Daniel Ortega

CPSC 490 Senior Project Proposal

Advisor: Scott Petersen

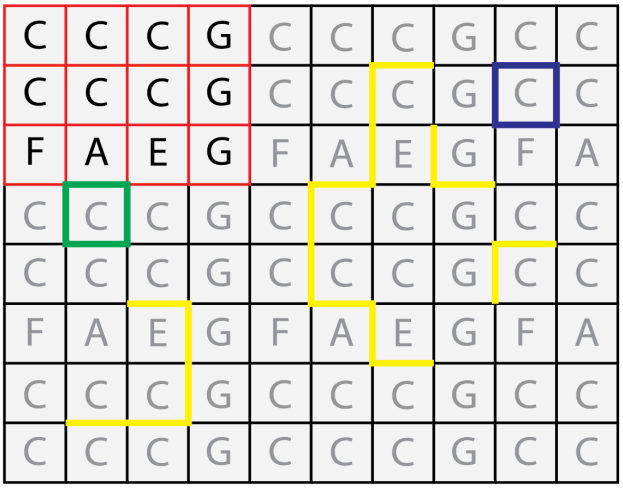
Making Music with Pathfinding Algorithms

Abstract

When first learning how pathfinding algorithms work, many students can be confused how different algorithms search the sample space. This naturally has led to the creation of pathfinding visualization programs. While these visualizations often help people better understand the general concept of how each algorithm searches for a path, adding some other kind of representation might be able to further enhance that understanding.

As a result, I propose creating a musical visualization of each algorithm as it progresses from start to finish. Within a given grid made up of cells, each cell will contain a musical note which will be played when an algorithm "visits" that cell. This should produce unique musical output for various pathfinding algorithms, distinguishing between them. Note that for different pathfinding algorithms, “visiting” a cell can easily vary in its definition, so this could feasibly be a customizable option.

Initially, the inputs to the program will be the following: which pathfinding algorithm is used, where walls are placed, the start cell, and the end cell. Grid values will be fixed to certain selectable presets, but eventually this can naturally expand to customizable grids.



Possible grid of cells: Red = tiled repetition of cells, Yellow = walls, Green = start cell, Blue = end cell

Primary Goal

The minimum viable product (MVP) will be a program with a default grid of notes. The user will select a certain algorithm from a menu (A*, Dijkstra's, etc.), select start and end cells, and optionally draw walls into the grid for the pathfinding algorithm to avoid. Upon starting the algorithm, the program should play a note whenever it visits a cell as a candidate in any way, as opposed to after it has found a particular path. After a path is found, or one is impossible, the user might edit the start cell, end cell, walls, and algorithm, and then begin the program again. This should be accompanied by some sort of visual indication that that cell was visited in real time, such as a change in color of the cell.

Stretch Goals

Feature	*Description*	*Priority*
Customizable grid cells	Allow the user to edit the note played by any given cell	High
Alternate grid layouts	E.g., triangular or hexagonal grid	High
Edit instrumentation	Allow the user to change what instrument is being played by the program	High
Parallel grid(s) which affect other musical features	Determine note duration, note amplitude, and/or high-level chord progressions from separate grid(s)	Medium
Saving and loading	Save and load wall/grid/start/end layout from a file	Medium
Exporting to mp3	Record the music outputted by the program and save as an mp3	Low
Machine learning which “plays” the musical pathfinding visualization	Train another program to place walls on the grid to make music that sounds "similar to" some input	Low

Deliverables

1. Abstract
2. Project Proposal
3. Project Report
4. Video demo of program
    a. Approximately 5-minute overview

5. GitHub repository
    a. All source files and code, including audio files and any other sample output of program

Timeline

Finish:	*By:*
Project Proposal	Feb 15
Set up code repository and research/begin building structure of program	Feb 22
Build MVP	Mar 5
Modify MVP to fit stretch goals or unforeseen needs	Mar 31
Begin final report, wrap up work on the program	Apr 15
Finalize code repository, finish final report	Apr 29

Resources

Papers

Collins, Tom, David Meredith, and Anja Volk. "Mathematics and Computation in Music." *Proceedings of 5th International Conference*, MCM. Vol. 9110. 2015.

Conklin, Darrell. "Music generation from statistical models." *Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*. 2003.

Lee, Sangkeun, et al. "Random walk based entity ranking on graph for multidimensional recommendation." *Proceedings of the fifth ACM conference on Recommender systems*. 2011.

Tymoczko, Dmitri. "The generalized tonnetz." *Journal of Music Theory* (2012): 1-52.

Wang, Mengsha, et al. "RNDM: A random walk method for music recommendation by considering novelty, diversity, and mainstream." *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI).* IEEE, 2018.

Relevant Classes

- CPSC 366 - Intensive Algorithms
- CPSC 431 - Algorithmic Computer Music
- CPSC 474 - Computational Intelligence for Games

Models and Examples

- https://github.com/reddragonnm/pathfinding-music
- https://qiao.github.io/PathFinding.js/visual/