KHULNA UNIVERSITY OF ENGINEERING & TECHNOLOGY

Department of Computer Science and Engineering

CSE 4110: Artificial Intelligence Laboratory

Game Title

Gold Rush: AI based game to collect gold by avoiding enemy

Presented by:

Sajidul Islam (1707010)

Tahsin Mahmud (1707019)

MD. Tamim Hossain (1707020)

Date of Submission: JULY 6, 2022

Contents

List of Figures

1 Introduction

Artificial intelligence (AI) is used in video games to create human-like intelligence in non-player characters (NPCs) by generating responsive, adaptable, or intelligent actions. Since the 1950s, when they first appeared, artificial intelligence has played a significant role in video games.Since Garry Kasparov was defeated by IBM's computer program Deep Blue in a chess match in 1997, artificial intelligence (AI) has assumed an increasingly significant and beneficial role in the gaming business. AI is utilized in a variety of ways to improve game elements, behaviors, and settings.

Figure 1: AI robot playing chess against human

Our game which was developed while attending the CSE4110 Laboratory similarly has the capability to read to human move and make the effort to win against human.It's a fun game to play while a human brain has to move wisely to beat the AI(or computer). Anyone who needs a little brainstorming will enjoy our game very much. This game has been developed using the PYTHON language while the User Interface was done using Pygame .The algorithms which we focused on while developing this game is Breadth First Search.

2 Game Overview

2.1 Modes of the Game

The goal of the game is to have you collect the gold scattered around a maze. The game will start by pressing the SPACE button. At he initial point when you start the game one character will start running and try to collect the gold around the maze. You can control the move of the character by using up, down, left and right button on the keyboard. At the same time four enemy also start running and try to catch you. the player have three lives in single play. At the end of the game, you can press SPACE bar for restart or ESC for quit the game.

2.2 Board Organization

There is a standard Board on which the game will be played. The maze size is 28x30. The smallest unit of the boxes will be called grid.

Basically a player can move only the valid grid and can collect the coin. The wall and coin arrangement is made by using wall.txt file where we use '1' for wall and 'C' for coins.

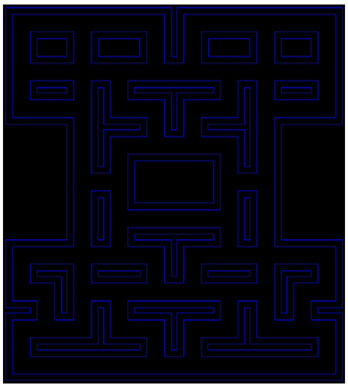
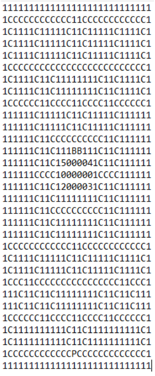


Figure 2: Board Organization of Our Game



```
1111111111111111111111111111
1CCCCCCCCCCCC11CCCCCCCCCCCC1
1C1111C11111C11C11111C1111C1
1C1111C11111C11C11111C1111C1
1C1111C11111C11C11111C1111C1
1CCCCCCCCCCCCCCCCCCCCCCCCCC1
1C1111C11C11111111C11C1111C1
1C1111C11C11111111C11C1111C1
1CCCCCC11CCCC11CCCC11CCCCCC1
111111C11111C11C11111C111111
111111C11111C11C11111C111111
111111C11CCCCCCCCCC11C111111
111111C11C111BB111C11C111111
111111C11C15000041C11C111111
111111CCCC10000001CCCC111111
111111C11C12000031C11C111111
111111C11C11111111C11C111111
111111C11CCCCCCCCCC11C111111
111111C11C11111111C11C111111
111111C11C11111111C11C111111
1CCCCCCCCCCCC11CCCCCCCCCCCC1
1C1111C11111C11C11111C1111C1
1C1111C11111C11C11111C1111C1
1CCC11CCCCCCCCCCCCCCCC11CCC1
111C11C11C11111111C11C11C111
111C11C11C11111111C11C11C111
1CCCCCC11CCCC11CCCC11CCCCCC1
1C1111111111C11C1111111111C1
1C1111111111C11C1111111111C1
1CCCCCCCCCCCCPCCCCCCCCCCCCC1
1111111111111111111111111111
```

Figure 3: Wall and Coin Orientation

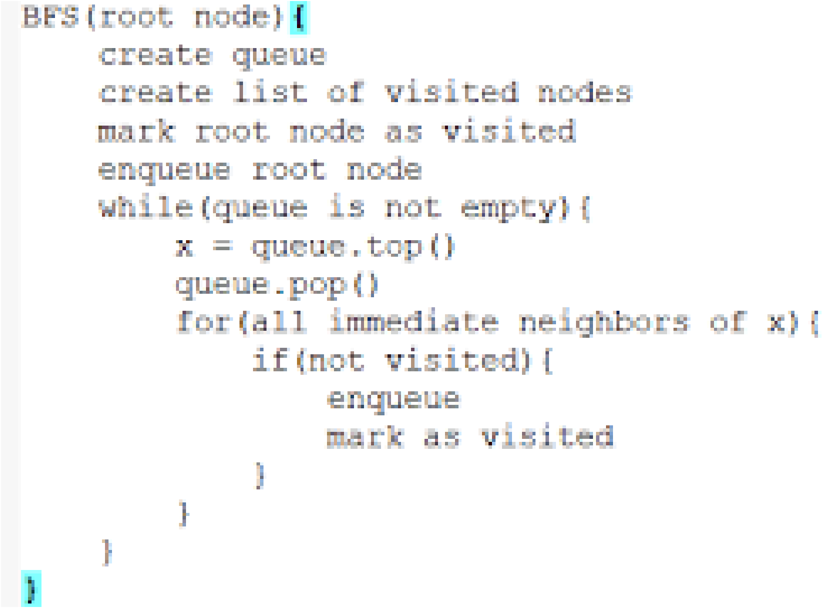
2.3 Deciding the Winner

If there is no gold left in the maze, the player is the winner. But if the enemy can catch you three times in a singe game, the player lose the game.

3 Code Decryption

3.1 Fundamental Algorithm

The pseudo-code used for is mentioned in Figure-4. The AI moves are based on this algorithm. Also one enemy maintain the random algorithm called by random library function used in python. Breadth-first search (BFS) is an algorithm that is used to graph data or searching tree or traversing structures. The full form of BFS is the Breadth-first search.The algorithm efficiently visits and marks all the key nodes in a graph in an accurate breadth wise fashion. This algorithm selects a single node (initial or source point) in a graph and then visits all the nodes adjacent to the selected node. Remember, BFS accesses these nodes one by one. Once the algorithm visits and marks the starting node, then it moves towards the nearest unvisited nodes and analyses them. Once visited, all nodes are marked. These iterations continue until all the nodes of the graph have been successfully visited and marked.



```
BFS(root node){
    create queue
    create list of visited nodes
    mark root node as visited
    enqueue root node
    while(queue is not empty){
        x = queue.top()
        queue.pop()
        for(all immediate neighbors of x){
            if(not visited){
                enqueue
                mark as visited
            }
        }
    }
}
```

Figure 4: Breadth First Search Pseudocode

3.2 Description of the important functions used in our code

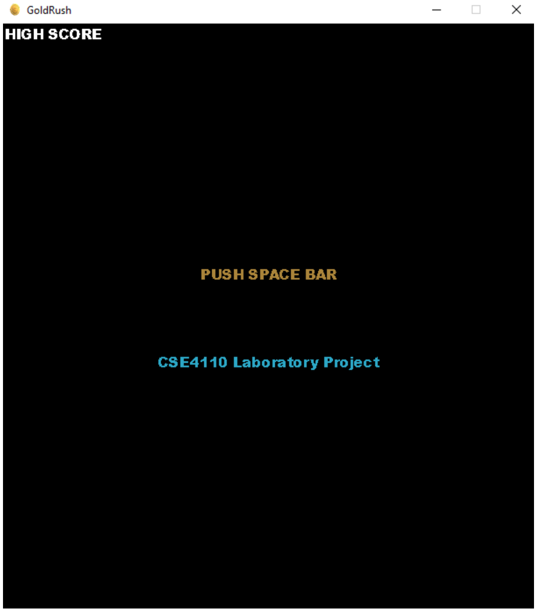
load : creates the game space taking the arguments(self)
set_speed : Sets the speed of the enemy
draw_coins : Draw the coin at the desired position on the grid.
set_personality : Set the personality parameter of the enemy i.e speedy, slow, random, scared.
get_random_direction : Update the speed of the enemy by using random library of python.
update : Update the position of the player in the grid.
find_next_cell_in_path : Find the next cell of the path by using breadth first search for the winning of the enemy
BFS : Implement the bfs algorithm by using queue for measuring the shortest path from the target and start moving that direction.
remove_life : If an enemy can catch the player in the grid the lives value will decrease by 1.
game_over_events : Determine the end of the game. If a player lose all 3 lives or can collect all the coins from the grid the game is ended.

4 Result and Analysis

4.1 How to open the Game

One needs to go to the folder and needs to go to the command prompt and write the following command : python main.py

The AI uses its best to dominate the game but as the human have the control to control the player moves, it gives AI a lot of challenge and AI needs to make the best use of its algorithm. The heuristic function helps to find the efficient path and if a human isn't careful or wise, he will be beaten by the AI easily. Overall, one will definitely have fun playing this game.



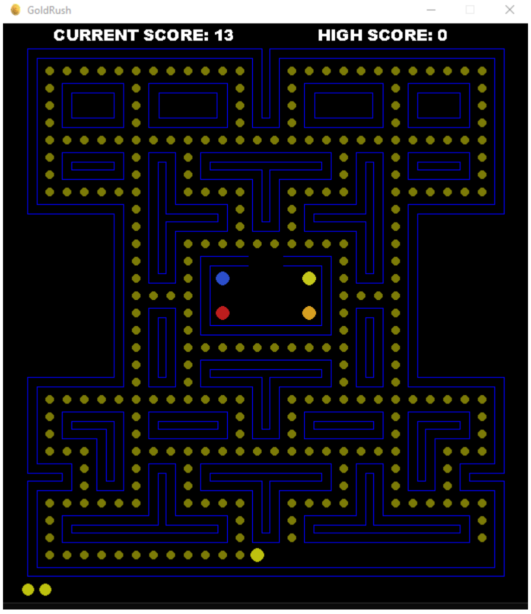
GoldRush

HIGH SCORE

PUSH SPACE BAR

CSE4110 Laboratory Project

Figure 5: Opening Window of the Game



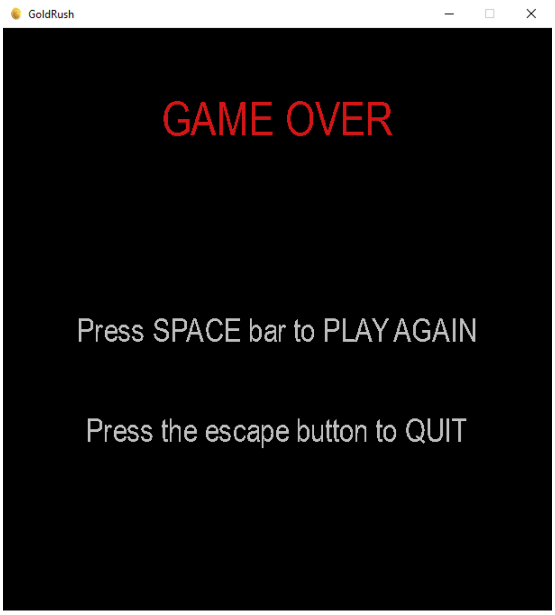
GoldRush

CURRENT SCORE: 13

HIGH SCORE: 0

Figure 6: Image of the Gameplay

The AI uses its best to dominate the game but as the human have the control to control the player moves, it gives AI a lot of challenge and AI needs to make the best use of its algorithm. The heuristic function helps to find the efficient path and if a human isn't careful or wise, he will be beaten by the AI easily. Overall, one will definitely have fun playing this game.



GAME OVER

Press SPACE bar to PLAY AGAIN

Press the escape button to QUIT

Figure 7: Gameover Window

5 Conclusions

AI has been improved day by day and our project is just the proof of that. Overall, we had a lot of fun and learnt a lot of new things while developing this game. This project made us walk-through all the classes we took in this amazing lab under the supervision of our respected teachers. We are hopeful that our game can be further taken to a new upgrade and many more new features can be added to it.

References

[1] https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/

[2] https://www.mygreatlearning.com/blog/best-first-search-bfs/

[3] https://www.mygreatlearning.com/blog/a-search-algorithm-in-artificial-intelligence/