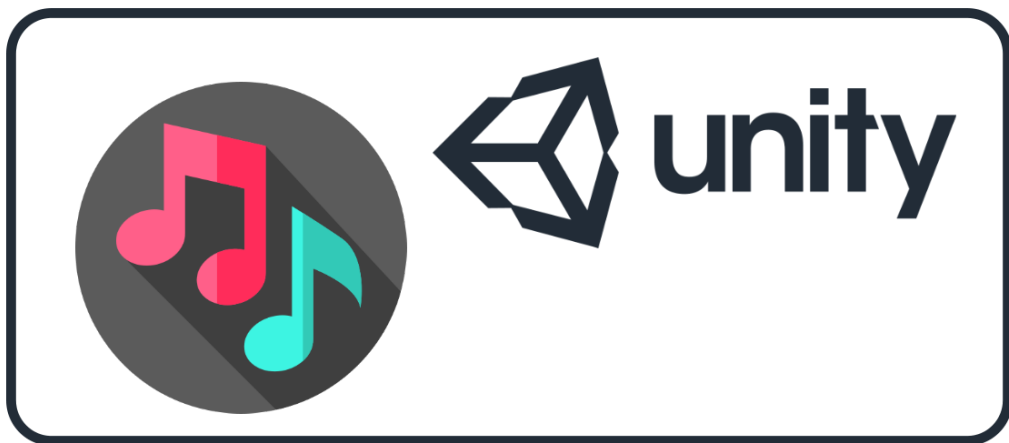


ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΠΑΙΧΝΙΔΙΟΥ
ΕΠΑΥΞΗΜΕΝΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΜΕ ΣΤΟΧΟ
ΤΗΝ ΔΙΑΔΡΑΣΤΙΚΗ ΔΙΔΑΣΚΑΛΙΑ ΤΟΥ
ΜΑΘΗΜΑΤΟΣ ΤΗΣ ΜΟΥΣΙΚΗΣ ΤΗΣ Δ'
ΔΗΜΟΤΙΚΟΥ



Της φοιτήτριας
Ζωής Δερμεντζόγλου
Αρ. Μητρώου: 144403

Επιβλέπων
Ευκλείδης Κεραμόπουλος
Καθηγητής

Τίτλος Π.Ε. Ανάπτυξη εκπαιδευτικού παιχνιδιού επαυξημένης πραγματικότητας με στόχο την
διαδραστική διδασκαλία του μαθήματος της Μουσικής της Δ' Δημοτικού

Κωδικός Π.Ε. 21350

Όνοματεπώνυμο φοιτήτριας Ζωή Δερμεντζόγλου
Όνοματεπώνυμο εισηγητή Ευκλείδης Κεραμόπουλος

Ημερομηνία ανάληψης Π.Ε. 14-10-2021

Ημερομηνία περάτωσης Π.Ε. 10-09-2023

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

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

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

«Στους φίλους μου που είναι η οικογένεια που διάλεξα»

Πρόλογος

Με τα χρόνια να περνούν και την τεχνολογία να αναπτύσσεται βρίσκουμε νέους τρόπους μεταλαμπάδευσης της γνώσης, τόσο στους μικρότερους αλλά όσο και στους μεγαλύτερους σε ηλικία ανθρώπους. Η επαυξημένη πραγματικότητα είναι μια τεχνολογία που μπορεί να προσφέρει πολλές δυνατότητες στην εκπαίδευση, με την χρήση επαυξημένων δραστηριοτήτων η εκμάθηση μαθημάτων τόσο θεωρητικών αλλά και πρακτικών γίνεται ευκολότερη και διασκεδαστική για όλους.

Περίληψη

Στόχος αυτής της πτυχιακής εργασίας είναι ο βελτιωμένος τρόπος εκμάθησης της μουσικής σε παιδιά Δ' δημοτικού με έναν ενδιαφέρον και ευχάριστο τρόπο. Το παιχνίδι χρησιμοποιεί την τεχνολογία της επαυξημένης πραγματικότητας, συνδυάζοντας quiz, μουσικά όργανα, ήχους, μελωδίες και άλλες δραστηριότητες. Η ανάπτυξη του παιχνιδιού πραγματοποιήθηκε μέσω της πλατφόρμας Unity, χρησιμοποιώντας το Vuforia Engine.

«Augmented reality educational game in the school subject of Music for the 4th grade of elementary school»

«Zoe Dermentzoglou»

Abstract

The purpose of this thesis is the improved way of learning music in 4th grade elementary school in an interesting and enjoyable way. The game uses augmented reality technology, combining quizzes, musical instruments, sounds, melodies and other activities. The development of the game takes place through the Unity platform using the Vuforia Engine.

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή κ.Κεραμόπουλο για την συνεργασία και για τα εφόδια που μου έδωσε στις σπουδές μου.

Κυρίως θα ήθελα να ευχαριστήσω τον αδερφό μου από άλλη μάνα, Χρυσόστομο Σέργιο Τασιό για όλα τα μερόνυχτα που προσπαθούσε να με κάνει να μην τα παρατήσω, με τον δικό του τρόπο. Τον αγαπημένο μου Χρήστο Μπαλαμπανίδη που χωρίς αυτόν και την τεράστια βοήθεια του ίσως να μην τελείωνα με την υλοποίηση του κώδικα καθώς και τον Νικόλα Πρόγιο για όλες τις φορές που ήταν εκεί να μου πει την γνώμη του και να μου δώσει την βοήθεια του. Έπειτα να ευχαριστήσω όλους τους φίλους μου, τα ΚΖ της καρδιάς μου, που ήταν εκεί να με παροτρύνουν και να με βοηθήσουν ψυχολογικά σε όλη την πορεία της δημιουργίας της πτυχιακής εργασίας.

Περιεχόμενα

Πρόλογος	v
Περίληψη	vi
Abstract	vii
Ευχαριστίες	viii
Περιεχόμενα	ix
Κατάλογος Εικόνων	xi
Συντομογραφίες	xvi
Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Δομή Εργασίας	1
1.2 Επίλογος	1
Κεφάλαιο 2ο: Επαυξημένη Πραγματικότητα	3
2.1 Εισαγωγή	3
2.2 Augmented Reality (AR)	3
2.3 Virtual Reality (VR)	3
2.4 Διαφορά AR από VR	4
2.5 Μικτή Πραγματικότητα (Mixed Reality - MR)	5
2.6 Διαφορά AR από MR	5
2.7 AR στον χώρο της εκπαίδευσης	6
2.8 Επίλογος	6
Κεφάλαιο 3ο: Εργαλεία Ανάπτυξης	7
3.1 Εισαγωγή	7
3.2 Unity	7
3.3 Unity Hub	7
3.4 Vuforia	8
3.4.1 Vuforia Engine	8
3.4.2 Visual Studio Code Editor	8
3.4.3 Δημιουργία Vuforia license key	9
3.4.4 Σύνδεση του Vuforia με το Unity	10
3.4.5 Δημιουργία QR codes	10
3.4.6 Δημιουργία Image Target στο Vuforia	11
3.4.7 Προσθήκη βάσης στο Unity	13
3.5 Blender	13
3.6 Adobe Photoshop	14
3.7 Audacity	14
3.8 Επίλογος	15
Κεφάλαιο 4ο: Ανάπτυξη 3D αντικειμένων	17
4.1 Εισαγωγή	17
4.2 Περιβάλλον Blender	17
4.3 Δημιουργία Αντικειμένων	17
4.4 Λίστα Αντικειμένων	22

4.5 Επίλογος	24
Κεφάλαιο 5ο: Παρουσίαση εφαρμογής	25
5.1 Εισαγωγή	25
5.2 Αρχικό Μενού	25
5.3 Quiz	30
5.3.1 Κεφάλαιο 11	33
5.3.2 Κεφάλαιο 13	45
5.3.3 Κεφάλαιο 18	55
5.3.4 Κεφάλαιο 21	56
5.3.5 Κεφάλαιο 22	58
5.4 Μεταλλόφωνο	59
5.5 Μουσείο Μουσικής	63
5.6 Άκου - Βρες	70
5.7 Επίλογος	75
Κεφάλαιο 6ο: Συμπεράσματα και προτάσεις για βελτίωση	77
ΒΙΒΛΙΟΓΡΑΦΙΑ	78

Κατάλογος Εικόνων

Εικόνα 2.1: AR vs VR	4
Εικόνα 2.2: Mixed reality	5
Εικόνα 2.3: Pokemon Go	6
Εικόνα 3.1: Unity Hub, Main	8
Εικόνα 3.2: Vuforia Engine AR	9
Εικόνα 3.3: Visual Studio Code Editor	9
Εικόνα 3.4: Vuforia License Manager, δημιουργία άδειας χρήσης	10
Εικόνα 3.5: Vuforia License Manager, δημιουργία κλειδιού άδειας	10
Εικόνα 3.6: Vuforia License Manager, παράδειγμα κλειδιού	11
Εικόνα 3.7: Vuforia Configuration	11
Εικόνα 3.8: QR editor	12
Εικόνα 3.9: Vuforia, δημιουργία βάσης	12
Εικόνα 3.10: Vuforia, προσθήκη Image Target	13
Εικόνα 3.11: Vuforia, λήψη βάσης	13
Εικόνα 3.12: Vuforia, λήψη βάσης για την πλατφόρμα του Unity	14
Εικόνα 3.13: Unity, εισαγωγή στόχου QR code.....	14
Εικόνα 4.1: Περιβάλλον Blender	17
Εικόνα 4.2: Edit Mode	18
Εικόνα 4.3: Viewport Shading, Wireframe	18
Εικόνα 4.4: Προβολή αντικειμένου σε Wireframe	18
Εικόνα 4.5: Πρότυπο Κιθάρας	19
Εικόνα 4.6: Εργαλείο Loop Cut	19
Εικόνα 4.7: Εργαλείο Scale	19
Εικόνα 4.8: Χρήση “Loop Cut” και “Scale”	19
Εικόνα 4.9: Εργαλείο “Extrude Region”	20
Εικόνα 4.10: Κορμός Κιθάρας	20
Εικόνα 4.11: Μορφή Κιθάρας	20
Εικόνα 4.12: Μέρη κιθάρας	21
Εικόνα 4.13: Material Properties	21
Εικόνα 4.14: Τελικό αποτέλεσμα	22
Εικόνα 4.15 : Συνθέτες	22
Εικόνα 4.16 : Μουσικά όργανα	23
Εικόνα 4.17 : Περιφερειακά αντικείμενα	23

Εικόνα 5.1: Main menu	25
Εικόνα 5.2: Έναρξη εφαρμογής	26
Εικόνα 5.3: Logo Container	26
Εικόνα 5.4: Button Component.....	27
Εικόνα 5.5: Main Container	27
Εικόνα 5.6: Κύριο Μενού	28
Εικόνα 5.7: Prompt Canvas	28
Εικόνα 5.8: Yes button component	28
Εικόνα 5.9: Yes button component	29
Εικόνα 5.10: OnClose μέθοδος	29
Εικόνα 5.11: Close button component	29
Εικόνα 5.12: Επίπεδο UI	29
Εικόνα 5.13: Quiz Button, Button component	30
Εικόνα 5.14: ContainerAnimatin class	30
Εικόνα 5.15: LeanTween package	31
Εικόνα 5.16: Button Component, onClick μέθοδος	31
Εικόνα 5.17: SetSceneName μέθοδος	31
Εικόνα 5.18: Back Button, Button component	32
Εικόνα 5.19: Session Container	32
Εικόνα 5.20: Κεφάλαια	32
Εικόνα 5.21: Κεφάλαια Buttons, Button component	33
Εικόνα 5.22: Session 11	33
Εικόνα 5.23: AR Camera, Vuforia Behaviour	33
Εικόνα 5.24: Stage Guide Dialogue script setup	34
Εικόνα 5.25: OnEnable script	34
Εικόνα 5.26: Unity Editor	34
Εικόνα 5.27: Stage Guide Popur	35
Εικόνα 5.28: OK button component	35
Εικόνα 5.29: CloseDialog μέθοδος	36
Εικόνα 5.30: Image Target component	36
Εικόνα 5.31: Image Target του Κεφαλαίου 11	36
Εικόνα 5.32: Προβολή σκηνης κεφαλαίου 11	37
Εικόνα 5.33: Canvas Button	37
Εικόνα 5.34: button component	37
Εικόνα 5.35: Button Manager	38

Εικόνα 5.36: Image component	38
Εικόνα 5.37: Answer μέθοδος	38
Εικόνα 5.38: Sound Manager Unity Editor	39
Εικόνα 5.39 Update Unity μέθοδος	39
Εικόνα 5.40: PlayMusic μέθοδος	40
Εικόνα 5.41: ενεργά Particles	40
Εικόνα 5.42: BackButton	40
Εικόνα 5.43: Update Unity μέθοδος	41
Εικόνα 5.44: Quiz Manager	41
Εικόνα 5.45: ShowQuiz μέθοδος	41
Εικόνα 5.46: GenerateQuestion μέθοδος	42
Εικόνα 5.47: setAnswer μέθοδος	42
Εικόνα 5.48: Question Format	42
Εικόνα 5.49: Question List Unity Editor	43
Εικόνα 5.50: onClick Answer μέθοδος	43
Εικόνα 5.51: μεταβλητή _isCorrect	43
Εικόνα 5.52: Answer μέθοδος	43
Εικόνα 5.53: GeneralQuestion μέθοδος	44
Εικόνα 5.54: TerminalQuiz και InitiateDialogCanvas μέθοδος	44
Εικόνα 5.55: Ενημέρωση του χρήστη για τις σωστές απαντήσεις	45
Εικόνα 5.56: RetryButton Button component	45
Εικόνα 5.57: LoadNextScene μέθοδος	45
Εικόνα 5.58: Session 13	45
Εικόνα 5.59: AR Camera, Vuforia Behaviour	46
Εικόνα 5.60: Stage Guide Dialogue script setup	46
Εικόνα 5.61: OnEnable μέθοδος	46
Εικόνα 5.62: Unity Editor	47
Εικόνα 5.63: Stage Guide Popur	47
Εικόνα 5.64: OK button component	47
Εικόνα 5.65: CloseDialog μέθοδος	48
Εικόνα 5.66: Image Target	48
Εικόνα 5.67: Image Target του Κεφαλαίου 13	48
Εικόνα 5.68: Προβολή σκηνής κεφαλαίου 13	49
Εικόνα 5.69: Canvas Button	49
Εικόνα 5.70: ButtonManagerModels και button component	50

Εικόνα 5.71: Image component	50
Εικόνα 5.72: Answer μέθοδος	50
Εικόνα 5.73: Quiz Manager Models	51
Εικόνα 5.74: ShowQuiz μέθοδος	51
Εικόνα 5.75: GenerateQuestion μέθοδος	52
Εικόνα 5.76: SetAnswer μέθοδος	52
Εικόνα 5.77: Question Format	52
Εικόνα 5.78: Question List Unity Editor	53
Εικόνα 5.79: Question List Unity Editor	53
Εικόνα 5.80: μεταβλητή _isCorrect	53
Εικόνα 5.81: Answer μέθοδος	53
Εικόνα 5.82: Button Left	54
Εικόνα 5.83: ActiveObject μέθοδος	54
Εικόνα 5.84: Answer μέθοδος	54
Εικόνα 5.85: WaitForNext μέθοδος	55
Εικόνα 5.86: GeneralQuestion μέθοδος	55
Εικόνα 5.87: Image Target του Κεφαλαίου 18	55
Εικόνα 5.88: Προβολή σκηνής κεφαλαίου 18	56
Εικόνα 5.89: Image Target του Κεφαλαίου 21.....	57
Εικόνα 5.90: Προβολή σκηνής κεφαλαίου 21	57
Εικόνα 5.91: Image Target του Κεφαλαίου 22	58
Εικόνα 5.92: Προβολή σκηνής κεφαλαίου 21	59
Εικόνα 5.93: Scene Metallofono	59
Εικόνα 5.94: Stage Guide Popur	60
Εικόνα 5.95: Image Target του Μεταλλόφωνο	60
Εικόνα 5.96: 3D Μεταλλόφωνο	61
Εικόνα 5.97: Mesh Collider	61
Εικόνα 5.98: XylophoneManager	61
Εικόνα 5.99: Unity μέθοδος Start	62
Εικόνα 5.100: Audio Clip list	62
Εικόνα 5.101: Update του Unity	62
Εικόνα 5.102: PlayMusic μέθοδος	63
Εικόνα 5.103: Μέγεθος του πίνακα.....	63
Εικόνα 5.104: Scene Μουσείο Μουσικής	63
Εικόνα 5.105: Stage Guide Popur	64

Εικόνα 5.106: Image Target του Μουσείο Μουσικής	64
Εικόνα 5.107: Canvas Buttons	64
Εικόνα 5.108: Unity Event OnClick με τη μέθοδος ChangeItem	65
Εικόνα 5.109: gameObject InGameItemHolder	65
Εικόνα 5.110: ItemSelection	65
Εικόνα 5.111: ChangeItem	66
Εικόνα 5.112: SelectItem	66
Εικόνα 5.113: objectNamesList	66
Εικόνα 5.114: SelectItem	67
Εικόνα 5.115: Box Collider	67
Εικόνα 5.116: ActivateText μέθοδος	67
Εικόνα 5.117: Melody button	68
Εικόνα 5.118: OnClick μέθοδος	68
Εικόνα 5.119: PlayMusic μέθοδος	68
Εικόνα 5.120: Πληροφορίες για τα αντικείμενα	69
Εικόνα 5.121: Αρχικοποίηση πίνακα _instrumentInfo	69
Εικόνα 5.122: Unity Editor	69
Εικόνα 5.123: Close Dialog	70
Εικόνα 5.124: Unity Editor	70
Εικόνα 5.125: Scene Άκου Βρες	71
Εικόνα 5.126: Stage Guide Popup	71
Εικόνα 5.127: Image Target του Μεταλλόφωνο	71
Εικόνα 5.128: Canvas Buttons	72
Εικόνα 5.129: OnClick Answer μέθοδος	72
Εικόνα 5.130: ButtonManager2 μέθοδος	72
Εικόνα 5.131: MusicalQuiz μέθοδος	72
Εικόνα 5.132: Update μέθοδο του Unity	73
Εικόνα 5.133: PlayMusic μέθοδος	73
Εικόνα 5.134: ShowQuiz μέθοδος	74
Εικόνα 5.135: _questionsList	74
Εικόνα 5.136: QuestionsList Unity Editor	74
Εικόνα 5.137: Σκηνή Άκου Βρες	75

Συντομογραφίες

ΔΠΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
VR	Virtual Reality
AR	Augmented Reality
VFX	Visual Effects
UI	User Interface
MR	Mixed Reality
3D	Three-dimensional

Κεφάλαιο 1ο: Εισαγωγή

Η παρούσα Π.Ε. ήταν μια προσπάθεια δημιουργίας διαφόρων δραστηριοτήτων/ασκήσεων με την χρήση τεχνολογιών επαυξημένης πραγματικότητας σε συσκευές Android. Οι μαθητές πρέπει να σκανάρουν τις εικόνες ή QR codes που θα τους δίνονται έτσι θα εμφανιστεί μια σκηνή με την άσκηση/δραστηριότητα και με αυτό τον τρόπο θα μπορούν να κάνουν τις ασκήσεις τους πιο ευχάριστα σαν να είναι ένα παιχνίδι.

Χρησιμοποιώντας την εφαρμογή, όπου θα επαυξάνει την εικόνα με εικονικά 3D γραφικά, οι μαθητές θα δέχονται ερωτήσεις από τα κεφάλαια του σχολικού βιβλίου, και θα έχουν να επιλέξουν σαν απάντηση 3D αντικείμενα που σχετίζονται με τον χώρο της μουσικής. Παράλληλα θα υπάρχουν και κάποιες έξτρα ενότητες όπου ο μαθητής θα μπορεί να εξασκηθεί ή να λάβει πληροφορίες μέσα από τα 3D αντικείμενα που βλέπει.

1.1 Δομή Εργασίας

Η Π.Ε. αποτελείται από έξι κεφάλαια. Στο 1ο κεφάλαιο γίνεται μια μικρή εισαγωγή για το σκοπό της Π.Ε. Στο 2ο κεφάλαιο υπάρχει αναδρομή στην AR. Στο 3ο κεφάλαιο γίνεται αναφορά στα εργαλεία και τις τεχνολογίες που χρησιμοποιήθηκαν για τη δημιουργία του project. Στο 4ο κεφάλαιο περιγράφεται ο τρόπος που δημιουργήθηκαν τα 3D αντικείμενα που χρησιμοποιήθηκαν στο project. Στο 5ο κεφάλαιο θα γίνει μια παρουσίαση της εφαρμογής, εικόνες και μερικά κομμάτια από την συγγραφή του κώδικα. Τέλος, στο 6ο κεφάλαιο αναφέρονται μερικά συμπεράσματα και προτάσεις για μελλοντικές βελτιώσεις.

1.2 Επίλογος

Στο πρώτο κεφάλαιο έγινε μια σύντομη αναφορά στην ιδέα της πτυχιακής εργασίας καθώς και στην δομή της.

Κεφάλαιο 2ο: Επαυξημένη Πραγματικότητα

2.1 Εισαγωγή

Η επαυξημένη πραγματικότητα (AR) είναι μια τεχνολογία που περιλαμβάνει την επικάλυψη γραφικών υπολογιστή στον πραγματικό κόσμο. Το AR βρίσκεται σε ένα γενικότερο πλαίσιο που ονομάζεται Μικτή Πραγματικότητα (MR), η οποία αναφέρεται σε ένα πολυαξονικό φάσμα περιοχών που καλύπτουν την Εικονική πραγματικότητα (VR), την AR και άλλες σχετικές τεχνολογίες.

Η εικονική πραγματικότητα είναι ένας όρος που χρησιμοποιείται για τρισδιάστατα περιβάλλοντα που δημιουργούνται από υπολογιστή και επιτρέπουν στον χρήστη να εισέλθει και να αλληλεπιδράσει με ψηφιακά περιβάλλοντα [7].

2.2 Augmented Reality (AR)

Η Επαυξημένη Πραγματικότητα (Augmented Reality - AR) αναφέρεται σε μια τεχνολογία που συνδυάζει τον φυσικό κόσμο με τον ψηφιακό κόσμο, παρέχοντας πληροφορίες που εμφανίζονται μπροστά στον χρήστη μέσω μιας συσκευής, όπως ένα smartphone, tablet, ή AR γυαλιά.

Η AR χρησιμοποιεί τον πραγματικό κόσμο ως βάση. Ο χρήστης βλέπει το περιβάλλον του και επιπλέον πληροφορίες ή γραφικά που προβάλλονται πάνω σε αυτό.

Η τεχνολογία AR ενσωματώνει τα ψηφιακά στοιχεία στην πραγματική σκηνή. Αυτά τα στοιχεία μπορεί να είναι εικόνες, βίντεο, 3D μοντέλα, πληροφορίες και άλλα. Έχει πολλές εφαρμογές, από παιχνίδια και ψυχαγωγία μέχρι εκπαιδευτικές, ιατρικές, βιομηχανικές και εμπορικές εφαρμογές. Παράδειγμα είναι η δυνατότητα να δει κανείς πληροφορίες για προϊόντα όταν κρατάει το smartphone πάνω από αυτά.

Μπορεί να χρησιμοποιηθεί για τη δημιουργία εκπαιδευτικών εμπειριών, όπως προβολή 3D μοντέλων ανατομίας ή ιστορικών σκηνών στον πραγματικό χώρο. Ανοίγει νέους τρόπους δημιουργίας περιεχομένου και διαδραστικών εμπειριών, δίνοντας τη δυνατότητα στους δημιουργούς να αλληλεπιδρούν με το περιβάλλον των χρηστών.

Η Επαυξημένη Πραγματικότητα έχει ευρεία εφαρμογή σε διάφορους τομείς και συνεχώς εξελίσσεται με νέες τεχνολογίες και εφαρμογές που προσφέρουν εντυπωσιακές και ενισχυμένες εμπειρίες στους χρήστες [8].

2.3 Virtual Reality (VR)

Η Εικονική Πραγματικότητα είναι μια τεχνολογία που δημιουργεί εικονικά περιβάλλοντα και εμπειρίες, τα οποία μοιάζουν με τον πραγματικό κόσμο. Οι χρήστες της βιώνουν τα εικονικά περιβάλλοντα με τη βοήθεια ειδικών συσκευών, όπως γυαλιά VR, και μπορούν να αλληλεπιδρούν με το περιβάλλον και τα αντικείμενα μέσω κινήσεων, φωνητικών εντολών και άλλων μεθόδων.

Είναι δημοφιλής τομέας της ψυχαγωγίας. Χρησιμοποιείται για τη δημιουργία εικονικών κόσμων και παιχνιδιών, παρέχοντας στους χρήστες μια πιο εμπλουτισμένη και συναρπαστική εμπειρία.

Στον τομέα της εκπαίδευσης μπορεί να παρέχει εξαιρετικά αληθοφανή εικονικά περιβάλλοντα, όπως ιστορικά γεγονότα, φυσικά φαινόμενα και επιστημονικές έννοιες, ώστε να παρέχει στους μαθητές μια πιο ολοκληρωμένη οπτικοακουστική εμπειρία για να κατανοήσουν και να ενσωματώσουν τις γνώσεις που λαμβάνουν.

Στον τομέα του τουρισμού, μπορεί να προσφέρει μοναδικές και συναρπαστικές εμπειρίες ταξιδιού για τους ταξιδιώτες. Αυτό μπορεί να βελτιώσει την προώθηση του τουρισμού, να διευρύνει τις γνώσεις των ταξιδιωτών για διάφορους προορισμούς και να παρέχει εντυπωσιακές εμπειρίες ακόμα και από την άνεση του σπιτιού τους. Οι ταξιδιώτες μπορούν να εξερευνήσουν διάφορους προορισμούς μέσω εικονικών περιηγήσεων. Μπορούν να περπατήσουν σε εμβληματικά αξιοθέατα, να απολαύσουν την πανοραμική θέα από κορυφές βουνών και να ανακαλύψουν πολιτιστικά αξιοθέατα. Επίσκεψη σε εικονικά μουσεία, αρχαιολογικούς χώρους και άλλα αξιοθέατα, εξερευνώντας την τέχνη, την ιστορία και τον πολιτισμό.

Δίνει μεγάλο πλεονέκτημα στην εκπαίδευση ικανοτήτων και αποτελεί έναν καινοτόμο τρόπο για την απόκτηση και τη βελτίωση δεξιοτήτων σε διάφορους τομείς. Επιτρέπει τη δημιουργία ρεαλιστικών περιβαλλόντων προσομοίωσης. Αυτό μπορεί να βοηθήσει τους μαθητές να ασκηθούν και να βελτιώσουν δεξιότητες, όπως η πιλοτική, η οδήγηση αυτοκινήτου ή ακόμα και ιατρικές διαδικασίες. Οι μαθητές μπορούν να αναπτύξουν δημιουργικές δεξιότητες μέσω εικονικών εργαστηρίων σχεδίασης, γλυπτικής, ζωγραφικής και άλλων καλλιτεχνικών δραστηριοτήτων.

2.4 Διαφορά AR από VR

Η Επαυξημένη Πραγματικότητα (Augmented Reality - AR) και η Εικονική Πραγματικότητα (Virtual Reality - VR) είναι δύο διαφορετικές τεχνολογίες που παρουσιάζουν τον ψηφιακό κόσμο με διαφορετικούς τρόπους. Αν και μοιάζουν, υπάρχουν σημαντικές διαφορές μεταξύ τους, όπως φαίνεται στην Εικόνα 2.1.

Στην AR, ο πραγματικός κόσμος εμφανίζεται μπροστά στον χρήστη, ενώ επιπλέον ψηφιακά στοιχεία προβάλλονται πάνω σε αυτόν. Ο χρήστης βλέπει το περιβάλλον του και τα επιπλέον ψηφιακά στοιχεία ταυτόχρονα. Στην VR, ο χρήστης βυθίζεται πλήρως σε ένα εικονικό περιβάλλον και διαχωρίζεται από τον πραγματικό κόσμο. Δεν βλέπει το πραγματικό περιβάλλον γύρω του.

Οι συσκευές AR, όπως τα smartphones, tablets και AR γυαλιά, χρησιμοποιούν κάμερες και αισθητήρες για να ανιχνεύουν το περιβάλλον και να προβάλλουν τα ψηφιακά στοιχεία πάνω σε αυτό. Ενώ στην VR, όπως τα VR headsets, καλύπτουν το πεδίο όρασης του χρήστη με εικονικό περιβάλλον και χρησιμοποιούν αισθητήρες κίνησης για να ανιχνεύουν την κίνηση του χρήστη [9].



Εικόνα 2.1: AR vs VR

2.5 Μικτή Πραγματικότητα (Mixed Reality - MR)

Η Μικτή Πραγματικότητα (Mixed Reality - MR) αναφέρεται σε μια ευρύτερη κατηγορία τεχνολογιών που συνδυάζουν τον φυσικό και τον ψηφιακό κόσμο με τρόπο που επιτρέπει την αλληλεπίδραση του χρήστη με τα εικονικά αντικείμενα στον πραγματικό χώρο. Η MR ενοποιεί τις έννοιες της Επαυξημένης Πραγματικότητας (Augmented Reality - AR) και της Εικονικής Πραγματικότητας (Virtual Reality - VR) με τρόπο που επιτρέπει στα εικονικά και τα φυσικά στοιχεία να αλληλεπιδρούν σε πραγματικό χρόνο, (Εικόνα 2.2).

Στη Μικτή Πραγματικότητα, οι εικονικές προσθήκες μπορούν να αλληλεπιδρούν με το περιβάλλον του χρήστη, και οι χρήστες μπορούν να αλληλεπιδρούν τόσο με τα εικονικά όσο και με τα φυσικά αντικείμενα. Αυτή η τεχνολογία επιτρέπει εμβυθισμένες και συναρπαστικές εμπειρίες, όπως η δημιουργία εικονικών αντικειμένων που μπορούν να τοποθετηθούν και να αλληλεπιδράσουν με τον πραγματικό χώρο [10].



Εικόνα 2.2: Mixed reality

2.6 Διαφορά AR από MR

Οι δύο τεχνολογίες έχουν εφαρμογές σε διάφορους τομείς όπως η ψυχαγωγία, η εκπαίδευση, η υγεία, η βιομηχανία και πολλούς άλλους. Η AR αντικαθιστά ή προσθέτει εικονικά αντικείμενα στο πραγματικό περιβάλλον, ενώ η MR δημιουργεί ένα συνδυασμένο περιβάλλον όπου εικονικά και πραγματικά αντικείμενα αλληλεπιδρούν μεταξύ τους.

Η MR αναφέρεται στη συνδυασμένη εμπειρία της πραγματικής και εικονικής πραγματικότητας συνήθως χρησιμοποιούνται συσκευές όπως τα HoloLens της Microsoft ή τα Magic Leap. Επιτρέπει στους χρήστες να αλληλεπιδρούν με εικονικά αντικείμενα που ενσωματώνονται στο πραγματικό περιβάλλον τους. Υπάρχουν διάφορα επίπεδα MR, από την απλή προσθήκη εικονικών αντικειμένων μέχρι τη δημιουργία σύνθετων αλληλεπιδράσεων.

Από την άλλη η AR αναφέρεται στην προσθήκη εικονικών στοιχείων στο πραγματικό περιβάλλον, χωρίς να το αντικαθιστά. Χρησιμοποιείται συνήθως μέσω συσκευών όπως smartphones, tablets και γυαλιά AR. Επιτρέπει στους χρήστες να βλέπουν εικονικά αντικείμενα ενσωματωμένα στο πραγματικό περιβάλλον μέσα από την οθόνη της συσκευής τους. Ένα παράδειγμα είναι τα παιχνίδια όπως το Pokémon GO (Εικόνα 2.3), που τοποθετούν εικονικά πλάσματα στον πραγματικό κόσμο [10].



Εικόνα 2.3: Pokemon Go

2.7 AR στον χώρο της εκπαίδευσης

Η Επαυξημένη Πραγματικότητα προσφέρει πολλά θετικά στοιχεία στον τομέα της εκπαίδευσης, μπορεί να κάνει τη μάθηση πιο συναρπαστική και διαδραστική. Οι μαθητές μπορούν να αλληλεπιδρούν με τα μαθήματα μέσα από εικονικά αντικείμενα και περιβάλλοντα, βοηθώντας τους να κατανοήσουν τα μαθήματα πιο βαθιά. Παρέχει πρόσβαση σε εικονικές εμπειρίες που είναι δύσκολο ή ακατόρθωτο να πραγματοποιηθούν στον πραγματικό κόσμο. Για παράδειγμα, μπορεί να επιτρέπει στους μαθητές να εξερευνήσουν τον ανθρώπινο σώμα σε 3D ή να ταξιδέψουν σε ιστορικά γεγονότα.

Μέσω του AR δίνεται η δυνατότητα δημιουργίας νέων εκπαιδευτικών περιεχομένων και εργαλείων. Οι εκπαιδευτικοί μπορούν να δημιουργήσουν εντυπωσιακά περιβάλλοντα μάθησης που προσαρμόζονται στις ανάγκες των μαθητών. Έτσι μπορούν να δημιουργηθούν εξατομικευμένες εκπαιδευτικές εμπειρίες. Οι μαθητές μπορούν να προχωρούν στον ρυθμό που τους ταιριάζει και να επικεντρώνονται σε αυτά που χρειάζονται περισσότερο. Επιπλέον, δίνει τη δυνατότητα στους μαθητές να αναλάβουν ενεργό ρόλο στη διαδικασία μάθησης. Αυτό μπορεί να τους εμπνεύσει και να τους βοηθήσει να αναπτύξουν αυτοδιαχειριστικές δεξιότητες.

Με την χρήση του AR παρέχονται εξατομικευμένες λύσεις για ανθρώπους με ειδικές ανάγκες, επιτρέποντάς τους να συμμετέχουν πλήρως στην εκπαιδευτική διαδικασία. Μπορούν να δημιουργηθούν εικονικές εμπειρίες που βοηθούν στην ανάπτυξη δεξιοτήτων και την κατανόηση. Καθώς επίσης βελτιώνεται η πρόσβαση στην εκπαίδευση για άτομα με ειδικές ανάγκες. AR εφαρμογές μπορούν να παρέχουν εναλλακτικούς τρόπους πρόσβασης, όπως φωνητικές εντολές ή κίνηση, για τα άτομα που δυσκολεύονται να χρησιμοποιήσουν συμβατικά μέσα. Για παράδειγμα, μπορούν να προσφέρουν εκπαιδευτικά παιχνίδια που βοηθούν τα παιδιά με αυτισμό να βελτιώσουν τις κοινωνικές τους δεξιότητες.

Εν κατακλείδι, η Επαυξημένη Πραγματικότητα μπορεί να μεταμορφώσει τον τρόπο που προσεγγίζουμε την εκπαίδευση, κάνοντάς την πιο συναρπαστική, αποτελεσματική και προσαρμοσμένη στις ανάγκες και τις δυνατότητες κάθε μαθητή [10].

2.8 Επίλογος

Στο δεύτερο κεφάλαιο έγινε αναφορά στην επαυξημένη πραγματικότητα, στη μικτή πραγματικότητα καθώς και στις διαφορές της επαυξημένη πραγματικότητας από την εικονική πραγματικότητα.

Κεφάλαιο 3ο: Εργαλεία Ανάπτυξης

3.1 Εισαγωγή

Στο παρόν κεφάλαιο θα παρουσιαστούν τα εργαλεία ανάπτυξης Unity, Vuforia, Blender και Audacity που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής καθώς και τα βήματα που πραγματοποιήθηκαν για την εγκατάσταση και σύνδεση των εργαλείων.

3.2 Unity

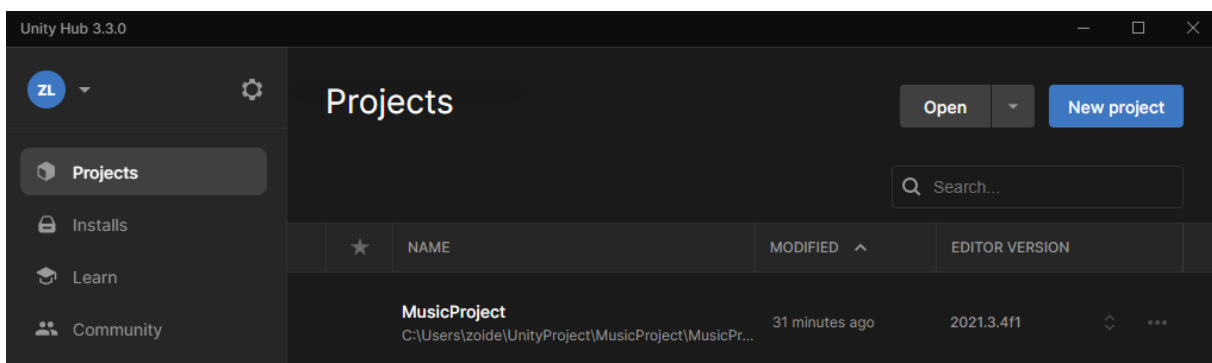
Το Unity αποτέλεσε την βασική πλατφόρμα υλοποίησης της εφαρμογής. Ωστόσο, δεν είναι η μόνη πλατφόρμα ανάπτυξης παιχνιδιών, παρόλα αυτά είναι ιδανική για συγκεκριμένα είδη όπως το AR και για τον λόγο αυτό κι επιλέχθηκε για την παρούσα Π.Ε. Αποτελεί μια από τις μεγαλύτερες πλατφόρμες στη δημιουργία παιχνιδιών καθώς το περιεχόμενο της δίνεται δωρεάν στους χρήστες της μαζί με συνεχείς αναβαθμίσεις.

Το Unity παρέχει στους χρήστες του το Asset Store μια βιβλιοθήκη όπου περιέχει δωρεάν assets (αντικείμενα) φτιαγμένα τόσο από χρήστες του Unity αλλά όσο και από την Unity Technologies. Τα assets μπορούν να αντικατοπτρίσουν αντικείμενα τόσο δισδιάστατα όσο και τρισδιάστατα [1][2].

Το Unity χρησιμοποιεί ένα πακέτο ονόματι AR Foundation το οποίο επιτρέπει την αλληλεπίδραση με πλατφόρμες επαυξημένης πραγματικότητας. Αυτό το πακέτο παρουσιάζει μια διεπαφή για χρήση από τους προγραμματιστές Unity όμως δεν εφαρμόζει από μόνη της καμία δυνατότητα AR. Για να χρησιμοποιηθεί σωστά το AR Foundation σε μια συσκευή, χρειάζεται επίσης ξεχωριστά πακέτα για τις πλατφόρμες που υποστηρίζονται επίσημα από το Unity. Στη παρούσα Π.Ε. χρησιμοποιήθηκε το ARCore XR Plugin μιας και η εφαρμογή φτιάχτηκε για Android συσκευές [3].

3.3 Unity Hub

Το Unity Hub είναι μια εφαρμογή μέσω της οποίας μπορεί να γίνει η διαχείριση, τόσο αυτόματα όσο και μη, των εγκαταστάσεων των εκδόσεων του Unity. Επιπλέον υπάρχει η δυνατότητα διαχείρισης του λογαριασμού του Unity και των αδειών που χρησιμοποιεί ο Editor. Αποτελεί επίσης μέσο διαχείρισης όλων των project που έχουν δημιουργηθεί με τον εκάστοτε Unity λογαριασμό καθώς και για τη δημιουργία νέων projects, [4], (Εικόνα 3.1).



Εικόνα 3.1: Unity Hub, Main

3.4 Vuforia

Το Vuforia είναι μια πλατφόρμα ανάπτυξης εφαρμογών Επαυξημένης Πραγματικότητας (AR) και Μικτής Πραγματικότητας (MR). Η σύνδεση του Vuforia με το Unity επιτρέπει τη δημιουργία εφαρμογών και παιχνιδιών για Android και iOS χρησιμοποιώντας μια ροή εργασίας drag-and-drop.

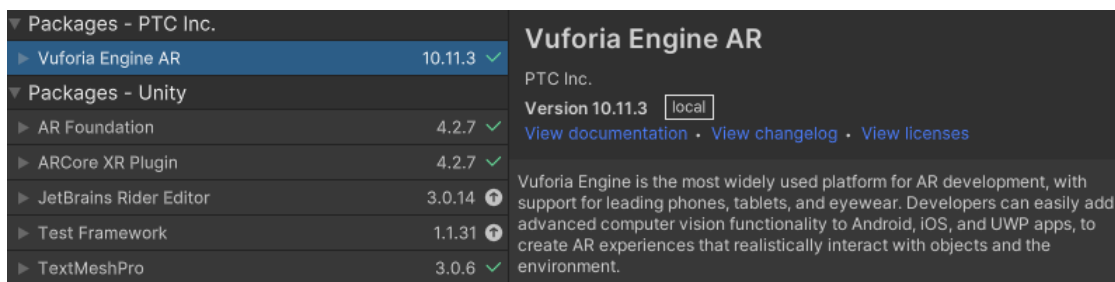
Το Vuforia υποστηρίζει πολλές συσκευές τρίτων κατασκευαστών (όπως γυαλιά AR/MR) και συσκευές VR με κάμερες με όψη στο πίσω μέρος (όπως το Gear VR).

3.4.1 Vuforia Engine

Το Vuforia Engine είναι η πιο ευρέως χρησιμοποιούμενη πλατφόρμα για ανάπτυξη AR, με υποστήριξη για τα περισσότερα τηλέφωνα, tablet και γυαλιά. Οι προγραμματιστές μπορούν εύκολα να προσθέσουν προηγμένες λειτουργίες όρασης υπολογιστή σε εφαρμογές Android, iOS και UWP, για να δημιουργήσουν εμπειρίες AR που αλληλεπιδρούν ρεαλιστικά με αντικείμενα και το περιβάλλον.

Η βιβλιοθήκη Vuforia Engine περιέχει την πιο πρόσφατη τεχνική τεκμηρίωση για τη δημιουργία εφαρμογών AR με το Vuforia Engine SDK. Ενημερώνεται συνεχώς με οδηγούς και πληροφορίες για τις πιο πρόσφατες δυνατότητες και υπηρεσίες που προσφέρει η Vuforia Engine. Επίσης, είναι οργανωμένη σύμφωνα με τις κύριες κατηγορίες χαρακτηριστικών της. Οι κύριες κατηγορίες χαρακτηριστικών είναι τα Images και Objects, τα Environments και η υποστήριξη πλατφόρμας.

Το Vuforia Engine υποστηρίζει την ανάπτυξη στο Unity Editor καθώς και στις τρεις κύριες εγγενείς πλατφόρμες: iOS, Android και Universal Windows Platform (UWP), [5], (Εικόνα 3.2).



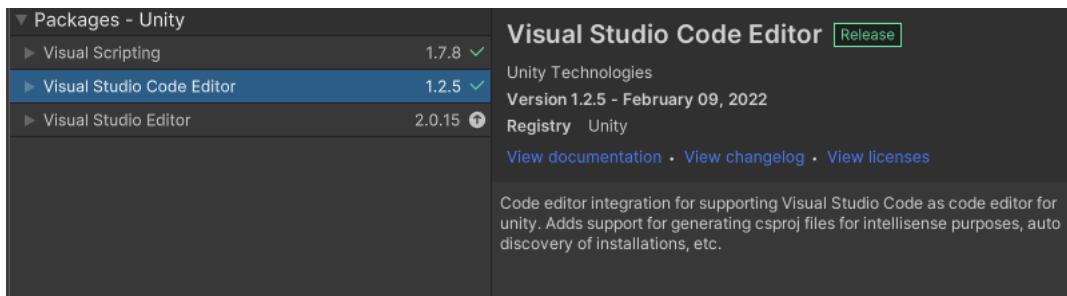
Εικόνα 3.2: Vuforia Engine AR

3.4.2 Visual Studio Code Editor

Το Visual Studio Code Editor, είναι πακέτο για την επεξεργασία κώδικα για το Unity, έχει αρκετά πλεονεκτήματα, που το καθιστούν μια δημοφιλή επιλογή μεταξύ των προγραμματιστών.

Ενώ το Unity χρησιμοποιεί κυρίως τη γλώσσα C# για τον προγραμματισμό, το VS Code υποστηρίζει μια ευρεία ποικιλία γλωσσών προγραμματισμού. Αυτό σημαίνει ότι μπορεί ο προγραμματιστής να εργαστεί σε έργα που περιλαμβάνουν πολλές γλώσσες ή τεχνολογίες.

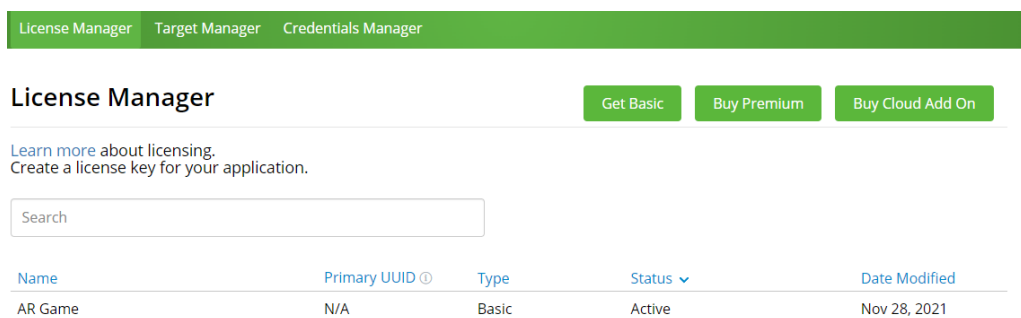
Είναι δωρεάν και Open Source. Το VS Code είναι διαθέσιμο σε πολλές πλατφόρμες, συμπεριλαμβανομένων των Windows, macOS και Linux. Αυτή η δυνατότητα διασυνδεσιμότητας επιτρέπει να εργάζεται ο κάθε προγραμματιστής ανεμπόδιστα σε διαφορετικά λειτουργικά συστήματα. Αυτό είναι ιδιαίτερα χρήσιμο εάν συνεργάζεται με άλλους προγραμματιστές που χρησιμοποιούν διαφορετικές πλατφόρμες, (Εικόνα 3.3).



Εικόνα 3.3: Visual Studio Code Editor

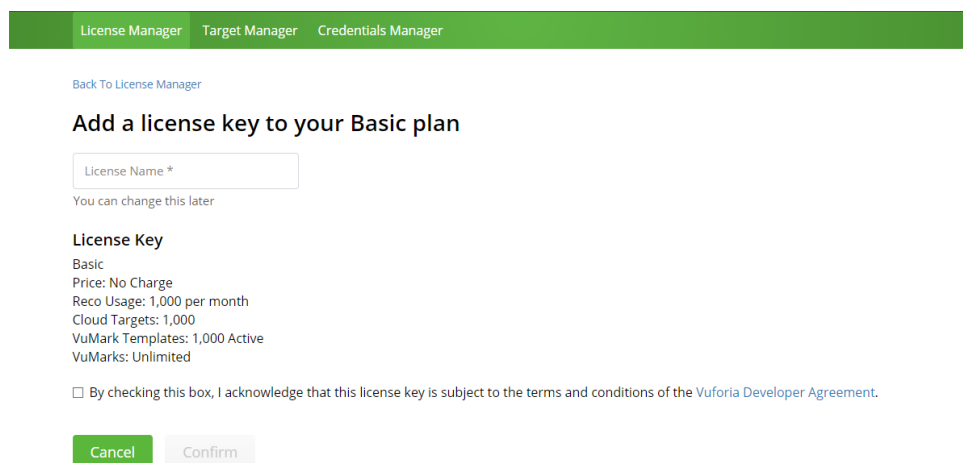
3.4.3 Δημιουργία Vuforia license key

Στο κεντρικό site του Vuforia υπάρχει η δυνατότητα δημιουργίας προφίλ. Έπειτα μεταβαίνοντας στη Διαχείριση άδειας χρήσης (License Manager), (Εικόνα 3.4)



Εικόνα 3.4: Vuforia License Manager, δημιουργία άδειας χρήσης

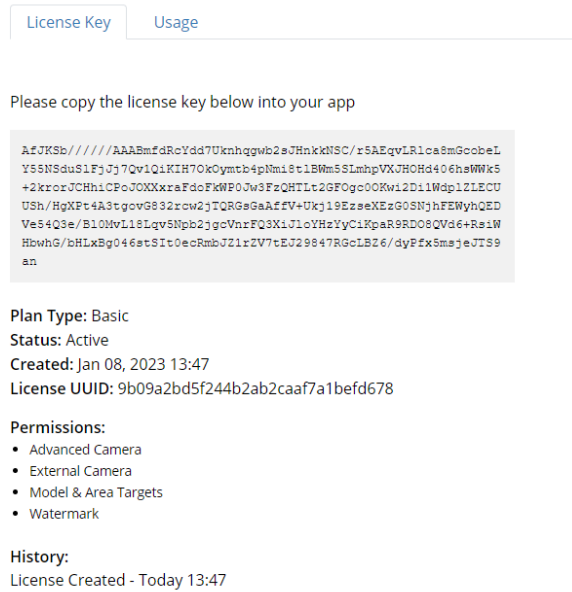
και κάνοντας κλικ στο κουμπί «Get Basic» θα ανοίξει η σελίδα Προσθήκη κλειδιού άδειας χρήσης, (Εικόνα 3.5).



Εικόνα 3.5: Vuforia License Manager, δημιουργία κλειδιού άδειας

Εδώ δίνεται επίσης η δυνατότητα ονομασίας του Licence για την εφαρμογή που θα το χρησιμοποιήσει ο χρήστης.

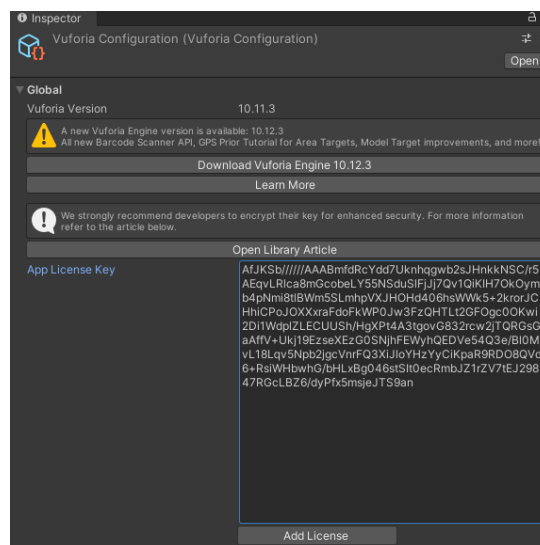
Αφού προστεθεί τιμή στο πεδίο License Name και αποδεχθούμε τους όρους του Vuforia Developer Agreement θα τελειώσει η διαδικασία δημιουργίας του κλειδιού και θα εμφανιστεί σε νέα σελίδα, (Εικόνα 3.6).



Εικόνα 3.6: Vuforia License Manager, παράδειγμα κλειδιού

3.4.4 Σύνδεση του Vuforia με το Unity

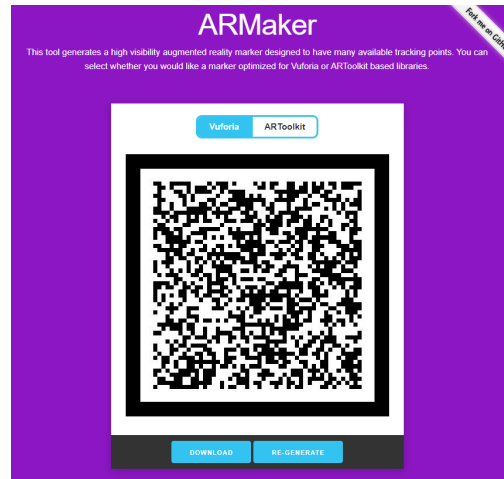
Το κλειδί που δημιουργήθηκε θα πρέπει να προστεθεί στο Unity οπότε αντιγράφηκε και επικολλήθηκε εντός του Unity στο πεδίο App License Key. Αυτό έγινε με τα παρακάτω βήματα (AR Camera → Vuforia Behaviour (Script) → Open Vuforia Engine configuration → Global), (Εικόνα 3.7).



Εικόνα 3.7: Vuforia Configuration

3.4.5 Δημιουργία QR codes

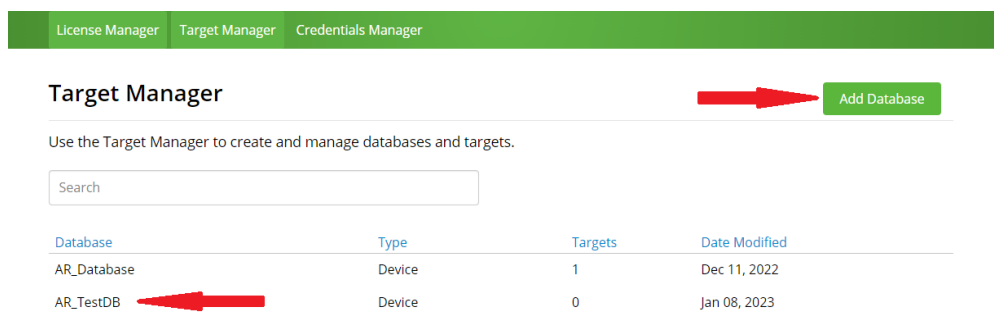
Με τα Image-Targets δίνεται η ευκαιρία στο Vuforia να αναγνωρίσει και να δημιουργήσει τα objects με ευκολότερο τρόπο. Ένα από τα πιο εύρηστα Image-Targets είναι τα QR codes. Τα QR codes είναι εύκολα στην δημιουργία εφόσον υπάρχουν πολλά sites που τα δημιουργούν δωρεάν. Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε το shawnlehner, [6], (Εικόνα 3.8).



Εικόνα 3.8: QR editor

3.4.6 Δημιουργία Image Target στο Vuforia

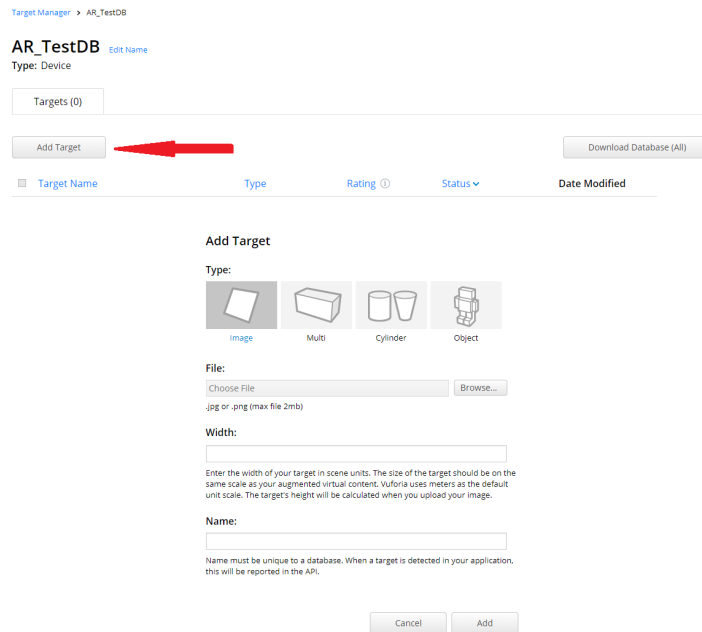
Αρχικά, αφού μεταβούμε στην ενότητα «Target Manager» της επίσημης ιστοσελίδας του Vuforia, κάνουμε κλικ στο κουμπί στο «Add Database» για να δημιουργηθεί η βάση που θα περιέχει τα Image Targets (QR codes), (Εικόνα 3.9).



Εικόνα 3.9: Vuforia, δημιουργία βάσης

Επιλέγοντας την βάση όπου δημιουργήθηκε νωρίτερα βλέπετε ότι ανοίγει μια νέα σελίδα όπως φαίνεται στη παρακάτω εικόνα 3.10.

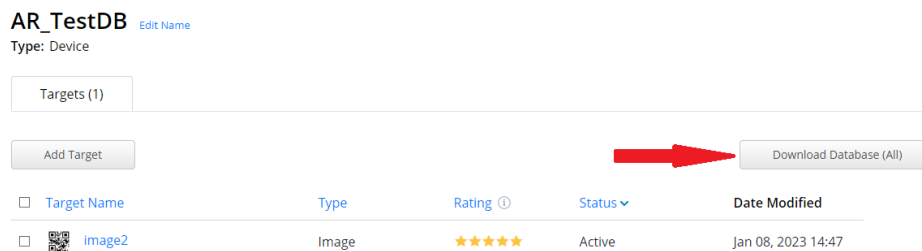
Κεφάλαιο 3



Εικόνα 3.10: Vuforia, προσθήκη Image Target

Σε αυτή τη σελίδα δίνεται η δυνατότητα προσθήκης των QR codes (images) που επιλέχθηκαν για να λειτουργήσουν σαν Image Targets στη βάση.

Αφού προστέθηκαν τα Targets που χρειάζονται για την εφαρμογή κάνουμε κλικ πάνω στο κουμπί «Download Database», (Εικόνα 3.11).



Εικόνα 3.11: Vuforia, λήψη βάσης

Το Vuforia περιέχει επίσης την επιλογή πλατφόρμας στην οποία θέλει ο χρήστης να χρησιμοποιήσει την βάση που έχει δημιουργήσει. Εδώ επιλέχθηκε η επιλογή του «Unity Editor», και η βάση κατέβηκε και αποθηκεύτηκε τοπικά, (Εικόνα 3.12).

Download Database

1 of 1 active targets will be downloaded

Name:
AR_TestDB

Select a development platform:

- Android Studio, Xcode or Visual Studio
- Unity Editor

Cancel

Download

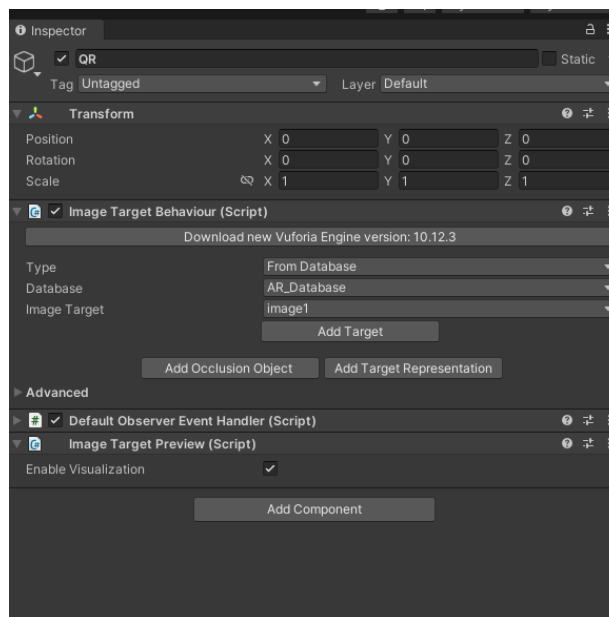
Εικόνα 3.12: Vuforia, λήψη βάσης για την πλατφόρμα του Unity

3.4.7 Προσθήκη βάσης στο Unity

Έχει δημιουργηθεί και κατέβει η βάση σε ένα τοπικό αρχείο.

Για να συνδεθεί το αρχείο με τη μηχανή Unity, πρέπει να γίνουν τα παρακάτω βήματα με την εξής σειρά: Asset → Import Package → Custom Package.

Στη συνέχεια, αφού επιλεγεί η ARCamera από το παράθυρο Hierarchy εμφανίζονται οι ρυθμίσεις της στο παράθυρο Inspector. Μεταβαίνοντας στο στοιχείο Vuforia Behavior (Script) και κάνοντας κλικ στο κουμπί «Open Vuforia configuration» προστίθεται το License Key μέσα στο πεδίο App License Key. Τέλος, ενώ έχει δημιουργηθεί το αντικείμενο στόχου (Vuforia Engine → Image Target) επιλέγεται η βάση με τα Image Targets όπως φαίνεται παρακάτω, (Εικόνα 3.13):



Εικόνα 3.13: Unity, εισαγωγή στόχου QR code

3.5 Blender

Το Blender είναι ένα ισχυρό και ευέλικτο λογισμικό δημιουργίας 3D που είναι ανοικτού κώδικα και χρησιμοποιείται για μια ευρεία γκάμα εργασιών, όπως 3D μοντελοποίηση, animation, rendering, επεξεργασία βίντεο, sculpting, και πολλά άλλα. Είναι δημοφιλές μεταξύ καλλιτεχνών, σχεδιαστών,

animator, προγραμματιστών παιχνιδιών και κινηματογραφιστών λόγω των εκτεταμένων δυνατοτήτων του και της ενεργής κοινότητάς του.

Μερικά από τα βασικά του χαρακτηριστικά είναι τα ακόλουθα [19] :

1. Μοντελοποίηση 3D: Το Blender προσφέρει μια ποικιλία εργαλείων και τεχνικών για τη δημιουργία τόσο οργανικών όσο και σκληρών επιφανειών σε 3D. Υποστηρίζει πολυγωνική μοντελοποίηση, sculpting, subdivision surfaces, και πολλά άλλα.
2. Animation: Με το Blender, υπάρχει η δυνατότητα δημιουργίας animations για αντικείμενα και χαρακτήρες χρησιμοποιώντας keyframes, rigging, inverse kinematics (IK), και περιορισμούς. Υποστηρίζει τόσο παραδοσιακά animation frame-by-frame όσο και πιο περίπλοκα animation για χαρακτήρες.
3. Rendering: Το Blender περιλαμβάνει ένα ισχυρό κινηματογραφικό μηχανισμό που ονομάζεται Cycles, επιτρέπει τη δημιουργία ρεαλιστικών εικόνων υψηλής ποιότητας.
4. Sculpting: Τα εργαλεία sculpting στο Blender επιτρέπουν να διαμορφώνονται ψηφιακά αντικείμενα σαν να πλάθουμε πηλό. Αυτό είναι ιδιαίτερα χρήσιμο για τη δημιουργία λεπτομερών οργανικών μοντέλων.
5. Open Source: Ένα από τα ξεχωριστά χαρακτηριστικά του Blender είναι ότι είναι open source. Αυτό σημαίνει ότι το λογισμικό είναι δωρεάν για χρήση, τροποποίηση και διανομή, ενισχύοντας μια ισχυρή κοινότητα προγραμματιστών και χρηστών που συνεχώς βελτιώνουν το λογισμικό.
6. Υποστήριξη Διαφορετικών Πλατφορμών: Το Blender είναι διαθέσιμο για Windows, macOS και Linux, εξασφαλίζοντας τη συνδεσιμότητα μεταξύ διαφορετικών λειτουργικών συστημάτων.

Στο κεφάλαιο 4 έχει γίνει η λεπτομερή αναφορά της αξιοποίησης του στην παρούσα Π.Ε.

3.6 Adobe Photoshop

Το Adobe Photoshop [21] είναι ένα ισχυρό λογισμικό επεξεργασίας ράστερ γραφικών (bitmap), ανεπτυγμένο από την Adobe Inc. Χρησιμοποιείται ευρέως από φωτογράφους, γραφίστες και δημιουργούς για την επεξεργασία εικόνων, τον σχεδιασμό γραφικών, τη δημιουργία ψηφιακής τέχνης κ.ο.κ.. Προσφέρει εργαλεία για την επεξεργασία φωτογραφιών, τη δημιουργία στρώσεων, τη διαχείριση χρωμάτων, την προσθήκη κειμένου και τον δημιουργικό σχεδιασμό. Με τη δυνατότητα συνεργασίας με άλλα προγράμματα Adobe, αποτελεί αναπόσπαστο εργαλείο για τη δημιουργία επαγγελματικών γραφικών και επεξεργασία φωτογραφιών.

Στη συγκεκριμένη Π.Ε., η αιτία της χρήσης του ήταν η δημιουργία των εικόνων που χρησιμοποιήθηκαν σαν κουμπιά, στην εφαρμογή του Unity. Επίσης, χρησιμοποιήθηκε και στην δημιουργία του logo και του app icon.

3.7 Audacity

Το Audacity είναι ένα δωρεάν, open source λογισμικό επεξεργασίας και ηχογράφησης ήχου, το οποίο χρησιμοποιείται ευρέως για διάφορες εργασίες που σχετίζονται με τον ήχο.

Μερικά βασικά στοιχεία του Audacity [22]:

1. Επεξεργασία Ήχου: Το Audacity επιτρέπει στον χρήστη να επεξεργάζεται ηχητικά αρχεία με μια μεγάλη ποικιλία εργαλείων. Να κόβει, να αντιγράφει, να επικολλά και να διαγράφει

ενότητες του ήχου, καθώς και να εφαρμόζεται διάφορα εφέ, όπως εξισορρόπηση, μείωση θορύβου, αλλαγή τόνου και άλλα.

2. Ηχογράφηση: Χρησιμοποιείται επίσης για να καταγράψει ήχο από διάφορες πηγές, συμπεριλαμβανομένων μικροφώνων, γραμμικών εισόδων και ακόμη και ήχου που ρέει από τον υπολογιστή σας. Παρέχει επιλογές για τη ρύθμιση των ρυθμίσεων ηχογράφησης και την παρακολούθηση της εισόδου.
3. Εισαγωγή και Εξαγωγή: Το Audacity υποστηρίζει μια ευρεία γκάμα μορφών αρχείων ήχου τόσο για την εισαγωγή όσο και για την εξαγωγή ήχου. Αυτό επιτρέπει στον χρήστη να εργαστεί με υπάρχοντα ηχητικά αρχεία και να μοιραστεί τα επεξεργασμένα έργα σας σε διάφορες μορφές.
4. Ανάλυση Ήχου: Το λογισμικό περιλαμβάνει εργαλεία για την ανάλυση ήχου, όπως φασματογραφήματα, οπτικοποιήσεις μορφών κυματώσεων και τη δυνατότητα οπτικοποίησης των συχνοτήτων και των πλατών ήχου.
5. Ανοικτού Κώδικα: Το Audacity είναι λογισμικό ανοικτού κώδικα, πράγμα που σημαίνει ότι είναι δωρεάν για χρήση, τροποποίηση και διανομή. Αυτό δημιουργεί μια κοινότητα συνεισφερόντων που συνεχώς βελτιώνουν το λογισμικό και παρέχουν υποστήριξη.

Στη παρούσα πτυχιακή εργασία το Audacity χρησιμοποιήθηκε για το τριμάρισμα και την μετατροπή ήχων σε αρχεία που δέχεται για επεξεργασία το Unity.

3.8 Επίλογος

Στο τρίτο κεφάλαιο έγινε αναφορά στα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της παρούσας πτυχιακής εργασίας καθώς και ο τρόπος σύνδεσης αναμεταξύ τους.

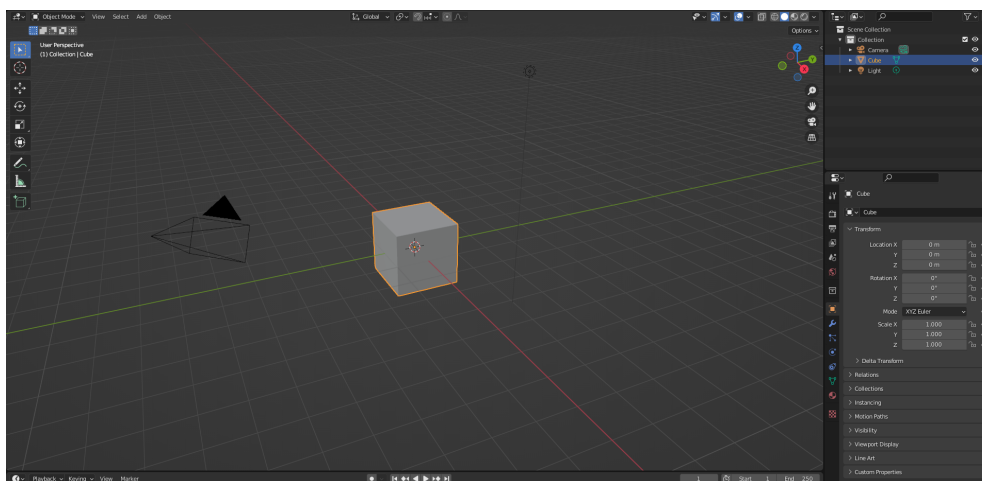
Κεφάλαιο 4ο: Ανάπτυξη 3D αντικειμένων

4.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζεται ο τρόπος δημιουργίας των αντικειμένων που χρειάστηκαν. Όλα τα αντικείμενα δημιουργήθηκαν με την βοήθεια της πλατφόρμας του Blender μιας και είναι ένα ισχυρό εργαλείο για τη δημιουργία ρεαλιστικών τρισδιάστατων αντικειμένων και σκηνών. Η επίτευξη ρεαλισμού στο Blender περιλαμβάνει διάφορες τεχνικές οι οποίες αναφέρθηκαν στο Κεφάλαιο 3.

Στην επόμενη ενότητα γίνεται λεπτομερής παρουσίαση της διαδικασίας που ακολουθήθηκε για την δημιουργία ενός εκ των αντικειμένων του πρακτικού μέρους της πτυχιακής εργασίας. Όσον αφορά τα υπόλοιπα αντικείμενα της εφαρμογής, γίνεται αναφορά σε αυτά μέσω εικόνας. Αυτή η απόφαση πάρθηκε με σκοπό να αποφευχθεί η επανάληψη στο κείμενο μιας και η δημιουργία τους έγινε με παραπλήσιο τρόπο.

4.2 Περιβάλλον Blender



Εικόνα 4.1: Περιβάλλον Blender

Μόλις ανοίξει η πλατφόρμα, ο χρήστης συναντά το παραπάνω περιβάλλον, (Εικόνα 4.1). Στα δεξιά της πλατφόρμας υπάρχουν δύο μέρη:

1. Η λίστα με τα αντικείμενα
2. Μια λίστα με διάφορες ρυθμίσεις που σχετίζονται με το επιλεγμένο αντικείμενο

Στα αριστερά της πλατφόρμας υπάρχει ένα μενού που περιλαμβάνει τις βασικές επιλογές που αλλάζουν ανάλογα στο edit mode που επιλέγει ο χρήστης.

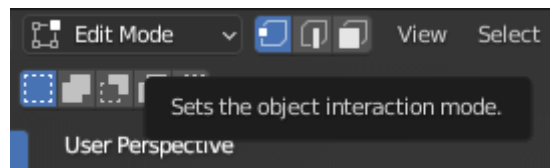
4.3 Δημιουργία Αντικειμένων

Αρχικά όπως βλέπουμε στην Εικόνα 4.1 καθώς ανοίγει το Blender το περιβάλλον του ήδη περιέχει έναν κύβο, στις επιλογές του Blender υπάρχουν διάφορα 3D σχήματα οποία είναι χρήσιμα για την δημιουργία του νέου 3D αντικειμένου.

Ένα από τα αντικείμενα που δημιουργήθηκαν για την παρούσα πτυχιακή εργασία ήταν η κιθάρα, παρακάτω αναλύεται βήμα βήμα ο τρόπος δημιουργία της καθώς και τα εργαλεία που χρησιμοποιήθηκαν για να της δώσουν το τελικό σχήμα της.

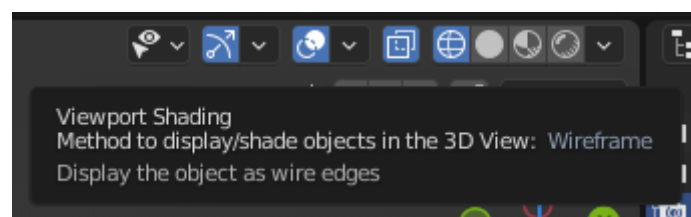
Κεφάλαιο 4

Σε πρώτη φάση, επιλέχθηκε ο κύβος σαν αντικείμενο επεξεργασίας. Η μετάβαση από ‘Object Mode’ σε ‘Edit Mode’ (Εικόνα 4.2) προσφέρει περισσότερες επιλογές για τον τρόπο επεξεργασίας του αντικειμένου. Παρακάτω αναφέρονται οι επιλογές που χρησιμοποιήθηκαν.



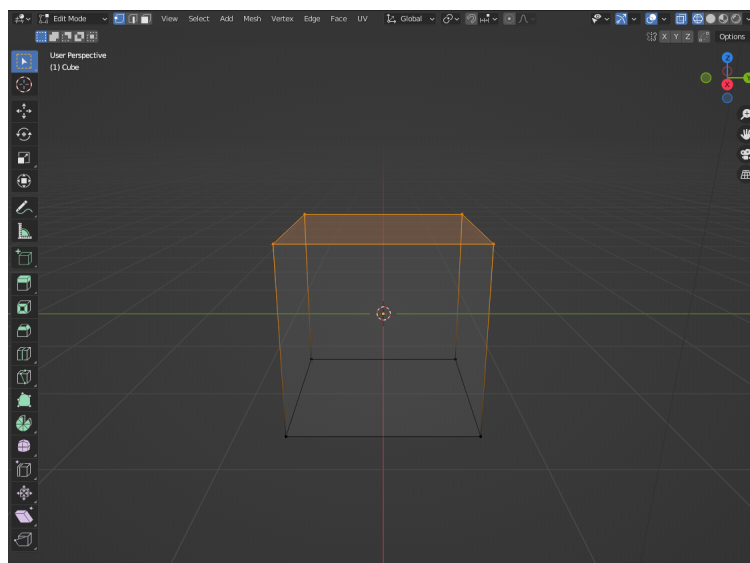
Εικόνα 4.2: Edit Mode

Επόμενη αλλαγή ήταν ο τρόπος που εμφανίζεται στην σκηνή το αντικείμενο, επιλέχθηκε το “Wireframe” (Εικόνα 4.3) που εμφανίζει μόνο τις άκρες (wireframes), (Εικόνα 4.4), των αντικειμένων στη σκηνή.



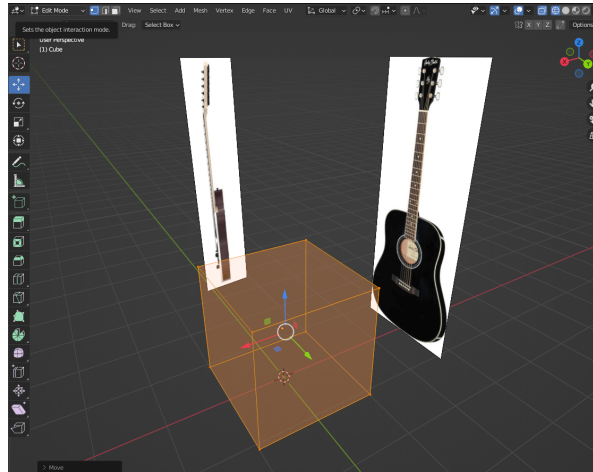
Εικόνα 4.3: Viewport Shading, Wireframe

Με αυτόν τον τρόπο η επιλογή των πτυχών του αντικειμένου ήταν ευκολότερη.



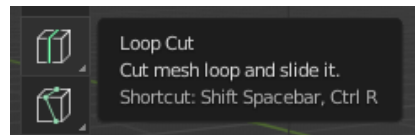
Εικόνα 4.4: Προβολή αντικειμένου σε Wireframe

Για την διευκόλυνση στην δημιουργία της κιθάρας, χρησιμοποιήθηκαν σαν πρότυπα δύο εικόνες της ίδιας κιθάρας, μπροστινή και πλαϊνή για να υπάρχει η αίσθηση του ύψους, όγκου κτλ του αντικειμένου και να φτιαχτεί αναλόγως, (Εικόνα 4.5).



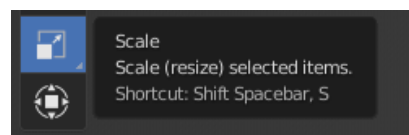
Εικόνα 4.5: Πρότυπο Κιθάρας

Για να μπορέσουμε να φτάσουμε τον κύβο σε παρόμοιο σχήμα όπως αυτό στην πρόσοψη χρησιμοποιήθηκε η επιλογή “Loop Cut”, (Εικόνα 4.6), το εργαλείο αυτό χωρίζει έναν βρόχο προσόψεων εισάγοντας νέους βρόχους άκρων που τέμνουν την επιλεγμένη άκρη.

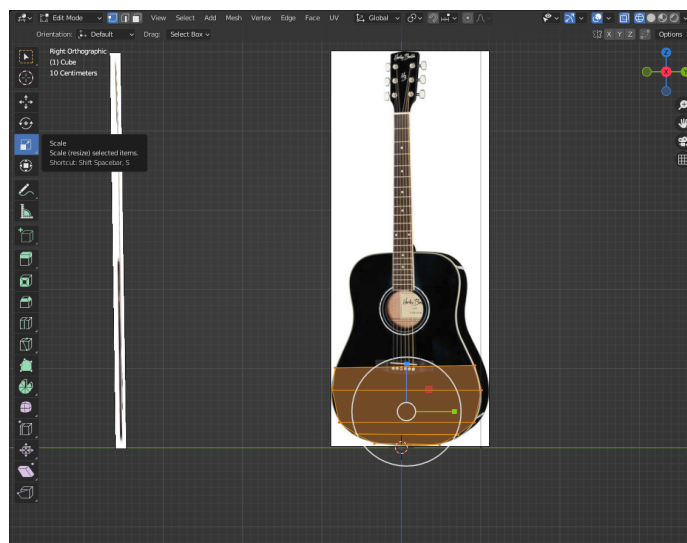


Εικόνα 4.6: Εργαλείο Loop Cut

Ακόμα ένα εργαλείο που χρησιμοποιήθηκε ήταν το “Scale”, (Εικόνα 4.7), που βοήθησε στην αλλαγή μεγέθους των άκρων, (Εικόνα 4.9).

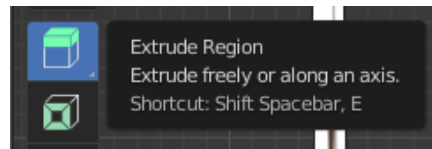


Εικόνα 4.7: Εργαλείο Scale



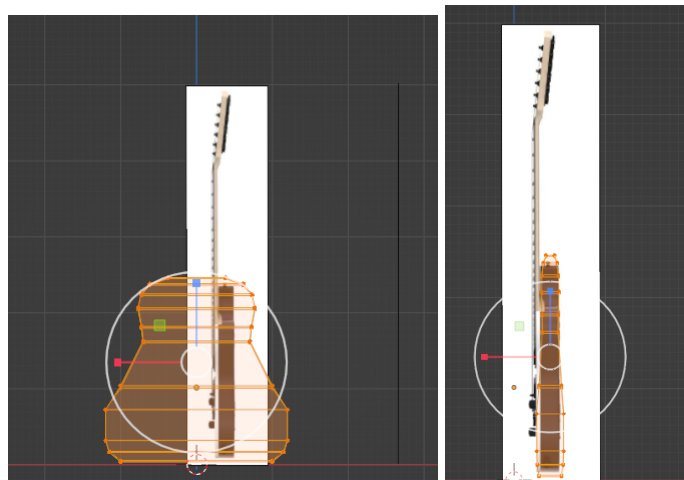
Εικόνα 4.8: Χρήση “Loop Cut” και “Scale”

Ο βασικός κορμός που φτιάχτηκε όμως χρειαζόταν μια επιμήκυνση γιατί ο αρχικός κύβος δεν ήταν αρκετός για τον κορμό. Σε αυτό βοήθησε το εργαλείο “Extrude Region” το οποίο αντιγράφει τις κορυφές, ενώ διατηρεί τη νέα γεωμετρία συνδεδεμένη με τις αρχικές κορυφές. Οι κορυφές μετατρέπονται σε άκρες και οι άκρες θα σχηματίσουν όψεις.



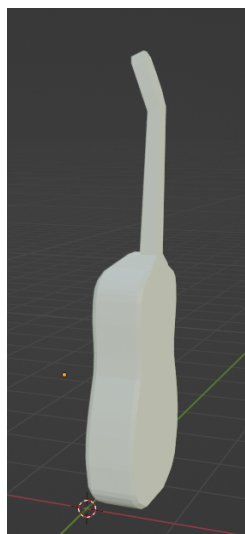
Εικόνα 4.9: Εργαλείο “Extrude Region”

Το αντικείμενο έφτασε σε μια μορφή όπως την αριστερή μεριά της (Εικόνα 4.10) και με το εργαλείο Scale έφτασε στη μορφή που βλέπουμε δεξιά, κοντά στον κορμό της κιθάρας στην εικόνα.



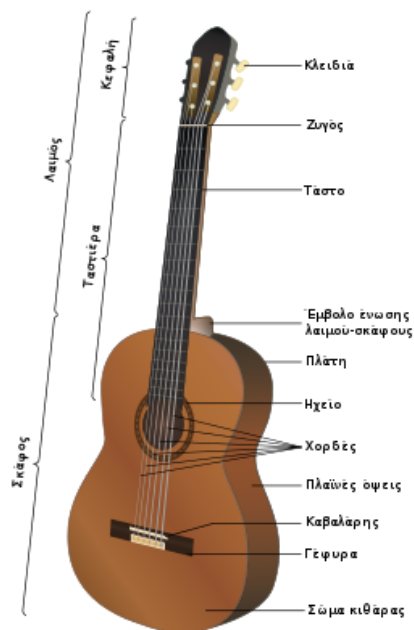
Εικόνα 4.10: Κορμός Κιθάρας

Έπειτα αλλάχτηκε ο άξονας της οπτικής γωνίας του αντικειμένου στον Z έτσι ώστε να γίνει πάλι η χρήση του “Loop Cut”. Με την χρήση του “Loop Cut” δημιουργήθηκαν γωνίες όπου επεκτάθηκαν με το εργαλείο “Extrude Region” και πήραν την μορφή του μπράτσου, (Εικόνα 4.11).



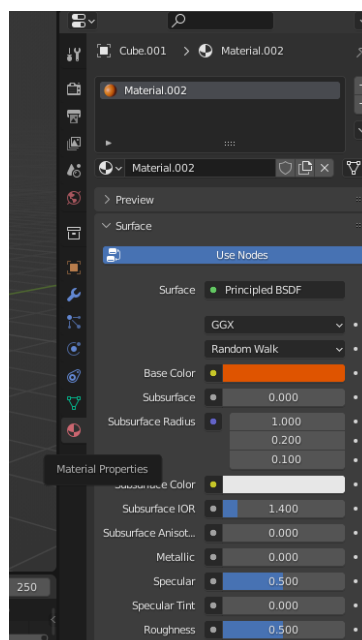
Εικόνα 4.11: Μορφή Κιθάρας

Με τον ίδιο τρόπο δημιουργήθηκαν όλα τα υπόλοιπα κομμάτια (Εικόνα 4.12) που συντελούν την κιθάρα δηλαδή τα κλειδιά, ο ζυγός, το ταστό, το έμβολο ένωσης λαιμού-σκάφους, το ηχείο, οι χορδές, ο καβαλάρης και η γέφυρα.



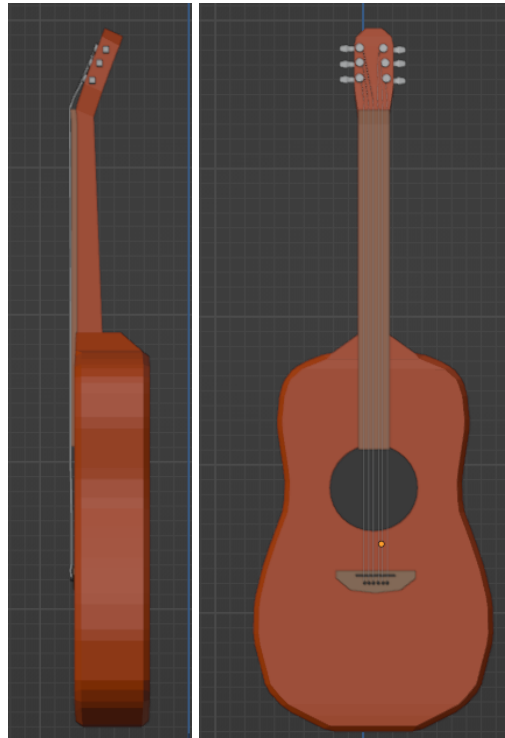
Εικόνα 4.12: Μέρη κιθάρας

Το επόμενο βήμα ήταν η επιλογή χρώματος σε κάθε ένα από αυτά τα κομμάτια. Το κάθε αντικείμενο έχει ρυθμίσεις για “Material Properties” (Εικόνα 4.13) όπου εκεί γίνεται μεταξύ άλλων και η επιλογή του χρώματος.



Εικόνα 4.13: Material Properties

Έτσι το τελικό αποτέλεσμα είναι το παρακάτω στην Εικόνα 4.14.



Εικόνα 4.14: Τελικό αποτέλεσμα

4.4 Λίστα Αντικειμένων

Τα αντικείμενα που δημιουργήθηκαν για την παρούσα πτυχιακή εργασία.

1. Συνθέτες

Δημιουργήθηκαν πέντε διαφορετικοί συνθέτες που εμφανίζονται στις σκηνές που θα αναφερθούν στο Κεφάλαιο 5. Αυτοί είναι ο Μπαχ, ο Μπετόβεν, ο Σοπεν, ο Ντεμπυσσύ και ο Μότσαρτ (με την ίδια σειρά στην Εικόνα 4.15).



Εικόνα 4.15 : Συνθέτες

2. Μουσικά όργανα

Για τις ερωτήσεις που καλείται ο χρήστης να απαντήσει στα quiz της εφαρμογής, δημιουργήθηκαν συνολικά δεκαπέντε μουσικά όργανα. Αυτά ήταν το τσέλο, κλαρινέτο,

ντραμς, φλογέρα, κιθάρα, ηλεκτρική κιθάρα, Κρητική λύρα, λαούτο, Ποντιακή λύρα, σαξόφωνο, Σύριγγα του Πανός, τρομπέτα, πιάνο (χωρίς ουρά), βιολί και μεταλλόφωνο, (Εικόνα 4.16).



Εικόνα 4.16 : Μουσικά όργανα

3. Αντικείμενα σκηνής

Δημιουργήθηκαν και διάφορα περιφερειακά αντικείμενα για την διακόσμηση της σκηνής. Στο σύνολο ήταν οκτώ. Τα αντικείμενα αυτά ήταν: πίνακας, τραπέζια, πικάπ, βιβλίο, αναλόγιο, η σκηνή και αστέρια (Εικόνα 4.17).



Εικόνα 4.17 : Περιφερειακά αντικείμενα

4.5 Επίλογος

Στο τρίτο κεφάλαιο έγινε αναφορά στα εργαλεία που χρησιμοποιήθηκαν για την δημιουργία των 3D αντικειμένων με την βοήθεια της πλατφόρμας Blender καθώς και η αναφορά όλων των αντικειμένων που δημιουργήθηκαν.

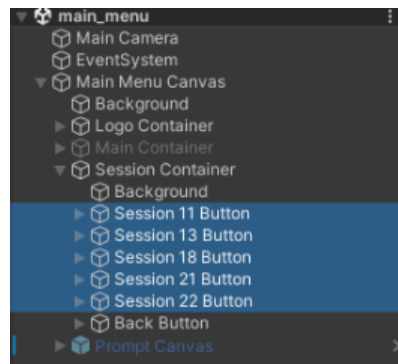
Κεφάλαιο 5ο: Παρουσίαση εφαρμογής

5.1 Εισαγωγή

Στο παρόν κεφάλαιο θα γίνει η παρουσίαση του πρακτικού κομματιού της πτυχιακής εργασίας. Θα ξεκινήσουμε με το αρχικό μενού της εφαρμογής. Έπειτα θα αναλύσουμε την κάθε σκηνή/quiz που έχει δημιουργηθεί για να βοηθήσει τους μαθητές με την ύλη που διδάσκονται καθώς η εφαρμογή καλύπτει κάποια από τα κεφάλαια του βιβλίου. Τέλος υπάρχουν κάποιες έξτρα επιλογές που θα βοηθήσουν τους μαθητές να εξασκηθούν και να λάβουν πληροφορίες με έναν πιο διασκεδαστικό τρόπο μέσα από οπτικοακουστικές δραστηριότητες.

5.2 Αρχικό Μενού

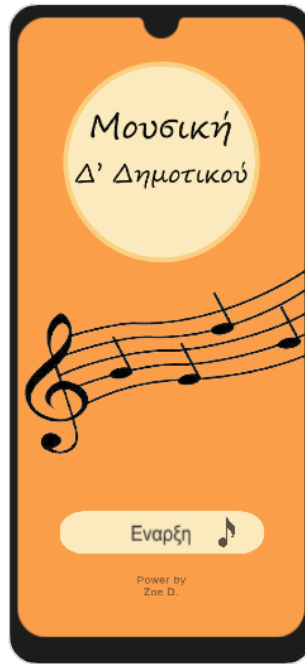
Ξεκινώντας με το αρχικό μενού, η κύρια δομή της σκηνής Main Menu είναι η ακόλουθη (Εικόνα 5.1).



Εικόνα 5.1: Main menu

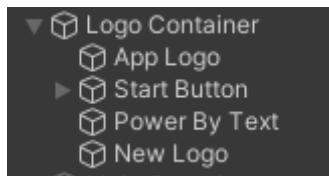
Το Main Menu Canvas είναι το κύριο gameObject της σκηνής. Αποτελείται από ένα αντικείμενο Background, τρία child gameObjects (Logo, Main και Session Container) που λειτουργούν ως “containers” για τις διάφορες επιλογές που προσφέρει το menu στον χρήστη και το Prompt Canvas.

Μόλις ανοίξει η εφαρμογή, ο χρήστης έρχεται πρώτα σε επαφή με το Logo Container έτσι αντικρίζει το αρχικό μενού της εφαρμογής (Εικόνα 5.2).



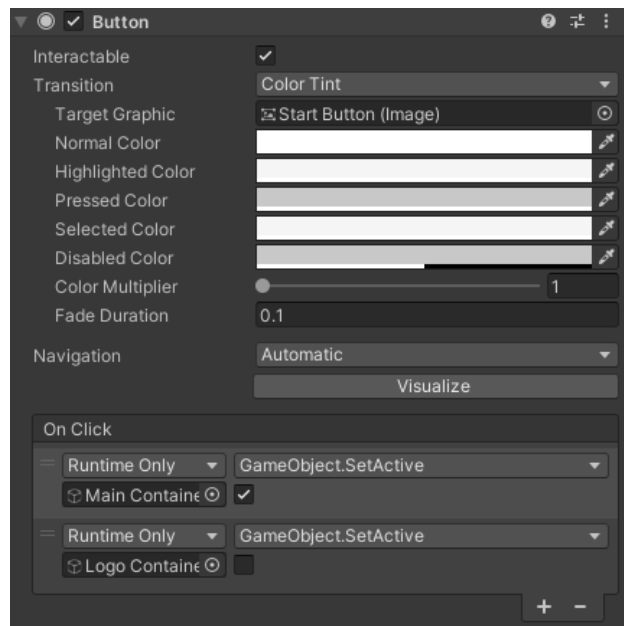
Εικόνα 5.2: Έναρξη εφαρμογής

Το Logo Container πέρα από τα sprites (εικόνες), διαθέτει και το κουμπί «Εναρξη» (Εικόνα 5.3).



Εικόνα 5.3: Logo Container

Το Start Button (Εναρξη), όπως και κάθε κουμπί στη πλατφόρμα του Unity, διαθέτει ένα button component. Η βασική λειτουργία του component αυτού είναι το UnityEvent OnClick, μια μέθοδος του Unity [10]. Όταν ο χρήστης κλικάρει πάνω στο Start Button, ενεργοποιεί το gameObject Main Container και απενεργοποιεί το Logo Container (Εικόνα 5.4).

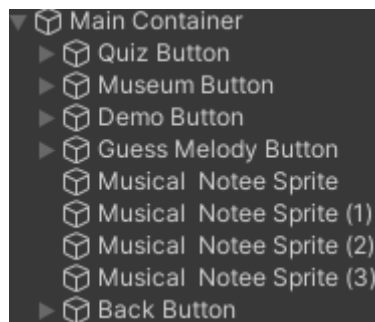


Εικόνα 5.4: Button Component

Με αυτόν τον τρόπο ο χρήστης μεταβαίνει στην κύρια «ενότητα» της Main Menu σκηνής. Εκεί θα συναντήσει όλες τις επιλογές-δραστηριότητες που προσφέρει το πρακτικό μέρος της πτυχιακής, AR-Music.

Το κύριο μενού χωρίζεται σε τέσσερις κατηγορίες (Εικόνα 5.5):

- Quiz
- Μουσείο Μουσικής
- Μεταλλόφωνο
- Άκου - Βρες!



Εικόνα 5.5: Main Container

Το κουμπί «Quiz» αφορά εξέταση γενικών γνώσεων του βιβλίου μέσω διαφόρων ερωτήσεων πολλαπλής επιλογής (πέντε ξεχωριστά κεφάλαια). Στην ενότητα «Μουσείο Μουσικής» ο χρήστης/μαθητής βρίσκει σημαντικές πληροφορίες σχετικά με μουσικούς συνθέτες και μουσικά όργανα που περιλαμβάνονται στην ενότητα «Quiz». Με το κουμπί «Μεταλλόφωνο» μεταβαίνει σε σκηνή όπου του δίνεται η δυνατότητα να παίξει με ένα AR Μεταλλόφωνο. Και τέλος στην ενότητα «Άκου - Βρες!» εξετάζεται σε ερωτήσεις πολλαπλής επιλογής, όπου ο χρήστης ακούει μια μελωδία και προσπαθεί να βρει τον τίτλο της (Εικόνα 5.6).



Εικόνα 5.6: Κύριο Μενού

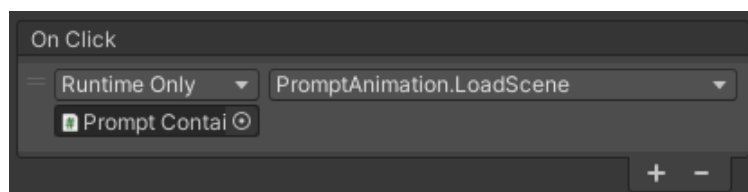
Πατώντας σε κάποιο από τα κουμπιά “Μουσείο Μουσικής”, “Μεταλλόφωνο” ή “Άκου - Βρες!” ενεργοποιείτε το αντικείμενο Prompt Canvas που έχει την ιδιότητα του prefab[16] (Εικόνα 5.7). Στο Unity, το "prefab" είναι ένα πολύ χρήσιμο κομμάτι της διαδικασίας ανάπτυξης παιχνιδιών. Ένα prefab είναι ουσιαστικά ένα προ-κατασκευασμένο αντικείμενο ή σύνολο αντικειμένων που μπορεί να δημιουργηθεί και να το επαναχρησιμοποιηθεί εύκολα σε διάφορα μέρη του παιχνιδιού.



Εικόνα 5.7: Prompt Canvas

Αυτό είναι ιδιαίτερα βολικό μιας και το συγκεκριμένο gameObject επαναχρησιμοποιείται σε όλες τις σκηνές που έχει η εφαρμογή και αν χρειαστεί να γίνει κάποια αλλαγή, αυτή εφαρμόζεται σε όλες τις σκηνές αυτόματα.

Το “Yes” Button εκτελεί την public μέθοδο LoadScene της κλάσης PromptAnimation (Εικόνα 5.8) και χρησιμοποιεί την private μεταβλητή _sceneName για να φορτώσει την όποια σκηνή μέσω της μεθόδου LoadScene του SceneManager. Υπενθυμίζεται ότι η μεταβλητή _sceneName παίρνει τιμές από τα διάφορα buttons του Main Menu Container και της μεθόδου SetSceneName (Εικόνα 5.9).



Εικόνα 5.8: Yes button component

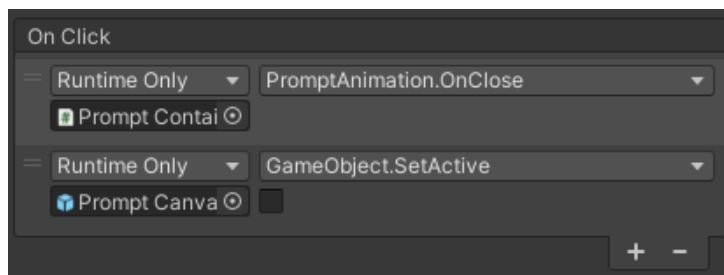
```
public void LoadScene()
{
    Debug.Log(_sceneName);
    if (_sceneName == null) { _sceneName = "main_menu"; }
    SceneManager.LoadScene(_sceneName);
}
```

Εικόνα 5.9: Yes button component

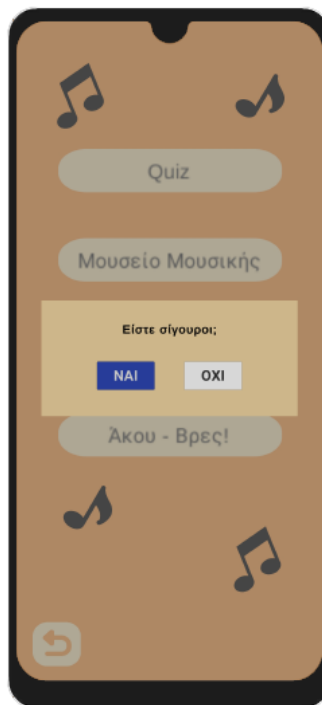
Από την άλλη, το “Close” Button κάνει trigger την public μέθοδο OnClose (Εικόνα 5.10) της ίδιας κλάσης και απενεργοποιεί (κάνει disable) το gameObject Prompt Canvas, αφού παίξει πρώτα το animation (Εικόνα 5.11-5.12).

```
0 references
public void OnClose()
{
    LeanTween.scale(_promptContainer, new Vector3(0, 0, 0), 0.25f).setEase(_outType);
}
```

Εικόνα 5.10: OnClose μέθοδος



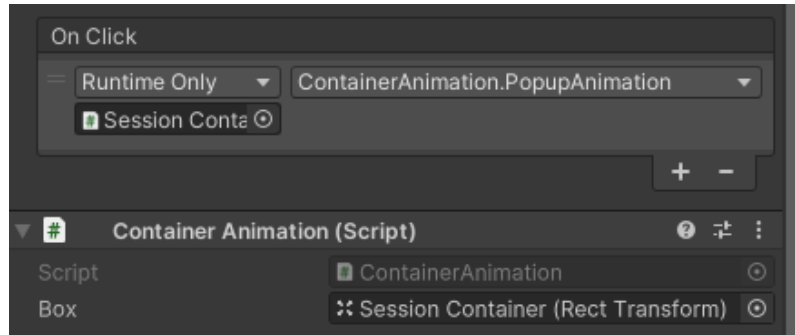
Εικόνα 5.11: Close button component



Εικόνα 5.12: Επίπεδο UI

5.3 Quiz

Πατώντας στο πρώτο κουμπί ο χρήστης θα πλοηγηθεί στην κατηγορία “Quiz”. Το ”Quiz” Button χρησιμοποιεί την κλάση ContainerAnimation η οποία διαχειρίζεται το Rect Transform [13] του Session Container, (Εικόνα 5.13).



Εικόνα 5.13: Quiz Button, Button component

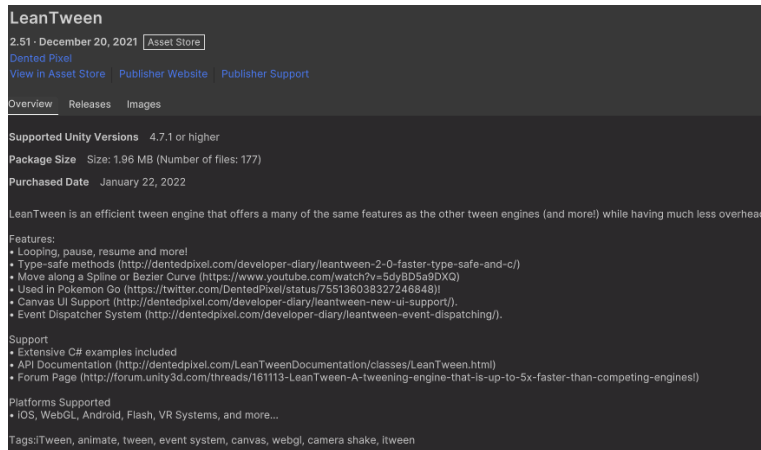
Πιο συγκεκριμένα, όπως και με το κουμπί «Εναρξη» έτσι και εδώ, γίνεται χρήση του UnityEvent OnClick το οποίο με την σειρά του κάνει trigger την public μέθοδο της κλάσης ContainerAnimatin ονόματι PopupAnimation , (Εικόνα 5.14), και αναδύεται από το κάτω μέρος της οθόνης το Session Container όπου εκεί περιέχονται τα “Κεφάλαια” προς εξέταση.

```

4 public class ContainerAnimation : MonoBehaviour
5 {
6     public Transform box;
7
8     @ Unity Message | 0 references
9     public void Awake()
10    {
11        box.localPosition = new Vector2(0.5f, -Screen.height);
12    }
13
14    @ references
15    public void PopupAnimation()
16    {
17        box.localPosition = new Vector2(0, -Screen.height);
18        box.LeanMoveLocalY(0, 0.5f).setEaseOutExpo().delay = 0.1f;
19    }
20
21    @ references
22    public void CloseDialog()
23    {
24        box.LeanMoveLocalY(-Screen.height, 0.5f).setEaseInExpo();
25    }
26 }
  
```

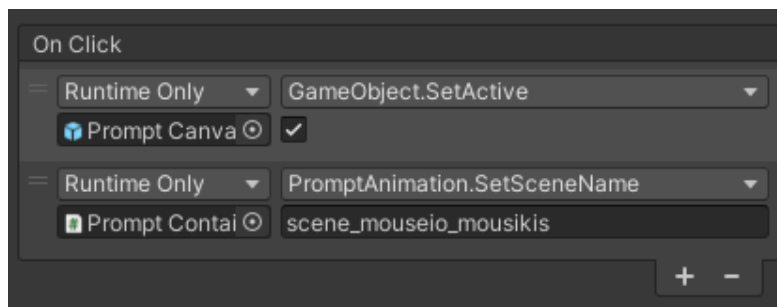
Εικόνα 5.14: ContainerAnimatin class

Για τα εφέ του animation έχει χρησιμοποιηθεί ένα third party package, το LeanTween (Εικόνα 5.15)

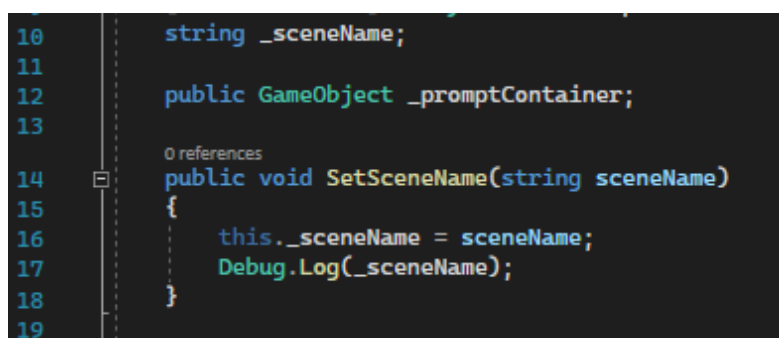


Εικόνα 5.15: LeanTween package

Οι υπόλοιπες ενότητες, “Μουσικό Μουσείο”, “Μεταλλόφωνο” και “Άκου-Βρες!”, έχουν πανομοιότυπη δομή όσον αφορά το button component. Μόλις κάνει κλικ ο χρήστης στο εκάστοτε κουμπί, ενεργοποιείται το gameObject Prompt Canvas και εκτελείται η public μέθοδος SetSceneName της κλάσης PromptAnimation (Εικόνα 5.16). Η μέθοδος αυτή λειτουργεί δυναμικά, με την έννοια ότι δέχεται ως παράμετρο το όνομα του scene έτσι μπορεί να χρησιμοποιηθεί σε πολλαπλά αντικείμενα χωρίς κανένα απολύτως πρόβλημα (Εικόνα 5.17).

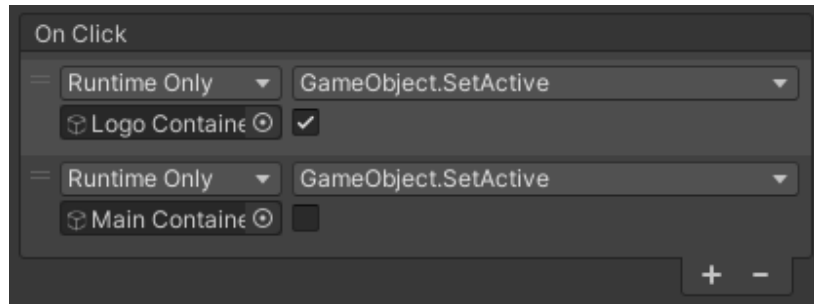


Εικόνα 5.16: Button Component, onClick μέθοδος



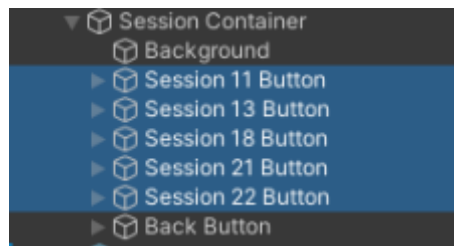
Εικόνα 5.17: SetSceneName μέθοδος

Το Back Button ακολουθεί αντίστοιχη φιλοσοφία με αυτή του Start Button που αναφέρθηκε παραπάνω. Εν ολίγοις, ενεργοποιεί το gameObject Logo Container και απενεργοποιεί το gameObject Main Container (Εικόνα 5.18) . Από την πλευρά του χρήστη, φαίνεται πως επιστρέφει στην αρχική “σελίδα”.



Εικόνα 5.18: Back Button, Button component

Το Session Container δεν διαφέρει σε μεγάλο βαθμό από όσα έχουν προηγηθεί. Πιο συγκεκριμένα, στο “container” αυτό υπάρχουν διαθέσιμα τα πέντε κεφάλαια προς εξέταση (Εικόνα 5.19).



Εικόνα 5.19: Session Container

Τα κεφάλαια που βλέπει ο χρήστης είναι:

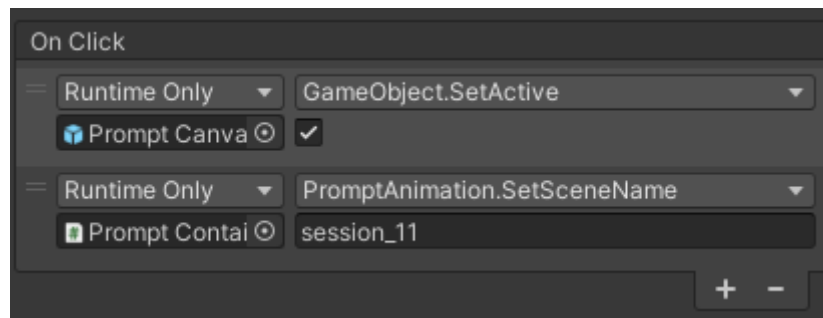
- Κεφάλαιο 11
- Κεφάλαιο 13
- Κεφάλαιο 18
- Κεφάλαιο 21
- Κεφάλαιο 22

όπως προδίδει και ο τίτλος τους σχετίζονται με κάποια από τα κεφάλαια που καλούνται οι διδάσκοντες με τους μαθητές να καλύψουν κατά την διάρκεια της σχολικής χρονιάς (Εικόνα 5.20).



Εικόνα 5.20: Κεφάλαια

Τα κουμπιά που φαίνονται στην παραπάνω εικόνα λειτουργούν όπως και τα κουμπιά του Main Container, πλην του “Quiz” Button. Το ίδιο ισχύει για το “Back” Button, (Εικόνα 5.21).

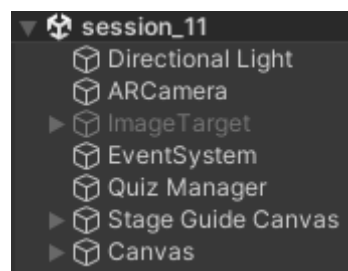


Εικόνα 5.21: Κεφάλαια Buttons, Button component

5.3.1 Κεφάλαιο 11

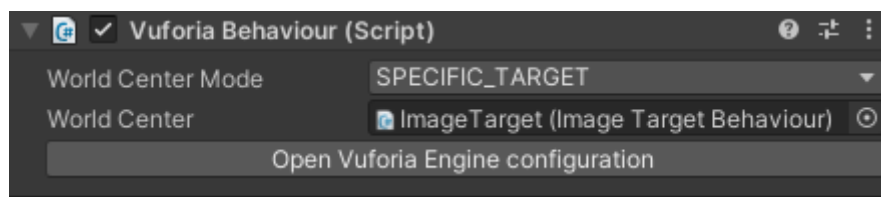
Η σκηνή του “Κεφαλαίου 11” σχετίζεται με το Κεφάλαιο 11 του βιβλίου μαθητή σελ. 39-40, όπου υπάρχουν ερωτοαπαντήσεις που ο χρήστης καλείται να απαντήσει .

Η δομή της σκηνής Session 11 (Κεφαλαίου 11) έχει ως εξής (Εικόνα 5.22):



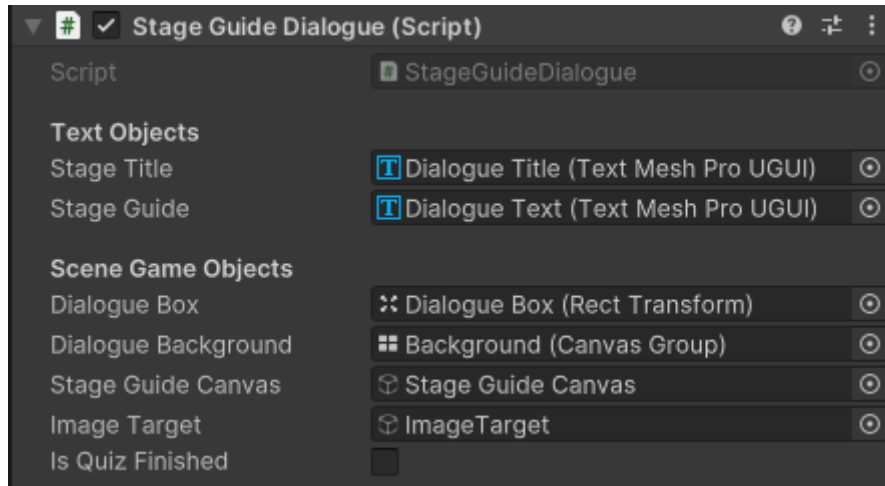
Εικόνα 5.22: Session 11

Το gameObject ARCamera (του Vuforia) έχει το παρακάτω setup (Εικόνα 5.23):



Εικόνα 5.23: AR Camera, Vuforia Behaviour

Αφού φορτώσει η σκηνή session_11 από την μηχανή γραφικών Unity, εμφανίζεται το Stage Guide Canvas στο οποίο περιέχονται οι οδηγίες της εκάστοτε σκηνής. Το child gameObject του Stage Guide Canvas, Dialogue Group χρησιμοποιεί την κλάση StageGuideDialogue (Εικόνα 5.24).



Εικόνα 5.24: Stage Guide Dialogue script setup

Αρχικά, μιας και το gameObject είναι ενεργό εκτελείται η μέθοδος OnEnable (Εικόνα 5.25) της Unity. Αυτό έχει ως αποτέλεσμα την εμφάνιση του UI αντικειμένου στην οθόνη της συσκευής μέσω ενός animation. Η υλοποίηση ενός “Stage Guide” πραγματοποιήθηκε για να διευκολύνει τον χρήστη/μαθητή στην κατανόηση του στόχου της κάθε σκηνής. Τέτοιες πρακτικές ακολουθούνται συνεχώς από εταιρείες λογισμικού στην βιομηχανία του gaming.

```

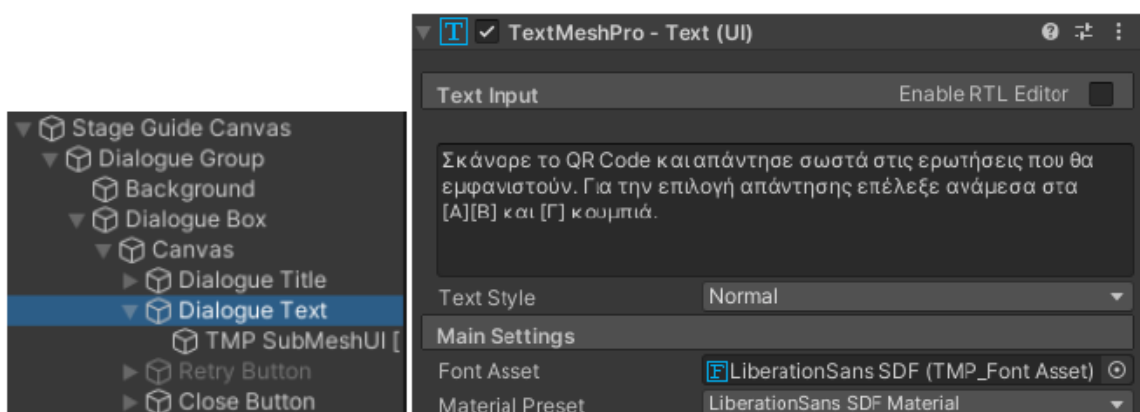
@ Unity Message | 0 references
void OnEnable()
{
    _dialogueBackground.alpha = 0;
    _dialogueBackground.LeanAlpha(1, 0.5f);

    _dialogueBox.localPosition = new Vector2(0, -Screen.height);
    _dialogueBox.LeanMoveLocalY(0, 0.5f).setEaseOutExpo().delay = 0.1f;
}

```

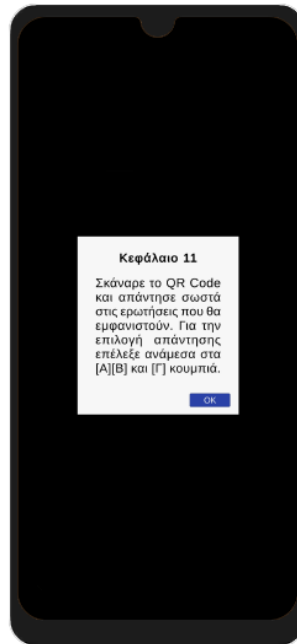
Εικόνα 5.25: OnEnable script

Τόσο ο τίτλος της σκηνής, όσο και το κείμενο με τις οδηγίες που υπάρχει δίνονται ως string texts μέσω του Unity Editor (Εικόνα 5.26).



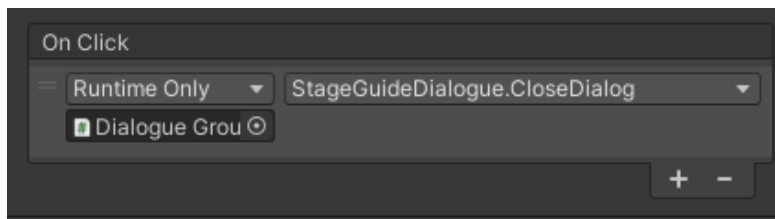
Εικόνα 5.26: Unity Editor

Ο χρήστης θα αντικρίσει το Popup με τις οδηγίες όπως φαίνεται στην (Εικόνα 5.27).



Εικόνα 5.27: Stage Guide Popup

Το κουμπί “OK” εκτελεί την public μέθοδο CloseDialog της κλάσης StageGuideDialogue. Αναλύοντας τον κώδικα που φαίνεται στην εικόνα, γίνεται αντιληπτό ότι η μέθοδος αυτή είναι υπεύθυνη για την απομάκρυνση του Stage Guide Canvas από την οθόνη της σκηνής και το initialization των απαραίτητων αντικειμένων για το σκανάρισμα του Image Target (Εικόνα 5.28) .



Εικόνα 5.28: OK button component

Στη συνέχεια, αφού προηγηθεί ο έλεγχος του αν τελείωσε ή όχι το Quiz, απενεργοποιείται το gameObject Stage Guide Canvas από την σκηνή με την βοήθεια της μεθόδου ToggleVisibility [16], (Εικόνα 5.29).

```

public void CloseDialog()
{
    _dialogueBackground.LeanAlpha(0, 0.5f);
    _dialogueBox.LeanMoveLocalY(-Screen.height, 0.5f).setEaseInExpo();
    StartCoroutine(ToggleVisibility(_imageTarget, true, 0));

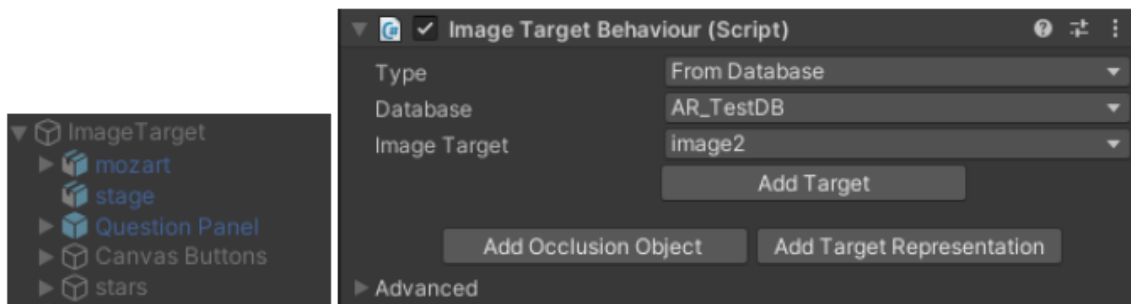
    if (_isQuizFinished == true)
    {
        LoadNextScene("main_menu");
    }
    else
    {
        StartCoroutine(ToggleVisibility(_stageGuideCanvas, false, 10));
    }
}

2 references
IEnumerator ToggleVisibility(GameObject gameObject, bool status, int seconds)
{
    gameObject.SetActive(status);
    yield return new WaitForSeconds(seconds);
}

```

Εικόνα 5.29: CloseDialog μέθοδος

Το αντικείμενο ImageTarget αποτελεί αντικείμενο του Vuforia [Εικόνα 5.30] και χρησιμοποιείται για την ανίχνευση και παρακολούθηση της εικόνας προς σκανάρισμα από την κάμερα της android συσκευής (Εικόνα 5.31).



Εικόνα 5.30: Image Target component



Εικόνα 5.31: Image Target του Κεφαλαίου 11

Μόλις ο χρήστης σκανάρει το Image Target της Εικόνας 5.27, εμφανίζονται αμέσως όλα τα αντικείμενα της σκηνής (Εικόνα 5.32). Ίδια πρακτική ακολουθείται και από τις υπόλοιπες σκηνές της εφαρμογής. Η σύνθεση του session_11 έχει ως εξής:

- ένα μουσικοσυνθέτη βοηθό

- τρία αναλόγια με ένα βιβλίο πάνω στο καθένα
- δύο αστέρια
- μία σκηνή
- έναν πίνακα



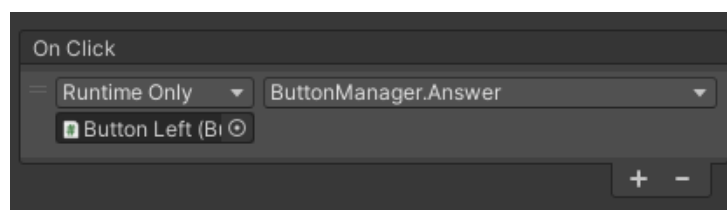
Εικόνα 5.32: Προβολή σκηνής κεφαλαίου 11

Έπειτα, όσον αφορά το στήσιμο της σκηνής και τον τρόπο των απαντήσεων, ο 3D πίνακας εμφανίζει πάνω του τις ερωτήσεις και 3 επιλογές απαντήσεων. Η απάντηση δίνεται από τον χρήστη πατώντας πάνω στα 3D αναλόγια τα οποία εμφανίζουν τα Α, Β και Γ σαν πιθανές απαντήσεις.

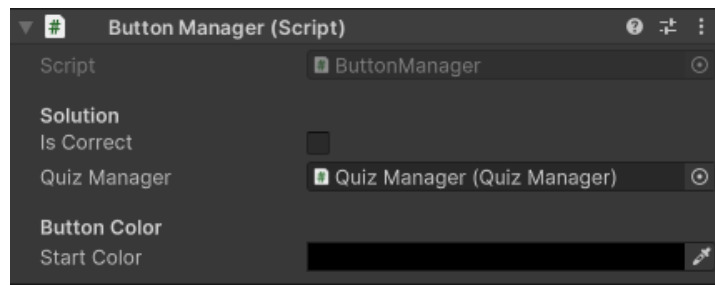
Τα αναλόγια είναι τρία buttons που περιέχει το Canvas Buttons (Εικόνα 5.33), τα οποία όπως ήταν αναμενόμενο ακολουθούν την ίδια δομή. Αυτό ουσιαστικά σημαίνει ότι το button component του κάθε button (Α, Β και Γ επιλογές) (Button Left, Button Middle και Button Right) (Εικόνα 5.34) εκτελεί την public μέθοδο της κλάσης ButtonManager (Εικόνα 5.35) μέσω του Unity Event OnClick.



Εικόνα 5.33: Canvas Button

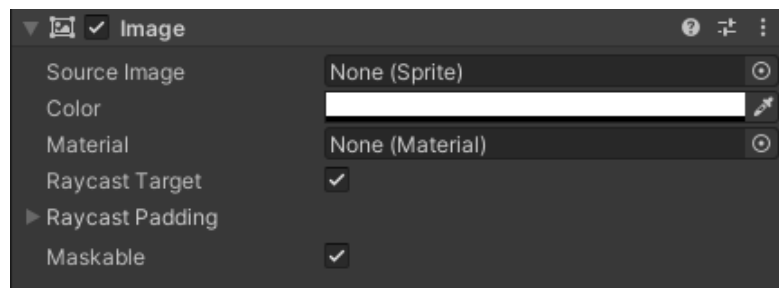


Εικόνα 5.34: button component



Εικόνα 5.35: Button Manager

Η μέθοδος αυτή απλά παίρνει το Image component (Εικόνα 5.36) του εκάστοτε button και αλλάζει το χρώμα του ανάλογα με την τιμή της μεταβλητής `_isCorrect` (Εικόνα 5.37).



Εικόνα 5.36: Image component

```
public void Answer()
{
    GetComponent<Image>().color = _isCorrect ? Color.green : Color.red;
    Debug.Log(_isCorrect ? "Correct answer!" : "Wrong answer!");

    _quizManager.Answer(_isCorrect);
}
```

Εικόνα 5.37: Answer μέθοδος

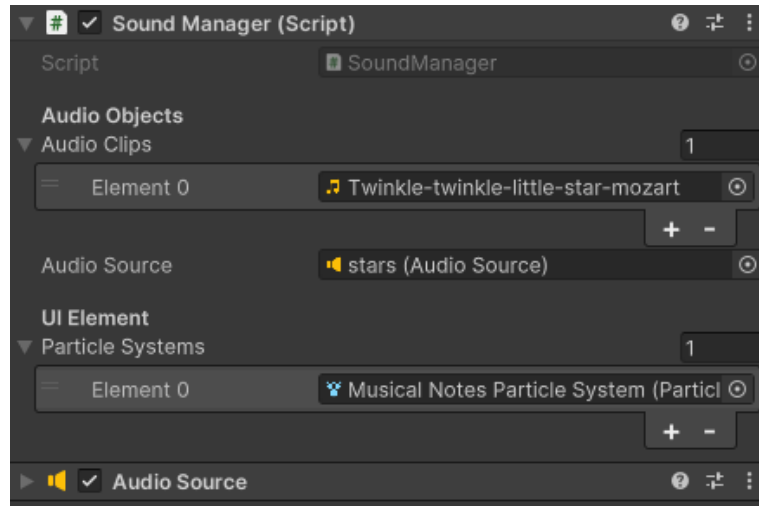
Όταν ο χρήστης επιλέγει μία από τις τρεις διαθέσιμες απαντήσεις εκτελείται η μέθοδος `QuizManager.RevealCorrectAnswer` (καλείται από την μέθοδο `QuizManager.Answer`). Η συγκεκριμένη μέθοδος φανερώνει στον χρήστη την σωστή απάντηση. Το background color του κουμπιού που περιέχει την σωστή απάντηση γίνεται πράσινο ενώ σε άλλη περίπτωση μετατρέπεται σε κόκκινο.

Στη σκηνή υπάρχει επίσης ο μουσικοσυνθέτης βοηθός, στην παρούσα σκηνή τον συγκεκριμένο ρόλο κατέχει ο Βόλφγκανγκ Αμαντέους Μότσαρτ. Η χρήση του είναι απλά διακοσμητική καθώς δεν υπάρχει κάποια υλοποίηση για να χρησιμοποιηθεί από τον χρήστη, προς το παρόν.

Ένα από τα πιο γνωστά μουσικά κομμάτια του Βόλφγκανγκ Αμαντέους Μότσαρτ είναι το “Twinkle Twinkle Little Star”, έτσι δημιουργήθηκε και η ιδέα της προσθήκης των δύο αστεριών στην σκηνή.

Το αντικείμενο Stars περιέχει τα 3D μοντέλα των δύο αστεριών, `star` και `starNote`, όπως επίσης τα particle systems για αυτά. Προκειμένου να γίνει εφικτή η αλληλεπίδραση με το συγκεκριμένο αντικείμενο ήταν απαραίτητη η προσθήκη ενός Box Collider [17].

Επιπλέον έχει γίνει attach η κλάση `SoundManager` αλλά και ενός Audio Source component (Εικόνα 5.38).



Εικόνα 5.38: Sound Manager Unity Editor

Όσον αφορά την αλληλεπίδραση καθαυτή (Εικόνα 5.39).

```

@ Unity Message | 0 references
void Update()
{
    if ((Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began))
    {
        Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);
        RaycastHit hit;
        if (Physics.Raycast(ray, out hit))
        {
            if (hit.transform != null)
            {
                string objectName = hit.transform.name;
                switch (objectName)
                {
                    case "hifi":
                        PlayMusic(0, _particleSystems[0]);
                        break;
                    case "piano":
                        PlayMusic(1, _particleSystems[1]);
                        break;
                    case "stars":
                        PlayMusic(0, _particleSystems[0]);
                        break;
                    default:
                        break;
                }
            }
        }
    }

    if (!_audioSource.isPlaying)
    {
        foreach (ParticleSystem particleSystem in _particleSystems) { particleSystem.Stop(); }
    }
}

```

Εικόνα 5.39: Update Unity μέθοδος

Δημιουργείται ένα Raycast με touch του χρήστη στην οθόνη. Όταν αυτό συγκρουστεί (κάνει collide) με κάποιο Collider component, γίνεται trigger η if statement μέσα στην μέθοδο Update του Unity. Η private μεταβλητή objectName καθορίζει τις παραμέτρους που θα χρησιμοποιηθούν στην μέθοδο PlayMusic μέσα στο switch case.

Στην περίπτωση του gameObject stars προφανώς εκτελείται το case “stars”. Η PlayMusic (Εικόνα 5.40) λειτουργεί ως εξής:

Αρχικά, ελέγχει αν το Audio Source (η μεταβλητή _audioSource) παίζει κάποιον ήχο ή όχι. Στην περίπτωση που κάτι τέτοιο ισχύει, σταματάει την όποια μελωδία και το particle system που δίνεται ως

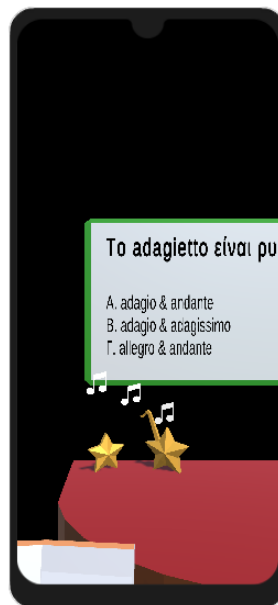
Κεφάλαιο 5

παράμετρος. Σε αντίθετη περίπτωση, το audio clip που θα παίξει το Audio Source component είναι αυτό που βρίσκεται στην θέση audioClipIndex του πίνακα _audioClips.

```
void PlayMusic(int audioClipIndex, ParticleSystem particleSystem)
{
    if (_audioSource.isPlaying)
    {
        _audioSource.Stop();
        particleSystem.Stop();
    }
    else
    {
        _audioSource.clip = _audioClips[audioClipIndex];
        _audioSource.Play();
        particleSystem.Play();
    }
}
```

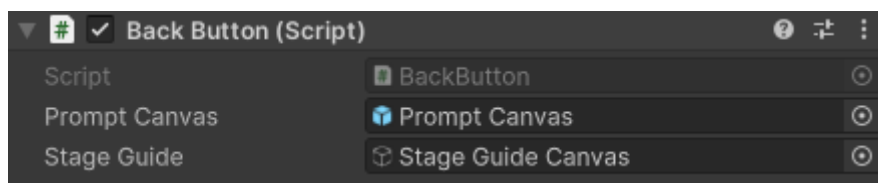
Εικόνα 5.40: PlayMusic μέθοδος

Τέλος, παίζει και το particle system (Εικόνα 5.41).



Εικόνα 5.41: ενεργά Particles

Στο αντικείμενο Event System έχει προστεθεί η κλάση BackButton η οποία είναι υπεύθυνη για την εμφάνιση του Prompt Canvas αφού γίνει κλικ στο back button της android συσκευής (Εικόνα 5.42).



Εικόνα 5.42: BackButton

Αυτό το tool είναι αναγκαίο ώστε να επιστρέψει ο χρήστης ανά πάσα στιγμή στην αρχική σκηνή, το Main Menu (Εικόνα 5.43).

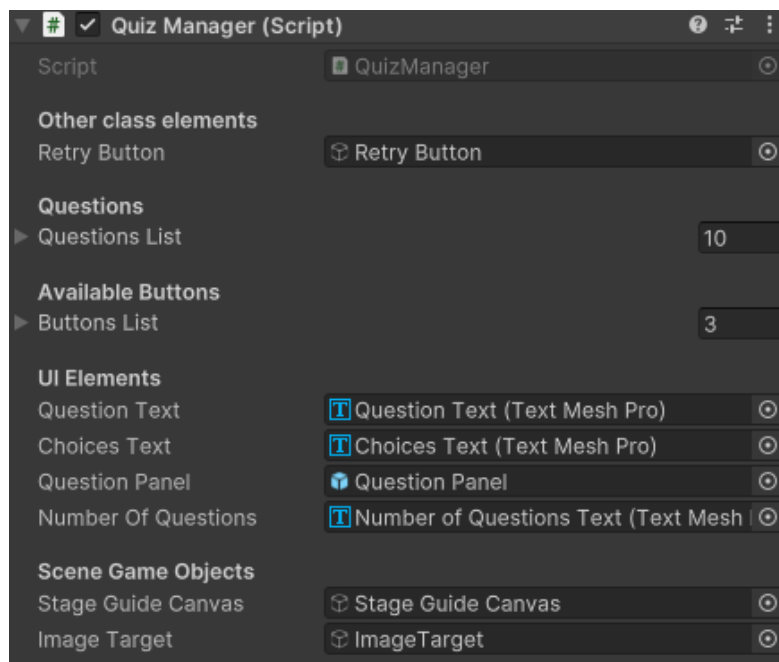
```

void Update()
{
    // Make sure user is on Android platform
    if (Application.platform == RuntimePlatform.Android)
    {
        // Check if Back was pressed this frame
        if (Input.GetKeyDown(KeyCode.Escape))
        {
            // Quit the application
            if (!_stageGuide.activeSelf)
                _promptCanvas.SetActive(true);
        }
    }
}

```

Εικόνα 5.43: Update Unity μέθοδος

Το κύριο gameObject της σκηνής και υπεύθυνο για την διαχείριση του quiz είναι το Quiz Manager. Γίνεται χρήση της ομώνυμης κλάσης (Εικόνα 5.44).



Εικόνα 5.44: Quiz Manager

Σε πρώτη φάση, εκτελείται η μέθοδος Start του Unity για να γίνει το initialization του quiz. Το text component του Number of Questions Text παίρνει ως τιμή το συνολικό μέγεθος της λίστας με τις ερωτήσεις, (Εικόνα 5.45).

```

@ Unity Message | 0 references
void Start() => ShowQuiz();

1 reference
public void ShowQuiz()
{
    _totalQuestions = _questionsList.Count;
    GenerateQuestion(); // Retrieve random question
}

```

Εικόνα 5.45: ShowQuiz μέθοδος

Έπειτα η μέθοδος GenerateQuestion (Εικόνα 5.46), όπως προδίδει και το όνομά της, επιλέγει τυχαία ένα index από την λίστα με τις ερωτήσεις και στη συνέχεια κάνει parse στο text component του Question Text (βλέπε 3D model πίνακας) την ερώτηση (που ανήκει σε αυτό το index).

```
void GenerateQuestion()
{
    if (_questionsList.Count > 0)
    {
        _numberOfQuestions.text = _answeredQuestions + " / " + _totalQuestions; //UI element
        _currentQuestionIndex = Random.Range(0, _questionsList.Count); // get a random number from zero to number of our questions
        _questionText.text = _questionsList[_currentQuestionIndex].question;
        ResetButtons(true);
        SetAnswers(); // get next answer
    }
    else
    {
        TerminateQuiz();
    }
}
```

Εικόνα 5.46: GenerateQuestion μέθοδος

Στη συνέχεια γίνεται το initialization των διαθέσιμων απαντήσεων και η αντιστοίχισή τους στα τρία buttons που υπάρχουν στη σκηνή (Εικόνα 5.47).

```
void SetAnswers()
{
    for (int i = 0; i < _buttonsList.Count; i++)
    {
        _buttonsList[i].GetComponent<ButtonManager>()._isCorrect = false; // to avoid having all buttons to be correct
        _choicesText.text += _questionsList[_currentQuestionIndex].answers[i] + "\n";

        if (_questionsList[_currentQuestionIndex].correctAnswer == i + 1)
        {
            _buttonsList[i].GetComponent<ButtonManager>()._isCorrect = true;
        }
    }
}
```

Εικόνα 5.47: setAnswer μέθοδος

Το ποια απάντηση είναι η σωστή καθορίζεται μέσω UI, δηλαδή μέσα από το Unity Editor και αυτό διότι η λίστα με τις ερωτήσεις (μεταβλητή _questionsList) είναι ενός custom τύπου. Στο παραπάνω κομμάτι κώδικα φαίνεται και ο τρόπος που αλλάζει τιμή η public μεταβλητή _isCorrect της κλάσης ButtonManager (Εικόνα 5.48).

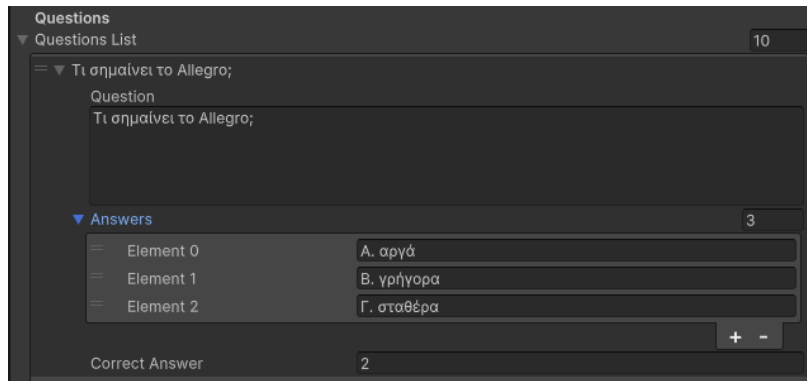
```
[Header ("Questions")]
[SerializeField] List<QuizFormat> _questionsList;
```

```
[System.Serializable]
4 references
public class QuizFormat
{
    [TextArea(5, 10)]
    public string question;
    public string[] answers;
    public int correctAnswer;
}
```

Εικόνα 5.48: Question Format

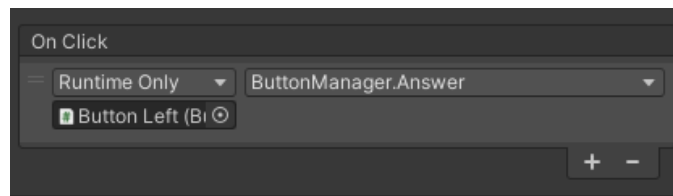
Γίνεται πιο εύκολο κατανοητό ρίχνοντας μια ματιά στην παρακάτω εικόνα. Η λίστα QuestionsList διαθέτει δέκα ερωτήσεις, η πρώτη (Element 0) ερώτηση (public string question) ισούται με “Τι σημαίνει το Allegro;”. Ο πίνακας Answers (public string[] answers) περιέχει τις πιθανές απαντήσεις.

Και τέλος, η μεταβλητή Correct Answer (public int correctAnswer) δηλώνει τη σωστή απάντηση (Εικόνα 5.49).



Εικόνα 5.49: Question List Unity Editor

Έτσι, όταν γίνεται κλικ σε κάποιο από τα τρία κουμπιά εκτελείται η public μέθοδος Answer της κλάσης ButtonManager με στόχο να αλλάξει το χρώμα του επιλεγμένου button. Επιπλέον, εκτελείται η public μέθοδος Answer της κλάσης QuizManager (Εικόνα 5.50). Δίνεται ως παράμετρος στη μέθοδο η bool public μεταβλητή _isCorrect, (Εικόνα 5.51).



Εικόνα 5.50: onClick Answer μέθοδος

```
public void Answer()
{
    GetComponent<Image>().color = _isCorrect ? Color.green : Color.red;
    Debug.Log(_isCorrect ? "Correct answer!" : "Wrong answer!");
    _quizManager.Answer(_isCorrect);
}
```

Εικόνα 5.51: μεταβλητή _isCorrect

Αν η παράμετρος που δόθηκε είναι αληθής τότε αυξάνεται κατά μια μονάδα ο συνολικός αριθμός των σωστών απαντήσεων. Το ίδιο πραγματοποιείται και για τις ερωτήσεις που έχουν απαντηθεί, (Εικόνα 5.52).

```
public void Answer(bool PPlayerAnswer)
{
    if (PPlayerAnswer == true)
        _correctAnswers++;

    _answeredQuestions++; // for every correct answer increase score by 1
    _questionsList.RemoveAt(_currentQuestionIndex); // remove answered question
    ResetButtons(false);
    StartCoroutine(WaitForNext()); // Get next question
}
```

Εικόνα 5.52: Answer μέθοδος

Αφαιρείται στη συνέχεια η ερώτηση από την λίστα. Απενεργοποιούνται για λίγα δεύτερα τα buttons (μέθοδος ResetButtons) και γίνεται retrieve η επόμενη ερώτηση με τον ίδιο τρόπο που αναλύθηκε προηγουμένως .

Αν το μέγεθος της λίστας δεν είναι πλέον μεγαλύτερο του μηδενός, αυτό σημαίνει ότι έχουν απαντηθεί όλες (παραπάνω αναφέρθηκε ότι με κάθε απάντηση διαγράφεται η εκάστοτε ερώτηση από την λίστα) και εκτελείται το “else” κομμάτι του κώδικά (Εικόνα 5.53).

```

void GenerateQuestion()
{
    if (_questionsList.Count > 0)
    {
        _numberOfQuestions.text = _answeredQuestions + " / " + _totalQuestions; //UI element
        _currentQuestionIndex = Random.Range(0, _questionsList.Count); // get a random number from zero to number of our questions
        _questionText.text = _questionsList[_currentQuestionIndex].question;
        ResetButtons(true);
        SetAnswers(); // get next answer
    }
    else
    {
        TerminateQuiz();
    }
}

```

Εικόνα 5.53: GeneralQuestion μέθοδος

Απενεργοποιείται το ImageTarget από την σκηνή και εμφανίζεται ξανά με animation το Stage Guide Canvas. Αυτή τη φορά όμως με διαφορετικό κείμενο στο οποίο αναφέρεται η επίδοση του χρήστη/μαθητή στο quiz (Εικόνα 5.54).

```

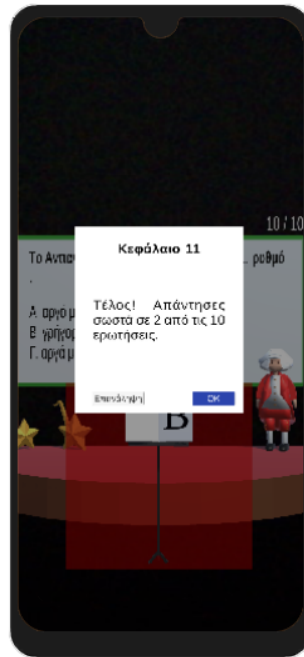
void TerminateQuiz()
{
    StopAllCoroutines();
    _answeredQuestions = 0;
    _imageTarget.SetActive(false);
    InitiateDialogueCanvas();
}

1 reference
void InitiateDialogueCanvas()
{
    _stageGuideCanvas.SetActive(true);
    _retryButton.SetActive(true);
    StageGuideDialogue stageDialogue = GameObject.Find("Dialogue Group").GetComponent<StageGuideDialogue>();
    stageDialogue._isQuizFinished = true;
    stageDialogue._stageGuide.text = "Τέλος! Απάντησες σωστά σε " + _correctAnswers + " από τις " + _totalQuestions + " ερωτήσεις.";
}

```

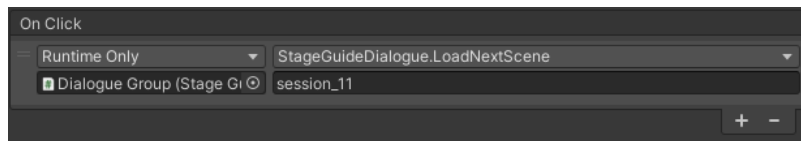
Εικόνα 5.54: TerminalQuiz και InitiateDialogCanvas μέθοδος

Τέλος, στο παράθυρο εκτός από το αποτέλεσμα, υπάρχουν και δυο κουμπιά, “Επανάληψη” και ‘OK’ (Εικόνα 5.55).

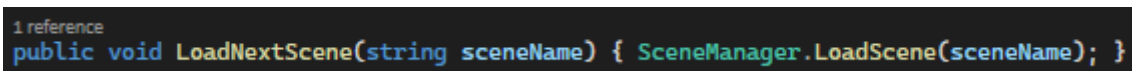


Εικόνα 5.55: Ενημέρωση του χρήστη για τις σωστές απαντήσεις

Πατώντας το κουμπί “Επανάληψη” γίνεται ενεργό του gameObject RetryButton του Stage Guide Canvas δίνοντας κατά αυτόν τον τρόπο τη δυνατότητα στον χρήστη να επαναλάβει το quiz (Εικόνα 5.56-57) ενώ πατώντας το κουμπί “OK” βγαίνει από την παρούσα σκηνή και μπορεί να συνεχίσει με τις επόμενες ενότητες.



Εικόνα 5.56: RetryButton Button component

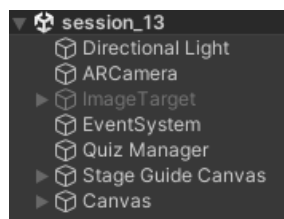


Εικόνα 5.57: LoadNexrScene μέθοδος

5.3.2 Κεφάλαιο 13

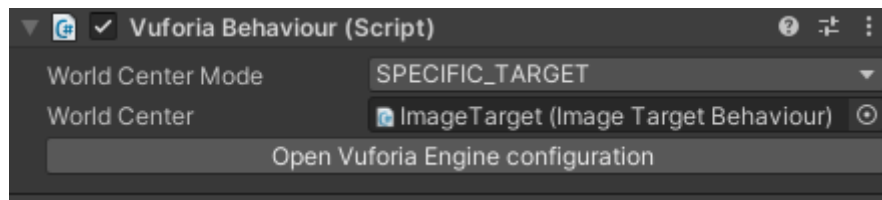
Η σκηνή του ‘Κεφαλαίου 11’ αντιστοιχεί στο Κεφάλαιο 13 του βιβλίου μαθητή σελ. 30-46, όπου και εδώ υπάρχουν ερωτοαπαντήσεις τις οποίες πρέπει να απαντήσει ο χρήστης. Σε αντίθεση με το προηγούμενο κεφάλαιο, σε αυτό υπάρχουν διάφορα 3D αντικείμενα σαν απαντήσεις.

Η δομή της σκηνής Session 13 (Κεφαλαίου 13) έχει ως εξής (Εικόνα 5.58):



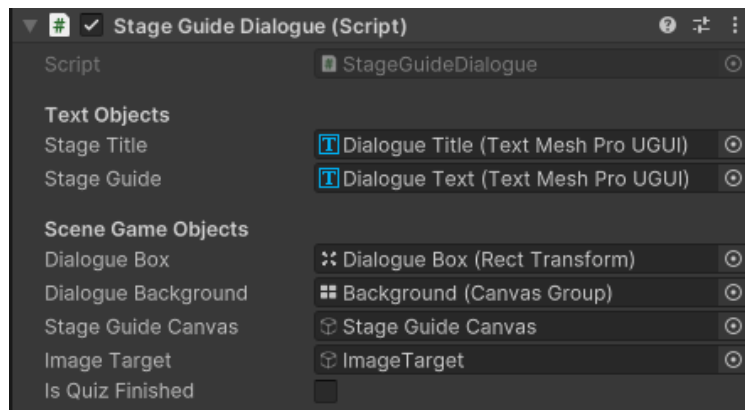
Εικόνα 5.58: Session 13

Το gameObject ARCamera (του Vuforia) έχει το παρακάτω setup (Εικόνα 5.59):



Εικόνα 5.59: AR Camera, Vuforia Behaviour

Αφού φορτώθει η σκηνή session_13 από την μηχανή γραφικών Unity, εμφανίζεται το Stage Guide Canvas στο οποίο περιέχονται οι οδηγίες της εκάστοτε σκηνής. Το child gameObject του Stage Guide Canvas, Dialogue Group χρησιμοποιεί την κλάση StageGuideDialogue (Εικόνα 5.60).



Εικόνα 5.60: Stage Guide Dialogue script setup

Αρχικά, μιας και το gameObject είναι ενεργό εκτελείται η μέθοδος OnEnable της Unity. Αυτό έχει ως αποτέλεσμα την εμφάνιση του UI αντικειμένου στην οθόνη της συσκευής μέσω ενός animation. Η υλοποίηση ενός “Stage Guide” πραγματοποιήθηκε για να διευκολύνει τον χρήστη/μαθητή στην κατανόηση του στόχου της κάθε σκηνής. Τέτοιες πρακτικές ακολουθούνται συνεχώς από εταιρείες λογισμικού στην βιομηχανία του gaming (Εικόνα 5.61).

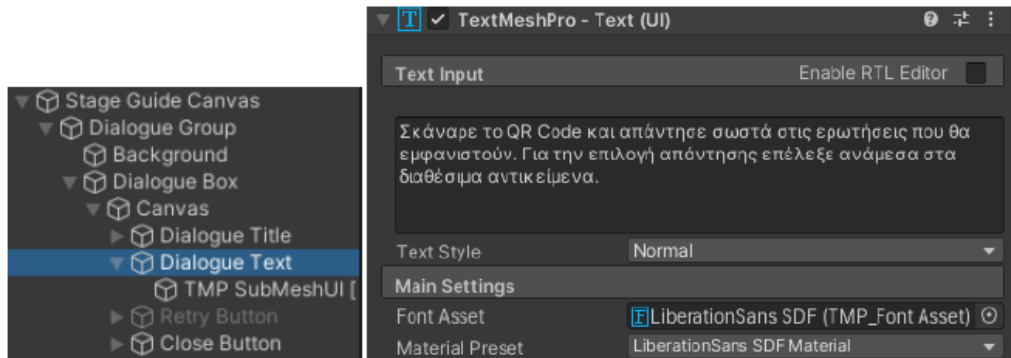
```

@ Unity Message | 0 references
void OnEnable()
{
    _dialogueBackground.alpha = 0;
    _dialogueBackground.LeanAlpha(1, 0.5f);

    _dialogueBox.localPosition = new Vector2(0, -Screen.height);
    _dialogueBox.LeanMoveLocalY(0, 0.5f).setEaseOutExpo().delay = 0.1f;
}
    
```

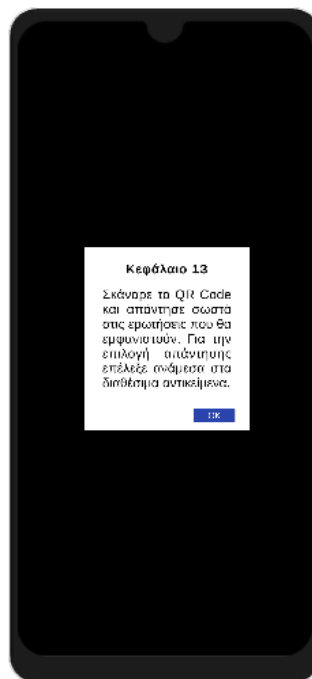
Εικόνα 5.61: OnEnable μέθοδος

Τόσο ο τίτλος της σκηνής, όσο και το κείμενο με τις οδηγίες που υπάρχει δίνονται ως string texts μέσω του Unity Editor (Εικόνα 5.62).



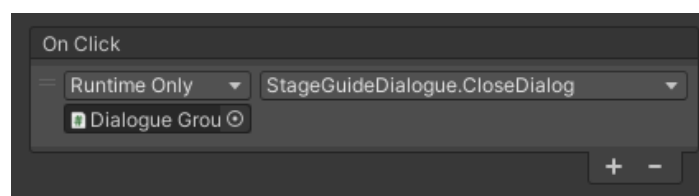
Εικόνα 5.62: Unity Editor

Ο χρήστης θα αντικρίσει το Popup με τις οδηγίες όπως φαίνεται στην (Εικόνα 5.63).



Εικόνα 5.63: Stage Guide Popup

Το κουμπί OK εκτελεί την public μέθοδο CloseDialog της κλάσης StageGuideDialogue. Αναλύοντας τον κώδικα που φαίνεται στην εικόνα, γίνεται αντιληπτό ότι η μέθοδος αυτή είναι υπεύθυνη για την απομάκρυνση του Stage Guide Canvas από την οθόνη της σκηνής και το initialization των απαραίτητων αντικειμένων για το σκανάρισμα του Image Target (Εικόνα 5.64).



Εικόνα 5.64: OK button component

Στη συνέχεια, αφού προηγηθεί ο έλεγχος του αν τελείωσε ή όχι το Quiz, απενεργοποιείται το gameObject Stage Guide Canvas από την σκηνή με την βοήθεια της μεθόδου ToggleVisibility (Εικόνα 5.65)

```

public void CloseDialog()
{
    _dialogueBackground.LeanAlpha(0, 0.5f);
    _dialogueBox.LeanMoveLocalY(-Screen.height, 0.5f).setEaseInExpo();
    StartCoroutine(ToggleVisibility(_imageTarget, true, 0));

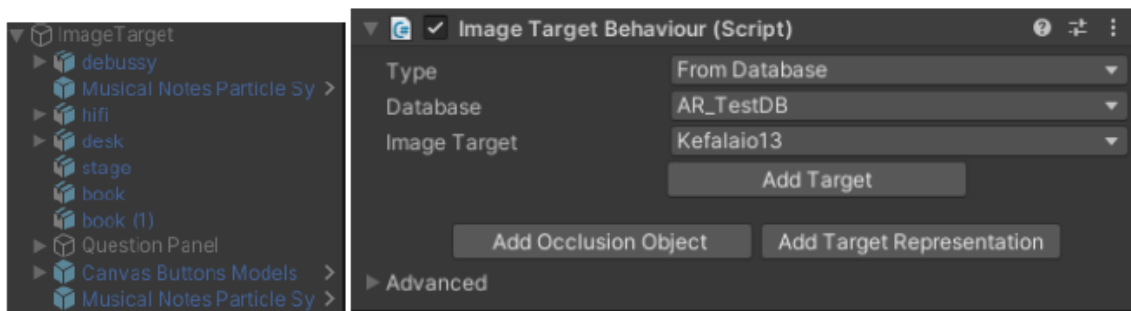
    if (_isQuizFinished == true)
    {
        LoadNextScene("main_menu");
    }
    else
    {
        StartCoroutine(ToggleVisibility(_stageGuideCanvas, false, 10));
    }
}

2 references
IEnumerator ToggleVisibility(GameObject gameObject, bool status, int seconds)
{
    gameObject.SetActive(status);
    yield return new WaitForSeconds(seconds);
}

```

Εικόνα 5.65: CloseDialog μέθοδος

Το αντικείμενο ImageTarget αποτελεί αντικείμενο του Vuforia (Εικόνα 5.66) και χρησιμοποιείται για την ανίχνευση και παρακολούθηση της εικόνας προς σκανάρισμα από την κάμερα της android συσκευής (Εικόνα 5.67).



Εικόνα 5.66: Image Target



Εικόνα 5.67: Image Target του Κεφαλαίου 13

Αφού σκαναριστεί το Image Target της Εικόνας 5.67, εμφανίζονται στο χώρο τα δεκαεπτά 3D αντικείμενα που δημιουργήθηκαν με το Blender (Εικόνα 5.68):

- ένας συνθέτης βοηθός
- τρία αναλόγια με ένα βιβλίο πάνω στο καθένα
- ένα πικάπ
- μία σκηνή
- ένα πίνακα
- ένα τραπέζι
- δύο βιβλία πάνω στη σκηνή
- τέσσερα διαφορετικά μουσικά όργανα
- τρεις εικόνες



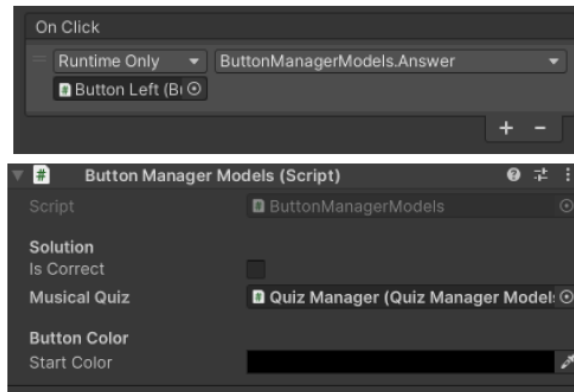
Εικόνα 5.68: Προβολή σκηνής κεφαλαίου 13

Έπειτα, όσον αφορά το στήσιμο της σκηνής και τον τρόπο των απαντήσεων, ο 3D πίνακας εμφανίζει πάνω του τις ερωτήσεις και τρεις επιλογές απαντήσεων. Η απάντηση δίνεται από τον χρήστη πατώντας πάνω στα 3D αντικείμενα.

Το Canvas Buttons Models περιέχει 3 buttons τα οποία όπως ήταν αναμενόμενο ακολουθούν την ίδια δομή. Αυτό ουσιαστικά σημαίνει ότι το button component του κάθε button (Button Left, Button Middle και Button Right) (Εικόνα 5.69) εκτελεί την public μέθοδο της κλάσης ButtonManagerModels (Εικόνα 5.70) μέσω του Unity Event OnClick.

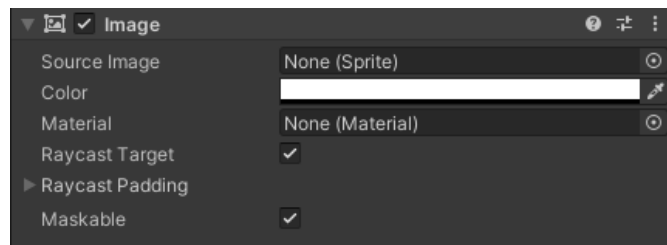


Εικόνα 5.69: Canvas Button



Εικόνα 5.70: ButtonManagerModels και button component

Η μέθοδος αυτή απλά παίρνει το Image component (Εικόνα 5.71) του εκάστοτε button και αλλάζει το χρώμα του ανάλογα με την τιμή της μεταβλητής `_isCorrect` (Εικόνα 5.72).



Εικόνα 5.71: Image component

```
public void Answer()
{
    GetComponent<Image>().color = _isCorrect ? Color.green : Color.red;
    Debug.Log(_isCorrect ? "Correct answer!" : "Wrong answer!");
    _musicalQuiz.Answer(_isCorrect);
}
```

Εικόνα 5.72: Answer μέθοδος

Αυτή την φορά, τον ρόλο του μουσικοσυνθέτη βοηθού έχει ο C. Debussy. Όπως και προηγουμένως, η χρήση του δεν είναι άλλη παρά διακοσμητική.

Το αντικείμενο `hifi` λειτουργεί με τον ίδιο τρόπο όπως και το αντικείμενο `stars` του Session 11. Αποτελείται από τα ίδια components, δηλαδή ένα `Box Collider` για την αλληλεπίδραση με το χρήστη, την κλάση `SoundManager` και ένα `Audio Source` component .

Όσον αφορά την αλληλεπίδραση, δημιουργείται ένα `Raycast` με `touch` του χρήστη στην οθόνη. Όταν αυτό συγκρουστεί (κάνει `collide`) με κάποιο `Collider` component, γίνεται `trigger` η `if` statement μέσα στην μέθοδο `Update` του Unity (ίδια μέθοδος `Update` που χρησιμοποιείται και στο αντικείμενο `stars` μπορείτε να δείτε την εικόνα 5.39) . Η `private` μεταβλητή `objectName` καθορίζει τις παραμέτρους που θα χρησιμοποιηθούν στην μέθοδο `PlayMusic` μέσα στο `switch case`.

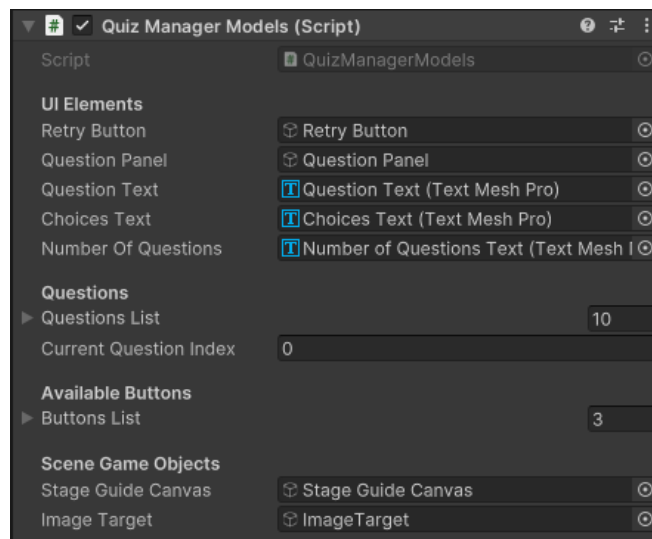
Στην περίπτωση του `gameObject hifi` προφανώς εκτελείται το `case "hifi"`. Η `PlayMusic` λειτουργεί ως εξής (ίδια μέθοδος `PlayMusic` που χρησιμοποιείται και στο αντικείμενο `stars` μπορείτε να δείτε την εικόνα 5.40):

Αρχικά, ελέγχει αν το `Audio Source` (η μεταβλητή `_audioSource`) παίζει κάποιον ήχο ή όχι. Στην περίπτωση που κάτι τέτοιο ισχύει, σταματάει την όποια μελωδία και το `particle system` που δίνεται ως

παράμετρος. Σε αντίθετη περίπτωση, το audio clip που θα παίζει το Audio Source component είναι αυτό που βρίσκεται στην θέση audioClipIndex του πίνακα _audioClips. Τέλος, παίζει και το particle system.

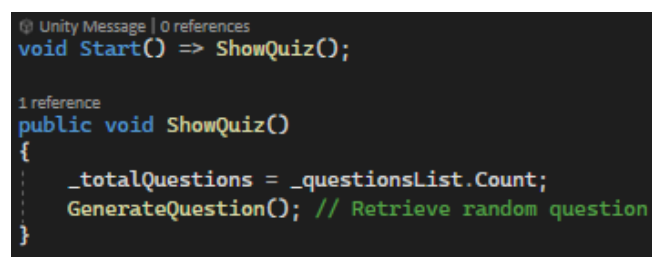
Στο αντικείμενο Event System έχει προστεθεί η κλάση BackButton η οποία είναι υπεύθυνη για την εμφάνιση του Prompt Canvas (βλέπε ενότητα Main Menu) αφού γίνει κλικ στο back button της android συσκευής. Αυτό το tool είναι αναγκαίο ώστε να επιστρέψει ο χρήστης ανά πάσα στιγμή στην αρχική σκηνή, το Main Menu.

Το κύριο gameObject της σκηνής και υπεύθυνο για την διαχείριση του quiz με τα μοντέλα είναι το QuizManagerModels (Εικόνα 5.73). Γίνεται χρήση της ομώνυμης κλάσης.



Εικόνα 5.73: Quiz Manager Models

Σε πρώτη φάση, εκτελείται η μέθοδος Start του Unity για να γίνει το initialization του quiz. Το text component του Number of Questions Text παίρνει ως τιμή το συνολικό μέγεθος της λίστας με τις ερωτήσεις, (Εικόνα 5.74).



Εικόνα 5.74: ShowQuiz μέθοδος

Έπειτα η μέθοδος GenerateQuestion (Εικόνα 5.75), όπως προδίδει και το όνομά της, αφού ελέγξει το μέγεθος της λίστας με τις ερωτήσεις επιλέγει τυχαία ένα index από αυτήν και στη συνέχεια κάνει parse στο text component του Question Text την ερώτηση.

```

void GenerateQuestion()
{
    if (_questionsList.Count > 0)
    {
        _numberOfQuestions.text = _answeredQuestions + " / " + _totalQuestions;
        _currentQuestionIndex = Random.Range(0, _questionsList.Count); // get a random number from zero to number of our questions
        _questionText.text = _questionsList[_currentQuestionIndex].question;
        _choicesText.text = string.Empty;
        ResetButtons(true);
        SetAnswers(); // get next answer
        ActivateObject();
    }
    else
    {
        TerminateQuiz();
    }
}

```

Εικόνα 5.75: GenerateQuestion μέθοδος

Η μεταβλητή `_numberOfQuestions` τύπου `TMP_Text` απεικονίζει σε UI επίπεδο το σημείο στο οποίο βρίσκεται ο χρήστης (απαντηθέντες(?) ερωτήσεις/σύνολο ερωτήσεων) και υπάρχει με απώτερο σκοπό την ενημέρωση του για το progression που έχει πραγματοποιήσει στο quiz.

Στη συνέχεια γίνεται το initialization των διαθέσιμων απαντήσεων και αντιστοίχισής τους στα τρία buttons που υπάρχουν στη σκηνή (Εικόνα 5.76).

```

void SetAnswers()
{
    for (int i = 0; i < _buttonsList.Count; i++)
    {
        _buttonsList[i].GetComponent<Image>().color = _buttonsList[i].GetComponent<ButtonManagerModels>()._startColor;
        _buttonsList[i].GetComponent<ButtonManagerModels>()._isCorrect = false; // to avoid having all buttons to be correct
        _choicesText.text += _questionsList[_currentQuestionIndex].answers[i] + "\n";

        if (_questionsList[_currentQuestionIndex].correctAnswer == i + 1)
        {
            _buttonsList[i].GetComponent<ButtonManagerModels>()._isCorrect = true;
        }
    }
}

```

Εικόνα 5.76: SetAnswer μέθοδος

Το ποια απάντηση είναι η σωστή καθορίζεται μέσω UI, δηλαδή μέσα από το Unity Editor και αυτό διότι η λίστα με τις ερωτήσεις (μεταβλητή `_questionsList`) είναι ενός custom τύπου. Στο παραπάνω κομμάτι κώδικα φαίνεται και ο τρόπος που αλλάζει τιμή η public μεταβλητή `_isCorrect` της κλάσης `ButtonManagerModels`, (Εικόνα 5.77).

```

[Header("Question")]
public QuizManagerModels _musicalQuiz;

```

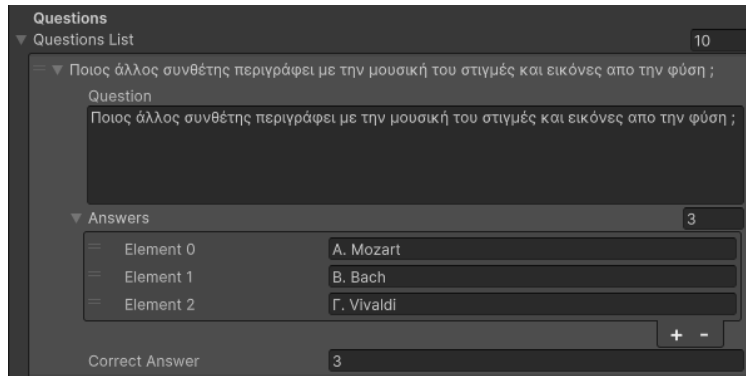
```

[System.Serializable]
4 references
public class QuizFormat
{
    [TextArea(5, 10)]
    public string question;
    public string[] answers;
    public int correctAnswer;
}

```

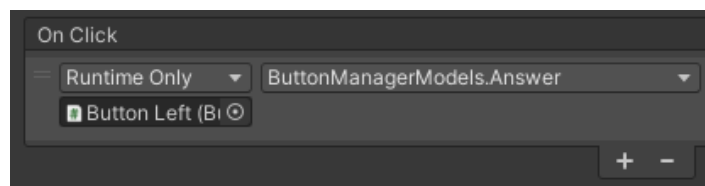
Εικόνα 5.77: Question Format

Γίνεται πιο εύκολο κατανοητό ρίχνοντας μια ματιά στην παρακάτω εικόνα. Η λίστα `QuestionsList` διαθέτει δέκα ερωτήσεις, η πρώτη (Element 0) ερώτηση (public string question) ισούται με “Ποιος άλλος συνθέτης περιγράφει με την μουσική του στιγμές και εικόνες από την φύση;”. Ο πίνακας `Answers` (public string[] answers) περιέχει τις πιθανές απαντήσεις. Τέλος, η μεταβλητή `Correct Answer` (public int correctAnswer) δηλώνει τη σωστή απάντηση (Εικόνα 5.78).

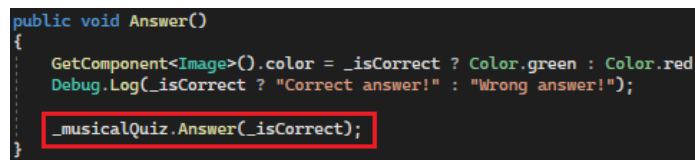


Εικόνα 5.78: Question List Unity Editor

Έτσι, όταν γίνεται κλικ σε κάποιο από τα τρία κουμπιά εκτελείται η public μέθοδος Answer της κλάσης ButtonManagerModels με στόχο να αλλάξει το χρώμα του επιλεγμένου button. Το background color του κουμπιού που περιέχει την σωστή απάντηση γίνεται πράσινο ενώ σε άλλη περίπτωση μετατρέπεται σε κόκκινο με παρόμοιο τρόπο όπως στο κεφάλαιο 11. Επιπλέον, εκτελείται η public μέθοδος Answer της κλάσης QuizManagerModels (Εικόνα 5.79). Δίνεται ως παράμετρος στη μέθοδο η bool public μεταβλητή _isCorrect (Εικόνα 5.80).

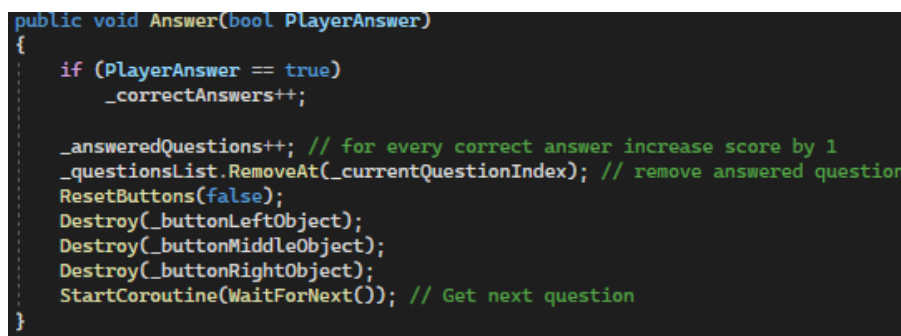


Εικόνα 5.79: Question List Unity Editor



Εικόνα 5.80: μεταβλητή _isCorrect

Αν λοιπόν η παράμετρος που δόθηκε είναι αληθής τότε αυξάνεται κατά μια μονάδα ο συνολικός αριθμός των σωστών απαντήσεων. Το ίδιο πραγματοποιείται και για τις ερωτήσεις που έχουν απαντηθεί (gameObject.Number of Questions Text), (Εικόνα 5.81).

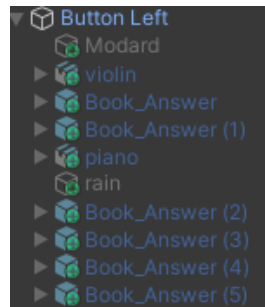


Εικόνα 5.81: Answer μέθοδος

Αφαιρείτε στη συνέχεια η ερώτηση από την λίστα. Απενεργοποιούνται για λίγα δεύτερα τα buttons (μέθοδος ResetButtons) και γίνεται retrieve η επόμενη ερώτηση με τον ίδιο τρόπο που αναλύθηκε προηγουμένως.

Η διαφορά με την κλάση QuizManager που παρουσιάστηκε στην προηγούμενη ενότητα (Κεφάλαιο 11) είναι το γεγονός ότι εδώ χρησιμοποιούνται 3D models και sprites ως child gameObjects για το κάθε button.

Για ευκολότερη κατανόηση δίνεται ως παράδειγμα, το Button Left[Εικόνα 5.82]:



Εικόνα 5.82: Button Left

Αν κάτσει κανείς και απαριθμήσει τα child gameObjects του Button Left διαπιστώνει πως το πλήθος τους ισούται με τον συνολικό αριθμό των ερωτήσεων. Και αυτό διότι, για κάθε ερώτηση αντιστοιχεί ένα σετ 3D μοντέλων (1 σετ = 3 buttons = 3 διαφορετικά 3D models).

Η αρχικοποίηση τους έχει γίνει μέσω της μεθόδου ActiveObject (Εικόνα 5.83). Η μεταβλητή `_currentQuestionIndex` που αναφέρθηκε προηγουμένως (βλέπε μέθοδο `GenerateQuestions`) χρησιμοποιείται και εδώ. Με αυτό τον τρόπο, ενεργοποιούνται το σωστό σετ μοντέλων για την σωστή ερώτηση (πχ Ερώτηση 9 - Σετ μοντέλων στη θέση 9 του εκάστοτε button). Υπενθυμίζεται ότι η μέθοδος αυτή βρίσκεται μέσα στην `GenerateQuestions` και καλείται κάθε φορά που το `gameObject QuizManager` κάνει `retrieve` μια ερώτηση.

```
void ActivateObject()
{
    _buttonLeftObject = _buttonsList[0].transform.GetChild(_currentQuestionIndex).gameObject;
    _buttonMiddleObject = _buttonsList[1].transform.GetChild(_currentQuestionIndex).gameObject;
    _buttonRightObject = _buttonsList[2].transform.GetChild(_currentQuestionIndex).gameObject;

    _buttonLeftObject.SetActive(true);
    _buttonMiddleObject.SetActive(true);
    _buttonRightObject.SetActive(true);
}
```

Εικόνα 5.83: ActiveObject μέθοδος

Συνεχίζοντας την επεξήγηση της μεθόδου Answer (Εικόνα 5.84), αφού αφαιρεθεί η ενεργή ερώτηση καταστρέφονται τα child gameObjects των buttons που αντιστοιχούν σε αυτή.

```
public void Answer(bool PlayerAnswer)
{
    if (PlayerAnswer == true)
        _correctAnswers++;

    _answeredQuestions++; // for every correct answer increase score by 1
    _questionsList.RemoveAt(_currentQuestionIndex); // remove answered question
    ResetButtons(false);
    Destroy(_buttonLeftObject);
    Destroy(_buttonMiddleObject);
    Destroy(_buttonRightObject);
    StartCoroutine(WaitForNext()); // Get next question
}
```

Εικόνα 5.84: Answer μέθοδος

Απενεργοποιούνται για λίγα δεύτερα τα buttons (μέθοδος ResetButtons) και γίνεται retrieve η επόμενη ερώτηση με τον ίδιο τρόπο που αναλύθηκε προηγουμένως, (Εικόνα 5.85).

```
IEnumerator WaitForNext()
{
    yield return new WaitForSecondsRealtime(1); // wait 1 second before executing generateQuestion() method
    GenerateQuestion();
}
```

Εικόνα 5.85: WaitForNext μέθοδος

Αν το μέγεθος της λίστας δεν είναι πλέον μεγαλύτερο του μηδενός, αυτό σημαίνει ότι έχουν απαντηθεί όλες (παραπάνω αναφέρθηκε ότι με κάθε απάντηση διαγράφεται η εκάστοτε ερώτηση από την λίστα) και εκτελείται το “else” κομμάτι του κώδικα, (Εικόνα 5.86).

```
void GenerateQuestion()
{
    if C_questionsList.Count > 0)
    {
        _numberOfQuestions.text = _answeredQuestions + " / " + _totalQuestions;
        _currentQuestionIndex = Random.Range(0, _questionsList.Count); // get a random number from zero to number of our questions
        _questionText.text = _questionsList[_currentQuestionIndex].question;
        _choicesText.text = string.Empty;
        ResetButtons(true);
        SetAnswers(); // get next answer
        ActivateObject();
    }
    else
    {
        TerminateQuiz();
    }
}
```

Εικόνα 5.86: GeneralQuestion μέθοδος

Απενεργοποιείται το ImageTarget από την σκηνή και εμφανίζεται ξανά με animation το Stage Guide Canvas. Αυτή τη φορά όμως με διαφορετικό κείμενο στο οποίο αναφέρεται η επίδοση του χρήστη/μαθητή στο quiz (χρησιμοποιήθηκε η ίδια μέθοδος με το κεφάλαιο 11).

Τέλος, γίνεται ενεργό του gameObject RetryButton του Stage Guide Canvas δίνοντας κατά αυτόν τον τρόπο τη δυνατότητα στον χρήστη να επαναλάβει το quiz (χρησιμοποιήθηκε η ίδια μέθοδος με το κεφάλαιο 11 [Εικόνα 5.58-9]).

5.3.3 Κεφάλαιο 18

Στο παρόν κεφάλαιο δε θα γίνει αναφορά στη δομή της σκηνής και των κομματιών του κώδικα καθώς είναι ο ίδιος με αυτόν του Κεφαλαίου 13. Θα γίνει αναφορά για την οπτική γωνία του χρήστη.

Η σκηνή του ‘Κεφαλαίου 18’ σχετίζεται με το Κεφάλαιο 18 του βιβλίου μαθητή σελ. 58-61.

Όταν ο χρήστης κλικάρει πάνω στο κουμπί “Κεφάλαιο 22” θα αντικρίσει το Popur με τις οδηγίες. Αφού πατήσει ο χρήστης το κουμπί “OK” είναι απαραίτητο το σκανάρισμα του Image Target, Εικόνα 5.87.



Εικόνα 5.87: Image Target του Κεφαλαίου 18

Μόλις σκανάρει το QR, θα εμφανιστεί η σκηνή και σε αυτή ο χρήστης θα δει είκοσι ένα 3D αντικείμενα που δημιουργήθηκαν με το Blender (Εικόνα 5.88):

- ένα συνθέτη βοηθό
- τρία αναλόγια με ένα βιβλίο πάνω στο καθένα
- ένα πικάπ
- μία σκηνή
- ένα πίνακα
- δύο τραπέζια
- δύο βιβλία πάνω στη σκηνή
- τέσσερα διαφορετικά μουσικά όργανα πάνω στο τραπέζι
- έξι διαφορετικά μουσικά όργανα στις απαντήσεις



Εικόνα 5.88: Προβολή σκηνής κεφαλαίου 18

Έπειτα, όσον αφορά το στήσιμο της σκηνής και τον τρόπο των απαντήσεων, ο 3D πίνακας εμφανίζει πάνω του τις ερωτήσεις και τρεις επιλογές απαντήσεων. Η απάντηση δίνεται από τον χρήστη πατώντας πάνω στα 3D αντικείμενα παρομοίως με το Κεφάλαιο 13.

Στη σκηνή υπάρχει επίσης ο μουσικοσυνθέτης βοηθός, όπου στην παρούσα σκηνή έχει δημιουργηθεί ο Bach. Η χρήση του είναι απλά διακοσμητική καθώς δεν υπάρχει καποια υλοποίηση για να χρησιμοποιηθεί από τον χρήστη.

Το αντικείμενο hiFi λειτουργεί με τον ίδιο τρόπο όπως και το αντικείμενο stars του Session 11.

Τέλος, όταν ο χρήστης απαντήσει όλες τις ερωτήσεις θα εμφανιστεί το κείμενο στο οποίο αναφέρεται η επίδοση του στο quiz δίνοντας του τη δυνατότητα να επαναλάβει το quiz ή να προχωρήσει στο επόμενο κεφάλαιο.

5.3.4 Κεφάλαιο 21

Στο παρόν κεφάλαιο δε θα γίνει αναφορά στη δομή της σκηνής και των κομματιών του κώδικα καθώς είναι ο ίδιος με αυτόν του Κεφαλαίου 13. Θα γίνει αναφορά για την οπτική γωνία του χρήστη.

Η σκηνή του ‘Κεφαλαίου 21’ σχετίζεται με το το Κεφάλαιο 21 του βιβλίου μαθητή σελ. 70-72.

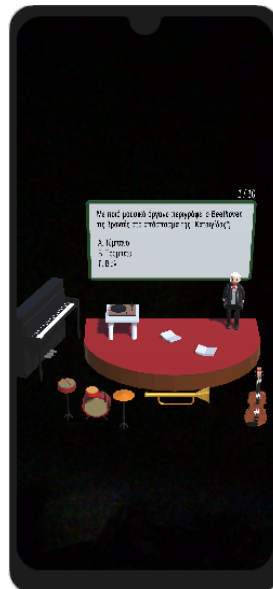
Όταν ο χρήστης κλικάρει πάνω στο κουμπί “Κεφάλαιο 21” θα αντικρίσει το Popup με τις οδηγίες. Αφού πατήσει ο χρήστης το κουμπί “OK” είναι απαραίτητο το σκανάρισμα του Image Target, Εικόνα 5.89.



Εικόνα 5.89: Image Target του Κεφαλαίου 21

Μόλις σκανάρει την εικόνα, θα εμφανιστεί η σκηνή και σε αυτή ο χρήστης θα δει εικοσιδύο 3D αντικείμενα που δημιουργήθηκαν με το Blender (Εικόνα 5.90):

- έναν συνθέτη βοηθό
- τρεία αναλόγια με ένα βιβλίο πάνω στο καθένα
- ένα πικάπ
- ένα πιάνο
- μία σκηνή
- ένα πίνακα
- ένα τραπέζι
- δύο βιβλία πάνω στη σκηνή
- πέντε διαφορετικά μουσικά όργανα και έξι εικόνες στις απαντήσεις



Εικόνα 5.90: Προβολή σκηνής κεφαλαίου 21

Έπειτα, όσον αφορά το στήσιμο της σκηνής και τον τρόπο των απαντήσεων, ο 3D πίνακας εμφανίζει πάνω του τις ερωτήσεις και τρεις επιλογές απαντήσεων. Η απάντηση δίνεται από τον χρήστη πατώντας πάνω στα 3D αντικείμενα παρομοίως με το Κεφάλαιο 13.

Στη σκηνή υπάρχει επίσης ο μουσικοσυνθέτης βοηθός, όπου στην παρούσα σκηνή έχει δημιουργηθεί ο Beethoven. Η χρήση του είναι απλά διακοσμητική καθώς δεν υπάρχει καποια υλοποίηση για να χρησιμοποιηθεί από τον χρήστη.

Το αντικείμενο hi-fi και piano λειτουργούν με τον ίδιο τρόπο όπως και το αντικείμενο stars του Session 11.

Τέλος, όταν ο χρήστης απαντήσει όλες τις ερωτήσεις θα εμφανιστεί το κείμενο στο οποίο αναφέρεται η επίδοση του στο quiz δίνοντας του τη δυνατότητα να επαναλάβει το quiz ή να προχωρήσει στο επόμενο.

5.3.5 Κεφάλαιο 22

Στο παρόν κεφάλαιο δε θα γίνει αναφορά στη δομή της σκηνής και των κομματιών του κώδικα καθώς είναι ο ίδιος με αυτόν του Κεφαλαίου 13. Θα γίνει αναφορά για την οπτική γωνία του χρήστη. Η σκηνή του 'Κεφαλαίου 22' σχετίζεται με το το Κεφάλαιο 22 του βιβλίου μαθητή σελ. 73-63.

Όταν ο χρήστης κλικάρει πάνω στο κουμπί "Κεφάλαιο 22" θα αντικρίσει το PopUp με τις οδηγίες. Αφού πατήσει ο χρήστης το κουμπί "OK" είναι απαραίτητο το σκανάρισμα του Image Target, Εικόνα 5.91.



Εικόνα 5.91: Image Target του Κεφαλαίου 22

Μόλις σκανάρει το QR, θα εμφανιστεί η σκηνή και σε αυτή ο χρήστης θα δει είκοσι 3D αντικείμενα που δημιουργήθηκαν με το Blender (Εικόνα 5.92):

- έναν συνθέτη βοηθό
- τρία αναλόγια με ένα βιβλίο πάνω στο καθένα
- μία σκηνή
- ένα πίνακα
- πέντε διαφορετικά μουσικά όργανα στη σκηνή
- εννιά διαφορετικά μουσικά όργανα στις απαντήσεις



Εικόνα 5.92: Προβολή σκηνής κεφαλαίου 21

Έπειτα, όσον αφορά το στήσιμο της σκηνής και τον τρόπο των απαντήσεων, ο 3D πίνακας εμφανίζει πάνω του τις ερωτήσεις και τρεις επιλογές απαντήσεων. Η απάντηση δίνεται από τον χρήστη πατώντας πάνω στα 3D αντικείμενα παρομοίως με το Κεφάλαιο 13.

Στη σκηνή υπάρχει επίσης ο μουσικοσυνθέτης βοηθός, όπου στην παρούσα σκηνή έχει δημιουργηθεί ο Chopin. Η χρήση του είναι απλά διακοσμητική καθώς δεν υπάρχει καποια υλοποίηση για να χρησιμοποιηθεί από τον χρήστη.

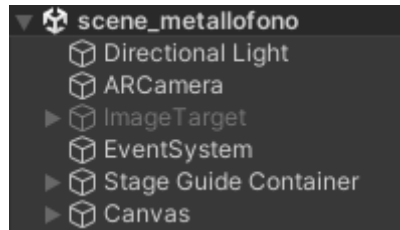
Τέλος, όταν ο χρήστης απαντήσει όλες τις ερωτήσεις θα εμφανιστεί το κείμενο στο οποίο αναφέρεται η επίδοση του στο quiz δίνοντας του τη δυνατότητα να επαναλάβει το quiz ή να προχωρήσει στο επόμενο.

5.4 Μεταλλόφωνο

Η παρούσα σκηνή προήλθε από την παρακάτω ιδέα.

Οι μαθητές αυτής της σχολικής χρονιάς καλούνται να αγοράσουν ένα μουσικό όργανο, αυτό είναι είτε φλογέρα είτε μεταλλόφωνο ώστε να εξασκήσουν τις μελωδίες που τους παραδίδουν οι διδάσκοντες. Η παροχή αυτού του μουσικού οργάνου σε μορφή AR μπορεί να βοηθήσει τον μαθητή να εξασκηθεί χωρίς να χρειαστεί να προβεί σε αγορά κάποιου μεταλλόφωνου και έτσι μπορεί να λάβει μέρος στο μάθημα ή να κάνει εξάσκηση με αυτό. Η συγκεκριμένη σκηνή προσφέρει ένα μουσικό όργανο σε μορφή AR (εδώ επιλέχθηκε ένα μεταλλόφωνο) και κατά αυτόν τον τρόπο ο μαθητής μπορεί να εξασκηθεί σε αυτό χωρίς να προβεί σε έξτρα αγορές για το μάθημα της Μουσικής.

Η δομή της σκηνής είναι η εξής (Εικόνα 5.93):



Εικόνα 5.93: Scene Metallofono

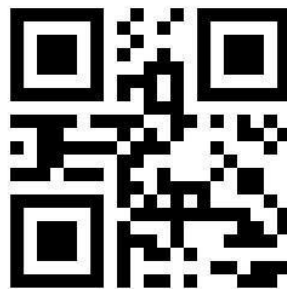
Δε θα γίνει αναφορά των κομματιών του κώδικα για την ARCamera και το παράθυρο με τις οδηγίες καθώς είναι ο ίδιος με αυτόν που χρησιμοποιήθηκε για τις προηγούμενες ενότητες.

Όταν ο χρήστης κλικάρει πάνω στο κουμπί “Μεταλλόφωνο” του Main Menu, θα εμφανιστεί το παράθυρο με τις οδηγίες για την σκηνή της ενότητας όπως φαίνεται στην (Εικόνα 5.94).



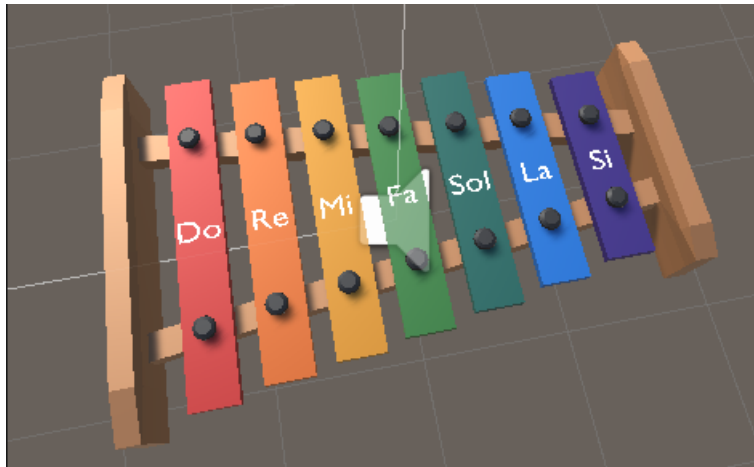
Εικόνα 5.94: Stage Guide Popup

Αφού πατήσει ο χρήστης το κουμπί “OK” είναι απαραίτητο το σκανάρισμα του QR (Image Target Εικόνα 5.95)



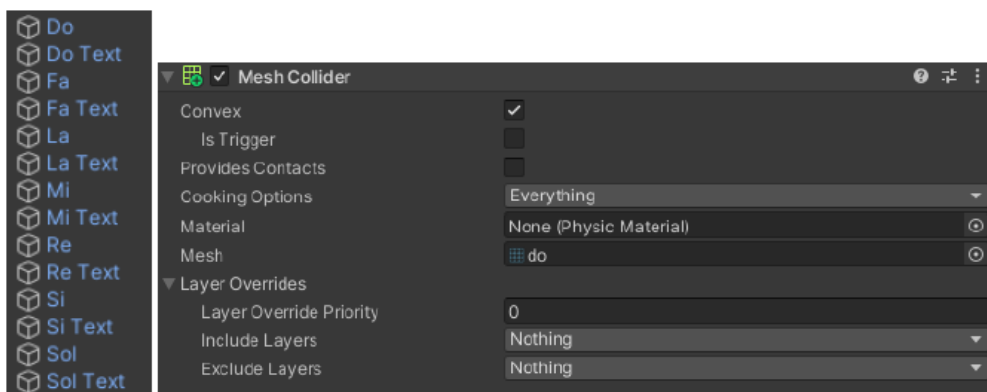
Εικόνα 5.95: Image Target του Μεταλλόφωνο

Το gameObject metallofono είναι το 3D μοντέλο ενός μεταλλόφωνου (Εικόνα 5.96) το οποίο εμφανίζεται αφότου ο χρήστης κλείσει το Stage Guide Container πατώντας πάνω στο κουμπί OK .



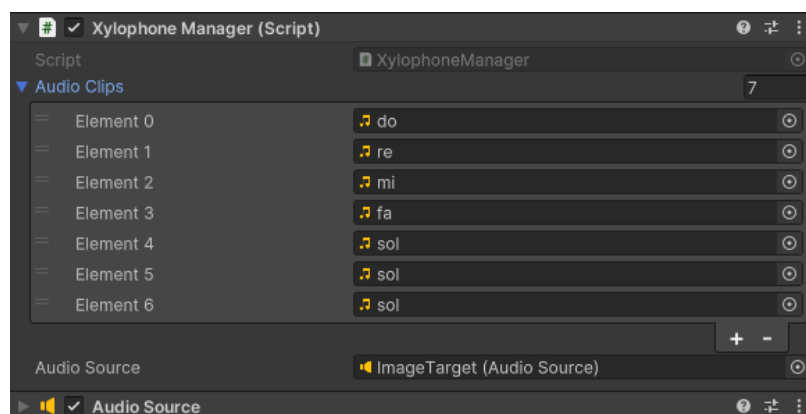
Εικόνα 5.96: 3D Μεταλλόφωνο

Κάθε νότα αποτελεί διαφορετικό gameObject και διαθέτει το δικό του Mesh Collider (Εικόνα 5.97) για να μπορέσει ο χρήστης να αλληλεπιδράσει μαζί του.



Εικόνα 5.97: Mesh Collider

Επιπλέον, το gameObject ImageTarget διαθέτει την κλάση XylophoneManager και ένα Audio Source component (Εικόνα 5.98).



Εικόνα 5.98: XylophoneManager

Η συγκεκριμένη κλάση είναι ιδιαίτερα απλή και ακολουθεί παρόμοια φιλοσοφία με αρκετά scripts που υπάρχουν στην εφαρμογή και έχουν περιγραφεί σε προηγούμενες σελίδες, (Εικόνα 5.99).

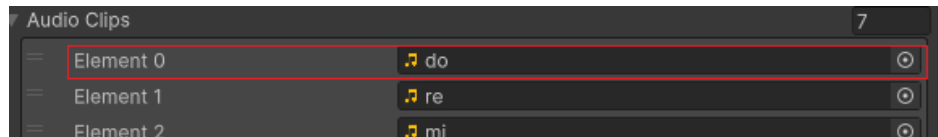
```
// Start is called before the first frame update
@ Unity Message | 0 references
void Start()
{
    _audioSource = GetComponent<AudioSource>();
}
```

Εικόνα 5.99: Unity μέθοδος Start

Η μέθοδος Start του Unity κάνει attach το Audio Source component του gameObject metallofono στην μεταβλητή _audioSource. Αυτή η ενέργεια είναι αναγκαία ώστε να μπορέσει να τροποποιηθούν οι ρυθμίσεις του συγκεκριμένου αντικειμένου.

Έτσι, όταν ο χρήστης κάνει κλικ πάνω σε μια νότα του μεταλλόφωνου (touch του χρήστη στην οθόνη), δημιουργείται ένα Raycast . Όταν αυτό συγκρουστεί (κάνει collide) με κάποιο Collider component οποιασδήποτε νότας, γίνεται trigger η if statement μέσα στην μέθοδο Update του Unity. Η private μεταβλητή buttonName καθορίζει τις παραμέτρους που θα χρησιμοποιηθούν στην μέθοδο PlayMusic μέσα στο switch case.

Για παράδειγμα, ο χρήστης κάνει touch πάνω στην νότα Do. Θα γίνει πρώτα έλεγχος αν το Raycast που δημιουργήθηκε από την ενέργεια του χρήστη συγκρουστεί με το Collider του αντικειμένου Do. Στη συνέχεια, η μέθοδος PlayMusic θα πάρει τις τιμές που αντιστοιχούν στον ήχο της νότας Do (element 0 του πίνακα _audioClips), (Εικόνα 5.100-101).



Εικόνα 5.100: Audio Clip list

```

// Update is called once per frame
@ Unity Message | 0 references
void Update()
{
    if ((Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began))
    {
        Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);
        RaycastHit hit;

        if (Physics.Raycast(ray, out hit))
        {
            _buttonName = hit.transform.name;
            switch (_buttonName)
            {
                case "Do":
                    PlayMusic(_buttonName, 0);
                    break;
                case "Re":
                    PlayMusic(_buttonName, 1);
                    break;
                case "Mi":
                    PlayMusic(_buttonName, 2);
                    break;
                case "Fa":
                    PlayMusic(_buttonName, 3);
                    break;
                case "Sol":
                    PlayMusic(_buttonName, 4);
                    break;
                case "La":
                    PlayMusic(_buttonName, 5);
                    break;
                case "Si":
                    PlayMusic(_buttonName, 6);
                    break;
                default:
                    break;
            }
        }
    }
}

```

Εικόνα 5.101: Update του Unity

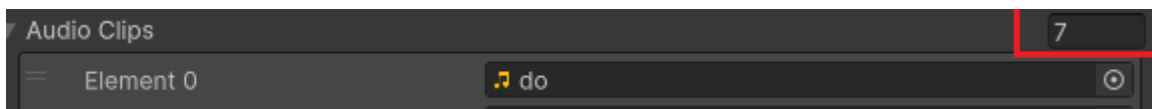
Οπότε το audio clip που θα παίζει αντιστοιχεί στο audio clip που βρίσκεται στη θέση audioClipIndex) μηδέν του πίνακα _audioClips τύπου Audio Clips. Το μέγεθος του πίνακα είναι ίσος με τον αριθμό των νότων που έχει το μεταλλόφωνο της σκηνής, δηλαδή επτά, (Εικόνα 5.102-103).

```

void PlayMusic(string buttontName, int audioClipIndex)
{
    Debug.Log("Debug message: " + buttontName + "!!!");
    _audioSource.clip = _audioClips[audioClipIndex];
    _audioSource.Play();
}

```

Εικόνα 5.102: PlayMusic μέθοδος



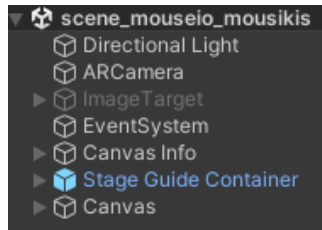
Εικόνα 5.103: Μέγεθος του πίνακα

5.5 Μουσείο Μουσικής

Η παρούσα σκηνή προήλθε από την παρακάτω ιδέα.

Οι μαθητές για να μπορέσουν να απαντήσουν κάποια από τις ερωτήσεις που δε θυμούνται καλά ή κάποια που δεν ξέρουν, ανατρέχουν στο σχολικό βιβλίο για να πάρουν παραπάνω πληροφορίες ώστε να μπορέσουν να απαντήσουν. Σε αυτή την σκηνή υπάρχει πηγή πληροφοριών που μπορούν να βοηθήσουν τον μαθητή στα κεφάλαια προς εξέταση χωρίς να χρειαστεί να ανατρέξει στο βιβλίο.

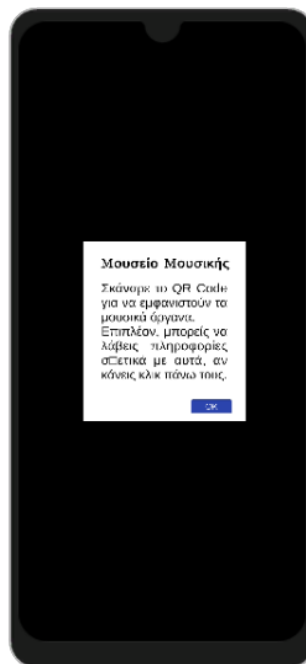
Η δομή της σκηνής είναι η εξής (Εικόνα 5.104):



Εικόνα 5.104: Scene Μουσείο Μουσικής

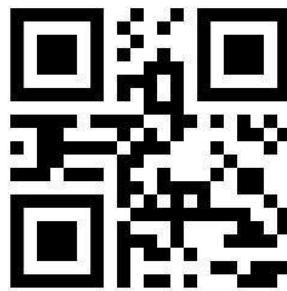
Δε θα γίνει αναφορά των κομματιών του κώδικα για την ARCamera και το παράθυρο με τις οδηγίες καθώς είναι ο ίδιος με αυτόν που χρησιμοποιήθηκε για τις προηγούμενες ενότητες.

Όταν ο χρήστης κλικάρει πάνω στο κουμπί “Μουσείο Μουσικής” θα αντικρίσει το παράθυρο με τις οδηγίες όπως φαίνεται στην (Εικόνα 5.105).



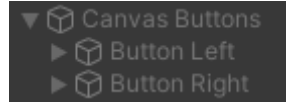
Εικόνα 5.105: Stage Guide Popup

Αφού πατήσει ο χρήστης το κουμπί “OK” είναι απαραίτητο το σκανάρισμα του QR (Image Target (Εικόνα 5.106))

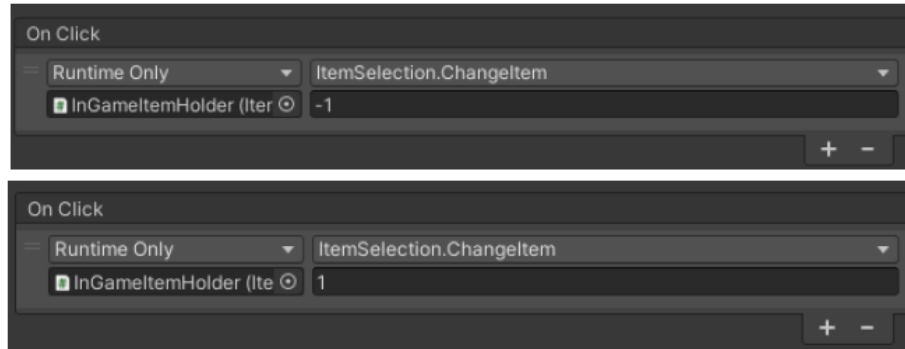


Εικόνα 5.106: Image Target του Μουσείο Μουσικής

Το Canvas Buttons (Εικόνα 5.107) περιέχει δύο buttons τα οποία διαθέτουν το ίδιο setup. Αυτό ουσιαστικά σημαίνει ότι το button component του κάθε button (Button Left και Button Right) εκτελεί την public μέθοδο `ChangeItem` (Εικόνα 5.108) της κλάσης `ItemSelection` μέσω του Unity Event `OnClick` με την μόνη διαφορά η τιμή της integer τιμής που δίνεται ως παράμετρος (περισσότερα για το πως λειτουργεί παρακάτω).

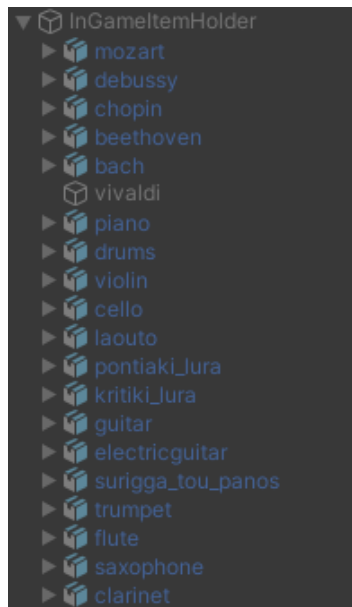


Εικόνα 5.107: Canvas Buttons



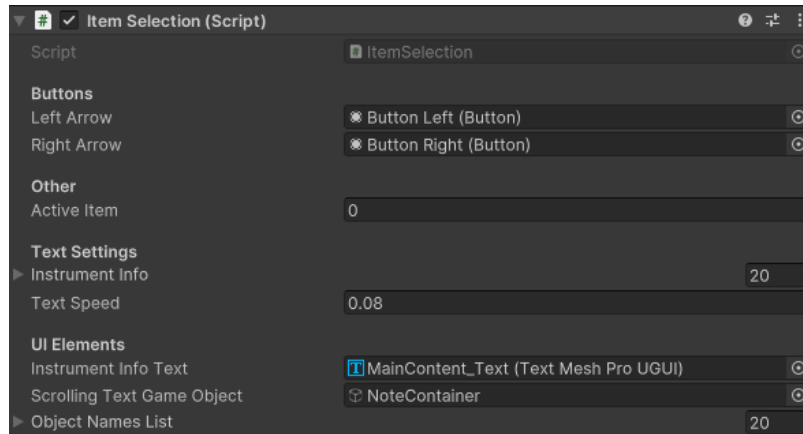
Εικόνα 5.108: Unity Event `OnClick` με τη μέθοδος `ChangeItem`

Το `gameObject` `InGameItemHolder` (Εικόνα 5.109), όπως προδίδει και η ονομασία του, έχει τον ρόλο ενός δοχείου που περιέχει όλα τα 3D models της εφαρμογής. Το πρώτο child `gameObject` (`mozart`) είναι ενεργό ενώ όλα τα υπόλοιπα βρίσκονται ανενεργή κατάσταση (`disabled`).



Εικόνα 5.109: `gameObject` `InGameItemHolder`

Χρησιμοποιεί την κλάση `ItemSelection` (Εικόνα 5.110) όπως φαίνεται παρακάτω:



Εικόνα 5.110: ItemSelection

Αφού ενεργοποιηθεί το gameObject InGameItemHolder από το script του Stage Guide Container (βλέπε προηγούμενες σελίδες) ο χρήστης μπορεί να σκανάρει το QR code και να εμφανιστούν στην οθόνη της συσκευής του το Canvas Buttons και το πρώτο child gameObject του InGameItemHolder (το 3D μοντέλο του moztart).

Κάνοντας κλικ στα κουμπιά (Button Left και Button Right) μπορεί να πλοηγηθεί στα περιεχόμενα του InGameItemHolder δοχείου. Η κατεύθυνση είναι είτε προς τα μπροστά, είτε προς τα πίσω. Αυτό είναι εφικτό εξαιτίας της ChangeItem (Εικόνα 5.111).

```
public void ChangeItem(int _change)
{
    _scrollingTextGameObject.SetActive(false);
    _activeItem += _change;

    if (_activeItem > transform.childCount - 1)
        _activeItem = 0;
    else if (_activeItem < 0)
        _activeItem = transform.childCount - 1;

    SelectItem(_activeItem);
}
```

Εικόνα 5.111: ChangeItem

Η μέθοδος SelectItem (Εικόνα 5.112) παίρνει τη θέση του child αντικειμένου και το εμφανίζει στο χώρο.

```
void SelectItem(int _index)
{
    for (int i = 0; i < transform.childCount; i++)
    {
        transform.GetChild(i).gameObject.SetActive(i == _index);
    }
}
```

Εικόνα 5.112: SelectItem

Πέρα από την τρισδιάστατη παρατήρηση του αντικειμένου, ο χρήστης/μαθητής έχει την δυνατότητα να λάβει πληροφορίες για το ενεργό 3D αντικείμενο αν κάνει κλικ πάνω του με παρόμοιο τρόπο όπως και στις προηγούμενες ενότητες.

Η private μεταβλητή _objectName παίρνει το όνομα του gameObject που ήρθε σε επαφή με το Raycast και μέσω της foreach λούπας διατρέχει τα περιεχόμενα της _objectNamesList λίστας (Εικόνα

5.113) μέχρις ούτε βρει το αντικείμενο με το συγκεκριμένο όνομα ώστε να ενεργοποιήσει το text message (Εικόνα 5.114) με τις πληροφορίες για αυτό.

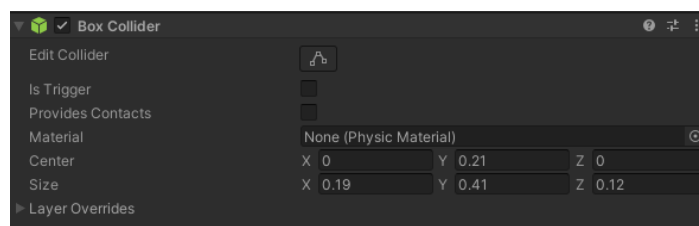
```
[SerializeField]
List<string> _objectNamesList = new List<string> {
    "mozart", "debussy", "chopin", "beethoven", "bach", "vivaldi", "piano",
    "drums", "violin", "cello", "laouto", "pontiaki_lura", "kritiki_lura",
    "guitar", "electricguitar", "surigga_tou_panos", "trumpet", "flute",
    "saxophone", "clarinet" };
```

Εικόνα 5.113: objectNamesList

```
// Update is called once per frame
@ Unity Message | 0 references
void Update()
{
    if ((Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began))
    {
        Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);
        RaycastHit hit;
        if (Physics.Raycast(ray, out hit))
        {
            _objectName = hit.transform.name;
            foreach (string name in _objectNamesList)
            {
                if (name.Equals(_objectName))
                    ActivateText(_activeItem);
            }
        }
    }
}
```

Εικόνα 5.114: SelectItem

Για την αλληλεπίδραση χρήστη-3D αντικειμένου είναι απαραίτητη η χρήση ενός Box Collider. Αυτό ισχύει για κάθε child gameObject του InGameItemHolder (Εικόνα 5.115).



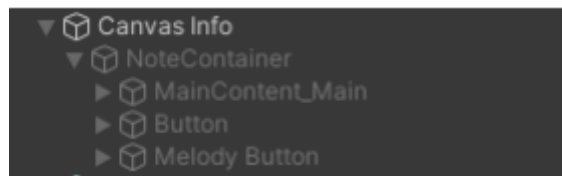
Εικόνα 5.115: Box Collider

Η μέθοδος ActivateText (Εικόνα 5.116) ετοιμάζει το κείμενο και το εμφανίζει στην οθόνη.

```
void ActivateText(int index)
{
    _instrumentInfoText.text = string.Empty;
    _currentDisplayingText = index;
    _scrollingTextGameObject.SetActive(true);
    _instrumentInfoText.text = _instrumentInfo[_currentDisplayingText];
}
```

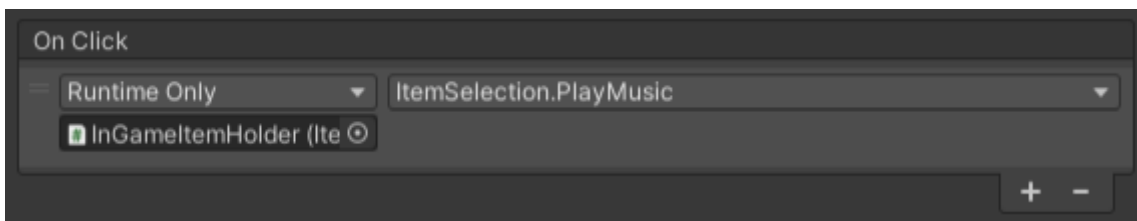
Εικόνα 5.116: ActivateText μέθοδος

Στο gameObject NoteContainer υπάρχει το κουμπί Melody. (Εικόνα 5.117)



Εικόνα 5.117: Melody button

Το οποίο διαθέτει το εξής setup, (Εικόνα 5.118) :



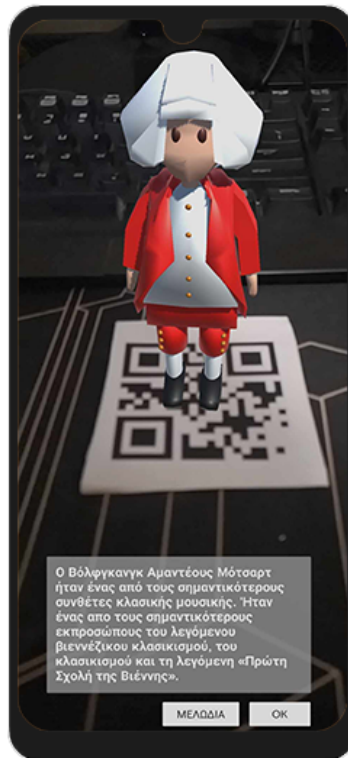
Εικόνα 5.118: OnClick μέθοδος

Όταν γίνει κλικ πάνω του εκτελείται η μέθοδος ItemSelection.PlayMusic παίζοντας μια μελωδία για το εκάστοτε ενεργό αντικείμενο, (Εικόνα 5.119).

```
public void PlayMusic()
{
    if (_audioSource.isPlaying)
    {
        _audioSource.Stop();
        Debug.Log("Debug message: Source and particle stop");
    }
    else
    {
        _audioSource.clip = _audioClips[_activeItem];
        _audioSource.Play();
        Debug.Log("Debug message: Source and particle play");
    }
}
```

Εικόνα 5.119: PlayMusic μέθοδος

Το αποτέλεσμα αυτής της μεθόδου φαίνεται στην εικόνα (Εικόνα 5.120):



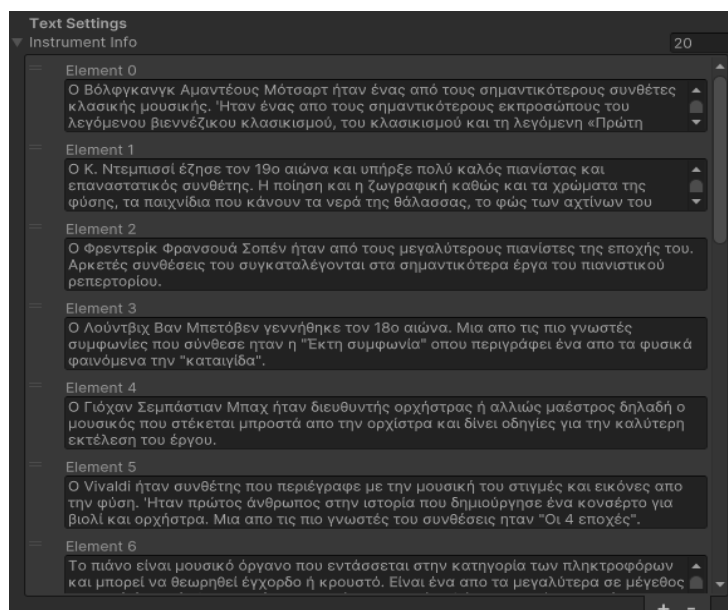
Εικόνα 5.120: Πληροφορίες για τα αντικείμενα

Αρχικά, αδειάζει οποιοδήποτε text μπορεί να έχει μείνει στο `gameObject_instrumentInfoText` και έπειτα δέχεται το κείμενο που υπάρχει στη θέση `_currentDisplayInText` του πίνακα `_instrumentInfo`, (Εικόνα 5.121).

```
[SerializeField] [TextArea] string[] _instrumentInfo;
```

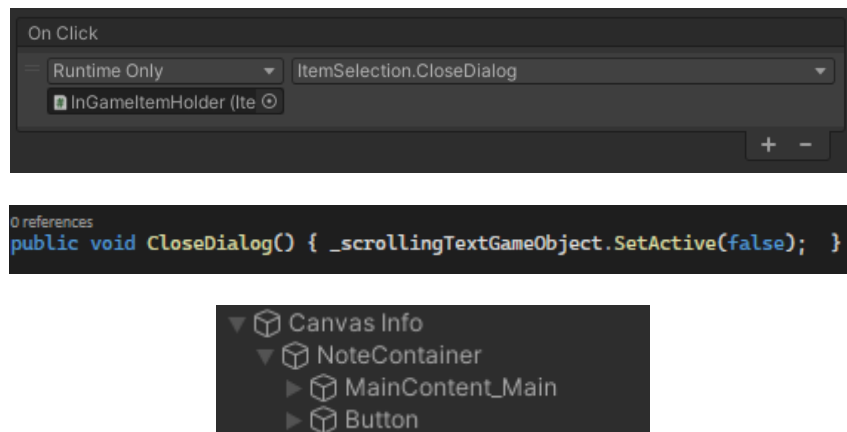
Εικόνα 5.121: Αρχικοποίηση πίνακα `_instrumentInfo`

Το κείμενο δίνεται μέσω UI, από τον Unity Editor (Εικόνα 5.122):



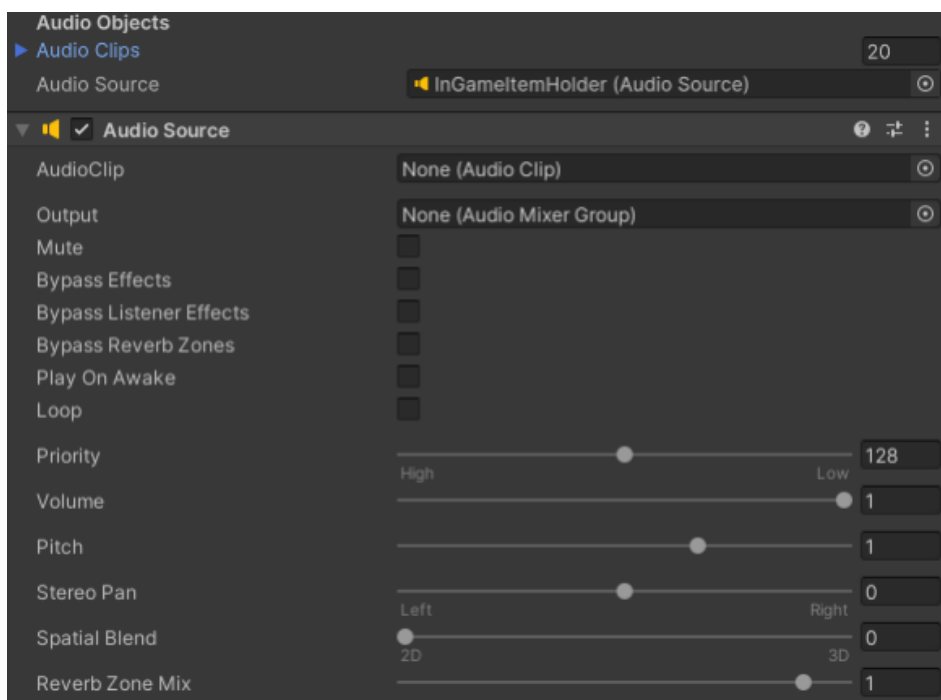
Εικόνα 5.122: Unity Editor

Κάνοντας κλικ στο κουμπί OK το gameObject NoteContainer γίνεται ανενεργό κρύβοντας έτσι το text message (Εικόνα 5.123).



Εικόνα 5.123: Close Dialog

Για να υλοποιηθεί η μελωδίας στα αντικείμενα χρειάστηκαν να προστεθούν νέες μεταβλητές στην κλάση ItemSelection (_audioClips, _audioSource) και ένα νέο component στο gameObject InGameItemHolder, (Εικόνα 5.124).



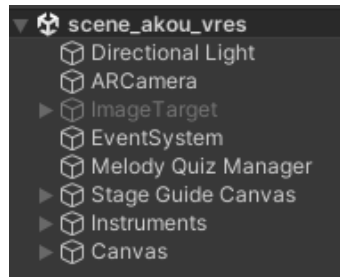
Εικόνα 5.124: Unity Editor

5.6 Άκου - Βρες

Η παρούσα σκηνή προήλθε από την παρακάτω ιδέα.

Η μαθητές την ώρα που μαθήματος καλούνται να ακούσουν κάποιες μελωδίες και να τις αναγνωρίσουν. Μέσα από αυτή την σκηνή οι μαθητές μαθαίνουν για τις συνθέσεις διάσημων μουσικών καθώς ακούν την μελωδία και απαντούν στις ερωτήσεις χωρίς την χρήση κάποια άλλης συσκευής.

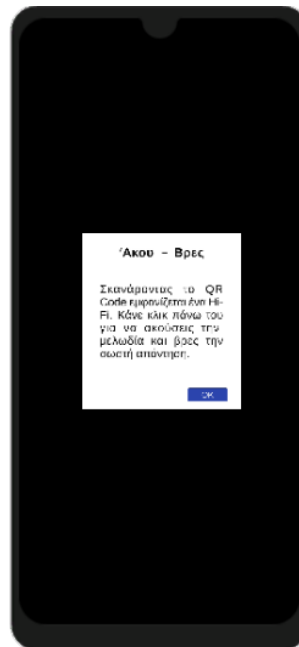
Η δομή της σκηνής είναι η εξής (Εικόνα 5.125):



Εικόνα 5.125: Scene Άκου Βρες

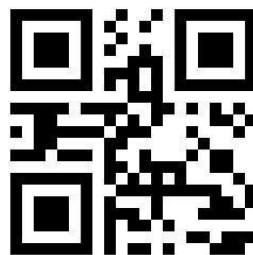
Δε θα γίνει αναφορά των κομματιών του κώδικα για την ARCamera και το παράθυρο με τις οδηγίες καθώς είναι ο ίδιος με αυτόν που χρησιμοποιήθηκε για τις προηγούμενες ενότητες.

Όταν ο χρήστης κλικάρει πάνω στο κουμπί “Άκου - Βρες” θα αντικρίσει το παράθυρο με τις οδηγίες όπως φαίνεται στην (Εικόνα 5.126).



Εικόνα 5.126: Stage Guide Pop-up

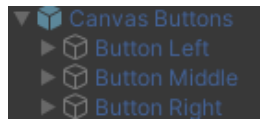
Αφού πατήσει ο χρήστης το κουμπί “OK” είναι απαραίτητο το σκανάρισμα του QR (Image Target (Εικόνα 5.127))



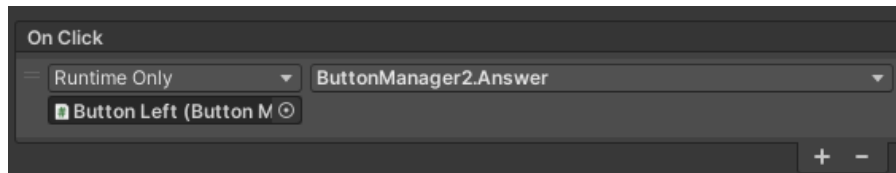
Εικόνα 5.127: Image Target του Μεταλλόφωνο

Κεφάλαιο 5

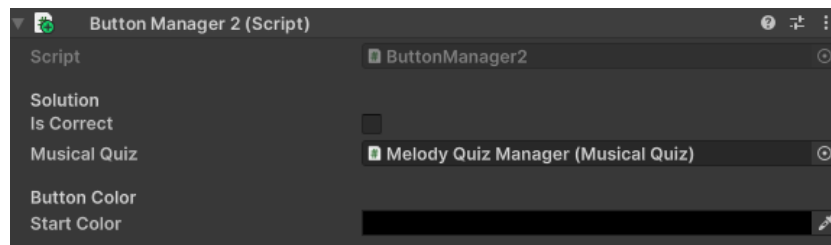
Το Canvas Buttons (Prefab) (Εικόνα 5.128) περιέχει τρία buttons τα οποία όπως ήταν αναμενόμενο ακολουθούν την ίδια δομή. Αυτό ουσιαστικά σημαίνει ότι το button component του κάθε button (Button Left, Button Middle και Button Right) (Εικόνα 5.129) εκτελεί την public μέθοδο της κλάσης ButtonManager2 μέσω του Unity Event OnClick (Εικόνα 5.130).



Εικόνα 5.128: Canvas Buttons



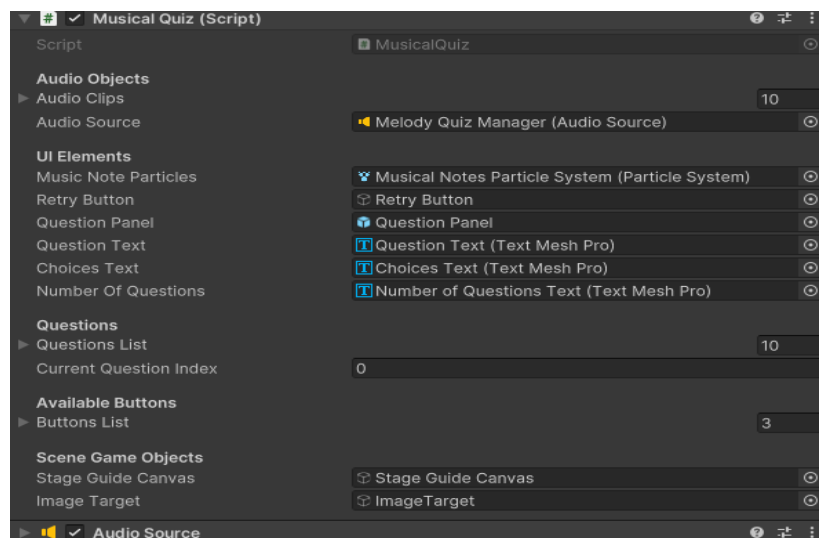
Εικόνα 5.129: OnClick Answer μέθοδος



Εικόνα 5.130: ButtonManager2 μέθοδος

Η μέθοδος αυτή απλά παίρνει το Image component του εκάστοτε button και αλλάζει το χρώμα του ανάλογα με την τιμή της μεταβλητής `_isCorrect`.

Το αντικείμενο `hifi` λειτουργεί με τον πανομοιότυπο τρόπο όπως και το αντικείμενο `stars` του Session 11. Αποτελείται από ένα `Box Collider` ωστόσο την αλληλεπίδραση του με τον χρήστη την διαχειρίζεται το `gameObject` `Melody Quiz Manager` με τη χρήση της κλάσης `MusicalQuiz` (Εικόνα 5.131).



Εικόνα 5.131: MusicalQuiz μέθοδος

Όσον αφορά την αλληλεπίδραση καθαυτή γίνεται στην Update μέθοδο του Unity (Εικόνα 5.132).

```

void Update()
{
    if ((Input.touchCount > 0 && Input.touches[0].phase == TouchPhase.Began))
    {
        Ray ray = Camera.main.ScreenPointToRay(Input.GetTouch(0).position);
        RaycastHit hit;

        if (Physics.Raycast(ray, out hit))
        {
            _objectName = hit.transform.name;
            switch (_objectName)
            {
                case "hifi":
                    PlayMusic(_audioClipIndex);
                    break;
                default:
                    break;
            }
        }
    }

    if (!_audioSource.isPlaying) { _musicNoteParticles.Stop(); }
}

```

Εικόνα 5.132: Update μέθοδο του Unity

Η private μεταβλητή `_objectName` καθορίζει τις παραμέτρους που θα χρησιμοποιηθούν στην μέθοδο `PlayMusic` μέσα στο switch case.

Στην περίπτωση του `gameObject hifi` προφανώς εκτελείται το case “hifi”. Η `PlayMusic` λειτουργεί ως εξής, (Εικόνα 5.133):

```

void PlayMusic(int audioClipIndex)
{
    if (_audioSource.isPlaying)
    {
        _audioSource.Stop();
        _musicNoteParticles.Stop();
        Debug.Log("Debug message: Source and particle stop");
    }
    else
    {
        _audioSource.clip = _audioClips[audioClipIndex];
        _audioSource.Play();
        _musicNoteParticles.Play();

        Debug.Log("Debug message: Source and particle play");
    }
}

```

Εικόνα 5.133: PlayMusic μέθοδος

Αρχικά, ελέγχει αν το Audio Source (η μεταβλητή `_audioSource`) παίζει κάποιον ήχο ή όχι. Στην περίπτωση που κάτι τέτοιο ισχύει, σταματάει την όποια μελωδία και το particle system. Σε αντίθετη περίπτωση, το audio clip που θα παίζει το Audio Source component είναι αυτό που βρίσκεται στην θέση `audioClipIndex` του πίνακα `_audioClips`.

Τέλος, παίζει και το particle system. Στο αντικείμενο Event System έχει προστεθεί η κλάση `BackButton` η οποία είναι υπεύθυνη για την εμφάνιση του Prompt Canvas αφού γίνει κλικ στο back button της android συσκευής. Αυτό το tool είναι αναγκαίο ώστε να επιστρέψει ο χρήστης ανά πάσα στιγμή στην αρχική σκηνή, το Main Menu.

Το κύριο `gameObject` της σκηνής και υπεύθυνο για την διαχείριση του musical quiz είναι το `Melody Quiz Manager`.

Σε πρώτη φάση, εκτελείται η μέθοδος `Start` του Unity για να γίνει το initialization του quiz. Το text component του `Number of Questions Text` παίρνει ως τιμή το συνολικό μέγεθος της λίστας με τις ερωτήσεις, (Εικόνα 5.134).

```

void Start() => ShowQuiz();

1 reference
public void ShowQuiz()
{
    _totalQuestions = _questionsList.Count;
    GenerateQuestion();
}

```

Εικόνα 5.134: ShowQuiz μέθοδος

Έπειτα η μέθοδος GenerateQuestion, έχει προαναφερθεί [Εικόνα 5.55], αφού ελέγξει το μέγεθος της λίστας με τις ερωτήσεις επιλέγει τυχαία ένα index από αυτήν και στη συνέχεια κάνει parse στο text component του Question Text (βλέπε 3D model πίνακα) την ερώτηση (που ανήκει σε αυτό το index).

Η μεταβλητή _numberOfQuestions τύπου TMP_Text απεικονίζει σε UI επίπεδο το σημείο στο οποίο βρίσκεται ο χρήστης και υπάρχει με απώτερο σκοπό την ενημέρωση του για το progression που έχει πραγματοποιήσει στο quiz.

Στη συνέχεια γίνεται το initialization των διαθέσιμων απαντήσεων και αντιστοιχισής τους στα 3 buttons που υπάρχουν στη σκηνή.

Το ποια απάντηση είναι η σωστή καθορίζεται μέσω UI, δηλαδή μέσα από το Unity Editor και αυτό διότι η λίστα με τις ερωτήσεις (μεταβλητή _questionsList) είναι ενός custom τύπου. Στο παραπάνω κομμάτι κώδικα φαίνεται και ο τρόπος που αλλάζει τιμή η public μεταβλητή _isCorrect της κλάσης ButtonManager2, (Εικόνα 5.135).

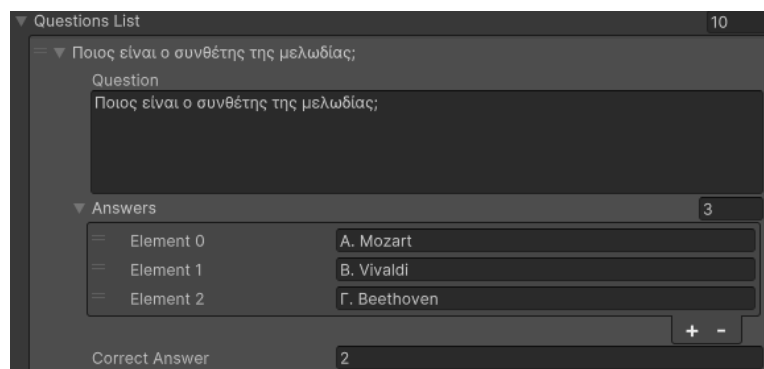
```

[Header("Questions")]
[SerializeField] List<QuizFormat> _questionsList;

```

Εικόνα 5.135: _questionsList

Η λίστα QuestionsList διαθέτει δέκα ερωτήσεις, η πρώτη (Element 0) ερώτηση (public string question) ισούται με “Ποιος είναι ο συνθέτης της μελωδίας;”. Ο πίνακας Answers (public string[] answers) περιέχει τις πιθανές απαντήσεις. Και τέλος, η μεταβλητή Correct Answer (public int correctAnswer) δηλώνει τη σωστή απάντηση, (Εικόνα 5.136).



Εικόνα 5.136: QuestionsList Unity Editor

Έτσι, όταν γίνεται κλικ σε κάποιο από τα τρία κουμπιά εκτελείται η public μέθοδος Answer της κλάσης ButtonManagerModels με στόχο να αλλάξει το χρώμα του επιλεγμένου button. Επιπλέον, εκτελείται η public μέθοδος Answer της κλάσης QuizManagerModels. Δίνεται ως παράμετρος στη μέθοδο η bool public μεταβλητή _isCorrect.

Αν λοιπόν η παράμετρος που δόθηκε είναι αληθής τότε αυξάνεται κατά μια μονάδα ο συνολικός αριθμός των σωστών απαντήσεων. Το ίδιο πραγματοποιείται και για τις ερωτήσεις που έχουν απαντηθεί (gameObject Number of Questions Text).

Αφαιρείτε στη συνέχεια η ερώτηση από την λίστα. Αυξάνεται κατά μία μονάδα ο αριθμός των ερωτήσεων που έχουν απαντηθεί/εμφανιστεί έως τώρα, σταματάει η μελωδία και αυξάνεται η τιμή του `_audioClipIndex`. Απενεργοποιούνται για λίγα δευτέρα τα buttons (μέθοδος `ResetButtons`) και γίνεται retrieve η επόμενη ερώτηση με τον ίδιο τρόπο που αναλύθηκε προηγουμένως, (Εικόνα 5.137).



Εικόνα 5.137: Σκηνή Άκου Βρές

Αν το μέγεθος της λίστας δεν είναι πλέον μεγαλύτερο του μηδενός, αυτό σημαίνει ότι έχουν απαντηθεί όλες (παραπάνω αναφέρθηκε ότι με κάθε απάντηση διαγράφεται η εκάστοτε ερώτηση από την λίστα) και εκτελείται το “else” κομμάτι του κώδικα..

Απενεργοποιείται το `ImageTarget` από την σκηνή και εμφανίζεται ξανά με animation το `Stage Guide Canvas`. Αυτή τη φορά όμως με διαφορετικό κείμενο στο οποίο αναφέρεται η επίδοση του χρήστη/μαθητή στο quiz.

Τέλος, γίνεται ενεργό του gameObject `RetryButton` του `Stage Guide Canvas` δίνοντας κατά αυτόν τον τρόπο τη δυνατότητα στον χρήστη να επαναλάβει το quiz.

5.7 Επίλογος

Στο πέμπτο κεφάλαιο έγινε η πλήρη αναφορά της ανάπτυξης του κώδικα καθώς και η παρουσίαση της εφαρμογής για κάθε σκηνή και βήμα που θα κάνει ο χρήστης.

Κεφάλαιο 6ο: Συμπεράσματα και προτάσεις για βελτίωση

Η Επαυξημένη Πραγματικότητα έχει ευρεία εφαρμογή σε διάφορους τομείς και συνεχώς εξελίσσεται με νέες τεχνολογίες και εφαρμογές που προσφέρουν εντυπωσιακές και ενισχυμένες εμπειρίες στους χρήστες.

Η διδασκαλία θα μπορεί να γίνει πιο ευχάριστη και πιο αποτελεσματική από τους διδάσκοντες καθώς δίνεται στους μαθητές η ευκαιρία να παίρνουν πληροφορίες και γνώσεις για τα μαθήματα τους μέσα από τις εφαρμογές επαυξημένης πραγματικότητας.

Η οπτικοακουστική διδασκαλία τις περισσότερες φορές βοηθάει τους μαθητές να αφομοιώσουν το μάθημα τους καλύτερα από το να ακούνε μόνο θεωρητικά τα μαθήματα τους από τους διδάσκοντες. Οι μαθητές λαμβάνουν ερεθίσματα εκτελώντας τις ασκήσεις τους μέσα από το παιχνίδι και έτσι η εκμάθηση γίνεται διασκεδαστική.

Η εφαρμογή ArMusic είναι μια beta έκδοση στην οποία μπορούν να γίνουν αρκετές βελτιώσεις.

Αρχικά, περισσότερες 3D φιγούρες μουσικών οργάνων με την βοήθεια του Blender θα έκαναν τις ασκήσεις (quiz) να εμπλουτιστούν περισσότερο π.χ. λεπτομέρειες πάνω στους συνθέτες βοηθούς που βρισκόταν σε κάθε σκηνή . Η προσθήκη νέων αντικειμένων στο χώρο (μουσικών οργάνων και μη) όπου ο χρήστης θα μπορούσε να μετακινήσει . Πάνω σε αυτή την βελτίωση θα μπορούσε να δημιουργηθεί ένα νέο quiz όπου θα δινόταν η δυνατότητα στον χρήστη με drag and drop κίνησης να εκτελεστεί κάποιου είδους άσκηση. Επιπλέον, θα μπορούσε να υπάρξει κάποιο quiz αναγραμματισμού ή σταυρόλεξου με την δυνατότητα των χρηστών να χρησιμοποιούν τα 3D αντικείμενα (γράμματα ή μουσικά όργανα) για να βρουν την λέξη που τους ζητείται στο quiz.

Η προσθήκη περισσότερων εφέ τόσο ηχητικών όσο και κίνησης στα αντικείμενα θα έκανε την εφαρμογή πιο ευχάριστη στον χρήστη. Κάποιες από αυτές τις προσθήκες θα ήταν τα ηχητικά εφέ στις σωστές και λάθος απαντήσεις, όπως, επίσης πάνω στα μουσικά όργανα για να μπορεί ο μαθητής να έχει μια πιο ρεαλιστική ακουστική εικόνα για το κάθε μουσικό όργανο που βλέπει μέσα στην εφαρμογή. Κίνηση πάνω στους χαρακτήρες των μουσικών για να δίνουν κάποια βοήθεια στον μαθητή.

Τέλος, μια μορφή πρόσκλησης προς τον χρήστη θα ήταν η προσθήκη χρονικού ορίου ανάμεσα στις ερωτήσεις.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Unity Documentation, "Unity User Manual", [Online]. Available: <https://docs.unity3d.com/Manual/index.html>
- [2] Unity Documentation, [Online]. Available: <https://unity.com>
- [3] Unity Documentation, "About AR Foundation", [Online]. Available: <https://docs.unity3d.com/Packages/com.unity.xr.foundation@4.1/manual/index.html>
- [4] Unity Documentation, "Unity Hub", [Online]. Available: <https://docs.unity3d.com/2020.1/Documentation/Manual/GettingStartedUnityHub.html>
- [5] Vuforia Developer Library, "Getting Started", [Online]. Available: <https://library.vuforia.com>
- [6] ARMaker - Augmented Reality Marker Generator, [Online]. Available: <https://shawnlehner.github.io/ARMaker/>
- [7] Introduction to Augmented Reality, R. Silva, J. C. Oliveira, G. A. Giraldi, National Laboratory for Scientific Computation, Av. Getulio Vargas, 333 - Quitandinha - Petropolis-RJ Brazil
- [8] Caudell, T.P., and Mizell, D.W. (1992)"Augmented Reality: an application of heads-up display technology to manual manufacturing processes." In Proceedings of the Twenty-Fifth Hawaii International Conference on Systems Science, Kauai, Hawaii, 7th-10th Jan. 1992
- [9] PCMag, "Augmented Reality (AR) vs. Virtual Reality (VR): What's the Difference?", [Online]. Available: <https://www.pcmag.com/news/augmented-reality-ar-vs-virtual-reality-vr-whats-the-difference>
- [10] Brigham, T.J. Reality check: Basics of augmented, virtual, and mixed reality. Med. Ref. Serv. Q. 2017, 36, 171–178.
- [11] Di Serio, Á., Ibáñez, M. B., & Kloos, C. D. (2013). "Impact of an augmented reality system on students' motivation for a visual art course". Computers & Education, Vol. 68, 586–596
- [12] Unity Documentation, "UI Button onClick", [Online]. Available: <https://docs.unity3d.com/530/Documentation/ScriptReference/UI.Button.onClick.html>
- [13] Unity Documentation, "Rect Transform", [Online]. Available: <https://docs.unity3d.com/ScriptReference/RectTransform.html>
- [14] Unity Documentation, "MonoBehaviour.OnEnable()", [Online]. Available: <https://docs.unity3d.com/530/Documentation/ScriptReference/MonoBehaviour.OnEnable.html>
- [15] Unity Documentation, "Prefabs", [Online]. Available: <https://docs.unity3d.com/Manual/Prefabs.html>
- [16] Unity Documentation, "Start Coroutine", [Online]. Available: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.StartCoroutine.html>
- [17] Unity Documentation, "Box Collider", [Online]. Available: <https://docs.unity3d.com/Manual/class-BoxCollider.html>
- [18] Flat Icons [Online]. Available: <https://www.flaticon.com>

[19] Blender [Online]. Available: <https://www.blender.org>

[20] Blender Manual, Extrude Region [Online]. Available: https://docs.blender.org/manual/en/latest/modeling/meshes/tools/extrude_region.html#

[21] Adobe Photoshop Manual, Photoshop User Guide [Online]. Available: <https://helpx.adobe.com/photoshop/user-guide.html>

[22] Audacity Manual, Audacity Reference Manual [Online]. Available: <https://manual.audacityteam.org>