**Underground Hero**

**Requirement Analysis Document**

By Team Cobra

(John Lam, Matt Smith, Kyle Cousins, King Lo)

1. **Introduction**

The game is about a person without any superpowers but is clever and a good fighter. He has infiltrated a super villain lair to stop an apocalypse from happening. The purpose of this document is to identify the core functionality, the entity, boundary, and control objects, and create class and sequence diagram based on the use cases.

1. **Proposed System**
   1. **Overview**

Underground Hero is a text based adventure game. The game will consist of at least 32+ rooms, 8+ puzzles, and 8+ monsters. The game action will be done by the player’s input command. The player will encounter monster and puzzles in order to progress the game. Once the player defeated the final boss, the game will end.

* 1. **Functional Requirements**
     1. **Rooms(King)**

ID: FR1

Title: Navigate Room

DESC: The player will always be facing north. The player can type a direction to move between rooms. Upon, entering a room a general description of the room will be given to the player.

DEP: N/A

ID: FR2

Title: Look Room

DESC: It will provide the player with specific information of the room

DEP: N/A

ID: FR3

Title: Move Up

DESC: Moving up means to go in the ceiling such as the vents.

DEP: FR1

ID: FR4

Title: Move North

DESC: When the player input “North”, the player will move the entrance to the north.

DEP: FR1

ID: FR5

Title: Move South

DESC: When the player input “South”, the player will move the entrance to the South.

DEP: FR1

ID: FR6

Title: Move East

DESC: When the player input “East”, the player will move the entrance to the East.

DEP: FR1

ID: FR7

Title: Move West

DESC: When the player input “West”, the player will move the entrance to the West.

DEP: FR1

ID: FR8

Title: Hidden entrance

DESC: Throughout the game the player can find hidden entrance to hidden rooms or short cuts to previous rooms, but it will not give a description to the player about the hidden entrance. These entrances can be found by typing the direction. If the direction of the hidden entrance exists then the player will move through the hidden entrance to the designated room.

DEP: FR1

* + 1. **Game System(John)**

ID: FR9

Title: Create Game

DESC: A user opts for create new game, the system will generate a new game state and load the required assets and data for the system to run properly.

DEP: none

ID: FR10

TITLE: Exit Game

DESC: A user selects exit game in which the system enables a user to force exit the application during the main screen or during in game menu.

DEP: FR9

ID: FR11

TITLE: Save Game

DESC: A user selects save game which will enable the system to create or overwrite a game state.

DEP: FR9

ID: FR 12

TITLE: Load Game

DESC: The system will search and locate the game save state and if one is present, it will load the game state with all information about character, artifacts, puzzles, and monster progression.

DEP: FR11

ID: FR13

TITLE: Check Save State

DESC: System will check whether a save state is exists. If not the system will prompt the user that a file does not exist and the system.

DEP: none

ID: FR14

TITLE: Delete Save State

DESC: System will delete a saved state upon user request.

DEP: FR13

**2.2.3 Monster (Kyle)**

ID: FR15

Title: Enter Battle

DESC: When a player enters a boss room where the boss has not yet been defeated, the player enters a battle. When a player enters a non-boss room, a battle has a 10% chance of starting.

DEP: N/A

ID: FR16

TITLE: Player Attack

DESC: When the player types “Attack” in battle, the enemy’s HP stat will be reduced by the player’s current Strength stat minus the enemy’s current Defense stat. This is a Speed-determined action.

DEP: FR1, FR15

ID: FR17

TITLE: Player Defend

DESC: When the player types “Defend” in battle, their Defense will be multiplied by 2 until the next turn. When defending, the player always acts first.

DEP: FR1, FR15

ID: FR18

TITLE: Player Run

DESC: If the player does not wish to battle, then they may attempt escaping by typing “Run” and a direction. When running, the player always acts first.

- The enemy may be blocking an exit; if the player attempts to run in that direction, the text will say, “The enemy blocks your escape!” and the player will be given another chance to act.

- If the player only types “Run” without a direction, the text will say, “Which direction?” and the player will be given another chance to attack.

- If the player attempts to run in a direction that does not have an exit, the text will say, “Can’t run in that direction!” and the player will be given another chance to act.

- If the player attempts to run in a direction that has an exit and is not blocked by the enemy, then there will be a chance that the player will escape. This chance is determined by Speed. If the player’s Speed is greater than or equal to the enemy’s, then the player has a 100% chance of successfully escaping. If the player’s Speed is less than the enemy’s, then the player has a 50% chance of escaping successfully.

- If the player successfully escapes, then they will move into the room in the direction they ran.

- If the player does not successfully escape, then the text will say, “Can’t escape!” and the player will NOT get another chance to act until the next turn.

DEP: FR1, FR15

ID: FR19

TITLE: Player Heal

DESC: If the player types “HEAL” and has a health restoration consumable in their inventory, the player will use one to restore health and have their turn end. If there is no health items in the inventory, the player will be notified of the fact and be forced to select a different option. The player healing always goes first.

DEP: FR1, FR15

ID: FR20

TITLE: Enemy Action

DESC: When the player chooses their action, the enemy will randomly choose an action. The enemy’s actions include Attack, Defend, and up to two Special Actions. The percentage chance of each action is determined for each enemy on an individual basis in the enemy’s document.

DEP: FR1

ID: FR21

TITLE: Enemy Attack

DESC: The enemy reduces the player’s HP stat by the enemy’s Attack stat minus the player’s Defense stat. This is a Speed-determined action.

DEP: FR1, FR15

ID: FR22

TITLE: Enemy Defend

DESC: The enemy’s defense is multiplied by 2 (or if Defense is 0, increased to 1) until the next turn. This action always goes first.

DEP: FR1, FR15

ID: FR23

TITLE: Enemy Special Action

DESC: The enemy performs a special action. What this special action does is determined in the enemy’s document.

DEP: FR1, FR15

ID: FR24

TITLE: Player Dies

DESC: If the player’s HP stat is reduced to zero or less, then the player dies and is sent to the Game Over screen.

DEP: FR1

ID: FR25

TITLE: Enemy Dies

DESC: If the enemy’s HP is reduced to zero or less, the enemy dies and combat is over.

DEP: FR1

**2.2.4 Puzzle (Matt)**

ID: FR26

Title: Initiate Puzzle

DESC: Upon entering a room with a puzzle, the player should be able to initiate the puzzle.

DEP: FR1

ID: FR27

Title: Use Puzzle Item

DESC: The player must use a key item to solve a puzzle.

DEP: FR26

ID: FR28

Title: Monster Puzzle

Desc: A monster will initiate a puzzle with the player as a attack. when the player solve the puzzle the monster will die immediately

Dep: FR21

ID: FR29

TITLE: Exit Puzzle

DESC: When the player is in a non monster puzzle, the player should be able to leave the puzzle without entering an answer.

DEP: FR26

ID: FR30

TITLE: Enter Answer

DESC: For non-object puzzle the player can enter the puzzle answers.

DEP: FR26

**2.2.5 Inventory**

ID: FR31

TITLE: Open Inventory

DESC: The user inputs into text parser open inventory and the system will output and display the list of artifacts in a table format using text. The order of artifacts will be determined by their types, and IDs. It accessible at all times.

DEP: none

ID: FR32

TITLE: Use Item

DESC: The user opts for use item in which the system will utilize an artifact and display its usage and prompts the user again for confirmation.

DEP: FR31, FR33

ID: FR33

TITLE: Check Item Compatible

DESC: The system will check artifact’s type and will respond to appropriate conditions depending on the user’s current environment and situation. System will compare artifact’s type and its usability to check whether it is possible to use in certain conditions. If able to utilize an artifact, the artifact will leave from the user’s inventory and vice versa.

DEP: none

ID: FR34

TITLE: View Item

DESC: User opts for select item in which the system will display more detailed information including type, usage, and any additional effects that affects to the environment or the user.

DEP: FR31

* 1. **Non-Functional Requirements**
* System should respond to user input within ½ second.
* The different features of the puzzle window should be clear and easy to use. Text parser should always be available. Journal should be directly opened and user should not have to open inventory first. Puzzle window should close automatically after user enters an answer.
* The system must store the data for only one saved game at a time. If a new game is started with a current saved game stored, then the current saved game data will remain until the user attempts to save the new game, in which case the new game’s data will overwrite the existing saved game’s data.
* Reliability that the system provides a response to user dialog. Provide feedback on 100% of user input.
* Internet is only needed to download the app and install any updates.
* Admin should have a specific way to log in to game. Prevent anyone who does not have administrative rights from editing game data such as monster and puzzle locations or HP for user or monsters.
* Test environments should be built for the application to allow testing of functions such as monster battles and puzzles.
* The software will be using text files to input and output data. Data containing monster, artifact, and puzzle assets can only be read, while saved data that are stored is able to read and write.
  1. **System Models**
     1. **Use case model**
        1. **Use cases description**
           1. **Room(King)**

|  |  |
| --- | --- |
| **Use Case Name** | Navigate Room |
| **Participating Actors** | player |
| **flow of events** | 1. Player enter a direction commands   2. System moves player to the designated room with a general description of the room |
| **entry condition** | Player is in a room |
| **exit condition** | Player moves to a new room |
| **Quality Requirement** | Player successfully enters a new room. |

|  |  |
| --- | --- |
| **Use Case Name** | View Room |
| **Participating Actors** | player |
| **flow of events** | 1. The player enter the look command   2. system display  specific information  about the room |
| **entry condition** | player is in the room |
| **exit condition** | player is shown specific details about the room |
| **Quality Requirement** | player successfully viewed the room |

* + - * 1. **Game System(John)**

|  |  |
| --- | --- |
| **Use Case Name** | Start Game |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player enters into text parser to initiate and generate a game state with assets loaded. |
| **Entry Condition** | * Application is executed |
| **Exit Condition** | * Game is loaded. |
| **Quality Requirements** | * The game is not corrupted and must be able to execute. |

|  |  |
| --- | --- |
| **Use Case Name** | Save Game |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player enters into text parser to create or overwrite a save state. |
| **Entry Condition** | * Game is already initiated. |
| **Exit Condition** | * Game save state is successfully created. |
| **Quality Requirements** | * Save state is empty, if not it is overwritten. |

|  |  |
| --- | --- |
| **Use Case Name** | Load Game |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player enters into text parser to load a save state. |
| **Entry Condition** | * A game save state must exist. |
| **Exit Condition** | * Game save state is successfully loaded. * Game loads all the player’s progression. |
| **Quality Requirements** | * Game save state is not corrupted or tampered. |

|  |  |
| --- | --- |
| **Use Case Name** | Exit Game |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player opts to exit game either through exiting the game application or entering into text parser. |
| **Entry Condition** | * Game is executed. |
| **Exit Condition** | * Game has been successfully closed. |
| **Quality Requirements** | * None |

* + - * 1. **Monster(Kyle)**

|  |  |
| --- | --- |
| **Use Case Name** | Attack |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player inputs the “Attack” command 2. The system allows whoever has the higher speed to have the first action 3. The system calculates the damage output toward the monster 4. The monster’s HP is reduced by this output |
| **Entry Condition** | * Player is in a battle and it is his/her turn |
| **Exit Condition** | * The player attacks |
| **Quality Requirements** |  |

|  |  |
| --- | --- |
| **Use Case Name** | Defend |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player inputs the “Defend” command 2. The system multiplies the player’s current defense by x2 for the duration of the monster’s next action |
| **Entry Condition** | * Player is in a battle and it is his/her turn |
| **Exit Condition** | * The player defends |
| **Quality Requirements** |  |

|  |  |
| --- | --- |
| **Use Case Name** | Heal |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player inputs the “Heal” command 2. The system restores an amount of HP depending on which consumable was selected 3. The system removes the consumable that was used from the player’s inventory |
| **Entry Condition** | * Player is in a battle and it is his/her turn and has at least one health consumable in their inventory |
| **Exit Condition** | * Player uses a health consumable |
| **Quality Requirements** |  |

|  |  |
| --- | --- |
| **Use Case Name** | Run |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player inputs the “Run” command 2. The system asks the player which direction they want to run 3. The player inputs to run in a direction that has an exit 4. The system determines through the player’s speed vs the enemy’s speed the percent chance that escape is successful |
| **Entry Condition** | * Player is in a battle that can be escapable and it is his/her turn |
| **Exit Condition** | * Player either escapes the battle successfully or fails the attempt |
| **Quality Requirements** |  |

* + - * 1. **Puzzle(Matt)**

|  |  |
| --- | --- |
| **Use Case Name** | Initiate Puzzle |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player finds a puzzle problem to solve.    1. Initiates a battle against Puzzle Monster    2. Initiates a Logic Puzzle    3. Initiates a Puzzle that requires a Key Item |
| **Entry Condition** | * Player is in a room that contains a puzzle problem. |
| **Exit Condition** | * If a non-monster puzzle, able to leave. |
| **Quality Requirements** | * None |

|  |  |
| --- | --- |
| **Use Case Name** | Solve Puzzle |
| **Participating Actors** | Player |
| **Flow of Events** | 1. If Monster Puzzle    1. Player defeats monster 2. If Logic Puzzle    1. Player solves puzzle logically 3. If Key Puzzle    1. Player uses key item to solve puzzle |
| **Entry Condition** | * Must be in the middle of a puzzle. |
| **Exit Condition** | * Puzzle is solved |
| **Quality Requirements** | * None |

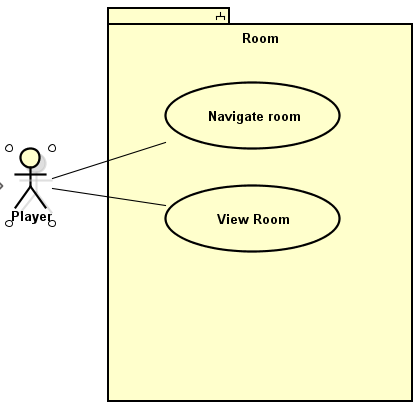
|  |  |
| --- | --- |
| **Use Case Name** | Exit Puzzle |
| **Participating Actors** | Player |
| **Flow of Events** | 1. Player exits puzzle. |
| **Entry Condition** | * Must initiate puzzle prior to exiting. |
| **Exit Condition** | * If a non-monster puzzle, able to leave |
| **Quality Requirements** | * None |

* + - * 1. **Inventory**

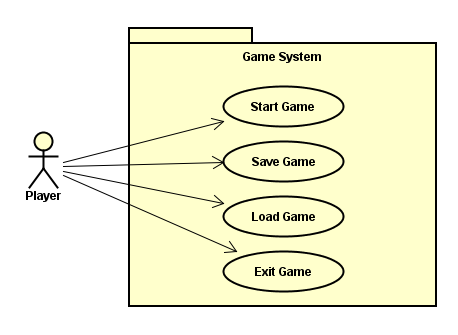
|  |  |
| --- | --- |
| **Use Case Name** | Open Inventory |
| **Participating Actors** | Player |
| **Flow of events** | 1. The player input the command “open inventory”   2.The system respond by giving a list of items in the inventory |
| **Entry condition** | Player is in the game |
| **Exit Condition** | Player opened the inventory |
| **Quality Requirement** | Player was able to open the inventory |

|  |  |
| --- | --- |
| **Use Case Name** | Use Item |
| **Participating Actors** | Player |
| **Flow of events** | 1. player type the name of the item to use   2.the system will scan for the item in the inventory and process the item name. tells the player that item has been used. |
| **Entry condition** | * Player opened the inventory * Item exist in inventory |
| **Exit Condition** | Player used the item |
| **Quality Requirement** | Player was able to use the item |

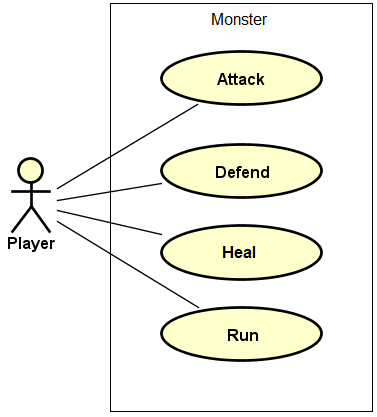
* + - 1. **Use case diagrams**
         1. **Room**



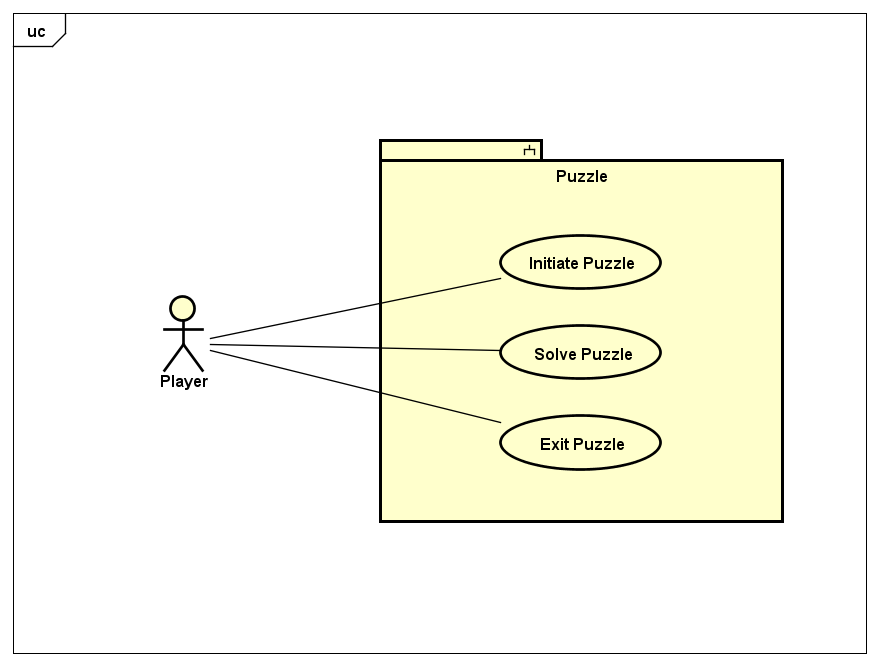
* + - * 1. **Game System**



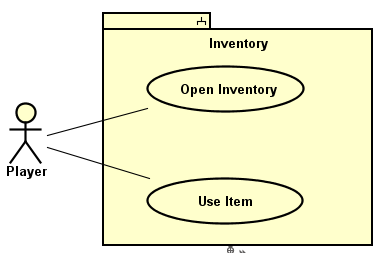
* + - * 1. **Monster**



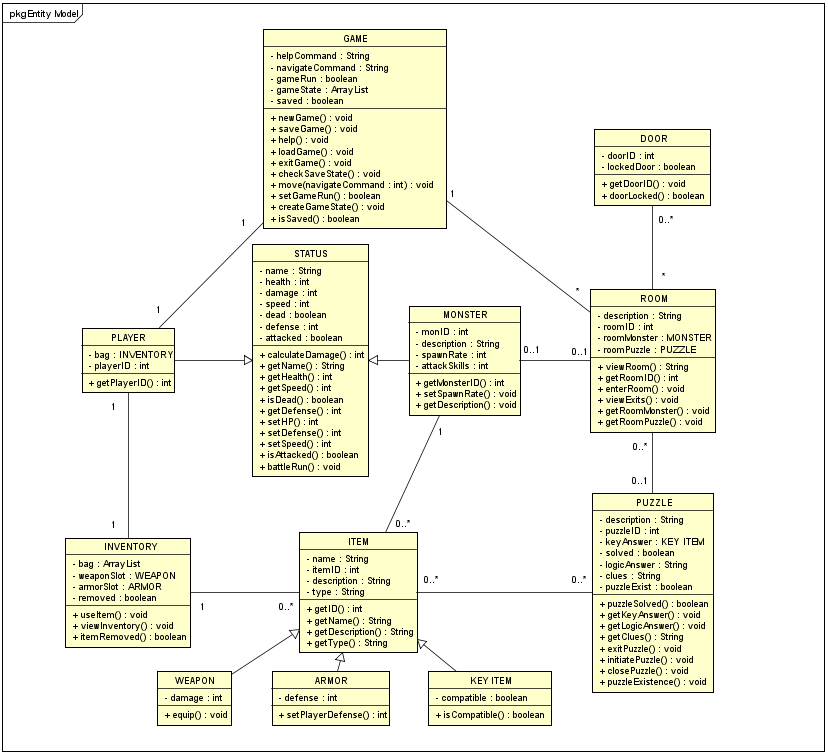
* + - * 1. **Puzzle**



* + - * 1. **Inventory**



* + 1. **Object Model**
       1. **Class Diagram**

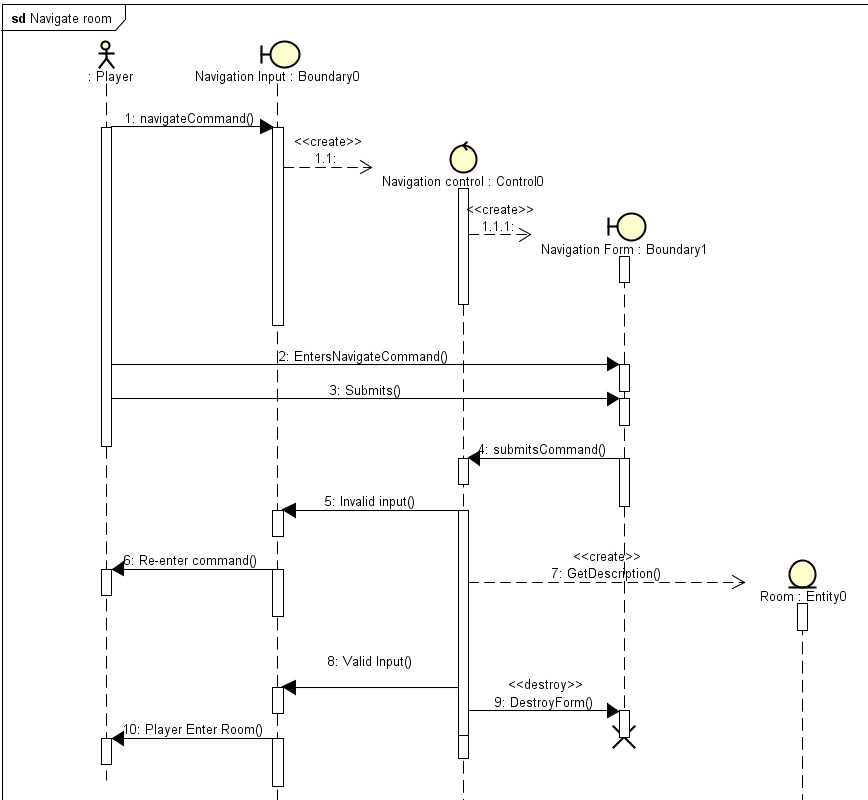


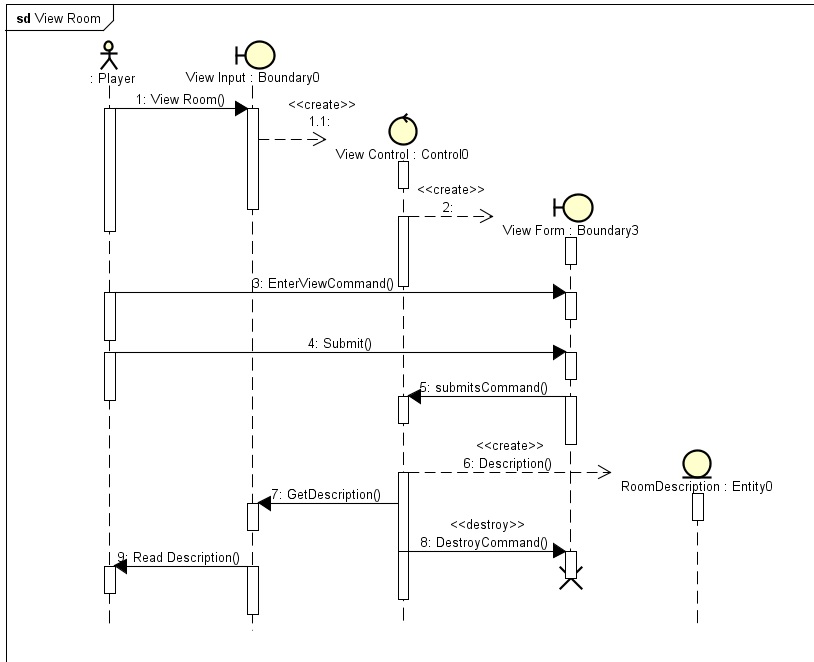
* + - 1. **Class Diagram description**

|  |  |
| --- | --- |
| **Class Name** | **Description(Responsibilities)** |
| **GAME** | Responsible for creating, saving, and overwriting game states. It contains functions for player’s movements within the game. |
| **ROOM** | Contains multiple doors or exits that the player can traverse through. It can contain a monster or puzzle within a room and a detailed description of a room will help players uniquely differentiate one room to another. |
| **DOOR** | Contains description of the door in the room. |
| **PUZZLE** | An obstacle that is required to progress through the game. Requirements include logical puzzles and/or key items that are used to solve them. |
| **STATUS** | Parent class for PLAYER and MONSTER. Setting and obtaining values for quantifiable attributes and calculating them between the player and monster. |
| **PLAYER** | Child class of STATUS. An entity that the user will control within the game system and their interaction with the inventory. |
| **MONSTER** | Child class of STATUS. An entity that acts to halt the player’s progression. Some monsters can act as puzzles and some are required to progress through the game. Monster will have a probability to spawn in certain rooms. |
| **INVENTORY** | To contain items in an infinite quantity. A player can view and use items if item is obtained. |
| **ITEM** | Parent class of WEAPON, ARMOR, and KEY ITEM. The class contains names and description that is divided into subsection depending on the type. Each item is to be uniquely identified by ID’s and its relationship and purpose between other classes. |
| **WEAPON** | An equippable item type that assists the players to progress towards the goal of the game. |
| **ARMOR** | An equippable item type that can be worned by the player to assist them against battle a monster. |
| **KEY ITEM** | A usable item type that is only usable depending on the condition of a certain puzzle or monster to progress through the game. |

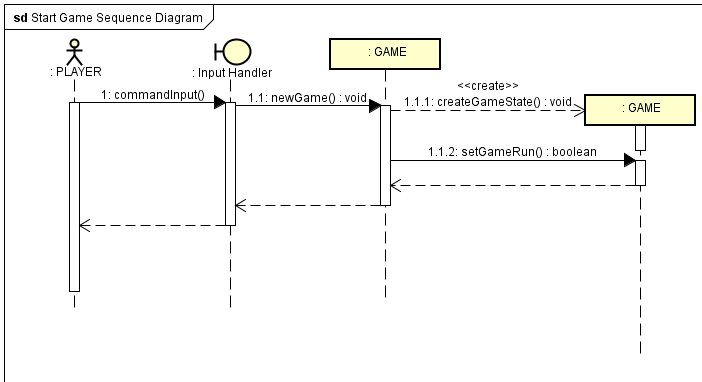
* + 1. **Dynamic Model**
       1. **Sequence Diagrams**

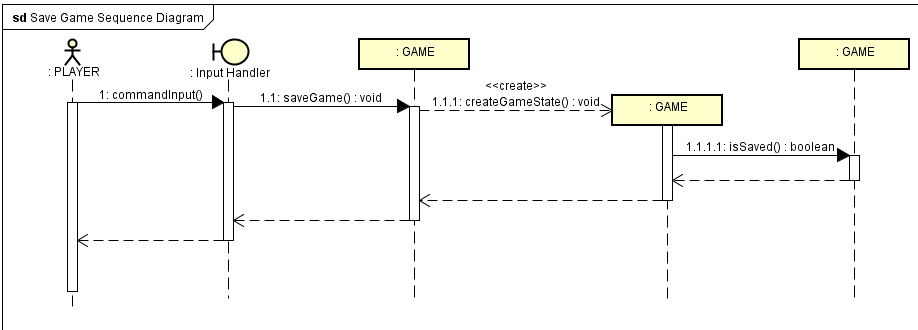
**ROOM SEQUENCE DIAGRAMS(King)**

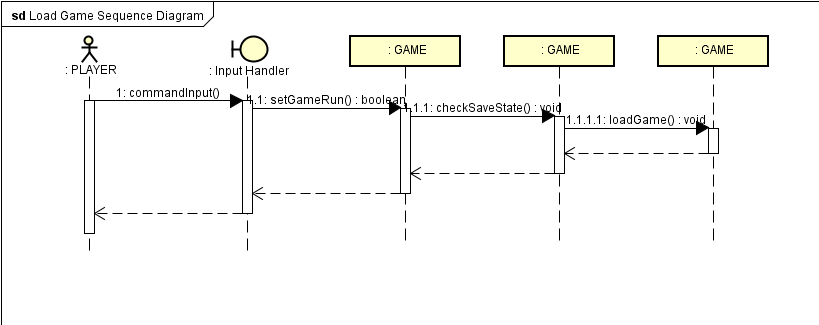


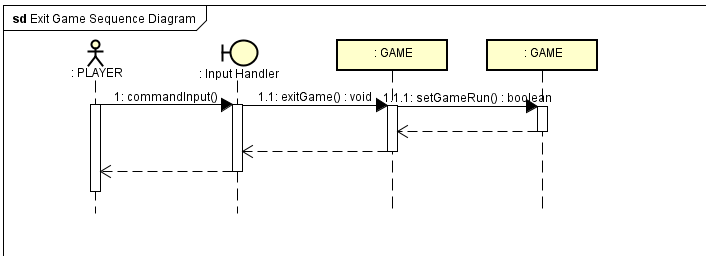


**GAME SYSTEM SEQUENCE DIAGRAMS(John)**

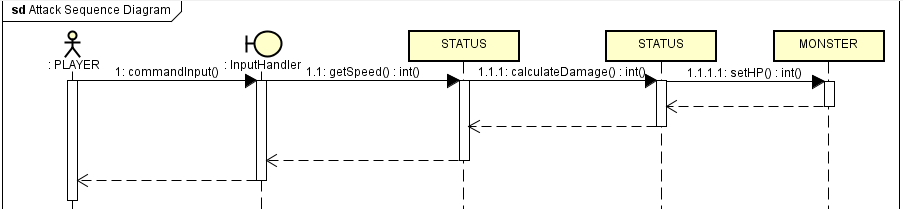


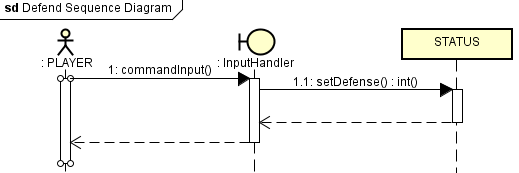


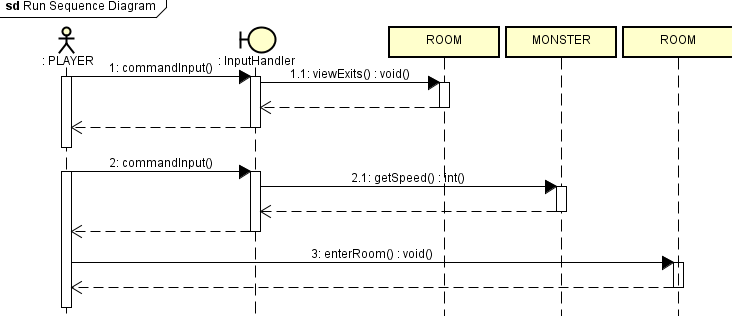




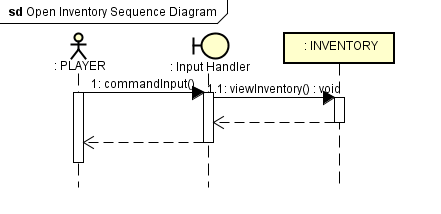
**MONSTER SEQUENCE DIAGRAMS(Kyle)**

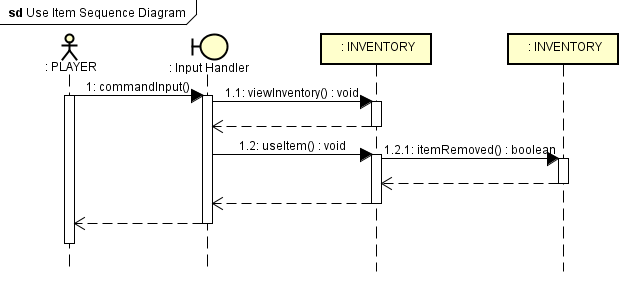




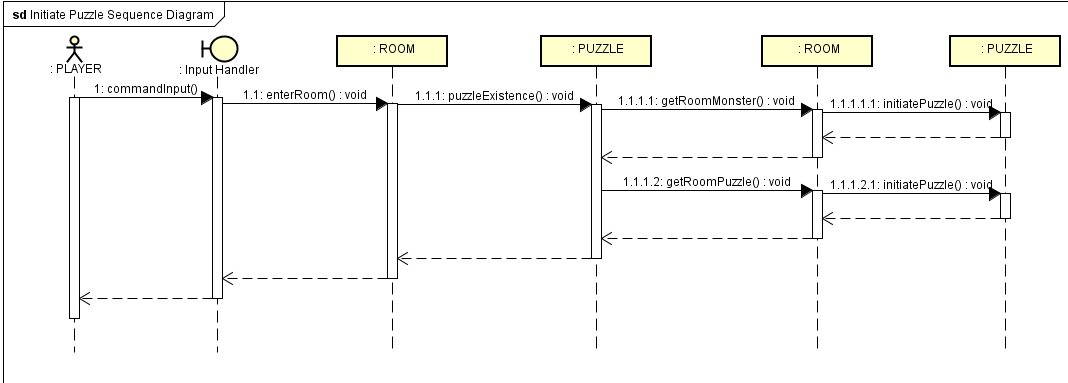


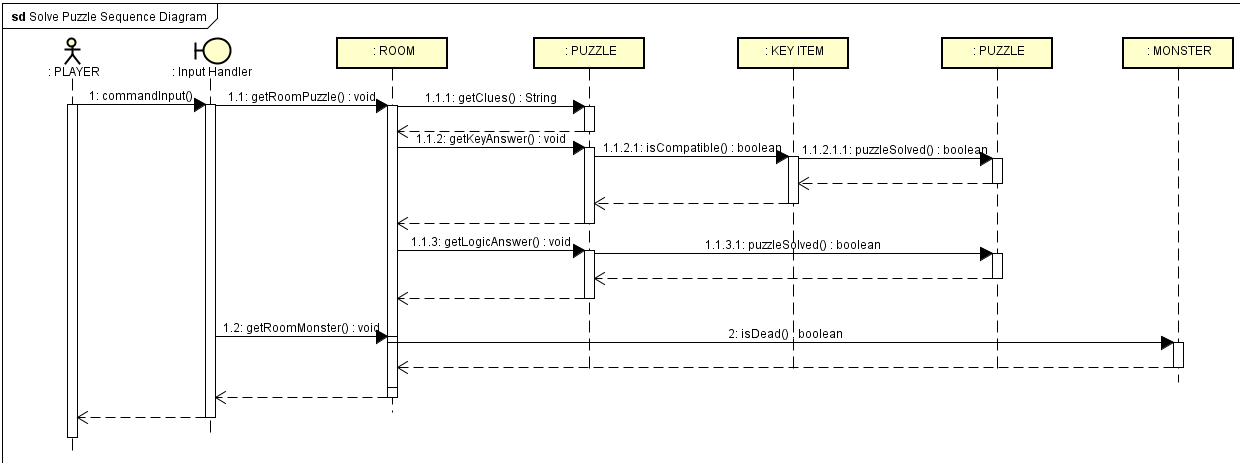
**INVENTORY SEQUENCE DIAGRAMS**

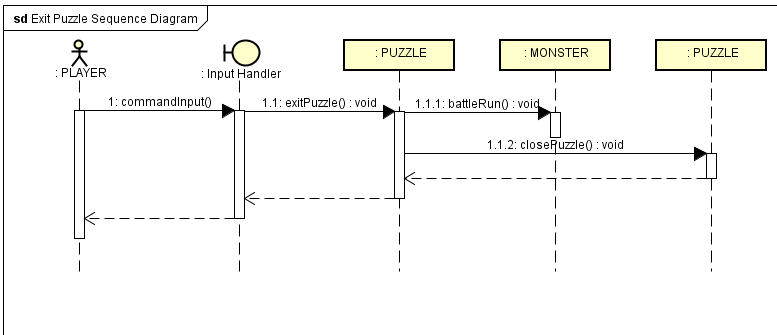




**PUZZLE SEQUENCE DIAGRAMS(Matt)**







* + 1. **User interface**
       1. **Screen mockups (Extra Credit)**
       2. **Navigation**

|  |  |
| --- | --- |
| **COMMANDS** | **DESCRIPTION** |
| HELP | Provides the player a detailed list of commands and its functionality that can be used within the game. |
| VIEW ROOM | Given a detailed description of the room and anything within it including objects and exits. |
| MOVE NORTH | If able, moves player to NORTH exit. |
| MOVE SOUTH | If able, moves player to SOUTH exit. |
| MOVE WEST | If able, moves player to SOUTH exit. |
| MOVE EAST | If able, moves player to EAST exit. |
| MOVE UP | If able, moves player to any entrances above them in a room. |
| EXIT GAME | Enables a player to force the application to terminate. |
| SAVE GAME | Enables a player to save or overwrite a game state. |
| LOAD GAME | Enables a player to load a game state. |
| START GAME | Enables a player to create a new game state. |
| ATTACK | Attacks the selected monster. |
| DEFEND | Defends against the monster’s attack. |
| RUN | If able, flees away from battle against monster. |
| HEAL | If healing item exists, restores health points to player. |
| OPEN INVENTORY | Enables a player to open inventory bag. |
| VIEW ITEM | Enables a player to view selected item and gives detailed information about it. |
| EQUIP ITEM | If item type is weapon or armor, enables player to equip item. |
| USE ITEM | If able, player uses item. |