόσα πλαίσια όσες και οι κρυμμένες λέξεις. Το κάθε πλαίσιο περιέχει μια λέξη μέσα του, που είναι κρυμμένη, και μόλις την ανακαλύπτει ο χρήστης, διαγράφεται. Το τελικό UI του παιχνιδιού παρουσιάζεται στο σχήμα 11.1.



Σχήμα 11.1 UI κρυπτολεξου.

### 11.3 Script

Στο παρόν παιχνίδι προς διευκόλυνση έχει δημιουργηθεί ένας custom editor στην unity. Οπότε αρκετές φορές εισάγονται δεδομένα από εκεί, π.χ. ο αριθμός των στηλών, οι λέξεις, και η τοποθεσία. Ο custom editor φαίνεται στο κάτωθι σχήμα 11.2.



Σχήμα 11.2 Custom editor.

### 11.3.1 Spawn grid

Αρχικά, το πρώτο πράγμα που γίνεται μέσω των scripts είναι το spawn του grid και η εισαγωγή των images μαζί με τα γράμματα πάνω τους. Αυτό πραγματοποιείται με την μέθοδο spawnGridSquares() που, σε πρώτο στάδιο, παίρνει τα δεδομένα από τον custom editor(rows, columns) και τις δημιουργεί. Έπειτα, εισάγει random γράμματα σε κάθε ένα κελί εκτός από εκείνα που έχουμε προκαθορίσει ότι θα είναι οι λέξεις που έχουμε ορίσει. Οι δύο μέθοδοι γίνονται φανερές στο σχήμα 11.3.

```
1 reference
private void spawnGridSquares()
{
    if (currentGameData != null)
    {
        var squareScale = GetSquareScale(new Vector3(1.5f, 1.5f, 0.1f));
        foreach (var squares in currentGameData.selectedBoardData.Board)
        {
            foreach (var squareLetter in squares.Row)
            {
                var normalLetterData = alphabetData.AlphabetNormal.Find(data => data.letter == squareLetter);
                var selectedLetterData = alphabetData.AlphabetHighlighted.Find(data => data.letter == squareLetter);
                var correctLetterData = alphabetData.AlphabetWrong.Find(data => data.letter == squareLetter);

                if (normalLetterData.image == null || selectedLetterData.image == null)
                {
                    Debug.LogError("error 1");

                    #if UNITY_EDITOR
                    if (UnityEditor.EditorApplication.isPlaying)
                    {
                        UnityEditor.EditorApplication.isPlaying = false;
                    }
                    #endif
                }
                else
                {
                    _squareList.Add(Instantiate(gridSquarePrefab));
                    _squareList[_squareList.Count - 1].GetComponent<GridSquare>().SetSprite(normalLetterData, correctLetterData, selectedLetterData);
                    _squareList[_squareList.Count - 1].transform.SetParent(this.transform);
                    _squareList[_squareList.Count - 1].GetComponent<Transform>().position = new Vector3(0f, 0f, 0f);
                    _squareList[_squareList.Count - 1].transform.localScale = squareScale;
                    _squareList[_squareList.Count - 1].GetComponent<GridSquare>().SetIndex(_squareList.Count - 1);
                }
            }
        }
    }
}

1 reference
private Vector3 GetSquareScale(Vector3 defaultScale)
{
    var finalScale = defaultScale;
    var adjustment = 0.01f;

    while (shouldScaleDown(finalScale))
    {
        finalScale.x -= adjustment;
        finalScale.y -= adjustment;

        if (finalScale.x <= 0 || finalScale.y <= 0)
        {
            finalScale.x = -adjustment;
            finalScale.y = -adjustment;
            return finalScale;
        }
    }
    return finalScale;
}
```

Σχήμα 11.3 Function spawnGridSquares().

### 11.3.2 Μέθοδος Drug

Ο χρήστης μπορεί να εντοπίσει τις λέξεις είτε αριστερά - δεξιά, είτε πάνω - κάτω, είτε διαγώνια. Για να είναι αντιληπτό σε ποιο σημείο ο χρήστης σέρνει το δάχτυλό του, χρησιμοποιήθηκε η τεχνική raycast. Από την στιγμή που ο χρήστης ακουμπήσει το δάχτυλό του, δημιουργείται μια νοητή γραμμή, η οποία τον ακολουθεί όπου και αν κατευθύνεται. Στο αποτέλεσμα αυτό μας οδηγεί η μέθοδος selectRay(), όπως φαίνεται στο σχήμα 11.4.

```
1 reference
private Ray SelectRay(Vector2 firstPosition, Vector2 secondPosition)
{
    var direction = (secondPosition - firstPosition).normalized;
    float tolerance = 0.01f;

    if (Math.Abs(direction.x) < tolerance && Math.Abs(direction.y - 1f) < tolerance)
    {
        return _rayUp;
    }

    if (Math.Abs(direction.x) < tolerance && Math.Abs(direction.y - (-1f)) < tolerance)
    {
        return _rayDown;
    }

    if (Math.Abs(direction.x - (-1f)) < tolerance && Math.Abs(direction.y - 1f) < tolerance)
    {
        return _rayLeft;
    }

    if (Math.Abs(direction.x - 1f) < tolerance && Math.Abs(direction.y - 1f) < tolerance)
    {
        return _rayRight;
    }

    if (direction.x < 0f && direction.y > 0f)
    {
        return _rayDiagonalLeftUp;
    }

    if (direction.x < 0f && direction.y < 0f)
    {
        return _rayDiagonalLeftDown;
    }

    if (direction.x > 0f && direction.y > 0f)
    {
        return _rayDiagonalRightUp;
    }

    if (direction.x > 0f && direction.y < 0f)
    {
        return _rayDiagonalRightDown;
    }

    return _rayDown;
}
```

Σχήμα 11.4 Function selectRay().

Πριν όμως “ζωγραφιστεί” το raycast, πρέπει να γίνει κατανοητό σε ποιο tile πάτησε ο χρήστης. Το κομμάτι αυτό το αναλαμβάνει η μέθοδος squareSelected, που αρχικά μόλις ο χρήστης πατήσει σε κάποιο tile, κατευθείαν δημιουργεί το raycast προς όλες τις κατευθύνσεις που μπορεί να κινηθεί, δηλαδή πάνω - κάτω, αριστερά - δεξιά και διαγώνια, όπως ακριβώς φαίνεται και στο σχήμα 11.5. Κάθε φορά όμως που περνάει από ένα γραμμάκι, το γραμμάκι αυτό κρατείται κατευθείαν σε έναν πίνακα, ώστε να μπορεί να ελεγχθεί αργότερα.

```

if (_assignedPoints == 0)
{
    _rayStartPosition = squarePosition;
    _correctSquareList.Add(squareIndex);
    _word += letter;

    _rayUp = new Ray(new Vector2 (squarePosition.x, squarePosition.y), new Vector2(0f, 1));
    _rayDown = new Ray(new Vector2(squarePosition.x, squarePosition.y), new Vector2(0f, -1));
    _rayLeft = new Ray(new Vector2(squarePosition.x, squarePosition.y), new Vector2(-1, 0f));
    _rayRight = new Ray(new Vector2(squarePosition.x, squarePosition.y), new Vector2(1, 0f));
    _rayDiagonalLeftUp = new Ray(new Vector2(squarePosition.x, squarePosition.y), new Vector2(-1, 1));
    _rayDiagonalLeftDown = new Ray(new Vector2(squarePosition.x, squarePosition.y), new Vector2(-1, -1));
    _rayDiagonalRightUp = new Ray(new Vector2(squarePosition.x, squarePosition.y), new Vector2(1, 1));
    _rayDiagonalRightDown = new Ray(new Vector2(squarePosition.x, squarePosition.y), new Vector2(1, -1));
}
else if(_assignedPoints == 1)
{
    _correctSquareList.Add(squareIndex);
    _currentRay = SelectRay(_rayStartPosition, squarePosition);
    GameEvents.SelectSquareMethod(squarePosition);
    _word += letter;
    CheckWord();
}
}

```

Σχήμα 11.5 Function squareSelected().

### 11.3.3 Μέθοδος checkWord

Αφού ο χρήστης αφήσει το δάχτυλό του από την οθόνη του κινητού του, καλείται η μέθοδος checkWord(). Πρώτα, αφού καλείται η μέθοδος, εκτελείται μια for loop για όλες τις λέξεις που έχουν εισαχθεί στον custom editor. Αν η λέξη που έχει πατήσει ο χρήστης υπάρχει σε αυτή την λίστα, τότε ο έλεγχος είναι αληθής και τότε εκτελούνται κάποιες λειτουργίες. Πρώτα αυξάνει τον αριθμό της παραμέτρου correctedWords, μετά καλείται το event CorrectWordMethod() που αλλάζει το background image από λευκό σε πράσινο, το οποίο σημαίνει ότι η λέξη είναι σωστή, διαγράφει την λέξη από τα πλαίσια του κάτω UI και τέλος, καλείται η μέθοδος CheckBoardCompleted(), όπως φαίνεται στο σχήμα 11.6.

```

2 references
private void CheckWord()
{
    foreach(var serchingWord in currentGameData.selectedBoardData.SearchWords)
    {
        Debug.Log("autes pou caxnoume" + "" +serchingWord.Word);
        Debug.Log("auth pou exoume" + "" + _word);
        if (_word == serchingWord.Word)
        {
            GameEvents.CorrectWordMethod(_word, _correctSquareList);
            _completedWords++;
            _word = string.Empty;
            _correctSquareList.Clear();
            CheckBoardCompleted();
            return;
        }
    }
}

```

Σχήμα 11.6 Function checkWord().

### 11.3.4 Μέθοδος CheckBoardCompleted

Στην μέθοδο που φαίνεται στο σχήμα 11.7 γίνεται απλώς ο έλεγχος για το αν η παράμετρος `completedWords`, που αυξανόταν προηγουμένως, είναι ίση με τον αριθμό και με τις λέξεις που έχουν οριστεί στον custom editor. Αν αυτός ο έλεγχος είναι αληθής, τότε καλείται το event `BoardCompletedMethod`, που αναλαμβάνει να κλείσει τον timer και να εμφανίσει την οθόνη νίκης.

```
1 reference
private void CheckBoardCompleted()
{
    if (currentGameData.selectedBoardData.SearchWords.Count == _completedWords)
    {
        GameEvents.BoardCompletedMethod();
    }
}
```

Σχήμα 11.7 Function `CheckBoardCompleted()`.

### 11.3.5 Timer

Ο timer στο παρόν παιχνίδι παίζει σημαντικό ρόλο, εφόσον ο χρήστης πρέπει να βρει όλες τις λέξεις μέσα σε αυτόν χρόνο, προκειμένου να κερδίσει το παιχνίδι. Ο timer εμφανίζεται απλώς σε ένα `textMeshPro` στην πάνω πλευρά της οθόνης. Ο χρόνος του παιχνιδιού είναι προκαθορισμένος στα 90 δευτερόλεπτα. Ο χρόνος μειώνεται σε μια `update` μέθοδο, κάθε φορά αν ο `_stopTimer` είναι `false` μειώνουμε `-time.deltaTime` από την μεταβλητή. Το `time.deltaTime` σημαίνει ότι “τρέχει” σε κάθε frame που ισοδυναμεί με τον πραγματικό χρόνο. Η μέθοδος υποδεικνύεται στο σχήμα 11.8.

```
Unity Message | 0 references
private void Update()
{
    if (_stopTimer == false)
    {
        _timeLeft -= Time.deltaTime;
    }

    if (_timeLeft <= 0)
    {
        _oneSecondDown = _timeLeft - 1f;
    }
}
```

Σχήμα 11.8 Function `Timer()`.

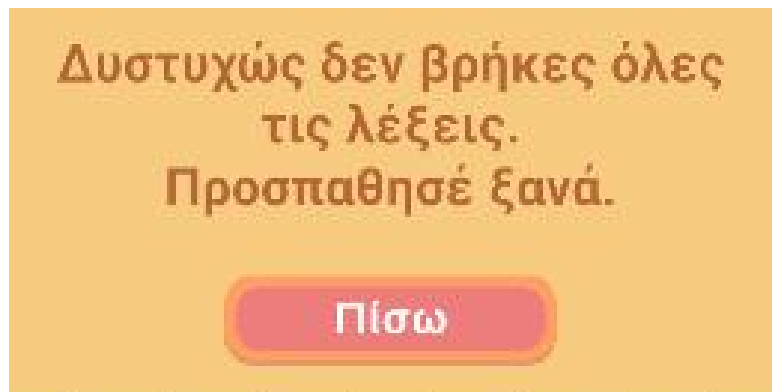
## 11.4 Οθόνη νίκης

Τέλος, αφού έχει καλεστεί το event `boardCompletedMethod()`, πρέπει να εμφανισθεί η οθόνη νίκης που σηματοδοτεί το τέλος του παιχνιδιού και ο χρήστης πρέπει να επιστρέψει στην αρχική οθόνη. Η οθόνη νίκης εμφανίζεται στο σχήμα 11.9.



Σχήμα 11.9 Οθόνη νίκης.

Σε αυτό το παιχνίδι υπάρχει μια ιδιαιτερότητα. Αν ο χρήστης δεν προλάβει να ανακαλύψει όλες τις λέξεις μέσα στο χρόνο, εμφανίζεται μια οθόνη ήττας, η οποία ενημερώνει τον χρήστη πως έχασε και στη συνέχεια, πλοηγείται στο αρχικό μενού. Το σχήμα 11.10 εμφανίζει την οθόνη ήττας.



Σχήμα 11.10 Οθόνη ήττας.

## 11.5 Επίλογος

Σε αυτό το κεφάλαιο παρουσιάστηκε ένα παιχνίδι μίξης και των δύο μαθημάτων. Το παρόν παιχνίδι έκανε χρήση δισδιάστατων στοιχείων. Για τη λειτουργία του παιχνιδιού δημιουργήθηκαν στοιχεία από το πρόγραμμα Photoshop. Το κρυπτόλεξο είναι ένα διασκεδαστικό παιχνίδι και έχει την ιδιαιτερότητα του μετρητή του χρόνου. Ο χρήστης πρέπει να εντοπίσει όλες τις κρυμμένες λέξεις προτού λήξει ο χρόνος. Τέλος, εμφανίζεται μια οθόνη νίκης ή οθόνη ήττας ανάλογα με το αν ο χρήστης κέρδισε ή έχασε.



## Κεφάλαιο 12ο: Κρεμάλα

### 12.1 Εισαγωγή

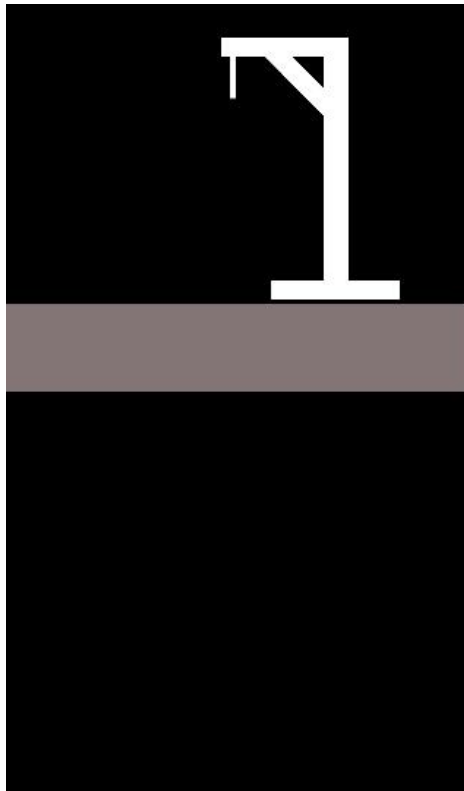
Σε αυτό το κεφάλαιο θα δημιουργηθεί ένα διάσημο παιχνίδι, η κρεμάλα. Το συγκεκριμένο παιχνίδι σχετίζεται και αυτό με τα δυο μαθήματα, αφού ο χρήστης θα προσπαθήσει να ανακαλύψει τις τυχαίες λέξεις ή τις λέξεις που έχουν εισαχθεί χειροκίνητα. Οι λέξεις που θα είναι κρυμμένες αφορούν και το μάθημα των “Εικαστικών” αλλά και το μάθημα της “Μουσικής”. Το ιδιαίτερο στοιχείο αυτού του παιχνιδιού είναι ότι οι δάσκαλοι και οι μαθητές μπορούν να το παίξουν μαζί. Ο δάσκαλος του αντίστοιχου μαθήματος μπορεί να εισάγει όποια λέξη επιθυμεί σχετικά με το μάθημα και ίσως να δώσει κάποια στοιχεία, ώστε να βρεθεί η λέξη. Το συγκεκριμένο παιχνίδι κάνει χρήση της AR επαυξημένης πραγματικότητας. Ο χρήστης σκανάρει μια εικόνα και τότε, το UI της σκηνής ξεκινάει την λειτουργία του.

### 12.2 UI

Σχεδόν όλο το UI του παιχνιδιού έχει δημιουργηθεί στο unity εκτός από κάποια κομμάτια που δημιουργήθηκαν στο photoshop. Τα κομμάτια αυτά είναι η οθόνη νίκης, η οθόνη ήττας, όπως επίσης και τα κουμπιά από το εισαγωγικό μενού. Τα γράμματα και ο έλεγχος των λέξεων βασίζονται στην ελληνική αλφάβητο. Στο πάνω μέρος και δεξιά υπάρχει ένα κουμπί που, όταν γίνεται κλικ πάνω του, ανοίγει ένα frame, το οποίο δείχνει στον μαθητή ποια εικόνα πρέπει να σκανάρει. Τέλος, στο πάνω μέρος και αριστερά υπάρχει ένα κουμπί, με το οποίο ο χρήστης μπορεί να επιστρέψει στην αρχική οθόνη. Η εικόνα σκαναρίσματος και το τελικό UI του παιχνιδιού παρουσιάζεται στα σχήματα 12.1 και 12.2 αντίστοιχως.



Σχήμα 12.1 Εικόνα σκαναρίσματος.



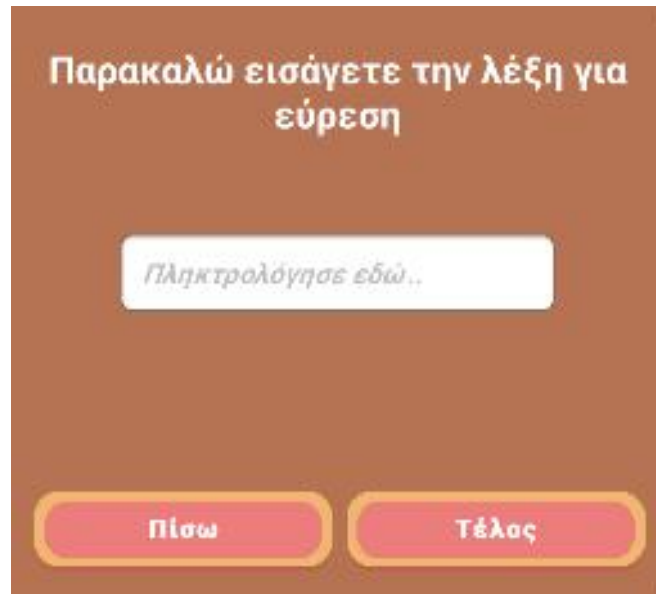
Σχήμα 12.2 UI κρεμάλας.

### 12.3 Script

Το παιχνίδι ξεκινάει, αφού ο χρήστης σκανάρει την εικόνα. Τότε εμφανίζεται ένα frame, όπου ο μαθητής πρέπει να διαλέξει, αν θα παίξει με την τυχαία λέξη ή αν προτιμάει κάποιος άλλος να κάνει εισαγωγή της λέξης. Αν αποφασισθεί η πρώτη επιλογή, τότε το παιχνίδι ξεκινάει απευθείας. Ειδιάλλως ανοίγει ένα άλλο frame, που εκεί υπάρχει ένα input field για αυτόν που θα εισάγει την λέξη, όπως δείχνουν τα σχήματα 12.3 και 12.4.



Σχήμα 12.3 Επιλογή mode παιχνιδιού.



Σχήμα 12.4 Εισαγωγή λέξης.

### 12.3.1 Τυχαία λέξη

Αν ο χρήστης επιλέξει το mode τυχαία λέξη, τότε καλείται η μέθοδος `InitialiseGame()`. Η μέθοδος αυτή, αρχικά, μηδενίζει για σιγουριά τον αριθμό των σωστών guesses και των λανθασμένων. Έπειτα, εμφανίζει όλα τα buttons για τα γράμματα που ο χρήστης μπορεί να επιλέξει. Ύστερα, βάζει σε τιμή `false` όλες τις εικόνες της κρεμάλας. Τέλος, επιλέγει μια τυχαία λέξη καλώντας την μέθοδο `GenerateWord()` και εμφανίζει τα αντίστοιχα πλακίδια των γραμμάτων από την τυχαία λέξη, όπως φανερώνει το σχήμα 12.5.

```

1 reference
public void InitialiseGame()
{
    incorrectGuesses = 0;
    correctGuesses = 0;
    foreach (Button child in keyboardContainer.GetComponentInChildren<Button>())
    {
        child.interactable = true;
    }
    foreach (Transform child in wordContainer.GetComponentInChildren<Transform>())
    {
        //if (child != wordContainer) Destroy(child.gameObject);
    }
    foreach (GameObject stage in hangmanStage)
    {
        stage.SetActive(false);
    }

    word = GenerateWord();

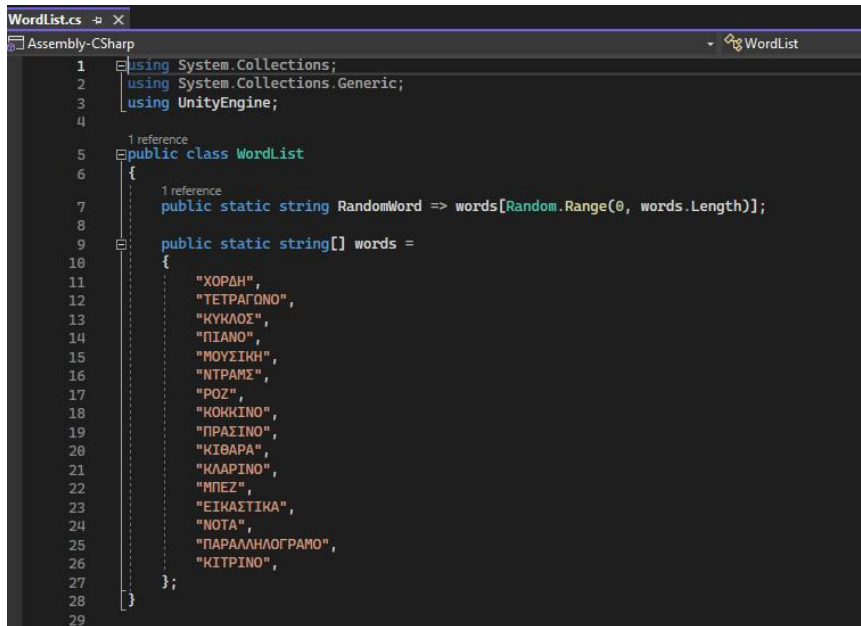
    foreach (char letter in word)
    {
        Instantiate(letterContainer, wordContainer);
    }
}

```

Σχήμα 12.5 Function `InitialiseGame()`.

### 12.3.2 Μέθοδος GenerateWord()

Η μέθοδος GenerateWord() είναι πολύ απλή. Επί της ουσίας καλεί ένα άλλο script με όνομα “WordList”, στο οποίο υπάρχουν διάφορες λέξεις. Εκεί μέσα πραγματοποιείται η επιλογή μιας τυχαίας λέξης και την επιστρέφει πίσω στον game manager για να την εμφανίσουμε στο UI. Η μέθοδος GenerateWord() γίνεται αντιληπτή στο κάτωθι σχήμα 12.6.



```

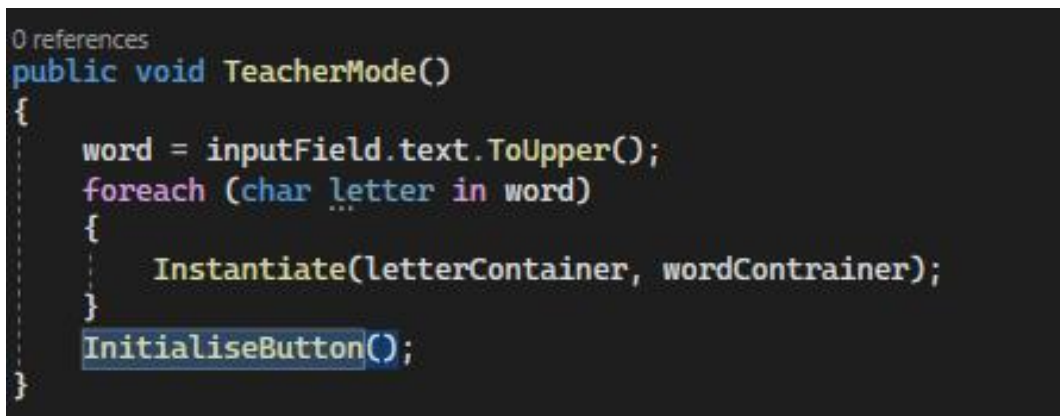
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class WordList
6 {
7     public static string RandomWord => words[Random.Range(0, words.Length)];
8
9     public static string[] words =
10     {
11         "ΧΟΡΔΗ",
12         "ΤΕΤΡΑΓΩΝΟ",
13         "ΚΥΚΛΟΣ",
14         "ΠΙΑΝΟ",
15         "ΜΟΥΣΙΚΗ",
16         "ΝΤΡΑΜΣ",
17         "ΡΟΖ",
18         "ΚΟΚΚΙΝΟ",
19         "ΠΡΑΣΙΝΟ",
20         "ΚΙΘΑΡΑ",
21         "ΚΛΑΡΙΝΟ",
22         "ΜΠΕΖ",
23         "ΕΙΚΑΣΤΙΚΑ",
24         "ΝΟΤΑ",
25         "ΠΑΡΑΛΛΗΛΟΓΡΑΜΟ",
26         "ΚΙΤΡΙΝΟ",
27     };
28 }
29

```

Σχήμα 12.6 Script WordList.

### 12.3.3 Εισαγωγή λέξης

Αν ο χρήστης επιλέξει το mode εισαγωγής λέξης, τότε καλείται η μέθοδος TeacherMode(), που φαίνεται στο σχήμα 12.7. Αρχικά, η μέθοδος αυτή κρατάει σε μια μεταβλητή την λέξη που έχει πάρει από το input field. Στη συνέχεια, για κάθε γράμμα από την λέξη δημιουργεί το αντίστοιχο πλακίδιο και εμφανίζει τα γράμματα, προκειμένου να επιλέξει ο μαθητής.



```

0 references
public void TeacherMode()
{
    word = inputField.text.ToUpper();
    foreach (char letter in word)
    {
        Instantiate(letterContainer, wordContainer);
    }
    InitialiseButton();
}

```

Σχήμα 12.7 Function TeacherMode().

### 12.3.4 Έλεγχος γραμμάτων

Κάθε φορά που ο χρήστης πατάει ένα γράμμα καλείται μια μέθοδος με όνομα “checkLetter”, όπως φαίνεται στο σχήμα 12.8., προκειμένου να ελέγξουμε αν το συγκεκριμένο γράμμα υπάρχει μέσα στην λέξη που έχει επιλεχθεί. Αρχικά, η μέθοδος αυτή εκτελεί μια for για όλα τα γράμματα των λέξεων και πραγματοποιεί τον έλεγχο στην περίπτωση που το γράμμα, που πατήθηκε, υπάρχει. Έπειτα αυξάνεται ο αριθμός των correctGuesses. Το κουμπί του γράμματος γίνεται μη λειτουργικό για να μην μπορεί να πατηθεί ξανά και τέλος, εμφανίζεται το γράμμα / γράμματα στα πλακίδια. Αν το γράμμα δεν υπάρχει στην λέξη, ο αριθμός των incorrectGuesses αυξάνεται και εμφανίζεται το stage της κρεμάλας, π.χ. το κεφάλι. Τέλος, αφού γίνουν όλοι αυτοί οι έλεγχοι, καλείται η μέθοδος checkOutcome(), για να γίνει αντιληπτό αν ο μαθητής κατάφερε να εντοπίσει ολόκληρη την λέξη.

```
1 reference
private void CheckLetter(char inputLetter)
{
    bool letterInWord = false;
    for (int i = 0; i < word.Length; i++)
    {
        if (inputLetter == word[i])
        {
            letterInWord = true;
            correctGuesses++;
            wordContainer.GetChild(i).GetComponentInChildren<TextMeshProUGUI>().SetText(inputLetter.ToString());
        }
    }
    if (letterInWord == false)
    {
        incorrectGuesses++;
        hangmanStage[incorrectGuesses - 1].SetActive(true);
    }
    CheckOutcome();
}
```

Σχήμα 12.8 Function CheckLetter().

### 12.3.5 Έλεγχος λέξης

Όπως είδαμε στο τέλος της προηγούμενης function, καλείται η μέθοδος checkOutcome(). Κάθε φορά που καλείται αυτή η μέθοδος γίνονται δύο έλεγχοι. Πρώτον, αν ο αριθμός των correctGuesses είναι ίσος με τον αριθμό των γραμμάτων της λέξης, αυτό σημαίνει ότι ο χρήστης βρήκε την λέξη και τότε η λέξη μετατρέπεται σε πράσινα γράμματα κάτω από την κρεμάλα, απενεργοποιείται το keyboard container και εμφανίζεται η οθόνη νίκης. Η επόμενη συνθήκη ελέγχει αν ο αριθμός των incorrectGuesses είναι ίσος με 6, δηλαδή όσα και τα stage της κρεμάλας (π.χ. κεφάλι, χέρια) τότε ο χρήστης δεν κατόρθωσε να ανακαλύψει την λέξη και έχασε. Όποτε εμφανίζεται η σωστή λέξη με κόκκινα γράμματα, απενεργοποιείται το keyboard container και εμφανίζεται η οθόνη ήττας. Η μέθοδος checkOutcome() φαίνεται στο σχήμα 12.9.

```
1 reference
private void CheckOutcome()
{
    if (correctGuesses == word.Length)
    {
        for (int i = 0; i < word.Length; i++)
        {
            wordContainer.GetChild(i).GetComponentInChildren<TextMeshProUGUI>().color = Color.green;
            keyboardContainer.SetActive(false);
            winPanel.SetActive(true);
        }
    }

    if (incorrectGuesses == 6)
    {
        for (int i = 0; i < word.Length; i++)
        {
            wordContainer.GetChild(i).GetComponentInChildren<TextMeshProUGUI>().color = Color.red;
            wordContainer.GetChild(i).GetComponentInChildren<TextMeshProUGUI>().text = word[i].ToString();
            keyboardContainer.SetActive(false);
            lossPanel.SetActive(true);
        }
    }
}
```

Σχήμα 12.9 Function CheckOutcome().

## 12.4 Οθόνη νίκης και οθόνη ήττας

Τέλος, αφού έχει γίνει αντιληπτό, αν ο χρήστης κέρδισε ή έχασε, εμφανίζονται τα αντίστοιχα UI για την οθόνη νίκης (σχήμα 12.10) και για την οθόνη ήττας (σχήμα 12.11), που σηματοδοτούν το τέλος του παιχνιδιού. Σε εκείνο το σημείο ο χρήστης επιλέγει αν θα παίξει ξανά ή αν θα επιστρέψει στην αρχική οθόνη. Το άνοιγμα των αντιστοίχως panel πραγματοποιείται με την εντολή `setActive(true)`.



Σχήμα 12.10 Οθόνη νίκης.



Σχήμα 12.11 Οθόνη ήττας.

## 12.5 Επίλογος

Σε αυτό το κεφάλαιο δημιουργήθηκε ακόμα ένα παιχνίδι μίξης και των δύο μαθημάτων. Το παρόν παιχνίδι έκανε χρήση των δισδιάστατων στοιχείων και της AR επαυξημένης πραγματικότητας. Το ιδιαίτερο στοιχείο αυτής της υλοποίησης έγκειται στο ότι ο δάσκαλος μπορεί να παίξει το παιχνίδι μαζί με τους μαθητές ή και μόνο οι μαθητές μεταξύ τους. Αυτό συνεπάγεται ότι θα υπάρχει μεγάλη διαδραστικότητα. Η εκμάθηση αυτών των δύο μαθημάτων γίνεται με έναν αρκετά διασκεδαστικό τρόπο και η ώρα του μαθήματος είναι ευχάριστη.



## Κεφάλαιο 13ο: Συμπεράσματα και προτάσεις βελτίωσης

Στο τέλος της πτυχιακής εργασίας, και αφού δημιουργήθηκε το παιχνίδι, προέκυψαν κάποια συμπεράσματα. Αναλυτικά:

- Χάρη στην τεχνολογία ο τρόπος διδασκαλίας και εκμάθησης μπορεί να γίνει με πιο εύκολο, αλλά και με πιο αποτελεσματικό τρόπο. Οι δάσκαλοι μπορούν να εξηγούν τα μαθήματα με μεγαλύτερη ευκολία, με αποτέλεσμα οι μαθητές να μπορούν να συμμετέχουν πιο ενεργά και να κατανοούν καλύτερα τα μαθήματα.
- Η ώρα του μαθήματος γίνεται πιο διασκεδαστική, αφού υπάρχει η διαδραστικότητα μεταξύ των μαθητών και των δασκάλων. Τα παραδείγματα και τα πειράματα γίνονται με έναν αρκετά πιο απλό και βατό τρόπο.
- Οι μαθητές είναι πιο ευδιάθετοι και πρόθυμοι να παρακολουθήσουν το μάθημα, διότι στην ουσία μέσω της AR/VR τεχνολογίας μπορούν να διασκεδάζουν και να μαθαίνουν ταυτόχρονα.
- Για την χρήση αυτών των τεχνολογιών απαιτείται κάποιος απαραίτητος εξοπλισμός πέραν της εφαρμογής. Οι τάξεις θα πρέπει να περιλαμβάνουν κάποιες συσκευές κινητών ή tablets, και φυσικά ένα monitor για να μεταδίδεται η εφαρμογή, ώστε να είναι ορατή σε όλους τους μαθητές.
- Η unity είναι ένα πολύ δυνατό game engine που προσφέρει πλούσια εργαλεία και πόρους για τη δημιουργία παιχνιδιών και εφαρμογών AR και όχι μόνο. Σε συνδυασμό με το vuforia engine γίνεται εύκολη η ενσωμάτωση αυτών των τεχνολογιών.

Αυτήν την στιγμή η εφαρμογή βρίσκεται σε ένα σημείο, στο οποίο είναι λειτουργική και μπορεί να χρησιμοποιηθεί από τους μαθητές και τους δάσκαλους. Ωστόσο, υπάρχουν βελτιώσεις που θα μπορούσαν να γίνουν σε μελλοντικά σχέδια. Παρακάτω αναφέρονται ορισμένες από αυτές:

- Εισαγωγή της εφαρμογής στο playstore.
- Δημιουργία της εφαρμογής για IOS συσκευές.
- Εισαγωγή ήχων και animation σε όλη την εφαρμογή.
- Δημιουργία της εφαρμογής για VR(εικονική πραγματικότητα).
- Βελτίωση του κώδικα.
- Δημιουργία περισσότερων παιχνιδιών, ώστε να καλύπτονται περισσότερα κεφάλαια από τα σχολικά βιβλία.
- Βελτίωση των 2D / 3D γραφικών και εφέ.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] A. Nikolaidis, "What Is Significant in Modern Augmented Reality: A Systematic Analysis of Existing Reviews" vol. 8, Issue 145, 21 May 2022.
- [2] Goldman Sachs Global Investment Research, Profiles in Innovation: Virtual & Augmented Reality, New York: The Goldman Sachs Group, Inc, 2016.
- [3] K. Alha, D. Leorke, E. Koskinen, J. Paavilainen, "Augmented play: An analysis of augmented reality features in location-based games" vol. 29, Issue 22, 13 Feb 2022.
- [4] T. Krodel, V. Schott, and J. Ovtcharova, "XR Technology Deployment in Value Creation" vol. 13, no. 8, 2023.
- [5] R. Wanger, D. Cozmiucv, "Extended Reality in Marketing—A Multiple Case Study on Internet of Things Platforms" vol. 13, no. 6, 2022.
- [6] A. Logeswaran, C. Munsch, J. Chong and N. Ralph "The role of extended reality technology in healthcare education: Towards a learner-centred approach" vol. 8, no. 1, 2021.
- [7] K. Takrouri, E. Causton, B. Simpson "AR Technologies in Engineering Education: Applications, Potential, and Limitations" vol. 2, 2022.
- [8] M. Farshid, J. Paschen, T. Eriksson and J. Kietzmann, "Go boldly!: Explore augmented reality (AR), virtual reality (VR), and mixed reality (MR) for business.,"vol. 61, no.5, 2018.
- [9] Alecu, F. (2010). Blender institute—the institute for open 3D projects. Open Source Science Journal, 36-45.
- [10] "Unity," : <https://unity.com>
- [11] "Vuforia Engine," : <https://developer.vuforia.com/library/getting-started/vuforia-features>
- [12] "Vuforia Engine," : <https://developer.vuforia.com/library/objects/image-targets>
- [13] "Photoshop," :<https://creativecloud.adobe.com/en-CA/learn/photoshop/web/ps-basics-fundamentals>.
- [14] "Blender," :[https://docs.blender.org/manual/en/latest/getting\\_started/about/introduction.html](https://docs.blender.org/manual/en/latest/getting_started/about/introduction.html).
- [15] "Unity," : <https://docs.unity3d.com/560/Documentation/Manual/CreatingScenes.html>.

## Παράρτημα Α: Αρχείο GameManager.cs αρχικής οθόνης

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

public class GameManager : MonoBehaviour
{
    public static GameManager Instance;
    public GameObject starBtn;
    public GameObject mainMenuUi;
    public GameObject ChooseGameUi;
    public GameObject goBackBtn;
    public GameObject quitBtn;

    private void Awake()
    {
        if(Instance == null)
        {
            Instance = this;
        }
        else
        {
            Destroy(gameObject);
        }
    }
    public void GoToChooseGame()
    {
        mainMenuUi.SetActive(false);
        ChooseGameUi.SetActive(true);
    }
    public void GoToMainMenu()
    {
        mainMenuUi.SetActive(true);
        ChooseGameUi.SetActive(false);
    }
    public void ExitGame()
    {
        Application.Quit();
    }
}
```

## Παράρτημα Β: Αρχείο CustomEditor.cs έβδομου παιχνιδιού

```
private void ConverToUpperButton()
{
    if(GUILayout.Button("To Upper"))
    {
        for(int i = 0; i < GameDataInstance.Columns; i++)
        {
            for(int j = 0; j < GameDataInstance.Rows; j++)
            {
                var errorCounter = Regex.Matches(GameDataInstance.Board[i].Row[j],
@"[a-w]").Count;

                if (errorCounter > 0)
                {
                    GameDataInstance.Board[i].Row[j] =
GameDataInstance.Board[i].Row[j].ToUpper();
                }
            }
        }

        foreach (var searchWord in GameDataInstance.SearchWords)
        {
            var errorCounter = Regex.Matches(searchWord.Word, @"[a-w]").Count;

            if (errorCounter > 0)
            {
                searchWord.Word = searchWord.Word.ToUpper();
            }
        }
    }
}

private void ClearBoardButton()
{
    if(GUILayout.Button("Clear Board"))
    {
        for (int i = 0; i < GameDataInstance.Columns; i++)
        {
            for (int j = 0; j < GameDataInstance.Rows; j++)
            {
                GameDataInstance.Board[i].Row[j] = "";
            }
        }
    }
}

private void FillUpWithRandomLettersButton()
{
    if (GUILayout.Button("Fill Up With Random"))
    {
        for (int i = 0; i < GameDataInstance.Columns; i++)
        {
            for (int j = 0; j < GameDataInstance.Rows; j++)
            {
                int errorCounter = Regex.Matches(GameDataInstance.Board[i].Row[j],
```

