CSCI 331 – Lab #1

Noah Alvard
Section 03

20200914

Usage

python lab1.py <terrainImage> <elevationFile> <pathFile> <season> <outputFileName>

Map Data Class

The program uses a Node class to store the terrain data. Each node has its own cords, the previous node it came from, its elevation, terrain, and f/g/h costs. This allows for a much easier implementation of a* in my opinion and allowed for an easier tracing of the path when a path is found.

Transition Cost

The true cost function between two points is implemented using three pieces of information.

- Distance between the center point of the two nodes in meters
- Estimated difficulty of the terrain in estimated seconds per 10 meters of terrain
- Walking speed at the angle between the two points based on Tobler's Hiking Function

The first two are multiplied, as the cost scales positively with those, and that result is divided by the walking speed as the cost scales negatively with speed

Heuristic Cost

The heuristic cost function between two points is implemented using the same information. However, to ensure an under bound some rounding is done. The Distance factor is done using straight line between the two points. Due to the "real" map being a 2d grid, this leads to a straight-line distance that is much shorter than any actual possible path, no matter the terrain in-between. Second, the terrain difficulty used is the easiest from the available terrains, road. Lastly, the angle fed into Tobler's function to is -2.86 as this is the highest walking speed obtainable through the function.

Walking Speed Per Elevation

The walking speed due to elevation changes is calculated using Tobler's Function (created by Waldo Tobler).

$$W = 6e^{-3.5|\frac{dh}{dx}+0.05|}$$

This returns the walking speed in kilmeters per hour, so my implementation dividies by 3.6 so that all units are meters per second.

Terrain Difficulty

While I feel that the distance and walking speed calculations are mostly correct, the main issue I had designing the cost function was the terrain difficulty. Is walking in a trail actually that much harder than walking on a road? It turns out, yes. I did some practical experiments of my own with a stopwatch in the local area, timing how long it took to walk about 10 meters, and used these measurements as the difficulty factor for each terrain. However, the final values are not perfect representations of these measurements. I also took into account how much I would actually want to traverse one terrain over another. As a result, road is actually much lower than the time I recorded, and water is python's sys.maxsize because there's no way I'm swimming, even if it would lead to a faster traversal.

Seasonal Changes

Season changes are done after the map is loaded in. If it's fall, every trail node is iterated through and if it's adjacent to an easy move forest, it's changed to a leafy trail. Spring and winter are a bit more involved. If it's spring or fall the map is iterated through and every water node adjacent to land is added to a list. Then, a breadth first search is conducted using a queue seeded with that list, either searching inwards for ice, or outwards from the water for the mud in the spring.

User Display

Any changes to the terrain for seasonal changes is reflected on the output image. Ice is a light blue, mud is a brown, and leafy trails are an orange. Additionally, and paths are marked with a line, and the endpoints with a 3x3 green box with a red dot in the center. The image is saved as the outputFileName, and also shown through whatever default image viewer your computer has. Additionally, the total path length in meters is printed in the console.