Procedural Music

McKade Umbenhower, Robert Randolph, Taylor Bleizeffer

COSC Senior Design, University of Wyoming

Introduction

Project Goal:
- Algorithmically generate new music based off a single input

Project Questions:
- Can an algorithm produce listenable music?
- Can we utilize the texture synthesis algorithm, Wave Function Collapse, to make the music?

Benefits and Uses:
- Inspire music creation
- Provide brand new, original music to listen to
- Create on-the-fly music to be paired with games, videos, podcasts, or any other desired medium
- Recreate tunes similar to other pieces

Tools Used:
- Java
- JMusic – Open source music programming library

Project Challenges:
- Determine the feature set to be used from the sample to produce more music
- Find a good similarity balance between the input and output

Related Work

Wave Function Collapse:
- Texture Synthesis
- Model Generation

Other procedural music:
- Markov Chains
- Cellular Automata

References:

Brown, Andrew R. JMusic – Computer Music Composition in Java. explodingart.com/jmusic/.

mxgmn. "mxgmn/WaveFunctionCollapse." GitHub, 14 Nov. 2018, github.com/mxgmn/WaveFunctionCollapse.

WolframTones, Wolfram Research, Inc., 2018, tones.wolfram.com/about/how-it-works/.

Implementation

- Input a sample

Midi File

E	0							0								
C		0					0			0						0
A			0		0						0		0			
F				0								0				

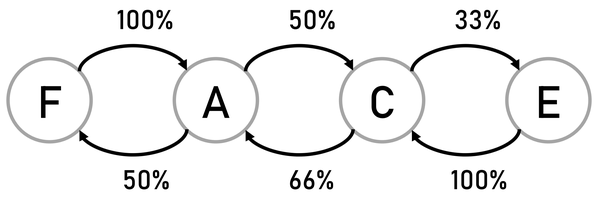
- Generate Markov Tables/Markov Chains

Markov Table

	E	C	A	F
E	-	2	-	-
C	1	-	2	-
A	-	2	-	2
F	-	-	2	-

Markov Chains

	E	C	A	F
E	-	100%	-	-
C	33%	-	66%	-
A	-	50%	-	50%
F	-	-	100%	-



- Apply Wave Function Collapse (WFC)

Super Position of all Notes

E	.	.	.	.	.	.	.	.	.	.	.	.
C	.	.	.	.	.	.	.	.	.	.	.	.
A	.	.	.	.	.	.	.	.	.	.	.	.
F	.	.	.	.	.	.	.	.	.	.	.	.

Collapsing on Super Position Note

E	.	.	.	.	.		.	.	.	.	.	.
C	.	.	.	.	.	0	.	.	.	.	.	.
A	.	.	.	.	.		.	.	.	.	.	.
F	.	.	.	.	.		.	.	.	.	.	.

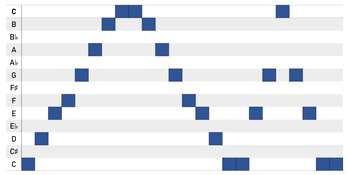
Constraint Propagation

E	.		.		.		.		.		.	
C		.		.		0		.		.		.
A	.		.		.		.		.		.	
F		.		.				.		.		.

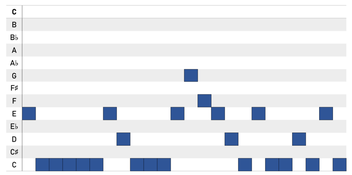
- Repeat WFC until all notes are collapsed

Results

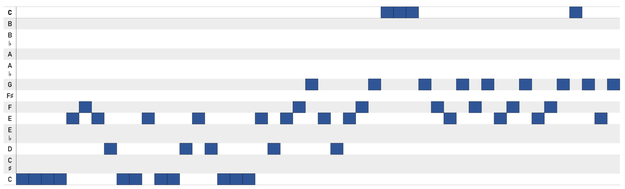
Current music productions:
- Input a simple sample with fixed-length notes
- Output a locally similar, original piece that inherits features from the sample
- Each run of the algorithm produces a new output
- Controllable output length



Example input sample



Short example output



Longer example output

Takeaways:
- Our current implementation is a good proof on concept
- The algorithm's current output music has a lot of room for improvement
- Lots of feature tuning will be needed

Future Work

User interface:
- Add a GUI
- Add note visualization
- Options to toggle sampling features
- Input/Save music

Improving the music:
- Note length
- Chords
- Phrasing
- Multiple Parts
- Noise