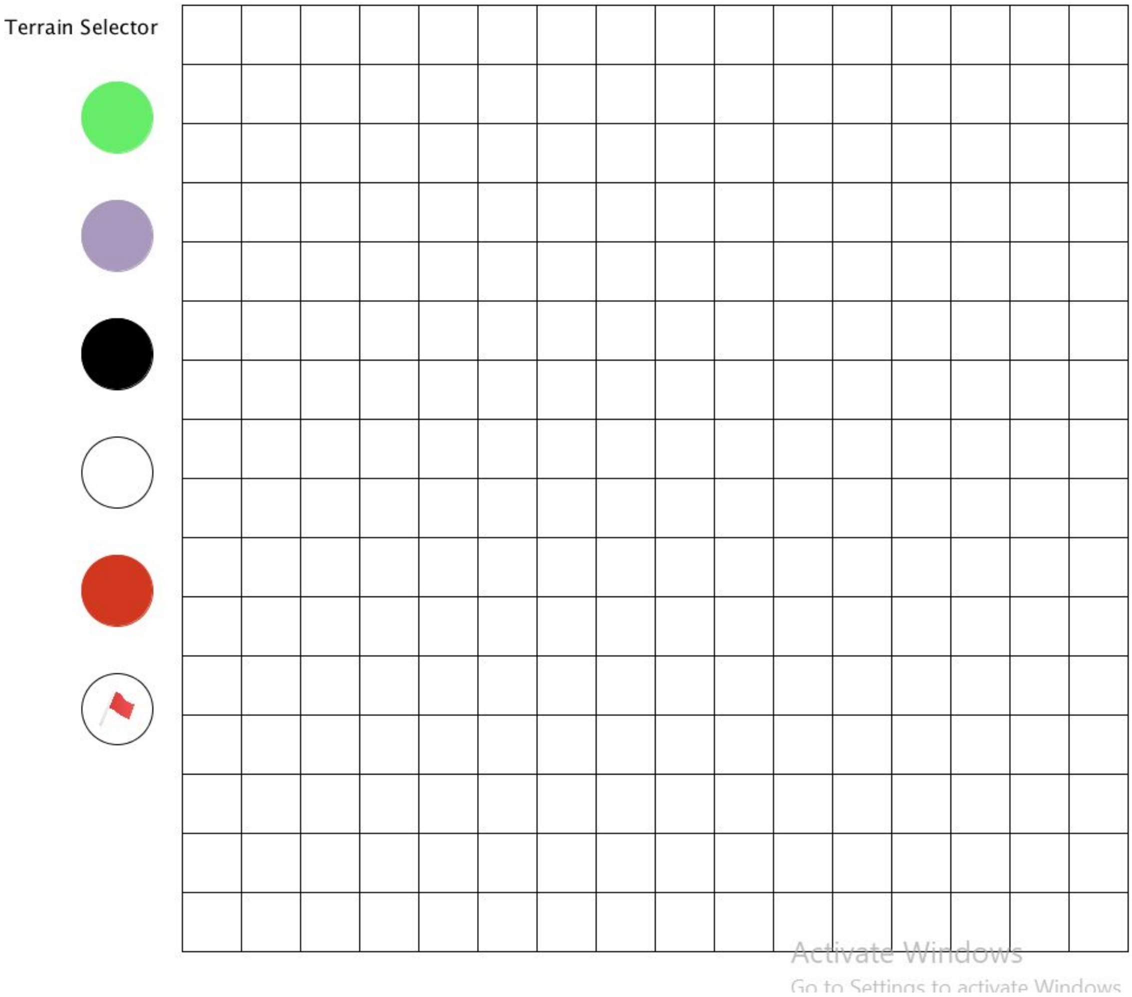
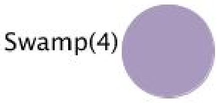
Game Description

This game is a simulation of pathfinding using the A* algorithm. This is the initial screen of the game:



You can use the terrain selector on the left to specify the type of tiles of the grid on the right. There are six types of tiles. You can first click on one of the circles to select a type, the type and cost of the tile with show up:



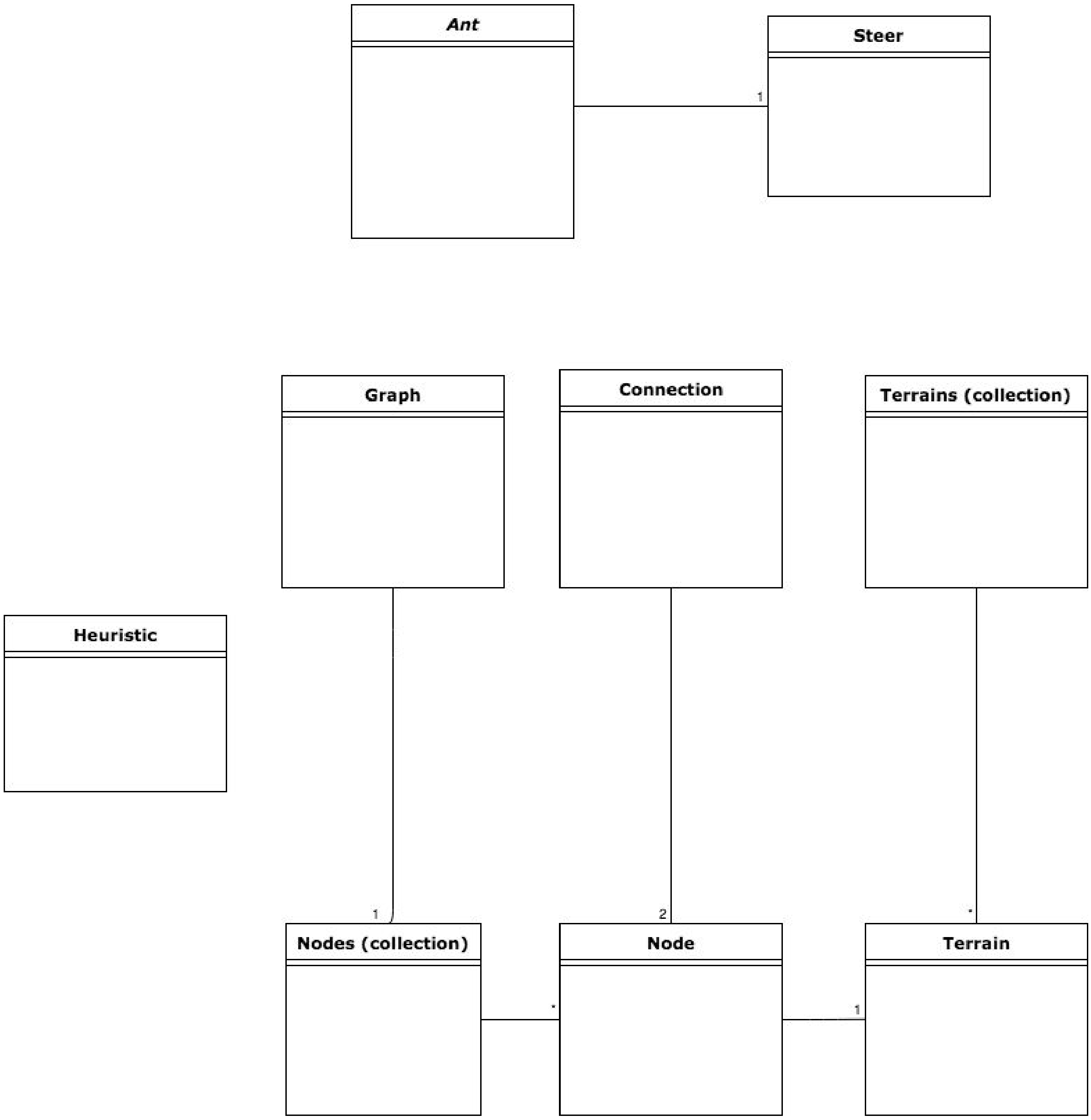
You can then click on a tile to apply the type to it. The “Obstacle” type has no cost because it indicates the tile is blocked and cannot be passed. There can only be one of the “Start” type tile (red) and one “Goal” type tile (flag).

Once you finish setting up the grid, press “Space” key to start pathfinding. The “ant” will start navigating if a path is found. Once the navigation is over, press “Space” key again to clean up the grid.

Code Structure

Class Diagram

Below is a simple class diagram showing the structure of the code:



Collection classes are indicated in the graph.

Class Descriptions

- Ant:
This is represented in the game as a black circle. It uses its pathfind method to navigate through the grid. The ant cannot move diagonally.

- Steer:
This class includes a very simple steering method to move the ant.

- Terrain:
This represents a tile on the grid.

- Terrains:
The collection class for Terrain class.

- Heuristic:
This class include a simple heuristic method to estimate the distance between a specific tile and the goal. (It calculates and returns a combination of the perpendicular distance of the two tiles plus the cost of the tile.)

- Graph:
This graph contains a list of all nodes. And it has a method to get all the outgoing connections from a node.

- Node:
This class includes a corresponding Terrain class and additional properties such as cost of path so far and estimated total cost of path through the tile.

- Nodes:
The collection class for Node class.

- Connection:
This represents the directed connection between two nodes.

Running game & View source code

Running the game

(Assuming you have Java installed on your system)

- For Windows systems, navigate to “astar.windows32/” directory and double click the file named “main” to start the game.

- For Linux systems, use terminal to navigate to “astar.linux32/” directory and run “./main” command.

Source code

To view the code, go to “source/” directory. Files with “.pde” extension are the original code, which can be view with any text editor or IDE. Processing uses Java syntax so if you are using Notepad++, you can select “Java” as language to highlight keywords.

The “.java” file was generated from the original code.

Development Environment

- Operating system: Windows 10 64-bit
- Language: Processing
- IDE: Processing 2.2.1