Graphical Music Compiler

A Domain-Specific Language for Visual Musical Expression

Esteban Alejandro Villalba Delgadillo, Daniel Felipe Barrera Suarez
Universidad Distrital Francisco José de Caldas, Bogotá D.C.

Abstract

This project proposes the design of a graphical-musical interpretive system that transforms structured textual instructions into visual representations of musical notation. Inspired by domain-specific language (DSL) principles, the system compiles simple commands into *piano roll* diagrams and audio playback using Python libraries like `matplotlib` and `pygame`.

Background

Domain-specific languages enable intuitive mappings between human intent and computational behavior. In music, DSLs like Sonic Pi or TidalCycles offer powerful tools, but often require steep learning curves.

Our approach simplifies this by defining a minimal grammar:

```
play C4 for 1.0 at 0.0
play E4 for 0.5 at 1.0
```

This grammar is interpreted to generate a time-pitch grid known as a **piano roll**, widely used in DAWs (Digital Audio Workstations).

Objectives

The main objective of this project is to develop a graphical music compiler capable of interpreting simple textual instructions and converting them into a piano roll-type visual representation using Python and Pygame. It aims to facilitate the creation of visual compositions from textual commands and to explore the intersection between programming, music, and visual representation.Design a symbolic interpreter for musical phrases. Enable text-based musical input. Visualize music using `matplotlib`. Play notes using `pygame.mixer`.

Methodology

The methodology begins with the definition of a minimal symbolic language that allows users to describe notes and their durations. For instance, inputs such as `["C4-1/4", "E4-1/4", "G4-1/2"]` are parsed using a simple recursive descent parser. The interpreter, implemented in Python, uses `pygame.mixer` to play notes sequentially. Once parsing is successful, a visualization is generated using `matplotlib`, where each note is represented as a bar on a piano roll diagram. Future work includes expanding the syntax to support chords, loops, and live manipulation.

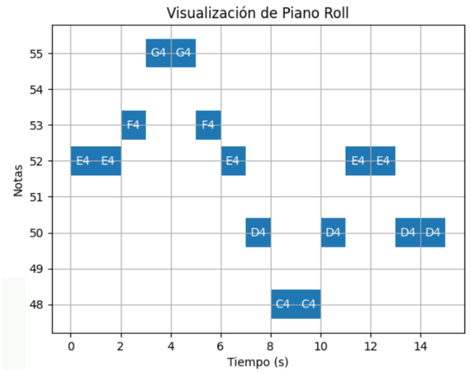

Figure 1: Example of piano roll visualization

Scope and Limitations

Scope:

This project focuses on rendering simple monophonic musical sequences using textual instructions. It supports basic rhythmic values such as quarter and eighth notes and operates at a fixed tempo, currently set at 120 beats per minute.

Limitations:

The system has several limitations due to its prototype nature. It does not support polyphonic textures or real-time interaction, and playback is restricted to .wav audio files. Additionally, the project does not generate traditional sheet music notation but instead relies solely on a piano roll-style visual representation.

Results to Expect

- Successfully implemented a functional parser.
- Generated clear visualizations for musical patterns.
- Enabled basic playback using predefined sound clips.

Conclusions

The development of the graphical music compiler demonstrates the feasibility of connecting programming and music in an accessible and educational way. Using a simple language, it interprets and visualizes basic musical instructions, promoting the learning of both computational and musical concepts. Although the system has limitations, its educational focus and expansion potential make it a valuable tool for interdisciplinary teaching.

References

[1] McLean, A., & Wiggins, G. (2010). *Tidal — Pattern language for live coding of music*. Proceedings of the 7th Sound and Music Computing Conference.

[2] Sonic Pi. https://sonic-pi.net

[3] FoxDot. https://foxdot.org

[4] Pygame Documentation. https://www.pygame.org/docs

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS