



Bilkent University  
Department of Computer Engineering

# Object-Oriented Software Engineering Project

*CS319 Project: River Adventure*

## Analysis Report

Nazlı Özge Uçan, Meder Kutbidin Uulu, Aras Heper, Hande Teke

Course Instructor: Uğur Doğrusöz

TA: İstemi Bahçeci

# Table of Contents

1. Introduction.....	4
2. Proposed System.....	5
2.1. Overview.....	5
2.1.1. Gameplay and Control.....	5
2.1.2. Characters.....	5
2.1.3. Obstacles.....	8
2.1.4. Items.....	9
2.1.4.1. Traps.....	10
2.1.4.2. Exclusive Boosts.....	10
2.1.4.3. Character Specified Boosts.....	15
2.1.4.4. Rare Boosts.....	16
2.1.5. Collectables.....	17
2.1.6. Store.....	18
2.1.7. Scoring.....	19
2.2. Functional Requirements:.....	19
2.2.1. Playing Game.....	19
2.2.2. Pausing the Game.....	19
2.2.3. Viewing Store and Purchasing Items.....	20
2.2.4. Setting Changes.....	20
2.2.5. Viewing Help.....	20
2.2.6. Viewing Credits.....	20
2.2.7. Playing Sound.....	21
2.3. Non-Functional Requirements.....	21
2.3.1. Usability.....	21
2.3.2. Reliability.....	21
2.3.3. Performance.....	21
2.3.4. Supportability.....	21
2.4. Constraints.....	21
2.5. Scenarios.....	22
2.6. Use Cases.....	33
2.6.1. Case Overview.....	33
2.6.2. Case Descriptions.....	33
2.7. Object Model.....	36
2.7.2. Class Diagram.....	36
2.8. Dynamic Models.....	40
2.8.1. State Chart Diagrams.....	40
2.8.2. Activity Diagrams.....	41
2.8.3. Sequence Diagrams.....	43
2.9. User Interface.....	52
2.9.1. Main Menu Screen.....	52
2.9.2. Play Screen.....	52
2.9.3. Pause Screen.....	54
2.9.4. End Game Screen.....	55

2.9.5. Score Screen.....	56
2.9.6. Settings Screen.....	57
2.9.7. Help Screen.....	58
2.9.8. Credits Screen.....	59
3. Conclusion.....	61

# 1.Introduction

The game River Adventure is a first-person game where the player is a character swimming along the river using right and left arrow keys on the keyboard. The aim of the River Adventure is to swim as much as you can while collecting the in-game money currency, coins. When the health of the character reaches zero, the game ends and the score of the player is calculated based on the time he plays and the coins he collected. The player can go into the store to unlock many items and/or characters to improve the game standard of his own. After the game finishes, the player can see his score on the HighScore table if he is worthy enough to enter the list.

In the beginning of the game, the player has a character of a duck. Duck is the most basic character and doesn't have any special abilities. In the store there are more character options for the player that he can unlock with coins. Those characters, from the lowest price to the highest, are the Beaver, the Deer, the Crocodile and the Hippopotamus. The Beaver is the first character that can be unlocked with lowest amount of coins. The Beaver has immunity to logs up to three times; if it clashes into logs it won't take any damage up to three times. After that logs would harm him as usual. The Deer is another character that can be unlocked with coins. If the player is playing with the Deer, during the game a box would come out randomly named "the Blessing of the Forest Goddess" (BFG for short). BFG is a surprise box that will give the player surprise gifts from the store. The Crocodile is the third character. If the player is playing with the Crocodile, there will be birds that he can collect while swimming which would give him bonus health. The last one is the Hippopotamus that can be bought with the highest amount of coins. While playing with the Hippopotamus, another special boost called "berserk" would come up which gives the character immunity to all the obstacles for some limited time.

Throughout the the gameplay many obstacles and items show up. Each obstacle reduces some amount of the character's health if the character clashes to it. Obstacles are categorised based on their damage. From lowest to the highest they are logs, stones, chemical hazards and nuclear bomb where the clashing into a nuclear bomb reduces the character's whole health. Along with obstacles, which show up randomly in the river, there are also items that can be categorised into traps, pricey boosts, character specified boosts and in-game boosts. Traps, like obstacles are hazardous for the character. However, they don't reduce its health. Whereas, they either speed the character up or minimise it as swimming would be harder when the character is fast and/or small. Traps have a time limit; until a specified amount of time passes, the character is trapped and has to play with the trap that it got into. Pricey boosts are the ones that helps the character to swim and collect coins easier. They are also time-limited. They

can be categorised into shields, minimise, slow-down power, invincibility and health packs. However, in order to run across to them during the game, the player should unlock them in the store using coins. After a they are unlocked they will show up during the game. Also, the player can upgrade them with more coins and boosts would be more powerful in terms of durability and strength. Another kind of item is character specified boosts. As described earlier, character specified boosts are only limited for the specified character and shows up like other items. For the last category of items, there are rare boosts. Those boosts are BONUS and Random Box. They don't come much often than other boosts, they are rare. When the character collects BONUS, all the obstacles in the river disappears and there will be coins all the way, until the time limit reaches and the game turns into regular. Random Box is an item that the player cannot see what it is until he collects it. There might some kind of boosts or some traps in that box!

## 2. Proposed System

### 2.1. Overview

#### 2.1.1. Gameplay and Control

The River Adventure is played using left and right arrow keys on the keyboard. The character is swimming continually on the river and the player tries to avoid the obstacles and collect the boosts and coins. While the character is swimming the speed of it increases. So the difficulty of the gameplay increases since it becomes harder to swim without clashing into obstacles.

#### 2.1.2. Characters

There are several characters in the game that can be unlocked from the store with coins.

- **The Duck**

The Duck is the initial character that the player starts playing. It doesn't have any special abilities.

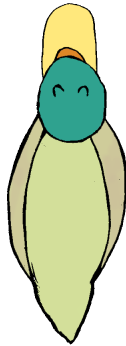


Figure 1. the Duck

- **The Beaver**

The Beaver can be unlocked from the store with coins. If the player has collected enough coins to unlock the Beaver, he can buy it, equip it and play as the Beaver in the game. The Beaver has a special ability to hit up to three logs as obstacles, without taking damage. When the player chooses to play as the Beaver, there will be a bar in the top of the screen that shows how much more logs that it can hit. After that bar is full, the Beaver would take damage from the logs as regular characters.



Figure 2. the Beaver

- **The Deer**

The Deer can be unlocked from the store with coins. If the player has collected enough coins to unlock the Deer, he can buy it, equip it and play as the Deer in the game. While playing with the Deer, there will be a box named “the Blessing of the Forest Goddess” will show up. The player can collect those boxes. Inside of them,

there are surprise gifts as coins or boosts.

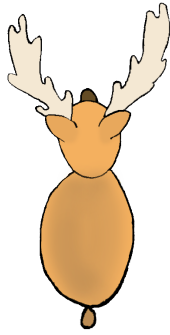


Figure 3. the Deer

- **The Crocodile**

The Crocodile can be unlocked from the store with coins. If the player has collected enough coins to unlock the Crocodile, he can buy it, equip it and play as the Crocodile in the game. The Crocodile has a special ability to heal itself when it collects birds. While playing as a crocodile, birds will show up as other boosts to heal the Crocodile if he is low on health. Immediately after collecting a bird, the health of the Crocodile fills as the amount of health as the bird gives.



Figure 4. the Crocodile

- **The Hippopotamus**

The Hippopotamus can be unlocked from the store with coins. If the player has collected enough coins to unlock the Hippopotamus, he can buy it, equip it and play as the Hippopotamus in the game. The Hippopotamus is the most costly character that the player can unlock and play with. While playing as the Hippopotamus, a special kind of boost for it shows up in the river. It is called “berserk”. If the player collects berserk, the Hippopotamus becomes invincible for some amount of time; it will be immune to damages.

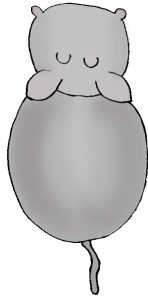


Figure 5. the Hippopotamus

### 2.1.3. Obstacles

While the player is swimming along the river, there will be many obstacles that he should avoid clashing into. Each obstacle reduces the character's health and after some clashes, the character loses all of its health and the game finishes. Obstacles can be list according to their damage as follows:

- **Logs**

Logs are the most basic obstacle that the player can clash into. They reduce some amount of the character's health each time the character clashes it.

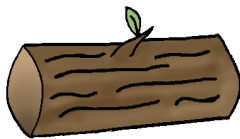


Figure 6. Logs

- **Stones**

Stones are another kind of obstacles that the player can clash into. They reduce the health of the player for each clash more than Logs but less than a Chemical Hazard.





Figure 7. Stones

- **Chemical Hazard**

Chemical Hazard is another obstacle that reduces the health of the player in each clash. However, the amount of health that is lost is significant and more than Stones and Logs.



Figure 8. Chemical Hazard

- **Nuclear Bomb**

Nuclear Bomb is another obstacle that reduces the health of the player when he hits it. They are very rare obstacles. However, when the player hits a Nuclear Bomb, it reduces all of the character's health and finishes the game.

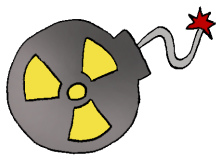


Figure 9. Nuclear Bomb

## 2.1.4. Items

There are many items that the player can collect while swimming. Some of those items are hazardous for the character whereas other are boosts that beneficial for the gameplay. All of the items are time limited.

### 2.1.4.1. Traps

During the game there are some traps that the player can collect. As obstacles, traps are also hazardous for the character. However, instead of reducing health, a trap makes the harder to play for the player. The traps are:

- **Accelerate**

When the player collects Accelerate, the character gathers speed. So, the player's risk of clashing into an obstacle gets higher. After some time limit the character goes back to its normal speed, if it hasn't lost all of its health and finishes the game.

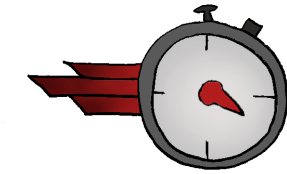


Figure 10. Accelerate

- **Maximise**

When the player collects Magnify, the character gets bigger in size. So, the player's risk of clashing into an obstacle gets higher. After some time limit the character goes back into its normal size, if it hasn't lost all of his health and finishes the game.



Figure 11. Maximise

#### 2.1.4.2. Exclusive Boosts

There are many boosts that the player can unlock and then upgrade from the store using coins. They are as follows:

- **Shield**

The player can unlock the shield boost from the store with coins. After he unlocks the shield, during the game he can run into shield boost and if he collects it, the character gains the shield. If the character has the shield, it will be immune to obstacles with regard to the level of the shield. Shield has three levels of upgrade. When the player first unlocks the shield from the store shield level-1 is unlocked. After that, if player wants to upgrade the shield power, he can upgrade it with more coin. Level-2 shield has more immunity to obstacles than level-1 shield and level-3

shield has more immunity to obstacles than level-2 shield.



Figure 12. Shield level-1



Figure 13. Shield level-2

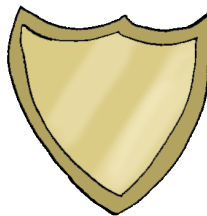


Figure 14. Shield level-3

- **Minimisation Power**

The player can unlock the minimisation power from the store with coins. After he unlocks the minimisation power, during the game he can run into the minimisation power boost and if he collects it, the character gets smaller in size. Minimisation power is time limited. Until the end of the time limit the character becomes smaller and will be able to avoid obstacles easier. Minimisation power has three levels of upgrade. When the player first unlocks the boost, it is available in the game as level-1 minimisation power. Level-2 minimisation power has longer durability than the level-1 and level-3 minimisation power has even more durability than the level-2.

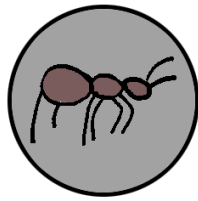


Figure 15. Minimisation power level-1

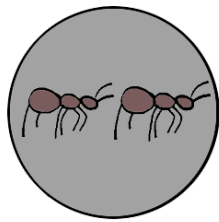


Figure 16. Minimisation power level-2

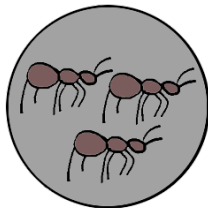


Figure 17. Minimisation power level-3

- **Deceleration**

The player can unlock the deceleration from the store with coins. After he unlocks deceleration, the character can run into and collect the boost during the game. When the character collects the boost, it becomes slower and will be able to avoid obstacles easier. Deceleration is time limited. Until the end of the time limit, the character becomes slower. Deceleration has three levels of upgrade. When the player first unlocks deceleration, level-1 is unlocked. After that, if player wants to upgrade the power, he can upgrade it with more coin. Level-2 deceleration has longer time limit than level-1 and level-3 has more time limit than level-2 deceleration.



Figure 18. Deceleration level-1



Figure 19. Deceleration level-2



Figure 20. Deceleration level-3

- **Invincibility**

The player can unlock invincibility from the store with coins. After he unlocks invincibility, during the game he can run into the invincibility boost and if he collects it, the character becomes immune to obstacles. Invincibility also has a time limit. After the character becomes invincible it can clash as many obstacles as it can until the time limit ends. After the time limit ends, the character takes damage from obstacles as usual. Invincibility has three levels of upgrade. When the player first unlocks invincibility, level-1 is unlocked. After that, if player wants to upgrade the power, he can upgrade it with more coin. Level-2 invincibility has longer time limit than level-1 and level-3 has longer time limit than level-2 invincibility.

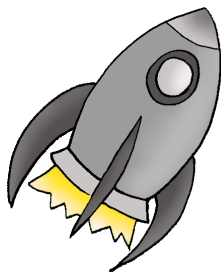


Figure 21. Invincibility level-1

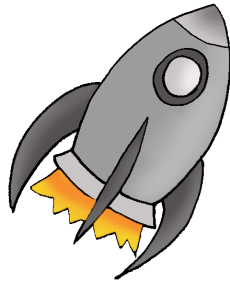


Figure 22. Invincibility level-2

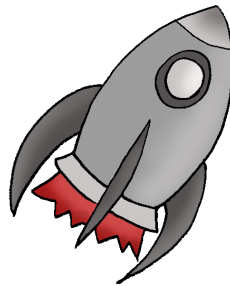


Figure 23. Invincibility level-3

- **Health Pack**

The player can unlock health pack boost from the store with coins. After he unlocks the health pack, during the game he can run into the health pack boosts and if he collects it, the health of the character increases. Health pack boost has three levels of upgrade. When the player first unlocks health pack, level-1 is unlocked. After that, if player wants to upgrade the boost, he can upgrade it with more coin. Level-2 health pack increases health more than level-1 and level-3 health pack increases health more than level-2.



Figure 24. Health Pack level-1



Figure 25 Health Pack level-2



Figure 26. Health Pack level-3

#### 2.1.4.3. Character Specified Boosts

There are some character specified boosts as well as the ones that sold in the store. Character specified boosts are not available in the store. When the player unlocks a character from the store with coins, character specified boosts of the specific character becomes available during the game when the player is playing with that specific character. They are as follows:

- **The Blessing of the Forest Goddess**

The Blessing of the Forest Goddess is a box that shows up during the game when the player is playing as the Deer. BFG is specific to the Deer and only shows up while playing with the Deer. When the character collects the box, it will give the player some random gift. During the game the Deer may run into many boxes and receives gifts from each of them.

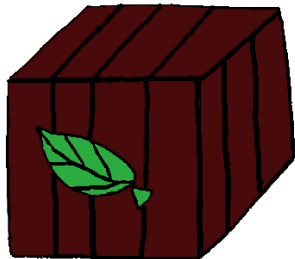


Figure 27. The Blessing of the Forest Goddess

- **Birds of the Crocodile**

Birds of the Crocodile is a boost that can only be seen while playing as the Crocodile. If the player is playing as the Crocodile, Birds of the Crocodile can show up and when the player collects them, the health of the character increases accordingly. During the game, the Crocodile may run into many birds and increase its health.



Figure 28. Birds of Crocodile

- **Berserk**

Berserk is the boost that only shows up when the player is playing as the Hippopotamus. If the player is playing as the Hippopotamus, Berserk can show up and when the player collects it, the Hippopotamus becomes immune to damage. Berserk is time limited. So when the time limit finishes, the Hippopotamus continues to swim regularly. During the game, the Hippopotamus may run into many Berserk boosts.

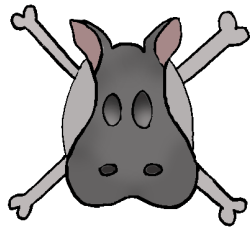


Figure 29. Berserk

#### 2.1.4.4. Rare Boosts

Rare boosts only show up during the game. They are not character-dependent and they cannot be bought from the store or upgraded. The possibility of running into one is rare as the name suggests. They are as follows:

- **BONUS**

BONUS is a boost that when collected, the obstacles along the river disappears and there will be coins along the way. BONUS is time limited. After the time limit ends, obstacles will reappear.





Figure 30. BONUS

- **Random Box**

Random Box is a box that rarely shows up during the game. When the character collects the box it turns into a random boost or a trap. The player cannot see what is in the box before he collects it.

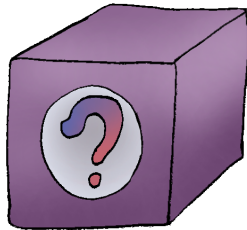


Figure 31. Random Box

### 2.1.5. Collectables

Collectables give player the in-game money currency, coin. The player can see how much coin he has earned and he should collect coins in order to unlock characters and boosts as well as upgrading them. There are two kinds of collectables in the game that give coin.

- **Coin**

Coins are most basic collectables in the game. A coin is worth one unit. There will be many coins to collect during the game, it will show up frequently.

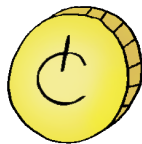


Figure 32. Coin

- **Coin Bag**

Coin Bag shows up less frequently than regular coins. However, when the player collects it, coin bag will give more coins than a basic coin.



Figure 33. Coin Bag

#### 2.1.6. Store

The store is the place where the player can unlock new characters, new boosts and upgrade them. It can be accessible through the main screen. In the store there will be a list of characters that can be unlocked and the player can see the amount of coin he has. If the player wants to unlock a character, he can pay the price and unlock the character. After he unlocks the character he can equip as that character and play. The list will contain all the characters which are previously unlocked and/or still locked ones. Along with characters there will be pricey/exclusive boosts that the player can unlock using coins. After a boost is unlocked, it will show up during the next game automatically as level-1. If the player wants to upgrade the boost he can upgrade it with more coins. In the store, there is also this boost list that will show all the boosts, that are unlocked and locked along with their levels, if unlocked.

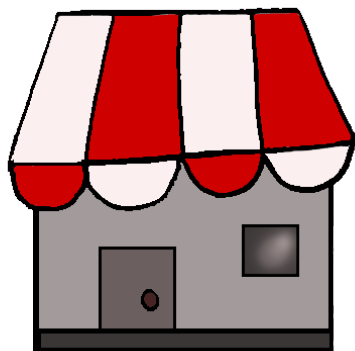


Figure 34. The Store

### 2.1.7. Scoring

The score of the game is calculated according to the time that the character survives and the amount of coin it collects. After the game finishes the player can see his score along with the HighScore table. Also, if he breaks a record and he is worthy enough to enter the table, he will also see his score on the table.

## 2.2. Functional Requirements:

### 2.2.1. Playing Game

- The Player should be able to move right or left within the river.
- The System should place Obstacles to certain locations in the screen
- The System should place Regular Boosts to certain locations in the screen
- The System should place Character Boosts, which appears only for specific characters, to certain locations in the screen
- The System must display the time left on the screen
- The System must display the health left on the screen
- The System must display the Boosts when the player collects one in the game
- The System should update the health of the player when a collision occurs between the Character and an Obstacle.
- The System should create an animation when a Character's health depletes.
- The System should increase the health when the player collects a health bonus
- The System should update the time each second.
- The System should notify the player about the change in health points/ hit points after each collision.
- The System should apply the respective bonus when a boost is collected.
- The System must notify the player about the current boost.
- The System must notify the player about the total points and coins earned at the end of the game.
- The System should display current speed level.
- The System should change generation and displacement speed of objects according to speed modifier.

### 2.2.2 Pausing the Game

- The Player should be able to pause the game while playing.
- The System should pause the game, stop all moving game objects and time when relative button is clicked.

- When the game is paused the System must display pause menu.
- The Player should be able to continue to play the game by pressing relative button in the Pause Menu.

### 2.2.3 Viewing Store and Purchasing Items

- The Player must be able to view Store.
- The System should display all Characters and Exclusive Boosts that are present in the game.
- The System should lock the items the Player cannot afford.
- The System should highlight the items the Player can afford.
- The System should indicate an already purchased item and should not allow it to be purchased again.
- The Player must be able to view details of an available item.
- The Player should be able to select an item to purchase.
- If the Player selects a Character to purchase the System should reduce the amount of coin from the Player.
- The Player should be able to specify Boost amount if he/she wants to purchase a Boost.
- The System should calculate price of selected item.
- The System must notify the Player if the price exceeds the number of Player coins.
- The System should notify the Player when purchase is completed.
- The System should update the Store view after a purchase.

### 2.2.4 Setting Changes

- The System should display the current settings to User.
- The Player must be able to turn the music on or off.
- The Player must be able to alter volume level.
- The System should record and update changed settings.

### 2.2.5 Viewing Help

- The System should display Help page and provide tutorials to the Player
- The System should specify basic concepts of the game
- The System must demonstrate User controls
- The System should explain Store operation.

### 2.2.6 Viewing Credits

- The System should display credits.
- The Player should be able to return to Main Menu by pressing relative button.

### 2.2.7 Playing Sound

- The System system should play music while the Player is on Menus and pages.
- The System should play animation sounds during gameplay when collision occurs.

## 2.3.Non-Functional Requirements

### 2.3.1. Usability

- The System should be User-friendly and easy to use.
- The System should use meaningful names for buttons and icons in order to ease navigation.
- The System should provide access to Main Menu in each screen.
- The Player should be able to access Settings and Help screens from Pause Menu.

### 2.3.2. Reliability

- The System should not loose the Player progress in a power-loss situation
- The System should not loose the Player progress if the System crashes
- The System should not loose the Player progress if the User closes the app abruptly

### 2.3.3. Performance

- The System should respond to the Player input within 1 second
- The System should move the game objects smoothly
- The System should handle animations smoothly

### 2.3.4. Supportability

- The System should be open to development, new features can be added to the game

## 2.4.Constraints

- The System shall be implemented in Java
- Adobe Photoshop shall be used in the design of game graphics

## 2.5. Scenarios

### Scenario 1

**Use Case Name:** OpeningTheGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on Main Menu

**Exit Conditions:**

- Player Max starts the game

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses play button
4. River Adventure starts

### Scenario 2

**Use Case Name:** PlayingTheGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on Main Menu AND
- Player Max presses play button

**Exit Conditions:**

- The health of the character becomes zero

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max hits play button
4. River Adventure starts
5. Player Max moves left or right to avoid obstacles
6. - Each time Player max hits an obstacle, the character's health decrease
7. Player Max tries to gather coins
8. - Each time Player Max collects a coin, the total amount of coins that Max has increase
9. Player Max hits obstacles and reduces the health of the character to zero
10. Game finishes
11. Total point of the Player Max is calculated using the time spent and coins gathered

### Scenario 3

**Use Case Name:** GoingToTheStore

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on Main Menu AND
- Player Max presses the store button

**Exit Conditions:**

- Player Max presses the Return Main Menu button

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the store button
4. The store page opens
5. Player Max presses Return Main Page button

**Scenario 4**

**Use Case Name:** UnlockingTheBeaver

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on the store page AND
- Player Max has enough coins to unlock the Beaver

**Exit Conditions:**

- Player Max presses unlock button of the Beaver

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the store button
4. The store page opens
5. Player Max presses the unlock button of the Beaver
6. The amount of the Beaver is reduced from the total coins
7. The Beaver is unlocked

**Scenario 5**

**Use Case Name:** NotEnoughCoinToUnlockTheCrocodile

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on the store page AND
- Player Max presses unlock button of the Crocodile

**Exit Conditions:**

- The warning message shows up

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the store button

4. The store page opens
5. Player Max presses the unlock button of the Crocodile
6. The warning that says not enough coin to unlock shows up

### **Scenario 6**

**Use Case Name:** EquippingPreviouslyBoughtDeer

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on the store page

**Exit Conditions:**

- Player Max presses the equip button under the Deer

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the store button
4. The store page opens
5. Player Max presses the equip button under the Deer

### **Scenario 7**

**Use Case Name:** UnlockingANewBoost

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on the store page AND
- Player Max has enough coins to unlock a new boost

**Exit Conditions:**

- Boost is unlocked

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the store button
4. The store page opens
5. Player Max presses the unlock button under the specific boost
6. Amount of that boost is reduced from Max's total coins
7. The boost is unlocked

### **Scenario 8**

**Use Case Name:** UpgradingABoost

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on the store page AND



- Player Max has enough coins to upgrade a boost

**Exit Conditions:**

- Boost is upgraded

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the store button
4. The store page opens
5. Player Max presses the upgrade button under the specific boost
6. Amount of the upgrade of that boost is reduced from Max's total coins
7. The boost is upgraded

### **Scenario 9**

**Use Case Name:** CollectingShieldDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button AND
- Player Max collects the shield

**Exit Conditions:**

- Capacity of the shield runs out

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max goes right and left using arrows from the keyboard
6. Player Max sees a shield boost
7. Player Max collects the shield boost by crossing over it
8. The character of Player Max has the shield
9. Player Max hits obstacles as much as the shield can tolerate
10. Shield runs out

### **Scenario 10**

**Use Case Name:** CollectingMinimisationPowerDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button AND
- Player Max collects the minimisation power

**Exit Conditions:**

- Time limit runs out

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up

3. Player Max presses the Start button
4. River Adventure starts
5. Player Max goes right and left using arrows from the keyboard
6. Player Max sees a minimisation power boost
7. Player Max collects the minimisation power boost by crossing over it
8. The character of Player Max has the minimisation power
9. Player Max's character gets smaller in size until the time limit
10. Time limit runs out

### **Scenario 11**

**Use Case Name:** CollectingDecelerationDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button AND
- Player Max collects the deceleration boost

**Exit Conditions:**

- Time limit runs out

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max goes right and left using arrows from the keyboard
6. Player Max sees a deceleration boost
7. Player Max collects the deceleration boost by crossing over it
8. The character of Player Max becomes slower until the time limit
9. Time limit runs out

### **Scenario 12**

**Use Case Name:** CollectingInvincibilityDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button AND
- Player Max collects the invincibility boost

**Exit Conditions:**

- Time limit runs out

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max goes right and left using arrows from the keyboard

6. Player Max sees a invincibility boost
7. Player Max collects the invincibility boost by crossing over it
8. The character of Player Max becomes immune to obstacles until the time limit
9. Time limit runs out

### **Scenario 13**

**Use Case Name:** CollectingHealthPackDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button AND
- Player Max collects the health pack

**Exit Conditions:**

- The health of the character of Player Max increases

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max goes right and left using arrows from the keyboard
6. Player Max sees a health pack
7. Player Max collects the health pack by crossing over it
8. The health of the character of Player Max increases

### **Scenario 14**

**Use Case Name:** CollectingTheBlessingOfTheForestGoddessDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button AND
- Player Max is playing as the Deer

**Exit Conditions:**

- Collecting the Blessing of the Forest Goddess

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the store button
4. Player Max presses the equip button under the Deer
5. Player Max presses the Return Main Page button
6. Player Max presses the Start button
7. River Adventure starts
8. Player Max goes right and left using arrows from the keyboard

9. Player Max sees the Blessing of the Forest Goddess
10. Player Max collects it by crossing over it
11. After the game finishes the gift inside the BFG reveals

### **Scenario 15**

**Use Case Name:** CollectingTheBirdsOfCrocodileDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button AND
- Player Max is playing as the Crocodile

**Exit Conditions:**

- Collecting the Birds of Crocodile

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the store button
4. Player Max presses the equip button under the Crocodile
5. Player Max presses the Return Main Page button
6. Player Max presses the Start button
7. River Adventure starts
8. Player Max goes right and left using arrows from the keyboard
9. Player Max sees the Birds of Crocodile
10. Player Max collects it by crossing over it
11. The health of the character increases

### **Scenario 16**

**Use Case Name:** CollectingTheBerserkDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button AND
- Player Max is playing as the Hippopotamus

**Exit Conditions:**

- Time limit ends

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the store button
4. Player Max presses the equip button under the Hippopotamus
5. Player Max presses the Return Main Page button

6. Player Max presses the Start button
7. River Adventure starts
8. Player Max goes right and left using arrows from the keyboard
9. Player Max sees the Berserk
10. Player Max collects it by crossing over it
11. The Hippopotamus becomes invincible for the time limit
12. Time limit ends and the Hippopotamus becomes regular

### **Scenario 17**

**Use Case Name:** CollectingBONUSDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button

**Exit Conditions:**

- The time limit ends

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max goes right and left using arrows from the keyboard
6. Player Max sees the BONUS
7. Player Max collects the BONUS by crossing over it
8. The obstacles along the way disappears and coins are spread along the river
9. Time limit ends

### **Scenario 18**

**Use Case Name:** CollectingRandomBoxDuringGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button

**Exit Conditions:**

- The unknown boost/trap reveals

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max goes right and left using arrows from the keyboard
6. Player Max sees the Random Box
7. Player Max collects the Random Box by crossing over it
8. The unknown boost/trap reveals

### **Scenario 19**

**Use Case Name:** TrippingIntoAccelerate

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button

**Exit Conditions:**

- Time limit ends

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max goes right and left using arrows from the keyboard
6. Player Max sees the Accelerate
7. Player Max trips into the Accelerate by crossing over it
8. The speed of the character of the Max speeds up
9. Player Max tries to survive until time limit ends
10. Time limit ends

### **Scenario 20**

**Use Case Name:** TrippingIntoMaximise

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the play button

**Exit Conditions:**

- Time limit ends

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max goes right and left using arrows from the keyboard
6. Player Max sees the Maximise
7. Player Max trips into the Maximise by crossing over it
8. The size of the character of the Max increase
9. Player Max tries to survive until time limit ends
10. Time limit ends

### **Scenario 20**

**Use Case Name:** PausingTheGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max is playing the game AND
- Player Max presses the Pause button

**Exit Conditions:**

- Player Max presses the Continue button

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max presses the Pause button
6. Player Max presses the Continue button

**Scenario 21**

**Use Case Name:** ChangeVolumeDuringTheGame

**Actors:** Player Max

**Entry Conditions:**

- Player Max is playing the game AND
- Player Max presses the Pause button

**Exit Conditions:**

- Player Max presses the Save & Exit button

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Start button
4. River Adventure starts
5. Player Max presses the Pause button
6. The settings appear
7. Player Max adjusts the volume
8. Player Max presses the Save & Exit button

**Scenario 22**

**Use Case Name:** ChangeVolumeFromTheMainScreen

**Actors:** Player Max

**Entry Conditions:**

- Player Max presses the Settings button

**Exit Conditions:**

- Player Max presses the Save & Exit button

**Main Flow of Events:**

1. Player Max opens the game
2. River Adventure Main Screen shows up
3. Player Max presses the Settings button
4. The settings appear

5. Player Max adjusts the volume
6. Player Max presses the Save & Exit button

### **Scenario 23**

**Use Case Name:** AccessHelp

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on the Main Screen

**Exit Conditions:**

- Player Max presses the Exit button

**Main Flow of Events:**

1. Player Max opens the game
2. Player Max presses the Help button
3. Help page opens
4. Player Max presses the Exit button

### **Scenario 24**

**Use Case Name:** ViewCredits

**Actors:** Player Max

**Entry Conditions:**

- Player Max is on the Main Screen

**Exit Conditions:**

- Player Max presses the Exit button

**Main Flow of Events:**

1. Player Max opens the game
2. Player Max presses the Credits button
3. Credits page opens
4. Player Max presses the Exit button



## 2.6.Use Cases

### 2.6.1. Case Overview:

**Play:** User plays the game.

**Pause Game:** User pauses the game, system presents the pause menu.

**Change Settings:** User views and can change settings.

**Access Help:** User views helpful information about the game..

**View Store:** User views store options, such as characters and boosts.

**Purchase an Item:** Invoked when user decides to buy a non-character item.

**Upgrade an Item:** Invoked when user decides to upgrade a previously bought non-character item.

**Purchase a Character:** Invoked when user decides to buy a character.

**Equip a Character:** Desired purchased character is equipped and users previous character becomes unequipped.

**View Credits:** User views the information about the systems development.

### 2.6.2. Case Descriptions:

#### Play Game:

**Use Case Name:** Play Game

**Actors:** User

**Entry Conditions:** User is on Main Menu.

**Exit Conditions:** User is on Main Menu.

#### **Main Flow of Events:**

1. Game map is initialized, user character is generated.
2. Objects are generated and pushed down according to time.
3. User can choose to move to left or right.
4. User character collides with an object.
  - a. If it is a coin or boost, user doesn't lose any health point and gains coins/ gains boosts effect.
    - i. jump to event 3.
  - b. If it is a harmful object AND user has shield, shield value is reduced
    - i. jump to event 3.
  - c. If it is a harmful object, user loses health according to object type
    - i. If health points reduce to/below 0 jump to event 5.
    - ii. else jump to event 3.
5. System informs the user about the score.
6. Use case ends.

#### **Alternative Flow of Events:**

1. User clicks on pause. ( at main flow 1, 2, 3, 4 )
  - a. Use case Pause Game is invoked.

### Pause Game:

**Use Case Name:** Pause Game

**Actors:** User

**Entry Conditions:** User is in-Game and extends Play Game.

**Exit Conditions:** User is in-Game. Play Game use case jumps to event 5 OR remains same.

**Main Flow of Events:**

1. System pops Pause Menu, the game stops.
2. User clicks on 'Continue'.
3. Use case ends.

**Alternative Flow of Events:**

1. User clicks on 'Settings'.
  - a. Use case Change Settings is invoked.
2. User clicks on 'end the game'.
  - a. Pause Game use case jumps to event 3.
  - b. Exit Condition "Play Game use case jumps to event 5" is invoked.

### Change Settings:

**Use Case Name:** Change Settings

**Actors:** User

**Entry Conditions:** User is on Main Menu OR user is at Pause Menu.

**Exit Conditions:** User is on Main Menu OR user is at Pause Menu.

**Main Flow of Events:**

1. System displays settings options
2. User clicks on 'Save & Exit'
3. Changes are recorded, and game settings are adjusted accordingly.
4. Use case ends.

**Alternative Flow of Events:**

1. User adjusts the sound volume.( Anytime after main event 1 and before main event 2)
2. User mutes the music.( Anytime after main event 1 and before main event 2)

### Access Help:

**Use Case Name:** Access Help

**Actors:** User

**Entry Conditions:** User is on Main Menu.

**Exit Conditions:** User is on Main Menu.

**Main Flow of Events:**

1. System displays helpful information.
2. User clicks on 'exit'.
3. Use case ends, user returns to Main Menu.

#### View Store:

**Use Case Name:** View Store

**Actors:** User

**Entry Conditions:** User is on Main Menu.

**Exit Conditions:** User is on Main Menu.

**Main Flow of Events:**

1. System displays store UI.
2. User clicks on 'exit'.
3. Use case end, user returns to Main Menu.

#### Purchase an Item:

**Use Case Name:** Purchase an Item

**Actors:** User

**Entry Conditions:** User is on Store Page AND this extends to use case View Store

**Exit Conditions:** User is on Store Page.

**Main Flow of Events:**

1. User clicks on buy icon next to an unbought item of interest.
2. Required amount of coins are withdrawn from users coins.
3. Item becomes bought.
4. Use case ends.

**Alternative Flow of Events:**

1. Use case ends.( If users coins are not sufficient after the event 1.)

#### Upgrade an Item:

**Use Case Name:** Purchase an Item

**Actors:** User

**Entry Conditions:** User is on store page AND store has at least one upgradeable bought item AND this extends to use case View Store.

**Exit Conditions:** User is on Store Page.

**Main Flow of Events:**

1. User clicks on upgrade icon next to an bought item of interest.
2. Required amount of coins are withdrawn from users coins.
3. Item becomes upgraded.
4. Use case ends.

**Alternative Flow of Events:**

1. Use case ends.( If users coins are not sufficient after the event 1.)

#### Purchase a Character:

**Use Case Name:** Purchase a Character

**Actors:** User

**Entry Conditions:** User is on Store Page AND this extends to use case View Store.

**Exit Conditions:** User is on Store Page.

**Main Flow of Events:**

1. User clicks on buy icon next to the character of interest.
2. Required amount of coins are withdrawn from users coins.
3. Character becomes bought/ equipable.
4. Use case ends.

**Alternative Flow of Events:**

1. Use case ends.( If users coins are not sufficient after the event 1.)

Equip a Character:

**Use Case Name:** Equip a Character

**Actors:** User

**Entry Conditions:** User is on Store Page AND store has at least one bought character AND this extends to use case View Store.

**Exit Conditions:** User is on Store Page.

**Main Flow of Events:**

1. User clicks on equip icon next to a bought character.
2. Character becomes equipped, previous character becomes unequipped.
3. Use case ends.

View Credits:

**Use Case Name:** View Credits

**Actors:** User

**Entry Conditions:** User is on Main Menu.

**Exit Conditions:** User is on Main Menu.

**Main Flow of Events:**

1. System displays credits.
2. User hits 'exit'.
3. Use case ends, user returns to Main Menu.



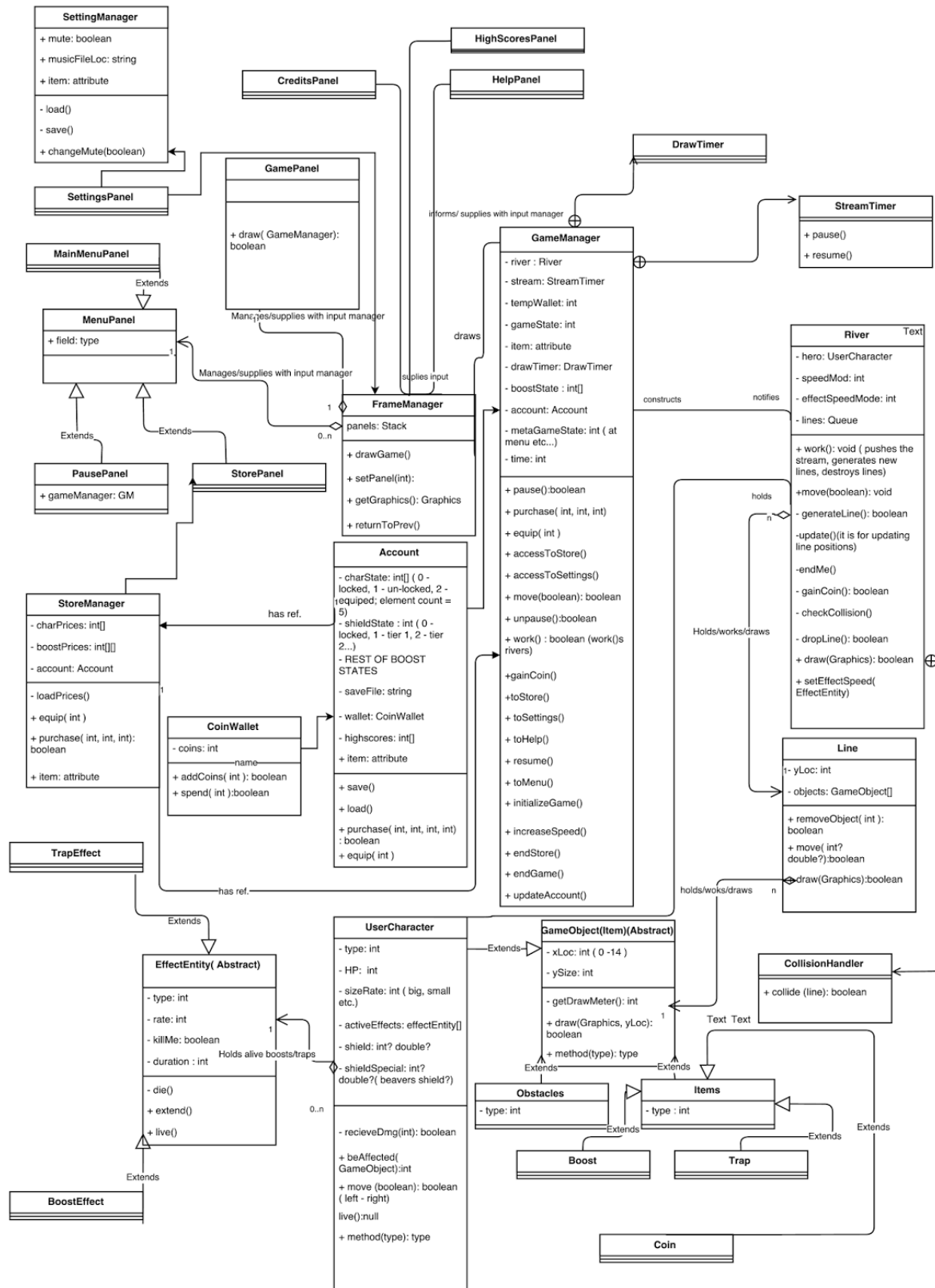


Figure 36. Class Diagram

## **Boundary Classes:**

**FrameManager:** Handles panels, gives them input listeners, or listens instead of them.

**MainMenuPanel:** Class that holds UI components related to Main Menu use case.

**SettingsPanel:** Class that holds UI components related to View Settings use case.

**StorePanel:** Class that holds UI components related to View Store use case.

**GamePanel:** Class that provides canvas to draw in-game objects.

**PausePanel:** Class that holds UI components related to Pause the Game use case.

**CreditsPanel:** Class that holds UI components related to View Credits use case.

**HelpPanel:** Class that holds UI components related to Access Help use case.

**HighScoresPanel:** Class that presents high scores to user via GUI.

**StreamTimer:** Inner-class of GameManager, listens the system clock to start a tick in the game process.

**DrawTimer:** Inner-class of GameManager, listens the system clock to start drawing phase.

## **Control Classes:**

**GameManager:** Holds the overall game state, either waits input from boundary classes to change it, or waits for River. Creates a River object when the Play Game use case starts.

**StoreManager:** Assists store GUI by creating a meaningful communication between Account and StorePanel, decides when to load()/save() from/to text files.

**SettingsManager:** Loads or saves from/to settings text file.

## **Entity Classes:**

**River:** Holds Line's and UserCharacter. Checks collisions, generates Line's, destroys Line's and moves Line's when work() is invoked by GameManager.

**Line:** Holds 15 objects, destroys an object if River requests so.

**GameObject:** Abstract-parent of Item's/UserCharacter/Obstacles's, defines all necessities needed to be an in-game entity.

**Items:** Abstract-parent of Trap's/Boost's. Invokes some effects when collide with UserCharacter.

**Traps:** Invokes harmful effects, category is defined by type:int property.

**Boosts:** Invokes beneficial effects, category is defined by type:int property.

**Obstacle:** Inflicts damage upon User Character upon collision, type( rock, wood,...) is defined by type:int.

**UserCharacter:** The object played by the user. When HP property reaches 0, game ends.

**EffectEntity:** Abstract-parent of BoostEffect's/TrapEffect's. Stores post-collision effects, and their lifetime/magnitude.

**BoostEffect:** Beneficial effects, differ by type:int property in between different boosts.

**TrapEffect:** Harmfull effects, differ by type:int property in between different boosts.

**Account:** Holds the location of the main save .txt file. Load()s or save()s them when requested by GameManager or StoreManager. Holds highscores, purchase/used store items and CoinWallet.

**CoinWallet:** Holds total coins of the account.



## 2.8. Dynamic Models

### 2.8.1. State Chart Diagrams

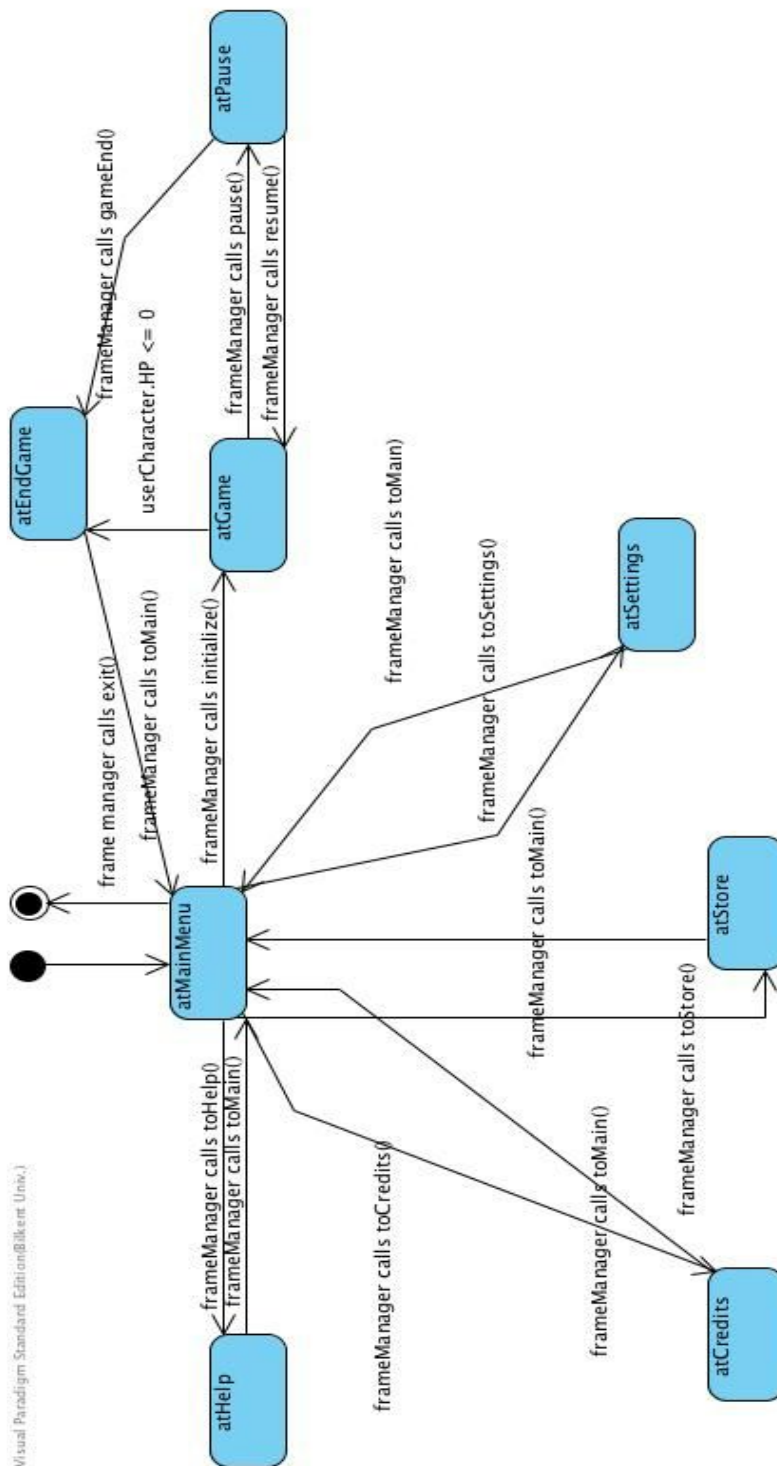


Figure 37. State Chart Diagram

## 2.8.2. Activity Diagrams

### General Activity Diagram:

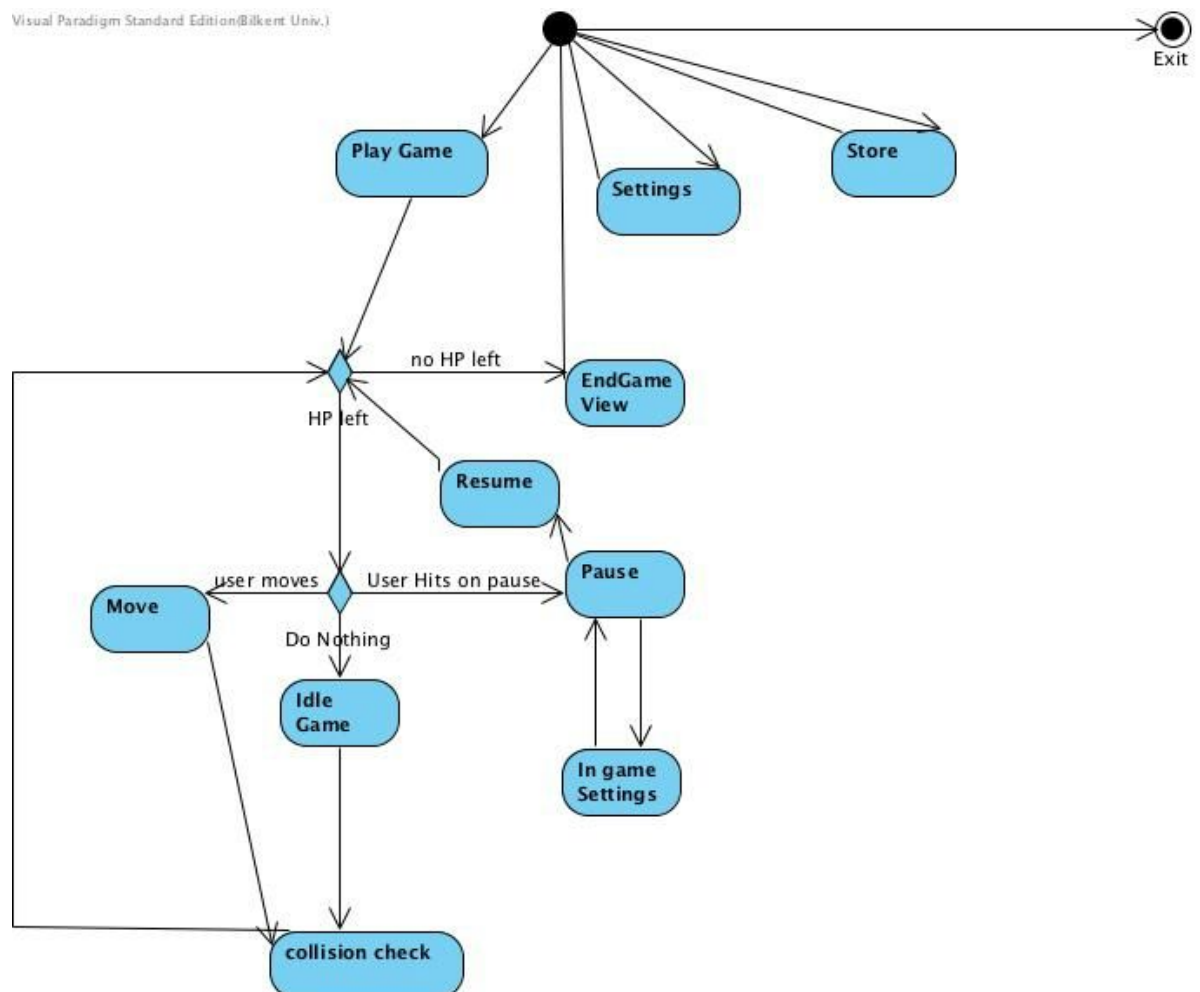


Figure 38. General Activity Diagram

The user is initially at the black circle. He has three options to choose, Play Game, Settings and Store. When he chooses Play Game option, there is a decision maker, which decides to go either Move, Pause, Idle Game (this continues when the user do not move the character) or EndGame View. If the user moves, the system checks if the user character collided with an obstacle or not, if not, then goes back to decision maker, and again makes decision. If the user chooses Pause option, he can go to Settings and from Settings back to Pause. From Pause to Resume, and back to decision maker. From that point, every specific amount of time, it checks whether the user character collided or not, if not then continues to play. If it is collided, then goes to EndGame View.

Collision Activity Diagram:

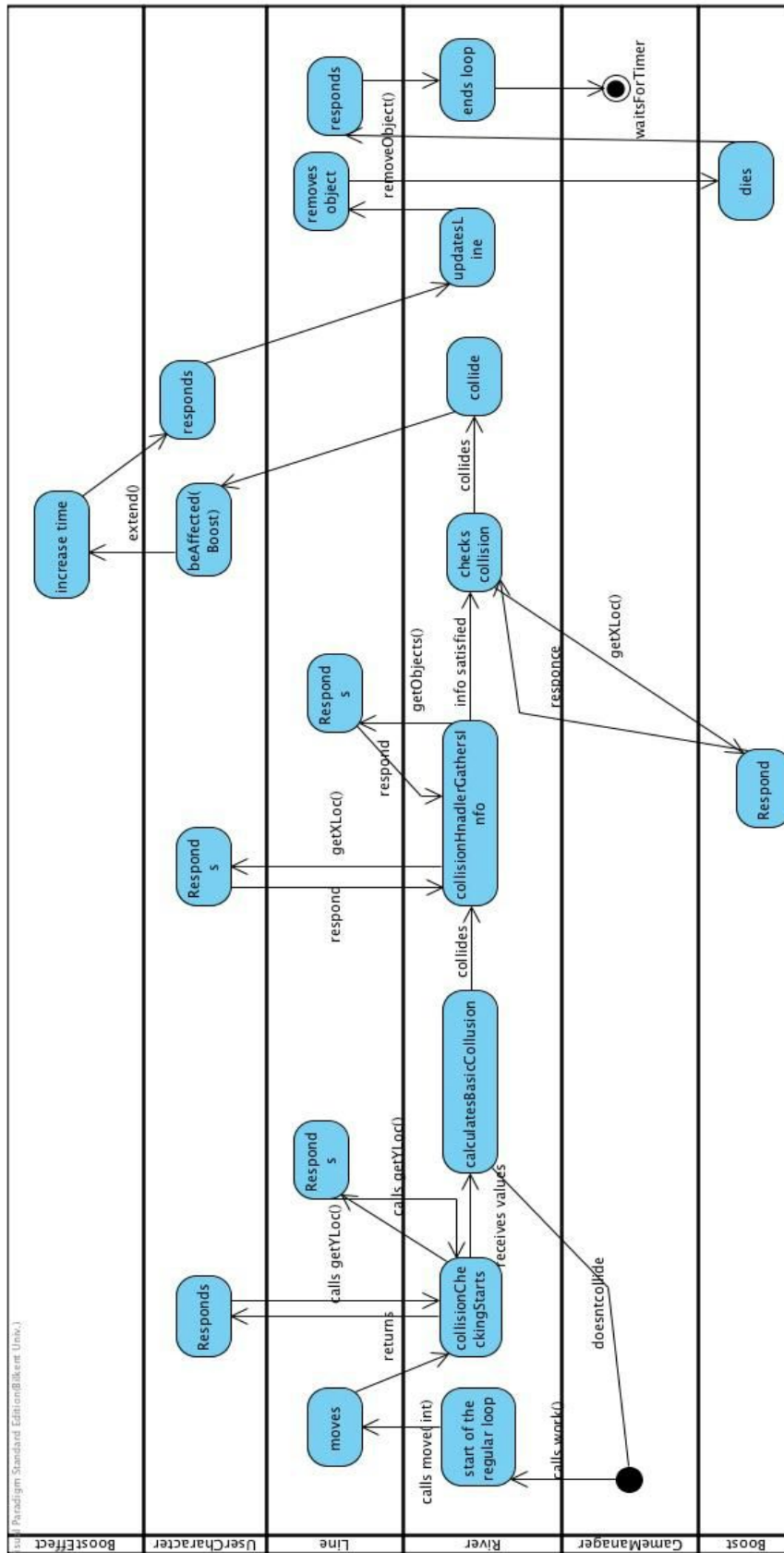


Figure 39. Collision Activity Diagram

Collision activity starts from a tick invoked by StreamTimer, which forces us to GameManager's initial activity before the main gameplay loop. It calls work() of River to continue on, only the check for a single line is being showed. After the action case ends, System waits for another check invoked by GameManager's inner class StreamTimer's tick. Note that at this activity case GameManager is constantly in 'atGame' state.

### 2.8.3. Sequence Diagrams

#### Game Initialization:

This is the game initialisation sequence diagram. Firstly, Meder is at Main Menu, clicks play game button. This button's controller is **FrameManger**, and it evokes *initializeGame()* method of **GameManager** class. Afterwards **River** class is instantiated, and creates **UserCharacter** instance with itself. They are being shown in **GamePanel**, which was created by **FrameManager**. In order to update the game flow, we have another **StreamTimer** class, which handles updates of all objects within a given specific time. Finally, we have a panel to play, a chosen character, lastly **Line** object is being created with *generateLine()* method, which is an instance of **River** class. In **River** class we have an array of **Line** objects that handles with obstacles, boosts and traps, therefore, those objects are being created by **Line** class. So far, all the objects needed are created and ready game to start.

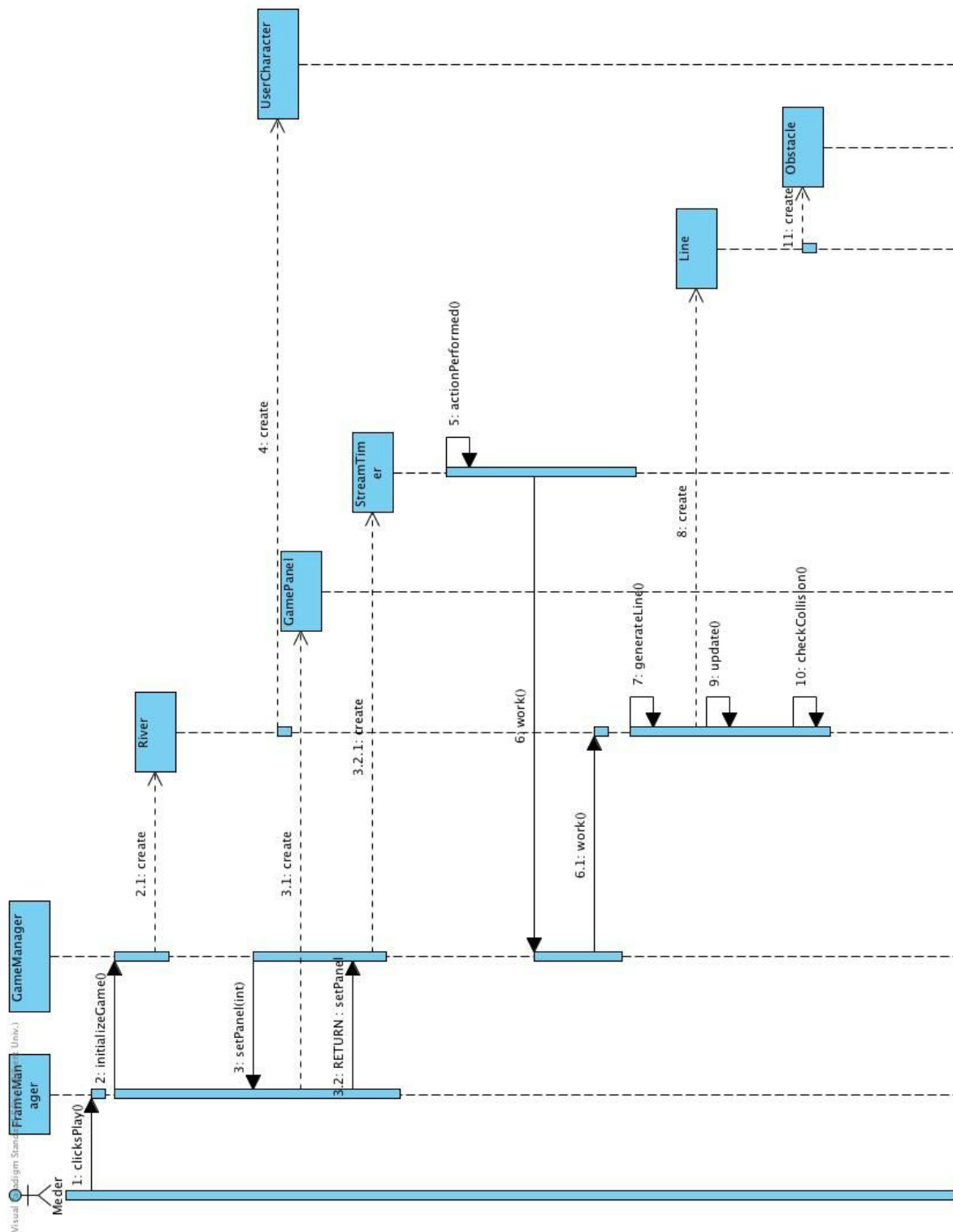


Figure 40. Game Initialization Diagram

## Go to Store and Purchase Character:

In this sequence diagram, Hande is at the Main Menu, and clicks go to Store button. Afterwards, **StorePanel** is appeared. This panel is for showing all the characters and boosts which are available at the market. After choosing a particular character or boost, method is called and the account of the user is updated and that particular purchased object is unlocked.

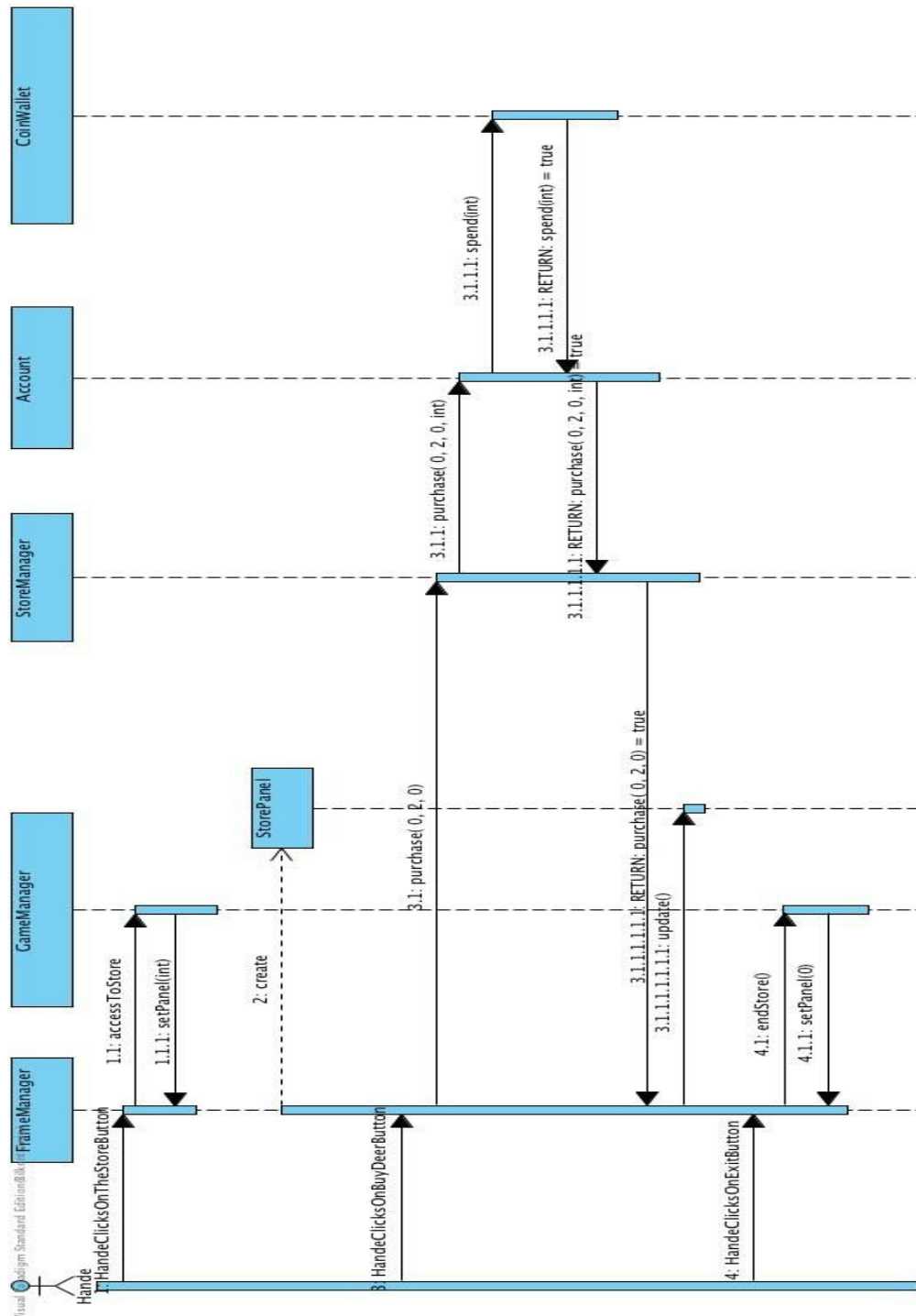


Figure 41. Go to Store and Purchase Character Diagram

### Pause the Game and Change Settings:

Nazli is playing a game, and while she is playing she presses a particular key in order to pause. Then, **FrameManager's** *pause()* method and **GameManager's** *pause()* method is called. **GameManager's** *pause()* method stops all the **TimerStream's** updates. After all necessary pauses are made, there **PausePanel** will appear. In this panel, there are three options, first is to resume the game, second is to back to Main Menu and the third one is to go to Settings Panel. When Setting option is pressed, **FrameManager** is notified and creates new instances of **SettingPanel** and **SettingManager**. In this panel, Nazli mutes the sound of the game, *changeMute(boolean)* is called, and all modifications are saved by *save()* method of **SettingManager**. By using *returnToPrev()* method of **SettingPanel**, we let **FrameManager** to know that all changes are made and **FrameManager** goes back to **PausePanel** with the help of *setActive()* method. Finally, Nazli clicks to *resume()* method of **PausePanel** class, and continues to play after changing sound settings.

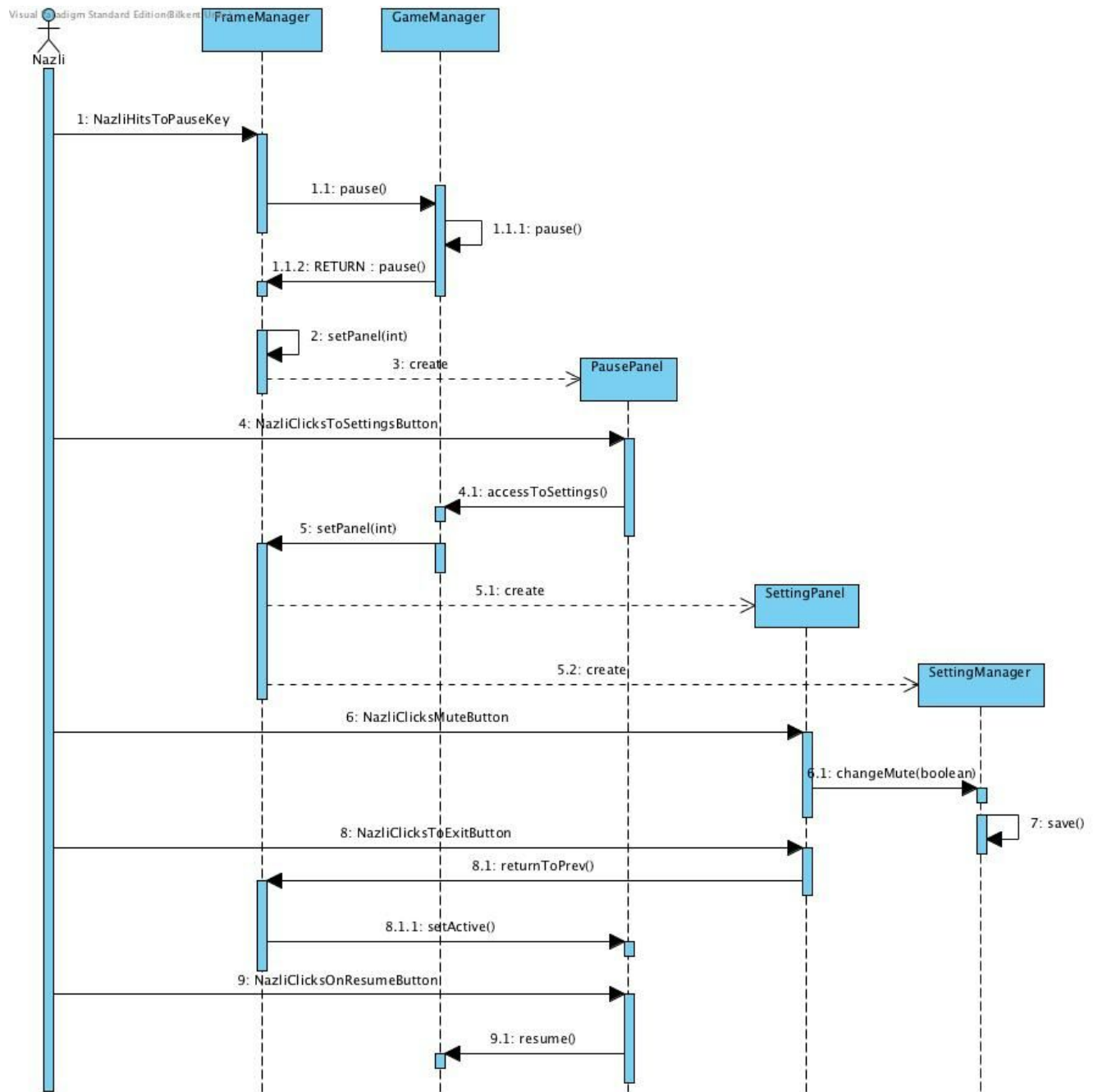


Figure 42. Pause the Game and Change Settings Diagram



## Hit Obstacle and Finish Game:

Aras is playing a game, and clicks to a right or left key in order to shift the position of a user character. A *move()* method of **GameManager** class is called, and afterwards *checkCollision()*, a method of **River** is called, this method checks whether a user character and an obstacle are at the same line. If they are at the same line, then *collide()* method of **River** is called, this is to check whether they overlap each other or not. In order to check, *getObject()* method is called to know which object is being overlapped. After getting the object, a *getXLoc()* method of **Obstacle** class is invoked. So having all these data, *collide()* method decide if they overlap or not, if yes then *beAffectedGameObject()* method of **UserCharacter** is called, it is to ensure the updates of health of the user character because when they collide, the health of the user supposed to decrease. Also, the same method checks whether the character has enough health to continue the game or not, if it is not enough, then *endMe()* method of **River** is evoked, which calls *endGame()*. Finally, the game finishes.

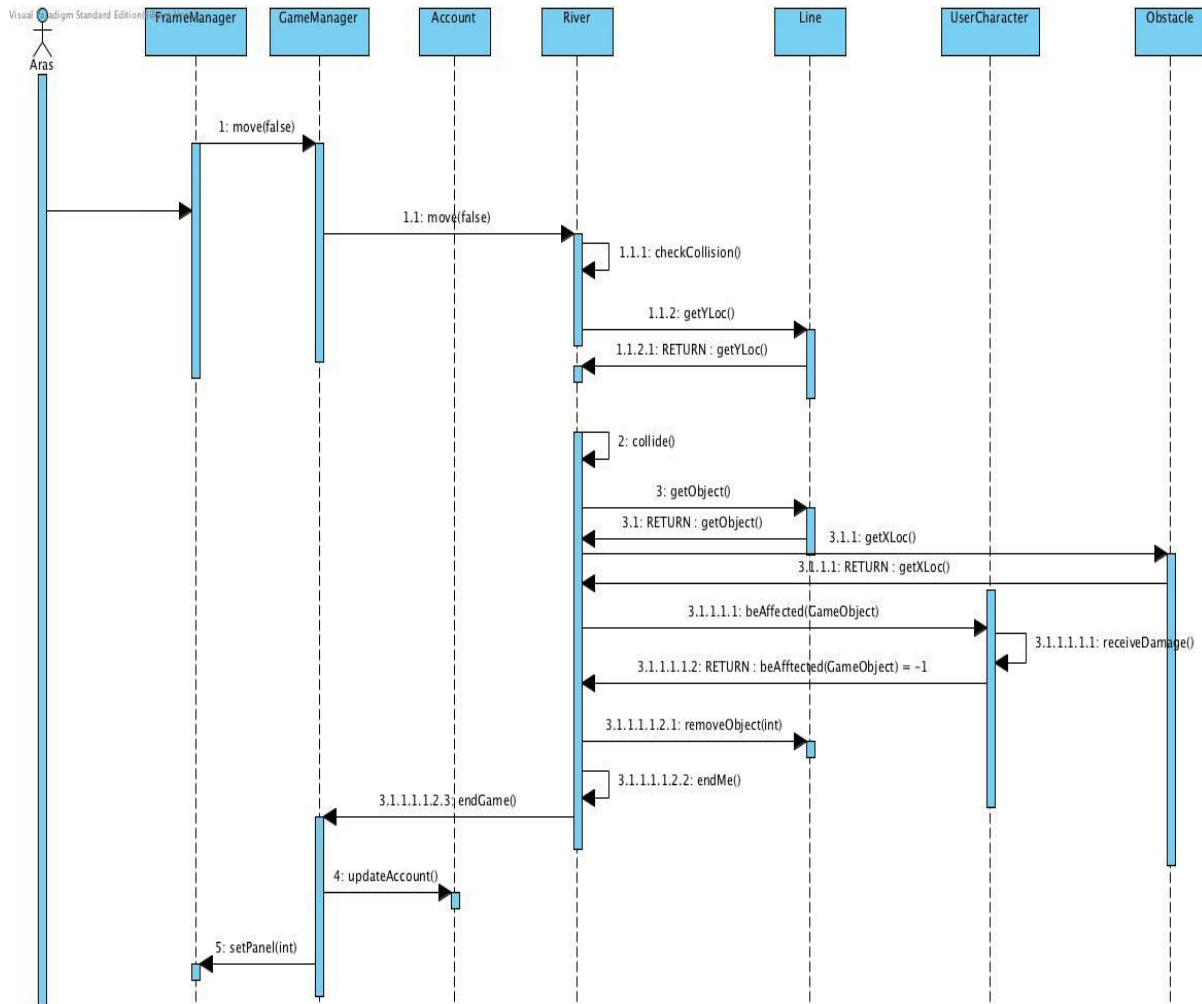


Figure 43. Hit Obstacle and Finish Game Diagram

## Move and Gain Coin:

In this sequence diagram, gaining coins in the game is demonstrated. Player **Aras2** hits the left arrow key so **Frame Manager** then **Game Manager** interacts with the **River**. **River** gets the necessary variables from other classes. From the **Line**, location, and object variables, from the **UserCharacter**, character's location and from the **Coin** class, coin's location are provided as their own specific integer values. Then **River** class compares all these to confirm the collision in *checkCollision()* method. If it is true, collide method runs which means player **Aras2** increases coin amount which is provided by *gainCoin()* method. Then, **River** removes the object that appeared on the current line.

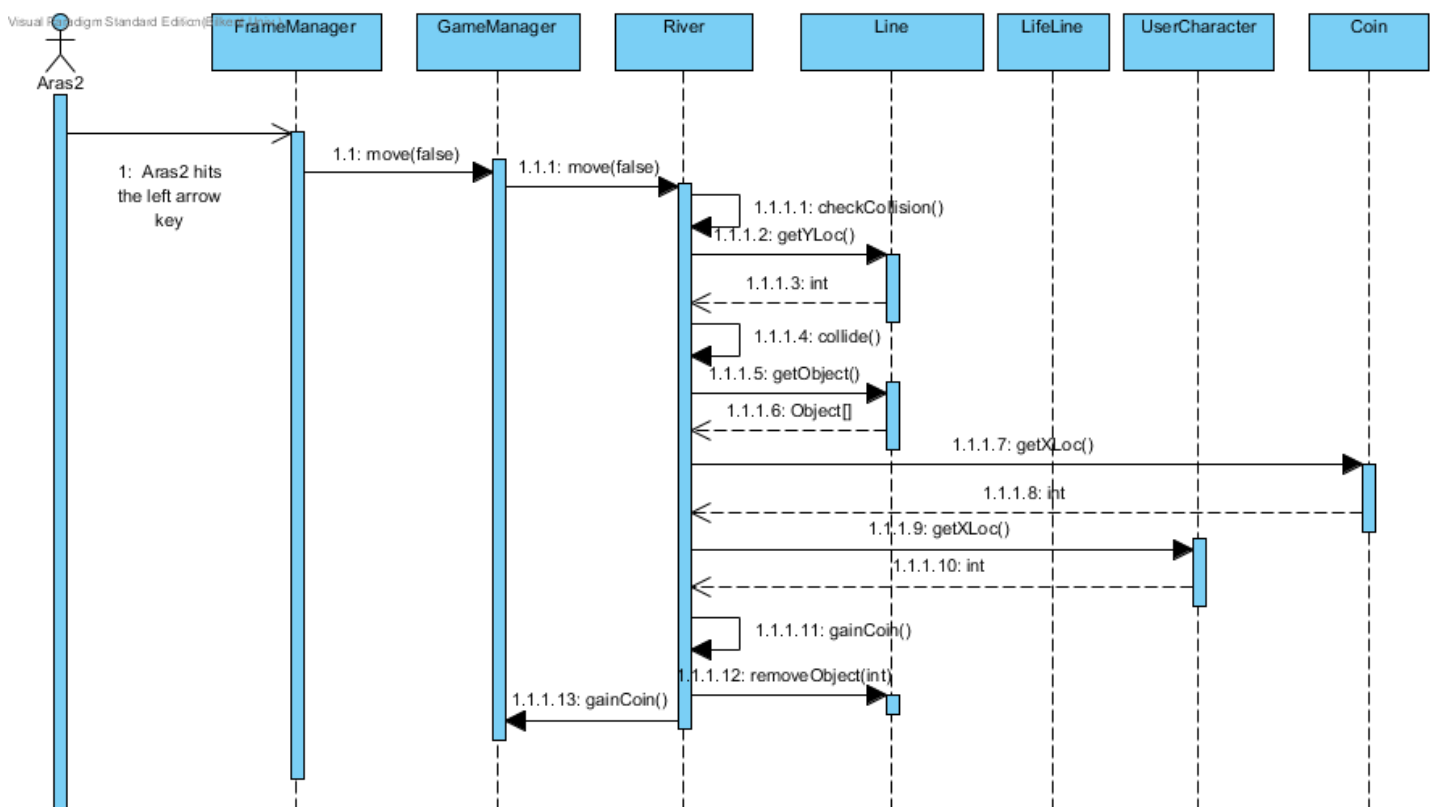


Figure 44. Move and Gain Coin Diagram

### Game Processes and Player Gains a Boost:

In this sequence diagram, getting affected by a boost is demonstrated. Main activities occur in the **River** class because it checks the current situation of the character and boosts and makes the effects happen. **StreamTimer**, **GameManager** runs work methods so the river updates itself. **River** takes the integer variables of locations and the identities of the boost from other classes. **Line** gives the move information as a boolean, location information as integer and the object information. **Boost** class gives the location of the boost. And the **UserCharacter** gives its location value as well. River class runs all the checking methods and when it detects a collision with a boost, *beAffected()* becomes true. As a result, **BoostEffect** be created. Finally, **River** removes the existing boost on the **Line**.

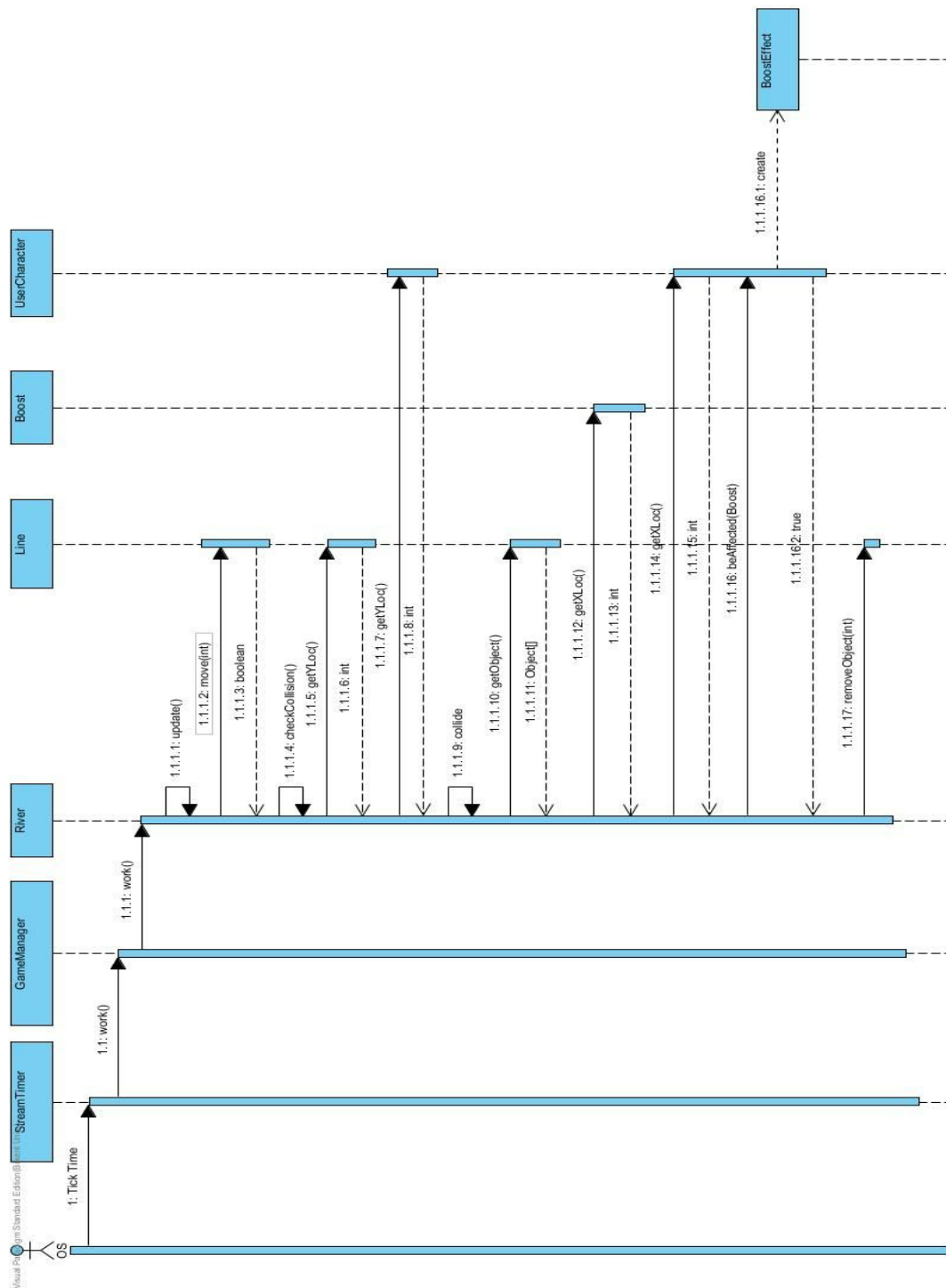


Figure 45. Game Processes and Player Gains a Boost Diagram

## 2.9. User Interface

### 2.6.1. Main Menu Screen

Main Menu screen is the first screen that shows up when the game starts. It has “Play”, “Store”, “Settings”, “Help” and “Credits” buttons under the relative images and when the player clicks a button related screen will show up.

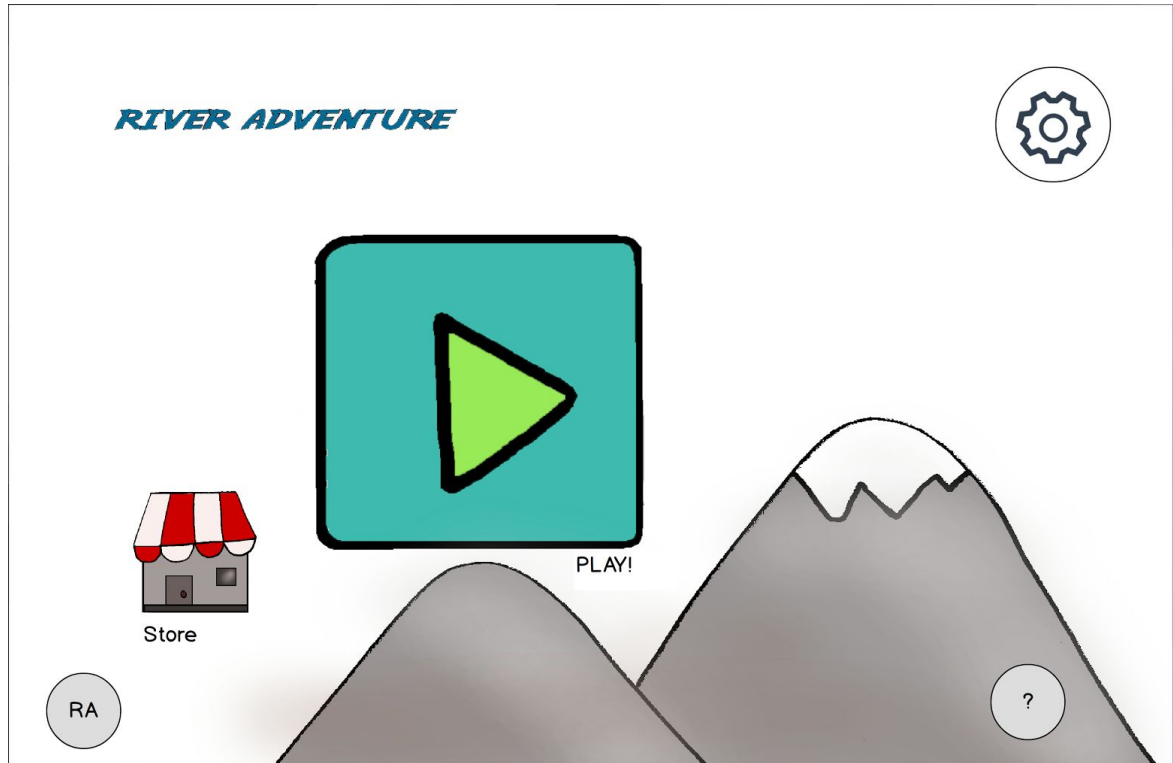


Figure 46. Main Menu Screen

### 2.6.2. Play Screen

Play screen that is showed below is an example gameplay. It shows “the Duck” as the character. The Duck currently has a level-1 shield that is shown at the bottom left of the screen. On the river there are shown two obstacles; stones and a log. Along the river there are coins that the Duck can collect. On the top right of the screen the coins that character collects during the game is shown. Below that the time of the game is shown. On the bottom right corner of the remaining health of the character is shown. During the game player can pause the game by pressing the “Pause” button that is on the top left corner of the screen.

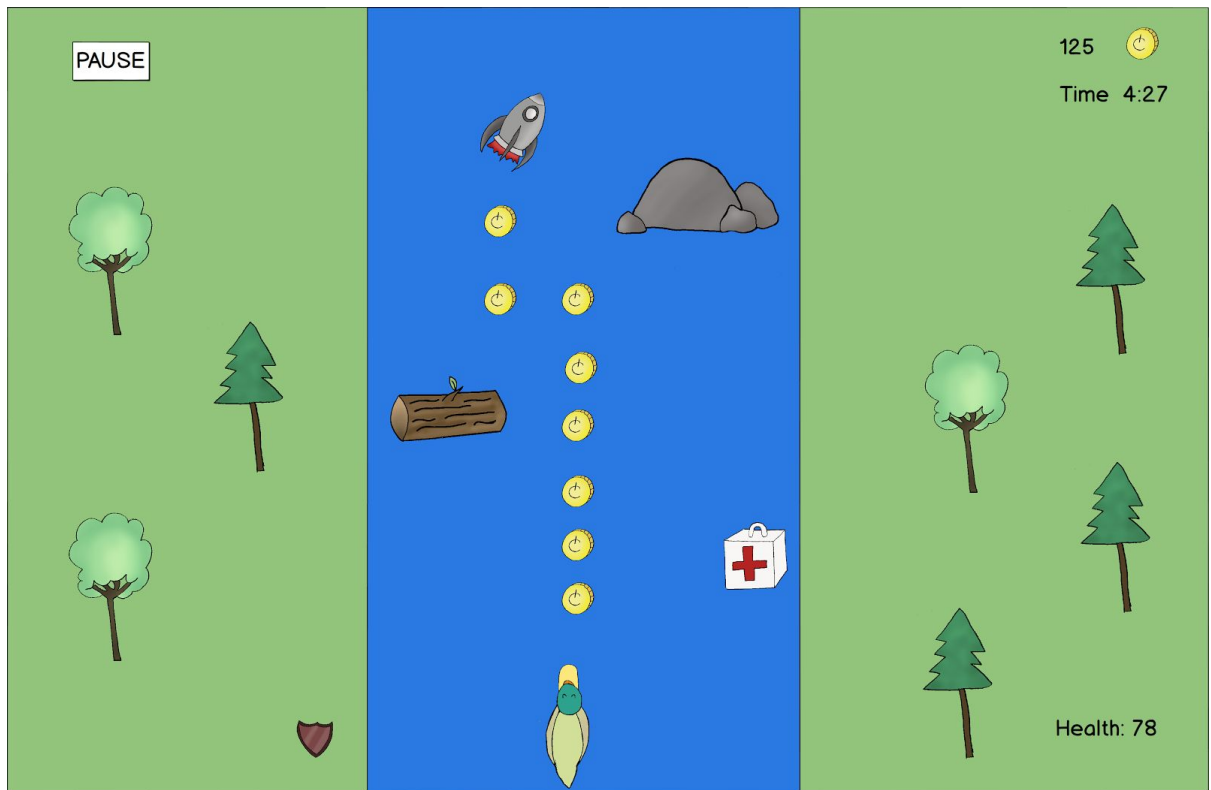


Figure 47. Play Screen

### 2.6.3. Pause Screen

When the game is paused a pop-up pause screen shows up. It has three options. First button is “Resume” game button. Second one is “Exit to Main Screen” button. When the player clicks it, the end game screen shows up. The third button is “Settings” button. When the player clicks it, settings screen shows up.

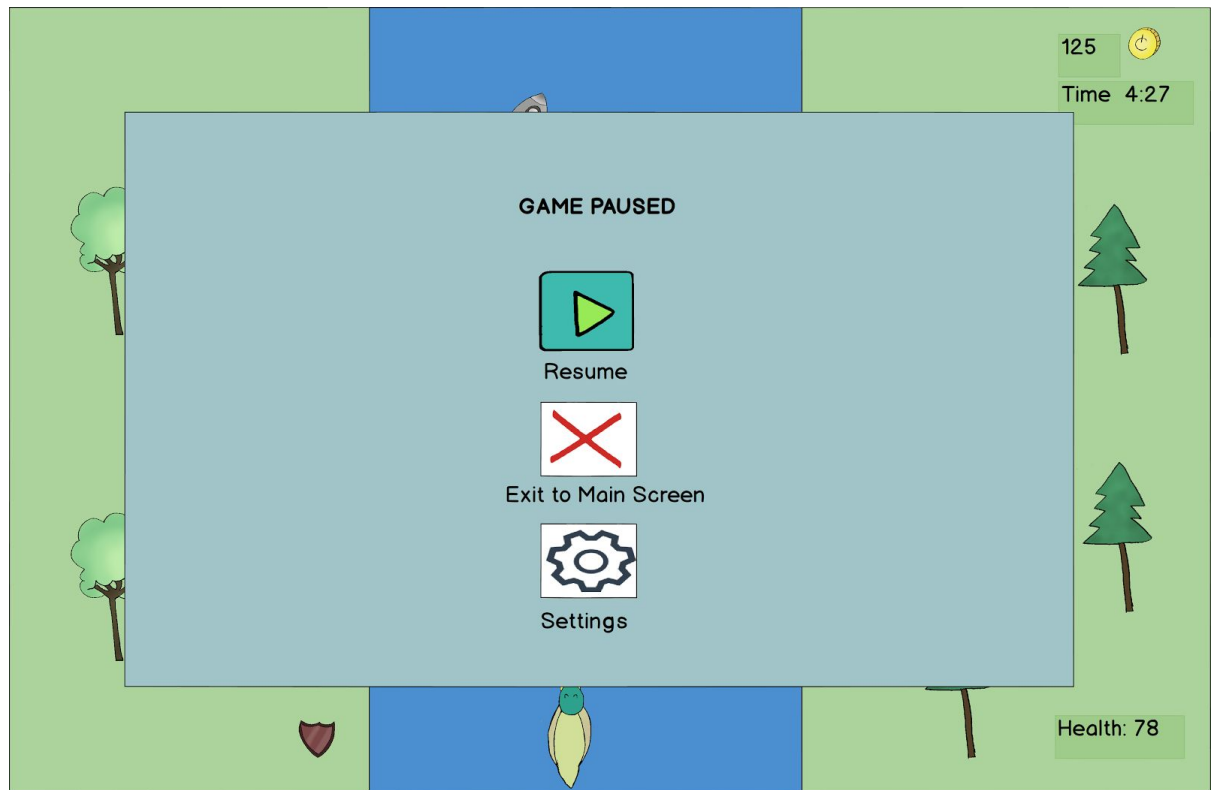


Figure 48. Pause Screen

#### 2.6.4. End Game Screen

When the game finishes, the end game screen shows up. In the end game screen the score of the player is shown along with the highscore table that shows the highest 5 scores.



Figure 49. End Game Screen

### 2.6.5. Store Screen

From the main menu screen, when the player clicks “Store” button store screen shows up. In the store all the characters and boosts that can be bought are shown. On the top right corner the coins that the player has is shown. In the character list, each character has an image, a button below it and a textual description is shown. The character that the player currently plays has “Equipped” button below it that is disabled. Other characters that the player previously bought has the button “Equip”. When the player has unlocked a character and presses “Equip” button, “Equip” button becomes “Equipped” and the previously equipped character update its button as “Equip”. The characters that are unlocked, i.e. haven’t been bought, has the button “Unlock” below them. When the player clicks, the required amount is reduced from the account and the button becomes “Equip”. In the boosts list, each boost is shown with an image, a description and a button below it as characters. Initially every boost is unlocked and has the button “Unlock”. When the player pays the price and unlocks a boost, it is unlocked as level 1 and has the button “Upgrade”. If a boost is already bought and it is level 1, it has the button “Upgrade”. When the player pays the price and clicks the button below it, the image and the textual description updates itself and the button stays as “Upgrade”. Level of each boost is shown in the description. If the level of the boost is 3, i.e. it has the maximum level of upgrade, the button becomes “Upgraded”. In each of the description of a boost, the name, the level, the price of the level if player buys or upgrades and a textual description according to the level of the boost is shown.



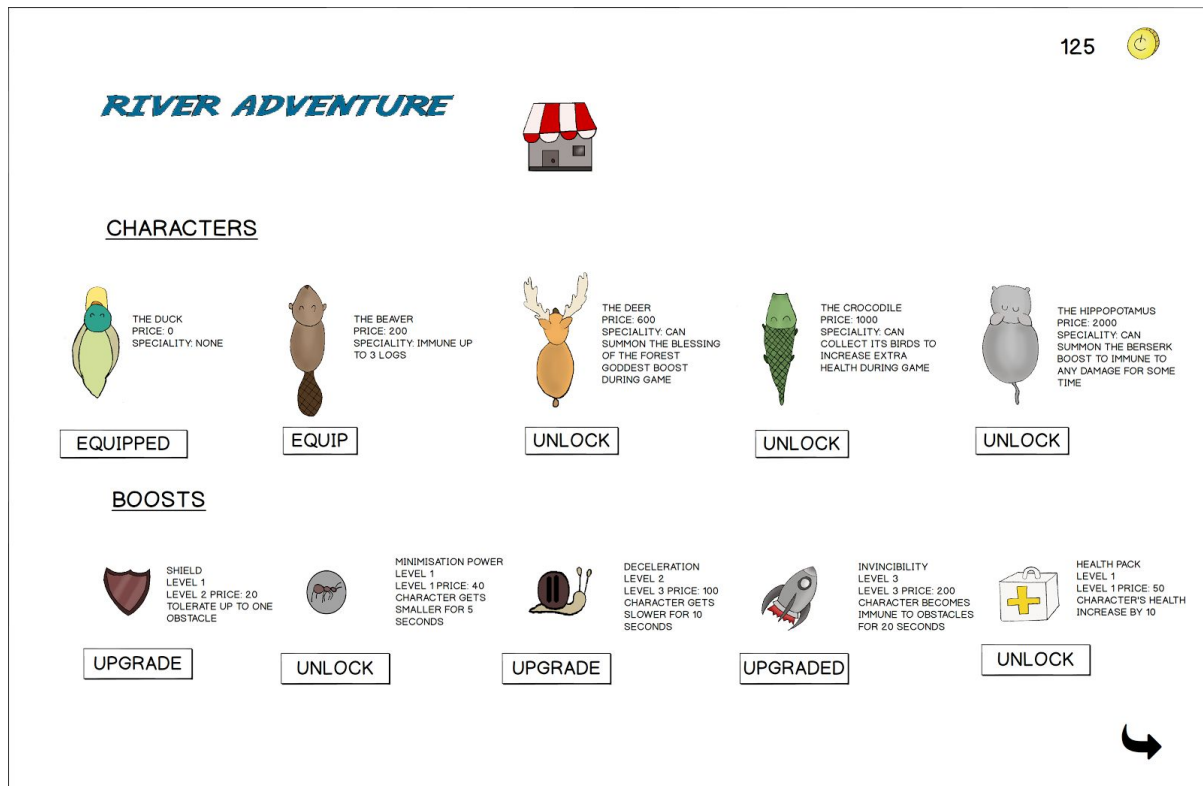


Figure 50. Store Screen

## 2.6.6. Settings Screen

When the player clicks the setting button from either pause screen or main menu screen, settings screen will show up. In the settings screen player can adjust the sound and/or mute the game.

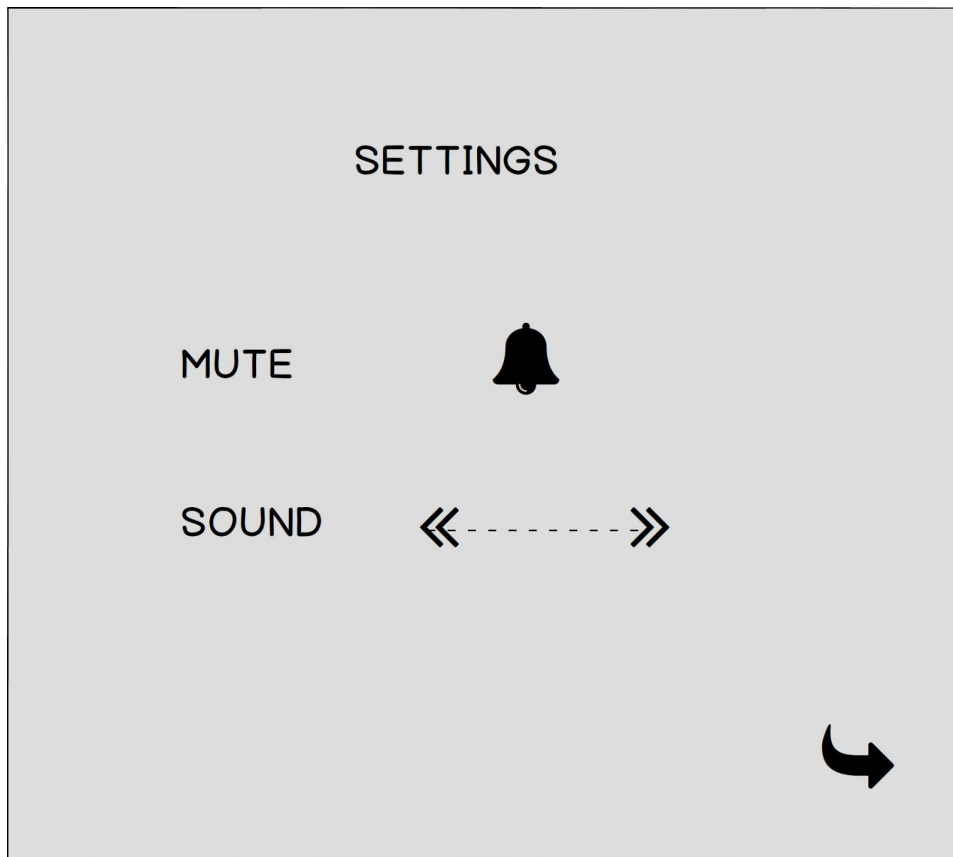


Figure 51. Settings Screen

### 2.6.7. Help Screen

In the main menu screen, when the player clicks help button, help screen shows up. In the help screen there is a textual description of the game and the gameplay.

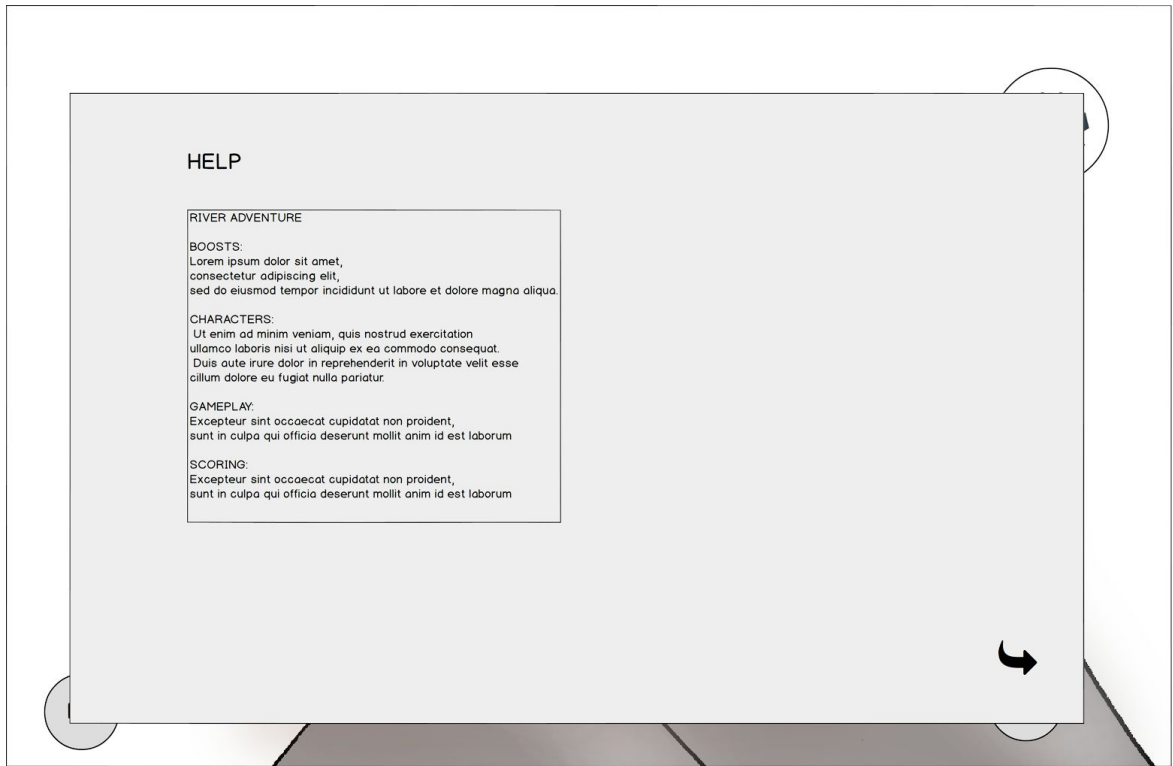


Figure 52. Help Screen

### 2.6.8. Credits Screen

In the main menu screen, when the player clicks credits button, credits screen will show up. In this screen the developers and the ones that are needed to be credited is shown.

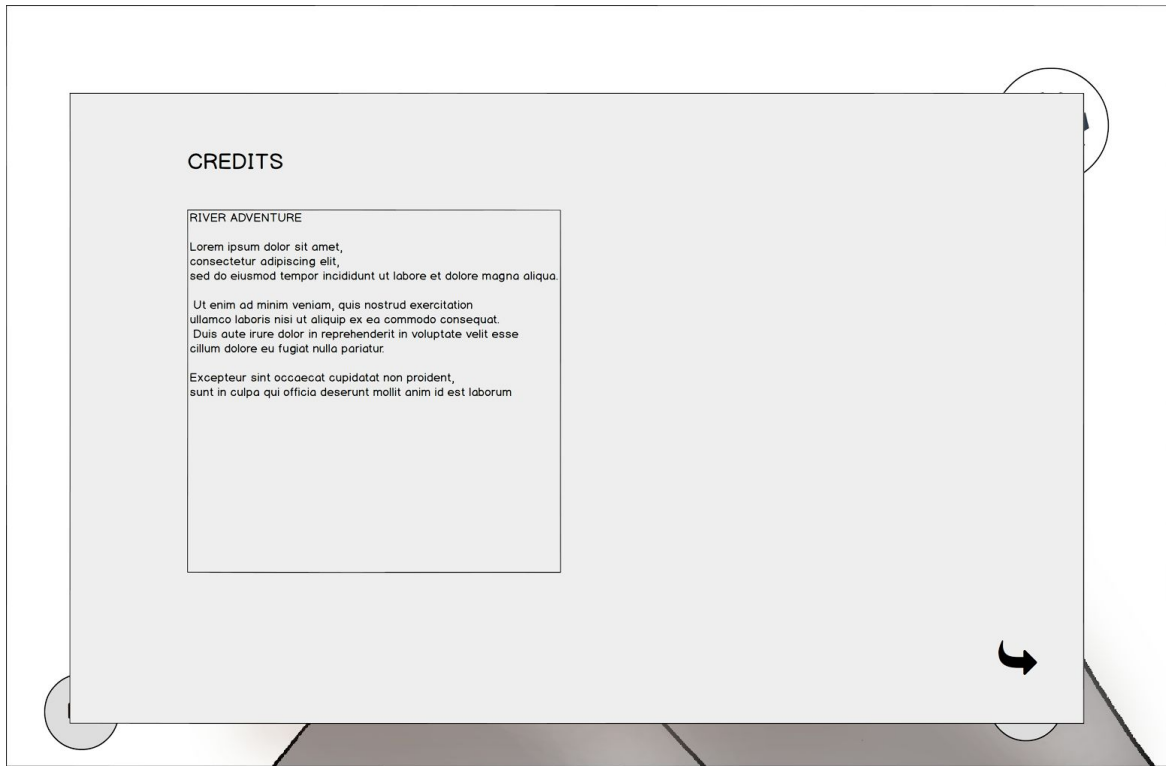


Figure 53. Credits Screen

### 3. Conclusion

In this report the game River Adventure is examined and its requirements are extracted. These requirements are analyzed in order to plan the next steps of building the game. Also, functionalities of the system are examined and modeled in this report.

Initially, general functions of the game are identified. The context and control of the game is specified. Gameplay and items are explained in detail. Hence, the answer to the questions about River Adventure that what is it, how it is played, what are the properties and tasks of the game, are answered.

Then, considering the description of the game, requirements are extracted. What River Adventure should be able to do for certain functionalities are identified. Furthermore, non-functional requirements and system constraints are specified. Based on the requirements, sample scenarios are created for various functionalities, gameplays and items. Then these scenarios are generalized as use cases.

After identifying the requirements and use cases, object model and dynamic models are created. The classes and their interactions are shown in the class diagram. The dynamic model is represented with activity diagrams, statecharts and sequence diagrams. All of these diagrams created a comprehensive notion of River Adventure along with its gameplay.

Furthermore, the user interface of River Adventure is included in the report. Screen designs and pictures of game objects are provided.