Erik Marrero

Nely Rodriguez

Project proposal

Dr. Neli Zlatareva

Cs-253

Algorithm: Dijkstra’s Algorithm

Graph type: Weighted Graph

The problem we are solving is finding the best and quickest way out of a maze. We will be creating an algorithm that finds the optimal path out of a maze. The algorithm that best fits what we are trying to solve is Dijkstra's algorithm. With dijkstra's algorithm we will be able to calculate which path to take based on the paths between the nodes in the weighted graph. The program will first start off as a game in which the user gets from the starting point of the maze to the ending point. After the user has completed getting out of the maze we will introduce the best path discovered while using Dijkstra's algorithm. The program will compare the users' choices to the best solution.The main objectives besides completing the maze is implementing Dijkstra's algorithm in the correct way and using a weighted graph. When playing a game the user has to just get from point A to point B whether they do it quickly or take a different approach. The user has many different options on how they can do it; this all depends on the path they chose to take. Dijkstra's algorithm finds the optimal path by doing calculations and is a straightforward set of thinking. While the user is just trying to play a game, the user's way of thinking is about finishing the game. The algorithm is programmed to function a certain way but a user has outside elements that can influence their choices. When using a weighted graph we need to take into consideration it is a simple graph with weighted edges. When doing calculations for a weighted graph we need to use a set of vertices, a set of edges and a number that is set by length. Without a weighted graph Dijkstra’s algorithm would not work properly. As long as both objectives are good the problem can be solved.

For the program we will use 2D Graphics to create a maze. First, we identify the start and end point of the maze. Then, the coded algorithm will determine all paths to find the way out of the maze.After identifying all possible paths for the maze it will search every route through the maze using Dijkstra’s Algorithm. We then find the best path to find the end of the maze. The algorithm will determine the best path by applying a weight to each path. So it will check to see which path has the lowest weight which will be determined by how many blocks it will take to get the end of the maze. So if one path has 50 blocks to get to the end of the made it will have a weight of 50. If another path contains 45 blocks to get to the end of the maze it will have a weight of 45. The program will identify that the best route is the one with 45 blocks because it is the shortest distance to get to the end of the maze. We will implement user input. The user will be able to go through the maze and choose the path they feel is the best way out of the maze. After receiving the user input the program will continue by displaying the best path and proceed to compare both outcomes.