Bilkent University

CS 319

Object Oriented Software Engineering

Analysis Report

OtoParker

Group 7 Members:

Arda Usman

Ali Çetin

Çelik Köseoğlu

Hüseyin Beyan

Table of Contents

1. Introduction

OtoParker is a basic car parking game. The purpose of the game is to park the car in a designated spot without colliding the obstacles, and in a limited time period. There are lots of similar games on the market with several different features.

We are mainly influenced by the following game:

link: http://www.kraloyun.com/Oyun/Taksi-park-et

The main difference with our game is that, it will include new features like upgrades and other customization options for your car. Every level will have time constraints which will determine how many stars you get. Collecting these stars will enable the player to purchase upgrades for the car.

We thought this rewarding star collection system incorporated by many modern games is the perfect option the introduce difficulty without confusing the user with several difficulty options. We could see that this system fits better with nowadays' gaming market because it gives the player some flexibility to make the game both challenging and enjoyable.

2. Overview

After entering the game screen, users will see the view of a parking place from above, and a spot that they should park their car in. Using the keyboard arrow keys, the user will navigate the car and park into the spot. They will have a time limit, and

if they park in this period successfully, they will be able to unlock the next level. Their score will gain them stars that shows their success in the level, each level can give the user 3 or less stars according to their time-score. The user will collect these stars and spend them for buying new models or upgrading. In short, there are two goals that users achieve in the game: unlocking all levels and gaining all stars to buy all cars and modifications.

2.1. Controls

Control of the car is done by using the keyboard arrow keys. Up and down keys are for accelerating the car in forward or backward direction. The longer the user presses these keys, the car will accelerate more, and when released, the car will slow down. During car's motion, left and right keys will be used for changing the direction of the car, but they won't give the car acceleration. If the car is upgraded with weapons, space key will be used for firing the weapon. Esc key will be used for pausing the game during gameplay.

2.2. Levels

There will be 10 levels in the first version game with increasing difficulties. Difficulty in these levels is defined by the number and positions of the obstacles. In the upper levels, obstacles will be put such that the user will have to upgrade their car with weapons for destroying them and parking to the spot for completing the level. At the start of the game, all levels expect the first will be locked, and unlocking the next will require finishing the current level.

Unlocked levels can be played again by the user to get a better score and collect more stars from that level.

2.3. Obstacles and Time Limit

In each level, when the car appears in the parking area, there will be several obstacles that user should avoid to hit, on the hit, the level is over and can be started again. In addition to movement constraints that the obstacles provide, there will also be time constraints which will more strict in increasing levels. When time limit is done, again the level will be over. The designed obstacles in this game are:

Other cars: Naturally in the all levels, there will be other cars in the parking area that user's car should not hit. In increasing levels, there will be cars parked in wrong positions or left their doors open, to increase the game's challenge. They will all be static.

Trees: Trees will be a part of the obstacles, they will either be planted in the parking area or fallen on the car's way.

Traffic Cones: They will be placed on various places in the area, to limit car's free space.

Dustbins: They function as the same as traffic cones, only they will be in bigger size.

When the user's car is upgraded with weapon and ammos, they will be able to destroy these obstacles.

Time limit in the game will be different from level to level, depending on the difficulty. Completing the level in a short time is important, because the score of the level will be calculated by using the amount of time left.

2.4. Scores and Stars

With the completion of each level, the user will earn some points based on the remaining time. According to the points earned, the user will gain stars based on their success. Each level will give the users at most 3 stars, meaning that the user can have 1 to 3 stars for each level they complete. If they finish the level in with minimum effort, they will gain 1 star, average success means 2 stars and 3 stars will be earned for the scores well above average.

Also, in the Levels screen, number of stars gained from each level and the total number of gained stars are shown. For example, the user plays the 4th level and their score gave 1 star. After playing it again, with a better score they can earn 2 stars, if this is the case, Levels screen is updated and their total star count is increased by 1. With this system, we try to achieve that to complete the game, user should gain some mastery on the game play. As they play better, they can reach more in-game content by using their stars to make modifications on their car.

2.5. Car Upgrades

The users will start with a standard car type in the game. Upgrades to the car will include changing modal, color, and adding weapon. User can buy these upgrades by spending the collected stars. By this way, the users can personalize their car and they can deal with more challenging obstacles using the weapons.

3. Functional Requirements

3.1. Landing Page

This view will provide navigation to the following game menus:

- Play
- Upgrade Car
- Instructions
- Credits

In addition to these, it will have a simple button at the bottom to give the players the ability to enable or disable in-game sound effects. Also, it will display the number of stars earned, so the play will be aware of possible upgrades.

3.2 Play Button and Levels

After clicking the play button, the user will be navigated to this screen which displays the available levels for the user to play. And it shows the player's progress towards completing all the levels and collecting stars.

3.3 Upgrade Car

After clicking the upgrades button, the user will be navigated to this screen which displays the upgrade options for the player's car. The player will use the stars earned from previous chapters he/she played to upgrade the car. The player will be able to customize the following options:

- the color of the car
- the model of the car
- steering radius (the amount of turn the wheels generate on full lock)
- add weapons for destroying the obstacles

3.4. Instructions

In-game controls and the instructions to play the game will be given. Controls part includes the keys and the mouse buttons to use the application. Mechanics part gives information about how the game is played.

3.5. Credits

The list of the contributors will be shown on this screen.

3.6. Enable/Disable Sound

When the user clicks the sound button on the landing page, this will enable and disable the in-game sound effects.

3.7. Game Engine Mechanics

The car will require certain functions like steering and acceleration.

3.8. Collision Detection

Every obstacle will have a collision box to detect if a crash has occurred or if the weapon has hit it.

3.9. Next Level

If the player could drive the car to the designated (highlighted) zone without colliding with any other objects and in the limited time period, then the current level is completed. However, the player also has the option to collect stars before completing the level. Collect stars to buy upgrades for your car. Next level proposes a higher difficulty and has more turns to complete the level.

3.10. End of the game

If the player collides with any other objects such as other cars, trees or gates, then the game will be over. At the end of the game, a menu will appear to ask the user if he/she wants to retry, go to the upgrades menu, exit to main menu or exit the game.

3.11. Pause Game

Player can pause the game by pressing the escape button on the keyboard. User can pause the game for that particular moment. The pause menu will show four options:

- Upgrades
- Main Menu
- Exit Game
- Resume

3.12. Exit Game

If the player clicks the exit button in the pause menu or closes the application window by clicking the X in the top right corner (red dot on the top left corner on a Mac), the application window will be closed. Current level progress will be lost.

3.13. Timer

Every level will have a time limit depending on its difficulty. The player has to complete the level in this time period or else, the level will be lost.

4.Non-functional Requirements

4.1. Ease of Use

The game should be easily played even without reading the instructions. Main menu navigation should be done with the mouse and in-game controls should be the space bar and the arrow keys like most of the other games on the market.

4.2. Performance

The game has to have a high frame-rate and very sensitive control response in order to help the player in those tight parking spots. Response time from the keyboard is very important because the user would not be expecting a delay after pressing the accelerate key on the keyboard. Otherwise, the user may collide with other objects without purpose.

4.3. Graphics

The visual elements of the game, such as the player's car, the designated parking spot and the obstacles should be easily distinguishable.

4.4. Extensibility

OtoParker's source code should be organized in such a way that it should allow new features to be added easily. For example we may change the star collection or scoring system in the future, add more obstacles and more customization options. These issues should be in consideration during the development process.

5. System Models

5.1. Use Case Model

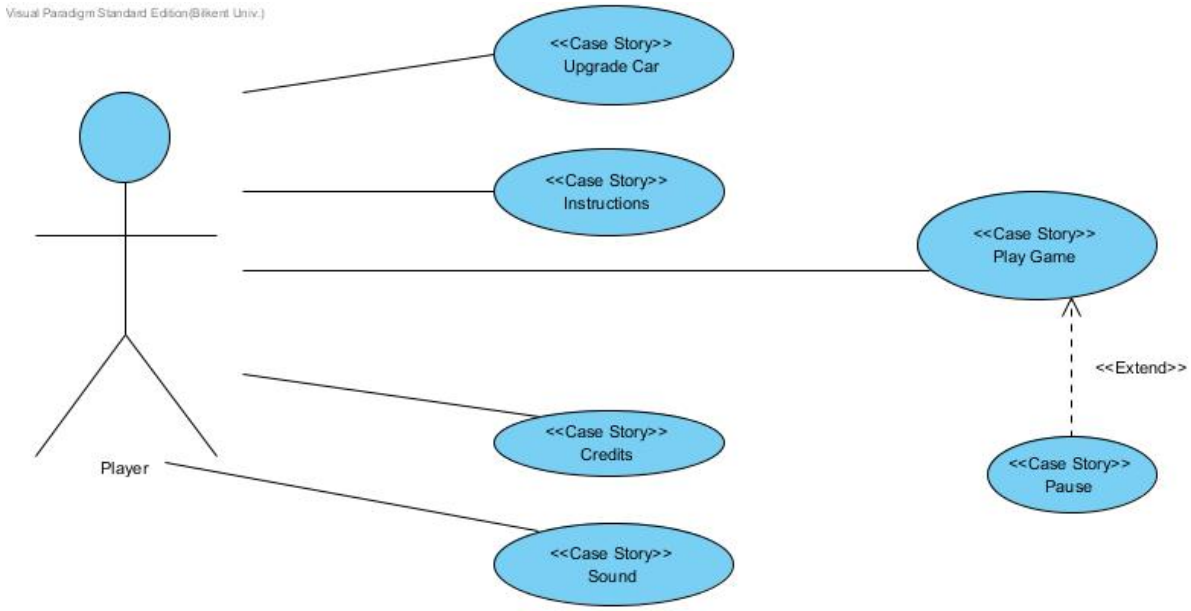

Figure 5.1.1 Illustrates the use case model of the game

5.1.1. View Instructions

Use Case Name: Instructions

Primary Actor: Player

Stakeholders and Interests:

- Player wants to learn how to play the game
- System shows the instructions that are necessary to play the game

Pre – condition:

- The designers have to design the instructions screen before the product screen
- Player should be in main menu

Post – condition: -

Entry condition: Player should choose "Instructions" button from the main menu.

Exit condition: Player selects 'Back' from the main menu'

Success Scenario Event Flow: If the player wants to learn how to play the game, this screen shows the instructions that are necessary.

Alternative Flows:

A. If player desires to return main menu at any time:

A.1. Player clicks "Back" button to return to main menu.

A.2. System displays Main Menu**.**

5.1.2. View Credits

Use Case Name: Credits

Primary Actor: Player

Stakeholders and Interests:

- Player wants to learn about the contributors of the game
- System displays the contributors

Pre – condition:

- Player should be in the Landing Screen.
- The screen must be designed before the product release

Post – condition: -

Entry condition: Player selects the credits button from the main menu

Exit condition: Player selects "Back" to return to main menu.

Success Scenario Event Flow:

1- The system gives information to the user about the developers of the game.

Alternative Flows:

A. If player wishes to return to main menu at any time:

A.1. Player clicks "Back" button to return to main menu.

A.2. System displays Main Menu.

5.1.4.Upgrade Car

Use Case Name: Upgrade Car

Primary Actor: Player

Stakeholders and Interests:

- Player upgrades a selected car in order to complete a level more easily. If the player has enough coins for that upgrade.

Pre – condition: Upgrades must have a specific purpose.

Post – condition: -

Entry condition: Player should press the "Upgrade Car" button.

Exit condition: Player clicks the done button.

Success Scenario Event Flow:

1- The system applies the selected upgrades to the selected car.

Alternative Flows:

1- The system applies the selected upgrades to the selected car.:

1A- Because the user does not have enough coins the upgrades cannot be applied to the selected car.

5.1.4. Play Game

Use Case Name: Upgrade Car

Primary Actor: Player

Stakeholders and Interests:

- Player wants to play the game and successfully pass the levels
- The system comes up with the list of all the levels in the game. The played ones and the locked ones are displayed distinctly from each other.

Pre – condition: The game system has to be ready for user to start playing.

Post – condition: From the levels that are listed in this screen the player has to select the one to start playing.

Entry Condition: Player selects "Play Game" button from the main menu.

Exit Condition: Player selects the "Quit" button on the screen.

Success Scenario Event Flow:

1. The screen to determine the preferences of the game is prepared
2. The level selection has been made successfully
3. Vehicle selection screen has been displayed

Alternative Flows:

2A. Player selects a locked level to play so the game does not allow player to play that game and wants another choice

2B. Player selects a level that has successfully completed before to replay it. The system accepts this as a valid level selection.

5.1.5.1.Actual Play Game

Primary Actor: Player

Stakeholders and Interests:

- Player aims to complete the level and make a high score.
- System keeps the score of the player.

Pre – condition: The preferences that are made by the user before are set for the user.

Post – condition: The score of the player is compared with other scores for that level if that score is the highest it is saved as a high score for that level.

Entry Condition: Player has selected the desired level from the previous screen.

Exit Condition: From the pause menu player selects quit

Success Scenario Event Flow:

1. Game is started by the system.
2. Player starts playing from the selected level.
3. Player plays the game until the car is parked to the pre-selected area successfully.
4. System displays the car upgrade screen before going to the next level.
5. The latest score is compared with the previous high score. If it is greater than the current high score, the high score changes to the latest score. If it is not greater than the current high score, no change occurs.

   This sequence repeats as long as there are levels to play for the player.

1. At the end of the game player sees the credits and then returns to the main menu.

Alternative Flows:

3A.layer plays the game until the car is parked to the pre-selected area successfully.:

3A1. Player presses the esc button to display the pause menu and then clicks 'Quit' to return to the main menu.

3A2. Player presses the esc button to display the pause menu and then changes the sound settings.

3A3. Player crashes the car to a obstacle thus a menu appears and allows player to choose between:

- retry that level
- back to main menu
- go back to upgrades menu

4A.System displays the car upgrade screen before going to the next level.:

4A1:User can select quit to return to the main menu

4A2:User can go directly to the next level without upgrading the car.

4A3: User upgrades the car that is used on this level.

5.2 Dynamic Models

5.2.1 State Chart

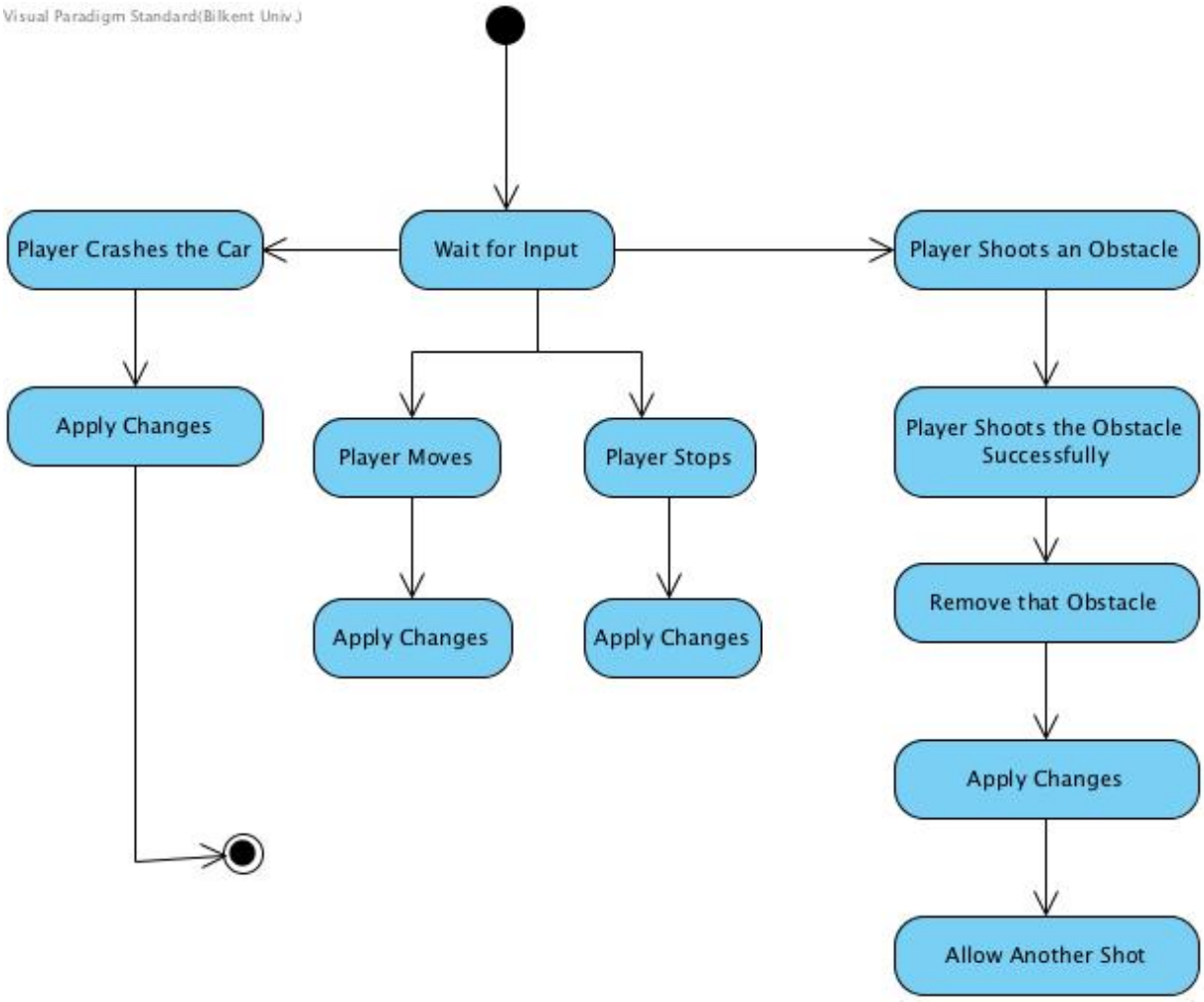

Figure 5.2.1.1: Illustrates the state chart diagram for the player

This state diagram describes the behavior of the player. If the player uses arrow keys, the car will steer and accelerate on the screen. If the player uses the space bar, the car will use the weapon to shoot the obstacle. The obstacle that has been shot is deleted from the screen. If the player stops using the arrow keys the car goes

with its own acceleration and it stops successfully unless it crashes into an obstacle. In the case that when the car crashes to an obstacle, the game is over and a menu pops up asking player between the choices; retry, upgrade the car, go back to main menu.

5.2.2 Activity Chart