FINAL DATA STRUCTURES PROJECT

Racing car game

By: Maham Imran & Fatima Wajahat

Introduction:

The racing car game we implemented for our final term project for data structures is a 2D console-based game. The game is a menu-based game, which allows users to choose between different modes of the game. Once a mode is selected, the game automatically generates a random maze(each time) that the user can navigate. Different objects are placed throughout the maze, such as blockades, coins, and powerups, each giving the game a push of excitement. For example, if the user is to hit a blockade, they get deducted a few points, furthermore, the user can collect coins which automatically add up. The game also uses CSV file handling to store the scores of users, which can be output to the console through the main menu.

Design:

The code uses several data structures such as linked lists,graphs, queues,tree to implement the game.

We also used many structures and classes to make the major functionality, which are explained as follows:

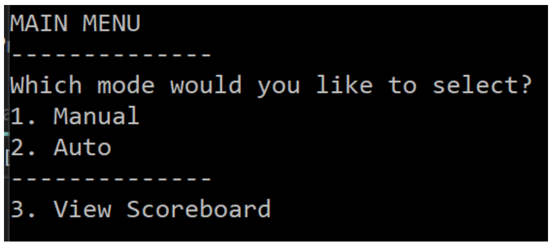
1. Struct obstacles
2. Struct Coins
3. Struct powerups
   (These three structures were created to make the objects of the game, they contained the symbol that they were to represent and their placement on the grid. Further in the code these structures were used to create queues to store all their instances in the code organised.)
4. Struct Node (this structure was the node for the tree, which functionalities are coded following the node as global functions)
5. Class graph (this is the class that creates the maze and grid using a graph data structure which was coded manually, not in built. This includes the finding shortest path and dijkstrasAlgorithm)
6. Class Car (This is the class that contains the entire functionality of the car, from the movement to the barriers and the finish line)

The major functionality of the code is implemented using functions inside classes, or globally both in header and source files. They are explained as follows(except getter setters, inserts, constructors and main):

1. bool isPlaceTakenByCoins: function to check if there is a coin present in a specific location
2. bool isPlaceTakenByObstacles: function to check if there is a blockade present in a specific location
3. bool isPlaceTakenByPowerUps: function to check if there is a powerup present in a specific location
4. vector<int> dijkstrasAlgorithm: the dijkstras algorithm to find the shortest path between two points in the graph
5. vector<int> findPath: To find the shortest path using the above algorithm
6. void moveUp, moveDown, moveRight, moveLeft, moveCar: functions to move the car according to the provided input
7. void checkForCollision: uses functions (1,2,3) to see if the car has collided in to any of the objects
8. void checkForLastVertex: checks if the user has reached the finish line or not
9. Void generateItems: This function generates the items on the already generated grid, randomly each time.
10. void display: displays the grid and objects and car
11. void generateMap: this functions generates a random grid using the number of rows and columns provided to it
12. void manualMode: of the two modes this function calls other functions to generate the manual mode of the game and also outputs information for the user.
13. Void autoMode: the second mode is generated through this function
14. void writeToFile: This functions writes the score board data to the CSV file
15. void viewScoreBoard: This function outputs the score board of the CSV file to the console
16. void displayAllQueues: This function stores the coins and powerups the user has collected in a queue, to calculate the final score once the game is over.

How it runs:

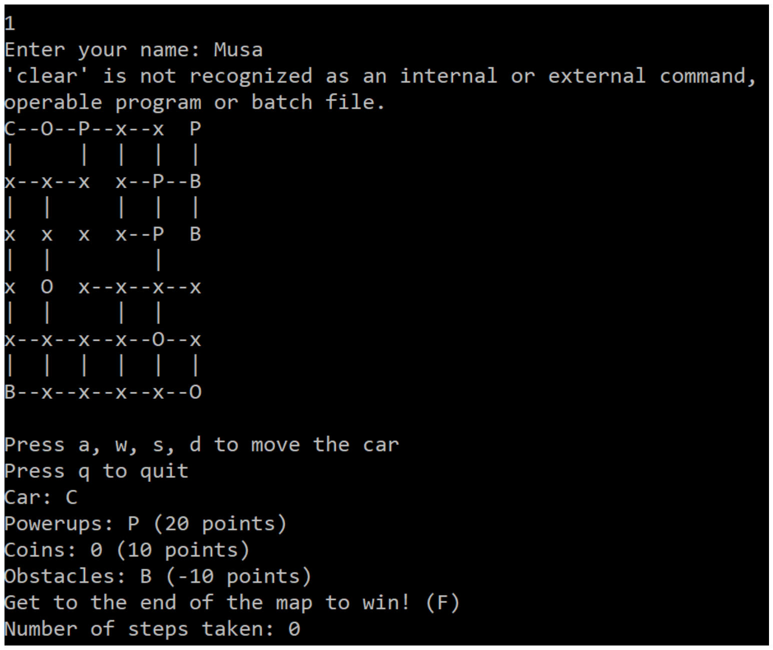
The code starts with a menu which takes an input from the user from three choices



```
MAIN MENU
--------------
Which mode would you like to select?
1. Manual
2. Auto
--------------
3. View Scoreboard
```

Now if you:

1. Select manual mode: The console asks the user for their name, in order to create a new or use an old entry in the scoreboard. Then proceeds to load a maze for the user along with some information and guidelines.

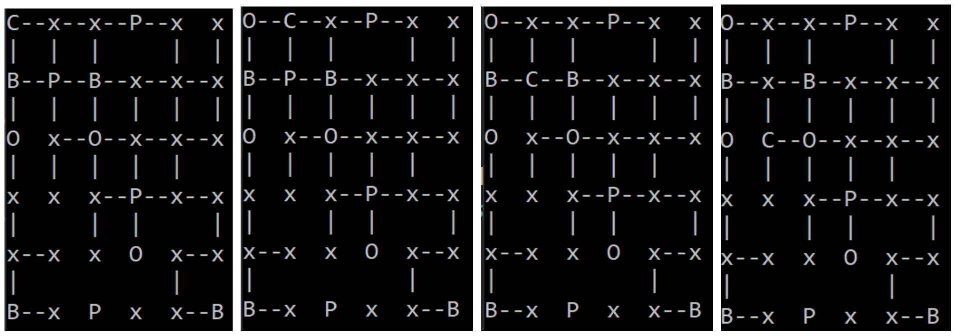


```
1
Enter your name: Musa
'clear' is not recognized as an internal or external command,
operable program or batch file.
C--O--P--x--x  P
|     |  |  |  |
x--x--x  x--P--B
|  |     |  |  |
x  x  x  x--P  B
|  |        |
x  O  x--x--x--x
|  |     |  |
x--x--x--x--O--x
|  |  |  |  |  |
B--x--x--x--x--O

Press a, w, s, d to move the car
Press q to quit
Car: C
Powerups: P (20 points)
Coins: 0 (10 points)
Obstacles: B (-10 points)
Get to the end of the map to win! (F)
Number of steps taken: 0
```

The user can move the car by using the a,s,d,w keys as you do in other games on line. If you hit a maze border the car does not move, if you hit a blockade “B” you get 10 points subtracted, if you hit a powerup “P” you get 20 points added, and if you hit a coin “C” you get coins added as well as 10 points added. The number of steps you take are also calculated and output.
Once you reach the “F” finish line the game terminates and you are congratulated using a prompt.

2. Select auto mode: The auto mode automatically creates a maze and moves the car through the maze on the shortest possible path, until it reaches the finish line.



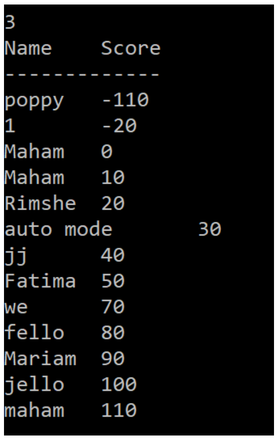
```
C--x--x--P--x  x
|  |  |     |  |
B--P--B--x--x--x
|  |  |  |  |  |
O  x--O--x--x--x
|  |  |  |  |
x  x  x--P--x--x
|     |  |     |
x--x  x  O  x--x
|           |
B--x  P  x  x--B
```

```
O--C--x--P--x  x
|  |  |     |  |
B--P--B--x--x--x
|  |  |  |  |  |
O  x--O--x--x--x
|  |  |  |  |
x  x  x--P--x--x
|     |  |     |
x--x  x  O  x--x
|           |
B--x  P  x  x--B
```

```
O--x--x--P--x  x
|  |  |     |  |
B--C--B--x--x--x
|  |  |  |  |  |
O  x--O--x--x--x
|  |  |  |  |
x  x  x--P--x--x
|     |  |     |
x--x  x  O  x--x
|           |
B--x  P  x  x--B
```

```
O--x--x--P--x  x
|  |  |     |  |
B--x--B--x--x--x
|  |  |  |  |  |
O  C--O--x--x--x
|  |  |  |  |
x  x  x--P--x--x
|     |  |     |
x--x  x  O  x--x
|           |
B--x  P  x  x--B
```

3. Select view scoreboard: The score board is output, via the scoreBoard.csv file that is given along with the header and source files.



```
3
Name    Score
-------------
poppy   -110
1       -20
Maham   0
Maham   10
Rimshe  20
auto mode       30
jj      40
Fatima  50
we      70
fello   80
Mariam  90
jello   100
maham   110
```