Bilkent University
Department of Computer Engineering

Object Oriented Software Engineering Project Report

Project short-name: Curve Fever

Analysis Report

Zeynep Delal Mutlu, Yunus Ölez, Barış Poyraz

Course Instructor: Bora Güngören

Analysis
October 29, 2016

Table Of Contents

1.Introduction

The game we decided to develop is Curve Fever which is an arcade game. Curve Fever is a game which can be played with at least 2 people. It is an offline multiplayer type game which can be played in the same computer.

Information about the game is in Overview and Gameplay part but, mainly, there are players, which are points in a map, where these points leaves a solid pattern, a pattern of their color.

The objective of the game is not to cross each others patterns and the boundaries of the map. The winner is also be the last player standing in the game. Also power ups might appear in randomly, which may either do harm or boost the player.The game we do replica is following:

"Curve Fever 2"

http://forum.curvefever.com/play2.php

The game will be a desktop application and will be played using keyboard.

2.Overview and Gameplay

As stated earlier, Curve Fever is an offline-multiplayer arcade game. The game is easy and fun to play. The entertainment you get in this game is directly proportional with the number of players. As the number of players increases, the game becomes more enjoyable.

In this game, the players are considered as points. The players are allowed to choose their color. The important thing here is that, each player shall have a different color than the others. When the game starts, the players are dropped in a map which has a black background. As the game starts, each point, players, starts to move to forward automatically, which gives players control over their left and right movement, and leave a pattern behind themselves while giving some gaps randomly. These gaps can be used for advantage during game. The players can turn left or right by using keyboard buttons.

Another description for the game is as follows:

"To briefly explain, "Curve Fever", is a free, online multiplayer game where players have the ability to turn left or right leaving a permanent, solid line behind them in their colour. If a player collides with another player's line or the outside boundary, the player loses. Gaps in the line are randomly generated, which players can use to their advantage. The goal of the game is to be "the last player standing" and have the most points!"(Curve Fever Wikia).

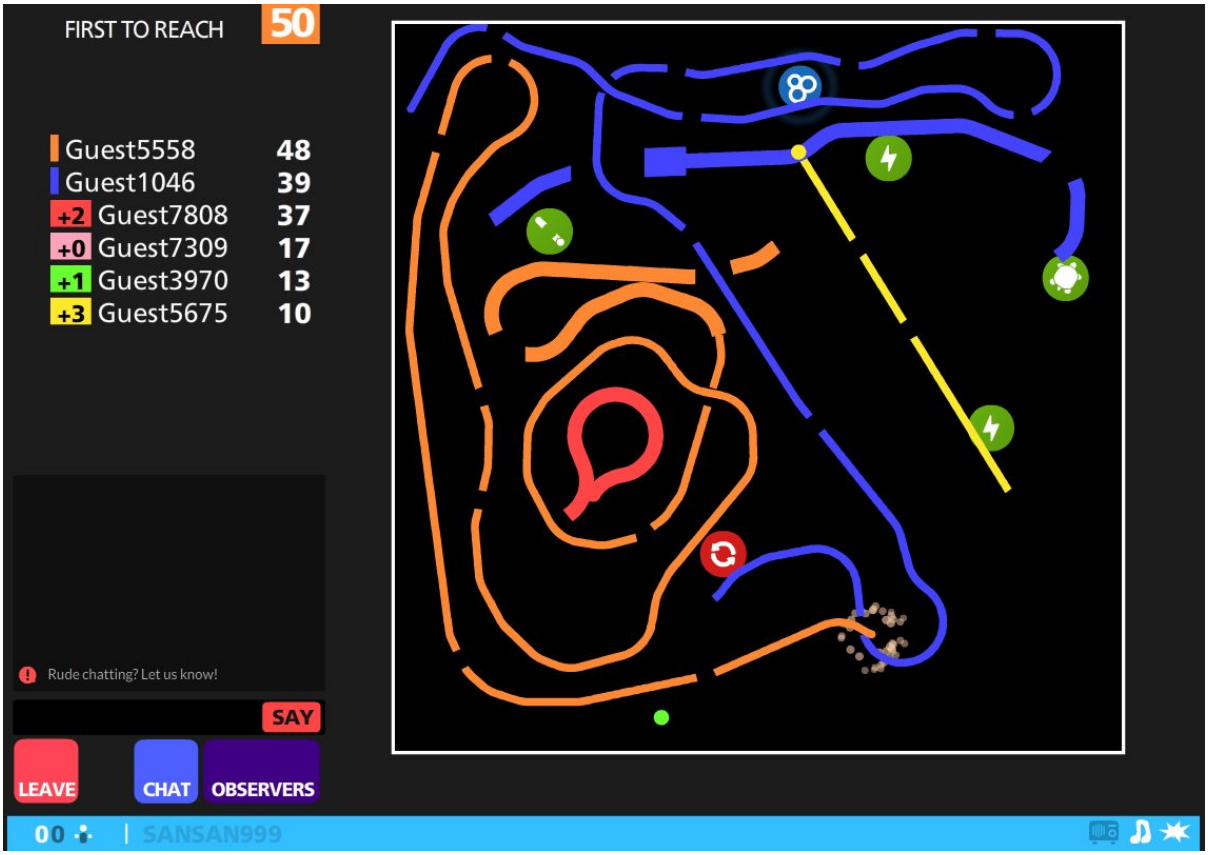


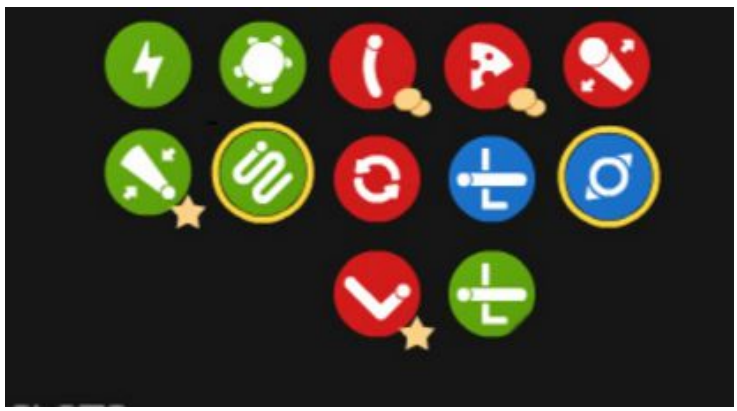
Figure 1. Gameplay from the Original Game
An image from the original Curve Fever found in Google Images. The original game is actually online which requires internet connection but in our version, it will be offline.

As seen from the figure above, there are 6 players playing the game simultaneously. Each of the player has names on the left side of the figure with their colors. The players which can also be told as players leaves a pattern, a solid line behind themselves. From the left part of the figure, where we see number of points

gained from the round, we can understand that, during that part of gameplay, orange and blue points continue playing. For example, the yellow point which leaves a yellow line behind it got 3 points, which means that it collided to one of the colors, blue in that case, after third elimination. The power-ups and players are discussed deeper in the subsections 2.1 & 2.2.

2.1.Power-Ups

During the gameplay, power-ups are very crucial. The power-ups pop up during the game within the boundaries from the set of power-ups that are created earlier. The power-ups are very crucial because, they can change win situation in any game. There are two types of power-ups. These differ in terms of their background color. Green power-ups have an effect on the player who touches to it whereas red power-ups have an effect on the other players and it doesn't not affect the player who takes it. There will be different types of power-ups, which will be taken as an example from the original game and implement. Examples of power-ups can be seen from the figure 2.i and from the image below



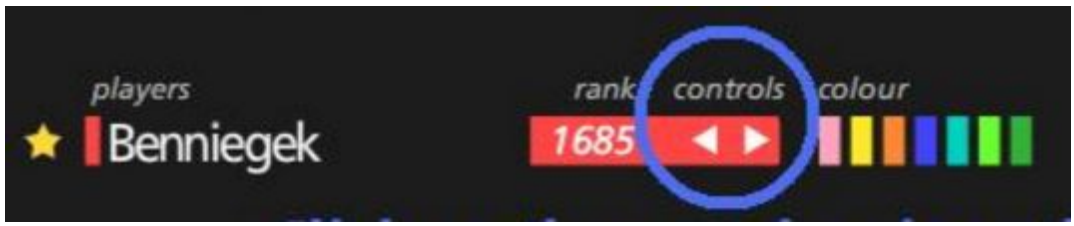
2.2.Player

Each player or points has a unique name, color and key configurations to play the game with in order to participate and play the game. During the player selection part, each person will have the opportunity to create a player with these specific requirements. Meanwhile, in the game, the players start in an area, which has boundaries colored with color white. Each user can control it's point by using the key

configurations they decided in player selection stage. Forward movement will be done automatically. Players can give left and right directions to their point.

2.3.Key Configurations

Key configurations allow control on points. These configurations will be selected in player selection stage. For the sake of offline multiplayer, only keyboard keys are allowed for the key configuration selection. The players will have to choose two keys. One key for giving left movement, the other for right movement in the game.



3.Requirement Specification

3.1 Functional Requirements

3.1.1 Play Game

- In order to move the curve some certain functions must be added to the game to get inputs from the keyboard. With these inputs the players should be allowed to move their curve to the left or right direction. The last player who survives wins that round and gets points. Players play the game simultaneously.
- To detect collisions between two curves or with the borders, some functions must be added. We are planning to detect collisions with checking the color of the position right in front of the curves head. If the color of that position matches with another curve's color, that would mean a collision. The important thing here is that the pattern of the player who is collapsed into the wall or a player's pattern stays in the game. This changes the gameplay. A figure in User Interface part shows this.

- In order to make the game more exciting and challenging power-ups must be added. Therefore, some functions should spawn power-up's randomly and at random locations.
- Maximum point can a player get in a round is found by (number of players - 1) which is the last one standing player. This means that, the first player who dies gets 0 points and this increases as the players die.

3.1.2 Pause Game

- During the game, players might want to pause or exit the game. In order to add such a capability to the game, a pause button which directs the players to a pause menu must be added. This button should be a button from the keyboard, for example the escape or 'p' button.
- In the pause menu, the players should be capable of pausing or exiting the game with two different buttons. These buttons should be GUI elements.
- When a player hits the pause button, the game should stop and no further progress should happen. The curves must stop their movement and power-ups should stop spawning.

3.1.3 View Help

- To make the game user-friendly and useful a help menu must be added to the game in order to guide the players. This menu should be reachable from the main menu of the program via a GUI button.
- In the help system, there must be some guidelines about the GUI elements, how to play the game and some tips for the novice players.

3.2 Non-Functional Requirements

3.2.1 Performance

Since, the game will have some dynamic movements, to make the players feel more flawless and satisfactory experience when playing the game, the performance of the game should be pleasant. In order to do so, there should be no issues in the code.

3.2.2 Arcade & Last One Standing Themed

In order to make the game more entertaining the GUI elements of the game and the designs of the curves must be matching. Since the players who cross the line or boundaries lose the game, the player that remains in the map wins the round and gets (# of players - 1) points. Pattern of the players who die remains in the map. This increases the difficulty of the game while it brings excitement.

3.2.3 Offline Multiplayer

Today most of the multiplayer games are online and thus, do not have the side-by-side challange and entertainment. With our prject game, it is desired to bring these pleasures back. To do so, this game is planned to be an offline multiplayer game which players can play the game using only one keyboard and one computer.

3.2.4 User-Friendly Interface

To make the game more understandable the GUI elements should be designed and positioned accurately to create an user-friendly interface. However this is not an essential proccess, it is very important to create the scene correctly in order to make the game enjoyable.

3.2.5 Color Selection

The players should be able to select the color of their curves in order to distinguish theirs from others. Therefore, a color selection option must be added to the game whenever a new player is added to the game.

3.3.Constraints

3.3.1.English Language

In terms of game development and marketing, developers should choose a language which is widely used. This way, accessibility of a game increases. For the sake of this purpose, we are constraining ourselves to use English language as our game's user interface.

3.3.2.Java

Java is our second constraint in this project. For the project developers and programmers, Java is a language which has big familiarity in the world of software. Java programs don't require to be installed and Java is famous for being portable.

Also, encountering some errors such as buffering, overflow and segmentation fault in C/C++ are not included to Java. Because of these reasons, we are forcing ourselves to use Java programming language.

3.3.3.Game Start

There is no point for a player to play the game on your own. Therefore, the game can be played only if there are 2 or more players. A practice feature may be implemented if the time allows it.

3.3.4.Playing Field

The dimensions of the gameplay field should be fixed. Players will not be allowed to move outside the gamefield. Also, hitting the borders of the gameplay field should conclude with the death of the player that hit it.

3.3.5.Multiplayer Settings

Every player should have a key configuration to move to the left or right direction during the game. The program should not allow the players to start the game if not all the players have a key configuration.

4.System Models

4.1.Scenarios

4.1.1.Play Game Success Scenario

In the “Main Menu, one of the players clicks on “Play Game”. It results with a “Player Selection” stage in which the players can add or remove players. Since in order to play a game at least two players are needed, the players need to add two players at minimum. Players go through name, color and key configuration stages. After each user adds his player to the game, they click on the “Play” button. Within a countdown by three, the first round starts.

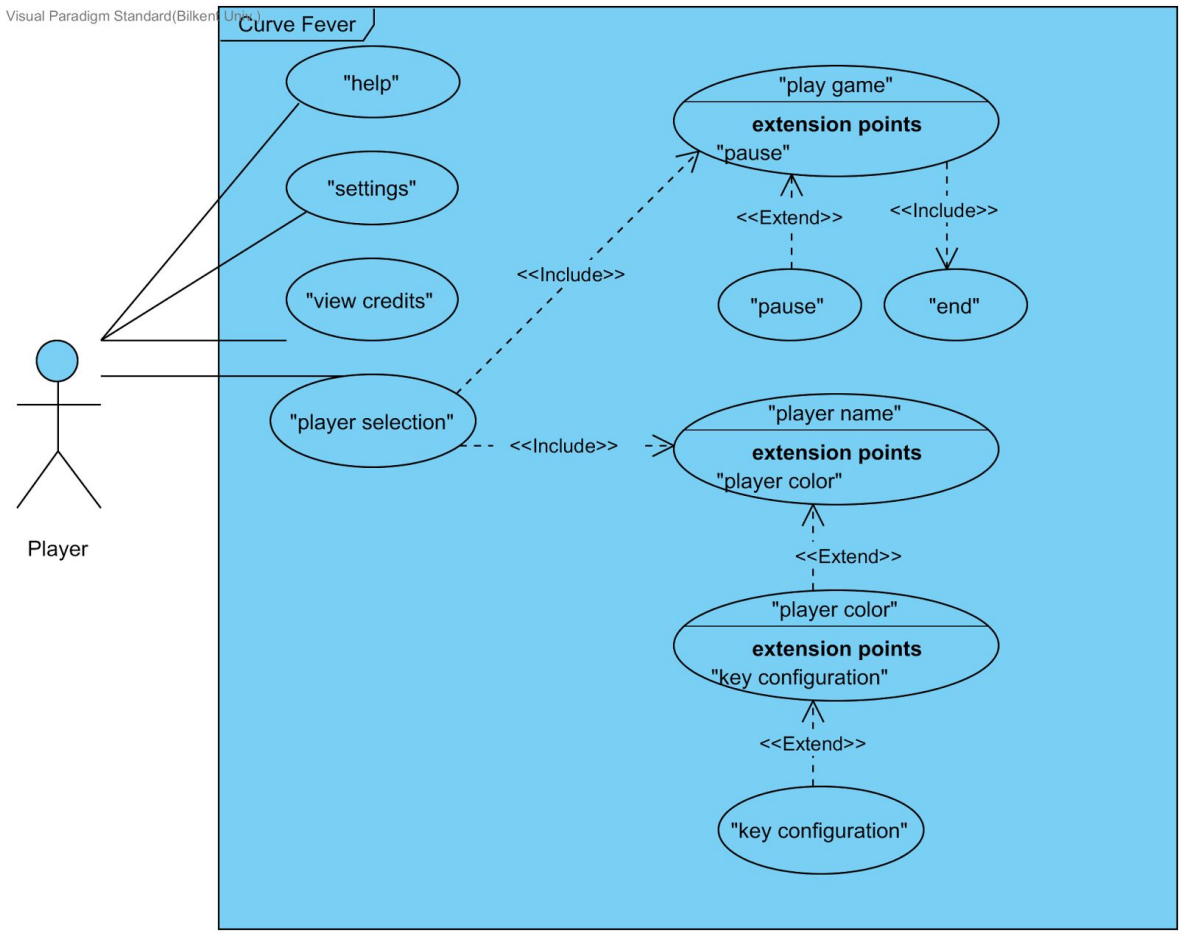
4.1.2.Help Success Scenario

User clicks on the “Help” button in “Main Menu” and reads the manual. After finishing the manual, user clicks on “Back” button to return to the “Main Menu”.

4.1.3. View Credits Scenario

User clicks on the “Credits” button in “Main Menu”. System displays contact information of the developers of the game. User clicks on “Back” button to return to the “Main Menu”.

4.2. Use Case Model



4.2.1. Play Game

Use Case Name: Play Game

Primary Actor: Players

Entry Condition: Each player is selected in appropriate way in terms of name, color and key configuration

One player selects “Play” button from Main Menu.

Exit Condition: One player selects “Exit” button from the main menu

	One player selects “Main Menu” button from Pause Menu.
Flow of event:	Each player selects Player Name. Each player selects Color. Each player selects Key Configuration. After identification, one player clicks “Play” button. Each player try to keep going on playing field without touching to other player’s/players’ pattern. The Player who is first eliminated gets “0” point. The other player/players win one more point than previous player in each time until one player wins the round. The player who gets the 20 points in total means, the one wins the “Play”. For each round, loser player/players wait for completion of round. After completion the play, just winner gets one point and the other player/players get zero. Scores are recorded to players’ total score list. System, returns to Main Menu.

4.2.2. View Help

Use case name:	ViewHelp
Participating actors:	Users
Entry condition(s):	Users can select this option from the main menu
Flow of events:	The gameplay guide will be appeared. Back button provides to close “Help” window and turn back main menı
Exit condition(s):	Click to back button
Postcondition(s):	To be in main menu

4.2.3. View Credit

Use case name:	View Credit
Participating actors:	Users
Entry condition(s):	Users can select this option from the main menu
Flow of events:	User selects the View Credit button. There are two options: 1. Load: Load the suggestions and complaints 2. Back:User turns back to Main Menu without saving any suggestions and complaints.
Exit condition(s):	Click to back button
Postcondition(s):	To be in Main Menu

4.2.4. Pause

Use case name:	Pause
Participating actors:	Player
Entry condition(s):	Players click to Pause button while the play is continuing.
Flow of event:	When the play is continuing, player selects Pause button. There are 2 options: 1.Continue: Continue to Play 2.Exit: Return the Main Menu
Exit condition(s):	User click to “Continue” button to continue to Play.
Postcondition(s):	Play is continuing.

4.3.Events

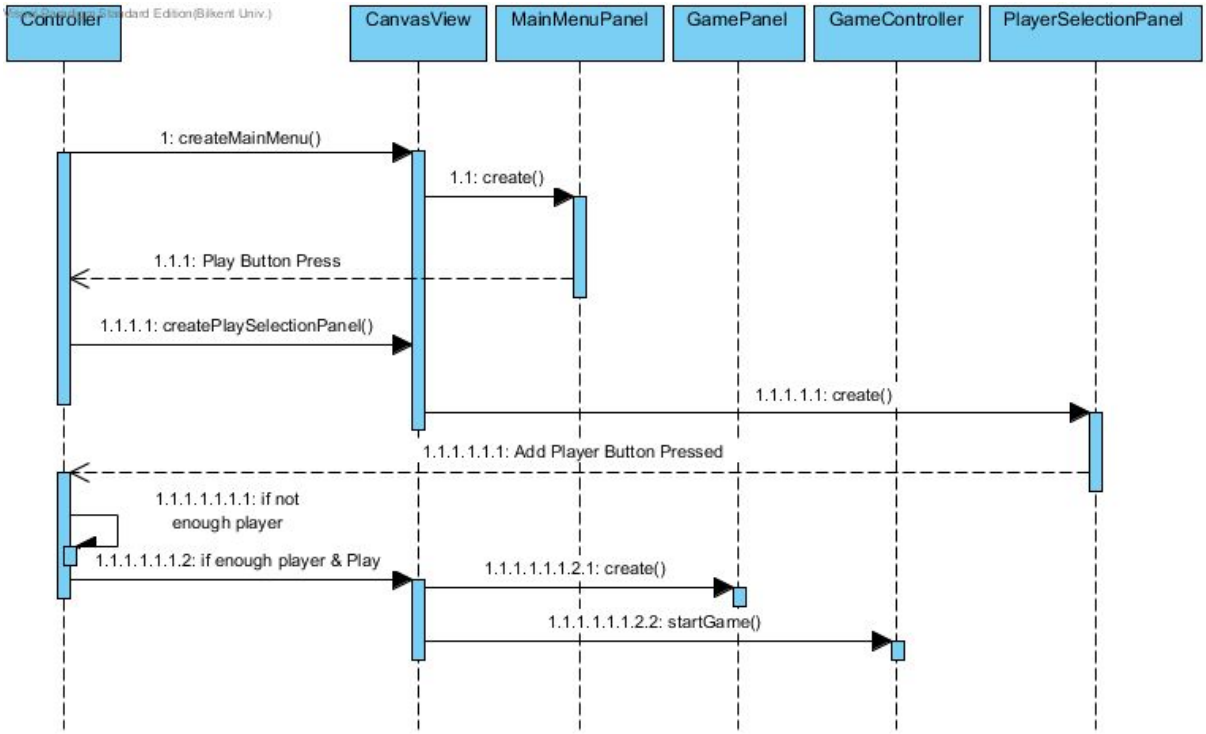
One of the features of our game is that it is event driven. By means of event, it is meant to be power-ups. The power-ups in the game are crucial in terms of the gameplay. They can change the gameplay anytime and can harm or help players in a way. We have decided on some of the power ups that may feature in our game but we may add additional power ups later, after we think and discuss about those in deeper. When they are taken, a timer will start. Until the timer is not equal to a second which we will decide later, the power-ups effect will be effective.

PowerUp	***Effects on the player who takes it***	***Effects on the other players***
Increase Speed	Increases the speed of the player	-
Decrease Speed	Decreases the speed of the player	-
Increase Speed2	-	Increases speeds of the other players
Decrease Speed2	-	Decreases speeds of the other players
Reverse Key Configurations	-	Reverses the key configurations such that if a player turns left using A and turns right using S, he/she will use reverse keys to do that things.

4.4.Dynamic Model

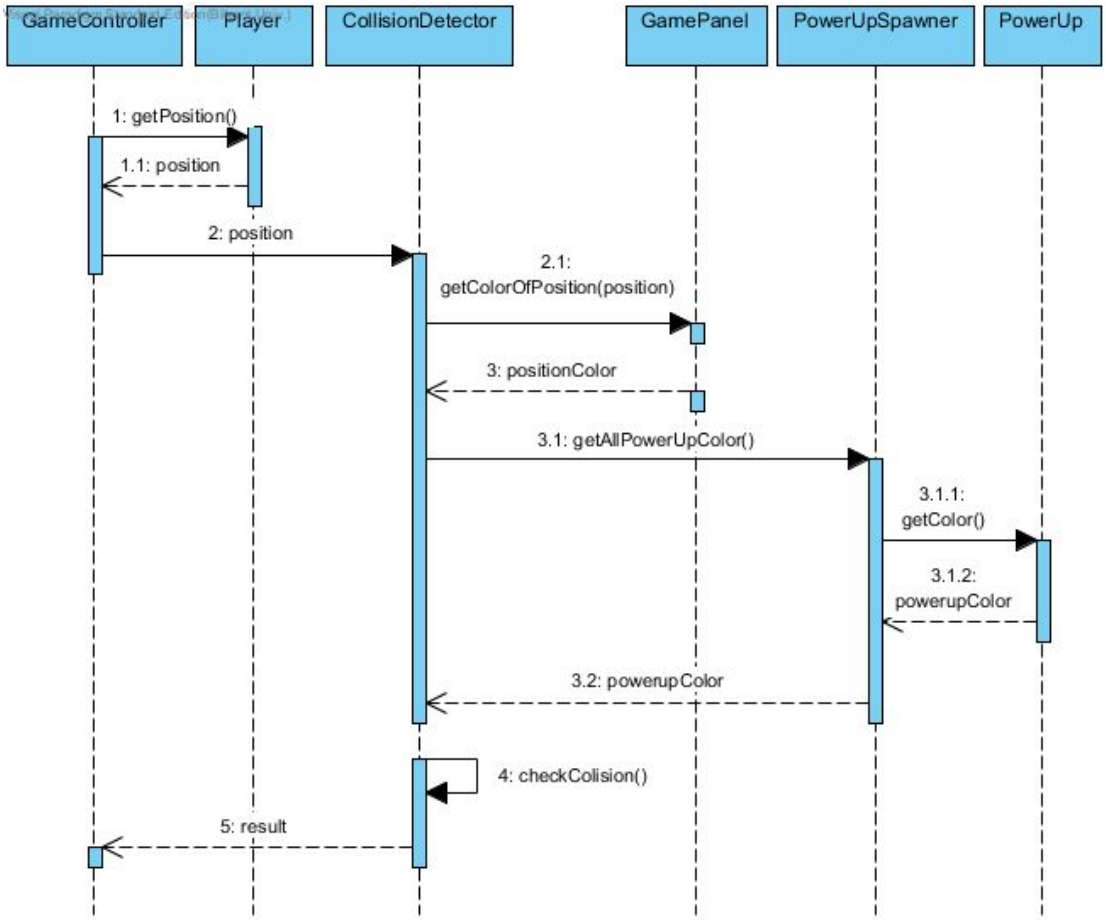
4.4.1.Sequence Diagrams

4.4.1.1.Start Game



When player presses the Play button on the main menu panel, Controller will connect to CanvasView class and main menu will be created by the MainMenuPanel class. Then, player selection panel will be created. On the player selection panel, therre will be adding player button. The adding player operation will continue until the specific player number is achieved. By finishing the adding operation and pressing the Play button, Controller will send a message to CanvasView class which states the players are ready for the game. Finally, CanvasView class will give an order to create GamePanel and pass the baton to GameController class.

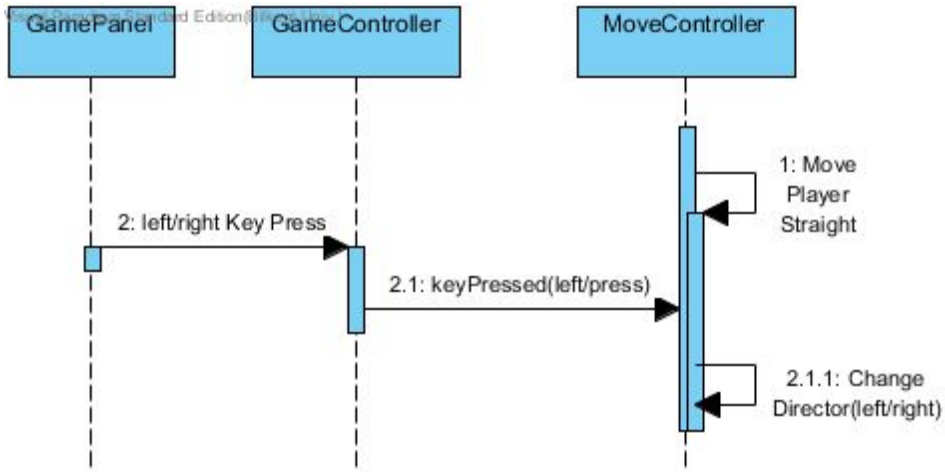
4.4.1.2.Collision Detector



A player can collide with the borders of the game panel, with a power-up, or with another player. The function of the CollisionDetector class is to detect whether a player has collided with one of the things above or not within certain periods of time. To check this, some variables like the position of the player, the colors of power-ups and the color of the pixel right in front of the player should be provided to the CollisionDetector class. The value of the position of the player is read from the Player class via GameController. The colors of power-ups are read from the PowerUp class via PowerUpSpawner and the color of the pixel right in front of the player is being read from the GamePanel. CollisionDetector compare the color of the background with the value read from the GamePanel. If the colors match it means that there are no collisions, however, if the colors do not match it means that there is a collision. If

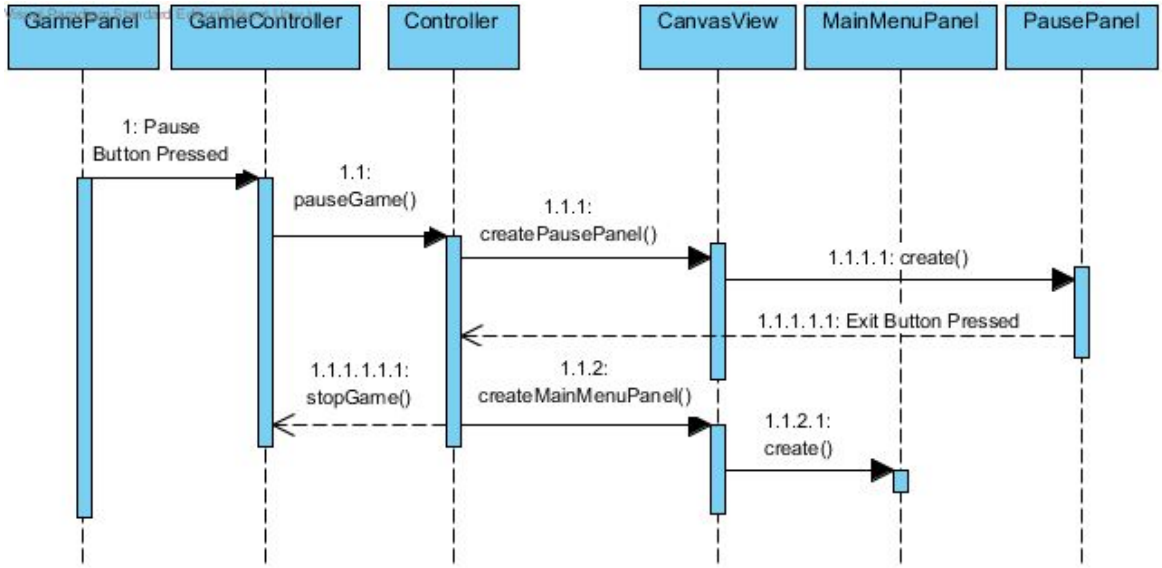
there is a collision, the CollisionDetector compares the color read from the GamePanel with the power-up and player colors. Lastly, it sends an integer to inform the GameController whether the player has collided with something or not.

4.4.1.3.Each Player Move



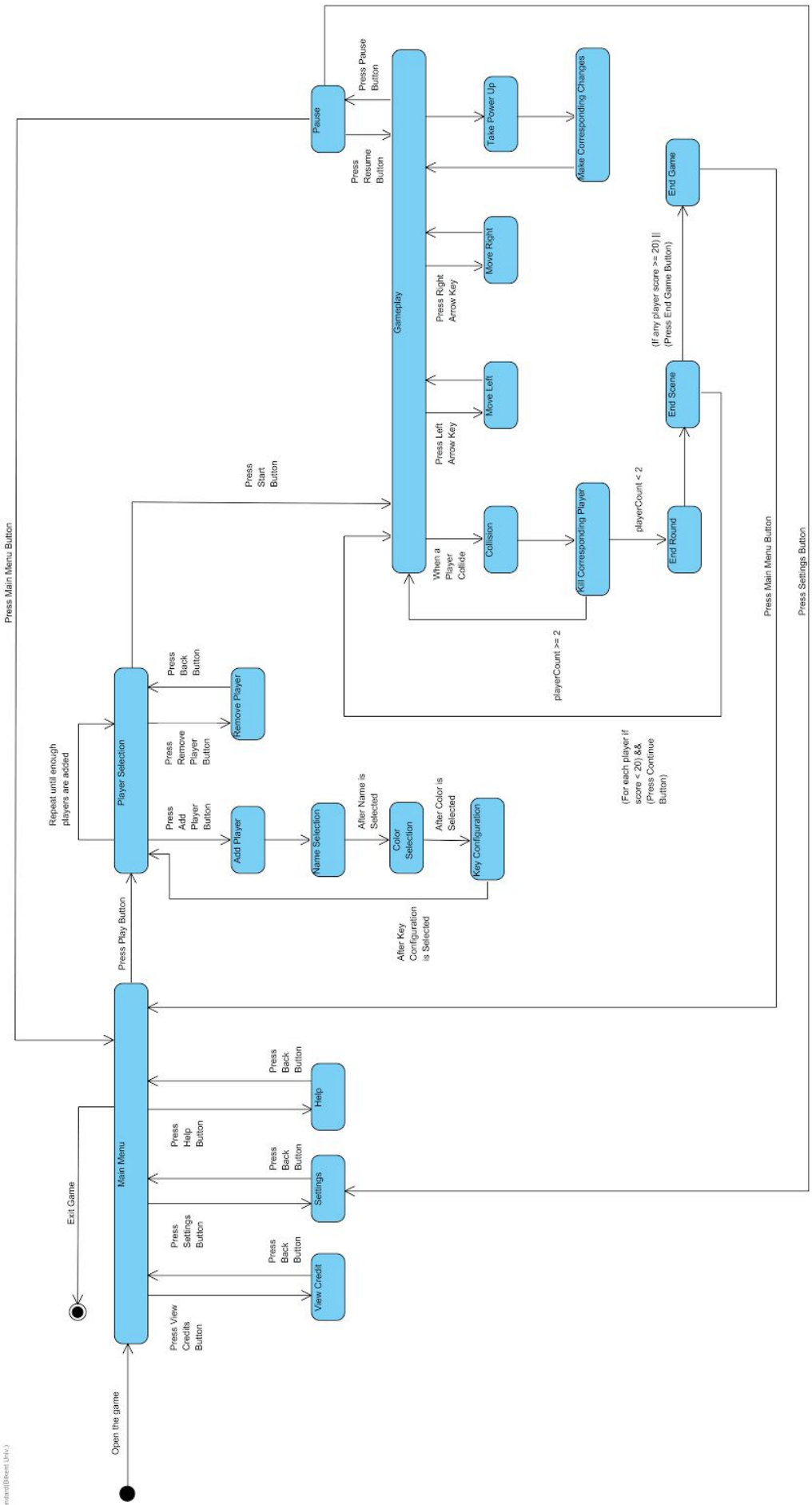
The sequence diagram which demonstrates each player's move has three life lines which are GamePanel, GameController and MoveController. In the game duration, the default action will be moving straight. If any player presses the left or right button, his/her head selectively turns right or left direction.

4.4.1.4.Pause-End Game



The players should be able to pause and exit the game while playing. In order to add such a functionality to the program, some panels and controllers should work together. To detect input from the keyboard GamePanel should be used. It should then send a pulse to the Controller via GameController to create the pausePanel. Simultaneously, the GameController should stop the game. After that, the pausePanel should wait to detect an input from the keyboard to exit the game and signal the Controller that the user wants to exit the game. Then the Controller should inform the CanvasView to create a mainMenuPanel and tell the Controller to end the game.

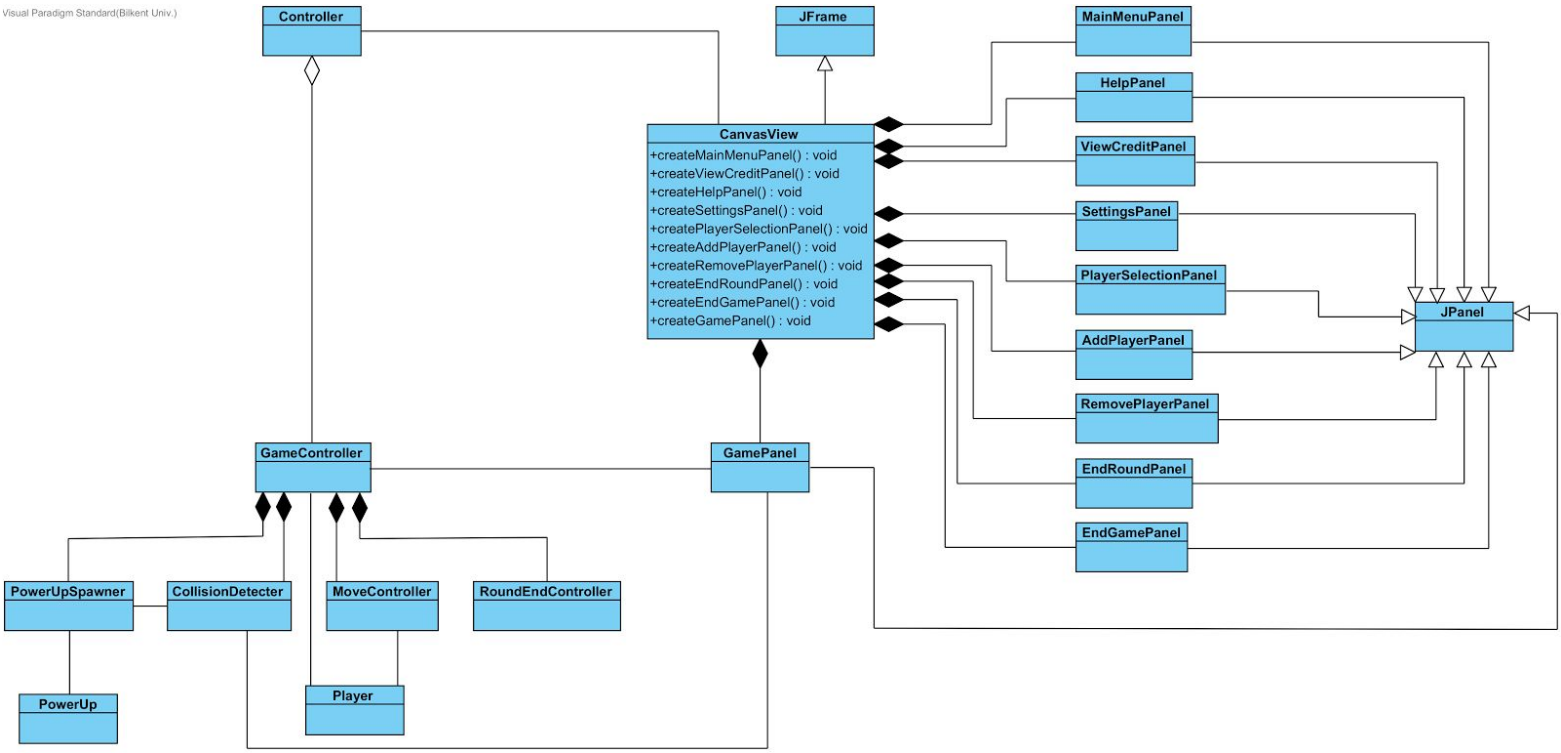
4.5.State Chart



The state diagram above demonstrates a complete overview over our game. When the user opens the game by using the executable file, “Main Menu” is shown directly. From “Main Menu”, user can access to “Settings”, “Help” and “Credits” by using the buttons. If user clicks on the “Settings” button, he/she will gain access to control some things, which will be decided later and whenever he/she clicks on the back button, “Main Menu” screen will be shown. Clicking on “Help” button will change the current screen where the user can learn how to play the game, and can get some tips for the gameplay until he/she clicks on the back button which will direct him/her back to “Main Menu”. Clicking on the “Credits” button will change the screen, and here the names of project members are shown. Finally, if he/she clicks on the play button, “Player Selection” screen is open. Within this panel, user is asked to enter an integer which corresponds to how many players are going to play the game. If user doesn’t enter 2 or more players, will be asked to enter again. When the user enters correct number, “Add Player” screen will come up and will ask user to enter name, select color and choose key configuration respectively. Since players will play on the same keyboard, each of them should select a different key configuration. When this step is done for each player, “Player Selection” screen will again open. This time, user will have a chance to remove players from the game. User can access the “Remove Player” screen from this screen and either can delete a player or go back to “Player Selection” screen by using the back button. When everything is set, user can start the game by pressing the start button. In the “Gameplay” screen, players will be dropped into the map, which will have specific boundaries. Each player will control his/her desired color with respect to their selections in key configuration part. The players will only be able to move their colors either left or right. By pressing the pause button, the game will be paused and “Pause” screen will come up. In this screen, players will have the opportunity, to either resume the game, go to “Settings” or go to “Main Menu” by clicking the specific buttons. In the gameplay, some powerups will be generated and pop up in the map. Each power-up will have a different effect on the game. If a player collides with a player or collapses to one of the player’s pattern, he/she will die for that round, but his/her pattern on the map will stay. When the number of players in the round becomes 1, game will open the “End

Round” screen and from that, if one of the players reach the max score “End Game” screen will open. Otherwise, the new round will start. “End Round” and “End Game” screen will have informations about the round and current situation of the game. In the “End Game” screen, players can go to “Main Menu” screen by pressing the main menu button.

4.6.Object/Class Diagram



The Controller is the head of all other classes and it is responsible of managing other classes and creating communications between them. It directly communicates with the GameController and the CanvasView. The CanvasView is basically a JFrame and is responsible of the GUI elements and creating and destroying panels which are the scenes of the game. The most important panel is the GamePanel and it is the scene where the game is being played. Thus, it has a direct communication with the GameController. The GameController is responsible for controlling the major features of the gameplay. It keeps the game data and reacts when certain things, like collisions, happen. There are 4 other classes which are attached to the GameController and responsible for certain executing and detecting certain things. The PowerUpSpawner

generates random power-ups within a random interval and on a random position on the game field. However, it should check the spawn position in order not to lap with player curves. The CollisionDetector, as it can be understood from its name, is responsible for detecting collisions. It checks the position of each player and whether it had collided with anything or not in a certain amount of time. If it detects a collision it also checks what the player has collided with and informs the GameController. In order to do those, it reads the color of the pixel right in front of the player and compares it with the background, power-ups and other players color. The MoveController is responsible for moving each of the players. It automatically moves players in a straight line in every frame. Other than that, it also changes the direction of a player when the player presses the left or right arrow key. In order to detect a keybord input, it communicates with the GamePanel via the GameController. Finally, the RoundEndController checks the current count of players that are alive. Then, it decides whether to end the round or not. If it decides to end the round it also checks whether any player has reached the target score. If so, it ends the game, else, it starts another round after displaying the score of each player.

4.7.User Interface

Most of the graphics for the user interface will be different from the original game since this one is an offline version of that game.

4.7.1.Main Menu

Figure 2. Main Menu
The "Main Menu" page. Users can access to "Settings", "Help", "Credits" page from here or they can continue to next step by clicking the play button.

4.7.2.Help

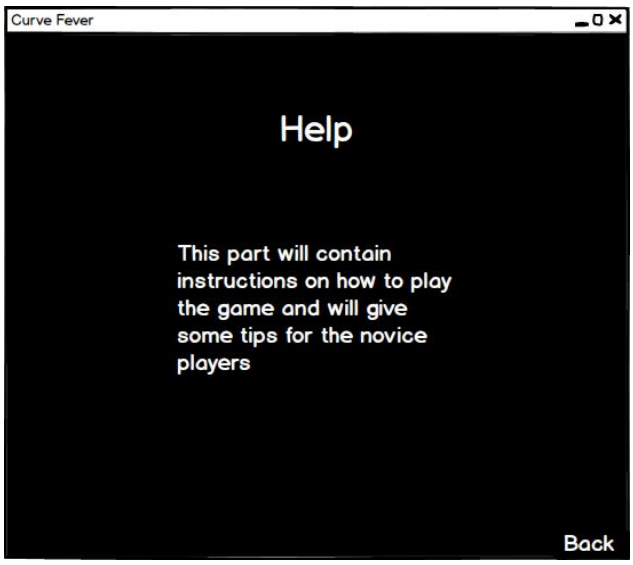


Figure 3. Help
The "Help" page. The basics of gameplay and the tips for gameplay will be summarized here for the user.

4.7.3.Settings

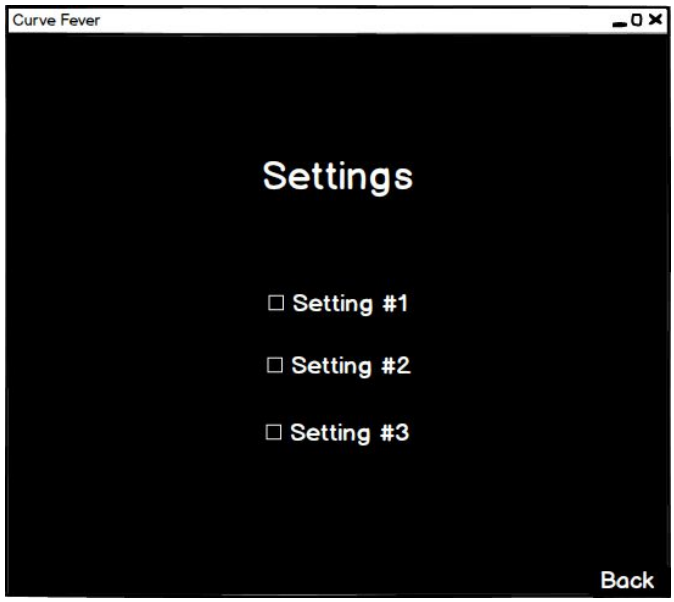


Figure 4. Settings
The "Settings" page. Here the user will have a chance to change some settings that will be decided later.

4.7.4.Credits

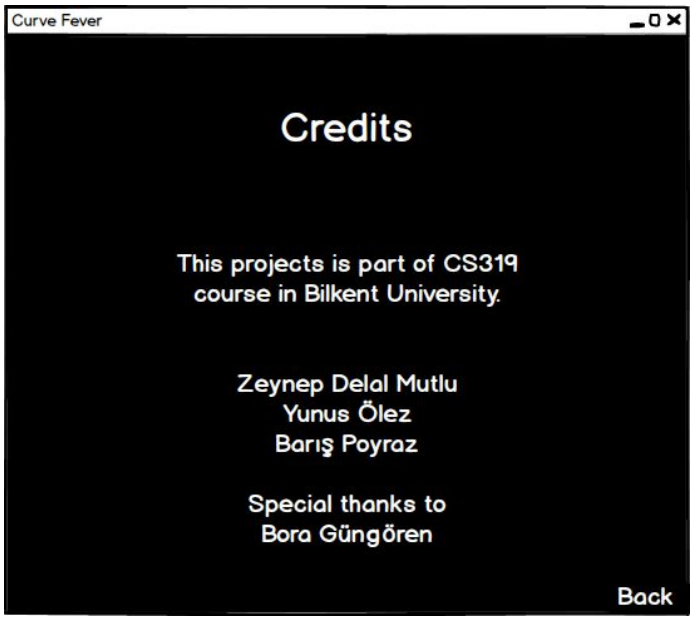


Figure 5. Credits
The "Credits" page. Here the user will see the developers of the game

4.7.5.Player Selection

Figure 6. Player Selection
The "Player Selection" page. Here the user will be asked to enter number of players that will play the game.

4.7.6.Add Player

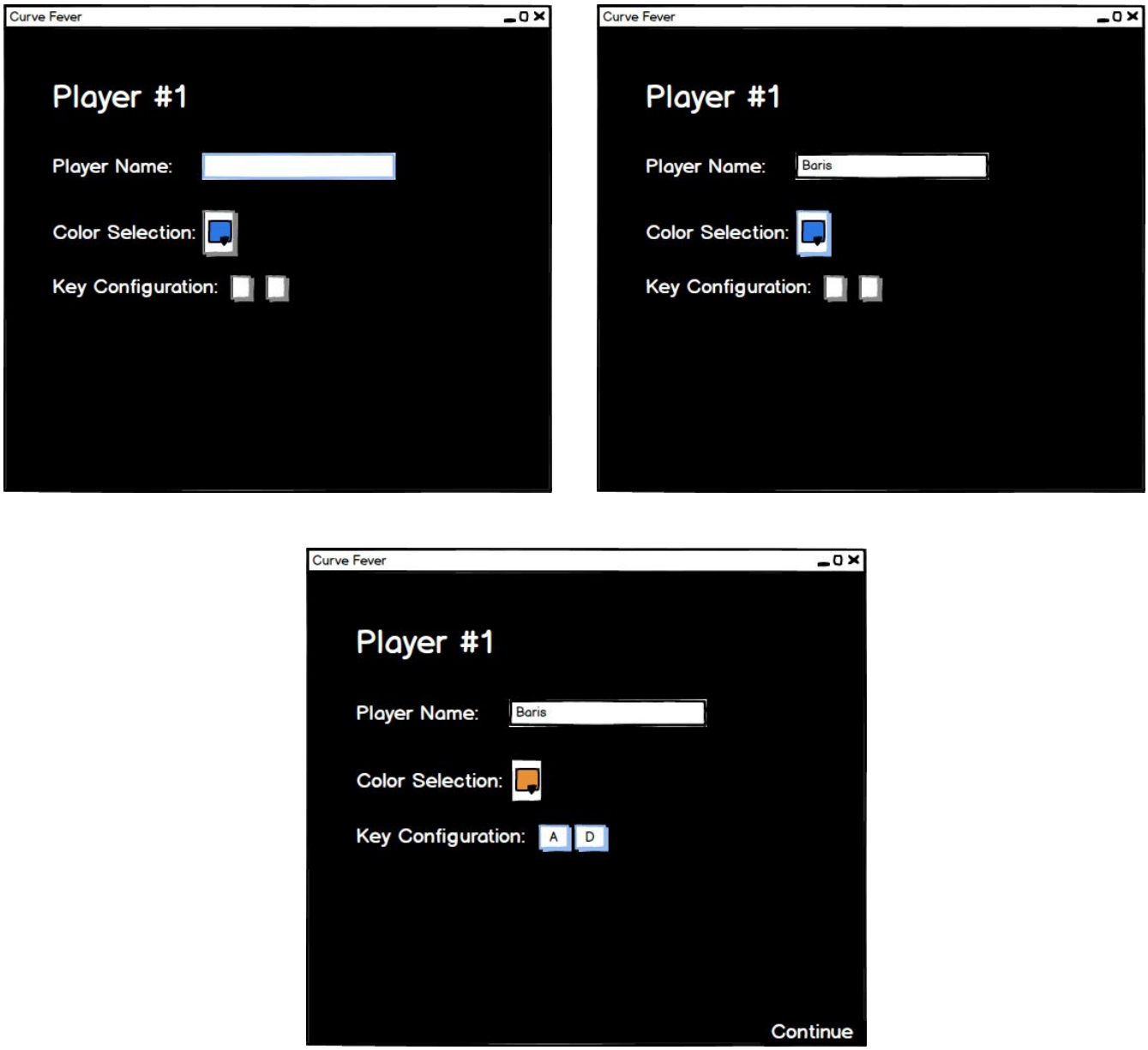


Figure 7, 8, 9. Add Player

"Add Player" screen looks like above the figures. The user must select, choose or type his/her choice with respective order: Player Name, Color Selection, Key Configuration. These figures are put to show that. In the upper left figure, Player Name is focused and the rest is disabled. In the upper right figure, user can select either change his/her player name or can select a color. In the figure that is below the other, user can select all of them, but must select a key configuration to continue.

4.7.7.Player Screen

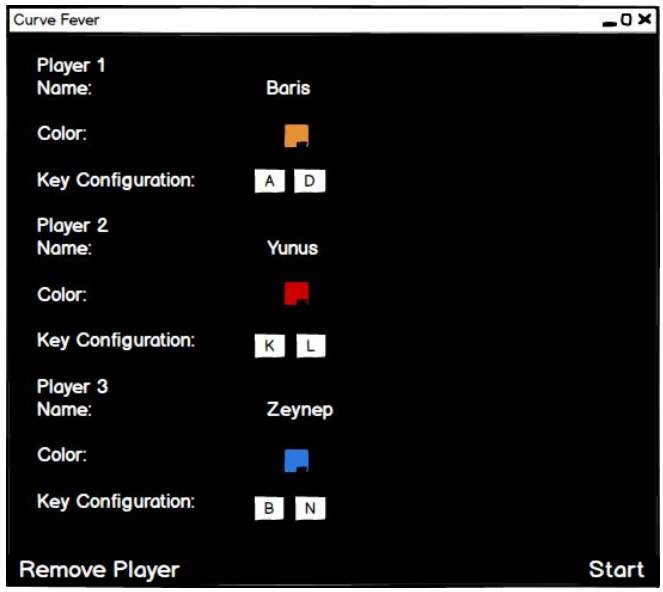


Figure 10. Player Screen

The "Player Screen" page. Here players will see their names, colors and key configurations in one page.

4.7.8.Remove Player

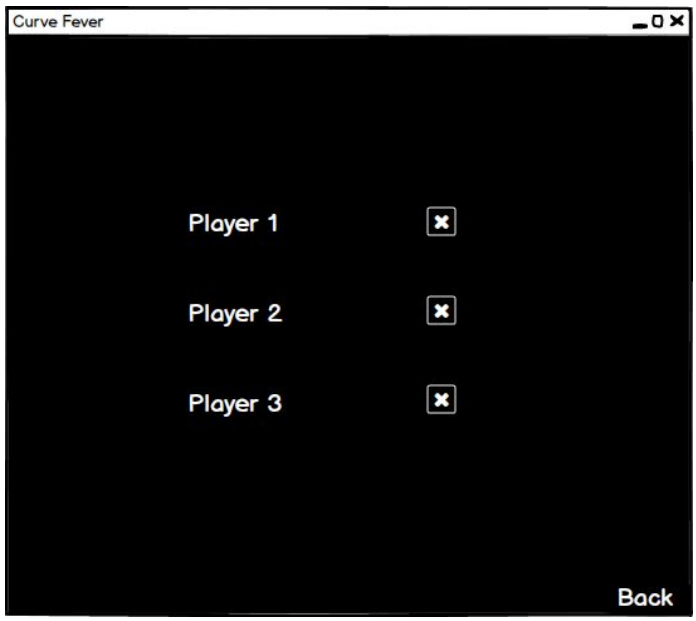


Figure 11. Remove Player

The "Remove Player" page. Here players are given a chance to remove their players.

4.7.9.Gameplay

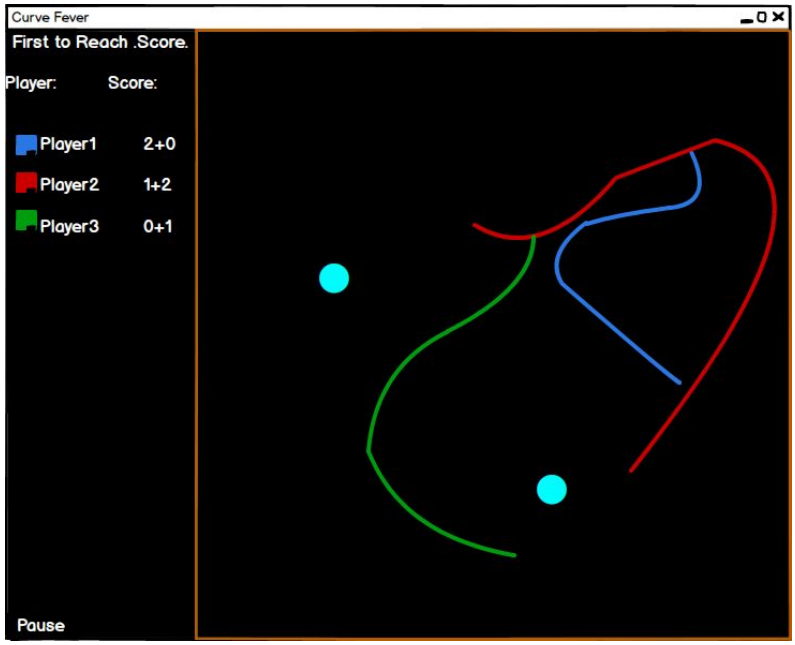


Figure 12. Gameplay

The "Gameplay" page. 3 players are playing the game. There are two power ups in the screen and, according to scores, it is understood that player 1 has died firstly, then player 3. Therefore player2 wins the round. On the top left part of the figure .Score. means for example for 3 players, it is "First to Reach 20". It can change according to number of players. If players die, their patterns stay on the map, but they stop playing for that round.

4.7.10.Pause

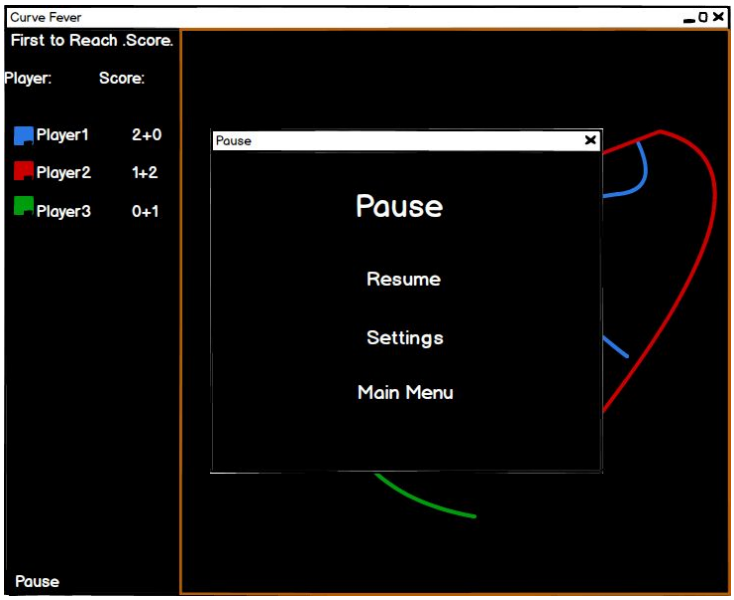


Figure 13. Pause

The "Pause" page. Here the players can resume the game or go to settings or they can stop playing and return to main menu.

4.7.11.End Round

Figure 14. End Round

The "End Round" page. At the end of each round, end round page comes up which gives updated informations about the game.

4.7.12.End Game

Figure 15. End Game

The “End Game” page. If the game is finished (if one of the players reached .Score. or if they want to return the main menu) “End Game” page comes. Here players can see the final update on their game.

5.Glossary & References

Curve Fever Wikia | Fandom powered by Wikia

http://curvefever.wikia.com/wiki/Curve_Fever_Wikia