Program 4 Overview

Description

- Program 4 is breath first search(BFS) algorithm, it utilizes a queue and O(|M|+|N|) time complexity, where M and N are the dimensions of the maze. The user is given the option of 3 mazes of respective dimensions, 5x5, 20x20 and 40x40. Once a maze is selected, the program will output the initial maze to the console, followed by an updated version every x iterations, x is dependant on the size of maze selected.

Class Functions

- `Maze::Maze()`
  - This is the constructor for our maze class
    - Asks user to select maze
    - reads into correct file
    - initializes startPosit, endPosit, length, width, and both 2d arrays
- `void Maze::solveMaze()`
  - This function implements the BFS, and solves the maze. Using a queue we keep track of all the nodes we need to visit and due to the FIFO nature of a queue we reach the destination using the shortest available path
    - initialize queue and enqueue the start position
    - loop while the queue is not empty
    - dequeue a coordinate, check its neighbors for availability
    - if available, enqueue, update the "visitedness" of that cell and update the 2d char array to the right ascii character.
- `bool Maze::available()`
  - this function is called specifically to check if a coordinate is out of bounds and whether or not it has been visited
- `void Maze::printMaze`
  - beautifully prints the beloved maze to the console