**3. Requirement Specification**

**3.1 Functional Requirements**

**3.1.1 Play Game**

Glory for Punch is a kind of an arcade game. The main purpose of the game is beating other boxes in the league and become champion. At the beginning of the game, Player chooses difficulty level which are Normal and God Mode. After that, the player chooses his boxer’s fighting style which are tank, butterfly, hammer. Depending of the Player’s choose his boxer starts the game with little advantages which is increasing his stats little bit. Tank increases his health. Butterfly increases his agility. Hammer increases his attack’s damage.

In Normal mode and God mode, after choosing his fighting style, Player enters the league. His first challenge is fighting against the best boxer in the league. This first match is absolute lose which indicates the Player must train his boxer in the gym by selecting different type of trainings which have different results on boxer’s stats. After completing some trainings, boxer starts to glorify his career by fighting with other boxers in the league. The player starts the league at the bottom, and he tries to increase his position in the league by beating his opponents. In order to win the game, the Player has to beat the best player. If the Player loses the final match, he loses the game. Otherwise, he wins the game.

Winning matches grant some money for paying his apartment’s rent and gym’s rent. However, winning matches won’t grant enough money to live. The player must work or fight in illegal box matches to earn money. Also, the Player can purchase some power ups with his money. These power ups and illegal box matches makes game much more challenging and enjoyable. Also, The Player has only 3 move options in a day. He should choose his actions strategically to win the game smoothly.

In Normal mode, losing a match has no penalty to the player. However, in the God mode, losing consecutive matches causes to lose the game.

**3.1.2 Change Settings**

**3.1.3 Pause Game**

The game can be paused during box matches. Then, the player can continue the match where he paused. However, closing the application causes to lose all the progress.

**3.1.4 View Help**

The player can get all the information about the game in that section. This information mainly aims to show purpose of the game, difference between Normal mode and God mode. Also, shows the controllers of the game.

The information that will be shown:

* Main purpose of the game
* Difference between Normal mode and God mode
* Controllers

**3.1.5 View Credits**

Player can reach information about the game’s developers. Also, it has a contact information. By using this information, the Player can communicate with the developers to help the development of the game in the future.

**3.2 Non-Functional Requirements**

**3.2.1 Game Performance**

The smoothness of the game is one of the most important features. While fighting game should not decrease its response time or frozen totally, or when go from somewhere to another place on the map game should not make user wait etc. We will make the game run smoothly every part of the game even in animation actions. The game will run in high smoothness and performance all times.

**3.2.2 Wide Range Skills**

In a fight game, different actions in fights increase playability of the game. Also it is provide more fun to user in fights and lead different tactics and attack types. Therefore we will develop a game has a wide choice of skills in database. We will show user a rich list of fighting skill so the user can choose three of them and use them in fights. These skill include kicks, punches, and even some other parts of the body.

**3.2.3 Interface**

We are going to build an user-friendly interface which provide users to understand how to play easily and not confused by complicated options or features in the interface. The main aim of our interface is going to be let people who even do not know using computer very well play our game easily. The instructions will be clear and simple interface of the game provide user-friendly game environment.

**3.2.4 Extendibility**

The game has an improvable foundation. There are lots of ways and points can be improved and extended for later updates. Maps can be added or different storylines can be added the game later on. We will also design the game to be easily extended later.

**5. Interface**

**5.1 Main Menu**

Main menu has a simple outline. It has only 5 buttons which are “Start Game, Settings, How To Play, Credits, and Exit”.

Start Game: Button for starting the game. It leads another panel which user will chose game mod.

Setting: This button leads the game options. Settings panel include sound control, appearance control. User will be able to open, close, and adjust sound level in sound control part. User will change the outfit of the player’s boxer. This is only changing in appearance so it does not affect gameplay or level of character.

How To Play: This button open a text panel which explain game controls. It shows which buttons can be used and what they are used for with a detailed explanation.

Credits: This button leads a text panel which contains information of game developers.

Exit: Button for exit the game.

**5.2 Map**

Map shows the overall map of the game. The user can see the place icons in the map which are home, gym, work. User will use map to travel between places in the game. Each place has specific purpose in the game.

Home: Used for rest end of the day.

Gym: The player will use gym to train his boxer in the game.

Work: The player needs money for buying items and rents of gym in the game so he needs to make his boxer to work sometimes. Work is the place where boxer make money other than fights.

**4. Use Cases**

Use Case Name: View Credits

Primary Actor: Player

Stakeholders and Interest:

- User aim to reach the text panel which contain contact information of game developers.

Pre-conditions: User should be in menu

Post-conditions: -

Entry Condition: User push the Credit button in menu.

Exit Condition: User push the Back button to return main menu.

Success Scenario Event Flow:

1. System shows text which contain contact information of the developers.

Alternative Flows:

A. If user wants to return main menu:

A.1. User push the Back button to return main menu.

A.2. System display main menu

Use Case Name: Change Settings

Primary Actor: Player

Stakeholders and Interests:

- Player wants to change game settings which are sound setting and appearance of the boxer.

- System save the changes.

Pre-condition: At first opening of game, system make settings as default. After changes system update new version.

Post-condition: Game settings are updated.

Entry Condition: Player push Settings button in main menu.

Exit Condition: Player push Back button to return main menu.

Success Scenario Event Flow:

1. Player select Settings button and reach the game settings panel.

2. Player can change game settings.

3. System update the changes.

Alternative Flows:

A. If Player wants to return main menu:

A.1. Player presses Back button on Settings screen.

A.2. System updates the changes.

A.3. Player returns main menu.

Use Case Name: Mode settings

Primary Actor: Player

Stakeholders and Interests:

- Player wants to choose game mode, and the boxer’s fighting class.

- System offer normal mode and God mode as difficult level.

- Player can select either Tank, Butterfly, or Hammer as fighting class of the boxer.

Pre-condition: At the first enter to game mode and class is chosen as default values. When Player make changes system will be updated.

Post condition: Mode and class settings are updated.

Entry Condition: Player presses Mode button in main menu.

Exit Condition: Player presses Back button in Mode screen.

Success Scenario Event Flow:

1. Player push the Mode button in the main menu.

2. Mode and fighting class settings are showed as choices.

3. Player choose one the choices.

4. System updates mode and class settings.

Alternative Flows:

A. When Player wishes to return back to the main menu:

A.1. Player presses Back button in the Mode screen.

A.2. System updates mode and class settings.

A.3. Player returns main menu.