

*Florida International University
School of Computing and Information Sciences*

Software Engineering Focus

Final Deliverable

VR-Gaming to Broad Participation in CS 1.0

Team Members: Daniel Perez, Armando Carrasquillo, Santiago Bolivar

Product Owner(s): Francisco Ortega

Mentor(s): Francisco Ortega

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

We propose a 3D environment in the form of a video game where the main idea is to increase Computer Science (CS) interest. We believe that by providing software that can be used by everyone, we can spark more interest in CS. We created a simple prototype emulating an Escape Room with the idea to attract individuals of any age range with a fun learning activity, but our primary focus is on teenagers and young adults. The puzzles in the game engage the player by giving them challenges that can be completed optimally by using computer science concepts. However, the game is presented as a typical puzzle game to avoid scaring away players who may have preconceived notions of computer science. The aim is to engage players through the puzzles to promote further interest in CS concepts.

Table of Contents

INTRODUCTION	5
CURRENT SYSTEM	5
PURPOSE OF NEW SYSTEM	5
USER STORIES	
IMPLEMENTED USER STORIES	6
PROJECT PLAN	
HARDWARE AND SOFTWARE RESOURCES	18
SPRINTS PLAN	19
<i>Sprint 1</i>	19
<i>Sprint 2</i>	20
<i>Sprint 3</i>	21
<i>Sprint 4</i>	22
<i>Sprint 5</i>	22
<i>Sprint 6</i>	23
SYSTEM DESIGN	
ARCHITECTURAL PATTERNS	25
SYSTEM AND SUBSYSTEM DECOMPOSITION	25
DEPLOYMENT DIAGRAM	26
DESIGN PATTERNS	26
SYSTEM VALIDATION	27
GLOSSARY	47
APPENDIX	48
APPENDIX A - UML DIAGRAMS	48
<i>Use Case Diagrams</i>	48
<i>Sequence Diagrams</i>	56
<i>Class Diagrams</i>	75
APPENDIX B - USER INTERFACE DESIGN	87
APPENDIX C - SPRINT REVIEW REPORTS	97
APPENDIX D - USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST DOCUMENT AND OTHER DOCUMENTS	101
REFERENCES	103

INTRODUCTION

The Computer Science (CS) major is high in demand but there's a well-documented lack of gender and racial diversity. Women, Blacks and Hispanics are underrepresented in obtaining CS degrees. Many of which steer away from CS due to its STEM nature, but not because of their ability to pursue the major. Educational institutions have attempted to make CS more appealing and accessible to these groups, yet the diversity gap endures. We propose Alchemist Escape, a full virtual environment video game with an “escape the room” theme. The game is designed to increase CS interest and awareness. Each room holds a basic concept of CS while the game itself maintains the entertainment factor of any other video game. This allows players who may have a misconception of CS to play the game, since no previous CS knowledge is required. Research will be done on the players performance to improve future iterations of the game. Also to quantify the efficacy of the software into bringing more interest to Computer Science.

Current System

Educational video games are developed to be instructional or tutorial based. Players who are interested in the subject tend to play these games, but they do not generate interest in the subject due to how they're perceived by many other players.

Purpose of New System

The purpose behind creating Alchemist Escape is to spark interest in Computer Science in an attempt to decrease the diversity gap. People are more open to play a game rather than attending a class. Moreover, by playing the game and then reinforcing the concepts of Computer Science, the user will be able to connect how everyday tasks can relate to concepts taught in Computer Science. The game includes a complete user interface and a total of three rooms that covers: If-Statements, Loops, and Arrays.

USER STORIES

The following section provides the detailed user stories that were implemented in this iteration of the Alchemist Escape project. These user stories served as the basis for the implementation of the project's features. This section also shows the user stories that are to be considered for future development.

Implemented User Stories

#696 Implement Function to Move Objects

Description

- As a developer, I want to implement a function for the user to move objects around in Unreal, so that I can avoid having to retype the code for grabbing and releasing objects.

Acceptance Criteria

- Verify that when an object is grabbed the object is being held until the user releases it.
- Verify that when an object is released the object is placed in the correct orientation.
- Verify that only one object can be moved at a time.

Use Case

- Name: Move Object
- Actor: User
- Preconditions: The object is a movable object.
- Description <Flow of events>:
 1. The user grabs an object.
 2. The user is able to move around with the held object.
 3. The user is able to release the object.

#697 Implement Adding Objects to Cauldron (Puzzle 1)

Description

- As a developer, I want to implement a function for the user to add potions to a cauldron in Unreal, so that the user can work on the puzzle.

Acceptance Criteria

- Verify that the function recognizes when objects are placed in the cauldron.
- Verify that the function recognizes the type of object that was added in the cauldron.
- Verify that the function is keeping record of the potions that are added.

- Verify that the function deletes the object after the object type was recognized.
- Verify that the function warns the user when an incorrect object is added in the cauldron.

Use Case

- Name: Place Object in Cauldron
- Actor: User
- Preconditions: The user is able to move objects around.
- Description <Flow of events>:
 1. The user grabs an object in the room.
 2. The user places the object in the cauldron.
 - a. If the object is not a potion, then warn the user.
 - b. If the potion was already added, then warn the user.
 - c. If the potion was not added, then record it.
 3. The object gets deleted from the cauldron.

#698 Implement Interaction to Heat Up Cauldron (Puzzle 1)

Description

- As a developer, I want to implement a function for the user to heat up the cauldron, so that the user can check if their solution to the current iteration was correct.

Acceptance Criteria

- Verify that the function only succeeds if the right combination of potions was added for the current iteration and that they're the only objects in the cauldron.
- Verify that the function notifies the user if their solution was correct.
- Verify that the iteration counter increments when the function succeeds.
- Verify that the function resets the potion flags after each iteration.

Use Case

- Name: Heat Up Cauldron
- Actor: User
- Preconditions: N/A
- Description <Flow of events>:
 1. The user presses a button to heat up the cauldron.
 2. The cauldron heats up.
 - a. If the solution is correct, then the user is notified and the iteration counter gets incremented.
 - b. Reset all the potion flags.
 3. Repeat steps 1 and 2 until the iteration counter equals the number of required successes.

#699 Implement Adding Objects to Scale (Puzzle 2)

Description

- As a developer, I want to implement a function for the user to add apples to a scale in Unreal, so that the user can work on the puzzle.

Acceptance Criteria

- Verify that the function recognizes when objects are placed on the scale.
- Verify that the function recognizes which type of objects are placed on the scale.
- Verify that the function is keeping one record for the objects placed on the left side of the scale and one record for the objects placed on the right side of the scale.
- Verify that the function warns the user when an incorrect object is placed on the scale and then remove that object from the scale.
- Verify that multiple objects can be placed on the scale.

Use Case

- Name: Place Object on Scale
- Actor: User
- Preconditions: The user is able to move objects around and there's available space on the scale to place objects.
- Description <Flow of events>:
 1. The user grabs an object in the room.
 2. The user places the object on the scale.
 3. The scale checks what type of object was placed.
 - a. If the object is not an apple, then warn the user and remove the object from the scale.

#700 Implement Interaction to Display Objects Weight (Puzzle 2)

Description

- As a developer, I want to implement a function to show the user which side of the scale has more weight, so that the user can collect the necessary information to solve the puzzle.

Acceptance Criteria

- Verify that the function succeeds if and only if apples are placed on the scale.
- Verify that the function notifies the user if the scale was activated with no apples placed on it.
- Verify that the function is able to read the weight of each apple and that the sum of apples on each side of the scale has the correct weight.

- Verify that the function plays an animation to notify the user that the scale is working when the user activates the scale to get the weight.
- Verify that the scale either tilts to one side or remains balanced depending on the weight when the user activates the scale.
- Verify that the function keeps record of how many times the scale was used.

Use Case

- Name: Display Weight of Objects
- Actor: User
- Preconditions: Objects must be placed on the scale and objects consist only of apples.
- Description <Flow of events>:
 1. The user activates the scale.
 2. The scale calculates the weight of the apples on each side.
 - a. If no apples are placed on the scale, then notify the user.
 3. An animation is played to notify the user that the scale is working.
 4. The scale displays the weight of the apples placed on it by either tilting to one side or by remaining balanced.
 5. The counter for the number of times the scale was used is incremented.

#701 Implement Placing Books on Shelf (Puzzle 3)

Description

- As a developer, I want to implement a function for the user to place books on a shelf in Unreal, so that the user can work on the puzzle.

Acceptance Criteria

- Verify that the function recognizes when books are placed on the shelf.
- Verify that the function is keeping record of the book's location on the shelf.
- Verify that the function checks the users solution once all the books have been placed on the shelf.
- Verify that once the puzzle is solved the user is notified and the system flags puzzle 3 as solved.

Use Case

- Name: Place Book on Shelf
- Actor: User
- Preconditions: The user is able to move objects around and only books are movable.
- Description <Flow of events>:
 1. The user grabs a book in the room.
 2. The user places the book on the shelf.

3. The system verifies the location of the book on the shelf.
4. After all the books have been placed on the shelf, check the user's solution.
5. An animation is played to notify the user of their current status.
6. Once the user solves the puzzle, notify the user and update the status for puzzle 3 in the system.

#702 Implement Interaction to Reset Puzzle (Puzzle 3)

Description

- As a developer, I want to implement a function for the user to reset Puzzle 3 in Unreal, so that if the user gets the incorrect solution they can easily restart the puzzle.

Acceptance Criteria

- Verify that the puzzle gets reset only when requested by the user.
- Verify that when the function is executed the books on the shelf automatically get moved back to their initial location.
- Verify that the function is keeping record of how many times the user reseted the puzzle.

Use Case

- Name: Reset Puzzle
- Actor: User
- Preconditions: N/A
- Description <Flow of events>:
 1. The user presses a button to reset the puzzle.
 2. All of the books get moved back to their initial location.
 3. The counter for the number of resets is incremented.

#723 Create Interactive Button

Description

- As a developer, I want to implement an interactive button in Unreal, so that the user can initiate a command on call.

Acceptance Criteria

- Verify that the interactive button can be assigned to any object in Unreal.
- Verify that when the user is within range of the button a message pops up informing the user what to press in order to interact with the button.
- Verify that the user can only interact with the button when within range.
- Verify that the command gets initiated when the user presses the correct key.
- Verify that the proper function is executed when the user initiates the command.

Use Case

- Name: Interactive Button
- Actor: User
- Preconditions: Interactive button must be assigned to an object.
- Description <Flow of events>:
 1. The user moves within range of the button.
 2. A message pops up informing the user what to press in order to interact with the button.
 - a. If the user moves out of range the message disappears.
 3. The user interacts with the button only if within range.
 4. The command gets initiated.
 5. The proper function gets executed.

#724 Implement Functionality for Doors

Description

- As a developer, I want to implement the functionality of a door in Unreal, so that the door can open and close on command.

Acceptance Criteria

- Verify that the door opens on command.
- Verify that the door closes on command.
- Verify that the functionality of the door remains when creating a copy of the door.

Use Case

- Name: Door Blueprint
- Actor: Developer
- Preconditions: Door object must be present in level.
- Description <Flow of events>:
 1. The door opens when a specific event occurs.
 2. The door closes when a specific event occurs.

#725 Implement HUD

Description

- As a developer, I want to implement a Heads Up Display (HUD), so that specific information is displayed to the user at all times.

Acceptance Criteria

- Verify that the HUD displays the total number of levels that the user has to complete.

- Verify that the HUD displays the current level the user is on.
- Verify that the HUD displays the achievement icons.
- Verify that the HUD displays the crosshair.

Use Case

- Name: HUD
- Actor: User
- Preconditions: Game must have at least one level.
- Description <Flow of events>:
 1. The user starts the game.
 2. Loop:
 - a. The HUD displays the current level the user is on.

#726 Implement Conditions to Escape Room (Puzzle 2)

Description

- As a developer, I want the user to place the bad apple they found on one plate and only one good apple on the other plate, so that the user can escape the room if those two conditions are met.

Acceptance Criteria

- Verify that the function recognizes the type of apple placed on each plate.
- Verify that the puzzle is solved if and only if the bad apple is on one plate and only one good apple is on the other plate.
- Verify that the system is notified when the puzzle is solved.
- Verify that the door opens when the puzzle is solved.
- Verify that the door stays open as long as the puzzle remains solved.

Use Case

- Name: Escape Room Puzzle 2
- Actor: User
- Preconditions: The user is able to move apples around the room.
- Description <Flow of events>:
 1. The user places one apple on each plate.
 2. Check the users solution.
 - a. If the bad apple is on one plate and only one good apple is on the other plate, then the puzzle is solved and the system is notified.
 - i. The door opens, allowing the user to escape the room.
 - ii. If the user changes their solution, then the door closes.
 - b. If the door remains closed, then the user must try a new solution.

#727 Implement Achievement System

Description

- As a developer, I want to implement an achievement system for the game, so that the user gets notified when a puzzle has been solved.

Acceptance Criteria

- Verify that the achievement is shown on screen when a puzzle is solved.
- Verify that the achievement can toggle the achievement menu option.
- Verify that the HUD is updated after the achievement has been unlocked.
- Verify that the menu is updated after the achievement has been unlocked.

Use Case

- Name: Achievement System
- Actor: User
- Preconditions: An achievement is unlocked.
- Description <Flow of events>:
 1. The achievement is shown on screen.
 - a. The user has an option to immediately view information about the achievement in the menu.
 2. The achievement icon in the HUD is updated.
 - a. The icon color changes from grey to gold.
 3. The menu is updated.
 - a. The achievement is now viewable from the menu at any time.

#728 Implement In-Game Menu

Description

- As a developer, I want to implement an in-game menu, so that the user can access help and information when needed.

Acceptance Criteria

- Verify that the following options are available: resume, controls, hints, achievements, and quit game.
- Verify that the resume option exits the in-game menu.
- Verify that the controls option displays a guide on how to play the game.
- Verify that the hints option displays a hint on how to solve the current puzzle the user is on.
- Verify that the achievements option displays all of the achievements by completion.
- Verify that the quit game option exits the game.

Use Case

- Name: In-Game Menu
- Actor: User
- Preconditions: The user started the game.
- Description <Flow of events>:
 1. The user presses the menu button.
 2. The in-game menu is displayed on screen.
 - a. If the user selects an option (other than resume or quit game) from the menu, then that option is displayed.
 - b. If the user presses the resume option, then the in-game menu is taken off screen.
 - c. If the user presses the quit game option, then the user is returned to the main menu.

#750 Collect Metrics

Description

- As a developer, I want to collect metrics for research purposes, so that relevant information can be extracted from the players data.

Acceptance Criteria

- Verify that the metrics start collecting data when the player clicks on start game.
- Verify that a unique player id is generated for each player.
- Verify that the date, game started time and game ended time is collected.
- Verify that for each room the following is collected: time room was entered, time puzzle was started, time puzzle was solved, time room was exited, and the number of attempts made to solve the puzzle.
- Verify that the players location and rotation is being collected every second.

Use Case

- Name: Collect Metrics
- Actor: Developer
- Preconditions: Player started the game.
- Description <Flow of events>:
 1. The date and game started time is collected.
 2. A unique player id is generated.
 3. The players location and rotation is collected every second.
 4. For each room the following is collected:
 - a. The time the room was entered.
 - b. The time the puzzle was started.

- c. The time the puzzle was solved.
 - d. The number of attempts made to solve the puzzle.
 - e. The time the room was excited.
5. The game ended time is collected.
 6. The players location and rotation stops being collected.

#751 Print Metrics

Description

- As a developer, I want to export the metrics to both a JSON and text file, so that researchers can view the collected data and parse the data as needed.

Acceptance Criteria

- Verify that all of the metrics except for the players path is appended to a text file, with commas separating the data and a new line separating each game.
- Verify that all of the metrics for each game is appended to a JSON file as an object.

Use Case

- Name: Print Metrics
- Actor: Developer
- Preconditions: The game is ended.
- Description <Flow of events>:
 1. Metrics are printed to a text file.
 2. Metrics are printed to a JSON file.

#752 Implement Main Menu

Description

- As a developer, I want to implement a main menu, so that the user can start a new game or view the credits.

Acceptance Criteria

- Verify that the following options are available: start game, credits, and quit game.
- Verify that the start game option loads a new game.
- Verify that the credits option displays the credits for the game.
- Verify that the quit game option turns off the game.

Use Case

- Name: Main Menu
- Actor: User

- Preconditions: The game is turned on.
- Description <Flow of events>:
 1. The user selects an option.
 - a. If the start game option is selected, then a new game is loaded.
 - b. If the credits option is selected, then the credits are displayed.
 - c. If the quit game option is selected, then the game is turned off.

#753 Create Enter Room Trigger

Description

- As a developer, I want to create a trigger for when the player enters a room, so that the proper events take place and the system gets updated with the current status.

Acceptance Criteria

- Verify that the trigger can only be activated once.
- Verify that when the trigger is activated the time is collected for the entered room metric.
- Verify that the system is updated with the players current level.
- Verify that the puzzle introduction is displayed.

Use Case

- Name: Enter Room Trigger
- Actor: User
- Preconditions: Player enters a new room in the game.
- Description <Flow of events>:
 1. The time is collected for the entered room metric.
 2. The system is updated with the players current level.
 3. The puzzle introduction is displayed.

#754 Create Game Completed Trigger

Description

- As a developer, I want to create a trigger for the player to complete the game, so that the proper events take place.

Acceptance Criteria

- Verify that when the trigger is activated the game is ended.
- Verify that the congratulation screen is displayed to the player.
- Verify that the player is returned to the main menu after viewing the congratulations.

Use Case

- Name: Game Completed Trigger
- Actor: User
- Preconditions: The game is completed.
- Description <Flow of events>:
 1. The game is ended.
 2. The congratulation screen is displayed to the player.
 3. The player is returned to the main menu after viewing the congratulations.

#755 Highlight Movable Objects

Description

- As a developer, I want to highlight movable objects, so that the player can distinguish which objects they can interact with and move around.

Acceptance Criteria

- Verify that movable objects are highlighted when the player is within range.
- Verify that movable objects are not highlighted when the player is out of range.

Use Case

- Name: Highlight Objects
- Actor: User
- Preconditions: The object is movable.
- Description <Flow of events>:
 1. If the player is within range, then the object gets highlighted.
 2. If the player is not within range, then the object is not highlighted.

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Hardware:

- HTC Vive
- Lenovo Legion Y720 Laptop
 - Intel Core i7-7700HQ CPU 2.80 GHz
 - 16GB RAM
 - NVIDIA GeForce GTX 1060
 - 6GB VRAM

Software:

- Unreal Engine 4 (version 4.18.3)
- Visual Studio 2017
- Blender
- JSON
- C++
- Lucidchart
- Git
- Google Drive
- Mingle
- Windows 10

Sprints Plan

Sprint 1

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 8:00 PM

End time: 8:30 PM

After discussion, the velocity of the team were estimated to be 89.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story #667 Setup Unreal Engine and Visual Studio (Computer 1)
- User Story #668 Setup Unreal Engine and Visual Studio (Computer 2)
- User Story #669 Setup Unreal Engine and Visual Studio (Computer 3)
- User Story #670 Research Unreal Engine Documentation (Developer 1)
- User Story #671 Research Unreal Engine Documentation (Developer 2)
- User Story #672 Research Unreal Engine Documentation (Developer 3)
- User Story #673 Research the C++ API (Developer 1)
- User Story #674 Research the C++ API (Developer 2)
- User Story #675 Research the C++ API (Developer 3)
- User Story #676 Research Game's that Women Play
- User Story #677 Setup VR Template Level
- User Story #678 Setup VR Headset with Unreal Engine

The team members indicated their willingness to work on the following user stories.

- Armado Carrasquillo
 - User Story #667 Setup Unreal Engine and Visual Studio (Computer 1)
 - User Story #670 Research Unreal Engine Documentation (Developer 1)
 - User Story #673 Research the C++ API (Developer 1)
 - User Story #676 Research Game's that Women Play
- Daniel Perez
 - User Story #668 Setup Unreal Engine and Visual Studio (Computer 2)
 - User Story #671 Research Unreal Engine Documentation (Developer 2)
 - User Story #674 Research the C++ API (Developer 2)

- User Story #677 Setup VR Template Level
- Santiago Bolivar
 - User Story #669 Setup Unreal Engine and Visual Studio (Computer 3)
 - User Story #672 Research Unreal Engine Documentation (Developer 3)
 - User Story #675 Research the C++ API (Developer 3)
 - User Story #678 Setup VR Headset with Unreal Engine

Sprint 2

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After discussion, the velocity of the team were estimated to be 72.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story #679 Setup Level Streaming in Unreal Engine
- User Story #680 Split Level in Unreal into Sub-Levels
- User Story #681 Create Survey for Game
- User Story #682 Research Puzzles and Storyline for Game (Developer 1)
- User Story #683 Research Puzzles and Storyline for Game (Developer 2)
- User Story #684 Research Puzzles and Storyline for Game (Developer 3)
- User Story #685 Add Functionality to Objects in Unreal Engine (Developer 1)
- User Story #686 Add Functionality to Objects in Unreal Engine (Developer 2)
- User Story #687 Collect Data from Survey

The team members indicated their willingness to work on the following user stories.

- Armando Carrasquillo
 - User Story #679 Setup Level Streaming in Unreal Engine
 - User Story #682 Research Puzzles and Storyline for Game (Developer 1)
 - User Story #685 Add Functionality to Objects in Unreal Engine (Developer 1)
- Daniel Perez
 - User Story #680 Split Level in Unreal into Sub-Levels
 - User Story #683 Research Puzzles and Storyline for Game (Developer 2)
 - User Story #686 Add Functionality to Objects in Unreal Engine (Developer 2)

- Santiago Bolivar
 - User Story #681 Create Survey for Game
 - User Story #684 Research Puzzles and Storyline for Game (Developer 3)
 - User Story #687 Collect Data from Survey

Sprint 3

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After discussion, the velocity of the team were estimated to be 88.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story #696 Implement Function to Move Objects
- User Story #697 Implement Adding Objects to Cauldron (Puzzle 1)
- User Story #698 Implement Interaction to Heat Up Cauldron (Puzzle 1)
- User Story #699 Implement Adding Objects to Scale (Puzzle 2)
- User Story #700 Implement Interaction to Display Objects Weight (Puzzle 2)
- User Story #701 Implement Placing Books on Shelf (Puzzle 3)
- User Story #702 Implement Interaction to Reset Puzzle (Puzzle 3)

The team members indicated their willingness to work on the following user stories.

- Armando Carrasquillo
 - User Story #697 Implement Adding Objects to Cauldron (Puzzle 1)
 - User Story #698 Implement Interaction to Heat Up Cauldron (Puzzle 1)
- Daniel Perez
 - User Story #699 Implement Adding Objects to Scale (Puzzle 2)
 - User Story #700 Implement Interaction to Display Objects Weight (Puzzle 2)
- Santiago Bolivar
 - User Story #696 Implement Function to Move Objects
 - User Story #701 Implement Placing Books on Shelf (Puzzle 3)
 - User Story #702 Implement Interaction to Reset Puzzle (Puzzle 3)

Sprint 4

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After discussion, the velocity of the team were estimated to be 96.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story #723 Create Interactive Button
- User Story #724 Implement Functionality for Doors
- User Story #725 Implement HUD
- User Story #726 Implement Conditions to Escape Room (Puzzle 2)
- User Story #727 Implement Achievement System
- User Story #728 Implement In-Game Menu

The team members indicated their willingness to work on the following user stories.

- Armando Carrasquillo
 - User Story #725 Implement HUD
 - User Story #728 Implement In-Game Menu
- Daniel Perez
 - User Story #723 Create Interactive Button
 - User Story #726 Implement Conditions to Escape Room (Puzzle 2)
- Santiago Bolivar
 - User Story #724 Implement Functionality for Doors
 - User Story #727 Implement Achievement System

Sprint 5

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After discussion, the velocity of the team were estimated to be 104.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story #750 Collect Metrics
- User Story #751 Print Metrics
- User Story #752 Implement Main Menu
- User Story #753 Create Enter Room Trigger
- User Story #754 Create Game Completed Trigger
- User Story #755 Highlight Movable Objects

The team members indicated their willingness to work on the following user stories.

- Armando Carrasquillo
 - User Story #752 Implement Main Menu
 - User Story #754 Create Game Completed Trigger
- Daniel Perez
 - User Story #750 Collect Metrics
 - User Story #753 Create Enter Room Trigger
- Santiago Bolivar
 - User Story #751 Print Metrics
 - User Story #755 Highlight Movable Objects

Sprint 6

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After discussion, the velocity of the team were estimated to be 120.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story #775 Write Research Paper for VIP (Developer 1)
- User Story #776 Write Research Paper for VIP (Developer 2)
- User Story #777 Write Research Paper for VIP (Developer 3)
- User Story #778 Port Game to VR (Developer 1)
- User Story #779 Port Game to VR (Developer 2)
- User Story #780 Port Game to VR (Developer 3)

- User Story #781 Alpha Testing (Developer 1)
- User Story #782 Alpha Testing (Developer 2)
- User Story #783 Alpha Testing (Developer 3)

The team members indicated their willingness to work on the following user stories.

- Armando Carrasquillo
 - User Story #775 Write Research Paper for VIP (Developer 1)
 - User Story #778 Port Game to VR (Developer 1)
 - User Story #781 Alpha Testing (Developer 1)
- Daniel Perez
 - User Story #776 Write Research Paper for VIP (Developer 2)
 - User Story #779 Port Game to VR (Developer 2)
 - User Story #782 Alpha Testing (Developer 2)
- Santiago Bolivar
 - User Story #777 Write Research Paper for VIP (Developer 3)
 - User Story #780 Port Game to VR (Developer 3)
 - User Story #783 Alpha Testing (Developer 3)

SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

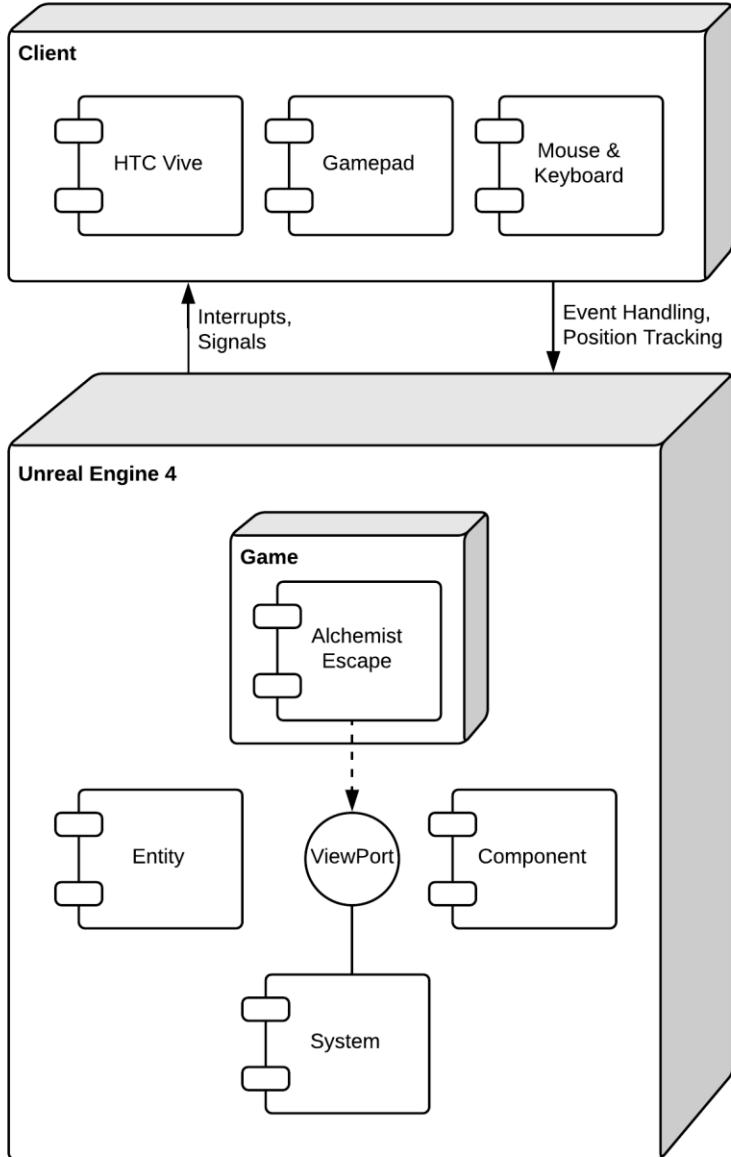
Architectural Patterns

Alchemist Escape was developed under the Entity-Component-System (ECS) architecture pattern. This type of architecture specializes in the creation and recognition of objects in the users viewport. Any function required for the program falls into the System level. These functions manage and control the components in the game. The collection of all these components sets the entity being processed by the system.

System and Subsystem Decomposition

- Entities, a set of components that are grouped on a common purpose, require the system to interact with them and generate events from users interaction.
 - One of the main entities present on Alchemist Escape is the player entity.
- Components are the different parts of the game that hold information for the system to interact with. It is important to denote that no logic is encapsulated on any of the components.
 - Some of the relevant components to mention are movement, rotation, location, physics, player input, and movable objects.
- The system is the logic and the functionality required for entities to work. On ECS, the system iterates through a group of entities that share components.
 - Alchemist Escape uses systems such as the motion system, render system, object movement system, player input system, notification system, and animation system.

Deployment Diagram



Design Patterns

There were no commonly occurring problems to create a general reusable solution.

SYSTEM VALIDATION

Unit Tests

User Story #696 Implement Function to Move Objects

- **Test case ID:** get_physics_body_in_reach
 - Description/Summary of Test: The system checks if there is a movable object in reach.
 - Pre-condition: User is located in front of an movable object.
 - Expected Results: User gets notified that object is movable.
 - Actual Result: User gets notified that object is movable.
 - Status (Fail/Pass): Pass
- **Test case ID:** grab_object
 - Description/Summary of Test: The system maintains object location and rotation in front of the user.
 - Pre-condition: User has clicked on movable object.
 - Expected Results: Object location and rotation is held in front of camera.
 - Actual Result: Object location and rotation is held in front of camera.
 - Status (Fail/Pass): Pass
- **Test case ID:** release_object
 - Description/Summary of Test: The system frees the object location and rotation from in front of user.
 - Pre-condition: User is currently holding object.
 - Expected Results: Object releases on the current location.
 - Actual Result: Object releases on the current location.
 - Status (Fail/Pass): Pass

User Story #697 Implement Adding Objects to Cauldron (Puzzle 1)

- **Test case ID:** identify_object_p1
 - Description/Summary of Test: Cauldron identifies the type of object that is placed within it.
 - Pre-condition: User places object in cauldron.
 - Expected Results: Identifies if the object is a potion, a player, or a miscellaneous object.

- Actual Result: Identified if the object is a potion, a player, or a miscellaneous object.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** record_potion_p1
 - Description/Summary of Test: Keeps track if a necessary potion object was placed in the cauldron.
 - Pre-condition: Place a potion object in the cauldron.
 - Expected Results: If a red or blue potion object is placed in the cauldron. The boolean for the respective potion is set to true.
 - Actual Result: When a red or blue potion object is placed in the cauldron. The boolean for the respective potion is set to true.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** delete_object_p1
 - Description/Summary of Test: Deletes the object after the necessary action has been done.
 - Pre-condition: Object has been identified and set to be deleted.
 - Expected Results: Deletes the object.
 - Actual Result: The object is deleted.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** error_msg_p1
 - Description/Summary of Test: Warns user if the user placed a miscellaneous object in the cauldron.
 - Pre-condition: The user placed a miscellaneous object in the cauldron.
 - Expected Results: An error message is displayed for the user letting them know that they placed an incorrect object.
 - Actual Result: An error message is displayed for the user letting them know that they placed an incorrect object.
 - Status (Fail/Pass): Pass

User Story #698 Implement Interaction to Heat Up Cauldron (Puzzle 1)

- **Test case ID:** check_solution_p1
- Description/Summary of Test: Checks to see if the correct potions were placed in the Cauldron.

- Pre-condition: Activate Heat Up button.
 - Expected Results: Checks to see if the correct potions were placed.
 - Actual Result: Checks to see if the correct potions were placed.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** notify_result_p1
 - Description/Summary of Test: Changes the particle effect on the Cauldron.
 - Pre-condition: Solution within cauldron is correct.
 - Expected Results: Changes particle effect within cauldron.
 - Actual Result: Particle effect changed.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** increment_counter_p1
 - Description/Summary of Test: Increments counter of successful solutions.
 - Pre-condition: A successful solution was made, and the particle effect changed.
 - Expected Results: The counter of total solutions is increased.
 - Actual Result: The counter of total solutions is increased.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** reset_flags_p1
 - Description/Summary of Test: Resets the potion flags.
 - Pre-condition: The solution was completed, and the next round of solutions began.
 - Expected Results: The potion flags should be set to false.
 - Actual Result: Potion flags set to false.
 - Status (Fail/Pass): Pass

User Story #699 Implement Adding Objects to Scale (Puzzle 2)

- **Test case ID:** identify_object_p2
- Description/Summary of Test: The scale identifies which object was placed on it.
- Pre-condition: The user places an object on the scale.
- Expected Results: Scale identifies the name of the object.
- Actual Result: Scale identified the name of the object.

- Status (Fail/Pass): Pass
- **Test case ID:** get_actors_left
 - Description/Summary of Test: The system checks which objects are placed on the left side of the scale and returns a list of those objects.
 - Pre-condition: Objects must be placed on the scale.
 - Expected Results: A list of objects placed on the left side of the scale.
 - Actual Result: A list of objects placed on the left side of the scale.
 - Status (Fail/Pass): Pass
- **Test case ID:** get_actors_right
 - Description/Summary of Test: The system checks which objects are placed on the right side of the scale and returns a list of those objects.
 - Pre-condition: Objects must be placed on the scale.
 - Expected Results: A list of objects placed on the right side of the scale.
 - Actual Result: A list of objects placed on the right side of the scale.
 - Status (Fail/Pass): Pass
- **Test case ID:** remove_object_p2
 - Description/Summary of Test: The object is removed from the scale only if the object is not an apple.
 - Pre-condition: The user places a non-apple object on the scale.
 - Expected Results: The non-apple object gets removed from the scale.
 - Actual Result: The non-apple object got removed from the scale.
 - Status (Fail/Pass): Pass
- **Test case ID:** error_msg_p2
 - Description/Summary of Test: The user gets a notification for placing a non-apple object on the scale.
 - Pre-condition: The user places a non-apple object on the scale.
 - Expected Results: The user receives the notification.
 - Actual Result: The user received the notification.
 - Status (Fail/Pass): Pass

User Story #700 Implement Interaction to Display Objects Weight (Puzzle 2)

- **Test case ID:** check_for_actors
 - Description/Summary of Test: The scale must have apples placed on it before the scale is activated. If no apples are on the scale, then the user is notified that they must place an apple on the scale before activating it.
 - Pre-condition: The user activates the scale.
 - Expected Results: The scale displays the weight on a successful event, otherwise the user is notified.
 - Actual Result: The scale displayed the weight on a successful event and the user was notified on an unsuccessful event.
 - Status (Fail/Pass): Pass
- **Test case ID:** get_weight_left
 - Description/Summary of Test: The weight of each apple on the left side of the scale is added to get the total sum of weight.
 - Pre-condition: Apples must be placed on the left side of the scale.
 - Expected Results: The total sum of weight of all the apples on the left side of the scale.
 - Actual Result: The total sum of weight of all the apples on the left side of the scale.
 - Status (Fail/Pass): Pass
- **Test case ID:** get_weight_right
 - Description/Summary of Test: The weight of each apple on the right side of the scale is added to get the total sum of weight.
 - Pre-condition: Apples must be placed on the right side of the scale.
 - Expected Results: The total sum of weight of all the apples on the right side of the scale.
 - Actual Result: The total sum of weight of all the apples on the right side of the scale.
 - Status (Fail/Pass): Pass
- **Test case ID:** scale_animation
 - Description/Summary of Test: An animation is played to notify the user that the scale is working.
 - Pre-condition: The user activates the scale and apples are placed on the scale.
 - Expected Results: The animation is played.
 - Actual Result: The animation is played.

- Status (Fail/Pass): Pass
- **Test case ID:** display_weight
- Description/Summary of Test: The scale either tilts to one side or remains balanced depending on the weight when the user activates the scale.
- Pre-condition: The user activates the scale and apples are placed on the scale.
- Expected Results: The scale either tilts to one side or remains balanced.
- Actual Result: When the left side had more weight the scale tilted to the left; when the right side had more weight the scale tilted to the right; when both sides had equal weight the scale remained balanced.
- Status (Fail/Pass): Pass
- **Test case ID:** increment_counter_p2
- Description/Summary of Test: A counter for the number of times the scale was used is incremented when the user successfully activates the scale.
- Pre-condition: The user activates the scale and apples are placed on the scale.
- Expected Results: The counter increments.
- Actual Result: The counter incremented.
- Status (Fail/Pass): Pass

User Story #701 Implement Placing Books on Shelf (Puzzle 3)

- **Test case ID:** verify_trigger_volume_is_assigned
- Description/Summary of Test: Makes sure there the trigger volume is not null.
- Pre-condition: N/A
- Expected Results: System notifies user that trigger volume is not active.
- Actual Result: System notifies user that trigger volume is not active.
- Status (Fail/Pass): Pass
- **Test case ID:** verify_trigger_volume_is_pressed
- Description/Summary of Test: Makes sure there is an object in the trigger volume.
- Pre-condition: Trigger volume is created and active.
- Expected Results: System notifies user that trigger volume is pressed.
- Actual Result: System notifies user that trigger volume is pressed.

- Status (Fail/Pass): Pass
- **Test case ID:** verify_correct_book
- Description/Summary of Test: Makes sure the current placed book matches the trigger volume.
- Pre-condition: Book has been placed on the trigger volume.
- Expected Results: System notifies correct book has been placed.
- Actual Result: System notifies correct book has been placed.
- Status (Fail/Pass): Pass
- **Test case ID:** verify_incorrect_book
- Description/Summary of Test: Notify user in case the current placed book does not match the trigger volume.
- Pre-condition: Book has been placed on the trigger volume.
- Expected Results: System notifies incorrect book has been placed.
- Actual Result: System notifies incorrect book has been placed.
- Status (Fail/Pass): Pass
- **Test case ID:** play_animation
- Description/Summary of Test: Displays animation of success or fail to user once all trigger volumes have been pressed.
- Pre-condition: All trigger volumes have been activated.
- Expected Results: System notifies user by spawning a particle system animation.
- Actual Result: System notifies user by spawning a particle system animation.
- Status (Fail/Pass): Pass

User Story #702 Implement Interaction to Reset Puzzle (Puzzle 3)

- **Test case ID:** reset_puzzle
- Description/Summary of Test: Allow the user to restart the puzzle.
- Pre-condition: N/A
- Expected Results: Books are sent back to table for user to start the puzzle again.
- Actual Result: Books are sent back to table for user to start the puzzle again.
- Status (Fail/Pass): Pass

- **Test case ID:** increment_counter
 - Description/Summary of Test: Every time that the user resets the puzzle, counter is increased.
 - Pre-condition: N/A
 - Expected Results: Counter increases once the puzzle has been reseted.
 - Actual Result: Counter increased once the puzzle has been reseted.
 - Status (Fail/Pass): Pass

- **Test case ID:** move_books
 - Description/Summary of Test: After user resets the puzzle all the book are moved to their original location.
 - Pre-condition: User presses reset button
 - Expected Results: Books are moved from the shelf to the table.
 - Actual Result: Books are moved from the shelf to the table.
 - Status (Fail/Pass): Pass

User Story #723 Create Interactive Button

- **Test case ID:** assign_button
 - Description/Summary of Test: Interactive button can be assigned to any object in Unreal.
 - Pre-condition: Object must exist in level.
 - Expected Results: The object gains the functionality of the interactive button.
 - Actual Result: The object gained the functionality of the interactive button.
 - Status (Fail/Pass): Pass

- **Test case ID:** toggle_message_on
 - Description/Summary of Test: A message is shown when the user is within range of the button.
 - Pre-condition: The user is within range.
 - Expected Results: Message is shown on screen.
 - Actual Result: Message is shown on screen.
 - Status (Fail/Pass): Pass

- **Test case ID:** toggle_message_off
 - Description/Summary of Test: The message is hidden when the user is not within range of the button.
 - Pre-condition: The user is not within range.
 - Expected Results: Message is hidden.
 - Actual Result: Message is hidden.
 - Status (Fail/Pass): Pass
- **Test case ID:** press_button_1
 - Description/Summary of Test: The user can only interact with the button when within range.
 - Pre-condition: The user is within range and the correct key is pressed.
 - Expected Results: The command gets initiated.
 - Actual Result: The command got initiated.
 - Status (Fail/Pass): Pass
- **Test case ID:** press_button_2
 - Description/Summary of Test: The user can only interact with the button when within range.
 - Pre-condition: The user is not within range and the correct key is pressed.
 - Expected Results: Nothing happens.
 - Actual Result: Nothing happened.
 - Status (Fail/Pass): Pass
- **Test case ID:** execute_function
 - Description/Summary of Test: The proper function is executed when the user initiates the command.
 - Pre-condition: The user is within range and the correct key is pressed.
 - Expected Results: The function gets executed.
 - Actual Result: The function got executed.
 - Status (Fail/Pass): Pass

User Story #724 Implement Functionality for Doors

- **Test case ID:** door_open
 - Description/Summary of Test: The door opens on command.

- Pre-condition: Door must exist in level.
 - Expected Results: Door opens.
 - Actual Result: Door opened.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** door_close
 - Description/Summary of Test: The door closes on command.
 - Pre-condition: Door must exist in level.
 - Expected Results: Door closes.
 - Actual Result: Door closed.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** door_functionality
 - Description/Summary of Test: The functionality of the door remains when creating a copy of the door.
 - Pre-condition: Door must exist in level.
 - Expected Results: The door copy retains the functionality of the door.
 - Actual Result: The door copy retained the functionality of the door.
 - Status (Fail/Pass): Pass

User Story #725 Implement HUD

- **Test case ID:** total_levels
 - Description/Summary of Test: The HUD displays the total number of levels that the user has to complete.
 - Pre-condition: Game must have at least one level.
 - Expected Results: HUD displays the total number of levels.
 - Actual Result: HUD displayed the total number of levels.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** current_level
 - Description/Summary of Test: The HUD displays the current level the user is on.
 - Pre-condition: Game must have at least one level.
 - Expected Results: HUD displays the current level.

- Actual Result: HUD displayed the current level.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** achievement_icons
 - Description/Summary of Test: The HUD displays the achievement icons.
 - Pre-condition: Game must have at least one achievement.
 - Expected Results: HUD displays the achievement icons.
 - Actual Result: HUD displayed the achievement icons.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** crosshair
 - Description/Summary of Test: The HUD displays the crosshair.
 - Pre-condition: N/A
 - Expected Results: HUD displays the crosshair.
 - Actual Result: HUD displayed the crosshair.
 - Status (Fail/Pass): Pass

User Story #726 Implement Conditions to Escape Room (Puzzle 2)

- **Test case ID:** identify_apple
 - Description/Summary of Test: The plates identify the type of apple placed on it.
 - Pre-condition: The user places an apple on the plates.
 - Expected Results: Plates identify the type of apple.
 - Actual Result: Plates identified the type of apple.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** check_solution_p2
 - Description/Summary of Test: The puzzle is solved if and only if the bad apple is on one plate and only one good apple is on the other plate.
 - Pre-condition: The user places an apple on each plate.
 - Expected Results: A boolean data type.
 - Actual Result: A boolean data type.
 - Status (Fail/Pass): Pass

- **Test case ID:** notify_system_p2
- Description/Summary of Test: The system is notified when the puzzle is solved.
- Pre-condition: The user places the bad apple on one plate and one good apple on the other plate.
- Expected Results: System is notified.
- Actual Result: System is notified.
- Status (Fail/Pass): Pass

- **Test case ID:** door_open_p2
- Description/Summary of Test: The door opens when the puzzle is solved.
- Pre-condition: Puzzle is solved.
- Expected Results: Door opens.
- Actual Result: Door opened.
- Status (Fail/Pass): Pass

- **Test case ID:** door_close_p2
- Description/Summary of Test: The door closes when the puzzle is not solved.
- Pre-condition: Puzzle is not solved.
- Expected Results: Door closes.
- Actual Result: Door closed.
- Status (Fail/Pass): Pass

User Story #727 Implement Achievement System

- **Test case ID:** show_achievement
- Description/Summary of Test: The achievement is shown on screen.
- Pre-condition: A puzzle was solved.
- Expected Results: The achievement is shown on screen.
- Actual Result: The achievement is shown on screen.
- Status (Fail/Pass): Pass

- **Test case ID:** toggle_achievement_menu
- Description/Summary of Test: The achievement can toggle the achievement menu option.
- Pre-condition: An achievement was shown on screen and the user accessed it.

- Expected Results: Display the achievement option in the menu.
 - Actual Result: Displayed the achievement option in the menu.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** update_achievement_icon
 - Description/Summary of Test: The achievement icon in the HUD is updated.
 - Pre-condition: An achievement was unlocked.
 - Expected Results: The icon color changes from grey to gold.
 - Actual Result: The icon color changed from grey to gold.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** update_achievement_menu
 - Description/Summary of Test: The achievement option in the menu is updated.
 - Pre-condition: An achievement was unlocked.
 - Expected Results: The achievement is viewable from the menu at any time.
 - Actual Result: The achievement is viewable from the menu at any time.
 - Status (Fail/Pass): Pass

User Story #728 Implement In-Game Menu

- **Test case ID:** menu_options
 - Description/Summary of Test: The following options are shown in the menu: resume, controls, hints, achievements, and quit game.
 - Pre-condition: User opened the menu.
 - Expected Results: The menu options are shown.
 - Actual Result: The menu options are shown.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** controls_option
 - Description/Summary of Test: The controls option displays a guide on how to play the game.
 - Pre-condition: User selected controls option.
 - Expected Results: The controls are displayed.
 - Actual Result: The controls are displayed.

- Status (Fail/Pass): Pass
- **Test case ID:** hints_option
 - Description/Summary of Test: The hints option displays a hint on how to solve the current puzzle the user is on.
 - Pre-condition: User selected hints option.
 - Expected Results: A hint is displayed.
 - Actual Result: A hint is displayed.
 - Status (Fail/Pass): Pass
- **Test case ID:** achievements_option
 - Description/Summary of Test: The achievements option displays all of the achievements by completion.
 - Pre-condition: User selected achievements option.
 - Expected Results: The achievements are displayed.
 - Actual Result: The achievements are displayed.
 - Status (Fail/Pass): Pass
- **Test case ID:** exit_option
 - Description/Summary of Test: The resume option exits the menu.
 - Pre-condition: User selected resume option.
 - Expected Results: The menu closes.
 - Actual Result: The menu closed.
 - Status (Fail/Pass): Pass

User Story #750 Collect Metrics

- **Test case ID:** metrics_start
 - Description/Summary of Test: Metrics start collecting data when the player clicks on start game.
 - Pre-condition: Player clicks on the “start game” button.
 - Expected Results: The metrics begin recording data.
 - Actual Result: The metrics began recording data.
 - Status (Fail/Pass): Pass

- **Test case ID:** generate_id
 - Description/Summary of Test: Generate a unique player ID at the start of the game.
 - Pre-condition: The game is started.
 - Expected Results: Generates a unique player id.
 - Actual Result: A unique player id was generated.
 - Status (Fail/Pass): Pass

- **Test case ID:** collect_date_and_time
 - Description/Summary of Test: The date and time that the game is started is recorded.
 - Pre-condition: The game is started.
 - Expected Results: The date and time is recorded.
 - Actual Result: The date and time was recorded.
 - Status (Fail/Pass): Pass

- **Test case ID:** collect_time_room_entered
 - Description/Summary of Test: The time for when a room is entered is collected.
 - Pre-condition: The player enters a room.
 - Expected Results: The entered room time is collected.
 - Actual Result: The entered room time was collected.
 - Status (Fail/Pass): Pass

- **Test case ID:** collect_time_puzzle_started
 - Description/Summary of Test: The time for when a puzzle is started is collected.
 - Pre-condition: The player starts the puzzle.
 - Expected Results: The puzzle started time is collected.
 - Actual Result: The puzzle started time was collected.
 - Status (Fail/Pass): Pass

- **Test case ID:** collect_time_puzzle_solved
 - Description/Summary of Test: The time for when a puzzle is solved is collected.
 - Pre-condition: The player solves the puzzle.
 - Expected Results: The puzzle solved time is collected.

- Actual Result: The puzzle solved time was collected.
- Status (Fail/Pass): Pass

- **Test case ID:** collect_time_room_exited
- Description/Summary of Test: The time for when a room is exited is collected.
- Pre-condition: The player exits a room.
- Expected Results: The exited room time is collected.
- Actual Result: The exited room time was collected.
- Status (Fail/Pass): Pass

- **Test case ID:** collect_total_number_of_tries
- Description/Summary of Test: The number of attempts made to solve the puzzle is collected.
- Pre-condition: The player solved the puzzle.
- Expected Results: The total number of tries is collected.
- Actual Result: The total number of tries was collected.
- Status (Fail/Pass): Pass

- **Test case ID:** record_player_location_rotation
- Description/Summary of Test: Players location and rotation is being collected every second.
- Pre-condition: The game is started.
- Expected Results: Payers location and rotation gets collected every second.
- Actual Result: Players location and rotation is collected every second.
- Status (Fail/Pass): Pass

User Story #751 Print Metrics

- **Test case ID:** metrics_to_text_file
- Description/Summary of Test: All of the metrics except for the players path are appended to a text file, with commas separating the data and a new line separating each game.
- Pre-condition: The game ends.
- Expected Results: All of the metrics except for the players path get appended to a text file, with commas separating the data and a new line separating each game.

- Actual Result: All of the metrics except for the players path are appended to a text file, with commas separating the data and a new line separating each game.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** metrics_to_json
 - Description/Summary of Test: All of the metrics for each game are appended to a JSON file as an object.
 - Pre-condition: The game ends.
 - Expected Results: All of the metrics for each game will be appended to a JSON file as an object.
 - Actual Result: All of the metrics for each game are appended to a JSON file as an object.
 - Status (Fail/Pass): Pass

User Story #752 Implement Main Menu

- **Test case ID:** menu_options
 - Description/Summary of Test: The following options are shown in the menu: start game, credits, quit game.
 - Pre-condition: User opened the menu.
 - Expected Results: The menu options are shown.
 - Actual Result: The menu options are shown.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** start_game_option
 - Description/Summary of Test: The start game option, starts the game.
 - Pre-condition: User selected start game option.
 - Expected Results: The game gets started.
 - Actual Result: The game is started.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** credits_option
 - Description/Summary of Test: The credits option displays the credits.
 - Pre-condition: User selected credits option.
 - Expected Results: The credits get displayed.

- Actual Result: The credits are displayed.
 - Status (Fail/Pass): Pass
-
- **Test case ID:** quit_game_option
 - Description/Summary of Test: The quit game option ends the game.
 - Pre-condition: User selected the quit game option.
 - Expected Results: The game is ended.
 - Actual Result: The game ended.
 - Status (Fail/Pass): Pass

User Story #753 Create Enter Room Trigger

- **Test case ID:** trigger_activates_once
 - Description/Summary of Test: The trigger can only be activated once.
 - Pre-condition: Player overlaps trigger.
 - Expected Results: The trigger can only be activated once.
 - Actual Result: The trigger is activated only once.
 - Status (Fail/Pass): Pass

- **Test case ID:** collect_time
 - Description/Summary of Test: Time is collected for the entered room metric.
 - Pre-condition: Trigger is activated.
 - Expected Results: Time gets collected for the entered room metric.
 - Actual Result: Time is collected for the entered room metric.
 - Status (Fail/Pass): Pass

- **Test case ID:** update_level_count
 - Description/Summary of Test: The system is updated with the players current level.
 - Pre-condition: Trigger is activated.
 - Expected Results: The system gets updated with the players current level.
 - Actual Result: The system is updated with the players current level.
 - Status (Fail/Pass): Pass

- **Test case ID:** puzzle_intro_displayed
- Description/Summary of Test: The puzzle introduction is displayed.
- Pre-condition: Trigger is activated.
- Expected Results: The puzzle introduction gets displayed.
- Actual Result: The puzzle introduction is displayed.
- Status (Fail/Pass): Pass

User Story #754 Create Game Completed Trigger

- **Test case ID:** game_end
- Description/Summary of Test: The game ends when the trigger is activated.
- Pre-condition: Trigger is activated.
- Expected Results: The game ends when the trigger is activated.
- Actual Result: The game ended when the trigger was activated.
- Status (Fail/Pass): Pass
- **Test case ID:** congrats_screen
- Description/Summary of Test: The congratulation screen is displayed to the player.
- Pre-condition: The game is ended through the trigger.
- Expected Results: The congratulations screen gets displayed to the player.
- Actual Result: The congratulations screen got displayed to the player.
- Status (Fail/Pass): Pass
- **Test case ID:** return_to_main_menu
- Description/Summary of Test: The player is returned to the main menu after viewing the congratulations.
- Pre-condition: The player viewed the congratulations screen.
- Expected Results: The player gets returned to the main menu after viewing the congratulations screen.
- Actual Result: The player is returned to the main menu after viewing the congratulations screen.
- Status (Fail/Pass): Pass

User Story #755 Highlight Movable Objects

- **Test case ID:** highlight_in_range
 - Description/Summary of Test: Movable objects are highlighted when the player is within range.
 - Pre-condition: Player is within range of movable objects.
 - Expected Results: Movable objects are highlighted.
 - Actual Result: Movable objects are highlighted.
 - Status (Fail/Pass): Pass
- **Test case ID:** highlight_out_of_range
 - Description/Summary of Test: Movable objects are not highlighted when the player is out of range.
 - Pre-condition: Player is not within range of the movable objects.
 - Expected Results: Movable objects out of the players range will not be highlighted.
 - Actual Result: Movable objects out of the players range are not highlighted.
 - Status (Fail/Pass): Pass

GLOSSARY

- **Blender:** A 3D computer graphics software toolset used for creating 3D models.
- **C++:** A general purpose object oriented programming language.
- **Comma Separated Values (CSV):** Text file containing comma separated values serves as repository.
- **Computer Science (CS):** The study of the theory, experimentation, and engineering that forms the basis for the design and use of computers.
- **Entity-Component-System (ECS):** Architectural pattern used on modern software development for game.
- **GamePad:** Type of controller used to interact with a system, held in the hands and input is mainly done by the thumbs.
- **Git:** A version control system designed to handle projects with speed and efficiency.
- **Headset:** Device used on virtual reality to generate virtual environments as a head mounted display.
- **HTC Vive:** Hardware that helps the user experience virtual reality, it includes sensors, motion controllers and headset.
- **JSON:** JavaScript Object Notation is a type of data collection that specializes in readability of text as objects.
- **Sensor:** Device that reads and track the motion of the user or another hardware piece and interacts with the system.
- **Unreal Engine 4 (UE4):** A suite of integrated tools for game developers to design and build games. Used to design and build the Alchemist Escape game. Blueprints and C++ were used to implement the logic for the game.
- **Visual Studio:** An IDE used for UE4 to write C++ scripts.
- **Virtual Reality (VR):** Type of software development that utilizes hardware to immerse the user on a virtual environment.

APPENDIX

Appendix A - UML Diagrams

Use Case Diagrams:

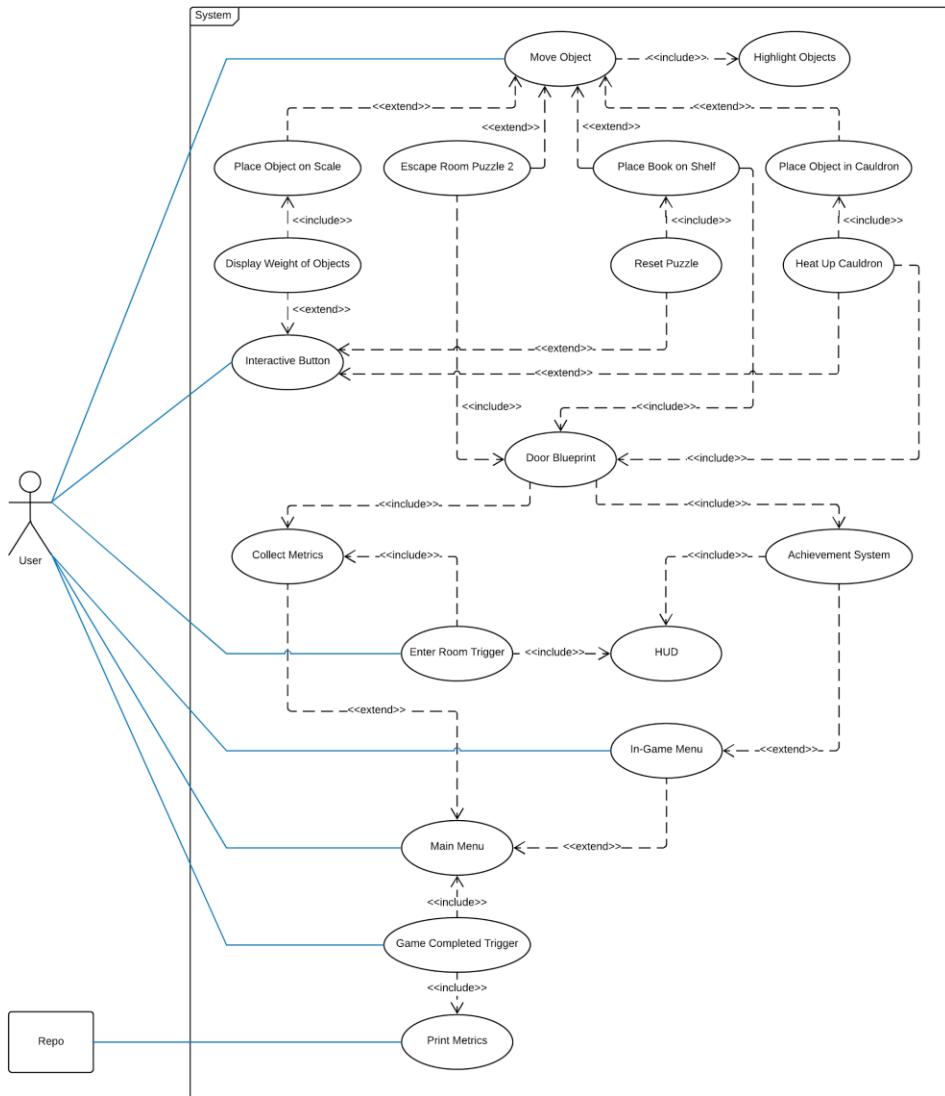


Figure 1: Complete System Use Case Diagram

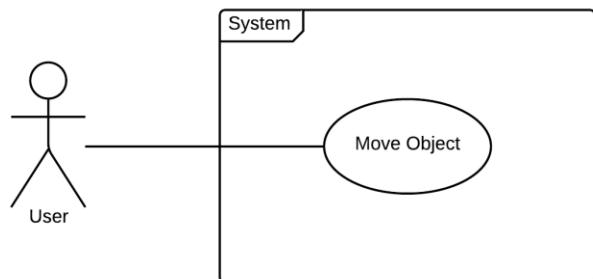


Figure 2: User Story #696 Use Case Diagram

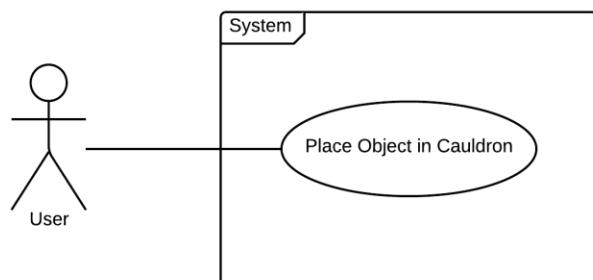


Figure 3: User Story #697 Use Case Diagram

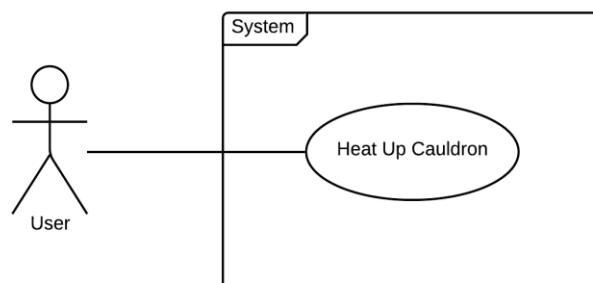


Figure 4: User Story #698 Use Case Diagram

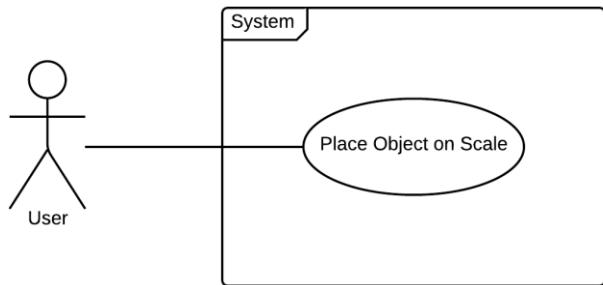


Figure 5: User Story #699 Use Case Diagram

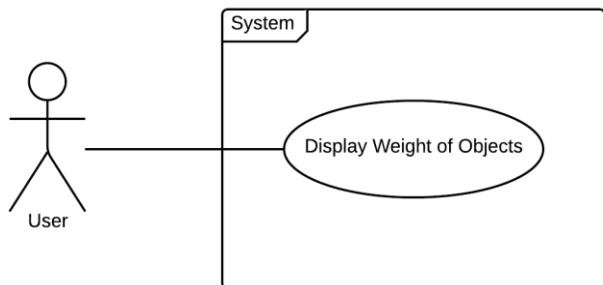


Figure 6: User Story #700 Use Case Diagram

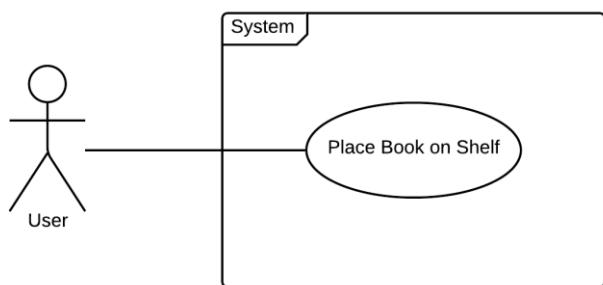


Figure 7: User Story #701 Use Case Diagram

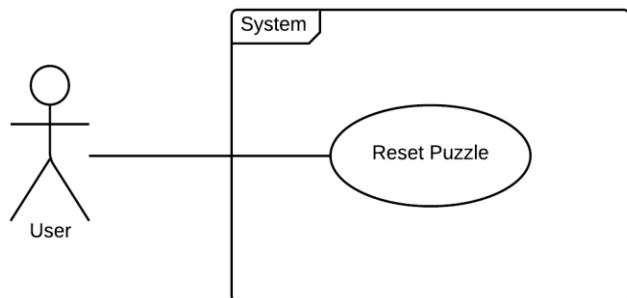


Figure 8: User Story #702 Use Case Diagram

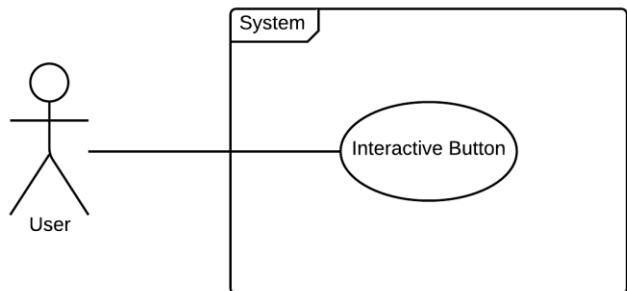


Figure 9: User Story #723 Use Case Diagram

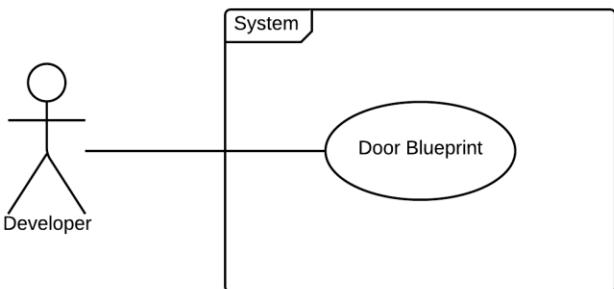


Figure 10: User Story #724 Use Case Diagram

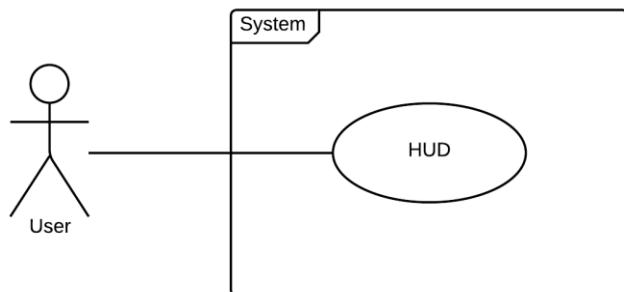


Figure 11: User Story #725 Use Case Diagram

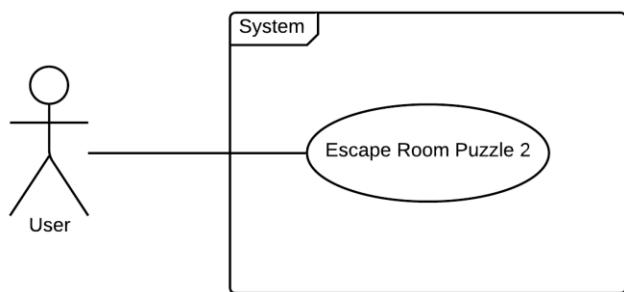


Figure 12: User Story #726 Use Case Diagram

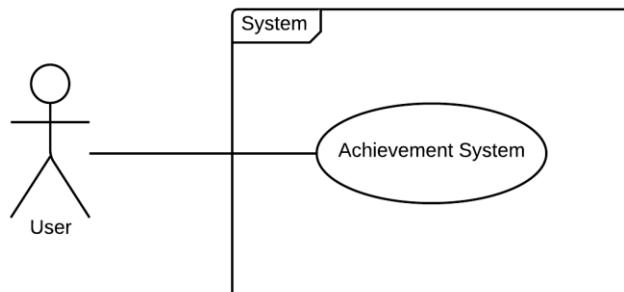


Figure 13: User Story #727 Use Case Diagram

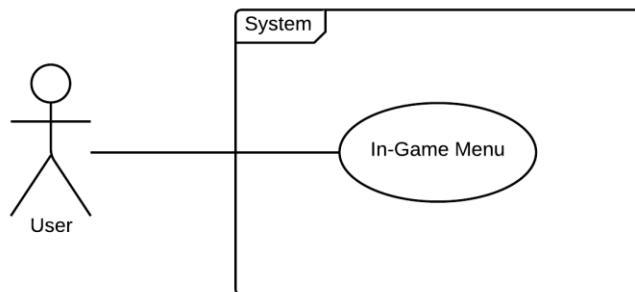


Figure 14: User Story #728 Use Case Diagram

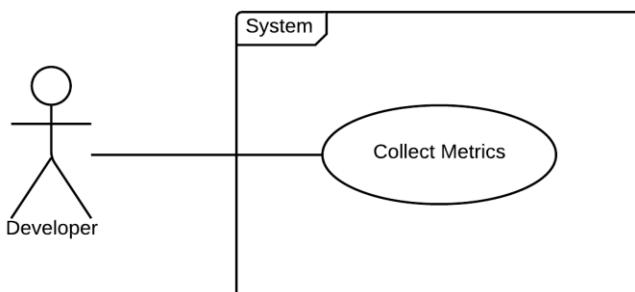


Figure 15: User Story #750 Use Case Diagram

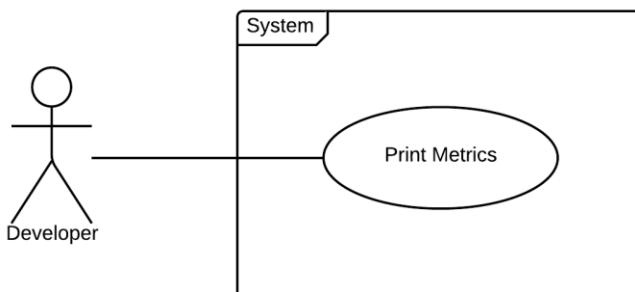


Figure 16: User Story #751 Use Case Diagram

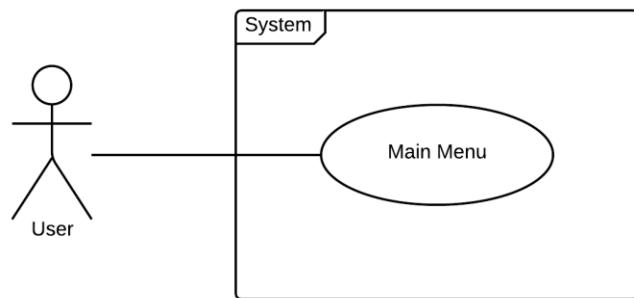


Figure 17: User Story #752 Use Case Diagram

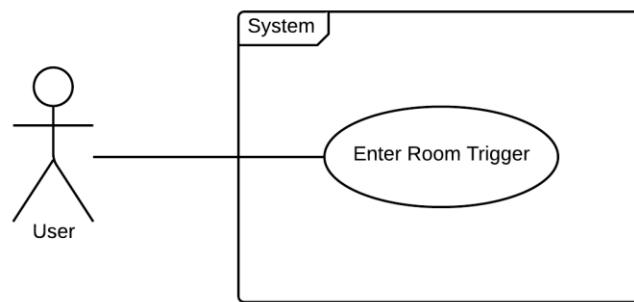


Figure 18: User Story #753 Use Case Diagram

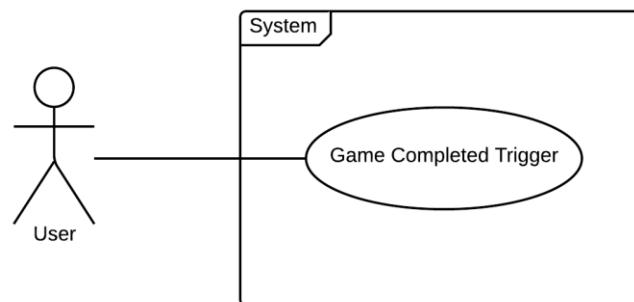


Figure 19: User Story #754 Use Case Diagram

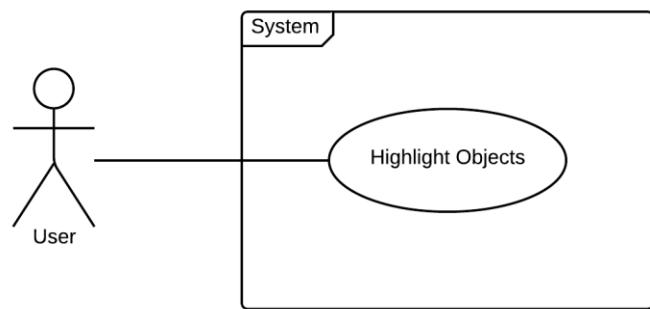


Figure 20: User Story #755 Use Case Diagram

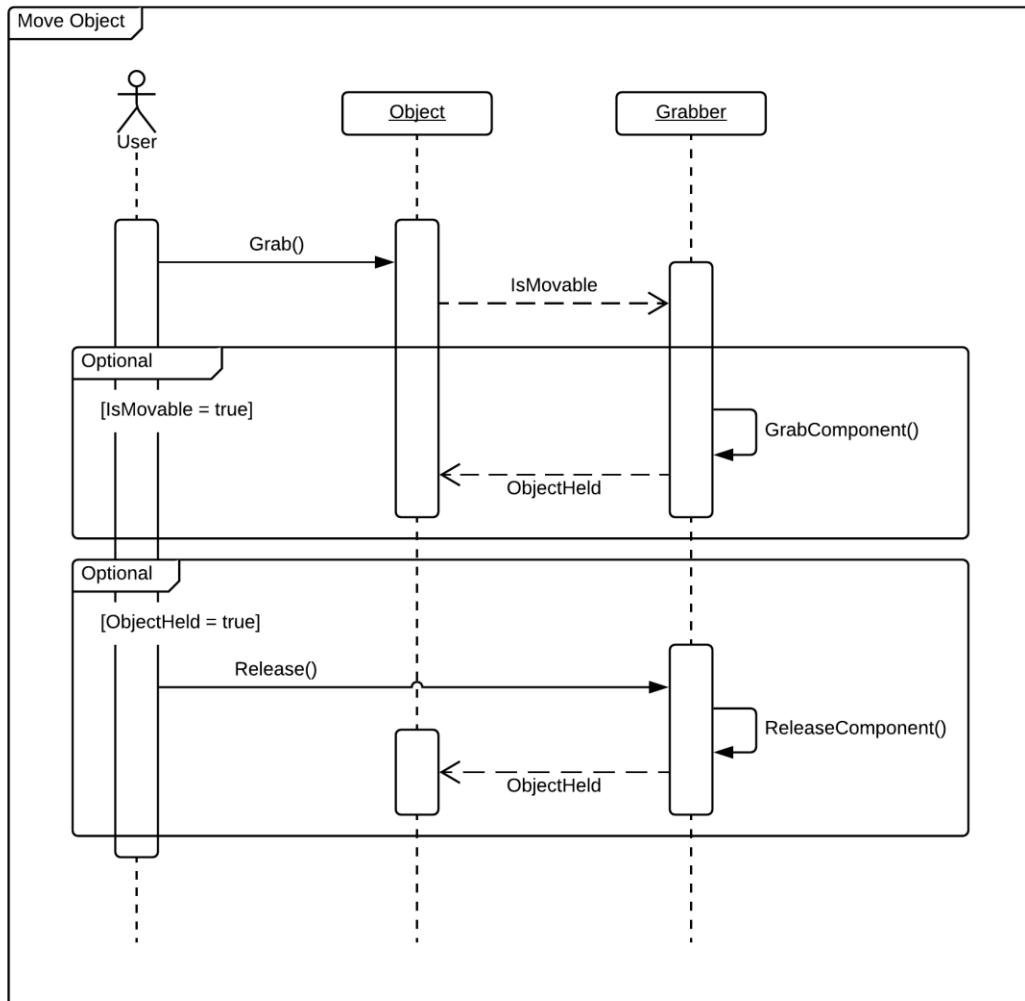
Sequence Diagrams:

Figure 1: User Story #696 Sequence Diagram

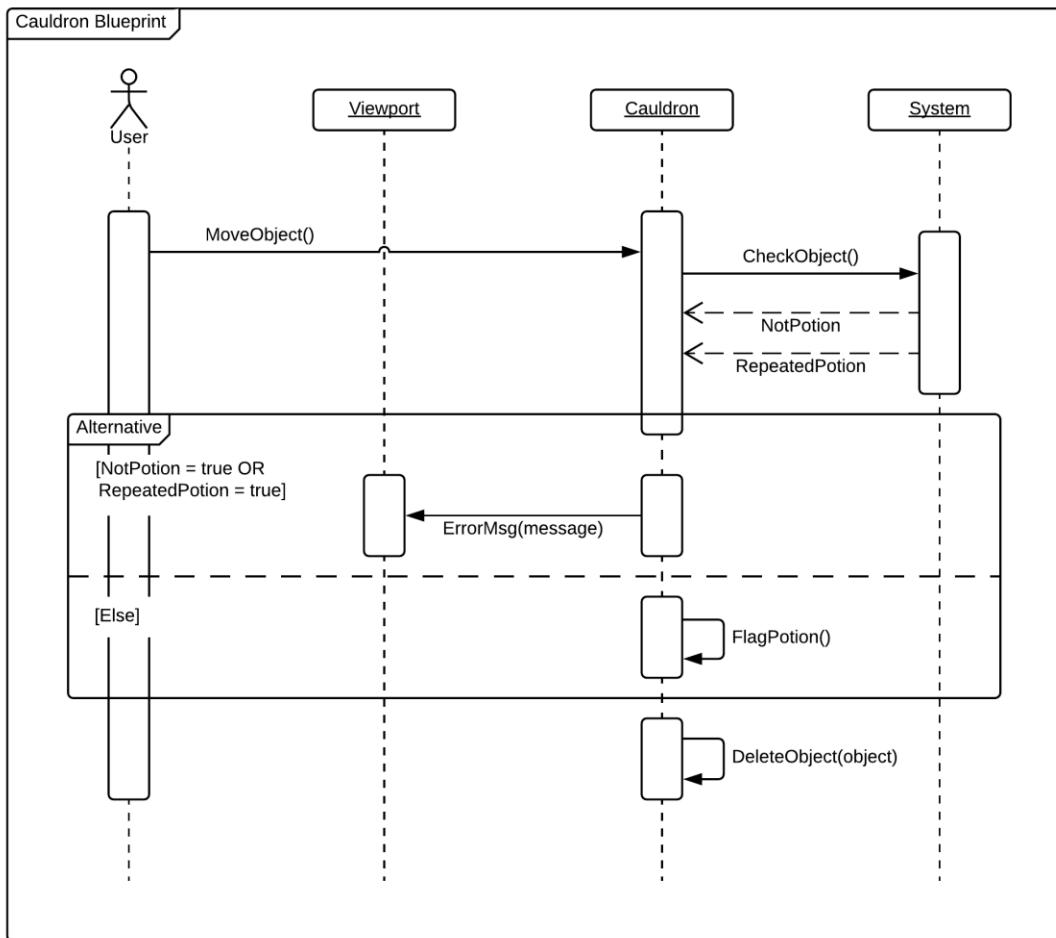


Figure 2: User Story #697 Sequence Diagram

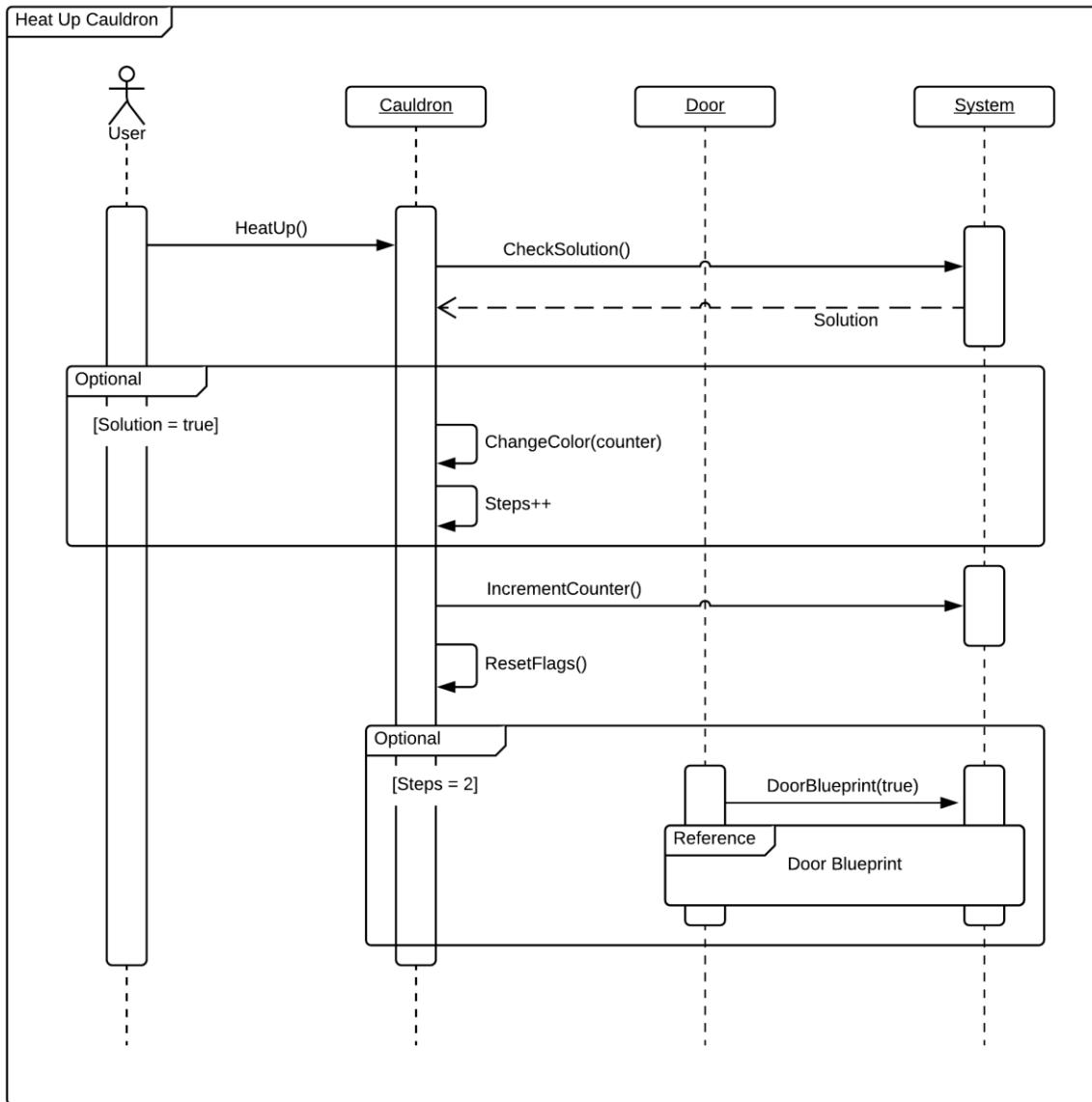


Figure 3: User Story #698 Sequence Diagram

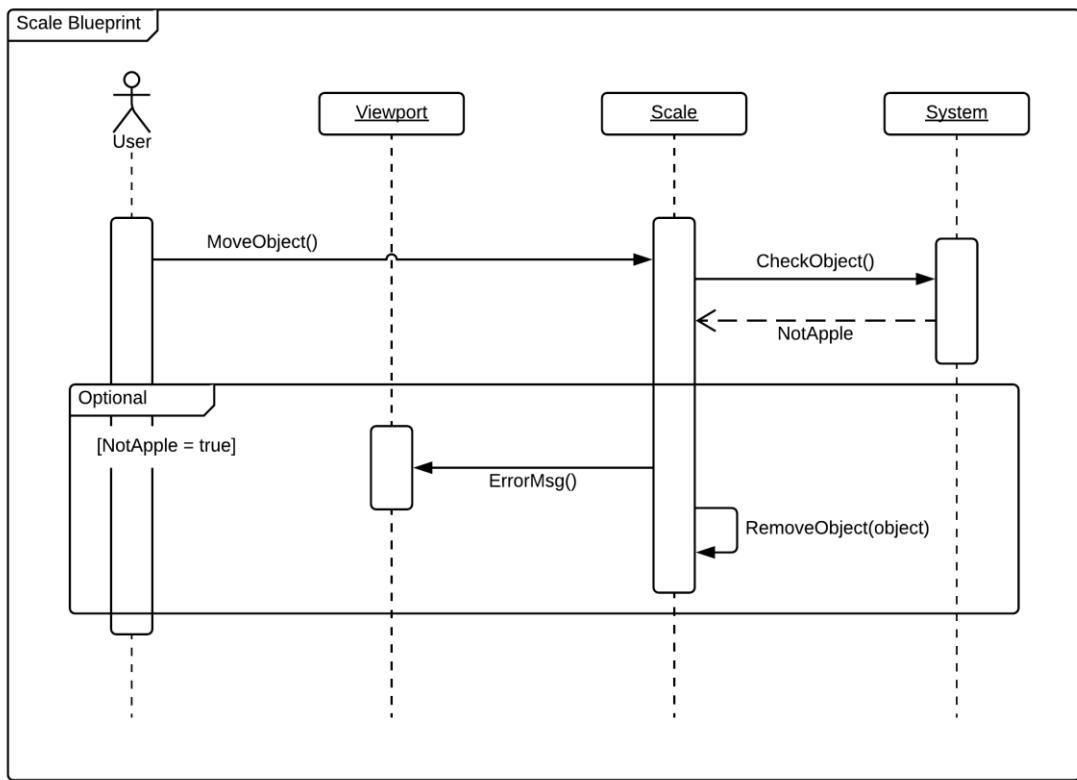


Figure 4: User Story #699 Sequence Diagram

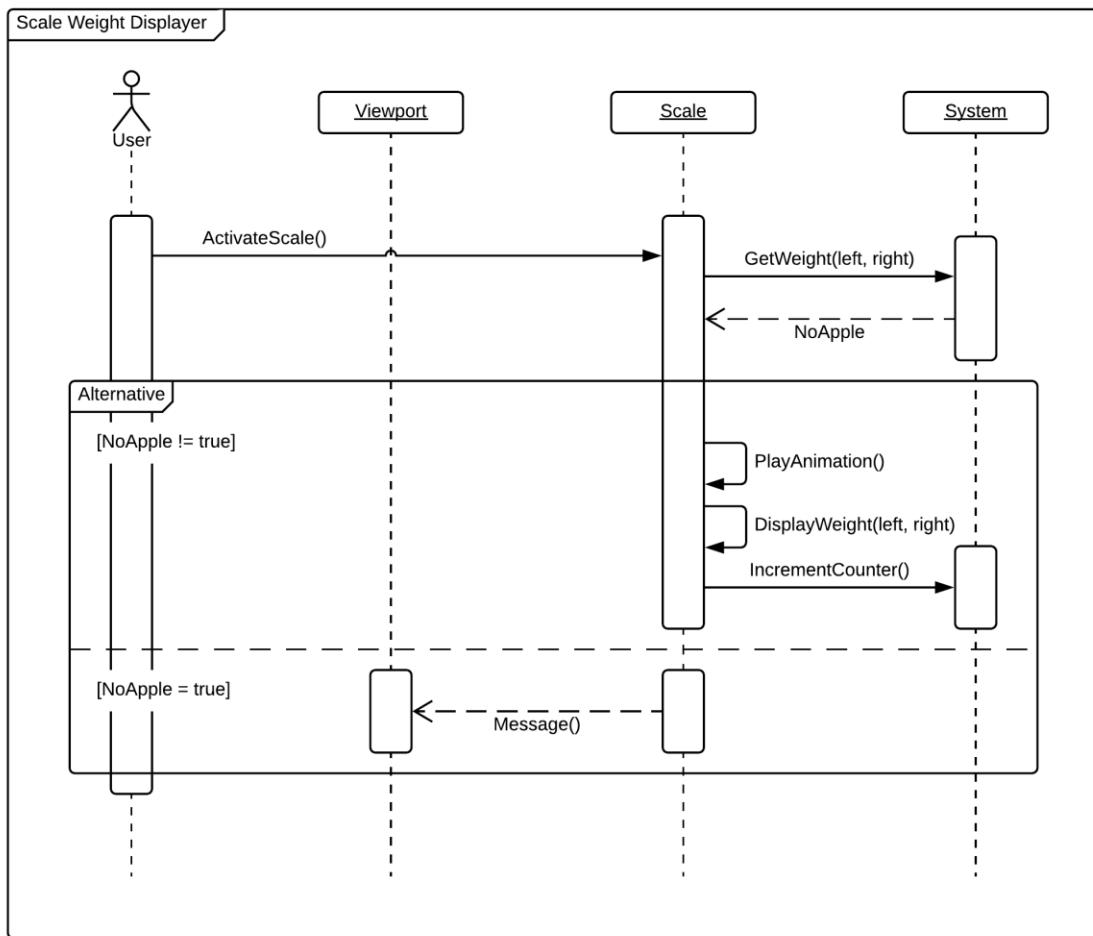


Figure 5: User Story #700 Sequence Diagram

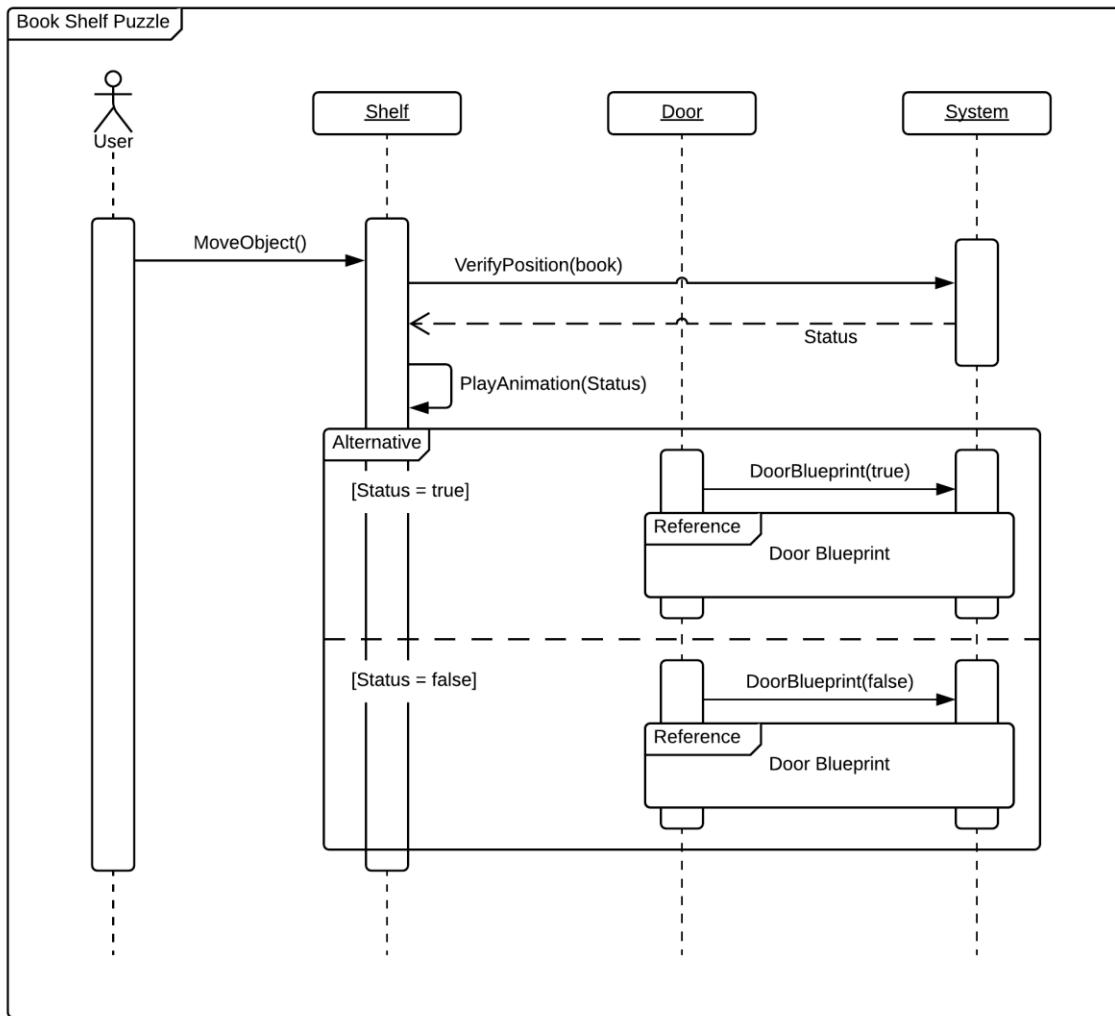


Figure 6: User Story #701 Sequence Diagram

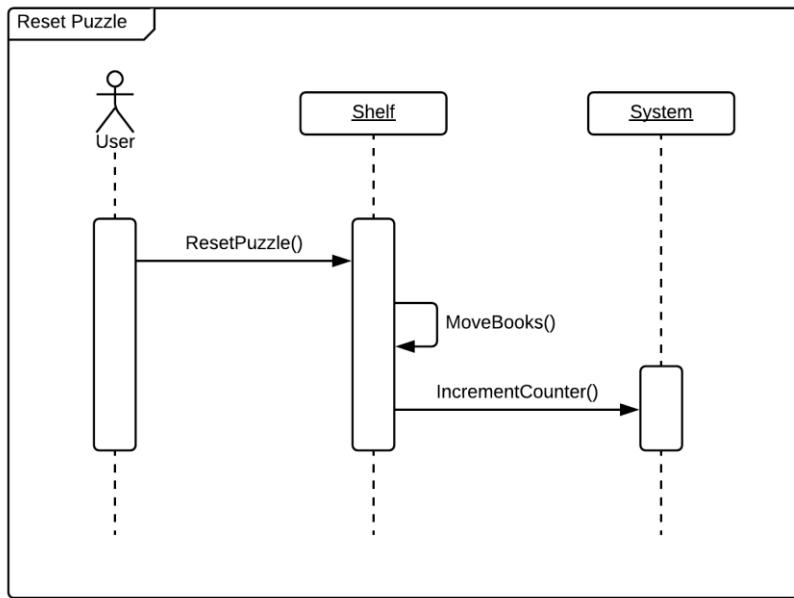


Figure 7: User Story #702 Sequence Diagram

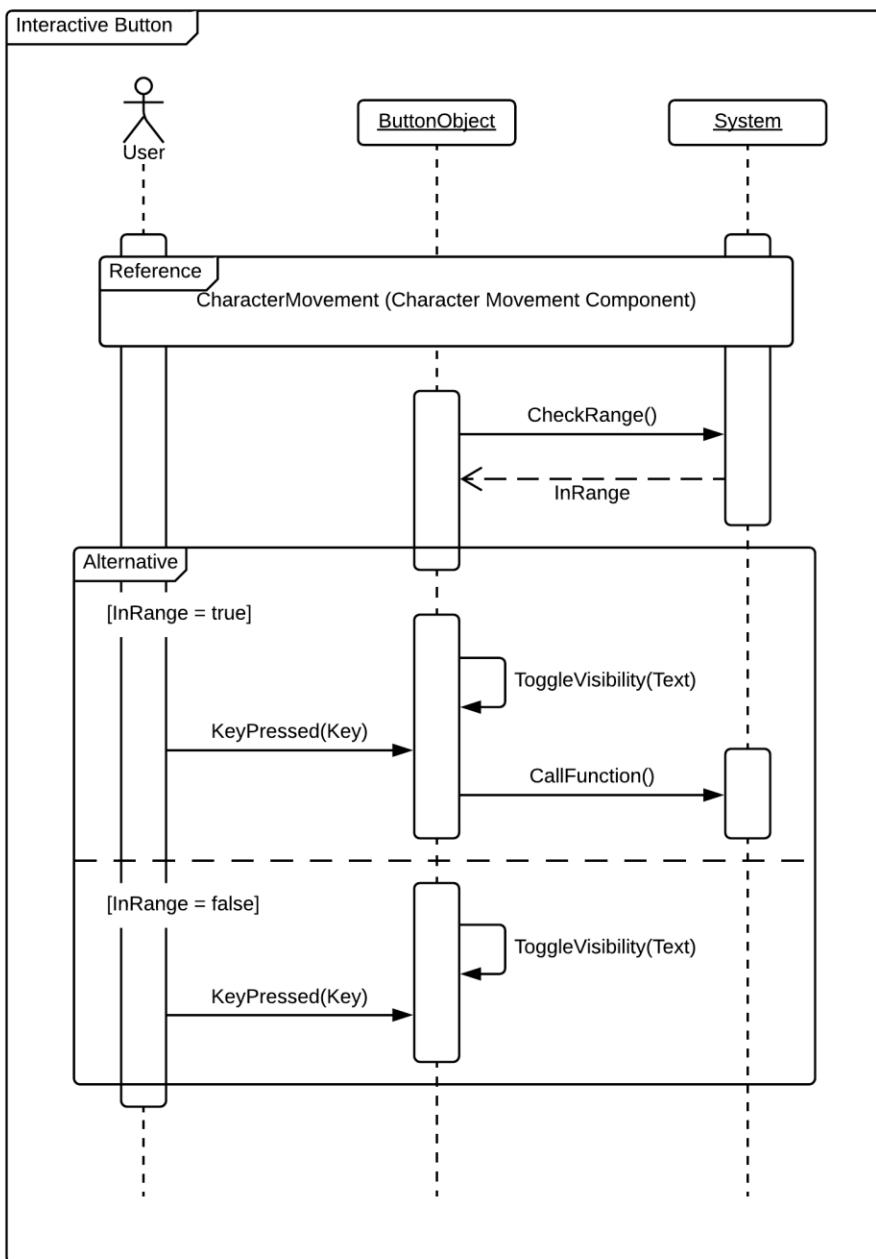


Figure 8: User Story #723 Sequence Diagram

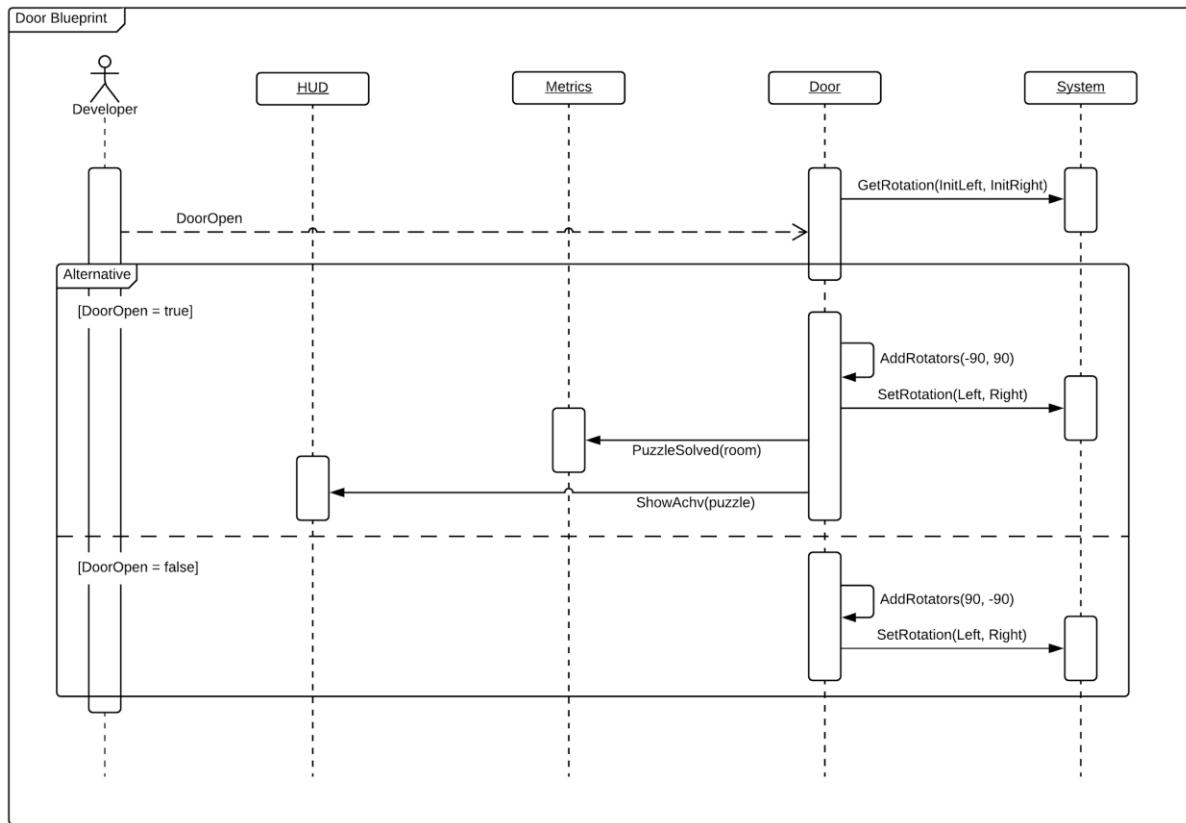


Figure 9: User Story #724 Sequence Diagram

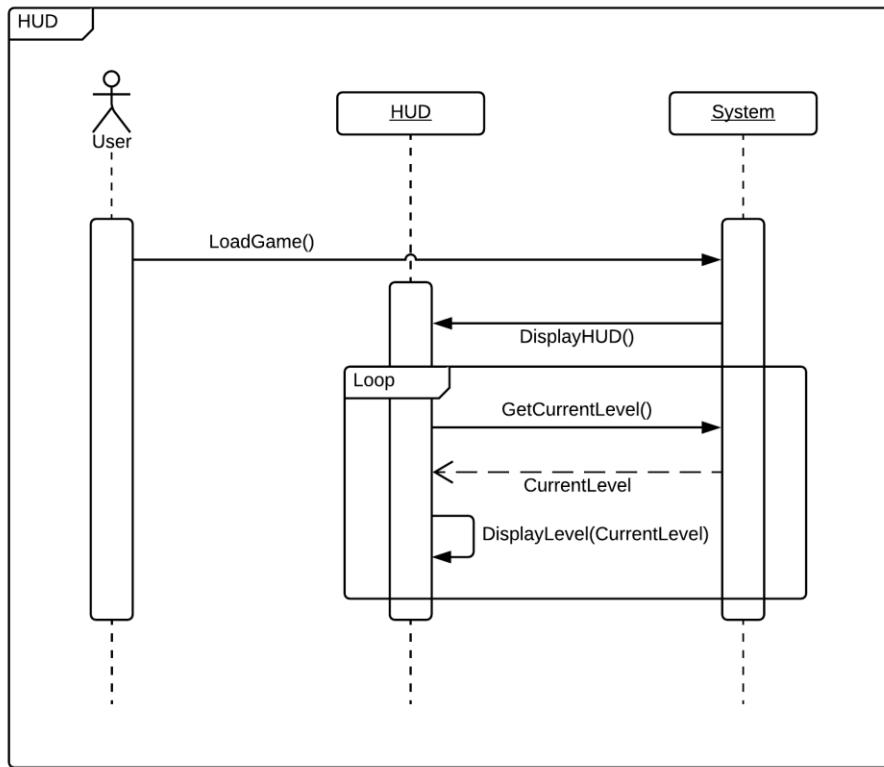


Figure 10: User Story #725 Sequence Diagram

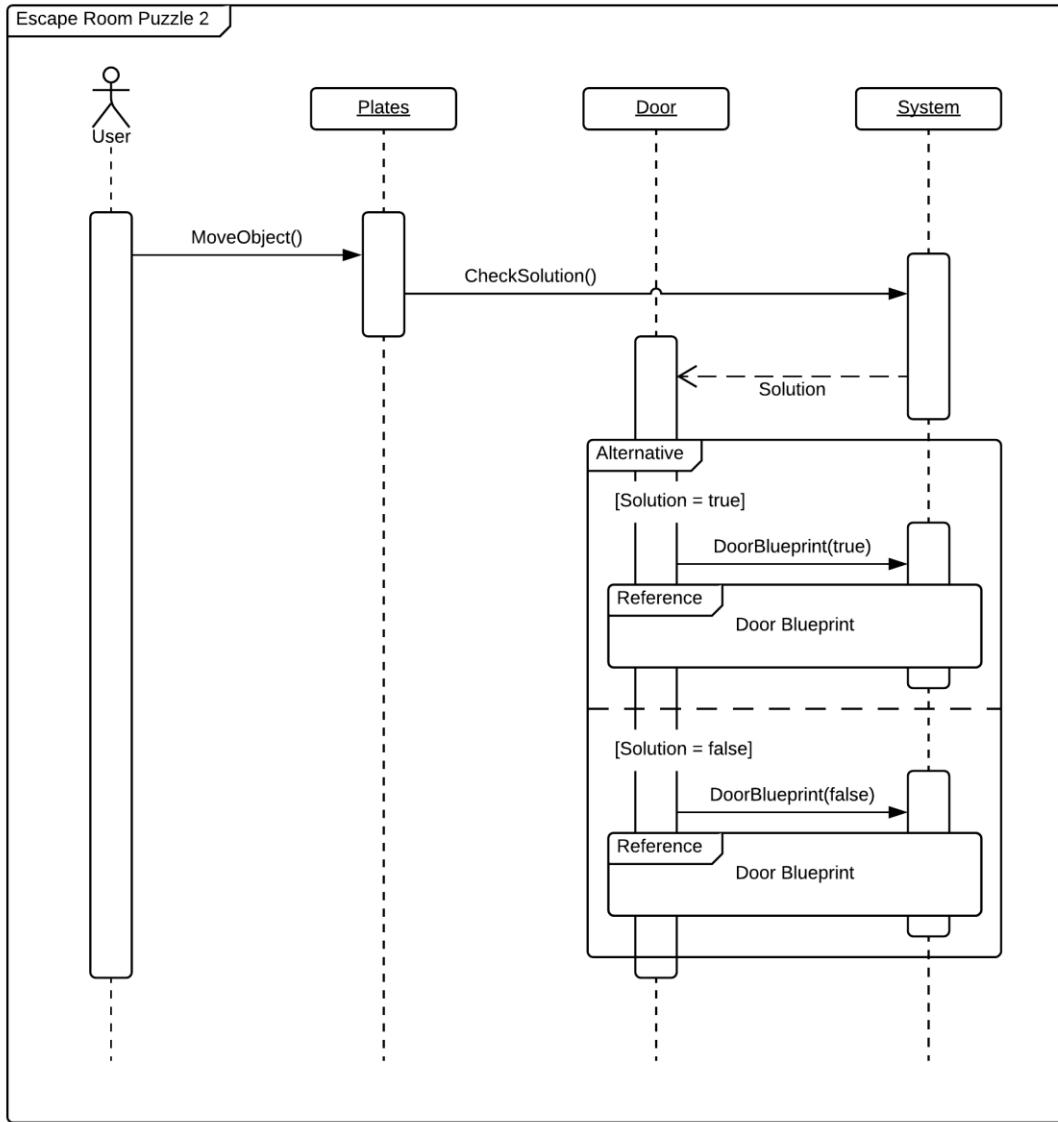


Figure 11: User Story #726 Sequence Diagram

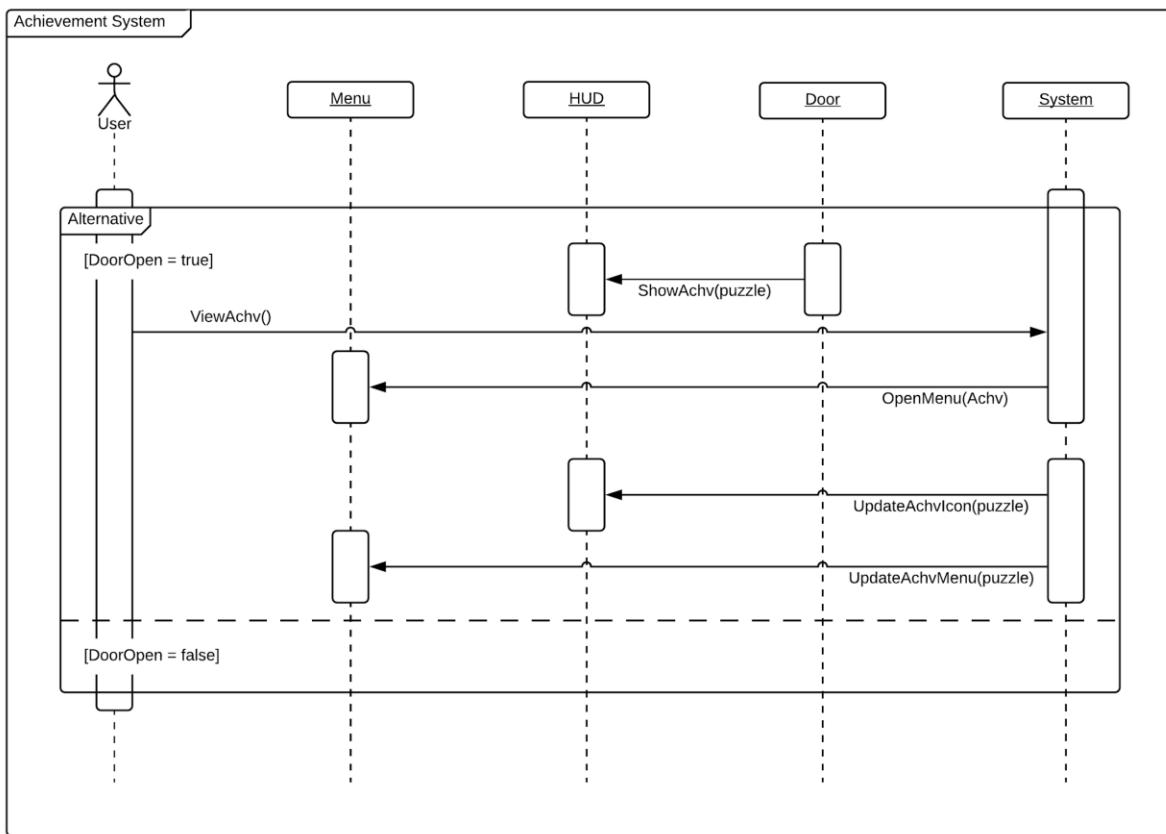


Figure 12: User Story #727 Sequence Diagram

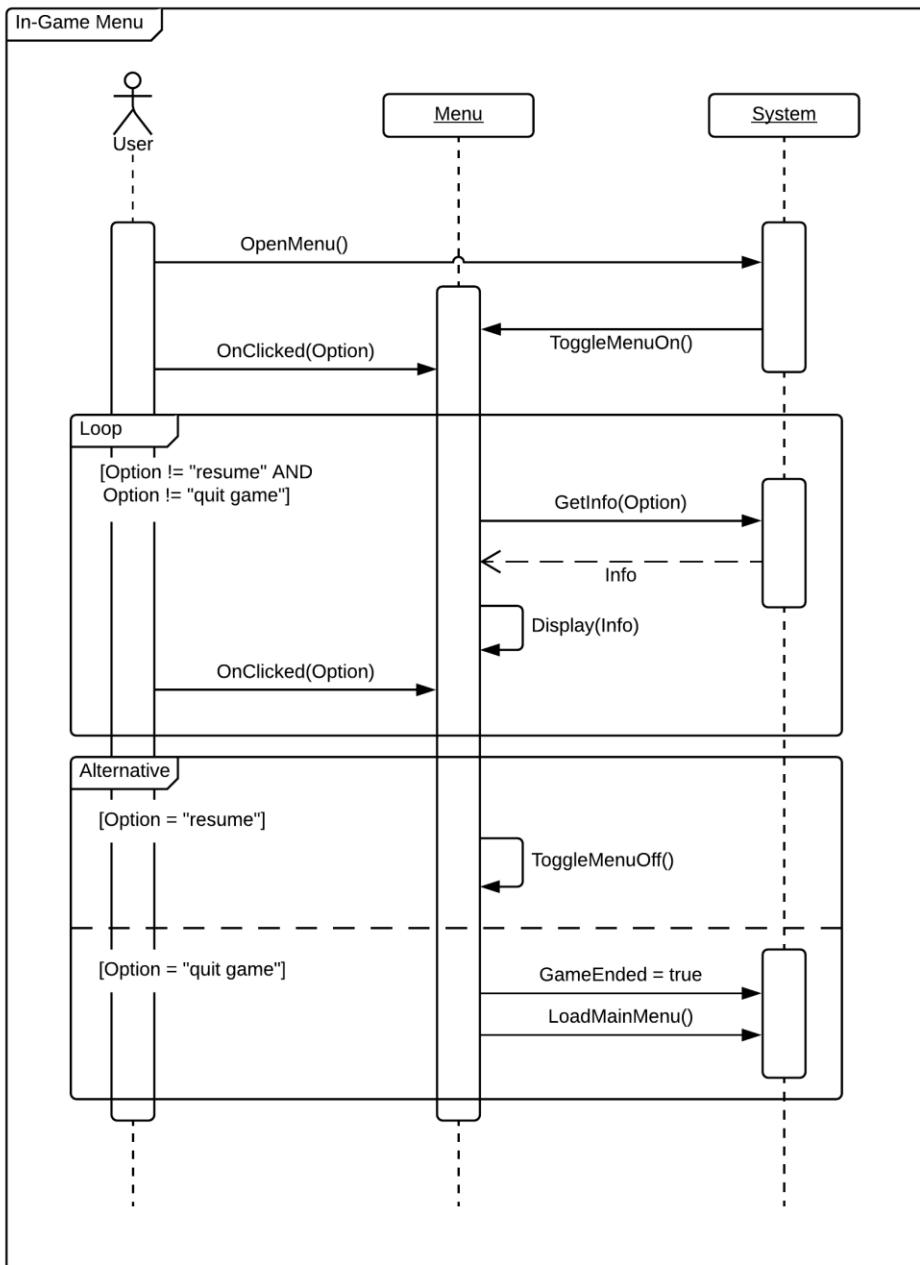


Figure 13: User Story #728 Sequence Diagram

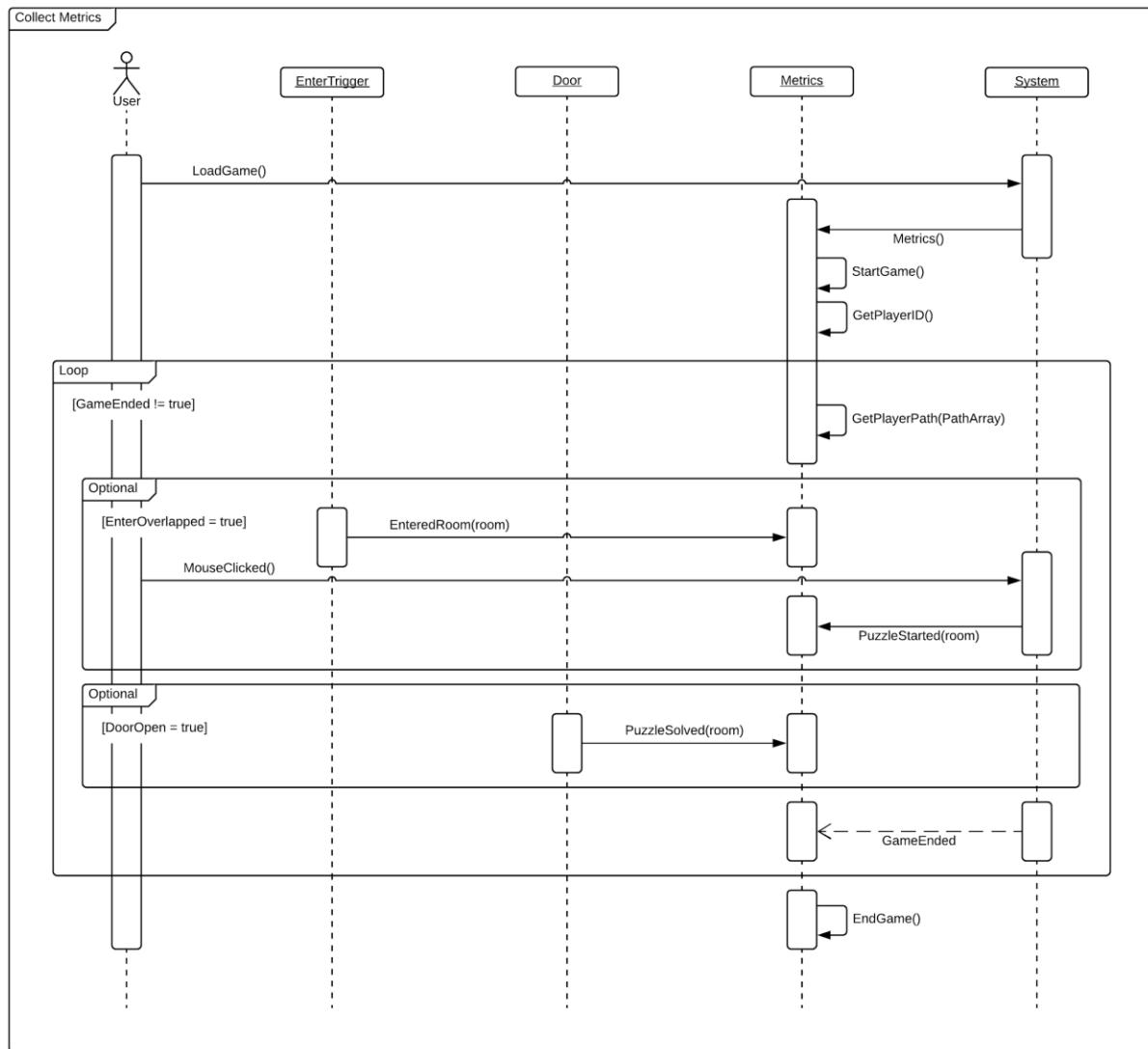


Figure 14: User Story #750 Sequence Diagram

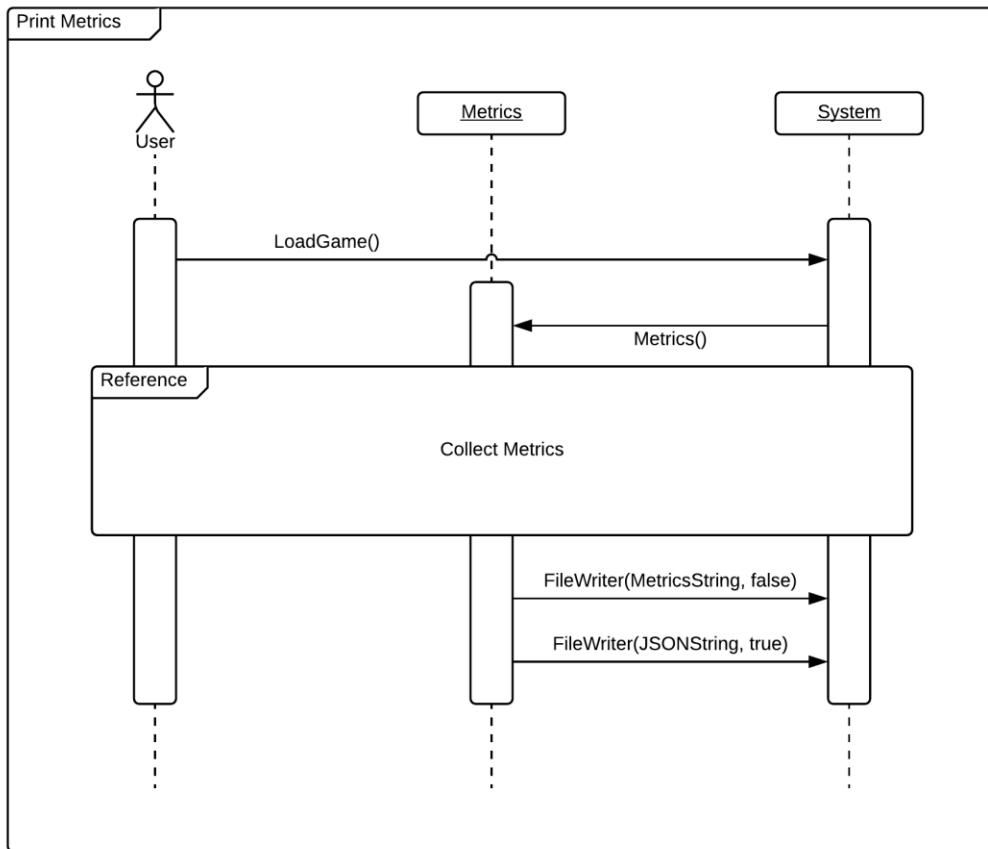


Figure 15: User Story #751 Sequence Diagram

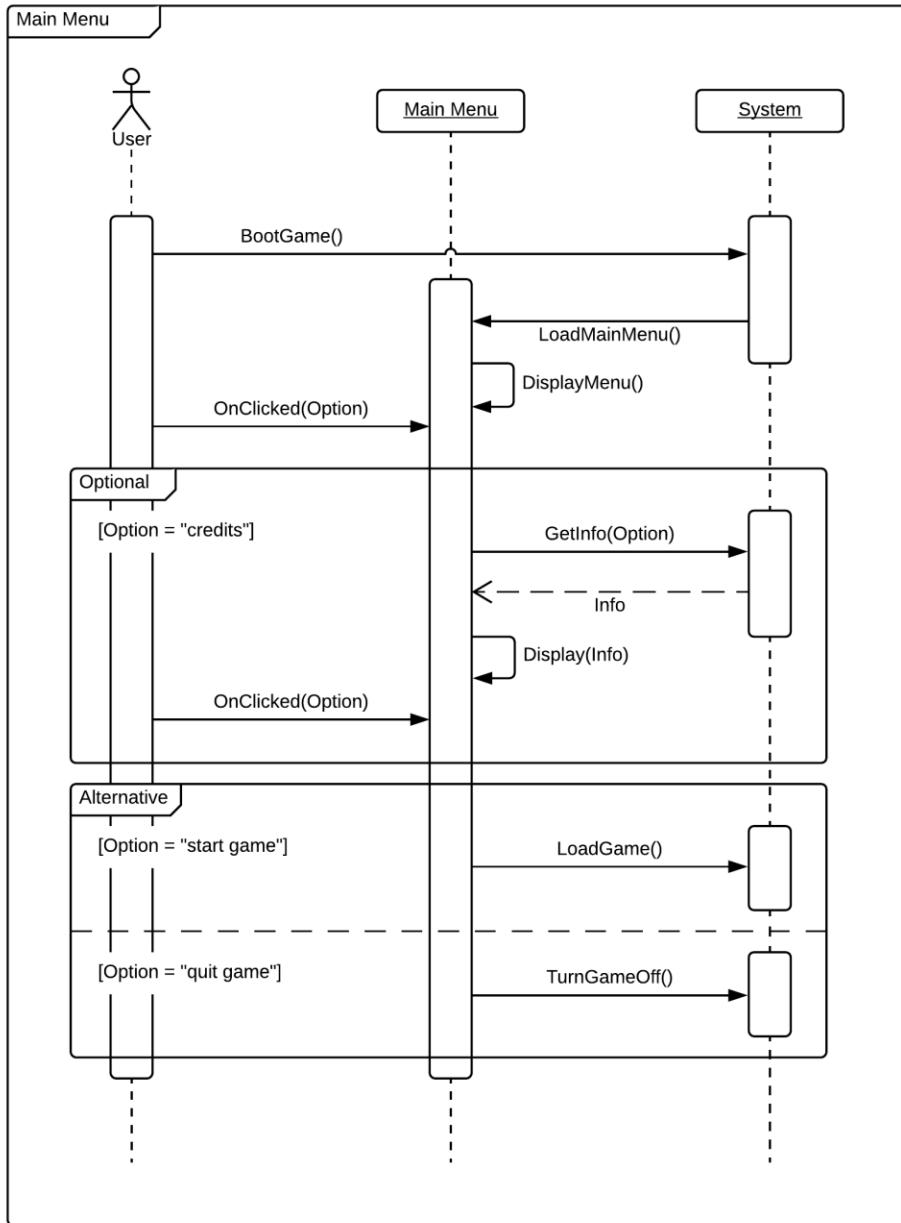


Figure 16: User Story #752 Sequence Diagram

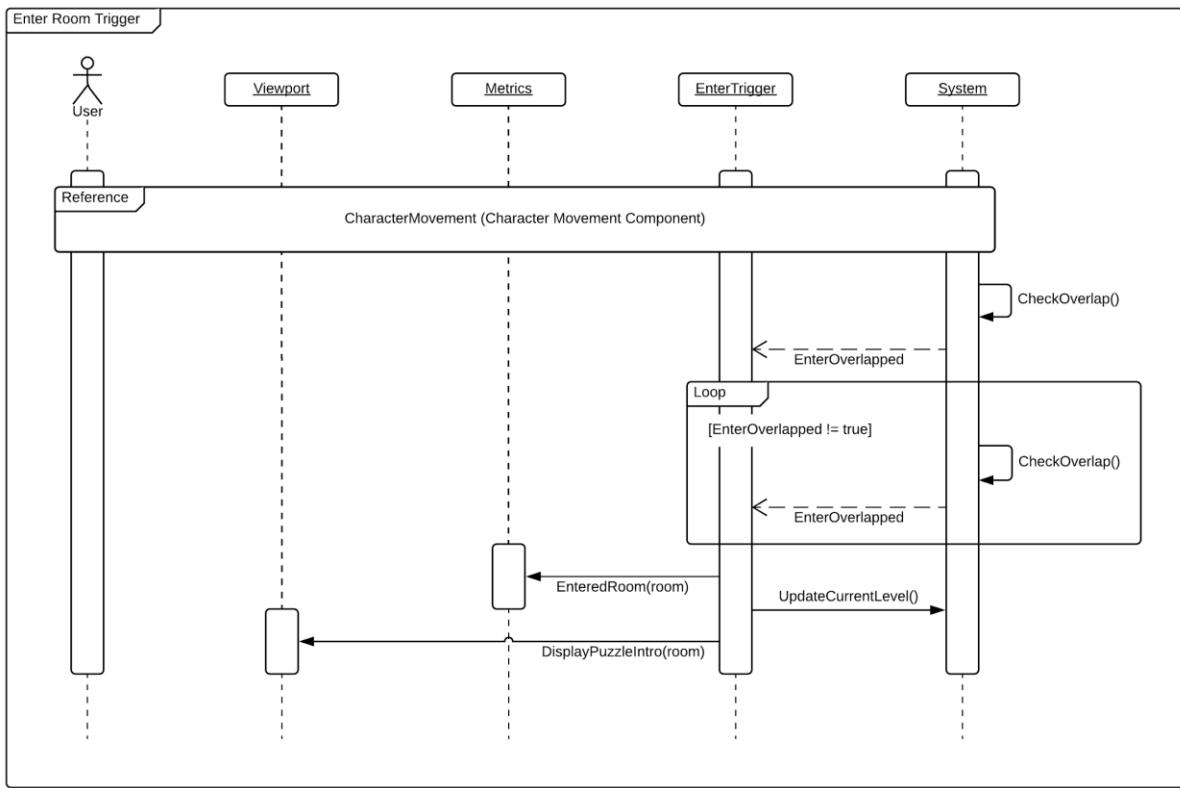


Figure 17: User Story #753 Sequence Diagram

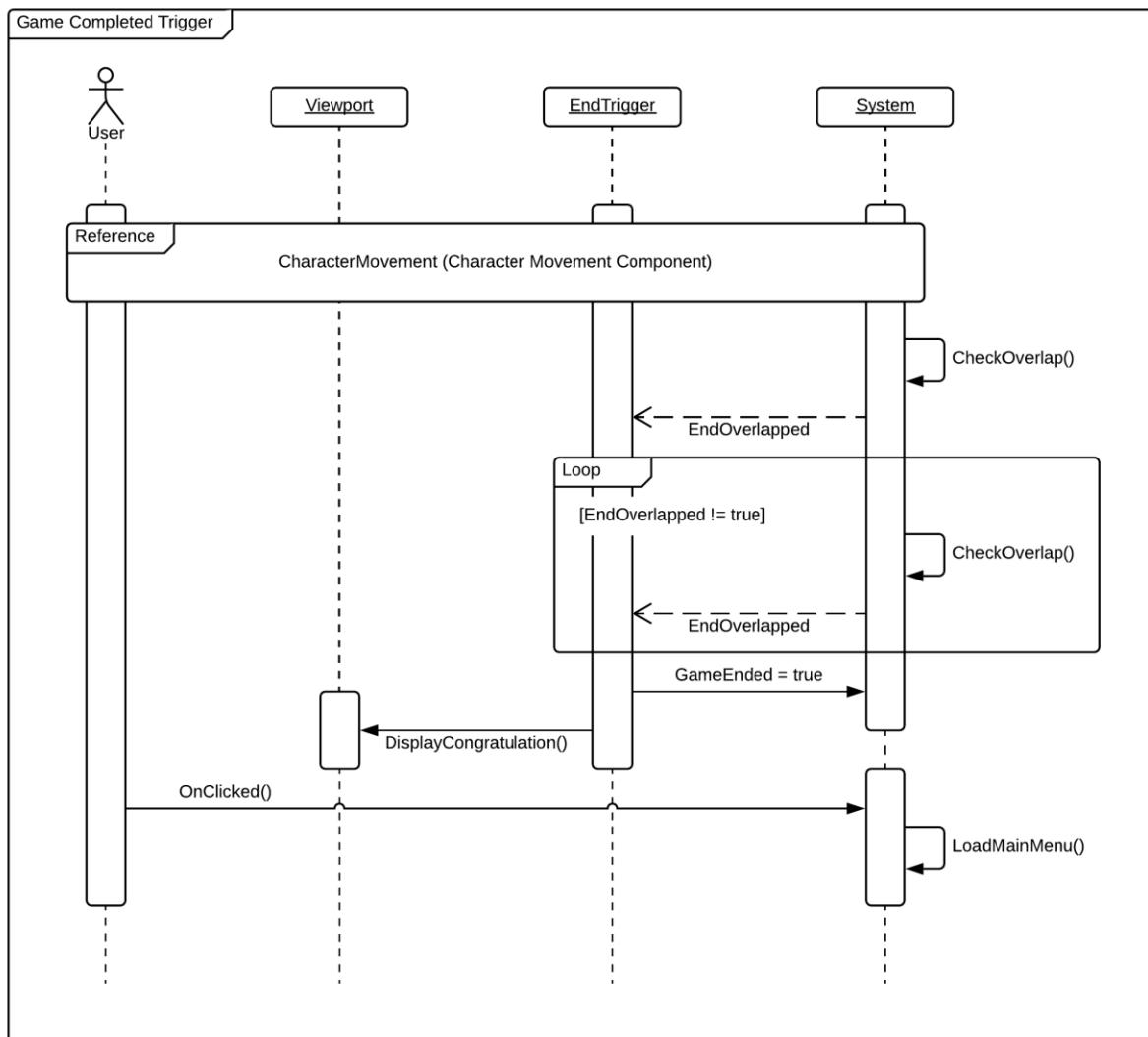


Figure 18: User Story #754 Sequence Diagram

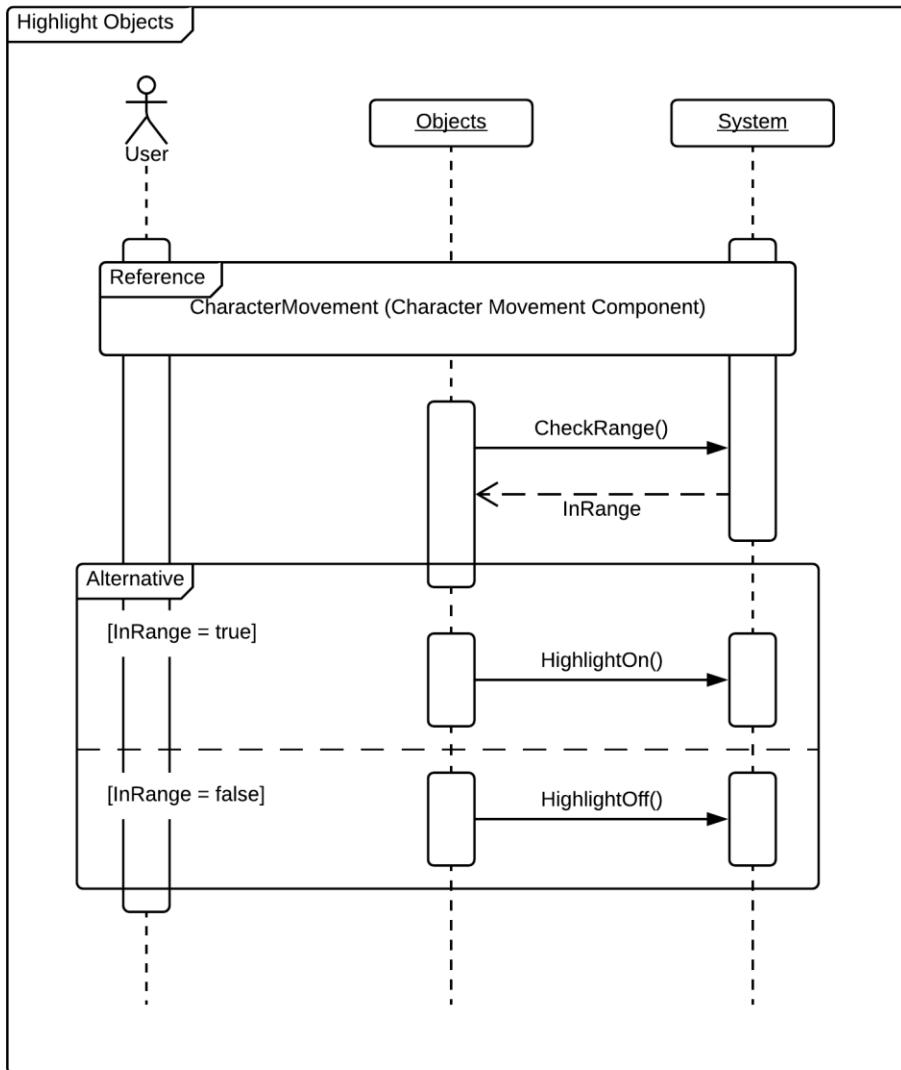


Figure 19: User Story #755 Sequence Diagram

Class Diagrams:

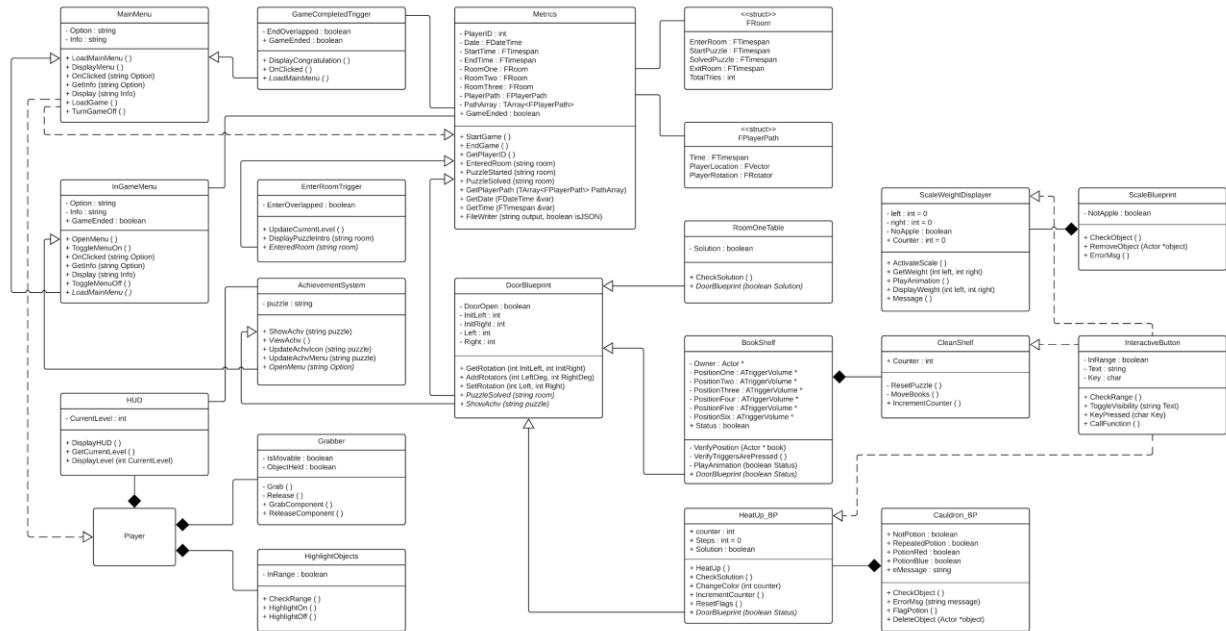


Figure 1: Complete System Class Diagram

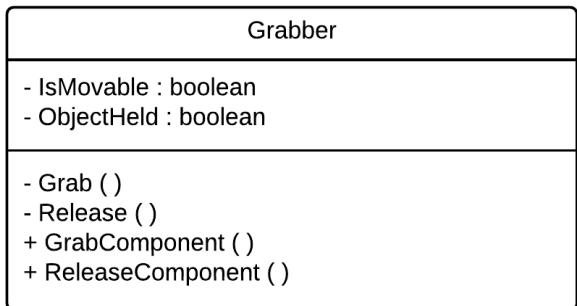


Figure 2: User Story #696 Class Diagram

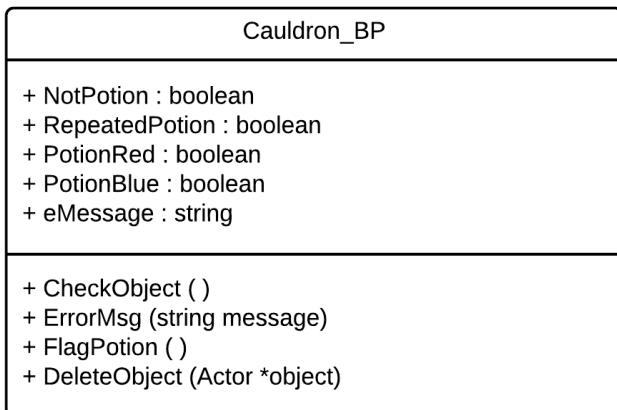


Figure 3: User Story #697 Class Diagram

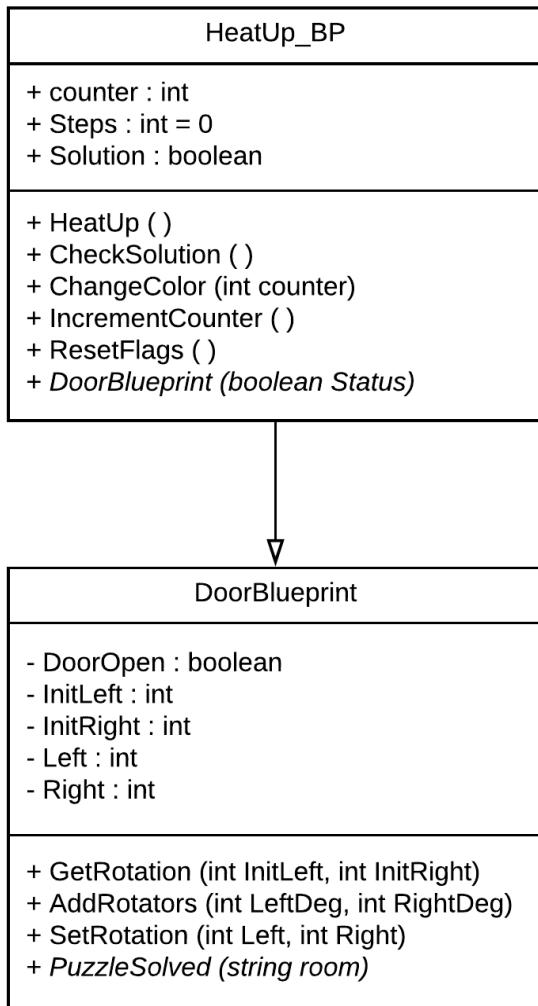


Figure 4: User Story #698 Class Diagram

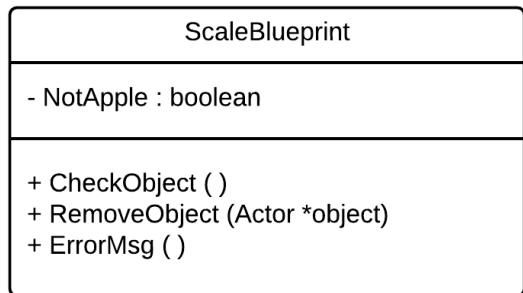


Figure 5: User Story #699 Class Diagram

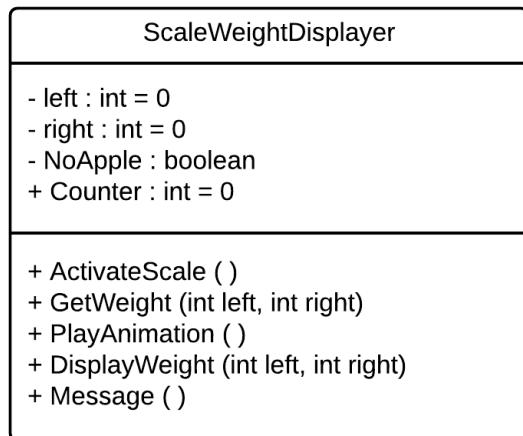


Figure 6: User Story #700 Class Diagram

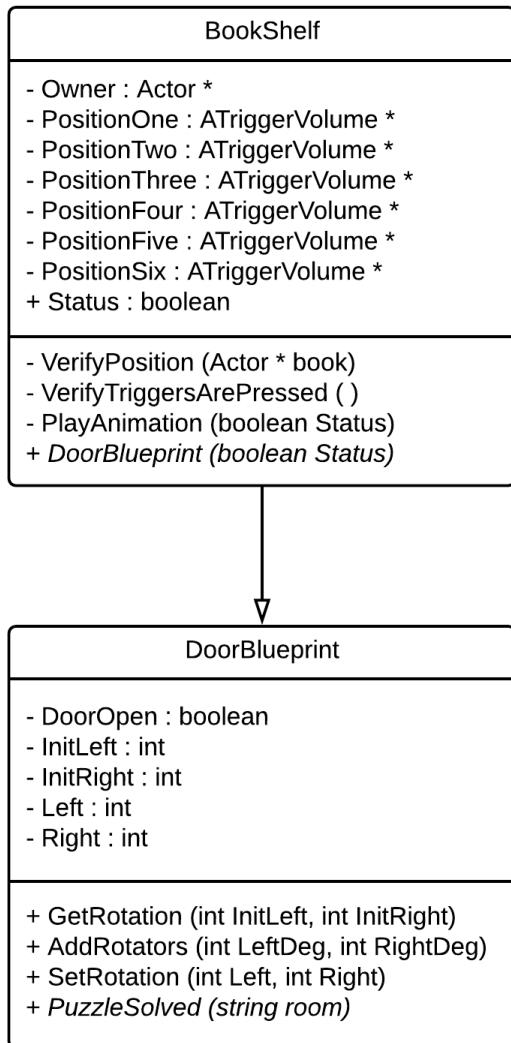


Figure 7: User Story #701 Class Diagram

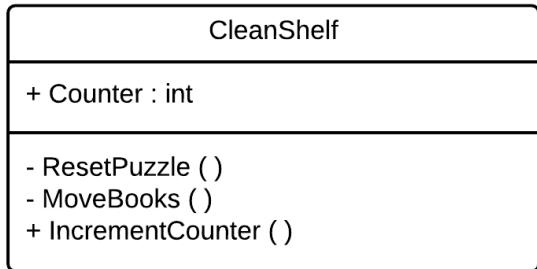


Figure 8: User Story #702 Class Diagram

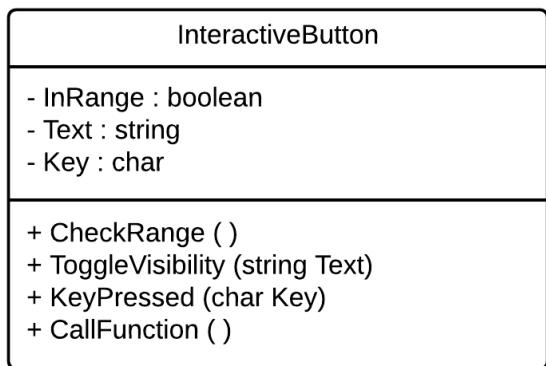


Figure 9: User Story #723 Class Diagram

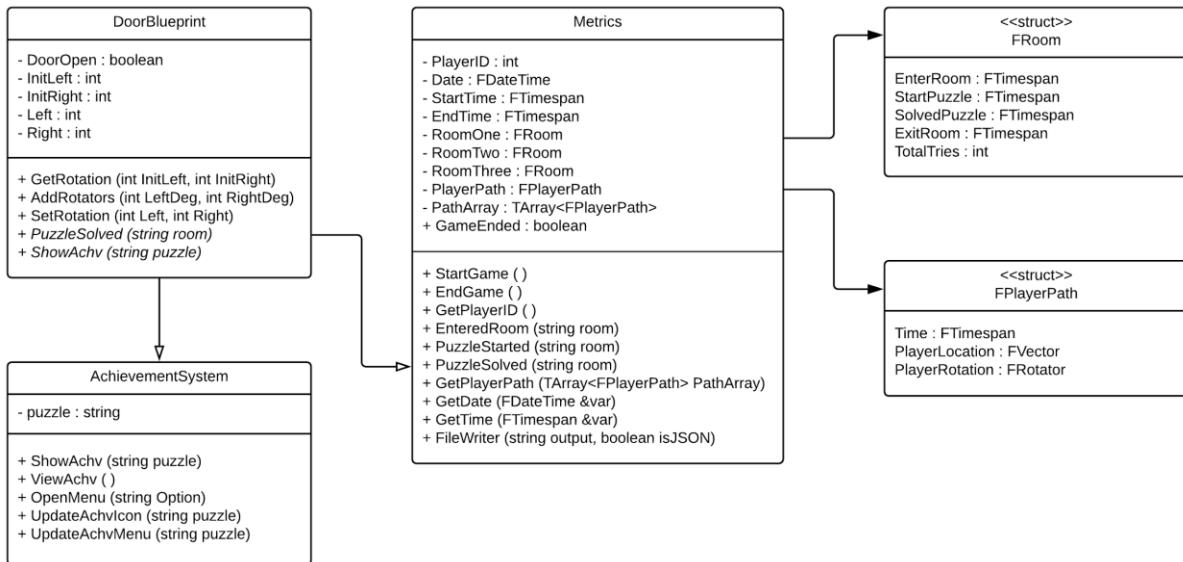


Figure 10: User Story #724 Class Diagram

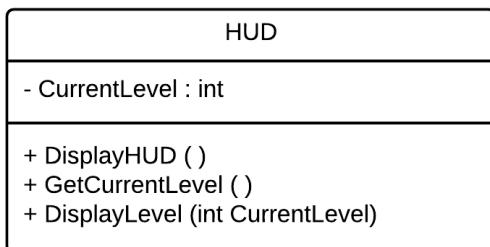


Figure 11: User Story #725 Class Diagram

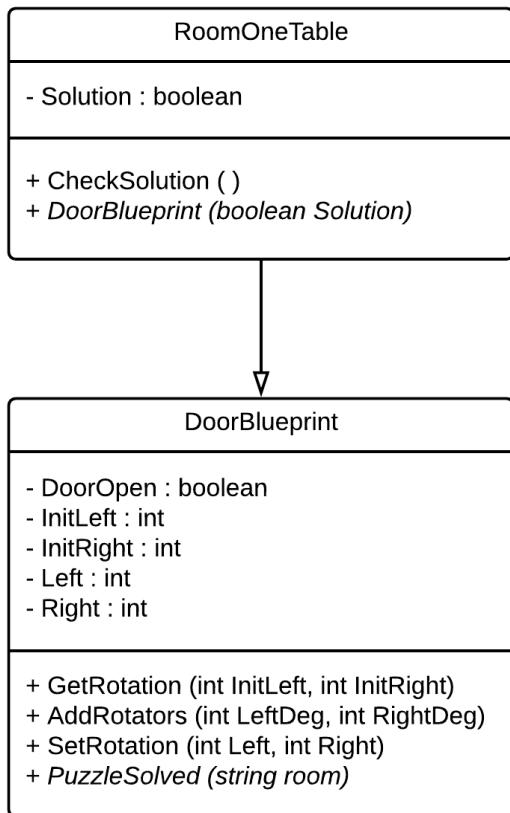


Figure 12: User Story #726 Class Diagram

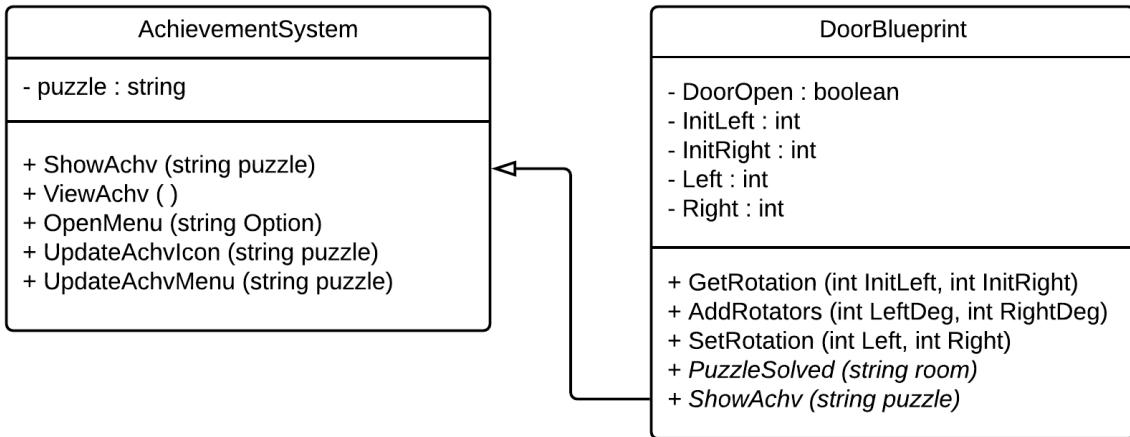


Figure 13: User Story #727 Class Diagram

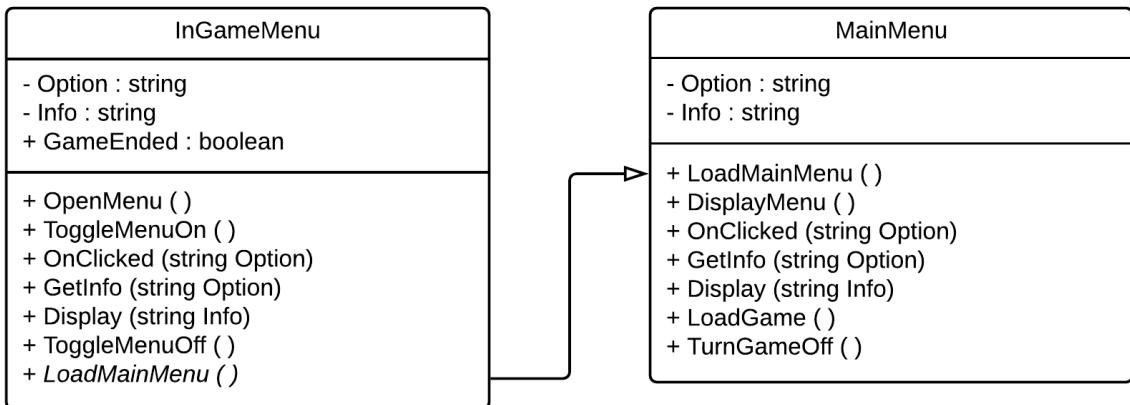


Figure 14: User Story #728 Class Diagram

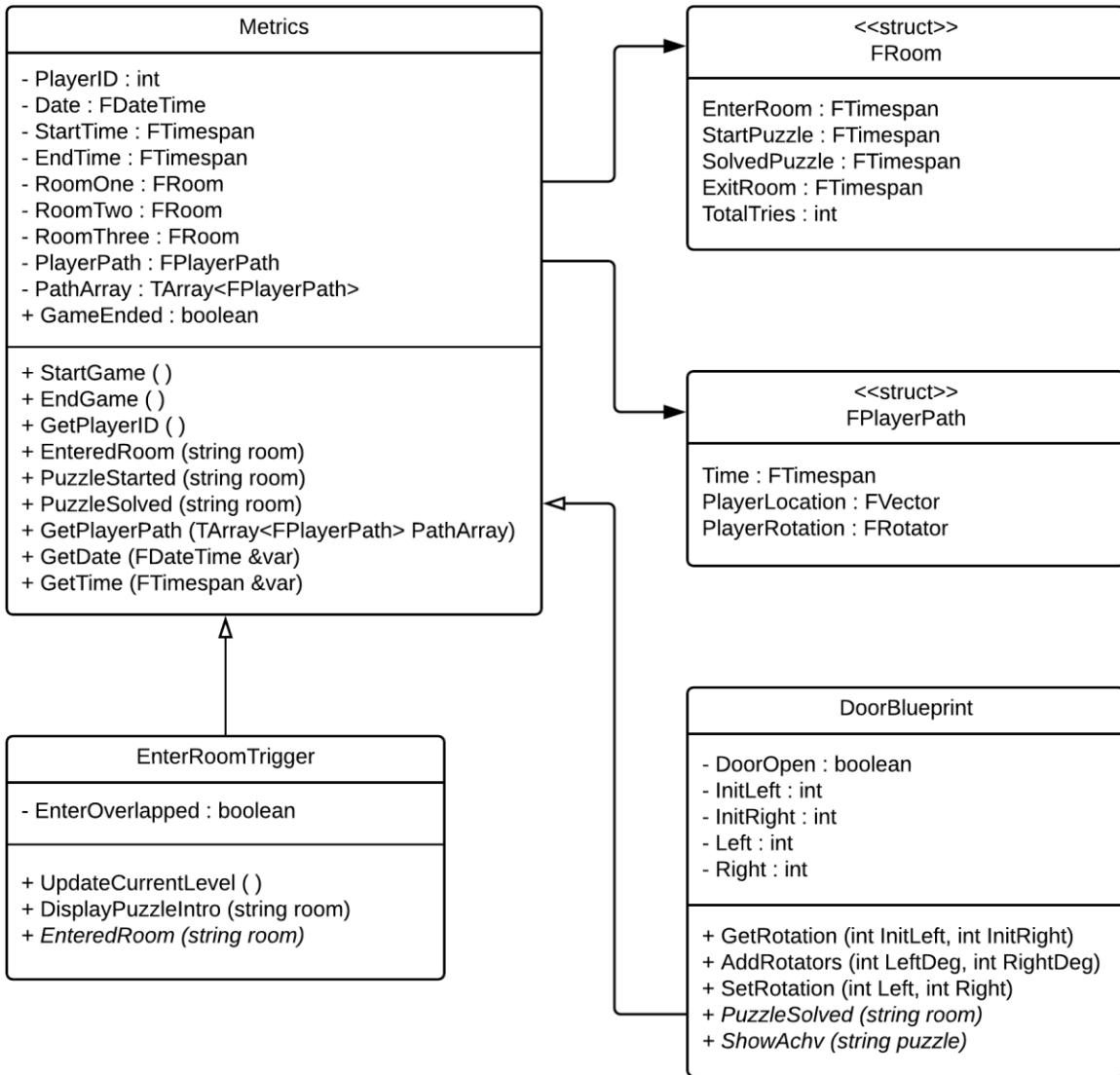


Figure 15: User Story #750 Class Diagram

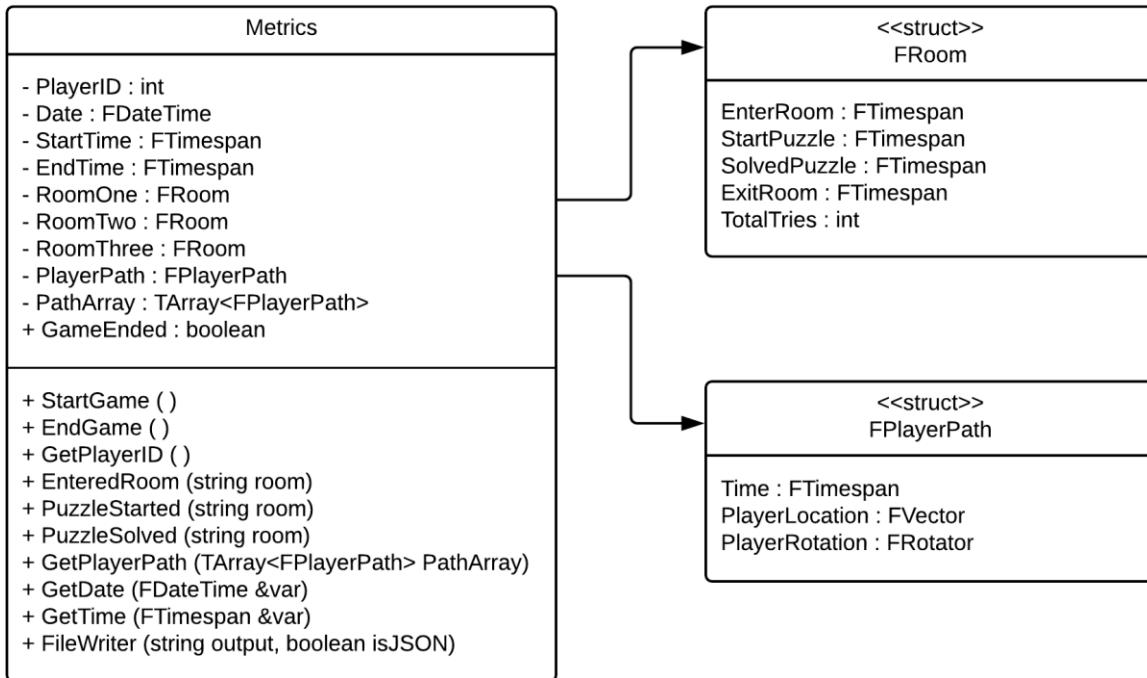


Figure 16: User Story #751 Class Diagram

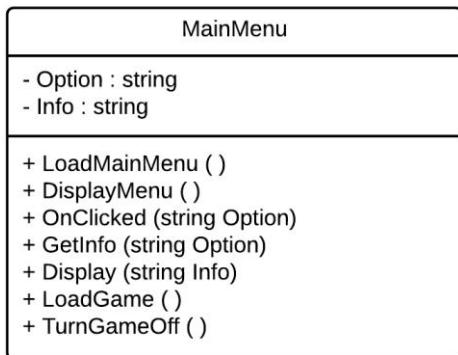


Figure 17: User Story #752 Class Diagram

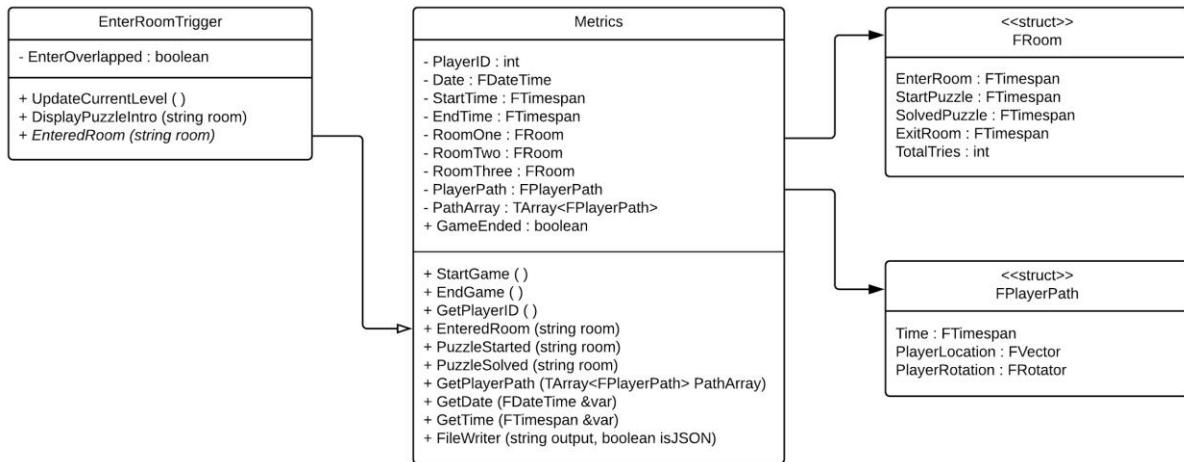


Figure 18: User Story #753 Class Diagram

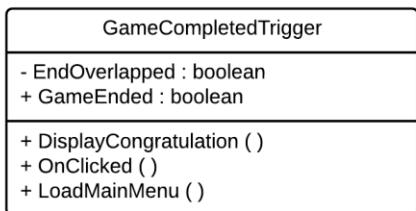


Figure 19: User Story #754 Class Diagram

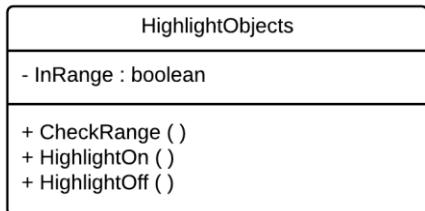


Figure 20: User Story #755 Class Diagram

Appendix B - User Interface Design



Figure 1: User Story #696 Move Objects



Figure 2: User Story #697 Add Objects to Cauldron



Figure 3: User Story #698 Heat Up Cauldron



Figure 4: User Story #699 Add Objects to Scale



Figure 5: User Story #700 Display Objects Weight

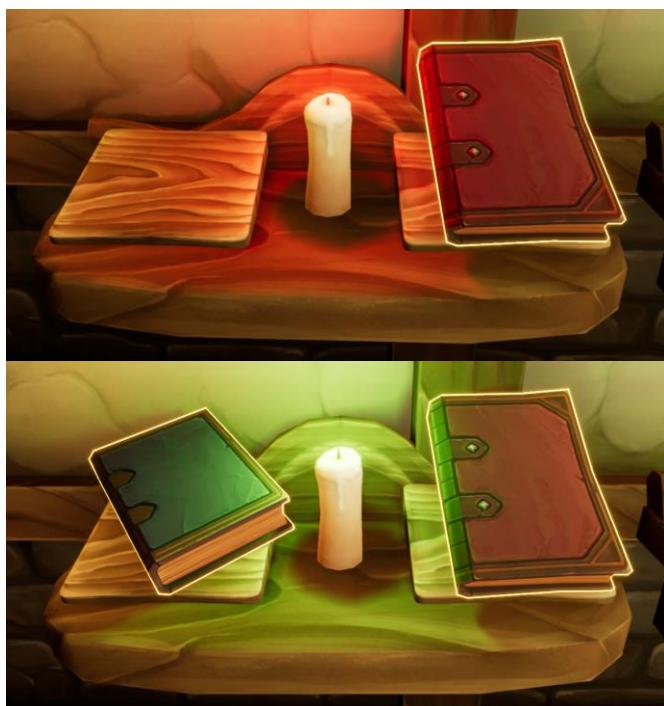


Figure 6: User Story #701 Place Books on Shelf



Figure 7: User Story #702 Clear Bookshelves



Figure 8: User Story #724 Open Doors



Figure 9: User Story #725 HUD



Figure 10: User Story #726 Room One Check Solution

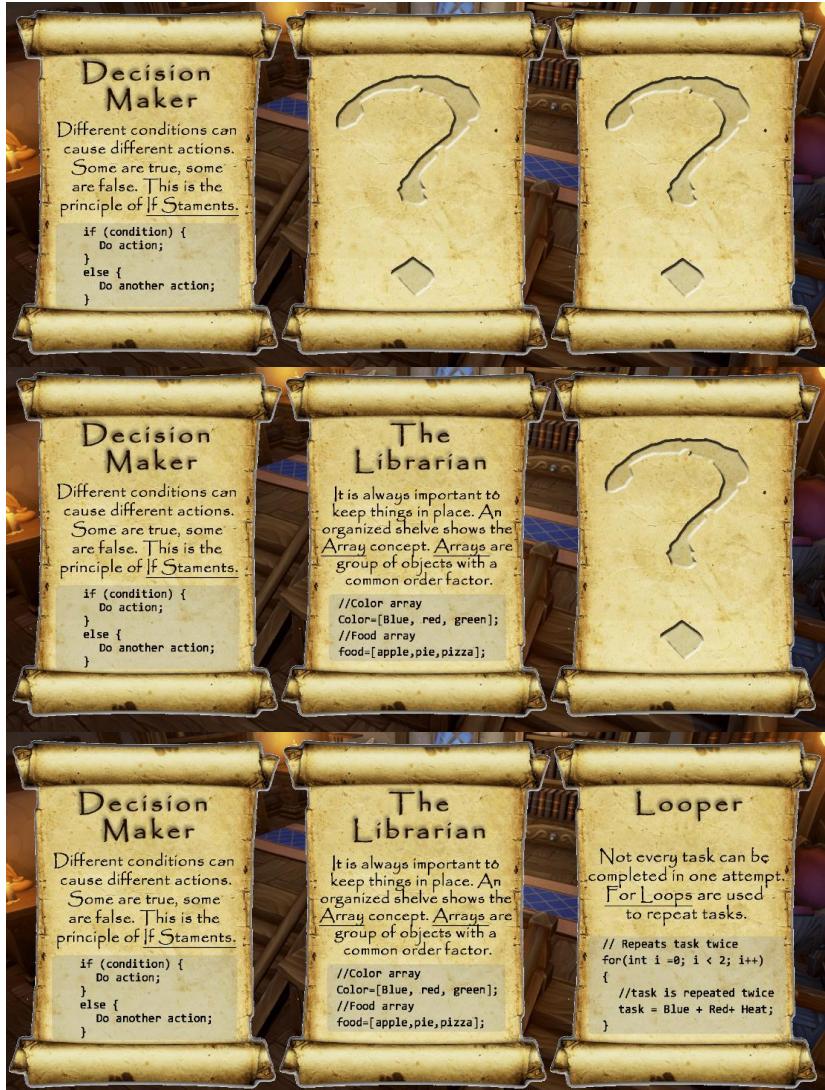


Figure 11: User Story #727 Achievement System

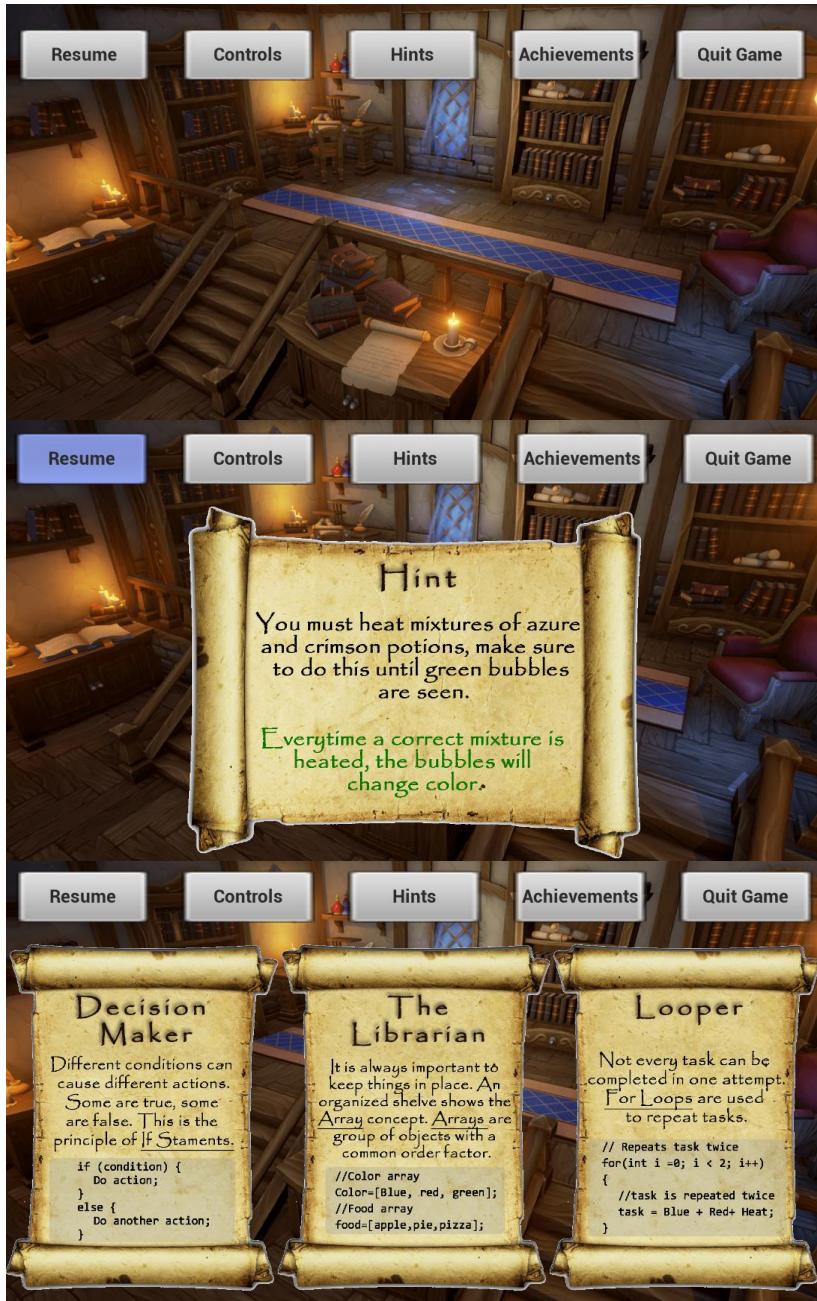


Figure 12: User Story #728 In-Game Menu

```
LogTemp: Warning: Date: 2018.04.17
LogTemp: Warning: Player ID: 817663
LogTemp: Warning: Game Started: +01:44:49.722
LogTemp: Warning: Room 1 Room 2 Room 3
LogTemp: Warning: Enter: +01:44:52.289 +01:45:09.730 +01:45:52.146
LogTemp: Warning: Start: +01:44:55.384 +01:45:14.358 +01:45:55.945
LogTemp: Warning: Solved: +01:45:04.849 +01:45:44.399 +01:46:07.101
LogTemp: Warning: Exit: +01:45:09.730 +01:45:52.146 +01:46:11.867
LogTemp: Warning: Tries: 0 0 2
LogTemp: Warning: Game Ended: +01:46:15.163
```

Figure 13: User Story #750 Collect Metrics

Figure 14: User Story #751 Export Metrics

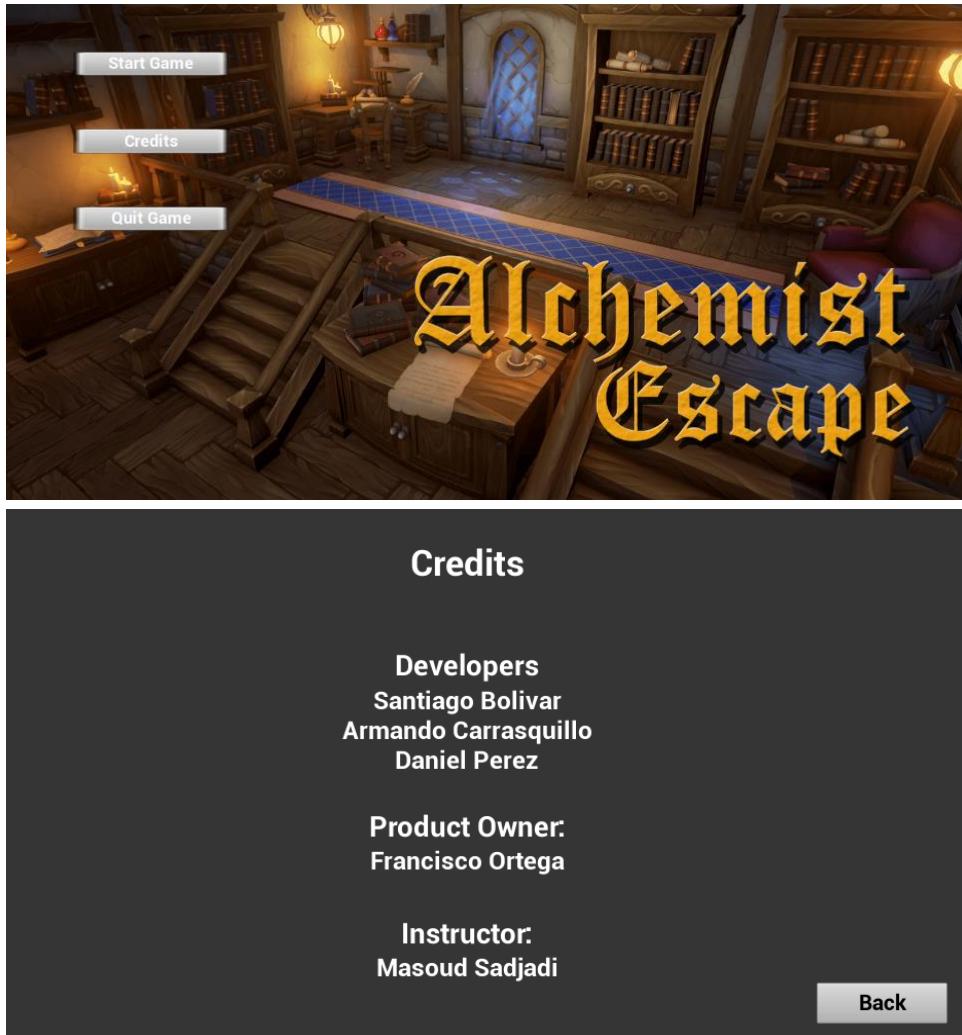


Figure 15: User Story #752 Main Menu



Figure 16: User Story #755 Highlight Objects

Appendix C - Sprint Review Reports

Sprint 1 Review Meeting Minutes

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 7:00 PM

End time: 7:30 PM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story #667 Setup Unreal Engine and Visual Studio (Computer 1)
- User Story #668 Setup Unreal Engine and Visual Studio (Computer 2)
- User Story #669 Setup Unreal Engine and Visual Studio (Computer 3)
- User Story #670 Research Unreal Engine Documentation (Developer 1)
- User Story #671 Research Unreal Engine Documentation (Developer 2)
- User Story #672 Research Unreal Engine Documentation (Developer 3)
- User Story #673 Research the C++ API (Developer 1)
- User Story #674 Research the C++ API (Developer 2)
- User Story #675 Research the C++ API (Developer 3)
- User Story #676 Research Game's that Women Play
- User Story #677 Setup VR Template Level
- User Story #678 Setup VR Headset with Unreal Engine

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- N/A

Sprint 2 Review Meeting Minutes

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story #679 Setup Level Streaming in Unreal Engine
- User Story #680 Split Level in Unreal into Sub-Levels
- User Story #681 Create Survey for Game
- User Story #682 Research Puzzles and Storyline for Game (Developer 1)
- User Story #683 Research Puzzles and Storyline for Game (Developer 2)
- User Story #684 Research Puzzles and Storyline for Game (Developer 3)
- User Story #685 Add Functionality to Objects in Unreal Engine (Developer 1)
- User Story #686 Add Functionality to Objects in Unreal Engine (Developer 2)
- User Story #687 Collect Data from Survey

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- N/A

Sprint 3 Review Meeting Minutes

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story #696 Implement Function to Move Objects
- User Story #697 Implement Adding Objects to Cauldron (Puzzle 1)
- User Story #698 Implement Interaction to Heat Up Cauldron (Puzzle 1)
- User Story #699 Implement Adding Objects to Scale (Puzzle 2)
- User Story #700 Implement Interaction to Display Objects Weight (Puzzle 2)
- User Story #701 Implement Placing Books on Shelf (Puzzle 3)
- User Story #702 Implement Interaction to Reset Puzzle (Puzzle 3)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- N/A

Sprint 4 Review Meeting Minutes

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story #723 Create Interactive Button
- User Story #724 Implement Functionality for Doors
- User Story #725 Implement HUD
- User Story #726 Implement Conditions to Escape Room (Puzzle 2)
- User Story #727 Implement Achievement System
- User Story #728 Implement In-Game Menu

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- N/A

Sprint 5 Review Meeting Minutes

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story #750 Collect Metrics
- User Story #751 Print Metrics
- User Story #752 Implement Main Menu
- User Story #753 Create Enter Room Trigger
- User Story #754 Create Game Completed Trigger
- User Story #755 Highlight Movable Objects

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- N/A

Sprint 6 Review Meeting Minutes

Attendees: Armado Carrasquillo, Daniel Perez, Santiago Bolivar

Start time: 9:30 AM

End time: 10:00 AM

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story #775 Write Research Paper for VIP (Developer 1)
- User Story #776 Write Research Paper for VIP (Developer 2)
- User Story #777 Write Research Paper for VIP (Developer 3)
- User Story #778 Port Game to VR (Developer 1)
- User Story #779 Port Game to VR (Developer 2)
- User Story #780 Port Game to VR (Developer 3)
- User Story #781 Alpha Testing (Developer 1)
- User Story #782 Alpha Testing (Developer 2)
- User Story #783 Alpha Testing (Developer 3)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- N/A

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

User Manuals

Unreal Engine 4 Documentation

- <https://docs.unrealengine.com/en-us>

Blender Documentation

- <https://docs.blender.org/>

Visual Studio Documentation

- <https://docs.microsoft.com/en-us/visualstudio/>

Installation/Maintenance Document

Installation

- Install Visual Studio
 - Download the latest version of Visual Studio Community Edition
 - <https://www.visualstudio.com/downloads/>
- Install Unreal Engine 4
 - Download Unreal Engine 4 ver 4.18 or higher
 - <https://www.unrealengine.com/download>
- Install Blender
 - Download the latest version of Blender
 - <https://www.blender.org/DOWNLOAD/>
- Pull the latest project revision from GitHub into a local repository.
- Run Unreal Engine 4 and open the project.
- Once the project is opened, select “file” and then select “generate c++ code” from the drop down menu to compile the c++ code.

Maintenance

- If the project needs to be migrated to a newer version of unreal engine, please be sure to comply with the code/naming conventions of the updated version.

Shortcomings/Wishlist Document

Things that were not completed in version 1.0

- Complete the VR implementation for the following:
 - HUD
 - Main menu
 - In-game menu
 - Achievement system
 - Triggers for entering a room

Things that we would like to be completed in the future

- Analyze the recorded data and improve on the current game based on the data.
- Add more puzzles to teach more CS concepts.
- Make a mobile version of the game.

REFERENCES

- YouTube Playlist:
<https://www.youtube.com/watch?v=0HYRRkzRQv8&list=PLG3lylzOg8Fj7BZp57IrP-CCdzeRuX5sl>