Software Requirements and Design Document

For

Korona Kingdom

Version 1.0

Authors:

Alexander Kostandarithes

Anthony Micciche

Ryan Goldberg

Karl Cooley

Alexander Jordan

1. Overview (5 points)

Korona Kingdom is a turn based RPG based on a parody of modern life under quarantine and inspired by early JRPGs like Dragon Warrior. It is written using the python arcade library. It will contain an inventory system, three dungeons, an overworld, a combat system, sprite animation, and the joy of adventure!

2. Functional Requirements (10 points)

Key: Working Issues Un-Implemented

Inventory System:

1. When the player hits [I] in the overworld the user's inventory should open. - High
2. When the player hits [I] in their inventory the user's inventory should close - High
3. When the inventory is open all the player's stats should be visible - High
    a. Level / HP / Energy / Strength / Defense / Intelligence / Immunity / Agility / Luck
        i. Stats wanted by Anthony for combat system
4. When the inventory is open all the player's equipped items should be shown - High
5. When the inventory is open the player's inventory should be separated by component type - Med
    a. Helmet / Weapon / Shirt / Gloves / Pants / Shoes / Consumables / Quest Items
6. When the inventory is opened the current component tab selected should be highlighted and be the leftmost tab - Med
7. When the player hits [→] or [←] the current component tab selected should change relative to what is currently selected - Med
8. When the player opens the inventory their should be a list of items related to the current selected component tab that shows what the player has picked up - High
    a. A single list Item shows: Name, Image
9. When the inventory is opened the first item in each of these component list should be the item that the player currently has equipped - Low
10. When the player hits [↑] or [↓] the player should be able to scroll up/down their list of items in the chosen component tab - High
11. When the player scrolls through these items the current selected item should show its description and if its stats are currently better than the item currently equipped on another panel- Low
12. When the player is in their inventory the user should not be able to enter a battle - High
13. When the player is in battle they should not be able to enter their inventory - High

Overlay System

Player Info

1. When the player walks around the overworld the user's HP should be shown - High
2. When the player walks around the overworld the user's energy should be shown - High
3. When the player walks around the overworld the user's image should be shown - Low

Dialogue Box

1. When interacting with certain items / people the dialogue box should display - High
2. When the dialogue box appears the user should know who is speaking - High
3. When the dialogue box appears the user should see an image of who is speaking - Low
4. When the dialogue box appears the user should know what is being said - High
5. Once the interactions are over the dialogue box should disappear - High

Menu Box

1. When the player walks around the overworld the user should see all the available menu options - Med
    a. Inventory / Map / Options

Map system:

Overworld:

1. When the game starts the player's beginning position should be right outside the 'house' in the overworld map - med
2. The overworld map should allow the user to follow the roads to the different dungeons. - high
3. The overworld map will have three 'dungeons' at each corner and the player's 'house' in the fourth corner. - high
4. The overworld map should also allow users to take shortcuts through the 'forests'. - med
5. The overworld map should not allow the user to walk more than one map tile outside of the 'road' unless they are walking through a forest. - high
6. Each dungeon should allow the user to enter that dungeon upon running into it on the overworld map. - high
7. Upon re-entering the overworld map from the dungeons the player should be located just outside of the door to the dungeon they were previously inside of. - high

Dungeons:

1. Upon entering the dungeon the player should be able to walk around the map. - high
2. Each dungeon should have a door that the player can run into in order to exit the dungeon. - high
3. Each dungeon should be relatively recognizable as the 'location' it represents - med
    a. I.e. 'the dollar store', 'the school', and 'MalMart'
4. Each dungeon should have a boss that must be 'encountered' before leaving the dungeon. - high

Encounter system:

1. The encounter system will be used for both random encounters as well as boss encounters. - high

2. The encounter system will halt all gameplay movement on the map behind it while an active encounter is occurring. - high
3. The encounter system will allow a list of strings to be passed in which will appear as menu options during the encounter. - high
4. The encounter system will allow the user to move an arrow key selector to different menu items in order to select their choice. - high
5. The encounter system will exit and allow the user to resume gameplay if the menu options ‘run’ or ‘hide’ are selected during an encounter.
6. The Dialogue box from the overlay system will be used by the encounter system to give the user updates about the current state of their encounter. - high
7. The encounter system will cause random encounters to occur while the user is in both the overworld and the dungeons. - high
8. The chances of a random encounter occurring will be increased upon entering a dungeon and will then decrease to the default probability again upon leaving the dungeon and entering the overworld again. - med
9. The encounter system will allow an enemy sprite object to be passed in and then will display this enemy within a UI window overlayed on top of the map. - high
10. The ‘trigger’ for the random encounters will be based on a directional key being released by the user (this means that the encounters may occur upon stopping movement, or direction change while still moving). - med

Combat system:

1. Battles will start using correct player stats, and upon completion will either give the player experience and currency or kill the player in some way. HIGH
    a. Hp will not be restored after combat
2. A master move list will be designed and implemented. Enemies and the player will contain a subset move list to choose moves from. HIGH
3. The combat system will be integrated with the UI. MED
4. The combat system will call enemies from an area dependent list. MED
5. Boss battles will be designed and implemented. MED

Animations:

1. When the player moves around the map the sprite should animate appropriately. - high
2. The player sprite should flip when the player changes directions - high
3. The player should have an accurate walk animation when moving around the map - high

3. Non-functional Requirements (10 points)

Inventory System:

1. When the player enters / leaves the inventory it should take less than a second

Overlay System

1. When the player walks around the map the overlay should not make the game run noticeably slower

Map system:

1. The overworld and dungeons should have pleasing artwork that leads to enjoyable gameplay
2. There should not be lags between entering/leaving the overworld or dungeons.

Encounter system:

1. The encounter system should not cause any undue slowdowns of the game
2. There should not be any lags when the encounter system is opening/closing
3. The artwork and menu UI of the encounter system should be visually appealing enough that it leads to enjoyable game play experience.

Animations:

1. The walk animations should be visually appealing.
2. Animation should not cause lag or a noticeable decrease in performance.

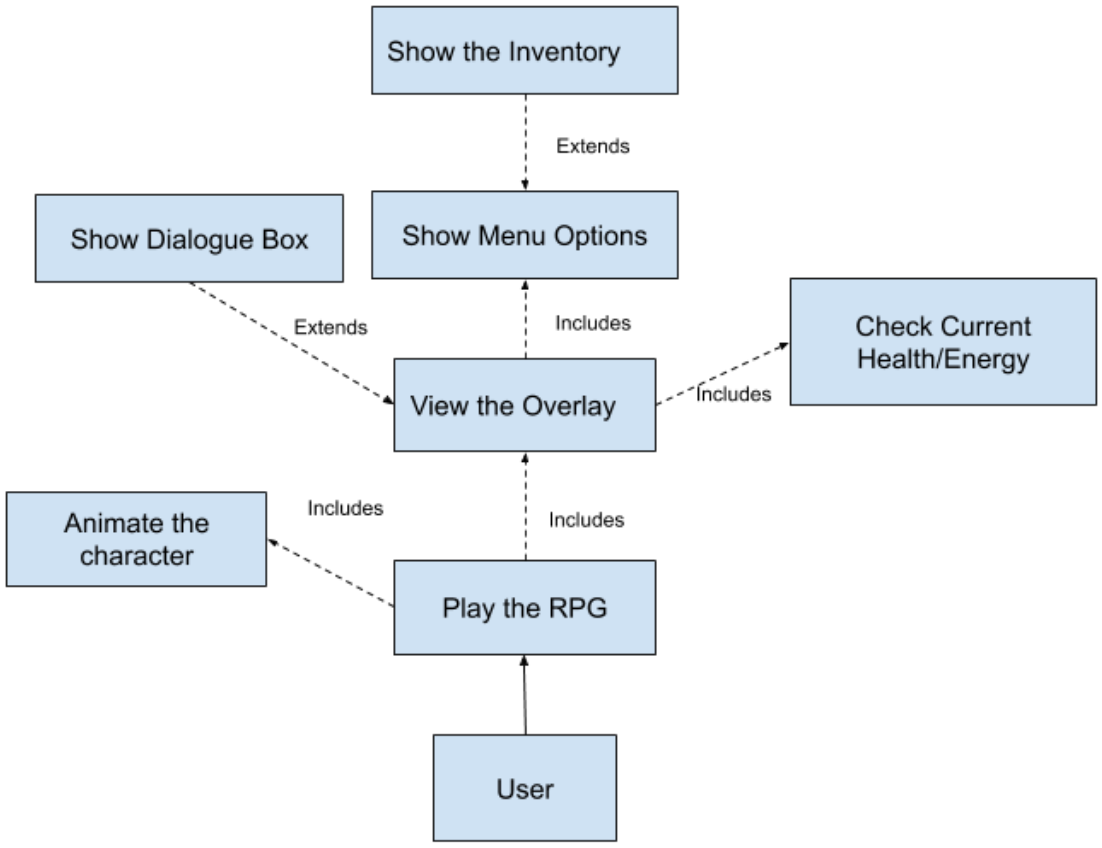
Combat:

1. It is imperative that the combat system be reliable as it is the heart of the game.

4. Use Case Diagram (10 points)

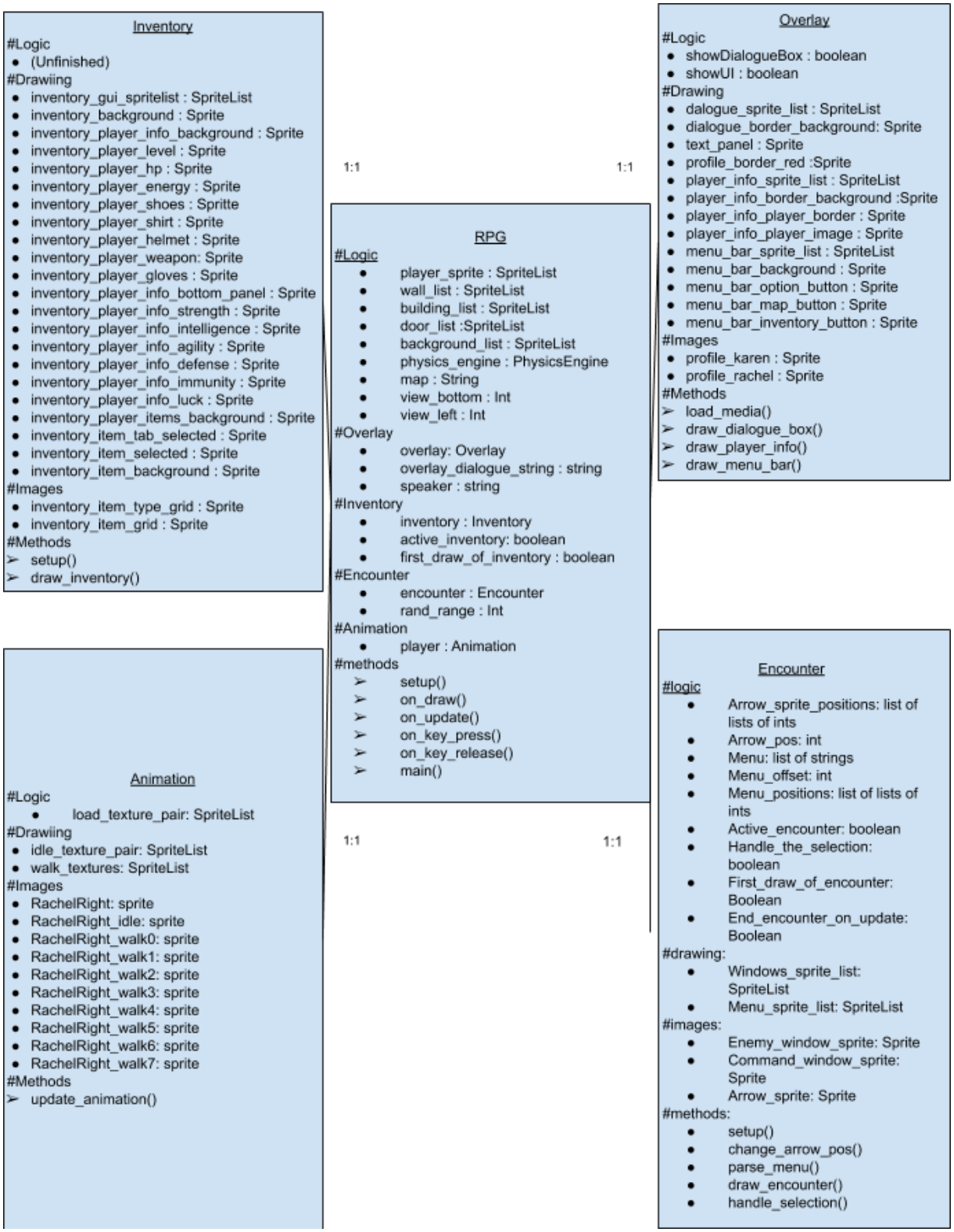
Extends - optional

Includes - mandatory

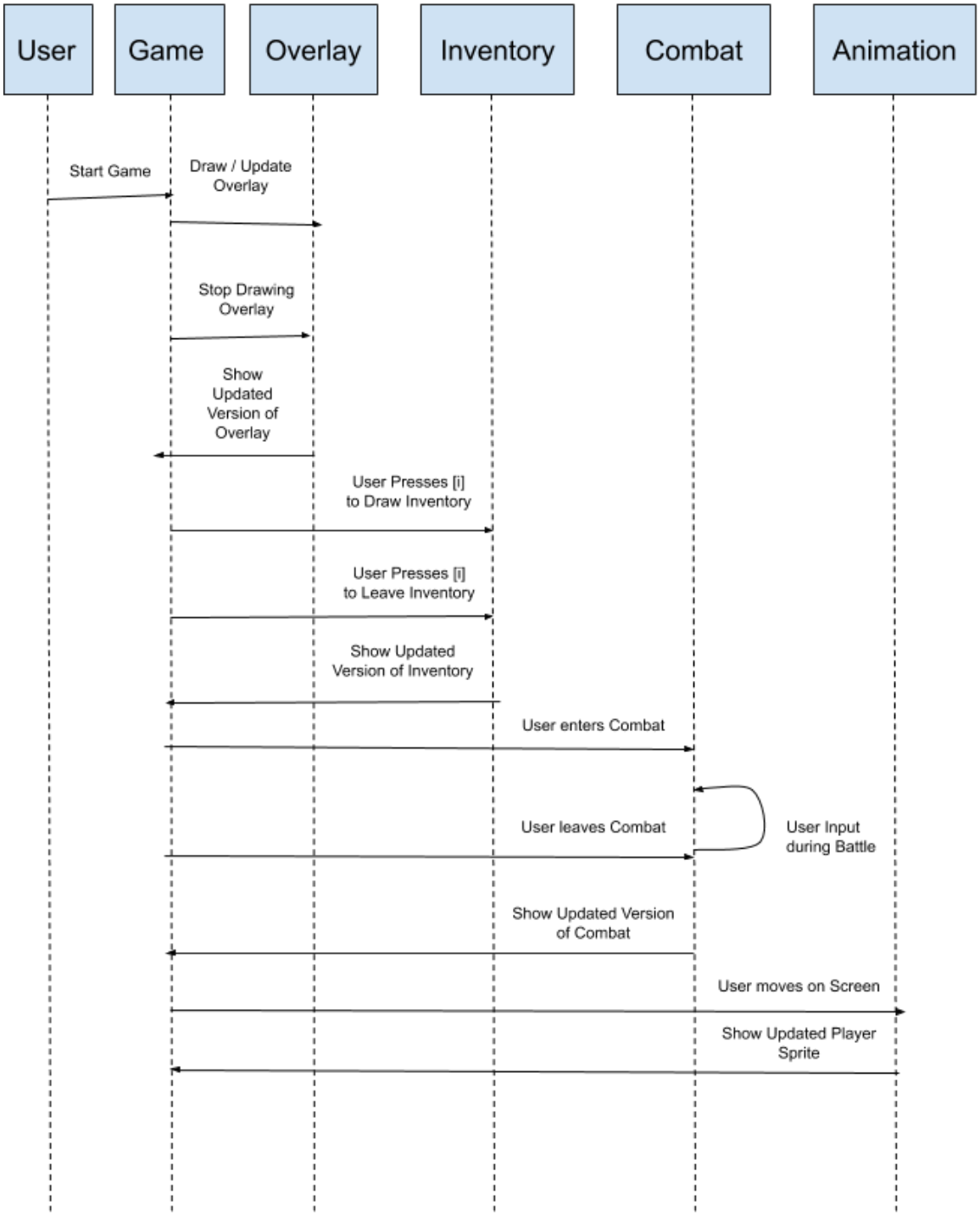


5. Class Diagram and/or Sequence Diagrams (15 points)

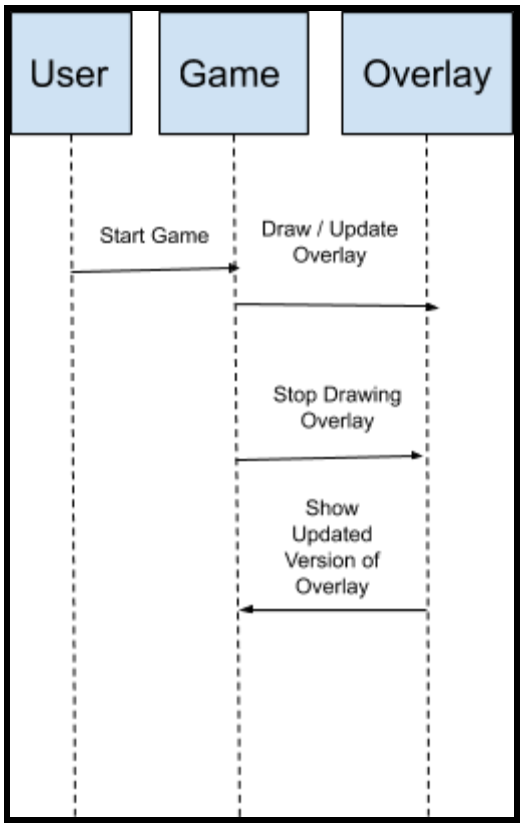
Class Diagram

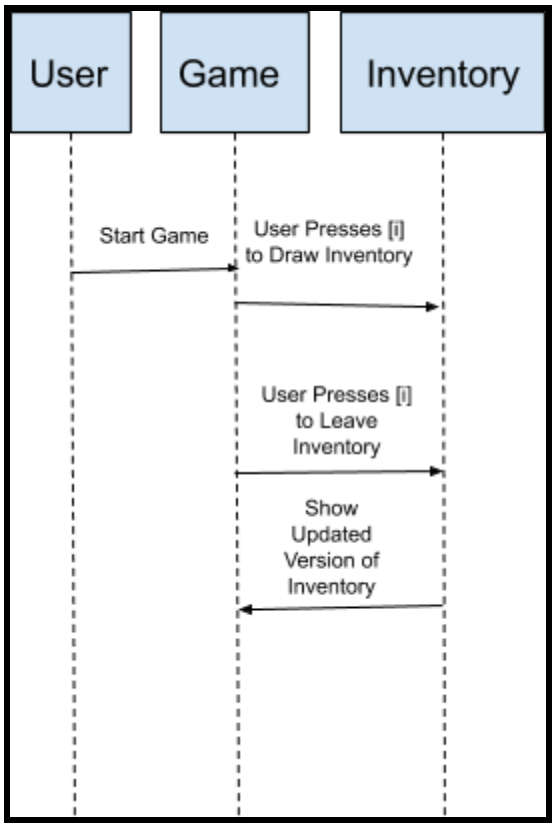


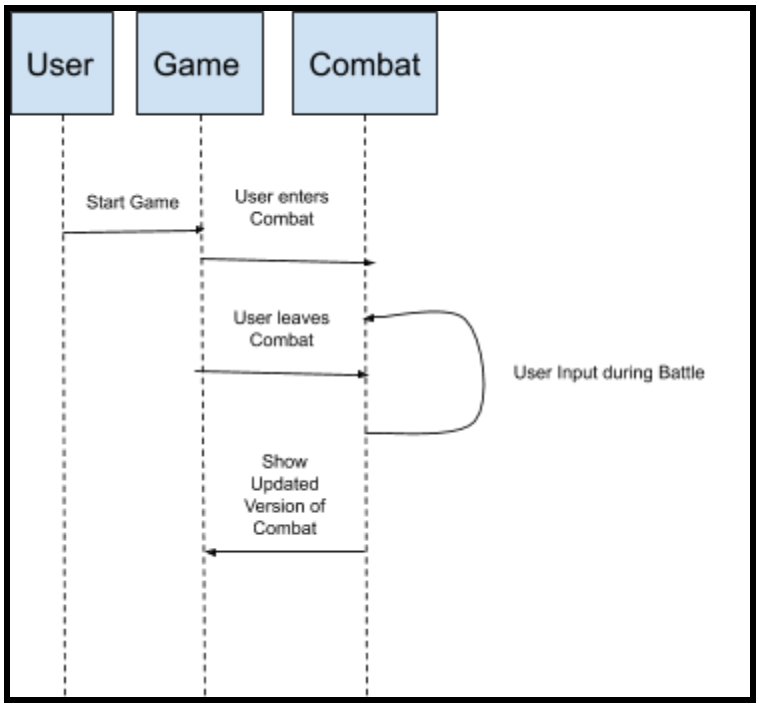
Sequence Diagram (whole)



Sequence Diagrams (Individual)







6. Operating Environment (5 points)

The software will operate on any PC whether it be desktop or laptop. The operating system/version of the PC should not have any effect on the user's ability to operate the software. In order for the software to run the user must have python3.6 or higher installed on their machine as well as the arcade library.

7. Assumptions and Dependencies (5 points)

Assumed factors:

- Assume that the arcade library is properly installed and remains functioning
- Assume that the user does not modify source code or integral components related to Korona Kingdom
- Assume that the player interphases with the game as intended (i.e not actively trying to exploit bugs or glitches)

Dependencies:

- Python3.6 or greater must be installed to run the software
- The python arcade library must be installed to run the software