CS-2001 Data Structures (FALL-2023)

Semester Project Statement: Design and Implement a Race Car Game

Pair Members

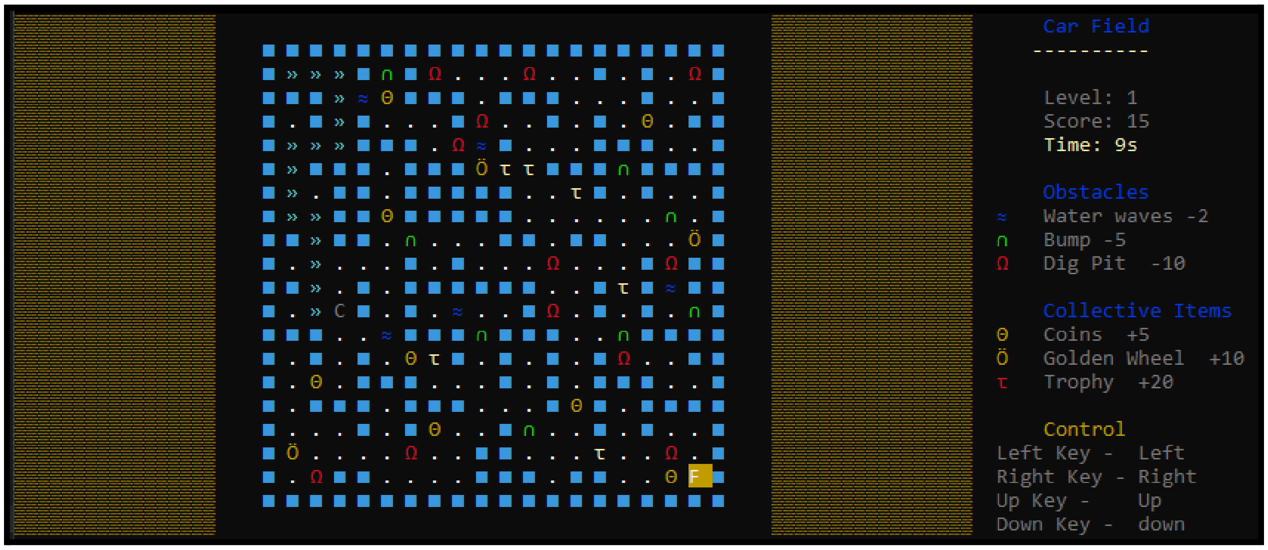
Amna Hassan 22i-8759

M Shuja Uddin 22i-2553

Submitted to:

Mam Maheen Arshad

CAR FIELD



```
■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■      Car Field
■ » » » ■ ∩ ■ Ω . . . Ω . . ■ . ■ . Ω ■      ----------
■ ■ ■ » ≈ Θ ■ ■ ■ . ■ ■ ■ . . . ■ . . ■
■ . ■ » ■ . . . ■ Ω . . ■ . ■ . Θ . ■ ■       Level: 1
■ » » » ■ ■ ■ . Ω ≈ ■ . . . ■ ■ ■ . . ■       Score: 15
■ » ■ ■ ■ . ■ ■ ■ Ö τ τ ■ ■ ■ ∩ ■ ■ ■ ■       Time: 9s
■ » . ■ ■ . ■ ■ ■ ■ ■ . . τ ■ . ■ . . ■
■ » » ■ ■ Θ ■ ■ ■ ■ ■ . . . . . . ∩ . ■      Obstacles
■ ■ » ■ ■ . ∩ . . . ■ ■ . ■ ■ . . . Ö ■   ≈  Water waves -2
■ . » . . . ■ . ■ . . . Ω . . . ■ Ω ■ ■   ∩  Bump -5
■ ■ » . ■ . ■ ■ ■ ■ ■ ■ . . ■ τ ■ ≈ ■ ■   Ω  Dig Pit  -10
■ . » C ■ . ■ . ≈ . . ■ Ω . ■ . ■ . ∩ ■
■ ■ ■ . . ≈ ■ ■ ■ ∩ ■ ■ ■ . . ∩ ■ ■ ■ ■      Collective Items
■ . ■ . ■ . Θ τ ■ . ■ . ■ . ■ Ω . . ■ ■   Θ  Coins   +5
■ . Θ . ■ ■ ■ . . . ■ . ■ . ■ ■ ■ . . ■   Ö  Golden Wheel   +10
■ . ■ ■ ■ . ■ ■ ■ . . . ■ Θ ■ . ■ ■ ■ ■   τ  Trophy  +20
■ . . . ■ . ■ Θ . . ■ ∩ . . ■ . ■ . . ■
■ Ö . . . . Ω . . ■ ■ . . . τ . . Ω . ■      Control
■ . Ω ■ ■ . . . . ■ ■ ■ . ■ ■ . . Θ F ■   Left Key -  Left
■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■   Right Key - Right
                                          Up Key -    Up
                                          Down Key -  down
```

CAR FIeld : Race Car Game Documentation

I. Introduction

Car Field, the Race Car Game, a 2D designed console-based adventure game that challenges players to navigate through a maze filled with obstacles and collective items like coins and trophies. It is developed in C++, this game provides an immersive experience, combining strategic thinking, fast-paced navigation, and the excitement of a race.

Purpose and Objectives

The purpose and objectives of the games includes giving users an entertaining experience. While playing the game it will engage the user intellectually and let them thick critically to go through the maze by dodging obstacles and collecting the most items.
The goal of the game is to provide engaging and interactive
game that incorporates various data structures, such as graphs, linked lists, or
queues, to manage game elements efficiently.

II. Game Overview

Description

The Race Car Game offers an engaging 2D console-based environment where players control a race car through a maze. The maze is dynamically generated, and players encounter obstacles and power-ups as they navigate toward the exit. As the player moves forward the maze randomly generates obstacles and collective items in the maze.

Text-Based User Interface (TUI)

The text-based user interface (TUI) utilizes ASCII art for maze representation, providing an immersive experience.A text-based user interface (TUI) relies on characters and text to convey information to the user instead of graphical elements. In a console-
based game, a TUI might involve using ASCII characters and text to represent
various game elements.

Game Modes

1. User-Operated Mode: Players manually control the race car using arrow keys, navigating through the maze and overcoming challenges an find the path.
2. Automated Mode: Players starts the game, and the game algorithm finds the shortest path, showcasing automated navigation and the player just have to follow the shortest path.

Gameplay Features and Dynamics

- Graph-based maze representation for dynamic and diverse gameplay.
- Obstacles deduct points, while collective items enhance scores and collective items are store in a list.
- Alternate routes based on graph dynamics provide strategic choices for players.

III. Project Objectives

Overview

Develop a 2D console-based race car game using C++ without utilizing any external graphics library. Implement a text-based user interface within the console (A text-based user interface (TUI) relies on characters and text to convey information to the user instead of graphical elements. In a console- ased game, a TUI might involve using ASCII characters and text to represent various game elements). The game must also feature player-controlled race cars, obstacles, tracks, and a scoring system.

Importance of Data Structures

By Implementing data structures, the Race Car Game should show a significant and efficient map, obstacle management, and item collection.

IV. Data Structures Integration

Data Structures Used In Project

a. Graphs for Map and Navigation:

- Map Representation: Graph to represent the game map. Nodes in the graph represent locations or waypoints in the game world, while edges represent paths between them. This graph can be used for player navigation. - Navigation and Pathfinding: Implement algorithms like Dijkstra's algorithm
or any other shortest path finding algorithms on the graph to find optimal paths for player-controlled race cars. The graph can help ensure that characters navigate the race track efficiently and avoiding obstacles

b. Queues for Obstacle:

- Obstacle : Queue data structure to manage the generation of obstacles. New obstacles can be added to the queue as they are generated, and the game loop can process the queue to introduce these elements at appropriate times.
- Example:
Game features randomly appearing obstacles on the race track, enqueue obstacle objects with information about their type and location. During each game update, dequeue these objects and add to the game world at the specified positions.

c. Linked Lists for Collected Coins and Game Trophies:

- Collected Items: Linked list to keep track of items collected by players during the game,i-e coins ,trophies and golden wheel.. Each node in the linked list represents an item collected, with relevant information (e.g., item type, score value).
- Example:
As players collect coins , add nodes to the linked list. The linked list allows easy tracking of collected items and can be used to update the player's score or display their achievements.

V. Game Logic

Player Controls

In playing manual mode the player has to use the arrow keys to navigate in the maze and find its path to finish the maze.
In automated mode, the player has to use the arrow keys to navigate in the maze but the system will show the shortest path and the player has just to follow that shortest path.

Collision Detection Mechanisms

For collision detection when the x,y coordinates of the car get equal to obstacles or collective items a collision is detected. Collision detection makes sure that the race car reacts correctly to obstacles and collective items. Score is deducted for collisions with obstacles, and scores are added for each collected item according to their values.

Scoring System Design

The scoring system is based on the number of steps, the numbers of collected item and obstacle clashed.There are three types of obstacle with values -2,-5 and -10. Each time any obstacle is collided the respective value is deducted from the score. Similarly there are three types of collective items with values 5,10 and 20. And added to the score each time any item is collected.

Win/Lose Conditions

Players win by successfully navigating the maze and reaching the exit. Losing conditions include colliding with too many obstacles or the score value drops less than 0.
Win/Lose conditions

Illusion of Movement Through Map Redrawing

Map redrawing creates the illusion of movement. The console refreshes with each player action, updating the display and providing a dynamic gaming experience.

VI. User Interface

Design Principles for TUI

The TUI is designed by keeping the clarity and aesthetics in focus. ASCII art is used in the maze, and a menu system guides players through different game modes and options.

Menus and Scoreboard Presentation

Menus are presented in an easily readable format, providing options for game modes, leaderboard,clearing leaderboard,instructions and quitting. Scoreboards displays previous players collectives and scores

VII. Graph-Based Map Representation

Detailed Map Design Using a Graph

The maze graph is strictly designed with nodes representing different features—path, collective item, and obstacles. Edges define connections between nodes and no edges defines walls.

Node Representation for Locations, Collective Items, and Obstacles

Each node type is represented with specific characters in the ASCII art. Collective items and obstacles are strategically placed to create challenges and rewards.

Graph-Based Game Dynamics

The graph structure introduces dynamic gameplay. Obstacles are special nodes that influence car movement, creating alternate routes and strategic choices for players.

VIII. Game Mechanics

Description of Intuitive Keyboard Controls

Arrow keys control,globally used for navigating, the race car's movement, allowing players to navigate through the maze. Key controls are standard for movement , providing a smooth gaming experience.

Different Game Modes and Their Functionalities

User-operated and automated modes offer distinct experiences. Players can enjoy the thrill of manual control or challenge themselves with algorithmic navigation in automated mode.

IX. Scoring System

Formula for Scoring

The scoring system is based on the number of steps, the numbers of collected item and obstacle clashed.There are three types of obstacle with values -2,-5 and -10. Each time any obstacle is collided the respective value is deducted from the score. Similarly there are three types of collective items with values 5,10 and 20. And added to the score each time any item is collected.

Storage of Scores in a File

Scores and collective items are stored in a file for future reference and leaderboard. The leaderboard showcases top performers and allows players to track their progress over multiple sessions.

X. Testing and Optimization

Overview of the Testing Process

Testing involves multiple gameplay sessions, with on user interaction focused , obstacle management , and scoring accuracy.

XI. Conclusion

Summary of the Project

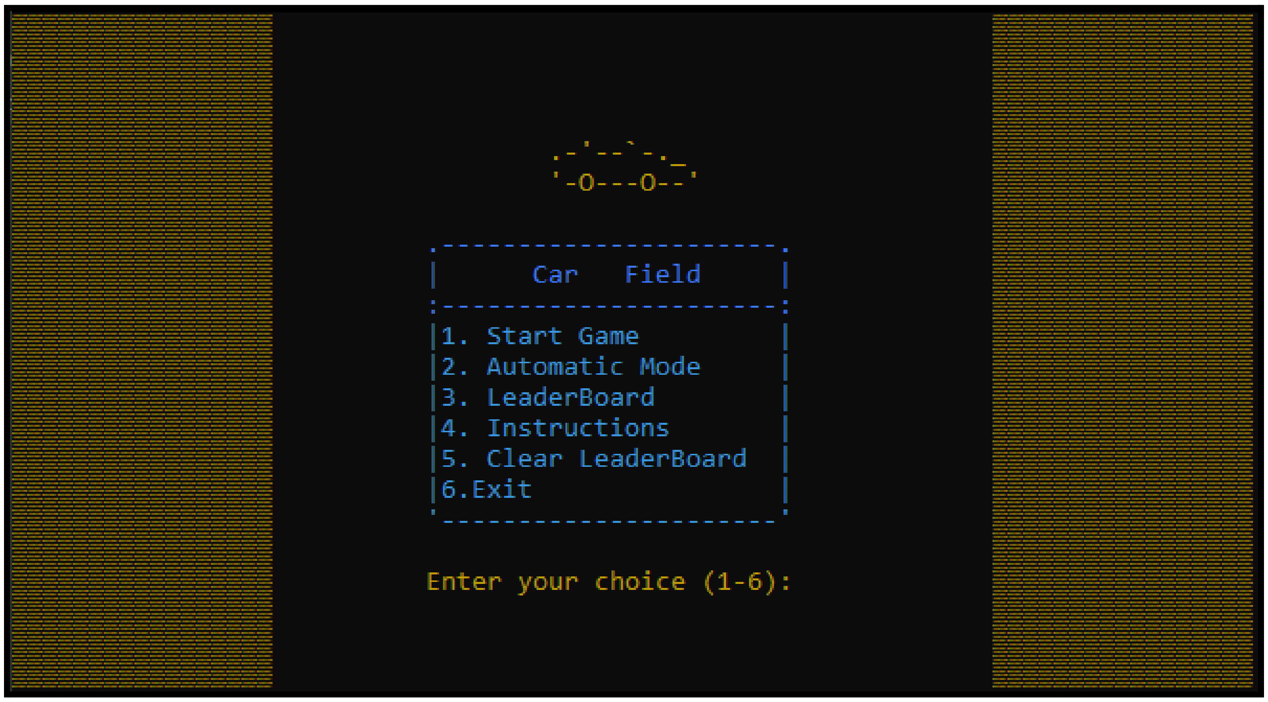
The Race Car Game project successfully combines entertainment with educational value, showcasing the practical application of data structures in game development.

Achievements and Challenges Faced During Development

Key achievements include the implementation of dynamic graph-based grid and a responsive user interface. Challenges also included balancing difficulty levels and refining the scoring system.

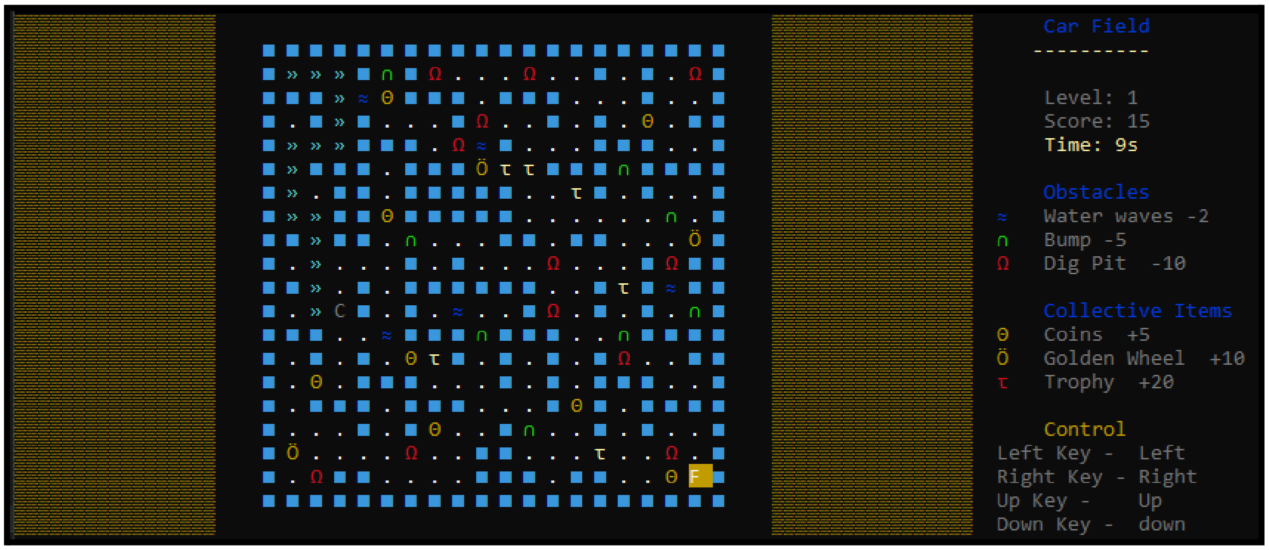
XII. Project Demo

Menu Screen



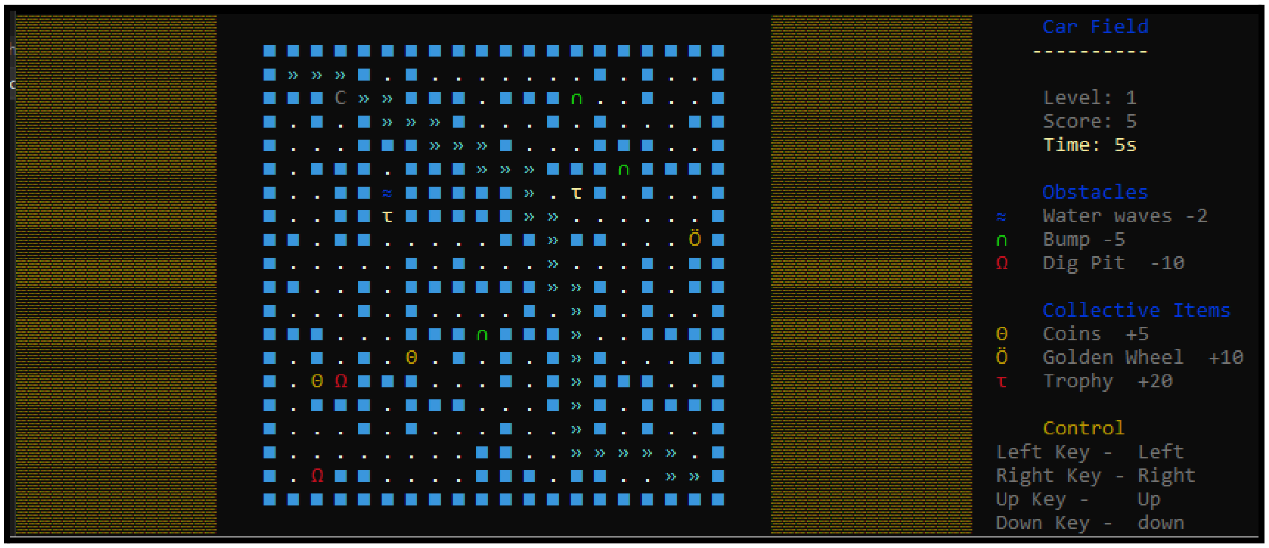
```
          .-'--`-._
          '-O---O--'

  .----------------------.
  |      Car   Field     |
  :----------------------:
  |1. Start Game         |
  |2. Automatic Mode     |
  |3. LeaderBoard        |
  |4. Instructions       |
  |5. Clear LeaderBoard  |
  |6.Exit                |
  '----------------------'

 Enter your choice (1-6):
```

Manual Game Screen



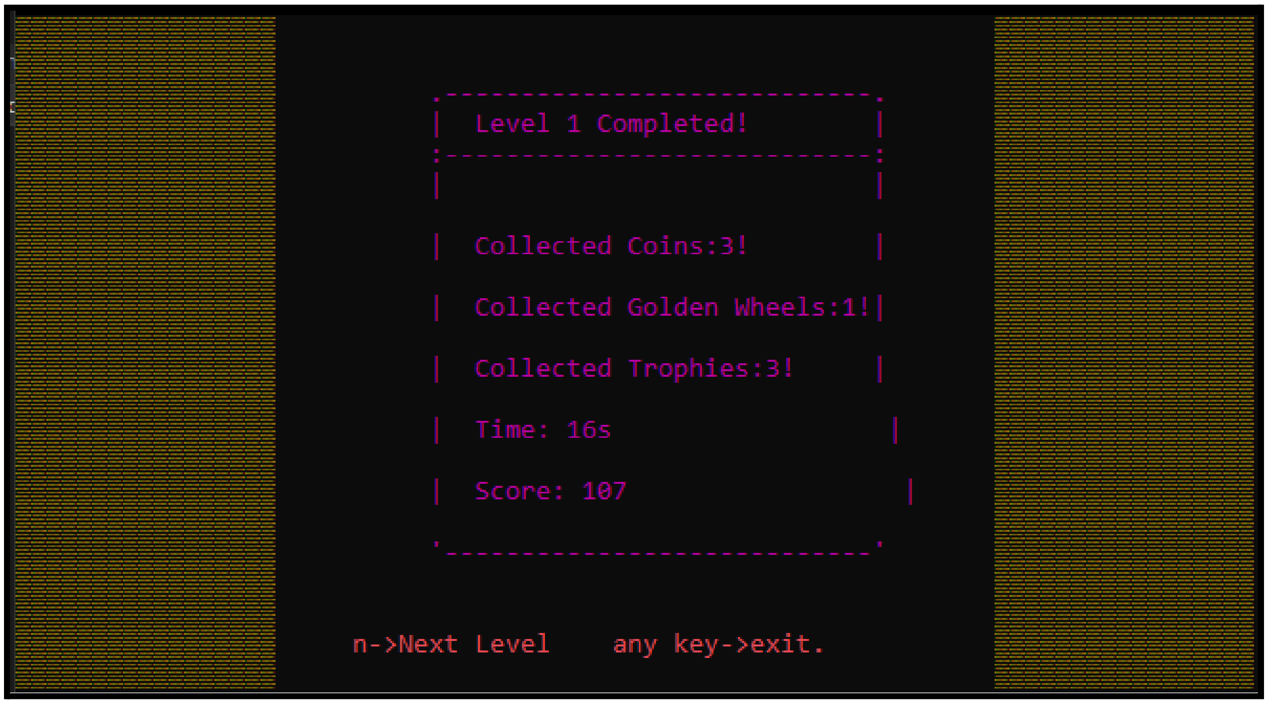
```
      Car Field
     ----------

      Level: 1
      Score: 15
      Time: 9s

      Obstacles
≈     Water waves -2
∩     Bump -5
Ω     Dig Pit  -10

      Collective Items
θ     Coins  +5
Ö     Golden Wheel  +10
τ     Trophy  +20

      Control
Left Key -  Left
Right Key - Right
Up Key -    Up
Down Key -  down
```

Automatic Game Screen



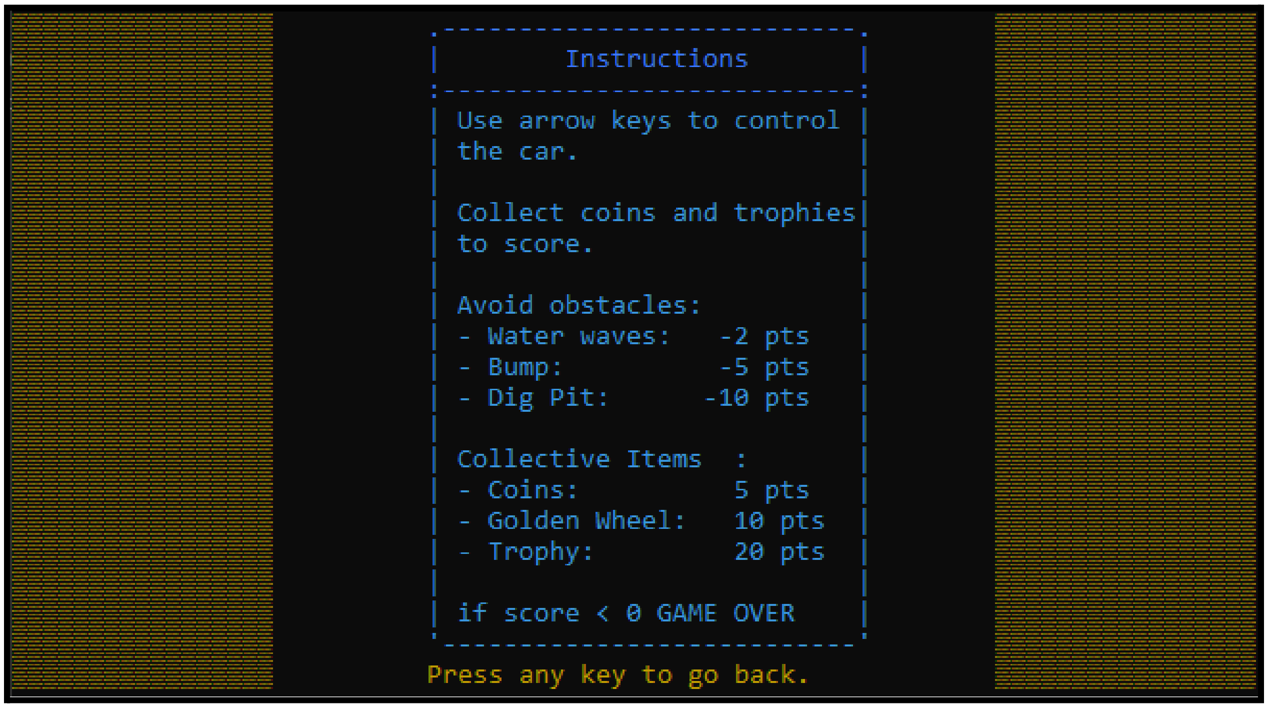
```
      Car Field
     ----------

      Level: 1
      Score: 5
      Time: 5s

      Obstacles
≈     Water waves -2
∩     Bump -5
Ω     Dig Pit  -10

      Collective Items
θ     Coins  +5
Ö     Golden Wheel  +10
τ     Trophy  +20

      Control
Left Key -  Left
Right Key - Right
Up Key -    Up
Down Key -  down
```

Leader Board Screen



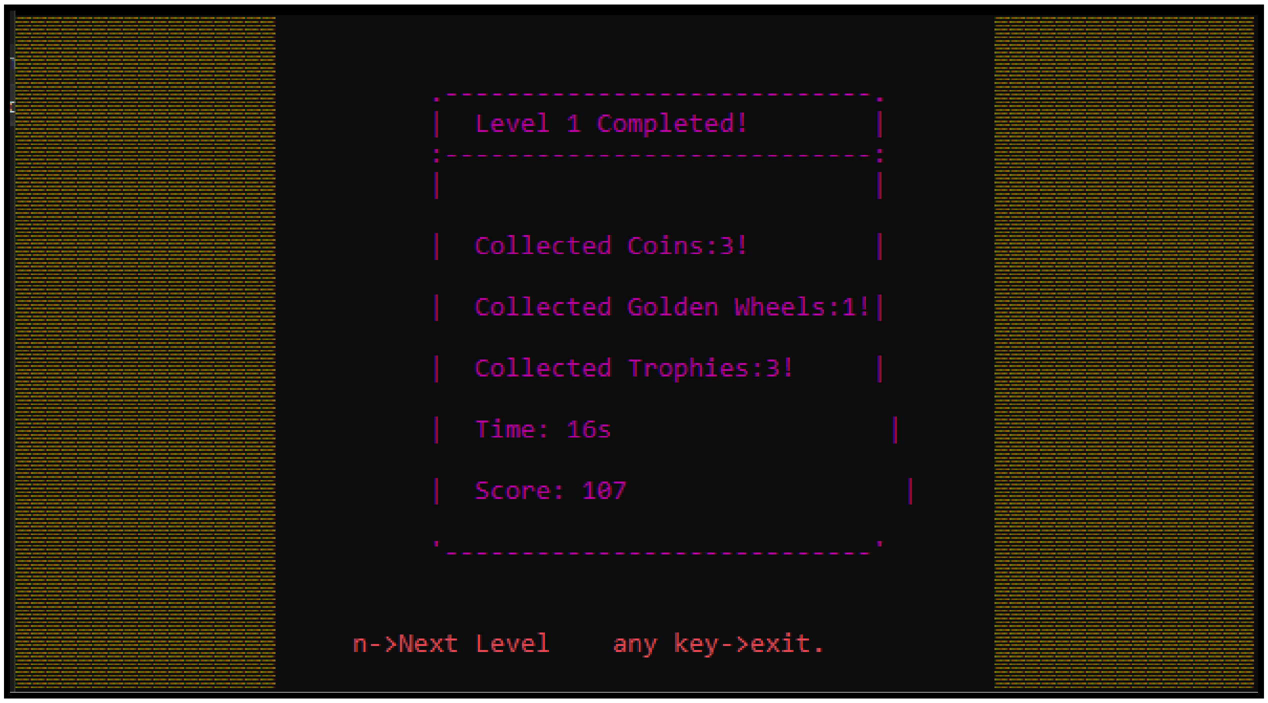
```
.-----------------------------.
|  Level 1 Completed!         |
:-----------------------------:
|                             |

|  Collected Coins:3!         |

|  Collected Golden Wheels:1!|

|  Collected Trophies:3!      |

|  Time: 16s                    |

|  Score: 107                     |

'-----------------------------'


n->Next Level    any key->exit.
```

Instructions Screen



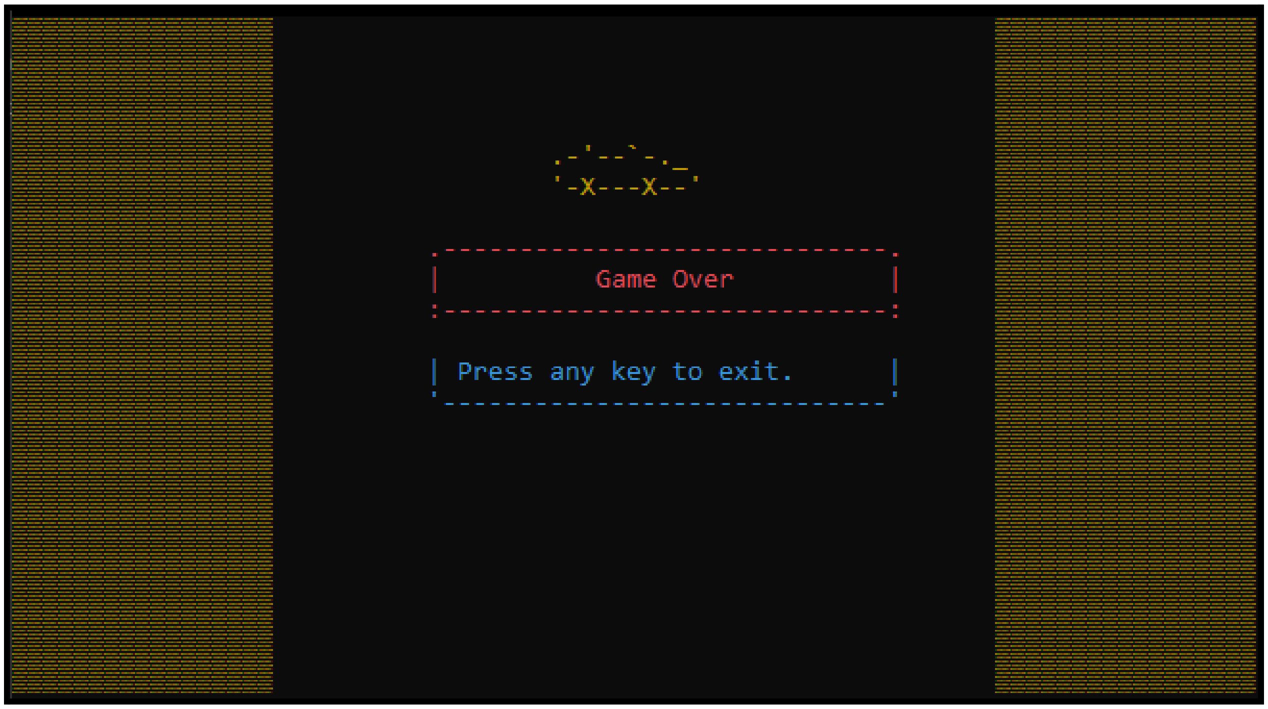
```
.---------------------------.
|        Instructions       |
:---------------------------:
| Use arrow keys to control |
| the car.                  |
|                           |
| Collect coins and trophies|
| to score.                 |
|                           |
| Avoid obstacles:          |
| - Water waves:    -2 pts  |
| - Bump:           -5 pts  |
| - Dig Pit:       -10 pts  |
|                           |
| Collective Items   :      |
| - Coins:           5 pts  |
| - Golden Wheel:    10 pts |
| - Trophy:          20 pts |
|                           |
| if score < 0 GAME OVER    |
'---------------------------'
Press any key to go back.
```

Level Complete Screen



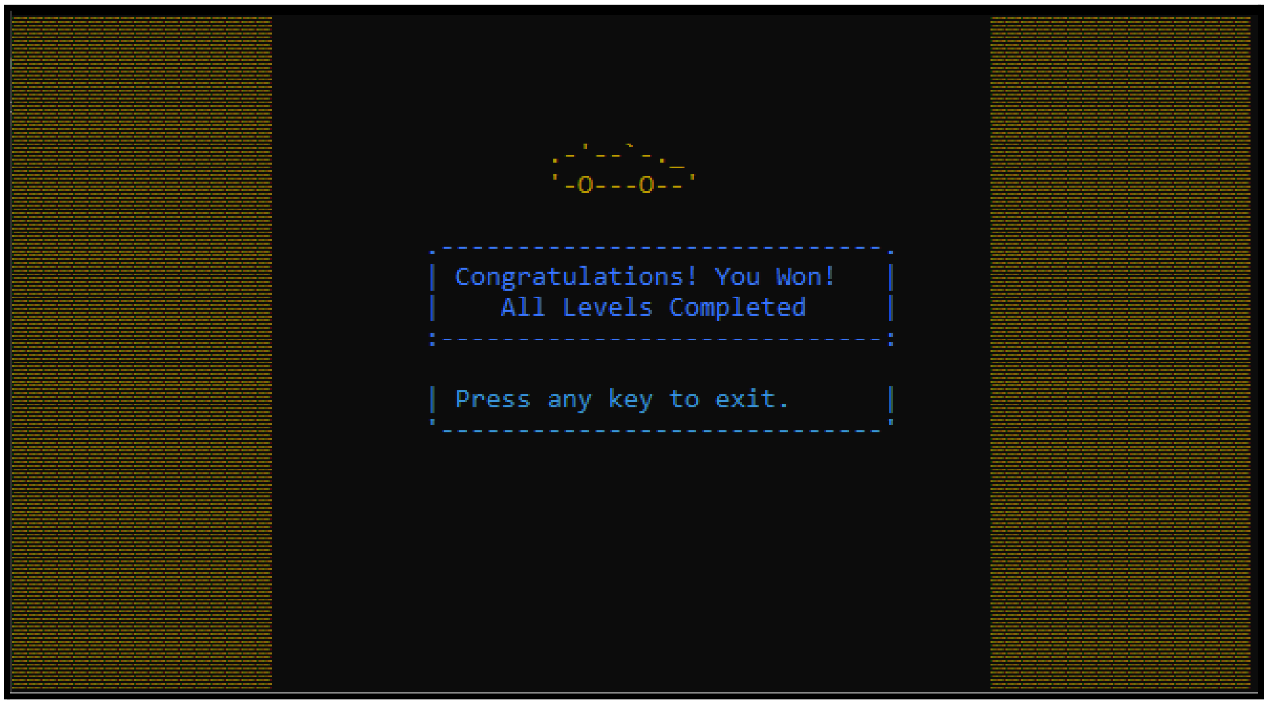
```
.----------------------------.
|  Level 1 Completed!        |
:----------------------------:
|                            |

|  Collected Coins:3!        |

|  Collected Golden Wheels:1!|

|  Collected Trophies:3!     |

|  Time: 16s                  |

|  Score: 107                  |

'----------------------------'


n->Next Level    any key->exit.
```

Game Over Screen



```
        .-'--`-._
        '-X---X--'

.------------------------------.
|          Game Over           |
:------------------------------:

| Press any key to exit.       |
'------------------------------'
```

All Levels Completed Screen



```
         .-'--`-._
         '-O---O--'

.-----------------------------.
| Congratulations! You Won!   |
|    All Levels Completed     |
:-----------------------------:

| Press any key to exit.      |
'-----------------------------'
```

XIII. Future Improvements

Suggestions for Future Enhancements

Potential improvements include additional game modes, more diverse maze structures, and integration of advanced algorithms for pathfinding.
More number of levels will be added in future to the game