



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΦΑΡΜΟΓΗ ΕΠΑΥΞΗΜΕΝΗΣ  
ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ  
ΜΟΥΣΙΚΟΥ ΟΡΓΑΝΟΥ ΣΕ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ

ΟΡΦΑΝΙΔΗΣ ΧΡΗΣΤΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΣΤΑΜΟΥΛΗΣ ΓΕΩΡΓΙΟΣ  
Καθηγητής

ΣΥΝΕΠΙΒΛΕΠΩΝ

ΛΟΥΚΑΚΗΣ ΓΕΩΡΓΙΟΣ  
Διδάσκων Π.Δ. 407

Λαμία 2021





ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΦΑΡΜΟΓΗ ΕΠΑΥΞΗΜΕΝΗΣ  
ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ  
ΜΟΥΣΙΚΟΥ ΟΡΓΑΝΟΥ ΣΕ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ

ΟΡΦΑΝΙΔΗΣ ΧΡΗΣΤΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

ΣΤΑΜΟΥΛΗΣ ΓΕΩΡΓΙΟΣ  
Καθηγητής

ΣΥΝΕΠΙΒΛΕΠΩΝ

ΛΟΥΚΑΚΗΣ ΓΕΩΡΓΙΟΣ  
Διδάσκων Π.Δ. 407

Λαμία 2021





UNIVERSITY OF  
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE &  
TELECOMMUNICATIONS

AUGMENTED REALITY APPLICATION FOR  
MUSICAL INSTRUMENT LEARNING ON  
MOBILE DEVICES

ORFANIDIS CHRISTOS

FINAL THESIS

ADVISOR

STAMOULIS GEORGIOS  
Professor

CO ADVISOR

LOUKAKIS GEORGIOS  
Adjunct Professor

Lamia 2021



«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις <sup>(1)</sup>, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 12/03/2021

Ο Δηλών



ΟΡΦΑΝΙΔΗΣ ΧΡΗΣΤΟΣ

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»







## ΠΕΡΙΛΗΨΗ

---

Η τεχνολογία της Επαυξημένης Πραγματικότητας (AR) κατάφερε να σπάσει το φράγμα ανάμεσα στον πραγματικό και τον ψηφιακό κόσμο. Αν και πρόκειται για νέα τεχνολογία, εφαρμόζεται σε πολλούς τομείς της καθημερινότητας και της επιστήμης. Στην παρούσα πτυχιακή εργασία παρουσιάζεται η ανάπτυξη μίας εφαρμογής Επαυξημένης Πραγματικότητας (AR) στον τομέα της επιστήμης της εκπαίδευσης, με εκτενή αναφορά στην διαδικασία της υλοποίησης και στην μεθοδολογία που εφαρμόστηκε στο πρόγραμμα Unreal Engine 4. Κύρια λειτουργία της είναι η οπτικοποίηση των συγχορδίων της κιθάρας με χρήση τρισδιάστατων (3D) μοντέλων που εμπλουτίζουν τον πραγματικό κόσμο, επιτυγχάνοντας δυνατότητα θέασης από διαφορετικές οπτικές γωνίες και κατ' επέκταση μετάδοση πληροφορίας που αδυνατεί να αποτυπωθεί σε έντυπο μέσο. Επιπλέον, η εφαρμογή δίνει την δυνατότητα αναπαραγωγής βιντεοσκοπημένου υλικού μέσω ασκήσεων μελέτης και παρέχει δύο σημαντικά εργαλεία, ένα ψηφιακό κουρδιστήρι και έναν ψηφιακό μετρονόμο. Στόχος της είναι η διάθεση χρήσιμων λειτουργιών, που καθιστούν την εκμάθηση κιθάρας ευκολότερη διαδικασία σε συνδυασμό με τα παραδοσιακά μέσα εκμάθησης (συγγράμματα).



## ABSTRACT

---

Augmented Reality (AR) technology has managed to break the barrier between the real and digital world. Although it is a new technology, it is applied in many areas of everyday life and science. This dissertation presents the development of an Augmented Reality (AR) application in the field of educational science, with extensive reference to the implementation process and methodology applied in the Unreal Engine 4 program. Its main function is to visualize the guitar chords with the use of three-dimensional (3D) models that enrich the real world, achieving the ability to view from different angles and consequently the transmission of information that can not be captured in print media. In addition, the application enables the reproduction of videotaped material of study exercises and provides two important tools, a digital tuner and a digital metronome. Its goal is to provide useful functions that make the guitar learning an easier process in combination with traditional learning tools (books).



## ΕΥΧΑΡΙΣΤΙΕΣ

---

Με την ολοκλήρωση της πτυχιακής εργασίας μου, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους όσους συνέβαλαν άμεσα ή έμμεσα στην εκπόνησή της.

Αρχικά, ευχαριστώ θερμά τον διδάσκοντα του μαθήματος «Αρχιτεκτονικής και Ανάπτυξης Παιγνίων», κύριο Γιώργο Λουκάκη, για την επιστημονική του καθοδήγηση, τις υποδείξεις του, τη συνεχή υποστήριξη και το αμείωτο ενδιαφέρον που έδειξε από την αρχή μέχρι το τέλος.

Επίσης, ευχαριστώ τον καθηγητή κιθάρας, κύριο Γιάννη Δημητρίου, που με σύστησε στον κόσμο της κιθάρας και κατ' επέκταση συνέβαλε στην έμπνευση του συγκεκριμένου θέματος.

Επιπλέον, ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στους Ανδρέα Εξουζιδη και Αρχοντούλα Γκουτζιούλη για την πολύτιμη βοήθειά τους.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένειά μου και στους φίλους μου για όλη τη στήριξη, τη συμπαράσταση και την κατανόησή τους, καθ' όλη την διάρκεια των σπουδών μου.







*How do I play guitar in three words?  
Practice, practice, practice.*

–Michael Angelo Batio

## Table of Contents

ΠΕΡΙΛΗΨΗ .....	I
ABSTRACT .....	III
ΕΥΧΑΡΙΣΤΙΕΣ .....	V
<b><u>ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ .....</u></b>	<b><u>3</u></b>
<b><u>ΚΕΦΑΛΑΙΟ 2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ .....</u></b>	<b><u>4</u></b>
<b><u>ΚΕΦΑΛΑΙΟ 3 ΣΚΟΠΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ &amp; ΜΕΘΟΔΟΛΟΓΙΑ.....</u></b>	<b><u>6</u></b>
3.1 ΣΚΟΠΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	6
3.2 ΜΕΘΟΔΟΛΟΓΙΑ .....	6
<b><u>ΚΕΦΑΛΑΙΟ 4 ΠΕΡΙΓΡΑΦΗ &amp; ΥΛΟΠΟΙΗΣΗ ΤΩΝ ΛΕΙΤΟΥΡΓΙΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....</u></b>	<b><u>7</u></b>
4.1 Η ΛΕΙΤΟΥΡΓΙΑ «AR CHORDS» .....	7
4.1.Α ΣΚΟΠΟΣ ΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ.....	7
4.1.Β ΔΗΜΙΟΥΡΓΙΑ 3D ΜΟΝΤΕΛΩΝ & ΔΙΑΓΡΑΜΜΑΤΩΝ .....	9
4.1.Γ ΔΗΜΙΟΥΡΓΙΑ LEVEL & GAME FRAMEWORK.....	14
4.1.Δ ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΠΑΥΣΗΜΕΝΩΝ ΕΙΚΟΝΩΝ & AR SESSION CONFIG .....	16
4.1.Ε ΔΗΜΙΟΥΡΓΙΑ WIDGET ΓΙΑ ΤΗΝ ΛΕΙΤΟΥΡΓΙΑ «AR CHORDS» .....	18
4.1.ΣΤ ΕΜΦΑΝΙΣΗ WIDGET & ΈΝΑΡΞΗ AR ΣΥΝΕΔΡΙΑΣ.....	19
4.1.Ζ ΑΝΑΓΝΩΡΙΣΗ ΕΙΚΟΝΑΣ, ΕΜΦΑΝΙΣΗ 3D ΜΟΝΤΕΛΟΥ & ΤΟΠΟΘΕΤΗΣΗ 3D ΜΟΝΤΕΛΟΥ ΣΤΗΝ ΦΥΣΙΚΗ ΕΙΚΟΝΑ .....	19
4.2 Η ΛΕΙΤΟΥΡΓΙΑ «EXERCISES» .....	25
4.2.Α ΣΚΟΠΟΣ ΤΗΣ ΛΕΙΤΟΥΡΓΙΑΣ.....	25
4.2.Β ΒΙΝΤΕΟΣΚΟΠΗΣΗ ΑΣΚΗΣΕΩΝ & ΣΥΓΓΡΑΦΗ ΕΚΦΩΝΗΣΕΩΝ .....	25
4.2.Γ ΕΙΣΑΓΩΓΗ ΑΡΧΕΙΩΝ ΒΙΝΤΕΟ & ΕΙΚΟΝΑΣ ΣΤΟ UE4 .....	26
4.2.Δ ΔΗΜΙΟΥΡΓΙΑ FILE MEDIA SOURCE & PLATFORM MEDIA SOURCE .....	27
4.2.Ε ΔΗΜΙΟΥΡΓΙΑ MEDIA PLAYER & MATERIAL .....	28
4.2.ΣΤ ΔΗΜΙΟΥΡΓΙΑ WIDGET ΓΙΑ ΚΑΘΕ ΆΣΚΗΣΗ .....	29
4.2.Ζ ΔΗΜΙΟΥΡΓΙΑ WIDGET ΓΙΑ ΕΠΙΛΟΓΗ ΜΕΤΑΞΥ LIST MODE/AR MODE.....	31
4.2.Η ΔΗΜΙΟΥΡΓΙΑ WIDGET ΓΙΑ ΤΟ LIST MODE .....	33
4.2.Θ ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΕΠΑΥΣΗΜΕΝΩΝ ΕΙΚΟΝΩΝ & AR SESSION CONFIG ΓΙΑ ΤΟ AR MODE .....	34
4.2.Ι ΔΗΜΙΟΥΡΓΙΑ WIDGET ΓΙΑ ΤΟ AR MODE .....	36
4.2.ΙΑ ΔΗΜΙΟΥΡΓΙΑ LEVEL ΓΙΑ ΤΟ AR MODE.....	37
4.3 Η ΛΕΙΤΟΥΡΓΙΑ «TUNER» .....	39
4.3.Α ΤΙ ΕΙΝΑΙ ΤΟ ΚΟΥΡΑΙΣΤΗΡΙ ΚΑΙ ΠΟΙΟΣ Ο ΣΚΟΠΟΣ ΤΟΥ.....	39
4.3.Β ΕΥΡΕΣΗ ΗΧΩΝ ΝΟΤΩΝ, ΕΙΣΑΓΩΓΗ ΣΤΟ UE4 & ΔΗΜΙΟΥΡΓΙΑ SOUND CUE.....	40
4.3.Γ ΔΗΜΙΟΥΡΓΙΑ WIDGET ΓΙΑ ΤΟ «TUNER» .....	40
4.3.Δ EVENT GRAPH ΤΟΥ «TUNER» .....	42
4.4 Η ΛΕΙΤΟΥΡΓΙΑ «METRONOME» .....	46
4.4.Α ΤΙ ΕΙΝΑΙ Ο ΜΕΤΡΟΝΟΜΟΣ ΚΑΙ ΠΟΙΟΣ Ο ΣΚΟΠΟΣ ΤΟΥ.....	46
4.4.Β ΕΥΡΕΣΗ ΗΧΩΝ ΧΤΥΠΩΝ, ΕΙΣΑΓΩΓΗ ΣΤΟ UE4 & ΔΗΜΙΟΥΡΓΙΑ SOUND CUE.....	46

4.4.Γ ΔΗΜΙΟΥΡΓΙΑ WIDGET ΓΙΑ ΤΟ «METRONOME» .....	47
4.4.Δ EVENT GRAPH ΤΟΥ «METRONOME» .....	49
<b><u>ΚΕΦΑΛΑΙΟ 5 ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ.....</u></b>	<b>56</b>
<b>5.1 ΥΛΟΠΟΙΗΣΗ ΤΟΥ ΒΑΣΙΚΟΥ ΜΕΝΟΥ (MAIN MENU) .....</b>	<b>56</b>
5.1.Α ΔΗΜΙΟΥΡΓΙΑ LEVEL ΓΙΑ ΤΟ MAIN MENU .....	56
5.1.Β ΔΗΜΙΟΥΡΓΙΑ WIDGET ΓΙΑ ΤΟ MAIN MENU .....	57
5.1.Γ EVENT GRAPH ΤΟΥ MAIN MENU .....	58
<b>5.2 ΠΛΟΗΓΗΣΗ ΣΤΗ ΔΙΕΠΑΦΗ ΧΡΗΣΤΗ .....</b>	<b>60</b>
<b><u>ΚΕΦΑΛΑΙΟ 6 ΑΠΑΡΑΙΤΗΤΕΣ ΡΥΘΜΙΣΕΙΣ, ΤΕΧΝΙΚΑ ΘΕΜΑΤΑ &amp; ΠΕΡΙΟΡΙΣΜΟΙ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....</u></b>	<b>65</b>
<b>6.1 ΡΥΘΜΙΣΕΙΣ ΣΤΟ PROJECT SETTINGS ΤΟΥ UE4 .....</b>	<b>65</b>
<b>6.2 ΣΥΝΔΕΣΗ ΚΙΝΗΤΗΣ ΣΥΣΚΕΥΗΣ ΜΕ ΤΟ UE4 .....</b>	<b>67</b>
<b>6.3 ΠΕΡΙΟΡΙΣΜΟΙ .....</b>	<b>68</b>
6.3.Α ΠΕΡΙΟΡΙΣΜΟΙ ΠΡΟΓΡΑΜΜΑΤΙΣΤΗ .....	68
6.3.Β ΠΕΡΙΟΡΙΣΜΟΙ ΧΡΗΣΤΗ .....	70
<b><u>ΚΕΦΑΛΑΙΟ 7 ΣΥΜΠΕΡΑΣΜΑΤΑ.....</u></b>	<b>71</b>
<b><u>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</u></b>	<b>72</b>

## ΚΕΦΑΛΑΙΟ 1 Εισαγωγή

---

Η ραγδαία ανάπτυξη του υλικού των Η/Υ και των κινητών συσκευών τις τελευταίες δεκαετίες επέτρεψε σε τεχνολογίες Επαυξημένης Πραγματικότητας (AR) να κάνουν την εμφάνισή τους και να εμπλακούν στην καθημερινότητά μας σε διάφορους τομείς, όπως τα βιντεοπαιχνίδια, τα μέσα κοινωνικής δικτύωσης, τη διασκέδαση, αλλά και σε επιστήμες, όπως η εκπαίδευση, η ιατρική και το μάρκετινγκ.

Όσον αφορά την εκπαίδευση, η ικανότητα αυτών των τεχνολογιών να εμπλουτίζουν το περιβάλλον του πραγματικού κόσμου με πρόσθετο ψηφιακό υλικό καταφέρνει να μεταδώσει στους μαθητές επιπλέον πληροφορία, δίνοντάς τους την δυνατότητα να οπτικοποιήσουν έννοιες και να παρατηρήσουν τρισδιάστατο υλικό από πολλές οπτικές γωνίες. Ως αποτέλεσμα, καταφέρνουν να παρέχουν στους μαθητές επιπλέον κίνητρο για μάθηση. Γι' αυτόν τον λόγο, δημιουργείται η ανάγκη για μεγαλύτερη εμπλοκή των AR τεχνολογιών στον τομέα αυτό, ειδικά στις μέρες μας όπου η εξ' αποστάσεως εκπαίδευση έχει επικρατήσει παγκοσμίως λόγω της πανδημίας.

Η παρούσα πτυχιακή εργασία προσπαθεί να συμβάλει σε αυτήν την ανάγκη και συγκεκριμένα αποσκοπεί στο να διευκολύνει την εκμάθηση κιθάρας με χρήση μιας εφαρμογής Επαυξημένης Πραγματικότητας. Η κιθάρα αποτελεί ένα από τα πιο διαδεδομένα και δημοφιλή μουσικά όργανα, με συμμετοχή σε πολλά είδη μουσικής παγκοσμίως. Μεγάλο ποσοστό αυτών που επιλέγουν την εκμάθησή της, προτιμούν να είναι αυτοδίδακτοι, κάνοντας χρήση συγγραμμάτων εκμάθησης κιθάρας και διαθέσιμου υλικού που υπάρχει στο διαδίκτυο. Αυτή η μέθοδος δημιουργεί ελλείψεις στα σημεία όπου απαιτείται παρατήρηση τρισδιάστατων στοιχείων, κάτι που αποτελεί πρόβλημα τόσο για τους αυτοδίδακτους κιθαρίστες, όσο και για αυτούς που πρέπει να συνεχίσουν τη μελέτη τους στο σπίτι μετά από ένα μάθημα με κάποιον δάσκαλο.

Στην παρούσα πτυχιακή εργασία παρουσιάζεται η εφαρμογή Επαυξημένης Πραγματικότητας που δημιουργήθηκε, η οποία αναπτύχθηκε στο πρόγραμμα Unreal Engine 4. Συνοπτικά, στο Κεφάλαιο 2 γίνεται επισκόπηση της υπάρχουσας βιβλιογραφίας με αναφορές σε επιστημονικά άρθρα συναφή με το αντικείμενο της εργασίας. Στο Κεφάλαιο 3 εξηγείται ο σκοπός της εφαρμογής και η μεθοδολογία που ακολουθήθηκε για την ανάπτυξή της με αναφορά στα απαραίτητα εργαλεία. Στο Κεφάλαιο 4 και στο Κεφάλαιο 5 γίνεται αναλυτική περιγραφή της υλοποίησης των λειτουργιών της εφαρμογής και της διεπαφής χρήστη αντίστοιχα. Στο κεφάλαιο 6 αναφέρονται τεχνικά θέματα και περιορισμοί της εφαρμογής. Τέλος, στο Κεφάλαιο 7 παρουσιάζονται κάποια σχετικά συμπεράσματα.

## ΚΕΦΑΛΑΙΟ 2 Βιβλιογραφική Επισκόπηση

Υπάρχει πληθώρα δημοσιευμένων εργασιών που αφορούν τη χρήση τεχνολογιών Επαυξημένης Πραγματικότητας γενικά στο χώρο της εκπαίδευσης και ειδικά στο χώρο της διδασκαλίας της μουσικής. Η αλίευση εργασιών και άρθρων έγινε σε μηχανές αναζήτησης σε βάσεις δεδομένων επιστημονικών άρθρων κάνοντας χρήση λέξεων-κλειδιών με τις λογικές εκφράσεις “augmented AND reality AND education”, “augmented AND reality AND education AND music” και “augmented AND reality AND guitar”.

Το πως μπορεί να χρησιμοποιηθεί η τεχνολογία της Επαυξημένης Πραγματικότητας για εκπαιδευτικούς σκοπούς έχει απαντηθεί από τους Klopfer και Squire (2008), οι οποίοι υποστήριζαν πως δίνει στους μαθητές την δυνατότητα να παρατηρήσουν επιστημονικά φαινόμενα τα οποία είναι αδύνατο να παρατηρήσουν στην καθημερινότητα τους, όπως για παράδειγμα οι χημικές αντιδράσεις. Άλλα παραδείγματα αναφέρουν οι Liu κ.α. (2007), όπως η παρατήρηση ενός ψηφιακού ηλιακού συστήματος και της διαδικασίας της φωτοσύνθεσης. Επίσης, οι μαθητές σύμφωνα με τους Kerawalla κ.α. (2006) έχουν την δυνατότητα να παρατηρήσουν το ψηφιακό υλικό από διάφορες οπτικές γωνίες. Στη μελέτη που πραγματοποίησαν, διεξήγαγαν το ίδιο μάθημα αστρονομίας, που αφορούσε την κατανόηση της περιστρεφόμενης Γης σε σχέση με τον Ήλιο και κατ' επέκταση τις εναλλαγές μέρας/νύχτας, σε δύο τάξεις δημοτικού, όπου στην πρώτη χρησιμοποιήθηκε τεχνολογία Επαυξημένης Πραγματικότητας, με εργαλεία όπως προτζέκτορας και πακέτο 3D μοντέλων, ενώ στην δεύτερη το μάθημα διεξήχθη με τον παραδοσιακό τρόπο με χρήση κειμένου και φυσικών αντικειμένων (π.χ. μπάλα τένις, φακός). Τα αποτελέσματα έδειξαν ότι υπήρξαν οφέλη από την χρήση 3D μοντέλων, αφού έγιναν στους μαθητές διαθέσιμες πληροφορίες, στις οποίες δεν είχαν πρόσβαση με τον παραδοσιακό τρόπο διδασκαλίας.

Εκτός από τον τρόπο ένταξης τεχνολογιών Επαυξημένης Πραγματικότητας στην εκπαίδευση, στη διεθνή βιβλιογραφία παρατηρείται ιδιαίτερο ενδιαφέρον όσον αφορά το κίνητρο για μάθηση και διδασκαλία που δημιουργείται σε μαθητές και εκπαιδευτικούς αντίστοιχα, όταν γίνεται χρήση των τεχνολογιών αυτών. Οι Nor Farhah Saidin κ.α. (2015) υποστηρίζουν ότι, παρόλο που σε προγενέστερες έρευνες υπήρξε ο φόβος πως η ένταξη νέων τεχνολογιών στην εκπαίδευση θα την επηρέαζε αρνητικά οδηγώντας σε παθητική μάθηση και αδυναμία ανάπτυξης κριτικής σκέψης, ο συνδυασμός της Επαυξημένης Πραγματικότητας με την εκπαίδευση απέδειξε από την αρχή της εφαρμογής του ότι δύναται να αυξήσει την ενεργητική μάθηση και να δώσει κίνητρα στους μαθητές, κάνοντας έτσι την μάθηση πιο αποτελεσματική. Αυτό συμβαίνει γιατί επιτρέπει στους μαθητές να αποκτούν ρεαλιστικές εμπειρίες. Επίσης, οι ίδιοι αναφέρουν ότι «*η Επαυξημένη Πραγματικότητα μπορεί να ελαχιστοποιήσει τις παρανοήσεις που δημιουργούνται λόγω της αδυναμίας των μαθητών να οπτικοποιούν έννοιες, επειδή η Επαυξημένη Πραγματικότητα επιτρέπει λεπτομερή οπτικοποίηση και δημιουργία κινούμενων αντικειμένων.[...] Η Επαυξημένη Πραγματικότητα απεικονίζει αντικείμενα και έννοιες με διαφορετικούς τρόπους και από διαφορετικές οπτικές γωνίες, βοηθώντας έτσι τους μαθητές στην καλύτερη κατανόηση των θεμάτων.*»(σελ. 3).

Οι Deepti Prit, Archana Mantri & Ben Horan (2019) πραγματοποίησαν μία μελέτη σχετικά με τα κίνητρα μάθησης που προσφέρουν οι τεχνολογίες Επαυξημένης Πραγματικότητας στην εκπαίδευση. Συγκεκριμένα, εφάρμοσαν ένα σύστημα μάθησης με χρήση AR τεχνολογίας σε προπτυχιακούς φοιτητές τμήματος Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών, το οποίο τους βοήθησε να οπτικοποιήσουν θεωρητικές έννοιες. Μετά το πέρας των διαλέξεων ζητήθηκε από τους φοιτητές να βαθμολογήσουν με κλίμακα 0-5 το κατά πόσο συμφωνούν με κάποιες προτάσεις. Η συμφωνία με την πρόταση δήλωνε ότι υπήρξε στη διάλεξη κάποιος παράγοντας κινήτρου, ενώ η διαφωνία απουσία κάποιου παράγοντα. Στο σύνολο των προτάσεων ο μέσος όρος βαθμολογίας ήταν πάνω από

4 (σε κλίμακα 0-5). Επομένως, κατέληξαν στο συμπέρασμα πως η χρήση τεχνολογιών Επαυξημένης Πραγματικότητας για την απεικόνιση δυσνόητων εννοιών προσφέρει κίνητρα μάθησης στους φοιτητές.

Στον τομέα της διδασκαλίας της μουσικής οι Ana G.D. Correa κ.α. (2016), οι οποίοι δημιούργησαν μία εφαρμογή Επαυξημένης Πραγματικότητας για την εκμάθηση του σωστού σχεδιασμού των νοτών στο μουσικό πεντάγραμμο για παιδιά, μετά από δοκιμή κατέληξαν στο συμπέρασμα ότι η μάθηση με την βοήθεια της εν λόγω εφαρμογής ήταν αποτελεσματική, με τους μαθητές να δείχνουν ενθουσιασμό και κίνητρο για μάθηση. Μάλιστα, αναφέρουν περιστατικό στο οποίο μαθήτρια η οποία δεν σχεδίασε σωστά τις νότες στο μουσικό πεντάγραμμο εντόπισε τα λάθη της, με την βοήθεια της εφαρμογής, και προέβη στην διόρθωσή τους χωρίς να παρέμβει κάποιος από τους δασκάλους. Οι Carlos A Torres Fernandez κ.α. (2016) επίσης υποστηρίζουν τη δημιουργία κινήτρου για μάθηση της μουσικής, μέσω της εφαρμογής Επαυξημένης Πραγματικότητας που δημιούργησαν για την σωστή εκτέλεση μουσικών συνθέσεων σε πληκτροφόρα μουσικά όργανα. Οι Valeria Ferinazzo Martins κ.α. (2015), που εστίασαν στον τομέα της μουσικής προπαιδείας (αναγνώριση χροιάς, έντασης, διάρκειας, τόνου), δημιούργησαν την εφαρμογή Επαυξημένης Πραγματικότητας “Music-AR” και μετά από δοκιμή σε περιβάλλον τάξης συμπέραναν πως οι μαθητές εξοικειώθηκαν εύκολα με την χρήση της εφαρμογής, η οποία βοήθησε στην κατανόηση των εννοιών του μαθήματος και παρείχε κίνητρο. Με τον ίδιο τομέα ασχολήθηκαν και οι Garyfallia Preka και Maria Rangoussi (2019), δημιουργώντας μία εφαρμογή Επαυξημένης Πραγματικότητας σε μορφή παιχνιδιού με στόχο την διδασκαλία μουσικής προπαιδείας σε παιδιά προσχολικής ηλικίας. Διεξήγαγαν τεστ σε 20 παιδιά πριν και μετά την ενασχόλησή τους με το παιχνίδι και τα αποτελέσματα έδειξαν σημαντική αύξηση των σωστών απαντήσεων.

Στον τομέα της εκμάθησης κιθάρας έχουν γίνει επίσης προσπάθειες ένταξης των τεχνολογιών Επαυξημένης Πραγματικότητας. Όσες εφαρμογές βρέθηκαν, έχουν παρόμοια λειτουργία και στοχοθεσία, με κοινό χαρακτηριστικό την εμφάνιση πληροφοριών και οδηγιών επάνω στην ταστιέρα της κιθάρας, όμως η κάθε μία έχει διαφορετική υλοποίηση. Οι Joseph R. Keebler κ.α. (2013) δημιούργησαν ένα σύστημα εκμάθησης κιθάρας, στο οποίο η κιθάρα διαθέτει φώτα LED σε κάθε πιθανό σημείο της ταστιέρας που μπορεί να γίνει τοποθέτηση δακτύλου και με την ενεργοποίηση συγκεκριμένων LED παρέχονται οδηγίες για την σωστή τοποθέτηση των δακτύλων. Αυτό το σύστημα απαιτεί ειδικά κατασκευασμένη κιθάρα με ενσωματωμένα φώτα LED ή τοποθέτηση ειδικού εξοπλισμού σε μία κοινή κιθάρα. Παρόμοιο σύστημα δημιούργησαν οι Markus Lochtefeld κ.α. (2011) με την διαφορά ότι, αντί για LED χρησιμοποιείται συσκευή η οποία φωτίζει με ακτίνες λέιζερ τα επιθυμητά σημεία της ταστιέρας. Οι Yoichi Motokawa κ.α. (2006), καθώς και οι Jorge Martin-Gutierrez κ.α. (2020) πρότειναν μία διαφορετική υλοποίηση, στην οποία η προβολή πληροφοριών και οδηγιών δεν γίνεται επάνω στην πραγματική ταστιέρα αλλά ψηφιακά. Πιο συγκεκριμένα, πραγματοποιείται καταγραφή και προβολή της εικόνας της κιθάρας σε πραγματικό χρόνο σε οθόνη που βρίσκεται απέναντι από τον χρήστη. Δημιουργείται, δηλαδή, ένα ψηφιακό είδωλο της κιθάρας και του χρήστη, με τις οδηγίες για την σωστή τοποθέτηση των δακτύλων να εμφανίζονται επάνω στο ψηφιακό είδωλο της ταστιέρας.

Στο ίδιο πλαίσιο με τις παραπάνω εφαρμογές κινείται και η εφαρμογή της παρούσης εργασίας, που αποσκοπεί, με το δικό της τρόπο, να διευκολύνει την εκμάθηση κιθάρας, λειτουργώντας συνοδευτικά με τα κλασικά εκπαιδευτικά εγχειρίδια.

# ΚΕΦΑΛΑΙΟ 3 Σκοπός της Εφαρμογής & Μεθοδολογία

---

## 3.1 Σκοπός της Εφαρμογής

---

Σκοπός της εφαρμογής είναι να συνοδεύει, σαν ένθετο, ένα σύγγραμμα εκμάθησης κιθάρας και να παρέχει στον αναγνώστη πληροφορίες και εργαλεία που αδυνατούν να αποτυπωθούν με έντυπο τρόπο. Στη διαδικασία εκμάθησης κιθάρας πολλές φορές απαιτείται τρισδιάστατη (3D) απεικόνιση πληροφορίας κυρίως όταν είναι αναγκαίο να παρουσιαστεί μία δακτυλοθεσία, δηλαδή η ακριβής τοποθέτηση των δακτύλων επάνω στην κιθάρα. Ωστόσο, τα συγγράμματα εκμάθησης κιθάρας περιέχουν μέσα (κείμενο, εικόνα, ήχο σε ένθετο Οπτικό Δίσκο (CD)), τα οποία δεν καταφέρνουν να μεταφέρουν στον αναγνώστη την απαιτούμενη πληροφορία για την πλήρη κατανόηση των εννοιών. Επομένως, η εφαρμογή προσπαθεί να καλύψει αυτήν την αδυναμία μετάδοσης πληροφορίας, κάνοντας χρήση της Επαυξημένης Πραγματικότητας για την τρισδιάστατη απεικόνιση των δακτυλοθεσιών και παρουσιάζοντας βιντεοσκοπημένο υλικό στον αναγνώστη. Επίσης, παρέχει δύο σημαντικά εργαλεία, χρήσιμα κατά την εκμάθηση και την καθημερινή μελέτη, τα οποία είναι ένα ψηφιακό κουρδιστήρι και ένας ψηφιακός μετρονόμος.

## 3.2 Μεθοδολογία

---

Η εφαρμογή υλοποιήθηκε στο πρόγραμμα Unreal Engine 4.24 μέσω Οπτικής Γλώσσας Προγραμματισμού (Visual Programming Language) που ονομάζεται Blueprints. Οι συναρτήσεις που αφορούν την τεχνολογία Επαυξημένης Πραγματικότητας (AR) έγιναν διαθέσιμες στο πρόγραμμα μέσω της επέκτασης ARCore 1.7. Η Διεπαφή Χρήστη δημιουργήθηκε και αυτή στο πρόγραμμα Unreal Engine 4.24 με την χρήση των Widget Blueprints που παρέχει το πρόγραμμα. Όλα τα στοιχεία τα οποία εμφανίζονται (κουμπιά, Εικόνες, Επικεφαλίδες κτλ.) δημιουργήθηκαν στο πρόγραμμα Adobe Photoshop 21. Χρησιμοποιήθηκε, επίσης, το πρόγραμμα Blender 2.83 για την δημιουργία των τρισδιάστατων μοντέλων που χρησιμοποιήθηκαν για την αναπαράσταση των δακτυλοθεσιών. Τα βίντεο, τα οποία αναπαράγονται εντός της εφαρμογής, βιντεοσκοπήθηκαν με χρήση ψηφιακής βιντεοκάμερας. Η επεξεργασία τους και η μετατροπή τους σε κατάλληλο αρχείο έγινε μέσω του προγράμματος Adobe Premiere Pro 14. Τέλος, η συγγραφή μουσικού πενταγράμμου/ταμπλατούρας έγινε στο πρόγραμμα Guitar Pro 5.

# ΚΕΦΑΛΑΙΟ 4 Περιγραφή & Υλοποίηση των Λειτουργιών της Εφαρμογής

---

## 4.1 Η Λειτουργία «AR Chords»

---

### 4.1.α Σκοπός της Λειτουργίας

---

Στα παραδοσιακά μέσα εκμάθησης κιθάρας γίνεται προσπάθεια απεικόνισης της δακτυλοθεσίας μιας συγχορδίας με δύο τρόπους:

1. Με χρήση φωτογραφίας, στην οποία ο κιθαρίστας κάνει επίδειξη του τρόπου εκτέλεσης της δακτυλοθεσίας:

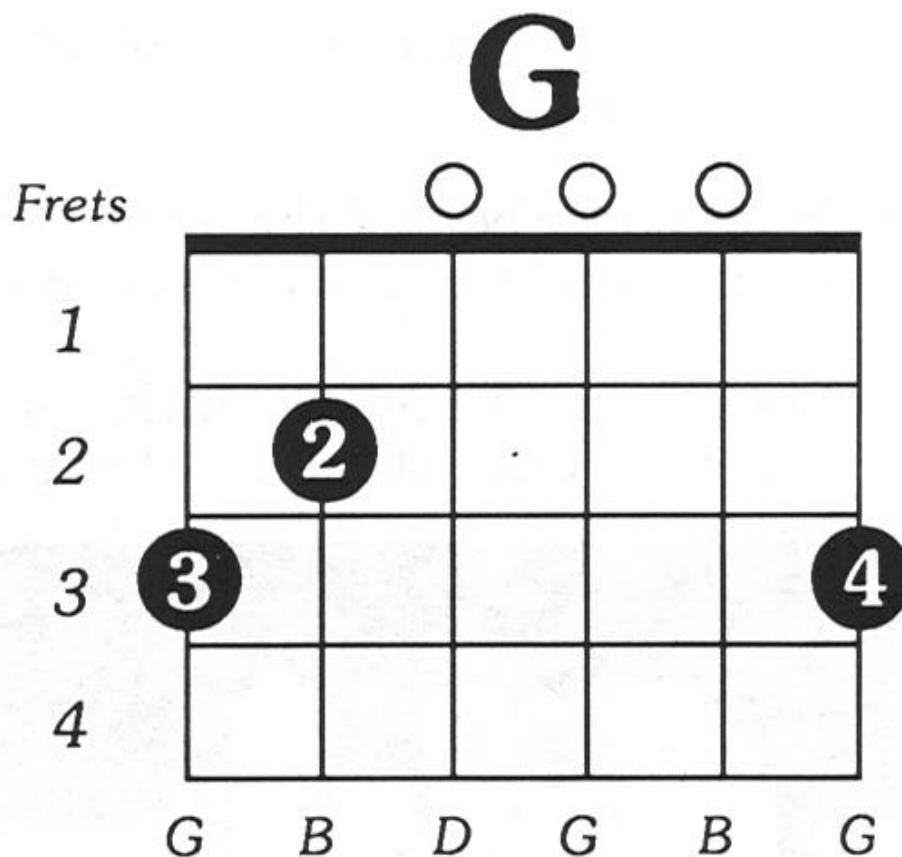
G major



Εικόνα 4.1: Παράδειγμα φωτογραφίας δακτυλοθεσίας

2. Με χρήση διαγράμματος, στο οποίο απαριθμούνται τα τάστα και οι χορδές της κιθάρας και με κύκλους υποδεικνύονται τα σημεία στα οποία θα τοποθετηθούν τα δάχτυλα. Ο κάθε κύκλος περιέχει έναν αριθμό ο οποίος αντιπροσωπεύει το κάθε δάχτυλο (Δείκτης = 1, Μέσος = 2, Παράμεσος = 3, Μικρός = 4):





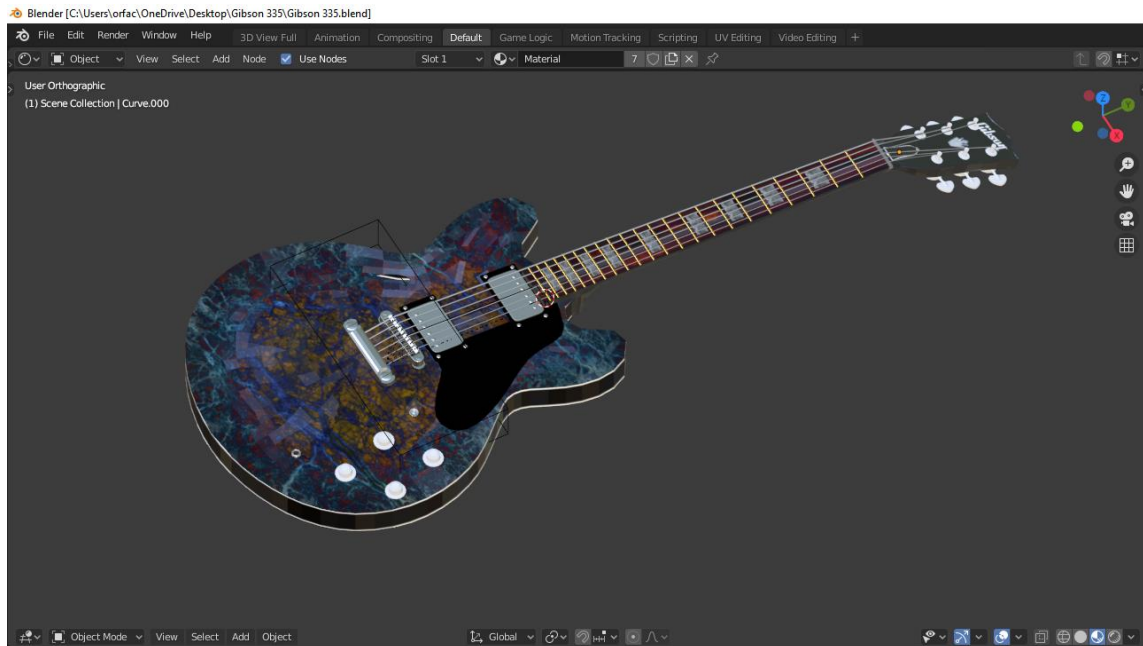
Εικόνα 4.2: Παράδειγμα διαγράμματος δακτυλοθεσίας

Και οι δύο τρόποι απεικόνισης δεν καταφέρνουν να μεταδώσουν στον μαθητή όλη την απαιτούμενη πληροφορία για τον τρόπο με τον οποίο θα πρέπει να γίνει η τοποθέτηση των δακτύλων. Αυτό είναι λογικό, διότι μια τρισδιάστατη πληροφορία είναι αδύνατο να απεικονιστεί σε διδιάστατο μέσο χωρίς ελλείψεις. Ο μαθητής έχει την ανάγκη να παρατηρήσει την δακτυλοθεσία από πολλές οπτικές γωνίες, ώστε να λάβει όλες τις απαιτούμενες πληροφορίες κάτι το οποίο δεν επιτυγχάνεται στα συμβατικά μέσα.

Η λειτουργία “AR Chords” επιχειρεί να καλύψει αυτό το κενό της διδιάστατης απεικόνισης, χρησιμοποιώντας 3D μοντέλα απεικόνισης της δακτυλοθεσίας, τα οποία τοποθετούνται πάνω στα παραδοσιακά διαγράμματα του συγγράμματος του μαθητή με τη χρήση της Επαυξημένης Πραγματικότητας. Πιο συγκεκριμένα, ενεργοποιείται η κάμερα της κινητής συσκευής του χρήστη και μόλις εμφανιστεί στο οπτικό πεδίο κάποιο διάγραμμα συγχορδίας του βιβλίου εμφανίζεται ένα 3D μοντέλο. Ο χρήστης, αλλάζοντας την κλίση και την απόσταση της κινητής συσκευής από την σελίδα του βιβλίου, μπορεί να επιλέξει διαφορετική οπτική γωνία θέασης της δακτυλοθεσίας. Με αυτόν τον τρόπο επιτυγχάνεται ολοκληρωμένη μετάδοση της πληροφορίας.

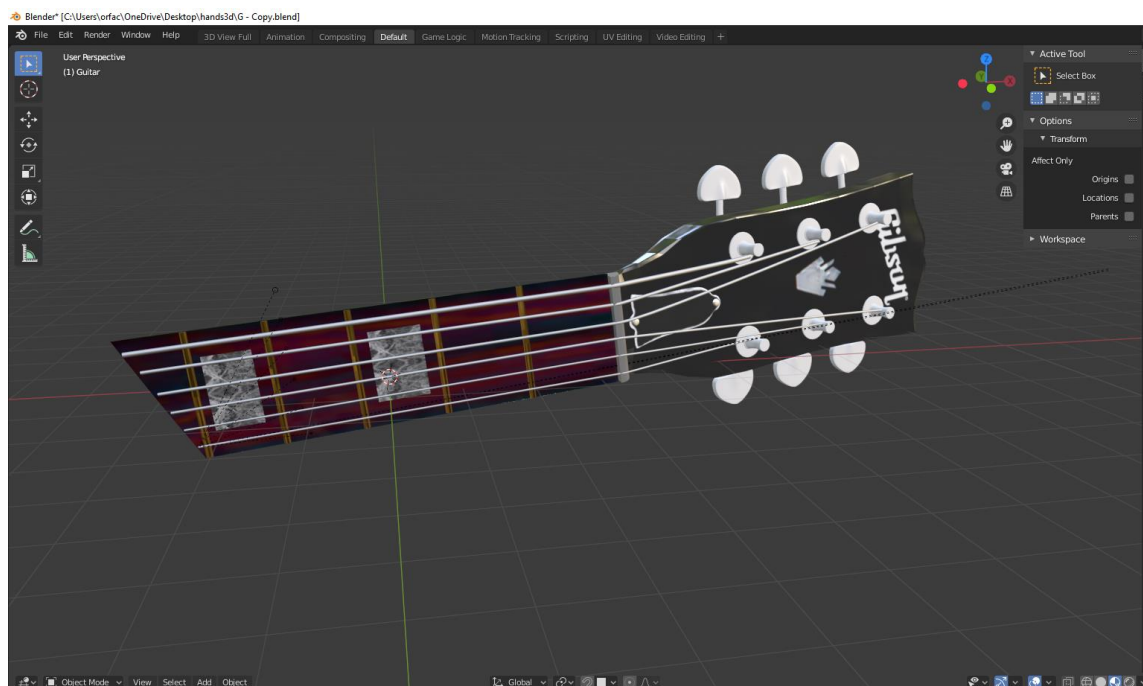
## 4.1.6 Δημιουργία 3D Μοντέλων & Διαγραμμάτων

Η δημιουργία των 3D μοντέλων έγινε με την χρήση του προγράμματος Blender. Αρχικά, βρέθηκε σε αποθετήριο 3D μοντέλων ένα μοντέλο κιθάρας<sup>1</sup>.



Εικόνα 4.3: 3D Μοντέλο Κιθάρας

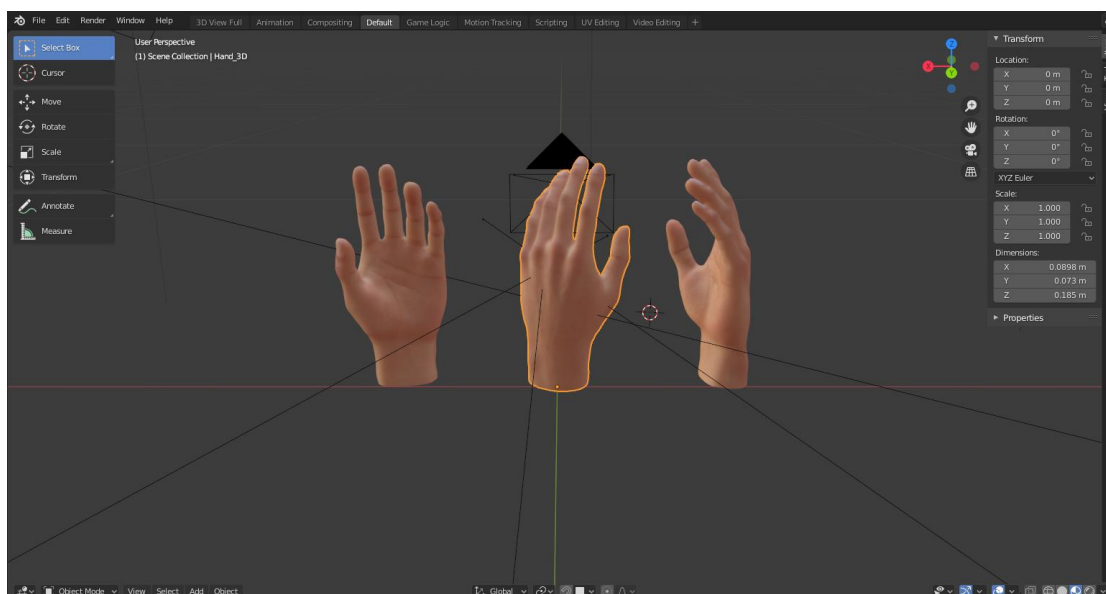
Έγινε διχοτόμηση του μοντέλου της κιθάρας και απομόνωση του μέρους που απαιτείται για την υλοποίηση της λειτουργίας. Πιο συγκεκριμένα απομονώθηκε το κομμάτι με τα πρώτα πέντε τάστα της κιθάρας, μιας και όλες οι δακτυλοθεσίες που θα απεικονιστούν εκτελούνται σε εκείνη τη περιοχή.



Εικόνα 4.4: 3D Μοντέλο Κεφαλής κιθάρας

<sup>1</sup> <https://free3d.com/3d-model/gibson-es-335-816888.html>

Στη συνέχεια, βρέθηκε σε αποθετήριο 3D μοντέλων ένα μοντέλο χεριού<sup>2</sup>.



Εικόνα 4.5: 3D Μοντέλο χεριού

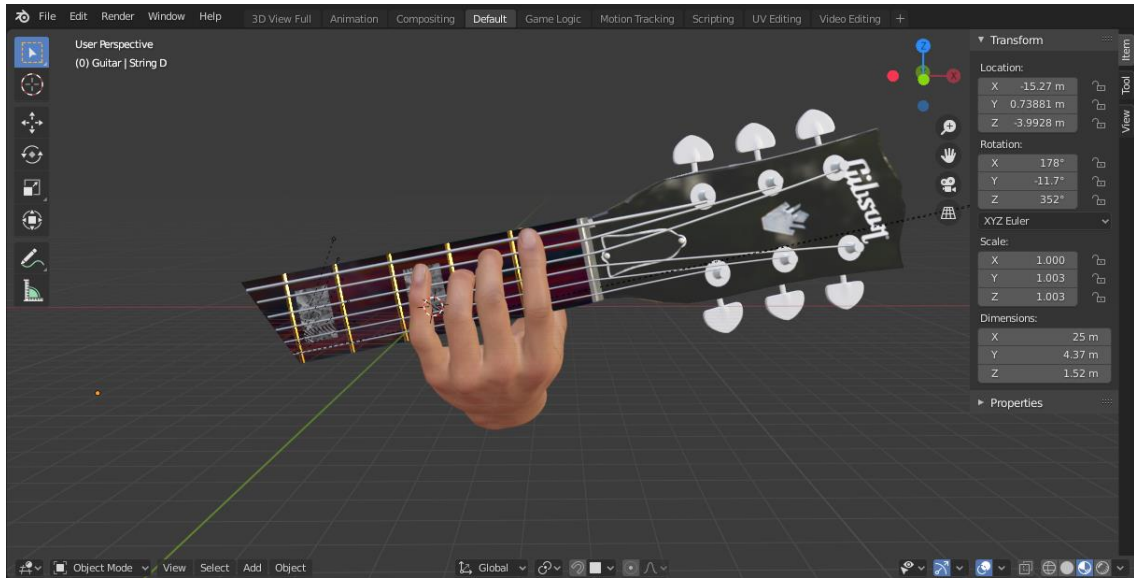
Για να μπορέσει το μοντέλο να πάρει τις κατάλληλες πόζες προστέθηκαν οστά (bones) εσωτερικά του χεριού.



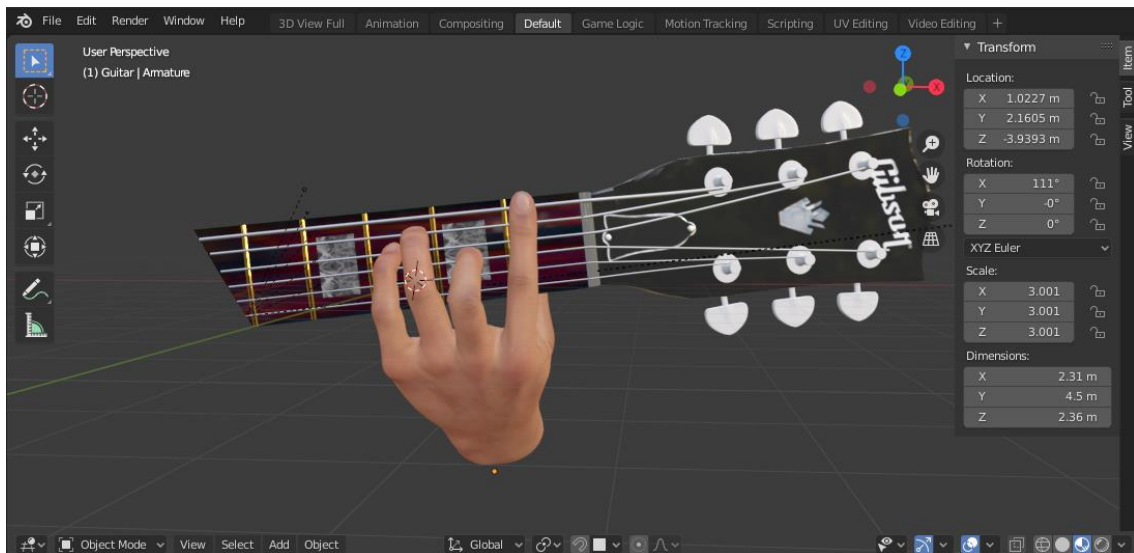
Εικόνα 4.6: 3D Μοντέλο χεριού με οστά

Στη συνέχεια, έγινε συνένωση των δύο μοντέλων (χεριού με οστά και κιθάρας) ώστε να βρίσκονται στην ίδια σκηνή. Με την κατάλληλη περιστροφή και αλλαγή θέσης των οστών παρήχθησαν οι απαιτούμενες πόζες.

<sup>2</sup> <https://blendswap.com/blend/16177>



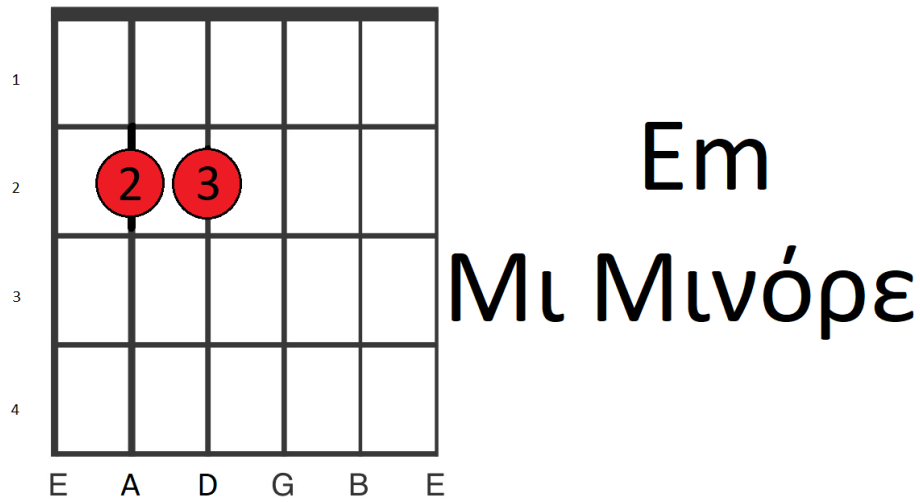
**Εικόνα 4.7: Πόζα Συγχορδίας Μι Μινόρε**



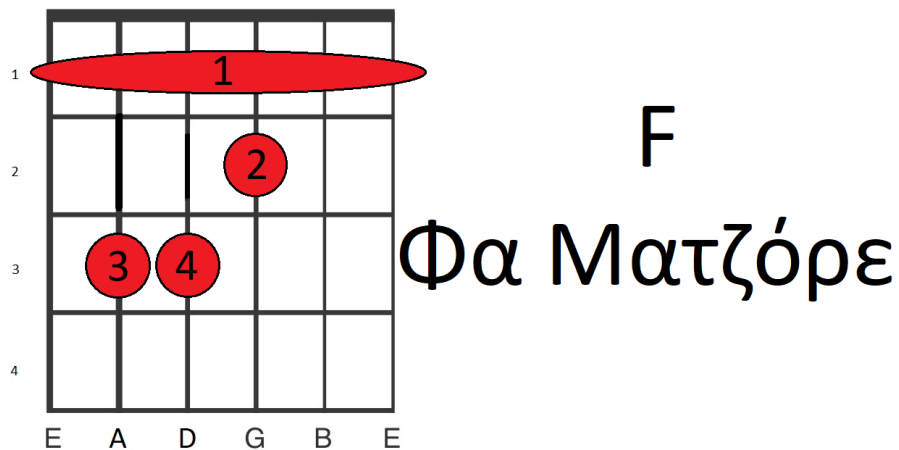
**Εικόνα 4.8: Πόζα Συγχορδίας Φα Ματζόρε**

Τέλος, έγινε εξαγωγή των μοντέλων σε αρχείο .OBJ ώστε να εισαχθούν στο πρόγραμμα Unreal Engine 4.

Τα διαγράμματα σχεδιάστηκαν στο πρόγραμμα Paint των Windows και εξήχθησαν σε αρχείο .png.

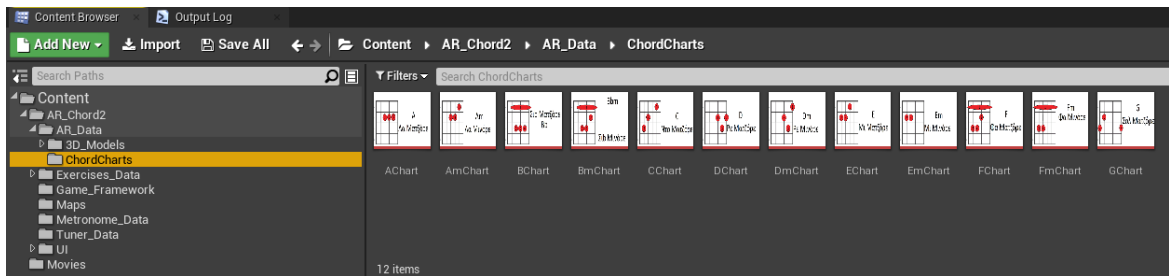


Εικόνα 4.9: Διάγραμμα Συγχορδίας Μι Μινόρε

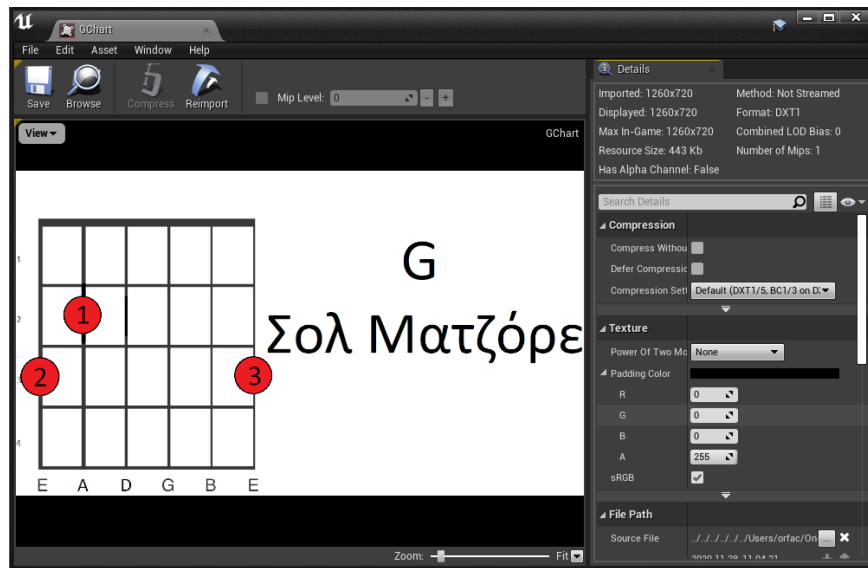


Εικόνα 4.10: Διάγραμμα Συγχορδίας Φα Ματζόρε

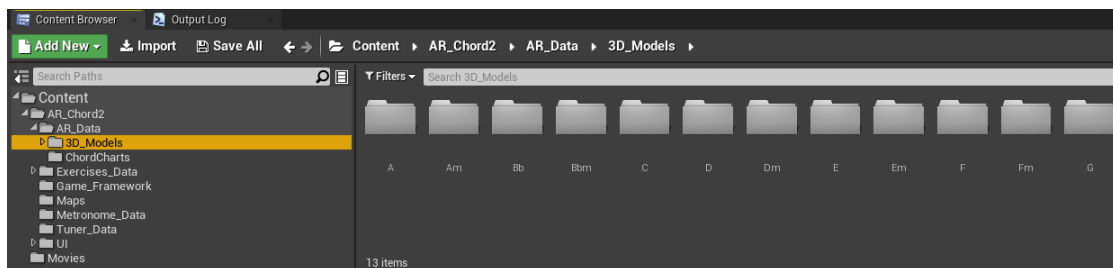
Τα διαγράμματα και τα 3D μοντέλα εισήχθησαν στο πρόγραμμα Unreal Engine 4 και δόθηκαν τα κατάλληλα ονόματα: “ΟνομαΣυγχορδίαςChart” στις εικόνες και “ΟνομαΣυγχορδίας\_Model” στα 3D Μοντέλα.



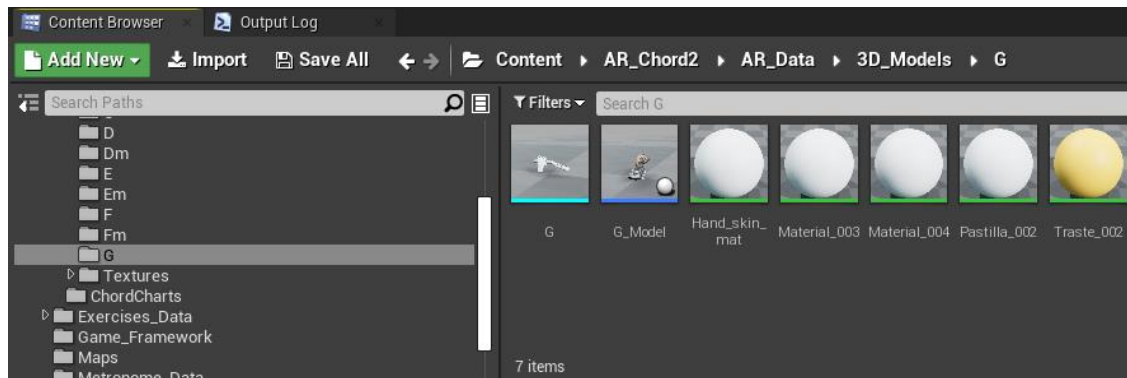
Εικόνα 4.11: Εικόνες Διαγραμμάτων στο Content Browser του UE4



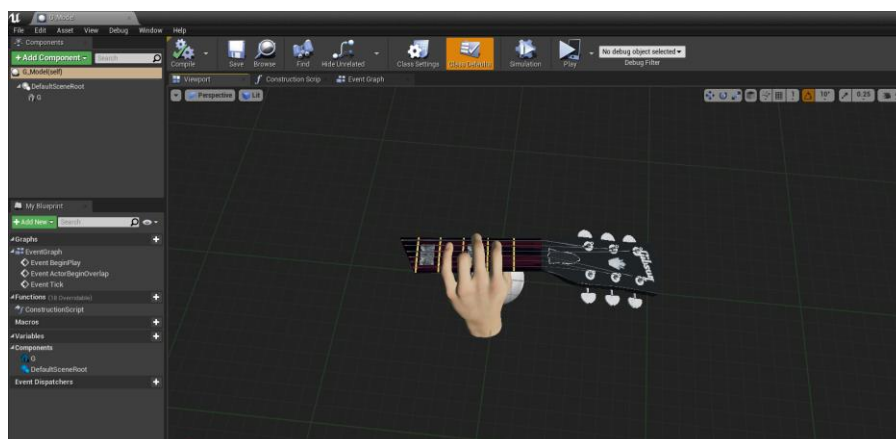
Εικόνα 4.12: Εικόνα Διαγράμματος Συγχορδίας Σολ Ματζόρε στο UE4



Εικόνα 4.13: 3D Μοντέλα στο Content Browser του UE4



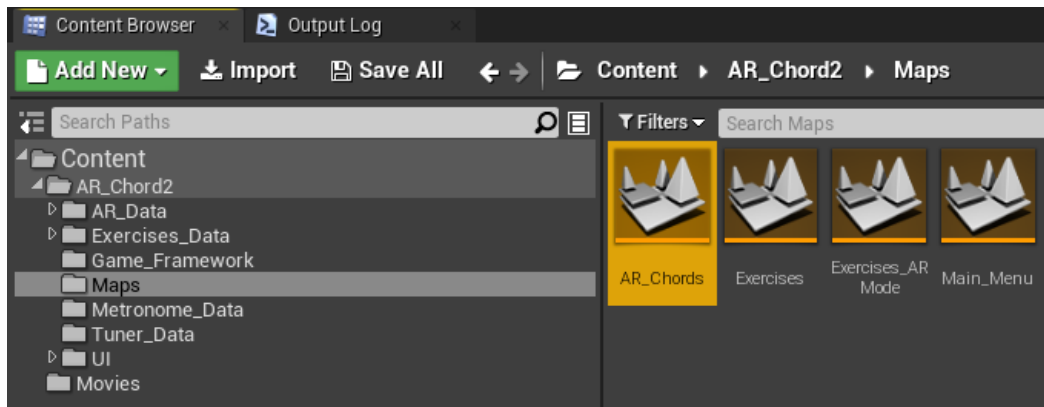
Εικόνα 4.14: Αρχεία 3D Μοντέλου Συγχορδίας Σολ Ματζόρε στο Content Browser του UE4



Εικόνα 4.15: 3D Μοντέλο Συγχορδίας Σολ Ματζόρε στο UE4

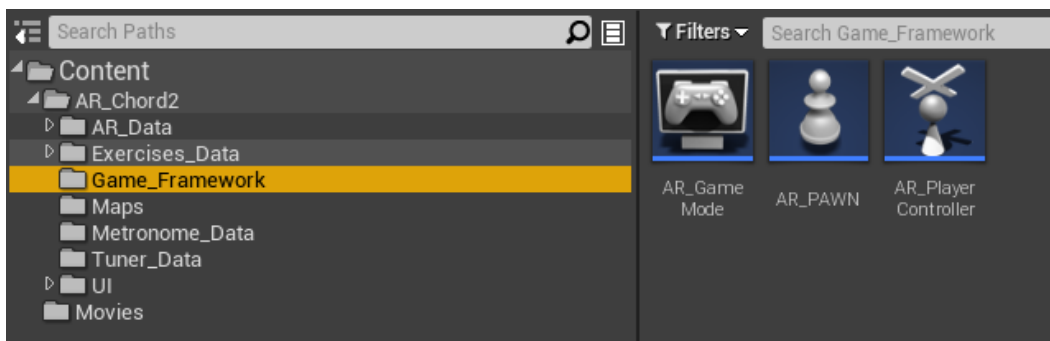
#### 4.1.γ Δημιουργία Level & Game Framework

Στο Project του προγράμματος Unreal Engine 4 δημιουργήθηκε νέο *Level* με όνομα “AR\_Chords”.



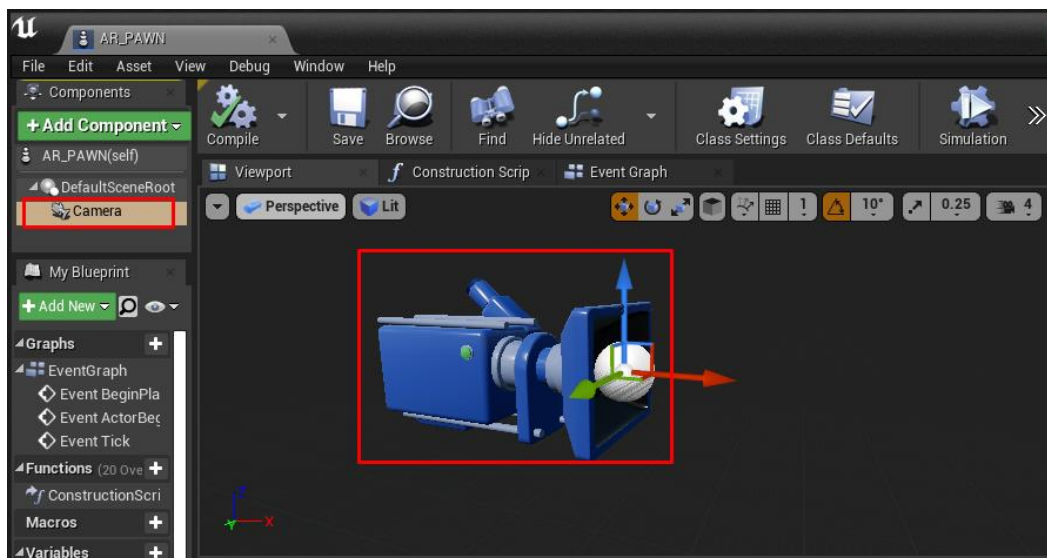
Εικόνα 4.16: Το Level "AR Chords" στο Content Browser του UE4

Δημιουργήθηκαν, επίσης, τα εξής *Blueprint Classes*: ένα *Game Mode Base* με όνομα “AR\_GameMode”, ένα *Pawn* με όνομα “AR\_Pawn” και ένα *Player Controller* με όνομα “AR\_PlayerController”.



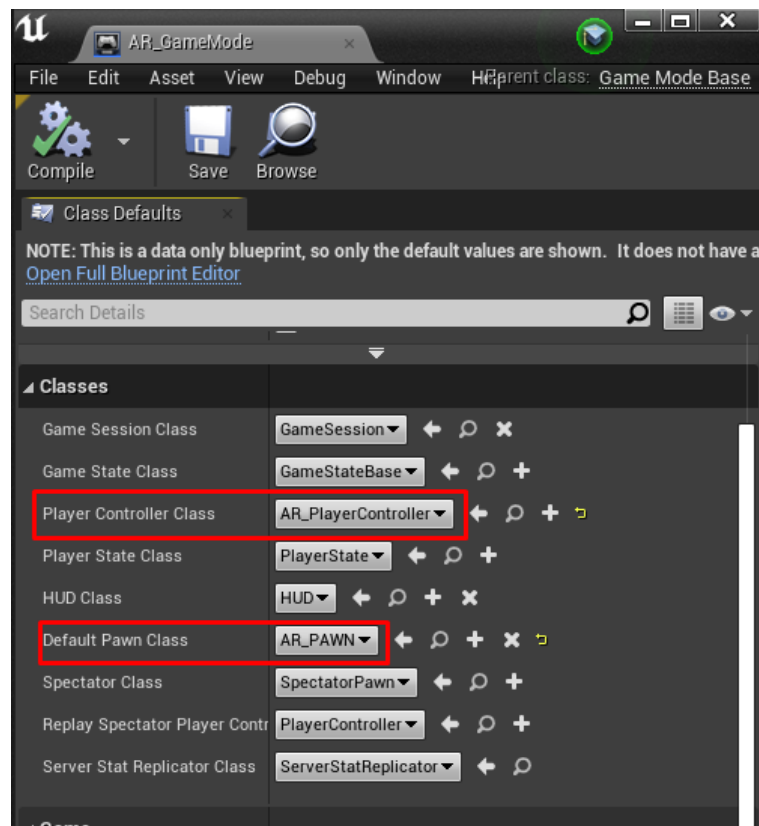
Εικόνα 4.17: Οι κλάσεις "AR\_GameMode", "AR\_PAWN" και "AR\_PlayerController" στο Content Browser του UE4

Στην κλάση “AR\_Pawn” εισήχθη το *component* “Camera”.



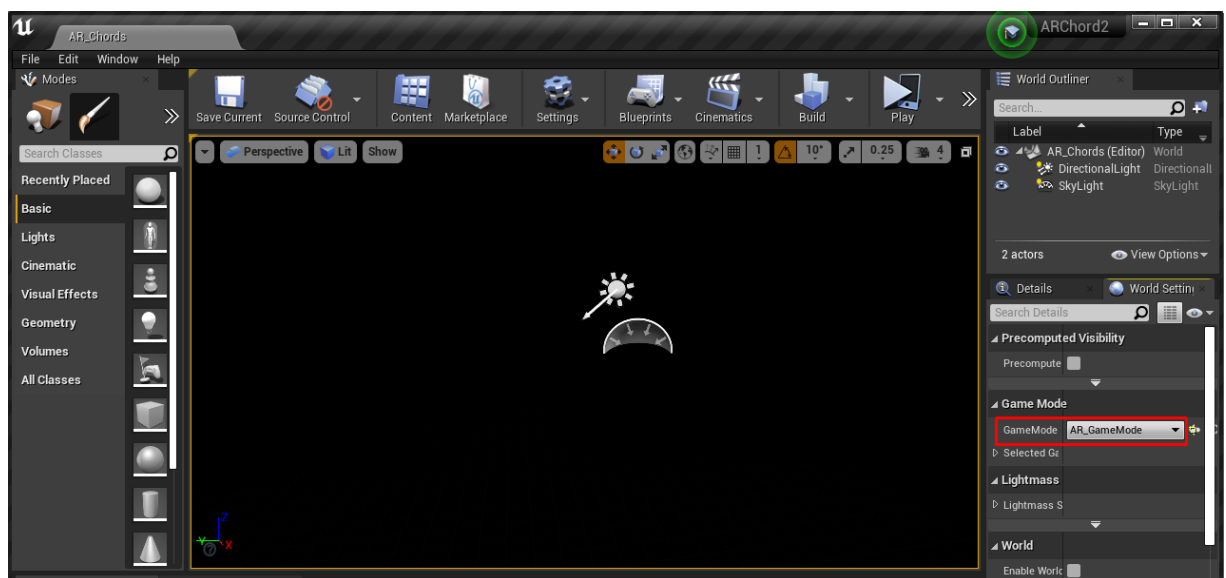
Εικόνα 4.18: Το Component Camera στην κλάση "AR\_PAWN"

Στην κλάση “AR\_GameMode” επιλέχθηκαν οι κλάσεις “AR\_PlayerController” και “AR\_Pawn” στα πεδία *Player Controller Class* και *Default Pawn Class* αντιστοίχα.



Εικόνα 4.19: Η κλάση "AR\_GameMode"

Τέλος, στα *World Settings* του *Level* “AR\_Chords” στο πεδίο *Game Mode* επιλέχτηκε η κλάση “AR\_GameMode”.

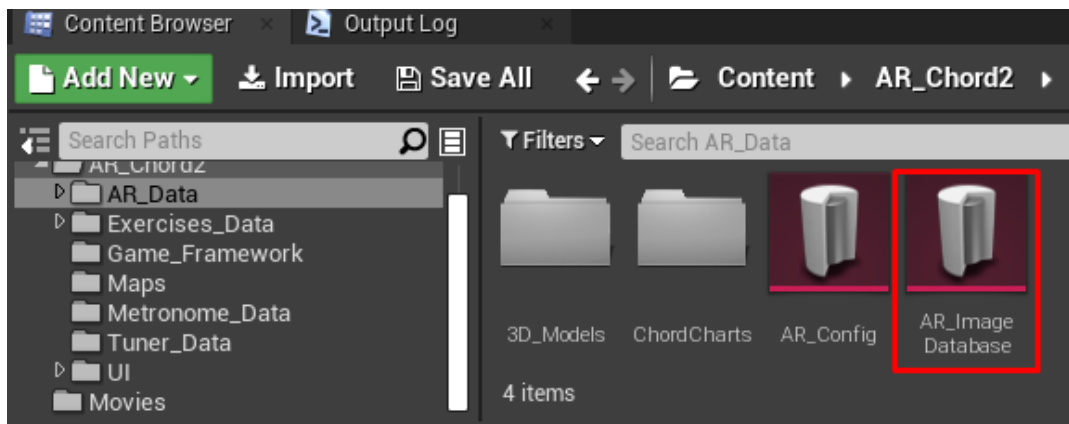


Εικόνα 4.20: Εκχώρηση του "AR\_GameMode" στο World Settings του Level "AR\_Chords"



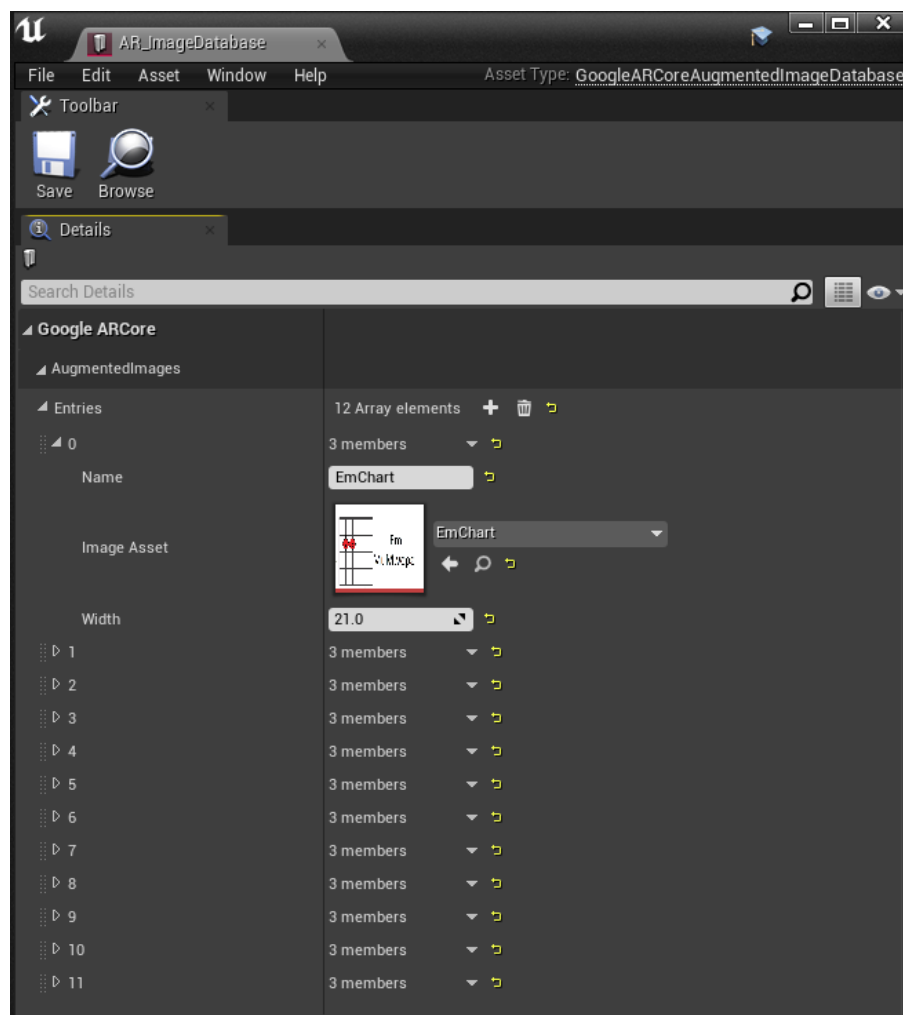
#### 4.1.6 Δημιουργία Βάσης Δεδομένων Επαυξημένων Εικόνων & AR Session Config

Δημιουργήθηκε *Data Asset* τύπου *GoogleARCoreAugmentedImageDatabase* με όνομα “AR\_ImageDatabase”.



Εικόνα 4.21: Η Βάση Δεδομένων Επαυξημένων Εικόνων στο Content Browser του UE4

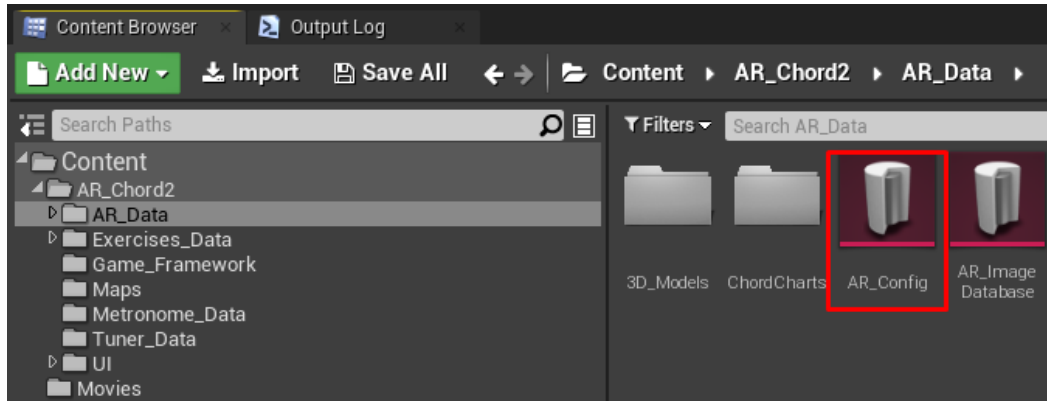
Σε αυτό εισήχθησαν οι εικόνες διαγραμμάτων που δημιουργήσαμε.



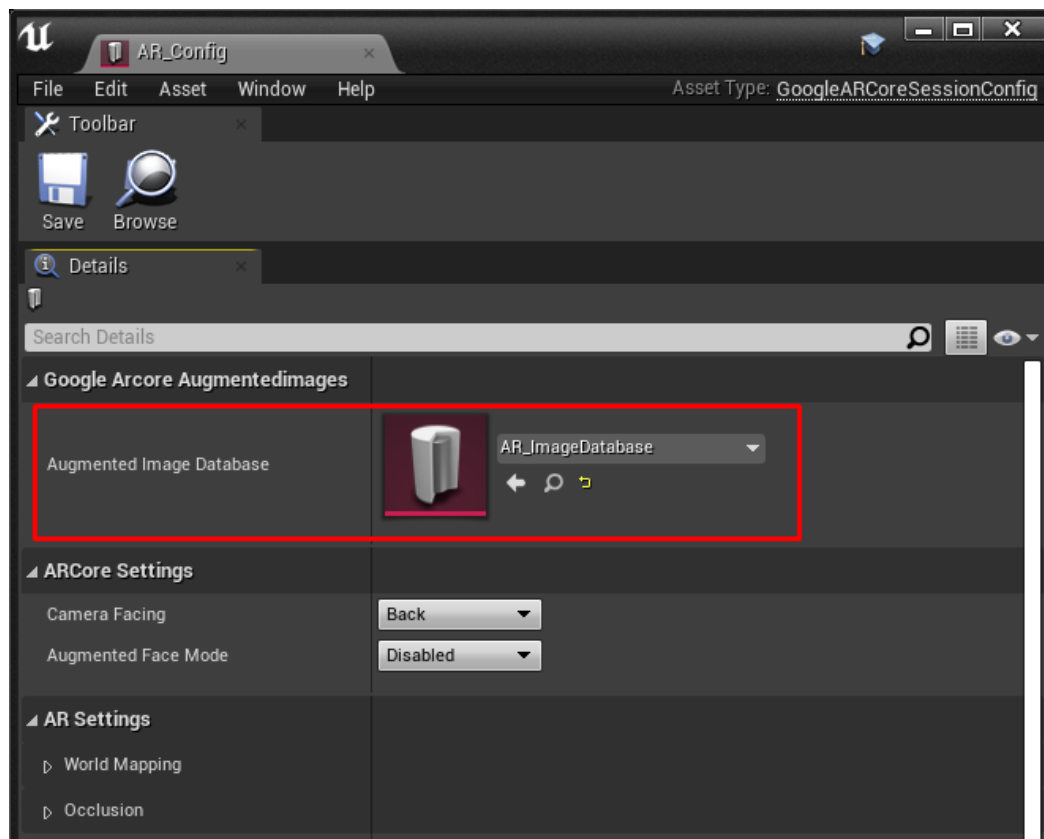
Εικόνα 4.22: Η Βάση Δεδομένων Επαυξημένων Εικόνων

Ως αποτέλεσμα, έχουμε 12 εικόνες που καταλαμβάνουν τις θέσεις 0-11 (όπως σε μία δομή δεδομένων τύπου πίνακα) και η κάθε εικόνα έχει το δικό της Index, το δικό της Όνομα και πληροφορίες για το πλάτος της στον φυσικό κόσμο.

Δημιουργήθηκε, επίσης, ένα *Data Asset* τύπου *GoogleARCoreSessionConfig* με όνομα "AR\_Config", στο οποίο φορτώθηκε η βάση δεδομένων "AR\_ImageDatabase".



Εικόνα 4.23: Το Αρχείο "AR\_Config" στο Content Browser του UE4

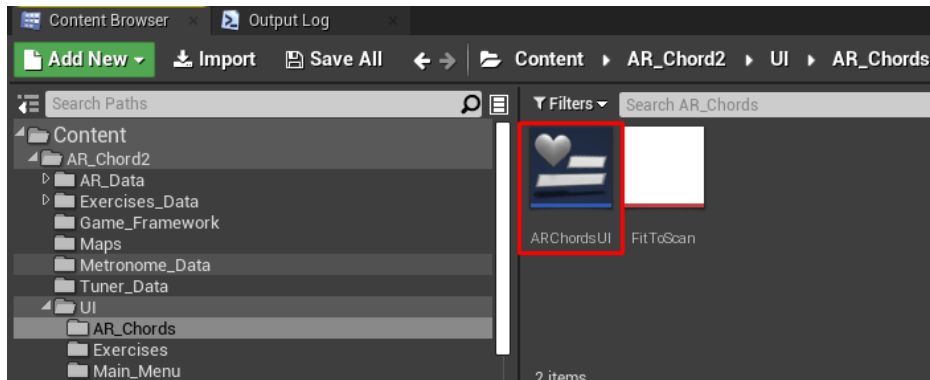


Εικόνα 4.24: Το Αρχείο "AR\_Config"

Το συγκεκριμένο *Data Asset* θα περνάει σαν είσοδος στο *ARCore*, ώστε να ξεκινήσει μια AR συνεδρία παρέχοντας του όλα τα δεδομένα που χρειάζεται.

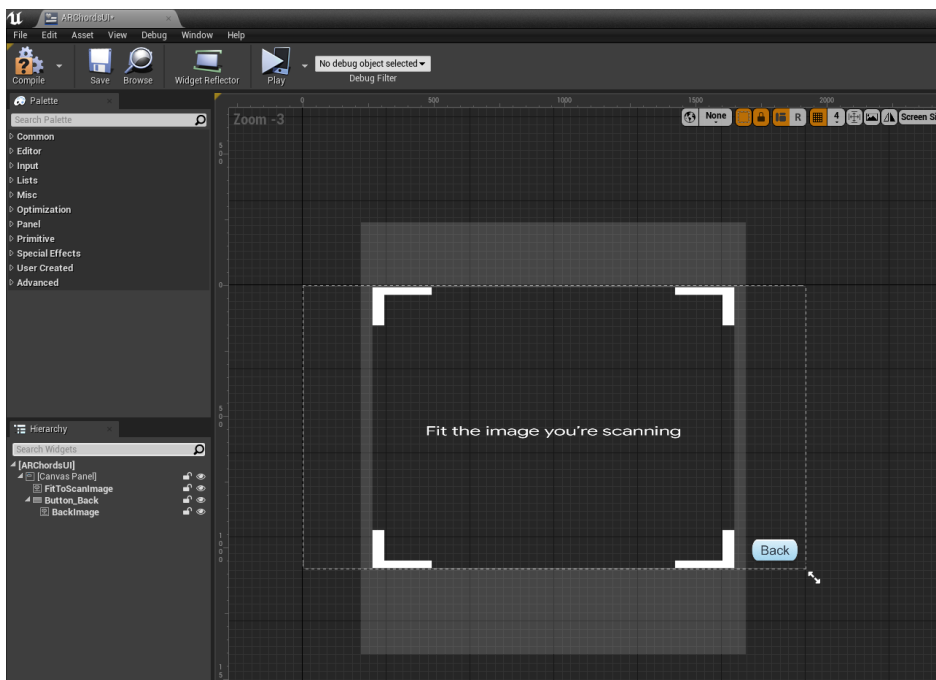
#### 4.1.ε Δημιουργία Widget για την λειτουργία «AR Chords»

Δημιουργήθηκε ένα *Widget Blueprint* με το όνομα “AR\_ChordsUI”.



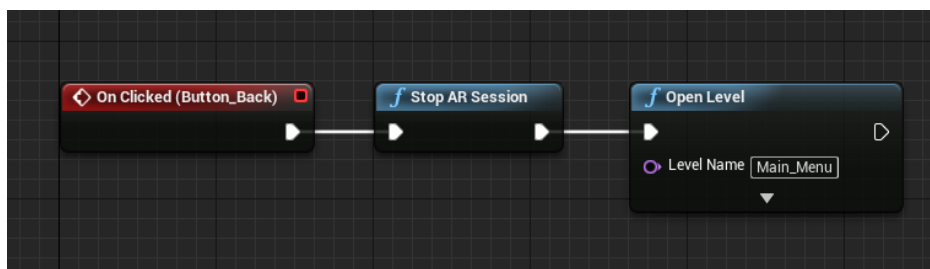
Εικόνα 4.25: Το Widget "ARChordsUI" στο Content Browser του UE4

Σε αυτό προστέθηκε μια εικόνα “FitToScan” και ένα κουμπι “Back”.



Εικόνα 4.26: Το Widget "ARChordsUI"

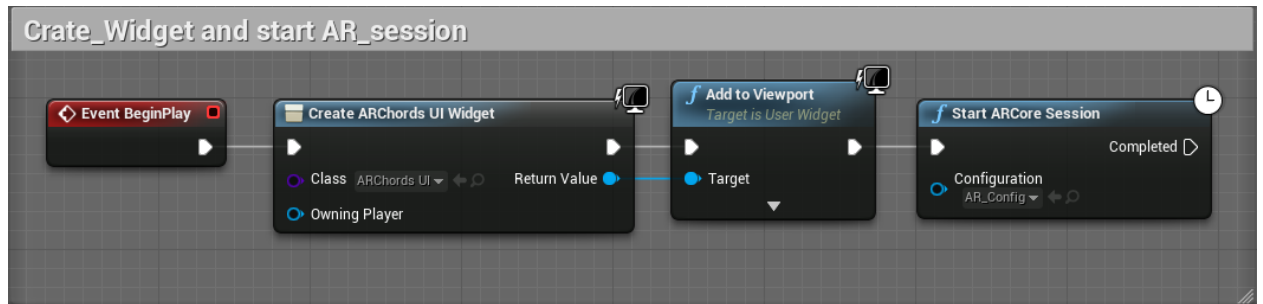
Ο χρήστης πατώντας το κουμπι “Back” επιστρέφει στο αρχικό μενού αφού γίνει διακοπή της AR συνεδρίας. Αυτό επιτυγχάνεται μέσω της συνάρτησης *Stop AR Session* και της συνάρτησης *Open Level*.



Εικόνα 4.27: Event Graph του Widget "ARChordsUI"

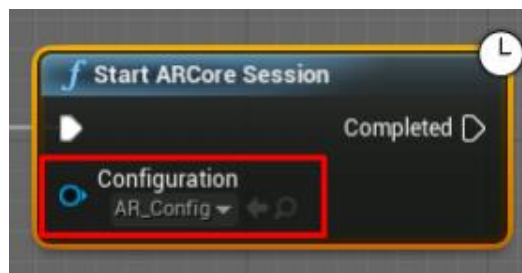
#### 4.1.στ Εμφάνιση Widget & Έναρξη AR Συνεδρίας

Στο *Level Blueprint* του “AR\_Chords” υλοποιήθηκε ο παρακάτω κώδικας.



Εικόνα 4.28: Κώδικας για Εμφάνιση Widget & Έναρξη AR Συνεδρίας

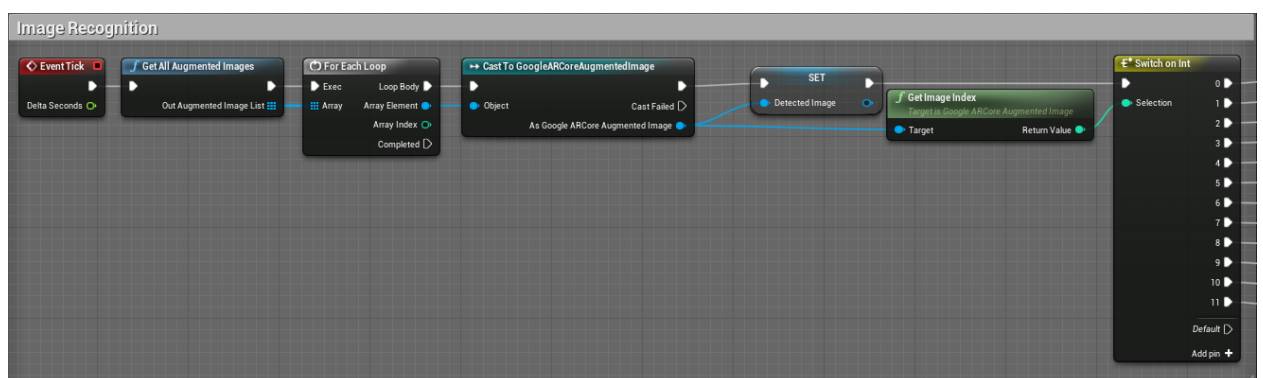
Ξεκινώντας με τον κόμβο *Event Begin Play*, αυτό που ακολουθεί θα ενεργοποιηθεί όταν ξεκινήσει το *Level*. Με τους κόμβους *Create Widget* και *Add to Viewport* θα εμφανιστεί στην οθόνη το *widget* που μόλις υλοποιήθηκε. Τέλος, με τον κόμβο *Start ARCore Session* ξεκινάει η AR συνεδρία, έχοντας δώσει σαν εισοδο το αρχείο “AR\_config” που αναφέρθηκε παραπάνω. Ενεργοποιείται, δηλαδή, η κάμερα της κινητής συσκευής και “ψάχνει” για τις εικόνες που έχουμε ορίσει στη Βάση Δεδομένων του “AR\_config”.



Εικόνα 4.29: Ο κόμβος Start ARCore Session

#### 4.1.ζ Αναγνώριση Εικόνας, Εμφάνιση 3D Μοντέλου & Τοποθέτηση 3D Μοντέλου στην φυσική εικόνα

Για την αναγνώριση της εικόνας απαιτείται ο παρακάτω κώδικας.



Εικόνα 4.30: Ο Κώδικας για την αναγνώριση εικόνας

Εδώ ξεκινάμε με τον κόμβο *Event Tick* που σημαίνει πως ότι ακολουθεί, θα επαναλαμβάνεται σε κάθε *frame*, δηλαδή κάθε καρέ που λαμβάνει η κάμερα της κινητής συσκευής. Ο κόμβος *Get All Augmented Images* θα συγκεντρώσει (αν υπάρχουν) τις

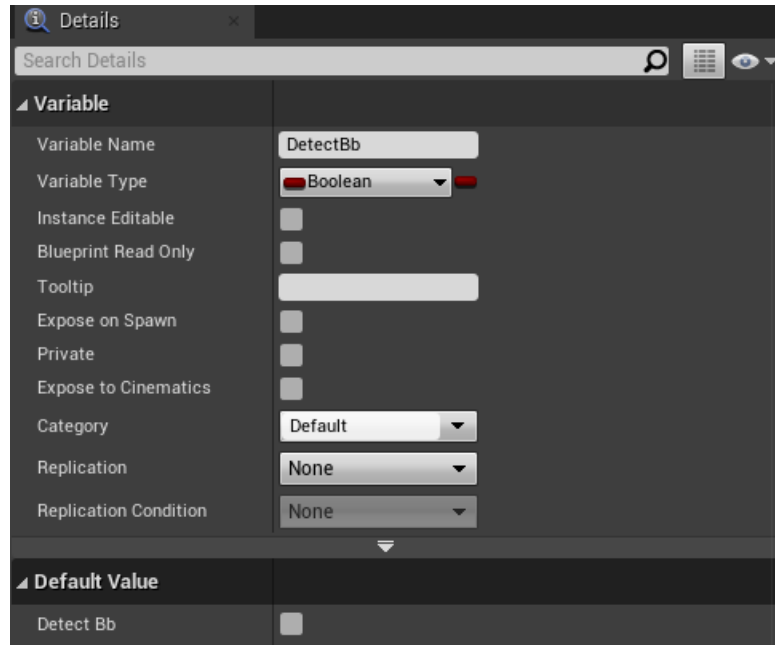
εικόνες της Βάσης Δεδομένων που βρίσκονται στο καρτέ και θα τις εισάγει σε έναν πίνακα. Σε αυτό το σημείο, τα αναγνωριστικά των εικόνων του πίνακα που εξάγεται, δεν ταυτίζονται με τα αναγνωριστικά της Βάσης Δεδομένων που ορίσαμε. Για παράδειγμα, αν αναγνωρίστηκε η εικόνα που βρίσκεται στη θέση 10 της Βάσης Δεδομένων με όνομα 'BbChart', στον πίνακα θα αποθηκευτεί στην θέση 0 με ένα όνομα το οποίο θα παραχθεί τυχαία από το πρόγραμμα. Ο κόμβος *For Each Loop* λαμβάνει σαν είσοδο τον πίνακα που εξήχθη από τον προηγούμενο κόμβο και για κάθε στοιχείο του θα επαναλάβει το κομμάτι κώδικα που συνδέεται στο *Loop Body*. Για παράδειγμα, αν εξαχθεί πίνακας με τρεις εικόνες, το κομμάτι του κώδικα δεξιά του *For Each Loop* θα εκτελεστεί τρεις φορές και κάθε φορά θα παρέχει στον επόμενο κόμβο το αντίστοιχο στοιχείο του πίνακα μέσω του πεδίου *Array Element*. Ο κόμβος *Cast To GoogleARCoreAugmentedImage* θα μεταφράσει τα αναγνωριστικά της εικόνας που έδωσε ο κόμβος *Get All Augmented Images* στα αναγνωριστικά που ορίσαμε στη Βάση Δεδομένων. Χρησιμοποιώντας το προηγούμενο παράδειγμα, η εικόνα στην θέση 0 με την τυχαία ονομασία, τώρα θα έχει index την τιμή 10 και όνομα 'BbChart'. Με αυτόν τον τρόπο πλέον μπορούμε να γνωρίζουμε πραγματικά ποια εικόνα αναγνωρίστηκε απ' την κάμερα. Την έξοδο του κόμβου *Cast To GoogleARCoreAugmentedImage* που μας δίνει την εικόνα που αναγνωρίστηκε την αποθηκεύουμε σε μία μεταβλητή με το όνομα "Detected\_Image" με την βοήθεια του κόμβου *SET*, γιατί θα τη χρειαστούμε στη συνέχεια του κώδικα. Ο κόμβος *Get Image Index* λαμβάνει το Index της εικόνας και το τροφοδοτεί στον επόμενο κόμβο. Στο παράδειγμά μας θα τροφοδοτήσει την τιμή 10. Ο τελευταίος κόμβος *Switch on Int* λειτουργεί όπως μια δομή επιλογής Switch με βάση έναν Integer. Ανάλογα με τον Integer που θα λάβει, θα επιλέξει την αντίστοιχη διαδρομή. Εφόσον έχουμε 12 εικόνες προς αναγνώριση, μπορεί να λάβει τις τιμές από 0 έως 11. Υπάρχουν αντίστοιχα 12 διαφορετικές διαδρομές στις οποίες θα εκτελεστούν η εμφάνιση μοντέλου και η τοποθέτηση μοντέλου πάνω στην εικόνα, που αφορούν την εικόνα που μόλις αναγνωρίστηκε. Εν προκειμένω, η ροή θα συνεχίσει την διαδρομή από την έξοδο 10.



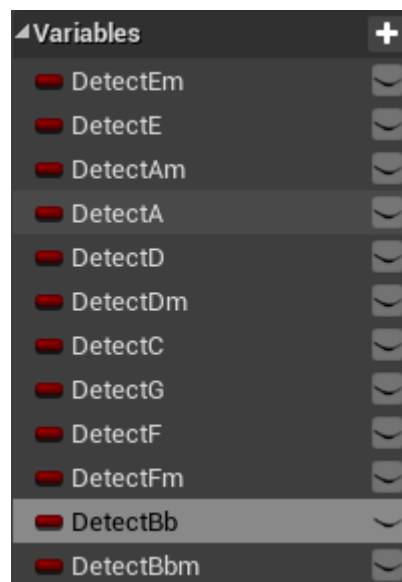
Εικόνα 4.31: Ο κόμβος Switch on Int

Για την εμφάνιση του μοντέλου και την τοποθέτησή του πάνω στην εικόνα έχουμε το ίδιο κομμάτι κώδικα για κάθε ζεύγος εικόνας-μοντέλου. Επομένως, θα εξηγηθεί ο κώδικας ενός ζεύγους ο οποίος θα ισχύει και για τα υπόλοιπα ζεύγη.

Αρχικά, θα οριστεί μία μεταβλητή τύπου Boolean για κάθε ζεύγος εικόνας-μοντέλου με όνομα “Detect*ΌνομαΣυγχορδίας*”. Π.χ. για το ζεύγος εικόνας BbChart - μοντέλου BbModel (δηλαδή της συγχορδίας Bb) θα οριστεί μεταβλητή Boolean με Όνομα “DetectBb”.

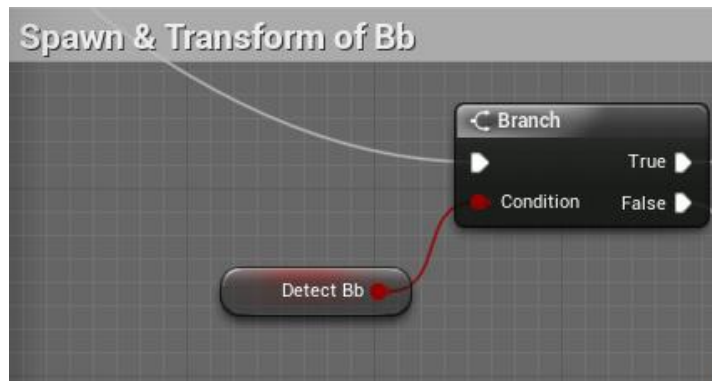


Εικόνα 4.32: Λεπτομέρειες μεταβλητής DetectBb



Εικόνα 4.33: Οι μεταβλητές Detect*ΌνομαΣυγχορδίας*

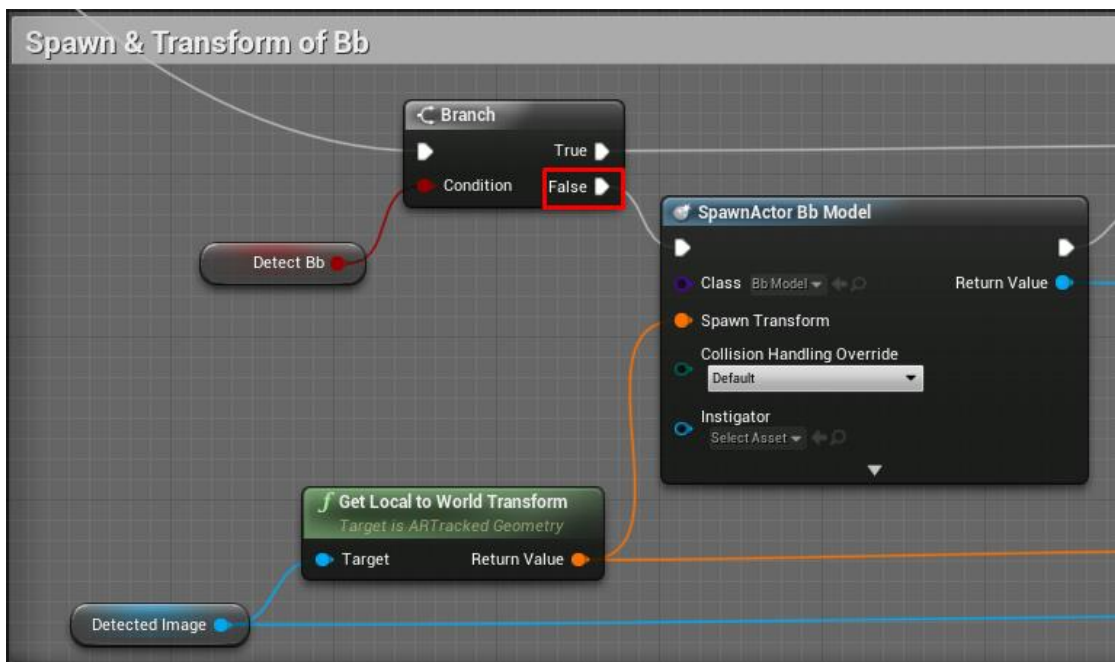
Θα ακολουθήσουμε την διαδρομή του παραδείγματος της συγχορδίας Bb. Θα πρέπει να γνωρίζουμε αν το συγκεκριμένο καρτέ είναι το πρώτο καρτέ στο οποίο εμφανίζεται η εικόνα “BbChart” για να εμφανίσουμε (Spawn) το μοντέλο “BbModel”. Εάν είναι η δεύτερη και πλέον φορά που εμφανίζεται θα πρέπει απλά να το επανατοποθετήσουμε πάνω στην εικόνα χωρίς να το εμφανίζουμε εκ νέου.



Εικόνα 4.34: Ο κόμβος Branch

Η ροή, επομένως, συνεχίζει με ένα κόμβο *Branch* (δομή επιλογής if) ο οποίος ελέγχει αυτό ακριβώς. Η Boolean μεταβλητή “DetectBb” τροφοδοτεί το *Condition* της *Branch* με μια τιμή True ή False. Εφόσον είναι η πρώτη φορά που αναγνωρίστηκε και αφού η “Detect Bb” έχει αρχική τιμή False, θα τροφοδοτήσει την *Branch* με False. Επομένως, η ροή θα συνεχιστεί από την έξοδο False.

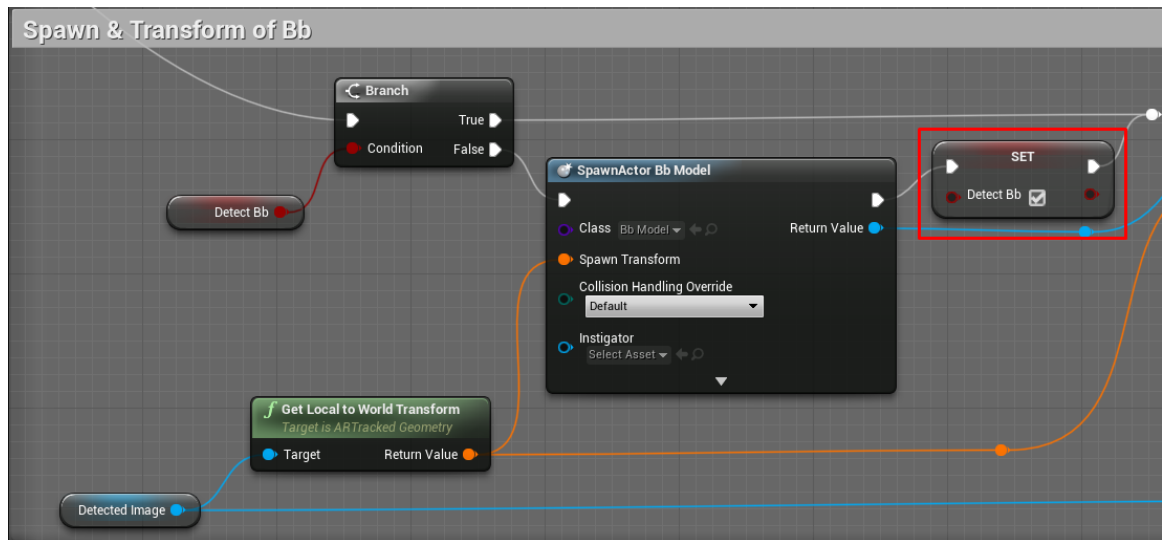
Στη συνέχεια της εξόδου False βρίσκεται ο κόμβος *Spawn Actor*.



Εικόνα 4.35: Ο κόμβος Spawn Actor στην έξοδο False

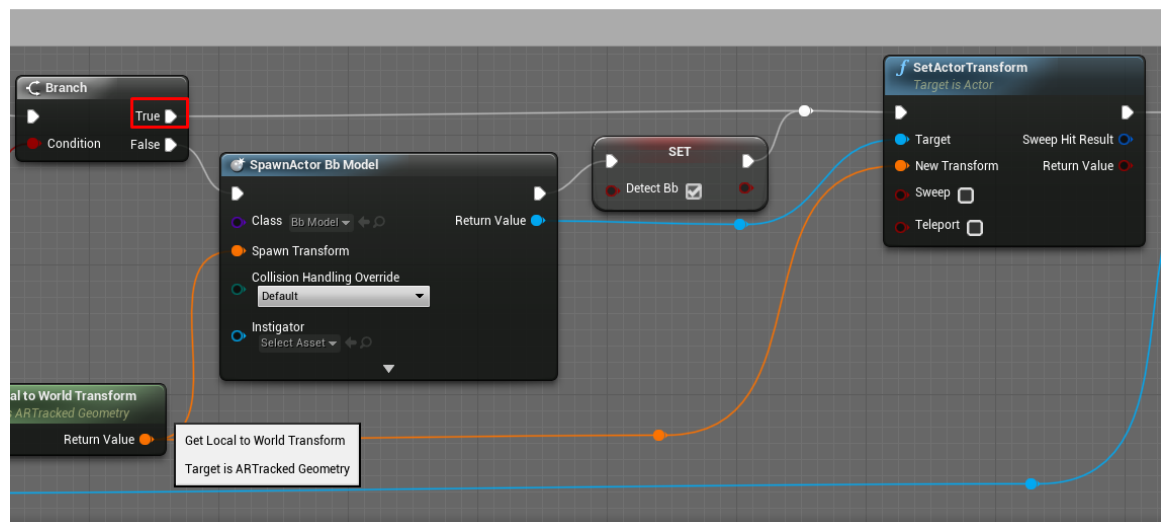
Αυτός ο κόμβος εμφανίζει το μοντέλο που επιλέγουμε στο πεδίο *Class* (“Bb Model”), στην θέση που παρέχει ο κόμβος *Get Local to World Transform*, δηλαδή πάνω στην εικόνα “BmChart”.

Στη συνέχεια, θα πρέπει να δώσουμε στην μεταβλητή “Detect Bb” την τιμή True ώστε στο επόμενο καρέ που θα αναγνωριστεί η εικόνα “BbChart” να μην εκτελεστεί ξανά ο κόμβος *Spawn Actor*, δηλαδή να μην ξαναεμφανιστεί το μοντέλο.



Εικόνα 4.36: Ο κόμβος SET

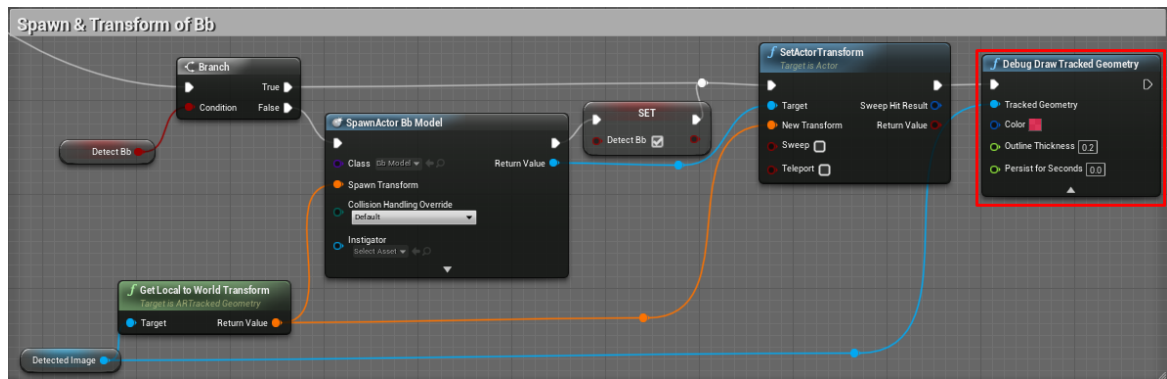
Με την βοήθεια του κόμβου *SET* η μεταβλητή “Detect Bb” λαμβάνει την τιμή True. Επομένως, στο επόμενο καρέ που θα αναγνωριστεί η εικόνα “BbChart” η ροή θα συνεχίσει από την έξοδο True της *Branch*.



Εικόνα 4.37: Ο κόμβος Set Actor Transform στην έξοδο True

Στη συνέχεια αυτής της εξόδου βρίσκεται ο κόμβος *Set Actor Transform*. Ο συγκεκριμένος κόμβος επανατοποθετεί σε κάθε καρέ το μοντέλο πάνω στην εικόνα. Το ποιο μοντέλο θα επανατοποθετηθεί ορίζεται από την έξοδο που του παρέχει ο κόμβος *Spawn Actor*. Η νέα τοποθεσία στην οποία θα επανατοποθετηθεί το μοντέλο καθορίζεται από την συνάρτηση *Get Local to World Transform*.



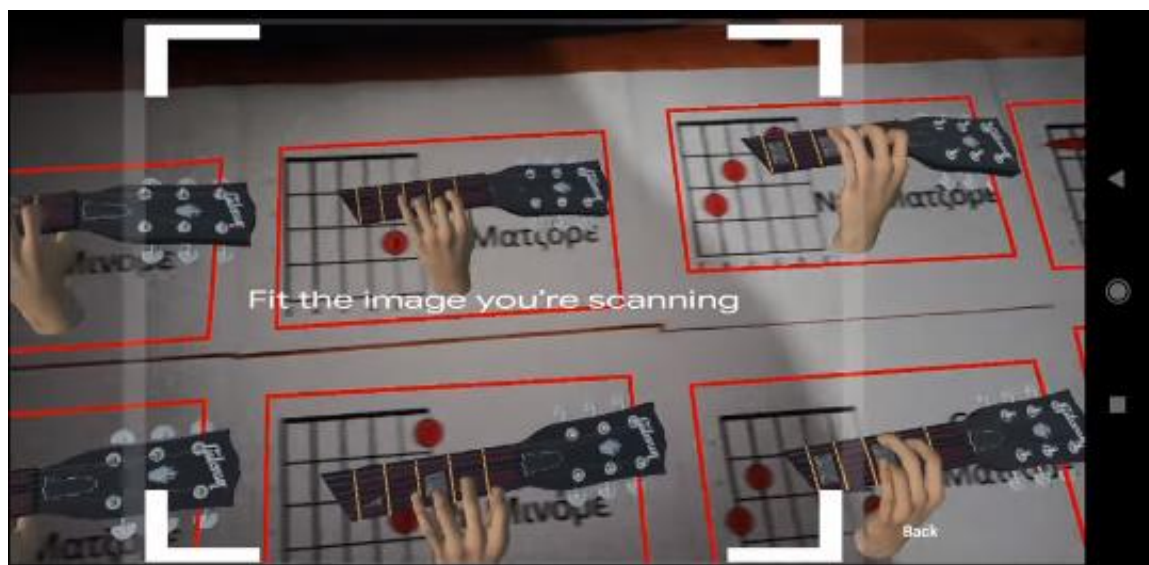


Εικόνα 4.38: Ο κόμβος Debug Draw Tracked Geometry

Τέλος υπάρχει ο κόμβος *Debug Draw Tracked Geometry*. Αυτός ο κόμβος “ζωγραφίζει” ένα πλαίσιο γύρω απ’ την εικόνα που αναγνωρίστηκε. Παίρνει σαν είσοδο την εικόνα στην οποία θα βάλει το πλαίσιο στο πεδίο *Tracked Geometry*, το χρώμα του πλαισίου στο πεδίο *Color* και το πάχος του πλαισίου στο πεδίο *Outline Thickness*.



Εικόνα 4.39: Επίδειξη Λειτουργίας ARChords στην συγχορδία Σιβ Ματζόρε



Εικόνα 4.40: Επίδειξη Λειτουργίας ARChords σε διάφορες Συγχορδίες

## 4.2 Η Λειτουργία «Exercises»

### 4.2.α Σκοπός της λειτουργίας

Τα περισσότερα συγγράμματα εκμάθησης κιθάρας συνοδεύονται από ένα CD, στο οποίο είναι αποθηκευμένα αρχεία ήχου με ηχογραφήσεις από τις ασκήσεις που περιέχει το σύγγραμμα. Τα τελευταία χρόνια, που γίνεται και έκδοση συγγραμμάτων διαδικτυακά σε μορφή pdf, ο εκδότης παρέχει, πέρα από το ίδιο το σύγγραμμα, και ένα πακέτο αρχείων, το οποίο περιέχει τις ηχογραφημένες ασκήσεις και πολλές φορές κάποιες από αυτές βιντεοσκοπημένες.

Σκοπός της λειτουργίας “Exercises” είναι να παρέχει τη δυνατότητα στο χρήστη να επιλέξει μέσω πλοήγησης στο μενού την βιντεοσκοπημένη άσκηση που θέλει να παρακολουθήσει (List Mode) ή να στρέψει την κάμερα της κινητής συσκευής προς την εκφώνηση της άσκησης και να ξεκινήσει αυτόματα η αναπαραγωγή (AR Mode).

### 4.2.β Βιντεοσκόπηση Ασκήσεων & Συγγραφή Εκφωνήσεων

Έγινε βιντεοσκόπηση της εκτέλεσης πέντε ενδεικτικών ασκήσεων με χρήση βιντεοκάμερας και στη συνέχεια έγινε μετατροπή των αρχείων σε .mp4, ώστε να γίνουν δεκτά από το Unreal Engine 4.



Εικόνα 4.41: Στιγμιότυπο από βίντεο εκτέλεσης Άσκησης

Οι εκφωνήσεις των ασκήσεων γράφτηκαν στο πρόγραμμα Microsoft Word και στη συνέχεια μετατράπηκαν σε αρχείο εικόνας .png. Για την συγγραφή μουσικού πενταγράμμου/ταμπλατούρας χρησιμοποιήθηκε το πρόγραμμα Guitar Pro 5.

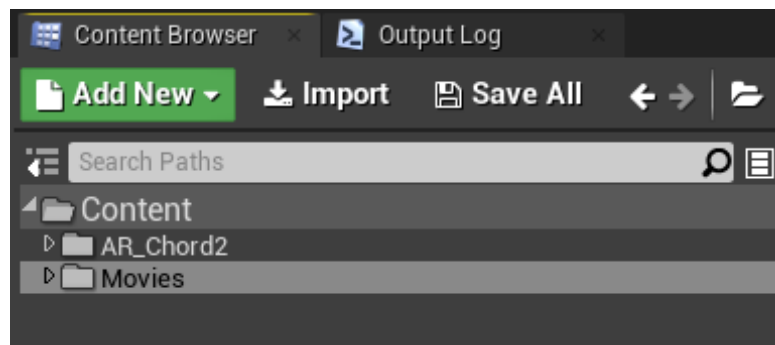
### Άσκηση 1

Παίξε τις συγχορδίες E, C#m, G#m και A χτυπώντας τέσσερις φορές την κάθε μία σε αξία ενός τετάρτου όπως φαίνεται παρακάτω:

Εικόνα 4.42 Εκφώνηση της Άσκησης 1

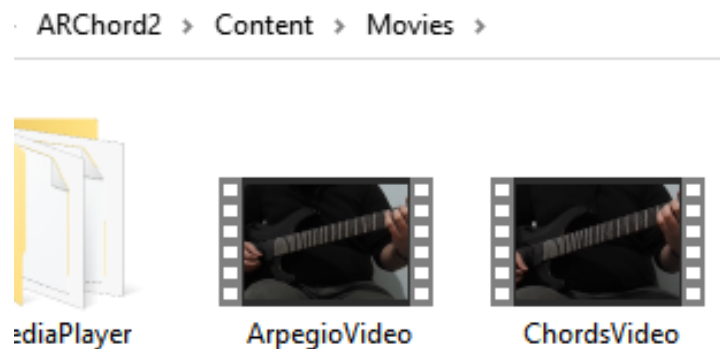
#### 4.2.γ Εισαγωγή Αρχείων Βίντεο & Εικόνας στο UE4

Για την εισαγωγή των αρχείων βίντεο στο Unreal Engine 4 δημιουργήθηκε φάκελος στο Content Browser με όνομα “Movies”.



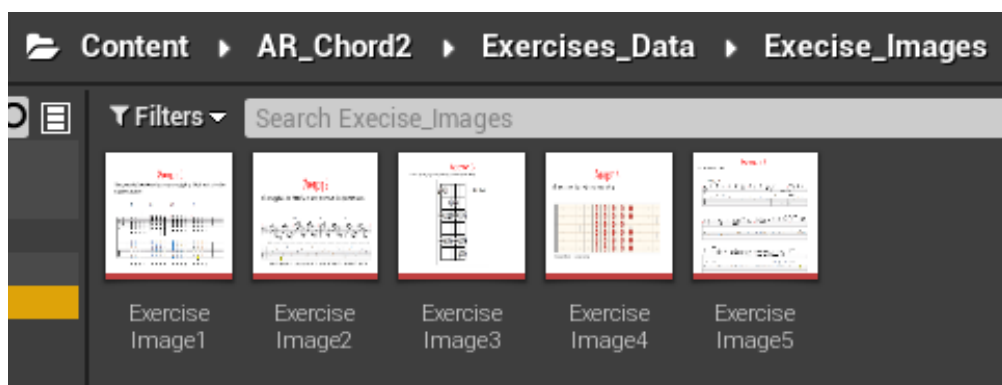
Εικόνα 4.43: Ο φάκελος Movies

Στη συνέχεια, βρέθηκε ο φάκελος “Movies” μέσω του File Explorer του λειτουργικού συστήματος και έγινε επικόλληση των αρχείων video εντός του.



Εικόνα 4.44: Ο φάκελος Movies στο File Explorer

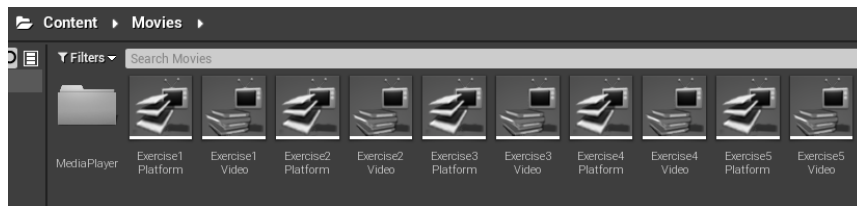
Η εισαγωγή των εικόνων έγινε με τον παραδοσιακό τρόπο από το Content Browser του προγράμματος και δόθηκαν ονόματα: ExerciseImageΑριθμόςΆσκησης.



Εικόνα 4.45: Οι Εκφωνήσεις των Ασκήσεων

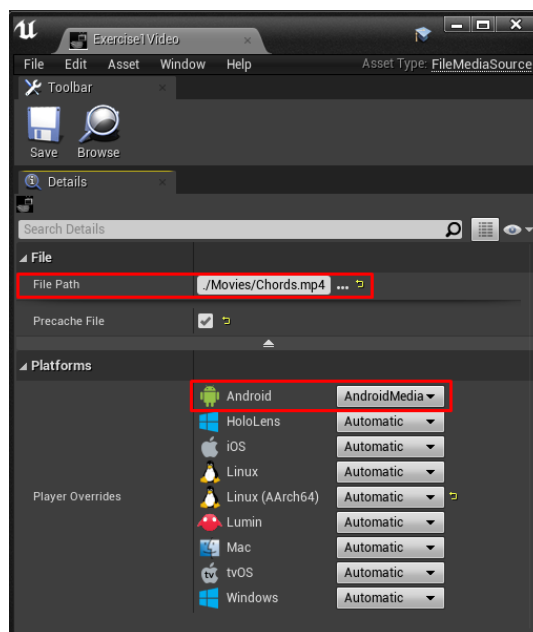
#### 4.2.6 Δημιουργία File Media Source & Platform Media Source

Δημιουργήθηκαν για κάθε Άσκηση ένα αρχείο *File Media Source* και ένα αρχείο *Platform Media Source*.



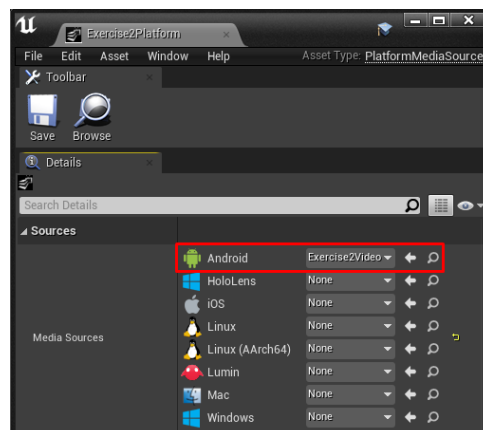
Εικόνα 4.46: Τα File Media Source & Platform Media Source

Σε κάθε αρχείο *File Media Source* στο πεδίο *File Path* επιλέχθηκε το path του αντίστοιχου βίντεο και στο πεδίο *Player Overrides*, *Android* επιλέχθηκε *Android Media*.



Εικόνα 4.47: Το File Media Source της Άσκησης 1

Σε κάθε αρχείο *Platform Media Source* επιλέχθηκε στο πεδίο *Media Sources*, *Android* το αντίστοιχο αρχείο *File Media Source*.

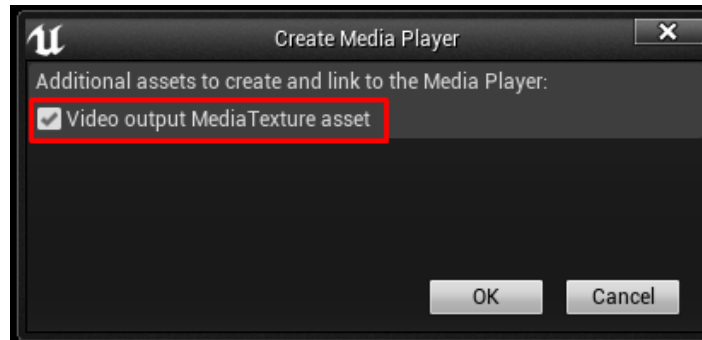


Εικόνα 4.48: Το Platform Media Source της Άσκησης 1

Αυτό έγινε για να μπορεί να γίνει αναπαραγωγή του βίντεο σε πλατφόρμες Android.

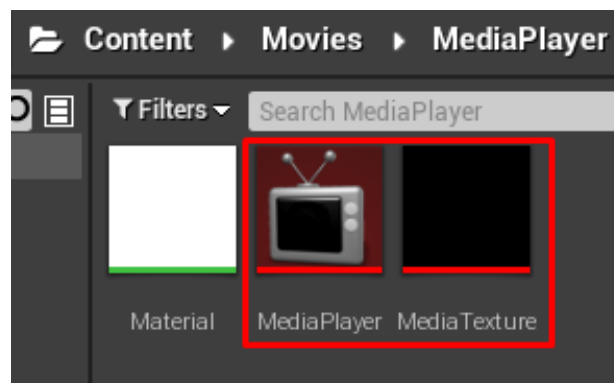
#### 4.2.ε Δημιουργία Media Player & Material

Δημιουργήθηκε αρχείο *Media Player* στο *Content Browser* του προγράμματος. Κατά τη δημιουργία εμφανίστηκε το παρακάτω παράθυρο στο οποίο τσεκαρίστηκε το *checkbox* “*Video output Media Texture asset*”.



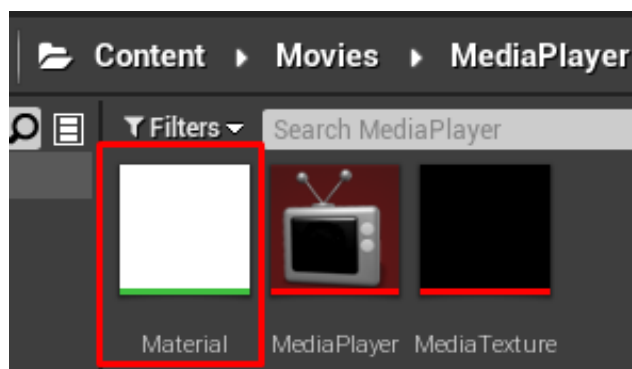
Εικόνα 4.49: Το παράθυρο Create Media Player

Πατώντας το κουμπι “OK”, εκτός του αρχείου *Media Player*, δημιουργήθηκε και ένα αρχείο *Media Texture*.



Εικόνα 4.50: Το Media Player & Media Texture

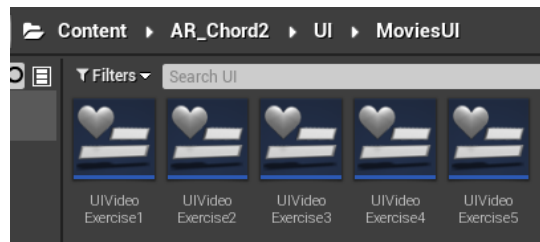
Από το αρχείο *Media Texture* και με την επιλογή *Create Material* δημιουργήθηκε ένα αρχείο *Material*.



Εικόνα 4.10: Το Material Που δημιουργήθηκε από την επιλογή Create Material

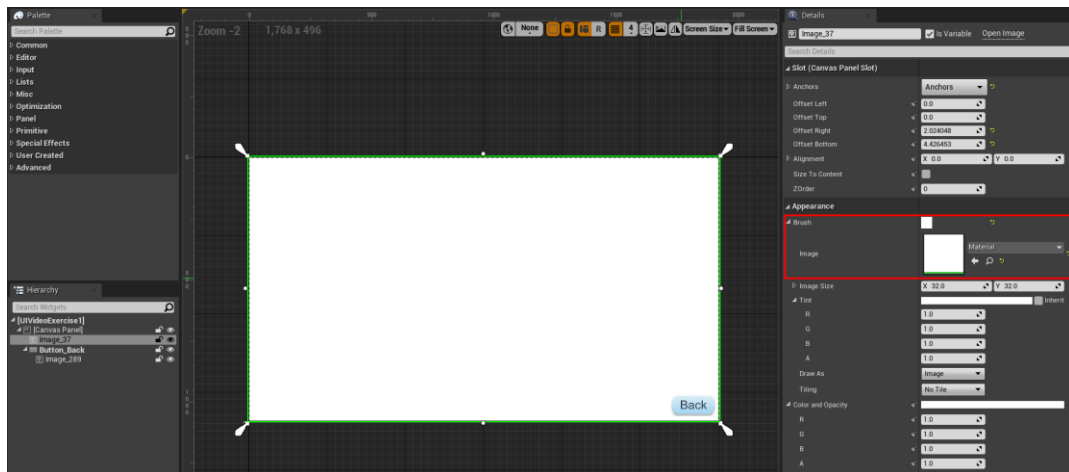
## 4.2.στ Δημιουργία Widget για κάθε Άσκηση

Δημιουργήθηκε ένα *Widget Blueprint* για κάθε άσκηση με όνομα “UIVideoExerciseΑριθμόςΆσκησης”.



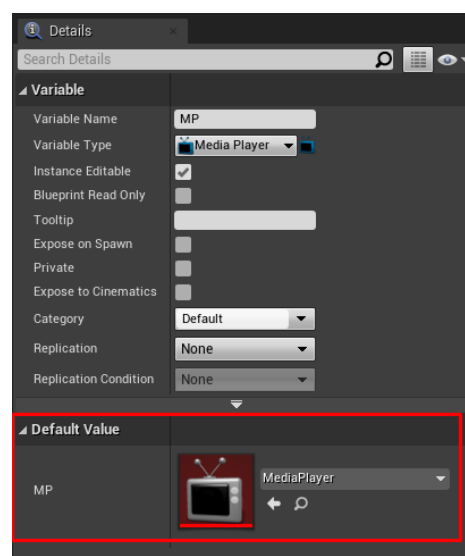
Εικόνα 4.52: Τα Widgets των Ασκήσεων

Στο κάθε *Widget* προστέθηκε ένα *Image*, το οποίο καλύπτει όλη την επιφάνεια της οθόνης και σαν είσοδο στο *Image* δόθηκε το *Material* που δημιουργήθηκε προηγουμένως. Επίσης, προστέθηκε και ένα κουμπί “Back”.



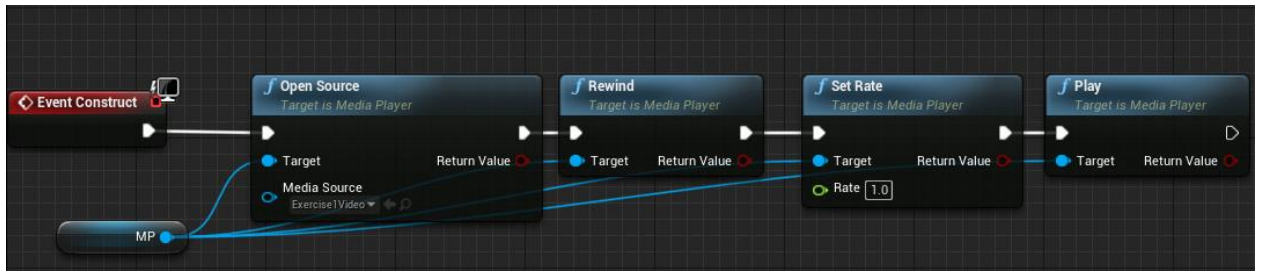
Εικόνα 4.53: Το Widget της Άσκησης 1

Στο *Event Graph* του κάθε *Widget* δημιουργήθηκε μεταβλητή τύπου *Media Player* με όνομα “MP” στην οποία αρχικοποιήθηκε το αρχείο “Media Player” που δημιουργήθηκε παραπάνω.



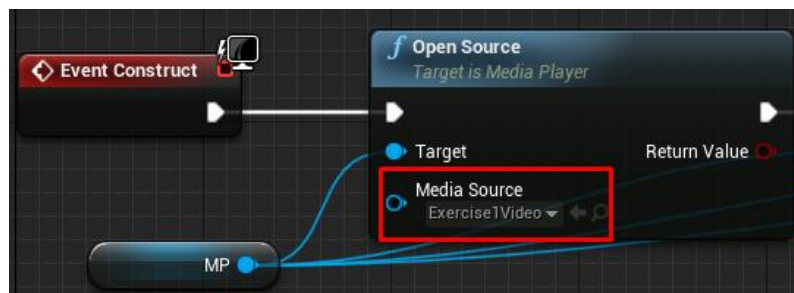
Εικόνα 4.54: Η μεταβλητή MP

Στη συνέχεια γράφτηκε ο παρακάτω κώδικας ο οποίος αναπαράγει το αντίστοιχο Video.



Εικόνα 4.55: Ο κώδικας για την αναπαραγωγή του βίντεο

Ο κόμβος *Event Construct* εκτελεί τον κώδικα που είναι συνδεδεμένος σε αυτόν όταν εμφανιστεί το *Widget Blueprint* στην οθόνη. Με τον κόμβο *Open Source* ανοίγει ένα *Media Player* για να γίνει η αναπαραγωγή του βίντεο. Εν προκειμένω, ανοίγει το *Media Player* που είναι αποθηκευμένο στη μεταβλητή “MP” για να γίνει αναπαραγωγή του αρχείου “Exercise1Video”.



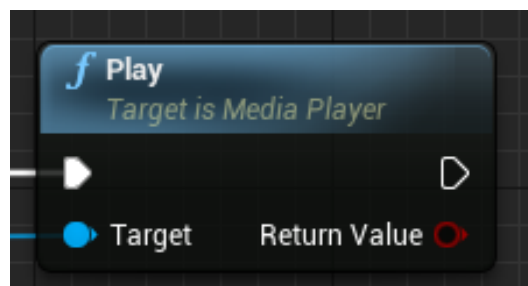
Εικόνα 4.56: Ο κόμβος Open Source

Στη συνέχεια, ο κόμβος *Rewind* προστέθηκε ώστε η αναπαραγωγή του βίντεο να ξεκινήσει από την αρχή. Ο κόμβος *Set Rate* ρυθμίζει την ταχύτητα και την φορά αναπαραγωγής του βίντεο. Η κατάλληλη τιμή *Rate* για την αναπαραγωγή στην κανονική φορά και στην κανονική ταχύτητα είναι “1.0”.



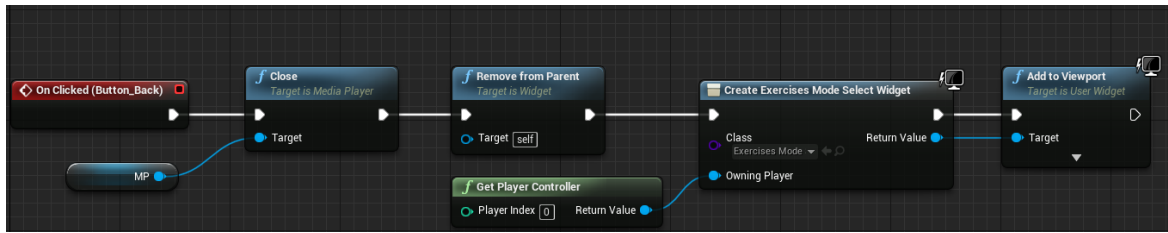
Εικόνα 4.57: Ο κόμβος Set Rate

Τέλος, με τον κόμβο *Play* ξεκινά η αναπαραγωγή του βίντεο.



Εικόνα 4.58: Ο κόμβος Play

Ο παρακάτω κώδικας εκτελείται μόλις πατηθεί το κουμπι “Back”.



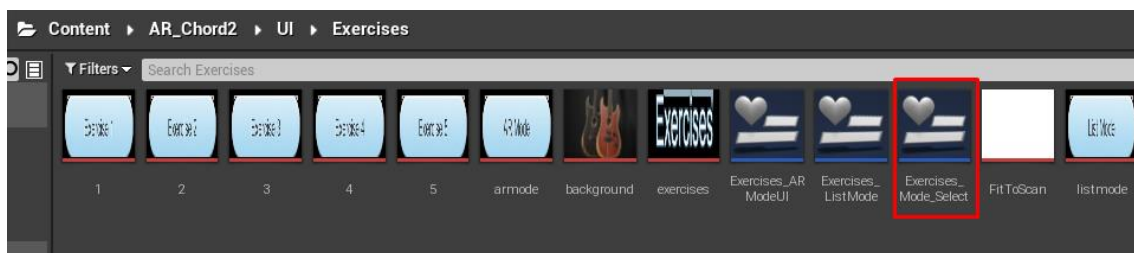
Εικόνα 4.59: Ο κώδικας για το πάτημα του κουμπιού Back

Ο κόμβος *Close* κλείνει το *Media Player* που είχε ανοίξει με τον κόμβο *Open Source*, ενώ ο κόμβος *Remove from Parent* αφαιρεί από την οθόνη το παρόν *Widget*. Με τους κόμβους *Create Widget* και *Add to Viewport* εμφανίζεται το *Widget* “Exercises Mode Select”, το οποίο θα εξηγηθεί παρακάτω.

#### 4.2.ζ Δημιουργία Widget για επιλογή μεταξύ List Mode/AR Mode

Σε αυτό το σημείο, είναι όλα έτοιμα για να γίνει η αναπαραγωγή των βίντεο ανοίγοντας ένα από τα *Widget Blueprints* που δημιουργήθηκαν παραπάνω για την κάθε άσκηση. Ο χρήστης θα πρέπει να διαλέξει την άσκηση που θέλει να παρακολουθήσει επιλέγοντας από λίστα (*List Mode*) ή μέσω της κάμερας της κινητής συσκευής και στοχεύοντας στην εκφώνηση της άσκησης (*AR Mode*).

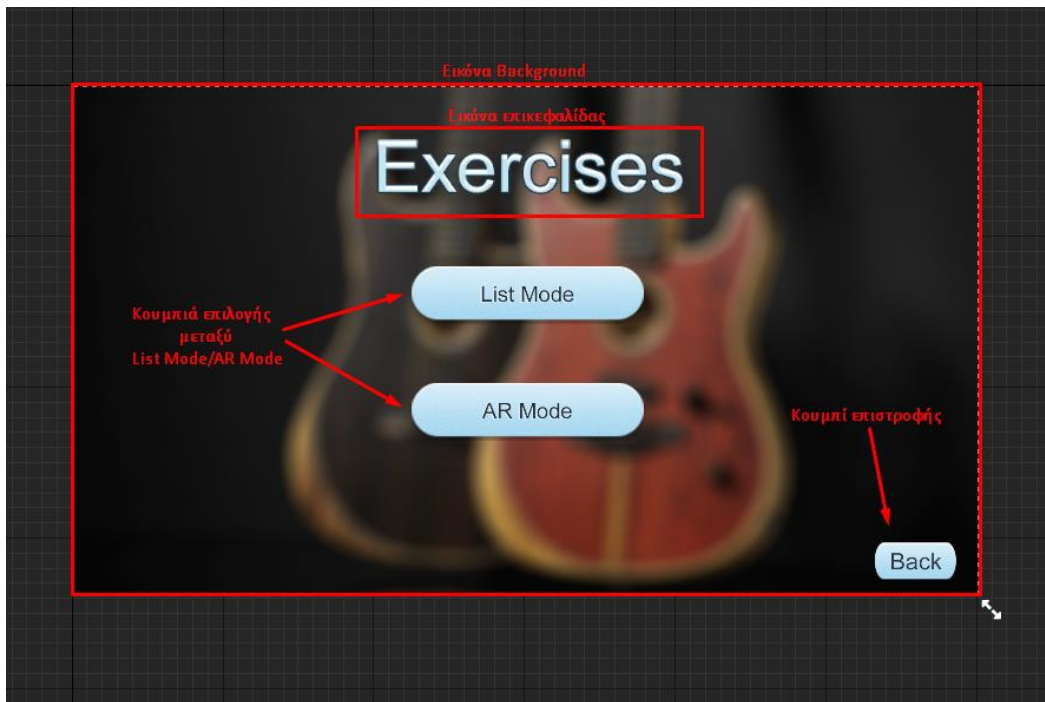
Για αυτό το λόγο, δημιουργήθηκε *Widget Blueprint* με όνομα “Exercises Mode Select”.



Εικόνα 4.60: Το Widget Exercises Mode Select στο Content Browser

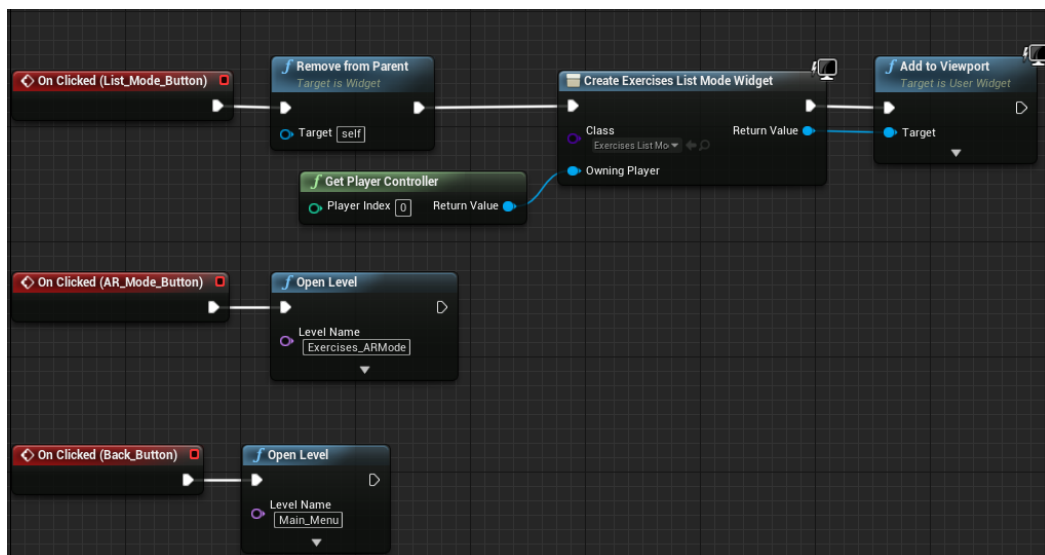
Σε αυτό έγινε εισαγωγή εικόνας *background*, εικόνας επικεφαλίδας, δύο κουμπιών για επιλογή μεταξύ *List Mode/AR Mode* και ενός κουμπιού επιστροφής στο αρχικό μενού (*Back*).





Εικόνα 4.61: Επεξήγηση του Widget Exercises Mode Select

Στο *Event Graph* του *Widget* γράφτηκε ο παρακάτω κώδικας.



Εικόνα 4.62: Event Graph του Widget Exercises Mode Select”

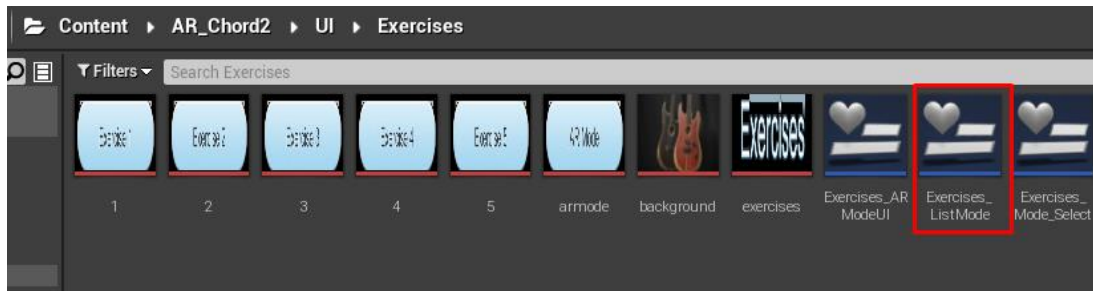
Ο κόμβος *On clicked (List\_Mode\_Button)* εκτελεί το κομμάτι κώδικα που είναι συνδεδεμένο σε αυτόν μόλις πατηθεί το κουμπι “List Mode”. Ο κόμβος *Remove from Parent* αφαιρεί από την οθόνη το παρόν *Widget* και με τους κόμβους *Create Widget* και *Add to Viewport* εμφανίζεται το *Widget* “Exercises List Mode”, το οποίο θα εξηγηθεί παρακάτω.

Ο κόμβος *On clicked (AR\_Mode\_Button)* εκτελεί το κομμάτι κώδικα που είναι συνδεδεμένο σε αυτόν μόλις πατηθεί το κουμπι “AR Mode”. Ο κόμβος *Open Level* ανοίγει το *Level* “Exercises\_ARMode”, το οποίο θα εξηγηθεί παρακάτω.

Ο κόμβος *On clicked (Back\_Button)* εκτελεί το κομμάτι κώδικα που είναι συνδεδεμένο σε αυτόν μόλις πατηθεί το κουμπι “Back”. Ο κόμβος *Open Level* ανοίγει το *Level* “Main\_Menu” και μεταφέρει τον χρήστη στο αρχικό μενού.

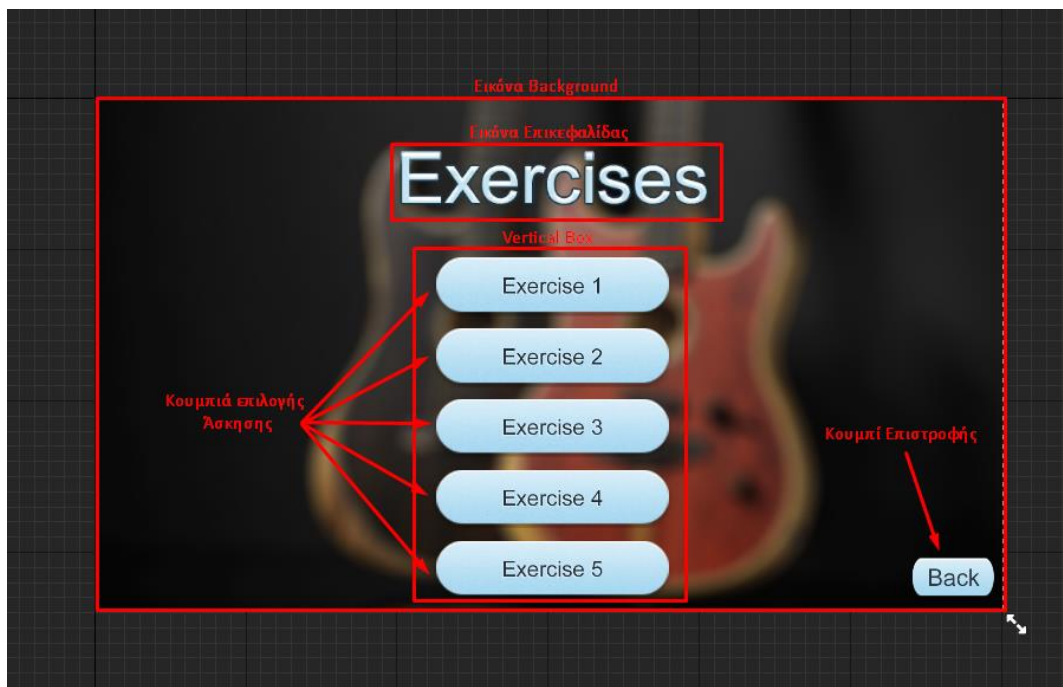
## 4.2.η Δημιουργία Widget για το List Mode

Δημιουργήθηκε *Widget Blueprint* με όνομα “Exercises List Mode”.



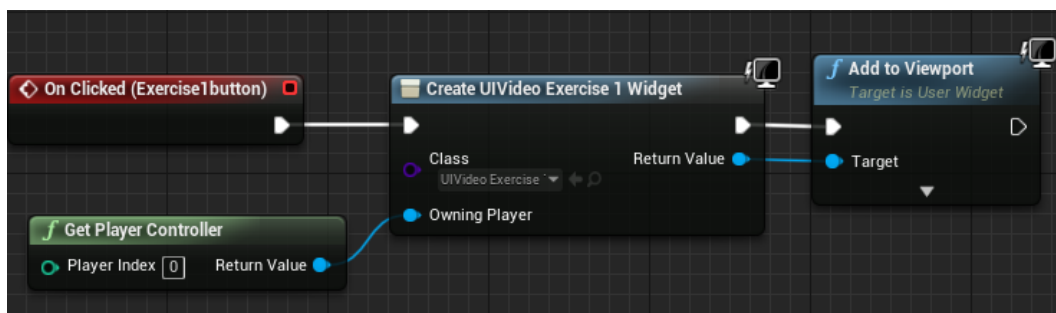
Εικόνα 4.63: Το Widget Exercises List Mode στο Content Browser

Σε αυτό έγινε εισαγωγή εικόνας background, εικόνας επικεφαλίδας, πέντε κουμπιών για επιλογή μεταξύ των ασκήσεων, ένα Vertical Box το οποίο περιέχει τα κουμπιά των ασκήσεων και ενός κουμπιού επιστροφής στο αρχικό μενού (Back).



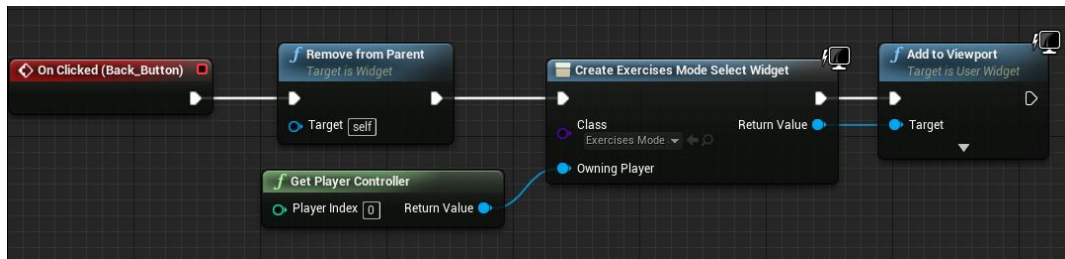
Εικόνα 4.64: Επεξήγηση του Widget Exercises List Mode

Στο *Event Graph* του *Widget* γράφτηκε ο παρακάτω κώδικας για κάθε ένα από τα κουμπιά επιλογής άσκησης ο οποίος εμφανίζει το αντίστοιχο widget όπου γίνεται η αναπαραγωγή του βίντεο της άσκησης.



Εικόνα 4.65: Ο κώδικας για το πάτημα του κουμπιού Exercise 1

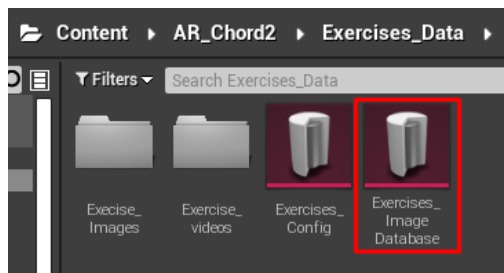
Επίσης γράφτηκε ο παρακάτω κώδικας για την εμφάνιση του *Widget* “Exercises Mode Select” μόλις πατηθεί το κουμπί “Back”.



Εικόνα 4.66: Ο κώδικας για το πάτημα του κουμπιού Back

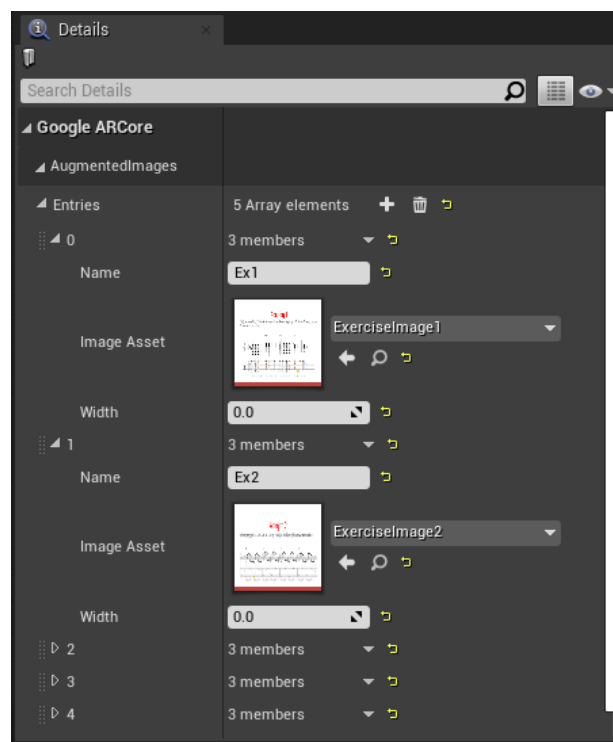
#### 4.2.θ Δημιουργία Βάσης Δεδομένων Επαυξημένων Εικόνων & AR Session Config για το AR Mode

Δημιουργήθηκε *Data Asset* τύπου *GoogleARCoreAugmentedImageDatabase* με όνομα “Exercises\_ImageDatabase”.



Εικόνα 4.67 Η Βάση Δεδομένων Επαυξημένων Εικόνων στο Content Browser του UE4

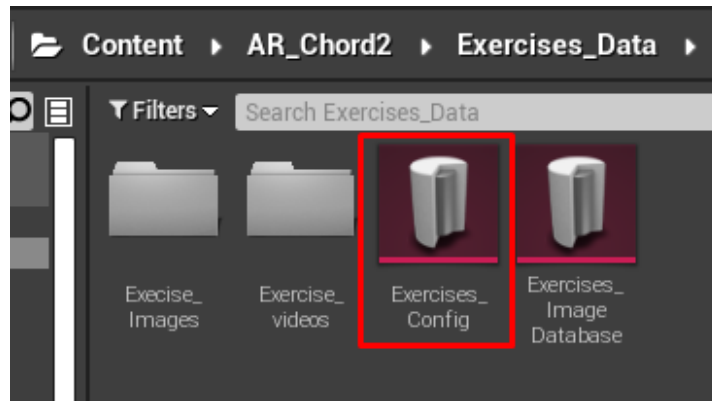
Σε αυτό εισήχθησαν οι εικόνες των εκφωνήσεων που δημιουργήθηκαν.



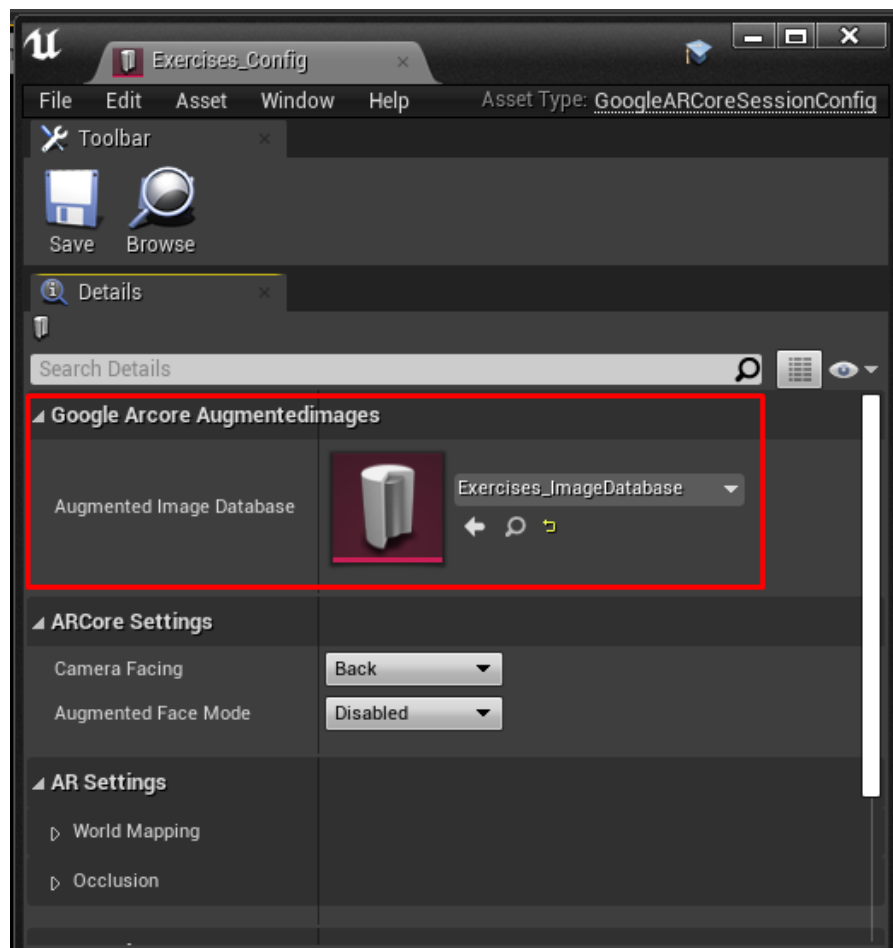
Εικόνα 4.68: Η Βάση Δεδομένων Επαυξημένων Εικόνων

Υπάρχουν 5 εικόνες που καταλαμβάνουν τις θέσεις 0-4 (όπως σε μια δομή δεδομένων τύπου πίνακα) και η κάθε εικόνα έχει το δικό της Index, το δικό της Όνομα και πληροφορίες για το πλάτος της στον φυσικό κόσμο.

Δημιουργήθηκε επίσης ένα *Data Asset* τύπου *GoogleARCoreSessionConfig* με όνομα “Exercises\_Config” στο οποίο φορτώθηκε η βάση δεδομένων “Exercises\_ImageDatabase”.



Εικόνα 4.69: Το Αρχείο Exercises\_Config στο Content Browser του UE4

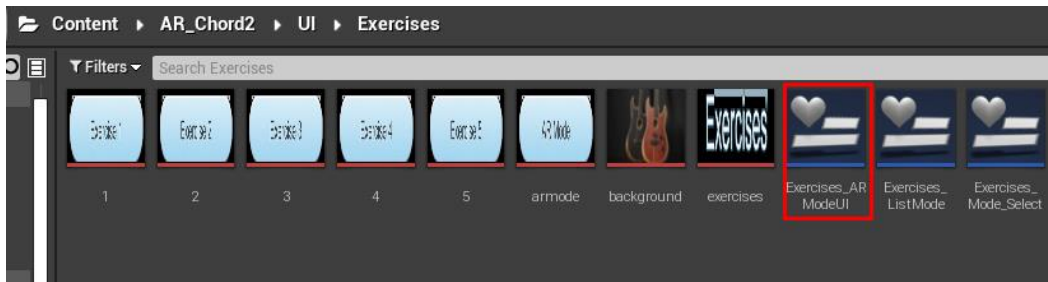


Εικόνα 4.70: Το Αρχείο Exercises\_Config

Το συγκεκριμένο *Data Asset* θα περάσει σαν είσοδος στο *ARCore* ώστε να ξεκινήσει μια AR συνεδρία παρέχοντας του όλα τα δεδομένα που χρειάζεται.

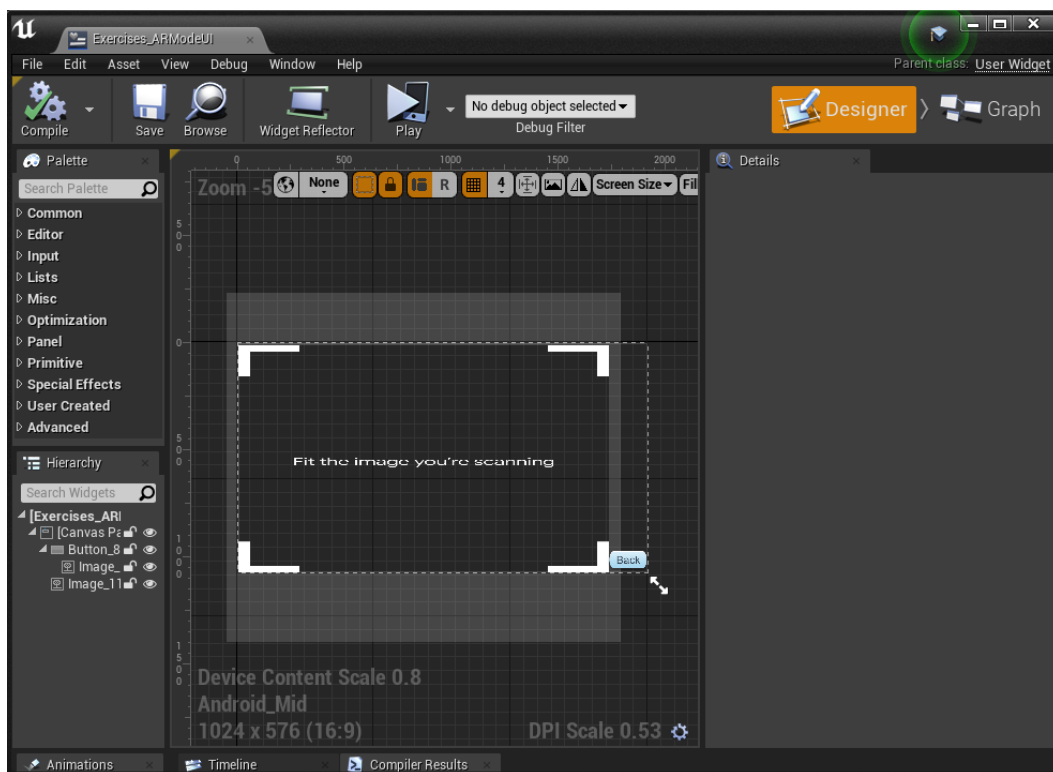
## 4.2.1 Δημιουργία Widget για το AR Mode

Δημιουργήθηκε ένα *Widget Blueprint* με το όνομα “Exercises\_ARModeUI”:



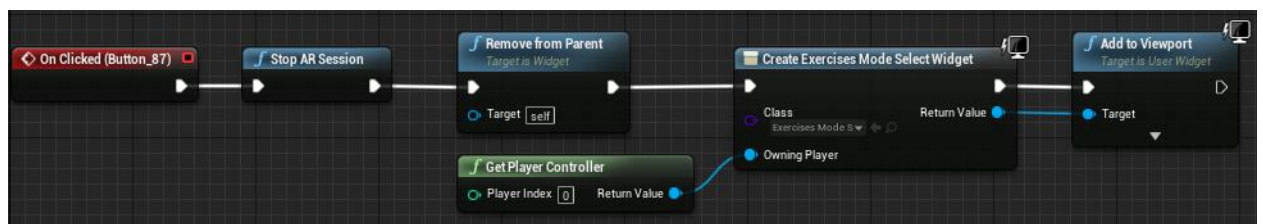
Εικόνα 4.71: Το Widget Exercises AR Mode στο Content Browser

Σε αυτό προστέθηκε μια εικόνα “FitToScan” και ένα κουμπί “Back”.



Εικόνα 4.72: Το Widget Exercises AR Mode

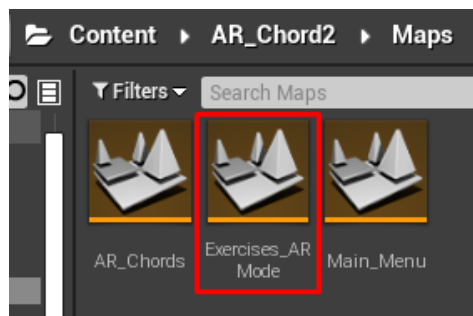
Ο χρήστης πατώντας το κουμπί “Back” επιστρέφει στο *Widget* “Exercises Mode Select” αφού γίνει διακοπή της AR συνεδρίας. Αυτό επιτυγχάνεται μέσω της συνάρτησης *Stop AR Session*, για την διακοπή της συνεδρίας, και μέσω των κόμβων *Remove from Parent*, *Create Widget* και *Add to Viewport*, για την μετάβαση στο *Widget*:



Εικόνα 4.73: Ο κώδικας για το πάτημα του κουμπιού Back

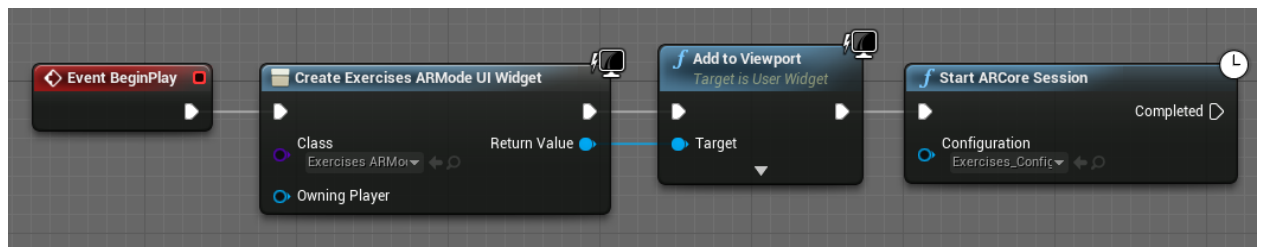
## 4.2.1α Δημιουργία Level για το AR Mode

Στο Project του προγράμματος Unreal Engine 4 δημιουργήθηκε νέο *Level* με όνομα “Exercises\_ARMode”.



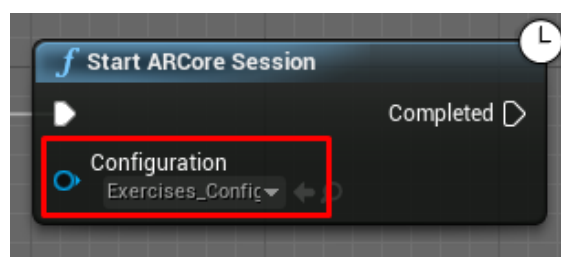
Εικόνα 4.74: Το Level "Exercises AR Mode" στο Content Browser στο UE4

Στο *Level Blueprint* του “Exercises\_ARMode” υλοποιήθηκε ο παρακάτω κώδικας.



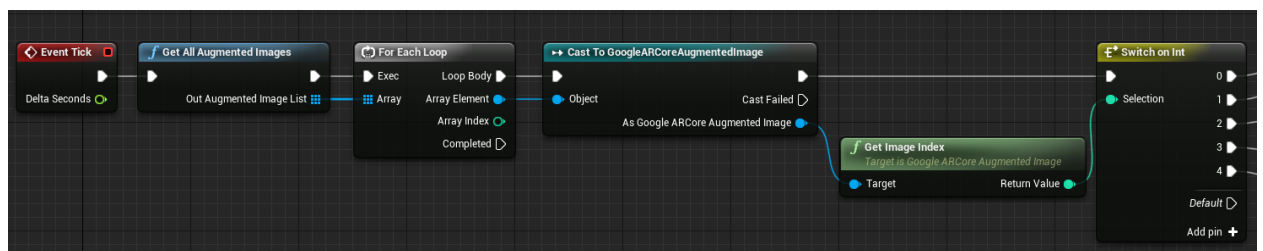
Εικόνα 4.75: Κώδικας για Εμφάνιση Widget & Έναρξη AR Συνεδρίας

Ξεκινάμε με τον κόμβο *Event Begin Play* που σημαίνει ότι αυτό που ακολουθεί θα ενεργοποιηθεί όταν ξεκινήσει το *Level*. Με τους κόμβους *Create Widget* και *Add to Viewport* θα εμφανιστεί στην οθόνη το *Widget* που μόλις υλοποιήθηκε. Τέλος, με τον κόμβο *Start ARCore Session* ξεκινάει η AR συνεδρία, έχοντας δώσει σαν εισοδο το αρχείο “Exercises\_Config” που αναφέρθηκε παραπάνω. Ενεργοποιείται, δηλαδή, η κάμερα της κινητής συσκευής και “ψάχνει” για τις εικόνες που έχουμε ορίσει στη Βάση Δεδομένων του “Exercises\_Config”.



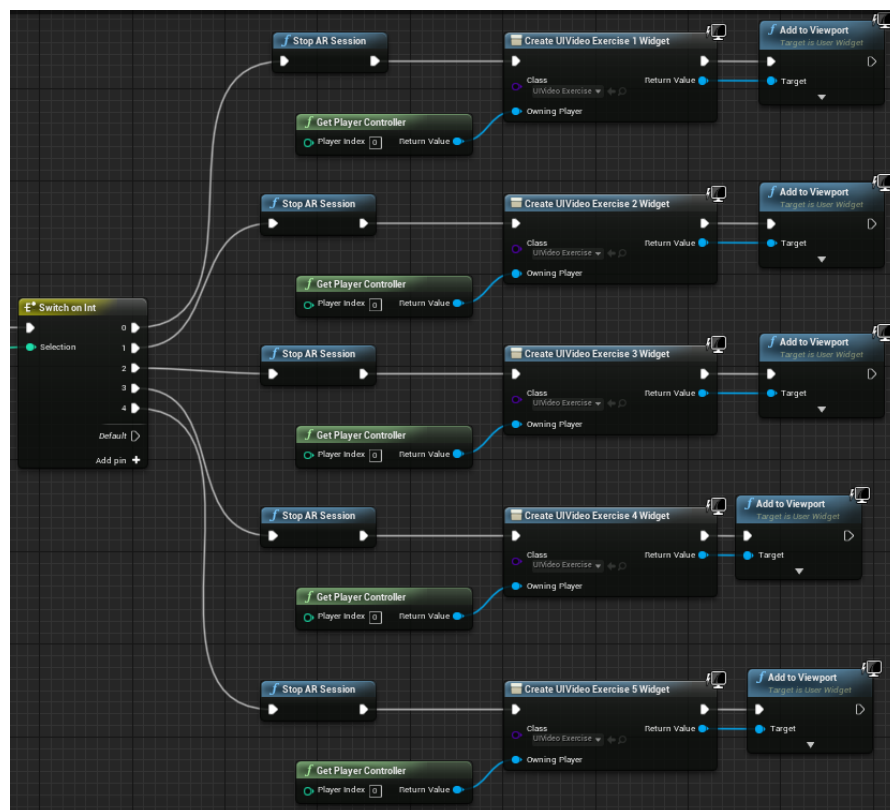
Εικόνα 4.76: Ο κόμβος Start ARCore Session

Για την αναγνώριση της εικόνας απαιτείται ο παρακάτω κώδικας:



Εικόνα 4.77: Ο Κώδικας για την αναγνώριση εικόνας

Εδώ ξεκινάμε με τον κόμβο *Event Tick* που σημαίνει πως ότι ακολουθεί θα επαναλαμβάνεται σε κάθε frame, δηλαδή σε κάθε καρέ που λαμβάνει η κάμερα της κινητής συσκευής. Ο κόμβος *Get All Augmented Images* θα συγκεντρώσει (αν υπάρχουν) τις εικόνες της Βάσης Δεδομένων που βρίσκονται στο καρέ και θα τις εισάγει σε έναν πίνακα. Σε αυτό το σημείο, τα αναγνωριστικά των εικόνων του πίνακα που εξάγεται δεν ταυτίζονται με τα αναγνωριστικά της Βάσης Δεδομένων που ορίσαμε. Π.χ. αν αναγνωρίστηκε η εικόνα που βρίσκεται στη θέση 3 της Βάσης Δεδομένων με όνομα 'Exe4', στον πίνακα θα αποθηκευτεί στην θέση 0 με ένα όνομα το οποίο θα παραχθεί τυχαία από το πρόγραμμα. Ο κόμβος *For Each Loop* λαμβάνει σαν είσοδο τον πίνακα που εξήχθη από τον προηγούμενο κόμβο, και για κάθε στοιχείο του, θα επαναλάβει το κομμάτι κώδικα που συνδέεται στο *Loop Body*. Για παράδειγμα, αν εξαχθεί πίνακας με τρεις εικόνες, το κομμάτι του κώδικα δεξιά του *For Each Loop* θα εκτελεστεί τρεις φορές και κάθε φορά θα παρέχει στον επόμενο κόμβο το αντίστοιχο στοιχείο του πίνακα μέσω του πεδίου *Array Element*. Ο κόμβος *Cast To GoogleARCoreAugmentedImage* θα μεταφράσει τα αναγνωριστικά της εικόνας που έδωσε ο κόμβος *Get All Augmented Images* στα αναγνωριστικά που ορίσαμε στη Βάση Δεδομένων. Χρησιμοποιώντας το προηγούμενο παράδειγμα, η εικόνα στην θέση 0 με την τυχαία ονομασία, τώρα θα έχει index την τιμή 3 και όνομα 'Ex4'. Με αυτόν τον τρόπο πλέον μπορούμε να γνωρίζουμε πραγματικά ποια εικόνα αναγνωρίστηκε απ' την κάμερα. Ο κόμβος *Get Image Index* λαμβάνει το index της εικόνας και το τροφοδοτεί στον επόμενο κόμβο. Στο παράδειγμά μας θα τροφοδοτήσει την τιμή 3. Ο τελευταίος κόμβος *Switch on Int* λειτουργεί όπως μια δομή επιλογής Switch με βάση έναν integer. Ανάλογα με τον integer που θα λάβει, θα επιλέξει την αντίστοιχη διαδρομή. Εφόσον έχουμε 5 εικόνες προς αναγνώριση, μπορεί να λάβει τις τιμές από 0 έως 4 και υπάρχουν αντίστοιχα 5 διαφορετικές διαδρομές στις οποίες θα εκτελεστεί η εμφάνιση του κατάλληλου Widget, δηλαδή η αναπαραγωγή του αντίστοιχου βίντεο της εκφώνησης που αναγνωρίστηκε αφού πρώτα γίνει διακοπή της AR συνεδρίας με τον κόμβο *Stop AR Session*. Για την εμφάνιση του Widget χρησιμοποιήθηκε ο ίδιος κώδικας με το List Mode.



Εικόνα 4.78: Ο κώδικας για την εμφάνιση του κατάλληλου Widget

## 4.3 Η Λειτουργία «Tuner»

### 4.3.α Τι είναι το Κουρδιστήρι και ποιος ο Σκοπός του

Σε ένα από τα δύο άκρα της χορδής της κιθάρας υπάρχει ένας μηχανισμός, ο οποίος ονομάζεται κλειδί και αυξομειώνει την τάση της χορδής. Η αυξομείωση της τάσης της χορδής επηρεάζει την νότα (δηλαδή τη συχνότητα ήχου) που παράγεται όταν η χορδή πάλλεται. Με την αύξηση της τάσης της χορδής επιτυγχάνεται μετάβαση σε υψηλότερη νότα (αύξηση της συχνότητας του ήχου) και με την μείωση της τάσης, μετάβαση σε χαμηλότερη νότα (μείωση της συχνότητας του ήχου). Ο μουσικός, για να μπορέσει να εκτελέσει σωστά τις μουσικές συνθέσεις και τις ασκήσεις, θα πρέπει να έχει την κιθάρα του κουρδισμένη, δηλαδή να έχει ρυθμίσει τα κλειδιά της κιθάρας με τέτοιο τρόπο, ώστε η κάθε χορδή, όταν πάλλεται, να παράγει μία συγκεκριμένη νότα. Το ποια νότα αντιστοιχεί σε κάθε χορδή, εξαρτάται από το κούρδιομα που έχει επιλέξει ο μουσικός. Υπάρχουν πολλά διαφορετικά κούρδια και το δημοφιλέστερο ονομάζεται “E Standard”.

Δυστυχώς, η διαδικασία του κούρδιαματος, χωρίς κάποιο βοήθημα, μπορεί να επιτευχθεί μόνο από έμπειρους μουσικούς, οι οποίοι μπορούν και αναγνωρίζουν την νότα που παράγεται από την χορδή όταν πάλλεται και ρυθμίζουν το κλειδί της χορδής στο επιθυμητό σημείο.

Οι υπόλοιποι μουσικοί, και κυρίως οι αρχάριοι, χρειάζονται για αυτή τη διαδικασία ένα εργαλείο που ονομάζεται κουρδιστήρι, το οποίο αναπαράγει την επιθυμητή νότα της κάθε χορδής. Με αυτό το τρόπο, ο μουσικός ρυθμίζει το κλειδί της χορδής που θέλει να κούρδισει, μέχρι η χορδή να παράγει την ίδια νότα με την νότα που παράγει το κουρδιστήρι. Το κουρδιστήρι παρέχει δηλαδή στον μουσικό ένα τονικό σημείο αναφοράς.



Εικόνα 4.79



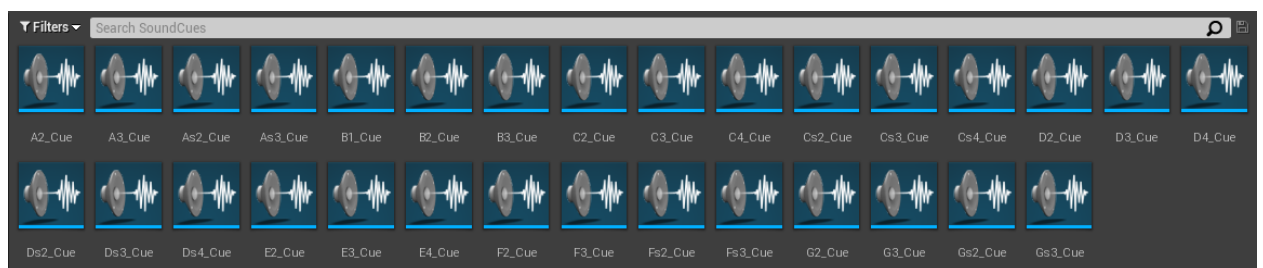
### 4.3.β Εύρεση ήχων νοτών, εισαγωγή στο UE4 & Δημιουργία Sound Cue

Για την υλοποίηση του Tuner βρέθηκαν σε αποθετήριο ήχων όλες οι νότες που απαιτούνται (30 στο σύνολο) σε μορφή .wav. Έγινε εισαγωγή των νοτών στο Unreal Engine 4 και δόθηκαν τα κατάλληλα ονόματα: *ΛατινικόΌνομαΝότας\_ΑριθμόςΟκτάβας* για τις καθαρές νότες και *ΛατινικόΌνομαΝότας\_s\_ΑριθμόςΟκτάβας* για τις αλλοιωμένες (Π.χ. η νότα Λα3 ονομάστηκε “A3” και η νότα “Λα#3” ονομάστηκε “As3”).



Εικόνα 4.80: Τα Sound Waves των νοτών στο Content Browser του προγράμματος UE4

Από αυτά τα αρχεία και με την επιλογή *Create Cue* δημιουργήθηκε ένα *Sound Cue* για την κάθε νότα με ίδιο όνομα αλλά με την κατάληξη *\_Cue* (Π.χ. το *Sound Cue* της νότας Λα3 ονομάστηκε A3\_Cue).

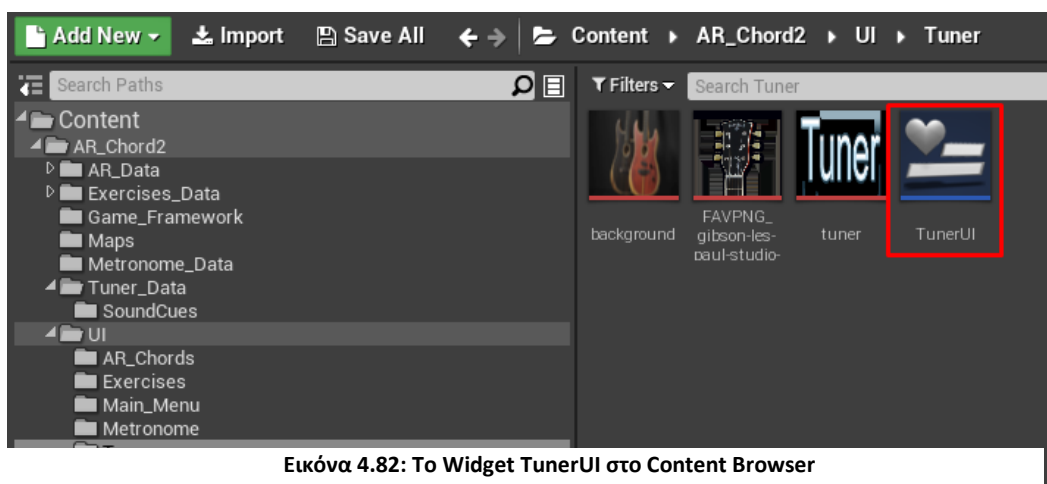


Εικόνα 4.81: Τα Sound Cues των νοτών στο Content Browser του προγράμματος UE4

### 4.3.γ Δημιουργία Widget για το «Tuner»

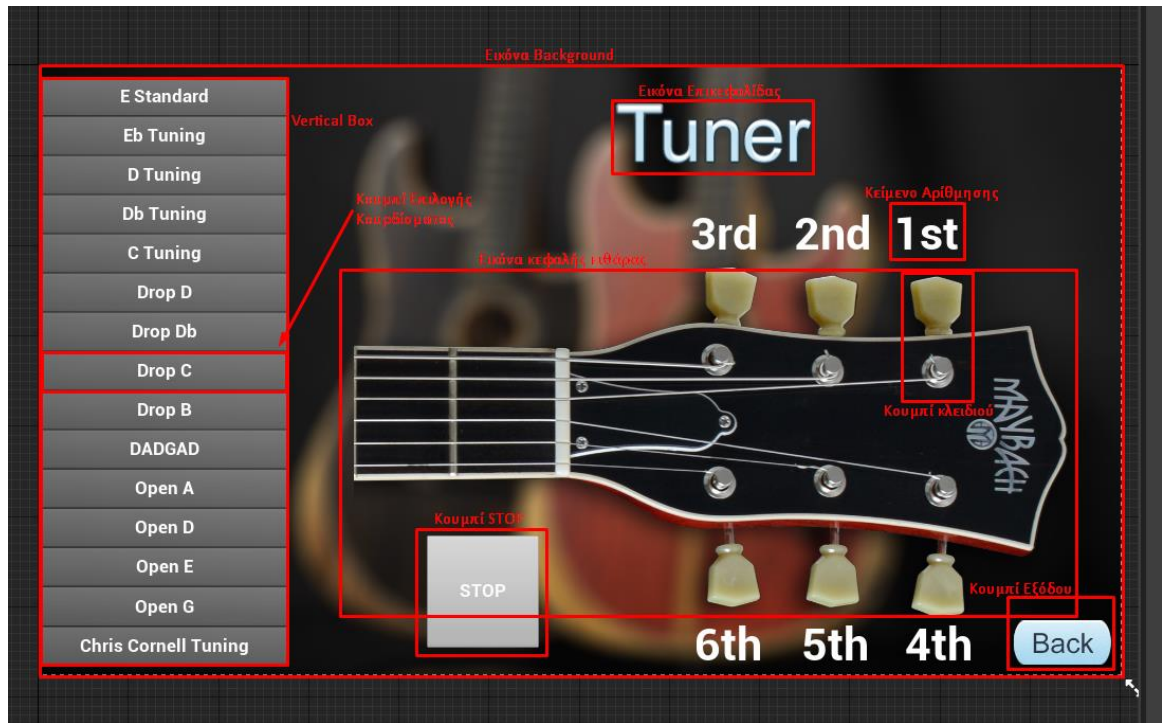
Το κουρδιστήρι μπορεί να υλοποιηθεί εξ' ολοκλήρου μέσω *Widget Blueprint* και δεν απαιτείται η δημιουργία νέου Level. Ο κώδικας γράφτηκε στο *Event Graph* του *Widget Blueprint*.

Δημιουργήθηκε *Widget Blueprint* με Όνομα “TunerUI”.



Εικόνα 4.82: Το Widget TunerUI στο Content Browser

Σε αυτό έγινε εισαγωγή εικόνας Background, εικόνας Επικεφαλίδας, εικόνας κεφαλής κιθάρας, έξι κειμένων για την αρίθμηση των κλειδιών, ένα κουμπι για κάθε κλειδί, κουμπι παύσης αναπαραγωγής νότας (Stop), δεκαπέντε κουμπιά επιλογής κουρδίσματος, Vertical Box το οποίο περιέχει τα κουμπιά επιλογής κουρδίσματος και κουμπι εξόδου από το κουρδιστήρι (Back).



Εικόνα 4.83: Επεξήγηση του Widget TunerUI



Εικόνα 4.84: Το Widget TunerUI

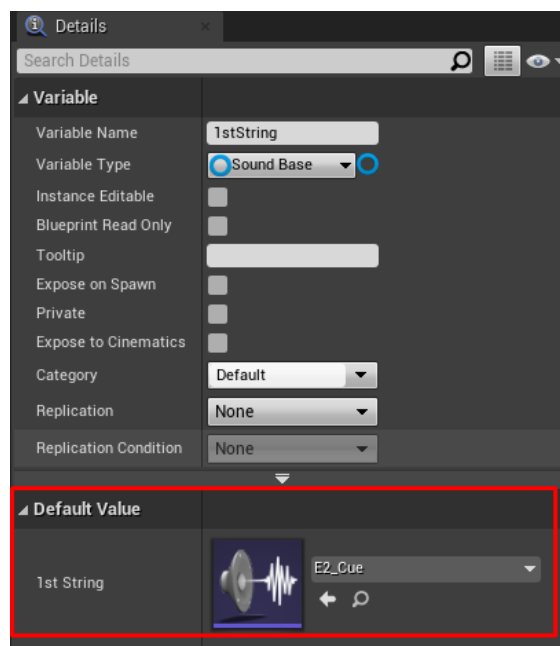
#### 4.3.6 Event Graph του «Tuner»

Αρχικά, δημιουργήθηκαν έξι μεταβλητές (μία για κάθε χορδή) τύπου *Sound Base*, με όνομα “1stString”, “2ndString”, ..., “6thString”.



Εικόνα 4.85: Οι έξι μεταβλητές τύπου *Sound Base*

Σε αυτές αρχικοποιήθηκαν τα *Sound Cues* των νοτών που αντιστοιχούν στο κούρδισμα “E Standard”.



Εικόνα 4.86: Λεπτομέρειες μεταβλητής 1st String

Στη συνέχεια γράφτηκε κώδικας για να αποδίδονται στις έξι μεταβλητές τα αντιστοίχα *Sound Cues* κάθε φορά που ο χρήστης επιλέγει να αλλάξει κούρδισμα πατώντας σε ένα από τα διαθέσιμα κουρδίσματα.

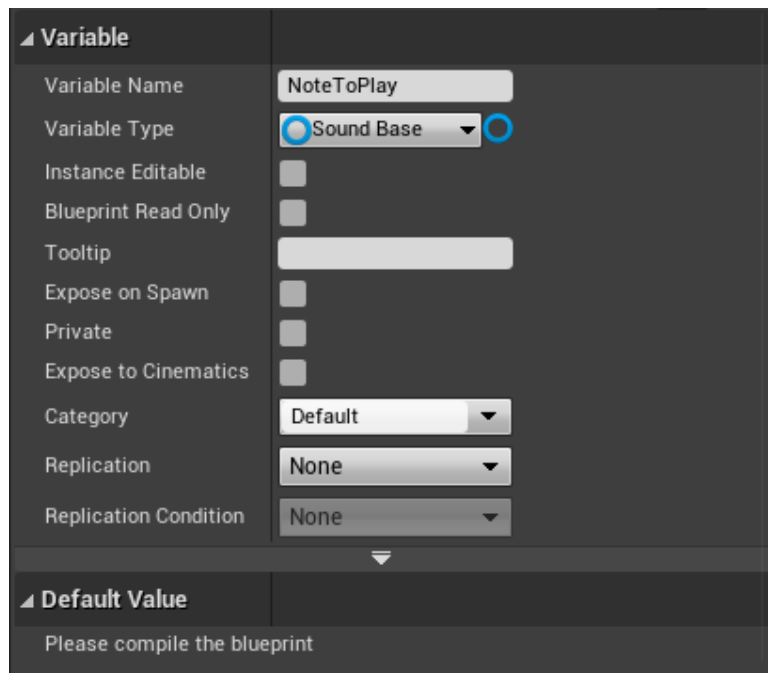
Για παράδειγμα, αν ο χρήστης επιλέξει το κούρδισμα “D Tuning” θα εκτελεστεί ο παρακάτω κώδικας:



Εικόνα 4.87: Ο κώδικας για το πάτημα του κουμπιού D Tuning

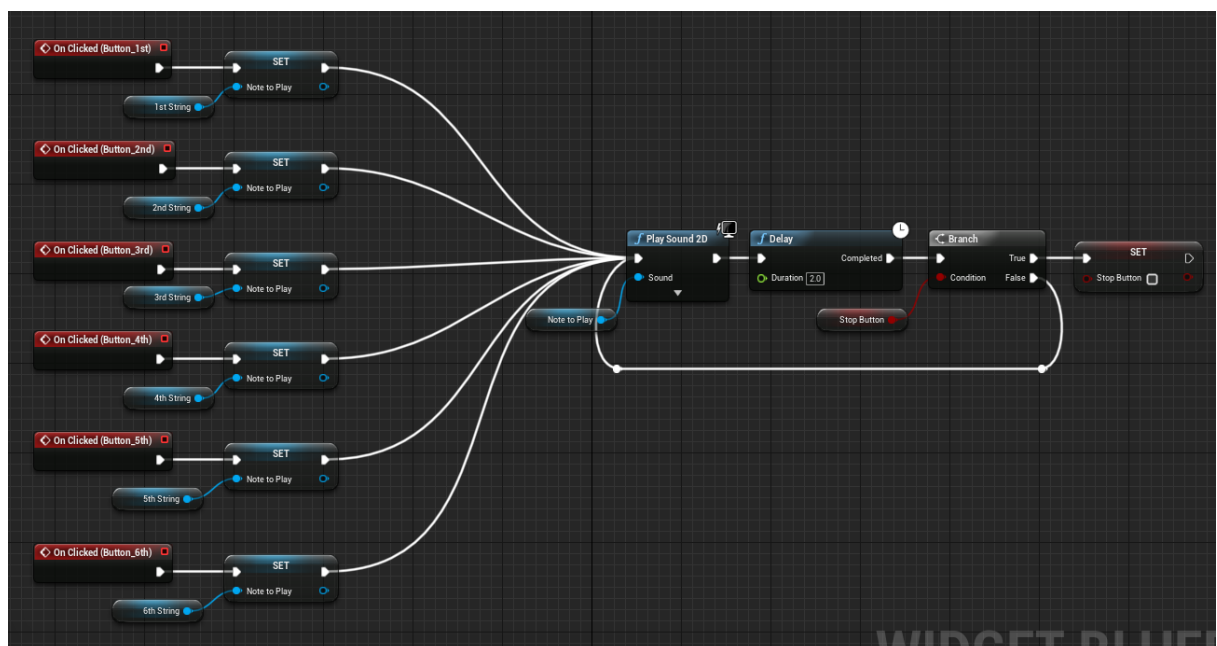
Ο κόμβος *On Clicked (Button\_D)* εκτελεί τον κώδικα που είναι συνδεδεμένος σε αυτόν, μόλις πατηθεί το κουμπί “D Tuning”. Στη συνέχεια, υπάρχουν έξι κόμβοι *SET*, οι οποίοι εκχωρούν στις έξι μεταβλητές τα *Sound Cues* που αντιστοιχούν στις νότες του συγκεκριμένου κουρδίσματος. Τέλος, υπάρχει ένας ακόμη κόμβος *SET*, ο οποίος θα εξηγηθεί παρακάτω. Γράφτηκε αντιστοιχος κώδικας για κάθε ένα από τα διαθέσιμα κουρδίσματα.

Στη συνέχεια, δημιουργήθηκε μεταβλητή τύπου *Sound Base* με όνομα “NoteToPlay”.



Εικόνα 4.88: Λεπτομέρειες της μεταβλητής Note to Play

Ο παρακάτω κώδικας εκτελεί την αναπαραγωγή και παύση των νοτών:



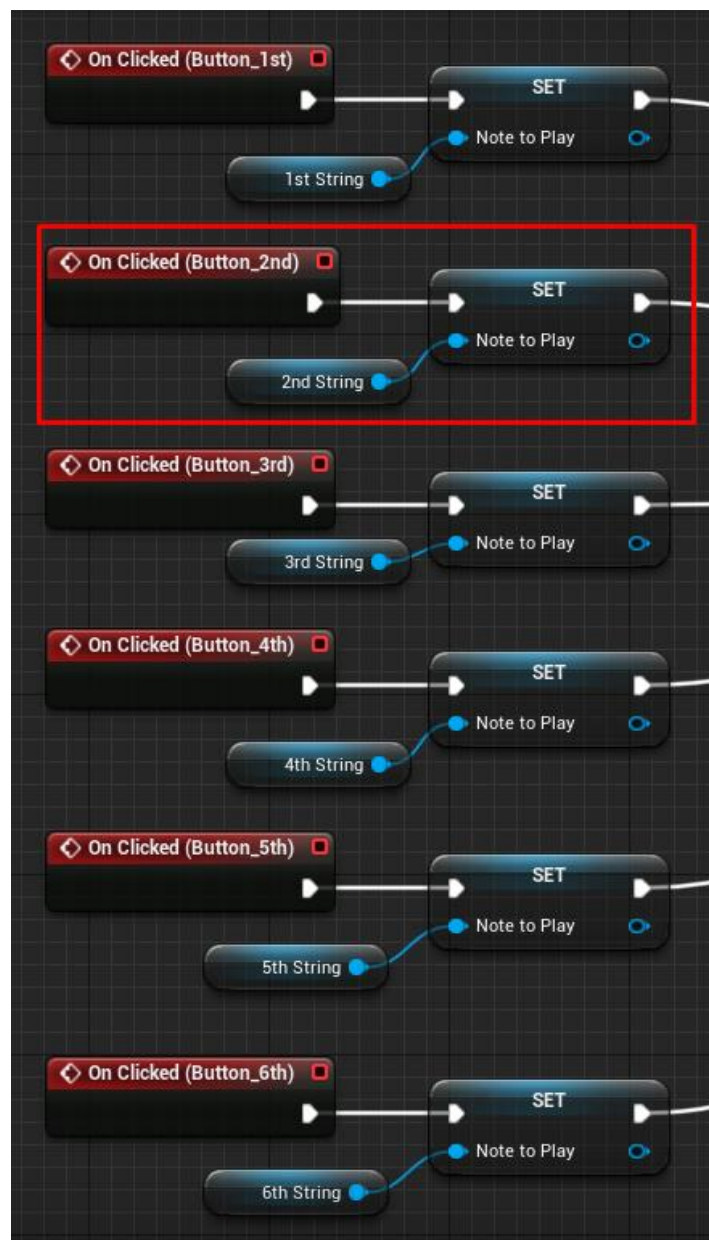
Εικόνα 4.89: Ο κώδικας για την αναπαραγωγή της κατάλληλης νότας

Στο πρώτο μέρος, έχουμε την εκχώρηση στην μεταβλητή “NoteToPlay” της νότας της χορδής που επέλεξε ο χρήστης πατώντας το αντίστοιχο κουμπι.

Για παράδειγμα, αν ο χρήστης έχει επιλέξει το κούρδισμα “D Tuning” και πατήσει την δεύτερη χορδή, στη μεταβλητή “NoteToPlay” θα εκχωρηθεί το *Sound Cue* που είχε εκχωρηθεί στην μεταβλητή “2ndString” επιλέγοντας το κούρδισμα “D Tuning”, δηλαδή το *Sound Cue* “G2\_Cue” που αντιστοιχεί στη νότα G2 (Σολ2).

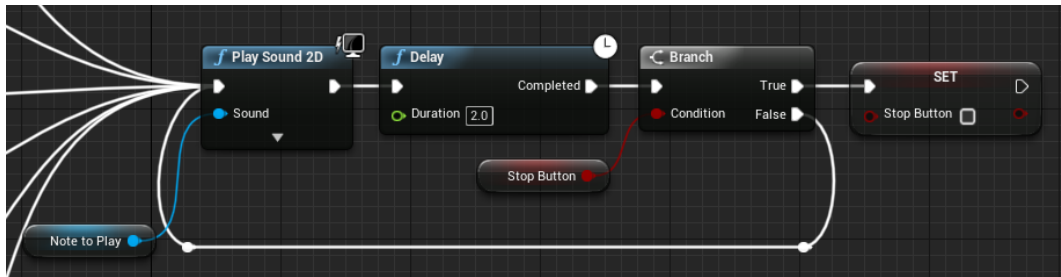


Εικόνα 4.90: Στην μεταβλητή 2nd String εκχωρείται ο ήχος G2Sound



Εικόνα 4.91: Στη μεταβλητή Note to Play εκχωρείται ο ήχος G2Sound

Σε δεύτερο μέρος, έχουμε την αναπαραγωγή του ήχου της νότας.



Εικόνα 4.92: Ο κώδικας αναπαραγωγής της νότας

Υπάρχει ο κόμβος *Play Sound 2D*, ο οποίος αναπαράγει την νότα που βρίσκεται στην μεταβλητή “NoteToPlay” και στη συνέχεια ο κόμβος *Delay* ο οποίος καθυστερεί την εκτέλεση του κώδικα κατά δύο δευτερόλεπτα. Στη συνέχεια, υπάρχει μια *Branch* η οποία ελέγχει αν η μεταβλητή “Stop Button” τύπου Boolean που δημιουργήθηκε περιέχει τιμή True ή False. Αν περιέχει τιμή False θα ξαναγυρίσει η ροή στον κόμβο *Play Sound 2D* και θα γίνει επανάληψη της αναπαραγωγής της νότας. Η αναπαραγωγή της νότας θα συνεχίσει επ’ άπειρον μέχρι να πατηθεί από τον χρήστη το κουμπι “STOP”, όπου θα εκτελεστεί κώδικας που θα εκχωρήσει στην μεταβλητή “Stop Button” την τιμή True, όπως φαίνεται παρακάτω.



Εικόνα 4.93: Ο κώδικας για το πάτημα του κουμπιού Stop

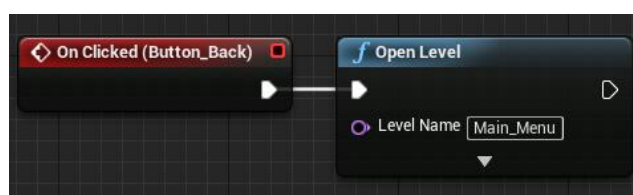
Σε περίπτωση που την στιγμή που αναπαράγεται νότα ενός κουρδίσματος ο χρήστης επιλέξει διαφορετικό κούρδισμα, θα πρέπει να γίνει παύση της αναπαραγωγής. Για αυτό το λόγο, υπάρχει κόμβος εκχώρησης τιμής True στην μεταβλητή “Stop Button” στο τέλος του event πατήματος κουμπιού επιλογής κουρδίσματος.



Εικόνα 4.94: Εκχώρηση τιμής True στην μεταβλητή Stop Button με κάθε πάτημα κουμπιού αλλαγής κουρδίσματος

Ο κόμβος *Delay* προστέθηκε για να υπάρχει χρονική απόσταση ανάμεσα στις διαδοχικές αναπαραγωγές.

Τέλος, αν πατηθεί το κουμπι Back ο χρήστης μεταφέρεται στο αρχικό μενού με τον κόμβο *Open Level*.



Εικόνα 4.95: Ο κώδικας για το πάτημα του κουμπιού Back

## 4.4 Η Λειτουργία «Metronome»

### 4.4.α Τι είναι ο Μετρονόμος και ποιος ο Σκοπός του

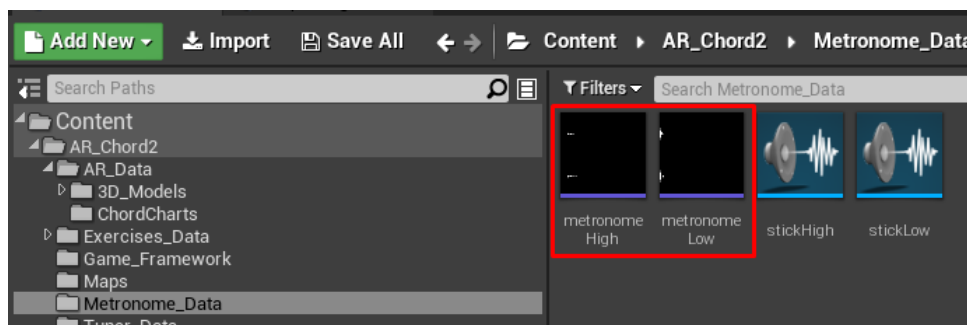
Σκοπός του μετρονόμου είναι να δίνει στον μουσικό ένα σταθερό ρυθμό για την εκτέλεση των συνθέσεων και των ασκήσεων. Αναπαράγει έναν χτύπο σε αυστηρά καθορισμένα χρονικά διαστήματα. Τα διαστήματα αυτά τα καθορίζει ο μουσικός ρυθμίζοντας την ταχύτητα του μετρονόμου (tempo). Η μονάδα μέτρησης που χρησιμοποιείται για το tempo είναι τα Beats per Minute (BPM) (Χτύποι ανά Λεπτό). Ο μετρονόμος αποτελεί απαραίτητο εργαλείο καθ' όλη τη διάρκεια εκμάθησης ενός μουσικού οργάνου και κατά την καθημερινή μελέτη.



Εικόνα 2.4 1

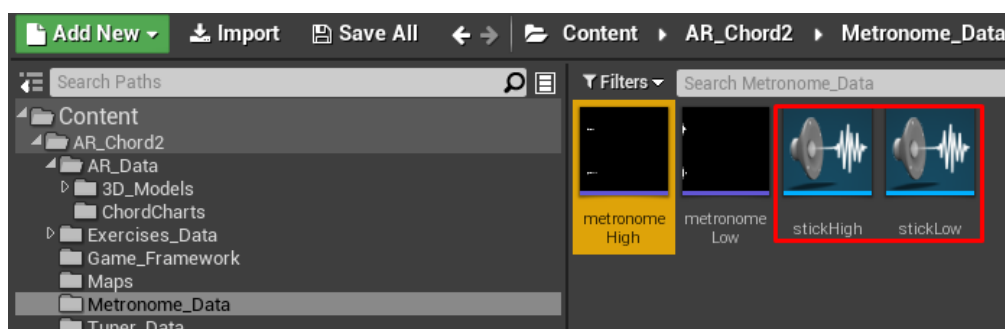
### 4.4.β Εύρεση ήχων χτύπων, εισαγωγή στο UE4 & Δημιουργία Sound Cue

Για την υλοποίηση του μετρονόμου βρέθηκαν σε αποθετήριο ήχων δύο ήχοι χτύπων σε μορφή .wav . Ο πρώτος χτύπος είναι τονικά χαμηλότερος από τον δεύτερο. Έγινε εισαγωγή των χτύπων στο Unreal Engine 4 και δόθηκαν τα κατάλληλα ονόματα: “MetronomeLow” για τον χαμηλότερο τονικά χτύπο και “MetronomeHigh” για τον υψηλότερο.



Εικόνα 4.96: Τα Sound Waves των χτύπων στο Content Browser του προγράμματος UE4

Από αυτά τα αρχεία και με την επιλογή *Create Cue* δημιουργήθηκε ένα *Sound Cue* για τον κάθε ήχο με ονόματα “StickLow” και “StickHigh”.

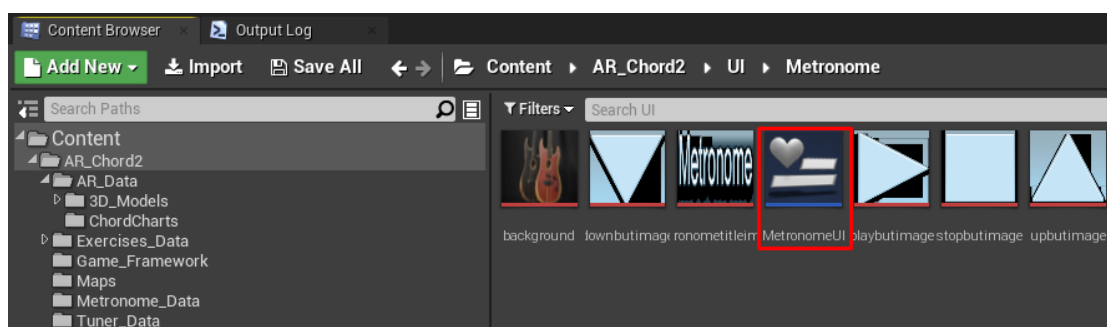


Εικόνα 4.97: Τα Sound Cues των χτύπων στο Content Browser του προγράμματος UE4

#### 4.4.γ Δημιουργία Widget για το «Metronome»

Ο μετρονόμος μπορεί να υλοποιηθεί εξ' ολοκλήρου μέσω *Widget Blueprint* και δεν απαιτείται η δημιουργία νέου Level. Ο κώδικας γράφτηκε στο *Event Graph* του *Widget Blueprint*.

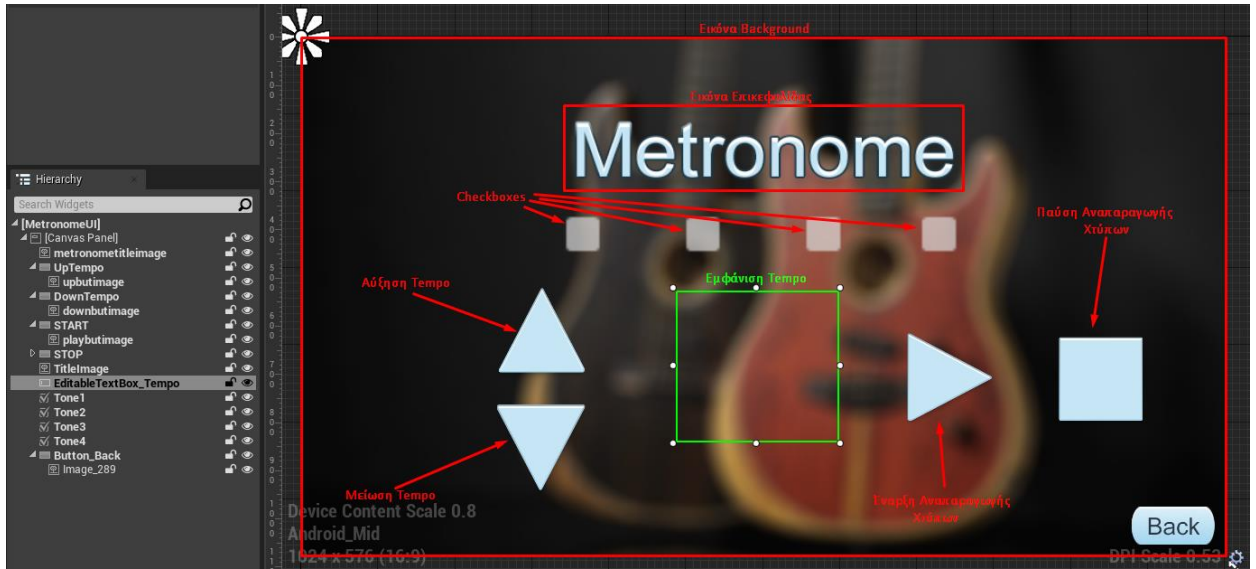
Δημιουργήθηκε *Widget Blueprint* με Όνομα “MetronomeUI”.



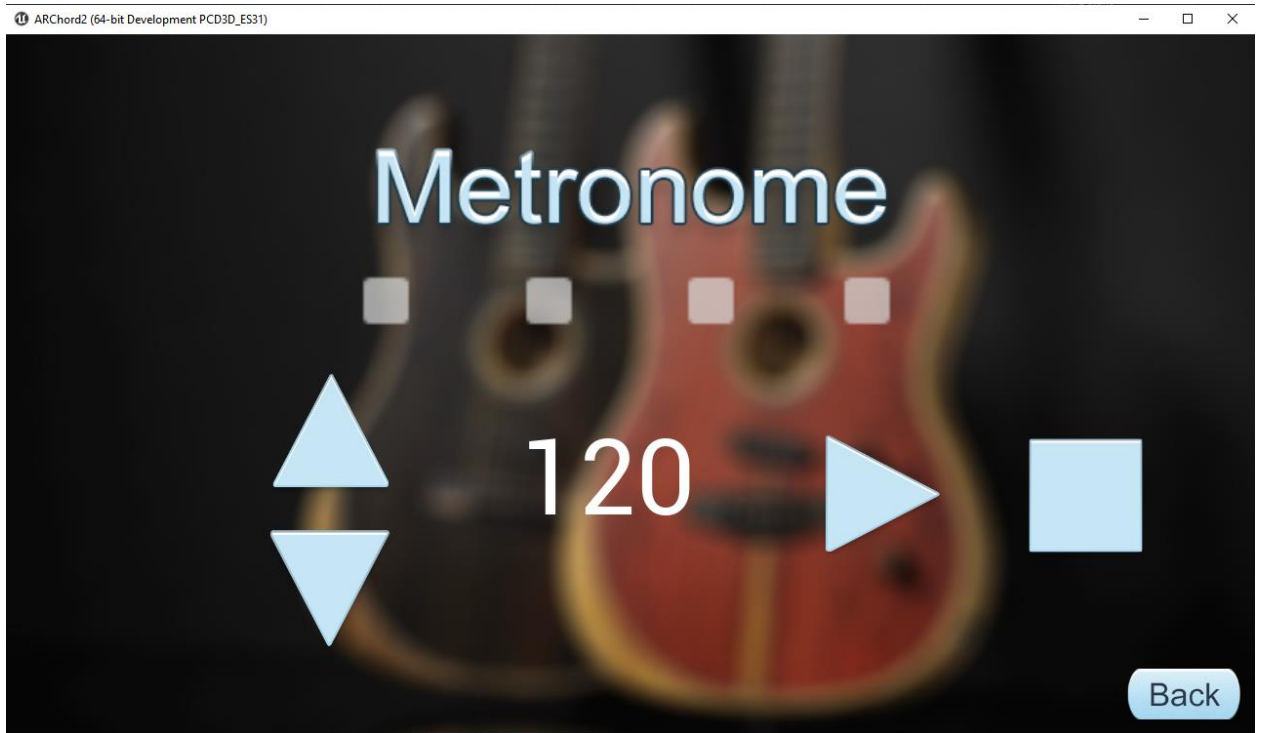
Εικόνα 4.98: Το Widget MetronomeUI στο Content Browser

Σε αυτό έγινε εισαγωγή εικόνας Background, εικόνας Επικεφαλίδας, τεσσάρων checkboxes, κουτιού κειμένου για την εμφάνιση του tempo (TextBox) και κουμπιών για: Έναρξη αναπαραγωγής χτύπων (Start), Παύση αναπαραγωγής χτύπων (Stop), Αύξηση Tempo (UpTempo), Μείωση Tempo (DownTempo) και έξοδο από τον μετρονόμο (Back).





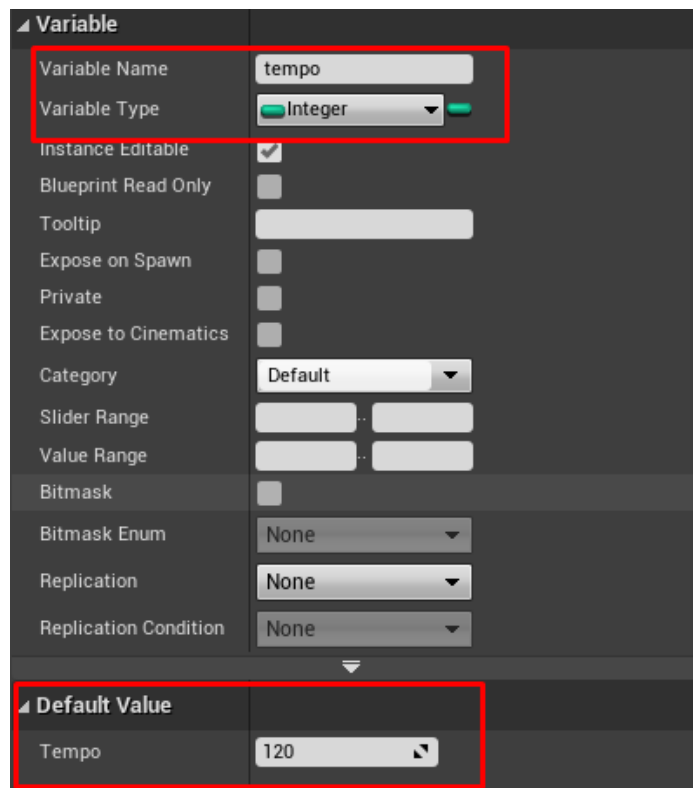
Εικόνα 4.99: Επεξήγηση του Widget MetronomeUI



Εικόνα 4.100: Το Widget MetronomeUI

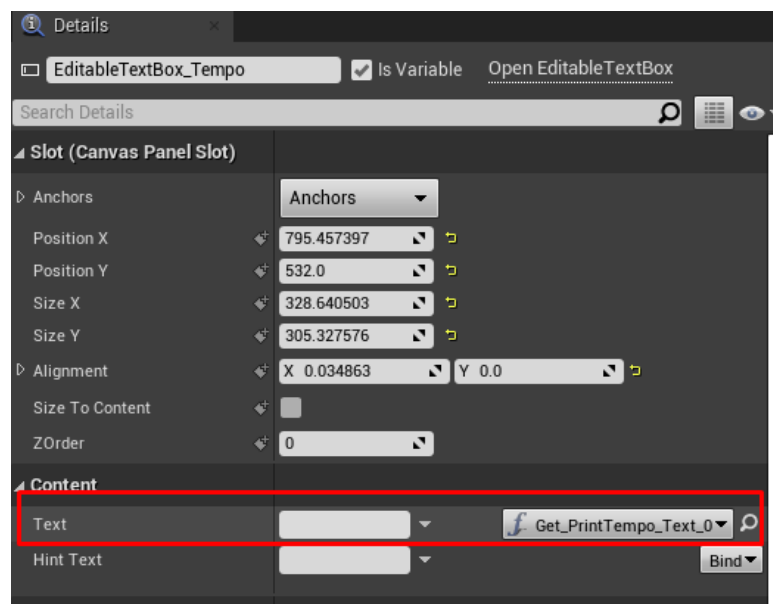
#### 4.4.6 Event Graph του «Metronome»

Αρχικά δημιουργήθηκε μία μεταβλητή τύπου Integer με όνομα tempo και με αρχική τιμή 120.



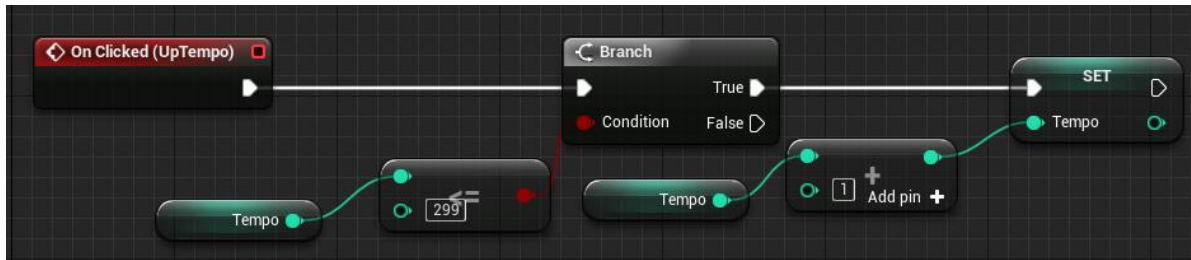
Εικόνα 4.101: Λεπτομέρειες μεταβλητής tempo

Έγινε δέσιμο (Bind) της μεταβλητής με το περιεχόμενο του κειμένου του *TextBox* ώστε να εμφανίζεται η τιμή της μεταβλητής στο κέντρο της οθόνης.



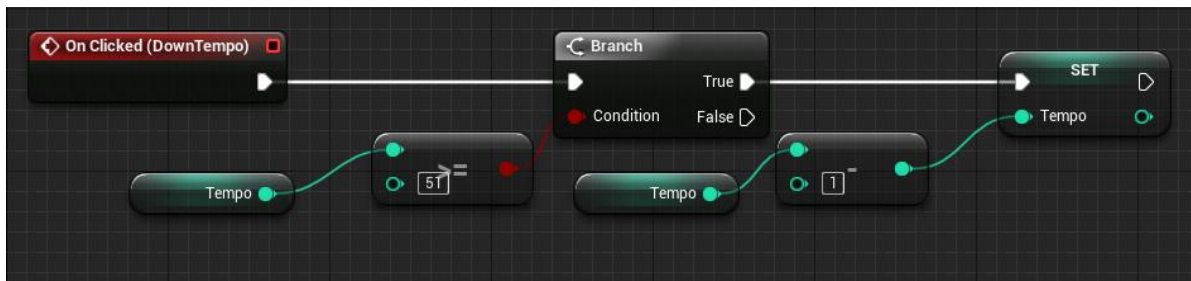
Εικόνα 4.102: Binding μεταβλητής tempo με TextBox

Η τιμή της μεταβλητής *Tempo* αυξάνεται και μειώνεται πατώντας τα κουμπιά *UpTempo* και *DownTempo* αντίστοιχα, λαμβάνοντας τιμές από 50 μέχρι 300.



Εικόνα 4.103: Ο κώδικας αύξησης του *tempo*

Ο παραπάνω κώδικας εκτελείται μόλις πατηθεί το κουμπι *UpTempo* μέσω του κόμβου *On Clicked(UpTempo)*. Ο κόμβος *Branch* ελέγχει αν η μεταβλητή *Tempo* περιέχει τιμή μικρότερη ή ίση του 299. Αν ισχύει, θα περάσει στον κόμβο *SET*, ο οποίος θα προσθέσει στο *Tempo* την τιμή 1 μέσω ενός αθροιστή. Αν δεν ισχύει (δηλαδή αν η τιμή του είναι 300), δεν εκτελείται τίποτα. Με αυτόν τον τρόπο επιτυγχάνεται η αύξηση του *Tempo*.

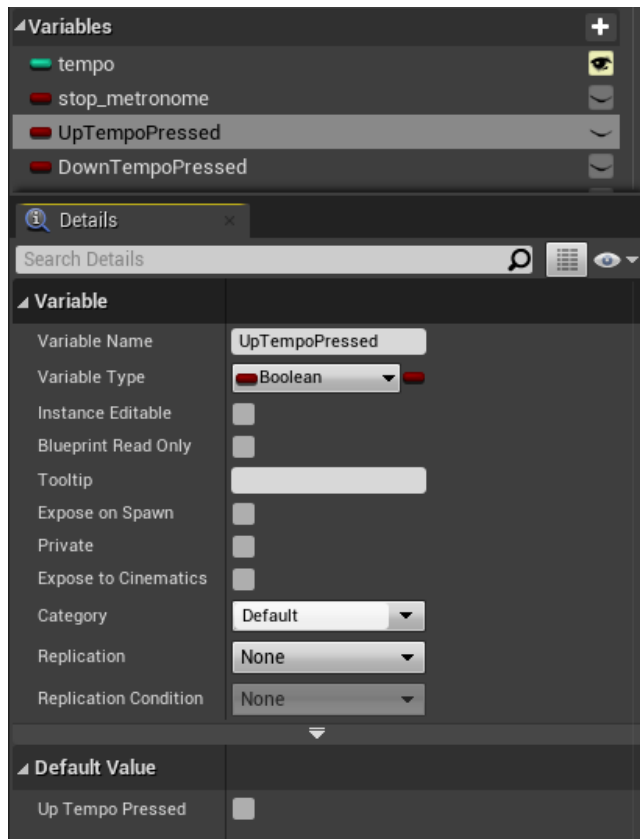


Εικόνα 4.24: Ο κώδικας μείωσης του *tempo*

Με τον ίδιο τρόπο επιτυγχάνεται η μείωση του *Tempo* με τη διαφορά ότι η *Branch* ελέγχει αν η τιμή είναι μεγαλύτερη ή ίση του 51 και αν ισχύει, η τιμή μειώνεται κατά 1 στον κόμβο *SET*.

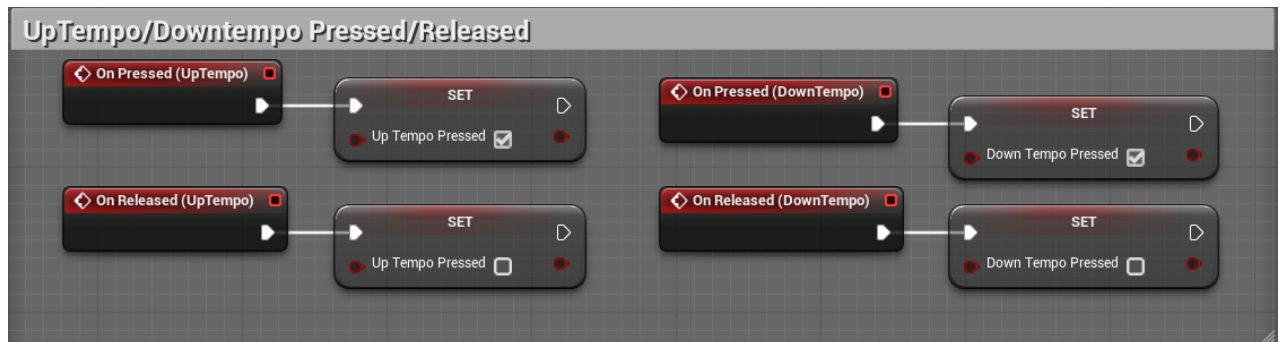
Ο παραπάνω κώδικας λειτουργεί κάθε φορά που ο χρήστης κάνει μεμονωμένο click στο κουμπι *UpTempo* ή *DownTempo* και για κάθε κλικ αντιστοιχεί αύξηση ή μείωση κατά 1. Στην περίπτωση, όμως, που ο χρήστης επιθυμεί να μεταβάλλει το *Tempo* κατά πολλές μονάδες (πχ. Από *Tempo* = 100 σε *Tempo* = 180), θα πρέπει να κάνει πολλές φορές κλικ στο αντίστοιχο κουμπι (80 φορές), κάτι το οποίο είναι χρονοβόρο και κουραστικό. Για το λόγο αυτό έχει γραφτεί κώδικας, ο οποίος αυξάνει/μειώνει το *Tempo* με ένα συνεχόμενο Click. Μόλις ο χρήστης φτάσει στην επιθυμητή τιμή *Tempo*, αφήνει το κουμπι (release) και σταματά η αύξηση/μείωση.

Για αυτήν την υλοποίηση δημιουργήθηκαν δύο μεταβλητές τύπου Boolean (μία για το κάθε κουμπι) με ονόματα: “*UpTempoPressed*” και “*DownTempoPressed*”.



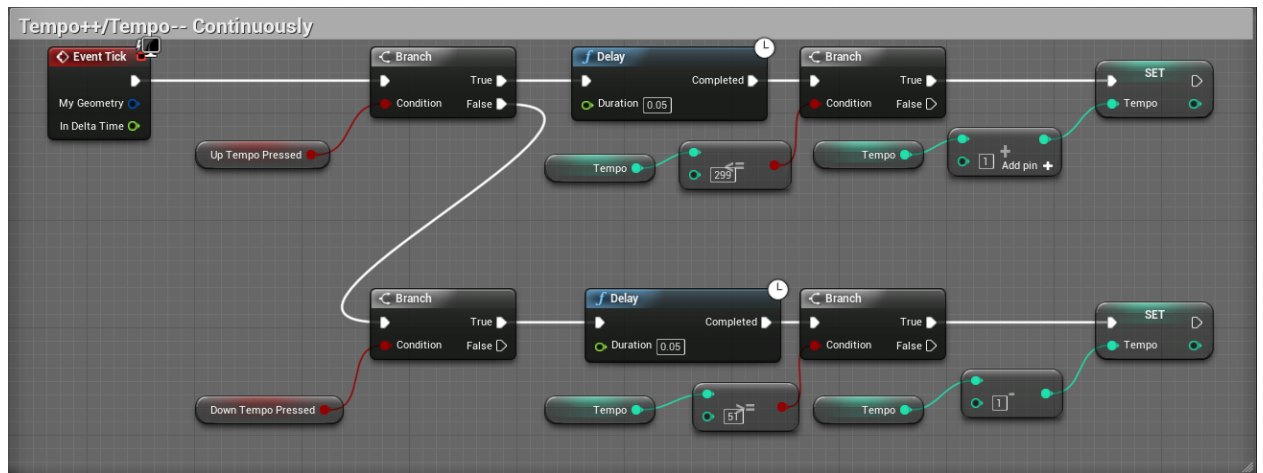
Εικόνα 4.105: Λεπτομέρειες μεταβλητής UpTempoPressed

Με τον παρακάτω κώδικα δίνεται η τιμή True στην “UpTempoPressed”, όταν το κουμπι UpTempo πατηθεί, και η τιμή False, όταν το κουμπι αφεθεί. Το ίδιο συμβαίνει και για το κουμπι DownTempo και την μεταβλητή ”DownTempoPressed”.



Εικόνα 4.106: Ο κώδικας για πάτημα και απελευθέρωση των κουμπιών UpTempo & DownTempo

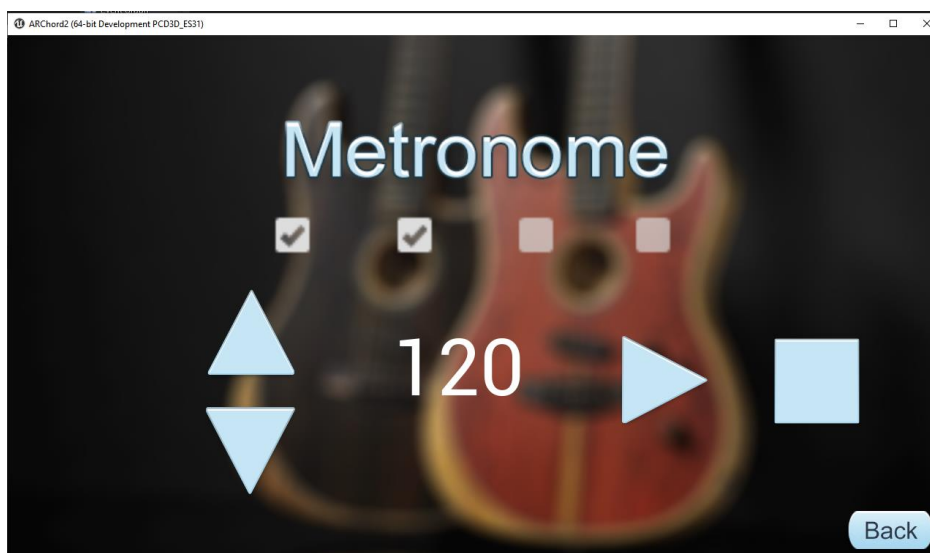
Τώρα επιθυμούμε να αυξήσουμε/μειώσουμε το Tempo κατά 1 για κάθε frame στο οποίο είναι πατημένο το κουμπι UpTempo/DownTempo. Για να επιτευχθεί αυτό χρησιμοποιήθηκε ο κόμβος *Event Tick*, ο οποίος επαναλαμβάνει το κομμάτι κώδικα που είναι συνδεδεμένο σε αυτόν σε κάθε frame.



Εικόνα 4.107: Ο κώδικας για συνεχή αύξομείωση του tempo

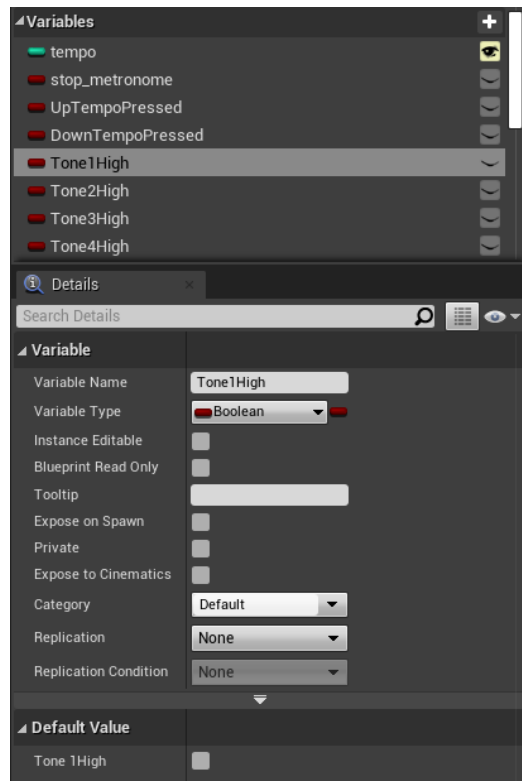
Στη συνέχεια του *Event Tick* υπάρχει ο κόμβος *Branch*, ο οποίος ελέγχει αν η μεταβλητή “UpTempoPressed” περιέχει τιμή True (δηλαδή αν το κουμπί UpTempo είναι πατημένο) και αν ισχύει, εκτελεί τον ίδιο κώδικα με αυτόν που χρησιμοποιήθηκε παραπάνω για την αύξηση του Tempo με μεμονωμένο click. Ο αντίστοιχος κώδικας γράφτηκε και για την μείωση του Tempo. Μία διαφορά με τον προηγούμενο κώδικα είναι ότι προστέθηκε σε κάθε περίπτωση ένας κόμβος *Delay*, ο οποίος καθυστερεί την διαδικασία κατά 0.05 δευτερόλεπτα σε κάθε frame. Αυτό έγινε ώστε να μην αυξομειώνεται η τιμή του Tempo ακαριαία, αλλά η μετάβαση στις διάφορες τιμές να γίνεται με ομαλό τρόπο.

Οι χτύποι αναπαράγοντα σε τετράδες. Υπάρχουν τέσσερα *Check Boxes* για καθένα από τους τέσσερις χτύπους που καθορίζουν το αν ο χτύπος θα είναι τονικά υψηλός ή χαμηλός. Αν είναι *Checked* ένα *Check Box*, θα ηχήσει ο υψηλότερος τονικά χτύπος, ενώ αν είναι *Unchecked* ο χαμηλότερος. Για παράδειγμα, αν τα δύο πρώτα checkboxes είναι checked και τα δύο τελευταία unchecked, θα ηχήσουν δύο τονικά υψηλότεροι χτύποι ακολουθούμενοι από δύο τονικά χαμηλότερους.



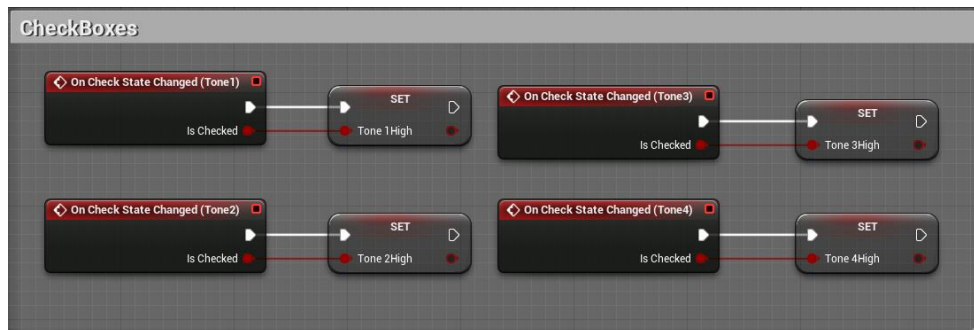
Εικόνα 4.108: Το Widget MetronomeUI με επιλεγμένα τα δύο πρώτα Checkboxes

Δημιουργήθηκαν, λοιπόν, τέσσερις Boolean μεταβλητές, μία για κάθε *Check Box*, οι οποίες αποθηκεύουν την τιμή True, όταν είναι checked και τιμή False, όταν είναι Unchecked. Τα ονόματα των μεταβλητών είναι: Tone1High, Tone2High, Tone3High, Tone4High.



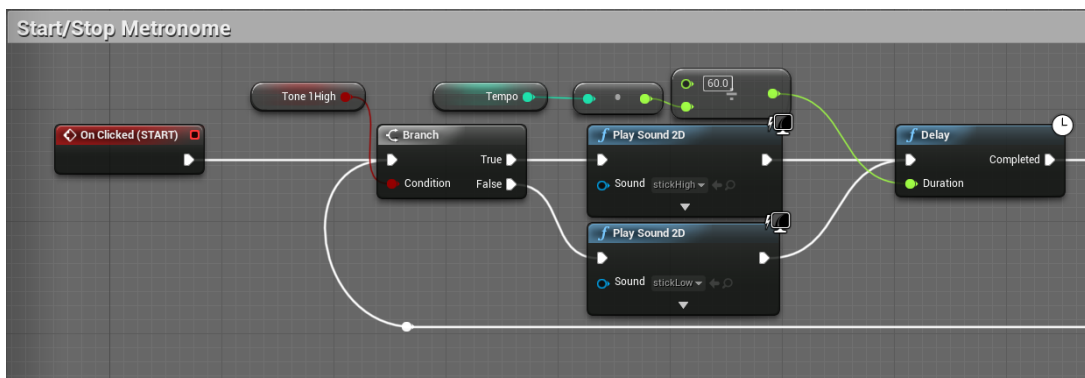
Εικόνα 4.109: Λεπτομέρειες της μεταβλητής Tone1High

Ο παρακάτω κώδικας δίνει τις τιμές True/False στις παραπάνω μεταβλητές ανάλογα με το αν είναι Checked ή Unchecked τα *Check Boxes*.



Εικόνα 4.110: Ο κώδικας για το τσεκάρισμα των Checkboxes

Παρακάτω φαίνεται το πρώτο μέρος του κώδικα που θα εκτελεστεί μόλις πατηθεί το κουμπι έναρξης αναπαραγωγής χτύπων.

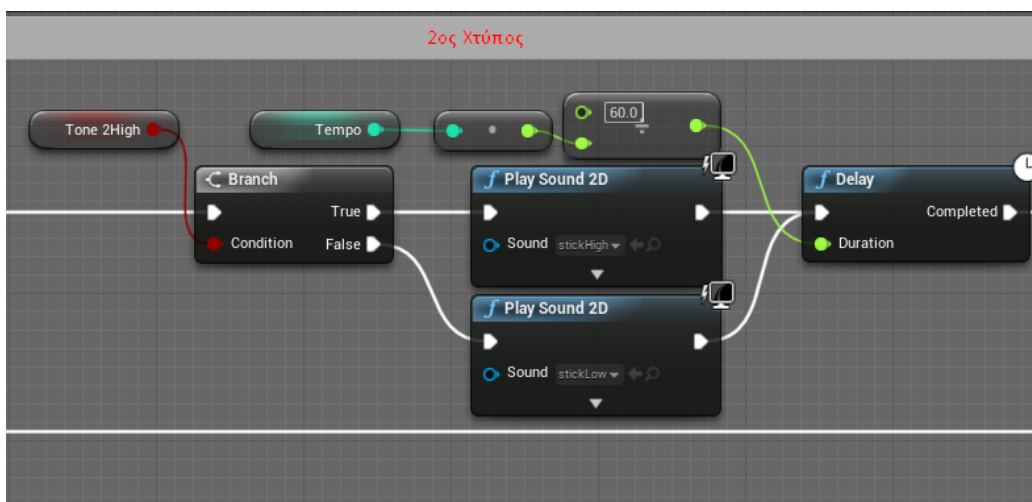


Εικόνα 4.111: Ο κώδικας για την αναπαραγωγή του πρώτου χτύπου

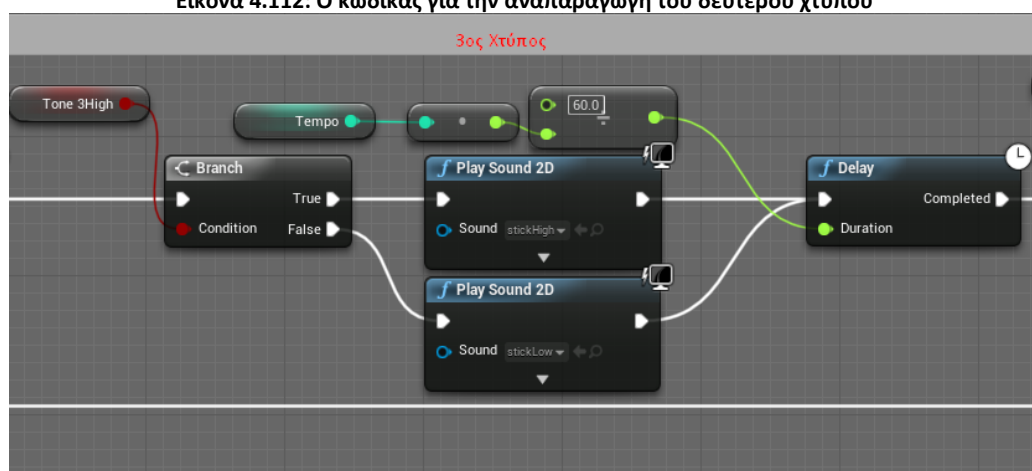
Αρχικά, υπάρχει ο κόμβος *On Clicked(START)*, ο οποίος εκτελεί το κομμάτι κώδικα που είναι συνδεδεμένο σε αυτόν όταν πατηθεί το κουμπί START. Στη συνέχεια, υπάρχει ο κόμβος *Branch*, ο οποίος ελέγχει αν η μεταβλητή “Tone1High” περιέχει τιμή True ή False, δηλαδή αν το πρώτο Check Box είναι checked ή όχι αντίστοιχα. Αν η τιμή είναι True, θα αναπαραχθεί ο υψηλός τονικά χτύπος με την συνάρτηση *Play Sound 2D*, έχοντας επιλέξει στο πεδίο *Sound* το *Sound Cue* “StickHigh”. Αν η τιμή είναι False θα αναπαραχθεί ο χαμηλός τονικά χτύπος με την συνάρτηση *Play Sound 2D*, έχοντας επιλέξει στο πεδίο *Sound* το *Sound Cue* “StickLow”. Στη συνέχεια, θα πρέπει να αναπαραχθεί ο δεύτερος χτύπος έχοντας μία χρονική απόσταση από τον πρώτο. Τη χρονική αυτή απόσταση, την καθορίζει το Tempo. Όπως αναφέρθηκε παραπάνω, η μονάδα μέτρησης του Tempo είναι τα BPM (Χτύποι ανά Λεπτό). Άρα, η χρονική απόσταση μεταξύ δύο διαδοχικών χτύπων ισούται με  $60/\text{Tempo}$ . Για παράδειγμα, αν το Tempo ισούται με 60 θα έχουμε 60 χτύπους ανά λεπτό, δηλαδή ένα χτύπο ανά 1 δευτερόλεπτο ( $60/60=1$ ), αν το Tempo ισούται με 120 θα έχουμε 120 χτύπους ανά λεπτό, δηλαδή ένα χτύπο ανά 0.5 δευτερόλεπτο ( $60/120=0.5$ ) κ.ο.κ.

Επομένως, υπάρχει, ακολούθως της αναπαραγωγής του πρώτου χτύπου, ο κόμβος delay, ο οποίος καθυστερεί το επόμενο κομμάτι κώδικα (στο οποίο βρίσκεται η αναπαραγωγή του δεύτερου χτύπου) κατά τόσα δευτερόλεπτα όσο είναι το  $60/\text{Tempo}$ .

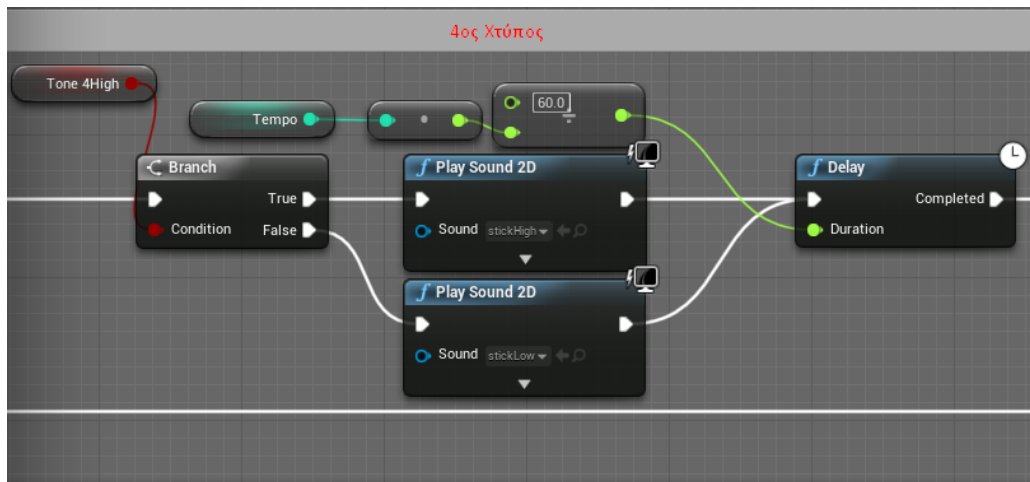
Ο κώδικας συνεχίζει με τον ίδιο τρόπο για τους επόμενους τρεις χτύπους με την κάθε *branch* να ελέγχει το αντίστοιχο *Check Box*.



Εικόνα 4.112: Ο κώδικας για την αναπαραγωγή του δεύτερου χτύπου

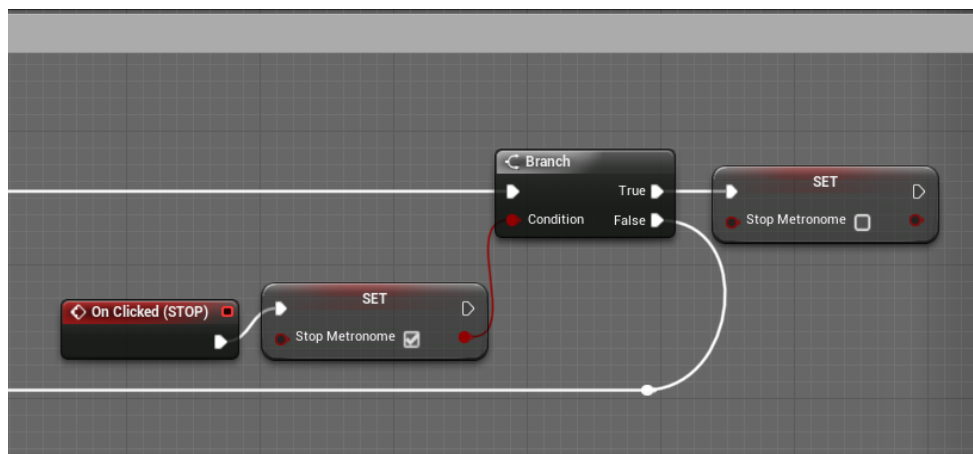


Εικόνα 4.113: Ο κώδικας για την αναπαραγωγή του τρίτου χτύπου



Εικόνα 4.114: Ο κώδικας για την αναπαραγωγή του τέταρτου χτύπου

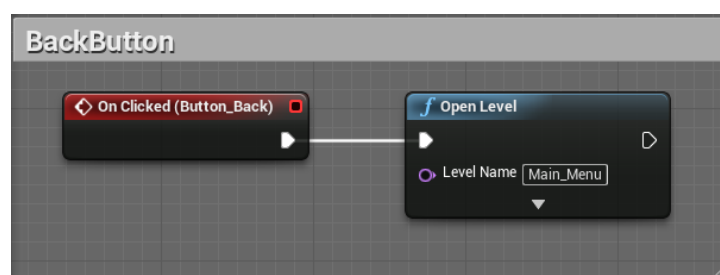
Μετά τον τέταρτο χτύπο υπάρχει ο παρακάτω κώδικας.



Εικόνα 4.115: Ο κώδικας μετά την αναπαραγωγή των χτύπων

Έχει δημιουργηθεί μία μεταβλητή τύπου Boolean με Όνομα “StopMetronome”, η οποία λαμβάνει τιμή True, όταν ο χρήστης πατήσει το κουμπί Παύσης Αναπαραγωγής Χτύπων (Stop). Η παραπάνω Branch ελέγχει την τιμή της “StopMetronome”. Αν είναι False, δηλαδή αν δεν έχει πατηθεί το κουμπί Stop, συνεχίζει την ροή από την πρώτη Branch του κώδικα, δημιουργώντας έτσι οκοπίμως έναν ατέρμονα βρόγχο, ώστε οι χτύποι να επαναλαμβάνονται επ’ άπειρον. Αν είναι True, δηλαδή αν έχει πατηθεί το κουμπί Stop, γίνεται έξοδος απ’ τον βρόγχο, τερματίζοντας την αναπαραγωγή των χτύπων και ο κόμβος SET επαναφέρει την μεταβλητή “StopMetronome” στην τιμή False.

Τέλος, αν πατηθεί το κουμπί Back, ο χρήστης μεταφέρεται στο αρχικό μενού με τον κόμβο *Open Level*.



Εικόνα 4.116: Ο κώδικας για το πάτημα του κουμπιού Back

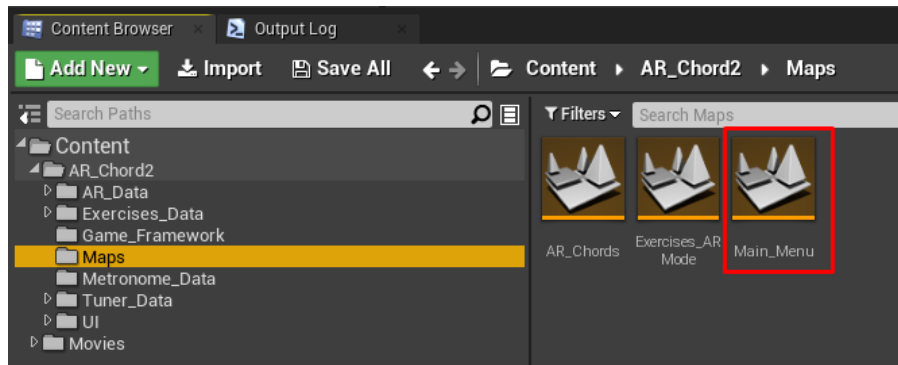


## ΚΕΦΑΛΑΙΟ 5 Διεπαφή Χρήστη

### 5.1 Υλοποίηση του Βασικού Μενού (Main Menu)

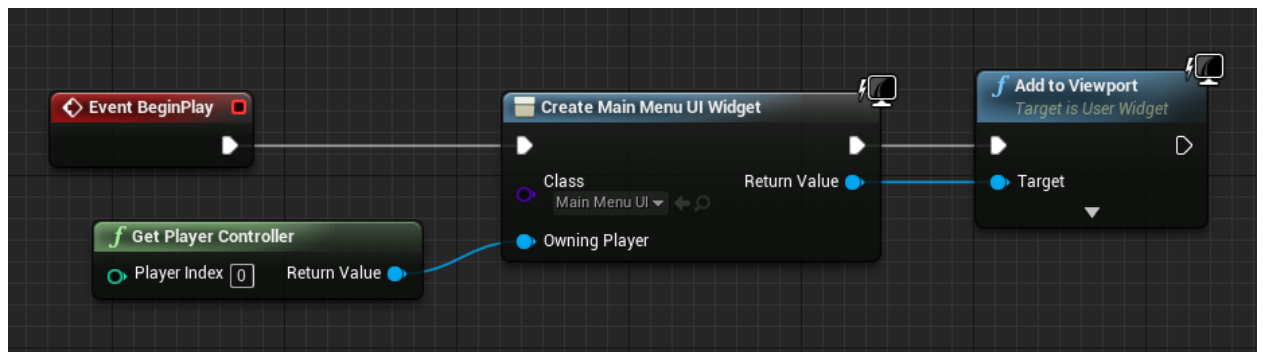
#### 5.1.α Δημιουργία Level για το Main Menu

Στο Project του προγράμματος Unreal Engine 4 δημιουργήθηκε νέο *Level* με όνομα "Main\_Menu".



Εικόνα 5.1: Το Level "Main Menu" στο Content Browser

Στο *Level Blueprint* γράφτηκε ο παρακάτω κώδικας.

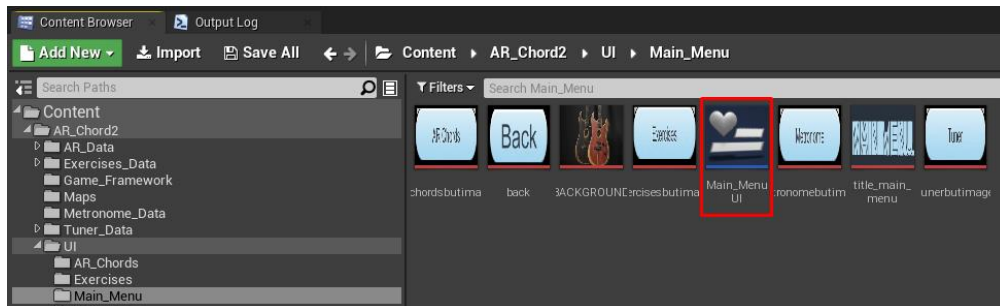


Εικόνα 5.2: Κώδικας για την εμφάνιση του Widget "Main Menu"

Ο κόμβος *Event Begin Play* εκτελεί τον κώδικα που είναι συνδεδεμένος σε αυτόν μόλις ανοίξει το Level. Με τους κόμβους *Create Widget* και *Add to Viewport* εμφανίζεται στην οθόνη το *Widget* "Main Menu UI", το οποίο θα εξηγηθεί παρακάτω.

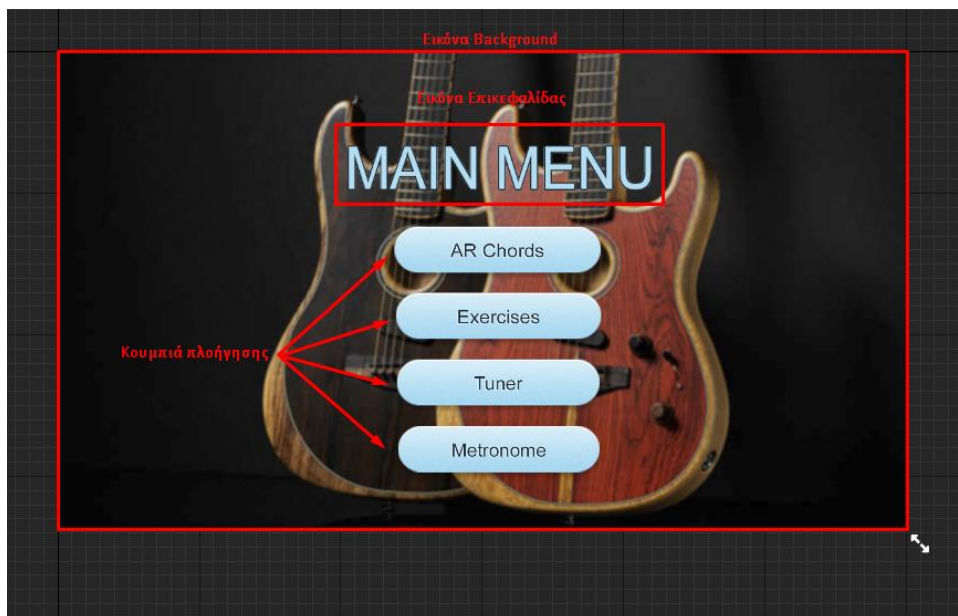
### 5.1.β Δημιουργία Widget για το Main Menu

Δημιουργήθηκε *Widget Blueprint* με Όνομα "Main Menu UI".



Εικόνα 5.3: Το Widget "Main Menu" στο Content Browser

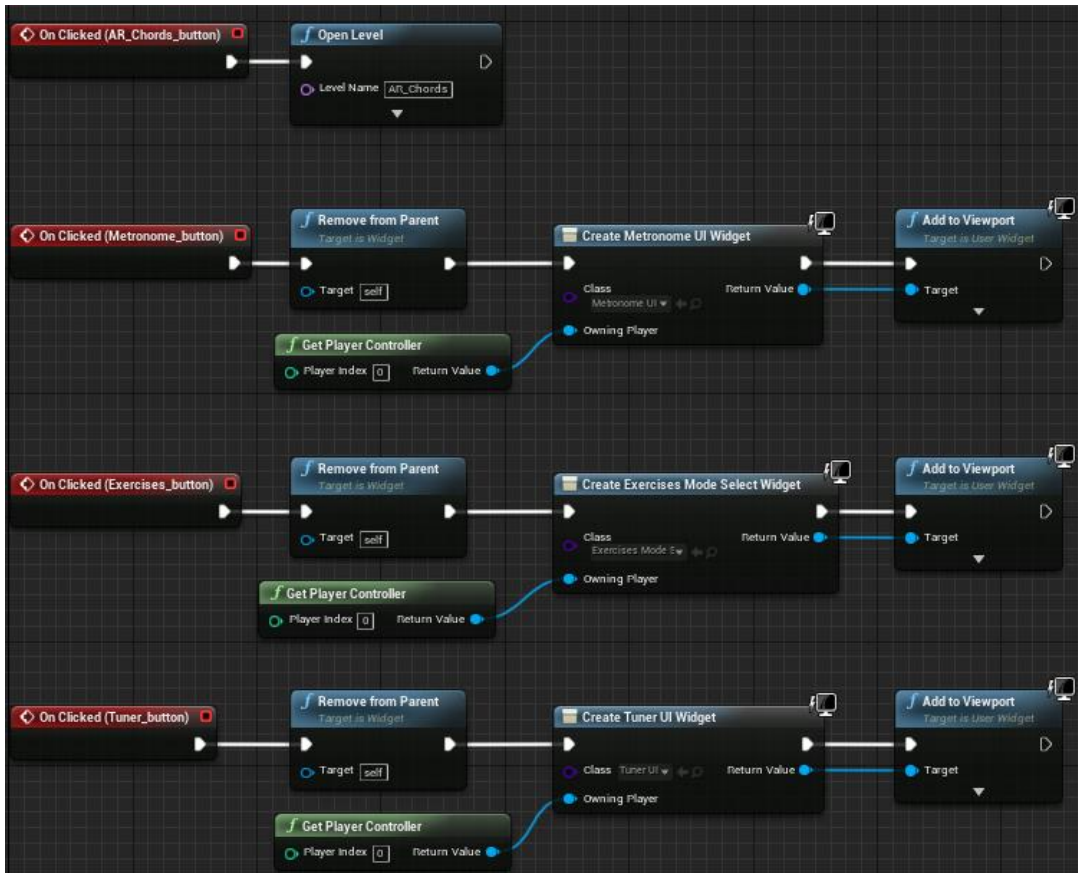
Σε αυτό έγινε εισαγωγή εικόνας Background, εικόνας Επικεφαλίδας, και τεσσάρων κουμπιών για πλοήγηση στις λειτουργίες της εφαρμογής (AR Chords, Exercises, Tuner, Metronome).



Εικόνα 5.4: Επεξήγηση του Widget "Main Menu"

## 5.1.γ Event Graph του Main Menu

Στο Event Graph του Widget γράφτηκε κώδικας για κάθε ένα από τα κουμπιά πλοήγησης:



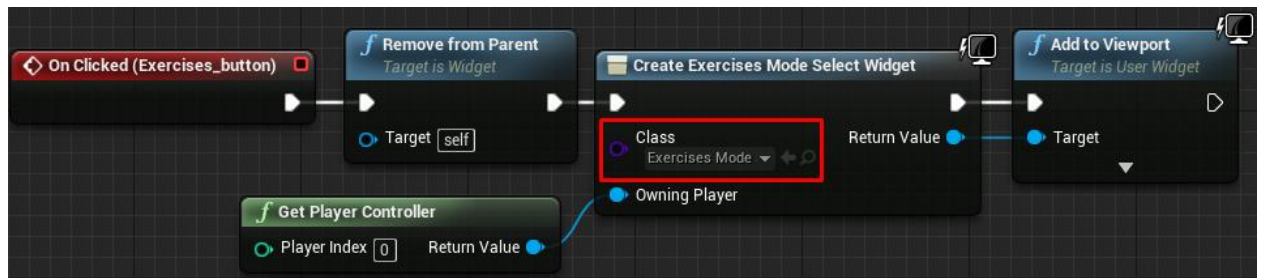
Εικόνα 5.5: Ο κώδικας του Widget "Main Menu"

Συγκεκριμένα, όταν πατηθεί το κουμπί "AR Chords", θα ανοίξει το *Level* "AR\_Chords" χρησιμοποιώντας τον κόμβο *Open Level*.



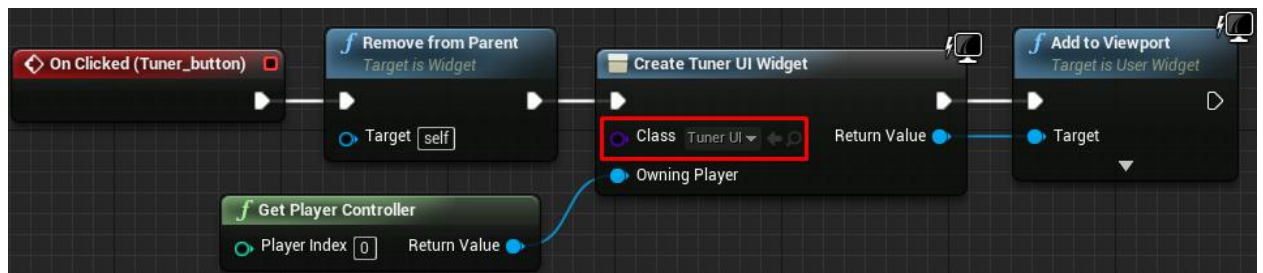
Εικόνα 5.6: Ο κώδικας για το πάτημα του κουμπιού "AR Chords"

Όταν πατηθεί το κουμπί Exercises, θα εκτελεστεί ο παρακάτω κώδικας, ο οποίος αφαιρεί το παρόν *Widget* από την οθόνη, χρησιμοποιώντας τον κόμβο *Remove from Parent*, και στη συνέχεια εμφανίζει το *Widget* "Exercises Mode Select", χρησιμοποιώντας τους κόμβους *Create Widget* και *Add to Viewport*.

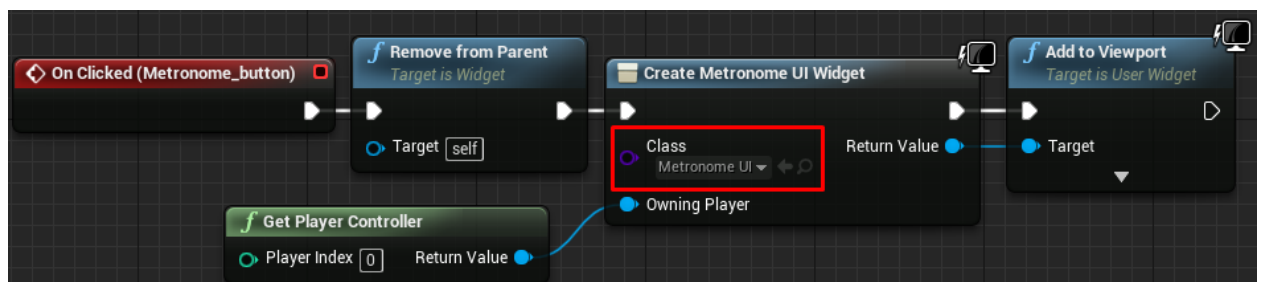


Εικόνα 5.7: Ο κώδικας για το πάτημα του κουμπιού "Exercises"

Ο ίδιος κώδικας θα εκτελεστεί και με το πάτημα των κουμπιών "Tuner" και "Metronome" με την διαφορά ότι στον κόμβο *Create Widget* έχει επιλεγθεί το αντίστοιχο *Widget* της κάθε λειτουργίας.



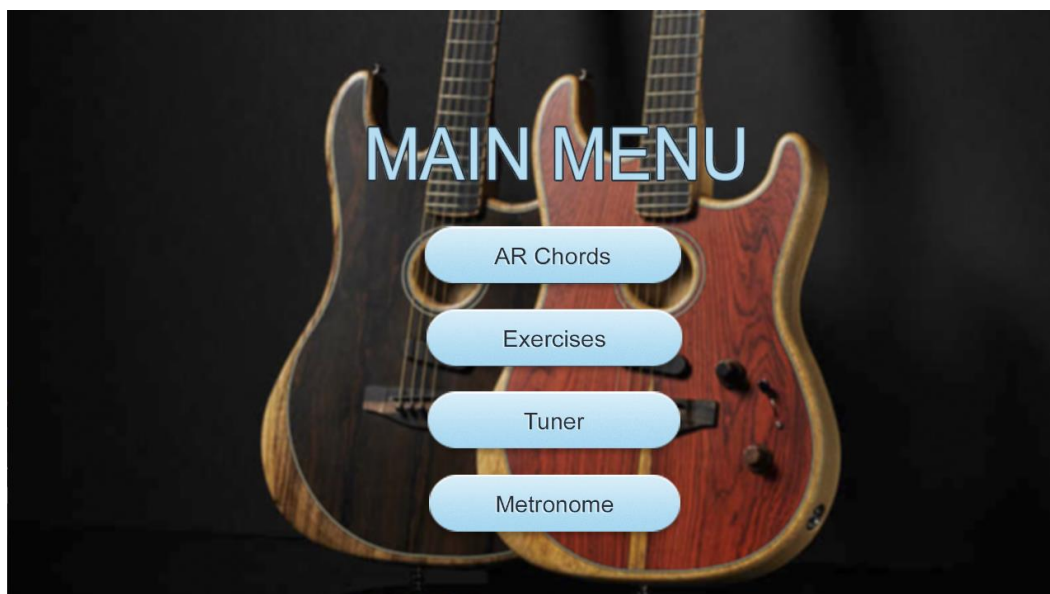
Εικόνα 5.8: Ο κώδικας για το πάτημα του κουμπιού "Tuner"



Εικόνα 5.9: Ο κώδικας για το πάτημα του κουμπιού "Metronome"

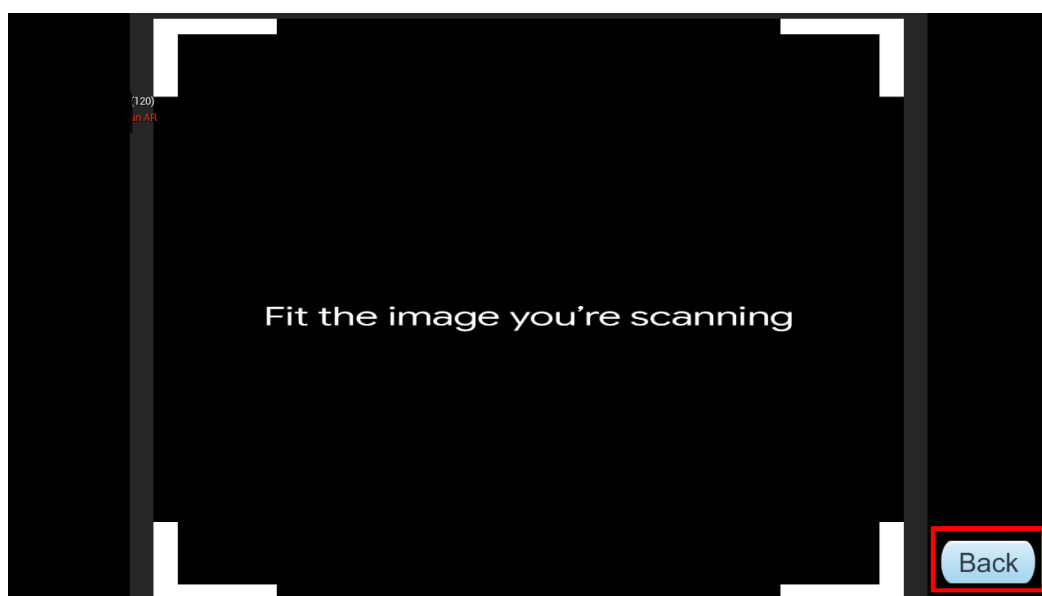
## 5.2 Πλοήγηση στη Διεπαφή χρήστη

Με την έναρξη της εφαρμογής εμφανίζεται στο χρήστη αυτόματα το *Widget* του Βασικού Μενού (Main Menu):



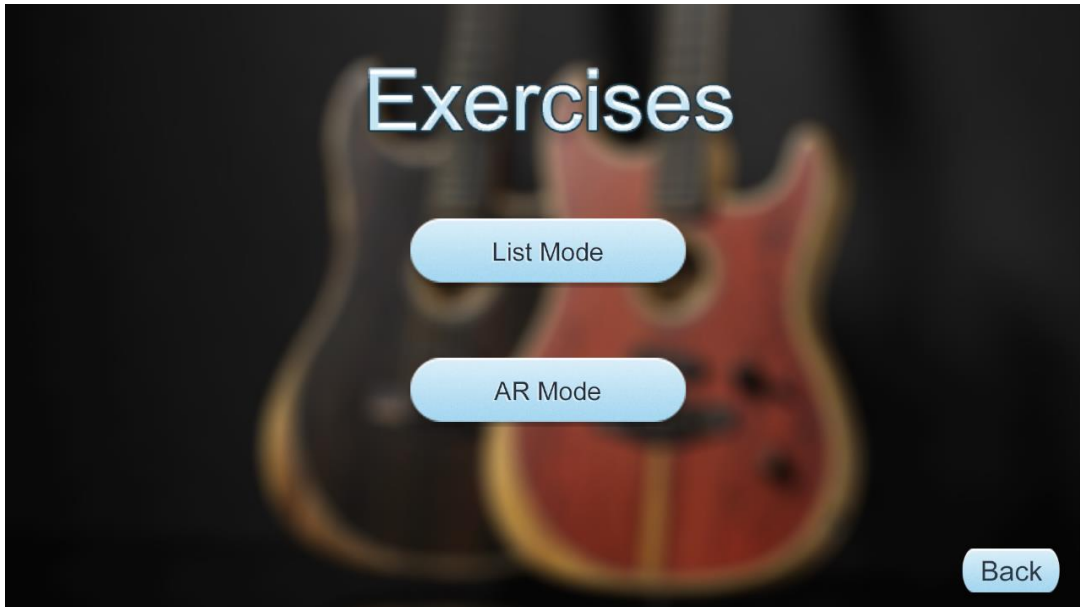
Εικόνα 5.10: Το *Widget* "Main Menu"

Η πλοήγηση στα *Widgets* των λειτουργιών της εφαρμογής επιτυγχάνεται πατώντας τα κατάλληλα κουμπιά. Συγκεκριμένα, ο χρήστης πατώντας το κουμπί AR Chords μεταφέρεται στο *Widget* της ομώνυμης λειτουργίας, στο οποίο υπάρχει το κουμπί Back που τον επαναφέρει στο Βασικό Μενού.



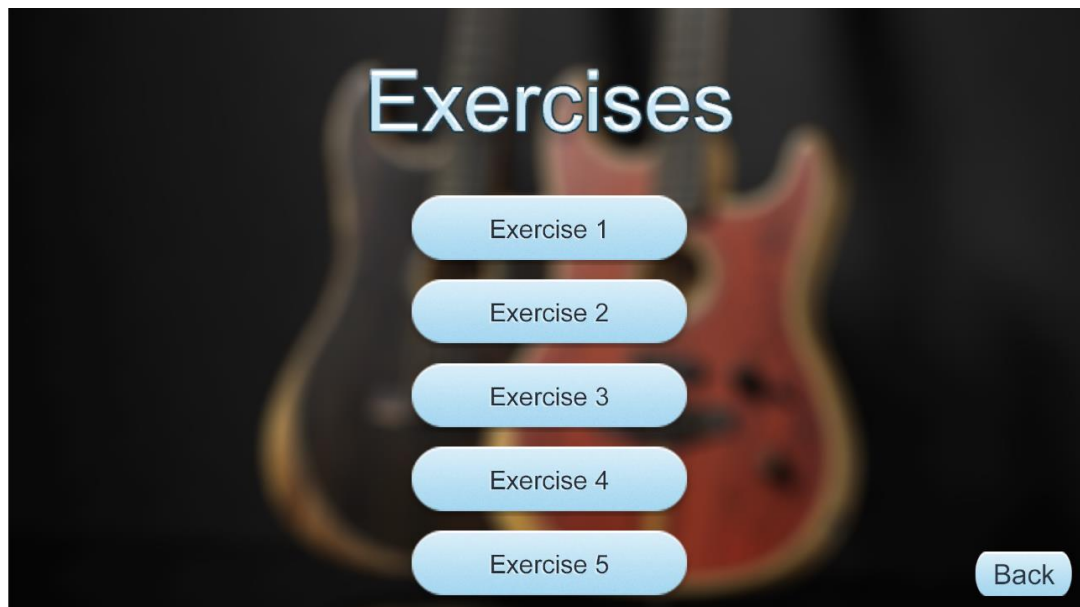
Εικόνα 5.11: Το *Widget* "AR Chords"

Πατώντας το κουμπί Exercises μεταφέρεται στο *Widget* επιλογής ανάμεσα σε List Mode και AR Mode στο οποίο, επίσης, υπάρχει επιλογή επιστροφής στο Βασικό Μενού μέσω του κουμπιού Back.



Εικόνα 5.12: Το Widget για επιλογή μεταξύ List Mode/AR Mode

Με το πάτημα του κουμπιού List Mode ο χρήστης μεταφέρεται στο *Widget* επιλογής παρακολούθησης άσκησης. Σε αυτό υπάρχουν πέντε διαθέσιμα κουμπιά τα οποία αντιστοιχούν στις πέντε διαθέσιμες ασκήσεις. Με το πάτημά τους εμφανίζεται το *Widget* στο οποίο γίνεται η αναπαραγωγή του βίντεο της άσκησης. Στο *Widget* επιλογής άσκησης, όπως επίσης και στα *Widgets* αναπαραγωγής των βίντεο, υπάρχει κουμπί Back το οποίο μεταφέρει τον χρήστη στο *Widget* επιλογής ανάμεσα σε List Mode και AR Mode.



Εικόνα 5.13: Το Widget "List Mode"



Εικόνα 5.14: Το Widget Αναπαραγωγής Βίντεο

Πατώντας το κουμπι AR Mode εμφανίζεται στην οθόνη του χρήστη το *Widget* για την επιλογή αναπαραγωγής του βίντεο με την χρήση της κάμερας της κινητής συσκευής.



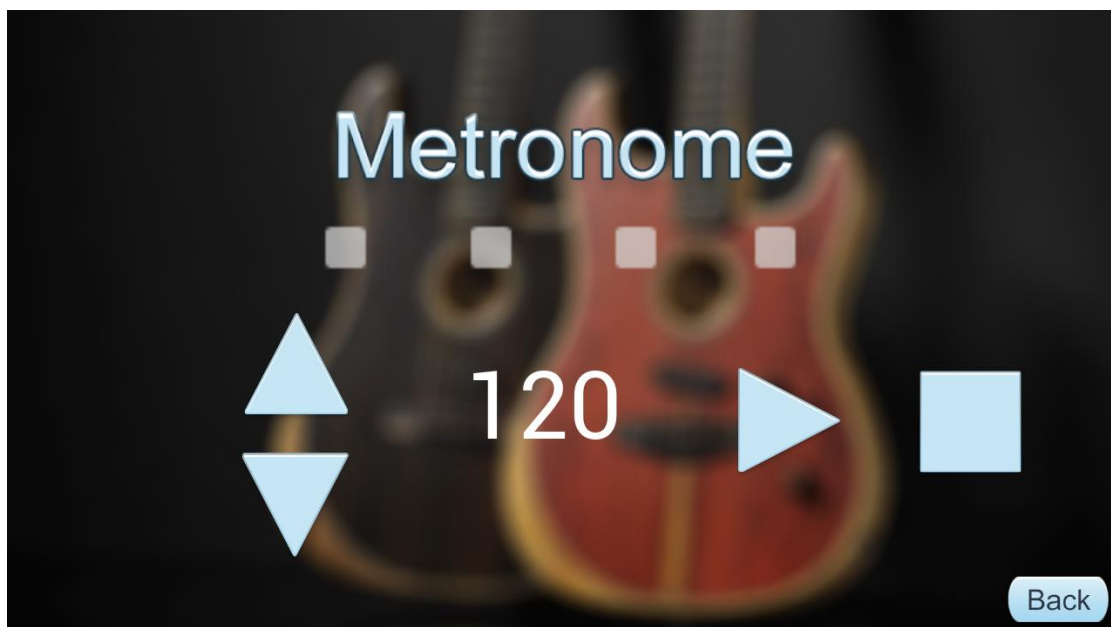
Εικόνα 5.15: Το Widget "AR Mode"

Πατώντας το κουμπι Back γίνεται επιστροφή στο *Widget* επιλογής Mode.

Από το Βασικό Μενού ο χρήστης πατώντας τα κουμπιά Tuner και Metronome μεταφέρεται στο *Widget* για το Κουρδιστήρι και τον Μετρονόμο αντίστοιχα, στα οποία υπάρχει κουμπι Back, το οποίο τον επαναφέρει στο Βασικό Μενού.



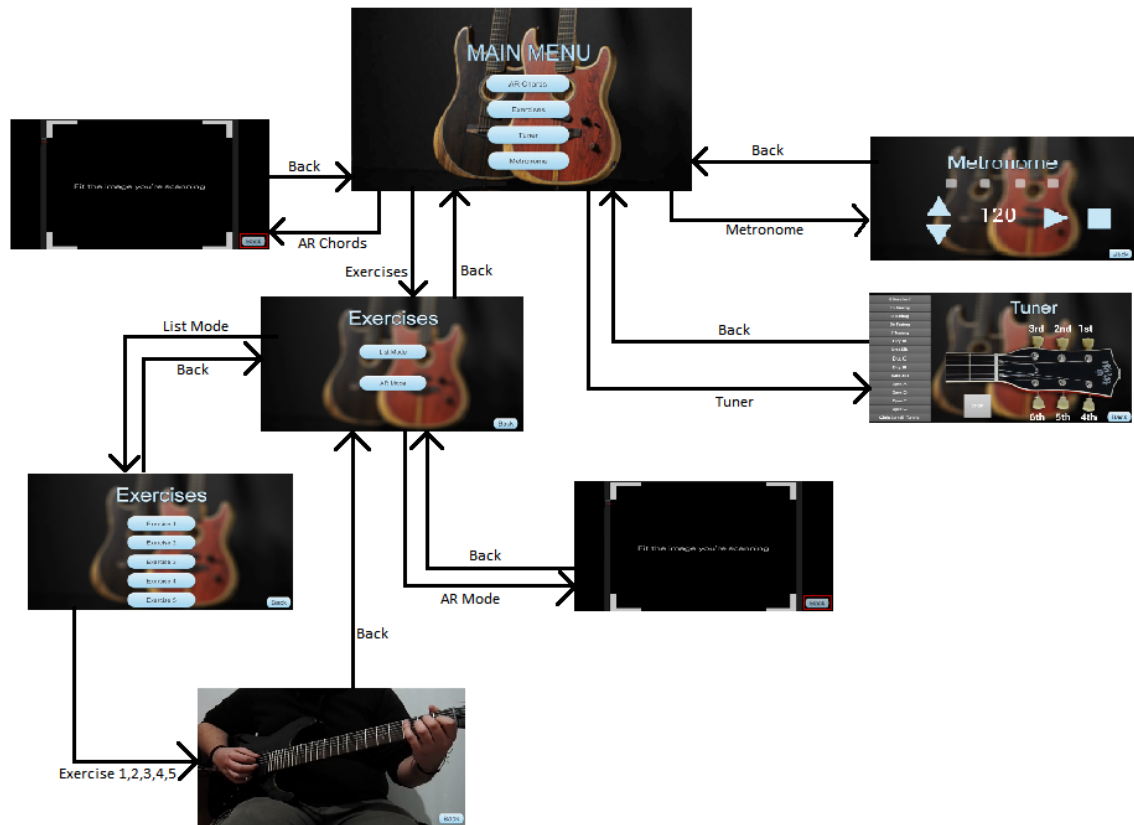
Εικόνα 5.16: Το Widget "Tuner"



Εικόνα 5.17: Το Widget "Metronome"

Στο παρακάτω διάγραμμα ροής σχεδιάστηκαν οι μεταβάσεις που μπορούν να πραγματοποιηθούν μεταξύ των Widgets της εφαρμογής. Τα βέλη συμβολίζουν την μετάβαση και η ετικέτα του βέλους το κουμπί που θα πρέπει να πατηθεί ώστε να πραγματοποιηθεί η μετάβαση.



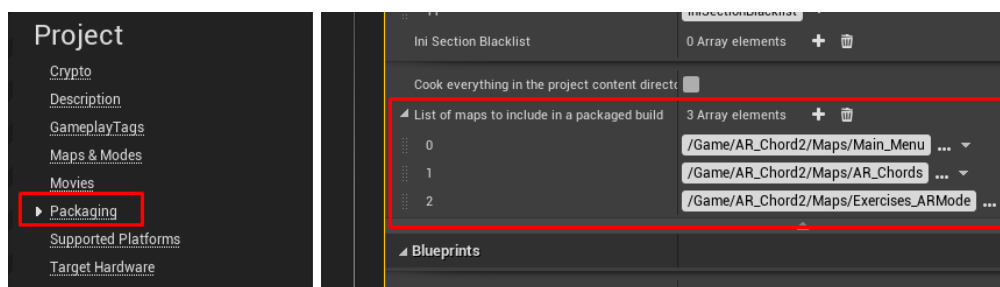


Εικόνα 5.18: Διάγραμμα ροής μεταβάσεων στη διεπαφή χρήστη

# ΚΕΦΑΛΑΙΟ 6 Απαραίτητες Ρυθμίσεις, Τεχνικά Θέματα & Περιορισμοί της Εφαρμογής

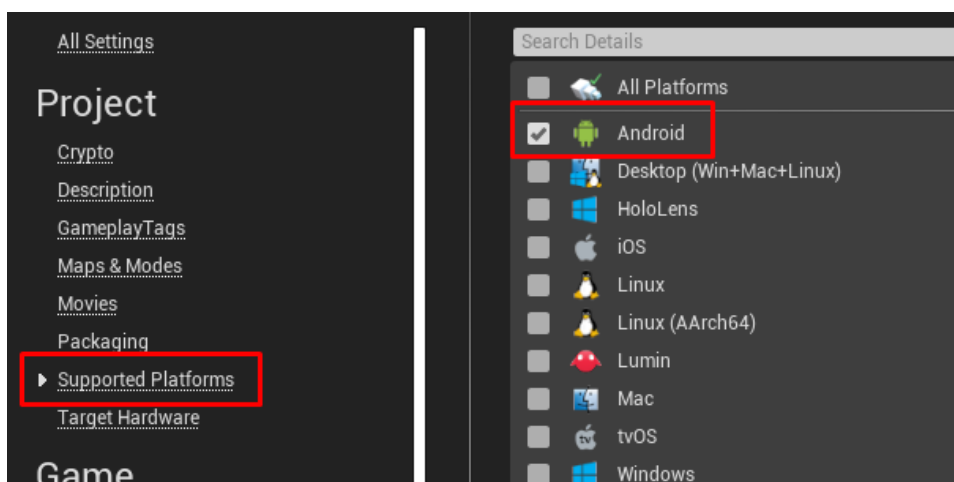
## 6.1 Ρυθμίσεις στο Project Settings του UE4

Για την σωστή φόρτωση και λειτουργία της εφαρμογής στην κινητή συσκευή έγιναν κάποιες απαραίτητες ρυθμίσεις στο *Project Settings* του προγράμματος Unreal Engine 4. Αρχικά, στην στήλη *Packaging* στο πεδίο *List of Maps to include in a packaged build* προστέθηκαν οι διαδρομές των αρχείων των Levels που δημιουργήθηκαν στην εφαρμογή. Αυτό έγινε ώστε, όταν φορτώνεται η εφαρμογή στη κινητή συσκευή, να μην φορτώνεται μόνο το Level το οποίο είναι ανοιχτό τη στιγμή που γίνεται η φόρτωση, αλλά όλα τα απαραίτητα Levels.



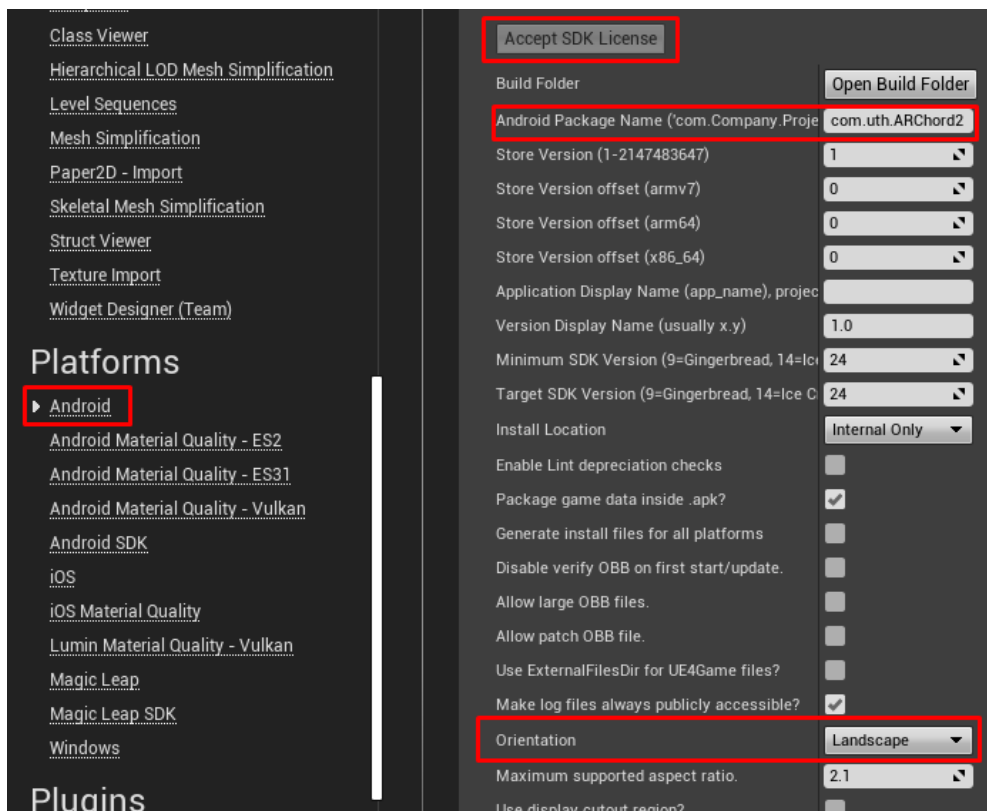
Εικόνα 6.1: Ρυθμίσεις Packaging στο Project Settings

Στη συνέχεια, στη στήλη *Supported Platforms* ενεργοποιήθηκε το πεδίο *Android* ώστε η εφαρμογή να υποστηρίζεται από πλατφόρμες Android.



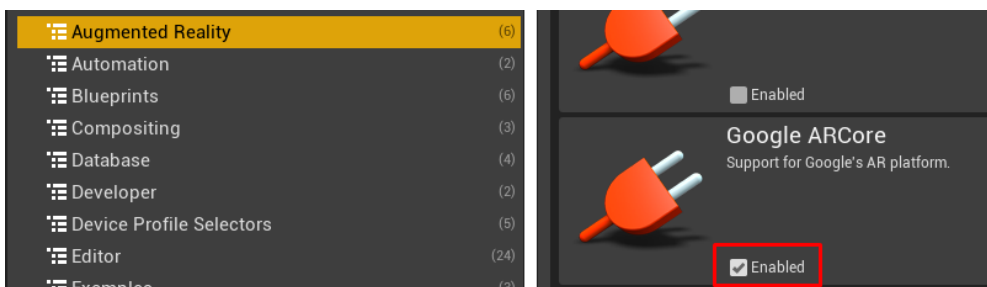
Εικόνα 6.2: Ρυθμίσεις Supported Platforms στο Project Settings

Στη στήλη *Android* έγινε αποδοχή του *SDK License*, δόθηκε το όνομα της εφαρμογής που θα φορτωθεί στην κινητή συσκευή από το πεδίο *Android Package Name* και επιλέχθηκε στο πεδίο *Orientation* η επιλογή *Landscape*.



Εικόνα 6.1: Ρυθμίσεις Android στο Project Settings

Τέλος, για να είναι διαθέσιμες οι συναρτήσεις της *ARCore* ήταν απαραίτητο να γίνει ενεργοποίηση της επέκτασης *Google ARCore* από την στήλη *Augmented Reality* του παραθύρου *Plugins*.

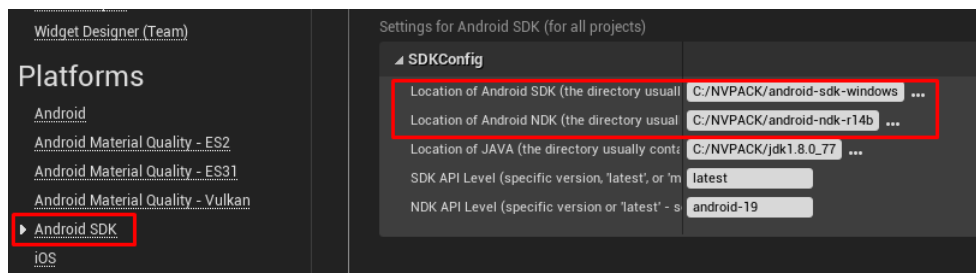


Εικόνα 6.2: Plugin Google ARCore

## 6.2 Σύνδεση Κινητής συσκευής με το UE4

Για την σύνδεση της κινητής συσκευής με το UE4 θα πρέπει αρχικά να εγκατασταθεί στον Η/Υ το πρόγραμμα *Android Studio 3.5.3*, ώστε να υπάρξει πρόσβαση σε δύο αρχεία τα οποία είναι απαραίτητα για την σύνδεση, το *Android Software Development Kit (SDK)* και το *Android Native Development Kit (NDK)*.

Στη συνέχεια, θα πρέπει να γνωστοποιήσουμε στο UE4 την τοποθεσία που αυτά τα αρχεία έχουν αποθηκευτεί. Αυτό επιτυγχάνεται μέσω της στήλης *Android SDK* του παραθύρου *Project Settings* στα πεδία *Location of Android SDK* και *Location of Android NDK*.



Εικόνα 6.5: Ρυθμίσεις Android SDK στο Project Settings

Τέλος, θα πρέπει να γίνουν κάποιες ρυθμίσεις στην κινητή συσκευή. Αφού ενεργοποιηθεί η *Λειτουργία Προγραμματιστή*, στη στήλη *Ρυθμίσεις Προγραμματιστή* θα πρέπει να ενεργοποιηθεί το πεδίο *USB Debugging*.

## 6.3 Περιορισμοί

### 6.3.α Περιορισμοί Προγραμματιστή

Για να γίνει η αναπαραγωγή βίντεο στο Android Media Player που δημιουργήθηκε για την λειτουργία “Exercises”, θα πρέπει τα αρχεία βίντεο που θα εισαχθούν να είναι υποστηριζόμενοι τύπου. Οι υποστηριζόμενοι τύποι αρχείων είναι αναρτημένοι στο Documentation της UE4.

	AndroidMedia Player	AvfMedia Player	Electra Player	PS4Media Player	WmfMedia Player	MfMedia Player (XboxOne)	MfMedia Player (Windows 7+)
.3g2		Y			Y		Y
.3gp		Y			Y		Y
.3gp2		Y			Y		Y
.3gpp	Y	Y			Y		Y
.aac	Y				Y		
.ac3		Y					Y
.adts					Y		
.aif		Y					Y
.aifc		Y					Y
.aiff		Y					Y
.amr		Y					Y
.asf					Y		
.au		Y					Y
.avi					Y		
.bwf		Y					Y
.caf		Y					Y
.cdda		Y					Y
.m2ts					Y		
.m3u8	Y		Y	Y			
.m3d				Y			
.m4a		Y		Y	Y		Y
.m4v		Y		Y	Y		
.mp4v							Y
.mov		Y		Y	Y		Y
.mp3		Y			Y		Y
.mp4	Y	Y	Y	Y	Y	Y	Y
.qt		Y					Y
.sami					Y		
.sdv		Y					Y
.smi					Y		
.snd		Y					Y
.wav		Y			Y		Y
.wave		Y					Y
.wma					Y		
.wmv					Y		

Εικόνα 6.6: Πίνακας υποστηριζόμενων τύπων αρχείου βίντεο<sup>3</sup>

<sup>3</sup> <https://docs.unrealengine.com/en-US/WorkingWithMedia/MediaFramework/TechReference/index.html>

Το ίδιο ισχύει και για τα αρχεία ήχου που χρησιμοποιήθηκαν στις λειτουργίες “Tuner” και “Metronome”.

## Import

The engine currently supports importing uncompressed little endian 16 bit wave files at any sample rate (although sample rates of 44100 Hz or 22050 Hz are recommended).

Specifications	PCM, ADPCM, DVI ADPCM
Format	.WAV
Bit Rate	16
Speaker Channels	Mono, Stereo, 2.1, 4.1, 5.1 6.1, 7.1

**Εικόνα 6.7: Πίνακας υποστηριζόμενων τύπων αρχείου ήχου<sup>4</sup>**

Τέλος, τα αρχεία εικόνας που χρησιμοποιήθηκαν για τις λειτουργίες “AR Chords” και “Exercises” εκτός του ότι θα πρέπει να είναι υποστηριζόμενου από το UE4 τύπου, θα πρέπει να πληρούν και κάποιες προδιαγραφές που ορίζει το ARCore, αφού θα προσπαθήσει να τις αναγνωρίσει (Image Recognition) σε μία AR συνεδρία.

When creating textures for use in UE4, only the following texture formats are supported:

- .bmp
- .float
- .pcx
- .png
- .psd
- .tga
- .jpg
- .exr
- .dds - Cubemap Texture (32bits/Channel, 8.8.8.8 ARGB 32 bpp, unsigned) See [Cubemap Tools](#) for more information.
- .hdr - Cubemap Texture (LongLat unwrap)

**Εικόνα 6.8: Πίνακας υποστηριζόμενων τύπων αρχείου εικόνας<sup>5</sup>**

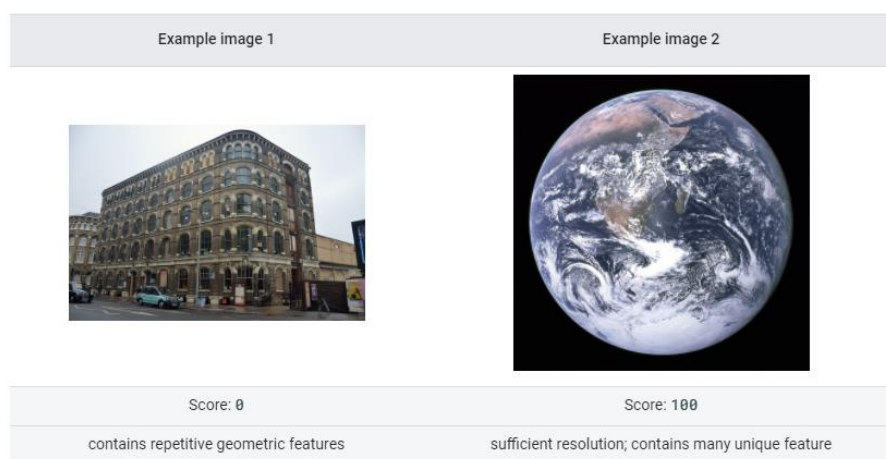
<sup>4</sup> <https://docs.unrealengine.com/en-US/WorkingWithMedia/Audio/WAV/index.html>

<sup>5</sup> <https://docs.unrealengine.com/en-US/RenderingAndGraphics/Textures/Importing/index.html>

Οι προδιαγραφές αυτές είναι αναρτημένες στο documentation του ARCore.

#### Tips for selecting reference images

- The image's resolution should be **at least 300 x 300 pixels**. Using images with high resolution does *not* improve performance.
- Reference images can be provided in **PNG or JPEG file format**.
- **Color information is not used**. Both color and equivalent grayscale images can be used as reference images or by users at runtime.
- **Avoid images with heavy compression** as this interferes with feature extraction.
- **Avoid images with that contain a large number of geometric features, or very few features** (e.g. barcodes, QR codes, logos and other line art) as this will result in poor detection and tracking performance.
- **Avoid images with repeating patterns** as this also can causes issues with detection and tracking.
- Use the `arccoreimg` tool included in the ARCore SDK to get a quality score between **0** and **100** for each image. **We recommend a quality score of at least 75**. Here are two examples:



Εικόνα 6.9: Προδιαγραφές εικόνας από την Google ARCore<sup>6</sup>

### 6.3.β Περιορισμοί Χρήστη

Ο χρήστης θα πρέπει να διαθέτει κινητή συσκευή με «καλής ποιότητας κάμερα, δυνατότητα ανίχνευσης κίνησης (Motion Sensor) και αρκετά ισχυρό επεξεργαστή για να εξασφαλίσει καλή απόδοση και αποτελεσματικούς υπολογισμούς σε πραγματικό χρόνο», όπως αναφέρεται στην ιστοσελίδα της ARCore. Μόνον εάν πληροί τις παραπάνω προϋποθέσεις η συσκευή θα είναι συμβατή με το ARCore, το οποίο έχει αναρτήσει στο Documentation του λίστα με όλες τις υποστηριζόμενες κινητές συσκευές. Οι συσκευές οι οποίες είναι υποστηριζόμενες συνήθως έχουν προ εγκατεστημένο το ARCore. Τέλος, η συσκευή θα πρέπει να διαθέτει 226 MB ελεύθερου αποθηκευτικού χώρου.

<sup>6</sup> <https://developers.google.com/ar/discover/supported-devices>

## ΚΕΦΑΛΑΙΟ 7 Συμπεράσματα

---

Στις μέρες μας, η τεχνολογική πρόοδος διευκολύνει σημαντικά την ανάπτυξη εφαρμογών Επαυξημένης Πραγματικότητας (AR) από την σκοπιά της υλοποίησης (με εργαλεία όπως το Google ARCore), αλλά και από την σκοπιά της χρήσης (κινητές συσκευές με επαρκή επεξεργαστική ισχύ, κάμερα υψηλής ανάλυσης με καταγραφή κίνησης κτλ.).

Στην παρούσα πτυχιακή εργασία παρουσιάζεται αναλυτικά η μεθοδολογία που ακολουθήθηκε για την ανάπτυξη μίας AR εφαρμογής με στόχο την εκμάθηση κιθάρας, χρησιμοποιώντας το πρόγραμμα Unreal Engine 4 και Οπτική Γλώσσα Προγραμματισμού (Visual Programming Language).

Δεδομένου ότι η απεικόνιση τρισδιάστατων (3D) στοιχείων δεν αποτελεί αναγκαιότητα μόνο στο κομμάτι των συγχορδίων της κιθάρας, που είναι η βασική λειτουργία της παρούσης εφαρμογής, αναδεικνύονται προοπτικές για επέκταση. Μία μελλοντική επέκταση μπορεί, για παράδειγμα, να αφορά άλλες ενότητες της κιθάρας αλλά και ενότητες άλλων μουσικών οργάνων. Έτσι, συνολικά η εκμάθηση της μουσικής καθίσταται μία πιο εύκολη και κατανοητή διαδικασία, μέσω του συνδυασμού πραγματικού και ψηφιακού κόσμου, παρέχοντας επιπλέον κίνητρο για μάθηση.

Θα είχε ιδιαίτερο ενδιαφέρον, επίσης, να εξεταστεί η συγκεκριμένη εφαρμογή στην πράξη και να αποτυπωθούν οι εντυπώσεις των μαθητών για την χρήση της.



## BIBΛΙΟΓΡΑΦΙΑ

---

- Correa, A. G., Lemos, B. H., Nascimento, M., & Lopes, R. D. (2016). AR Musical App for Children's Musical Education. In *2016 IEEE International Symposium on Consumer Electronics (ISCE)* (pp. 125-126). IEEE.
- Fernandez, C. A. T., Paliyawan, P., Yin, C. C., & Thawonmas, R. (2016). Piano learning application with feedback provided by an ar virtual character. In *2016 IEEE 5th Global Conference on Consumer Electronics* (pp. 1-2). IEEE.
- Keebler, J. R., Wiltshire, T. J., Smith, D. C., & Fiore, S. M. (2013). Picking up STEAM: educational implications for teaching with an augmented reality guitar learning system. In *International Conference on Virtual, Augmented and Mixed Reality* (pp. 170-178). Springer, Berlin, Heidelberg.
- Kerawalla, L., Luckin, R., Seljeflot, S., & Woolard, A. (2006). "Making it real": exploring the potential of augmented reality for teaching primary school science. *Virtual reality*, *10*(3-4), 163-174.
- Klopfer, E., & Squire, K. (2008). Environmental Detectives—the development of an augmented reality platform for environmental simulations. *Educational technology research and development*, *56*(2), 203-228.
- Liu, W., Cheok, A. D., Mei-Ling, C. L., & Theng, Y. L. (2007). Mixed reality classroom: learning from entertainment. In *Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts* (pp. 65-72).
- Löchtefeld, M., Gehring, S., Jung, R., & Krüger, A. (2011). guitAR: supporting guitar learning through mobile projection. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems* (pp. 1447-1452).
- Martin-Gutierrez, J., Del Rio Guerra, M. S., Lopez-Chao, V., Soto Gastelum, R. H., & Valenzuela Bojórquez, J. F. (2020). Augmented reality to facilitate learning of the acoustic guitar. *Applied Sciences*, *10*(7), 2425.
- Martins, V. F., Gomez, L., & Corrêa, A. G. D. (2015). Teaching Children Musical Perception with MUSIC-AR. *EAI Endorsed Trans. e Learn.*, *2*(5), e3.
- Motokawa, Y., & Saito, H. (2006). Support system for guitar playing using augmented reality display. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality* (pp. 243-244). IEEE.
- Preka, G., & Rangoussi, M. (2019). Augmented Reality and QR Codes for Teaching Music to Preschoolers and Kindergarteners: Educational Intervention and Evaluation. In *CSEDU (1)* (pp. 113-123).
- Prit Kaur, D., Mantri, A., & Horan, B. (2019). Design implications for adaptive augmented reality based interactive learning environment for improved concept comprehension in engineering paradigms. *Interactive Learning Environments*, 1-19.
- Saidin, N. F., Halim, N. D. A., & Yahaya, N. (2015). A review of research on augmented reality in education: Advantages and applications. *International education studies*, *8*(13), 1-8.

