COMP521 project proposal

Yuanhang Yang, Alfred Kenny
email: {firstname}.{lastname}@mail.mcgill.ca

I. Introduction

Two major problems involved in the production of computer games are the procedural generation of gameplay areas that a player or non-player character (NPC) will traverse. Combined, these topics can enrich the gameplay experience by creating a new world at upon each game initialization and also creating allies or opponents who appear to possess intelligence and an understanding of the world around them.

II. Topic Discussion

A. *Procedural town Generation*

The first topic we will explore is random town generation. As part of the field of content generation, this is a topic important in many games based in town environments, as town geometry is where the character(s) inhabit in. In the lectures we learned about techniques such as Perlin Noise, but many other procedural techniques have been well-studied, such as fractals, L-systems, tiling systems and cellular basis [1], for town generation, or terrain generation in general.

One of the goals of this COMP521 project is to explore these town generation techniques. We will learn about the ideas of different procedural town generation techniques, and pick the one(s) that will best serve the deliverable design. There have been a number of well-studied procedures available for this purpose, such as ”Roadmap → Allotment → Buildings → Geometry” procedure with L-systems [2]. In addition, we will explore how to combine different techniques to create a town with appropriate level of complexity for the path-finding topic of this project.

B. *Pathfinding*

The second topic we will explore is the task of determining how to move a character from one point in a map or shape to another. This topic is especially important in regard to maneuvering a character through a town, as towns are neither purely open spaces nor purely grids. Instead, towns exist somewhere between these two extrema, with some regions appearing more gridlike and others more open, but with both aspects existing everywhere to some degree.

In addition to the layout of a town, certain aspects of a town may change over time. While some intersections may have simple stop signs in which a pedestrian always has the right of way, other may have traffic lights prohibiting the movement in one direction at a given time. With this in mind, a good pathfinding algorithm in such an environment should take the dynamic nature of the polygon to be traversed.

III. Final deliverable

The final deliverable of this project is expected to be a game demo, rather than a full game. Users will be able to observe an AI character’s behaviors navigating around surrounding environment.

Upon starting, a 2D town map is randomly generated, with a certain level of complexity and different visual objects (e.g., road, park, households, town hall, etc.). There are always two objects whose positions in the map are fixed: the ”My House” and the ”town Hall”. We will attempt for the underlying town generation algorithm to generate reasonably realistic-looking town maps.

Optionally, there will a bus route covering major roads of this town, with buses’ locations clearly visible on the town map, possibly pre-determined. The bus route will be computed so as to cover up a maximum number of households, given certain constraints (e.g. number of bus routes, max length of each route, etc.).

The (AI) character will always start at ”My House” and her goal is to get to ”town Hall” as fast as possible. Upon planning the path, she will take into consideration of:

- whether or not she can traverse different terrain types (e.g. she cannot swim, or pass through buildings)
- the current availability of paths with respect to street lights
- (optionally) whether there are public transportation she can take advantage of along the way.

In order to achieve all of these goals, the dynamic Lifelong Planning A* [3] will be implemented on the procedurally generated town. This algorithm begins as A* but reacts to changes in the graph/polygon as time goes on.

IV. Project Division

The workload will be divided in the following manner: one member will work on the generation of the town and the other will work on the implementation of a patfinding algorithm. The town is not dependent on the pathfinder but the pathfinder is inherently dependent on the town. To minimize this, the pathfinder will be implemented to work on a grid with edges randomly removed and introduced in emulation of the finished town.

References

[1] G. Kelly and H. McCabe, “A survey of procedural techniques for city generation,” *ITB Journal*, vol. 14, pp. 87–130, 2006.

[2] Y. I. Parish and P. Müller, “Procedural modeling of cities,” in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques.* ACM, 2001, pp. 301–308.

[3] S. Koenig, M. Likhachev, and D. Furcy, “Lifelong planning A*,” *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93–146, 2004.