**CAR RACE GAME (DS PROJECT)**

**TEAM MEMBERS:**

**NAME: ZAID \_ 22i-2513**

**NAME: M ALI\_22i-1516**

**DOCUMENTATION:**

**GRAPH CLASS:**

**Purpose:** Represents the main class for the car game.

**Members:**

Various matrices and variables representing the game state.

Instances of queue and linked list classes for game elements.

**Functions:**

**addEdge:** Adds an edge between two vertices in the graph. And we have created manual path in such a way that display of

**addObstacle:** obstacles are created in such a way that are present in queue so that it is added in display function.

**addPowerup:** are create in such a way that are present in queue so that it is added in display function.

**addTrophy:** are create in such a way that are present in queue so that it is added in display function.

**addCoins:** Adds game elements at specified positions.

**display:** Displays the current state of the game. As game run same time clear function is implement in such a way that

**moveCar:** Moves the car based on user input.

**bfs:** Performs breadth-first search traversal.

**totalT, totalC:** Display total trophies and coins collected, respectively.

**display:** Displays the current state of the game. As game run same time clear function is implement in such a way that

**moveCar:** Moves the car based on user input.

**bfs:** Performs breadth-first search traversal.

**totalT, totalC:** Display total trophies and coins collected, respectively.

**Randfun Function:**

**Purpose:** Generates random elements in the game world.

**Parameters:** Takes a copy of the graph (consider passing by reference).

**Note:** The function currently takes a copy of the graph, and changes made won't affect the original graph. You may want to pass it by reference (graph& g) instead.

**2.Created Data Structure Usage:**

**LinkedList:** Used to represent a simple linked list for various game elements.

**Queue:** Represents a queue using linked list nodes for game-related operations.

**Graph:** Utilizes matrices to represent the game world, and various linked lists and queues for handling game elements.

**3. Code Explanation:**

The code represents a car game with a graph-based game world.

The graph class includes functionalities for adding obstacles, power-ups, trophies, and coins.

The game state is displayed on a grid, and the user can move the car based on input.

Breadth-first search is used for traversal.

Randomly generated game elements are handled by the Randfun function.

**4. Notes:**

Consider passing the graph by reference in Randfun to modify the original graph.

Ensure proper resource management and error handling, especially with memory allocations.

**5. Colours:**

As we have used color with obstacles, power up and coin in such a way that it looks attractive.

**6. Menu:**

Our menu have there option as auto game play, manual game play, exit and leader board.