Generating original music from user-created melodies.

G5038 Final year Project - Interim Report
Candidate 246518

Department of Informatics and Engineering
University of Sussex
Supervised by Dr. Kingsley Sage
November 2023

Table of Contents

1 Introduction

In Recent Years, there has been a notable surge in the application of AI-Driven productivity tools which aim to enhance user creativity and productivity. Many of these tools enable users to augment, improve and expand their digital content with Machine-Generated elements.

Adobe's Flagship image editor, Photoshop, has recently introduced a new feature which implements this concept [1], bringing AI technology to one of the most popular applications within the creative industry [2].

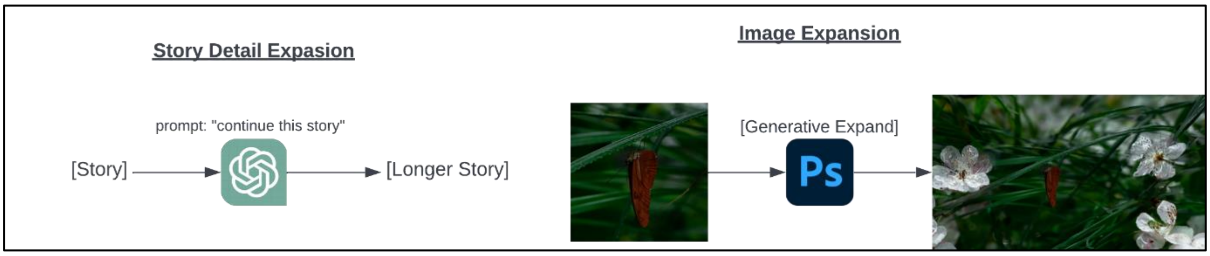
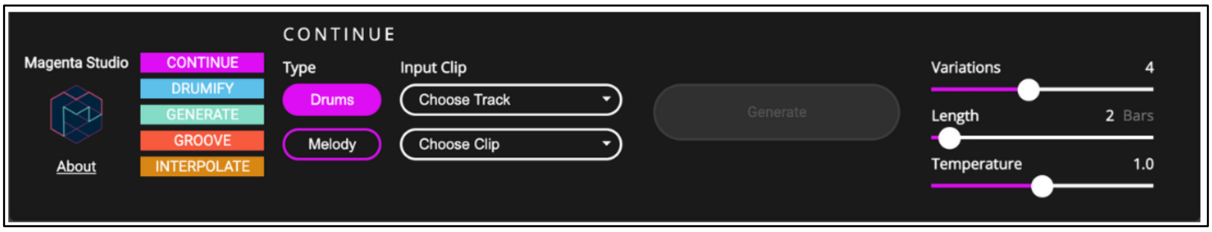


Figure 1: Examples of Content Expansion

The growing interest in AI-powered music generation research is indicative of the impact this technology is having on creative industries beyond digital art applications. We can see research initiatives coming from prominent organisations such as OpenAI [3], Google [4], Microsoft [5], and Meta [6], all with the shared goals of providing AI assistance in the music creation process.

Google's generative music research project 'Magenta Studio' stands as an interesting and noteworthy example for this project, with specific focus on providing AI tools to the widely used 'Ableton Live' music production software in the form of a set of plugins. Magenta Studio's 'continue' plugin allows users to generate MIDI Tracks that seamlessly follow user-provided melodies from a chosen MIDI Clip [7].



Variations 4

Length 2 Bars

Temperature 1.0

Figure 2: The User Interface of Magenta Studio's 'Continue' Tool.

The primary focus of this project is to design and develop a similar system to Magenta's continue tool, presented as an easy-to-use, interactive online application which aims to generate original music tracks, the composition of which will be influenced by a short melody which the user shall supply through a piano roll, for users to listen to, download and enjoy beyond the confines of a specific Digital Audio Workstation (DAW).

2 Aims

As stated within my introduction, the aim of this project is to create a web-based system that allows users to generate music which is influenced by a user-submitted melody.

I have broken this aim down into the following core, stretch and fallback objectives:

2.1 Core Objectives

1. Design and implement a website frontend which allows users to input short, note-by-note melodies into a piano roll, improving it with feedback from test users.
2. Explore various music generation techniques which can be 'flavoured' by an input melody using python notebooks, such as LSTMs, GANs, VMMs & more.
3. Develop a python backend which implements the most suitable generation technique determined from prototyping.
4. Connect the front and backend using Flask.
5. Host the tool online.
6. Evaluate the tool's effectiveness through qualitative assessment.
7. Reflect on the successes and challenges encountered during the project.

2.2 Stretch Objectives

1. Add complexity to the notes which users can input (longer notes, velocity adjustment).
2. Generate multiple melodies per user inputted melody for users to pick from.
3. Cleanly stitch the generated music onto the user's inputted melody.
4. Investigate the plausibility of implementing the system into a Digital Audio Workstation (DAW) as a plugin.

2.3 Fallback Objectives

1. Create an offline tool which allows users to input short melodies into a piano roll, producing MIDI files.
2. Research and experiment with music generation techniques to make new, original melodies.

2.4 Objective alterations

These objectives have been updated from my original project proposal, as my progress so far has given me a better idea of the project's potential scope; the capabilities of the languages & frameworks I'm using; and my own technical ability.

More detail will be provided regarding the tool's expected behaviour and functionality within Section 5.3 – Requirements.

3 Professional Considerations

This project primarily involves research and development of a web application, with minimal ethical concerns until later project stages where minimal testing shall occur.

3.1 Public Interest

The final product of this project will not contain any content which would negatively impact the public health, privacy, security or wellbeing of others and the environment. Any third-party libraries, software or ideas used will be rightfully credited and licensed for their work and contribution to this project.

My activities regarding this project will be conducted without any discrimination on the ground of sex, sexual orientation, marital status, nationality, colour, race, ethnic origin, religion, age, disability, or of any other condition or requirement.

The product of this project will ideally be a publicly accessible website that anyone can use, providing equal access to any benefits the product shall provide once completed.

3.2 Professional Competence and Integrity

The development of this project uses skills which I have already learned throughout this course. Any elements of this project that require learning new skills have been determined to be within my capabilities, as suitable university and external resources exist to acquire these skills.

I will continue to evaluate the suitability of my project's scope during development and will choose to alter my requirements as needed if any issues arise.

I will ensure that I am knowledgeable of any relevant legislation related to this project, and will avoid injuring others, their property, reputation, or employment by any false, malicious, or negligent action or inaction. Any forms of bribery or unethical inducement I encounter during this project will be rejected and reported to my supervisor.

3.3 Duty to Relevant Authority

I will avoid situations that may create a conflict of interest between me and my project supervisor and follow any guidelines that are imposed by my supervisor or the University.

I will not disclose confidential information without my project supervisor's permission, except as required by legislation.

I will be honest about my project's performance, avoiding misrepresentation or withholding of information, and will not exploit others' lack of knowledge or experience, unless bound by a duty of confidentiality.

4 Background Research

So far within this project, most of my efforts have been focused on development rather than research. My initial research has covered the following topics and technologies:

4.1 MIDI & .mid Files

Unlike a typical audio file which stores a waveform, MIDI, short for Musical Instrument Digital Interface, operates descriptively, storing instructions for musical performance which can be played by a piece of software called a MIDI player. [8]

These instructions which are known as 'events', with MIDI files consisting of tracks containing many timestamped 'events' as well as a metadata-rich 'header' containing information like the file's time signature, key signature, tempo, and copyright information.

There are many types of events, but for this project I will mainly be considering 'KeyOn' and 'KeyOff' events, which correspond to the activation and release of a specific note on a virtual instrument. [9]

These KeyOn and KeyOff events are easily added, removed, and manipulated using DAWs, and more importantly for this project, code.

Many external libraries exist to create and manipulate MIDI tracks through code, allowing for easy experimentation with MIDI files. I've been familiarising myself with their structure by tinkering with Mido [10], a python library I'm using within a python notebook, and have found that they have the following structure:

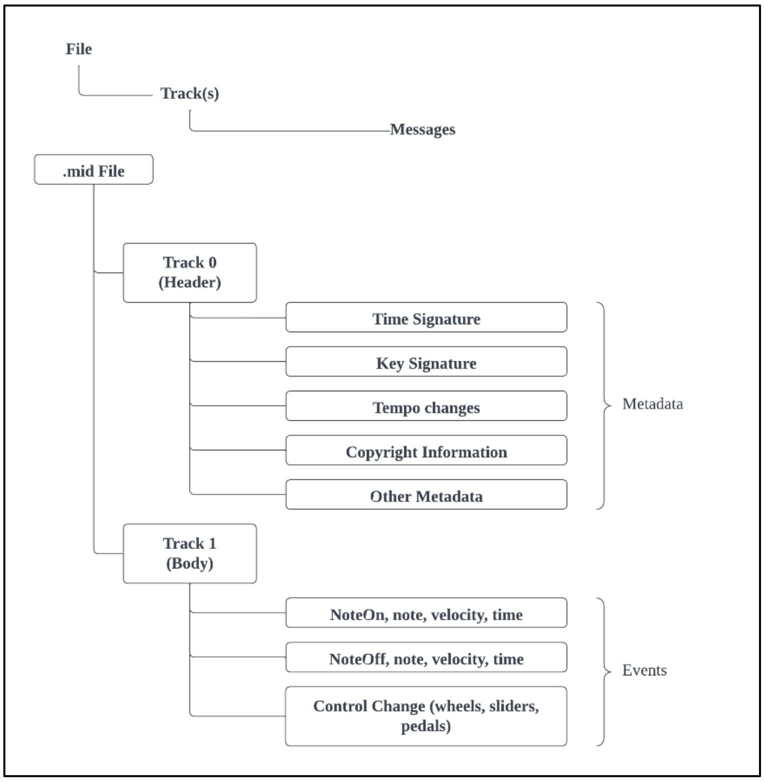


Figure 3: A tree visualisation of a MIDI File's structure.

Reading and writing new music into these files is as easy as adding or removing messages from the file's tracks, making MIDI a suitable format for this melody extension project.

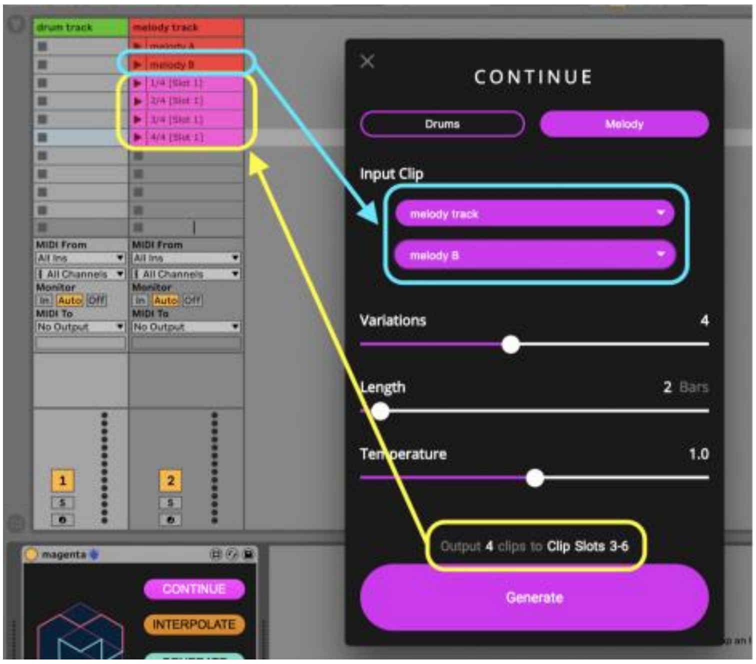
The abundance of MIDI datasets online ([11], [12], [13] to name a few) also adds merit to MIDI within this project, given the amount (and variety) of datasets which are available for training purposes, with some of the AI Music projects mentioned in my introduction using MIDI at steps of their generation process.

4.2 Magenta, Magenta Studio & 'Continue'

Magenta is an open-source research project formed by Google AI to explore the role of machine learning as a tool in creative processes [7]. Magenta Studio is one of their larger projects which is aimed at building AI tools for Ableton's 'live' Digital Audio Workstation, in the form of a set of downloadable plugins.

Magenta Studio consists of 5 plug-ins which all provide unique functionality to generate melodies and drum tracks from user input. 'Continue' is the most relevant plug-in to this project.

Continue is a compositional tool that can be used to generate notes that are likely to follow an input drum beat or melody. The user provides an input clip, and the plug-in will extend it by up to 32 measures. This plug-in can help a user add variation to a drum beat or create new material for a melodic track, typically picking up on things like durations, key signatures, and timing. Users can control how close the continuation will stick to the input pattern by adjusting the temperature used during sampling [7]. Continue's music generation is using a Long-Short-Term-Memory Network (LSTM), yielding impressive results. [14]



Input Clip: melody track, melody B

Variations 4

Length 2 Bars

Temperature 1.0

Output 4 clips to Clip Slots 3-6

Figure 4: Example usage of Continue. The user selects an input clip (annotated in blue) and the plug-in outputs multiple continuations in the same track (annotated in yellow). Source: [7]

As stated in my introduction, continue’s functionality is a primary inspiration for my project and I hope to achieve similar results to what Magenta outputs.

4.3 – Music Generation Techniques

Much of the work I’ve done so far in this project has been dedicated towards frontend development, rather than backend research. I’ve done some surface level reading on some generation techniques (primarily LSTMs) and aim to do more thorough research on other techniques after writing this report (see section 6 – project plan).

Below, I have summarised some techniques which I am interested in experimenting with:

4.3.1 – LSTMs

[15] [16] [17]

The order of notes and the timing of their occurrence are crucial elements of a piece of music. Long Short-Term-Memory networks (better known as LSTMs) are a type of Recurrent Neural Network (RNN) which are specifically designed to process sequential data. LSTM Networks are made up of memory cells which can ‘remember’ patterns over long sequences, making them suitable for tasks where the order of data matters, such as creating notes in music.

These memory cells can cause the generation of a new note to be influenced by previous notes in a sequence, which can be previously generated by the LSTM itself, or can be supplied by a user, which is clearly relevant to this project.

[18] and [17] are walkthroughs which I aim to follow and play around with to gain a better understanding of how to create and use LSTMs in Python.

4.3.2 – Other Generation Techniques to Research

I have been recommended by my supervisor to investigate other music generation techniques, so I’ve found a set of resources which I aim to read and work through to improve my understanding of how each of these generation techniques can be used, and to inform me of the most suitable generation technique for my project’s backend.

GANs

Generative Adversarial Networks (GANs) use a generator to generate new musical examples while a discriminator is used to classify whether these examples are real or fake. The generator tries to produce samples that the discriminator cannot distinguish from real examples, and vice versa. This process continues until the generator produces samples that the discriminator cannot distinguish from real samples, producing human-like output. [19]

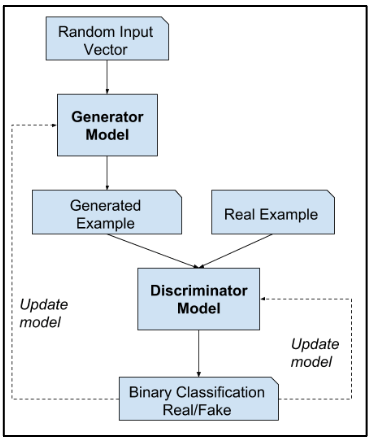


Figure 5: Example of the Generative Adversarial Network Model Architecture, Source: [19]

Relevant resources: [19], [20]

Markov Chains

In a Markov Chain, each note or chord within a piece of music is considered a state. How a Markov Chain transitions from one state to another (to select the next note/chord to be played) is determined by a probability distribution, based on the current state. This probability distribution is calculated based on the frequency of occurrence of each note/chord following a particular note/chord from a training dataset.

Once the model is trained, it can generate new music by starting with a random note or chord and then choosing the next note or chord based on the learned probability distribution. [21]

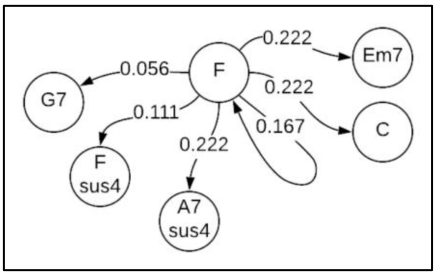


Figure 6: An illustration of the probabilities of different chords, given a chord is played. Source: [21]

Relevant Resources: [21], [22]

5 Requirements Analysis

5.1 Project Motivation

The motivation behind this project is two-fold,

Applying my skills to this project is an exciting yet challenging opportunity which will help me build upon and fully exercise the skills I’ve learned during my time at Sussex.

It is also a personal goal of mine to learn how to develop and host web content, I've never tried doing web development before and I believe that this project is a great opportunity to pick this vastly applicable skill up.

5.2 Project Relevance

This project will borrow and build upon skills which I have acquired during my time at Sussex, including soft skills like algorithmic thinking, software engineering skills, programming skills and hard skills such as my proficiency with Python and Machine Learning libraries.

As my course has an Artificial Intelligence focus, I believe this project will be a good reflection of the skills and knowledge I have acquired during my course to any parties interested in my degree.

5.3 Requirements

Below I have created tables of required and desirable requirements for the tool. Minor details of the tool's functionality may be subject to change based on any new discoveries made while prototyping.

5.3.1 Frontend Requirements

Requirement	Requirement Description	Required/ Desirable	Functional/ Non-Functional
F1	Users shall be able to input notes onto a piano roll using the mouse.	Req.	F
F2	The piano roll shall be suitable size for entering short melodies, currently determined to be 24 notes (A to G over 2 octaves) which can be placed over 16 beats, henceforth known as the 'base melody'.	Req.	F
F3	Notes could be more or less than 1 beat in length, inputted by pressing and dragging on the piano roll.	Des.	F
F4	Inputted notes could be moved around the piano roll by clicking and dragging.	Des.	F
F5	Users shall be able to play and stop the playback of their inputted base melody using a play/stop playback button.	Req.	F
F6	Users could be able to change the number of bars their melody will generate.	Des.	F
F7	Users could be able to change a 'chaos' parameter of the backend implementations using a slider, if one is implementable.	Des.	F
F8	Users shall be able to change the Tempo of the base melody they've inputted from a suitable range using a slider, currently determined to range from 60 Beats per Minute (BPM) to 240 Beats per Minute.	Req.	F
F9	Users shall be given audio feedback when a note is inputted.	Req.	F
F10	Users shall be able to clear their inputted base melody using a button.	Req.	F
F11	Once a user has inputted their base melody, they shall be able to submit their melody to be extended using a button.	Req.	F
F12	The tool shall provide a visual representation of any generated melodies.	Req.	F
F13	The tool shall provide an opportunity for users to download any generated melodies.	Req.	F

F14	The UI shall be intuitive and not require any sort of tutorial for use.	Req.	NF
F15	Any audio which is played from the site shall be free from distortion and lag.	Req.	NF
F16	Inputted notes shall be in a colour which is distinguishable from the background	Req.	NF

Table 1: Frontend Requirements.

5.3.2 Backend Requirements

Requirement	Requirement Description	Required/ Desirable	Functional/ Non-Functional
B1	Once a user has submitted a Melody, an original melody shall be generated by the website's backend, henceforth referred to as 'the generator'.	Req.	F
B2	The generator shall generate a random, 'un-influenced' melody if the piano roll is left empty.	Req.	F
B3	Generated melodies shall be 'influenced' by the melody which the user supplies through the piano roll	Req.	F
B4	The generator could generate multiple different melodies using the same base melody, allowing the user to select from a variety of generated melodies.	Des.	F
B5	Generation wait times shouldn't be longer than 15 seconds	Req.	NF

Table 2: Backend Requirements.

5.3.3 Other Requirements

Requirement	Requirement Description	Required/ Desirable	Functional/ Non-Functional
O1	The tool shall be hosted online and accessible through the web.	Req.	NF

Table 3: Other Requirements.

6 Project Plan

Within my project proposal I mentioned a Notion workspace which I planned to use for week-by-week planning within my project. I've since abandoned the planning side of that workspace due to the amount that this project has changed week-by-week based on new research discoveries and testing/prototyping.

I've instead created a GANTT chart which defines a more general plan of the project's stages which I aim to follow, rather than detailed week-by-week tasks, which I found difficult to stick to.

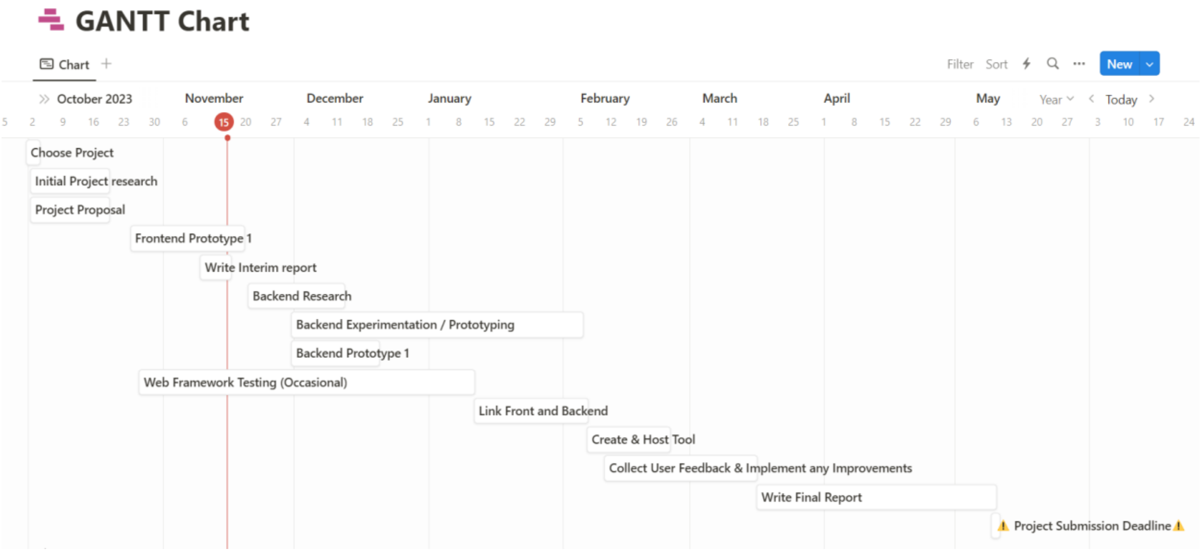


Figure 7: The Project's GANTT Chart

7 Development Progress to Date

My initial focus in the project has been primarily on the tool’s frontend. I made the decision to get the frontend out of the way towards the beginning of the project as I had no prior experience with frontend development when starting.

This lack of experience has pushed me towards using standard HTML, CSS, and JavaScript for my webpages, as I’d rather learn these core skills before moving onto something more complex like React.

This early learning process has provided me with valuable insights into the project’s overall structure and has been a great learning experience to crack down on the basics of web development.

I’m happy with my current rate of progress and will soon be shifting priority away from UI and Frontend development, as most parts of the frontend are in place to allow for unimpeded backend prototyping.

Aside from my frontend focus, I’ve also invested time into shaping the overall structure of the project. I’ve undertaken a high-level approach to designing and refining plans for the tool’s overall structure and operation. To illustrate, I’ve created a flowchart below that outlines the key elements and the operation of the tool, as well as the proposed languages and frameworks I will use.

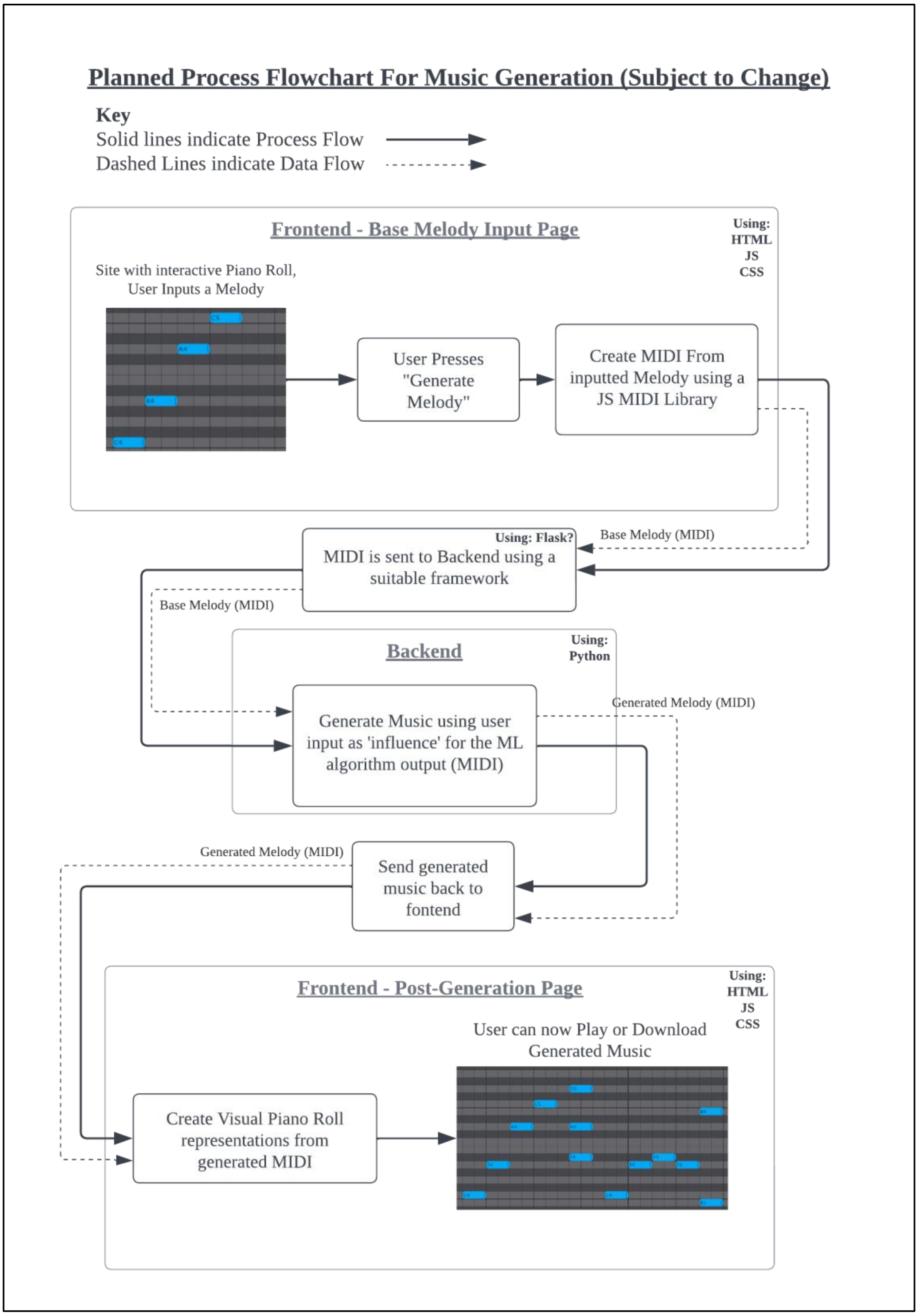

Planned Process Flowchart For Music Generation (Subject to Change)
Key
Solid lines indicate Process Flow
Dashed Lines indicate Data Flow
Frontend - Base Melody Input Page
Using: HTML JS CSS
Site with interactive Piano Roll, User Inputs a Melody
User Presses "Generate Melody"
Create MIDI From inputted Melody using a JS MIDI Library
Base Melody (MIDI)
Using: Flask?
MIDI is sent to Backend using a suitable framework
Base Melody (MIDI)
Backend
Using: Python
Generate Music using user input as 'influence' for the ML algorithm output (MIDI)
Generated Melody (MIDI)
Generated Melody (MIDI)
Send generated music back to fontend
Frontend - Post-Generation Page
Using: HTML JS CSS
User can now Play or Download Generated Music
Create Visual Piano Roll representations from generated MIDI

Figure 8: A High-Level flowchart of the tool's operation process.

This broader perspective will be important to ensure the project's cohesiveness and to give me a well-rounded understanding of how the frontend and backend components will work with each other during future development.

I've experimented with using Django as a web framework for linking the project's frontend and eventual backend, but quickly found that it's 'batteries included' nature was rather overkill, with it including tools which I've found no need for within this project. I've been recommended to use Flask as a lightweight alternative by my peers, and surface level reading of online developer forums and flask's own documentation [23] makes it seem like an attractive option for this project, which I aim to investigate.

7.1 Development Progress – Prototype 1

My work so far has culminated in a first prototype for the tool's frontend.

Most of my progress so far has been dedicated towards the site's interactive piano roll and functionality surrounding it. Progress in this area has been slow but extremely valuable as I've spent most the time learning JavaScript, HTML and CSS from scratch.

The underlying structure of the site was based off a loosely followed YouTube tutorial for a basic web-based minesweeper board [24],which has since been altered and built upon to create an interactive piano roll with a BPM slider and play/clear buttons.

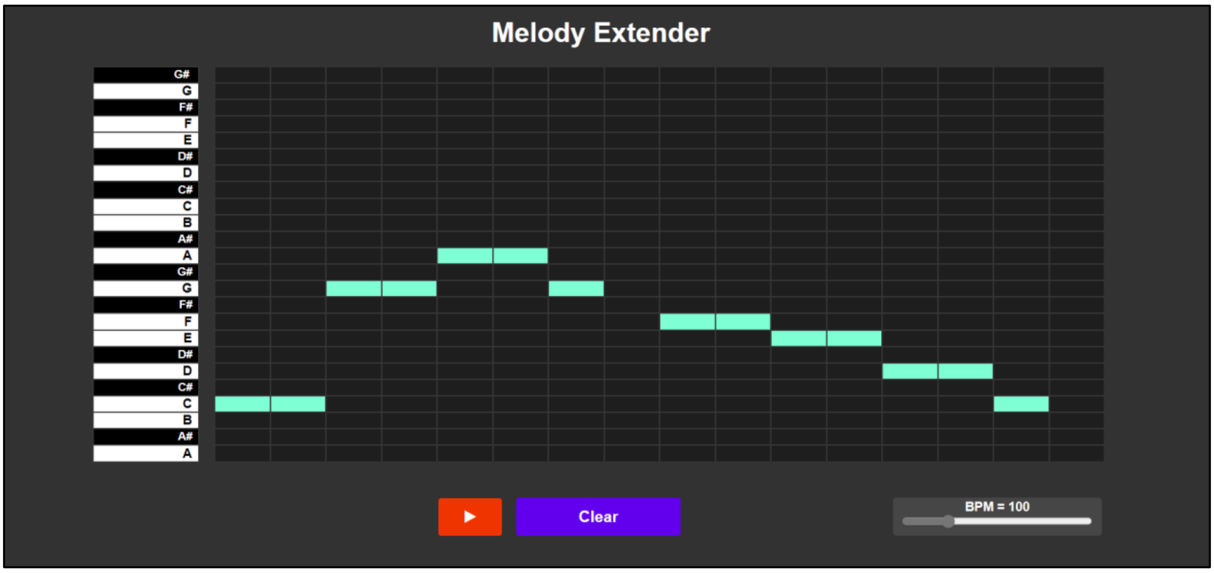


Figure 9: A Screenshot of the tool with 'twinkle twinkle little star' inputted.

This prototype has the following feature set:

- Users can input individual notes onto the piano roll using the mouse. Each part of the grid representing a certain note at a timing.
- Inputting a note will sound an appropriate noise from a digital piano.
- There is a visual representation of a piano's keys to the left of the grid, aimed at assisting users in positioning their notes.
- Users can change the BPM of their melody using the BPM slider.
- Pressing the play button towards the bottom of the screen will play the user's melody at the specified BPM, from left to right.
- A 'clear' button can be found next to this play button, which clears the piano roll when pressed.

As well as this melody input page, I've also made a skeleton for a results page which will be shown after the user's melodies have been generated.

As of right now, the shaded sections are empty, but will eventually contain visual representations of the music generated by the tool's backend. Each result has a play and download button which currently have no functionality. Work on this page will resume once I have made a suitable amount of progress with my backend to get it to generate 'flavoured' melodies from user input.

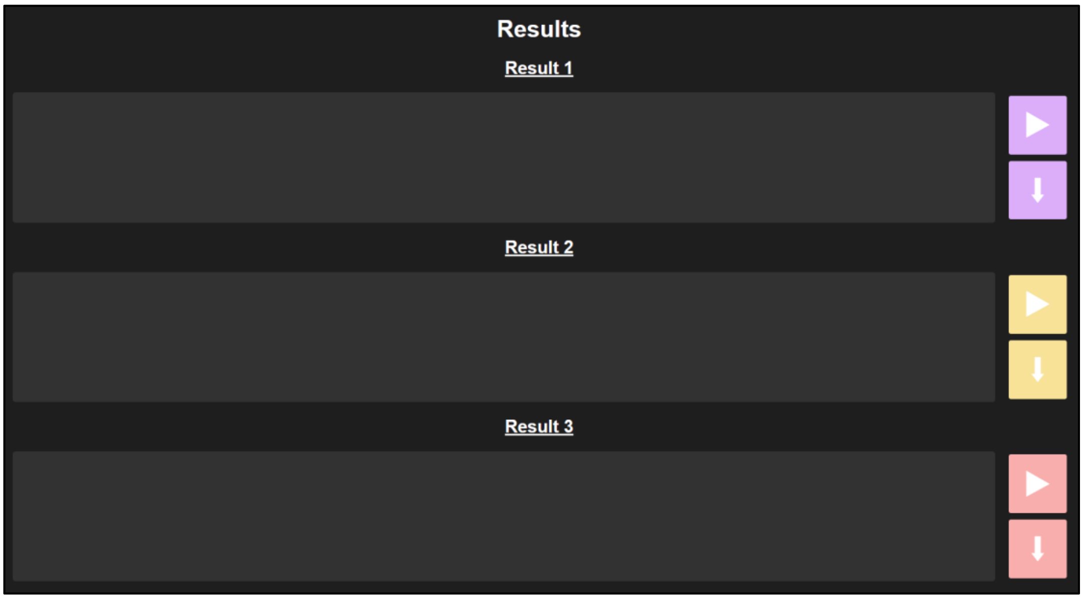
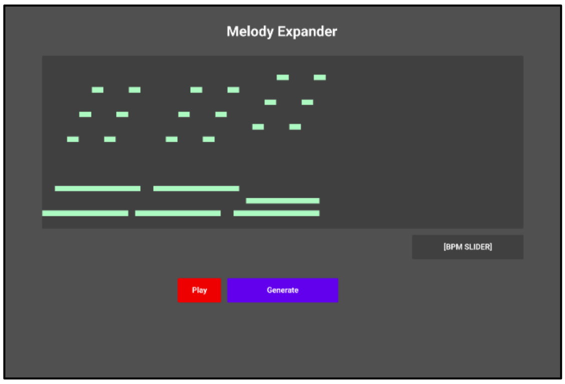


Figure 10: The current 'skeleton' implementation of the Results page.

All the progress I have made with the frontend has been based on designs I made earlier in the project using Figma. The original designs for the Piano roll and results pages can be seen below.



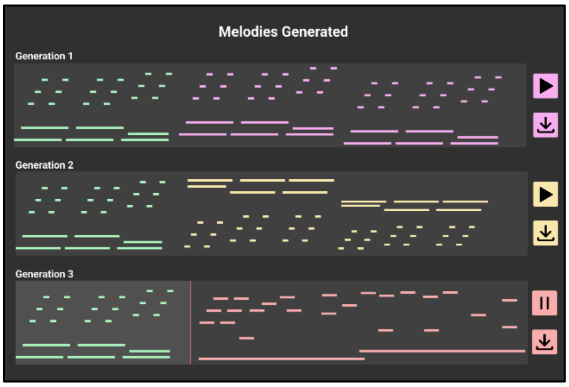


Figure 11a & 11b: Designs of the Piano roll & results pages.

I'm quite happy with how close the frontend currently looks to my designs, given my lack of web dev experience.

7.1.1 Issues

As this is a prototype, there are some noticeable bugs, especially when it comes to audio playback on the piano roll page.

Current known bugs to date are:

- Identical notes do not give audio feedback when they are placed back-to-back in rapid succession.
- Identical notes do not play during playback if placed back-to-back.
- Pressing the play button during playback will break the player.

Fixing these bugs isn't at a very high priority as they won't have any impact on the MIDI files that the frontend will eventually create. They'll most likely be fixed later when refining the frontend in a later prototype.

7.2 Next steps

As stated beforehand, most parts of this prototype are functional enough to allow for backend prototyping to begin. I aim to add one more feature which will write the inputted melody into a series of note events into a MIDI file using MIDI-Writer-JS. After this feature is added, I will be completely shifting my focus away from frontend development and towards researching the music generation research techniques discussed in Section 4.3 for my backend, with the aim to develop my first backend prototype which will generate music by late December.

8 References & Appendices

8.1 References

[1] J. Schneider, "Photoshop's New 'Generative Fill' Uses AI to Expand or Change Photos," PetaPixel, 23 May 2023. [Online]. Available: https://petapixel.com/2023/05/23/photoshops-new-generative-fill-uses-ai-to-expand-or-change-photos/. [Accessed 17 October 2023].

[2] Enlyft, "Computer graphics and photo editing software market share worldwide in 2023, by product," 17 July 2023. [Online]. Available: https://www.statista.com/statistics/1223620/graphics-photo-editing-software-market-share/. [Accessed 14 November 2023].

[3] C. Payne, "MuseNet.," OpenAI, 25 April 2019. [Online]. Available: https://openai.com/research/musenet. [Accessed 19 October 2023].

[4] Google Brain Team, "Welcome to Magenta!," 1 June 2016. [Online]. Available: https://magenta.tensorflow.org/blog/2016/06/01/welcome-to-magenta/. [Accessed 18 October 2023].

[5] Microsoft, "AI Music," 1 August 2019. [Online]. Available: https://www.microsoft.com/en-us/research/project/ai-music/. [Accessed 14 November 2023].

[6] F. K. I. G. T. R. D. K. G. S. Y. A. A. D. Jade Copet, "Simple and Controllable Music Generation," Arxiv, 7 November 2023. [Online]. Available: https://arxiv.org/abs/2306.05284. [Accessed 14 November 2023].

[7] J. E. Y. M. J. G. C. K. S. N. M. D. C. R. C. H. D. E. Adam Roberts, "Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live," in *MUME*.

[8] The MIDI Association, "About MIDI-Part 1: Overview," [Online]. Available: https://www.midi.org/midi-articles/about-midi-part-1-overview. [Accessed 2 November 2023].

[9] The MIDI Association, “About MIDI-Part 4: MIDI Files,” [Online]. Available: https://www.midi.org/midi-articles/about-midi-part-4-midi-files. [Accessed 2 November 2023].

[10] S. Agnew, “Working with MIDI data in Python using Mido,” Twilio Blog, 18 July 2017. [Online]. Available: https://www.twilio.com/blog/working-with-midi-data-in-python-using-mido. [Accessed 18 October 2023].

[11] B. Krueger, “Classical Piano MIDI Page,” 1996. [Online]. Available: http://www.piano-midi.de/.

[12] D. McKenzie, “Jazz Piano MIDI Library,” [Online]. Available: https://bushgrafts.com/midi/.

[13] E. F. P. M. M. a. G. w. t. M. Dataset, “Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, Douglas Eck,” International Conference on Learning Representations, 2019. [Online]. Available: https://openreview.net/forum?id=r1lYRjC9F7.

[14] Google AI, “Magenta Studio (Ableton Live Plugin) 2.0,” 24 August 2023. [Online]. Available: https://magenta.tensorflow.org/studio#continue. [Accessed 18 October 2023].

[15] R. M. P. J. Sanidhya Mangal, “LSTM Based Music Generation System,” *IARJSET,* vol. 6, no. 5, pp. 47-54, 2019.

[16] L. G. K. A. D. H. R. S. A. N. Michael Conner, “Music Generation Using an LSTM,” 2022. [Online]. Available: https://arxiv.org/abs/2203.12105. [Accessed 14 November 2023].

[17] V. Velardo, “Generating Melodies with LSTM Nets: Series Overview,” Youtube, 4 May 2020. [Online]. Available: https://youtu.be/FLr0r-QhqH0. [Accessed 14 November 2023].

[18] R. Bali, “LSTMs for Music Generation,” Towards Data Science, 1 July 2021. [Online]. Available: https://towardsdatascience.com/lstms-for-music-generation-8b65c9671d35. [Accessed 11 November 2023].

[19] J. Brownlee, “A Gentle Introduction to Generative Adversarial Networks (GANs),” Machine Learning Mastery, 19 July 2019. [Online]. Available: https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/. [Accessed 14 November 2023].

[20] W.-Y. H. L.-C. Y. Y.-H. Y. Hao-Wen Dong, “MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment,” in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI),* 2018.

[21] A. Osipenko, “Markov Chain for music generation,” Towards Data Science, 8 July 2019. [Online]. Available: https://towardsdatascience.com/markov-chain-for-music-generation-932ea8a88305. [Accessed 14 November 2023].

[22] R. G. Conrad Hsu, “Comparative Assessment of Markov Models and Recurrent Neural Networks for Jazz Music Generation,” 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2309.08027.

[23] Pallets, “Flask Documentation,” [Online]. Available: https://flask.palletsprojects.com/en/3.0.x/. [Accessed 31 October 2023].

[24] K. Y. Coding, “Build Minesweeper with Javascript HTML CSS,” Youtube, 14 March 2022. [Online]. Available: https://www.youtube.com/watch?v=AfhfAxKFP-s. [Accessed 24 October 2023].

8.2 Project Proposal

Project Background

In recent years, there has been a notable surge in the application of AI-Driven productivity tools to enhance user creativity and productivity. Many of these tools enable users to augment their digital content with AI-Generated elements, Adobe’s generative expand tool in photoshop being a prominent example.

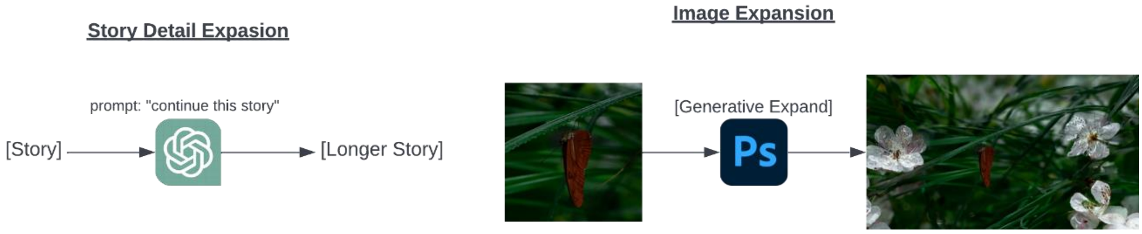


Figure 12: Examples of content expansion

The recent trend has sparked my curiosity and led to a question: Can this concept be applied to music production? It seems that researchers have the same question and have created research initiatives and projects with this aim. Google’s generative music research project ‘Magenta Studio’ stands as an interesting and noteworthy example, with specific focus on providing AI tools to the widely used Ableton music production software in the form of a plugin.

Magenta's ‘continue’ tool directly addresses the question I've posed, allowing users to generate music that seamlessly follows user-provided melodies within a chosen track.

Objectives

The general aim of this project is to develop a similar online musical tool akin to magenta’s ‘continue’ tool, allowing the generation of ‘original’ music based on a user-provided melody.

Specifically, the project will investigate the following primary, extension and fallback objectives:

Primary objectives

1. Research and evaluate existing work in the field of AI Music generation to gather an idea of currently existing technologies.
2. Identify Successes and Failures of existing AI Music tools in professional software, research studies and online toys.
3. Design and plan the tool’s frontend using the insights gained from Objectives 1 & 2.
4. Explore the effectiveness of various backend implementations through prototyping.
5. Conduct an assessment to determine how successful the tool is.
   (As it stands, I would like the tool to create melodies which are indistinguishable from human made melodies. This may be subject to change).

6. Reflect on the successes and challenges encountered during the project.

Extension Objectives

1. Modify the tool to generate music around the user's melody, rather than creating a separate track.
2. Enhance the tool to provide multiple melodies, giving users the option to select from a variety of generated tracks.
3. Enable users to export their generated melodies as MIDI files for importation into music production software.

Fallback Objectives

1. Create a web-based tool for users to create short MIDI melodies.

Relevance

This project's significance lies in its potential to aid music creation, empowering users to generate original compositions from their own melodies with the assistance of AI technology.

The objectives I've outlined above will serve a dual purpose to drive the development of my tool while also providing an opportunity for me to assess and enhance my existing technical skills and learn new languages and frameworks and parts of computing which are currently a mystery to me.

I currently have technical experience with machine learning libraries using Python from my modules in prior years and slight UI design experience from personal projects, with no experience creating web content.

As it stands, these skills alone will not allow me to complete this project. I anticipate that I will need to learn the following skills and technologies to be able to fulfil my primary and extension objectives:

- Proficiency in HTML, CSS and JavaScript for the Tool's frontend.
- Familiarity with a framework for integrating the frontend with A python backend (Django seems like an attractive option).
- Website Design.
- Website Hosting.
- Fundamental knowledge of music theory.

Resources Required

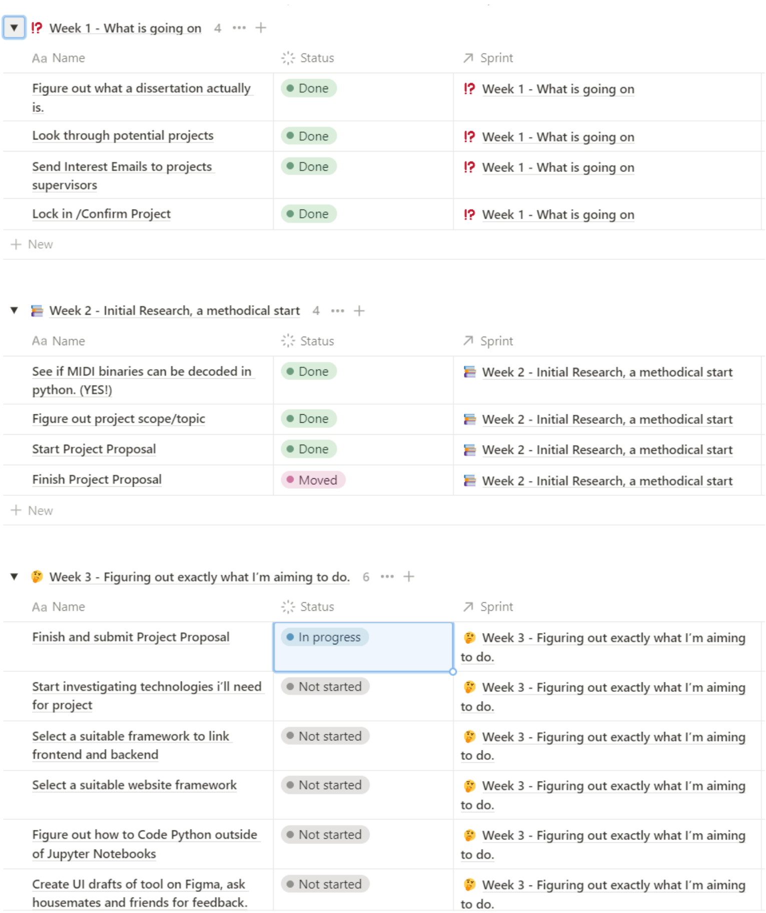
This project may involve reserving study or meeting rooms for conducting user testing on an as-needed basis.

Although I currently lack experience in web hosting, I anticipate the need for some sort of online hosting resource to make my tool available on the web. I plan to specify the necessary resources once I research the hosting strategy for my tool.

Planning

I've created a notion workspace to facilitate my project's planning process, as I've found it to be an extremely helpful tool for organisation in a prior project (Software engineering Group 4's documentation).

My intention is to use this workspace as a central hub to track my background research, references, project timeline, weekly work updates and supervisor meetings. Below is a screenshot of one part of my workspace, the current task list, which shows my week-by-week tasks.



▼ ⁉ Week 1 - What is going on 4 ··· +

Aa Name	Status	Sprint
Figure out what a dissertation actually is.	● Done	⁉ Week 1 - What is going on
Look through potential projects	● Done	⁉ Week 1 - What is going on
Send Interest Emails to projects supervisors	● Done	⁉ Week 1 - What is going on
Lock in /Confirm Project	● Done	⁉ Week 1 - What is going on

+ New

▼ 📚 Week 2 - Initial Research, a methodical start 4 ··· +

Aa Name	Status	Sprint
See if MIDI binaries can be decoded in python. (YES!)	● Done	📚 Week 2 - Initial Research, a methodical start
Figure out project scope/topic	● Done	📚 Week 2 - Initial Research, a methodical start
Start Project Proposal	● Done	📚 Week 2 - Initial Research, a methodical start
Finish Project Proposal	● Moved	📚 Week 2 - Initial Research, a methodical start

+ New

▼ 🤔 Week 3 - Figuring out exactly what I'm aiming to do. 6 ··· +

Aa Name	Status	Sprint
Finish and submit Project Proposal	● In progress	🤔 Week 3 - Figuring out exactly what I'm aiming to do.
Start investigating technologies i'll need for project	● Not started	🤔 Week 3 - Figuring out exactly what I'm aiming to do.
Select a suitable framework to link frontend and backend	● Not started	🤔 Week 3 - Figuring out exactly what I'm aiming to do.
Select a suitable website framework	● Not started	🤔 Week 3 - Figuring out exactly what I'm aiming to do.
Figure out how to Code Python outside of Jupyter Notebooks	● Not started	🤔 Week 3 - Figuring out exactly what I'm aiming to do.
Create UI drafts of tool on Figma, ask housemates and friends for feedback.	● Not started	🤔 Week 3 - Figuring out exactly what I'm aiming to do.

Figure 13: A screenshot of the tasks from weeks 1-3 of my notion workspace.

I currently do not have an overarching plan for the whole year. I would like to discuss this in a meeting to get an idea of how long each stage of a project like this would take.

Availability

Below is my calendar's typical weekday plan. Each day is subject to change and may differ slightly from how I have planned it. My timetable will of course change between terms.

My work shifts are consistently on Monday, Tuesday and weekends. I've made it clear to my work about my academic commitments and will not hesitate to quit if my job interferes too much with my studies.

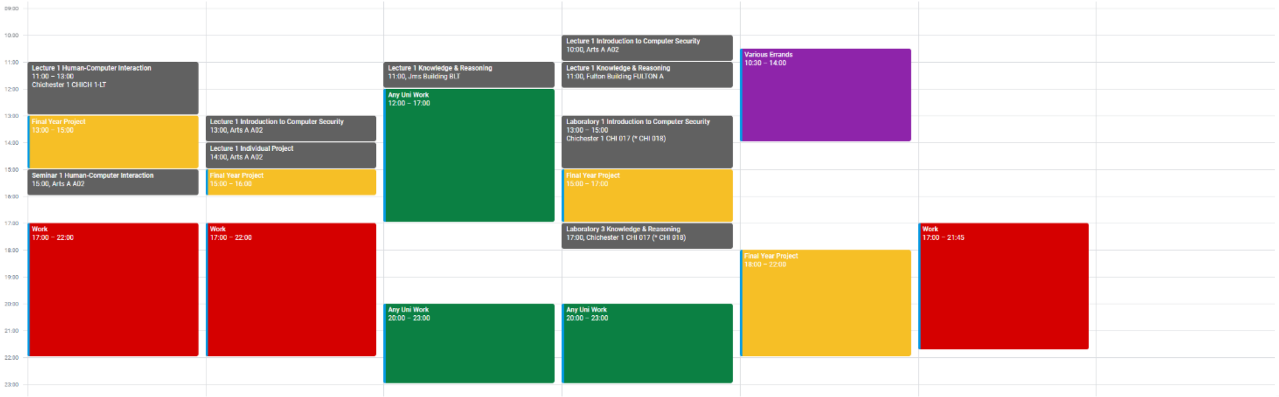
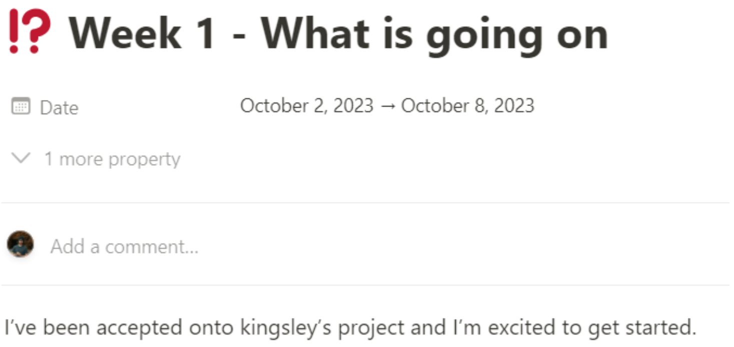


Figure 14: Calendar Availability

I am available for meetings at any time that I am not in a lecture or at work.

8.3 Weekly Log



⁉ Week 1 - What is going on

Date October 2, 2023 → October 8, 2023

1 more property

Add a comment...

I've been accepted onto kingsley's project and I'm excited to get started.

📚 Week 2 - Initial Research

Date October 9, 2023 → October 15, 2023

1 more property

Add a comment...

Things went quite slow this week. I've been dealing with a mouldy house and ended up getting covid on the final half of the week which pretty much floored any plans I had with my project.

I've looked at the structure of MIDI files and see that they're quite easy to grasp, as well as throwing around ideas about what my project should actually be.

🤔 Week 3 - Figuring out exactly what I'm aiming to do.

Date October 17, 2023 → October 21, 2023

1 more property

Add a comment...

I've not got covid any more so that's something, I'm in uni now and can start working on my project without feeling like i'm going to perish.

18.11.2023

Has been good, I've completed my proposal a day before submission and I think I've fully figured out my project scope.

🌻

Week 4/5 - Frontend Time

Date	@October 23, 2023 → November 5, 2023

Using The below video for the Grid.

Build Minesweeper with Javascript HTML CSS

Learn how to code minesweeper in javascript html and css!
Mine: 💣
Flag: 🚩

https://www.youtube.com/watch?v=AfhfAxKFP-s

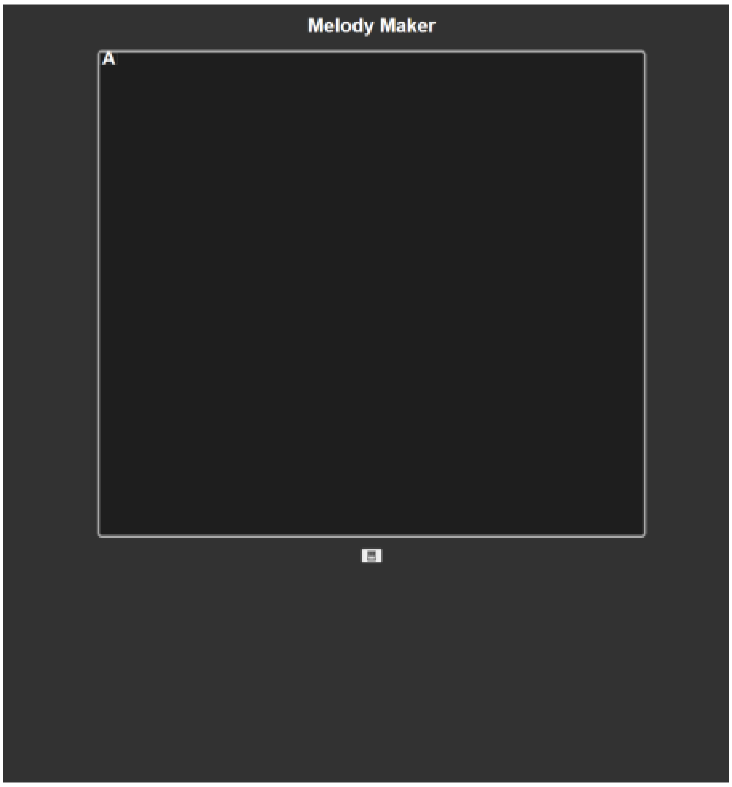
G#
G
F#
F
E
D#
D
C#
C
B
A#
A

Let's say we want 3 octaves, we'll need 36 keys.

~~Each key will be around 32px * 12px~~ keys now use viewport width and height to disable issues with sizing

24 Keys over 16 "Time blocks"

~~Dimensions will be 432 * 512 (doubled.)~~



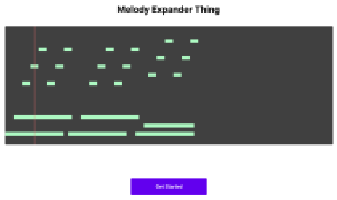
Clicked values are stored within a 2D bool array if the note is clicked or not.

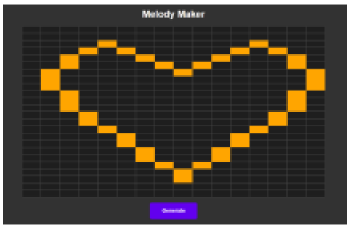
Notes can be clicked on or off.

I think this should be changed into a “note on note off” type representation rather than a 2D bool Array due to its similarity with MIDI Note-on Note-off events, making it easier.

This may come to bite me in the ass if i don’t do it sooner rather than later.

Progress looks pretty good compared to Original Design



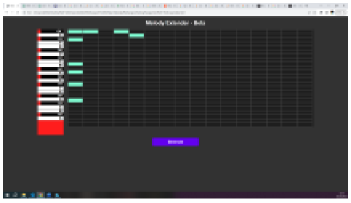


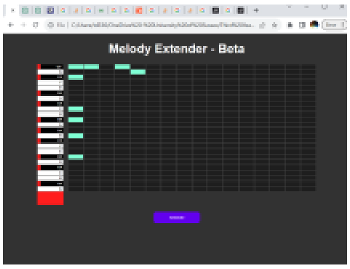
25.10.2023

Made progress on grid, changed everything from being Pixel values to ViewWidth values

Using the pixel values introduced issues when scaling the site.

Swapping to Viewwidth/Height values has fixed those issues and now allows the site to scale.

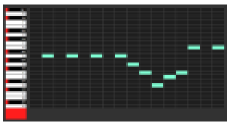


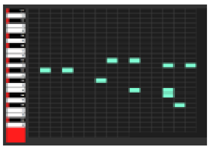


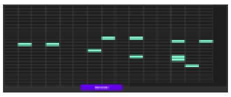
However, this has introduced a couple of issues.

When scaling the site, the piano grid sometimes goes from 16 rows wide to 15 rows wide.

This completely messes up the visual representation of the inputted melody.

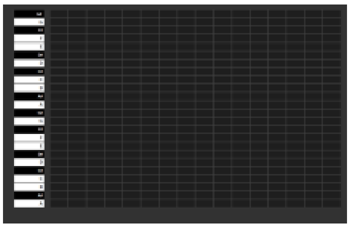


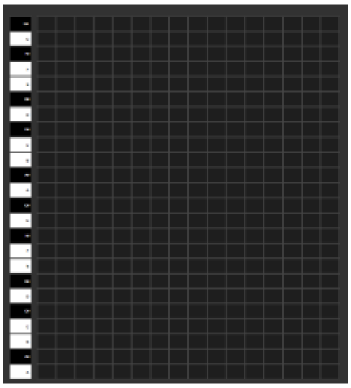


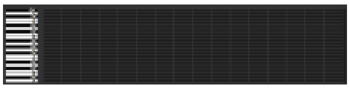


27.10.2023

Fixed the scaling issues by using a 2D div grid instead of 1 big list of div which wraps around the box.







There are still some issues with text scaling but the grid itsself is now fine.

All elements (apart from the title and some corner radii) are now sized using vertical height and vertical width, this helped to remove some box scaling issues

Added Sounds to the keyboard when notes are inputted

(note: inputted sounds currently are clunky, delayed and can't play at the same time, I need to find a fix for this.)

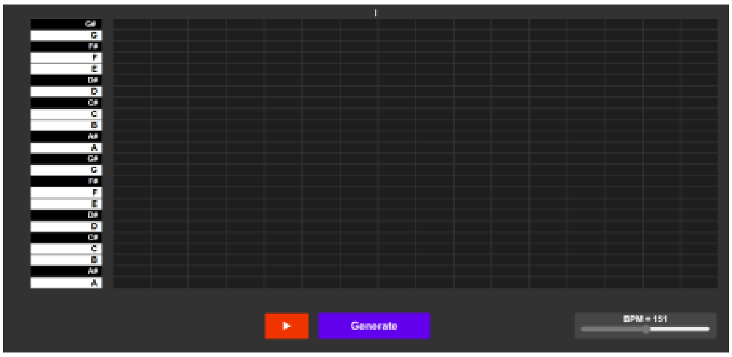
30.10.2023

Added functionality to clear the board when “generate” is pressed.

31.10.2023

Added dynamic BPM slider which corresponds to a JS variable.

Added a play button at the bottom of the screen, currently has no functionality



Plan: Work on playing user melodies using the play button.

1.11.2023

Added Player Functionality!!!

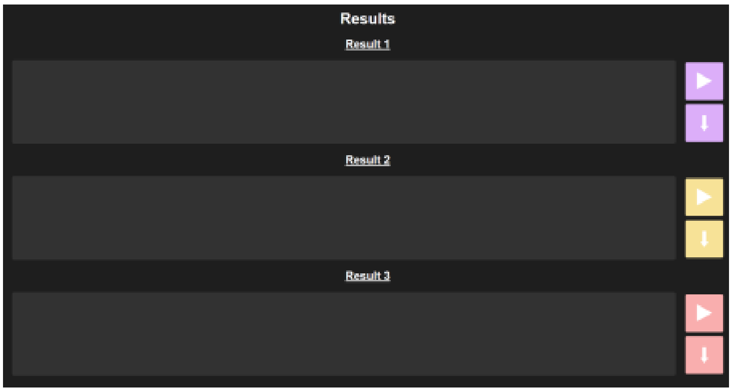
Struggling with same notes not playing at the same time due to how JS/HTML audio works

Still need to add the player head to board, thinking of using a red right border on playing elements. Will be annoying to figure out.

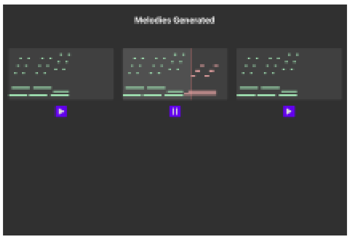
Given that I've progressed so far, i may have to change my fallback objectives. ("revised" objectives).

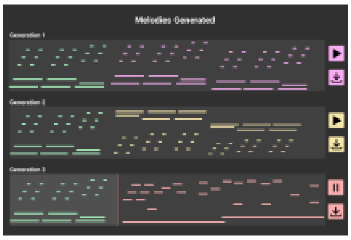
08.11.2023

I've added a second page which will show after the music has generated.



The design has closely followed the design I made on Figma. (first design also pictured below, scrapped due to an excess of unused space.)





📝 Week 6 & 7: Interim Report

Date November 6, 2023 → November 16, 2023

1 more property

Add a comment...

Doing my interim report has pushed me to do more research about backend implementations which has helped a lot.

I've not done any sort of development progress just due to how busy i've been on my interim.

I'll just have to implement MIDI-Writer-JS onto my frontend and i'll be able to fully focus on my backend.

I've received feedback from kingsley on my first draft and have tried to implement the imporvements

I'm aiming to finish and submit on the 16th to not get a late penalty.

8.4 Supervisor Meetings

20.10.2023

Had a zoom session with Kingsley to discuss my project progress so far.

Went through project proposal and all was well with that.

Discussed my approach to the project (research vs half/half vs full development) and concluded that I'm doing a sort of half/half project.

Have been recommended some papers and generation techniques.

08.11.2023

Had an in person meeting with Kingsley to discuss process so far.

Mentioned how I've been struggling with writing and taking the first step towards completing the report. Suggestion about starting with top-down progress kicked me into gear and helped me get into a writing state.

Went over my prototype, both of us agreed that It has enough functionality and that I can start focussing on my backend.

Understood why I've ditched Django for a more lightweight framework.