**Software Requirements Specification**

***Kitten Math Game***

CPSC 362

By: Benjamin Heng, Jake Wong, Quyen Tsai

Professor: Dr. Lidia Morrison

Department of Computer Science

California State University, Fullerton

Fall, 2020

**Table of Contents**

[**1. Introduction 3**](#_Toc58460323)

[**1.1 Purpose 3**](#_Toc58460324)

[**1.2 Scope of the Problem 3**](#_Toc58460325)

[**1.3 Intended Audience 4**](#_Toc58460326)

[**2. Overall Description 4**](#_Toc58460327)

[**2.1 User Objectives 4**](#_Toc58460328)

[**2.2 Operating Environment 4**](#_Toc58460329)

[**2.3 Constraints 5**](#_Toc58460330)

[**3. Functional Requirements 5**](#_Toc58460331)

[**3.1 The app shall display a main menu 5**](#_Toc58460332)

[**3.2 - The app shall let the user choose a different game mode 5**](#_Toc58460333)

[**3.3 - The app shall allow the user to choose the “addition” game mode 5**](#_Toc58460334)

[**3.4 - The app shall generate a randomized “addition” math problem 6**](#_Toc58460335)

[**3.5 - The app shall allow the user to choose the “subtraction” game mode 6**](#_Toc58460336)

[**3.6 - The app shall generate a randomized “subtraction” math problem 6**](#_Toc58460337)

[**3.7 - The app shall allow the user to choose the “count” game mode 6**](#_Toc58460338)

[**3.8 - The app shall generate a randomized “count” math problem 7**](#_Toc58460339)

[**3.9 - The app shall allow the user to add and remove kittens in the game mode. This applies to all game modes. 7**](#_Toc58460340)

[**3.10 - The app shall allow the user to submit the answer to the problem. This applies to all game modes. 7**](#_Toc58460341)

[**3.11 - The app shall let the user know if their answer was right or wrong. This applies to all game modes. 8**](#_Toc58460342)

[**3.12 - The app shall decrease user’s tries if the user answers the question incorrectly. This applies to all game modes. 8**](#_Toc58460343)

[**3.13 - The app shall display the amount of tries the user has. This applies to all game modes. 8**](#_Toc58460344)

[**3.14 - The app shall display the number of lives the user has as “Apples” on the screen. This applies to all game modes. 9**](#_Toc58460345)

[**3.15 - The app shall decrease the user's lives if the user uses up his or her two tries. This applies to all game modes. 9**](#_Toc58460346)

[**3.16 - The app shall display that the user loses a life after the user gets the same question wrong twice. This applies to all game modes. 9**](#_Toc58460347)

[**3.17 - The app shall continue to generate problems as long as the user has lives left. This applies to all game modes. 9**](#_Toc58460348)

[**3.18 - The app shall display a “game over” screen if all lives are lost. This applies to all game modes. 10**](#_Toc58460349)

[**3.19 - The app shall give the user options such as to “Try again” or “return to the game menu” on the game over screen. This applies to all game modes. 10**](#_Toc58460350)

[**3.20 - The app shall bring the user back to the game the user was playing before they lost if the user clicked the button “Try again” from the “Game over” screen. This applies to all game modes. 10**](#_Toc58460351)

[**3.21 - The app shall bring the user back to the game menu if the user clicks the button “Game Menu” from the “Game over” screen..  This applies to all game modes. 11**](#_Toc58460352)

[**3.22 - The app shall take the user back to the “Main Menu” screen if the user clicks the button “Back” in the “Game Menu” screen. 11**](#_Toc58460353)

[**3.23 - The app shall take the user back to the “Game Menu” screen if the user clicks the button “Back” while playing any game mode. This applies to all game modes. 11**](#_Toc58460354)

[**3.24 - The app shall allow the user to access “option” screen from “Main Menu” 11**](#_Toc58460355)

[**3.25 - The app shall allow the user to adjust the game volume in options 12**](#_Toc58460356)

[**3.26 - The app shall allow the user to increase the game volume in options 12**](#_Toc58460357)

[**3.27 - The app shall allow the user to increase the game volume in options 12**](#_Toc58460358)

[**3.28 - The app shall allow the user to exit the game 12**](#_Toc58460359)

[**4.0 Non-Functional Requirements 12**](#_Toc58460360)

[**5.0 UML Diagrams 13**](#_Toc58460361)

[**5.1 Use Case Diagram 13**](#_Toc58460362)

[**5.2 Activity Diagram for a Game Mode 14**](#_Toc58460364)

[**APPENDIX A: User Operations Manual 15**](#_Toc58460366)

[**APPENDIX B: References and Tools Used 21**](#_Toc58460375)

# **Table of Figures**

[**Figure 1  - Use Case Diagram for All User Options 13**](#_Toc58460363)

[**Figure 2 - Activity Diagram for a Game Mode 15**](#_Toc58460365)

[**Figure 3 - Screenshot of the Main Menu 16**](#_Toc58460367)

[**Figure 4 - Screenshot of the Option screen 17**](#_Toc58460368)

[**Figure 5 - Screenshot of the Game mode menu 17**](#_Toc58460369)

[**Figure 6 - Screenshot of the Addition game 18**](#_Toc58460370)

[**Figure 7 - Screenshot of Addition game after adding answer 18**](#_Toc58460371)

[**Figure 8 - Screenshot of Subtraction game 19**](#_Toc58460372)

[**Figure 9 - Screenshot of Counting game 20**](#_Toc58460373)

[**Figure 10 - Screenshot of end game screen 20**](#_Toc58460374)

# **1. Introduction**

## **1.1 Purpose**

The following document offers a description of the software requirements for our 362 group project. It will explain the general purpose and reasoning behind our game, Kitten Math, which will include the software requirements and goals that we wish to accomplish through the program.

## **1.2 Scope of the Problem**

These days, parents are trying to find supplemental ways to ensure that their children are staying on top or get ahead of what they’re learning in school, whether it’s through, but not limited to, after-school academies or textbooks that the children can use. The game Kitten Math offers a solution for parents by allowing first grade children to hone their mathematical skills through a fun, engaging game, which offers a simplified interface and cutesy graphics.

Kitten Math allows children to choose between three different game modes (addition, subtraction, and counting) as a way to have them practice different areas of mathematics at their current grade level. The games are designed in an arcade-like fashion in that children will be rewarded for consecutive questions answered correctly, while challenging them to not lose a set amount of lives before the game is over.

## **1.3 Intended Audience**

The intended audience of this document includes Professor Lidia Morrison, and the members of Group 11, who are in charge of designing, implementing, and executing the functionality of the software. Other potential users include all students enrolled in this section of CPSC 362 for Fall 2020 at California State University, Fullerton. As mentioned earlier, this application is intended for first-grade children seeking to have fun while practicing their addition, subtraction, and counting skills.

# **2. Overall Description**

## **2.1 User Objectives**

Kitten Math is a desktop game intended for children at the first-grade level, who want to improve their mathematical skills outside of a classroom setting. The three game modes that children can engage in are addition, subtraction, and counting to 20. Each game mode is designed in an arcade-like fashion, with a set amount of lives and score multipliers for consecutive questions answered correctly, to ensure that learning and practicing mathematical concepts is fun for the children.

## **2.2 Operating Environment**

The Kitten Math game is an executable program that was developed using the Python programming language, using the PyGame library, responsible for making up the bulk of the graphical user interface (GUI) of the game, while the rest of Python is responsible for handling the logic behind the overall function of the program. The Target OS (Operating System) for this project is Windows 10, as the developers of the program who created and tested the game all possessed Windows 10 computers.

## **2.3 Constraints**

As of now, because Kitten Math was designed and tested on Windows 10 desktops, the developers’ native operating system, the program can for sure run in that environment. It is not designed for mobile use since it requires the usage of mouse clicks. The developers also attempted to test the program in a Mac OS environment, concluding that it does not work for that operating system, as the .exe file extension was designed for a Windows environment.

# **3. Functional Requirements**

## **3.1 The app shall display a main menu**

**Description**

On clicking the app icon, the program shall display a main menu, listing three options (Play, Options, Quit) that a user can choose.

**Pre-condition** - User has downloaded and installed Kitten Math

**Post-condition** - The main menu is displayed

## **3.2 - The app shall let the user choose a different game mode**

**Description**

Once the user is in the main menu and clicks on play, there shall be 3 different game modes (addition, subtraction, count) that the user can choose.

**Pre-condition** - User click “Play”

**Post-condition** - Game mode menu is displayed

## **3.3 - The app shall allow the user to choose the “addition” game mode**

**Description**

Once the user is in the game mode menu, the user can click on the “Addition” button to play the game.

**Pre-condition** - User in Game mode menu and click the “Addition” button

**Post-condition** - User in the “Addition” game and the addition game screen is displayed.

## **3.4 - The app shall generate a randomized “addition” math problem**

**Description**

When inside the “addition” game, the programs shall generate random problems, displaying kittens on the left box and the right box for the user to determine the answer.

**Pre-condition** - User is playing in “addition” game

**Post-condition** - The numbers of cats generated on the right box and left box are displayed randomly for each problem.

## **3.5 - The app shall allow the user to choose the “subtraction” game mode**

**Description**

Once the user is in the “Game Mode” menu, the user can click on “Subtraction” game to play.

**Pre-condition** - User in “Game Mode” menu and click the “Subtraction” button

**Post-condition** - User in the “Subtraction” game and the “Subtraction” game screen is displayed

## **3.6 - The app shall generate a randomized “subtraction” math problem**

**Description**

When playing the “Subtraction” game, the programs shall generate random problems, displaying kittens on the left box and the right box for the user to determine the answer.

**Pre-condition** - User is playing “Subtraction” game

**Post-condition** - The numbers of cats generated on the right box and left box are random for each problem, which generate and display randomized problems with randomized answers.

## **3.7 - The app shall allow the user to choose the “count” game mode**

**Description**

Once the user is in the game mode menu, the user can click on “Count” game to play

**Pre-condition** - User in Game mode menu and click the “Count” button

**Post-condition** - User in the “Count” game and the “Count” game screen is displayed.

## **3.8 - The app shall generate a randomized “count” math problem**

**Description**

When playing the “Count” game, the programs shall generate random problems, displaying randomized kittens in the whitebox for the user to count.

**Pre-condition** - User is playing the “Count”

**Post-condition** - The numbers of kittens generated on the Whitebox is randomized, and the randomized number of kittens shall be displayed

## **3.9 - The app shall allow the user to add and remove kittens in the game mode. This applies to all game modes.**

**Description**

The application shall add or remove the kittens in the answer box if the user clicks on the button “add” or the button “remove”.

**Pre-condition** - User inside a game mode.

**Post-condition** - User click the “remove” button and it remove a kitten, or User click the “add” button and the program shall add a kitten

## **3.10 - The app shall allow the user to submit the answer to the problem. This applies to all game modes.**

**Description**

The app shall allow the user to click the “submit” button and submit the answer for the program to check.

**Pre-condition** - The user is playing a game from any game mode.

**Post-condition** - Once user click ‘submit’ button, the user’s answer is compared to the correct answer

## **3.11 - The app shall let the user know if their answer was right or wrong. This applies to all game modes.**

**Description**

The app shall make a sound if the answer is correct and move to the next problem, if the answer is incorrect, the app shall make a different sound and the user loses a “tries”. This applies to all game modes.

**Pre-condition** - User is in one of the game and click the button ‘submit’

**Post-condition** - “Mario coin” sound (https://www.youtube.com/watch?v=RfkcI8dhfsQ) shall play if the answer is correct and “Roblox Death Sound” (https://www.youtube.com/watch?v=f49ELvryhao) shall play if the answer is incorrect.

## **3.12 - The app shall decrease user’s tries if the user answers the question incorrectly. This applies to all game modes.**

**Description**

The app shall decrease the amount of times the user can do the same problem if the user answers the problem incorrectly. Users get two chances to answer the problem correctly.

**Pre-condition** - User answer the problem incorrectly

**Post-condition** - The number of “tries” decrease

## **3.13 - The app shall display the amount of tries the user has. This applies to all game modes.**

**Description**

The app shall display the amount of “tries” user has for each question.

**Pre-condition** - User is playing one of the game modes

**Post-condition** - The amount of “tries” get displayed as Tries: [number]

## **3.14 - The app shall display the number of lives the user has as “Apples” on the screen. This applies to all game modes.**

**Description**

The amount of lives a user has (user has 3 lives every time they play a game) to play the game shall be display as Apple

**Pre-condition** - The user play any game mode

**Post-condition** - Three apples representing the lives the user has in the game shall be displayed on the top right corner of the screen.

## **3.15 - The app shall decrease the user's lives if the user uses up his or her two tries. This applies to all game modes.**

**Description**

If the user answers a problem incorrectly twice on the same problem, their lives shall be decreased by one.

**Pre-condition** - The user play a game and get the same question wrong twice

**Post-condition** - User loses a life

## **3.16 - The app shall display that the user loses a life after the user gets the same question wrong twice. This applies to all game modes.**

**Description**

If the user answers a problem incorrectly twice on the same problem, the amount of apple that is being display on the screen shall decrease by one

**Pre-condition** - User answer the same problem wrong twice in a row in any game mode

**Post-condition** - The number of apples being displayed decreases by one.

## **3.17 - The app shall continue to generate problems as long as the user has lives left. This applies to all game modes.**

**Description**

As long as the user has lives left, the game shall continuously generate new problems.

**Pre-condition** - User play any game mode

**Post-condition** - The user plays the game until there are no lives left or the user left the game.

## **3.18 - The app shall display a “game over” screen if all lives are lost. This applies to all game modes.**

**Description**

If the user lose all their lives (3), a “game over” screen shall appear with options for user to choose

**Pre-condition** - User play any game mode and lose all their lives

**Post-condition** - “Game over” screen appear with clickable options

## **3.19 - The app shall give the user options such as to “Try again” or “return to the game menu” on the game over screen. This applies to all game modes.**

**Description**

If the user loses the game and gets to the “game over” screen, the user shall get the option to “Try again”, or to return to the “Game Menu”. Users can choose their option by clicking on the “Try again” or “Game Menu” options.

**Pre-condition** - User is in the “Game over” screen

**Post-condition** - User click “Try again” or “Game Menu”

## **3.20 - The app shall bring the user back to the game the user was playing before they lost if the user clicked the button “Try again” from the “Game over” screen. This applies to all game modes.**

**Description**

The application shall take the user back to the game they were playing if they click the button “Try again” from the “game over” screen

**Pre-condition** - User click the “Try again” button

**Post-condition** - The game that the user was playing shall be displayed and the user can replay the game.

## **3.21 - The app shall bring the user back to the game menu if the user clicks the button “Game Menu” from the “Game over” screen.  This applies to all game modes.**

**Description**

The application shall take the user back to the “Game Menu” if they click the button “Game Menu” from the “Game over” screen

**Pre-condition** - User click the “Game Menu” button from the “Game over” screen

**Post-condition** - Display the “Game Menu”

## **3.22 - The app shall take the user back to the “Main Menu” screen if the user clicks the button “Back” in the “Game Menu” screen.**

**Description**

If the user clicks the button “Back” while in the “Game Menu” the app shall take the user to the “Main Menu”

**Pre-condition** - user is currently in the “Game Menu”

**Post-condition** - “Main Menu” is displayed

## **3.23 - The app shall take the user back to the “Game Menu” screen if the user clicks the button “Back” while playing any game mode. This applies to all game modes.**

**Description**

If the user clicks the button “Back” while playing “Addition”, “Subtraction”, or “Count” games, the application shall take the user back to the “Game Menu”

**Pre-condition** - user is currently playing a game from any game mode.

**Post-condition** - “Game Menu” is displayed

## **3.24 - The app shall allow the user to access “option” screen from “Main Menu”**

**Description**

The app shall allow the user to click on the “Option” button and access the option screen.

**Pre-condition** - User is in “Main Menu”

**Post-condition** - User in “Option” screen

## **3.25 - The app shall allow the user to adjust the game volume in options**

**Description**

User shall be allowed to adjust the volume, whether through decreasing or increasing the volume

**Pre-condition** - User in “Option” screen

**Post-condition** - The volume is adjusted.

## **3.26 - The app shall allow the user to increase the game volume in options**

**Description**

User shall be allowed to increase the volume through clicking the “+” buttons in “Option” screen

**Pre-condition** - User in “Option” screen

**Post-condition** - The volume increased

## **3.27 - The app shall allow the user to increase the game volume in options**

**Description**

User shall be allowed to decrease the volume through clicking the “-” buttons in “Option” screen

**Pre-condition** - User in “Option” screen

**Post-condition** - The volume decreased

## **3.28 - The app shall allow the user to exit the game**

**Description**

User shall be allowed to exit the game through clicking the “Quit” button in “Main Menu”

**Pre-condition** - User is in “Main Menu”

**Post-condition** - The program ended

# **4.0 Non-Functional Requirements**

**Usability:**

* The program *will* be easy to handle and is relatively straight forward in execution.
* It *will* switch between states within 1 second to no delay.

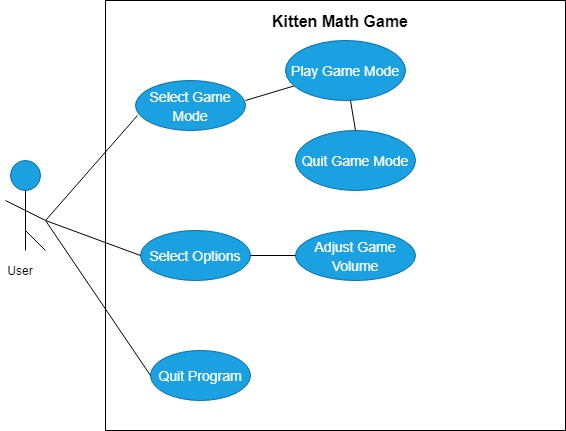
**Supportability:**

* *Will* only run in Windows 10.

# **5.0 UML Diagrams**

## **5.1 Use Case Diagram**

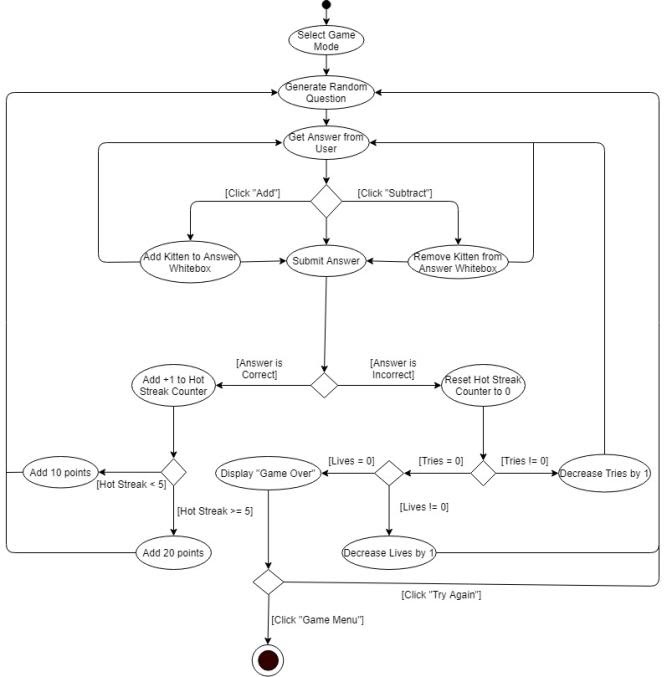
As soon as the user loads the program, they are presented with three options: they can “Play” the game, go to “Options” to adjust in-game volume, or “Exit” the game. When the user clicks “Play” they can choose between three different game modes: addition, subtraction, and counting. Each game mode works similarly, in that the user is given an addition, subtraction, or counting prompt, and they are required to click “add” or “remove” to increment or decrement the number of cats in their response, respectively. They can also submit their answer once they feel like their answer is correct. When clicking “Options” the user can adjust the in-game volume accordingly using a slider. Lastly, when the user clicks the “Exit” button, the program is terminated.



# **Figure 1 - Use Case Diagram for All User Options**

## **5.2 Activity Diagram for a Game Mode**

After clicking “Play” in the main menu and choosing any of the three game options: Addition, Subtraction, or Counting, the game will first generate a random question. It is there that the game will await the answer from the user, to which the user can either click the “ADD” button to add a kitten to the answer whitebox, which contains the user’s answer to the question, or conversely click “REMOVE” to remove a kitten to the answer whitebox. At any point, regardless of whether or not the user chooses to fill the answer whitebox with kittens, they can click “submit” to send their answer for judgment. If the answer is correct, depending on whether or not there is a hot streak will affect how many points the user gets for each correct answer. A user gets two tries per question before a life is deducted and they are forced to move on to the next question, or if they run out of the three lives that they are initially provided. At any point the user can quit the game, to which they will return to the game mode selection screen.



# **Figure 2 - Activity Diagram for a Game Mode**

# **APPENDIX A: User Operations Manual**

This section of this document provides an understanding of how to use the Kitten Math App.

1. **Main Menu:**

Upon opening kitten\_math.exe, the program’s main menu shall be displayed and shall remain at the home screen unless the user clicks “Start Game”, “Option” or “Quit”. Users can click “Start Game” to be direct to the “Game Menu”, “Option” to be direct to the option screen, or “Quit” to quit the game.



# **Figure 3 - Screenshot of the Main Menu**

1. **Change Volume:**

Volume can be changed by clicking the “Option” menu from the “Main Menu”. Once the “Option” screen shows up, click the symbol “+” to increase the volume, or click the symbol “-” to decrease the volume.  User can click button “Back” to go back to “Main Menu”



# **Figure 4 - Screenshot of the Option screen**

1. **Choose game mode from “Game Menu”**

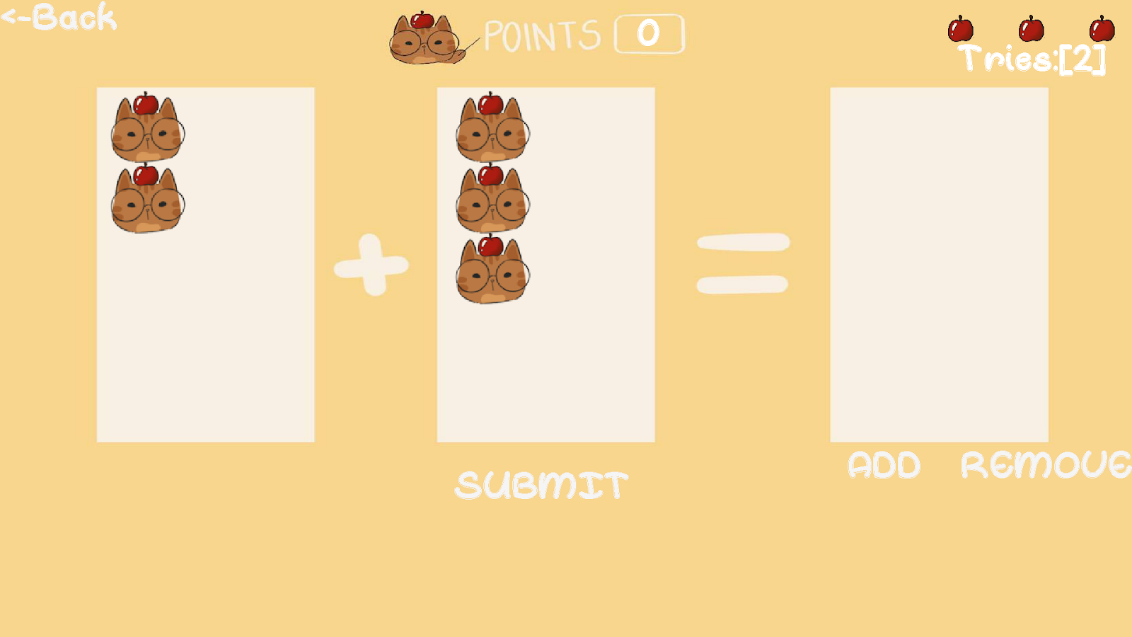
Click on a game mode from the “Game Menu” to play. User can click button “Back” to go back to “Main Menu”



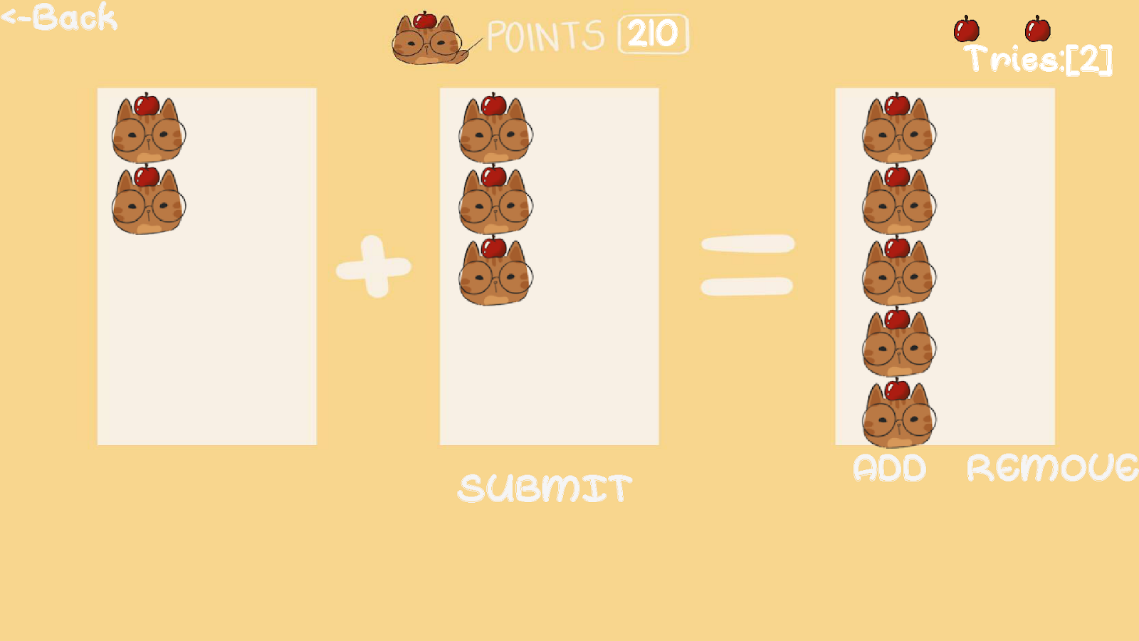
# **Figure 5 - Screenshot of the Game mode menu**

1. **How to play “Addition” game mode:**

To play the game, count the kittens of the left and right, then click “ADD” to add kittens according to the user's calculation, so in the picture, there are 2 kittens on the left white box, 3 on the right. After adding, it totaled out to 5 kittens. Users can then click “ADD” to add kittens as an answer or “REMOVE” to remove kittens from the answer whitebox. After adding the number of kittens as an answer, the user can click “SUBMIT” to submit the answer. User can click the button “Back” to go back to “Game Menu”

****

# **Figure 6 - Screenshot of the Addition game**



# **Figure 7 - Screenshot of Addition game after adding answer**

1. **How to play “Subtraction” game:**

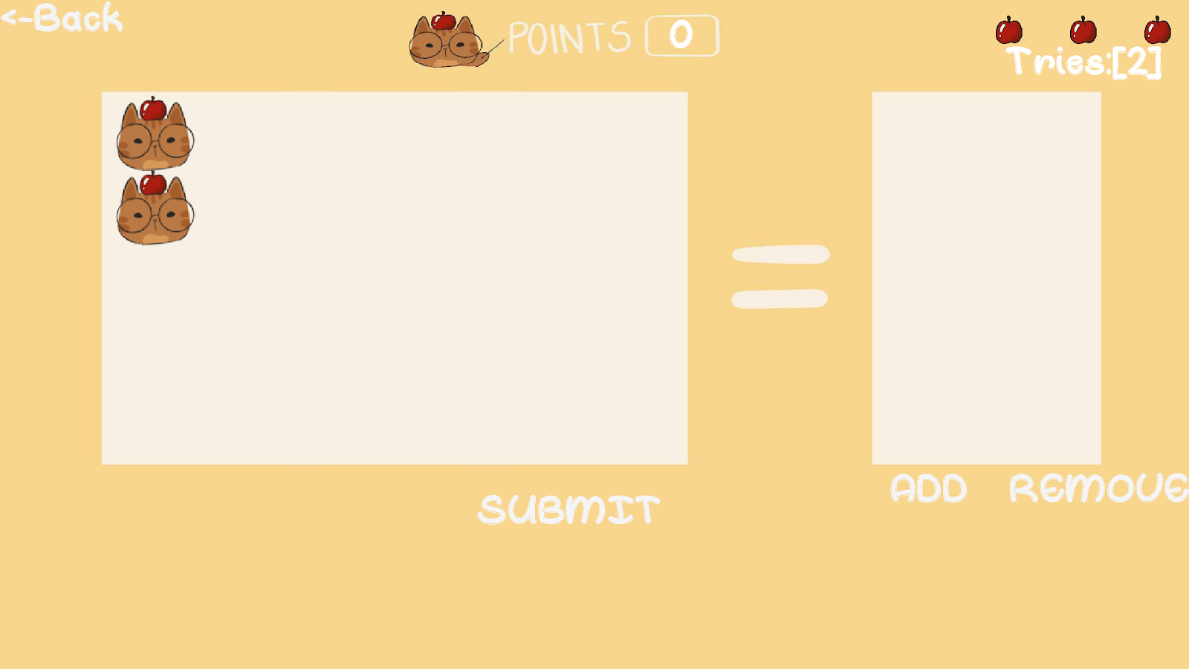
To play the game, count the kittens on the left and subtract the amount of kitten on the right, then click “ADD” to add kittens according to the user's calculation, so in the picture, there are 3 kittens on the left white box, 2 on the right. After subtracting, it totaled out to 1 kittens. Users can then click “ADD” to add kittens as an answer or “REMOVE” to remove kittens from the answer whitebox. After adding the number of kittens as an answer, the user can click “SUBMIT” to submit the answer. User can click the button “Back” to go back to “Game Menu”



# **Figure 8 - Screenshot of Subtraction game**

1. **How to play the “Count” game:**

To play the game, count the kittens in the question whitebox, then click “ADD” to add kittens according to the user's calculation. In the picture, there are 2 kittens. After counting, there are 2 kittens. Users can then click “ADD” to add kittens as an answer or “REMOVE” to remove kittens from the answer whitebox. After adding the number of kittens as an answer, the user can click “SUBMIT” to submit the answer. User can click the button “Back” to go back to “Game Menu”

****

# **Figure 9 - Screenshot of Counting game**

1. **How to play again:**

After losing all “lives” in game, click “Play again” to replay the game mode that was being played. If user doesn’t want to play again, the user can click the button “Game Menu” to return to the game mode menu.



# **Figure 10 - Screenshot of end game screen**

# **APPENDIX B: References and Tools Used**

**References**

SRS Example for Banker Buddy from Fall 2014 - Provided by Professor Lidia Morrison

Lecture Notes and Slides by Professor Morrison

Functional vs Non-Functional Requirements: The Definitive Guide - Qracorp

**Tools Used**

The following tools were used for developing this SRS and drawing all UML diagrams:

* SRS Document: Google Docs, Microsoft Word
* UML Diagrams: https://draw.io

The following program was used to create the Kitten Math Program:

* PyCharm Community Edition

Sound Effects Used in Program (credits):

* Game Music - “Survive the Fall” from Fall Guys Game, by Mediatonic (https://www.youtube.com/watch?v=h2TpZXnWVsw)
* Correct Answer - “Super Mario Bros.-Coin Sound Effect” from Nintendo (https://www.youtube.com/watch?v=RfkcI8dhfsQ)
* Incorrect Answer - “Roblox Death Sound Effect” from Roblox, by Roblox Corporation (https://www.youtube.com/watch?v=f49ELvryhao)