

## ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Δημιουργία εφαρμογής κινητού τηλεφώνου που μετατρέπει την μουσική σε δονήσεις, με χρήση απτικής αλληλεπίδρασης και οπτικής αναπαράστασης για άτομα με προβλήματα ακοής»



**Φοιτήτρια:**  
**Τσουρονάκη Ζωή**  
**Αριθμός Μητρώου: 518169**

**Επιβλέπων:**  
**Κοκκώνης Γεώργιος**  
**Επίκουρος Καθηγητής**

Δημιουργία εφαρμογής κινητού τηλεφώνου που μετατρέπει την μουσική σε δονήσεις, με χρήση απτικής αλληλεπίδρασης και οπτικής αναπαράστασης για άτομα με προβλήματα ακοής

Κωδικός: 24135

Όνοματεπώνυμο φοιτήτριας: Τσουρουνάκη Ζωή

Όνοματεπώνυμο εισηγητή: Κοκκώνης Γεώργιος

Ημερομηνία ανάληψης Π.Ε.: 01-03-2024

Ημερομηνία περάτωσης Π.Ε.: 12-09-2025

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

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

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

## Πρόλογος

Η τεχνολογία εξελίσσεται με ταχύτατους ρυθμούς και επηρεάζει πλέον κάθε πτυχή της ανθρώπινης δραστηριότητας. Από την επικοινωνία και την ψυχαγωγία μέχρι την εκπαίδευση και την υγεία, οι σύγχρονες τεχνολογικές λύσεις διαμορφώνουν έναν νέο τρόπο ζωής, όπου οι ψηφιακές εφαρμογές κατέχουν πρωταγωνιστικό ρόλο. Σημαντική θέση ανάμεσα σε αυτές έχουν οι κινητές συσκευές, οι οποίες χάρη στις προηγμένες δυνατότητές τους, αποτελούν ισχυρά εργαλεία αλληλεπίδρασης, πρόσβασης στη γνώση και δημιουργίας καινοτόμων εμπειριών. Ιδιαίτερη έμφαση δίνεται σήμερα στην ανάπτυξη τεχνολογιών που ενισχύουν την προσβασιμότητα και εξασφαλίζουν ίσες ευκαιρίες συμμετοχής για όλους τους πολίτες. Η χρήση της απτικής ανάδρασης, σε συνδυασμό με την οπτικοποίηση δεδομένων, προσφέρει νέες δυνατότητες για άτομα με προβλήματα ακοής, καθιστώντας τη μουσική εμπειρία πιο κατανοητή και βιωματική μέσω άλλων αισθήσεων. Η παρούσα διπλωματική εργασία εντάσσεται σε αυτό το πλαίσιο, καθώς διερευνά τον τρόπο με τον οποίο οι σύγχρονες τεχνολογικές εφαρμογές μπορούν να μετατρέψουν τη μουσική σε δονήσεις και οπτικές αναπαραστάσεις, ανοίγοντας νέους δρόμους για την κατανόηση και την απόλαυση της τέχνης πέρα από τα παραδοσιακά ακουστικά όρια.

## Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας είναι η διερεύνηση του τρόπου με τον οποίο μπορούν να αξιοποιηθούν σύγχρονες τεχνολογίες για τη βελτίωση της καθημερινότητας των ατόμων με προβλήματα ακοής. Η εργασία επικεντρώνεται στην ανάπτυξη μίας εφαρμογής κινητού τηλεφώνου, η οποία μετατρέπει τη μουσική σε δονήσεις, παρέχοντας μια εναλλακτική μορφή αντίληψης του ήχου μέσω της απτικής αλληλεπίδρασης. Επιπλέον, η εφαρμογή ενσωματώνει οπτική αναπαράσταση της μουσικής, ενισχύοντας την εμπειρία και επιτρέποντας στα άτομα με προβλήματα ακοής να προσεγγίσουν την τέχνη και την ψυχαγωγία με καινοτόμο τρόπο. Η εφαρμογή αναπτύχθηκε για συσκευές με λειτουργικό σύστημα Android και αξιολογήθηκε προκειμένου να εξαχθούν συμπεράσματα σχετικά με τη χρηστικότητα, την αποτελεσματικότητα, αλλά και τον αντίκτυπο της χρήσης απτικής και οπτικής τεχνολογίας στην κοινωνική ένταξη και στην πρόσβαση στην πολιτιστική δημιουργία.

# « Development of a Mobile Application that Converts Music into Vibrations, Utilizing Haptic and Visual Feedback for Individuals with Hearing Impairments. »

Tsourounake Zoe

## **Abstract**

The aim of this thesis is to explore how modern technologies can be utilized to improve the daily lives of people with hearing impairments. The work focuses on the development of a mobile application that converts music into vibrations, providing an alternative form of sound perception through haptic interaction. In addition, the application incorporates a visual representation of music, enhancing the experience and enabling people with hearing impairments to approach art and entertainment in an innovative way. The application was developed for devices running the Android operating system and was evaluated in order to draw conclusions regarding usability, effectiveness, and the impact of combining haptic and visual technologies on social inclusion and access to cultural creation.

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω τον πρώτα από όλα τον Θεό, έπειτα τον κύριο Κοκκώνη Γεώργιο για την ανάθεση της διπλωματικής εργασίας και την στήριξη του από την αρχή ως και την περάτωση της. Τέλος θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένεια μου και το αγόρι μου που με στηρίζουν τόσα χρόνια.

# Περιεχόμενα

Πρόλογος.....	iii
Περίληψη .....	iv
Abstract.....	v
Ευχαριστίες.....	vi
Περιεχόμενα .....	vii
Κατάλογος Εικόνων.....	x
Ελληνικές Συντομογραφίες.....	xiii
Ξένες Συντομογραφίες.....	xiv
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Σκοπός και Περιγραφή Εφαρμογής.....	1
1.2 Κίνητρα Υλοποίησης.....	1
1.3 Χρήση στην καθημερινότητα και στην τεχνολογία.....	1
1.4 Δομή της Εργασίας.....	2
Κεφάλαιο 2ο: Τι είναι το Visualizer .....	3
2.1 Εισαγωγή στην Οπτικοποίηση Δεδομένων.....	3
2.2 Τύποι οπτικοποιήσεων: waveform, spectrum bars, abstract patterns .....	3
2.3 Σχέση οπτικοποιήσεων με τη μουσική δομή και τον ρυθμό.....	4
2.4 Επιλογές χρωμάτων και σχήματα για καλύτερη κατανόηση της μουσικής οπτικά .....	5
2.5 Οπτικοποίηση μέσω Spectrogram.....	6
2.5 Συμπεράσματα .....	7
Κεφάλαιο 3ο: Τι είναι το Haptic Feedback.....	8
3.1 Εισαγωγή στην Ανάδραση Αφής – Ορισμοί, Ιστορική Εξέλιξη και Θεμελιώδεις Έννοιες.....	8
3.2 Τεχνολογίες που χρησιμοποιούνται για απτική δόνηση σε κινητά.....	8
3.3 Διαφορές Τύπων Δονήσεων (Ενταση, Διάρκεια, Μοτίβα).....	11
3.4 Απτική Μουσική Αντίληψη.....	12
3.4.1 Νευροφυσιολογική Βάση και Ψυχοφυσικά Μοντέλα Αφής.....	12
3.4.2 Τεχνολογικές Βάσεις και Απόδοση Απτικής Πληροφορίας.....	14
3.5 Μετατροπή Ηχητικών Σημάτων σε Απτικά Ερεθίσματα .....	16
3.5.1 Fourier.....	17
3.6 Η σημασία της αφής στην εμπειρία του χρήστη (UX) .....	18
3.6.1 Ρυθμική Δόνηση.....	18

3.6.2	Απόδοση Ύψους Ήχου (Pitch).....	19
3.6.3	Απτική Αναπαράσταση Μελωδίας.....	20
3.6.4	Απόδοση Χροιάς (Timbre).....	21
3.6.5	Απόδοση Έντασης (Loudness).....	22
3.7	Συμπεράσματα .....	23
Κεφάλαιο 4ο:	Haptic Music Player (HMP) με Visualizer.....	24
4.1	Εισαγωγή — Ορισμός και Ιστορική Αναδρομή.....	24
4.2	Συνολική Αρχιτεκτονική και Ροή Δεδομένων .....	24
4.3	Ενσωμάτωση Απτικής και Οπτικής Ανάδρασης.....	25
4.4	Διασύνδεση Συστήματος (Software & Hardware σε υψηλό επίπεδο) .....	26
4.5	Real-Time Συγχρονισμός και Προκλήσεις.....	28
4.6	Τεχνολογικές Υλοποιήσεις και Εφαρμογές.....	29
4.7	Συμπεράσματα .....	30
Κεφάλαιο 5ο:	Θεωρητικό Υπόβαθρο στην Αντίληψη των Ήχων και της Μουσικής από άτομα με προβλήματα ακοής.....	31
5.1	Εισαγωγή.....	31
5.2	Η Σημασία της Πολυαισθητηριακής Προσέγγισης στα Άτομα με Προβλήματα Ακοής. .....	31
5.3	Η Νευρολογική Βάση της Πολυαισθητηριακής Αντίληψης .....	32
5.3.1	Νευρωνική Κωδικοποίηση .....	33
5.4	Αντίληψη Ρυθμού και Δόνησης από Κωφούς/Βαρήκοους .....	34
5.4.1	Ψυχοακουστική – Ψυχοαφής Αντίληψη .....	34
5.4.2	Μοντέλα Αντίληψης Ρυθμού μέσω Δόνησης.....	35
5.4.3	Μουσικολογικές Εφαρμογές και Τεχνολογίες Δόνησης .....	37
5.5	Έρευνες για τη Μουσική Βιωσιμότητα μέσω Δονήσεων .....	37
5.6	Συμπεράσματα .....	38
Κεφάλαιο 6ο:	Σχεδιασμός της Εφαρμογής.....	39
6.1	Εισαγωγή.....	39
6.2	Άδειες Πρόσβασης, Λειτουργικότητα και App Icon .....	39
6.3	Περιγραφή Κώδικα Αρχικής Οθόνης (Home Page).....	42
6.3.1	XML .....	42
6.3.2	KOTLIN.....	43
6.4	Βιβλιοθήκη Τραγουδιών (Activity Library).....	49
6.4.1	XML .....	49
6.4.2	KOTLIN.....	51

6.4.3	SongsAdapter .....	54
6.5	Κατάσταση Αναπαραγωγής (Player Mode).....	58
6.5.1	XML .....	58
6.5.2	KOTLIN.....	62
6.6	Οπτικοποίηση Μουσικής (Visualizer View).....	70
6.7	Σελίδα Δόνησης Ρυθμού (Page RhythmVibrator).....	79
6.8	Διάγραμμα Ροής (Flowchart).....	83
6.9	Παρατηρήσεις και Συμπεράσματα .....	84
Κεφάλαιο 7ο:	Δοκιμές και Αξιολόγηση.....	86
7.1	Εισαγωγή.....	86
7.2	Ερωματολόγιο .....	86
7.3	Στατιστικά στοιχεία:.....	90
7.3.1	Δημογραφικά Στοιχεία.....	90
7.3.2	Αξιολόγηση Δονήσεων από Χρήστες .....	92
7.3.3	Αξιολόγηση Οπτικοποίησης από Χρήστες .....	95
7.3.4	Αξιολόγηση Συνδυασμού Οπτικοποίησης με Δονήσεις από Χρήστες – Γενική Εμπειρία .....	97
7.3.5	Προτάσεις και Παρατηρήσεις.....	99
7.4	Συμπεράσματα .....	99
Κεφάλαιο 8ο:	Συμπεράσματα και Δυνατότητες Βελτίωσης.....	100
BIBΛΙΟΓΡΑΦΙΑ.....		102

## Κατάλογος Εικόνων

Εικόνα 3.1: VCA [15].....	9
Εικόνα 3.2: LRA [17] .....	10
Εικόνα 3.3: Pizeoelectric [21] .....	39
Εικόνα 6.1: Mic Permission .....	40
Εικόνα 6.2: Media Permission.....	41
Εικόνα 6.3: Άδειες και εικονίδιο εφαρμογής .....	41
Εικόνα 6.4: Αρχική Οθόνη.....	42
Εικόνα 6.5: Διαχωρισμός Οθόνης xml.....	43
Εικόνα 6.6: Gif xml .....	43
Εικόνα 6.7: Gif .....	43
Εικόνα 6.8: MainActivity imports .....	44
Εικόνα 6.9: MainActivity μεταβλητές .....	44
Εικόνα 6.10: MainActivity onCreate .....	45
Εικόνα 6.11: MainActivity animation και onTouch .....	45
Εικόνα 6.12: MainActivity εσωτερική κλήση (swipe) .....	46
Εικόνα 6.13: MainActivity navigateToActivity().....	46
Εικόνα 6.14: MainActivity checkAndRequestPermissions() .....	47
Εικόνα 6.15: MainActivity onRequestPermissionsResult() .....	47
Εικόνα 6.16: MainActivity onPermissionsGranted ().....	48
Εικόνα 6.17: MainActivity arePermissionsGranted ().....	48
Εικόνα 6.18: ActivityLibrary xml.....	49
Εικόνα 6.19: Λίστα Τραγουδιών display στο κινητό.....	49
Εικόνα 6.20: ActivityLibrary1 xml.....	50
Εικόνα 6.21: ActivityLibrary2 xml.....	50
Εικόνα 6.22: ActivityLibrary imports.....	51
Εικόνα 6.23: ActivityLibrary μεταβλητές και onCreate().....	51
Εικόνα 6.24: ActivityLibrary loadSongs() 1 .....	52
Εικόνα 6.25: ActivityLibrary loadSongs() 2 .....	53
Εικόνα 6.26: ActivityLibrary openPlayerMode().....	54
Εικόνα 6.27: SongsAdapter xml .....	55
Εικόνα 6.28: SongsAdapter2 xml .....	55
Εικόνα 6.29: SongsAdapter3 xml.....	56
Εικόνα 6.30: SongsAdapter imports .....	56
Εικόνα 6.31: SongsAdapter.kt data class .....	57
Εικόνα 6.32: SongsAdapter.kt .....	57
Εικόνα 6.33: SongsAdapter.kt 2 .....	58
Εικόνα 6.34: PlayerMode xml 1 .....	58
Εικόνα 6.35: PlayerMode xml 2 .....	59
Εικόνα 6.36: PlayerMode xml 3 .....	59
Εικόνα 6.37: PlayerMode xml 4.....	60
Εικόνα 6.38: PlayerMode xml 5.....	60
Εικόνα 6.39: PlayerMode xml 6 .....	61
Εικόνα 6.40: PlayerMode xml 7 .....	61
Εικόνα 6.41: PlayerMode xml 8 .....	62

Εικόνα 6.42: PlayerMode xml 9 .....	62
Εικόνα 6.43: PlayerMode imports .....	63
Εικόνα 6.44: PlayerMode μεταβλητές .....	63
Εικόνα 6.45: PlayerMode onCreate() 1 .....	64
Εικόνα 6.46: PlayerMode onCreate() 2 .....	65
Εικόνα 6.47: PlayerMode onCreate() 3 .....	65
Εικόνα 6.48: PlayerMode playSong() 1 .....	66
Εικόνα 6.49: PlayerMode playSong() 2 .....	66
Εικόνα 6.50: PlayerMode updateSongInfo() .....	67
Εικόνα 6.51: PlayerMode playNextSong() .....	67
Εικόνα 6.52: PlayerMode playPreviousSong() .....	67
Εικόνα 6.53: PlayerMode updatePlayPauseButton() .....	68
Εικόνα 6.54: PlayerMode handlePlayPauseAction() .....	68
Εικόνα 6.55: PlayerMode updateStartTimeTextView() .....	69
Εικόνα 6.56: PlayerMode updateEndTimeTextView() .....	69
Εικόνα 6.57: PlayerMode startUpdatingTime() .....	69
Εικόνα 6.58: PlayerMode onPause() .....	70
Εικόνα 6.59: PlayerMode onDestroy() .....	70
Εικόνα 6.60: visualizer .....	71
Εικόνα 6.61: Visualizer imports .....	71
Εικόνα 6.62: Visualizer μεταβλητές .....	72
Εικόνα 6.63: Visualizer setOnBeatListener() .....	72
Εικόνα 6.64: Visualizer linkToMediaPlayer() .....	73
Εικόνα 6.65: Visualizer detectBeats() 1 .....	74
Εικόνα 6.66: Visualizer detectBeats() 2 .....	75
Εικόνα 6.67: Visualizer detectBeats() 3 .....	75
Εικόνα 6.68: Visualizer detectBeats() 4 .....	76
Εικόνα 6.69: Visualizer vibrateBasedOnBeat .....	77
Εικόνα 6.70: Visualizer colors .....	78
Εικόνα 6.71: Visualizer onDraw() .....	78
Εικόνα 6.72: Rhythm imports .....	79
Εικόνα 6.73: Rhythm vibrateToBeat() .....	80
Εικόνα 6.74: Rhythm vibrateWithWaveform() .....	81
Εικόνα 6.75: Rhythm stopVibration() .....	81
Εικόνα 6.76: Rhythm mapIntensityToDuration() .....	82
Εικόνα 6.77: Rhythm mapIntensityToDurationAndAmplitude() .....	83
Εικόνα 7.1: Διάγραμμα Ηλικιακής κατανομής .....	91
Εικόνα 7.2: Διάγραμμα κατανομής φύλου .....	91
Εικόνα 7.3: Διάγραμμα για άτομα με προβλήματα ακοής .....	92
Εικόνα 7.4: Διάγραμμα εμπειρίας χρηστών με ειδικές εφαρμογές μουσικής .....	92
Εικόνα 7.5: Διάγραμμα αξιολόγησης της ευχαρίστησης από τις δονήσεις .....	93
Εικόνα 7.6: Διάγραμμα αξιολόγησης διάρκειας δονήσεων .....	93
Εικόνα 7.7: Διάγραμμα αξιολόγησης αντιστοίχισης δονήσεων-έντασης .....	94
Εικόνα 7.8: Διάγραμμα ακρίβειας δονήσεων-ρυθμού .....	94
Εικόνα 7.9: Διάγραμμα διαφορών δονήσεων ανά είδος μουσικής .....	95
Εικόνα 7.10: Διάγραμμα αναγνωσιμότητας οπτικής αναπαράστασης .....	95

Εικόνα 7.11: Διάγραμμα επίδρασης οπτικών εφέ στον ρυθμό της μουσικής .....	96
Εικόνα 7.12: Διάγραμμα αντίληψης πολυπλοκότητας μέσω οπτικής αναπαράστασης .....	96
Εικόνα 7.13: Διάγραμμα οπτικής παρακολούθησης πολλαπλών χαρακτηριστικών μουσικής .....	97
Εικόνα 7.14: Διάγραμμα ευχρηστίας εφαρμογής .....	97
Εικόνα 7.15: Διάγραμμα αξιολόγησης ικανοποίησης εφαρμογής .....	98
Εικόνα 7.16: Διάγραμμα αξιολόγησης συγχρονισμού οπτικοποίησης–δονήσεων .....	98
Εικόνα 7.17: Διάγραμμα αξιολόγησης χρησιμότητας για άτομα με προβλήματα ακοής.....	99

## Ελληνικές Συντομογραφίες

B.M.	Βαθιά Μάθηση
Δ.Ε.	Διπλωματική Εργασία
ΔΙ.ΠΑ.Ε.	Διεθνές Πανεπιστήμιο Ελλάδος
M.M.	Μηχανική Μάθηση
Π.Ε.	Πτυχιακή Εργασία
T.N.	Τεχνητή Νοημοσύνη
Υ.Τ.	Υποβοηθητική Τεχνολογία

## Ξένες Συντομογραφίες

A.T.R.	Audio Tactile Rendering
C.A.S.R.	Contact Area Spread Rate
C.N.N.	Convolutional Neural Network
C.O.CO.	Common Objects in Context
D.A.C.	Digital to Analog Converter
D.F.T.	Discrete Fourier Transform
D.M.A.	Dual Mode Actuators
D.S.P.	Digital Signal Processing
E.C.G.	Electrocardiography
E.E.G.	Electroencephalography
E.R.M.	Eccentric Rotating Mass
F.F.T.	Fast Fourier Transform
G.M.M.	Gaussian Mixture Models
H.C.I.	Human Computer Interaction
H.M.P.	Haptic Music Players
J.N.D.	Just Noticeable Difference
L.R.A.	Linear Resonant Actuators
M.E.M.S.	Micro Electro Mechanical Systems
M.I.D.I.	Musical Instrument Digital Interface
M.R.I.	Magnetic Resonance Imaging
P.W.T.	Playing With Tagging
P.I.D.	Proportional Integral Derivative
Y.O.L.O.	You Only Look Once
V.C.A.	Voice Coil Actuators

## Κεφάλαιο 1ο: Εισαγωγή

### 1.1 Σκοπός και Περιγραφή Εφαρμογής

Η παρούσα εργασία έχει ως σκοπό τη μελέτη, ανάπτυξη και αξιολόγηση μιας εφαρμογής κινητού τηλεφώνου που μετατρέπει τη μουσική σε δονήσεις και την αποδίδει ταυτόχρονα μέσω οπτικής αναπαράστασης. Ο συνδυασμός αυτός στοχεύει στη δημιουργία μιας πολυαισθητηριακής εμπειρίας που απευθύνεται κυρίως σε άτομα με προβλήματα ακοής, επιτρέποντάς τους να «αισθανθούν» και να «δουν» τη μουσική, ακόμη και αν δεν μπορούν να την ακούσουν.

Η εφαρμογή αξιοποιεί τις δυνατότητες των σύγχρονων κινητών συσκευών, όπως τον ενσωματωμένο μηχανισμό δόνησης και την οθόνη υψηλής ανάλυσης, ώστε να μετατρέπει τα ηχητικά χαρακτηριστικά (ρυθμό, ένταση, συχνότητα) σε απτικά και οπτικά σήματα. Έτσι, η μουσική παύει να αποτελεί αποκλειστικά ακουστικό φαινόμενο και μετατρέπεται σε εμπειρία πολλαπλών αισθήσεων.

### 1.2 Κίνητρα Υλοποίησης

Το βασικό κίνητρο για την υλοποίηση της εφαρμογής είναι η ανάγκη για **προσβασιμότητα**. Τα άτομα με μερική ή ολική απώλεια ακοής στερούνται σε μεγάλο βαθμό τη μουσική εμπειρία, η οποία για τους περισσότερους ανθρώπους αποτελεί σημαντικό κομμάτι της καθημερινότητας, της ψυχαγωγίας και της συναισθηματικής έκφρασης. Η τεχνολογία μπορεί να λειτουργήσει ως γέφυρα που γεφυρώνει αυτό το χάσμα, δίνοντας σε αυτά τα άτομα τη δυνατότητα να αλληλεπιδράσουν με τη μουσική με διαφορετικό αλλά εξίσου ουσιαστικό τρόπο.

Επιπλέον, τα κίνητρα σχετίζονται με την **καινοτομία** στη χρήση των αισθήσεων. Η απτική και η οπτική πληροφορία, όταν συνδυάζονται, μπορούν να αναπαραστήσουν πολύπλοκα μουσικά μοτίβα και να εμπλουτίσουν τη μουσική εμπειρία όχι μόνο για τα άτομα με προβλήματα ακοής, αλλά και για όλους τους χρήστες που αναζητούν νέες μορφές αλληλεπίδρασης με την τέχνη.

### 1.3 Χρήση στην καθημερινότητα και στην τεχνολογία

Η εφαρμογή μπορεί να έχει πολλαπλές χρήσεις στην καθημερινή ζωή:

- **Ψυχαγωγία:** επιτρέπει στους χρήστες να βιώνουν τη μουσική με έναν νέο τρόπο, συνδυάζοντας εικόνα και αίσθηση.
- **Εκπαίδευση:** μπορεί να αξιοποιηθεί σε σχολεία κωφών και βαρήκοων, για να διδάξει ρυθμούς, μοτίβα και βασικές έννοιες της μουσικής μέσω αφής και εικόνας.

- **Καλλιτεχνικές εφαρμογές:** μπορεί να χρησιμοποιηθεί σε συναυλίες ή πολυμεσικές εγκαταστάσεις, προσφέροντας μια εναλλακτική εμπειρία κοινού που συνδυάζει τεχνολογία και τέχνη.
- **Τεχνολογική προέκταση:** ανοίγει τον δρόμο για περαιτέρω αξιοποίηση σε φορετές συσκευές (wearables), όπου η δόνηση μπορεί να ενισχύσει την εμπειρία του χρήστη σε εφαρμογές ψυχαγωγίας, υγείας ή εικονικής πραγματικότητας (VR/AR).

### 1.4 Δομή της Εργασίας

Η εργασία αναπτύσσεται σε οκτώ κεφάλαια. Στο πρώτο κεφάλαιο παρουσιάζονται ο σκοπός, τα κίνητρα και η χρησιμότητα της εφαρμογής. Στο δεύτερο κεφάλαιο αναλύεται η έννοια της οπτικοποίησης δεδομένων και οι μαθηματικές βάσεις που υποστηρίζουν τη μετάφραση του ήχου σε οπτικά ερεθίσματα. Στο τρίτο κεφάλαιο εξετάζεται η απτική ανάδραση, οι τεχνολογίες και τα μοντέλα που τη στηρίζουν, καθώς και η δυνατότητα απόδοσης μουσικών χαρακτηριστικών μέσω δονήσεων. Το τέταρτο κεφάλαιο περιγράφει την αρχιτεκτονική του Haptic Music Player, ενώ το πέμπτο εστιάζει στη θεωρητική κατανόηση της μουσικής από άτομα με προβλήματα ακοής. Στο έκτο κεφάλαιο παρουσιάζεται ο σχεδιασμός και η υλοποίηση της εφαρμογής, ενώ το έβδομο περιλαμβάνει τη διαδικασία δοκιμών και αξιολόγησης από χρήστες. Τέλος, στο όγδοο κεφάλαιο καταγράφονται τα συνολικά συμπεράσματα και προτείνονται δυνατότητες βελτίωσης και μελλοντικής ανάπτυξης.

## Κεφάλαιο 2ο: Τι είναι το Visualizer

### 2.1 Εισαγωγή στην Οπτικοποίηση Δεδομένων

Η οπτικοποίηση (visualization) είναι η διαδικασία μετατροπής αφηρημένων ή μη απτών δεδομένων σε οπτικές ή απτικές αναπαραστάσεις, με στόχο την ευκολότερη αντίληψη και κατανόησή τους. Στο πεδίο της ανάλυσης των δεδομένων, η οπτικοποίηση χρησιμοποιείται για την καλύτερη κατανόηση μεγάλων και περίπλοκων συνόλων πληροφορίας. Εφαρμόζεται σε διάφορους τομείς, όπως τα μαθηματικά, η επιστήμη και η μηχανική, αποκτώντας όλο μεγαλύτερη σημασία για τη λήψη αποφάσεων. Στο πλαίσιο της μουσικής, η οπτικοποίηση αναφέρεται στη διαδικασία μετατροπής των μουσικών χαρακτηριστικών (όπως η ένταση, ο ρυθμός, η συχνότητα και η χροιά) σε οπτικά ή απτικά σήματα, επιτρέποντας στους χρήστες να «δούν» τη μουσική [1].

Η διαδικασία αυτή μπορεί να συμβάλει στην καλύτερη κατανόηση των στοιχείων της μουσικής, ειδικά για άτομα με μερική ή ολική απώλεια ακοής, δίνοντάς τους την ευκαιρία να αλληλεπιδράσουν με τη μουσική μέσω άλλων αισθήσεων, όπως η όραση και η αφή. Τα μαθηματικά που αξιοποιούνται για την οπτικοποίηση της μουσικής περιλαμβάνουν τεχνικές αναλύσεις ήχου, όπως οι **Μετασχηματισμοί Fourier** και οι **αναλύσεις φασμάτων** (spectral analysis), οι οποίες επιτρέπουν την καταγραφή και την απεικόνιση των μουσικών χαρακτηριστικών στον χρόνο και στην συχνότητα [1],[2].

### 2.2 Τύποι οπτικοποιήσεων: waveform, spectrum bars, abstract patterns

Η οπτικοποίηση του ήχου αποτελεί μια θεμελιώδη τεχνική για την ενίσχυση της αντιληπτικής και γνωστικής εμπειρίας του χρήστη. Ανάλογα με τον στόχο και το πλαίσιο χρήσης, έχουν αναπτυχθεί διαφορετικοί τύποι γραφικών αναπαραστάσεων που μετασχηματίζουν το ακουστικό σήμα σε οπτική μορφή. Στην παρούσα ενότητα εξετάζονται τρεις βασικές κατηγορίες: **waveform**, **spectrum bars** και **abstract patterns** [1].

Η **waveform οπτικοποίηση** αποτυπώνει άμεσα το ηχητικό σήμα στον χρόνο. Σε διακριτό πεδίο, το σήμα εκφράζεται ως:

$$x[n], \quad n = 0, 1, 2, \dots, N-1 \quad (2.1)$$

- $x[n]$ : το διακριτό σήμα στον χρόνο, όπου  $n$  είναι δείκτης δειγμάτων (samples). [1]

ή, σε συνεχή μορφή, ως:

$$x(t), t \in \mathbb{R} \quad (2.2)$$

- $x(t)$ : το συνεχές σήμα στον χρόνο, όπου  $t$  είναι η συνεχής μεταβλητή του χρόνου. [1]

Ο οριζόντιος άξονας αναπαριστά τον χρόνο, ενώ ο κατακόρυφος το πλάτος (amplitude). Πρόκειται για μία από τις πιο παραδοσιακές και άμεσα κατανοητές μορφές οπτικοποίησης, καθώς αντανακλά με πιστότητα το αρχικό σήμα και χρησιμοποιείται εκτενώς σε λογισμικά ηχογράφησης, επεξεργασίας και ανάλυσης ήχου. Η μορφή αυτή παρέχει πληροφορίες σχετικά με την ένταση, τη διάρκεια και τις παύσεις του ήχου, χωρίς ωστόσο να αποκαλύπτει φασματικά χαρακτηριστικά όπως η συχνοτική κατανομή [3].

Αντίθετα, οι **spectrum bars** (ή “spectrogram-like” αναπαραστάσεις) εστιάζουν στο πεδίο της συχνότητας. Μέσω της εφαρμογής μετασχηματισμών, όπως ο ταχέως μετασχηματισμός Fourier (FFT), ο οποίος θα παρουσιαστεί αναλυτικά στο Κεφ. 3.5.1, το ακουστικό σήμα αναλύεται σε συνιστώσες συχνοτήτων και αποδίδεται γραφικά σε μορφή κάθετων ράβδων ή συνεχόμενων καμπυλών. Οι αναπαραστάσεις αυτές είναι ιδιαίτερες χρήσιμες για τη μελέτη της τονικής δομής, τον εντοπισμό θεμελιωδών συχνοτήτων και την οπτική παρακολούθηση της μουσικής δυναμικής [4].

Τέλος, τα **abstract patterns** αποτελούν μια πιο καλλιτεχνική και πειραματική προσέγγιση. Σε αυτή την κατηγορία, ο ήχος μεταφράζεται σε δυναμικά σχήματα, χρώματα ή γεωμετρικές μορφές που δεν στοχεύουν απαραίτητα στη γραμμική απεικόνιση των ακουστικών χαρακτηριστικών, αλλά στη δημιουργία μιας οπτικής εμπειρίας που συνοδεύει και ενισχύει τη μουσική. Τέτοιου είδους οπτικοποιήσεις συναντώνται κυρίως σε εφαρμογές ψυχαγωγίας, συναυλίες και πολυμεσικά έργα, όπου προτεραιότητα έχει η αισθητική και η συναισθηματική διάσταση της εμπειρίας [5].

Συνοψίζοντας, οι τρεις αυτοί τύποι οπτικοποίησης καλύπτουν διαφορετικά επίπεδα αναπαράστασης: από την πιστή αποτύπωση της κυματομορφής, στη συχνοτική ανάλυση και, τέλος, στην αφηρημένη και καλλιτεχνική απεικόνιση. Η επιλογή του κατάλληλου τύπου εξαρτάται από τον σκοπό της εφαρμογής και τις ανάγκες του τελικού χρήστη.

### 2.3 Σχέση οπτικοποιήσεων με τη μουσική δομή και τον ρυθμό

Η μουσική δομή και ο ρυθμός αποτελούν θεμελιώδη στοιχεία της μουσικής εμπειρίας, καθώς οργανώνουν το χρόνο, την ανάβαση των μελωδικών φράσεων, αλλά και τον τρόπο αντίληψης της μουσικής συνοχής. Τα τρία βασικά είδη οπτικοποιήσεων που παρουσιάστηκαν στην προηγούμενη ενότητα **waveform**, **spectrum bars** και **abstract patterns** συνεισφέρουν με διαφορετικούς τρόπους στην κατανόηση των ρυθμικών και δομικών χαρακτηριστικών της μουσικής.

Η **waveform** λειτουργεί ως άμεση απεικόνιση των χρονικών γεγονότων, διευκολύνοντας τον εντοπισμό χτύπων, παύσεων και μεταβολών στην ένταση. Η δυνατότητα να αναγνωρίζονται κορυφές πλάτους επιτρέπει την ανάλυση της ρυθμικής αγωγής (tempo) και τη σύγκριση διαφορετικών τμημάτων ενός κομματιού. Στην πράξη, αυτή η μορφή χρησιμοποιείται συχνά σε εργαλεία ανάλυσης

ανίχνευσης ρυθμού και διαχωρισμού μουσικών τμημάτων, όπου η μουσική δομή αναγνωρίζεται μέσω σημαντικών σημείων αλλαγής μέσα στο κομμάτι [3].

Αντίθετα, οι **spectrum bars** προσφέρουν ένα πιο σύνθετο μέσο για την κατανόηση της μουσικής δομής, καθώς αποκαλύπτουν την περιοδικότητα των χαμηλών συχνοτήτων που σχετίζεται με τον ρυθμό, αλλά και την οργάνωση των τονικών περιοχών που συμβάλλουν στη μορφολογική ανάλυση. Με τον τρόπο αυτό, οι φασματικές αναπαραστάσεις έχουν αξιοποιηθεί σε μελέτες αυτόματης ανάλυσης της μουσικής δομής (structural segmentation), όπου οι μεταβολές στο φασματικό περιεχόμενο λειτουργούν ως ενδείξεις για τον διαχωρισμό του κομματιού σε διακριτά τμήματα [4].

Τα **abstract patterns**, αν και λιγότερο αναλυτικά σε σχέση με τα προηγούμενα, συνδέονται με τον ρυθμό και τη δομή σε ένα πιο αντιληπτικό και εμπειρικό επίπεδο. Μέσω γεωμετρικών σχημάτων ή δυναμικών κινήσεων που αντιδρούν στον ρυθμό, ενισχύεται η υποκειμενική αίσθηση της περιοδικότητας και των δομικών μεταβάσεων. Τα **abstract patterns**, αν και λιγότερο αναλυτικά σε σχέση με τα προηγούμενα, συνδέονται με τον ρυθμό και τη δομή σε ένα πιο αντιληπτικό και εμπειρικό επίπεδο. Μέσω γεωμετρικών σχημάτων ή δυναμικών κινήσεων που αντιδρούν στον ρυθμό, ενισχύεται η υποκειμενική αίσθηση της περιοδικότητας και των δομικών μεταβάσεων. Συνεπώς, αυτού του είδους οι οπτικοποιήσεις συχνά αξιοποιούνται σε περιβάλλοντα όπου στοχεύουν στη δημιουργία μιας άμεσης και έντονης εμπειρίας της μουσικής, όπως σε συναυλίες και πολυμεσικά έργα [5].

Συνολικά, οι τρεις τύποι οπτικοποίησης δεν λειτουργούν απλώς ως αισθητικές απεικονίσεις του ήχου, αλλά συμβάλλουν ουσιαστικά στην κατανόηση της μουσικής δομής και του ρυθμού σε διαφορετικά επίπεδα: από την αντικειμενική χαρτογράφηση γεγονότων (waveform), στην ανάλυση συχνοτικών μοτίβων (spectrum bars), έως την αισθητηριακή ενίσχυση της ρυθμικής εμπειρίας (abstract patterns).

## 2.4 Επιλογές χρωμάτων και σχήματα για καλύτερη κατανόηση της μουσικής οπτικά

Σε αυτήν την παράγραφο εξετάζονται οι παράμετροι σύζευξης χρωμάτων και σχημάτων με εκφραστικά μουσικά χαρακτηριστικά. Η επιλογή τους δεν λειτουργεί μόνο ως αισθητικό συμπλήρωμα, αλλά και ως εργαλείο κατανόησης της μουσικής για διαφορετικές κατηγορίες χρηστών. Ιδιαίτερη σημασία αποκτά για άτομα με προβλήματα ακοής, τα οποία μπορούν να προσεγγίσουν τη μουσική εμπειρία μέσω της οπτικής διάστασης.

### Χρωματικές επιλογές και προσβασιμότητα

Η χρήση χρωμάτων σε waveforms και spectrum bars δεν περιορίζεται στην καλλιτεχνική διάσταση, αλλά λειτουργεί ως “μεταφραστής” ακουστικών διαφορών σε οπτικά σήματα. Η εφαρμογή αποχρώσεων που εκτείνονται από ψυχρά (μπλε) έως θερμά (κόκκινα) χρώματα μπορεί να αντιπροσωπεύει εντάσεις ή συναισθηματικές μεταπτώσεις. Για παράδειγμα, οι υψηλές συχνότητες μπορούν να απεικονιστούν με φωτεινές μπλε ή μωβ αποχρώσεις και συνδέονται με “λυπηρές” καταστάσεις, ενώ οι χαμηλές με θερμά χρώματα όπως κόκκινο, πορτοκαλί και συνδέονται με “χαρούμενες”. Αυτό επιτρέπει σε άτομα με απώλεια ακοής να κατανοήσουν την πολυπλοκότητα του ήχου χωρίς να χρειάζεται να τον ακούσουν άμεσα. Έρευνες έχουν δείξει ότι η χρωματική

διαφοροποίηση βοηθά στην αναγνώριση ρυθμικών και τονικών μοτίβων, δημιουργώντας μια "οπτική μουσική γραμματική" [6].

Παράλληλα, οι χρωματικοί κύκλοι που απεικονίζουν τόνους ή κλίμακες προσφέρουν έναν τρόπο στους χρήστες με περιορισμένη ακουστική πρόσβαση να αναγνωρίζουν δομικές μεταβολές στη μουσική. Με τον τρόπο αυτό, οι αλλαγές τονικότητας δεν γίνονται μόνο αντιληπτές ακουστικά, αλλά και οπτικά [7].

### **Συμβολική χρήση σχημάτων**

Η εφαρμογή γεωμετρικών μορφών επιτρέπει την απόδοση ρυθμικών ή μελωδικών χαρακτηριστικών σε οπτικό επίπεδο. Η χρήση αφηρημένων σχημάτων, όπως αιχμηρά ή στρογγυλεμένα μοτίβα, βασίζεται σε γνωστικά φαινόμενα όπως το Brouha/Kiki effect, που δείχνουν ότι οι άνθρωποι συνδέουν τις αισθήσεις του ήχου με οπτικές μορφές [8]. Για τους χρήστες με προβλήματα ακοής, τέτοιες αντιστοιχίσεις διευκολύνουν τη δημιουργία μιας εναλλακτικής, πολυαισθητηριακής αναπαράστασης της μουσικής.

Επίσης, οι δυναμικές μορφές που κινούνται στον χώρο, όπως στο παράδειγμα του "Kinetic Lighting for Music Visualization" [9] ενισχύουν την κατανόηση της ροής και του ρυθμού μέσω φωτός και κίνησης. Αυτό προσφέρει στα άτομα με ακουστική αναπηρία μια βιωματική πρόσβαση στη μουσική, μετατρέποντας την ηχητική ενέργεια σε οπτικοκινητικό γεγονός.

### **Συνδυασμός χρωμάτων και σχημάτων για ενισχυμένη κατανόηση**

Ο συνδυασμός χρωμάτων και σχημάτων στην οπτικοποίηση αποτελεί κρίσιμο παράγοντα για την ενίσχυση της κατανόησης και την αποτελεσματική επικοινωνία των δεδομένων. Η σύζευξη αυτών των δύο στοιχείων δεν λειτουργεί απλώς διακοσμητικά, αλλά προσδίδει μια πολυεπίπεδη διάσταση στην ερμηνεία της πληροφορίας, καθώς η αλληλεπίδρασή τους μπορεί να διαφοροποιήσει, να ιεραρχήσει και να καθοδηγήσει την προσοχή του θεατή σε συγκεκριμένα σημεία. Ιδιαίτερη σημασία έχει αυτός ο σχεδιαστικός συνδυασμός για άτομα με προβλήματα ακοής, τα οποία στηρίζονται σε μεγαλύτερο βαθμό στα οπτικά ερεθίσματα για την πρόσληψη και κατανόηση της πληροφορίας. Μέσω της προσεκτικής τους συνδυαστικότητας, επιτυγχάνεται όχι μόνο η αποφυγή γνωστικής υπερφόρτωσης αλλά και η ενίσχυση της αναγνωσιμότητας, ιδίως σε σύνθετα ή πολυδιάστατα σύνολα δεδομένων. Επομένως, η στρατηγική αξιοποίηση των χρωματικών τόνων σε συνάρτηση με τις μορφολογικές διαφοροποιήσεις των σχημάτων συνιστά αναπόσπαστο εργαλείο για την ανάπτυξη οπτικοποιήσεων που ισορροπούν ανάμεσα στην αισθητική, την επιστημονική ακρίβεια και την προσβασιμότητα.

## **2.5 Οπτικοποίηση μέσω Spectrogram**

Η οπτικοποίηση του ήχου μέσω spectrogram αποτελεί μία από τις πιο διαδεδομένες μεθόδους ανάλυσης σήματος, καθώς παρέχει ταυτόχρονα πληροφορία για τη συχνότητα, την ένταση και τον χρόνο. Το spectrogram προκύπτει από τη διαδοχική εφαρμογή του Μετασχηματισμού Fourier σε τμήματα (παράθυρα) του σήματος, γνωστή και ως Short-Time Fourier Transform (STFT). Με αυτόν τον τρόπο, κάθε χρονικό τμήμα του σήματος μετατρέπεται σε φάσμα συχνοτήτων και η απεικόνιση

αποδίδεται σε μορφή τριδιάστατης πληροφορίας: ο άξονας  $x$  αντιπροσωπεύει τον χρόνο, ο άξονας  $y$  τη συχνότητα, ενώ η ένταση (ή το χρώμα) αποτυπώνει το πλάτος των συχνοτήτων [10]. Τυπικά, το spectrogram ορίζεται ως:

$$S(t,f) = \left| \sum_{n=-\infty}^{\infty} x[n] \cdot w[n - \tau] \cdot e^{-j2\pi f n} \right|^2 \quad (2.3)$$

Όπου:

- $x[n]$ : το διακριτό σήμα εισόδου,
- $w[n]$ : η συνάρτηση **παραθύρου**, η οποία «κόβει» το σήμα σε μικρά τμήματα ώστε να μπορέσουμε να δούμε πώς αλλάζει το φάσμα του στον χρόνο. Συνήθως χρησιμοποιούνται παράθυρα Hamming, Hanning, Blackman κ.ά.
- $\tau$ : συνεχής μεταβλητή χρόνου
- $f$ : συχνότητα (σε Hz) στη διακριτή μορφή,
- $n$ : διακριτός δείκτης χρόνου. [10]

Η αναπαράσταση αυτή είναι ιδιαίτερα χρήσιμη για την κατανόηση μουσικών μοτίβων ή ρυθμικών στοιχείων από άτομα με προβλήματα ακοής, καθώς τα χρώματα και τα σχήματα που προκύπτουν αποδίδουν οπτικά την «δομή» του ήχου. Στο πλαίσιο της παρούσας εργασίας, το spectrogram λειτουργεί ως βασικό εργαλείο οπτικοποίησης, υποστηρίζοντας την απτική διάσταση της εφαρμογής και διευκολύνοντας την πολυαισθητηριακή εμπειρία του χρήστη.

## 2.5 Συμπεράσματα

Το δεύτερο κεφάλαιο ανέδειξε τη σημασία της οπτικοποίησης ως μέσο κατανόησης και εμπλουτισμού της μουσικής εμπειρίας, ειδικά για άτομα με προβλήματα ακοής. Παρουσιάστηκαν τρεις βασικές κατηγορίες αναπαράστασης: η κυματομορφή, οι φασματικές μπάρες και τα αφηρημένα μοτίβα. Κάθε μία με διαφορετικά πλεονεκτήματα όσον αφορά την ανάδειξη του ρυθμού, της δομής και της έντασης του ήχου. Ιδιαίτερη βαρύτητα δόθηκε στη χρήση χρωμάτων και σχημάτων, τα οποία λειτουργούν ως οπτικοί «μεταφραστές» των ακουστικών χαρακτηριστικών, διευκολύνοντας την πρόσβαση στη μουσική μέσω της όρασης.

Τέλος, ιδιαίτερη αναφορά έγινε στο spectrogram το οποίο αποτελεί την πλέον ολοκληρωμένη μέθοδο οπτικοποίησης, συνδυάζοντας χρόνο, συχνότητα και ένταση σε μία ενιαία αναπαράσταση. Στο πλαίσιο της παρούσας εργασίας, το spectrogram δεν αποτελεί μόνο εργαλείο τεχνικής ανάλυσης, αλλά και βασικό στοιχείο σύνδεσης της οπτικής με την απτική διεπαφή. Έτσι, συμβάλλει στη δημιουργία μιας πολυαισθητηριακής εμπειρίας, που ενισχύει την κατανόηση και την αλληλεπίδραση με τη μουσική πέρα από τα όρια της ακουστικής πρόσληψης.

## Κεφάλαιο 3ο: Τι είναι το Haptic Feedback

### 3.1 Εισαγωγή στην Ανάδραση Αφής – Ορισμοί, Ιστορική Εξέλιξη και Θεμελιώδεις Έννοιες

Η έννοια της **απτικής ανατροφοδότησης** (*haptic feedback*) αναφέρεται στην αξιοποίηση της **αφής** ως μέσου διαδραστικής πληροφορίας μεταξύ ανθρώπου και μηχανής. Πρόκειται για μία τεχνολογία που βασίζεται στη **μηχανική διέγερση** του ανθρώπινου σώματος μέσω **δυνάμεων, δονήσεων ή πιέσεων**, με σκοπό τη **μεταφορά πληροφορίας, την ενίσχυση της φυσικότητας και τη βελτίωση της απόδοσης** κατά την αλληλεπίδραση με εικονικά ή φυσικά περιβάλλοντα [7], [8]. Η αλληλεπίδραση αυτή δεν περιορίζεται μόνο στην **επιφανειακή αφή** (*cutaneous feedback*), αλλά επεκτείνεται και στην **κινητική ανάδραση** (*kinesthetic feedback*), η οποία αφορά την αντίληψη δύναμης, θέσης και κίνησης μέσω ιδιοδεκτικών μηχανισμών [11].

Ο όρος "haptic" προέρχεται από το αρχαιοελληνικό ρήμα "ἅπτομαι" και περιλαμβάνει κάθε μορφή αισθητικής πληροφορίας που προέρχεται από το **απτικό σύστημα**, το οποίο περιλαμβάνει τόσο **μηχανοποδοχείς του δέρματος** όσο και **ιδιοδεκτικούς μηχανισμούς** σε μύες, τένοντες και αρθρώσεις [12]. Το ανθρώπινο σύστημα αφής μπορεί να διακρίνει συχνότητες έως και ~500 Hz, καθιστώντας το κατάλληλο για την αντίληψη πολλών στοιχείων της μουσικής και του ήχου μέσω δόνησης [4].

Η ιστορική εξέλιξη της ανάδρασης αφής ξεκινά από εφαρμογές σε αεροπλάνα κατά τον Β΄ Παγκόσμιο Πόλεμο, όπου κραδασμοί στο χειριστήριο βοηθούσαν τον πιλότο να αντιληφθεί επικίνδυνες καταστάσεις [1], [2].

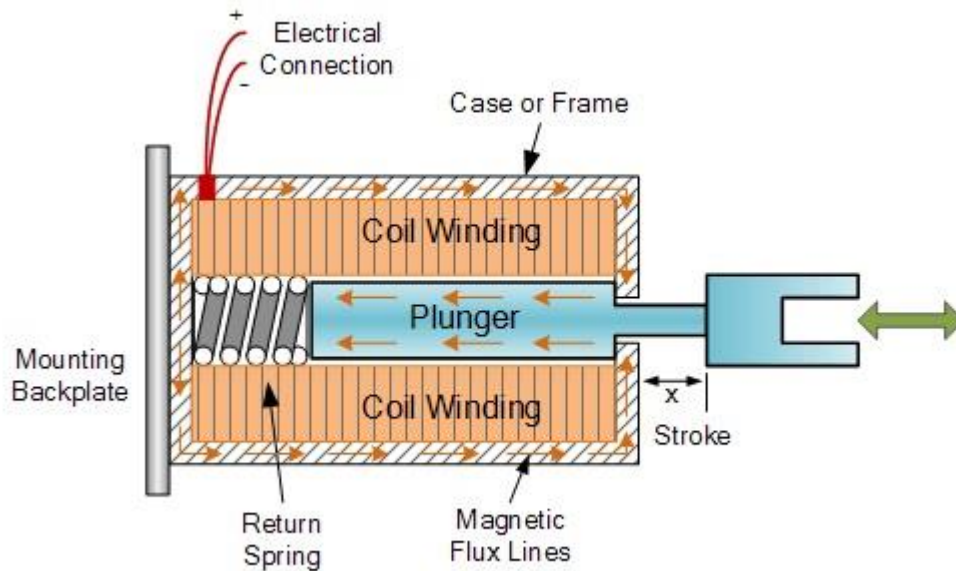
Η ιστορία των δονήσεων ως μέσο επικοινωνίας και αλληλεπίδρασης ξεκινάει από απλές μηχανικές συσκευές που χρησιμοποιούσαν δονήσεις για ειδοποιήσεις, όπως τα κινητά τηλέφωνα της δεκαετίας του 1990. Στη συνέχεια, η τεχνολογία εξελίχθηκε στο λεγόμενο **haptic feedback**, που επιτρέπει στους χρήστες να αισθάνονται φυσικές αντιδράσεις σε ψηφιακά περιβάλλοντα. Τα πρώτα συστήματα haptic χρησιμοποιήθηκαν σε προσομοιωτές πτήσης και χειρουργικής εκπαίδευσης, ενώ σήμερα η τεχνολογία εφαρμόζεται σε smartphones, παιχνίδια, VR/AR και wearable συσκευές, προσφέροντας μια πιο ρεαλιστική και διαδραστική εμπειρία [13].

### 3.2 Τεχνολογίες που χρησιμοποιούνται για απτική δόνηση σε κινητά

Στις σύγχρονες κινητές συσκευές, η δημιουργία απτικών ερεθισμάτων δηλαδή οι δονήσεις που ενισχύουν την αλληλεπίδραση με τον χρήστη βασίζεται σε διαφορετικές **τεχνολογίες ενεργοποιητών**, οι οποίες διαφοροποιούνται ως προς το μηχανισμό δράσης, την ακρίβεια, την κατανάλωση ενέργειας και το κόστος. Οι πιο διαδεδομένοι **τύποι ενεργοποιητών** περιλαμβάνουν:

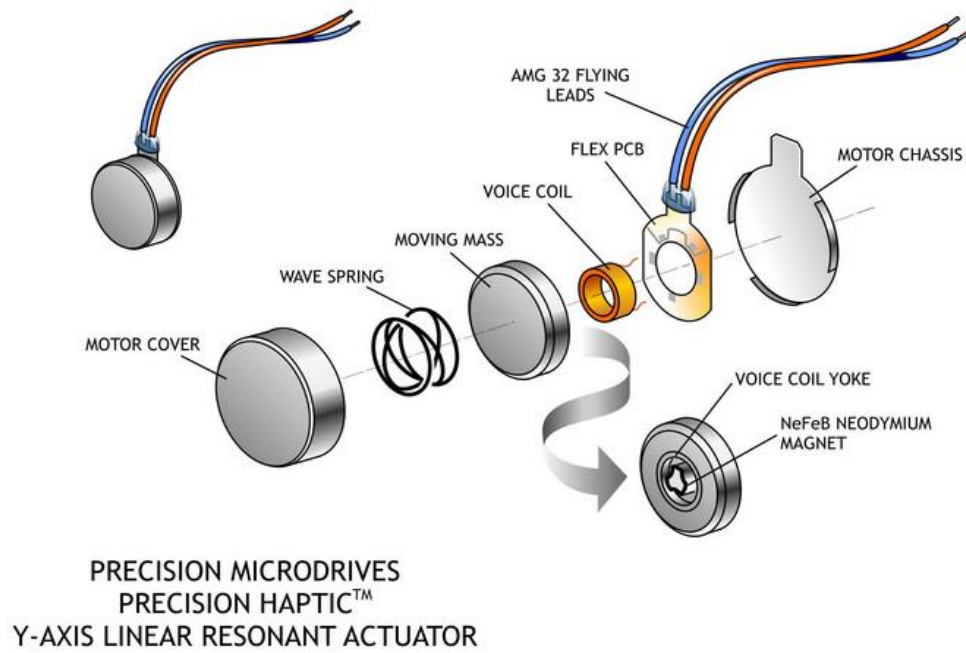
1. **Voice Coil Actuators (VCA):** Οι VCAs μεταφράζουν απευθείας το ηλεκτρικό ηχητικό σήμα σε μηχανικές δονήσεις. Η συχνότητα του εισερχόμενου ήχου αντιστοιχεί στη συχνότητα της

δόνησης, ενώ η ένταση εκφράζεται στο πλάτος αυτής, επιτρέποντας πιστή αναπαραγωγή μουσικών χαρακτηριστικών. Διακρίνονται για ταχεία απόκριση, διαχείριση δυναμικών μεταβολών και χαμηλό κόστος, καθιστώντας τα κατάλληλα για εμπορικές και πειραματικές εφαρμογές [14].



Εικόνα 3.1: VCA [15]

2. **Linear Resonant Actuators (LRA):** Αξιοποιούν ένα γραμμικό σύστημα μαγνήτη και ελατηρίου με βάση την αρχή της συντονισμένης ταλάντωσης, βελτιστοποιημένοι για συγκεκριμένο εύρος συχνοτήτων. Έχουν υψηλή αποδοτικότητα και χαμηλή κατανάλωση ενέργειας προσφέροντας ταχύτερη και πιο καθαρή απτική εμπειρία., ιδανικά για φορετές εφαρμογές όπου απαιτείται ελαχιστοποίηση όγκου και κατανάλωσης. Γι' αυτόν τον λόγο έχουν υιοθετηθεί ευρέως στα σύγχρονα smartphones. Ωστόσο, το στενό φάσμα συχνοτήτων τους απαιτεί προσεκτική αντιστοίχιση με το απτικό εύρος αντίληψης του ανθρώπινου δέρματος [16].



Εικόνα 3.2: LRA [17]

3. **Eccentric Rotating Mass (ERM):** Λειτουργούν μέσω περιστρεφόμενης μάζας εκτός κέντρου, προκαλώντας δόνηση κατά την κίνηση του άξονα. Παρότι είναι οικονομικοί και ευρέως διαδεδομένοι σε φορητές συσκευές όπως κινητά τηλέφωνα, δεν επιτρέπουν ανεξάρτητο έλεγχο συχνότητας και έντασης, έχουν περιορισμένη ακρίβεια, έχουν υψηλή κατανάλωση ενέργειας και αργή απόκριση [18],[19].
4. **Πιεζοηλεκτρικοί ενεργοποιητές:** Βασίζονται σε υλικά που μεταβάλλουν το σχήμα τους όταν εφαρμόζεται ηλεκτρικό πεδίο, για παραγωγή δονήσεων υψηλής ανάλυσης και ευρείας απόκρισης συχνοτήτων. Παρουσιάζουν εξαιρετική ακρίβεια και μπορούν να αποδώσουν σύνθετα μουσικά μοτίβα, αλλά απαιτούν εξειδικευμένα κυκλώματα οδήγησης και έχουν περιορισμένο πλάτος δόνησης [20].



Εικόνα 3.3: Piezoelectric [21]

5. **Dual Mode Actuators (DMA):** Σχεδιασμένοι να παράγουν ταυτόχρονα δύο διαφορετικές συχνότητες, επιτρέποντας την απόδοση πιο σύνθετων μουσικών σημάτων και βελτιώνοντας την απτική εμπειρία. Διακρίνονται για υψηλή ενεργειακή απόδοση και δυνατότητα λεπτομερούς ελέγχου [22].
6. **Ηχεία (Subwoofers):** Παρόλο που προορίζονται για ακουστική αναπαραγωγή, μπορούν να χρησιμοποιηθούν και για δονητική απόδοση, ειδικά σε απτικές καρέκλες και πλατφόρμες. Προσφέρουν ευρύ φάσμα συχνοτήτων και ισχυρές δονήσεις, αλλά το μέγεθος και η ενεργειακή κατανάλωση περιορίζουν τη χρήση τους σε φορητές συσκευές [23].

Πέραν αυτών, οι πρόσφατες έρευνες εστιάζουν σε νέες μορφές ενεργοποιητών, όπως πιεζοηλεκτρικά thin-film υλικά, lead-free συνθέσεις για περιβαλλοντική βιωσιμότητα, καθώς και υπερηχητικούς μετατροπείς (ultrasonic transducers) για ανέπαφη αλληλεπίδραση [5], [6]. Η εξέλιξη αυτών των τεχνολογιών δείχνει την τάση προς πιο ρεαλιστική, ακριβή και οικολογικά βιώσιμη απτική ανάδραση, η οποία αναμένεται να ενισχύσει περαιτέρω την εμπειρία χρήσης στα μελλοντικά κινητά.

### 3.3 Διαφορές Τύπων Δονήσεων (Ένταση, Διάρκεια, Μοτίβα)

Η απτική ανάδραση (haptic feedback) στα κινητά βασίζεται σε τρεις κύριες διαστάσεις: ένταση (intensity), διάρκεια (duration) και μοτίβα (patterns), οι οποίες διαμορφώνουν τη συνολική εμπειρία αφής του χρήστη. Η **ένταση** αναφέρεται στο μέγεθος της δόνησης και καθορίζεται από την τιμή της επιταχυνσιμότητας ή της πίεσης που αισθάνεται το δέρμα. Στις ψυχοφυσικές μελέτες, έχει διαπιστωθεί ότι οι χρήστες μπορούν να διακρίνουν περίπου 3–5 επίπεδα έντασης, με διαφορές αντίληψης έως 2,5 dB, ενώ η μέγιστη ευαισθησία του δέρματος σε δονήσεις εντοπίζεται γύρω στα 200–300 Hz [24].

Η **διάρκεια** της δόνησης επηρεάζει επίσης την αντίληψη: μεγαλύτερες χρονικές διάρκειες συχνά αντιλαμβάνονται ως πιο έντονες, αλλά μπορεί να επιφέρουν αποστέρηση αισθητηριακής αντίδρασης ή ακόμα και ερεθισμό, ειδικά όταν υπερβαίνει τα 200 ms [25]. Ειδικότερα, μελέτες καταγράφουν ότι διάστημα ελέγχου δόνησης μεταξύ 50 και 200 ms θεωρείται βέλτιστο για ευχάριστη αντίληψη χωρίς ερεθισμό [26].

Τα **μοτίβα** δόνησης συνιστούν σύνθεση μεταβολών στις παραμέτρους έντασης και διάρκειας, δημιουργώντας πιο σύνθετες αφαιρετικές δομές μεταφοράς πληροφορίας. Πειράματα με διαδοχικές (sequential) και ταυτόχρονες (simultaneous) παρουσιάσεις μοτίβων στον βραχίονα έδειξαν ότι η αναγνώριση ακολουθιών είναι σημαντικά πιο ακριβής όταν οι δονήσεις εμφανίζονται διαδοχικά περίπου 97% για σύντομες και 89% για μακρές ακολουθίες σε σχέση με ταυτόχρονες, όπου η ακρίβεια φθάνει μόνο στο 15–38% [27]. Η χρήση υψηλής έντασης σε συνδυασμό με μικρό ISI αυξάνει την αντίληψη επείγοντος αισθητηριακού μηνύματος [28].

Συνοψίζοντας, η διακύμανση στην ένταση, τη διάρκεια και τα μοτίβα (όπως ακολουθίες ή διαλείμματα) αποτελεί θεμέλιο για την σχεδίαση αποδοτικών, κατανοητών και άνετων απτικών συστημάτων. Η ρύθμιση της έντασης δίνει την αίσθηση δύναμης, η διάρκεια ελέγχει την ευχρηστία και αποφυγή υπερέρεσης, ενώ τα μοτίβα επιτρέπουν τη μεταφορά πιο σύνθετων πληροφοριών.

### 3.4 Απτική Μουσική Αντίληψη

Η διασύνδεση ήχου και απτικής ανάδρασης είναι μια **διεπιστημονική πρόκληση**, καθώς απαιτεί γνώσεις από την **ψυχοακουστική**, τη **μηχανολογία**, τη **νευροεπιστήμη** και τη **σχεδίαση διεπαφών ανθρώπου-μηχανής (HCI)**. Οι τρέχουσες έρευνες διερευνούν τρόπους βελτιστοποίησης της αντιστοιχίας ανάμεσα σε **χαρακτηριστικά του ήχου** (όπως ένταση, ρυθμός, αρμονικές) και **απτικά μοτίβα** ώστε να επιτευχθεί **φυσική, κατανοητή και χρήσιμη εμπειρία** για τον χρήστη [29].

Η τεχνολογία αυτή αξιοποιείται σε εφαρμογές όπου η **ακουστική πληροφόρηση πρέπει να αποδοθεί μέσω αφής**, όπως για παράδειγμα σε περιβάλλοντα με υψηλό θόρυβο, σε άτομα με **βαρηκοΐα** ή **κώφωση**, ή σε συστήματα επαυξημένης εμπειρίας χρήστη (όπως στα βιντεοπαιχνίδια και την εικονική πραγματικότητα). Στις περιπτώσεις αυτές, το **ηχητικό σήμα** αναλύεται και μετατρέπεται σε **δονήσεις με συγκεκριμένα χαρακτηριστικά συχνότητας και έντασης**, που εφαρμόζονται στο σώμα μέσω απτικών διεπαφών (π.χ. απτικών μετατροπέων – *tactile transducers*) [30], [31].

#### 3.4.1 Νευροφυσιολογική Βάση και Ψυχοφυσικά Μοντέλα Αφής

Η απτική αντίληψη (*haptic perception*) είναι η ικανότητα του ανθρώπινου νευρικού συστήματος να ανιχνεύει, να μεταδίδει και να ερμηνεύει πληροφορίες αφής, υφής, θερμοκρασίας, πίεσης και δόνησης μέσω εξειδικευμένων υποδοχέων του δέρματος και του ιδιοδεκτικού συστήματος. Η λειτουργία αυτή αποτελεί συνδυασμό **περιφερικών αισθητηρίων μηχανισμών και κεντρικών νευρωνικών διεργασιών**, που επεξεργάζονται τα σήματα από το δέρμα μέχρι τον εγκεφαλικό φλοιό [32].

Η ανθρώπινη απτική αίσθηση θεμελιώνεται σε τέσσερις κύριους **τύπους μηχανοϋποδοχέων**, οι οποίοι διαφοροποιούνται ως προς τα φυσικά ερεθίσματα στα οποία αποκρίνονται, τον ρυθμό προσαρμογής τους, καθώς και την τοπογραφική τους κατανομή στο σώμα.

- **Merkel** (πίεση). Οι **υποδοχείς (Merkel cells, SA I)** αποτελούν βραδέως προσαρμοζόμενους μηχανοϋποδοχείς που ανταποκρίνονται σε στατικά ερεθίσματα χαμηλής συχνότητας (<5 Hz). Εξειδικεύονται στην ανίχνευση της στατικής πίεσης και της μορφολογίας της επιφάνειας, συμβάλλοντας κατ' αυτόν τον τρόπο στην ακριβή διάκριση υφών, ακμών και σχημάτων. Η υψηλή τους συγκέντρωση στις άκρες των δακτύλων και στα χείλη καθιστά δυνατή την αναγνώριση μικρολεπτομερειών, όπως στην ανάγνωση της γραφής Braille [33].
- **Meissner** (ελαφρές δονήσεις **10-50 Hz**). Χαρακτηρίζονται από ταχεία προσαρμογή (RA I) και αποκρίνονται κυρίως σε ελαφρές δονήσεις και δυναμικές αλλαγές αφής (**10–50 Hz**). Η πυκνή παρουσία τους στις παλάμες και τα δάχτυλα τους καθιστά ιδιαίτερα σημαντικούς για την ανίχνευση ταχύτατων μεταβολών στην επιφανειακή επαφή, όπως η ολίσθηση ενός αντικειμένου πάνω στο δέρμα. Κατά συνέπεια, διαδραματίζουν κρίσιμο ρόλο στη ρύθμιση του απτικού ελέγχου κατά τη διάρκεια της χειριστικής δραστηριότητας [33].
- **Ruffini** (διάταση), Οι **υποδοχείς (Ruffini endings, SA II)**, είναι επίσης βραδέως προσαρμοζόμενοι, οι οποίοι ενεργοποιούνται κυρίως από την **διάταση του δέρματος** και τη

**συνεχή παραμόρφωση των ιστών.** Σε αντίθεση με τους Meissner και Pacinian, που ανταποκρίνονται σε δονήσεις, οι Ruffini καταγράφουν πιο «στατικές» πληροφορίες, σχετικές με την τάση και το τέντωμα του δέρματος. Αυτό τους καθιστά ιδιαίτερα σημαντικούς για την **αντίληψη της θέσης και της κίνησης των δακτύλων**, καθώς παρέχουν σήματα που βοηθούν τον εγκέφαλο να εκτιμήσει τον προσανατολισμό του χεριού και τη δύναμη που ασκείται κατά τη σύλληψη ενός αντικειμένου [33].

- **Pacinian** (υψηλές συχνότητες). Οι **Pacinian corpuscles**, αποτελούν τους πιο ευαίσθητους δονητικούς υποδοχείς. Αποκρίνονται σε συχνότητες από **40 έως 500 Hz**, με μέγιστη ευαισθησία γύρω στα **250 Hz**. είναι ιδιαίτερα σημαντικοί για τις απτικές εφαρμογές [34], [35]. Η κατανομή τους είναι ανομοιογενής στο ανθρώπινο σώμα, με υψηλή πυκνότητα σε περιοχές όπως οι παλάμες και τα δάχτυλα. Η λειτουργία τους αφορά την ανίχνευση ταχύτατων δονητικών ερεθισμάτων και μικροδονήσεων που μεταδίδονται μέσα από αντικείμενα σε επαφή με το δέρμα, ενισχύοντας σημαντικά την ικανότητα διάκρισης λεπτών διαφορών στην απτική εμπειρία [33].

Η **αντίληψη έντασης** ενός απτικού ερεθίσματος δεν αυξάνεται γραμμικά με τη φυσική του τιμή. Σύμφωνα με τον **Νόμο του Weber–Fechner**, η ψυχολογική ένταση ενός ερεθίσματος σχετίζεται λογαριθμικά με το πραγματικό μέγεθος:

$$S = k \cdot \log\left(\frac{I}{I_0}\right) \quad (3.1)$$

Όπου:

- S: η αντιλαμβανόμενη ένταση,
- I: ένταση ερεθίσματος,
- $I_0$ : το κατώφλι αντίληψης,
- k: μία σταθερά εξαρτώμενη από το αισθητήριο σύστημα. [36]

Αυτό το μοντέλο είναι καθοριστικό για τον σχεδιασμό απτικών συστημάτων ώστε να αποδίδουν αντιληπτικά διακριτές διαβαθμίσεις ερεθισμάτων.

Το **equal-loudness contour** αναφέρεται στις καμπύλες ίσης ακουστότητας που περιγράφουν πώς η ανθρώπινη ακοή δεν έχει την ίδια ευαισθησία σε όλες τις συχνότητες, αλλά διαφοροποιείται ανάλογα με το φάσμα του ήχου και το επίπεδο έντασης. Με άλλα λόγια, δύο ήχοι με διαφορετικές συχνότητες αλλά διαφορετικές εντάσεις μπορούν να γίνουν αντιληπτοί ως το ίδιο «δυνατοί» από τον ακροατή. Οι καμπύλες αυτές καθορίστηκαν αρχικά από τους Fletcher και Munson (1933) και στη συνέχεια βελτιώθηκαν από την ISO 226:2003, η οποία αποτελεί το διεθνές πρότυπο για τις καμπύλες ίσης

ακουστότητας [37], [38]. Μαθηματικά, η αντιληπτή ακουστότητα εκφράζεται σε phons και σχετίζεται με το φυσικό επίπεδο πίεσης του ήχου (SPL, σε dB) μέσω ενός μη γραμμικού μοντέλου που εξαρτάται από τη συχνότητα:

Το **απτικό κατώφλι αντίχενωσης** καθορίζεται από τον νόμο του Weber:

$$\frac{\Delta I}{I} = k \quad (3.2)$$

Όπου:

- $\Delta I$  είναι η ελάχιστη μεταβολή που μπορεί να γίνει αντιληπτή σε σχέση με την αρχική ένταση  $I$ , και
- $k$  είναι μια σταθερά εξαρτώμενη από το είδος του ερεθίσματος (π.χ. δύναμη, δόνηση) [39].

Για δονήσεις, η τιμή του  $k$  κυμαίνεται συνήθως μεταξύ 0.08–0.2, αναλόγως της συχνότητας και της θέσης [39].

### 3.4.2 Τεχνολογικές Βάσεις και Απόδοση Απτικής Πληροφορίας

Τα συστήματα απτικής ανατροφοδότησης χρησιμοποιούν **ενεργοποιητές** (actuators), που αναλύθηκαν στο κεφάλαιο 3.2, που μετατρέπουν ηλεκτρικά σήματα σε μηχανική κίνηση, είτε γραμμική είτε περιστροφική. Οι ενεργοποιητές αυτοί τοποθετούνται σε φορητές ή σταθερές διεπαφές (π.χ. γάντια, ζώνες, επιφάνειες), επιτρέποντας την αναπαραγωγή αισθήσεων όπως δονήσεις, πίεση ή εικονική αφή [40], [41].

Στο τεχνολογικό επίπεδο:

1. η **δυναμική μοντελοποίηση** των απτικών συστημάτων γίνεται μέσω συστημάτων δεύτερης τάξης:

$$F(t) = m \cdot x''(t) + b \cdot x'(t) + k \cdot x(t) \quad (3.3)$$

Όπου:

- $F(t)$ : η εφαρμοζόμενη δύναμη,
- $m$ : η μάζα του συστήματος (ή της διεπαφής),
- $b$ : ο συντελεστής απόσβεσης,
- $k$ : ο συντελεστής ελαστικότητας,

- $x(t)$ : η μετατόπιση ως συνάρτηση του χρόνου.

[40], [42]

Το μοντέλο αυτό επιτρέπει την προσομοίωση ιδιοτήτων όπως σκληρότητα ή ελαστικότητα σε εικονικά περιβάλλοντα [42]. Ο Colgate εισήγαγε την έννοια της **Z-width**, η οποία ορίζει το εύρος δυνατών απτικών συμπεριφορών που μπορεί να αποδώσει ένα σύστημα χωρίς αστάθεια [43]. Το μέτρο αυτό καθορίζει πόσο ρεαλιστικά μπορεί ένα σύστημα να αναπαραγάγει την αίσθηση δύναμης, και εξαρτάται από την απόκριση του ενεργοποιητή, το bandwidth και την υστέρηση του συστήματος.

Ο τύπος της **Z-width** είναι: [44]:

$$Z_{max} = \frac{2b}{T_s} \quad (3.4)$$

όπου  $T_s$  είναι το χρονικό βήμα δειγματοληψίας. Με άλλα λόγια, όσο πιο γρήγορη η δειγματοληψία, τόσο πιο «σκληρά» περιβάλλοντα μπορούν να αποδοθούν με ασφάλεια [44], [45].

Αυτός ο τύπος αναδεικνύει τη σημασία της **χρονικής ανάλυσης και εύρους ζώνης (bandwidth)** του συστήματος, καθώς κάθε απτική διεπαφή έχει ένα φυσικό όριο στο τι μπορεί να αναπαράγει με ασφάλεια.

**2. Ο απτικός προσομοιωτής (haptic rendering engine)**, πρέπει να υπολογίζει με πολύ υψηλό ρυθμό (συνήθως  $> 1\text{kHz}$ ) την δύναμη που θα ανατροφοδοτηθεί στον χρήστη [44].

Η **σταθερότητα** ενός τέτοιου συστήματος εξαρτάται από το **χρόνο δειγματοληψίας**, τη **καθυστέρηση απόκρισης**, και την αριθμητική **ολοκλήρωση** του δυναμικού μοντέλου. Σύμφωνα με την **Θεωρία της Παθητικότητας** (Passivity Theory), ένα απτικό σύστημα είναι παθητικό εάν δεν μπορεί να αποδώσει περισσότερη ενέργεια από όση εισάγει ο χρήστης. Αυτό εκφράζεται μαθηματικά ως:

$$\int_0^T F(t) \cdot \dot{x}(t) dt \geq 0 \quad (3.5)$$

Για κάθε χρονική διάρκεια  $T$ . Εάν αυτή η συνθήκη παραβιαστεί λόγω αριθμητικών σφαλμάτων ή λανθασμένων παραμέτρων, το σύστημα μπορεί να καταστεί **ασταθές**, δημιουργώντας ανεξέλεγκτες δυνάμεις [46], [47].

**3. Στα δυναμικά μοντέλα**, ο **ρυθμός εξάπλωσης της επιφάνειας επαφής (CASR)** χρησιμοποιείται για την αναπαράσταση της μαλακότητας. Αυτός ο ρυθμός είναι ο παράγωγος της επιφάνειας επαφής ως προς τον χρόνο, και σχετίζεται γραμμικά με την αντιλαμβανόμενη απαλότητα:

$$CASR(t) = \frac{dA(t)}{dt} \quad (3.6)$$

όπου  $A(t)$  είναι η επιφάνεια επαφής σε χρόνο  $t$ . [48]

Ο ρόλος της **επιφάνειας επαφής** είναι κρίσιμος στην αντίληψη μαλακότητας και υφής. Οι Bicchi et al. απέδειξαν ότι η μεταβολή της «ταχύτητας διάχυσης της επιφάνειας επαφής» (Contact Area Spread Rate, CASR) σχετίζεται άμεσα με την αντίληψη απτικής απαλότητας, κάτι που επηρεάζει και την αναπαράσταση μουσικών χαρακτηριστικών όπως το «timbre» [48].

### 3.5 Μετατροπή Ηχητικών Σημάτων σε Απτικά Ερεθίσματα

#### Μοντέλα Δόνησης

Οι δονήσεις της μουσικής μπορούν να περιγραφούν με μαθηματικά ως ημιτονοειδείς συναρτήσεις της μορφής:

$$x(t) = A \cdot \sin(2\pi ft + \varphi) \quad (3.7)$$

όπου:

- $A$  είναι το πλάτος της δόνησης (σχετίζεται με την ένταση),
- $f$  η συχνότητα (Hz),
- $\varphi$  η φάση,
- $t$  ο χρόνος.

Η αίσθηση της δόνησης εξαρτάται σημαντικά από το φάσμα των συχνοτήτων που απαρτίζουν τη μουσική. Το ανθρώπινο απτικό σύστημα έχει διαφορετική ευαισθησία σε συχνότητες, με κορυφές στα 200-300 Hz για το Pacinian σύστημα [49].

Με χρήση τεχνικών **ανάλυσης φασματικού περιεχομένου** (π.χ. FFT, wavelet transforms) και **ανίχνευσης ρυθμικών χαρακτηριστικών**, η μουσική μπορεί να μετατραπεί σε δομές απτικών σημάτων [50], [51].

Τα απτικά αυτά μοτίβα χαρτογραφούνται σε **φορητές συσκευές** που παρέχουν δονήσεις σε διαφορετικά μέρη του σώματος, με στόχο την αναπαράσταση ρυθμού, δυναμικής και αρμονικής πληροφορίας. Ένα παράδειγμα είναι η συσκευή **Ontenna**, η οποία μεταφράζει ένταση ήχου σε ταχύτερες παλμικές δονήσεις [34].

## Φασματική Ανάλυση

Για πιο πολύπλοκα μουσικά σήματα, εφαρμόζουμε την **Μετασχηματισμό Fourier**. Η φασματική ανάλυση αποκαλύπτει τις επιμέρους συχνότητες που συνθέτουν το μουσικό κομμάτι, και μπορούμε να συσχετίσουμε αυτές τις συχνότητες με την ενεργοποίηση των διαφορετικών απτικών υποδοχέων [52].

### 3.5.1 Fourier

Ένα ηχητικό σήμα καταγράφεται αρχικά στο **χρονικό πεδίο**:  $x(t)$  ή διακριτότερα  $x[n]$ .

Ο **Μετασχηματισμός Fourier** αποτελεί θεμελιώδες μαθηματικό εργαλείο για την ανάλυση σημάτων, καθώς επιτρέπει τη μετατροπή τους από το πεδίο του χρόνου στο πεδίο της συχνότητας. Αυτή η διαδικασία επιτρέπει την ανάλυση των στοιχείων του ήχου (όπως η ένταση, η συχνότητα και ο ρυθμός) και την αναπαράστασή τους με γραφικά στοιχεία. Στην περίπτωση ενός σήματος  $x(t)$ , ο Μετασχηματισμός **Fourier** εκφράζεται ως:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt \quad (3.8)$$

Όπου:

- $X(f)$  είναι το φάσμα του σήματος  $x(t)$ , δηλαδή η αναπαράσταση του σήματος σε συνάρτηση με τη συχνότητα.
- $t$  είναι ο χρόνος.
- $f$  είναι η συχνότητα. [52]

Διακριτός Μετασχηματισμός Fourier (**DFT**):

$$x[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \dots, N-1 \quad (3.9)$$

- $\mathbf{x[n]}$ : η είσοδος του σήματος στον χρόνο ή δείγμα  $n$ , για  $n = 0, 1, \dots, N-1$ . Πρόκειται για τα διακριτά δεδομένα που θέλουμε να αναλύσουμε.
- $\mathbf{x[k]}$ : η έξοδος του DFT στο δείγμα συχνότητας  $k$ . Αντιπροσωπεύει την ποσότητα (πλάτος και φάση) της συχνότητας  $k$  που περιέχεται στο σήμα  $x[n]$ .
- $\mathbf{N}$ : ο συνολικός αριθμός δειγμάτων του σήματος.
- $\mathbf{j}$ : η μονάδα φανταστικού αριθμού ( $j^2 = -1$ )
- $\frac{2\pi}{N}kn$ : το γωνιακό πολλαπλασιαστικό που καθορίζει την αντιστοίχιση μεταξύ δείγματος χρόνου  $n$  και δείγματος συχνότητας  $k$ . [52]

Ο άμεσος υπολογισμός του **DFT** απαιτεί αριθμό πράξεων που αυξάνεται με το τετράγωνο του πλήθους των δειγμάτων, κάτι που τον καθιστά αναποτελεσματικό για μεγάλες τιμές του  $N$ . Αντιθέτως, η **FFT**, με πιο διαδεδομένη εκδοχή αυτή των Cooley και Tukey, μειώνει σημαντικά το υπολογιστικό κόστος, καθώς ο αριθμός των απαιτούμενων πράξεων αυξάνεται πολύ πιο αργά, περίπου ανάλογα με το γινόμενο του πλήθους των δειγμάτων και του λογαρίθμου του [52], [53].

Ο **FFT** χρησιμοποιείται ευρέως σε επιστημονικές και τεχνολογικές εφαρμογές, όπως στην επεξεργασία ήχου και εικόνας, στις ψηφιακές επικοινωνίες, στη φασματική ανάλυση, καθώς και σε πεδία όπως οι υπολογιστικές επιστήμες και η μηχανική μάθηση. Ενδεικτικά, στην περίπτωση ηχητικών σημάτων, ο **FFT** επιτρέπει την ανάλυση των συχνοτήτων που τα συνθέτουν, διευκολύνοντας την εφαρμογή τεχνικών συμπίεσης ή αποθορυβοποίησης. Παράλληλα, αξιοποιείται σε κρίσιμες εφαρμογές, όπως η επεξεργασία σεισμικών δεδομένων, η αναγνώριση ομιλίας και η ιατρική απεικόνιση, με χαρακτηριστικό παράδειγμα τη μαγνητική τομογραφία (MRI) [52].

### 3.6 Η σημασία της αφής στην εμπειρία του χρήστη (UX)

#### Απόδοση Ηχητικών Χαρακτηριστικών μέσω Αφής (Audio-Tactile Rendering)

Η «απτική αναπαράσταση του ήχου» (ATR) είναι η διαδικασία κατά την οποία ο ήχος μετατρέπεται σε δονήσεις που μπορούν να γίνουν αντιληπτές μέσω της αφής. Ο στόχος αυτής της τεχνολογίας είναι να μεταφέρει μουσική πληροφορία είτε για να ενισχύσει την αισθητηριακή εμπειρία, είτε για να προσφέρει εναλλακτικούς τρόπους πρόσβασης σε άτομα με προβλήματα ακοής [30].

Η τεχνική αυτή στηρίζεται στην ανάλυση βασικών μουσικών στοιχείων – όπως ο ρυθμός, το ύψος του ήχου, η μελωδία, η χροιά και η ένταση – τα οποία μετασχηματίζονται σε φυσικά χαρακτηριστικά δονήσεων, όπως η συχνότητα, η ένταση, η διάρκεια και η θέση τους στον χώρο [30].

#### 3.6.1 Ρυθμική Δόνηση

Ο ρυθμός είναι ένα από τα μουσικά στοιχεία που μπορούν να μεταφερθούν πιο άμεσα στην αφή, καθώς η περιοδική φύση του επιτρέπει να αποδοθεί αποτελεσματικά με επαναλαμβανόμενους δονητικούς παλμούς. Μέσα από αυτούς, ο χρήστης μπορεί να αισθανθεί τους μουσικούς χτύπους ή να αναγνωρίσει ρυθμικά μοτίβα [30].

#### Ενδεικτικές Μέθοδοι:

- Χρήση χαμηλοπερατών φίλτρων (συχνότητες κάτω από 150 Hz) για την εξαγωγή των χαμηλών ήχων, όπως τύμπανα ή μπάσο, που φέρουν το βασικό ρυθμικό σχήμα [30].

- Δημιουργία δονήσεων μέσω κυματομορφών, όπως ημιτονοειδών, τετραγωνικών ή σύντομων παλμών, για να αποδοθούν τα «χτυπήματα» του ρυθμού [30].
- Σύνθεση δονητικών σημάτων από την αρχή, με βάση καθορισμένα χρονικά διαστήματα και ένταση, ώστε να επιτευχθεί ακρίβεια στον ρυθμό [30].

### Μαθηματική Περιγραφή Ρυθμικού Παλμού:

$$v(t) = A \cdot \sin(2\pi ft) \cdot e^{-at} \quad (3.10)$$

Όπου:

- **A**: αντιστοιχεί στην ένταση της δόνησης,
- **f**: είναι η συχνότητα του παλμού (π.χ. 40 Hz για χαμηλές συχνότητες),
- **a**: καθορίζει πόσο γρήγορα μειώνεται η ένταση (απόσβεση),
- **t**: είναι ο χρόνος.

**Επιπλέον:** Ο συγχρονισμός των δονήσεων με οπτικά ερεθίσματα, όπως φώτα ή animations, μπορεί να ενισχύσει ακόμη περισσότερο την αντίληψη του ρυθμού από τον χρήστη [30].

### 3.6.2 Απόδοση Ύψους Ήχου (Pitch)

Το ύψος του ήχου, το οποίο αντιστοιχεί στη θεμελιώδη συχνότητα μιας μουσικής νότας, αποτελεί ένα ακουστικό χαρακτηριστικό με περιορισμένη δυνατότητα απτικής απεικόνισης, εξαιτίας των φυσιολογικών ορίων της ανθρώπινης απτικής αντίληψης. Το φάσμα των συχνοτήτων που μπορούν να γίνουν αντιληπτές μέσω της αφής κυμαίνεται περίπου από 20 έως 500 Hz, με τη μέγιστη απτική ευαισθησία να παρατηρείται κοντά στα 250 Hz. Ως εκ τούτου, η ακριβής απόδοση του πλήρους φάσματος των μουσικών νοτών δεν είναι εφικτή [30].

#### Ενδεικτικές τεχνικές απτικής απόδοσης του ύψους:

- **Χωρική χαρτογράφηση συχνοτήτων (Frequency-to-Location Mapping):** Η συχνότητα μεταφράζεται σε διαφορετική απτική θέση στο σώμα. Χαμηλές συχνότητες μπορούν να αποδίδονται μέσω δονητών τοποθετημένων σε περιοχές όπως οι γοφοί ή η κοιλιακή χώρα, ενώ υψηλότερες συχνότητες αντιστοιχίζονται σε σημεία όπως οι ώμοι ή τα άνω άκρα. Αυτή η προσέγγιση αξιοποιεί τη χωρική διαφοροποίηση ως φορέα πληροφορίας [30].

- **Απτικές μεταφορές (Tactile Metaphors):** Η ποιοτική διαφοροποίηση των δονητικών σημάτων π.χ. απότομες, σύντομης διάρκειας δονήσεις για υψηλές νότες και αργές, βαθύτερες για χαμηλές χρησιμοποιείται για τη δημιουργία υποκειμενικής αντιστοιχίας μεταξύ ακουστικών και απτικών χαρακτηριστικών [30].

#### **Ελάχιστη διακριτή διαφορά συχνότητας (JND):**

Η κατώτατη διαφορά συχνότητων που μπορεί να γίνει αντιληπτή μέσω απτικής διεπαφής εκτιμάται ότι κυμαίνεται μεταξύ 8 και 10 Hz. Η τιμή αυτή εξαρτάται από παράγοντες όπως το σημείο εφαρμογής της δόνησης στο σώμα, η ένταση του σήματος και τα τεχνικά χαρακτηριστικά του δονητικού ενεργοποιητή [30].

### **3.6.3 Απτική Αναπαράσταση Μελωδίας**

Η μελωδία, η οποία ορίζεται ως η χρονικά οργανωμένη ακολουθία μεταβολών στο μουσικό ύψος (pitch), παρουσιάζει σημαντικές δυσκολίες κατά την απτική απόδοσή της, σε σύγκριση με άλλα μουσικά χαρακτηριστικά. Η δυσκολία αυτή απορρέει από τους εξής παράγοντες:

- Η απτική αισθητηριακή οδός έχει περιορισμένη ικανότητα διάκρισης συχνότητων, γεγονός που δεν επιτρέπει την ακριβή αναπαράσταση των μουσικών διαστημάτων σε όλο το φάσμα ακουστικών συχνότητων [30].
- Σε αντίθεση με το ακουστικό σύστημα, η απτική αντίληψη δεν προσφέρει καθαρό διαχωρισμό μεταξύ διαδοχικών μουσικών νοτών, καθιστώντας ασαφή την αίσθηση τονικών βαθμίδων και μελωδικών μοτίβων [30].

#### **Προτεινόμενες τεχνικές για την απτική απόδοση της μελωδίας:**

- **Εξαγωγή κυρίαρχων φθόγγων και μετατροπή σε MIDI σήματα:** Μέσω ανάλυσης του αρχικού ακουστικού σήματος (π.χ. μέσω FFT ή pitch tracking), εντοπίζονται οι επικρατούσες νότες, οι οποίες μετατρέπονται σε MIDI δεδομένα. Τα δεδομένα αυτά χρησιμοποιούνται για τον έλεγχο απτικών ενεργοποιητών με σκοπό την αναπαράσταση της μελωδικής πληροφορίας [30].
- **Χρήση συστοιχίας δονητικών ενεργοποιητών (tactile arrays):** Μέσω της σειριακής ενεργοποίησης πολλαπλών δονητών, το σύστημα μπορεί να μεταδώσει τη μελωδία απτικά, δημιουργώντας την αίσθηση της μελωδικής συνέχειας μέσα από χωρική και χρονική διαφοροποίηση [30].

- **Προσομοίωση μελωδικής κίνησης μέσω χρονικών μεταβολών:** Η δυναμική μεταβολή της έντασης, διάρκειας και συχνότητας των δονήσεων επιτρέπει τη δημιουργία μιας απτικής αναπαράστασης της ανόδου ή καθόδου της μελωδίας, ακόμη και αν δεν αναπαρίσταται πλήρως κάθε διακριτή νότα [30].

### Μαθηματικό μοντέλο:

Αν θεωρηθεί μια ακολουθία από νότες με συχνότητες  $f_1, f_2, \dots, f_n$ , και αντίστοιχους χρόνους ενεργοποίησης  $T_i$ , η απτική αναπαράσταση της μελωδίας μπορεί να μοντελοποιηθεί ως εξής:

$$M(t) = \sum_{i=1}^n A_i \cdot \sin(2\pi f_i \cdot (t - T_{start,i})) \cdot w_i(t) \quad (3.11)$$

όπου:

- $A_i$ : το πλάτος (ένταση) της δόνησης που αντιστοιχεί στη νότα  $i$ ,
- $f_i$ : η θεμελιώδης συχνότητα της κάθε νότας,
- $T_{start,i}$ : η χρονική στιγμή έναρξης της νότας,
- $w_i(t)$ : συνάρτηση παραθύρου (π.χ. Gaussian, Hanning) που ορίζει τη διάρκεια και τη μορφή της δόνησης στο χρόνο [30].

Αυτό το υπόδειγμα παρέχει ένα βασικό θεωρητικό πλαίσιο για την απτική σύνθεση μελωδίας και μπορεί να προσαρμοστεί με βάση τις παραμέτρους ευαισθησίας του ανθρώπινου απτικού συστήματος [30].

### 3.6.4 Απόδοση Χροιάς (Timbre)

Η χροιά, γνωστή και ως "χρώμα" του ήχου, καθορίζει την ποιότητα και την ταυτότητα του ήχου, προσδιορίζοντας το χαρακτήρα του πέρα από το ύψος και την ένταση. Προκύπτει από τη φασματική ανάλυση του σήματος, περιλαμβάνοντας τον αριθμό και την ένταση των αρμονικών συνιστωσών του ήχου. Αν και το ανθρώπινο σύστημα αφής δεν διαθέτει την ίδια φασματική διακριτική ικανότητα με την ακοή, είναι σε θέση να ανιχνεύσει βασικά χαρακτηριστικά που συνδέονται με τη χροιά, όπως:

- Τον τύπο της κυματομορφής, για παράδειγμα διαφορές μεταξύ πριονωτών και ημιτονοειδών κυματομορφών.

Ποικιλία στην ένταση και στην πυκνότητα των δονήσεων [30].

### Τεχνικές Απόδοσης Χροιάς:

- **Παραμετροποίηση ADSR (Attack, Decay, Sustain, Release):** Η εφαρμογή διαφορετικών καμπυλών έντασης στον χρόνο, οι οποίες αναπαριστούν τα στάδια του ήχου, επιτρέπει την αναπαραγωγή της χροιάς μέσω απτικών ερεθισμάτων.
- **Συνδυασμός καθαρών τόνων και λευκού θορύβου:** Αυτή η τεχνική χρησιμοποιείται για να αναπαραστήσει ηχητικά χαρακτηριστικά όπως η "τραχιά" ή "καθαρή" υφή ενός ήχου.

Αυτές οι μέθοδοι καθιστούν δυνατή την αναπαραγωγή και αντίληψη της χροιάς του ήχου μέσω του αισθητηρίου της αφής, επιτρέποντας τη δημιουργία μιας ολοκληρωμένης απτικής αναπαράστασης του ηχητικού φάσματος [30].

### 3.6.5 Απόδοση Έντασης (Loudness)

Η ένταση αποτελεί ένα από τα πιο απλά χαρακτηριστικά που μπορούν να αποδοθούν μέσω της αφής. Η αύξηση της ακουστικής έντασης οδηγεί σε μεγαλύτερη δόνηση, η οποία αντιστοιχεί σε μεγαλύτερο πλάτος του σήματος.

#### Εξίσωση για την ένταση:

$$L = k \cdot A \quad (3.12)$$

Όπου:

- **L:** Η αντιληπτή ένταση,
- **A:** Το πλάτος του σήματος,
- **k:** Η σταθερά αναλογίας. [30]

#### Ψυχοφυσικοί παράγοντες:

Η αντίληψη της έντασης εξαρτάται από τη συχνότητα του σήματος. Επίσης, υπάρχει φαινόμενο άθροισης (summation) και καταστολής (suppression) όταν πολλά σήματα παράγονται ταυτόχρονα, κάτι που οδηγεί σε συγχρονισμό των σημάτων σε κοινά κανάλια [30].

#### Σύνοψη

Η αποτελεσματική απόδοση μουσικής μέσω της αφής απαιτεί όχι μόνο προηγμένη τεχνολογία, αλλά και μια βαθιά κατανόηση της ανθρώπινης απτικής φυσιολογίας. Κάθε μουσικό χαρακτηριστικό

απαιτεί την εφαρμογή εξειδικευμένων τεχνικών και συμβιβασμών, με σκοπό την παροχή της πιο πλούσιας δυνατής εμπειρίας για τον χρήστη.

### 3.7 Συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάστηκαν οι βασικές αρχές της απτικής ανατροφοδότησης και η σημασία της για την ανάπτυξη μιας εφαρμογής που μετατρέπει τη μουσική σε δονήσεις. Αρχικά, έγινε κατανοητό ότι η απτική αλληλεπίδραση αποτελεί κρίσιμο μέσο επικοινωνίας ανάμεσα στον χρήστη και το σύστημα, αξιοποιώντας τις δυνατότητες του ανθρώπινου αισθητηριακού μηχανισμού για να μεταφέρει πληροφορίες μέσω δονήσεων. Η ιστορική αναδρομή έδειξε την εξέλιξη της τεχνολογίας από απλά μηχανικά συστήματα έως τα σύγχρονα κινητά τηλέφωνα και τις φορητές συσκευές. Παρουσιάστηκαν οι κύριοι τύποι ενεργοποιητών (VCA, LRA, ERM, πιεζοηλεκτρικοί, DMA), με τα πλεονεκτήματα και τους περιορισμούς τους. Η ανάλυση αυτή ανέδειξε πως η επιλογή του κατάλληλου μηχανισμού δόνησης είναι καθοριστική για την ποιότητα της εμπειρίας του χρήστη, καθώς επηρεάζει τόσο την ακρίβεια όσο και την κατανάλωση ενέργειας. Επιπλέον, αναδείχθηκαν οι διαφορές στην ένταση, τη διάρκεια και τα μοτίβα των δονήσεων, που αποτελούν θεμέλια για τη μετάδοση μουσικών χαρακτηριστικών μέσω αφής. Η μελέτη της νευροφυσιολογικής βάσης της αφής επιβεβαίωσε ότι οι διαφορετικοί υποδοχείς (Merkel, Meissner, Ruffini, Pacinian) έχουν εξειδικευμένους ρόλους στην αντίληψη πίεσης, υφής και δόνησης, γεγονός που ενισχύει τη δυνατότητα δημιουργίας σύνθετων απτικών εμπειριών. Παράλληλα, παρουσιάστηκαν μαθηματικά και ψυχοφυσικά μοντέλα (όπως ο Νόμος Weber–Fechner και οι καμπύλες ίσης ακουστότητας), τα οποία καθοδηγούν τον σχεδιασμό απτικών συστημάτων ώστε οι δονήσεις να γίνονται αντιληπτές με φυσικό και διακριτό τρόπο. Η τεχνολογική βάση των συστημάτων αυτών, μέσα από δυναμικά μοντέλα και φασματική ανάλυση (FFT), επιτρέπει την αξιόπιστη μετατροπή των ακουστικών σημάτων σε απτικά μοτίβα. Τέλος, τονίστηκε η σημασία της απτικής αναπαράστασης του ήχου (Audio-Tactile Rendering) τόσο για την ενίσχυση της εμπειρίας χρήστη όσο και για τη δημιουργία εργαλείων που προσφέρουν σε άτομα με προβλήματα ακοής μια εναλλακτική οδό πρόσληψης της μουσικής. Το κεφάλαιο αυτό έθεσε τις επιστημονικές και τεχνολογικές βάσεις πάνω στις οποίες θα στηριχθεί η υλοποίηση της εφαρμογής που προτείνεται στη συγκεκριμένη διπλωματική εργασία.

## Κεφάλαιο 4ο:

### Κεφάλαιο 4ο: Haptic Music Player (HMP) με Visualizer

#### 4.1 Εισαγωγή — Ορισμός και Ιστορική Αναδρομή

Οι συσκευές απτικής αναπαραγωγής μουσικής, γνωστές ως Haptic Music Players (**HMPs**), έχουν σχεδιαστεί με σκοπό τη μετατροπή της ηχητικής πληροφορίας σε μηχανικές δονήσεις που γίνονται αισθητές μέσω του δέρματος, προσφέροντας μια πλούσια απτική εμπειρία. Η αρχιτεκτονική ενός HMP στηρίζεται σε μια αλληλουχία υποσυστημάτων που συνεργάζονται για την ολοκληρωμένη μετατροπή και απόδοση του ακουστικού σήματος σε απτικές εντολές και, στο σύγχρονο πλαίσιο, σε οπτικές αναπαραστάσεις μέσω visualizer [54],[55].

**Ιστορικά**, οι πρώτες προσπάθειες δημιουργίας συστημάτων μετατροπής ήχου σε δόνηση εμφανίστηκαν ήδη από τις δεκαετίες του 1960 και 1970, με απλούς μηχανικούς μηχανισμούς που τοποθετούνταν σε καθίσματα ή δάπεδα συναυλιακών χώρων [55]. Οι περιορισμοί εκείνης της εποχής σχετίζονταν κυρίως με την ακρίβεια μεταφοράς του ήχου και την περιορισμένη δυνατότητα ελέγχου της έντασης και της συχνότητας των δονήσεων.

Ωστόσο, η τεχνολογική πρόοδος των τελευταίων δεκαετιών ιδιαίτερα στους τομείς των **μικροηλεκτρομηχανικών συστημάτων (MEMS)** και της **υπολογιστικής ανάλυσης σήματος σε πραγματικό χρόνο**, επέτρεψε τη δημιουργία και ανάπτυξη φορητών και υψηλής πιστότητας HMP συσκευών. Αυτές οι συσκευές είναι ικανές να αποδώσουν με μεγάλη λεπτομέρεια ακόμη και σύνθετα μουσικά έργα, ενώ ο σύγχρονος συνδυασμός με οπτικοποιητές (**visualizers**) επιτρέπει ταυτόχρονα την οπτική αναπαράσταση των μουσικών χαρακτηριστικών [56].

Η **σύγχρονη** θεώρηση του HMP υπερβαίνει τον απλό στόχο της προσβασιμότητας και εξετάζει την απτική απόδοση μουσικής ως **νέα αισθητηριακή εμπειρία** που μπορεί να εμπλουτίσει τη συναισθηματική αντίδραση του ακροατή. Μελέτες δείχνουν ότι η απτική αντίληψη μουσικών δομών μπορεί να ενισχύσει την αίσθηση του ρυθμού, να δημιουργήσει μια πιο «σωματική» σχέση με το έργο και να λειτουργήσει συμπληρωματικά προς την ακουστική εμπειρία και σε συνδυασμό με οπτικές αναπαραστάσεις αυξάνει την αλληλεπίδραση και την κατανόηση της μουσικής [57].

Συνεπώς, η μελέτη και ο σχεδιασμός της αρχιτεκτονικής των HMP με visualizer αποτελεί πεδίο υψηλής επιστημονικής και καλλιτεχνικής αξίας, συνδέοντας τεχνολογία, αισθητική και πρόσβαση σε μουσική με καινοτόμο τρόπο.

#### 4.2 Συνολική Αρχιτεκτονική και Ροή Δεδομένων

Η διαδικασία ξεκινά με το **στάδιο εισόδου (input stage)**, όπου το ηχητικό σήμα λαμβάνεται είτε ως προηχογραφημένο αρχείο (π.χ. WAV, MP3, FLAC) είτε ως ζωντανό σήμα από μικρόφωνο ή άλλη πηγή. Στο αρχικό αυτό στάδιο πραγματοποιείται προεπεξεργασία για την απομάκρυνση θορύβου και τη ρύθμιση της στάθμης, εξασφαλίζοντας βέλτιστη ποιότητα σήματος για την επόμενη φάση ανάλυσης [39].

Το επόμενο κρίσιμο στάδιο είναι η **εξαγωγή χαρακτηριστικών** (feature extraction), κατά το οποίο εφαρμόζονται αλγόριθμοι όπως ο μετασχηματισμός Fourier (FFT), ο Μετασχηματισμός Wavelet ή νευρωνικά δίκτυα, με σκοπό την απομόνωση και ανάλυση βασικών μουσικών παραμέτρων όπως ο ρυθμός (tempo), η φασματική κατανομή, η δυναμική (loudness envelope) και οι μελωδικές δομές [58],[59]. Αυτά τα μουσικά χαρακτηριστικά αποτελούν τη βάση για τη δημιουργία των απτικών σημάτων που θα αποδοθούν στον χρήστη.

Η **βασική αρχιτεκτονική** ενός τέτοιου συστήματος περιλαμβάνει τέσσερα κύρια συστατικά, τα οποία συνεργάζονται για την ολοκληρωμένη απτική εμπειρία:

- **Λογισμικό επεξεργασίας σήματος ήχου:** Το σύστημα λαμβάνει το ηχητικό σήμα από πηγές όπως αρχεία ήχου ή μικρόφωνα. Μέσω ανάλυσης, εντοπίζονται βασικά μουσικά χαρακτηριστικά, όπως ο ρυθμός, το ύψος και η χροιά, τα οποία αποτελούν τη βάση για τη δημιουργία των απτικών σημάτων [59].
- **Μονάδα χαρτογράφησης (haptic rendering engine):** Μετατρέπει τα μουσικά χαρακτηριστικά σε παραμέτρους δόνησης. Η χαρτογράφηση μπορεί να είναι άμεση (συχνότητα → δόνηση) ή αφαιρετική (ένταση → μοτίβο/ρυθμός). Σε πιο προηγμένα συστήματα η διαδικασία προσαρμόζεται δυναμικά σε πραγματικό χρόνο [54], [55],
- **Μετατροπέας ψηφιακού σε αναλογικό σήμα (DAC):** Απαραίτητος για να μετατρέψει τα ψηφιακά δεδομένα σε αναλογικά σήματα, κατάλληλα για την ενεργοποίηση των δονητικών μηχανισμών [60].
- **Ενισχυτές:** Ρυθμίζουν την ισχύ του σήματος ώστε να οδηγούν αποτελεσματικά τους δονητικούς ενεργοποιητές (actuators) [60].
- **Δονητικοί ενεργοποιητές (actuators):** Είναι τα στοιχεία που παράγουν τις δονήσεις και μεταφέρουν την απτική πληροφορία στο δέρμα του χρήστη [60].
- **Οπτικοποιητής (Visualizer):** Χρησιμοποιεί τα ίδια δεδομένα ανάλυσης ήχου για να δημιουργήσει οπτικές αναπαραστάσεις, όπως φασματογραφήματα, κυματομορφές και ρυθμικά μοτίβα, που προβάλλονται σε πραγματικό χρόνο [61].

Η **συνολική ροή** της πληροφορίας περιλαμβάνει τα εξής βήματα: εισερχόμενο ηχητικό σήμα → προεπεξεργασία → εξαγωγή χαρακτηριστικών → χαρτογράφηση → μετατροπή σε απτικά σήματα → ψηφιακό-αναλογική μετατροπή → ενίσχυση σήματος → δόνηση μέσω των ενεργοποιητών (απτική έξοδος) & Visualizer (οπτική έξοδος) [59],[62].

Τέλος, πολλές συσκευές ενσωματώνουν διαδραστικά στοιχεία, όπως κουμπιά ή επιλογείς, που επιτρέπουν στον χρήστη να προσαρμόζει τις παραμέτρους της απτικής εμπειρίας σύμφωνα με τις προτιμήσεις του [59].

### 4.3 Ενσωμάτωση Απτικής και Οπτικής Ανάδρασης

Η ενότητα αυτή στοχεύει να παρουσιάσει την **ολοκληρωμένη πολυαισθητηριακή εμπειρία** του συστήματος Haptic Music Player (HMP) με Visualizer, εξετάζοντας πώς η αφή και η όραση συνεργάζονται αρμονικά σε πραγματικό χρόνο. Ο στόχος είναι να αποδοθεί με σαφήνεια ο τρόπος σύνδεσης των δύο αυτών μέσων στην αναπαράσταση της μουσικής.

### **Συγχρονισμός Αίσθησης και Εικόνας: Χρονική Ευαισθησία**

Η **χρονική αντιστοίχιση** ανάμεσα σε απτικά και οπτικά ερεθίσματα είναι κρίσιμη για την αίσθηση και κατανόηση της μουσικής. Έρευνες δείχνουν ότι άνθρωποι αντιλαμβάνονται ασυνέπειες άνω των ~100 ms στον συγχρονισμό ανάμεσα σε haptic και οπτική ή auditory ερέθισμα. Η διαδικασία αυτή πρέπει να εκτελείται με εξαιρετικά χαμηλή καθυστέρηση (latency), συνήθως κάτω από 10 ms, ώστε η οπτική και απτική εμπειρία να παραμένει συγχρονισμένη με το ακουστικό ερέθισμα [57]. Επομένως, το σύστημα πρέπει να στοχεύει χρονικές καθυστερήσεις πιο κάτω από αυτές τις τιμές ώστε να διατηρείται ο συναισθηματικός και αισθητηριακός συγχρονισμός.

### **Συναισθηματική Εμπλοκή και Αισθητική Εμπειρία μέσω Αφής**

Η μετάφραση ηχητικών εμπειριών σε απτικά μοτίβα δεν αποσκοπεί μόνο στην πληροφοριακή αναπαράσταση, αλλά και στην **αισθητική και συναισθηματική πρόσληψη**. Έρευνες δείχνουν ότι συνδυασμένα οπτικο-απτικά ερεθίσματα μπορούν να **ενεργοποιήσουν εγκεφαλικά κυκλώματα** που σχετίζονται με την απόλαυση της μουσικής ακόμα και σε απουσία ακουστικής εισόδου [63].

### **Πολυαισθητηριακή Σικήψη και Αισθητηριακή Προσαρμογή**

Η **οπτική κυριαρχία** (visual dominance) είναι ένα φαινόμενο όπου η όραση υπερτερεί άλλων αισθήσεων στον καθορισμό της αντίληψης ενός ερεθίσματος. Παρ' όλα αυτά, σε περιπτώσεις όπου ο απτικός ερεθισμός παρέχει χρονικά πιο αξιόπιστες πληροφορίες, όπως στον ρυθμό, η αφή μπορεί να υπερτερεί. Στο HMP αυτό σημαίνει πως, αν η δόνηση ανταποκρίνεται πιο άμεσα στον ρυθμό της μουσικής απ' ό,τι το visualizer, τότε οι χρήστες μπορεί να αισθανθούν τον ρυθμό πιο έντονα μέσω της αφής. Η οπτικοποίηση, αντιθέτως, μπορεί να ενισχύει την αντίληψη της μελωδίας ή των χρωμάτων της μουσικής [64].

## **4.4 Διασύνδεση Συστήματος (Software & Hardware σε υψηλό επίπεδο)**

Η υλοποίηση ενός Haptic Music Player (HMP) απαιτεί τον προσεκτικό σχεδιασμό και την ακριβή συνεργασία υλικού (hardware) και λογισμικού (software) για τη δημιουργία μιας συνεκτικής, χαμηλού λανθάνοντος χρόνου και υψηλής πιστότητας απτικής εμπειρίας [54]. Στο επίπεδο του υλικού, οι **ενεργοποιητές**, που αναλύθηκαν στο κεφάλαιο 3.2, αποτελούν τον πυρήνα κάθε συστήματος απτικής αναπαραγωγής μουσικής, καθώς είναι υπεύθυνοι για τη φυσική δημιουργία των δονήσεων που αντιστοιχούν στα χαρακτηριστικά της μουσικής, όπως ο ρυθμός, η μελωδία και η δυναμική. Η επιλογή του κατάλληλου ενεργοποιητή είναι κρίσιμη, καθώς η ποιότητα της απτικής αντίληψης εξαρτάται από παραμέτρους όπως η απόκριση συχνοτήτων, το πλάτος των δονήσεων, η ενεργειακή απόδοση και η εργονομία της συσκευής [61].

### Παράγοντες επιλογής ενεργοποιητή

Η σωστή αντιστοίχιση μεταξύ των χαρακτηριστικών του ενεργοποιητή και των αναγκών του χρήστη αποτελεί καθοριστικό παράγοντα για την αποτελεσματικότητα των HMP και την αποδοχή τους από τα άτομα με ακουστική απώλεια [44].

### Λογικοί Δομικοί Πυρήνες του Συστήματος

Σε υψηλό επίπεδο, το σύστημα συνίσταται από **τρία βασικά modules**:

1. **Διαχειριστής Αλληλεπίδρασης (Interaction Manager)** – Σαν "κόμβος" του συστήματος, συλλέγει τα μουσικά δεδομένα, αποστέλλει συμβάντα στον σωστό προορισμό και διασφαλίζει τον συγχρονισμό μεταξύ των μονοπατιών (οπτικό και απτικό), σύμφωνα με τη φιλοσοφία των πολυτροπικών αρχιτεκτονικών [65].
2. **Visual-Logic Module** – Λαμβάνει δεδομένα ήχου (π.χ. συχνότητες ή ένταση) μέσω του Interaction Manager και παράγει real-time οπτικοποίηση, αναλόγως του συστήματος (2D/3D γραφικά, animations κ.ά.) [65].
3. **Haptic-Logic Module** – Παρέχει απτική ανατροφοδότηση βασισμένη στην ίδια είσοδο, μέσω κατάλληλων χαρτογραφήσεων (π.χ. ένταση σε δύναμη δόνησης), εξασφαλίζοντας σύνδεση με τον visualizer χωρίς επανάληψη [65].

Η επικοινωνία μεταξύ των modules βασίζεται σε χαλαρά συνδεδεμένες δομές, ώστε να διευκολύνεται η συντήρηση και επεκτασιμότητα (low coupling, high modularity) [65].

### Ροή Δεδομένων και Συγχρονισμός

Κατά την αναπαραγωγή μουσικής:

1. Το σύστημα λαμβάνει **audio input** (stream ή αρχείο).
2. Ο Interaction Manager αποστέλλει → Visual-Logic & Haptic-Logic.
3. Το **Visual-Logic** μετασχηματίζει δεδομένα (π.χ. FFT ή amplitude analysis) σε γραφικά μοτίβα, ενώ το **Haptic-Logic** τα μεταφράζει σε δονήσεις.
4. Τα δύο outputs εκπέμπονται παράλληλα στον χρήστη — η **πολυτροπική ροή δεδομένων** υποστηρίζει συναισθηματική συνοχή και αισθητηριακή συνέπεια. [65].

### Διανομή, Πλατφόρμες και Υποστήριξη

Το μοντέλο αυτό επιτρέπει:

- **Κεντρική υλοποίηση** (όλα τα modules σε μία συσκευή) για εύκολη ανάπτυξη και δοκιμή.
- **Κατανεμημένη υλοποίηση** (π.χ. κεντρικός server με οπτικοποίηση και κινητή συσκευή με haptic output), αξιοποιώντας δίκτυα και συμβατές διεπαφές, εφόσον διατηρείται το χαμηλό latency [66].

## Εργαλεία, APIs και Ενσωμάτωση

Για δημιουργία της πλατφόρμας:

- Σε Android: APIs όπως **MediaPlayer**, **AudioTrack** και **VibrationEffect** για αλληλεπίδραση ήχου-δόνησης.
- Σε Unity/VR: **Meta Haptics SDK** ή άλλα haptic SDKs όπως περιγράφονται σε οδηγούς πολυαισθητηριακού σχεδιασμού.
- Για οπτικοποίηση: χρήση OpenGL, shaders ή game engines για real-time rendering του Visual module.

Η συνεργασία μεταξύ λογισμικού και hardware διέπεται από πρότυπα για επεκτασιμότητα και επανεκμετάλλευση συστατικών [67].

## 4.5 Real-Time Συγχρονισμός και Προκλήσεις

Η επιτυχία του Haptic Music Player (HMP) με Visualizer εξαρτάται σε μεγάλο βαθμό από την ικανότητά του να συγχρονίζει με ακρίβεια απτική και οπτική έξοδο σε πραγματικό χρόνο. Η απαίτηση αυτή δημιουργεί ένα σύνολο τεχνικών προκλήσεων, που σχετίζονται με την καθυστέρηση, τη μεταβλητότητα της απόκρισης και τους περιορισμούς του υλικού και του λογισμικού.

### 1. Πηγές Καθυστέρησης στο Σύστημα

Η συνολική απόδοση επηρεάζεται από διαφορετικά τμήματα της γραμμής επεξεργασίας (pipeline):

- **Audio processing:** η ανάλυση φάσματος (π.χ. FFT), ο εντοπισμός ρυθμού και η εξαγωγή χαρακτηριστικών απαιτούν υπολογιστικούς πόρους που προκαλούν καθυστέρηση στην απόκριση (latency).
- **Haptic driver & APIs:** η δημιουργία μοτίβων δόνησης και η ενεργοποίηση του κινητήρα μέσω του λειτουργικού συστήματος συνεπάγονται αστάθεια στον χρόνο απόκρισης.
- **Απεικόνιση γραφικών:** η παραγωγή των γραφικών σε GPU εξαρτάται από τη συχνότητα εμφάνισης καρτέ (frame rate) και τις καθυστερήσεις στη γραμμή επεξεργασίας (pipeline).
- **Λειτουργικό σύστημα:** η διαχείριση διεργασιών και οι προσωρινές μνήμες (buffers) μπορούν να αυξήσουν την ακανόνιστη καθυστέρηση (jitter), ιδίως σε κινητές συσκευές. [66].

### 3. Τεχνικές Συγχρονισμού

Πέρα από την κλασική χρήση timestamps, νεότερες μέθοδοι αξιοποιούν **key-frame alignment**, δηλαδή συγχρονισμό με βάση σημεία αναφοράς στα δεδομένα (π.χ. κορυφές ρυθμού) ώστε να παρακάμπτεται η ανάγκη ακριβούς ρολογιού. Αυτό καθιστά το σύστημα πιο ανθεκτικό σε jitter και αστάθεια.

#### 4. Προοπτικές και Βέλτιστες Πρακτικές

Για μείωση των προβλημάτων προτείνονται:

- **Ελαχιστοποίηση αλγοριθμικής πολυπλοκότητας** στην ανάλυση ήχου.
- **Real-time schedulers** σε λειτουργικά συστήματα για προτεραιότητα στο haptic/visual pipeline.
- **Χρήση edge computing και 5G** για εφαρμογές με κατακευματισμένη αρχιτεκτονική.
- Εμπειρία από το χώρο του **VR/AR**, όπου αντίστοιχες τεχνικές συγχρονισμού έχουν ήδη εφαρμοστεί με επιτυχία. [66]

#### 4.6 Τεχνολογικές Υλοποιήσεις και Εφαρμογές

Τα συστήματα αυτά ενσωματώνουν σύγχρονες μεθόδους ψηφιακής επεξεργασίας ήχου, αλγορίθμους ανάλυσης φάσματος και τεχνικές χαρτογράφησης των ηχητικών χαρακτηριστικών σε δονήσεις και οπτικά ερεθίσματα.

##### **Reactable Music Visualizer**

Το Reactable Music Visualizer είναι ένα διαδραστικό σύστημα που μετατρέπει τη μουσική σε απτικά και οπτικά ερεθίσματα σε πραγματικό χρόνο. Το σύστημα επιτρέπει στους χρήστες να αντιλαμβάνονται την ένταση, τον ρυθμό και τη χροιά της μουσικής μέσω δονήσεων που παράγονται από actuators και παράλληλης οπτικής απεικόνισης με δυναμικά γραφικά, ενισχύοντας τη πολυαισθητηριακή εμπειρία ακρόασης [68].

##### **Audio-to-Haptics**

Η τεχνολογία *Audio-to-Haptics* της bHaptics προσφέρει μια καινοτόμο πολυαισθητηριακή εμπειρία, μετατρέποντας ήχους από παιχνίδια, μουσική και ταινίες σε απτική ανατροφοδότηση μέσω συσκευών όπως το TactSuit και το TactVisor. Αυτό επιτρέπει στους χρήστες να αισθάνονται τον ήχο στο σώμα τους, ενισχύοντας την αίσθηση παρουσίας και αλληλεπίδρασης. Η δυνατότητα προσαρμογής μέσω του *bHaptics Studio* δίνει τη δυνατότητα δημιουργίας εξατομικευμένων προφίλ, ρυθμίζοντας ένταση, συχνότητες και σημεία ανατροφοδότησης, προσφέροντας έτσι μια ολοκληρωμένη εμπειρία πολυαισθητηριακής αλληλεπίδρασης. [69]

##### **PWT (Playing With Tagging)**

Ο PWT είναι πειραματικός μουσικός αναπαραγωγέας που δεν περιορίζεται στην απλή αναπαραγωγή ήχου, αλλά εισάγει μια «tag-based» προσέγγιση, συνδέοντας τα μουσικά δεδομένα με σημασιολογικές

πληροφορίες μέσω τεχνικών μηχανικής μάθησης. Συγκεκριμένα, η ανάλυση του μουσικού σήματος επιτρέπει την αυτόματη απόδοση χαρακτηριστικών, όπως είδος, διάθεση ή δομή του έργου, σε μορφή ετικετών. Οι ετικέτες αυτές χρησιμοποιούνται για την παραγωγή δυναμικών οπτικοποιήσεων που προσεγγίζουν τον ακροατή οπτικά και ακουστικά, ενισχύοντας την εμπειρία αντίληψης της μουσικής [4].

### 4.7 Συμπεράσματα

Στο κεφάλαιο αυτό αναλύθηκε η αρχιτεκτονική και η λειτουργία του Haptic Music Player με Visualizer, ενός συστήματος που επιτρέπει τη μετατροπή του ήχου σε δονήσεις και οπτικά μοτίβα με στόχο την παροχή μιας ολοκληρωμένης πολυαισθητηριακής εμπειρίας. Περιγράφηκε η ροή των δεδομένων από το στάδιο εισαγωγής του μουσικού σήματος, την προεπεξεργασία και την εξαγωγή χαρακτηριστικών, μέχρι τη χαρτογράφησή τους σε δονήσεις μέσω ενεργοποιητών και την ταυτόχρονη οπτική αναπαράσταση με τη βοήθεια γραφικών. Τονίστηκε η σημασία της συνεργασίας υλικού και λογισμικού, καθώς και η ανάγκη για χαμηλό λανθάνοντα χρόνο, ώστε η απτική και η οπτική έξοδος να παραμένουν συγχρονισμένες με τη μουσική. Επισημάνθηκαν οι τεχνικές προκλήσεις που σχετίζονται με τον συγχρονισμό σε πραγματικό χρόνο, όπως οι καθυστερήσεις και το jitter, αλλά και οι πρακτικές λύσεις που μπορούν να εφαρμοστούν. Τέλος, παρουσιάστηκαν εφαρμογές και παραδείγματα, όπως το Reactable και το HapticMuse, που δείχνουν τις προοπτικές της τεχνολογίας αυτής. Συνολικά, το κεφάλαιο κατέδειξε ότι το HMP με Visualizer δεν αποτελεί μόνο εργαλείο προσβασιμότητας για άτομα με προβλήματα ακοής, αλλά και μια καινοτόμα προσέγγιση που ενισχύει τη λειτουργικότητα, την αισθητική εμπειρία και τη συναισθηματική σύνδεση με τη μουσική.

## Κεφάλαιο 5ο: Θεωρητικό Υπόβαθρο στην Αντίληψη των Ήχων και της Μουσικής από άτομα με προβλήματα ακοής

### 5.1 Εισαγωγή

Ο ήχος, σε όλες του τις μορφές, αποτελεί θεμελιώδες στοιχείο της καθημερινότητας, της επικοινωνίας και της καλλιτεχνικής έκφρασης. Από τους απλούς περιβαλλοντικούς ήχους που λειτουργούν ως σήματα προσανατολισμού και ασφάλειας, μέχρι τη μουσική που εκφράζει συναισθήματα και δημιουργεί συλλογικές εμπειρίες, η ακουστική πληροφορία είναι βαθιά συνδεδεμένη με την ανθρώπινη ζωή. Για τα άτομα με προβλήματα ακοής, η περιορισμένη ή ανύπαρκτη πρόσβαση σε αυτό το φάσμα ηχητικών εμπειριών δημιουργεί σημαντικά εμπόδια τόσο στην καθημερινή λειτουργικότητα όσο και στην κοινωνική και πολιτισμική συμμετοχή.

Για την αντιμετώπιση αυτών των προκλήσεων, η έρευνα και η τεχνολογία έχουν στραφεί στην ανάπτυξη εναλλακτικών καναλιών πρόσληψης της ακουστικής πληροφορίας, όπως η οπτικοποίηση, η απτική αναπαράσταση και οι πολυαισθητηριακές διεπαφές, που παρουσιάστηκαν στα προηγούμενα κεφάλαια. Μέσα από αυτές τις προσεγγίσεις, επιχειρείται να επαναπροσδιοριστεί η εμπειρία του ήχου και της μουσικής, ώστε να είναι πιο προσιτή και συμμετοχική για τα άτομα με προβλήματα ακοής.

Ενώ στα προηγούμενα κεφάλαια παρουσιάστηκαν οι τεχνικές μέθοδοι μετατροπής μουσικής σε οπτικο-απτικά σήματα, εδώ εξετάζεται πώς αυτές οι μέθοδοι γίνονται αντιληπτές από άτομα με προβλήματα ακοής.

### 5.2 Η Σημασία της Πολυαισθητηριακής Προσέγγισης στα Άτομα με Προβλήματα Ακοής

Η πολυαισθητηριακή προσέγγιση στην αντίληψη της μουσικής αναδεικνύει τη σημασία της ταυτόχρονης ενεργοποίησης πολλαπλών αισθητηριακών διαύλων για την εμπλουτισμένη και ολοκληρωμένη εμπειρία του ήχου. Σε επίπεδο νευρολογίας, η ενσωμάτωση ακουστικών, απτικών και οπτικών ερεθισμάτων μπορεί να περιγραφεί μαθηματικά μέσω συστημάτων πολυδιάστατων συναρτήσεων ενεργοποίησης νευρώνων. Θεωρητικά, αν θεωρήσουμε  $x_a(t)$  ως το ακουστικό σήμα,  $x_h(t)$  ως την απτική δόνηση και  $x_v(t)$  ως οπτικά ερεθίσματα, η συνδυασμένη **πολυαισθητηριακή εμπειρία**  $y(t)$  μπορεί να μοντελοποιηθεί ως:

$$y(t) = f_a(x_a(t)) + f_h(x_h(t)) + f_v(x_v(t)) + \epsilon(t) \quad (5.1)$$

όπου  $f_a, f_h, f_v$  είναι οι μετασχηματισμοί των αντίστοιχων αισθητηριακών σημάτων στο κεντρικό νευρικό σύστημα και  $\epsilon(t)$  αναπαριστά θόρυβο ή τυχαίες νευρωνικές διακυμάνσεις [70]. Η θεωρία της πολυαισθητηριακής συνένωσης υποστηρίζει ότι η σύγκλιση αυτών των σημάτων σε περιοχές όπως ο

φλοιός του προσαγωγίου και ο ημισφαιρικός φλοιός της ωχράς κηλίδας ενισχύει την επεξεργασία των μουσικών μοτίβων, οδηγώντας σε υψηλότερη ευαισθησία σε ρυθμικές και μελωδικές δομές [71].

Η μαθηματική ανάλυση της **πολυαισθητηριακής ενίσχυσης** μπορεί να προσεγγιστεί με τη χρήση μοντέλων πληροφορικής όπως η θεωρία πληροφορίας Shannon. Αν  $I_a$  και  $I_v$  αντιπροσωπεύουν την πληροφορία που παρέχεται αντίστοιχα από ακουστικά, απτικά και οπτικά κανάλια, τότε η συνολική πληροφορία  $I_{total}$  υπό υπόθεση ανεξαρτησίας μπορεί να γραφεί ως:

$$I_{total} = I_a + I_h + I_v - I_{redundancy} \quad (5.2)$$

όπου  $I_{redundancy}$  μετρά την επικαλυπτόμενη πληροφορία μεταξύ των αισθήσεων [72]. Η πρακτική εφαρμογή αυτού του μοντέλου στην εκπαίδευση ατόμων με βαρηκοΐα ή κώφωση έχει δείξει ότι η ενίσχυση της μουσικής εμπειρίας μέσω απτικών δονήσεων και οπτικών δεικτών οδηγεί σε σημαντική βελτίωση της αντίληψης ρυθμού και μελωδίας, υπογραμμίζοντας τη σημασία της συνδυασμένης πολυαισθητηριακής στρατηγικής [53].

Επιπλέον, η πολυαισθητηριακή προσέγγιση συνδέεται με τις θεωρίες **νευρωνικής πλαστικότητας**. Η ταυτόχρονη διέγερση διαφορετικών αισθητηριακών καναλιών μπορεί να ενισχύσει τις συνδέσεις μεταξύ νευρώνων σε συστήματα όπως ο θάλαμος και ο ακουστικός φλοιός, επιτρέποντας την αναδιαμόρφωση νευρωνικών κυκλωμάτων με βάση τη διαθεσιμότητα αισθητηριακής πληροφορίας [73]. Αυτή η διαδικασία μπορεί να μοντελοποιηθεί μέσω εξισώσεων διαφορών ή διαφορικών εξισώσεων που περιγράφουν την εξέλιξη της συνδεσιμότητας  $C(t)$  με τον χρόνο:

$$\frac{dC}{dt} = a \sum_i f_i(x_i(t)) - \beta C(t) \quad (5.3)$$

όπου  $a$  είναι ο ρυθμός ενίσχυσης συνδέσεων,  $\beta$  η φυσική διάσπαση των συνδέσεων,  $f_i(x_i(t))$  και οι εισροές από τα πολυαισθητηριακά κανάλια [74]. Με αυτόν τον τρόπο, η πολυαισθητηριακή εμπειρία δεν αποτελεί απλά πρόσθετο αισθητηριακό ερέθισμα αλλά οδηγεί σε μακροχρόνιες νευρωνικές προσαρμογές που ενισχύουν την αντίληψη της μουσικής σε άτομα με περιορισμένη ακουστική πρόσβαση.

Συνοψίζοντας, η πολυαισθητηριακή προσέγγιση αποτελεί κρίσιμο εργαλείο για την ενίσχυση της μουσικής εμπειρίας, τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο, συνδυάζοντας μαθηματικά μοντέλα επεξεργασίας πληροφορίας, νευρολογικές δομές και εφαρμογές στην ειδική εκπαίδευση [75]. Η ενσωμάτωση ακουστικών, απτικών και οπτικών στοιχείων δημιουργεί ένα πλουσιότερο πεδίο αντιληπτικής επεξεργασίας, ενισχύοντας τη βιωσιμότητα και την κατανόηση της μουσικής μέσω αισθητηριακής συνέργειας.

### 5.3 Η Νευρολογική Βάση της Πολυαισθητηριακής Αντίληψης

Νευρολογικά, η πολυαισθητηριακή αντίληψη βασίζεται σε ένα δίκτυο περιοχών του εγκεφάλου που περιλαμβάνει τον φλοιό του εγκεφαλικού φλοιού (cerebral cortex), τους υποφλοιώδεις πυρήνες και τις περιοχές σύνθεσης των αισθητηριακών ερεθισμάτων. Ιδιαίτερα σημαντικές θεωρούνται οι περιοχές του εγκεφαλικού φλοιού που σχετίζονται με την αίσθηση της αφής και την οπτική επεξεργασία, όπως ο πρωτογενής σωματοαισθητικός φλοιός (S1) και ο οπτικός φλοιός (V1), καθώς και οι πολυαισθητηριακοί φλοιϊκοί κόμβοι στον ανώτερο κροταφικό λοβό, όπου γίνεται η ενσωμάτωση των διαφορετικών ερεθισμάτων [76].

Έρευνες με τεχνικές νευροαπεικόνισης, όπως η λειτουργική μαγνητική τομογραφία (fMRI), έχουν δείξει ότι η απουσία ή η μείωση της ακουστικής εισροής προκαλεί επανακατανομή της νευρωνικής δραστηριότητας σε άλλες περιοχές του εγκεφάλου. Συγκεκριμένα, ο φλοιός που κανονικά συμμετέχει στην επεξεργασία ήχων μπορεί να ενεργοποιείται σε οπτικά ή απτικά ερεθίσματα, μια διαδικασία γνωστή ως νευρωνική πλαστικότητα. Αυτή η πλαστικότητα επιτρέπει στα άτομα με βαρηκοΐα να αποκτούν εμπλουτισμένη αντίληψη της μουσικής μέσω ερεθισμάτων που δεν βασίζονται αποκλειστικά στον ήχο, όπως η δόνηση και η οπτική παρατήρηση της κίνησης ή των δονήσεων των οργάνων [77].

### 5.3.1 Νευρωνική Κωδικοποίηση

Η νευρωνική κωδικοποίηση (neural coding) περιγράφει τον τρόπο με τον οποίο οι νευρώνες μετατρέπουν και μεταφέρουν πληροφορίες μέσω ηλεκτρικών σημάτων, γνωστών ως δυναμικά ενέργειας ή "spikes". Αυτή η διαδικασία επιτρέπει στον εγκέφαλο να αναπαριστά και να επεξεργάζεται τα αισθητηριακά ερεθίσματα από το περιβάλλον. Στην αίσθηση της αφής, η πληροφορία κωδικοποιείται κυρίως με δύο μηχανισμούς: **rate coding** και **temporal coding** [78].

**1. Rate coding:** Ο ρυθμός εκφόρτισης ενός νευρώνα υπολογίζεται από τη σχέση:

$$\lambda = \frac{N_{spikes}}{\Delta t} \quad (5.4)$$

- $\lambda$  → Ρυθμός εκφόρτισης (firing rate), δηλαδή πόσα δυναμικά ενέργειας παράγει ένας νευρώνας ανά μονάδα χρόνου (μετράται συνήθως σε Hz = εκφορτίσεις/δευτερόλεπτο).
- $N_{spikes}$  → Ο αριθμός των "spikes" (δυναμικών ενέργειας) που κατέγραψε ο νευρώνας σε ένα συγκεκριμένο χρονικό διάστημα.
- $\Delta t$  → Το χρονικό διάστημα παρατήρησης (π.χ. 100 ms, 1 sec κλπ).

Ο ρυθμός εκφόρτισης συσχετίζεται με την ένταση του αισθητηριακού ερεθίσματος, παρέχοντας πληροφορία για τη δύναμη ή τη διάρκεια της αφής, της πίεσης ή της δόνησης [79].

2. **Temporal coding:** Η χρονική διάταξη των spikes παρέχει πληροφορία για γρήγορες ή δυναμικές αλλαγές στα αισθητηριακά ερεθίσματα, όπως οι δονήσεις υψηλής συχνότητας ή οι λεπτές μεταβολές υψής. Η πληροφορία κωδικοποιείται μέσω του χρονικού διαστήματος μεταξύ δύο διαδοχικών spikes ( $\Delta t_i$ ), και η **ένταση ή συχνότητα ενός ερεθίσματος** μπορεί να περιγραφεί ως:

$$f(t) \sim \frac{1}{\Delta t_i} \quad (5.5)$$

όπου μικρότερα χρονικά διαστήματα μεταξύ spikes αντιστοιχούν σε υψηλότερη συχνότητα ερεθισμάτων, όπως στις δονήσεις ή τη μουσική αντίληψη [80].

Η νευρωνική κωδικοποίηση χρησιμοποιείται από τα αισθητηριακά συστήματα με διάφορους τρόπους:

1. **Ακοή:** Η ένταση ενός ήχου αναπαρίσταται κυρίως μέσω του **rate coding**, ενώ η **temporal coding** επιτρέπει την αναγνώριση συχνοτήτων και ρυθμών. Αυτό επιτρέπει στους ανθρώπους να διακρίνουν μελωδίες, ρυθμούς και ομιλία [79].
2. **Όραση:** Στο οπτικό σύστημα, ο ρυθμός των **spikes** κωδικοποιεί την ένταση φωτός, ενώ η temporal coding διευκολύνει την ανίχνευση γρήγορων αλλαγών, κίνησης και λεπτομερειών σε δυναμικά οπτικά ερεθίσματα [78].
3. **Αφή:** Η temporal coding είναι ιδιαίτερα σημαντική για την αντίληψη δονήσεων υψηλής συχνότητας και λεπτών υφών, ενώ το rate coding μεταφέρει πληροφορίες για την ένταση της πίεσης. Σε άτομα με μειωμένη ακοή, η αφή χρησιμοποιείται για την αντίληψη μουσικών ρυθμών και δονήσεων, ενισχύοντας την πολυαισθητηριακή εμπειρία [80].

Η συνδυαστική χρήση των δύο μηχανισμών σε κάθε αισθητηριακό σύστημα επιτρέπει την ολοκληρωμένη αντίληψη του περιβάλλοντος και παρέχει δυνατότητες αντιστάθμισης των περιορισμών ενός αισθητηριακού καναλιού, όπως συμβαίνει σε άτομα με ακουστικές δυσκολίες [79].

## 5.4 Αντίληψη Ρυθμού και Δόνησης από Κωφούς/Βαρήκοους

Η **αντίληψη του ρυθμού** και η **αντίληψη δόνησης** σε άτομα με απώλεια ακοής αποτελούν ένα πεδίο έρευνας, που συνδυάζει τη φυσιολογία της αίσθησης, τη νευροεπιστήμη, τη μουσικολογία και τη μηχανική σημάτων. Παρόλο που η ακουστική είσοδος είναι ο κύριος φορέας ρυθμικής πληροφορίας στον γενικό πληθυσμό, σε κωφούς και βαρήκοους το σώμα χρησιμοποιεί εναλλακτικές αισθητηριακές οδούς, κυρίως την **απτική** και την **σωματοαισθητηριακή** αντίληψη [1].

### 5.4.1 Ψυχοακουστική – Ψυχοαφής Αντίληψη

Η μεταφορά ρυθμικής πληροφορίας μέσω δόνησης υπόκειται σε ψυχοφυσικούς νόμους, όπως ο νόμος του **Stevens**, όπου η αντιληπτή ένταση  $P$  σχετίζεται με το φυσικό ερέθισμα  $S$  μέσω της σχέσης:

$$P = k \cdot S^n \quad (5.6)$$

Για vibrotactile ερεθίσματα σε μεσαία συχνότητα (~60 Hz), μετρήσεις δείχνουν ότι  $n \approx 0,95$ , τιμή κοντά στο 1, γεγονός που επιτρέπει την πρακτική εφαρμογή του νόμου για την εκτίμηση της αντιληπτής έντασης δονήσεων [36].

Η αντίληψη του ρυθμού μέσω δόνησης είναι ιδιαίτερα σημαντική για κωφούς και βαρήκοους, καθώς οι δονήσεις μπορούν να μεταφέρουν πληροφορίες για παλμούς, tempo, και μοτίβα μουσικής που διαφορετικά θα ήταν αδύνατο να γίνουν αντιληπτά ακουστικά. Σε ελεγχόμενα πειράματα, η αναγνώριση παλμών μέσω δόνησης απαιτεί stimuli διάρκειας περίπου **50 ms** για να γίνουν αντιληπτά, ενώ η **μόλις αντιληπτή διαφορά (JND)** για ερεθίσματα 500 ms στα 100 Hz κυμαίνεται μεταξύ 100–300 ms, ανάλογα με την εμπειρία του χρήστη [2],[3].

Η αντίληψη του ρυθμού επηρεάζεται επίσης από την ένταση και τη χωρική κατανομή των αισθητήρων. Η χρήση πολλαπλών δονητών σε διαφορετικά σημεία του σώματος, όπως καρποί ή μέση, ενισχύει την αντίληψη πολυφωνικών μοτίβων και τη μουσική δομή μέσω αφής [3].

Οι παρατηρήσεις αυτές παρέχουν τη θεωρητική βάση για τον σχεδιασμό συσκευών απτικής επικοινωνίας, όπως εφαρμογές που μετατρέπουν μουσική σε δονήσεις, ή βοηθητικά συστήματα παρακολούθησης ρυθμού σε εκπαιδευτικά και ψυχαγωγικά περιβάλλοντα για κωφούς και βαρήκοους.

#### 5.4.2 Μοντέλα Αντίληψης Ρυθμού μέσω Δόνησης

Η αντίληψη ρυθμού μπορεί να μοντελοποιηθεί με βάση την **απόκριση φάσης** των υποδοχέων και την κεντρική χρονική ολοκλήρωση [82].

##### 1. Μοντέλο Ανίχνευσης Παλμών

Υποθέτουμε σήμα:

$$x(t) = \sum_n A \cdot \sin(2\pi f_c t) \cdot w(t - n \cdot T) \quad (5.7)$$

όπου

- $x(t)$  : Χρονική μορφή του δονητικού σήματος που προσλαμβάνεται από τον υποδοχέα
- $A$  : Πλάτος του σήματος (μονάδες όπως παραπάνω)
- $f_c$  : Φέρουσα συχνότητα της δόνησης (Hz)
- $w(t)$  : Παράθυρο/περιβάλλουσα παλμού (π.χ. Gaussian ή rectangular)
- $n$  : Δείκτης αριθμού παλμού
- $T$  : Περίοδος ρυθμού (s), δηλαδή  $T=1/f_r$  όπου  $f_r$  η ρυθμική συχνότητα σε Hz. [83]

## 2. Ανάλυση στο Κεντρικό Νευρικό Σύστημα

Το σήμα υφίσταται φασματική ανάλυση μέσω μηχανισμών χρονικής δειγματοληψίας (~5–10 ms βήμα), και η αντιληπτή περιοδικότητα προκύπτει μέσω αυτοσυσχέτισης:

$$R(\tau) = \int_0^{T'} x(t) \cdot x(t + \tau) dt \quad (5.8)$$

- $R(\tau)$  : Συνάρτηση αυτοσυσχέτισης, δείκτης ομοιότητας σήματος για χρονική μετατόπιση  $\tau$
- $\tau$  : Χρονική μετατόπιση (s)
- $T'$  : Συνολικό διάστημα παρατήρησης (s)
- $x(t)$  : Σήμα όπως ορίστηκε παραπάνω. [83]

## 3. Συνδυασμός Πληροφορίας και Αντίληψη

Μετά την ανάλυση στο Κεντρικό Νευρικό Σύστημα, τα σήματα από πολλαπλούς υποδοχείς και κανάλια ενσωματώνονται για να σχηματιστεί η συνολική αντίληψη του ρυθμού. Αυτός ο συνδυασμός μπορεί να περιγραφεί μαθηματικά μέσω της **συνάρτησης αυτοσυσχέτισης** ή της έντασης των συνδυασμένων σημάτων:

$$R_{total}(\tau) = \sum_{i=1}^N w_i \int_0^{T'} x_i(t) \cdot x_i(t + \tau) dt \quad (5.9)$$

όπου:

- $R_{total}(\tau)$ : συνολική αυτοσυσχέτιση για χρονική μετατόπιση  $\tau$ , δείκτης ομοιότητας του συνολικού σήματος.
- $x_i(t)$  : σήμα από τον  $i$ -οστό υποδοχέα ή κανάλι.
- $w_i$ : βάρος που αντιστοιχεί στη συνεισφορά κάθε υποδοχέα στην αντίληψη.
- $N$  : συνολικός αριθμός υποδοχέων/καναλιών.
- $T'$  : συνολικό διάστημα παρατήρησης. [84]

Η παραπάνω σχέση δείχνει πώς η πολυκαναλική πληροφορία ενσωματώνεται και συγχρονίζεται για να δημιουργηθεί η αντιληπτή περιοδικότητα.

### Πλαστικότητα Εγκεφάλου και Διααισθητηριακή Αντίληψη

Η **νευροπλαστικότητα** σε άτομα με πρόωμη ή συγγενή κώφωση οδηγεί σε αξιοσημείωτες αλλαγές στην οργάνωση του εγκεφάλου. Ο **ακουστικός φλοιός** δεν μένει ανενεργός αντίθετα, επαναπρογραμματίζεται για να επεξεργάζεται μη ακουστικά σήματα, όπως δονήσεις ή οπτικά μοτίβα [3].

## Μηχανισμός Διααισθητηριακής Ενσωμάτωσης

- Η **περιοχή A1** (πρωτογενής ακουστικός φλοιός) και η **περιοχή belt/parabelt** ενεργοποιούνται από δονητικά ερεθίσματα μέσω συνδέσεων με τον σωματοαισθητικό φλοιό (S1).
- Το φαινόμενο αυτό έχει παρατηρηθεί σε fMRI και MEG μελέτες, όπου η παρουσία δονητικού ρυθμού προκαλεί BOLD σήμα παρόμοιο με αυτό που θα προκαλούσε ακουστικός ρυθμός [4].

### 5.4.3 Μουσικολογικές Εφαρμογές και Τεχνολογίες Δόνησης

#### Συστήματα Παρουσίασης Δόνησης

##### 1. Δάπεδα Δόνησης (tactile floors)

Ολόκληρη η επιφάνεια μετατρέπεται σε μεγάλης κλίμακας μετατροπέα (actuator), με συχνοτική απόκριση έως ~250 Hz. Οι χαμηλές συχνότητες (< 40 Hz) γίνονται αντιληπτές κυρίως από τα πέλματα και μεταφέρονται μέσω του σκελετού (bone conduction) [85].

##### 2. Γιλέκα Δόνησης (tactile vests)

Περιέχουν πολλαπλούς transducers σε διαφορετικές περιοχές του σώματος, επιτρέποντας πολυκαναλική παρουσίαση (π.χ., μπάσο στο στήθος, μεσαίες συχνότητες στην πλάτη). Αυτό επιτρέπει την **χωρική χαρτογράφηση** της μουσικής πληροφορίας [86].

##### 3. Συστήματα Οπτικο-Απτικής Συγχώνευσης

Χρήση φωτεινών ερεθισμάτων (LEDs) σε συγχρονισμό με τις δονήσεις ενισχύει την ανίχνευση του ρυθμού μέσω **multisensory integration** [87].

## 5.5 Έρευνες για τη Μουσική Βιωσιμότητα μέσω Δονήσεων

Η μελέτη της μουσικής βιωσιμότητας μέσω δονήσεων αποτελεί σημαντικό πεδίο της απτικής μουσικής αντίληψης, συνδυάζοντας φυσική, νευροεπιστήμες και γνωστική ψυχολογία. Οι έρευνες επικεντρώνονται στο πώς οι δονήσεις, που παράγονται από μουσικά όργανα ή ηχητικά συστήματα, μετατρέπονται σε αισθητηριακές πληροφορίες αντιληπτές ακόμη και από άτομα με σοβαρές ακουστικές δυσκολίες.

Η διάδοση των δονήσεων σε ελαστικά μέσα περιγράφεται από την εξίσωση κύματος:

$$\frac{\partial^2 u}{\partial t^2} = v^2 \nabla^2 u \quad (5.10)$$

όπου  $u(x,t)$  η μετατόπιση σε ένα σημείο του μέσου,  $v$  η ταχύτητα διάδοσης του κύματος στο μέσο και  $\nabla^2$  ο χωρικός Λαπλασιανός τελεστής. Η μαθηματική αυτή περιγραφή επιτρέπει την ανάλυση συχνοτήτων και amplitudes των δονήσεων, κρίσιμων για την αντίληψη ρυθμού και αρμονίας μέσω της αφής [81].

Για την αποσύνθεση των δονητικών σημάτων σε συχνοτικά συστατικά χρησιμοποιούνται μοντέλα Fourier:

$$u(t) = \sum_{n=0}^{\infty} A_n \cos(2\pi f_n t + \Phi_n) \quad (5.11)$$

όπου  $A_n$  η amplitude κάθε συχνοτικού συνιστώσα,  $f_n$  η συχνότητα και  $\Phi_n$  η φάση. Αυτή η αναπαράσταση εξηγεί πώς οι χρήστες μπορούν να αναγνωρίσουν βασικά μουσικά μοτίβα, όπως ρυθμούς, παλμούς και μελωδίες, ακόμη και σε πολυσύνθετα έργα με πολλαπλά όργανα. Επιπλέον, έρευνες έχουν δείξει ότι η επαναλαμβανόμενη έκθεση σε δονητικά μοτίβα ενισχύει τη γνωστική και κινητική μνήμη, δημιουργώντας έναν μηχανισμό μάθησης παρόμοιο με αυτόν της ακουστικής μουσικής εκπαίδευσης [81].

## 5.6 Συμπεράσματα

Στο πέμπτο κεφάλαιο εξετάστηκε η αντίληψη των ήχων και της μουσικής από άτομα με προβλήματα ακοής, με έμφαση στον τρόπο που η πολυαισθητηριακή προσέγγιση μπορεί να εμπλουτίσει την εμπειρία τους. Αναλύθηκε η σημασία της ταυτόχρονης ενεργοποίησης ακουστικών, οπτικών και απτικών καναλιών, τόσο σε θεωρητικό όσο και σε νευρολογικό επίπεδο, αναδεικνύοντας πώς η νευροπλαστικότητα και η πολυαισθητηριακή συνένωση συμβάλλουν στην καλύτερη κατανόηση ρυθμού και μελωδίας. Ιδιαίτερη βαρύτητα δόθηκε στη νευρωνική κωδικοποίηση, όπου μηχανισμοί όπως το *rate coding* και το *temporal coding* εξηγούν πώς ο εγκέφαλος αναπαριστά και αντιλαμβάνεται δονήσεις και ρυθμούς, προσφέροντας μια εναλλακτική οδό πρόσληψης της μουσικής. Εξετάστηκε επίσης η ψυχοαφής διάσταση, με παραδείγματα για το πώς οι δονήσεις μπορούν να μεταφέρουν ρυθμικά μοτίβα και παλμούς, επιτρέποντας σε κωφούς και βαρήκοους να αναγνωρίσουν tempo και μουσικές δομές. Παράλληλα, παρουσιάστηκαν εφαρμογές και τεχνολογίες όπως γιλέκα δόνησης, δάπεδα δόνησης και συστήματα οπτικο-απτικής σύζευξης, που δείχνουν πώς η τεχνολογία μπορεί να αξιοποιήσει τη φυσιολογία της αφής για να επαναπροσδιορίσει την πρόσβαση στη μουσική. Τέλος, οι έρευνες για τη βιωσιμότητα της μουσικής μέσω δονήσεων έδειξαν ότι η επαναλαμβανόμενη έκθεση σε απτικά μοτίβα ενισχύει τη μνήμη και τη μάθηση, προσφέροντας μια βιωματική και συμμετοχική εμπειρία.

## Κεφάλαιο 6ο: Σχεδιασμός της Εφαρμογής

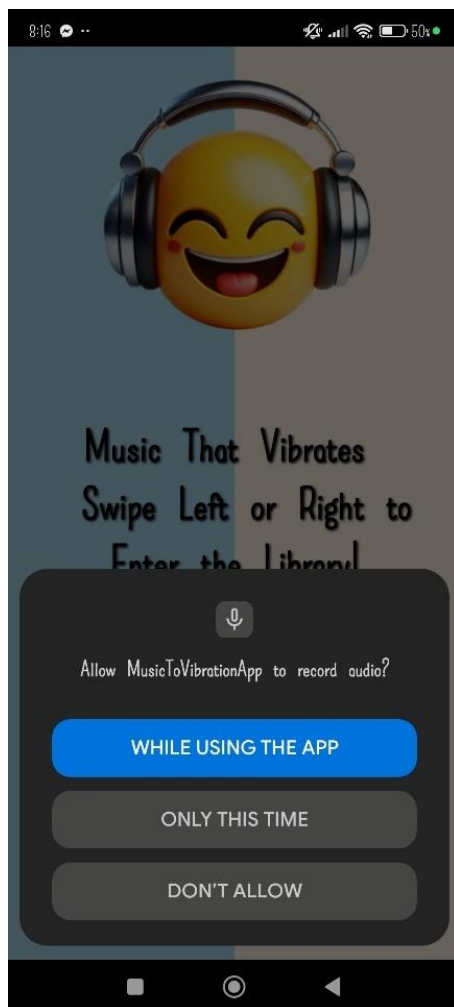
### 6.1 Εισαγωγή

Σε αυτό το κεφάλαιο βλέπουμε πώς οι έννοιες και οι τεχνολογίες που παρουσιάστηκαν στα προηγούμενα κεφάλαια εφαρμόζονται στην πράξη μέσα στην ανάπτυξη της εφαρμογής. Η συγκεκριμένη εφαρμογή αξιοποιεί βασικές έννοιες της **ψηφιακής επεξεργασίας σήματος (DSP)** για την ανίχνευση ρυθμικών μοτίβων (Beat Detection). Η διαδικασία στηρίζεται στον **Μετασχηματισμό Fourier (FFT)**, όπου επιτρέπει τη μετάβαση από το πεδίο του χρόνου στο πεδίο της συχνότητας, ώστε να αναλυθεί το μουσικό σήμα και να εντοπιστούν περιοδικές αυξήσεις στην ένταση (Κεφάλαιο 3). Το φάσμα χωρίζεται σε ζώνες συχνότητων (χαμηλές, μεσαίες, υψηλές), ενώ η ένταση κάθε ζώνης υπολογίζεται από τα πραγματικά και φανταστικά μέρη του FFT, στηριζόμενο στις αρχές της φασματικής ανάλυσης (Κεφάλαιο 2). Για τη βελτίωση της ακρίβειας εφαρμόζεται **κινούμενος μέσος όρος (moving average)** ώστε να μειωθεί ο θόρυβος, ενώ η τελική απόφαση για την παρουσία beat βασίζεται σε **adaptive thresholding**, δηλαδή σε δυναμικά κατώφλια που προσαρμόζονται στην ένταση του σήματος. Όπως αναλύθηκε στο Κεφάλαιο 2, η φασματική ανάλυση και οι τεχνικές εξομάλυνσης προσφέρουν το μαθηματικό υπόβαθρο για την ανίχνευση χαρακτηριστικών, ενώ όπως παρουσιάστηκε στο Κεφάλαιο 3 η απτική αλληλεπίδραση αξιοποιεί τα αποτελέσματα αυτής της διαδικασίας, και στο Κεφάλαιο 4 η αρχιτεκτονική του Haptic Music Player με Visualizer τα ενσωματώνει σε μια ολοκληρωμένη εμπειρία.

### 6.2 Άδειες Πρόσβασης, Λειτουργικότητα και App Icon

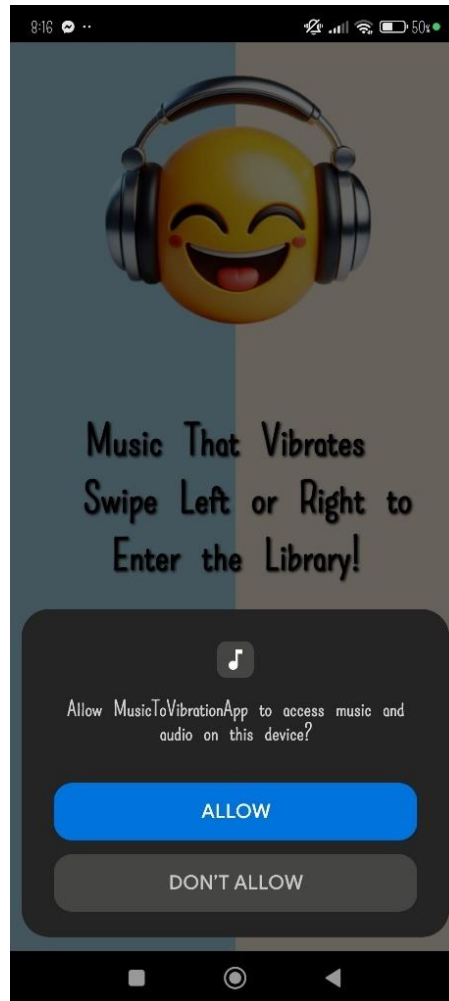
Κατά την εκκίνηση, η εφαρμογή ζητά από τον χρήστη την παροχή των απαραίτητων αδειών πρόσβασης, προκειμένου να διασφαλιστεί η πλήρης λειτουργικότητά της. Συγκεκριμένα, απαιτείται πρόσβαση:

- Στο **μικρόφωνο** όπου για τεχνικούς λόγους απαιτήθηκε η άδεια **RECORD\_AUDIO**. Η άδεια χρησιμοποιήθηκε μόνο για να επιτραπεί η λειτουργία του Visualizer σε ορισμένες εκδόσεις Android.



Εικόνα 6.1: Mic Permission

- Στα **πολυμέσα** της συσκευής ώστε η εφαρμογή να μπορεί να διαβάζει και να αναπαράγει τα αποθηκευμένα μουσικά αρχεία, διαφορετικά δεν εμφανίζεται κανένα τραγούδι.



Εικόνα 6.2: Media Permission

Η εξασφάλιση των παραπάνω αδειών είναι αναγκαία ώστε η εφαρμογή να υποστηρίζει τόσο την ανάλυση και επεξεργασία μουσικής σε πραγματικό χρόνο όσο και την αλληλεπίδραση με υπάρχουσες βιβλιοθήκες πολυμέσων της συσκευής.

```

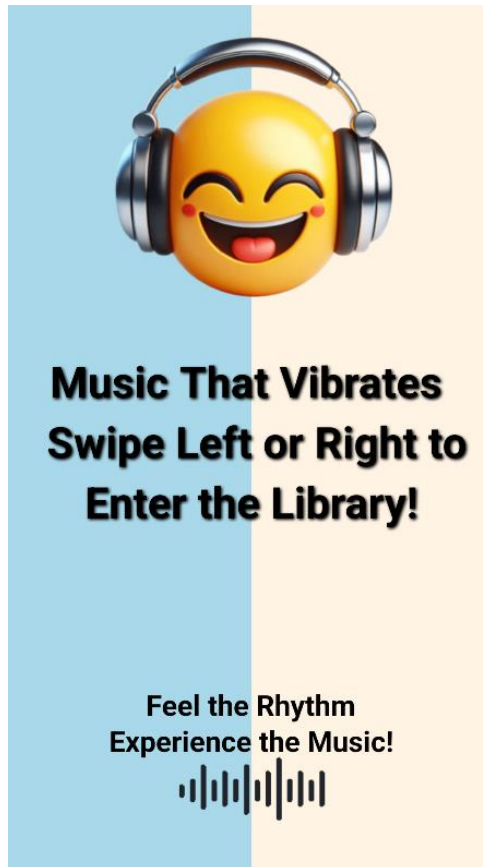
5 <uses-permission android:name="android.permission.VIBRATE" />
6 <uses-permission android:name="android.permission.RECORD_AUDIO" />
7 <uses-permission android:name="android.permission.READ_MEDIA_AUDIO" />
8 <uses-permission android:name="android.permission.ACCESS_NOTIFICATION_POLICY" />
9
10 <application
11     android:allowBackup="true"
12     android:dataExtractionRules="@xml/data_extraction_rules"
13     android:fullBackupContent="@xml/backup_rules"
14     android:icon="@mipmap/ic_launcher"
15     android:label="MusicToVibrationApp"
16     android:roundIcon="@mipmap/ic_launcher_round"
17     android:supportsRtl="true"
18     android:theme="@style/Theme.MusicToVibrationApp"
19     tools:targetApi="31">
20     <activity android:name=".PlayerMode" />

```

Εικόνα 6.3: Άδειες και εικονίδιο εφαρμογής

### 6.3 Περιγραφή Κώδικα Αρχικής Οθόνης (Home Page)

Για την αρχική οθόνη της εφαρμογής σχεδιάστηκε ένα απλό layout σε XML, το οποίο χωρίζει την οθόνη σε δύο μέρη με διαφορετικά χρώματα φόντου και περιλαμβάνει βασικά στοιχεία παρουσίασης, όπως το λογότυπο, ενημερωτικά μηνύματα και ένα κινούμενο γραφικό που παραπέμπει στην κίνηση και τον παλμό της μουσικής.



Εικόνα 6.4: Αρχική Οθόνη

#### 6.3.1 XML

Ο κώδικας που παρουσιάζεται αφορά τον αρχικό χειρισμό της διεπαφής χρήστη (UI) της εφαρμογής, με στόχο τον διαχωρισμό της οθόνης σε δύο περιοχές δεξιά, αριστερά και την προβολή της κινούμενης εικόνας που αναπαριστά τον ήχο.

```

12      <!-- Left half layout -->
13      <RelativeLayout
14          android:id="@+id/leftHalf"
15          android:layout_width="0dp"
16          android:layout_height="match_parent"
17          android:background="#A8D8EA"
18          app:layout_constraintStart_toStartOf="parent"
19          app:layout_constraintEnd_toStartOf="@+id/rightHalf"
20          app:layout_constraintTop_toTopOf="parent"
21          app:layout_constraintBottom_toBottomOf="parent">
22      </RelativeLayout>
23
24      <!-- Right half layout -->
25      <RelativeLayout
26          android:id="@+id/rightHalf"
27          android:layout_width="0dp"
28          android:layout_height="match_parent"
29          android:background="#FFF3E4"
30          app:layout_constraintStart_toEndOf="@+id/leftHalf"
31          app:layout_constraintEnd_toEndOf="parent"
32          app:layout_constraintTop_toTopOf="parent"
33          app:layout_constraintBottom_toBottomOf="parent">
34      </RelativeLayout>

```

Εικόνα 6.5: Διαχωρισμός Οθόνης xml

```

95      <!-- Gif at the bottom center -->
96      <ImageView
97          android:id="@+id/soundWaveAnimation"
98          android:layout_width="150dp"
99          android:layout_height="150dp"
100         app:layout_constraintBottom_toBottomOf="parent"
101         app:layout_constraintStart_toStartOf="parent"
102         app:layout_constraintEnd_toEndOf="parent"
103         app:layout_constraintHorizontal_bias="0.5"
104         android:src="@drawable/sound_wave_animation" />

```

Εικόνα 6.6: Gif xml



Εικόνα 6.7: Gif

### 6.3.2 KOTLIN

**Imports** (γραμμές 3-16): Εισαγωγή απαραίτητων βιβλιοθηκών που χρειάζεται η κλάση για να λειτουργήσει.

```

1 package com.example.musicvibrationapp
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.view.GestureDetector
6 import android.view.MotionEvent
7 import android.widget.ImageView
8 import android.widget.RelativeLayout
9 import android.widget.Toast
10 import androidx.appcompat.app.AppCompatActivity
11 import com.bumptech.glide.Glide
12 import androidx.core.content.ContextCompat
13 import android.content.pm.PackageManager
14 import androidx.core.app.ActivityCompat
15 import android.Manifest
16 import android.os.Build

```

Εικόνα 6.8: MainActivity Imports

#### Δήλωση Κλάσης και Μεταβλητών (γραμμές 20-24)

Όλα τα στοιχεία της οθόνης περιλαμβάνουν: σταθερά για αιτήματα δικαιωμάτων (**PERMISSION\_REQUEST\_CODE**), ανίχνευση χειρονομιών (**gestureDetector**), κινούμενη εικόνα ηχητικού κύματος (Gif) (**soundWaveAnimationView**) και βασικό layout με listener αφής (**rootLayout**).

```

18 class MainActivity : AppCompatActivity() {
19
20     private val PERMISSION_REQUEST_CODE = 101
21
22     private lateinit var gestureDetector: GestureDetector
23     private lateinit var soundWaveAnimationView: ImageView
24     private lateinit var rootLayout: RelativeLayout

```

Εικόνα 6.9: MainActivity μεταβλητές

#### Η μέθοδος onCreate() (γραμμές 26-47)

Η μέθοδος **onCreate()** στην κύρια δραστηριότητα καλεί **setContentview()** για να ορίσει το layout, **showAnimation()** για την εμφάνιση της κινούμενης εικόνας, αρχικοποιεί στοιχεία διεπαφής. Εφαρμόζεται **listener** αφής στο **rootLayout**, για πλοήγηση με χειρονομίες και ελέγχει μέσω **SharedPreferences** αν είναι πρώτη εκτέλεση ή αν λείπουν δικαιώματα, τα οποία ζητούνται με **checkAndRequestPermissions()** ή συνεχίζεται κανονικά με **onPermissionsGranted()**

```

26 override fun onCreate(savedInstanceState: Bundle?) {
27     super.onCreate(savedInstanceState)
28     setContentView(R.layout.activity_main)
29
30     showAnimation()
31
32     rootLayout = findViewById(R.id.rootLayout)
33
34     // Set up gesture detector
35     gestureDetector = GestureDetector(context: this, SwipeGestureListener())
36
37     // Set touch listener on the root layout (covers both halves)
38     rootLayout.setOnTouchListener { _, event -> gestureDetector.onTouchEvent(event) }
39
40     val prefs = getSharedPreferences(name: "MyAppPrefs", MODE_PRIVATE)
41     val firstLaunch = prefs.getBoolean(key: "FIRST_LAUNCH", defValue: true)
42     if (firstLaunch || !arePermissionsGranted()) {
43         checkAndRequestPermissions()
44     } else {
45         onPermissionsGranted()
46     }
47 }

```

Εικόνα 6.10: MainActivity onCreate

### Μέθοδοι showAnimation() και onTouchEvent() (γραμμές 49-59)

Η μέθοδος **showAnimation()** φορτώνει και προβάλλει το animation με τη βιβλιοθήκη Glide, ενώ ο listener αφής προωθεί τα γεγονότα στο **gestureDetector** για αναγνώριση χειρονομιών και ενεργοποίηση λειτουργιών.

```

49 private fun showAnimation() {
50     soundWaveAnimationView = findViewById(R.id.soundWaveAnimation)
51
52     Glide.with(activity: this) RequestManager
53         .load(R.drawable.sound_wave_animation) RequestBuilder<Drawable!>
54         .into(soundWaveAnimationView)
55 }
56
57 ZoZoTs99
58 override fun onTouchEvent(event: MotionEvent): Boolean {
59     return gestureDetector.onTouchEvent(event)
60 }

```

Εικόνα 6.11: MainActivity animation και onTouch

**Εσωτερική κλάση SwipeGestureListener** (γραμμές 62-87)

Επεκτείνει το **GestureDetector.SimpleOnGestureListener** για ανίχνευση οριζόντιων swipes, χρησιμοποιεί τα όρια **SWIPE\_THRESHOLD** και **SWIPE\_VELOCITY\_THRESHOLD**, και στη μέθοδο **onFling()** εντοπίζει τα swipes ώστε να καλέσει τη **navigateToActivity()**.

```

62     inner class SwipeGestureListener : GestureDetector.SimpleOnGestureListener() {
63         private val SWIPE_THRESHOLD = 100
64         private val SWIPE_VELOCITY_THRESHOLD = 100
65
66         ZoZoTs99 *
67         override fun onFling(
68             e1: MotionEvent?,
69             e2: MotionEvent,
70             velocityX: Float,
71             velocityY: Float
72         ): Boolean {
73             if (e1 != null && e2 != null) {
74                 val diffX = e2.x - e1.x
75                 val diffY = e2.y - e1.y
76                 // Only check for horizontal swipes
77                 if (Math.abs(diffX) > Math.abs(diffY)) {
78                     if (Math.abs(diffX) > SWIPE_THRESHOLD &&
79                         Math.abs(velocityX) > SWIPE_VELOCITY_THRESHOLD) {
80                         // Trigger navigation to the desired activity
81                         navigateToActivity()
82                         return true
83                     }
84                 }
85             }
86             return false
87         }
88     }

```

Εικόνα 6.12: MainActivity εσωτερική κλάση (swipe)

**navigateToActivity()** (γραμμές 89-93)

Η συνάρτηση αυτή είναι υπεύθυνη για την πλοήγηση του χρήστη στην οθόνη **ActivityLibrary**.

```

89     private fun navigateToActivity() {
90         // Navigating to ActivityLibrary:
91         val intent = Intent(packageContext, this, ActivityLibrary::class.java)
92         startActivity(intent)
93     }

```

Εικόνα 6.13: MainActivity navigateToActivity()

**checkAndRequestPermissions()** (γραμμές 95-114)

Η συνάρτηση ελέγχει, μέσω της μεθόδου **ContextCompat.checkSelfPermission()**, αν έχουν δοθεί τα δικαιώματα για πρόσβαση στα αρχεία ήχου και στο μικρόφωνο. Αν κάποιο λείπει, προστίθεται σε μια λίστα και ζητείται από τον χρήστη να το παραχωρήσει μέσω του συστήματος δικαιωμάτων του Android.

```

95     private fun checkAndRequestPermissions() {
96         var permissions = mutableListOf<String>()
97
98         if (ContextCompat.checkSelfPermission(context: this, Manifest.permission.READ_MEDIA_AUDIO)
99             != PackageManager.PERMISSION_GRANTED
100        ) {
101             permissions.add(Manifest.permission.READ_MEDIA_AUDIO)
102        }
103
104        if (ContextCompat.checkSelfPermission(context: this, Manifest.permission.RECORD_AUDIO)
105            != PackageManager.PERMISSION_GRANTED
106        ) {
107            permissions.add(Manifest.permission.RECORD_AUDIO)
108        }
109
110        if (permissions.isNotEmpty()) {
111            ActivityCompat.requestPermissions(activity: this, permissions.toTypedArray(),
112                PERMISSION_REQUEST_CODE)
113        }
114    }

```

Εικόνα 6.14: MainActivity checkAndRequestPermissions()

Η μέθοδος `onRequestPermissionsResult()` (γραμμές 115-132) ελέγχει το αποτέλεσμα του αιτήματος για δικαιώματα. Αν όλα τα απαιτούμενα δικαιώματα έχουν παραχωρηθεί, καλείται η συνάρτηση `onPermissionsGranted()` ώστε η εφαρμογή να συνεχίσει κανονικά. Σε διαφορετική περίπτωση, εμφανίζεται σχετικό μήνυμα στον χρήστη μέσω `Toast` και η εφαρμογή τερματίζεται.

```

115     override fun onRequestPermissionsResult(
116         requestCode: Int,
117         permissions: Array<out String>,
118         grantResults: IntArray
119     ) {
120         super.onRequestPermissionsResult(requestCode, permissions, grantResults)
121         if (requestCode == PERMISSION_REQUEST_CODE) {
122             // If any permission is denied, close the app
123             if (grantResults.isNotEmpty() &&
124                 grantResults.all { it == PackageManager.PERMISSION_GRANTED }) {
125                 onPermissionsGranted()
126             } else {
127                 Toast.makeText(context: this, text: "Permissions denied. App will close.",
128                     Toast.LENGTH_SHORT).show()
129                 finish()
130             }
131         }
132     }

```

Εικόνα 6.15: MainActivity onRequestPermissionsResult()

**onPermissionsGranted()** (γραμμές 134-144)

Η μέθοδος αυτή εκτελείται όταν όλα τα απαιτούμενα δικαιώματα έχουν παραχωρηθεί από τον χρήστη. Αρχικά, αποθηκεύει στις **SharedPreferences** ότι η εφαρμογή έχει εκκινηθεί και τα δικαιώματα έχουν δοθεί, ώστε να μην επαναλαμβάνεται ο έλεγχος σε επόμενες εκκινήσεις. Στη συνέχεια, φορτώνει το κύριο layout της εφαρμογής και εμφανίζει το **animation** του ηχητικού κύματος.

```

134     private fun onPermissionsGranted() {
135         // Save that permissions were granted and app initialized
136         sharedPreferences.edit()
137         sharedPreferences.putBoolean("FIRST_LAUNCH", false)
138         sharedPreferences.apply()
139
140
141         // proceed to main content
142         setContentView(R.layout.activity_main)
143         showAnimation()
144     }

```

Εικόνα 6.16: MainActivity onPermissionsGranted()

Η **arePermissionsGranted()** (γραμμές 146-160) ελέγχει προκαταβολικά αν τα απαιτούμενα δικαιώματα είναι ήδη διαθέσιμα, δηλαδή λειτουργεί πριν γίνει οποιοδήποτε αίτημα. Σε **παιλιότερες εκδόσεις Android** όπου το δικαίωμα για αρχεία ήχου δεν υποστηρίζεται, εμφανίζει μήνυμα και τερματίζει την εφαρμογή.

```

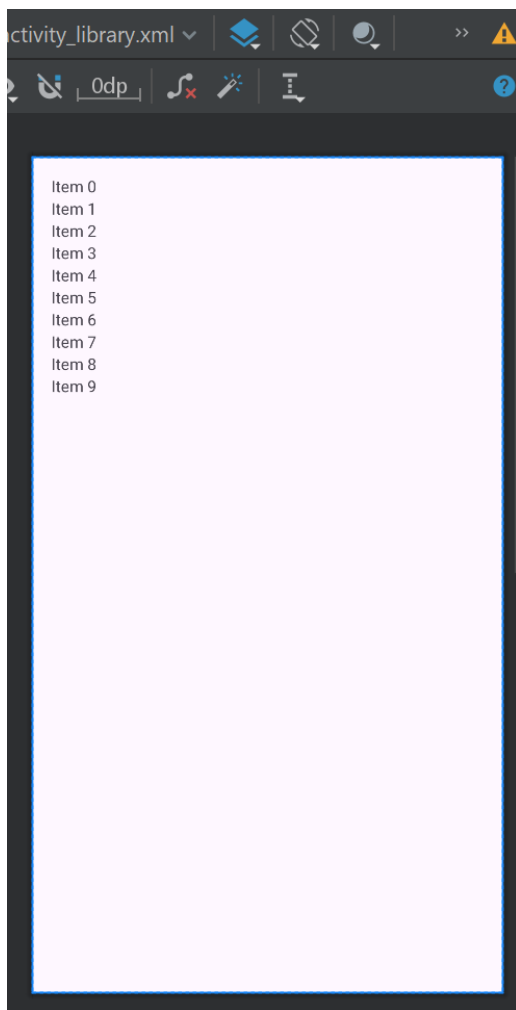
146     private fun arePermissionsGranted(): Boolean {
147         val mic = ContextCompat.checkSelfPermission(context, this,
148             Manifest.permission.RECORD_AUDIO) == PackageManager.PERMISSION_GRANTED
149         val media = if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
150             ContextCompat.checkSelfPermission(context, this,
151                 Manifest.permission.READ_MEDIA_AUDIO) == PackageManager.PERMISSION_GRANTED
152         } else {
153             Toast.makeText(context, this, "This application is not supported on this " +
154                 "version of Android.", Toast.LENGTH_SHORT).show()
155             finish()
156             false
157         }
158
159         return mic && media
160     }
161 }

```

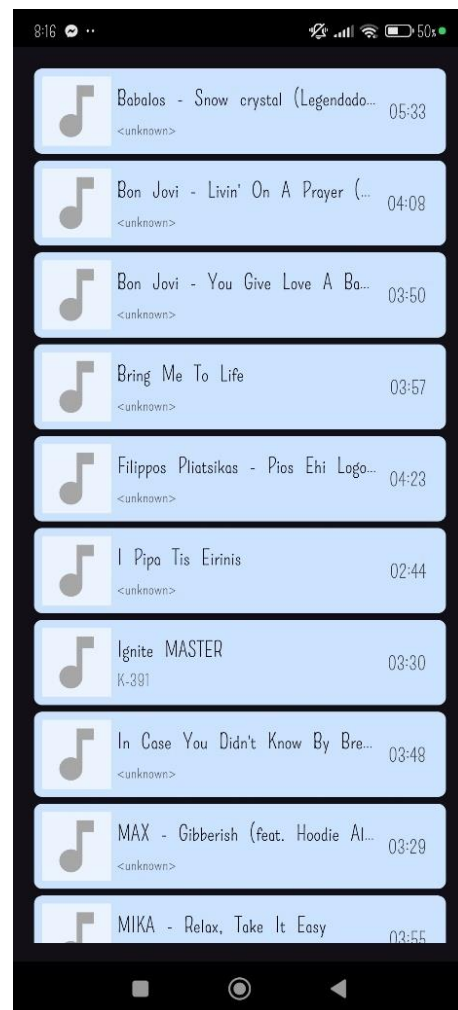
Εικόνα 6.17: MainActivity arePermissionsGranted()

## 6.4 Βιβλιοθήκη Τραγουδιών (Activity Library)

Η **ActivityLibrary** είναι η δραστηριότητα που αναλαμβάνει τη διαχείριση και την εμφάνιση της μουσικής βιβλιοθήκης της εφαρμογής. Συγκεκριμένα, φορτώνει τα τραγούδια της συσκευής μέσω του **MediaStore**, τα αποθηκεύει σε μια λίστα αντικειμένων τύπου **Song** και τα εμφανίζει στον χρήστη με τη βοήθεια ενός **RecyclerView**. Αν η λίστα είναι άδεια, εμφανίζεται σχετικό μήνυμα, ενώ κατά τη φόρτωση προβάλλεται ένα **ProgressBar**. Επιπλέον, όταν ο χρήστης επιλέγει κάποιο τραγούδι, η δραστηριότητα δημιουργεί μια λίστα με όλα τα κομμάτια και εκκινεί την **PlayerMode** activity, περνώντας τα απαραίτητα δεδομένα για την αναπαραγωγή. Με αυτόν τον τρόπο, η **ActivityLibrary** λειτουργεί ως γέφυρα ανάμεσα στη συλλογή τραγουδιών της συσκευής και τη λειτουργία του player.



Εικόνα 6.18: ActivityLibrary xml



Εικόνα 6.19: Λίστα Τραγουδιών display στο κινητό

### 6.4.1 XML

Το παρακάτω κομμάτι κώδικα αποτελεί το **UI layout** της οθόνης που εμφανίζει τη λίστα με τα τραγούδια. Περιλαμβάνει τρία βασικά στοιχεία: μια **ProgressBar** που εμφανίζεται προσωρινά κατά τη φόρτωση, μια **RecyclerView** για την προβολή των τραγουδιών σε λίστα, και μια **TextView** που εμφανίζει μήνυμα σε περίπτωση που δεν βρεθούν τραγούδια.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent">
7
8      <!-- Progress Bar shown while loading songs -->
9      <ProgressBar
10         android:id="@+id/progressBar"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:visibility="gone"
14         app:layout_constraintTop_toTopOf="parent"
15         app:layout_constraintBottom_toBottomOf="parent"
16         app:layout_constraintStart_toStartOf="parent"
17         app:layout_constraintEnd_toEndOf="parent" />
18
19     <!-- RecyclerView for displaying the list of songs -->
20     <androidx.recyclerview.widget.RecyclerView
21         android:id="@+id/recyclerView"
22         android:layout_width="0dp"
23         android:layout_height="0dp"
24         android:layout_margin="16dp"
25         app:layout_constraintTop_toTopOf="parent"
26         app:layout_constraintBottom_toBottomOf="parent"
27         app:layout_constraintStart_toStartOf="parent"
28         app:layout_constraintEnd_toEndOf="parent" />

```

Εικόνα 6.20: ActivityLibrary1 xml

```

30     <!-- TextView for an empty state message -->
31     <TextView
32         android:id="@+id/emptyStateText"
33         android:layout_width="wrap_content"
34         android:layout_height="wrap_content"
35         android:text="No songs found"
36         android:textSize="5sp"
37         android:textColor="#777"
38         android:visibility="gone"
39         app:layout_constraintTop_toTopOf="parent"
40         app:layout_constraintBottom_toBottomOf="parent"
41         app:layout_constraintStart_toStartOf="parent"
42         app:layout_constraintEnd_toEndOf="parent" />
43
44 </androidx.constraintlayout.widget.ConstraintLayout>

```

Εικόνα 6.21: ActivityLibrary2 xml

## 6.4.2 KOTLIN

**Imports** (γραμμές 3-12): Εισαγωγή απαραίτητων βιβλιοθηκών που χρειάζεται η κλάση για να λειτουργήσει.

```

1 package com.example.musicvibrationapp
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.provider.MediaStore
6 import android.view.View
7 import android.widget.ProgressBar
8 import android.widget.TextView
9 import androidx.appcompat.app.AppCompatActivity
10 import androidx.recyclerview.widget.LinearLayoutManager
11 import androidx.recyclerview.widget.RecyclerView
12 import android.database.Cursor

```

Εικόνα 6.22: ActivityLibrary imports

### Δήλωση Κλάσης και Μεταβλητών (γραμμές 14-18)

Οι μεταβλητές **recyclerView**, **songsAdapter** και **songs** αναπαριστούν αντίστοιχα το UI στοιχείο, τον adapter που χειρίζεται εμφάνιση και αλληλεπίδραση κάθε στοιχείου, και τη λίστα τραγουδιών από το MediaStore.

### onCreate() (γραμμές 20-34)

Αρχικοποιεί την ActivityLibrary, ορίζει το layout της οθόνης, συνδέει το **RecyclerView** με τον **SongsAdapter** και καλεί τη **loadSongs()** για να φορτωθούν τα τραγούδια και να εμφανιστούν δυναμικά στη λίστα.

```

14 class ActivityLibrary : AppCompatActivity() {
15
16     private lateinit var recyclerView: RecyclerView
17     private lateinit var songsAdapter: SongsAdapter
18     private val songs = mutableListOf<Song>()
19
20     override fun onCreate(savedInstanceState: Bundle?) {
21         super.onCreate(savedInstanceState)
22         setContentView(R.layout.activity_library)
23
24         recyclerView = findViewById(R.id.recyclerView)
25         recyclerView.layoutManager = LinearLayoutManager(context, this)
26
27         // Initialize adapter with click listener
28         songsAdapter = SongsAdapter(context, this, songs) { selectedSong ->
29             openPlayerMode(selectedSong)
30         }
31         recyclerView.adapter = songsAdapter
32
33         loadSongs()
34     }

```

Εικόνα 6.23: ActivityLibrary μεταβλητές και onCreate()

Η μέθοδος **loadSongs()** (γραμμές 38-86) ανακτά τα μουσικά αρχεία της συσκευής, δημιουργεί αντικείμενα **Song** και ενημερώνει τη λίστα της διεπαφής χρήστη, προετοιμάζοντας τα τραγούδια για απτική και οπτική αναπαράσταση.

```
38     private fun loadSongs() {
39         findViewById<ProgressBar>(R.id.progressBar).visibility = View.VISIBLE
40
41         val projection = arrayOf(
42             MediaStore.Audio.Media._ID,
43             MediaStore.Audio.Media.ARTIST,
44             MediaStore.Audio.Media.TITLE,
45             MediaStore.Audio.Media.DATA,
46             MediaStore.Audio.Media.DURATION
47         )
48
49         val cursor: Cursor? = contentResolver.query(
50             MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
51             projection,
52             selection: "${MediaStore.Audio.Media.IS_MUSIC} != 0",
53             selectionArgs: null,
54             sortOrder: MediaStore.Audio.Media.TITLE + " ASC"
55         )
56
57         songs.clear()
58         cursor?.use { it: Cursor
59             val idIndex = it.getColumnIndex(MediaStore.Audio.Media._ID)
60             val artistIndex = it.getColumnIndex(MediaStore.Audio.Media.ARTIST)
61             val titleIndex = it.getColumnIndex(MediaStore.Audio.Media.TITLE)
62             val pathIndex = it.getColumnIndex(MediaStore.Audio.Media.DATA)
63             val durationIndex = it.getColumnIndex(MediaStore.Audio.Media.DURATION)
64         }
```

Εικόνα 6.24: ActivityLibrary loadSongs() 1

```

65     while (it.moveToNext()) {
66         val id = it.getLong(idIndex)
67         val artist = it.getString(artistIndex) ?: "Unknown Artist"
68         val title = it.getString(titleIndex) ?: "Unknown Title"
69         val path = it.getString(pathIndex) ?: "Unknown Path"
70         val duration = it.getInt(durationIndex)
71
72         songs.add(Song(id, title, path, artist, duration))
73     }
74 }
75
76 findViewById<ProgressBar>(R.id.progressBar).visibility = View.GONE
77
78 if (songs.isEmpty()) {
79     findViewById<TextView>(R.id.emptyStateText).visibility = View.VISIBLE
80     recyclerView.visibility = View.GONE
81 } else {
82     findViewById<TextView>(R.id.emptyStateText).visibility = View.GONE
83     recyclerView.visibility = View.VISIBLE
84     songsAdapter.notifyDataSetChanged()
85 }
86 }
87

```

Εικόνα 6.25: ActivityLibrary loadSongs() 2

Η μέθοδος **openPlayerMode(song: Song)** (γραμμές 88-110) προετοιμάζει τα δεδομένα της λίστας τραγουδιών για την αναπαραγωγή. Συγκεκριμένα, δημιουργεί **λίστες** με τίτλους, καλλιτέχνες και διαδρομές των τραγουδιών και εντοπίζει τη θέση του επιλεγμένου τραγουδιού. Στη συνέχεια, ξεκινάει τη δραστηριότητα **PlayerMode**, μεταφέροντας όλα τα δεδομένα μέσω **Intent**, ώστε η εφαρμογή να μπορεί να αναπαράγει το επιλεγμένο τραγούδι και να παρέχει απτική και οπτική αναπαράσταση.

```

88     private fun openPlayerMode(song: Song) {
89         val songTitles = ArrayList<String>()
90         val songArtists = ArrayList<String>()
91         val songListPaths = ArrayList<String>()
92         var currentIndex = 0
93
94         for (i in songs.indices) {
95             songTitles.add(songs[i].title)
96             songArtists.add(songs[i].artist)
97             songListPaths.add(songs[i].path)
98             if (songs[i].path == song.path) {
99                 currentIndex = i
100             }
101         }
102
103         val intent = Intent(packageContext: this, PlayerMode::class.java).apply { this: Intent
104             putStringArrayListExtra(name: "SONG_TITLES", songTitles)
105             putStringArrayListExtra(name: "SONG_ARTISTS", songArtists)
106             putStringArrayListExtra(name: "SONG_LIST", songListPaths)
107             putExtra(name: "CURRENT_INDEX", currentIndex)
108         }
109         startActivity(intent)
110     }
111 }

```

Εικόνα 6.26: ActivityLibrary openPlayerMode()

### 6.4.3 SongsAdapter

Η **ActivityLibrary** αξιοποιεί την κλάση **SongsAdapter.kt**, η οποία είναι υπεύθυνη για τη σύνδεση των δεδομένων με το γραφικό περιβάλλον. Ο adapter χρησιμοποιεί το δικό του layout (**songs\_item\_layout.xml**) για την αναπαράσταση ενός τραγουδιού, προβάλλοντας τα αντίστοιχα στοιχεία (τίτλος, καλλιτέχνης κ.λπ.). Με αυτόν τον τρόπο, το **RecyclerView** επαναλαμβάνει αυτό το layout για κάθε εγγραφή της λίστας, προσαρμόζοντάς το δυναμικά στα δεδομένα του κάθε τραγουδιού. Έτσι εξασφαλίζεται μια αποδοτική και ευέλικτη παρουσίαση της μουσικής βιβλιοθήκης.

#### XML

- Το XML ορίζει το UI layout ενός τραγουδιού στο **RecyclerView**, με **CardView** για στρογγυλεμένες γωνίες και σκιά, **ImageView** για thumbnail, **LinearLayout** για τίτλο και καλλιτέχνη, και εμφάνιση διάρκειας στη δεξιά πλευρά.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.cardview.widget.CardView
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      android:layout_width="match_parent"
6      android:layout_height="wrap_content"
7      android:layout_marginHorizontal="2dp"
8      android:layout_marginVertical="3dp"
9      android:backgroundTint="#CCE3FF"
10     android:elevation="10dp"
11     app:cardCornerRadius="7dp">
12
13     <androidx.constraintlayout.widget.ConstraintLayout
14         android:layout_width="match_parent"
15         android:layout_height="match_parent"
16         android:padding="7dp">
17
18         <!-- Thumbnail Image on the left -->
19         <ImageView
20             android:id="@+id/album_thumbnail"
21             android:layout_width="60dp"
22             android:layout_height="60dp"
23             android:layout_marginEnd="10dp"
24             android:background="#EBF3FF"
25             android:src="@drawable/music_note"

```

Εικόνα 6.27: SongsAdapter xml

```

26         app:layout_constraintStart_toStartOf="parent"
27         app:layout_constraintTop_toTopOf="parent"
28         app:layout_constraintBottom_toBottomOf="parent" />
29
30         <!-- Title and Artist Information -->
31         <LinearLayout
32             android:layout_width="0dp"
33             android:layout_height="wrap_content"
34             android:orientation="vertical"
35             android:paddingEnd="10dp"
36             android:layout_marginStart="5dp"
37             app:layout_constraintStart_toEndOf="@id/album_thumbnail"
38             app:layout_constraintTop_toTopOf="parent"
39             app:layout_constraintBottom_toBottomOf="parent"
40             app:layout_constraintEnd_toStartOf="@id/songDuration">
41
42             <TextView
43                 android:id="@+id/songTitle"
44                 android:layout_width="wrap_content"
45                 android:layout_height="wrap_content"
46                 android:singleLine="true"
47                 android:text="Song Title"
48                 android:textColor="@color/black"
49                 android:textSize="18sp" />

```

Εικόνα 6.28: SongsAdapter2 xml

```

51     <TextView
52         android:id="@+id/artistName"
53         android:layout_width="wrap_content"
54         android:layout_height="wrap_content"
55         android:layout_marginTop="5dp"
56         android:text="Artist Name"
57         android:textColor="#5E5E5E"
58         android:textSize="14sp" />
59     </LinearLayout>
60
61     <!-- Duration on the right side -->
62     <TextView
63         android:id="@+id/songDuration"
64         android:layout_width="wrap_content"
65         android:layout_height="wrap_content"
66         android:text="3:45"
67         android:textSize="16sp"
68         android:textColor="#333"
69         app:layout_constraintEnd_toEndOf="parent"
70         app:layout_constraintTop_toTopOf="parent"
71         app:layout_constraintBottom_toBottomOf="parent"
72         android:paddingEnd="10dp" />
73
74 </androidx.constraintlayout.widget.ConstraintLayout>
75 </androidx.cardview.widget.CardView>

```

Εικόνα 6.29: SongsAdapter3.xml

## KOTLIN

**Imports** (γραμμές 3-7): Εισαγωγή απαραίτητων βιβλιοθηκών που χρειάζεται η κλάση για να λειτουργήσει.

```

1     package com.example.musictovibrationapp
2
3     import android.view.LayoutInflater
4     import android.view.View
5     import android.view.ViewGroup
6     import android.widget.TextView
7     import androidx.recyclerview.widget.RecyclerView

```

Εικόνα 6.30: SongsAdapter imports

Αρχικά, ορίζεται η κλάση **Song** (γραμμές 10-16), ένα **data class** που αναπαριστά κάθε τραγούδι με **id** (μοναδικός αναγνωριστικός δείκτης που δίνει το Android σε κάθε αρχείο μέσα στο **MediaStore**), **τίτλο**, **διαδρομή**, **διάρκεια** και **καλλιτέχνη**, και περιλαμβάνει τη μέθοδο **getFormattedDuration()** για μετατροπή της διάρκειας σε λεπτά και δευτερόλεπτα.

```

9      data class Song(
10         val id: Long,
11         val title: String,
12         val path: String,
13         val artist: String,
14         val duration: Int // Duration in milliseconds
15     ) {
16         fun getFormattedDuration(): String {
17             val minutes = (duration / 1000) / 60
18             val seconds = (duration / 1000) % 60
19             return String.format("%02d:%02d", minutes, seconds)
20         }
21     }

```

Εικόνα 6.31: SongsAdapter.kt data class

**SongsAdapter** (γραμμές 23-54)

Για κάθε τραγούδι, δημιουργεί και συνδέει το αντίστοιχο στοιχείο προβολής (**ViewHolder**), εμφανίζοντας τον τίτλο, τον καλλιτέχνη και τη διάρκεια. Επιπλέον, διαχειρίζεται τα πατήματα του χρήστη σε κάθε στοιχείο της λίστας μέσω μιας συνάρτησης **onItemClick**, ώστε να μπορεί να ανοίγει η λειτουργία αναπαραγωγής.

```

23     class SongsAdapter(
24         private val songs: List<Song>,
25         private val onItemClick: (Song) -> Unit // Lambda for handling item clicks
26     ) : RecyclerView.Adapter<SongsAdapter.SongsViewHolder>() {
27
28         override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): SongsViewHolder {
29             val view = LayoutInflater.from(parent.context).inflate(R.layout.song_item_layout,
30                 parent, attachToRoot: false)
31             return SongsViewHolder(view)
32         }
33
34         override fun onBindViewHolder(holder: SongsViewHolder, position: Int) {
35             val song = songs[position]
36
37             // Set the song title, artist, and formatted duration
38             holder.songTitle.text = song.title
39             holder.artistName.text = song.artist
40             holder.songDuration.text = song.getFormattedDuration()
41
42             // Trigger the click listener with the selected song
43             holder.itemView.setOnClickListener { it: View!
44                 onItemClick(song) // Pass the clicked song to the lambda
45             }
46         }
47     }

```

Εικόνα 6.32: SongsAdapter.kt

```

46 | override fun getItemCount(): Int = songs.size
47 |
48 | // ViewHolder class to hold views for each song item
49 | ZoZoTs99
50 | inner class SongsViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
51 |     val songTitle: TextView = itemView.findViewById(R.id.songTitle)
52 |     val artistName: TextView = itemView.findViewById(R.id.artistName)
53 |     val songDuration: TextView = itemView.findViewById(R.id.songDuration)
54 | }

```

Εικόνα 6.33: SongsAdapter.kt 2

## 6.5 Κατάσταση Αναπαραγωγής (Player Mode)

Η **PlayerMode** αποτελεί την οθόνη και την κεντρική λειτουργικότητα του **media player**, γίνεται η αναπαραγωγή τραγουδιών συνδυάζοντας ήχο, οπτικοποίηση και δονήσεις.

### 6.5.1 XML

Το layout του **PlayerMode** ορίζει την οθόνη αναπαραγωγής μουσικής, περιλαμβάνοντας φόντο, εικόνα, **custom view** για την **οπτικοποίηση** του ήχου, TextView για τίτλο και καλλιτέχνη, TextView για χρόνο έναρξης και λήξης, ένα **SeekBar** για την πρόοδο αναπαραγωγής, βασικά κουμπιά ελέγχου (προηγούμενο, play/pause, επόμενο) και τέλος κουμπί επιστροφής.

```

1 | <?xml version="1.0" encoding="utf-8"?>
2 | <androidx.constraintlayout.widget.ConstraintLayout
3 |     xmlns:android="http://schemas.android.com/apk/res/android"
4 |     xmlns:app="http://schemas.android.com/apk/res-auto"
5 |     xmlns:tools="http://schemas.android.com/tools"
6 |     android:layout_width="match_parent"
7 |     android:layout_height="match_parent"
8 |     android:background="@drawable/background"
9 |     tools:context=".PlayerMode">
10 |
11 |
12 |     <ImageView
13 |         android:id="@+id/album_art"
14 |         android:layout_width="400dp"
15 |         android:layout_height="300dp"
16 |         android:contentDescription="Album"
17 |         android:src="@drawable/music_note"
18 |         app:layout_constraintBottom_toBottomOf="parent"
19 |         app:layout_constraintEnd_toEndOf="parent"
20 |         app:layout_constraintHorizontal_bias="0.454"
21 |         app:layout_constraintStart_toStartOf="parent"
22 |         app:layout_constraintTop_toTopOf="parent"
23 |         app:layout_constraintVertical_bias="0.252" />

```

Εικόνα 6.34: PlayerMode xml 1

```

25 <com.example.musicvibrationapp.VisualizerView
26     android:id="@+id/VisualizerView"
27     android:layout_width="wrap_content"
28     android:layout_height="100dp"
29     android:background="#E4113B68"
30     app:layout_constraintBottom_toTopOf="@+id/songTitle"
31     app:layout_constraintEnd_toEndOf="parent"
32     app:layout_constraintHorizontal_bias="0.0"
33     app:layout_constraintStart_toStartOf="parent"
34     app:layout_constraintTop_toBottomOf="@+id/album_art" />
35
36 <ImageButton
37     android:id="@+id/back"
38     android:layout_width="30dp"
39     android:layout_height="30dp"
40     android:background="@drawable/back_button"
41     app:layout_constraintBottom_toBottomOf="parent"
42     app:layout_constraintEnd_toEndOf="parent"
43     app:layout_constraintHorizontal_bias="0.05"
44     app:layout_constraintStart_toStartOf="parent"
45     app:layout_constraintTop_toTopOf="parent"
46     app:layout_constraintVertical_bias="0.05" />

```

Εικόνα 6.35: PlayerMode xml 2

```

48 <!-- Song Title TextView -->
49 <TextView
50     android:id="@+id/songTitle"
51     android:layout_width="match_parent"
52     android:layout_height="40dp"
53     android:layout_marginHorizontal="45dp"
54     android:ellipsize="marquee"
55     android:focusable="true"
56     android:focusableInTouchMode="true"
57     android:marqueeRepeatLimit="10"
58     android:singleLine="true"
59     android:textAlignment="center"
60     android:textSize="25sp"
61     android:textColor="#ffffff"
62     android:text="Song Title"
63     app:layout_constraintBottom_toBottomOf="parent"
64     app:layout_constraintStart_toStartOf="parent"
65     app:layout_constraintEnd_toEndOf="parent"
66     app:layout_constraintTop_toTopOf="parent"
67     app:layout_constraintVertical_bias="0.75"
68     android:gravity="center"
69     android:scrollHorizontally="true" />

```

Εικόνα 6.36: PlayerMode xml 3

```

71 <!-- Song Artist TextView -->
72 <TextView
73     android:id="@+id/songArtist"
74     android:layout_width="match_parent"
75     android:layout_height="30dp"
76     android:layout_marginHorizontal="45dp"
77     android:layout_marginTop="5dp"
78     android:textSize="18sp"
79     android:textColor="#ffffff7"
80     android:text="Artist Name"
81     app:layout_constraintTop_toBottomOf="@id/songTitle"
82     app:layout_constraintEnd_toEndOf="parent"
83     app:layout_constraintStart_toStartOf="parent"
84     app:layout_constraintVertical_bias=".6"
85     android:gravity="center" />

```

Εικόνα 6.37: PlayerMode xml 4

```

87 <RelativeLayout
88     android:id="@+id/player"
89     android:layout_width="match_parent"
90     android:layout_height="wrap_content"
91     android:paddingHorizontal="5dp"
92     app:layout_constraintBottom_toBottomOf="parent"
93     app:layout_constraintTop_toTopOf="parent"
94     app:layout_constraintVertical_bias=".85">
95
96     <TextView
97         android:id="@+id/start_time"
98         android:layout_width="wrap_content"
99         android:layout_height="wrap_content"
100        android:layout_alignParentStart="true"
101        android:layout_marginStart="5dp"
102        android:text="--:--"
103        android:textColor="@color/white"
104        android:textSize="10dp" />

```

Εικόνα 6.38: PlayerMode xml 5

```

106 <TextView
107     android:id="@+id/end_time"
108     android:layout_width="wrap_content"
109     android:layout_height="wrap_content"
110     android:layout_alignParentEnd="true"
111     android:layout_marginEnd="5dp"
112     android:text="--:--"
113     android:textColor="@color/white"
114     android:textSize="10dp" />
115
116 <SeekBar
117     android:id="@+id/seekBar"
118     android:layout_width="match_parent"
119     android:layout_height="35dp"
120     android:backgroundTint="@color/white" />
121
122 </RelativeLayout>

```

Εικόνα 6.39: PlayerMode xml 6

```

124 <!-- Previous Button -->
125 <ImageButton
126     android:id="@+id/previous"
127     android:layout_width="50dp"
128     android:layout_height="50dp"
129     android:background="@android:color/transparent"
130     app:srcCompat="@drawable/previous"
131     android:clickable="true"
132     android:focusable="true"
133     android:scaleType="fitCenter"
134     android:contentDescription="previous_button"
135     app:layout_constraintBottom_toBottomOf="parent"
136     app:layout_constraintEnd_toStartOf="@id/play_pause"
137     android:layout_marginEnd="20dp"
138     android:layout_marginBottom="25dp" />

```

Εικόνα 6.40: PlayerMode xml 7

```

140     <ImageButton
141         android:id="@+id/play_pause"
142         android:layout_width="70dp"
143         android:layout_height="70dp"
144         android:background="@android:color/transparent"
145         app:srcCompat="@drawable/play"
146         android:clickable="true"
147         android:focusable="true"
148         android:scaleType="fitCenter"
149         android:contentDescription="Play/Pause Button"
150         app:layout_constraintBottom_toBottomOf="parent"
151         app:layout_constraintStart_toStartOf="parent"
152         app:layout_constraintEnd_toEndOf="parent"
153         android:layout_marginBottom="20dp" />

```

Εικόνα 6.41: PlayerMode xml 8

```

155     <!-- Next Button -->
156     <ImageButton
157         android:id="@+id/next"
158         android:layout_width="50dp"
159         android:layout_height="50dp"
160         android:background="@android:color/transparent"
161         app:srcCompat="@drawable/next"
162         android:clickable="true"
163         android:focusable="true"
164         android:scaleType="fitCenter"
165         android:contentDescription="next_button"
166         app:layout_constraintBottom_toBottomOf="parent"
167         app:layout_constraintStart_toEndOf="@id/play_pause"
168         android:layout_marginStart="20dp"
169         android:layout_marginBottom="25dp" />
170 </androidx.constraintlayout.widget.ConstraintLayout>

```

Εικόνα 6.42: PlayerMode xml 9

## 6.5.2 KOTLIN

Η κλάση **PlayerMode** υλοποιεί την κύρια λειτουργικότητα του media player της εφαρμογής. Κατά την αναπαραγωγή ενός τραγουδιού ενεργοποιείται ταυτόχρονα η **ηχητική**, **οπτική** και **απτική αναπαράσταση** του ρυθμού.

**Imports** (γραμμές 3-11): Εισαγωγή απαραίτητων βιβλιοθηκών που χρειάζεται η κλάση για να λειτουργήσει.

```

1 package com.example.musicvibrationapp
2
3 import android.media.MediaPlayer
4 import android.os.Bundle
5 import android.os.Handler
6 import android.os.Looper
7 import android.widget.ImageButton
8 import android.widget.SeekBar
9 import android.widget.TextView
10 import androidx.appcompat.app.AppCompatActivity
11 import android.net.Uri

```

Εικόνα 6.43: PlayerMode imports

### Μεταβλητές (γραμμές 15-37)

Περιλαμβάνουν τον `mediaPlayer` για αναπαραγωγή, `rhythmVibrator` και `visualizerView` για δονήσεις και οπτικοποίηση ήχου, τα κουμπιά `playPauseButton`, `nextButton`, `prevButton`, `TextView` για πληροφορίες τραγουδιού, `seekBar` για πρόοδο, `songUri` για το τρέχον τραγούδι (αποθηκεύει το URI-διεύθυνση), λίστες και δείκτη θέσης τραγουδιών, καθώς και `isPlaying` και `handler` για παρακολούθηση κατάστασης και ενημέρωση UI.

```

14 class PlayerMode : AppCompatActivity() {
15     private lateinit var mediaPlayer: MediaPlayer
16     private lateinit var rhythmVibrator: RhythmVibrator
17     private lateinit var visualizerView: VisualizerView
18
19     private lateinit var playPauseButton: ImageButton
20     private lateinit var nextButton: ImageButton
21     private lateinit var prevButton: ImageButton
22
23     private lateinit var startTimeTextView: TextView
24     private lateinit var endTimeTextView: TextView
25     private lateinit var songTitleTextView: TextView
26     private lateinit var songArtistTextView: TextView
27
28     private lateinit var seekBar: SeekBar
29     private var songUri: String? = null
30     private var songList: ArrayList<String>? = null
31     private var songTitles = ArrayList<String>()
32     private var songArtists = ArrayList<String>()
33     private var currentIndex = 0
34     private var isPlaying = true // To track play/pause state
35
36     // Use MainLooper for UI thread
37     private val handler = Handler(Looper.getMainLooper())

```

Εικόνα 6.44: PlayerMode μεταβλητές

Η μέθοδος **onCreate** (γραμμές 39-107) της **PlayerMode** αρχικοποιεί την **Activity**, φορτώνει το **layout**, συνδέει τα **UI** στοιχεία με τις **μεταβλητές**, ρυθμίζει τους **listeners** για τα κουμπιά και τον **VisualizerView**, ξεκινά την αναπαραγωγή του τρέχοντος τραγουδιού και ενημερώνει το **SeekBar**. Κατά τη διαδικασία καλεί τις βασικές μεθόδους του **player**: **playSong()** για έναρξη αναπαραγωγής, **handlePlayPauseAction()** και **updatePlayPauseButton()** για διαχείριση play/pause, **playNextSong()** / **playPreviousSong()** για μετάβαση στα τραγούδια, **updateSongInfo()** για ενημέρωση τίτλου και καλλιτέχνη, και **updateStartTimeTextView()** / **startUpdatingTime()** για ενημέρωση και περιοδική ανανέωση της προόδου αναπαραγωγής.

```

39 override fun onCreate(savedInstanceState: Bundle?) {
40     super.onCreate(savedInstanceState)
41     setContentView(R.layout.activity_player_mode)
42     // Retrieve song data from intent
43     songList = intent.getStringArrayListExtra( name: "SONG_LIST")
44     songTitles = intent.getStringArrayListExtra( name: "SONG_TITLES") ?: arrayListOf()
45     songArtists = intent.getStringArrayListExtra( name: "SONG_ARTISTS") ?: arrayListOf()
46     currentIndex = intent.getIntExtra( name: "CURRENT_INDEX", defaultValue: 0)
47
48     if (songList.isNullOrEmpty() || currentIndex !in songList!!.indices) {
49         println("Error: Invalid song list or index!")
50         finish()
51         return
52     }
53     // Initialize UI elements
54     songTitleTextView = findViewById(R.id.songTitle)
55     songArtistTextView = findViewById(R.id.songArtist)
56     startTimeTextView = findViewById(R.id.start_time)
57     endTimeTextView = findViewById(R.id.end_time)
58     seekBar = findViewById(R.id.seekBar)
59     playPauseButton = findViewById(R.id.play_pause)
60     nextButton = findViewById(R.id.next)
61     prevButton = findViewById(R.id.previous)
62     visualizerView = findViewById(R.id.VisualizerView)

```

Εικόνα 6.45: PlayerMode onCreate() 1

Λειτουργία **backButton** (γραμμές 64-67), κλείνει την **Activity PlayerMode** και επιστρέφει στην προηγούμενη οθόνη.

Ο **visualizerView.setOnBeatListener** (γραμμή 77), ανιχνεύει τα **beats** του τραγουδιού σε πραγματικό χρόνο.

- **intensity** → η ένταση του **beat**.
- **frequencyBand** → η συχνότητα ή το φάσμα του **beat**, δηλαδή σε ποιο εύρος συχνοτήτων εμφανίζεται η ένταση.

Κάθε φορά που ανιχνεύεται ένα **beat**, καλείται η μέθοδος **rhythmVibrator.vibrateToBeat(intensity, frequencyBand)** (γραμμή 78) για να δημιουργηθούν δονήσεις με βάση ένταση και συχνότητα. Η ένταση (**intensity**) και η συχνότητα (**frequencyBand**) του **beat** χρησιμοποιούνται για να δημιουργηθούν αντίστοιχες δονήσεις. Αρχικοποιεί το αντικείμενο **RhythmVibrator** (γραμμή 83), ώστε να μπορεί να μετατρέπει τα **beats** σε δονήσεις κατά την αναπαραγωγή.

```

64     val backButton = findViewById<ImageButton>(R.id.back)
65     backButton.setOnClickListener { it: View!
66         finish() // Closes PlayerMode and returns to ActivityLibrary
67     }
68
69     updateSongInfo() // Update song title and artist
70     updatePlayPauseButton() // Set initial button state to play
71     playPauseButton.setOnClickListener { it: View!
72         isPlaying = !isPlaying // Toggle play/pause state
73         handlePlayPauseAction()
74         updatePlayPauseButton() // Update the button image based on state
75     }
76     // Set the listener to detect beats and trigger vibrations
77     visualizerView.setOnBeatListener {intensity, frequencyBand ->
78         rhythmVibrator.vibrateToBeat(intensity, frequencyBand)
79     }
80
81     nextButton.setOnClickListener { playNextSong() }
82     prevButton.setOnClickListener { playPreviousSong() }
83     rhythmVibrator = RhythmVibrator( context: this)
84     playSong()

```

Εικόνα 6.46: PlayerMode onCreate() 2

```

86     // Set SeekBar listener to update song position
87     seekBar.setOnSeekBarChangeListener(object : SeekBar.OnSeekBarChangeListener {
88         override fun onProgressChanged(seekBar: SeekBar?, progress: Int, fromUser: Boolean) {
89             if (fromUser) {
90                 mediaPlayer.seekTo(progress)
91                 updateStartTimeTextView(progress)
92             }
93         }
94
95         override fun onStartTrackingTouch(seekBar: SeekBar?) {
96             // Stop updating time while the user is dragging the seek bar
97             handler.removeCallbacksAndMessages( token: null)
98         }
99
100        override fun onStopTrackingTouch(seekBar: SeekBar?) {
101            if (isPlaying) {
102                // Resume the time update once the user stops dragging the seek bar
103                startUpdatingTime()
104            }
105        }
106    })
107 }

```

Εικόνα 6.47: PlayerMode onCreate() 3

Η `playSong()` (γραμμές 108-140), ξεκινά την αναπαραγωγή του τρέχοντος τραγουδιού, αφού ελέγξει ότι η λίστα τραγουδιών και ο δείκτης είναι έγκυροι. Ορίζει το **MediaPlayer**, ξεκινά την

αναπαραγωγή, ενημερώνει το **seekBar** και τα **TextView** με τους χρόνους, ενημερώνει την κατάσταση αναπαραγωγής και την εικόνα του κουμπιού play/pause.

Τέλος συνδέει το **VisualizerView** με τον MediaPlayer (γραμμή 136), χρησιμοποιώντας το **audio session ID** του. Αυτό επιτρέπει στο VisualizerView να **ανιχνεύει τα ηχητικά δεδομένα** του τραγουδιού σε πραγματικό χρόνο. Με βάση αυτά τα δεδομένα, μπορεί να εμφανίζει **οπτικές αναπαραστάσεις του ήχου** και να ανιχνεύει **beats**, που στη συνέχεια ενεργοποιούν το RhythmVibrator.

```

108     private fun playSong() {
109         val safeSongList = songList ?: return // Avoid null issues
110         if (currentIndex !in safeSongList.indices) {
111             println("Error: Invalid song index!")
112             return
113         }
114
115         val uriString = safeSongList[currentIndex]
116         songUri = uriString
117         val uri = Uri.parse(uriString)
118         // If mediaPlayer exists, reset it instead of creating a new one
119         if (::mediaPlayer.isInitialized) {
120             // Instead of stopping/releasing, reset it for reuse
121             mediaPlayer.reset()
122         } else {
123             mediaPlayer = MediaPlayer()
124         }

```

Εικόνα 6.48: PlayerMode playSong() 1

```

126         // Setup the media player
127         mediaPlayer.apply { this: MediaPlayer
128             setDataSource(applicationContext, uri)
129             prepare()
130             start()
131             seekBar.max = duration
132             updateEndTimeTextView(duration)
133             updateStartTimeTextView( currentTime: 0)
134             startUpdatingTime()
135         }
136         visualizerView.linkToMediaPlayer(mediaPlayer.audioSessionId)
137
138         isPlaying = true // Update play state
139         updatePlayPauseButton()
140     }

```

Εικόνα 6.49: PlayerMode playSong() 2

Η μέθοδος **updateSongInfo()** (γραμμές 142-147), ενημερώνει τα **TextView** της οθόνης με τον τίτλο και τον καλλιτέχνη του τρέχοντος τραγουδιού.

```

142     private fun updateSongInfo() {
143         // Update song title
144         songTitleTextView.text = songTitles[currentIndex]
145         // Update artist name
146         songArtistTextView.text = songArtists[currentIndex]
147     }

```

Εικόνα 6.50: PlayerMode updateSongInfo()

Η μέθοδος **playNextSong()** (γραμμές 148-157), μεταβαίνει στο επόμενο τραγούδι της λίστας (αν υπάρχει), καλεί τη **playSong()** για να ξεκινήσει η αναπαραγωγή του και ενημερώνει τα **TextView** στο UI μέσω της **updateSongInfo()**.

```

148     private fun playNextSong() {
149         songList?.let { list ->
150             if (currentIndex < list.size - 1) {
151                 currentIndex++
152                 playSong()
153                 // Update UI after switching songs
154                 updateSongInfo()
155             }
156         }
157     }

```

Εικόνα 6.51: PlayerMode playNextSong()

Παρόμοια η **playPreviousSong()** (γραμμές 158-167), επιστρέφει στο προηγούμενο τραγούδι (αν υπάρχει), καλεί τη **playSong()** για να ξεκινήσει η αναπαραγωγή του και ενημερώνει τα **TextView** στο UI μέσω της **updateSongInfo()**.

```

158     private fun playPreviousSong() {
159         songList?.let { list ->
160             if (currentIndex > 0) {
161                 currentIndex--
162                 playSong()
163                 // Update UI after switching songs
164                 updateSongInfo()
165             }
166         }
167     }

```

Εικόνα 6.52: PlayerMode playPreviousSong()

Η μέθοδος **updatePlayPauseButton()** (γραμμές 169-175), ενημερώνει την εικόνα του κουμπιού play/pause ώστε να αντανακλά την τρέχουσα κατάσταση αναπαραγωγής: pause όταν παίζει μουσική και play όταν είναι σε παύση.

```

169     private fun updatePlayPauseButton() {
170         if (isPlaying) {
171             playPauseButton.setImageResource(R.drawable.pause)
172         } else {
173             playPauseButton.setImageResource(R.drawable.play)
174         }
175     }

```

Εικόνα 6.53: PlayerMode updatePlayPauseButton()

Η μέθοδος **handlePlayPauseAction()** (γραμμές 177-196), ελέγχει την κατάσταση αναπαραγωγής: αν το τραγούδι παίζει, το διακόπτει, σταματάει τις δονήσεις και την ενημέρωση του UI, αν δεν παίζει, ξεκινά την αναπαραγωγή και ενημερώνει τον χρόνο και το κουμπί play/pause.

```

177     private fun handlePlayPauseAction() {
178         // Prevent crashes if MediaPlayer is uninitialized
179         if (!::mediaPlayer.isInitialized) return
180         if (mediaPlayer.isPlaying) {
181             mediaPlayer.pause()
182             isPlaying = false
183             rhythmVibrator.stopVibration()
184             mediaPlayer.setOnCompleteListener { it: MediaPlayer!
185                 // Stop the visualizer when music finishes
186                 visualizerView.release()
187             }
188             // Stop seek bar updates
189             handler.removeCallbacksAndMessages(token: null)
190         } else {
191             mediaPlayer.start()
192             isPlaying = true
193             startUpdatingTime()
194         }
195         updatePlayPauseButton() // Update UI
196     }

```

Εικόνα 6.54: PlayerMode handlePlayPauseAction()

Η **updateStartTimeTextView(currentTime: Int)** (γραμμές 198-202) υπολογίζει λεπτά και δευτερόλεπτα από την τρέχουσα θέση του τραγουδιού (**currentTime**) και εμφανίζει την τιμή στο **startTimeTextView**.

```

198     private fun updateTimeTextView(currentTime: Int) {
199         val minutes = currentTime / 1000 / 60
200         val seconds = currentTime / 1000 % 60
201         startTimeTextView.text = String.format("%02d:%02d", minutes, seconds)
202     }

```

Εικόνα 6.55: PlayerMode updateTimeTextView()

Η **updateEndTimeTextView(duration: Int)** (γραμμές 203-207) μετατρέπει τη συνολική διάρκεια του τραγουδιού (**duration**) σε λεπτά και δευτερόλεπτα και την εμφανίζει στο **endTimeTextView**.

```

203     private fun updateEndTimeTextView(duration: Int) {
204         val minutes = duration / 1000 / 60
205         val seconds = duration / 1000 % 60
206         endTimeTextView.text = String.format("%02d:%02d", minutes, seconds)
207     }

```

Εικόνα 6.56: PlayerMode updateEndTimeTextView()

Η **startUpdatingTime()** (γραμμές 208-219) χρησιμοποιεί τον **handler** για να εκτελείται περιοδικά κάθε δευτερόλεπτο. Όσο το τραγούδι παίζει, ενημερώνει τον τρέχοντα χρόνο με τη **updateStartTimeTextView()** και προσαρμόζει την πρόοδο της **seekBar**.

```

208     private fun startUpdatingTime() {
209         handler.postDelayed(object : Runnable {
210             override fun run() {
211                 if (mediaPlayer.isPlaying) {
212                     val currentPosition = mediaPlayer.currentPosition
213                     updateTimeTextView(currentPosition) // Update the start time
214                     seekBar.progress = currentPosition // Update the seek bar progress
215                     handler.postDelayed(this, delayMillis: 1000) // Continue updating every second
216                 }
217             }
218         }, delayMillis: 1000)
219     }

```

Εικόνα 6.57: PlayerMode startUpdatingTime()

Η μέθοδος **onPause()** (γραμμές 221-230) εκτελείται όταν η δραστηριότητα μπαίνει σε παύση, σταματά την αναπαραγωγή αν ο **mediaPlayer** παίζει, αλλάζει το κουμπί σε εικονίδιο **play** αν η μουσική ήταν ενεργή, και καλεί την **stopVibration()** όπου διακόπτει τις δονήσεις του **rhythmVibrator**.

```

221  override fun onPause() {
222      super.onPause()
223      if (::mediaPlayer.isInitialized && mediaPlayer.isPlaying) {
224          mediaPlayer.pause()
225      }
226      if (isPlaying) {
227          playPauseButton.setImageResource(R.drawable.play) // Set to play icon
228      }
229      rhythmVibrator.stopVibration()
230  }

```

Εικόνα 6.58: PlayerMode onPause()

Η μέθοδος **onDestroy()** (γραμμές 231-242) εκτελείται όταν καταστρέφεται η δραστηριότητα, αποδεσμεύει τους πόρους του **mediaPlayer**, καλεί την **stopVibration()** που σταματά τις δονήσεις του **rhythmVibrator**, καλεί την **release()** που ελευθερώνει το **visualizerView** και αφαιρεί όλα τα προγραμματισμένα callbacks του **handler** ώστε να μην συνεχίζονται ενημερώσεις στο UI.

```

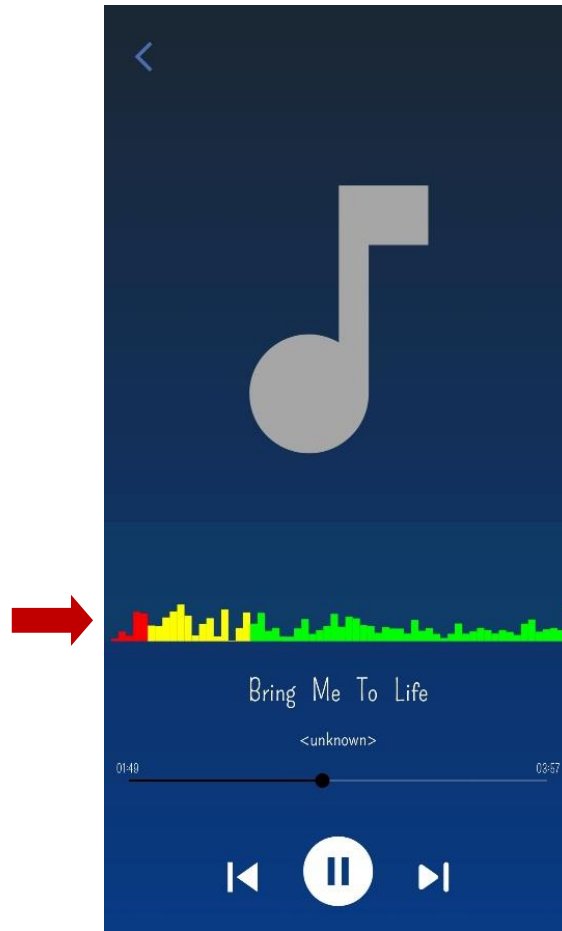
231  override fun onDestroy() {
232      super.onDestroy()
233      // Release MediaPlayer resources when activity is destroyed
234      if (::mediaPlayer.isInitialized) {
235          mediaPlayer.release()
236      }
237      rhythmVibrator.stopVibration()
238      visualizerView.release()
239      // Remove all handler callbacks for UI updates
240      handler.removeCallbacksAndMessages(token: null)
241  }
242  }

```

Εικόνα 6.59: PlayerMode onDestroy()

## 6.6 Οπτικοποίηση Μουσικής (Visualizer View)

Σε αυτό το κεφάλαιο θα εξηγηθεί πως έχει χρησιμοποιηθεί ένας visualizer για να γίνει καλύτερη η απτική ανάδραση.



Εικόνα 6.60: visualizer

## KOTLIN

**Imports** (γραμμές 3-12): Εισαγωγή απαραίτητων βιβλιοθηκών που χρειάζεται η κλάση για να λειτουργήσει.

```

1 package com.example.musictovibrationapp
2
3 import android.content.Context
4 import android.graphics.Canvas
5 import android.graphics.Color
6 import android.graphics.Paint
7 import android.media.audiofx.Visualizer
8 import android.util.AttributeSet
9 import android.view.View
10 import kotlin.math.log10
11 import kotlin.math.min
12 import kotlin.math.sqrt

```

Εικόνα 6.61: Visualizer imports

**Μεταβλητές** (γραμμές 16-24)

- **visualizer** (γραμμή 16): το αντικείμενο του Android **Visualizer API** που «ακούει» τον ήχο από το MediaPlayer και παρέχει τα δεδομένα **FFT** για ανάλυση συχνοτήτων.

- **Paint** (γραμμή 17): εργαλείο σχεδίασης που χρησιμοποιείται στην **onDraw()** για να αποδίδει τις μπάρες του φάσματος με διαφορετικά χρώματα.
- **fftBytes** (γραμμή 18): πίνακας δεδομένων **FFT** που κρατά τις τιμές συχνοτήτων από το ηχητικό σήμα σε πραγματικό χρόνο.
- **onBeatDetected** (γραμμή 19): callback συνάρτηση που καλείται όταν εντοπιστεί beat, ενημερώνει άλλες λειτουργίες της εφαρμογής (π.χ. απτική ανάδραση).
- **rhythmVibrator** (γραμμή 20): custom αντικείμενο που μεταφράζει τα beats σε δονήσεις, ενισχύοντας την απτική εμπειρία.
- **lowMagnitudes**, **midMagnitudes**, **highMagnitudes** (γραμμές 22-24): λίστες που αποθηκεύουν τις τιμές έντασης χαμηλών, μεσαίων και υψηλών συχνοτήτων αντίστοιχα, ώστε να γίνεται εξομάλυνση (με κινούμενο μέσο όρο) και να αποφεύγονται θόρυβοι στις μετρήσεις.

```

14 class VisualizerView(context: Context, attrs: AttributeSet) : View(context, attrs) {
15
16     private var visualizer: Visualizer? = null
17     private val paint = Paint()
18     private var fftBytes: ByteArray? = null
19     private var onBeatDetected: ((Int, String) -> Unit)? = null // Callback for beats
20     private val rhythmVibrator = RhythmVibrator(context)
21     // Store previous magnitudes for smoothing (low, mid, and high frequencies)
22     private var lowMagnitudes = mutableListOf<Float>()
23     private var midMagnitudes = mutableListOf<Float>()
24     private var highMagnitudes = mutableListOf<Float>()

```

Εικόνα 6.62: Visualizer μεταβλητές

Η μέθοδος **setOnBeatListener()** (γραμμές 27-29) ορίζει ένα **callback**, δηλαδή μια συνάρτηση που εκτελείται αργότερα ως απάντηση σε ένα γεγονός, το οποίο καλείται κάθε φορά που ανιχνεύεται ένα beat και δίνει την ένταση και τη ζώνη συχνοτήτων του, επιτρέποντας στην εφαρμογή να αντιδράσει κατάλληλα.

```

27 fun setOnBeatListener(listener: (Int, String) -> Unit) {
28     this.onBeatDetected = listener
29 }

```

Εικόνα 6.63: Visualizer setOnBeatListener()

Η μέθοδος **linkToMediaPlayer()** συνδέει το **VisualizerView** με το session του **MediaPlayer** (γραμμή 33), ώστε να αρχίσει να λαμβάνει τα δεδομένα ήχου σε πραγματικό χρόνο. Ορίζει το μέγεθος των δεδομένων που θα συλλέγονται (captureSize) και εγκαθιστά έναν **DataCaptureListener** (γραμμή 34). Ο **listener** (γραμμές 35-44) παρακολουθεί δύο τύπους δεδομένων:

1. **Waveform Data:** η κυματομορφή του ήχου (δεν χρησιμοποιείται στην παρούσα υλοποίηση).
2. **FFT Data** (γραμμές 39-43): καλείται κάθε φορά που ο **Visualizer** συλλέγει δεδομένα FFT από τον ήχο. Αποθηκεύει τα δεδομένα FFT στο **fftBytes** για χρήση στην οπτική αναπαράσταση (γραμμή 40). Καλεί τη μέθοδο **detectBeats()**, η οποία αναλύει τα δεδομένα για να εντοπίσει beats και ρυθμικά μοτίβα (γραμμή 41). Καλεί το **invalidate()**, αναγκάζοντας το **VisualizerView** να ανανεώσει την εμφάνιση του και να δείξει σε πραγματικό χρόνο τη νέα

οπτικοποίηση του ήχου (γραμμή 42). Τέλος ρυθμίζει τον Visualizer να συλλέγει δεδομένα ήχου με τη μέγιστη δυνατή συχνότητα δειγματοληψίας (**Visualizer.getMaxCaptureRate()**). Οι παράμετροι `false`, `true` σημαίνουν ότι δεν συλλέγεται waveform αλλά μόνο FFT data (γραμμή 44).

Ενεργοποιεί τον Visualizer, ώστε να αρχίσει η συλλογή και η ανάλυση των δεδομένων σε πραγματικό χρόνο (γραμμή 45).

Με αυτόν τον τρόπο, η μέθοδος συνδέει την ανάλυση του ήχου με την οπτική και απτική αναπαράσταση του ρυθμού.

```

31 fun linkToMediaPlayer(mediaPlayerSessionId: Int) {
32     release()
33     visualizer = Visualizer(mediaPlayerSessionId).apply { this: Visualizer
34         captureSize = Visualizer.getCaptureSizeRange()[1]
35         setDataCaptureListener(object : Visualizer.OnDataCaptureListener {
36             override fun onWaveFormDataCapture(visualizer: Visualizer?, waveform: ByteArray?,
37                 samplingRate: Int) {}
38
39             override fun onFftDataCapture(visualizer: Visualizer?, fft: ByteArray?, samplingRate: Int) {
40                 fftBytes = fft //Save the FFT data for drawing
41                 detectBeats(fft) //Function to analyze beat patterns
42                 invalidate() //Forces the view to redraw
43             }
44         }, Visualizer.getMaxCaptureRate(), waveform: false, fft: true)
45         enabled = true //turns the visualizer on so it starts capturing
46     }
47 }

```

Εικόνα 6.64: Visualizer linkToMediaPlayer()

Η μέθοδος **detectBeats()** (γραμμές 50-115) έχει ως στόχο την ανίχνευση ρυθμικών μοτίβων (beats) από τα δεδομένα ήχου που συλλέγονται σε πραγματικό χρόνο από τον **Visualizer**. Χρησιμοποιεί τις αρχές της ψηφιακής επεξεργασίας σήματος (**DSP**), συγκεκριμένα την **ανάλυση φάσματος μέσω FFT**.

Η μέθοδος δέχεται ως παράμετρο τον πίνακα FFT (γραμμή 50). Ο έλεγχος στη γραμμή 51 διασφαλίζει ότι η ανάλυση γίνεται μόνο αν υπάρχουν δεδομένα.

Στη συνέχεια, το φάσμα χωρίζεται σε τρεις ζώνες συχνότητας:

- **Low (χαμηλές συχνότητες):** 2–6
- **Mid (μεσαίες συχνότητες):** 6–20
- **High (υψηλές συχνότητες):** 20–40

Αυτή η κατηγοριοποίηση βασίζεται στη φυσιολογική αντιληπτικότητα του ήχου: οι ρυθμικές και μπάσες νότες συνήθως βρίσκονται στις χαμηλές συχνότητες, τα περισσότερα μουσικά όργανα καλύπτουν τις μεσαίες, ενώ τα υψηλά φωνητικά ή τριγωνικές νότες κατατάσσονται στις υψηλές.

```

50     private fun detectBeats(fft: ByteArray?) {
51         fft?.let { bytes ->
52             // Frequency bands for low, mid, and high
53             val lowBandStart = 2
54             val lowBandEnd = 6 // For bass frequencies (lower frequencies)
55             val midBandStart = 6
56             val midBandEnd = 20 // For mid frequencies
57             val highBandStart = 20
58             val highBandEnd = 40 // For high frequencies (higher frequencies)

```

Εικόνα 6.65: Visualizer detectBeats() 1

Για κάθε ζώνη υπολογίζεται η **ένταση του σήματος (magnitude)** χρησιμοποιώντας τα πραγματικά και φανταστικά μέρη των FFT δεδομένων:

$$magnitude = \sqrt{real^2 + imaginary^2} \quad (6.1)$$

Αυτό αντιστοιχεί στην υπολογιστική εκτίμηση της ισχύος του σήματος σε κάθε συχνότητα, βάσει των βασικών αρχών της **Fourier Analysis** (κεφάλαιο 3.5.1) . Ο υπολογισμός γίνεται για όλες τις συχνότητες κάθε ζώνης και τα αποτελέσματα αθροίζονται, δίνοντας ένα συνολικό μέτρο έντασης ανά ζώνη [88].

Με τον τρόπο αυτό:

- Το πρόγραμμα μπορεί να **αντιληφθεί πότε υπάρχει έντονο beat** σε κάθε ζώνη, βασισμένο στην απότομη αύξηση της magnitude.
- Οι χαμηλές συχνότητες βοηθούν στην ανίχνευση μπάσου ρυθμού, οι μεσαίες στην αναγνώριση μελωδικών στοιχείων, και οι υψηλές σε πιο λεπτομερή ηχητικά στοιχεία.

### Σχέση με απτική ανατροφοδότηση

Οι τιμές magnitude που υπολογίζονται σε αυτό το σημείο χρησιμοποιούνται στη συνέχεια για:

- **Ρυθμική δόνηση:** μέσω του RhythmVibrator, η συσκευή δονείται ανάλογα με την ένταση κάθε ζώνης.
- **Οπτικοποίηση:** τα δεδομένα χρησιμοποιούνται για να σχεδιαστούν οι μπάρες στο VisualizerView, δίνοντας οπτική αναπαράσταση των συχνοτήτων

```

61     var lowMagnitude = 0f
62     for (i in lowBandStart ≤ until < lowBandEnd) {
63         val real = bytes[i * 2].toFloat()
64         val imaginary = bytes[i * 2 + 1].toFloat()
65         lowMagnitude += sqrt(x: real * real + imaginary * imaginary)
66     }
67     var midMagnitude = 0f
68     for (i in midBandStart ≤ until < midBandEnd) {
69         val real = bytes[i * 2].toFloat()
70         val imaginary = bytes[i * 2 + 1].toFloat()
71         midMagnitude += sqrt(x: real * real + imaginary * imaginary)
72     }
73     var highMagnitude = 0f
74     for (i in highBandStart ≤ until < highBandEnd) {
75         val real = bytes[i * 2].toFloat()
76         val imaginary = bytes[i * 2 + 1].toFloat()
77         highMagnitude += sqrt(x: real * real + imaginary * imaginary)
78     }

```

Εικόνα 6.66: Visualizer detectBeats() 2

#### Αποθήκευση και εξομάλυνση τιμών

Οι τιμές **magnitude** κάθε ζώνης προστίθενται σε αντίστοιχες λίστες (`lowMagnitudes`, `midMagnitudes`, `highMagnitudes`) (γραμμές 80-82). Αυτό επιτρέπει την **αναλυτική παρακολούθηση της έντασης του ήχου** σε μια χρονική σειρά, αντί να εξετάζεται μόνο το τελευταίο δείγμα.

Οι λίστες περιορίζονται στα τελευταία 10 δείγματα (γραμμές 85-87), ώστε να διατηρείται ένα **παράθυρο κινούμενου μέσου όρου (moving average)**. Αυτή η τεχνική εξομάλυνσης μειώνει τον θόρυβο και τα τυχαία spikes που μπορεί να προκαλέσουν ψευδή ανίχνευση beats. Θεωρητικά, αυτή η μέθοδος βασίζεται σε **time-domain smoothing**, ένα κοινό εργαλείο στην επεξεργασία σήματος για σταθερότερη εκτίμηση χαρακτηριστικών.

```

80     lowMagnitudes.add(lowMagnitude)
81     midMagnitudes.add(midMagnitude)
82     highMagnitudes.add(highMagnitude)
83
84     // Keep only the last N samples for smoothing
85     if (lowMagnitudes.size > 10) lowMagnitudes.removeAt(index: 0)
86     if (midMagnitudes.size > 10) midMagnitudes.removeAt(index: 0)
87     if (highMagnitudes.size > 10) highMagnitudes.removeAt(index: 0)

```

Εικόνα 6.67: Visualizer detectBeats() 3

Υπολογίζεται η **μέση ένταση (average magnitude)** για κάθε ζώνη (γραμμές 90-92), η οποία λειτουργεί ως βάση για τη διάκριση σημαντικών αυξήσεων έντασης από τον τυχαίο θόρυβο.

Κάθε ζώνη αποκτά ένα **δυναμικό threshold**, πολλαπλασιάζοντας τον μέσο όρο επί 1.5 (γραμμές 95-97). Αυτή η τεχνική είναι γνωστή ως **adaptive thresholding** και επιτρέπει στο σύστημα να προσαρμόζεται σε διαφορετικά επίπεδα έντασης, π.χ. τραγούδια με χαμηλή ή υψηλή ισχύ.

Όταν η τρέχουσα ένταση σε μια ζώνη ξεπεράσει το αντίστοιχο `threshold`, θεωρείται ότι έχει εμφανιστεί ένα **beat**. Στη συνέχεια:

- Καλείται το **callback `onBeatDetected`**, ενημερώνοντας την εφαρμογή για την ένταση και τη ζώνη του **beat**.
- Καλείται η μέθοδος **`vibrateBasedOnBeat()`**, η οποία παράγει απτική ανάδραση μέσω του `RhythmVibrator`.

```

90     val avgLow = lowMagnitudes.average().toFloat()
91     val avgMid = midMagnitudes.average().toFloat()
92     val avgHigh = highMagnitudes.average().toFloat()
93
94     // Dynamic thresholds based on the moving averages
95     val lowThreshold = avgLow * 1.5f
96     val midThreshold = avgMid * 1.5f
97     val highThreshold = avgHigh * 1.5f
98
99     // Trigger beat detection based on the thresholds
100    when {
101        lowMagnitude > lowThreshold -> {
102            onBeatDetected?.invoke(lowMagnitude.toInt(), "low")
103            vibrateBasedOnBeat(lowMagnitude.toInt(), frequencyBand: "low")
104        }
105        midMagnitude > midThreshold -> {
106            onBeatDetected?.invoke(midMagnitude.toInt(), "mid")
107            vibrateBasedOnBeat(midMagnitude.toInt(), frequencyBand: "mid")
108        }
109        highMagnitude > highThreshold -> {
110            onBeatDetected?.invoke(highMagnitude.toInt(), "high")
111            vibrateBasedOnBeat(highMagnitude.toInt(), frequencyBand: "high")
112        }
113    }
114 }

```

Εικόνα 6.68: Visualizer detectBeats() 4

Η **`vibrateBasedOnBeat()`** ενεργοποιεί τον **`RhythmVibrator`**, παράγοντας δονήσεις ανάλογα με:

- **Magnitude:** την ένταση του **beat**, ώστε δυνατά **beats** να δίνουν πιο έντονες δονήσεις.
- **FrequencyBand:** τη ζώνη συχνοτήτων (χαμηλή, μεσαία, υψηλή), ώστε διαφορετικά είδη **beats** να παράγουν διακριτές απτικές ενδείξεις.

Η μέθοδος **`release()`** απελευθερώνει τους πόρους του `Visualizer` όταν δεν χρειάζεται, αποτρέποντας διαρροές μνήμης και περιττή κατανάλωση πόρων.

```

118     private fun vibrateBasedOnBeat(magnitude: Int, frequencyBand: String) {
119         rhythmVibrator.vibrateWithWaveform(magnitude, frequencyBand)
120     }
121     // Function to release the visualizer when not needed
122     @ZoZoTs99
123     fun release() {
124         visualizer?.release()
125         visualizer = null
126     }

```

Εικόνα 6.69: Visualizer vibrateBasedOnBeat

Η μέθοδος `onDraw()` (γραμμές 128-154) αναλαμβάνει την **οπτική αναπαράσταση του φάσματος συχνοτήτων**:

- Υπολογίζεται ο αριθμός των μπαρών (`barCount`) και το πλάτος τους (`barWidth`) για την οθόνη. (γραμμές 131-132)
- Για κάθε δείγμα FFT υπολογίζεται η **ένταση (`magnitude`)** [88]. (γραμμές 134-136)
- Το ύψος κάθε μπάρας κλιμακώνεται λογαριθμικά με τον δείκτη της (**`scaledIndex`**), προκύπτει από τον συνδυασμό της κανονικοποιημένης έντασης ( $\frac{M_k}{256}$ ), του μέγιστου ύψους  $H$  και ενός λογαριθμικού παράγοντα ανά δείκτη ( $\log_{10}(k+1)$ ). Ο λογάριθμος μειώνει την υπερβολική μεγέθυνση των μπαρών υψηλών δεικτών, δίνοντας μεγαλύτερη έμφαση στις χαμηλές συχνοτήτες και πιο ευανάγνωστη οπτικοποίηση. Ο τύπος προκύπτει ως φυσικός συνδυασμός των παραπάνω παραγόντων: κανονικοποιημένη ένταση  $\times$  μέγιστο ύψος  $\times$  λογαριθμικός δείκτης. Είναι αποτέλεσμα λογικής κλιμάκωσης για οπτική ισορροπία.

Το **ύψος κάθε μπάρας** υπολογίζεται ως:

$$Height_k = \left(\frac{M_k}{256}\right) \cdot H \cdot \log_{10}(k + 1) \quad (6.2)$$

όπου:

- $M_k$  είναι η ένταση,
  - $H$  = ύψος καμβά (π.χ. οθόνη κινητού),
  - $\log_{10}(k+1)$  = λογαριθμική κλιμάκωση για έμφαση στις χαμηλές συχνοτήτες. Αυτή η κλιμάκωση αντισταθμίζει την ψυχοακουστική αντίληψη του ανθρώπου, που αντιλαμβάνεται πιο έντονα τα μπάσα, και εξηγεί γιατί οι μπάρες των χαμηλών συχνοτήτων φαίνονται μεγαλύτερες.
- Οι μπάρες χρωματίζονται ανά ζώνη συχνοτήτων, **κόκκινο** για χαμηλές (bass), **κίτρινο** για μεσαίες, **πράσινο** για υψηλές. (γραμμές 140-144)



Εικόνα 6.70: Visualizer colors

- Σχεδιάζεται το σύνολο στο canvas (γραμμές 145-151), δημιουργώντας μια **ζωντανή, δυναμική οπτικοποίηση του ήχου** που ενημερώνεται σε πραγματικό χρόνο με τα νέα FFT δεδομένα, (εικόνα 6.70).

```

128 override fun onDraw(canvas: Canvas) {
129     super.onDraw(canvas)
130     fftBytes?.let { bytes ->
131         val barCount = min( a: 64, b: bytes.size / 2)
132         val barWidth = width / barCount.toFloat()
133         for (i in 0 until barCount) {
134             val real = bytes[i * 2].toFloat()
135             val imaginary = bytes[i * 2 + 1].toFloat()
136             val magnitude = sqrt( x: real * real + imaginary * imaginary)
137             val scaledIndex = log10((i + 1).toDouble()).toFloat()
138             val barHeight = (magnitude / 256) * height * scaledIndex
139             // Color bars based on frequency range
140             paint.color = when {
141                 i < 6 -> Color.RED // Low frequencies (bass)
142                 i < 20 -> Color.YELLOW // Mid frequencies
143                 else -> Color.GREEN // High frequencies
144             }
145             canvas.drawRect(
146                 left: i * barWidth,
147                 top: height - barHeight,
148                 right: (i + 1) * barWidth,
149                 height.toFloat(),
150                 paint
151             )
152         }
153     }
154 }
155 }

```

Εικόνα 6.71: Visualizer onDraw()

Θεωρητικά, η μέθοδος αυτή εφαρμόζει **graphical spectrum visualization**, που είναι κοινή στην επεξεργασία ήχου και την HCI, επιτρέποντας στον χρήστη να «δει» τον ήχο και να αντιληφθεί ρυθμούς και δυναμική.

## 6.7 Σελίδα Δόνησης Ρυθμού (Page RhythmVibrator)

Η απτική ανατροφοδότηση βασίζεται στον ρυθμό και στην ένταση του τραγουδιού. Μέσω ειδικού αλγορίθμου, υπολογίζεται η χρονική ένταση των δονήσεων, με στόχο την πιστή αναπαράσταση των ηχητικών χαρακτηριστικών.

### KOTLIN

**Imports** (γραμμές 3-6): Εισαγωγή απαραίτητων βιβλιοθηκών που χρειάζεται η κλάση για να λειτουργήσει.

```

1 package com.example.musictovibrationapp
2
3 import android.content.Context
4 import android.os.Build
5 import android.os.VibrationEffect
6 import android.os.Vibrator

```

Εικόνα 6.72: Rhythm imports

### Μεταβλητές (γραμμή 10)

Δημιουργείται ένα **αντικείμενο vibrator**, το οποίο είναι υπεύθυνο για τον έλεγχο του μηχανισμού δόνησης του κινητού. Αυτό εξασφαλίζει ότι η εφαρμογή μπορεί να παράγει δονήσεις σε συγχρονισμό με τη μουσική.

Στο Android, το **Vibrator** είναι μια **system service API** που δίνει πρόσβαση στο υλικό (hardware) της συσκευής το οποίο δημιουργεί μηχανικές δονήσεις.

- Οι δονήσεις παράγονται μέσω **ERM** ή **LRA**, ανάλογα με τη συσκευή (κεφ 3.2).
- Ο προγραμματιστής δεν χρειάζεται να γνωρίζει τον ακριβή τύπο του μηχανισμού, αρκεί να χρησιμοποιεί το API.

Η **vibrateToBeat()** (γραμμές 12-23) είναι υπεύθυνη για να δημιουργήσει μια δόνηση που συγχρονίζεται με το μουσικό beat, χρησιμοποιώντας την **ένταση** και τη **συχνότητα** του ήχου.

Η **mapIntensityToDuration** (γραμμή 13) μετατρέπει την ένταση του beat σε **χρόνο δόνησης (milliseconds)** και ο χρόνος προσαρμόζεται ανάλογα με το frequency band:

- χαμηλές συχνότητες (bass) → πιο μεγάλη διάρκεια,
- μεσαίες → μέτρια διάρκεια,
- υψηλές → σύντομη δόνηση.

Γίνεται ο έλεγχος έκδοσης Android (γραμμές 15-22). Στα νεότερα Android (API 26+, Android 8.0 και πάνω) χρησιμοποιείται η πιο σύγχρονη κλάση **VibrationEffect**. Ενώ σε παλιότερες εκδόσεις (κάτω από Android 8.0), χρησιμοποιείται η παλαιότερη **vibrator.vibrate(duration)**. Έτσι εξασφαλίζεται **συμβατότητα** με όλες τις συσκευές. Δημιουργείται μια **μοναδική δόνηση** διάρκειας duration (γραμμή

17). Το `DEFAULT_AMPLITUDE` αφήνει το Android να αποφασίσει την ένταση της δόνησης, ανάλογα με το hardware.

```

8  class RhythmVibrator(private val context: Context) {
9
10     private val vibrator: Vibrator = context.getSystemService(Context.VIBRATOR_SERVICE) as Vibrator
11     // Function to vibrate based on the beat detected
12     @ZoZoTs99
13     fun vibrateToBeat(intensity: Int, frequencyBand: String) {
14         val duration = mapIntensityToDuration(intensity, frequencyBand)
15
16         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
17             // API 26+ (Android 8.0+)
18             val effect = VibrationEffect.createOneShot(duration, VibrationEffect.DEFAULT_AMPLITUDE)
19             vibrator.vibrate(effect)
20         } else {
21             // For older versions (below API 26)
22             vibrator.vibrate(duration)
23         }
24     }
25 }

```

Εικόνα 6.73: Rhythm vibrateToBeat()

Η συνάρτηση **vibrateWithWaveform** (γραμμές 26-39) είναι υπεύθυνη για την ενεργοποίηση της δόνησης της συσκευής με βάση την ένταση και τη συχνότητα ενός συγκεκριμένου μουσικού εύρους (frequency band). Ο αλγόριθμος λαμβάνει ως παραμέτρους την ένταση (intensity) και την κατηγορία συχνότητας (frequencyBand) και προσαρμόζει τη δόνηση αναλόγως.

Αρχικά ελέγχει την έκδοση του λειτουργικού συστήματος Android (γραμμή 27). Αν είναι **8.0+ (API 26)** υποστηρίζεται το **VibrationEffect.createWaveform** (γραμμή 32), το οποίο επιτρέπει λεπτομερή έλεγχο της διάρκειας και της έντασης σε κάθε στάδιο.

Συγκεκριμένα:

1. Καλείται η συνάρτηση **mapIntensityToDurationAndAmplitude** (γραμμή 29) η οποία αντιστοιχίζει την ένταση και το εύρος συχνότητας σε δύο βασικές παραμέτρους δόνησης: τη διάρκεια και το πλάτος της δόνησης.
2. Δημιουργούνται δύο πίνακες (γραμμές 30,31): Ο **timings** καθορίζει ότι η δόνηση ξεκινά άμεσα (0 ms) και διαρκεί όσο ορίζει η μεταβλητή duration. Ο **amplitudes** καθορίζει ότι αρχικά υπάρχει σιωπή (0) και στη συνέχεια η δόνηση με το υπολογισμένο πλάτος (amplitude).
3. Κατασκευάζεται το αντικείμενο **VibrationEffect** με τη μέθοδο **createWaveform** (γραμμή 32), με επανάληψη -1, που σημαίνει ότι η δόνηση εκτελείται μόνο μία φορά. Έτσι, κάθε παλμός δόνησης συγχρονίζεται αποκλειστικά με τον εκάστοτε ρυθμό που ανιχνεύεται.
4. Τέλος, η δόνηση ενεργοποιείται με την κλήση **vibrator.vibrate(effect)** (γραμμή 34).

Σε περίπτωση που η έκδοση είναι παλαιότερη (γραμμές 35-38) χρησιμοποιείται η μέθοδος **vibrateToBeat** (γραμμή 37), η οποία υλοποιεί δόνηση χωρίς κυματομορφή, πιθανόν βασισμένη μόνο σε σταθερή διάρκεια και ένταση.

Συνοπτικά, η συνάρτηση επιτρέπει στον χρήστη να αισθανθεί τη μουσική μέσω της συσκευής, προσαρμόζοντας τη δόνηση δυναμικά σύμφωνα με την ένταση και το φάσμα συχνότητας του ήχου, αξιοποιώντας πλήρως τις δυνατότητες των σύγχρονων εκδόσεων Android.

```

26 fun vibrateWithWaveform(intensity: Int, frequencyBand: String) {
27     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
28
29         val (duration, amplitude) = mapIntensityToDurationAndAmplitude(intensity, frequencyBand)
30         val timings = longArrayOf(0, duration) // Start immediately, then vibrate
31         val amplitudes = intArrayOf(0, amplitude) // Silence first, then actual vibration
32         val effect = VibrationEffect.createWaveform(timings, amplitudes, repeat: -1) // -1 = no repeat
33
34         vibrator.vibrate(effect)
35     } else {
36         // Fallback to simpler vibration
37         vibrateToBeat(intensity, frequencyBand)
38     }
39 }

```

Εικόνα 6.74: Rhythm vibrateWithWaveform()

**stopVibration** (γραμμές 42-44): χρησιμοποιείται για να διακόψει οποιαδήποτε δόνηση είναι σε εξέλιξη, εξασφαλίζοντας ότι η συσκευή δεν θα συνεχίσει να δονείται πέρα από το επιθυμητό χρονικό διάστημα.

```

42 fun stopVibration() {
43     vibrator.cancel()
44 }

```

Εικόνα 6.75: Rhythm stopVibration()

Η συνάρτηση **mapIntensityToDuration** (γραμμές 47-63) υλοποιήθηκε με σκοπό να **μετατρέψει την ένταση ενός μουσικού σήματος σε κατάλληλη διάρκεια δόνησης**, προσαρμοσμένη στο φάσμα συχνοτήτων του σήματος. Η ανάγκη για διαφορετική αντιμετώπιση ανά εύρος συχνοτήτων προκύπτει από το γεγονός ότι οι χαμηλές, μεσαίες και υψηλές συχνότητες αντιλαμβάνονται διαφορετικά από τον χρήστη: οι χαμηλές συχνότητες συνήθως απαιτούν μεγαλύτερη διάρκεια για να γίνουν αντιληπτές, ενώ οι υψηλές χρειάζονται πιο σύντομες και διακριτές δονήσεις. Έτσι:

- Για **χαμηλές συχνότητες** (low), η διάρκεια διπλασιάζεται και περιορίζεται σε 500ms, ώστε η δόνηση να είναι αισθητή αλλά να μην γίνεται υπερβολικά μεγάλη.
- Για **μεσαίες συχνότητες** (mid), η διάρκεια είναι ίση με την ένταση και περιορίζεται σε 300ms, παρέχοντας μια ισορροπημένη αίσθηση.
- Για **υψηλές συχνότητες** (high), η διάρκεια μειώνεται στο ήμισυ της έντασης και περιορίζεται σε 150ms, ώστε η δόνηση να παραμένει σύντομη και καθαρή.
- Σε κάθε άλλη περίπτωση (**else**), εφαρμόζεται η ίδια λογική με τις μεσαίες συχνότητες, εξασφαλίζοντας μια σταθερή προεπιλεγμένη συμπεριφορά.

**Αρα Απτική Απόκριση:**

- Οι **μπάσες συχνότητες** οδηγούν σε **ισχυρότερες και μεγαλύτερες** δονήσεις.
- Οι **πρίμες συχνότητες** παράγουν **σύντομες και ήπιες** δονήσεις.

Η κλιμάκωση είναι συνεπής με τις ιδιότητες της ανθρώπινης αφής:

Χαμηλές συχνότητες → βαθύτερη διέγερση δερματικών υποδοχέων.

Υψηλές συχνότητες → επιφανειακή αίσθηση.

```

47 private fun mapIntensityToDuration(intensity: Int, frequencyBand: String): Long {
48     return when (frequencyBand) {
49         "low" -> {
50             (intensity * 2).toLong().coerceAtMost( maximumValue: 500) // Cap at 500ms for low
51         }
52         "mid" -> {
53             (intensity).toLong().coerceAtMost( maximumValue: 300) // Cap at 300ms for mid
54         }
55         "high" -> {
56             (intensity / 2).toLong().coerceAtMost( maximumValue: 150) // Cap at 150ms for high
57         }
58         else -> {
59             // Default case, medium vibration duration
60             (intensity).toLong().coerceAtMost( maximumValue: 300)
61         }
62     }
63 }

```

Εικόνα 6.76: Rhythm mapIntensityToDuration()

Η συνάρτηση **mapIntensityToDurationAndAmplitude** επεκτείνει τη βασική λογική της αντιστοίχισης έντασης σε διάρκεια δόνησης, εισάγοντας παράλληλα τον υπολογισμό της έντασης της δόνησης (amplitude). Ο σκοπός της είναι η παραγωγή δονήσεων που αντανακλούν πιο ρεαλιστικά τα χαρακτηριστικά του μουσικού σήματος, τόσο σε διάρκεια όσο και σε ένταση, λαμβάνοντας υπόψη το εύρος συχνοτήτων. Συγκεκριμένα:

- Υπολογισμός διάρκειας (duration) (γραμμές 68-73): ακριβώς το ίδιο με την **mapIntensityToDuration**.
- Υπολογισμός έντασης δόνησης (amplitude) (γραμμή 74): Η ένταση δόνησης υπολογίζεται ως  $intensity \cdot 2$  και περιορίζεται στο εύρος 1–255, ώστε να παραμένει αντιληπτή και να μην ξεπερνά τα τεχνικά όρια του κινητήρα δόνησης.
- Η συνάρτηση επιστρέφει ένα ζεύγος (γραμμή 76) όπου οι τιμές αυτές χρησιμοποιούνται στη συνάρτηση **vibrateWithWaveform** για τη δημιουργία της κυματομορφής δόνησης.

```

66     private fun mapIntensityToDurationAndAmplitude(intensity: Int,
67                                                    frequencyBand: String): Pair<Long, Int> {
68         val duration = when (frequencyBand) {
69             "low" -> (intensity * 2).toLong().coerceAtMost( maximumValue: 500)
70             "mid" -> intensity.toLong().coerceAtMost( maximumValue: 300)
71             "high" -> (intensity / 2).toLong().coerceAtMost( maximumValue: 150)
72             else -> intensity.toLong().coerceAtMost( maximumValue: 300)
73         }
74         val amplitude = (intensity * 2).coerceIn(1, 255) // 0 = silence, 255 = max intensity
75
76         return Pair(duration, amplitude)
77     }
78 }

```

Εικόνα 6.77: Rhythm mapIntensityToDurationAndAmplitude()

## 6.8 Διάγραμμα Ροής (Flowchart)

### Από το Ηχητικό Σήμα στην Πολυαισθητηριακή Αναπαράσταση

Η ροή μετασχηματισμού περιγράφεται ως εξής:

Audio Signal → FFT → Frequency Magnitudes → Band Partitioning → Visual Display (Bars)/Haptic Feedback (Vibration)

Δηλαδή:

- Το ηχητικό σήμα  $x[n]$  μετασχηματίζεται μέσω FFT σε  $X[k]$ , Μαθηματική σχέση (3.9)[52].
- Υπολογίζεται το μέτρο  $M_k$ , σχέση (6.1) για κάθε συχνοτικό bin  $k$  [88].
- Τα  $M_k$  ομαδοποιούνται σε τρεις ζώνες συχνοτήτων (low, mid, high).
- Η συνολική ενέργεια κάθε ζώνης οδηγεί:
  - Στην **οπτική αναπαράσταση** μέσω ράβδων (bars),
  - Στην **απτική αναπαράσταση** μέσω δονήσεων.

Η κλάση **VisualizerView** υλοποιεί την απεικόνιση και την ανάλυση του μουσικού σήματος σε πραγματικό χρόνο, συνδέοντας την οπτική αναπαράσταση του φάσματος συχνοτήτων με την απτική απόκριση μέσω της κλάσης **RhythmVibrator**. Το σύστημα λειτουργεί ως εξής: το σήμα ήχου συνδέεται με τον **Visualizer** της Android μέσω του `linkToMediaPlayer`, ο οποίος συλλέγει **FFT** δεδομένα σε καθορισμένο ρυθμό δειγματοληψίας. Η ανάλυση των FFT δεδομένων χωρίζει το φάσμα σε τρεις βασικές ζώνες: χαμηλές (bass), μεσαίες (mid) και υψηλές (treble) συχνοτήτες [52].

Για κάθε ζώνη υπολογίζεται το μέγεθος (magnitude) χρησιμοποιώντας τον τύπο 6.1 και εφαρμόζεται **κινητός μέσος όρος** για εξομάλυνση των τιμών και μείωση θορύβου. Η ανίχνευση beat

πραγματοποιείται όταν η τρέχουσα τιμή magnitude υπερβαίνει ένα δυναμικό όριο που βασίζεται στον μέσο όρο της ζώνης [88], δηλαδή:

$$Mag_{band}(t) > 1.5 \cdot Avg_{band}(t) \quad [88] \quad (6.3)$$

Όταν ανιχνεύεται beat, καλείται η **vibrateWithWaveform** της **RhythmVibrator**, η οποία μετατρέπει το μέγεθος του beat και τη ζώνη συχνοτήτων σε **διάρκεια και ένταση δόνησης**. Έτσι, οι χαμηλές συχνοτήτες οδηγούν σε μεγαλύτερες και πιο έντονες δονήσεις, ενώ οι υψηλές παράγουν σύντομες και διακριτές δονήσεις, δημιουργώντας μια **πιο ρεαλιστική απτική εμπειρία**.

## 6.9 Παρατηρήσεις και Συμπεράσματα

### Παρατηρήσεις

1. Κατά τη διάρκεια των δοκιμών παρατήρησα ότι όταν οι μπάρες οπτικοποίησης εμφανίζονταν με μεγάλο ύψος, η απτική ανατροφοδότηση (haptic feedback) γινόταν αντιληπτά πιο αδύναμη και λιγότερο ρεαλιστική σε σχέση με την οπτική ένδειξη. Αυτό συμβαίνει επειδή η οπτική κλίμακα μπορεί να υπερτονίζει τις διακυμάνσεις του σήματος, χωρίς να αυξάνει αντίστοιχα την ένταση της δόνησης που περιορίζεται από τα φυσικά όρια του κινητήρα. Για την αντιμετώπιση αυτού του φαινομένου, προσαρμόστηκε η κλίμακα των μπαρών ώστε να μειωθεί το εύρος του ύψους τους, διατηρώντας παράλληλα σταθερή και ρεαλιστική την ένταση της απτικής ανατροφοδότησης.
2. Στην υλοποίηση της κλάσης **RhythmVibrator** υπάρχουν δύο συναρτήσεις για την αντιστοίχιση της έντασης του ήχου: η **mapIntensityToDuration** και η **mapIntensityToDurationAndAmplitude**. Παρά το γεγονός ότι και οι δύο συναρτήσεις υπολογίζουν τη διάρκεια της δόνησης ανάλογα με το εύρος συχνοτήτων, η δεύτερη παρέχει επιπλέον τη δυνατότητα υπολογισμού της έντασης (amplitude) της δόνησης.

Η ύπαρξη δύο συναρτήσεων έχει τεχνικό και σχεδιαστικό λόγο:

- a) Η **mapIntensityToDuration** χρησιμοποιείται στην **vibrateToBeat**, όπου η δόνηση είναι απλή και χρειάζεται μόνο η διάρκεια.
- b) Η **mapIntensityToDurationAndAmplitude** χρησιμοποιείται στην **vibrateWithWaveform**, όπου απαιτείται **λεπτομερής έλεγχος και της έντασης**, ώστε η δόνηση να ακολουθεί πιστά την ένταση του ήχου και να δημιουργείται πιο ρεαλιστικό αποτέλεσμα.

Με αυτόν τον τρόπο, η υλοποίηση παραμένει **ευέλικτη και συμβατή με διαφορετικές εκδόσεις Android**, ενώ ταυτόχρονα επιτρέπει και απλές δονήσεις χωρίς δυναμικό amplitude. Η ύπαρξη δύο συναρτήσεων δεν αποτελεί πλεονασμό, αλλά **πρακτική λύση για την υποστήριξη διαφορετικών τύπων δόνησης**, ανάλογα με τις δυνατότητες του συστήματος και τον τύπο εφέ που επιθυμούμε να δημιουργήσουμε.

### Πρακτικά στην αρχιτεκτονική της εφαρμογής:

Σε **Android 13+** (TIRAMISU / API 33), η άδεια `READ_MEDIA_AUDIO` είναι μια «άδεια κατά το runtime» (runtime permission), και μετά από δύο φορές που ο χρήστης την αρνήθηκε, το Android **σταματά αυτόματα να εμφανίζει το παράθυρο αδειών κατά το άνοιγμα της εφαρμογής**. Το Android υποθέτει ότι ο χρήστης δεν θέλει πλέον να του εμφανίζονται αιτήματα αδειών, και πρέπει να πάει στις **ρυθμίσεις** της εφαρμογής.

### Συμπεράσματα

Η σύνδεση της οπτικής απεικόνισης (**VisualizerView**) με την απτική ανατροφοδότηση (**RhythmVibrator**) παρέχει ένα **συνολικό σύστημα που ενισχύει την αντίληψη του ρυθμού** με πολλαπλές αισθήσεις. Σε σύγκριση με άλλα συστήματα που προσφέρουν είτε μόνο οπτική είτε μόνο απτική ανατροφοδότηση, αυτό το σύστημα προσφέρει συγχρονισμένη **multimodal εμπειρία**, ενισχύοντας την αντίληψη του χρήστη για τον μουσικό ρυθμό.

### Μαθηματικά

- Ο υπολογισμός του **magnitude** από τα **FFT** δεδομένα αντιστοιχεί σε βασικές αρχές της φασματικής ανάλυσης [88].
- Ο κινητός μέσος όρος και ο δυναμικός πολλαπλασιαστής για **thresholds** εξασφαλίζουν σταθερή και αξιόπιστη αντίληψη **beat** [88].
- Η κατανομή σε ζώνες και η αναγωγή σε διάρκεια και ένταση δόνησης αντιστοιχεί σε μαθηματικά μοντέλα για απτική κλιμάκωση ανά συχνότητα [88].

### Φυσική

Το σύστημα συνδέεται άμεσα με έννοιες φυσικής, καθώς η απτική απόκριση μεταφράζει **ηχητικά κύματα σε μηχανικές δονήσεις** μέσω του κινητήρα δόνησης της συσκευής. Η ένταση της δόνησης (amplitude) αντιστοιχεί στην **δύναμη** που ασκείται στους δερματικούς υποδοχείς, ενώ η διάρκεια της δόνησης σχετίζεται με την **ενέργεια και την αντίληψη χρόνου της δόνησης** από τον άνθρωπο.

Με αυτόν τον τρόπο, η συνδυαστική χρήση οπτικής και απτικής ανατροφοδότησης δημιουργεί ένα πλήρες, δυναμικό σύστημα που ξεπερνά τις δυνατότητες συμβατικών μεθόδων αναπαράστασης μουσικού σήματος.

Το σύστημα **VisualizerView** και **RhythmVibrator** αποτελεί ένα εξαιρετικό παράδειγμα **διεπιστημονικής συνένωσης**:

- **Μαθηματικών** (FFT, Στατιστικής Εξομάλυνσης),
- **Φυσικής** (Ηχητικά Φαινόμενα, Απτική Αντίληψη),
- **Πληροφορικής** (Real-time Signal Processing, UI rendering).

## Κεφάλαιο 7ο: Δοκιμές και Αξιολόγηση

### 7.1 Εισαγωγή

Στο πλαίσιο της παρούσας εργασίας, ζητήθηκε από ένα δείγμα ανθρώπων να χρησιμοποιήσουν την εφαρμογή και να την αξιολογήσουν απαντώντας σε ερωτήσεις, ταξινομημένες σε πέντε βασικές κατηγορίες, καταγράφοντας τις σχετικές απαντήσεις τους.

### 7.2 Ερωτηματολόγιο

Οι πέντε κατηγορίες που χωρίζεται το ερωτηματολόγιο είναι οι εξής:

- **Δημογραφικά Στοιχεία**
- **Δόνηση**
- **Οπτικοποίηση**
- **Γενική Εμπειρία**
- **Προτάσεις και Παρατηρήσεις**

#### Δημογραφικά Στοιχεία:

##### 1. Ηλικία

Απαντήσεις:

- Κάτω των 18
- 18 - 35
- 35 - 60
- 60+

##### 2. Φύλλο

Απαντήσεις:

- Άνδρας
- Γυναίκα

##### 3. Έχετε προβλήματα ακοής;

Απαντήσεις:

- Ναι

- Όχι

4. Έχετε χρησιμοποιήσει ποτέ εφαρμογές για μουσική με απτική ή οπτική αναπαράσταση;

Απαντήσεις:

- Ναι
- Όχι

**Δόνηση:**

1. Οι δονήσεις είναι ευχάριστες.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

2. Η διάρκεια των δονήσεων είναι κατάλληλη.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

3. Οι δονήσεις αντικατοπτρίζουν σωστά τις αλλαγές στην ένταση της μουσικής.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

4. Οι δονήσεις αντιπροσωπεύουν με ακρίβεια τον ρυθμό της μουσικής.

## Κεφάλαιο 7

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

5. Υπάρχουν διαφορές στις δονήσεις ανάλογα με το είδος της μουσικής.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

### **Οπτικοποίηση:**

1. Η οπτική αναπαράσταση είναι ξεκάθαρη και εύκολα αναγνωρίσιμη.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

2. Τα οπτικά εφέ βοηθούν να κατανοήσω τον ρυθμό της μουσικής.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

3. Η πολυπλοκότητα των μουσικών κομματιών αποδίδεται με σαφήνεια στην οπτική αναπαράσταση.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

**4.** Η οπτική αναπαράσταση διευκολύνει την ταυτόχρονη παρακολούθηση πολλών χαρακτηριστικών της μουσικής (ένταση, ρυθμός, όργανα).

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

**Γενική Εμπειρία:**

**1.** Είναι εύκολη η χρήση της εφαρμογής.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

**2.** Είμαι ικανοποιημένος/η συνολικά από την εφαρμογή.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

**3.** Η οπτικοποίηση είναι συγχρονισμένη σωστά με τις δονήσεις.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

**4.** Θεωρώ ότι η εφαρμογή είναι χρήσιμη για άτομα με προβλήματα ακοής.

(1:Διαφωνώ Απόλυτα – 5:Συμφωνώ Απόλυτα)

- 1
- 2
- 3
- 4
- 5

**Προτάσεις και Παρατηρήσεις:**

Υπάρχουν στοιχεία της εφαρμογής που θα θέλατε να βελτιωθούν; Αν ναι τι;

Ελεύθερη απάντηση.

**7.3 Στατιστικά στοιχεία:**

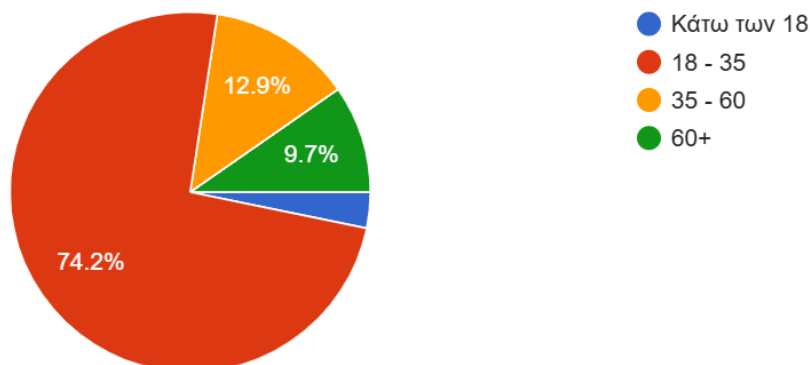
Στην παρούσα υποενότητα παρουσιάζονται στατιστικά δεδομένα που προέκυψαν από τη διαδικασία αξιολόγησης, με στόχο την αποτύπωση της εμπειρίας χρήσης της εφαρμογής και τον εντοπισμό πιθανών σημείων βελτίωσης για μελλοντικές αλλαγές.

**7.3.1 Δημογραφικά Στοιχεία**

Για την αξιολόγηση της εφαρμογής ερωτήθηκαν 31 άτομα εκ των οποίων το 74.2% περίπου ήταν 18-35, το 12,9% ήταν 35-60, το 9.7% άνω των 60 και το 3.2% κάτω των 18 ετών όπως φαίνεται και στο διάγραμμα 7.1:

## Ηλικία

31 responses

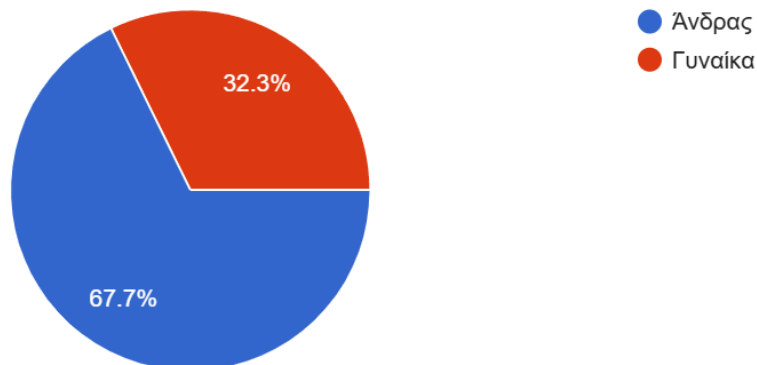


Εικόνα 7.1: Διάγραμμα ηλικιακής κατανομής δείγματος

Στην συνέχεια οι συμμετέχοντες ερωτήθηκαν για το φύλλο, με το 67.7% του δείγματος να αποτελείται από άντρες, ενώ το 32.3% να αποτελείται από γυναίκες σύμφωνα με το διάγραμμα 7.2:

## Φύλο

31 responses

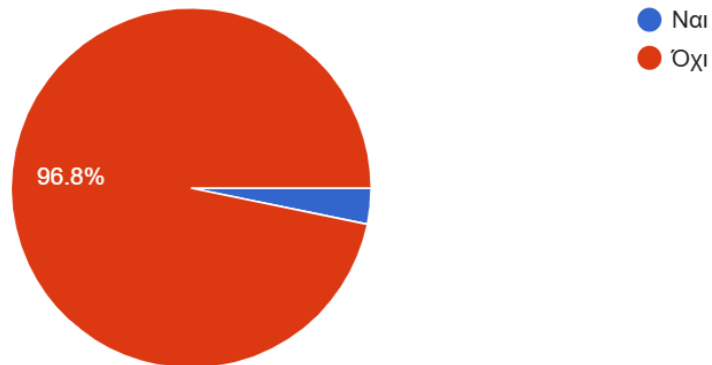


Εικόνα 7.2: Διάγραμμα κατανομής φύλου

Όσον αφορά πιθανά προβλήματα ακοής, ζητήθηκε από τους συμμετέχοντες να απαντήσουν με “Ναι” ή “Όχι”. Σύμφωνα με τα αποτελέσματα, το 3.2% δήλωσε ότι αντιμετωπίζει προβλήματα ακοής, ενώ το 96.8% απάντησε αρνητικά όπως βλέπουμε στο διάγραμμα 7.3:

Έχετε προβλήματα ακοής;

31 responses

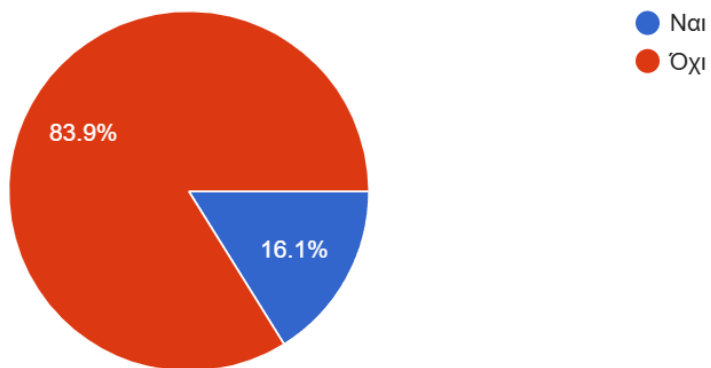


Εικόνα 7.3: Διάγραμμα για άτομα με προβλήματα ακοής

Στους συμμετέχοντες ζητήθηκε να απαντήσουν εάν έχουν χρησιμοποιήσει ποτέ εφαρμογές μουσικής που παρέχουν απτική ή οπτική αναπαράσταση. Το 16.1% απάντησε “Ναι” και το 83.9% “Όχι” όπως βλέπουμε στο διάγραμμα 7.4:

Έχετε χρησιμοποιήσει ποτέ εφαρμογές για μουσική με απτική ή οπτική αναπαράσταση;

31 responses



Εικόνα 7.4: Διάγραμμα εμπειρίας χρηστών με ειδικές εφαρμογές μουσικής

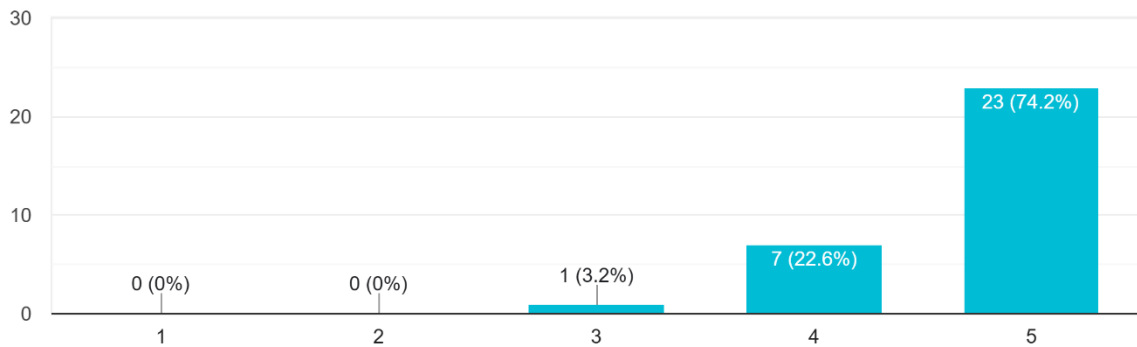
### 7.3.2 Αξιολόγηση Δονήσεων από Χρήστες

Οι συγκεκριμένες ερωτήσεις είναι πολλαπλών επιλογών με χαμηλότερη βαθμολογία το 1 (Διαφωνώ Απόλυτα) και υψηλότερη βαθμολογία το 5 (Συμφωνώ Απόλυτα).

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο οι δονήσεις της εφαρμογής τους άρεσαν, χρησιμοποιώντας την κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Οι δονήσεις είναι ευχάριστες.

31 responses

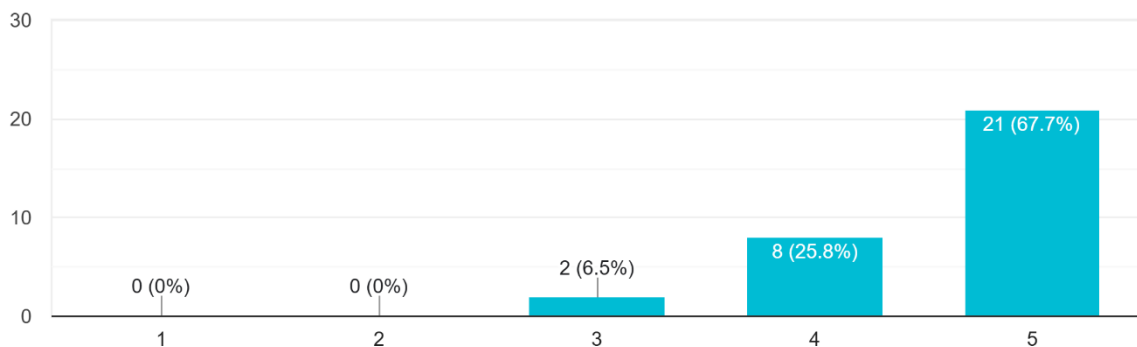


Εικόνα 7.5: Διάγραμμα αξιολόγησης της ευχαρίστησης από τις δονήσεις

Επίσης στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο η διάρκεια των δονήσεων της εφαρμογής είναι κατάλληλη, χρησιμοποιώντας την κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Η διάρκεια των δονήσεων είναι κατάλληλη.

31 responses



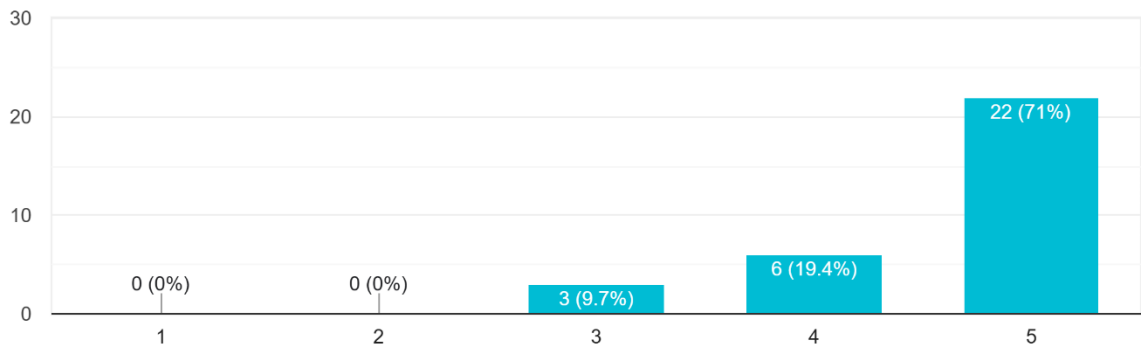
Εικόνα 7.6: Διάγραμμα αξιολόγησης διάρκειας δονήσεων

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο οι δονήσεις της εφαρμογής αντικατοπτρίζουν με ακρίβεια τις αλλαγές στην ένταση της μουσικής, χρησιμοποιώντας την κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

## Κεφάλαιο 7

Οι δονήσεις αντικατοπτρίζουν σωστά τις αλλαγές στην ένταση της μουσικής.

31 responses

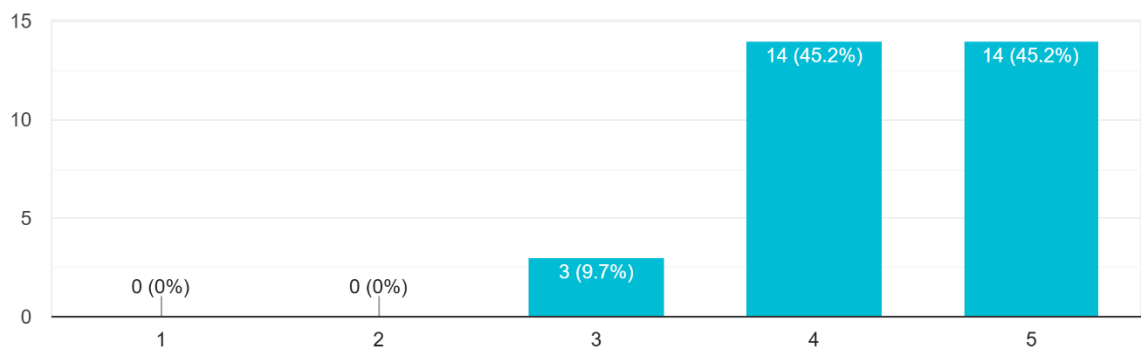


Εικόνα 7.7: Διάγραμμα αξιολόγησης αντιστοίχισης δονήσεων-έντασης

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο οι δονήσεις της εφαρμογής αντιπροσωπεύουν με ακρίβεια τον ρυθμό της μουσικής, χρησιμοποιώντας την κλίμακα 1-5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Οι δονήσεις αντιπροσωπεύουν με ακρίβεια τον ρυθμό της μουσικής.

31 responses

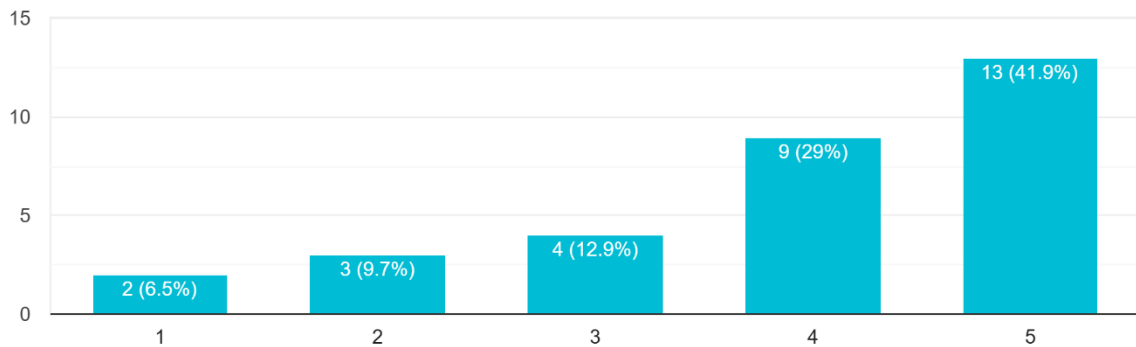


Εικόνα 7.8: Διάγραμμα ακρίβειας δονήσεων-ρυθμού

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο παρατηρούν διαφορές στις δονήσεις ανάλογα με το είδος της μουσικής, χρησιμοποιώντας την κλίμακα 1-5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Υπάρχουν διαφορές στις δονήσεις ανάλογα με το είδος της μουσικής.

31 responses



Εικόνα 7.9: Διάγραμμα διαφορών δονήσεων ανά είδος μουσικής

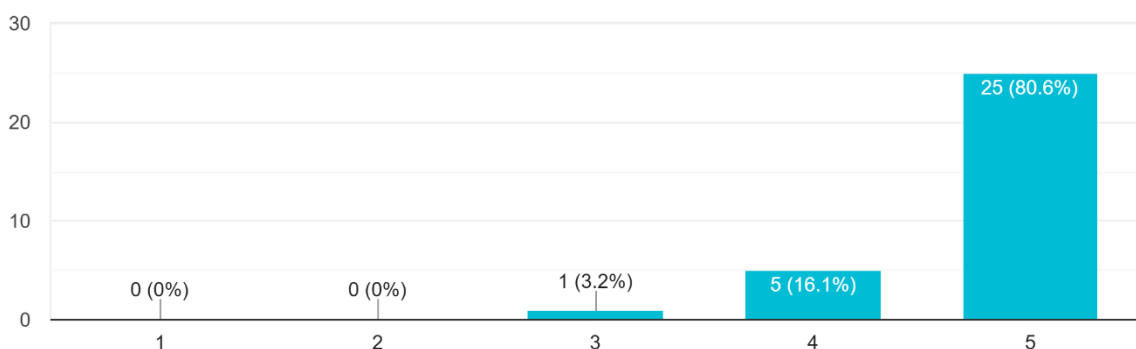
### 7.3.3 Αξιολόγηση Οπτικοποίησης από Χρήστες

Οι συγκεκριμένες ερωτήσεις είναι πολλαπλών επιλογών με χαμηλότερη βαθμολογία το 1 (Διαφωνώ Απόλυτα) και υψηλότερη βαθμολογία το 5 (Συμφωνώ Απόλυτα).

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο η οπτική αναπαράσταση της εφαρμογής είναι ξεκάθαρη και εύκολα αναγνωρίσιμη, χρησιμοποιώντας την κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Η οπτική αναπαράσταση είναι ξεκάθαρη και εύκολα αναγνωρίσιμη.

31 responses

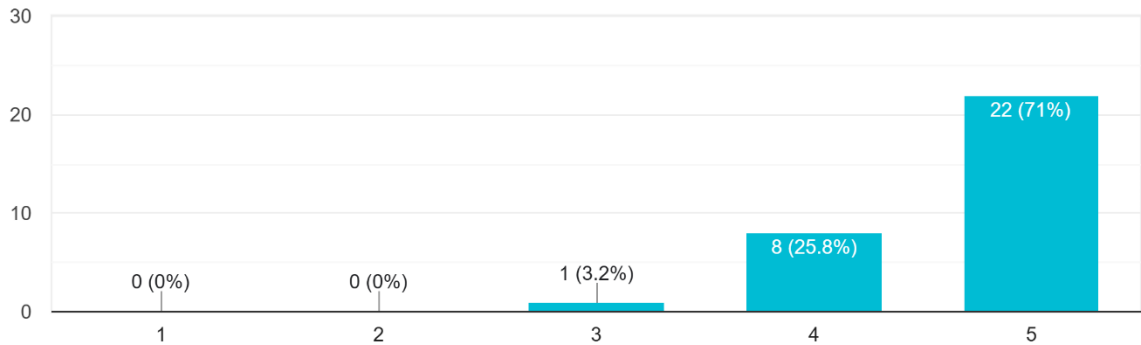


Εικόνα 7.10: Διάγραμμα αναγνωσιμότητας οπτικής αναπαράστασης

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο τα οπτικά εφέ της εφαρμογής τους βοηθούν να κατανοήσουν τον ρυθμό της μουσικής, χρησιμοποιώντας την κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Τα οπτικά εφέ βοηθούν να κατανοήσω τον ρυθμό της μουσικής.

31 responses

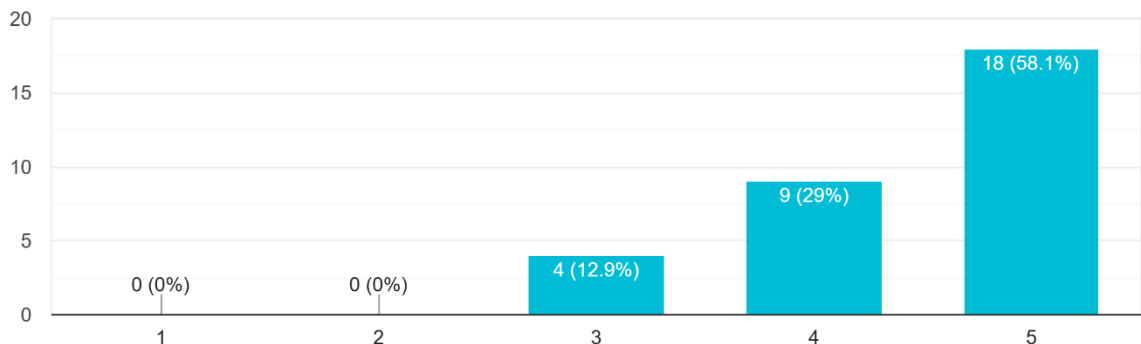


Εικόνα 7.11: Διάγραμμα επίδρασης οπτικών εφέ στον ρυθμό της μουσικής

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο η πολυπλοκότητα των μουσικών κομματιών αποδίδεται με σαφήνεια μέσω της οπτικής αναπαράστασης της εφαρμογής, χρησιμοποιώντας την κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Η πολυπλοκότητα των μουσικών κομματιών αποδίδεται με σαφήνεια στην οπτική αναπαράσταση.

31 responses

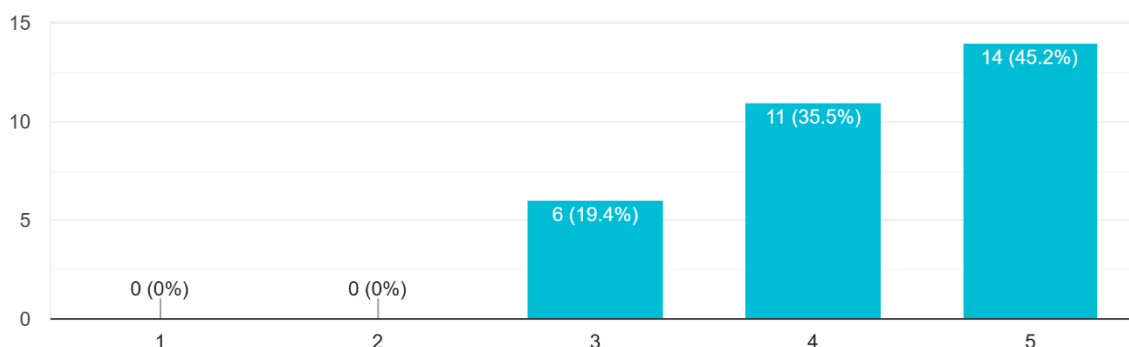


Εικόνα 7.12: Διάγραμμα αντίληψης πολυπλοκότητας μέσω οπτικής αναπαράστασης

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο η οπτική αναπαράσταση της εφαρμογής διευκολύνει την ταυτόχρονη παρακολούθηση πολλών χαρακτηριστικών της μουσικής (ένταση, ρυθμός, όργανα), χρησιμοποιώντας την κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Η οπτική αναπαράσταση διευκολύνει την ταυτόχρονη παρακολούθηση πολλών χαρακτηριστικών της μουσικής (ένταση, ρυθμός, όργανα).

31 responses



Εικόνα 7.13: Διάγραμμα οπτικής παρακολούθησης πολλαπλών χαρακτηριστικών μουσικής

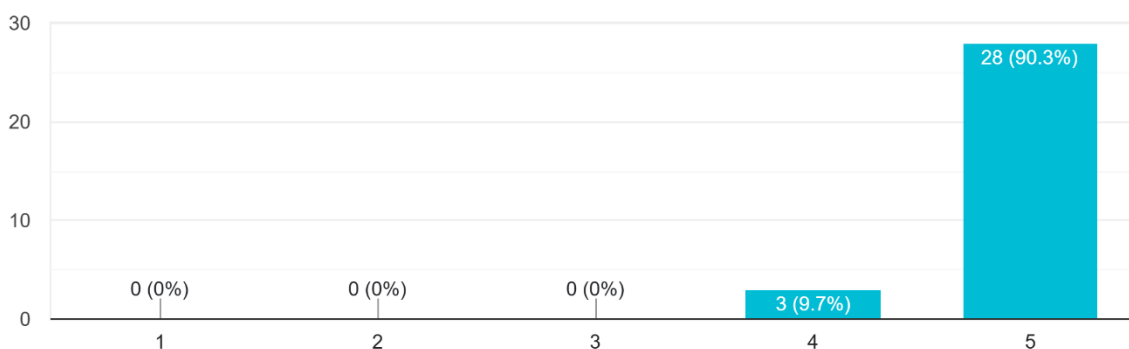
### 7.3.4 Αξιολόγηση Συνδυασμού Οπτικοποίησης με Δονήσεις από Χρήστες – Γενική Εμπειρία

Οι συγκεκριμένες ερωτήσεις είναι πολλαπλών επιλογών με χαμηλότερη βαθμολογία το 1 (Διαφωνώ Απόλυτα) και υψηλότερη βαθμολογία το 5 (Συμφωνώ Απόλυτα).

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο η χρήση της εφαρμογής είναι εύκολη, χρησιμοποιώντας κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Είναι εύκολη η χρήση της εφαρμογής

31 responses

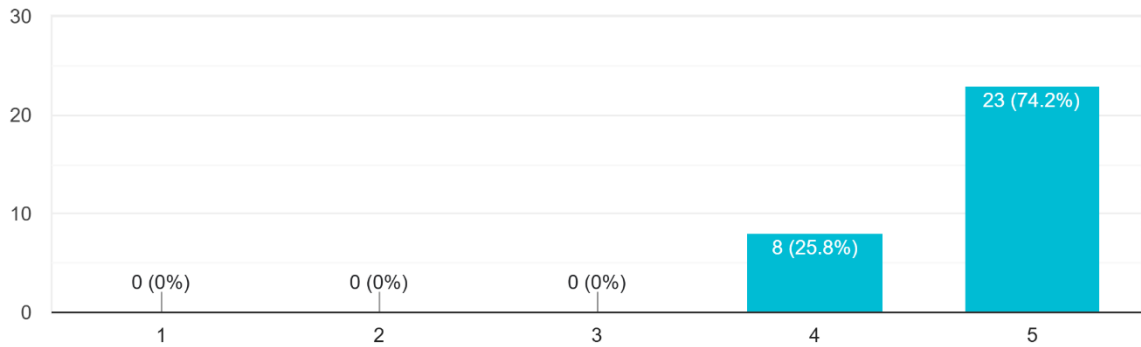


Εικόνα 7.14: Διάγραμμα ευχρηστίας εφαρμογής

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο είναι συνολικά ικανοποιημένοι από την εφαρμογή, χρησιμοποιώντας κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Είμαι ικανοποιημένος/η συνολικά από την εφαρμογή

31 responses

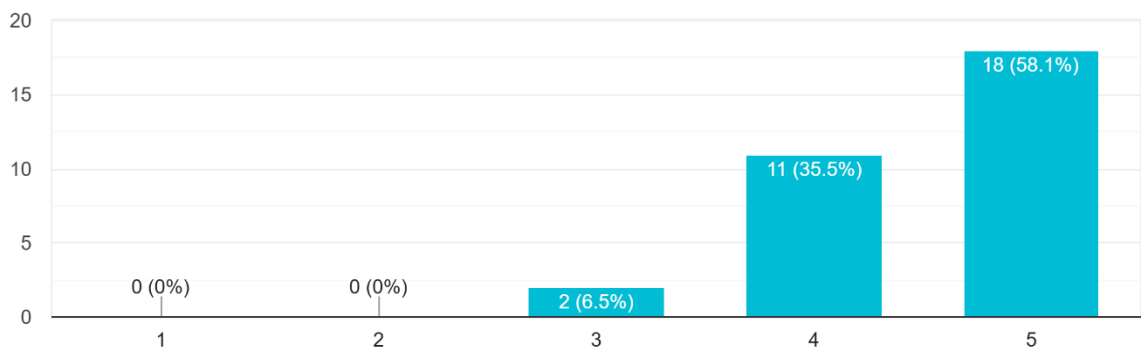


Εικόνα 7.15: Διάγραμμα αξιολόγησης ικανοποίησης εφαρμογής

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο η οπτικοποίηση της εφαρμογής είναι σωστά συγχρονισμένη με τις δονήσεις, χρησιμοποιώντας κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Η οπτικοποίηση είναι συγχρονισμένη σωστά με τις δονήσεις.

31 responses

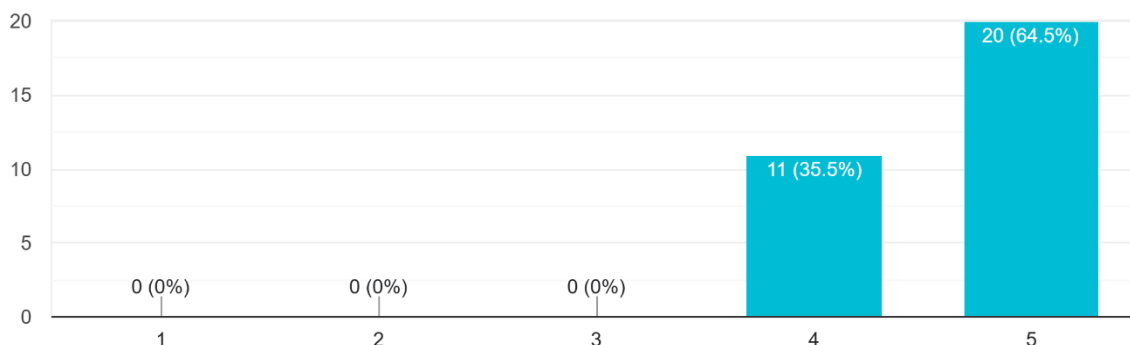


Εικόνα 7.16: Διάγραμμα αξιολόγησης συγχρονισμού οπτικοποίησης–δονήσεων

Στους συμμετέχοντες ζητήθηκε να αξιολογήσουν κατά πόσο θεωρούν ότι η εφαρμογή είναι χρήσιμη για άτομα με προβλήματα ακοής, χρησιμοποιώντας κλίμακα 1–5, τα αποτελέσματα παρουσιάζονται στο παρακάτω διάγραμμα:

Θεωρώ ότι η εφαρμογή είναι χρήσιμη για άτομα με προβλήματα ακοής.

31 responses



Εικόνα 7.17: Διάγραμμα αξιολόγησης χρησιμότητας για άτομα με προβλήματα ακοής

### 7.3.5 Προτάσεις και Παρατηρήσεις

Στο τέλος της αξιολόγησης, ζητήθηκε από τους συμμετέχοντες να καταθέσουν τυχόν προτάσεις ή παρατηρήσεις για σημεία της εφαρμογής που θα μπορούσαν να βελτιωθούν.

Κάποιοι χρήστες έδωσαν τις εξής απαντήσεις:

1. Να συνεχίζονται οι δονήσεις ακόμα και όταν η ένταση της μουσικής είναι μηδενική (mute),
2. Να υπάρχει δυνατότητα αναζήτησης,
3. Βελτίωση της ακρίβειας των δονήσεων,
4. Δυνατότητα λειτουργίας με ηχογράφιση.

## 7.4 Συμπεράσματα

Τα αποτελέσματα δείχνουν ότι η εφαρμογή αξιολογείται πολύ θετικά από τους χρήστες. Οι δονήσεις θεωρούνται ευχάριστες και κατάλληλης διάρκειας, ενώ η οπτική αναπαράσταση κρίνεται καθαρή, εύκολα αναγνωρίσιμη και βοηθητική στην κατανόηση του ρυθμού, της έντασης και άλλων χαρακτηριστικών της μουσικής. Παρά τα θετικά σχόλια, εντοπίστηκαν ορισμένα σημεία που χρήζουν βελτίωσης. Η ακρίβεια των δονήσεων σε σχέση με τον ρυθμό δεν αξιολογήθηκε απόλυτα υψηλά, ενώ η αντίληψη διαφορών στις δονήσεις ανάλογα με το είδος της μουσικής παρουσίασε μικρή ποικιλία στις απαντήσεις. Η γενική εμπειρία χρήσης θεωρήθηκε πολύ ικανοποιητική, με τους περισσότερους συμμετέχοντες να δηλώνουν ότι η εφαρμογή είναι εύκολη στη χρήση και χρήσιμη, ιδιαίτερα για άτομα με προβλήματα ακοής. Οι προτάσεις των χρηστών υποδεικνύουν ότι, αν και η εφαρμογή είναι λειτουργική και ευχάριστη, υπάρχει περιθώριο για βελτιώσεις, όπως η ακρίβεια των δονήσεων, η δυνατότητα αναζήτησης και η δυνατότητα λειτουργίας με ηχογράφιση.

Συνολικά, τα δεδομένα καταδεικνύουν ότι η εφαρμογή είναι επιτυχημένη στην παροχή μιας εναλλακτικής μουσικής εμπειρίας μέσω δονήσεων και οπτικής αναπαράστασης, προσφέροντας ταυτόχρονα υψηλό βαθμό ικανοποίησης στους χρήστες.

## Κεφάλαιο 8ο: Συμπεράσματα και Δυνατότητες Βελτίωσης

Η παρούσα εργασία εστίασε στη δημιουργία μιας εφαρμογής κινητού που μετατρέπει τη μουσική σε δονήσεις και ταυτόχρονα την αποδίδει οπτικά, προσφέροντας στα άτομα με προβλήματα ακοής μια νέα πολυαισθητηριακή εμπειρία. Αρχικά, παρουσιάστηκαν οι βασικές θεωρητικές έννοιες της οπτικοποίησης δεδομένων και της απτικής ανάδρασης, οι οποίες αποτέλεσαν το υπόβαθρο για την ανάπτυξη της εφαρμογής. Μέσα από το δεύτερο και τρίτο κεφάλαιο αναδείχθηκε πώς τα μαθηματικά εργαλεία (Fourier, spectrogram κ.ά.) και οι τεχνολογίες απτικής δόνησης (VCA, LRA, ERM, DMA) συνδυάζονται για να αποδώσουν τη μουσική μέσω της όρασης και της αφής. Στη συνέχεια, στο τέταρτο κεφάλαιο αναλύθηκε η αρχιτεκτονική του Haptic Music Player με Visualizer, όπου φάνηκε η σημασία του συγχρονισμού ανάμεσα σε ήχο, δόνηση και οπτικοποίηση. Ιδιαίτερη έμφαση δόθηκε στις τεχνικές προκλήσεις που σχετίζονται με την απόκριση σε πραγματικό χρόνο, αλλά και στις λύσεις που εφαρμόστηκαν για τη σταθερότητα και την ακριβή απτική αναπαράσταση. Το πέμπτο κεφάλαιο ανέδειξε τη σημασία της πολυαισθητηριακής προσέγγισης για τα άτομα με προβλήματα ακοής, τεκμηριώνοντας νευρολογικά και ψυχοακουστικά πώς οι δονήσεις μπορούν να αποτελέσουν εναλλακτικό μέσο αντίληψης του ρυθμού και της μουσικής. Στο έκτο κεφάλαιο παρουσιάστηκε αναλυτικά ο σχεδιασμός και η υλοποίηση της εφαρμογής, ενώ το έβδομο κεφάλαιο αναλύθηκαν τα αποτελέσματα της αξιολόγησης από χρήστες.

Η αξιολόγηση της εφαρμογής από τους συμμετέχοντες κατέδειξε ότι η συνολική εμπειρία χρήσης θεωρήθηκε ιδιαίτερα θετική. Οι χρήστες χαρακτήρισαν τις δονήσεις ευχάριστες και κατάλληλης διάρκειας, ενώ η οπτική αναπαράσταση εκτιμήθηκε ως καθαρή, εύκολα αναγνωρίσιμη και βοηθητική για την κατανόηση βασικών χαρακτηριστικών της μουσικής, όπως ο ρυθμός και η ένταση. Η εφαρμογή εκτιμήθηκε ως ένα καινοτόμο εργαλείο που καθιστά τη μουσική πιο προσιτή και προσφέρει μια πολυαισθητηριακή εμπειρία. Παρά τα θετικά αποτελέσματα, υπάρχουν περιθώρια βελτίωσης που ανοίγουν τον δρόμο για μελλοντική εξέλιξη:

- **Βελτιστοποίηση συστημάτων δόνησης:** με χρήση μαθηματικών προσομοιώσεων και πιο εξελιγμένων αλγορίθμων, ώστε οι δονήσεις να αποδίδουν πιο πιστά μουσικά χαρακτηριστικά όπως η μελωδία και η χροιά.
- **Προσαρμογή έντασης δόνησης:** ο χρήστης θα μπορούσε να ρυθμίζει το επίπεδο της δόνησης ανάλογα με τις προτιμήσεις του ή την ευαισθησία του, ώστε η εμπειρία να είναι πιο άνετη και εξατομικευμένη.
- **Διαφοροποίηση δονήσεων ανά είδος μουσικής:** ο χρήστης θα μπορούσε να επιλέγει ρύθμιση ανάλογα με τα διαφορετικά μουσικά είδη (π.χ. κλασική, ποπ, ροκ, ηλεκτρονική) και να αποδίδονται δονήσεις με ξεχωριστά μοτίβα που ταιριάζουν καλύτερα σε κάθε ύφος.
- **Εξειδικευμένα μοτίβα δόνησης:** δυνατότητα επιλογής διαφορετικών τρόπων απόδοσης (π.χ. δόνηση μόνο στον ρυθμό, δόνηση σε δυνατά σημεία, δόνηση σε συγκεκριμένο όργανο ή μόνο στη φωνή), ώστε να ενισχυθεί η κατανόηση της μουσικής.

Συνολικά, η εργασία ανέδειξε ότι η τεχνολογία μπορεί να λειτουργήσει ως γέφυρα ανάμεσα στην τέχνη και την ανθρώπινη αντίληψη, δημιουργώντας νέες μορφές πρόσβασης και αλληλεπίδρασης με τη μουσική. Ο συνδυασμός δόνησης και οπτικής αναπαράστασης δεν αποτελεί μόνο μια πρακτική λύση για την ενίσχυση της εμπειρίας ατόμων με προβλήματα ακοής, αλλά ανοίγει τον δρόμο για μια

πιο πολυαισθητηριακή προσέγγιση που μπορεί να εμπλουτίσει την απόλαυση και για το ευρύτερο κοινό. Η αξιοποίηση μεθόδων όπως η οπτικοποίηση δεδομένων και η απτική ανάδραση δείχνει πώς η καινοτομία επεκτείνει τα όρια της κατανόησης και της καλλιτεχνικής έκφρασης, προσφέροντας νέες δυνατότητες σε πεδία που μέχρι πρόσφατα έμοιαζαν περιορισμένα. Οι μελλοντικές εξελίξεις υπόσχονται ακόμη πιο εξελιγμένα εργαλεία, ικανά να ενισχύσουν την πολυδιάστατη εμπειρία της μουσικής και να συμβάλουν στη γενικότερη προσπάθεια για μεγαλύτερη προσβασιμότητα και συμπερίληψη. Συνολικά, η εργασία έδειξε ότι η μουσική μπορεί να γίνει προσβάσιμη μέσω της τεχνολογίας, ενώ οι μελλοντικές βελτιώσεις υπόσχονται μια ακόμη πιο ολοκληρωμένη και εξατομικευμένη εμπειρία, που θα ενισχύσει την απόλαυση και τη σύνδεση των χρηστών με τη μουσική.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] T. Hermann, A. Hunt, and J. G. Neuhoff, *The Sonification Handbook*. Berlin, Germany: Logos Publishing House, 2011.
- [2] S. C. Nanayakkara, E. Taylor, L. Wyse, and S. H. Ong, "An enhanced musical experience for the deaf: Design and evaluation of a music display and a haptic chair," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2009.
- [3] C. Cannam, C. Landone, M. B. Sandler, and J. P. Bello, "The Sonic Visualiser: A Visualization Platform for Semantic Descriptors from Musical Signals," 2006. [Ηλεκτρονικό]. <http://www.sonicvisualiser.org>
- [4] L. Wang, G. Sun, Y. Wang, J. Ma, X. Zhao, and R. Liang, "AFExplorer: Visual Analysis and Interactive Selection of Audio Features," *Visual Informatics*, vol. 6, no. 1, pp. 47–55, 2022. [Ηλεκτρονικό]. <http://doi.org/10.1016/j.visinf.2022.02.003>
- [5] M. Brehmer και T. Munzner, "A Multi-Level Typology of Abstract Visualization Tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2376–2385, 2013. <http://doi.org/10.1109/TVCG.2013.124>
- [6] H. B. Lima, C. G. R. Dos Santos, and B. S. Meiguins, "A Survey of Music Visualization Techniques," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, art. 143, pp. 1–29, Jul. 2021. <https://doi.org/10.1145/3461835>
- [7] P. Ciuha, B. Klemenc, and F. Solina, "Visualization of concurrent tones in music with colours," in *Proceedings of the 18th ACM International Conference on Multimedia*, Firenze, Italy, pp. 1677–1680, Oct. 2010. <https://doi.org/10.1145/1873951.1874320>
- [8] R. Gold and O. Segal, "The Bouba–Kiki Effect in Persons with Prelingual Auditory Deprivation," *Language Learning and Development*, vol. 16, no. 1, pp. 1–12, 2019. [Ηλεκτρονικό]. <http://doi.org/10.1080/15475441.2019.1685386>
- [9] W.-U. Jeong and S.-H. Kim, "Synesthesia visualization of music waveform: 'Kinetic lighting for music visualization'," *Int. J. Asia Digital Art Des. Assoc.*, vol. 23, no. 2, pp. 22–27, 2019.
- [10] J. B. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235–238, 1977. [Ηλεκτρονικό]. <http://doi.org/10.1109/TASSP.1977.1162950>
- [11] G. Burdea, *Force and Touch Feedback for Virtual Reality*. New York, NY, USA: Wiley-Interscience, 1996.
- [12] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nature Reviews Neuroscience*, vol. 10, no. 5, pp. 345–359, 2009.

- [13] R. J. Stone, "Haptic feedback: A brief history from telepresence to virtual reality," *Lecture Notes in Computer Science*, vol. 2058, pp. 1–16, Jan. 2000. [https://doi.org/10.1007/3-540-44589-7\\_1](https://doi.org/10.1007/3-540-44589-7_1)
- [14] N. Kastor, B. Dandu, V. Bassari, G. Reardon, and Y. Visell, "Ferrofluid electromagnetic actuators for high-fidelity haptic feedback," *Sensors and Actuators A: Physical*, vol. 355, art. 114252, 2023. [Ηλεκτρονικό]. <https://doi.org/10.1016/j.sna.2023.114252>
- [15] Power Electronic Tips, "Voice-coil actuators, Part 2: Application issues," *Power Electronic Tips*, Jun. 25, 2018. [Online]. Available: <https://www.powerelectronicstips.com/voice-coil-actuators-part-2-application-issues-faq/>
- [16] J. I. C. Lindsay, I. Jiang, E. Larson, R. Adams, S. N. Patel, and B. Hannaford, "Good vibrations: an evaluation of vibrotactile impedance matching for low power wearable applications," in *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*, pp. 515–520, 2013. <http://dx.doi.org/10.1145/2501988.2502051>
- [17] Power Electronic Tips, "Haptics components, part 1: LRA, ERM, and piezo actuators," *Power Electronic Tips*, Jun. 2, 2021. [Ηλεκτρονικό]. <https://www.powerelectronicstips.com/haptics-components-pt-1-lra-erm-and-piezo-actuators/>
- [18] S. Brewster, F. Chohan, and L. Brown, "Tactile feedback for mobile interactions," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2007)*, pp. 159–162, 2007. <https://doi.org/10.1145/1240624.1240649>
- [19] E. Hoggan, A. Crossan, S. A. Brewster, and T. Kaaresoja, "Audio or tactile feedback: Which modality when?," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2009)*, pp. 2253–2256, 2009. <https://doi.org/10.1145/1518701.1519045>
- [20] X. Zhou, S. Wu, X. Wang, Z. Wang, Q. Zhu, J. Sun, P. Huang, X. Wang, W. Huang, and L. Qianbo, "Review on piezoelectric actuators: Materials, classifications, applications, and recent trends," *Frontiers of Mechanical Engineering*, vol. 19, no. 1, 2024. <https://doi.org/10.1007/s11465-023-0772-0>
- [21] C. Jehle, "Piezo Haptic Actuators for Touch Applications," *EETPower – Technical Articles*, Sep. 20, 2019. [Online]. Available: <https://www.eepower.com/technical-articles/piezo-haptic-actuators-for-touch-applications/>
- [22] M. S. Hashem, A. Raza, S. E. Shan, and S. Jeon, "Pneumatic multi-mode silicone actuator with pressure, vibration, and cold thermal feedback," *arXiv preprint*, 2025. <https://arxiv.org/abs/2503.22247>
- [23] O. Schneider, K. E. MacLean, C. Swindells, and K. Booth, "Haptic experience design: What hapticians do and where they need help," *International Journal of Human-Computer Studies*, vol. 107, pp. 5–21, 2017. <https://doi.org/10.1016/j.ijhcs.2017.04.004>
- [24] A. Akshita, *Multimodal Affective Feedback: Role of Haptics*, M.S. thesis, Dept. of Computer Science and Engineering,, July 2015.

- [25] M.-H. Choi, K.-B. Kim, Y.-J. Kim, J.-S. Kim, H.-S. Kim, and S.-C. Chung, "Study on the cognitive characteristics induced by changes in the intensity, frequency and duration of vibratory stimuli," *Behavioral Sciences*, vol. 13, no. 5, p. 350, Apr. 2023.
- [26] P. Laitinen and J. Mawnpaa, "Enabling mobile haptic design: Piezoelectric actuator technology properties in hand held devices," in *Proc. IEEE International Workshop on Haptic Audio Visual Environments and their Applications (HAVE)*, 2006. [Ηλεκτρονικό]. <http://doi.org/10.1109/HAVE.2006.283787>
- [27] N. Yeganeh, I. Makarov, Á. Kristjánsson, and R. Unnthorsson, "Discrimination accuracy of sequential versus simultaneous vibrotactile stimulation on the forearm," *Applied Sciences*, vol. 14, no. 1, p. 43, Dec. 2023. [Ηλεκτρονικό]. <https://doi.org/10.3390/app14010043>
- [28] T. L. White and A. S. Krausman, "Effects of inter-stimulus interval and intensity on the perceived urgency of tactile patterns," *Applied Ergonomics*, vol. 48, pp. 121–129, May 2015. [Ηλεκτρονικό]. <http://doi.org/10.1016/j.apergo.2014.11.010>
- [29] P. N. Juslin και J. A. Sloboda, "Music and emotion," σε D. Deutsch, επιμ., *The Psychology of Music*, 3rd ed. San Diego, CA, USA: Academic Press, 2013, pp. 583–645. <https://doi.org/10.1016/B978-0-12-381460-9.00015-8>
- [30] B. Remache-Vinueza, A. Trujillo-León, M. Zapata, F. Sarmiento-Ortiz, και F. Vidal-Verdú, "Audio-Tactile Rendering: A Review on Technology and Methods to Convey Musical Information through the Sense of Touch," *Sensors*, vol. 21, no. 19, Art. 6575, 2021. [Ηλεκτρονικό]. <https://doi.org/10.3390/s21196575>
- [31] J. B. F. van Erp και A. Toet, "Social touch in human–computer interaction," *Frontiers in Digital Humanities*, vol. 2, Art. 2, 2015. [Ηλεκτρονικό]. <http://doi.org/10.3389/fdigh.2015.00002>
- [32] K. O. Johnson, "The roles and functions of cutaneous mechanoreceptors," *Current Opinion in Neurobiology*, vol. 11, no. 4, pp. 455–461, 2001.
- [33] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, A. J. Hudspeth, και S. Mack, *Principles of Neural Science*, 5th ed. New York, NY, USA: McGraw-Hill Education, 2014. <https://neurology.mhmedical.com/content.aspx?bookid=1049&sectionid=59138139>
- [34] T. Honda, T. Baba, και M. Okamoto, "Ontenna: Design and social implementation of auditory information transmission devices using tactile and visual senses," σε *Computers Helping People with Special Needs*, Lecture Notes in Computer Science, vol. 130–138, Springer, 2022. [https://doi.org/10.1007/978-3-031-08645-8\\_16](https://doi.org/10.1007/978-3-031-08645-8_16)
- [35] S. J. Bensmaïa, "Tactile intensity and population codes," *Behavioural Brain Research*, vol. 190, no. 2, pp. 165–173, 2008.
- [36] S. S. Stevens, *Psychophysics: Introduction to its perceptual, neural, and social prospects*. New York, NY, USA: Wiley, 1975.

- [37] H. Fletcher και W. A. Munson, "Loudness, its definition, measurement and calculation," *Journal of the Acoustical Society of America*, vol. 5, pp. 82–108, 1933. <https://doi.org/10.1121/1.1915637>
- [38] Y. Suzuki, H. Takeshima, και K. Kurakata, "Revision of ISO 226 'Normal Equal-Loudness-Level Contours' from 2003 to 2023 edition: The background and results," *Acoustical Science and Technology*, vol. 45, no. 1, pp. 1–8, 2024. <https://doi.org/10.1250/ast.e23.66>
- [39] G. A. Gescheider, *Psychophysics: The Fundamentals*. Hove, UK: Psychology Press, 1997.
- [40] A. M. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery," *Industrial Robot*, vol. 31, no. 6, pp. 499–508, 2004.
- [41] V. G. Chouvardas, A. N. Miliou, και M. K. Hatalis, "Tactile displays: Technologies and applications," *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 412–428, 2008.
- [42] T. H. Massie και J. K. Salisbury, "The PHANToM haptic interface: A device for probing virtual objects," σε *ASME Winter Annual Meeting*, 1994.
- [43] J. E. Colgate και J. M. Brown, "Factors affecting the Z-Width of a haptic display," σε *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1994, pp. 3205–3210.
- [44] V. Hayward, O. R. Astley, M. Cruz-Hernandez, D. Grant, και G. Robles-De-La-Torre, "Haptic interfaces and devices," *Sensor Review*, vol. 24, no. 1, pp. 16–29, 2004. <https://doi.org/10.1108/02602280410515770>
- [45] R. B. Gillespie και M. R. Cutkosky, "Stable user-specific haptic rendering," σε *Proceedings of the ASME International Mechanical Engineering Congress and Exposition (IMECE)*, 1996. <http://doi.org/10.1115/IMECE1996-0362>
- [46] R. J. Adams και B. Hannaford, "Stable haptic interaction with virtual environments," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 465–474, 1999. <https://doi.org/10.1109/70.768179>
- [47] B. Hannaford και J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002. <https://doi.org/10.1109/70.988969>
- [48] A. Bicchi, E. P. Scilingo, και D. De Rossi, "Haptic discrimination of softness in teleoperation: The role of the contact area spread rate," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 496–504, 2000. <https://doi.org/10.1109/70.880800>
- [49] S. J. Bensmaïa και M. Hollins, "Pacian representations of fine surface texture," *Perception & Psychophysics*, vol. 67, no. 5, pp. 842–854, 2005.
- [50] M. D. Fletcher, "Can haptic stimulation enhance music perception in hearing-impaired listeners?" *Frontiers in Neuroscience*, vol. 15, 2021.

- [51] A. Kumar, A. K. Singh, A. Kapoor, και A. Majumder, "A haptic wearable for translating music into tactile signals," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3274–3282, 2021. <https://doi.org/10.1108/02602280410515770>
- [52] J. B. J. Fourier, *Théorie analytique de la chaleur*. Paris, France: Firmin Didot, 1822.
- [53] J. W. Cooley και J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965. <https://doi.org/10.2307/2003354>
- [54] R. V. Moora και G. Prabhakar, "Tactile Melodies: A Desk-Mounted Haptic Music Player for Perceiving Musical Experiences," *arXiv*, 2024. [Ηλεκτρονικό]. <https://doi.org/10.48550/arXiv.2408.06449>
- [55] M. Barron, "Developments in concert hall acoustics in the 1960s: Theory and practice," *Acoustics*, vol. 1, no. 3, Art. 30, 2019. <https://doi.org/10.3390/acoustics1030030>
- [56] T. Venkatesan και Q. J. Wang, "Feeling connected: The role of haptic feedback in VR concerts and the impact of haptic music players on the music listening experience," *Arts*, vol. 12, no. 4, Art. 148, 2023. <https://doi.org/10.3390/arts12040148>
- [57] I. Hwang, H. Lee, και S. Choi, "Real-time dual-band haptic music player for mobile devices," *IEEE Transactions on Haptics*, vol. 6, no. 3, pp. 340–351, 2013. <https://doi.org/10.1109/TOH.2013.7>
- [58] R. L. Klatzky, S. J. Lederman, και C. Reed, "There's more to touch than meets the eye: The salience of object attributes for haptics with and without vision," *Journal of Experimental Psychology: General*, vol. 116, no. 4, pp. 356–369, 1987. <https://doi.org/10.1037/0096-3445.116.4.356>
- [59] E. Pezent, B. Cambio, και M. K. O'Malley, "Syntacts: Open-source software and hardware for audio-controlled haptics," *IEEE Transactions on Haptics*, vol. 14, no. 1, pp. 225–233, 2021. <https://doi.org/10.1109/TOH.2020.3002696>
- [60] S. Oxnard, E. Stanhope, L. J. Hobden, και M. Masri, "Characterisation and excursion modelling of audio haptic transducers," σε *Proceedings of the 27th International Conference on Digital Audio Effects (DAFx-24)*, Guildford, United Kingdom, 2024.
- [61] S. Choi και K. J. Kuchenbecker, "Vibrotactile display: Perception, technology, and applications," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 2093–2104, 2013. <https://doi.org/10.1109/JPROC.2012.2221071>
- [62] J. Li, H. Lin, και L. Zhou, "Emotion tag based music retrieval algorithm," σε *Proceedings of the Sixth Asia Information Retrieval Symposium (AIRS 2010)*, Springer, 2010, pp. 599–609. [https://doi.org/10.1007/978-3-642-17187-1\\_56](https://doi.org/10.1007/978-3-642-17187-1_56)

- [63] D. Cavdir, F. Ganis, R. Paisa, P. Williams, και S. Serafin, "Multisensory Integration Design in Music for Cochlear Implant Users," σε R. Michon, L. Pottier, και Y. Orlarey, επιμ., *SMC/JIM/IFC 2022 - Proceedings of the 19th Sound and Music Computing Conference*, Sound and Music Computing Network, 2022, pp. 334–341. <https://doi.org/10.5281/zenodo.6573315>
- [64] S. A. Whitton and F. Jiang, "Sensorimotor synchronization with visual, auditory, and tactile modalities," *Psychological Research*, 2023. [Ηλεκτρονικό]. <http://doi.org/10.1007/s00426-023-01801-3>.
- [65] N. Almeida, A. Teixeira, S. Silva, και M. Ketsmur, "The AM4I Architecture and Framework for Multimodal Interaction and Its Application to Smart Environments," *Sensors*, vol. 19, no. 11, 2019. [Ηλεκτρονικό]. <https://doi.org/10.3390/s19112587>
- [66] S. Lin, B. Rizk, M. Jun, A. Artze, C. Sullivan, S. Mozgai, and S. Fisher, "Estuary: A framework for building multimodal low-latency real-time socially interactive agents," *Proceedings of the 24th ACM International Conference on Intelligent Virtual Agents (IVA)*, 2024. [Ηλεκτρονικό]. <https://doi.org/10.1145/3652988.3696198>
- [67] T. Baier, S. B. Santamaría, and P. Vossen, "A modular architecture for creating multimodal agents," *arXiv*, Jun. 1, 2022. [Online]. <https://arxiv.org/abs/2206.00636>
- [68] S. Jordà, M. Kaltenbrunner, G. Geiger, and M. Alonso, "The reacTable: a tangible tabletop musical instrument and collaborative workbench," *ACM SIGGRAPH 2006 Sketches*, Article 91, Jan. 2006. [Ηλεκτρονικό]. <https://doi.org/10.1145/1179849.1179963>
- [69] bHaptics, "Audio-to-Haptics," *bHaptics*, 2025. [Ηλεκτρονικό]. <https://www.bhaptics.com/games/audio-to-haptic/>
- [70] B. E. Stein και T. R. Stanford, "Multisensory integration: Current issues from the perspective of the single neuron," *Nature Reviews Neuroscience*, vol. 9, no. 4, pp. 255–266, 2008.
- [71] G. A. Calvert, C. Spence, και B. E. Stein, *The Handbook of Multisensory Processes*. Cambridge, MA, USA: MIT Press, 2004.
- [72] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>
- [73] A. Pascual-Leone και R. Hamilton, "The metamodal organization of the brain," *Progress in Brain Research*, vol. 134, pp. 427–445, 2001.. [https://doi.org/10.1016/S0079-6123\(01\)34028-1](https://doi.org/10.1016/S0079-6123(01)34028-1)
- [74] P. Dayan και L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA, USA: MIT Press, 2001.
- [75] J. Driver και T. Noesselt, "Multisensory interplay reveals crossmodal influences on 'sensory-specific' brain regions, neural responses, and judgments," *Neuron*, vol. 57, no. 1, pp. 11–23, 2008. <https://doi.org/10.1016/j.neuron.2007.12.013>

- [76] A. Kral and A. Sharma, "Crossmodal plasticity in hearing loss," *Trends in Neurosciences*, vol. 46, no. 5, pp. 377–393, May 2023. [Ηλεκτρονικό]. <http://doi.org/10.1016/j.tins.2023.02.004>
- [77] Q. Wang, Q. Chen, P. Liu, J. Zhang, L. Zhou, και L. Peng, "Functional magnetic resonance imaging reveals early connectivity changes in the auditory and vestibular cortices in idiopathic sudden sensorineural hearing loss with vertigo: A pilot study," *Frontiers in Human Neuroscience*, vol. 15, 2021. <https://doi.org/10.3389/fnhum.2021.719254>
- [78] R. Quian Quiroga and S. Panzeri, *Principles of Neural Coding*, 1st ed, 2013. [Ηλεκτρονικό]. <http://doi.org/10.1201/b14756>
- [79] X. Wang, T. Lu, D. Bendor, and E. Bartlett, "Neural coding of temporal information in auditory thalamus and cortex," *Neuroscience*, vol. 154, no. 1, pp. 294–303, Jul. 2008. [Ηλεκτρονικό]. <http://doi.org/10.1016/j.neuroscience.2008.03.065>
- [80] P. A. Cariani, "Temporal codes and computations for sensory representation and scene analysis," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1100–1111, Sep. 2004. [Ηλεκτρονικό]. Available: <http://doi.org/10.1109/TNN.2004.833305>
- [81] L. Turchet, R. Rosaia, A. Diodati, και M. Carner, "Exposure to vibrotactile music improves audiometric performances in individuals with cochlear implants," *Scientific Reports*, vol. 15, Art. 10456, 2025. <https://doi.org/10.1038/s41598-025-02946-4>
- [82] C. Bernard, J. Monnoyer, M. Wiertelowski, and S. Ystad, "Rhythm perception is shared between audio and haptics," *Scientific Reports*, vol. 12, no. 1, Art. no. 4188, Mar. 2022. [Ηλεκτρονικό]. <https://doi.org/10.1038/s41598-022-08152-w>
- [83] L. Zhan, Y. Huang, Z. Guo, J. Yang, L. Gu, S. Zhong, and X. Wu, "Visual over auditory superiority in sensorimotor timing under optimized condition," *Frontiers in Psychology*, vol. 13, p. 1048943, 2022. [Ηλεκτρονικό]. <https://doi.org/10.3389/fpsyg.2022.1048943>
- [84] V. de Lafuente, M. Jazayeri, H. Merchant, O. García-Garibay, J. Cadena-Valencia, and A. M. Malagón, "Keeping time and rhythm by internal simulation of sensory stimuli and behavioral actions," *Science Advances*, vol. 10, no. 2, 2024. [Ηλεκτρονικό]. <https://doi.org/10.1126/sciadv.adh8185>
- [85] E. Frid and H. Lindetorp, "Haptic Music: Exploring Whole-Body Vibrations and Tactile Sound for a Multisensory Music Installation," *Proceedings of the International Sound and Music Computing Conference*, Torino, Italy, Jun 2020.
- [86] L. Turchet, T. West, and M. M. Wanderley, "Touching the audience: Musical haptic wearables for augmented and participatory live music performances," *Personal and Ubiquitous Computing*, vol. 25, pp. 749–769, 2021. <http://dx.doi.org/10.1007/s00779-020-01395-2>
- [87] P. Tranchant, M. M. Shiell, M. Giordano, A. Nadeau, I. Peretz, και R. J. Zatorre, "Feeling the Beat: Bouncing Synchronization to Vibrotactile Music in Hearing and Early Deaf People," *Frontiers in Neuroscience*, Sec. Auditory Cognitive Neuroscience, vol. 11, Art. 507, 12-Sep-2017. [Online]. Available: <http://doi.org/10.3389/fnins.2017.00507>

- [88] M. Cerna and A. F. Harvey, The Fundamentals of FFT-Based Signal Analysis and Measurement, National Instruments, Application Note 041, July 2000. [Ηλεκτρονικό]. <https://www.ni.com/pdf/manuals/371226a.pdf>