Abraham Arocha

CRC Cards for a Simple Car Racing Game Prototype

1. Class: Race
   1. Responsibilities:
      1. Manage race flow: start, pause, reset
      2. Update race status periodically
      3. Check for race completion
      4. Maintain list of participating cars
      5. Initialize cars at the start
      6. Calculate total time taken by each car
      7. Determine and declare the winner
      8. Receive commands from user interface
      9. Update race display with real-time information
      10. Send race results to leaderboard
   2. Collaborators:
      1. Car
      2. UserInterface
      3. RaceDisplay
      4. Randomizer
      5. Leaderboard
2. Class: Car
   1. Responsibilities:
      1. Participate in the race
      2. Move along the assigned route
      3. Update speed and position
      4. Calculate current speed based on engine, wheels, and obstacles
      5. Calculate total time taken
      6. Notify race of current status (position, speed)
      7. Inform race when route is completed
   2. Collaborators:
      1. Engine
      2. Wheel
      3. Route
      4. Obstacle
      5. Randomizer
      6. Race
3. Class: Engine
4. Responsibilities:
   1. Provide horsepower for acceleration calculations
   2. Define maximum speed
   3. Influence car’s acceleration and speed capabilities
5. Collaborators:
   1. Car
6. Class: Wheel
7. Responsibilities:
   1. Track condition and wear
   2. Adjust grip based on condition
   3. Influence car’s handling and speed
8. Collaborators:
   1. Car
9. Class: Route
   1. Responsibilities:
      1. Contain sequence of locations for the car
      2. Determine distances between stops
      3. Supply next location for the car
      4. Indicate when route is complete
   2. Collaborators:
      1. Car
      2. Location
10. Class: Location
    1. Responsibilities:
       1. Define a point with name and coordinates
       2. Serve as a waypoint
       3. Indicate if an obstacle is present
    2. Collaborators:
       1. Route
       2. Obstacle
11. Class: Obstacle
    1. Responsibilities:
       1. Specify type of obstacle or event
       2. Determine impact on car’s performance
       3. Apply modifications to the car (e.g., reduce speed, cause damage)
    2. Collaborators:
       1. Car
       2. Randomizer
       3. Location
12. Class: Randomizer
    1. Responsibilities:
       1. Provide random speeds within a range
       2. Create random obstacles or events
       3. Introduce unpredictability into the race
    2. Collaborators:
       1. Car
       2. Race
       3. Obstacle
13. Class: RaceDisplay
    1. Responsibilities:
       1. Display race in real time
       2. Show car positions and movements
       3. Reflect changes in car status (speed, position)
       4. Announce race events and winner
    2. Collaborators:
       1. Race
       2. Car
       3. UserInterface
14. Class: UserInterface
    1. Responsibilities:
       1. Capture user input (buttons, controls)
       2. Provide options to start, pause, reset, select parameters
       3. Show menus, settings, race data
       4. Provide feedback messages and status updates
    2. Collaborators:
       1. Race
       2. RaceDisplay
       3. Leaderboard
15. Class: Leaderboard
    1. Responsibilities:
       1. Store results of completed races
       2. Keep track of cars, times, rankings
       3. Provide list of top performers
       4. Show historical data of past races
    2. Collaborators:
       1. Race
       2. UserInterface