CS 1004 Object Oriented Programming
Semester Project

May 07, 2025

1 Instructions

1. Plagiarism in course project will result in F grade in the course.
2. This is not a group project and each person will be working on the project individually.
3. Make sure you submit your project before the submission time.
4. You can earn bonus marks by implementing extra features in the project.
5. Use OOP concepts to implement the project.
6. Combine all your work in one folder and compress it into a zip file. The folder must contain .cpp files.
7. Use proper naming convention to name the zip folder and the file containing source code. E.g. i24xxxx$_project.cpp, replacei21xxxxwithyourrollnumber.$

2 Rush Hour Game

Rush Hour is a game for one player, in which the player takes up the role of a taxi driver who must collect money by dropping passengers to their destinations.

2.1 General Overview

In this project, you have to create a version of the Rush Hour game, in which a user can choose between two modes i.e. being a taxi driver and a delivery driver in addition to introducing the concept of money per completion of task which will be used to refuel the car.

You will be required to utilize object oriented concepts to complete this project i.e. creating classes for each entity including cars, delivery package, passenger etc along with attributes. Use and implement concepts like composition, association, polymorphism and inheritance in the project. Projects failing to implement any OOP concepts will be not be marked.

2.2 Specifications

In this project, you have to make the Rush Hour game of 1 player with the following features:

2.2.1 Game Menu

- When the game starts, a user should be given the choice in the menu to view the leaderboard (displaying top 10 scores along with player names). The high-scores and player names should be read from the file (named highscores.txt). Store the data into an array. If the score of the current game makes to the list of high-scores, the lowest score is removed from the array and the new contents of the array overwrite the file. Use binary file handling to implement these constraints.
- If the game is being played for the first time the leaderboard will be empty.
- The game allows players to choose between working as a taxi driver or a delivery driver. Players will be prompted to choose their role and they can either select their preferred role or opt to be assigned a random job.
- Once the player has selected their role, they will enter their name.

2.2.2 The Board

- Draw a square game Board that matches your screen size. A sample board is attached below but you are free to be more creative and make as beautiful board as possible.
- The board has four distinct areas, roads(shown in white), usual buildings (shown in black), delivery package pickup points and fuel stations. Cars can be driven anywhere on the roads.
- The number of delivery pickups and fuel stations can range from 2-3 on the board.
- The board consists of the player's taxi or delivery car and other cars, obstacles, passengers, delivery package pickup points and fuel stations.
- At the start of the game, the player's car will be at the top left corner of the board as shown in the picture. Other cars, obstacles, pickup points, fuel stations and passengers will be randomly spread out on the board.
- The board shall always have at least 2 and at most 4 passengers waiting to be picked up. The passengers can be present at any location on the board except the position where other cars or obstacles are present.

- There should always be at least one delivery order available at a pickup point. Each time the game restarts, the delivery pickup points and fuel stations must be randomly placed. However, they should always appear along the side of the road where the buildings are located. Drivers will be able to stop near these points to interact with them.
- There will be a station at any corner of the map where a player can change his role mid game.
- Display the user's current score and earnings in the top right corner along with the time spent in the game.
- The grey colored garage at the bottom left side represents the station where you can switch player roles.

2.2.3 General Gameplay

- The player's car can move in all four directions (up, down, right and left using the four arrow keys.
- The player's car color should always be distinct from other cars on the grid.
- Design separate cars with appropriate colors for each player role.
- All other cars on the board are also moving, they can move up, down, right and left. The other cars cannot stop anywhere during the game.
- Their should be a fuel counter. The fuel will decrease as the player drives his car.
- The car can be refilled at the fuel station using the money earned from taxi driving or delivery driving.
- Please note that when playing as a delivery driver, the map will not show any passengers. Similarly, when driving a taxi, the delivery pickup points must be hidden on the map. However user can change their player role mid match by going to a station on the map and pressing "P". This will open a menu where user can change his user role. Once the user accepts the new role, the game resumes and the map is updated according to the new player role.
- When a player completes two jobs, a new car is added to the grid. and the speed of all the cars is increased.

2.2.4 Taxi Gameplay

- The taxi has to avoid obstacles and other cars and get to the people waiting for a taxi. Once the taxi is next to a person, the player will press the Spacebar to pick the passenger.

- Each passenger would have a desired destination on the board, that would be highlighted in GREEN once the taxi picks up the passenger. The taxi has to get the passenger to the desired location as fast as possible.
- The desired destinations of the passenger will be a randomly generated position on the board. The destination cannot be a place occupied by an obstacle.
- Once the taxi has picked up a passenger, they cannot drop them anywhere except for their desired destination.
- Every time the taxi picks up a passenger and drops them to their desired location the player gets 10 points.
- However, there are also some negative points,
  - If the taxi hits a person -5 points
  - Hits an obstacle (tree or box) -2 points
  - Hits another car -3 points
- Each time the taxi drops two passenger successfully at their desired destination the number of cars on the board will increase by 1.
- Each drive will have some fare which will be added to the player's wallet once the passenger is successfully dropped of at their location.

2.2.5 Delivery Driver Gameplay

- As a delivery driver the player has to avoid obstacles and other cars to get to the delivery pick up point. Once the car is near to the pickup point, press the Spacebar to pick the package.
- The drop off point of the package will be a randomly generated position on the board. The destination cannot be a place occupied by an obstacle.
- Once the player has picked up a package, they cannot drop them anywhere except for their desired drop off point.
- Every time the player picks up a package and drops it at the destination, the player gets 20 points.
- However, there are also some negative points,
  - If the delivery car hits a person -8 points
  - Hits an obstacle (tree or box) -4 points
  - Hits another car -5 points
- Each drive will have some fare which will be added to the player's wallet once the delivery is successfully completed.

2.2.6 Winning Criteria

- The game would have a total running time of 3 minutes.
- The game ends if one of the following criteria is met.
  - The game time of 3 minutes is reached.
  - Player's car runs out of fuel.
  - Players scores gets below zero due to excessive foul play.
  - If the player is able to score 100 points or more in 3 minutes, the player wins the game

2.3 Graphics

You can create a stick figure to represent passengers, a rectangle with two circles to represent a car. A red colored rectangular box to represent a fuel station and a brown square box to represent a delivery package.

2.4 Bonus

- A new board is generated every time a new game is started, however every corner in the board must be reachable. 35% of the board should be buildings and the remaining 65% should be roads.
- Add sound effects on passenger/ delivery pickup/drop off.
- Add sound effects on collisions, game start and completion.

3 Starter Code

We have provided you the starter code in C++ to draw circle, square and triangles of different colors and sizes. You are allowed to change it and add functionality according to the project statement.

4 Deliverables

The project is divided into two phases.

- Phase 1 of the project includes the design of UML diagram for the project. A UML diagram is used to identify and depict the classes/entities in the project along with their relationships (composition, inheritance etc) between them.
- Phase 2 of the project includes the design and implementation of the actual project. Use the UML diagram designed in phase 1 as a reference to design your classes and their relationships with each other.

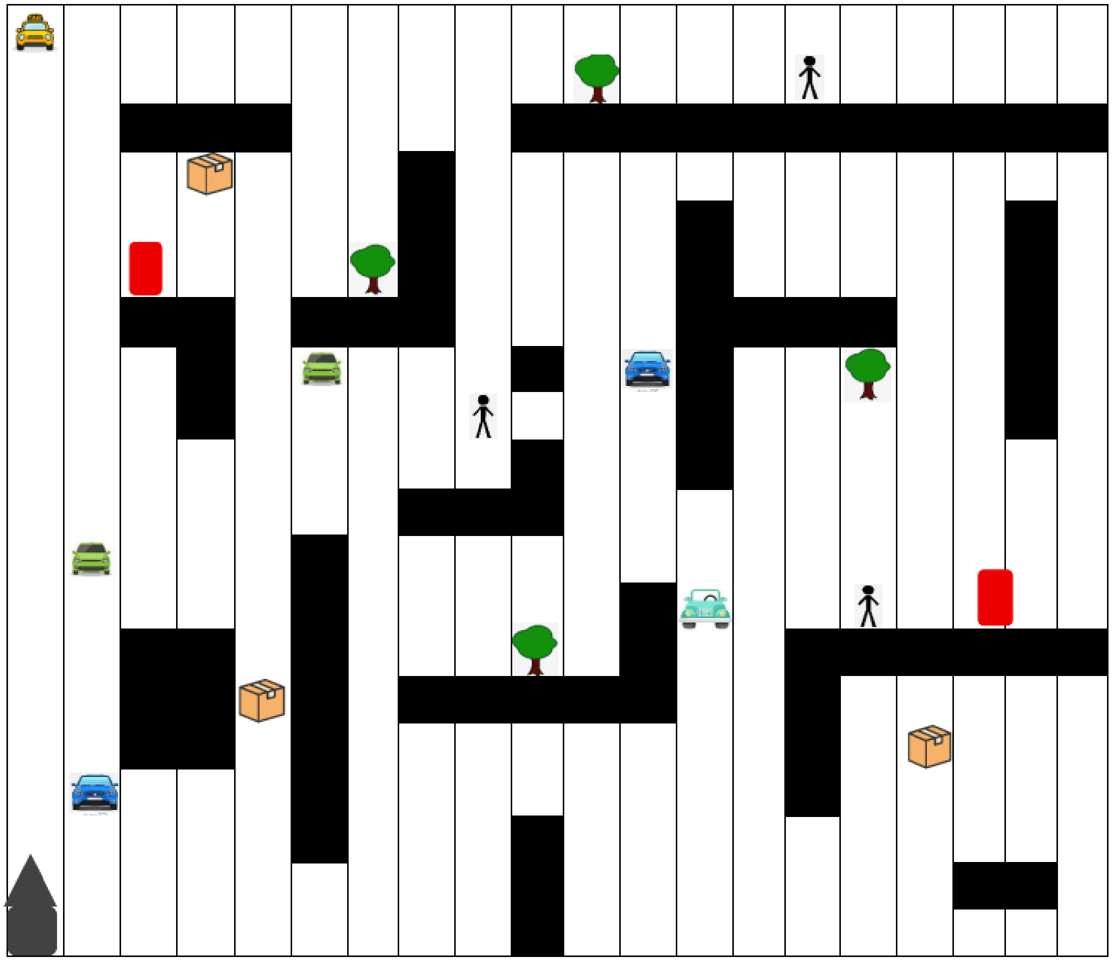

Figure 1: Sample Board