PROJECT DESIGN SPECIFICATIONS FOR SOUND VISUALIZER

Ethan Huang
ethanhuang@uvic.ca

Brian Pham
npham49@uvic.ca

Benjamin Say
benbsay@gmail.com

ABSTRACT

This project aims to recreate popular visualization features similar to those built into classic multimedia programs. We seek to implement the visualization of different frequencies in music through the use of Python and Pygame.

1. INTRODUCTION

1.1 Background

During the 2000s, multimedia software such as iTunes and Windows Media Player came equipped with a functionality called visualization. Behind the scenes, the visualizer takes different frequencies over the track and generates detailed geometric sequences. Our project goal is to recreate this functionality with Python.

1.2 High-level summary

On a high level, sounds are combinations of different frequencies; through the usage of Pygame's rendering capabilities, each frequency can be mapped into different objects in a 2-dimensional space, allowing for visualization. Pygame's rendering engine allows for different visual manipulation techniques such as resizing, colour switch, and masking, which can handle complex frequency changes and combinations.

1.3 Related works

- Music Visualization using frequencies `https://gitlab.com/avirzayev/music-visualizer`
- Music Visualization GUI `https://github.com/djfun/audio-visualizer-python/`

© F. Author, S. Author, and T. Author. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** F. Author, S. Author, and T. Author, "Project Design Specifications for Sound Visualizer", submitted to *ISMIR*, 2025.

2. METHODOLOGY

2.1 Datasets

Four different datasets were chosen for testing. The goal of selecting multiple datasets was to introduce as much variance as possible into the testing to ensure the product will be effective on any type of sound.

Environmental audio was chosen for the range of frequencies it presents, as well as its relatively low volume levels. This ensures that the product is both sensitive enough and can work on a wide enough range of frequencies. Likewise, testing with the Goodsounds dataset (solo musical instruments) will help ensure that the visuals "fit" subjectively with the perception of the musicality of the audio file. For example, the audio clips of musical instruments playing scales will be used to ensure the visualizer can move linearly if necessary.

The Mixed Emotional Music Soundscape is created from an assortment of 6-second Creative Commons licensed audio clips. These clips have been used to study variance in emotional response depending on the mixing and composition of music. The Freesound loop dataset contains clips from music within a range of BPM and genre. Testing with these datasets will ensure that the visualizer can represent the perceived "tone" of the audio file.

- Environmental Audio: `https://zenodo.org/records/1069747#.Xlj0vi2ZN24`
- Solo Musical Instruments Dataset: `https://zenodo.org/records/820937#.Xlj1by2ZN24`
- Mixed Emotional Music Soundscape: `https://www.metacreation.net/projects/emo-soundscapes/`
  - Research Paper: `https://static1.squarespace.com/static/64487a14945a646fa6f7a229/t/64f94111f82e5559e265c0cb/1715876819379/2017-Emo-Soundscapes-ADatasetforSoundsca pdf`
- Freesound Loop Dataset: `https://zenodo.org/records/3967852`
  - Research Paper: `https://program.ismir2020.net/poster_2-16.html`

2.2 Tools

- Python: main programming language https://www.python.org/
- Pygame: library for graphical processing https://www.pygame.org/news
- Numpy: library for data processing https://librosa.org/doc/latest/index.html
- Librosa: library for handling sound input and processing https://librosa.org/doc/latest/index.html
- GitHub: for hosting and managing source code https://github.com/
- Overleaf: for creating reports and summaries in LaTeX https://www.overleaf.com/

3. TIMELINE, OBJECTIVES AND ROLES

3.1 Timeline and objectives

Timeline assuming the project submission date is April 4th, 2025.

1. Define Minimum Viable Product (MVP) specifications using requirements and user stories and define a set of functionalities to be present in the MVP. Initial deadline: Feb 26th, 2025
   (a) Brian Pham: Create a series of user stories for graphical output
   (b) Ethan Huang: Create a series of user stories for Graphical User Interface (GUI)
   (c) Benjamin Say: Create a series of requirements for data processing
   (d) Whole team: Agree on the functionalities to be developed for MVP
2. Develop MVP with agreed-upon functionalities. Initial deadline: Mar 10th, 2025
   (a) Brian Pham: Develop with a focus on graphical output
   (b) Ethan Huang: Develop with a focus on GUI
   (c) Benjamin Say: Develop with a focus on music data processing
   (d) Whole team: Combine work, resolve merge-conflicts
3. Initial run and bug fixes. Initial deadline: Mar 17th, 2025
   (a) Whole team: Test run against user stories and create a log of bugs and defects
   (b) Brian Pham: Fix graphical output-related bugs
   (c) Ethan Huang: Fix GUI-related bugs
   (d) Benjamin Say: Fix music data processing-related bugs
4. Create a demo for TA and Prof and receive feedback on project direction. Initial deadline: Mar 19th, 2025
   (a) Whole team: Record demo and send to TA and prof
5. Create final project specifications based on feedback. Initial deadline: Mar 22nd, 2025
   (a) Brian Pham: Create a set of specifications based on feedback for graphical output
   (b) Ethan Huang: Create a set of specifications based on feedback for GUI
   (c) Benjamin Say: Create a set of specifications based on feedback for music data processing
6. Develop the final product with agreed-upon specifications. Initial deadline: Mar 31st, 2025
   (a) Brian Pham: Develop with a focus on graphical output
   (b) Ethan Huang: Develop with a focus on GUI
   (c) Benjamin Say: Develop with a focus on music data processing
   (d) Whole team: Combine work, resolve merge-conflicts
7. Create demo, complete documentation, and submit the project. Original deadline: April 4th, 2025
   (a) Whole team: Create demo, final report, and submit project

3.2 Roles

- Brian Pham: Graphical Output Development
- Benjamin Say: Music Data Processing Development
- Ethan Huang: Graphical User Interface Development

4. REFERENCES

[1] J. Savelsberg, "Visualizing Music Structure Using Spotify Data," in *ISMIR*, 2021. [Online]. Available: `https://archives.ismir.net/ismir2021/latebreaking/000003.pdf`

[2] E. Isaacson, "What You See Is What You Get: On Visualizing Music," in *ISMIR*, 2005. [Online]. Available: `https://ismir2005.ismir.net/proceedings/1129.pdf`

[3] T. Takashi, S. Fukayama, and M. Goto, "Instrudive: A music Visualization System Based On Automatically Recognized Instrumentation," in *ISMIR*, 2018. [Online]. Available: `https://archives.ismir.net/ismir2018/paper/000063.pdf`

[4] J. Paulus, M. Müller, and A. Klapuri, "Audio-based Music Structure Analysis," in *ISMIR*, 2010. [Online]. Available: `https://ismir2010.ismir.net/proceedings/ismir2010-107.pdf`

[5] I. A. Thottathil and S. Thivaharan, "Virtual Musical Instruments with Python and OpenCV," *Journal of Ubiquitous Computing and Communication Technologies*, vol. 5, no. 1, pp. 1–20, Mar. 2023, doi: `https://doi.org/10.36548/jucct.2023.1.001`.

[6] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," in *Proc. Python Sci. Conf.*, 2015, pp. 18–24, doi: `https://doi.org/10.25080/majora-7b98e3ed-003`.

[7] S. Suman, K. S. Sahoo, C. Das, N. Z. Jhanjhi, and A. Mitra, "Visualization of Audio Files Using Librosa," in *Proc. 2nd Int. Conf. Math. Model. Comput. Sci.*, ser. Advances in Intelligent Systems and Computing, vol. 1422. Singapore: Springer, 2022, pp. 1–10, doi: `https://doi.org/10.1007/978-981-19-0182-9_41`.

[8] H. Kinsley and W. McGugan, "Creating Visuals," in *Beginning Python Games Development*. Berkeley, CA: Apress, 2015, pp. 1–20, doi: `https://doi.org/10.1007/978-1-4842-0970-7_4`.

[9] T. Ishibashi, Y. Nakao, and Y. Sugano, "Investigating Audio Data Visualization for Interactive Sound Recognition," in *Proc. 25th Int. Conf. Intell. User Interfaces (IUI '20)*, New York, NY, USA: Association for Computing Machinery, 2020, pp. 67–77, doi: `https://doi.org/10.1145/3377325.3377483`.

[10] M. Lagrange, M. Rossignol, and G. Lafay, "Visualization Of Audio Data Using Stacked Graphs," in *ISMIR*, 2018. [Online]. Available: `https://zenodo.org/records/1492531`.

[11] V.-V. Eklund, "DBR Dataset," Zenodo, Dec. 03, 2017. [Online]. Available: `https://doi.org/10.5281/zenodo.1069747`.

[12] O. Romani Picas, H. Parra Rodriguez, D. Dabiri, and X. Serra, "Good-sounds Dataset," Zenodo, Jun. 29, 2017. [Online]. Available: `https://doi.org/10.5281/zenodo.820937`.

[13] J. Fan, M. Thorogood, and P. Pasquier, "Emo-Soundscapes: A Dataset for Soundscape Emotion Recognition," in *Proc. Int. Conf. Affective Comput. Intell. Interaction (ACII)*, 2017.

[14] A. Ramiresm, "Freesound Loop Dataset," Zenodo, Jul. 30, 2020. [Online]. Available:`https://doi.org/10.5281/zenodo.3967852`

[15] P. Knees, M. Schedl, and M. Goto, "Intelligent User Interfaces for music Discovery: the past 20 years and what's to come." in *ISMIR*, 2019. [Online]. Available: `http://archives.ismir.net/ismir2019/paper/000003.pdf`