



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΜΑΤΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

Βασάλος Γιώργος

Επιβλέποντες: **Καμαλάκης Θωμάς**
Δαλάκας Βασίλειος
Μιχαλακέλης Χρήστος

ΑΘΗΝΑ

ΙΟΥΛΙΟΣ 2016

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

**Βασάλος Γιώργος
ΑΜ: 21103**

ΕΠΙΒΛΕΠΟΝΤΕΣ:

**Καμαλάκης Θωμάς
Δαλάκας Βασίλειος
Μιχαλακέλης Χρήστος**

Περίληψη

Ο ήχος παίζει ένα μεγάλο ρόλο στην καθημερινή μας ζωή. Μέσω αυτού μπορούμε να αντιλαμβανόμαστε καλύτερα το περιβάλλον μας και πολλές φορές τον χρησιμοποιούμε για να κατανοήσουμε βαθύτερα τις περίπλοκες ιδιότητες του κόσμου γύρω μας. Το πιο χαρακτηριστικό παράδειγμα χρήσης του ήχου από τους ανθρώπους είναι η δημιουργία μουσικής ως μια αρμονική αλληλουχία ήχων.

Η μουσική ήταν ανέκαθεν και συνεχίζει να είναι ένα σημαντικό κομμάτι στη ζωή κάθε ανθρώπου στο σύγχρονο, ακόμα και στον αρχαίο πολιτισμό. Ως μια μορφή τέχνης, έχει την ιδιότητα να την αντιλαμβάνεται διαφορετικά και μοναδικά κάθε άτομο. Μέσω αυτής, πολλοί καλλιτέχνες εκφράζονται και προσπαθούν να μοιραστούν μηνύματα, συναισθήματα και εμπειρίες της ζωής τους με άλλους ανθρώπους κοινής ιδιοσυγκρασίας.

Με την εξέλιξη της τεχνολογίας ο διαμοιρασμός της μουσικής γίνεται ολοένα και πιο εύκολος. Η πρόσβαση στη μουσική κάθε καλλιτέχνη/συγκροτήματος είναι εφικτή από το internet με μικρό ή μηδενικό αντάλλαγμα. Επίσης η παραγωγή μουσικής είναι πλέον πολύ πιο εύκολη με χρήση εξειδικευμένων μηχανημάτων (π.χ. μικρόφωνο, κάρτα ήχου, κονσόλα) και λογισμικού (π.χ. λογισμικό ηχογράφησης, επεξεργασίας).

Η παρούσα πτυχιακή στοχεύει στην ανάπτυξη λογισμικού για την αποτύπωση της μουσικής, με χρήση κάποιου ειδικού εξοπλισμού, μέσω ηλεκτρονικών υπολογιστών. Απώτερος σκοπός είναι η χρήση της εφαρμογής από μουσικούς – συνθέτες για να μπορούν γρήγορα και εύκολα να αποτυπώνουν τις ιδέες τους σε κάποιο αρχείο απλά αναπαράγοντας και ηχογραφώντας τη μελωδία.

Συγκεκριμένα, η εφαρμογή δέχεται ηχητικό σήμα από την προεπιλεγμένη θήρα του υπολογιστή και το ηχογραφεί. Στη συνέχεια, αναλύει αυτό το σήμα και καταλήγει στη δημιουργία ενός αρχείου που αναπαριστά, με τρόπο κατανοητό για το μουσικό, την ηχογραφημένη μελωδία.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Επεξεργασία σήματος ήχου, εφαρμογή Python

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Επεξεργασία Σήματος, Σήματα Ήχου, Μουσική, Ανάλυση Fourier, Python

Abstract

Sound plays an important role in our every-day lives. Through it we can better perceive our surroundings and many times we use it for the deeper understanding of the world around us. The most characteristic example of the use of sound from humans is the creation of music as a harmonic sequence of sounds.

Music has always been and still is an important piece in the life of every human in the modern, and even in the ancient civilisation. As a form of art, each person perceives it in a different and unique way. Through it, many artists express themselves and try to share messages, emotions and experiences of their lives with other people who have similar temperament.

With the constant evolution of technology, sharing music is becoming easier and easier. Reaching the music of any artist or band is possible using the internet with a low or zero price. In addition, music production is nowadays much easier with the use of specified hardware (e.g. microphone, sound card, console) and software (e.g. recording, editing software).

This thesis aims at the development of a software that imprints music, with the use of specified hardware, via a computer. The ultimate goal is the use of the application by musicians – composers so they can easily and quickly imprint their ideas to a file just by reproducing and recording the melody.

Specifically, the application takes sound signal as the input from the default computer slot and records it. Consequently, it analyses the given signal and ends up creating a file which represents, in a way understandable by the musician, the recorded melody.

SUBJECT AREA: Sound signal analysis, Python application

KEYWORDS: Signal Analysis, Sound Signal, Music, Fourier Analysis, Python

Ευχαριστίες

Η παρούσα πτυχιακή εργασία διεκπεραιώθηκε με τη συμβολή κάποιων φίλων και συνεργατών. Κάθε ένας με βοήθησε να αντιμετωπίσω δυσκολίες που βρέθηκαν και να συνεχίσω να προσπαθώ να βελτιώνω και να ολοκληρώσω την εργασία και γενικά τις προπτυχιακές σπουδές μου. Θα ήθελα λοιπόν να ευχαριστήσω θερμά αυτά τα άτομα για το χρόνο που αφιέρωσαν ώστε να γίνω εγώ καλύτερος.

Αρχικά, θα ήθελα να ευχαριστήσω τους καθηγητές που είχα όλα αυτά τα χρόνια και όλο το προσωπικό του τμήματος Πληροφορικής & Τηλεματικής του Χαροκοπείου Πανεπιστημίου Αθηνών, οι οποίοι μου μετέδωσαν πολλές γνώσεις και όποτε είχα ανάγκη ήταν εκεί για να με συμβουλέψουν με υπομονή και αφοσίωση.

Εν συνεχεία θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της πτυχιακής μου εργασίας, κύριο Καμαλάκη Θωμά, κυρίως για την εμπιστοσύνη που μου έδειξε επιτρέποντάς μου να αναλάβω το θέμα που ήθελα και για την υπομονή που έδειξε από τη στιγμή που ξεκίνησα έως ότου τελείωσα.

Τέλος, θα ήθελα να ευχαριστήσω τους κοντινούς μου ανθρώπους, συγγενείς και φίλους, για την κατανόησή τους όλο το διάστημα των προπτυχιακών μου σπουδών και κυρίως την περίοδο εκπόνησης της πτυχιακής εργασίας

Περιεχόμενα

1 Εισαγωγή	12
1.1 Σκοπός της πτυχιακής εργασίας	12
1.1.1 Παρόμοιες εφαρμογές	13
1.2 Σημειογραφία	13
1.2.1 Παλαιότερες τεχνικές	13
1.2.2 Πεντάγραμμο	14
1.2.3 Ταμπλατούρα	14
1.3 Python και ανάλυση ήχου	15
1.3.1 Python	15
1.3.1.1 Ιστορική εξέλιξη	15
1.3.1.2 Βασικά χαρακτηριστικά	16
1.3.2 Ανάλυση ήχου	18
1.4 Δομή της εργασίας	18
2 Θεωρητικό Υπόβαθρο	20
2.1 Μουσική Θεωρία	20
2.1.1 Ρυθμός	20
2.1.1.1 Ρυθμική αγωγή	20
2.1.1.2 Αξίες	21
2.1.1.3 Μέτρα	22
2.1.1.4 Μετρικός σπλισμός	23
2.1.2 Τόνος	23
2.1.2.1 Συχνότητα	23
2.1.2.1.1 Κούρδισμα	24
2.1.2.2 Αρμονικοί	24
2.1.2.2.1 Χροιά	25

2.2	Μαθηματική θεωρία	25
2.2.1	Μετασχηματισμός Fourier	25
2.2.1.1	DFT (Descrete Fourier Transform)	26
2.2.1.1.1	FFT (Fast Fourier Transform)	26
2.2.1.2	Φασματική Ανάλυση	27
3	Σχεδιασμός	29
3.1	Αρχιτεκτονική Εφαρμογής	29
3.1.1	Ροή σήματος ήχου	29
3.1.1.1	Ηχογράφηση	30
3.1.1.2	Επεξεργασία	31
3.1.1.2.1	Ρυθμική ανάλυσης	31
3.1.1.2.2	Συχνотική ανάλυση	31
3.1.2	Μετρονόμος	31
3.1.3	Κουρδιστήρι	32
3.2	Εργαλεία υλοποίησης της εφαρμογής	32
3.2.1	Βιβλιοθήκες Python	32
3.2.1.1	PyAudio	33
3.2.1.2	SciPy	33
3.2.1.3	NumPy	33
3.2.2	Γραφικό περιβάλλον	34
3.2.2.1	PyQt	34
3.2.2.1.1	Qt Designer	34
4	Υλοποίηση	35
4.1	Λειτουργικότητα	35
4.1.1	Metronome	35
4.1.2	Recorder	36
4.1.3	Rhythm_analyser	36
4.1.4	FFT_tuner	39
4.1.5	Frequency_analyser	40
4.2	Οργάνωση	41

4.2.1	InstaScore_project	41
4.3	Γραφικό περιβάλλον	41
4.3.1	Παραγόμενες κλάσεις	41
4.3.2	Παραμετροποιημένες υποκλάσεις	41
4.4	Έξοδος	42
4.4.1	Extractor	42
5	Αποτελέσματα	43
5.1	Παραδείγματα εκτέλεσης	43
5.1.1	Σύγκριση αναμενόμενης – πραγματικής εξόδου	43
5.2	Δυσκολίες κατά την ανάπτυξη & προβλήματα	44
5.3	Συμπεράσματα	44
6	Συμπεράσματα	46
6.1	Συμπεράσματα	46
6.2	Μελλοντικές επεκτάσεις	46
	Βιβλιογραφία	48

Κατάλογος Σχημάτων

2.1 Τα σύμβολα των αξιών για νότες και παύσεις και οι διάρκειά τους	21
2.2 Αξία παρεστιγμένης νότας	22
2.3 Συνάρτηση μετασχηματισμού Fourier	26
2.4 Συνάρτηση DFT	26
2.5 Φασματόγραμμα σειρήνας	27
3.1 Ροή του ηχητικού σήματος κατά τη διάρκεια ηχογράφησης και επεξεργασίας	30
4.3 Αποτελέσματα μετά το πρώτο στάδιο της ρυθμικής ανάλυσης	37
4.4 Αποτελέσματα μετά το δεύτερο στάδιο της ρυθμικής ανάλυσης	38
5.1 White Stripes – Seven Nation Army	43
5.2 Deep Purple – Smoke On The Water	43
5.3 Star Wars – Imperial March	44

Κεφάλαιο 1

Εισαγωγή

Η μουσική πλέον αποτελεί αναπόσπαστο κομμάτι της ζωής κάθε ανθρώπου στη σύγχρονη κοινωνία. Θα ήταν δύσκολο έως ακατόρθωτο να φανταστεί κάποιος τη ζωή του δίχως αυτή. Με τη βοήθεια της τεχνολογίας, και συγκεκριμένα του internet, οι μουσικοί μπορούν χωρίς κόπο να μοιραστούν τη μουσική τους με όλο τον κόσμο. Επίσης, τη σημερινή εποχή είναι πολύ πιο εύκολη η ηχογράφηση και η επεξεργασία του ήχου με τη χρήση ειδικού εξοπλισμού, όπως τα μικρόφωνα και οι κονσόλες. Ένα πράγμα που έχει μείνει πρακτικά ίδιο από τότε που ξεκίνησε να γράφεται μουσική είναι η διαδικασία αποτύπωσής της. Ο μουσικός είναι αναγκασμένος, ακόμα και με την τόση εξέλιξη της τεχνολογίας, να γράφει την μελωδία που σκέφτεται στο χέρι, είτε σε χαρτί είτε σε υπολογιστή.

1.1 Σκοπός της πτυχιακής εργασίας

Ο σκοπός αυτής της εργασίας είναι η ανάπτυξη μιας εφαρμογής με την οποία οι μουσικοί θα μπορούν εύκολα να αποτυπώσουν τις ιδέες τους σε γνώριμη και επεξεργάσιμη μορφή. Συγκεκριμένα, ο μουσικός με την αναπαραγωγή της μελωδίας και μόνο θα μπορεί να την αποθηκεύσει ως ένα αρχείο στον υπολογιστή του, το οποίο θα έχει τη δυνατότητα στη συνέχεια να επεξεργαστεί, να αλλάξει ή να διορθώσει.

1.1.1 Παρόμοιες εφαρμογές

Υπάρχουν αρκετές εφαρμογές με τις οποίες ένας μουσικός μπορεί να αποτυπώσει μουσική από midi σήμα. Παραδείγματα τέτοιων εφαρμογών είναι το 8notes [1] και το musescore [2]. Η διαφορά του midi σήματος από κανονικό ήχο είναι πως, επί της ουσίας, το midi δεν είναι ήχος αλλά πληροφορία που έχει ό,τι χρειάζεται κανείς για να αποτυπώσει τη μουσική.

Κατά την έρευνά μου βρήκα άλλη μια εφαρμογή που λαμβάνει ως είσοδο αναλογικό σήμα ήχου και από αυτό μπορεί να αποτυπώσει μουσική. Το όνομα της εφαρμογής είναι ScoreCloud [3] και δουλεύει σε Windows και Mac OS X.

Ως κάτι επιπλέον που δίνει η εφαρμογή που αναπτύχθηκε κατά την εκπόνηση της παρούσας πτυχιακής εργασίας, είναι η δυνατότητα χρήσης παρόμοιου λογισμικού σε Linux.

1.2 Σημειογραφία

1.2.1 Παλαιότερες τεχνικές

Πρώτη φορά συναντάμε μουσική σημειογραφία στην περιοχή του Ιράκ πάνω σε ένα πλακίδιο σφηνοειδούς γραφής το οποίο χρονολογείται στα 2000 π.Χ. και περιέχει εκτελεστικές οδηγίες. Ακόμα, βλέπουμε σε ένα πλακίδιο του 1250 π.Χ. να αποτυπώνεται μια εξελιγμένη μορφή της ίδιας σημειογραφίας.

Στην αρχαία Ελλάδα χρησιμοποιούσαν σύστημα με το οποίο μπορούσαν να απεικονίσουν τόσο την διάρκεια όσο και το τονικό ύψος κάθε νότας, ενώ υπήρχε και η δυνατότητα αναπαράστασης λειτουργικής αρμονίας μέχρι ενός βαθμού. Σε αυτή τη σημειογραφία, ειδικά σύμβολα τοποθετούνταν πάνω από το κείμενο και προσδιόριζαν τη μελωδία.

Η σημειογραφία της βυζαντινής περιόδου μοιράζεται κοινά με την αντίστοιχη Δυτική. Στο σύστημα υπάρχουν μουσικά μέτρα και η κατεύθυνση είναι δεξιόστροφη. Σημαντική διαφορά είναι πως στη βυζαντινή μουσική τα σύμβολα δεν αναπαριστούν απόλυτους μουσικούς φθόγγους αλλά σχετικές

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

τονικές αλλαγές (πτώση ή άνοδος). Επιπροσθέτως, ανάμεσα στα μέτρα μπορούν να συμβολιστούν αλλαγές ρυθμικής αγωγής και τρόπου.

Με το πέρασμα των αιώνων, ήρθε η ανάγκη για καταγραφή απόλυτων τονικών υψών αντί για σχετικών αλλαγών. Έτσι, επινοήθηκε το πεντάγραμμο, αρχικά έχοντας μία μόνο γραμμή, στη συνέχεια τέσσερις και αρκετά μετά πέντε. [4]

1.2.2 Πεντάγραμμο

Το πεντάγραμμο έχει επικρατήσει για την απεικόνιση της δυτικής μουσικής από τον 15ο μ.Χ. αιώνα. Αποτελείται από πέντε παράλληλες, οριζόντιες γραμμές, στην αρχή των οποίων αναγράφεται ένα εκ των τριών μουσικών κλειδιών (σολ, φα, ντο) και ορίζει σε ποια γραμμή γράφεται η εκάστοτε νότα. Τονικές διαφορές απεικονίζονται με τη καταγραφή νοτών σε διαφορετικές γραμμές ή διαστήματα (ανάμεσα σε δύο γραμμές). Δίπλα στο κλειδί αναγράφεται ο μετρικός οπλισμός. Ο μετρικός οπλισμός είναι ένα κλάσμα που σαν παρονομαστή έναν ακέραιο αριθμό που δείχνει την μετρική μονάδα, ενώ σαν αριθμητή έναν ακέραιο αριθμό που περιγράφει πόσες μετρικές μονάδες υπάρχουν σε ένα πλήρες μέτρο. Τα μουσικά μέτρα χωρίζονται με κάθετες γραμμές που ονομάζονται διαστολές και οι νότες μέσα σε αυτά έχουν διαφορετικό σύμβολο ανάλογα με τη σχετική διάρκειά τους. Αντίστοιχα σύμβολα υπάρχουν και για παύσεις. Για να προσδιοριστεί η απόλυτη διάρκεια των νοτών χρειάζεται η ρυθμική αγωγή η οποία υπάρχει σαν σημείωση και είναι ένας ακέραιος αριθμός που δείχνει πόσοι χτύποι υπάρχουν σε ένα λεπτό. [5]

1.2.3 Ταμπλατούρα

Η ταμπλατούρα θεωρείται ένα είδος πενταγράμμου που χρησιμοποιείται κυρίως για έγχορδα όργανα. Πάνω σε αυτή δεν συμβολίζονται τονικές αλλαγές ή τονικά ύψη, αλλά δακτυλοθεσίες ως ακέραιοι αριθμοί. Κανονικά δεν περιέχουν πληροφορία ούτε για την διάρκεια της νότας, αλλά υπάρχουν παραλλαγές, στις οποίες κοσμούν τον αριθμό με γραμμές παρόμοιες με τα σύμβολα του

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

κανονικού πενταγράμμου. Συνήθως αποτελούνται από τόσες γραμμές όσες χορδές έχει το εκάστοτε όργανο. Οι ταμπλατούρες χρησιμοποιούνται γιατί είναι πολύ πιο εύκολες στην ανάγνωση και στην εκμάθηση λόγω του σημαντικά μικρότερου όγκου πληροφορίας που προσφέρει. [5]

1.3 Python και ανάλυση ήχου

1.3.1 Python

Η Python είναι μία γλώσσα προγραμματισμού υψηλού επιπέδου και γενικού σκοπού που επιτρέπει την πιο γρήγορη εργασία και την αποδοτικότερη υλοποίηση συστημάτων. [6] Στην επίσημη ιστοσελίδα της Python αναφέρεται:

*“Python is powerful... and fast;
plays well with others;
runs everywhere;
is friendly & easy to learn;
is Open”* [7]

1.3.1.1 Ιστορία κι εξέλιξη

Η υλοποίηση της Python ξεκίνησε στην Ολλανδία το 1989, από τον Guido van Rossum και βασίστηκε σε μία άλλη γλώσσα προγραμματισμού, την ABC. Σκοπός του Rossum ήταν να κρατήσει κάποια από τα καλά χαρακτηριστικά της ABC και να αφαιρέσει κάποια άλλα. Ο κύριος λόγος για την δημιουργία της γλώσσας ήταν να μπορεί να διαχειρίζεται σφάλματα (exception handling) αλλά και να είναι κατάλληλη για αλληλεπίδραση με το λειτουργικό σύστημα Amoeba.

Η πρώτη έκδοση της γλώσσας (0.9.0) δημοσιεύτηκε τον Φεβρουάριο του 1991 από τον Rossum. Από αυτήν την έκδοση ακόμα υπήρχαν κλάσεις και κληρονομικότητα, διαχείριση

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

σφαλμάτων, συναρτήσεις και βασικοί τύποι δεδομένων. Από την πρώτη της μορφή, η Python φαίνεται ότι έχει επηρεαστεί από άλλες γλώσσες προγραμματισμού, όπως από την Modula-3 για το σύστημα ενοτήτων της (module system) και την διαχείριση σφαλμάτων. Το 1994 σχηματίζεται το πρώτο forum συζήτησης για την Python, το comp.lang.python, κάτι που αποτέλεσε ορόσημο για το ευρύ κοινό φίλων της γλώσσας.

Ακολούθησαν μερικές εκδόσεις ακόμη με σημαντικότερες την έκδοση 2.0 στις 16 Οκτωβρίου 2000 και την 3.0 στις 3 Δεκεμβρίου του 2008. Στην έκδοση 2.0 ξαναπαρουσιάζεται η επιρροή της Python από άλλες γλώσσες με την νέα λειτουργικότητα της “κατανόησης λίστας” (list comprehension), η οποία ήταν γνωστή από τις γλώσσες SETL και Haskell. Ακόμη, σε αυτήν την έκδοση εμφανίστηκε και το σύστημα “συλλογής σκουπιδιών” (garbage collection) που αποτελεί βασικό χαρακτηριστικό της γλώσσας. Στις εκδόσεις της ίδιας σειράς που ακολούθησαν ξεχωρίζει η 2.2, όπου συνενώθηκαν οι τύποι της Python (γραμμένοι σε C) και οι κλάσεις της (γραμμένες σε Python) σε μία κοινή ιεραρχία, κάτι που κατέστησε το μοντέλο της γλώσσας ξεκάθαρα αντικειμενοστραφή. Στην ίδια έκδοση προστέθηκαν και οι γεννήτριες (generators), εμπνευσμένες από την γλώσσα Icon.

Η έκδοση 3.0 σχεδιάστηκε για να διορθώσει ορισμένα σχεδιαστικά ελαττώματα της γλώσσας και ο λόγος που απέκτησε καινούργιο όνομα σειράς (η σειρά 2 σταμάτησε στην έκδοση 2.7) ήταν ότι δεν ήταν εφικτή η συμβατότητα με τις προηγούμενες εκδόσεις. Στόχος της νέας σειράς ήταν να μειώσει τον αριθμό των ενοτήτων (modules) που είχαν σχεδιαστεί για την εκπλήρωση της ίδιας εργασίας, με βασική αρχή το “Θα έπρεπε να υπάρχει ένας – και κατά προτίμηση μόνο ένας – προφανής τρόπος να το κάνεις”. Παρά ταύτα και αυτή η έκδοση διατήρησε την ύπαρξη πολλών προγραμματιστικών παραδειγμάτων.

Σήμερα επικρατούν δύο βασικές εκδόσεις της Python, η 2.7.10 και η 3.4.3. και με τις δύο να είναι ευρέως διαδεδομένες, όμως με την 3.4 να έχει πιο βέβαιο μέλλον κι εξέλιξη, αφού η 2.7 θα υποστηρίζεται μέχρι και το 2020. Αυτό γιατί αναμένεται οι χρήστες να μεταβούν στις εκδόσεις 3.4+ το συντομότερο. [8]

1.3.1.2 Βασικά χαρακτηριστικά

Multi-paradigm

Βασικό χαρακτηριστικό της Python, είναι η υποστήριξη πολλών διαφορετικών

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή
προγραμματιστικών παραδειγμάτων, όπως αντικειμενοστραφής (object-oriented), δομημένος/διαδικαστικός (structured/procedural) και συναρτησιακός (functional) προγραμματισμός.

Dynamic typing

Η Python κάνει χρήση δυναμικού ελέγχου τύπων, δηλαδή ελέγχει την συνάφεια των τύπων συναρτήσεων ή μεταβλητών κατά την εκτέλεση του προγράμματος. Έτσι, δεν χρειάζεται ο προγραμματιστής να δηλώνει τον τύπο των μεταβλητών και των συναρτήσεων.

Memory Management

Η Python συνδυάζοντας ένα σύστημα μέτρησης των αναφορών σε πόρους (reference counting) με έναν “συλλέκτη σκουπιδιών” (garbage collector), μπορεί να κάνει αποδοτική διαχείριση της μνήμης και να αποδεσμεύει πόρους που δεν χρησιμοποιούνται πλέον.

Dynamic name resolution (late binding)

Ενώ άλλες γλώσσες αντιστοιχίζουν τα ονόματα και τους τύπους των συναρτήσεων κατά το compilation, η Python το κάνει κατά την εκτέλεση. Αν κληθεί μία συνάρτηση, για παράδειγμα, τότε θα ψάξει εκείνη την ώρα με βάση το όνομα της αν υπάρχει και στην συνέχεια θα την καλέσει.

Semantics and simple syntax

Η Python χαρακτηρίζεται από το απλό συντακτικό της και τους κανόνες που το διέπουν, κάτι που στοχεύει στην ευκολότερη εκμάθηση της γλώσσας και την απλούστερη συγγραφή κώδικα.

Συμβατότητα με C

Η Python παρέχει τη δυνατότητας συγγραφής ενοτήτων σε γλώσσα C. Το μεγάλο πλεονέκτημα σε αυτό είναι ο συνδυασμός της ταχύτητας της C και της απλότητας της Python.

Εύκολη ανάπτυξη γραφικού περιβάλλοντος

Η ανάπτυξη γραφικού περιβάλλοντος για μια Python εφαρμογή είναι εξαιρετικά εύκολη με τη χρήση έτοιμων βιβλιοθηκών.

1.3.2 Ανάλυση ήχου

Ως ανάλυση ήχου αναφέρεται η εξαγωγή πληροφοριών από ηχητικά σήματα για οποιοδήποτε σκοπό (ανάλυση, μελέτη, αποθήκευση, σύνθεση, κ.α.). [9] Ο ήχος είναι ένα κύμα πίεσης που ταξιδεύει σε ένα μέσο. Για να το μελετήσουμε με τους σύγχρονους υπολογιστές χρειάζεται πρώτα να μετατραπεί σε ηλεκτρικό σήμα. Η διαδικασία αυτή ονομάζεται ψηφιοποίηση.

Οι ψηφιακές ηχογραφήσεις δεν είναι συνεχόμενες, αλλά διακριτές διαδικασίες συλλογής δεδομένων. Αυτό σημαίνει πως δείγματα του ηχητικού κύματος λαμβάνονται από τον υπολογιστή σε ηλεκτρική μορφή σε τακτά χρονικά διαστήματα. Μία συνήθης συχνότητα λήψης τέτοιων πακέτων είναι 44.100 Hz.

Άλλη μια ιδιότητα της ψηφιοποίησης είναι η ανάθεση κατάλληλων πλατών σε κάθε δείγμα (quantisation). Συνήθως, τα συστήματα ηχογράφησης χρησιμοποιούν σύστημα 65536 (2^{16}) bit για να αντιπροσωπεύουν τις πιθανές τιμές πλάτους γύρω από το μηδέν. Όσα περισσότερα bits τόσο λιγότερη απόκλιση από την πραγματικότητα έχει η έξοδος, ενώ με λίγα bits η διαφορές μπορεί να είναι πολύ μεγάλες. [10]

1.4 Δομή της εργασίας

Η παρούσα πτυχιακή εργασία απαρτίζεται από 6 κεφάλαια. Κάθε ένα από αυτά εστιάζει σε ένα διαφορετικό συστατικό της στοιχείο και το αναλύει, επεξηγώντας βασικές έννοιες και στοιχεία που η εργασία πραγματεύεται.

Αρχικά, παρουσιάζονται διάφορες εισαγωγικές πληροφορίες, όπως ο σκοπός της πτυχιακής εργασίας, βασικές έννοιες της μουσικής και μια ιστορική αναδρομή για την μέχρι τώρα χρησιμοποιούμενη σημειογραφία. Ακόμα, παρατίθενται και οι βασικές έννοιες της ανάλυσης ήχου.

Στη συνέχεια, στο δεύτερο κεφάλαιο, αναλύεται το θεωρητικό υπόβαθρο της πτυχιακής εργασίας, εξηγώντας λεπτομερώς τη μουσική και τη μαθηματική θεωρία που χρειάζεται η εφαρμογή για τη σωστή λειτουργία της.

Το τρίτο κεφάλαιο αναλύεται ο σχεδιασμός της εφαρμογής. Περιγράφεται η ροή του ηχητικού Βασάλος Γιώργος

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

σήματος από την πυγή μέχρι το τελικό παραγόμενο αρχείο και αναφέρονται και βασικές βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση.

Το τέταρτο κεφάλαιο δίνει βάση στην υλοποίηση και παρουσιάζονται οι βασικές κλάσεις και ενότητες τις εφαρμογής. Ξεκαθαρίζονται οι αρμοδιότητες κάθε στοιχείου και περιγράφεται η έξοδος του προγράμματος.

Στο πέμπτο κεφάλαιο παρατίθενται παραδείγματα εκτέλεσης και πληρέστερη κατανόηση. Διαφαίνονται προβλήματα κατά τη σύγκριση των πραγματικών αποτελεσμάτων από την εφαρμογή σε σχέση με τα αναμενόμενα αποτελέσματα. Επίσης, βγαίνουν συμπεράσματα για την αποτελεσματικότητα των μηχανισμών που χρησιμοποιήθηκαν.

Στο τελευταίο κεφάλαιο αναφέρονται τα συμπεράσματα μετά το πέρας της εκπόνησης της εργασίας, που αφορούν τα διάφορα ζητήματα και τους περιορισμούς που έπρεπε να ληφθούν υπ' όψιν. Τέλος, παρουσιάζονται ορισμένες πιθανές επεκτάσεις και βελτιώσεις που μπορούν να γίνουν στο μέλλον.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Μουσική θεωρία

Η μουσική θεωρία έχει σημαντικό ρόλο στην ανάπτυξη της εφαρμογής. Πολλά είναι βασισμένα πάνω σε αυτή, καθώς απώτερος σκοπός είναι η διευκόλυνση μουσικών. Ό,τι χρειάζεται να είναι γνωστό, για την κατανόηση της λειτουργίας της εφαρμογής, αναλύεται παρακάτω.

2.1.1 Ρυθμός

2.1.1.1 Μέτρα
















Κατά την αποτύπωση της μουσικής σε παρτιτούρα, ομάδες νοτών χωρίζονται σε μέτρα. Αυτό γίνεται κυρίως για τη διευκόλυνση του παίκτη κατά την εκτέλεση. Με τον διαχωρισμό σε μέτρα, ο μουσικός είναι σε θέση να ξέρει γρήγορα σε ποιο σημείο της παρτιτούρας βρίσκεται. Επίσης είναι σε σημαντικό βαθμό ευκολότερη η ανάγνωση και μελέτη ενός νέου κομματιού, καθώς τα μέτρα οργανώνουν τις νότες, ορίζουν υπομέρη και κάνουν πιο ευδιάκριτες πιθανές επαναλήψεις.

2.1.1.2 Αξίες

Κάθε νότα που γράφεται σε ένα μέτρο έχει διάρκεια και τόνο. Η σχετική διάρκεια γίνεται κατανοητή από το σχήμα της νότας πάνω στο πεντάγραμμο και λέγεται αξία. Λέγεται σχετική διάρκεια διότι, από μόνη της, δεν είναι αρκετή πληροφορία για να ξέρουμε πόση χρονική διάρκεια θα έχει η νότα. Αντ' αυτού μας γνωστοποιεί πόσους χτύπους θα διαρκέσει. Οι καθαρές αξίες που μπορεί να πάρει μία νότα είναι επτά (ό,τι ακολουθεί υποθέτει ως μετρική μονάδα το τέταρτο. Οι τιμές μπορεί να αυξομειωθούν ανάλογα με αυτή):

- ολόκληρο – διαρκεί τέσσερις χτύπους
- μισό – διαρκεί δύο χτύπους
- τέταρτο – διαρκεί ένα χτύπο
- όγδοο – διαρκεί μισό χτύπο
- δέκατο έκτο – διαρκεί ένα τέταρτο του χτύπου
- τριακοστό δεύτερο – διαρκεί ένα όγδοο του χτύπου
- εξηκοστό τέταρτο – διαρκεί ένα δέκατο έκτο του χτύπου

Αντίστοιχα σύμβολα υπάρχουν και για τις παύσεις, ενώ είναι εύκολα αντιληπτό πως με την συνέχιση της λογικής σχέσης των αξιών μπορούμε να παράξουμε αξία εκατοστού εικοστού ογδού κ.ο.κ..

NOTE & REST CHART				
Name	Note	Rest	Beats	1 ⁴ / ₄ measure
Whole			4	
Half			2	
Quarter			1	
Eighth			$\frac{1}{2}$	
Sixteenth			$\frac{1}{4}$	

Σχήμα 2.1: τα σύμβολα των αξιών για νότες και παύσεις και η διάρκειά τους
lawfieldmusic.commons.hwdsb.on.ca

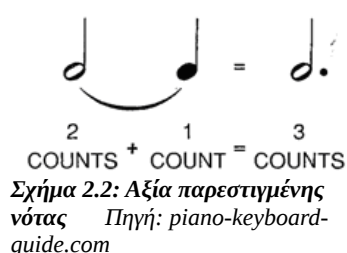
Πηγή:

Στην περίπτωση που η διάρκεια μίας νότας είναι περισσότερο από αυτή του ολόκληρου ή δεν ταιριάζει

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

με καμιά από τις προαναφερόμενες, τότε υπάρχει η δυνατότητα ένωσης δύο ή περισσότερων αξιών για να επιτευχθεί το σωστό αποτέλεσμα. Λέγεται σύζευξη και συμβολίζεται με μια καμπύλη γραμμή που ξεκινάει στην πρώτη διακριτή αξία και καταλήγει στην τελευταία.

Βλέποντας πως είναι συχνό φαινόμενο η σύζευξη, ειδικά σε ακόλουθες αξίες, αποφάσισαν να του δώσουν ξεχωριστό συμβολισμό, χάριν ευκολίας στην ανάγνωση. Έτσι, δημιουργήθηκαν οι παρεστιγμένες νότες. Παρεστιγμένη νότα λέγεται αυτή που η συνολική της διάρκεια είναι η διάρκεια της αξίας της συν τη διάρκεια της αμέσως μικρότερης αξίας. Συμβολίζεται με μία μικρή τελεία ακριβώς μετά το σύμβολο της νότας. [11]



2.1.1.3 Μετρικός οπλισμός

Μετρικός οπλισμός, όπως προαναφέρθηκε στην παράγραφο 1.2.2, είναι ένα κλάσμα το οποίο σαν παρονομαστή έχει έναν ακέραιο αριθμό που δείχνει την μετρική μονάδα, ενώ σαν αριθμητή έναν ακέραιο αριθμό που περιγράφει πόσες μετρικές μονάδες υπάρχουν σε ένα πλήρες μέτρο.

Επί της ουσίας, ο μετρικός οπλισμός λέει στον μουσικό πως να μετρήσει τις νότες στο μέτρο. Συγκεκριμένα, ο αριθμητής αναφέρει πόσες αξίες υπάρχουν στο μέτρο και ο παρονομαστής τι είδους αξίες. Για παράδειγμα, αν ο μετρικός οπλισμός σε ένα μέτρο είναι 3/4, τότε αυτό σημαίνει πως το μέτρο έχει διάρκεια τριών τετάρτων, σε ό,τι αξίες και αν είναι μοιρασμένη.

Ο μετρικός οπλισμός είναι πιθανό να αλλάζει από μέτρο σε μέτρο. Όταν υπάρχει αλλαγή, τότε απλά γράφεται το καινούργιο κλάσμα στην αρχή του μέτρου και συνεχίζεται η καταγραφή κανονικά με τα νέα δεδομένα.

2.1.1.4 Ρυθμική αγωγή

Ό,τι έχει αναφερθεί μέχρι τώρα για τη διάρκεια των νοτών είναι σχετικό. Οι αξίες, όπως αναφέρθηκε, δεν προσδιορίζουν χρόνο, αλλά διάρκεια χτύπων. Για να υπολογίσουμε λοιπόν την χρονική διάρκεια των νοτών, χρειαζόμαστε ένα μέτρο που να προσδιορίζει τους χτύπους. Αυτό είναι η ρυθμική αγωγή (tempo). Πιο συγκεκριμένα, η ρυθμική αγωγή είναι ένας ακέραιος αριθμός που ορίζει πόσοι χτύποι υπάρχουν σε ένα λεπτό. Δεν έχει συγκεκριμένη απεικόνιση και δεν είναι απαραίτητη στην τελική αποτύπωση της μουσικής. Σε περίπτωση που δεν έχει προσδιοριστεί, τότε είναι στην ευχέρεια του εκτελεστή να διαλέξει ό,τι πιστεύει ότι ακούγεται καλύτερο.

2.1.2 Τόνος

Εκτός από τη ρυθμική αξία, τη διάρκεια, μία νότα προσδιορίζεται και από το τονικό ύψος, τον τόνο. Ο τόνος είναι μια ιδιότητα του ήχου που μας επιτρέπει να τον χωρίσουμε σε διακριτές νότες βάσει αυτού. Έτσι, προκύπτουν οι επτά νότες που ξέρουμε: ντο, ρε, μι, φα, σολ, λα, σι. Αν σε αυτές προσθέσουμε 5 ενδιάμεσες, καταλήγουμε σε 12 νότες: ντο, ντο#/ρε^b, ρε, ρε#/μι^b, μι, φα, φα#/σολ^b, σολ, σολ#/λα^b, λα, λα#/σι^b, σι, ενώ με αγγλικούς χαρακτήρες αντίστοιχα: C, C#/D^b, D, D#/E^b, E, F, F#/G^b, G, G#/A^b, A, A#/B^b, B. Τα σύμβολα # και ^b δηλώνουν αύξηση και μείωση ενός διαστήματος από την νότα που αναγράφεται.

2.1.2.1 Συχνότητα

Όπως αναφέρθηκε, ο ήχος είναι ένα κύμα που διαδίδεται σε κάποιο φυσικό μέσο. Το ξεχωριστό που έχει κάθε νότα και προσδιορίζει τον τόνο της είναι η συχνότητα. Παρ' όλο που υπάρχει μεγάλη σχέση μεταξύ τους, ο τόνος και η συχνότητα δεν απευθύνονται στο ίδιο φαινόμενο. Η συχνότητα είναι μία ιδιότητα του κύματος, την οποία μπορούμε να μετρήσουμε. Ο τόνος είναι μία υποκειμενική αντίληψη του ηχητικού κύματος, η οποία δεν μπορεί να μετρηθεί. Αυτό δε σημαίνει απαραίτητα πως οι περισσότεροι άνθρωποι δε θα συμφωνούν για την τονική σχέση δύο νοτών. [12]

2.1.2.1.1 Κούρδισμα

Κάθε νότα είναι ένα ηχητικό κύμα με διαφορετική συχνότητα. Έτσι μπορούμε και αντιλαμβανόμαστε τις επτά καθαρές νότες και τις πέντε επιπλέον ενδιάμεσες.

Λόγω της ψυχοακουστικής αλληλεπίδρασης των τόνων, μερικοί συνδυασμοί νοτών μας φαίνονται πιο φυσικοί. Για παράδειγμα, δύο νότες με διπλάσιες μεταξύ τους συχνότητες δημιουργούν μια οκτάβα, ενώ αν η συχνότητα της μίας νότας είναι τριπλάσια της άλλης τότε δημιουργείται διάστημα τέλει δωδέκατης ή τέλει πέμπτης, αν ελαττωθεί κατά μια οκτάβα. [13]

Το κούρδισμα πραγματοποιείται αναπαράγοντας δύο ήχους και ταιριάζοντάς τους. Υπάρχουν διάφορες συσκευές που παράγουν ήχους αναφοράς, με τη βοήθεια των οποίων κουρδίζονται τα όργανα. [14]

Είναι αδύνατο να κουρδιστεί κάποιο όργανο με όλες τις νότες τέλειες. Οι αναλογίες δεν το επιτρέπουν έχοντας δώδεκα νότες. Υπάρχουν τεχνικές που βασίζονται σε κάποια διαστήματα που ακούγονται πιο ωραία στο αυτί και τα λάθη δεν είναι τόσο μεγάλα. Παρ' όλ' αυτά, σήμερα η πιο συνήθης τεχνική είναι το χρωματικό κούρδισμα. Συγκεκριμένα, βασίζεται στο γεγονός ότι κάθε οκτάβα έχει διπλάσια συχνότητα από τη βασική νότα. Συνεπώς, η μόνη σχέση που πρέπει να τηρηθεί είναι της οκτάβας: $f_1 = 2f_0$. Αν λάβουμε υπ' όψιν μας ότι οι νότες είναι δώδεκα τότε η φόρμουλα για την σχέση συχνοτήτων δύο γειτονικών νοτών είναι: $f_{n+1} = f_n \cdot 2^{1/12}$. Με αυτόν τον τρόπο, όλες οι ενδιάμεσες νότες είναι κατά λίγο ξεκούρδιστες, αλλά σε ανεκτό επίπεδο.

2.1.2.2 Αρμονικοί

Όταν παράγεται μία νότα στον φυσικό κόσμο, εκτός από την βασική συχνότητα, παράγονται και οι αρμονικοί. Οι αρμονικοί είναι ακέραια πολλαπλάσια της βασικής συχνότητας. [15]

2.1.2.2.1 Χροιά

Χροιά μιας νότας θεωρείται η ποιότητά της. Είναι αυτό που κάνει δύο συχνοτικά ίδιες νότες να ξεχωρίζουν αν παράγονται από διαφορετικά όργανα. Η χροιά προσδιορίζεται από το φάσμα και από την περιβάλλουσα του σήματος.

Οι αρμονικοί που παράγονται από κάποιο όργανο καθορίζουν πόσο πλούσιος και ολοκληρωμένος θα ακουστεί ο ήχος. Ο χαμηλότερος αρμονικός είναι η βασική συχνότητα, ο τόνος της οποίας προσδιορίζει τη νότα, ενώ η κυρίαρχος συχνότητα είναι αυτή που ακούγεται περισσότερο.

Η χροιά του ήχου επηρεάζεται και από την περιβάλλουσα συνάρτηση του σήματος. Η περιβάλλουσα δείχνει τα χαρακτηριστικά attack, decay, sustain, release τα οποία προσδιορίζουν τη συμπεριφορά του ήχου από τη στιγμή που θα αρχίσει μέχρι να χαθεί. [16]

2.2 Μαθηματική θεωρία

Για τη διαχείριση του σήματος και την επεξεργασία είναι προφανής η ανάγκη μαθηματικών εργαλείων. Όσον αφορά το ρυθμικό κομμάτι της ανάλυσης, δεν υπάρχει κάποια σημαντική συνεισφορά. Σίγουρα όμως μπορούν να χρησιμοποιηθούν τύποι για να κάνουμε φασματική ανάλυση.

2.2.1 Μετασχηματισμός Fourier

Ο μετασχηματισμός Fourier μετατρέπει ένα σήμα σε μία συνάρτηση, της οποίας η μονάδα μέτρησης είναι η συχνότητα με την οποία εμφανίζει μονάδες κύκλου / δευτερόλεπτο (Hz). [17] Επί της ουσίας, περιγράφεται το ίδιο φαινόμενο από δύο διαφορετικές οπτικές γωνίες. Η πρώτη είναι του χρόνου και φαίνεται στο σήμα εισόδου, ενώ η δεύτερη είναι στο πεδίο των συχνοτήτων και φαίνεται στη συνάρτηση εξόδου.

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

$$X(f) = \int_{-\infty}^{\infty} x(t) \times e^{-i2\pi ft} dt$$

Σχήμα 2.3: Συνάρτηση μετασχηματισμού Fourier

Πηγή: themobilestudio.net

2.2.1.1 DFT (Descrete Fourier Transform)

Ο μετασχηματισμός Fourier μπορεί να εφαρμοστεί σε ένα φυσικό, αναλογικό σήμα, όπως φαίνεται από τη συνάρτηση στο σχήμα 2.3. Ένας υπολογιστής είναι αδύνατο να μπορεί να διαχειριστεί και να επεξεργαστεί τέτοια σήματα. Η λύση είναι η χρήση του DFT αλγορίθμου, ο οποίος αντικαθιστά το άπειρο ολοκλήρωμα με ένα πεπερασμένο άθροισμα.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-i2\pi nk}{N}}, k=0 \dots N-1$$

Σχήμα 2.4: Συνάρτηση DFT Πηγή: quora.com

Με αυτόν τον τρόπο, ο υπολογιστής μπορεί να λάβει το ψηφιοποιημένο ηχητικό σήμα, να το επεξεργαστεί και να βγάλει αποτελέσματα σε σχέση με το πεδίο του φάσματος. Είναι λογικό πως όσο περισσότερα σημεία έχει το σήμα, τόσο πιο κοντά θα είναι το αποτέλεσμα του DFT στον κανονικό μετασχηματισμό Fourier. Συνεπώς, για υψηλότερη ακρίβεια χρειαζόμαστε είτε συχνότερη δειγματοληψία είτε μεγαλύτερο σήμα. [18]

2.2.1.1.1 FFT (Fast Fourier Transform)

Ο FFT υπάρχει για να λύσει το πρόβλημα της ταχύτητας. Μεγαλώνοντας τα δύο μεγέθη που αναφέραμε, η επεξεργασία από έναν υπολογιστή γίνεται βαριά. Ο DFT αλγόριθμος έχει πολυπλοκότητα n^2 , ενώ ο FFT $n \log n$. Αυτό σημαίνει πως είναι πιο αποδοτικός, όσο το μέγεθος του σήματος εισόδου μεγαλώνει.

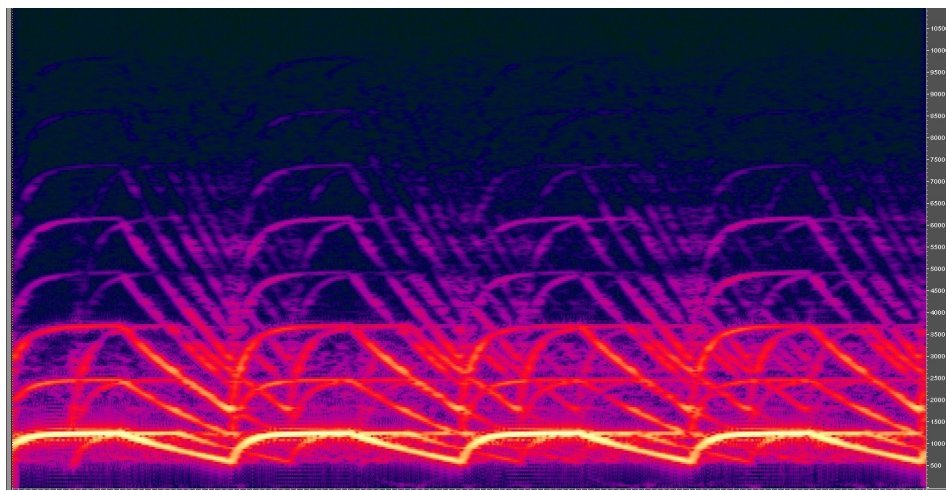
Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

Βασική αρχή του μετασχηματισμού Fourier, και συνεπώς του DFT, είναι ο έλεγχος κάθε συχνότητας πολλαπλασιάζοντας το σήμα με ένα ημίτονο και ένα συνημίτονο αυτής της συχνότητας και ύστερα υπολογίζοντας ένα μέσο όρο των αποτελεσμάτων.

Ο FFT χρησιμοποιεί έναν ειδικό αλγόριθμο για τον υπολογισμό των αρμονικών σαν ομάδα και όχι κάθε έναν ξεχωριστά. Χρειάζεται πολύ λιγότερη υπολογιστική ισχύ σε σχέση με τον DFT, αλλά έχει ως προαπαιτούμενο, το μήκος του σήματος να είναι δυαδικός αριθμός, π.χ. δύναμη του δύο. Αυτό συνεπάγεται πως θα υπάρχουν μεγαλύτερες αποκλείσεις από την πραγματικότητα, διότι με τον FFT το σήμα πολλαπλασιάζεται με ημίτονα και συνημίτονα τα οποία δεν ταιριάζουν σε έναν κύκλο της κυματομορφής. [19]

2.2.1.2 Φασματική ανάλυση

Αποτέλεσμα της χρήσης ενός από τους παραπάνω αλγόριθμους είναι η παραγωγή μιας συνάρτησης με τιμές στο πεδίο του φάσματος. Σε συνδυασμό με την πληροφορία του χρόνου, μπορούμε να παράξουμε φασματογράμματα και να αναλύσουμε το σήμα εισόδου ως προς τον χρόνο και το φάσμα.



Σχήμα 2.5: Φασματογράμμα σειρήνας

Πηγή: commons.wikimedia.org

Στο σχήμα 2.5 φαίνεται το φασματογράμμα μίας σειρήνας. Στον άξονα των x απεικονίζεται ο

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

χρόνος, ενώ στον κάθετο άξονα οι συχνότητες. Εντονότερα χρώματα (άξονας των z) σημαίνουν μεγαλύτερη ισχύ της συγκεκριμένης συχνότητας τη δεδομένη χρονική στιγμή για το σήμα εισόδου.

Κεφάλαιο 3

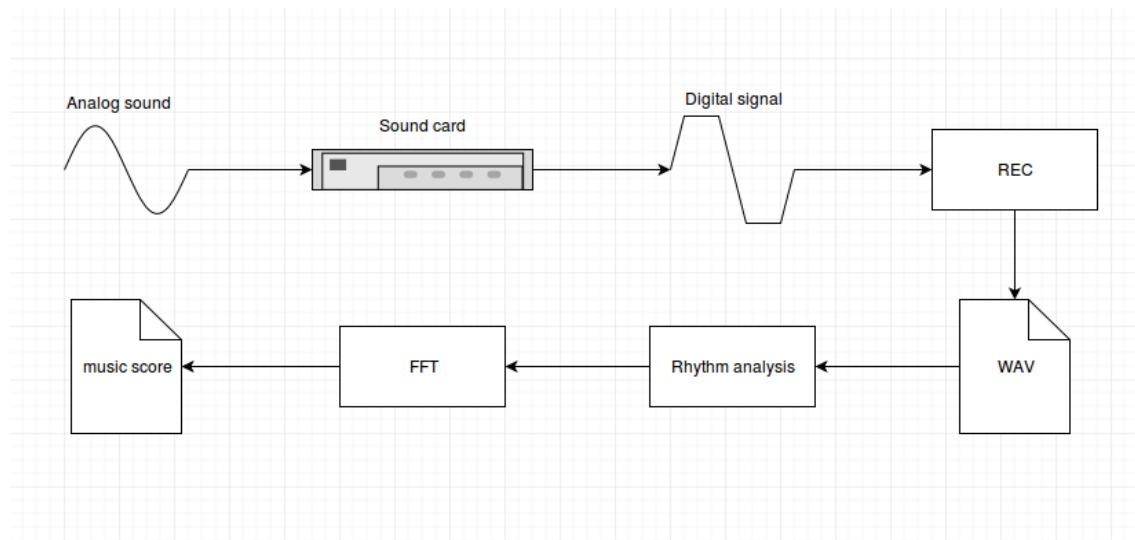
Σχεδιασμός

3.1 Αρχιτεκτονική εφαρμογής

Ο σχεδιασμός και η αρχιτεκτονική της εφαρμογής βασίζεται σε τέσσερα δομικά στοιχεία, τα οποία χωρίζονται σε τρεις “υποεφαρμογές”. Μαζί είναι η ηχογράφηση και η επεξεργασία του ήχου ως το βασικό και πιο μεγάλο κομμάτι της εφαρμογής. Εν συνεχεία, υπάρχει ο μετρονόμος για να μπορεί να συντονιστεί το παίξιμο του μουσικού με την ηχογράφηση και τις τελικές αξίες που θα πάρουν οι νότες. Τέλος, ως επιπρόσθετο, είναι το κουρδιστήρι για να ξέρει ο μουσικός ότι ο ίδιος και το πρόγραμμα αντιλαμβάνονται ίδιους τόνους.

3.1.1 Ροή σήματος ήχου

Το ηχητικό σήμα περνάει από διάφορα βήματα από τη στιγμή που θα αναπαραχθεί από το όργανο μέχρι να αποτυπωθεί σε ένα αρχείο. Αυτή η ροή απεικονίζεται καλύτερα στο παρακάτω σχήμα.



Σχήμα 3.1: Ροή του ηχητικού σήματος κατά τη διάρκεια ηχογράφησης και επεξεργασίας

3.1.1.1 Ηχογράφηση

Η καταγραφή του ήχου διαφέρει από όργανο σε όργανο. Ακουστικά όργανα όπως ακουστικές κιθάρες, βιολιά, πνευστά κ.α. πρέπει να ηχογραφηθούν με τη βοήθεια μικροφώνου. Ηλεκτρικά μουσικά όργανα, όπως ηλεκτρικές κιθάρες και μπάσα παίρνουν το σήμα από τους μαγνήτες και το μεταδίδουν με καλώδιο.

Είναι συχνή η χρήση εξωτερικών καρτών ήχου κατά την ηχογράφηση, διότι είναι πιο δυνατές από τις εσωτερικές κάρτες των προσωπικών ηλεκτρονικών υπολογιστών, έχουν καλύτερη δειγματοληψία και λιγότερη καθυστέρηση κατά την αναπαραγωγή. Ο ρόλος της κάρτας ήχου στη ροή είναι να μετατρέψει το αναλογικό σήμα το οποίο λαμβάνει σε ψηφιακό παλμό ώστε να είναι αναγνώσιμο και επεξεργάσιμο από τον υπολογιστή.

Αφού ψηφιοποιηθεί το σήμα, ένα λογισμικό αναλαμβάνει να γράψει τις τιμές που λαμβάνει σε κάποιο αρχείο για να αποθηκευτούν καταλλήλως. Τα αρχεία wav είναι συχνά. Καταγράφουν όλη την πληροφορία που λαμβάνουν χωρίς συμπίεση. Έτσι έχουν την καλύτερη πιθανή ποιότητα, αλλά δεσμεύουν πολύ χώρο στη μνήμη. Υπάρχουν άλλοι τύποι αρχείων ήχου (π.χ. mp3), οι οποίοι δεσμεύουν σημαντικά λιγότερο χώρο στη μνήμη, αλλά έχουν υποστεί συμπίεση.

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

Η εφαρμογή που αναπτύχθηκε για την παρούσα εργασία, αναλαμβάνει την ηχογράφηση ήχου και την αποθήκευση σε αρχεία wav. Ο χρήστης πατάει το αντίστοιχο κουμπί στο παράθυρο και από εκεί και πέρα ο,τιδήποτε ήχος καταγραφεί, από την προκαθορισμένη πόρτα εισόδου ήχου του υπολογιστή, γράφεται σε ένα αρχείο.

3.1.1.2 Επεξεργασία

Σκοπός της επεξεργασίας είναι να λάβει το ηχητικό σήμα και να το μετατρέψει, εν τέλει, σε κάτι ουσιώδες για τον μουσικό. Βάσει λογικής, έχει χωριστεί σε δύο μεγάλα στάδια.

3.1.1.2.1 Ρυθμική ανάλυση

Η ρυθμική ανάλυση προσπαθεί να χωρίσει το σήμα σε κομμάτια – ομάδες δειγμάτων που απαρτίζουν μία νότα. Λαμβάνει ολόκληρο το ηχογραφημένο σήμα και εξάγει μία λίστα με νότες, με τη χρονική σειρά με την οποία τις βρήκε.

3.1.1.2.2 Συχνοτική ανάλυση

Σκοπός της συχνοτικής ανάλυσης είναι να προσδιορίσει τη συχνότητα κάθε νότας που βρέθηκε νωρίτερα από τη ρυθμική ανάλυση. Χρησιμοποιεί τον FFT όπως περιγράφηκε στην παράγραφο 2.2.1.1.1 και καταλήγει σε κάποιες συχνότητες. Από εκεί και πέρα προσπαθεί να καταλάβει ποια είναι η βασική συχνότητα για να προσδιορίσει τον τόνο της εκάστοτε νότας.

3.1.2 Μετρονόμος

Ο μετρονόμος είναι σημαντικό κομμάτι της ρυθμικής ανάλυσης και ταυτόχρονα κάτι ξεχωριστό. Όπως δείχθηκε στην παράγραφο 2.1.1.4 για να βρούμε την πραγματική χρονική διάρκεια μιας νότας, πρέπει να γνωρίζουμε τη ρυθμική αγωγή. Εδώ ο μετρονόμος κάνει το αντίστροφο. Μέσω

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

αυτού, ορίζονται ο μετρικός οπλισμός και η ρυθμική αγωγή. Με την ηχογράφηση και το αρχικό στάδιο της ρυθμικής ανάλυσης παίρνουμε την απόλυτη διάρκεια της νότας. Συνδέοντας τις δύο πληροφορίες μπορούμε να καταλήξουμε στην αξία της νότας που θα καταλάβει ο μουσικός.

Για να μπορέσει ο εκτελεστής να είναι σταθερός και να τηρήσει τη ρυθμική αγωγή που όρισε, είναι αναγκαίο να ακούγεται ο μετρονόμος κατά τη διάρκεια της ηχογράφησης, χωρίς όμως να επηρεάζει την κυματομορφή του σήματος. Έτσι κατά την εκκίνηση της ηχογράφησης η εφαρμογή της εργασίας ξεκινάει έναν μετρονόμο, όπως αυτός ορίστηκε από τον χρήστη.

3.1.3 Κουρδιστήρι

Το κουρδιστήρι δεν είναι απαραίτητο στοιχείο για τη λειτουργία της εφαρμογής. Η σημαντική βοήθεια που παρέχει, είναι ο συντονισμός των τόνων που αντιλαμβάνεται ο χρήστης με αυτούς που αντιλαμβάνεται ο υπολογιστής. Ξεκινάει από το γραφικό περιβάλλον και δεν έχει καμία ανάμιξη με την επεξεργασία του σήματος.

3.2 Εργαλεία υλοποίησης της εφαρμογής

Για την ανάπτυξη της εφαρμογής, πέραν του προσωπικού κώδικα, χρειάστηκαν βιβλιοθήκες και εργαλεία που υλοποιούν σημαντικές εργασίες.

3.2.1 Βιβλιοθήκες Python

Ένα από τα μεγάλα προσόντα της Python είναι η ευρύτητα των βιβλιοθηκών και ενοτήτων που έχουν αναπτυχθεί και μπορούν να χρησιμοποιηθούν για να κάνουν τη συγγραφή νέου κώδικα πιο εύκολη. Μέσα σε αυτές υπάρχουν βιβλιοθήκες για αναπαραγωγή και εγγραφή ήχου και επιστημονικές βιβλιοθήκες που χρειάστηκαν στην εφαρμογή.

3.2.1.1 PyAudio

Η PyAudio παρέχει διασυνδέσεις σε Python για την cross-platform βιβλιοθήκη Ε/Ε ήχου PortAudio. Με την PyAudio, είναι εύκολο κανείς να χρησιμοποιήσει την Python για να αναπαράξει και να ηχογραφήσει ήχο σε διαφορετικές πλατφόρμες, όπως GNU/Linux, Microsoft Windows και Apple Mac OS X. [20]

3.2.1.2 SciPy

Η βιβλιοθήκη SciPy είναι ένα οικοσύστημα ανοιχτού κώδικα, βασισμένο σε Python, με λογισμικό για μαθηματικά, επιστήμη και μηχανική. Συγκεκριμένα, μερικά από τα σημαντικότερα πακέτα είναι:

- NumPy – “Base N-dimensional array package”
- SciPy library – “Fundamental library for scientific computing”
- Matplotlib – “Comprehensive 2D Plotting”
- IPython – “Enhanced Interactive Console”
- SymPy – “Symbolic mathematics”
- pandas – “Data structures & analysis” [21]

3.2.1.3 NumPy

Το NumPy είναι ένα θεμελιώδες πακέτο για επιστημονικούς υπολογισμούς με Python. Ανάμεσα σε άλλα, περιέχει:

- ένα δυνατό n -διάστατο αντικείμενο πίνακα
- εκλεπτυσμένες συναρτήσεις
- εργαλεία για να ενοποιείται κώδικας C/C++ και Fortran
- χρήσιμη γραμμική άλγεβρα, μετασχηματισμό Fourier και δυνατότητες τυχαίου αριθμού

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

Πέραν των προφανών επιστημονικών χρήσεων, το NumPy μπορεί να χρησιμοποιηθεί ως ένα αποδοτικό πολυδιάστατο δοχείο για δεδομένα γενικής φύσεως. Αυθαίρετοι τύποι μπορούν να οριστούν. Αυτό επιτρέπει στο NumPy να ενοποιείται γρήγορα και χωρίς πρόβλημα με ένα μεγάλο εύρος βάσεων δεδομένων. [22]

3.2.2 Γραφικό περιβάλλον

Για την εφαρμογή αναπτύχθηκε και γραφικό περιβάλλον για την αλληλεπίδραση του χρήστη με τις λειτουργίες της. Όπως και στα υπόλοιπα, η Python παρέχει ένα εύρος βιβλιοθηκών και ενοτήτων για την ανάπτυξη γραφικού περιβάλλοντος.

3.2.2.1 PyQt

Το PyQt είναι ένα σύνολο διεπαφών Python v2 και v3 για το framework της εφαρμογής Qt και τρέχει σε όλες τις πλατφόρμες που υποστηρίζει το Qt, συμπεριλαμβανομένων των Windows, OS X, Linux, iOS και Android. Οι διεπαφές είναι υλοποιημένες ως ένα σύνολο Python ενοτήτων και περιέχουν πάνω από 1000 κλάσεις. [23]

3.2.2.1.1 Qt Designer

Το Qt Designer είναι ένα εργαλείο του Qt για τη σχεδίαση και διεπαφή γραφικών περιβαλλόντων από συστατικά του Qt. Τα widget και οι φόρμες που δημιουργούνται με το Qt Designer ενοποιούνται γρήγορα και εύκολα με κώδικα χρησιμοποιώντας το μηχανισμό σημάτων του Qt. Οποιαδήποτε στοιχεία δημιουργηθούν με το Qt Designer μπορούν αργότερα να αλλάξουν δυναμικά μέσω του κώδικα. [24]

Κεφάλαιο 4

Υλοποίηση

Σε αυτό το κεφάλαιο περιγράφεται η υλοποίηση της εφαρμογής που αναπτύχθηκε στο πλαίσιο της παρούσας πτυχιακής εργασίας. Στόχος της εφαρμογής είναι, όπως αναφέρθηκε παραπάνω, η υλοποίηση εφαρμογής που θα είναι σε θέση να αποτυπώσει τον ήχο που λαμβάνει ως είσοδο με έναν ουσιώδη για τον μουσικό τρόπο. Στο κεφάλαιο θα περιγραφούν αναλυτικά οι λεπτομέρειες υλοποίησης της ηχογράφησης, των σταδίων επεξεργασίας του ήχου, του μετρονόμου και του κουρδιστηριού που χρησιμοποιούνται. Οι κλάσεις γραφικού περιβάλλοντος και τέλος τα παραγόμενα αρχεία σαν έξοδος.

4.1 Λειτουργικότητα

4.1.1 Metronome

Καταλυτική σημασία για την εφαρμογή έχει ο μετρονόμος. Λόγω αυτού υπάρχει η δυνατότητα εύρεσης της αξίας κάθε νότας.

Ο μετρονόμος που υλοποιήθηκε στην εφαρμογή τρέχει σε ξεχωριστό νήμα για να μην διακόπτει την κύρια χρήση του προγράμματος. Κατά τη δημιουργία ενός αντικειμένου Metronome πρέπει να παρέχονται οι αλλαγές που προκύπτουν στα μέτρα μέσα στο κομμάτι όσον αφορά τον μετρικό σπλισμό και την ρυθμική αγωγή.

Αυτές οι αλλαγές μπορούν να διαβαστούν είτε μέσω ενός αρχείου κειμένου, συγκεκριμένης μορφής, είτε σαν αλφαριθμητικά. Παραδείγματα τέτοιων αρχείων φαίνονται στα παρακάτω σχήματα

```
|1: 122  
10:150  
#half tempo  
12:75
```

Σχήμα 4.1

```
|1:4/4  
7:5/4  
8: 4 / 4  
15 :3/4  
18 : 7/8
```

Σχήμα 4.2

Κάθε χτύπος του μετρονόμου υπολογίζεται σύμφωνα με τη θεωρία που περιγράφηκε στην παράγραφο 2.1.1. Στις αλλαγές που έχουν οριστεί, προστίθενται από προεπιλογή ακόμα δύο μέτρα στην αρχή ως `precount`. Το `precount` συνηθίζεται στις ηχογραφήσεις ώστε ο μουσικός να αποκτήσει την αίσθηση του ρυθμού πριν ξεκινήσει να παίζει.

Επιπλέον, κάθε χτύπος τρέχει σε ξεχωριστό νήμα. Αυτό συμβαίνει γιατί τα διαστήματα μεταξύ των χτύπων είναι μικρά, στις περισσότερες περιπτώσεις, οπότε δε θα υπήρχε χρόνος να παίζουν όλοι οι ήχοι αν όλοι έπαιζαν στο ίδιο νήμα.

4.1.2 Recorder

Σημαντική, αν και όχι απαραίτητη, λειτουργία της εφαρμογής είναι η ηχογράφηση. Αυτό επιτυγχάνεται μέσω της κλάσης `Recorder`, η οποία χρησιμοποιεί την βιβλιοθήκη `PyAudio`. Χρησιμοποιεί συχνότητα δειγματοληψίας 44100 Hz και μπορεί να τρέξει με δύο τρόπους. Ή με προεπιλογή χρόνου ηχογράφησης ή με σταμάτημα της ηχογράφησης μετά από εντολή του χρήστη.

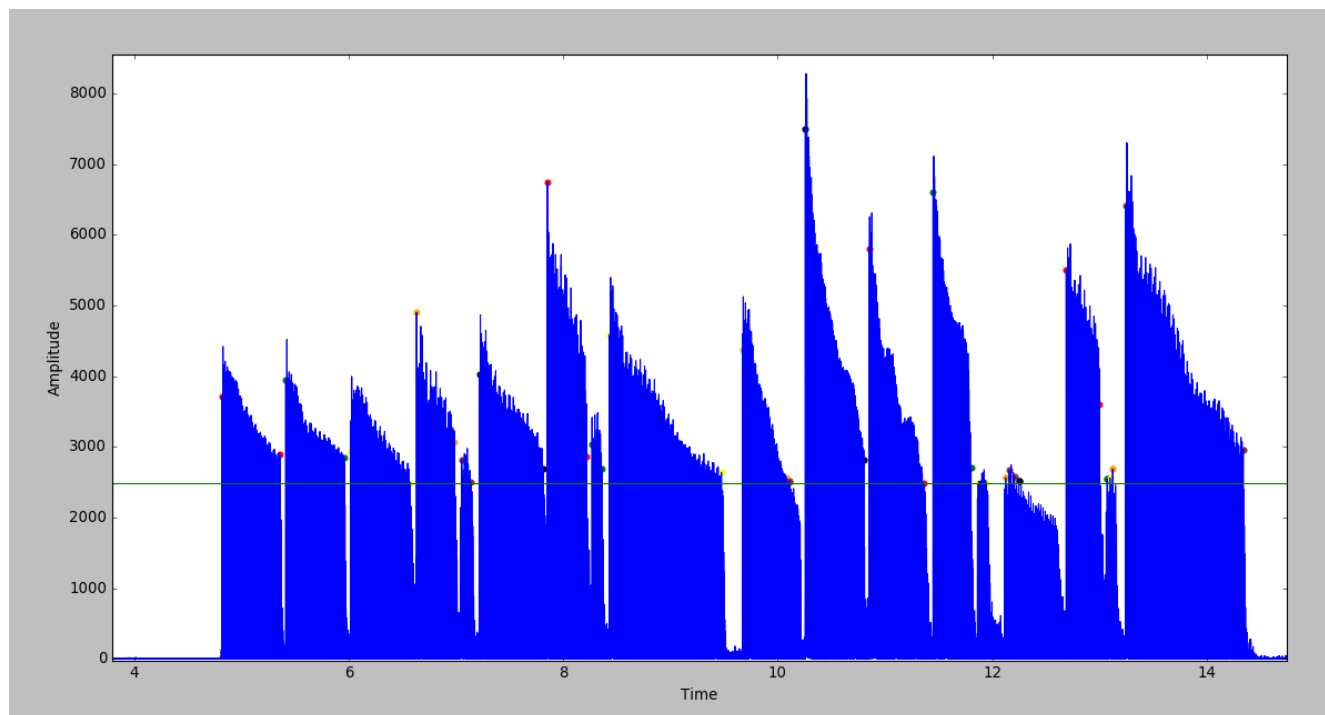
Και με τους δύο τρόπους, η ηχογράφηση είναι μια συλλογή κομματιών 1024 δειγμάτων, τα οποία γράφονται σε ένα αρχείο, μόλις ληφθούν. Η ουσιαστική διαφορά κατά την ηχογράφηση είναι πως στην περίπτωση του προεπιλεγμένου χρόνου η συλλογή γίνεται στο κύριο νήμα, ενώ στην άλλη, κάθε φορά που μαζεύεται η κατάλληλη ομάδα δειγμάτων, δημιουργείται ένα ξεχωριστό νήμα που είναι υπεύθυνο για την εγγραφή του αρχείου.

4.1.3 Rhythm_analyser

Η μεγαλύτερη κλάση της εφαρμογής είναι υπεύθυνη για τη ρυθμική ανάλυση του σήματος

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

εισόδου. Το πρώτο στάδιο της ανάλυσης είναι η εύρεση αρχής και τέλους, όχι αναγκαστικά με απόλυτη ακρίβεια, της κάθε νότας που έχει παιχτεί στο κομμάτι. Για να επιτευχθεί αυτό, ορίζεται στο ξεκίνημα της ανάλυσης ένα κατώφλι με τιμή το 30% του μέγιστου πλάτους στο σήμα. Στη συνέχεια γίνεται μια προσπάθεια εύρεσης της περιβάλλουσας συνάρτησης του σήματος μέσω των τοπικών μεγίστων. Οποιοδήποτε τοπικό μέγιστο έχει πλάτος μεγαλύτερο από το ορισμένο κατώφλι μπαίνει σε μια λίστα ως μέρος παιγμένης νότας. Με αυτόν τον τρόπο, δημιουργούνται διακριτές ομάδες σημείων, οι οποίες απαρτίζουν κάθε νότα. Μία μέγιστη απόσταση μεταξύ δύο τοπικών μεγίστων καθορίζει τα όρια των ομαδοποιημένων σημείων. Για αυτό το λόγο, η ρυθμική ανάλυση χρησιμοποιεί την απόλυτη τιμή του σήματος. Σε περίπτωση που δύο γειτονικά τοπικά μέγιστα απέχουν απόσταση μεγαλύτερη από αυτή, τότε θεωρείται πως τελειώνει μια νότα και ξεκινάει μια νέα. Τέλος, παραλείπονται όλα τα ενδιάμεσα σημεία και μένουν μόνο το πρώτο και το τελευταίο κάθε νότας, αφού είναι αρκετά για να περιγράψουν τη χρονική διάρκειά της.

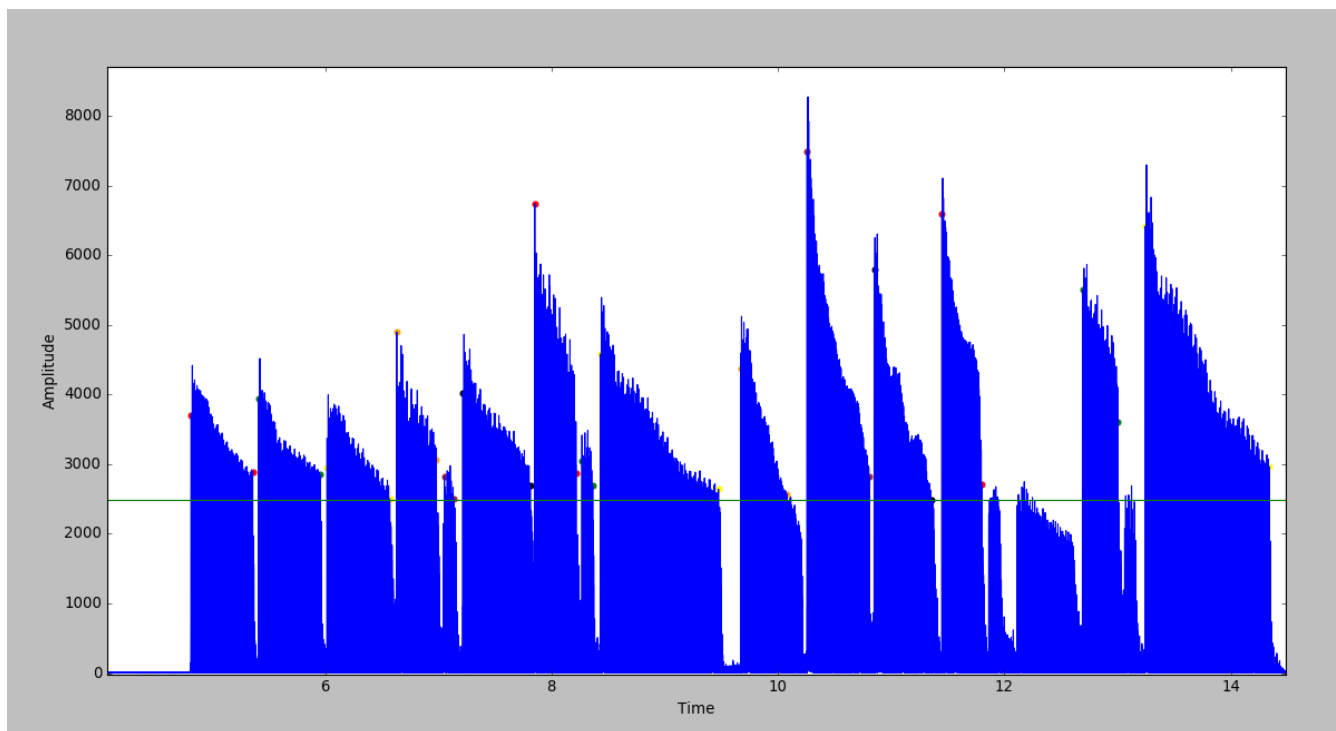


Σχήμα 4.3 Αποτελέσματα μετά το πρώτο στάδιο της ρυθμικής ανάλυσης

Στο σχήμα 4.3 φαίνεται η πρόοδος της ρυθμικής ανάλυσης μετά το πέρας του πρώτου σταδίου. Αναλυτικά, με μπλέ χρώμα είναι η απόλυτη τιμή του σήματος εισόδου, η πράσινη οριζόντια γραμμή είναι το κατώφλι που ορίζεται στην αρχή και τα πολύχρωμα σημεία ,σε ζευγάρια, δηλώνουν αρχή και

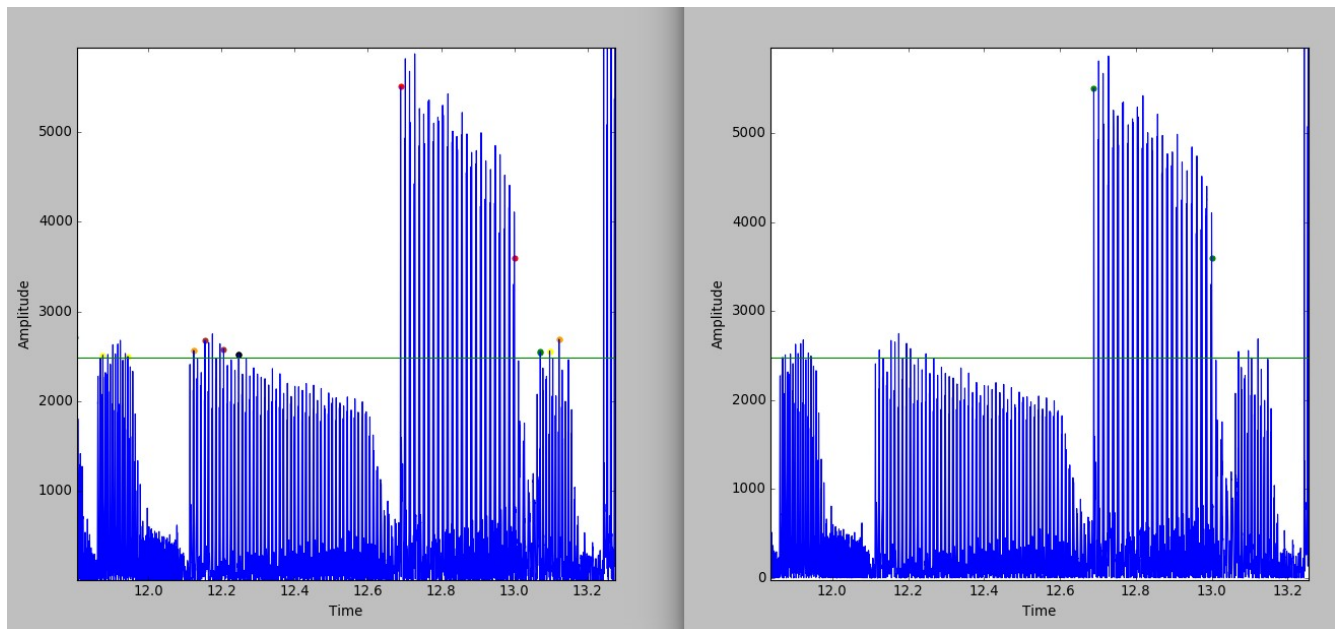
Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή
τέλος μιας νότας.

Αν προσέξει κανείς το σχήμα 4.3 βλέπει ότι υπάρχουν και ομάδες σημείων που αποτελούνται από ένα μόνο τοπικό μέγιστο. Το δεύτερο στάδιο της ρυθμικής ανάλυσης αναλαμβάνει να διαχειριστεί αυτά τα σημεία και τις νότες που φαίνεται να είναι πολύ μικρής διάρκειας για να είναι μέρος της μελωδίας. Εν ολίγοις, το δεύτερο στάδιο είναι υπεύθυνο για την απαλοιφή του θορύβου. Για τις ανάγκες της πτυχιακής εργασίας, αυτό το στάδιο απλά αγνοεί αυτά τα σημεία, άρα και τις νότες, διαγράφοντάς τα.



Σχήμα 4.4 Αποτελέσματα μετά το δεύτερο στάδιο της ρυθμικής ανάλυσης

Το σχήμα 4.4 περιέχει τα ίδια στοιχεία με το σχήμα 4.3 αλλά μετά την ολοκλήρωση του δεύτερου σταδίου της ρυθμικής ανάλυσης, ενώ στο σχήμα 4.5 φαίνονται ξεκάθαρα οι διαφορές.



Σχήμα 4.5

Το τρίτο και τελευταίο στάδιο της ρυθμικής ανάλυσης το ονομάζω `snar` και είναι υπεύθυνο για τη δημιουργία των μέτρων, την ταξινόμηση κάθε νότας σε αυτά και την τελική απόδοση αξιών και θέσεων στις νότες. Αρχικά, κάθε νότα τοποθετείται στο πρώτο μέτρο που έχει κενό χώρο. Μόλις μπει στο μέτρο, η αξία της υπολογίζεται με βάση το τέλος της προηγούμενης νότας (ή την αρχή του μέτρου αν είναι η πρώτη νότα σε αυτό) και το τέλος του μέτρου. Όταν μια καινούργια νότα μπει στο μέτρο, τότε θα επανυπολογιστεί η αξία της τελευταίας νότας βάσει της αρχής της νεοεισαχθείσας. Ο διπλός υπολογισμός των αξιών είναι απαραίτητος, διότι κανένας άνθρωπος δεν μπορεί να έχει την ακρίβεια του υπολογιστή όταν παίζει. Συνεπώς υπάρχουν λάθη που πρέπει να ληφθούν υπ' όψιν. Η έξοδος σε αυτό το στάδιο είναι τα μέτρα με τις περιεχόμενες νότες.

4.1.4 FFT_tuner

Η κλάση `FFT_tuner` παρέχει δύο υπηρεσίες κάνοντας ένα πράγμα. Η βασική λειτουργία της είναι η εφαρμογή του FFT αλγορίθμου όπως αυτός περιγράφηκε στην παράγραφο 2.2.1.1.1. Μέσω αυτού μπορεί να χρησιμεύσει ως κουρδιστήρι πραγματικού χρόνου και ως συχνοτικός αναλυτής για τις

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

νότες που έχουν παραχθεί από τη ρυθμική ανάλυση. Κατά τη δημιουργία ενός αντικειμένου FFT_tuner, παρέχεται πληροφορία για τη συχνότητα όλως των νοτών που μπορεί να διαχειριστεί η εφαρμογή.

Κατά την χρήση ως κουρδιστήρι, είσοδος λαμβάνεται από την προεπιλεγμένη είσοδο ήχου και γίνεται ανάλυση σε κάθε 2^{16} σημεία. Αντίθετα, στην περίπτωση της ανάλυσης σήματος, είσοδος είναι το κομμάτι του σήματος που θα αναλυθεί, με μέγεθος ακαθόριστο.

Η έξοδος της βασικής μεθόδου της κλάσης FFT_tuner είναι μία λίστα που περιέχει τον τόνο της νότας που βρέθηκε, αν μπορεί να θεωρηθεί σωστή ή αν απέχει αρκετά προς ψηλότερες ή χαμηλότερες συχνότητες και η απόλυτη διαφορά από την ακριβή συχνότητα του τόνου.

Η μέθοδος υπολογισμού της εξόδου είναι απλή. Πρώτα το σήμα περνάει από τον FFT αλγόριθμο. Το αποτέλεσμα του FFT είναι μία συνάρτηση που απεικονίζει ποιες νότες παίχτηκαν και με πόση δύναμη. Λόγω των παραγόμενων αρμονικών, δεν μπορούμε να βρούμε την βασική συχνότητα παίρνοντας την μέγιστη δύναμη. Σε πολλά όργανα η βασική νότα δε συνάδει με την κυρίαρχη συχνότητα. Άρα, αυτό που γίνεται, επί της ουσίας, είναι ότι βρίσκει και επιστρέφει τον πρώτο αρμονικό, άρα τη βασική συχνότητα.

Στη συνέχεια, το αποτέλεσμα συγκρίνεται με όλες τις καταγεγραμμένες νότες και βρίσκεται η πλησιέστερη. Λόγω της διακριτότητας των DFT και FFT, δεν ελέγχονται όλες οι συχνότητες και έτσι το αποτέλεσμα από τον μετασχηματισμό να μην είναι ακριβές. Συνεπώς, για να θεωρήσουμε ίσες δύο συχνότητες ορίζεται δυναμικά ένα περιθώριο λάθους. Αν οι συγκρινόμενες συχνότητες διαφέρουν για λιγότερο από αυτό, τότε θεωρούνται ίσες.

4.1.5 Frequency_analyser

Η κλάση Frequency_analyser είναι ένα wrapper για τη λειτουργικότητα της FFT_tuner, όσον αφορά την ανάληψη του σήματος. Το μόνο που πρέπει να κάνει η Frequency_analyser είναι να δώσει σωστές εισόδους στην FFT_tuner. Μόλις ολοκληρωθεί η διαδικασία, επιστρέφονται τα μέτρα του κομματιού που αναλύθηκε με επιπρόσθετη πληροφορία για τους τόνους των νοτών.

4.2 Οργάνωση

4.2.1 InstaScore_project

Η InstaScore_project είναι μια μικρή αλλά σημαντική κλάση. Είναι υπεύθυνη για την οργάνωση των προαναφερθέντων κλάσεων σε ένα project. Ένα project αποτελείται από τις ηχογραφήσεις, τα αρχεία που αναγράφεται η μουσική, και τα αρχεία αλλαγής ρυθμικής ανάλυσης και μετρικού οπλισμού. Επίσης, κατά τη διάρκεια εκτέλεσης υπάρχει ο μετρονόμος που είναι μέρος του project.

4.3 Γραφικό περιβάλλον

Για την ανάπτυξη του γραφικού περιβάλλοντος χρησιμοποιήθηκε το Qt Designer framework. Ο τρόπος λειτουργίας του Qt Designer είναι η εξαγωγή ui αρχείων με την πληροφορία των σχεδίων. Μετά αυτά τα αρχεία περνούν από compilation και δημιουργούνται Python κλάσεις.

4.3.1 Παραγόμενες κλάσεις

Η βασική μέθοδος σε κάθε παραγόμενη κλάση, είναι η setupUi και είναι υπεύθυνη για τον καθορισμό των αλλαγών που έχουμε κάνει μέσω της εφαρμογής.

4.3.2 Παραμετροποιημένες υποκλάσεις

Είναι καλή πρακτική να μην διορθώνονται οι παραγόμενες κλάσεις με το χέρι. Αυτό γιατί σε

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

περίπτωση που αποφασιστεί κάποια αλλαγή στο γραφικό περιβάλλον, τότε οι διορθώσεις που γίνανε θα χαθούν. Έτσι, αντ' αυτού, πρέπει να δημιουργηθούν νέες κλάσεις ως παιδιά των παραγόμενων κλάσεων. Σε αυτές τις υποκλάσεις, μπορούν να γίνουν αλλαγές και παραμετροποιήσεις που δε θα επηρεαστούν σε περίπτωση που αλλάξει η βασική κλάση.

4.4 Έξοδος

Η εφαρμογή παράγει ως έξοδο αρχεία που αποτυπώνουν με δικό τους τρόπο την μουσική που αναπαράχθηκε. Δημιουργούνται αρχεία κειμένου, κάθε γραμμή των οποίων αναπαριστά ένα μουσικό μέτρο. Στην αρχή αναγράφονται η ρυθμική αγωγή και ο μετρικός σπλισμός. Εν συνεχεία, υπάρχουν οι νότες ως ζευγάρια τόνου και αξίας.

4.4.1 Extractor

Η κλάση Extractor είναι υπεύθυνη για τη σωστή αποτύπωση των νοτών στο αρχείο.

Κεφάλαιο 5

Αποτελέσματα

5.1 Παραδείγματα εκτέλεσης

5.1.1 Σύγκριση αναμενόμενης – πραγματικής εξόδου

Κατά τη διάρκεια της ανάπτυξης της εφαρμογής, έγιναν δοκιμές με πραγματική είσοδο. Μερικά παραδείγματα φαίνονται στα ακόλουθα σχήματα, στα οποία φαίνονται και η αναμενόμενη και η πραγματική έξοδος.

```
1 tempo: 120, 4/4, notes: (E2, quarter_dotted), (E2, eighth), (G2, eighth_dotted), (E2, eighth_dotted), (D2, eighth)
2 tempo: 120, 4/4, notes: (C2, half), (B1, half)
3 tempo: 120, 4/4, notes: (E2, quarter_dotted), (E2, eighth), (G2, eighth_dotted), (E2, eighth_dotted), (D2, eighth)
4 tempo: 120, 4/4, notes: (C2, eighth_dotted), (D2, eighth_dotted), (C2, eighth), (B1, half)
5 tempo: 120, 4/4, notes: (E2, quarter_dotted), (E2, eighth), (G2, eighth_dotted), (E2, eighth_dotted), (D2, eighth)
6 tempo: 120, 4/4, notes: (C2, half), (B1, half)
7 tempo: 120, 4/4, notes: (E2, quarter_dotted), (E2, eighth), (G2, eighth_dotted), (E2, eighth_dotted), (D2, eighth)
8 tempo: 120, 4/4, notes: (C2, eighth_dotted), (D2, eighth_dotted), (C2, eighth), (B1, half)
9 tempo: 120, 4/4, notes: (G1, eighth), (G1, eighth), (G1, eighth), (G1, eighth), (G1, eighth), (G1, eighth), (G1, eighth), (G1, eighth)
10 tempo: 120, 4/4, notes: (A1, eighth), (A1, eighth), (A1, eighth), (A1, eighth), (A1, eighth), (A1, eighth), (A1, eighth), (A1, eighth)
```

Σχήμα 5.1: *White Stripes - Seven Nation Army*

```
1 tempo: 115, 4/4, notes: (G1, quarter), (A#1, quarter), (C2, quarter_dotted), (G1, eighth)
2 tempo: 115, 4/4, notes: (tie, eighth), (A#1, quarter), (C#2, eighth), (C2, half)
3 tempo: 115, 4/4, notes: (G1, quarter), (A#1, quarter), (C2, quarter_dotted), (A#1, eighth)
4 tempo: 115, 4/4, notes: (tie, eighth), (G1, half_dotted, eighth)
5 tempo: 115, 4/4, notes: (G1, quarter), (A#1, quarter), (C2, quarter_dotted), (G1, eighth)
6 tempo: 115, 4/4, notes: (tie, eighth), (A1, quarter), (C#2, eighth), (C2, half)
7 tempo: 115, 4/4, notes: (G1, quarter), (A#1, quarter), (C2, quarter_dotted), (A#1, eighth)
8 tempo: 115, 4/4, notes: (tie, eighth), (G1, quarter), (G1, eighth), (G1, eighth), (G1, eighth), (G1, eighth), (G1, eighth)
```

Σχήμα 5.2: *Deep Purple - Smoke On The Water*

```
1 tempo: 120, 4/4, notes: (E2, quarter_dotted), (E2, eighth), (G2, eighth_dotted), (E2, eighth_dotted), (D2, eighth)
2 tempo: 120, 4/4, notes: (C2, half), (B1, half)
3 tempo: 120, 4/4, notes: (E2, quarter_dotted), (E2, eighth), (G2, eighth_dotted), (E2, eighth_dotted), (C#2, eighth)
4 tempo: 120, 4/4, notes: (C2, eighth_dotted), (D2, eighth_dotted), (C2, eighth), (B1, half)
5 tempo: 120, 4/4, notes: (E2, quarter_dotted), (E2, eighth), (G2, eighth_dotted), (E2, eighth_dotted), (D2, eighth)
6 tempo: 120, 4/4, notes: (C2, half), (B1, half)
7 tempo: 120, 4/4, notes: (E2, quarter_dotted), (E2, eighth_dotted), (G2, eighth), (E2, eighth_dotted), (D2, eighth)
8 tempo: 120, 4/4, notes: (C2, eighth_dotted), (D2, eighth_dotted), (C2, eighth), (B1, half)
9 tempo: 120, 4/4, notes: (G1, eighth), (F#1, eighth), (G1, eighth_dotted), (F#1, sixteenth_dotted), (G1, eighth), (G1, eighth), (F#1, eighth), (F#1, sixteenth_dotted)
10 tempo: 120, 4/4, notes: (A1, eighth_dotted), (A#1, sixteenth_dotted), (A1, eighth), (G#1, eighth), (A1, eighth), (G#1, eighth), (A1, eighth), (G#1, eighth), (A1, eighth), (G#1, sixteenth_dotted)
```

```
1 tempo: 115, 4/4, notes: (G1, quarter), (B1, quarter), (C2, quarter_dotted), (G1, eighth)
2 tempo: 115, 4/4, notes: (tie, eighth), (A#1, quarter), (C#2, eighth), (C2, half)
3 tempo: 115, 4/4, notes: (F#1, quarter), (A1, quarter), (C2, quarter_dotted), (A#1, eighth)
4 tempo: 115, 4/4, notes: (tie, eighth), (G1, half_dotted, eighth)
5 tempo: 115, 4/4, notes: (G1, quarter), (A#1, quarter), (C2, quarter_dotted), (G1, eighth)
6 tempo: 115, 4/4, notes: (tie, eighth_dotted), (A1, quarter), (C#2, eighth), (C2, quarter_dotted)
7 tempo: 115, 4/4, notes: (G1, quarter), (A#1, quarter), (C2, quarter_dotted), (A#1, eighth)
8 tempo: 115, 4/4, notes: (tie, eighth), (G1, quarter), (G1, eighth), (F#1, eighth), (G1, eighth_dotted), (F#1, sixteenth_dotted), (G2, sixteenth_dotted)
```

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

```
1 tempo: 100, 4/4, notes: (G2, quarter), (G2, quarter), (G2, quarter), (D#2, eighth_dotted), (A#2, sixteenth)
2 tempo: 100, 4/4, notes: (G2, quarter), (D#2, eighth_dotted), (A#2, sixteenth), (G2, half)
3 tempo: 100, 4/4, notes: (D3, quarter), (D3, quarter), (D3, quarter), (D#3, eighth_dotted), (D3, sixteenth)
4 tempo: 100, 4/4, notes: (G2, quarter), (D#2, eighth_dotted), (A#2, sixteenth), (G2, half)
5 tempo: 100, 4/4, notes: (E3, quarter), (E2, quarter), (E3, quarter), (D#3, eighth_dotted), (D#3, sixteenth)
6 tempo: 100, 4/4, notes: (C#3, sixteenth), (C, sixteenth), (C#3, quarter), (E2, eighth), (C#3, quarter), (C3, eighth_dotted), (C3, sixteenth)
7 tempo: 100, 4/4, notes: (A#2, sixteenth), (A2, sixteenth), (A#2, quarter), (E2, eighth), (G2, quarter), (D#2, eighth_dotted), (A#2, sixteenth)
8 tempo: 100, 4/4, notes: (G2, quarter), (D#2, eighth_dotted), (A#2, sixteenth), (G2, quarter), (D#2, eighth_dotted), (A#2, sixteenth)
9 tempo: 100, 4/4, notes: [G2, whole]
```

Σχήμα 5.3: *Star Wars - Imperial March*

Στα σχήματα 5.1, 5.2 και 5.3 συγκρίνονται αναμενόμενες έξοδοι (αριστερά) με εξόδους από την εφαρμογή (δεξιά)

5.2 Δυσκολίες κατά την ανάπτυξη & προβλήματα

Καθώς συνέγραφα τον κώδικα για την εφαρμογή, αντιμετώπισα αρκετές δυσκολίες. Αρχικά, δεν υπάρχει κάποια παρόμοια εφαρμογή στην οποία μπορώ να βασιστώ και να αναπτύξω τις δικές μου επιπλέον ιδέες. Οπότε η μεγαλύτερη δυσκολία ήταν να σκεφτώ τρόπους αντιμετώπισης βασικών προβλημάτων στην ανάλυση.

Ακόμα, όσον αφορά τη συχνотική ανάλυση, εφαρμόζοντας τη θεωρία, είδα πως λαμβάνονται λάθος αποτελέσματα σε πολλές περιπτώσεις. Είναι ένας περιορισμός του FFT καθώς, με λίγα σημεία, τα αποτελέσματα δεν είναι τόσο ακριβή. Επίσης, ψάχνοντας την βασική συχνότητα μιας νότας έψαχνα το μέγιστο, καθώς θεωρούσα πως θα είναι το σωστό αποτέλεσμα. Αργότερα κατάλαβα πως κάθε όργανο παράγει διαφορετικούς αρμονικούς και πολλές φορές η βασική νότα δεν είναι η κυρίαρχη.

5.3 Συμπεράσματα

Με βάση τα παραδείγματα εκτέλεσης που παρουσιάστηκαν πιο πάνω και λαμβάνοντας υπ' όψιν τα προβλήματα και τις δυσκολίες που ανέφερα, είναι φανερό πως τα αποτελέσματα δεν είναι τέλεια. Υπάρχουν περιθώρια βελτίωσης για να εξάγονται καλύτερα αποτελέσματα. Τα τελικά αρχεία δεν είναι τελείως λάθος και πιστεύω ταιριάζουν σε ικανοποιητικό βαθμό με τα αναμενόμενα. Νομίζω πως με

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

περαιτέρω ανάπτυξη και προσπάθεια θα μπορούσε η εφαρμογή να φτάσει σε καλύτερα αποτελέσματα και να χρησιμοποιηθεί ευρέως.

Κεφάλαιο 6

Συμπεράσματα

6.1 Συμπεράσματα

Στην παρούσα πτυχιακή εργασία αναπτύχθηκε μία εφαρμογή για ηλεκτρονικούς υπολογιστές, με την χρήση της γλώσσας προγραμματισμού Python, με σκοπό τη διευκόλυνση των μουσικών κατά την αποτύπωση της μουσικής.

Δημιουργήθηκαν κατάλληλες διασυνδέσεις και αλγόριθμοι για να επιτευχθεί ο στόχος. Βασικά χαρακτηριστικά της εφαρμογής που δημιουργήθηκαν με σκοπό τη διευκόλυνση του μουσικού είναι ο μετρονόμος, το κουρδιστήρι και η δυνατότητα ηχογράφησης.

Προσωπικά, μέσα από την εργασία έμαθα πολλά πράγματα για την μουσική, που δε θα είχα άλλοτε την ευκαιρία και, λόγω των πολλών προβλημάτων που αντιμετώπισα, μπόρεσα να σκεφτώ διαφορετικούς τρόπους προσέγγισης που ίσως να μου χρειαστούν στο μέλλον.

Μπορεί η εφαρμογή να μην παρουσιάζει τέλεια αποτελέσματα, παρ' όλ' αυτά, πιστεύω, πως πέτυχε τον στόχο της, καθώς τα αρχεία που εξάγονται είναι σε αρκετό ποσοστό κοντά στην πραγματικότητα και είναι εύκολα αναγνώσιμα από έναν μουσικό.

6.2 Μελλοντικές επεκτάσεις

Σίγουρα υπάρχουν πολλά περιθώρια βελτίωσης. Το πρώτο βήμα για αυτό είναι η δοκιμή της εφαρμογής σε διαφορετικά περιβάλλοντα. Μελέτη και έρευνα χρειάζεται και για την παρατήρηση των διαφορετικών αρμονικών που παράγονται από κάθε όργανο. Με αυτήν την προσαρμογή πιστεύω πως η

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

εφαρμογή θα μπορέσει να αναγνωρίσει και πολυφωνίες, όπως τα ακόρντα μιας κιθάρας, και όχι μόνο μεμονωμένες νότες.

Κατά τη ρυθμική ανάλυση θα μπορούσαν να χρησιμοποιούνται πιο έξυπνα τα σημεία που παραλείπονται, ώστε να βρίσκονται πιο αδύναμες νότες. Επιπλέον, θα ήθελα να εστιάσω στην αρχή της ανάλυσης και να βελτιώσω τον αλγόριθμο, καθώς νομίζω πως παίζει καταλυτικό ρόλο για τη συνέχεια.

Για τη συχνοτική ανάλυση, θα ήθελα να εφαρμόσω άλλες τεχνικές για να εξετάσω αν τα αποτελέσματα θα είναι καλύτερα και δε θα επηρεάζονται από το πλήθος των σημείων.

Τέλος, θα ήθελα να κάνω αλλαγές στο γραφικό περιβάλλον ώστε να είναι πιο φιλικό προς τον χρήστη. Μερικές λειτουργίες της εφαρμογής είναι “κρυμμένες” από το γραφικό περιβάλλον και αλλάζοντάς το ο χρήστης θα μπορεί να έχει σε όλες τις δυνατότητες που παρέχονται εύκολη πρόσβαση.

Βιβλιογραφία

[1] 8notes. Midi to Sheet Music Converter.

http://www.8notes.com/midi_sheet_music/ [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[2] muscore. MIDI import.

<https://musescore.org/en/handbook/midi-import> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[3] ScoreCloud.

<http://scorecloud.com> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[4] Wikipedia, The Free Encyclopedia. Μουσική σημειογραφία.

https://el.wikipedia.org/wiki/Μουσική_σημειογραφία [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[5] Wikipedia, The Free Encyclopedia. Μουσικό πεντάγραμμο.

https://el.wikipedia.org/wiki/Μουσικό_πεντάγραμμο [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[6] Python

<https://www.python.org/> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

Αποτύπωση μουσικής σε παρτιτούρα/ταμπλατούρα μέσω υπολογιστή

[7] Python

<https://www.python.org/about/> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[8] Wikipedia, The Free Encyclopedia. History of Python.

https://en.wikipedia.org/wiki/History_of_Python [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[9] Wikipedia, The Free Encyclopedia. Audio analysis.

https://en.wikipedia.org/wiki/Audio_analysis [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[10] Jérôme Sueur. A very short introduction to sound analysis for those who like elephant trumpet calls or other wildlife sound. Museum national d'Histoire naturelle CNRS UMR 7205 OSEB, Paris, France, July 11, 2016.

[11] MusicHeaven. Στιγμή διαρκείαςκαι σύζευξη διαρκείας.

<http://www.musicheaven.gr/html/modules.php?name=News&file=article&id=2266> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[12] Wikipedia, The Free Encyclopedia. Pitch (music).

https://en.wikipedia.org/wiki/Pitch_%28music%29#Pitch_and_frequency [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[13] Wikipedia, The Free Encyclopedia. Musical tuning.

https://en.wikipedia.org/wiki/Musical_tuning#Tuning_systems [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[14] GET-TUNED. The Science of Tuning Musical Instruments.

http://www.get-tuned.com/tuning_science.php [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[15] the Physics Classroom. Fundamental Frequency and Harmonics.

<http://www.physicsclassroom.com/class/sound/Lesson-4/Fundamental-Frequency-and-Harmonics>

[Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[16] Wikipedia, The Free Encyclopedia. Timbre.

<https://en.wikipedia.org/wiki/Timbre#Harmonics> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[17] Wikipedia, The Free Encyclopedia. Μετασχηματισμός Φουριέ.

https://en.wikipedia.org/wiki/Μετασχηματισμός_Φουριέ [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[18] Julius O. Smith III. Mathematics Of The Discrete Fourier Transform (DFT) with Audio Applications, Second Edition, pages 1-7

[19] DFT Vs FFT For Fourier Analysis of Waveforms

<http://file.yizimg.com/320527/2010112211541556.pdf> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[20] PyAudio.

<https://people.csail.mit.edu/hubert/pyaudio/> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[21] SciPy.

<https://www.scipy.org/> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[22] NumPy.

<http://www.numpy.org/> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[23] Riverbank Computing Limited.

<https://riverbankcomputing.com/software/pyqt/intro> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].

[24] Qt.

<http://doc.qt.io/qt-4.8/designer-manual.html> [Online; τελευταία προσπέλαση 10-Ιουλίου-2016].