**Milestone P0 — Project Proposal and Motivation**

**Project Title: Symbolic Music Generation using Transformer-Based Architectures**

**Author:**

Cade Stocker  
University of South Carolina  
Email: cstocker@email.sc.edu

**Problem Statement and Motivation**

Music generation using artificial intelligence has become an active area of research, driven by advances in deep learning and symbolic music representation. However, most state-of-the-art AI music generators such as **MusicLM** and **Riffusion** produce *raw audio waveforms*, which are difficult to edit or interpret. This limits their usefulness to composers and producers who rely on **MIDI (Musical Instrument Digital Interface)** — a symbolic representation that encodes note, velocity, and timing information in a structured format.

MIDI-based generation offers interpretability and flexibility, allowing human musicians to edit or rearrange compositions directly within Digital Audio Workstations (DAWs). Despite this advantage, symbolic generation remains challenging: models must learn long-term musical structure, handle polyphony, and maintain coherence across measures and instruments.

This project aims to explore **transformer-based and recurrent models** for symbolic music generation using MIDI data. By following the architecture proposed in **Musenet**, which employs separate **discriminator** and **generator** components, the goal is to reproduce and extend the approach on a manageable scale. Using a smaller dataset such as the **Nottingham folk tune collection** (~1,200 songs), this project enables rapid experimentation and comparative evaluation of model architectures.

**Related Work**

Early neural music generation methods primarily used **Recurrent Neural Networks (RNNs)** and **Long Short-Term Memory (LSTM)** architectures to capture temporal dependencies in sequences. For example, Yang et al. [1] introduced **MidiNet**, a convolutional generative adversarial network for symbolic-domain music generation, achieving coherent melodies but limited long-term structure.

The **Musenet** architecture from OpenAI combines transformer-based sequence modeling with discriminative and generative stages to capture longer musical dependencies. Recent works like **Text2MIDI** [2] and **XMusic** [3] explore controllable symbolic generation, integrating natural language inputs or user-defined constraints. This project originally aimed to replicate the Text2MIDI architecture to create MIDI files based on a user’s textual description.

Survey papers such as Briot et al. [4] and Zhu et al. [5] highlight the evolution from RNN-based to transformer-based models and the growing focus on controllability and structure. However, there is still limited exploration of **lightweight, interpretable architectures** trained on smaller symbolic datasets for efficient local experimentation.

This project builds upon these foundations by re-implementing and adapting Musenet’s [7] concepts — specifically, a **generator–discriminator** pair — to MIDI-level data with a focus on **quantitative evaluation** (e.g., note density, pitch range, polyphony).

**Initial Hypotheses**

* **Hypothesis 1:** Transformer-based generators can produce more coherent and musically structured sequences than LSTM or GRU-based models when trained on symbolic (MIDI) data.
* **Hypothesis 2:** Incorporating a discriminator that selects harmonic context (e.g., chords for subsequent measures) improves tonal consistency and reduces abrupt transitions between measures.
* **Hypothesis 3:** Even with a small dataset like Nottingham, training lightweight models can yield measurable improvements in symbolic composition metrics such as polyphony and pitch diversity.

**Proposed Solution**

The proposed system follows a **two-stage architecture** inspired by Musenet:

1. **Discriminator (Harmonic Selector):**  
   Predicts the next chord or harmonic context based on previous measures.  
   This module captures high-level harmonic structure and progression. It can be implemented as a transformer, LSTM, or MLP.
2. **Generator (Melody Composer):**  
   Produces note sequences conditioned on the discriminator’s harmonic predictions and preceding measures.  
   It can be implemented as a transformer, LSTM, or GRU-based sequence generator.

Both components will be trained using symbolic data from the **Nottingham MIDI dataset**, which contains folk tunes with clear melodic and harmonic structure. The generator output will be converted to standard MIDI files, allowing analysis and playback.

A **factory design pattern** enables easy switching between architectures for both modules, facilitating controlled comparisons between models.

**Evaluation Plan**

The system will be evaluated along both **machine learning** and **musical quality** dimensions:

**1. Quantitative Evaluation**

Using the PrettyMIDI and Music21 libraries, generated MIDI outputs will be analyzed based on:

* **Note density** – notes per measure
* **Pitch range** – min/max note spread
* **Polyphony** – simultaneous note count
* **Duration statistics** – average note length
* **Chord consistency** – alignment between predicted and generated harmonies

**2. Qualitative Evaluation**

Generated sequences will be listened to and visually inspected (via piano roll visualizations) to assess:

* Harmonic coherence
* Rhythmic stability
* Repetition vs. variation

**3. Baseline Comparison**

Baseline models (e.g., LSTM without a discriminator) will be compared against transformer-based architectures using both metrics and subjective evaluation.

All experiments will be conducted on a **MacBook Air M1** with **PyTorch 2.9**, using **MPS acceleration**.

**References**

[1] Yang, L., Chou, S., & Yang, Y. (2017). **MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation.** *arXiv:1703.10847*  
[2] Bhandari, K., Roy, A., Wang, K., Puri, G., Colton, S., & Herremans, D. (2024). **Text2MIDI: Generating Symbolic Music from Captions.** *arXiv:2412.16526*  
[3] Tian, S., Zhang, C., Yuan, W., Tan, W., & Zhu, W. (2025). **XMusic: Towards a Generalized and Controllable Symbolic Music Generation Framework.** *arXiv:2501.08809*  
[4] Briot, J., Hadjeres, G., & Pachet, F. (2017). **Deep Learning Techniques for Music Generation — A Survey.** *arXiv:1709.01620*  
[5] Zhu, Y., Baca, J., Rekabdar, B., & Rawassizadeh, R. (2023). **A Survey of AI Music Generation Tools and Models.** *arXiv:2308.12982*  
[6] Raffel, C. (2016). *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching.* Ph.D. Thesis, Columbia University.

[7] Pandey, R., Kaur, G., & Mehra, P. (2022). **Musenet: Music Generation using Abstractive and Generative Methods.** ResearchGate preprint. [Online]. Available: <https://www.researchgate.net/publication/363856706_Musenet_Music_Generation_using_Abstractive_and_Generative_Methods>