

**UNIVERSITY OF SOUTHAMPTON**  
Faculty of Physical Sciences and Engineering

A project report submitted for the award of  
Bachelor's in Computer Science

Supervisor: David Millard  
Examiner: Zehor Belkhatir

**Project Audyssey - A Platform for  
Smooth Multidimensional Journeys  
Through Song Libraries**

by Kathirvelan Arounassalam

April 10, 2025



UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING

A project report submitted for the award of Bachelor's in Computer Science

by Kathirvelan Arounassalam



## Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

**I have acknowledged all sources, and identified any content taken from elsewhere.**

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

**I have not used any resources produced by anyone else.**

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

**I did all the work myself, or with my allocated group, and have not helped anyone else.**

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

**The material in the report is genuine, and I have included all my data/-code/designs.**

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for this assessment.

**I have not submitted any part of this work for another assessment.**

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

**My work did not involve human participants, their cells or data, or animals.**

# Contents

<b>1</b>	<b>Introduction (870 words)</b>	<b>1</b>
1.1	Problem . . . . .	1
1.2	Method . . . . .	2
1.2.1	Complete Knowledge over the Collection . . . . .	2
1.2.2	Replayability/Queue Building . . . . .	3
<b>2</b>	<b>Background</b>	<b>7</b>
<b>3</b>	<b>Design</b>	<b>9</b>
<b>4</b>	<b>Development</b>	<b>11</b>
<b>5</b>	<b>Testing</b>	<b>13</b>
5.1	Results . . . . .	13
<b>6</b>	<b>Evaluation</b>	<b>15</b>
<b>7</b>	<b>Project Planning Retrospective</b>	<b>17</b>
<b>8</b>	<b>Conclusions</b>	<b>19</b>
8.1	Future Work . . . . .	19
	<b>Bibliography</b>	<b>21</b>
	<b>Appendix A: Photos</b>	<b>23</b>
	<b>Appendix B: Code Listings</b>	<b>25</b>





# List of Figures



# List of Tables



# Listings



# Chapter 1

## Introduction (870 words)

### 1.1 Problem

As technology has advanced over the years, people's personal music collections have become more and more digital [?].

Initially, if a listener wanted to listen to a song or some music, it would have to be performed live, usually by the creator of said music. As such music started to be performed in concerts, where many people could listen to songs from one or more artists.

Radio allowed people to listen to music together and in the comfort of their own home. However, there wasn't full control over what to listen to.

Vinyls and CDs allowed for listening to any song in any place, with the right equipment. This made it possible for people to start treating music as a collectible physical item and build personal music collections. However these physical collections were limited by purchase cost and storage space.

As technology progressed further, we entered the streaming era, where physical collections were replaced by digital collections. These digital collections are significantly cheaper, and are much less constrained by storage space.

As such music collections have the capability to be significantly larger than their physical counterpart. The process of adding songs is much simpler digitally, allowing these digital collections to grow at a significantly faster rate than physical collections.

To navigate the landscape of a personal music collection the predominant method employed by streaming applications is to simply arrange the items in a list. This method works well on a small scale, but falls apart with the large scale that digital music collections can reach.

As such, as a workaround, most users split their mental collections into sub-collections (playlist)s, folders that allow for a lesser mental load and better maintainability.

Whilst playlists help manage the scale of songs, they themselves' have the issue "when a user's playlists become overwhelmingly numerous, streaming services begin to appear inefficient and unmanageable as collection systems" [?].

This hierarchical system allows for the better handling of large scale collections that grow at the pace most collections do **CITE** but if these playlists grow too large, then they require reorganising which can become a chore.

As such the organisational benefit provided by these playlists is lost due to the 'perceived' high mental load required.

Whilst the capability and format of personal music collections have changed dramatically with technological advancements, the structure of these collections has stagnated in the streaming era.

This project will investigate different structures to see how they can improve either of the following key aspects for personal music collections:

- **Complete Knowledge of the Collection** - understanding the entire contents of the collection (to not lose/forget about songs in the collection over time) without exacting a heavy mental load
- **Replayability/Queue Building** - being able to quickly and frictionlessly create song queues to listen to (where the order of the song queue exists on a spectrum between fixed and random)

## 1.2 Method

First we will try to understand people's mental organisational models for their personal music collections and how they create song queues using their organisational model.

Then we will see the relationship between their mental understanding and their digital music collection.

### 1.2.1 Complete Knowledge over the Collection

For song organisation, the current method employed by streaming applications is a folder-based list structure (using playlists as folders).

Other methods using graphical spatial methods have been researched



- there has been research into creating 2d and 3d visualisations of song Libraries
- although there hasn't been research into making these usable from a software application point of view

This project will employ three different organisational models:

- **Clustered Table** →
- **Static Graph** → each song is mapped onto a cartesian grid (both 2D and 3D) where the co-ordinates of the song is determined by the numerical attribute for that axis
- **Dynamic Graph** → each song is spaced out from every other song depending on how similar they are to each other, based on both metadata (artist, genre, etc.) and attributes

The aforementioned attributes are mid-level features analysed from the audiofile of each song → these include the instrumentalness, loudness, energy, etc. of a track

### 1.2.2 Replayability/Queue Building

When listening to songs, the next song to be played is chosen somewhere on a spectrum between fully deterministically (manually selected) or fully non-deterministically (randomly selected):

- fixed deterministic choices occur when the listener knows which song they want to listen to next
- when the listener doesn't know which song to listen to next then the next song should be randomly chosen (such as with Spotify's shuffle feature on a playlist)

With the advent of software now being the medium with which songs are listened to, song queues can now be generated randomly from a set of songs, such as a listener's playlist.

However, there exists no framework for creating song queues where their order is elsewhere on the spectrum than near the two ends. Essentially there is no way to control this randomness without significant overhead (such as having to create a new playlists with the desired songs).

By using the graph-based views as the foundation, this project will investigate a song-queue building algorithm which allows for more of a guided randomness, that is allowing for songs to be randomly selected under the constraints of song metadata and attributes.

This project will use a software application to act as a vehicle for testing and evaluating the aforementioned new organisational structures.

The problem is that I want to be able to see my full collection, not just see it in bits. Every time i see the collection (rather part of it) my mental experience and idea of it is influenced by what I saw I want full knowledge over the whole collection, not just vague notion of the parts

When listening to my songs I want to be able to have more movement on the spectrum  
- fully ordered/active - manually choose songs - fully random/passive - just hit shuffle



## Chapter 2

# Background

Here we will have a systematic literature review of the prior approaches to this problem.

And some more definitions and explanation



## Chapter 3

# Design





## Chapter 4

# Development



## Chapter 5

# Testing

### 5.1 Results



## Chapter 6

# Evaluation

The static graph view helped in maybe actually figuring out what was truly in the collection. Although this could also only be due to the fact that they were sitting down with me to actually look at and discuss their collection



## Chapter 7

# Project Planning Retrospective





## Chapter 8

# Conclusions

It works

### 8.1 Future Work



# Bibliography



# Appendix A: Photos

This is an appendix



# Appendix B: Code Listings

This is an appendix