Abdulrahman Abdulrahman, Brian Moran, Fares Easa, Khalid Masuod

CS440

10/20/20

Assignment 1: Heuristic Search

Objective: For this project we were tasked with creating a 2D discretized terrain grid that will support an A\* path finding Algorithm. The A\* algorithm will be run with three different heuristics in mind: uniform cost search, normal A\* search algorithm, and weighted A\* search.

Submission:

Assign1.zip containing:

* Algorithm.java
* AlternateAlgorithm.java
* CS440 Readme.docx is the current Read Me file
* DrawGrid.java
* Exev.java is the execution file that creates the grid object and runs the algorithm
* GridGenerator.java has the Grid object file, has constructors to create grid, and can write the file to grid
* Node.java is the node object file that creates and holds the nodes to be placed in the grid.
* Pair.java

Program Design:

Grid Generator Description:

Grid class has a 2D array of Nodes, from NodeClass

* + An array of 2 integers that represents the start Nodes
  + An array of 2 integers representing the end nodes;
  + Cells that are hard to traverse;

5 Kinds of constructors:

* + One Constructor that generates the complete grid with start and end points
  + The file name is the file to save the grid, rand is to separate the constructor from others
  + One Constructor that takes in arrays with a start and end location;
  + Reads in a files grid and creates a new start and stop location
  + One Constructor that takes in a string that is a file name to import the data;

Functions for Manipulation:

* + Print to stream the values within the grid - Warning! Huge output for the stream.

traverse()

* + Prints to the stream the grids designated start and end point;

stats()

* + Generates start and stop and ensures the file

generate()

* + Generates start for the path process, in array format

genStart()

* + Generates Goals for the path process, in array format

genEnd()

* + Generates the Harder to traverse cells
    - hardCells()
  + Generates the blocked cells that you cannot traverse
    - blockedCells()
  + Generates the Normal cells within the grid
    - normalCells(){
  + Generates 4 highways within the map
    - highways()
  + Creates a new file with the first two lines being the start and end points and then 160 rows of data;
    - writeToFile()