SPL-1 Project Report

PACMAN

A Maze-chase video game

SE 305: Software Project Lab-1

Submitted by:

Imamul Hossain Rafi

Class Roll No: 1323

Exam Roll No: 115530

Session: 2020-2021

Supervised by:

Dr. Mohammed Shafiul Alam Khan

Professor and Director

IIT

University of Dhaka

Project Name: *Pacman, a mase chase video game*

Student Name: **Imamul Hossain Rafi**

Date Submitted: 21 May, 2023

Supervised By: **Dr. Mohammed Shafiul Alam Khan**

Professor & Director

Institute of Information Technology

University of Dhaka

Supervisor's

Approval : ____

Table of Contents

1.Introduction:

Pac-Man is a maze chase video game; the player controls the eponymous character through an enclosed maze.

The objective of the game is to eat all of the dots placed in the two different level mazes while avoiding the ghosts that pursue Pac-Man. When Pac-Man eats all of the dots from two different levels, the player wins.

If Pac-Man is caught by a ghost, the game ends for that current level.

1.1 Background Study:

i. Path-finding Algo:

Studied multiple pathfinding algorithm, i.e, BFS, DFS, Dijkstra, A*, Floyd-Warshall, and many other algorithms such as Mini-max algorithm, Alpha-beta pruning algorithm and chose the suitable one, ***A****.

ii. Console Interface:

Studied some process to colorize the C/C++ console since all characters and sign were white.

iii. File I/O:

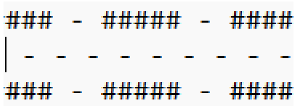
Studied Files and Streams to store data about a users' information and scores in a certain file.

1.2 Challenges:

I have faced a lot of challenges throughout the project.

Specially

i. ***Designing*** the mazes since the space between two row wise food is 1 but the space between two column wise food is 2. Hence sometimes



```
### - ##### - ####
| - - - - - - - - -
### - ##### - ####
```

faced these type of problems , while designing the mazes where a

food under a food is not exactly beneath.

ii. ***Applying and adjusting*** the A* path finding algorithm for Manhattan heuristic search across the mazes, since for the gap differences between rows and columns.

iii. ***Testing and bug fixing*****:** A lots of bugs arrived throughout the entire project. Some well recognized bugs and tests are mentioned below:

➢ When two ghost happens to come into contact to each other, they seemed to go to Scattered mode and lose their ability to functioning with the A* algorithm.

➢ When for a ghost to follow the minimum path to the Pacman, sometimes there were several paths with the same “minPath” values, but the actual path containing minimum value was one.

➢ After adjusting A* to all the 4 ghosts, the game seemed very much difficult to complete, almost impossible. Hence two upper ghosts were given a random number generator with 60% probability to close the gap between Pacman and ghosts.

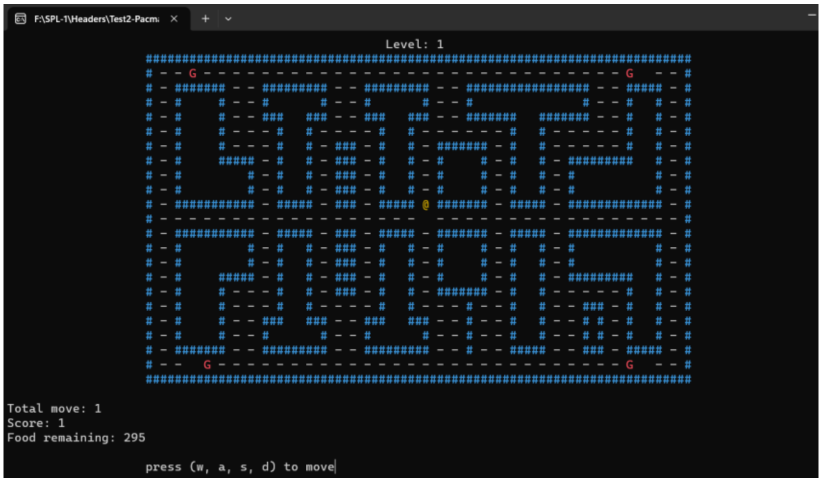
➢ After developing the code of a certain amount, the ghosts started to consume the foods on the path. Thus had to fix this with a foodtoken variable.

2. **Project Overview:**

In the first portion of project, I dealt with the designing the maze and movement of the player.

The first level maze is “IIT” theme based maze. It was first designed in a text file then inserted manually in the code with a 2D character array. Then there was another maze for level two of the game. After passing level 1 with consuming all the foods by the Pacman, the player moves to level 2. Then there was implemented another maze with a higher difficulty structure. If the player can pass all the two levels, he can win the game.

The player movement was dealt manually with “if, else if” code lines. The game take one input from player with “getInput” method to function one move for the Pacman and after one movement from the player, all the ghosts give one move following A* and random generator functions i.e, “GhostLogic”, “minPath” methods. If player move meets any wall on the maze, the Pacman won’t move but the ‘movecount’ will increase as well as the ghosts will give a valid move. Player can give moves until all the foods are consumed or until any ghost catches the Pacman.



```
                                          Level: 1
                    ###########################################################################
                    # - - G - - - - - - - - - - - - - - - - - - - - - - - - - - - - - G   - - #
                    # - ####### - - ######### - - ######### - - ################# - - ##### - #
                    # - #     # - - #       # - - #       # - - #               # - - #   # - #
                    # - #     # - - ###   ### - - ###   ### - - #######   ####### - - #   # - #
                    # - #     # - - - #   # - - - - #   # - - - - - - #   # - - - - - #   # - #
                    # - #     # - - - #   # - ### - #   # - ####### - #   # - - - - - #   # - #
                    # - #     ##### - #   # - ### - #   # - #     # - #   # - #########   # - #
                    # - #         # - #   # - ### - #   # - #     # - #   # - #           # - #
                    # - #         # - #   # - ### - #   # - #     # - #   # - #           # - #
                    # - ########### - ##### - ### - ##### @ ####### - ##### - ############# - #
                    # - - - - - - - - - - - - - - - - - -   - - - - - - - - - - - - - - - - - #
                    # - ########### - ##### - ### - ##### - ####### - ##### - ############# - #
                    # - #         # - #   # - ### - #   # - #     # - #   # - #           # - #
                    # - #         # - #   # - ### - #   # - #     # - #   # - #           # - #
                    # - #     ##### - #   # - ### - #   # - #     # - #   # - #########   # - #
                    # - #     # - - - #   # - ### - #   # - ####### - #   # - - - - - #   # - #
                    # - #     # - - - #   # - - - - #   # - - - # - - #   # - - ### - #   # - #
                    # - #     # - - ###   ### - - ###   ### - - # - - #   # - - # # - #   # - #
                    # - #     # - - #       # - - #       # - - # - - #   # - - # # - #   # - #
                    # - ####### - - ######### - - ######### - - # - - ##### - - ### - ##### - #
                    # - -   G - - - - - - - - - - - - - - - - - - - - - - - - - - - - G   - - #
                    ###########################################################################

Total move: 1
Score: 1
Food remaining: 295

                    press (w, a, s, d) to move
```

Then the second portion of the project dealt with the movement of the ghosts. There are 4 ghosts in both the mazes. All the ghosts give one move for any input from the player for the Pacman. The upper two ghost in the first row use “random” method that generates a number from 0 to 9 and for the first four numbers; 0-3, it makes a random directional move for the ghost. But for the last 6 number; 4-9, it gives a moves decisively. The decision is made based upon the position of the Pacman and the ghosts that eventually close the gap between the Pacman and the ghosts. Then the bottom most two ghosts make their moves according to A* path pathing algorithm. These ghosts call for a function “GhostLogic” that returns a value for their environmental 4 blocks, i.e, UP, DOWN, RIGHT, LEFT. Based on the returned values, these two ghosts make their move. Inside the “GhostLogic” function there is another function called, named “minPath”. Actually this “minPath” function deals with the A* pathfinding algorithm. What “minPath” does:

```
1. Initialize the open list

2. Initialize the closed list and put the starting node on the open list.

3. while the open list is not empty
    a) find the node with the least “fCost” on
       the open list, call it "q"

    b) pop q off the open list

    c) generate q's 4 successors and set their
       parents to q

    d) for each successor
        i) if successor is the goal, stop search

        ii) else, compute both “gCost” and “hCost” for successor
          successor.g = q.g + distance between
                              successor and q
          successor.h = distance from goal to
          successor
          successor.f = successor.g + successor.h

        iii) if a node with the same position as
            successor is in the OPEN list which has a
            lower f than successor, skip this successor

        iv) if a node with the same position as
            successor is in the CLOSED list which has
            a lower f than successor, skip this successor
            otherwise, add the node to the open list.

    e) push q on the closed list
```

This “minPath” returns the minimum value for the 4 successors of the ghost to “GhostLogic” which gives a direction for the ghost to move.

3. **User Manual:**

The user first has to open an IDE that can run C++ code on console. Then after installing all the files including header files in the same path, he should run the code on the console of the operating system.

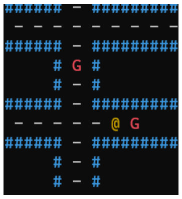
A screen will show with the message “Press any key”. After pressing any key, the first maze will arrive with all the Ghosts, foods and Pacman in initial position. A player can give decision for any of 4 blocks around the Pacman to move.

Up with ‘w’,

Down with ‘s’,

Left with ‘a’,

Right with ‘d’.



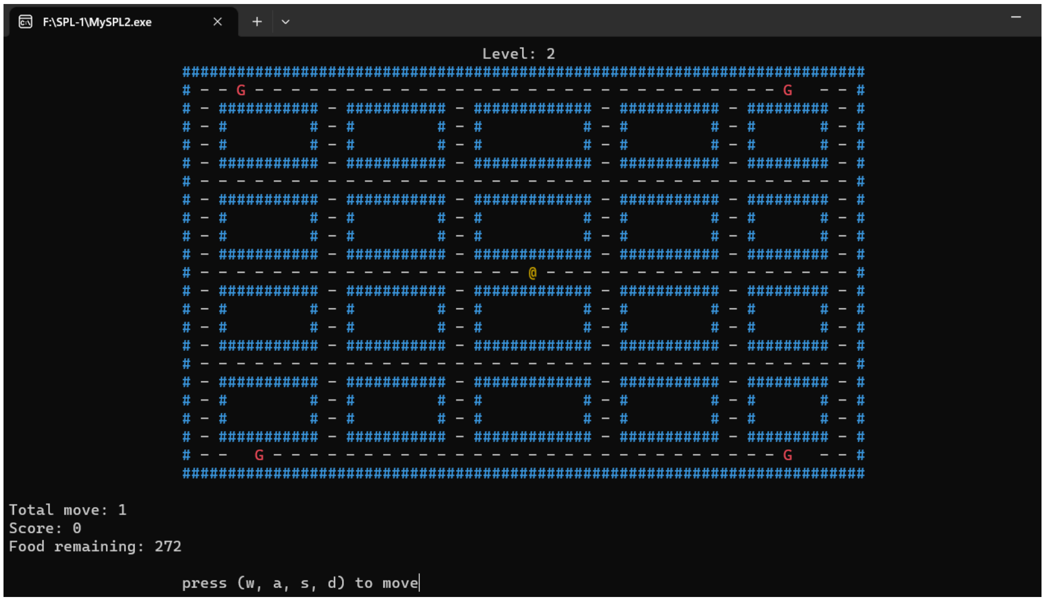
```
###### - #########
 - - - - - - - - -
###### - #########
     # G #
     # - #
###### - #########
 - - - - - @ G
###### - #########
     # - #
     # - #
```

The Pacman is indicated with ‘@’ and the ghosts are indicated with ‘G’, and the foods are indicated with ‘-’.

It is a valid move for the Pacman, if the move meets a space block or a block with food or a block with ghost. But if the move faces a wall, ‘#’, the move will be invalid, thus the Pacman will not make a move. But the ghosts will make their moves for this move. So, the player has to check his/her every move so that the Pacman doesn’t crash with the wall.

The player should check all the ghosts at the same time as they all intend to enclose their gap from Pacman. Thus a situation may come where the Pacman can give a valid move without touching the ghosts, but it’s path will be eventually blocked with 2, 3 or 4 ghosts.

After finishing level 1, the player will move into level 2. In this level the difficulty will be increased as the movements for ghosts will be more freedomised. The chance of getting Pacman's path blocked is high. So, the player should watchout.



```
                                                    Level: 2
                   ###########################################################################
                   # - - G - - - - - - - - - - - - - - - - - - - - - - - - - - - - - G   - - #
                   # - ########### - ########### - ############# - ########### - ######### - #
                   # - #         # - #         # - #           # - #         # - #       # - #
                   # - #         # - #         # - #           # - #         # - #       # - #
                   # - ########### - ########### - ############# - ########### - ######### - #
                   # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - #
                   # - ########### - ########### - ############# - ########### - ######### - #
                   # - #         # - #         # - #           # - #         # - #       # - #
                   # - #         # - #         # - #           # - #         # - #       # - #
                   # - ########### - ########### - ############# - ########### - ######### - #
                   # - - - - - - - - - - - - - - - - - - @ - - - - - - - - - - - - - - - - - #
                   # - ########### - ########### - ############# - ########### - ######### - #
                   # - #         # - #         # - #           # - #         # - #       # - #
                   # - #         # - #         # - #           # - #         # - #       # - #
                   # - ########### - ########### - ############# - ########### - ######### - #
                   # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - #
                   # - ########### - ########### - ############# - ########### - ######### - #
                   # - #         # - #         # - #           # - #         # - #       # - #
                   # - #         # - #         # - #           # - #         # - #       # - #
                   # - ########### - ########### - ############# - ########### - ######### - #
                   # - -   G - - - - - - - - - - - - - - - - - - - - - - - - - - - - G   - - #
                   ###########################################################################

Total move: 1
Score: 0
Food remaining: 272

                   press (w, a, s, d) to move
```

After finishing level 2, the game will be over and the player will be shown a Congratulatory message on the screen.

4. Implementation and Tools:

i. Language: C++

ii. Lines of codes: 1200+

iii. Number of files: 7

iv. Number of methods: 15

iii. Number of Github commits: 40+

iv. Project duration: January 2023 to May 25th, 2023.

5. Conclusion:

In this project I am trying to build a simple console based arcade game. There are some flaws, such as after every move the screen refreshes. It is not eye soothing at all. As I didn’t use any STL or any graphic tools, I couldn’t overcome these flaws. But this project surely gave me inspiration to work more as look forward to be a game developer. I think my critical thinking has developed throughout the entire project.

My Supervisor and my seniors from IIT helped a lot during the whole session. This developed my communication skill. During the session, I have explore many websites, that helped me developing my information gathering skill.

Reference:

Manhattan Huristic Search; “GeeksforGeeks”

https://www.geeksforgeeks.org/a-search-algorithm/

A* pathfinding algorithm; “StackOverFlow”

https://stackoverflow.com/questions/23189211/a-algorithm-with-manhattan-distance-heuristic

Youtube

https://www.youtube.com/results?search_query=a*+algorithm+pacman

My SPL-1 GitHub Link:

https://github.com/Hack-it-Rafi/SPL_1