

Initial Plan - 2D Game: connect all tiles with a continuous path

Author: Adam Kosak

Supervisor: Martin Caminada

Moderator: George Theodorakopoulos

2021-CM3203 One Semester Individual Project - 40 credits

Project Description

The basic premise of the project is to create a simple game based around trying to fill a grid in one continuous path. The main inspiration for this is the snake game, or rather its hypothetical end goal of eventually filling the entire board with the snake's body as it gets longer (view the gif for reference: <https://i.imgur.com/oo6abA0.gif>), and various puzzles with the goal of connecting all pathways with one continuous line.

My main goal and motivation was to make an at least somewhat original game, without having to go into unnecessarily deep mechanics, and to flesh out a simple idea as best as I could.

The main difference between my idea and the traditional snake game, is that there are no randomly generated spots where the snake feeds to grow, the snake instead expands with every movement, and all tiles it was on will remain occupied until the game ends. The other differences are that snake doesn't move automatically, the player makes each move tile by tile, and each level can have pre-filled spaces in form of walls, giving the board a unique layout.

Project Aims and Objectives

Primary features:

- Basic Game (Windows Java Application)
 - display game grid in command line
 - traverse the grid within constraints
 - win condition (grid fully filled)
 - lose condition (no moves available when grid isn't filled)
 - level select
- Level creator
 - set size of the new level
 - change the type of each space (empty/wall/starting point)
 - save level and allow it to be played
- Level solver
 - implement an algorithm that finds a solution, or if a solution is possible

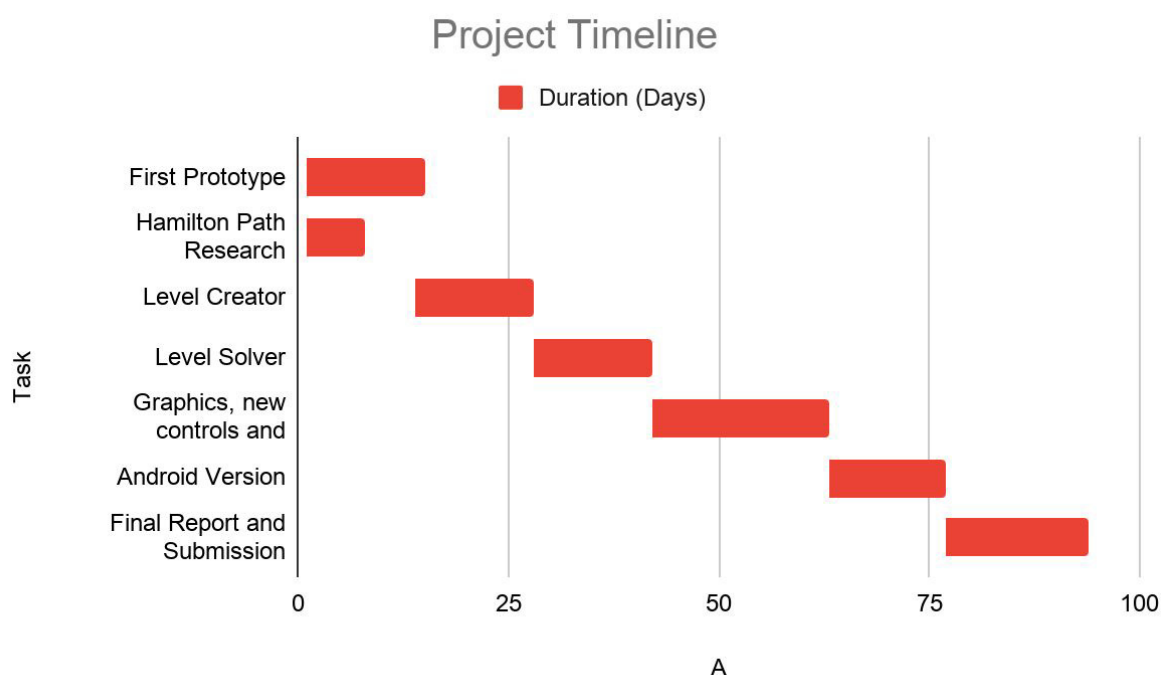
Additional features

- Level solver
 - in addition to reporting if a full solution isn't possible, report the best partial solution (most spaces filled)
- Challenge modes
 - Find X number of solutions
 - Automatic movement (like traditional snake) at various selectable speeds
- Graphical representation
 - Instead of command line display, the game features 2D graphics
- Android version
 - The game is ported and fully playable on an android system

Work Plan

For this project I'm adopting the methodology of Agile Development. The reasons for that are that I want to be flexible with what features I'm going to include by the end, and do more research about the possible extensions I want to include. While the Gantt chart presented further presents a more linear approach akin to Waterfall methodology, I want to maintain a safety net over any additional features I may or may not be able to implement during the development.

My main development and target platform is Java on Windows PC, as that is the programming language I have the most experience in, and the platform I have the easiest access to. My secondary target platform is android, as likewise I have access to development tools and devices to test on.



To go into more detail for each step:

First Prototype

February 9th to February 23rd

During this step I will create the basis for the whole program in Java. I will make the data container for my map grid, and functions to traverse it, and recognise when the game is complete, whether by victory or loss. The game has to represent and display in the command line the state of the board, recognise available moves, change the state of the

board according to the move chosen by the player and end the game once no moves are available. The victory would occur with every space of the grid being occupied, while loss would happen should there still be empty spaces on the board once end condition is satisfied. During this step I want to either test or research various solutions for handling the map data, whether by a one or two dimensional array, or a quad tree.

Hamilton Path Research

February 9th to February 16th

In the first week I also want to take some time to do research for the Level Solver and the Final Report, that being how my problem relates or can be represented by a Hamilton Path.

- Milestone 1 (*February 23rd*): Program can create a grid of selected size and traverse it, win and lose conditions are implemented

This step creates the bare minimum for the game to be playable, thus creating a solid base to expand on, while passing for a functional program to report on. Additionally, there is extra background material to add to the final report.

Level Creator and Selector

February 23rd to March 9th

During this step I have to expand the program to allow it to create new stages, save them, and load them to play. First, the new creation mode will start by asking the user for input on the dimensions of the board, its width and height. By modifying the regular traversal function for this mode, I'm going to allow it to go over every type of tile, and change the active tile to either be a wall, an empty space or a starting point. For the last one, the creator mode also has to recognise that there has to be only one starting point, and either preferably overwrite the old one. The program also has to be able to save the level to an external file, either as a compiled object or a simple text file, as well as read all the saved levels in a specified directory, and allow the user to select which one to play.

- Milestone 2 (*March 9th*): Level creator and selector complete, the game's main levels can now be provisionally developed.

Level Solver

March 9th to March 23rd

During this step I will develop the level solver. It should check whether the level currently being created can be completed, by brute forcing every possible solution, or by another method I may find from research on the Hamilton's Path. It should also be able to find and show the solution to the user for the level currently being played.

- Milestone 3(*March 23rd*): Level Solver complete, the game's levels can now be developed and automatically tested to see if they can be completed.

This step completes all primary features of the game. For that reason, I want to hold the first **review** meeting following it, on 23rd of March.

Graphics, new controls and additional features

March 23rd to April 13th

For this step I want to develop a new graphical representation of the game to replace the command line representation, and develop additional controls, including simple directional button press, or a mouse press on the selected tile. While this technically doesn't add anything to the game's functionality, it would make it significantly easier and more enjoyable to test and play. I also want to use the timeframe after I complete the primary features to try to implement other additional features I've mentioned in my aims and objectives.

Android Port

April 13th to April 27th

During this period I want to try to port the game to Android systems. Since Java is a common development platform on both Windows and Android, I'm hoping to reuse as much of my previous work as possible. While I have experience working with Android Studio in a team from last year's group project, this time I will be working on it alone.

- Milestone 4 (*April 27th*): Secondary Features development complete and Final Report first draft (Review)

Since the report is the primary object of rating, I want to have a dedicated review session to help begin writing it, as well as evaluate any of the additional work.

Final Report

April 27th to May 14th

The last weeks of the project period will be dedicated entirely to writing the report based on the development notes, documentation and feedback from the supervisor.

The final deliverables will include a functional game playable on at least one platform, with all the primary features and the final report, detailing its development.