Team Avengers

Analysis Document

1. Introduction (Estephanie Gonzalez)

A Player can traverse the game through a menu. The game will include 3 floors with multiple rooms per floor that may contain a monster or a puzzle to solve. The system will track a score based on defeated monsters, and puzzles solved. The goal is to reach the last room. Administrators have the ability to edit game content and controls.

2. Proposed System (Estephanie Gonzalez)

The system is a standalone product. Enabling users to play the game will be the main functionality. The game/user data is stored into text files for the system to have access to read and write files. A constant internet connection is not required by the system with the exception when downloading the software.

2.1 Overview

Will include functional requirements for each feature, along with non-functional requirements for the whole system.

2.2 Functional Requirements

Command menu (Jory Alexander)	**3.2.1.1 - Functional Requirement 1.1** **ID: FR1.1** TITLE: Create New Game DESC: A user opts for create new game state, the system will generate a new game state and load the required assets and data for the system to run properly. DEP: n/a **3.2.1.2 - Functional Requirement 1.2** **ID: FR1.2** 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: FR1.1 **3.2.1.3 - Functional Requirement 1.3** **ID: FR1.3** TITLE: Save Game DESC: A user selects save game which will enable the system to create or overwrite a game state. DEP: FR1.1 **3.2.1.4 - Functional Requirement 1.4** **ID: FR1.4** 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: FR1.5 **3.2.1.5 - Functional Requirement 1.5** **ID: FR1.5** 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: n/a **3.2.1.6 - Functional Requirement 1.6** **ID: FR1.6** TITLE: Open Inventory DESC: The user selects open inventory and the system will output and display the list of artifacts in text. The order of artifacts will be determined by their types, and IDs. DEP: n/a **3.2.1.7 - Functional Requirement 1.7** **ID: FR1.7** TITLE: Check Inventory Space DESC: System checks for empty space in the container that holds artifacts. The container cannot exceed a certain limit. DEP: FR1.6

	3.2.1.8 - Functional Requirement 1.8 **ID: FR1.8** TITLE: Select 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: FR1.6 **3.2.1.9 - Functional Requirement 1.9** **ID: FR1.9** TITLE: Check Item Type DESC: The system checks for an artifact's type and unique ID. DEP: n/a **3.2.1.10 - Functional Requirement 1.10** **ID: FR1.11** 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: FR1.9 **3.2.1.11 - Functional Requirement 1.11** **ID: FR1.12** 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: FR1.9 **3.2.1.12 - Functional Requirement 1.12** **ID: FR1.13** TITLE: Record Time DESC: The system will display a number counter to track how long a user has been playing the game. DEP: n/a **3.2.1.12 - Functional Requirement 1.13** **ID: FR1.14** TITLE: Display Score DESC: The system will continuously display a number counter for users to keep track of score. DEP: FR1.15 **3.2.1.12 - Functional Requirement 1.14** **ID: FR1.15** TITLE: Calculate Score DESC: The system will constantly check for user's activities within the game and calculate a score depending on activities regarding monsters, artifacts, puzzles, and time. The given dependencies will specify the conditions to calculate overall score. DEP: FR2.7(Defeat Monster), FR2.14(Lose Battle), FR3.8(Save Puzzle Attempts), FR4.4(Pick up artifacts/objects), FR1.13(Record Time)

Monster (Aaron Knobloch)	**3.2.2.1 - Functional Requirement 2.1** **ID: FR2.1** TITLE: Encounter Monster DESC: The system runs a Monster Check (FR2.2), then presents a list of contextual actions the player can take, and performs an Action Restrict (FR2.3) for the duration of the battle. DEP: N/A **3.2.2.2 - Functional Requirement 2.2** **ID: FR2.2** TITLE: Monster Check DESC: The system checks which monsters are tied to the current room, compares the encounter probability on each of those monsters, and selects a monster to present to the player based on the probability percentages. DEP: FR2.1 **3.2.2.3 - Functional Requirement 2.3** **ID: FR2.3** TITLE: Action Restrict DESC: The system temporarily disables the usage of certain actions the player could previously take when out of a battle, such as interacting with the room, leaving the room, and saving or loading a game. DEP: FR2.1 **3.2.2.4 - Functional Requirement 2.4** **ID: FR2.4** TITLE: Attack Monster DESC: The system runs a Player Attack Calculation (FR2.5), and displays the damage amount done to the monster. DEP: FR2.1 **3.2.2.5 - Functional Requirement 2.5** **ID: FR2.5** TITLE: Player Attack Calculation DESC: The system checks the player's base attack stat integer, adds any modifying integers in play (equipped weapons, buffs/debuffs) and sets this sum as player total attack power. System then subtracts the monster's current HP integer with player total attack power. DEP: FR2.4 **3.2.2.6 - Functional Requirement 2.6** **ID: FR2.6** TITLE: Enemy HP Check DESC: The system checks the current HP of the enemy after each party's attack. If the enemy's HP is less than or equal to zero, run Defeat Monster (FR2.7) DEP: FR2.4, FR2.11 **3.2.2.7 - Functional Requirement 2.7** **ID: FR2.7** TITLE: Defeat Monster DESC: The system declares to the player that the monster is defeated, runs an Item Check (FR2.8) and stores those items in the player's inventory. System then runs an Action Unrestrict (FR2.9). DEP: FR2.6 **3.2.2.8 - Functional Requirement 2.8**

ID: FR2.8
TITLE: Item Check
DESC: The system checks which items are tied to the monster, compares the dropping probability on each of those items, and determines from there which items are presented to the player.
DEP: FR2.7

3.2.2.9 - Functional Requirement 2.9

ID: FR2.9
TITLE: Action Un-restrict
DESC: The system re-enables the usage of actions that were restricted during the battle (FR2.3)
DEP: FR2.3, FR2.7

3.2.2.10 - Functional Requirement 2.10

ID: FR2.10
TITLE: Escape Battle
DESC: The system randomly determines if the player can escape the battle during their turn, with a 15% chance of success. (25% if the Tengu's Fan is equipped) If successful, the system removes the monster from the room and tells the player that they have successfully escaped from the battle. If unsuccessful, the system tells the player that they have failed in escaping, and grants the monster their turn.
DEP: FR2.2

3.2.2.11 - Functional Requirement 2.11

ID: FR2.11
TITLE: Take Damage
DESC: The system runs an Enemy Attack Calculation (FR2.12), and displays the damage amount done to the player.
DEP: FR2.1

3.2.2.12 - Functional Requirement 2.12

ID: FR2.12
TITLE: Enemy Attack Calculation
DESC: The system checks the enemy's base attack strength, adds any modifying integers in play (buffs/debuffs) and sets this sum as enemy total attack power. System then subtracts the player's current HP integer with enemy total attack power.
DEP: FR2.11

3.2.2.13 - Functional Requirement 2.13

ID: FR2.13
TITLE: Player HP Check
DESC: The system checks the current HP of the player after each party's attack. If the player's HP is less than or equal to zero, run Lose Battle (FR2.14)
DEP: FR2.4, FR2.11

3.2.2.14 - Functional Requirement 2.14

ID: FR2.14
TITLE: Lose Battle
DESC: The system declares to the player that they have died and their game is over, prompting the player if they want to load their last saved game or exit the application.
DEP: FR2.13

Puzzle (Caleb Sears)	**3.2.3.1 - Functional Requirement 3.1** **ID: FR3.1** TITLE: Initiate Puzzle DESC: The user must be able to initiate the puzzle in a given room through main dialog. DEP: N/a **3.2.3.2 - Functional Requirement 3.2** **ID: FR3.2** TITLE: Create Puzzle Window DESC: A new puzzle window must open when player initiates a puzzle. DEP: FR3.1 **3.2.3.3 - Functional Requirement 3.3** **ID: FR3.3** TITLE: Enter Puzzle Answer DESC: The puzzle window must allow the user to type and submit an answer. DEP: FR3.2 **3.2.3.4 - Functional Requirement 3.4** **ID: FR3.4** TITLE: Open Journal DESC: The puzzle window must provide a command to directly open the journal from inventory. DEP: FR3.2 **3.2.3.5 - Functional Requirement 3.5** **ID: FR3.5** TITLE: Show Puzzle Status DESC: The puzzle window must show the status of the puzzle (solved or not solved), as well as the current number of incorrect attempts since the last monster attack. DEP: FR3.2, FR3.8. **3.2.3.6 - Functional Requirement 3.6** **ID: FR3.6** TITLE: Exit Puzzle Window DESC: The puzzle window must provide an option to exit the puzzle without attempting to answer. DEP: FR3.2 **3.2.3.7 - Functional Requirement 3.7** **ID: FR3.7** TITLE: Show Puzzle Hint DESC: The puzzle window must provide a command to display the puzzle hint if desired by player. DEP: FR3.2 **3.2.3.8 - Functional Requirement 3.8** **ID: FR3.8** TITLE: Save Puzzle Attempts DESC: The system must save the puzzle attempts, which only include actual answer submissions. DEP: FR3.3

	3.2.3.9 - Functional Requirement 3.9 **ID: FR3.8** TITLE: Puzzle Monster Attack DESC: The system must initiate a monster attack after every third incorrect puzzle answer. DEP: FR3.8, FR2.4
Room (Estephanie Gonzalez)	**3.2.4.1 - Functional Requirement 4.1** **ID:FR4.1** Title: Check Locked Room DESC: Locked Room required a key item to enter the room DEP: N/A **3.2.4.2 - Functional Requirement 4.2** **ID:FR4.2** Title: Enter Room DESC: Move the user into the designated room. DEP: FR4.1 **3.2.4.3 - Functional Requirement 4.3** **ID:FR4.3** Title: Look Room DESC: Display a description and useful artifacts/objects in the room DEP: FR4.2 **3.2.4.4 - Functional Requirement 4.4** **ID:FR4.4** Title: Pick up artifacts/objects DESC: Allows the user to take artifacts/objects in the room DEP: FR4.2 **3.2.4.6 - Functional Requirement 4.6** **ID:FR4.6** Title: Trap Room DESC: Contains fake puzzles that can initiate monster DEP: FR4.2 **3.2.4.7 - Functional Requirement 4.7** **ID:FR4.7** Title: Exit Room DESC: Return the user to the previous room DEP: FR4.2

2.3 Non-Functional Requirements (All members)

ID:QR3.1
Title: Saved Game Data
DESC: 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.

ID:QR3.2
Title: Response Time
Scale: Response time for user input.
DESC: System should respond to user input within ½ second.

ID:QR3.3
Title: Usage of Puzzle Window
DESC: 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.

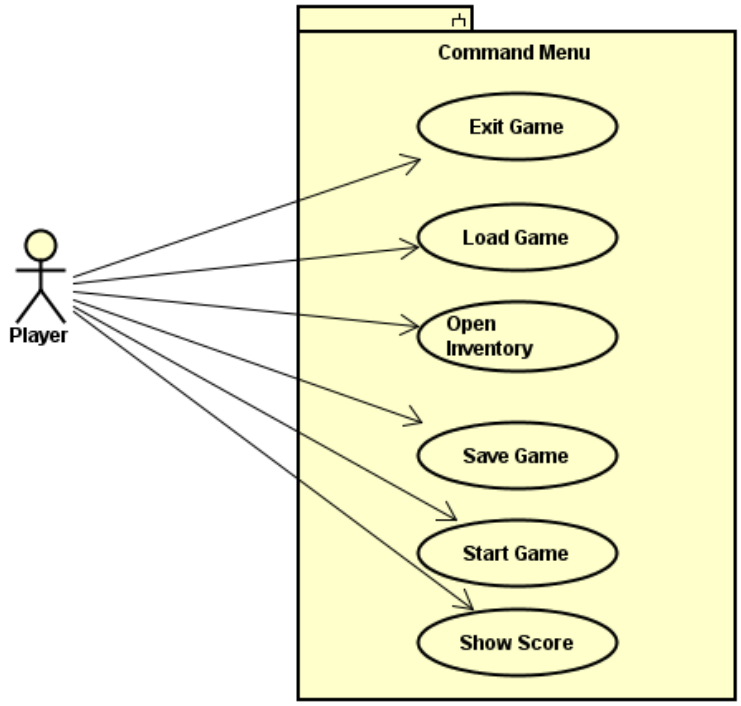
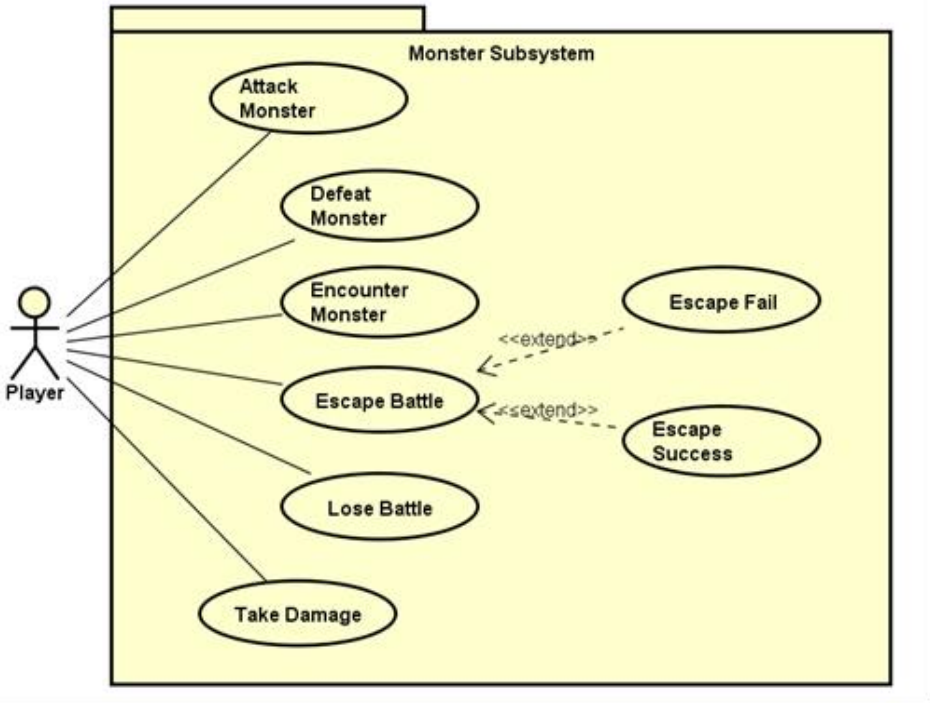
2.4 System Models

2.4.1. Use case model

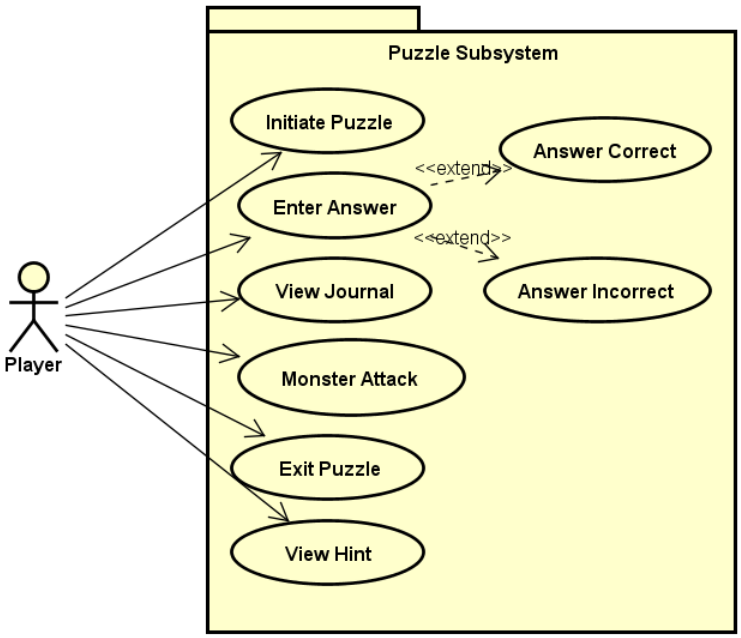
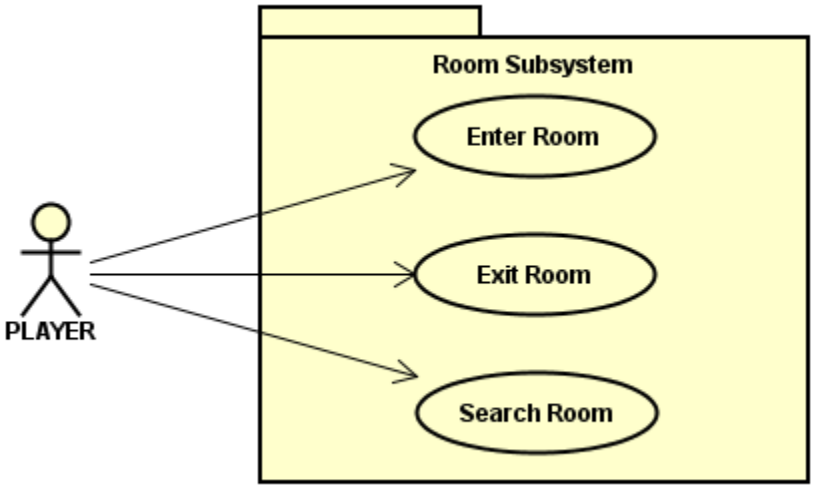
2.4.1.1. Use cases description

Features	**Use Cases**
Command Menu (Jory Alexander) 3.2.1	Save Game: *Player will enter the save game command to the options menu. A save game file will be created of the current game object* Load Game: *Player will enter the load game command into the start menu. The system will then check for a game object within the save game file and load it.* Open Inventory: *Player will enter the open inventory command. The inventory menu will then be displayed to the player.* Start Game: *Start Game will take in player input via the start menu. A game object will then be created.* Exit Game: *The player will enter the exit command into the options menu. The game object will be destroyed.* Show Score: *Player will enter the show score command to the game menu. The score will be displayed to the player.*
Monster (Aaron Knobloch) 3.2.2	Escape Battle (Success): *This allows the player to attempt to escape a battle, with the success determined by a percent chance. This use case presents the scenario wherein the user successfully escapes.* Escape Battle (Fail): *This allows the player to attempt to escape a battle, with the success determined by a percent chance. This use case presents the scenario wherein the user does not successfully escape.* Encounter Monster: *This allows the player to encounter monsters in room, by process of random generation from a list of possible monsters.* Defeat Monster: *This allows the player to defeat a monster, by attacking it to the point that it has no health points left.* Take Damage: *This allows the player's hero to take damage by a monster attack.* Lose Battle: *This allows the player to lose a battle by having a monster deal damage exceeding the health points of the hero.* Attack Monster: *This allows the player to choose to attack a monster, after initiating a battle.*
Puzzle (Caleb Sears) 3.2.3	Initiate Puzzle: *The Player will enter a command to the game menu to engage the puzzle and the system will respond by displaying the puzzle.* Enter Answer: *The Player will enter their answer to the puzzle in the puzzle window and the system will respond by calculating if the answer is correct or incorrect. If the Player is incorrect three times in a row, the system will have a monster attack the player. If the Player is correct, then the system will display the status and points to the payer.* View Journal: *The Player will enter the view journal command to the puzzle window and the system will respond by accessing and displaying the journal to the player.* View Hint: *The Player will enter a command to the puzzle window to view a puzzle's hint and the system will respond by displaying the hint to the Player.* Exit Puzzle: *The Player will enter a command to the puzzle window to disengage the puzzle and the system will respond by exiting the puzzle window.*
Room (Estephanie Gonzalez) 3.2.4	Enter Room: *The Player will enter a command to the game menu to enter a room and the system will respond by updating the player's location* Search/View/Describe Room: *The player will enter a command to the game menu to search, view or describe the room and the system will respond by displaying the corresponding description related to the command.* Exit Room: *The Player will enter a command to the game menu to exit a room and the system will respond by updating the player's location.*

2.4.1.2. Use case diagrams

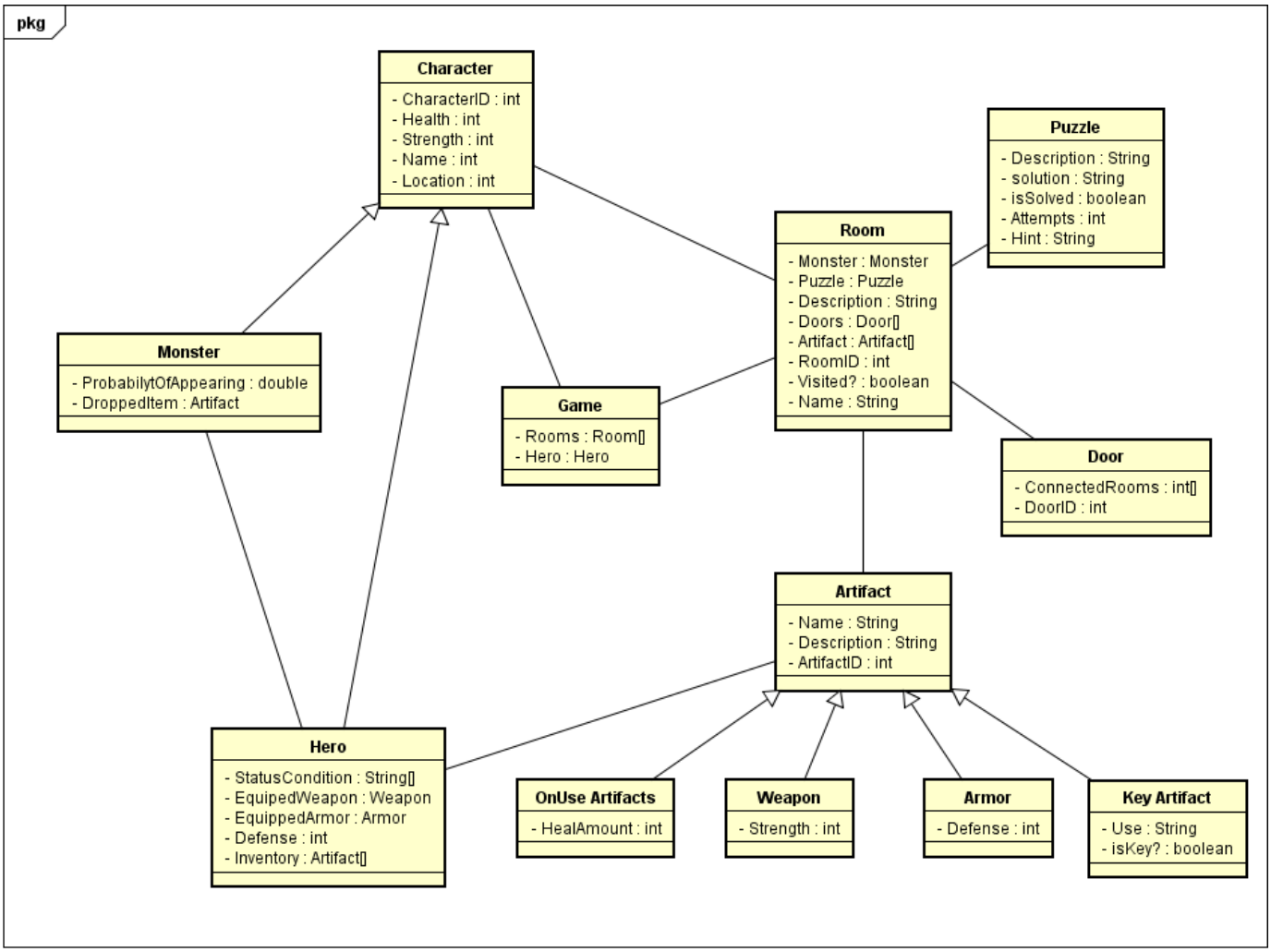
Command Menu (Jory Alexander)	
Monster (Aaron Knobloch)	

Team Avengers

Puzzle (Caleb Sears)	
Room (Estephanie Gonzalez)	

2.4.1. Object Model

2.4.1.1. Class Diagram (All members)

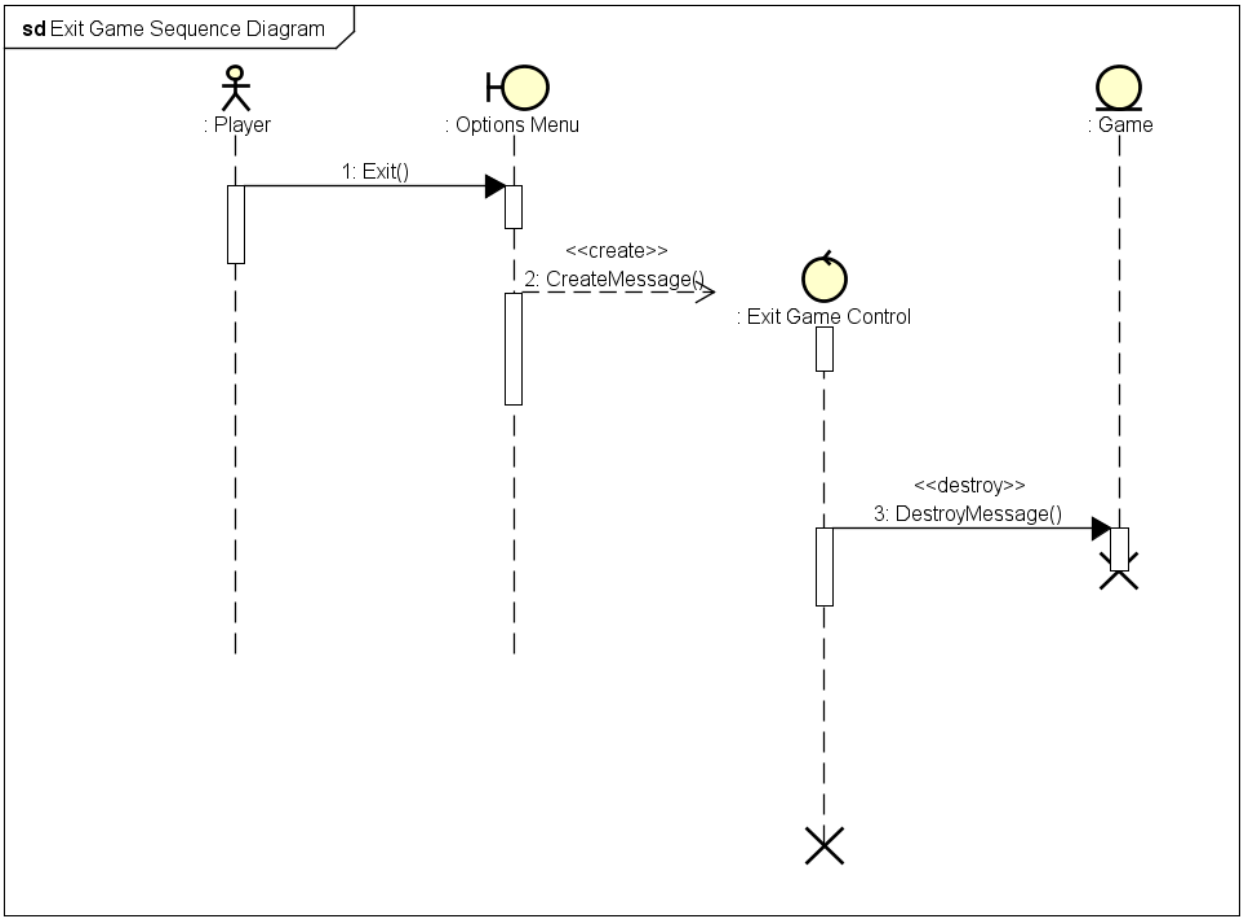


2.4.1.2. Class Diagram description (Jory Alexander)

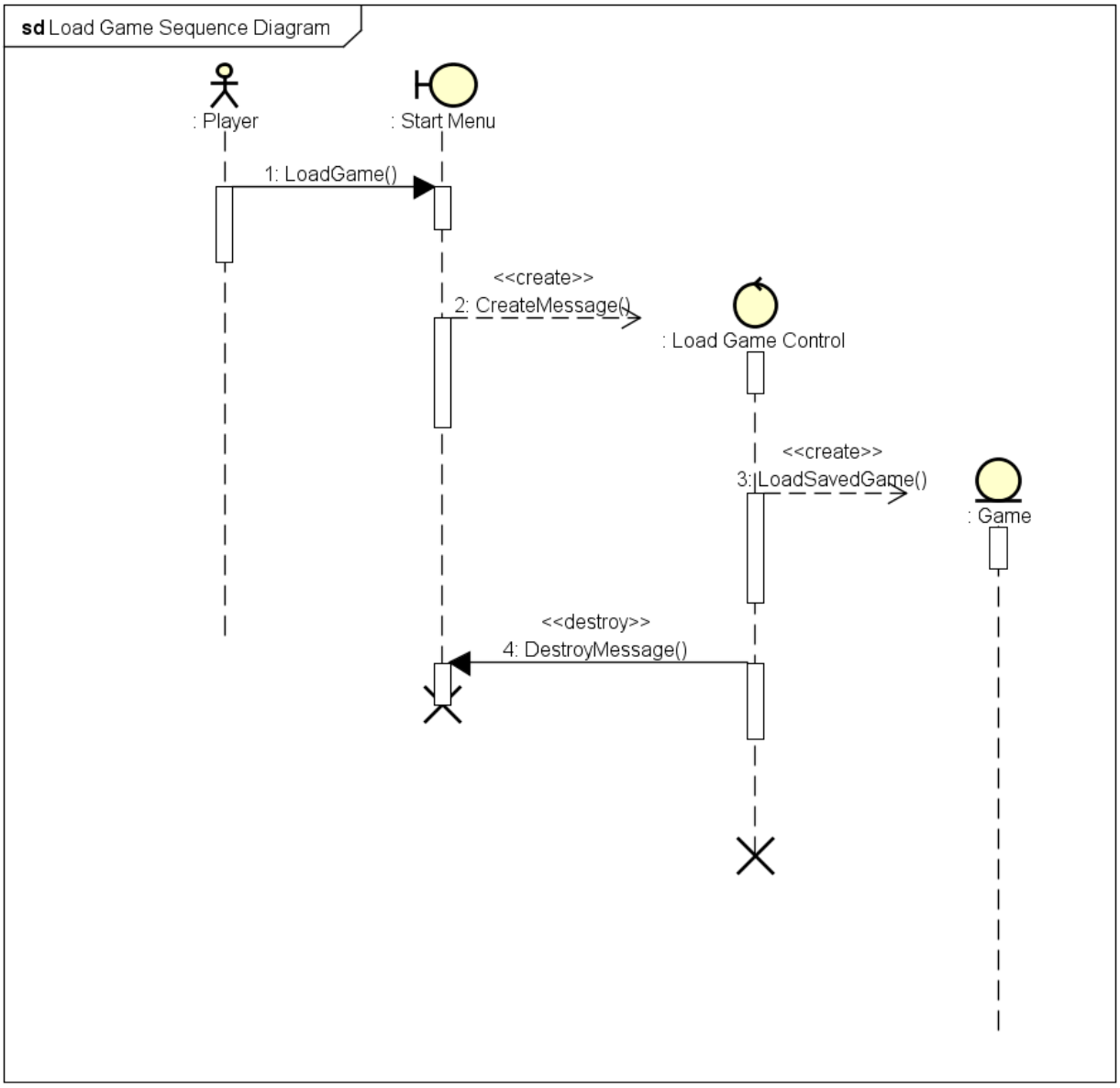
- **Character:** *The character class is a generalized class. It contains information related to the health strength and name of the character. It also contains a unique character ID.*
- **Monster:** *Monster is a subclass of Character. Monsters have a probability of appearing in a room as well as an artifact that they drop.*
- **Hero:** *Hero is a subclass of Character. The hero can have various status conditions, equip both weapons and armor, has a defense stat, as well as an array of Artifact object called inventory.*
- **Game:** *The game object will be responsible for holding all the rooms as well as the hero. The game object will be able to keep track of information regarding turns and duration of status conditions.*
- **Artifact:** *Artifact objects contain information regarding the name of the artifact. A Description of the artifact and a unique id. It is a generalized class.*
- **OnUse Artifact:** *OnUse Artifacts contain information regarding the amount of health that they will restore.*
- **Weapon:** *Weapon artifacts contain information regarding the amount of strength they will add to the hero.*
- **Armor:** *Armor contains information regarding the amount of defense they will add to the hero.*
- **Key Artifact:** *Key Artifacts contain a marking that they are a key artifact. And a description of their use.*
- **Door:** *Doors connect rooms to one another. Doors have a unique id and an array of the two rooms that they connect.*
- **Puzzle:** *Puzzles contain information describing the puzzle. They also contain a solution. Puzzles can be marked as solved or unsolved, and have a certain number of attempts. Puzzles can also contain a hint.*
- **Rooms:** *Rooms contain monster, puzzle, door, and artifact objects. They have a description of the room, a unique id, a name, and an indicator that they have been visited.*

2.4.2. Dynamic Model

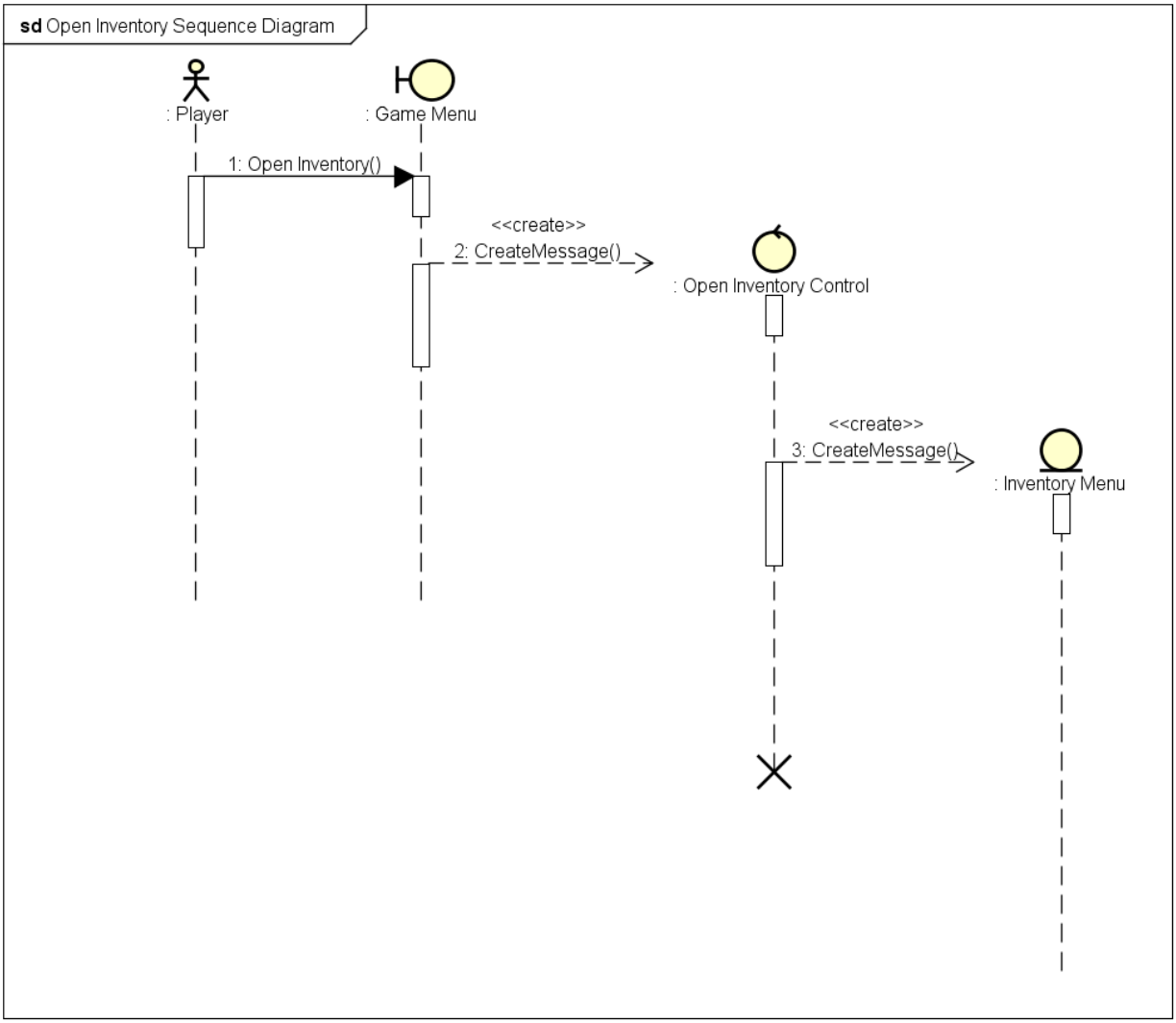
2.4.2.1. Sequence Diagrams

Command Menu (Jory Alexander)	

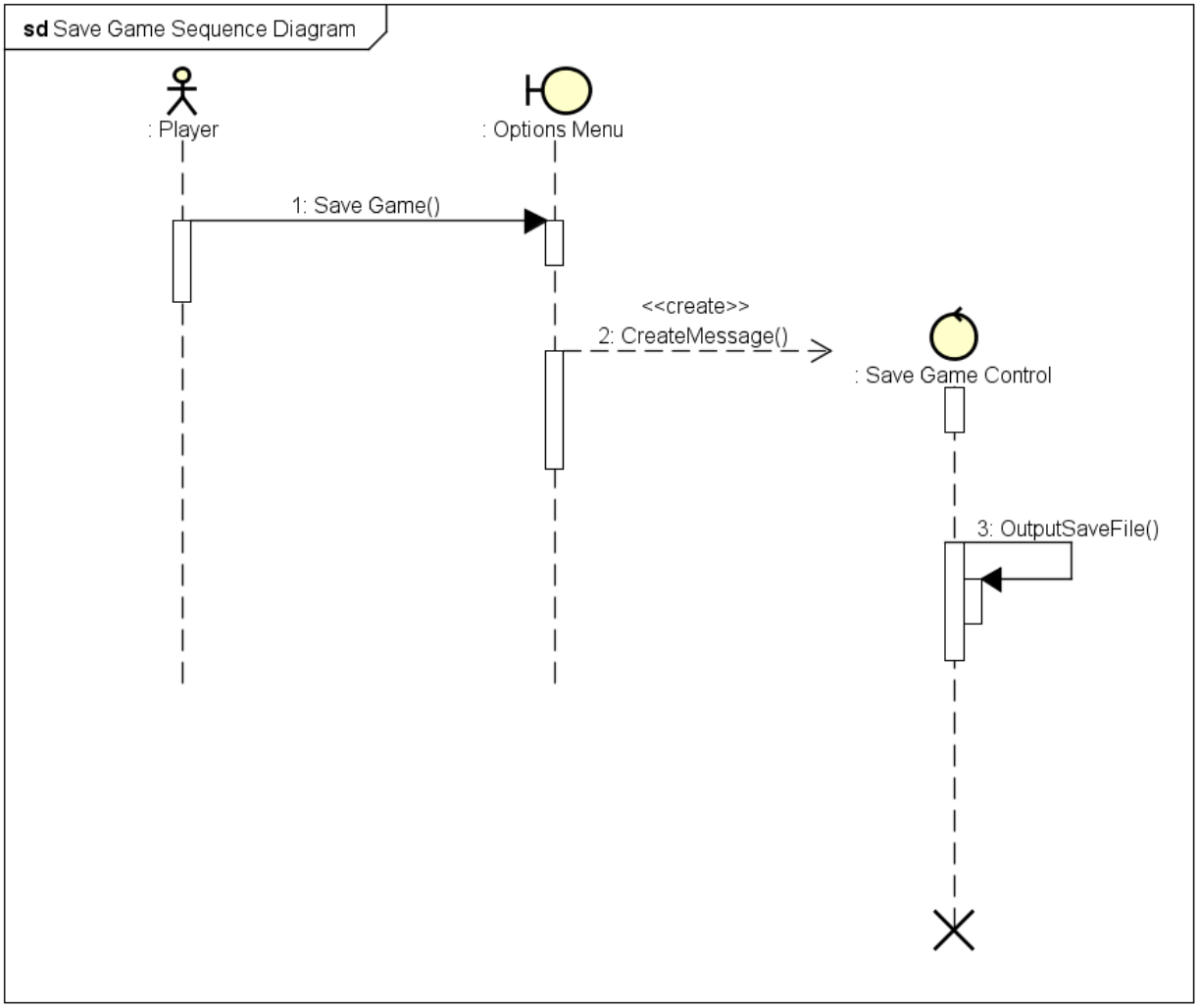
Team Avengers



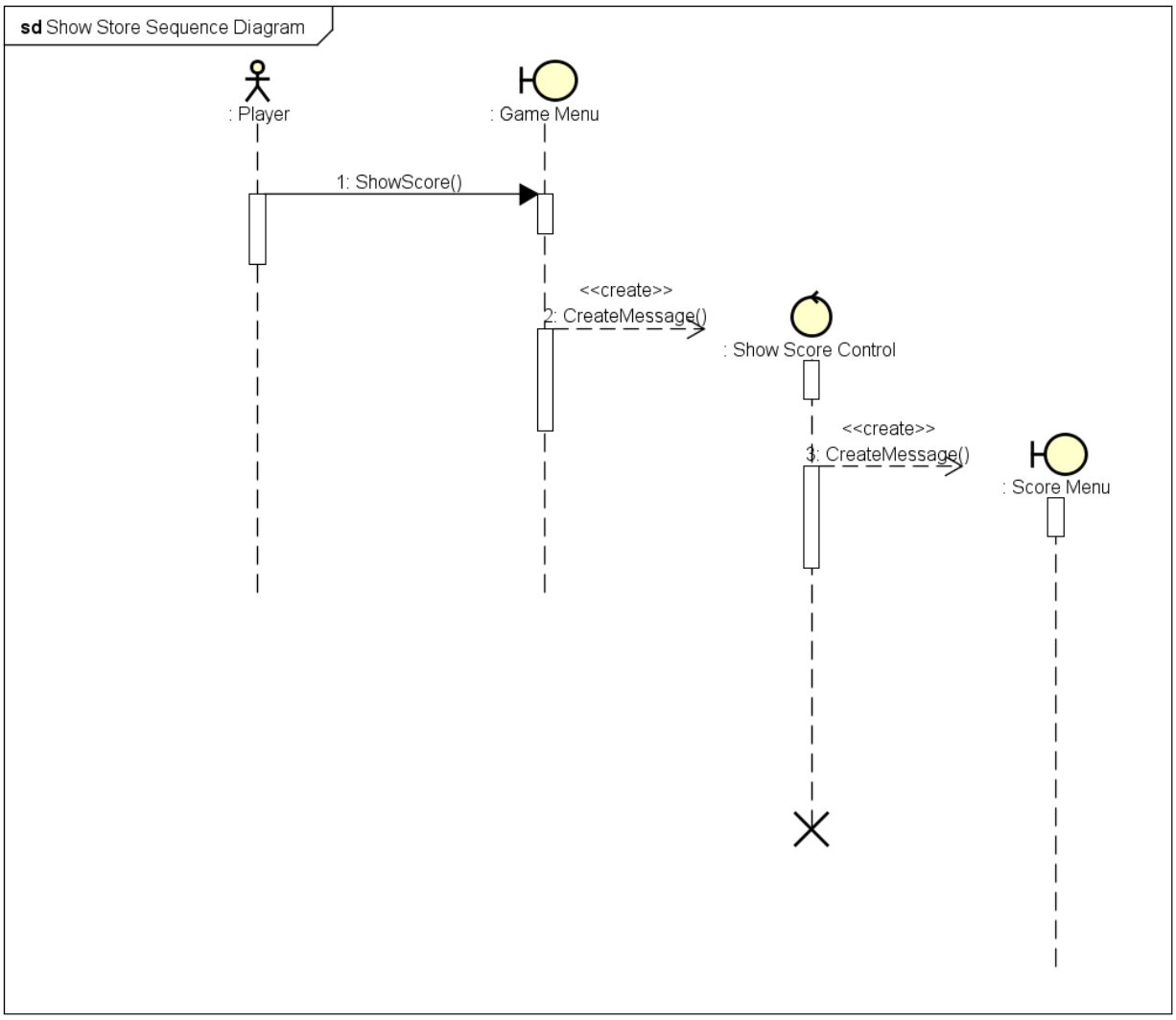
Team Avengers



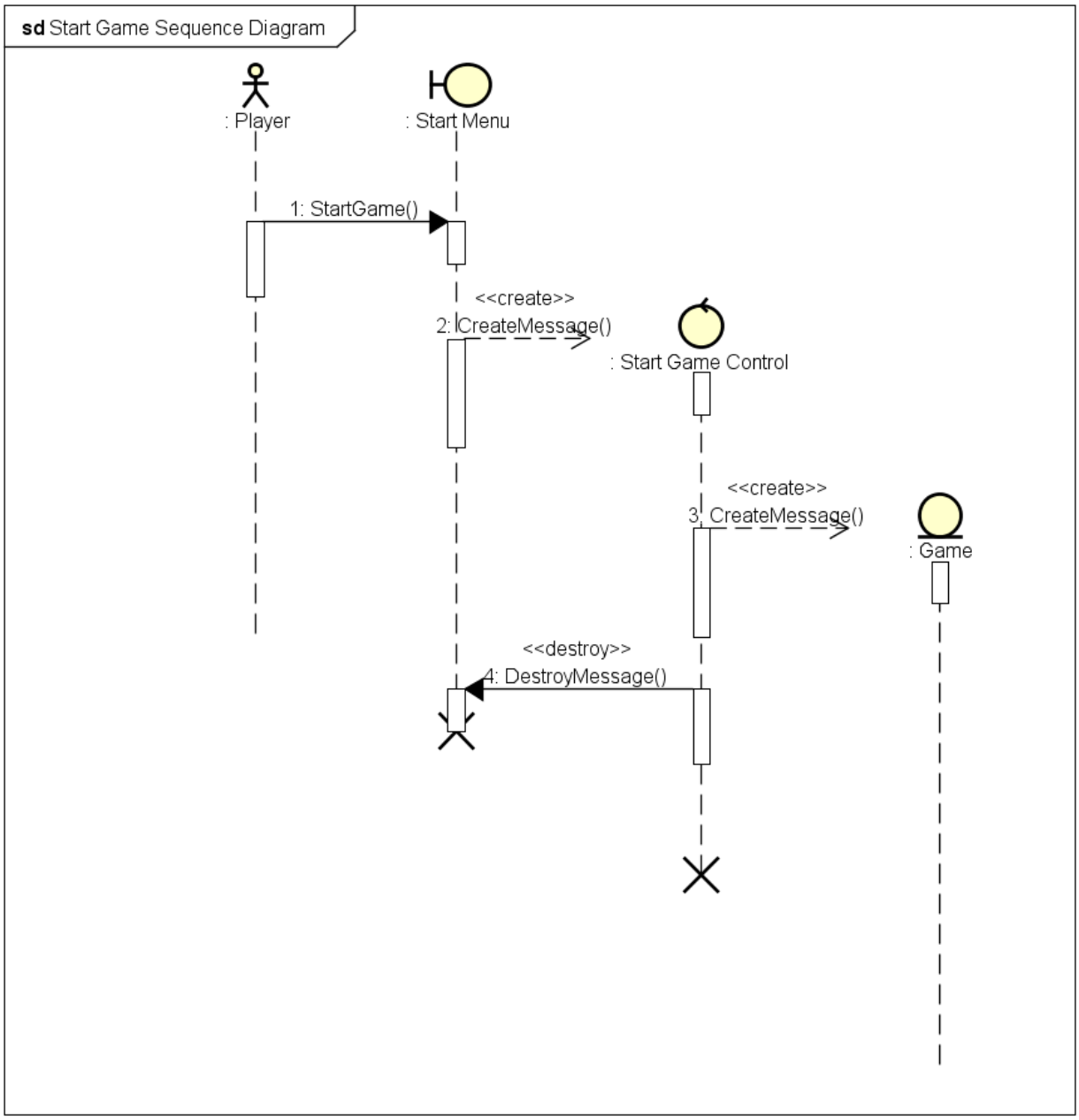
Team Avengers



Team Avengers



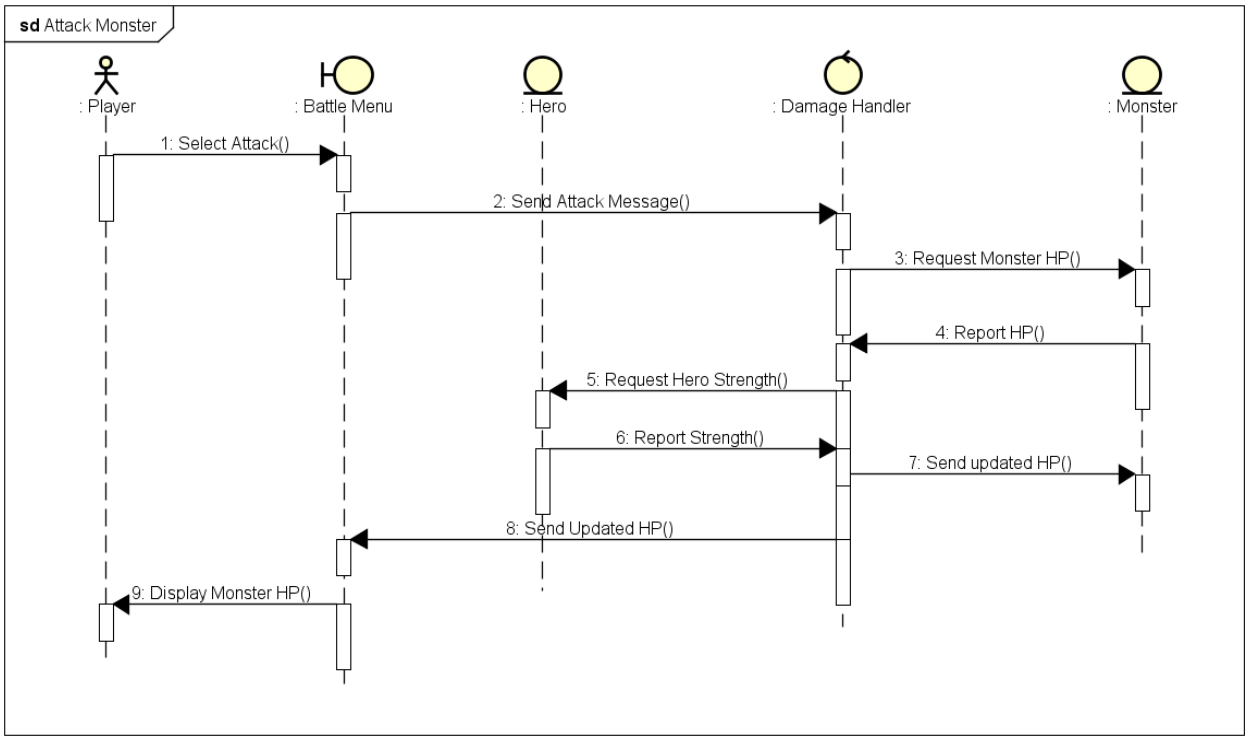
Team Avengers

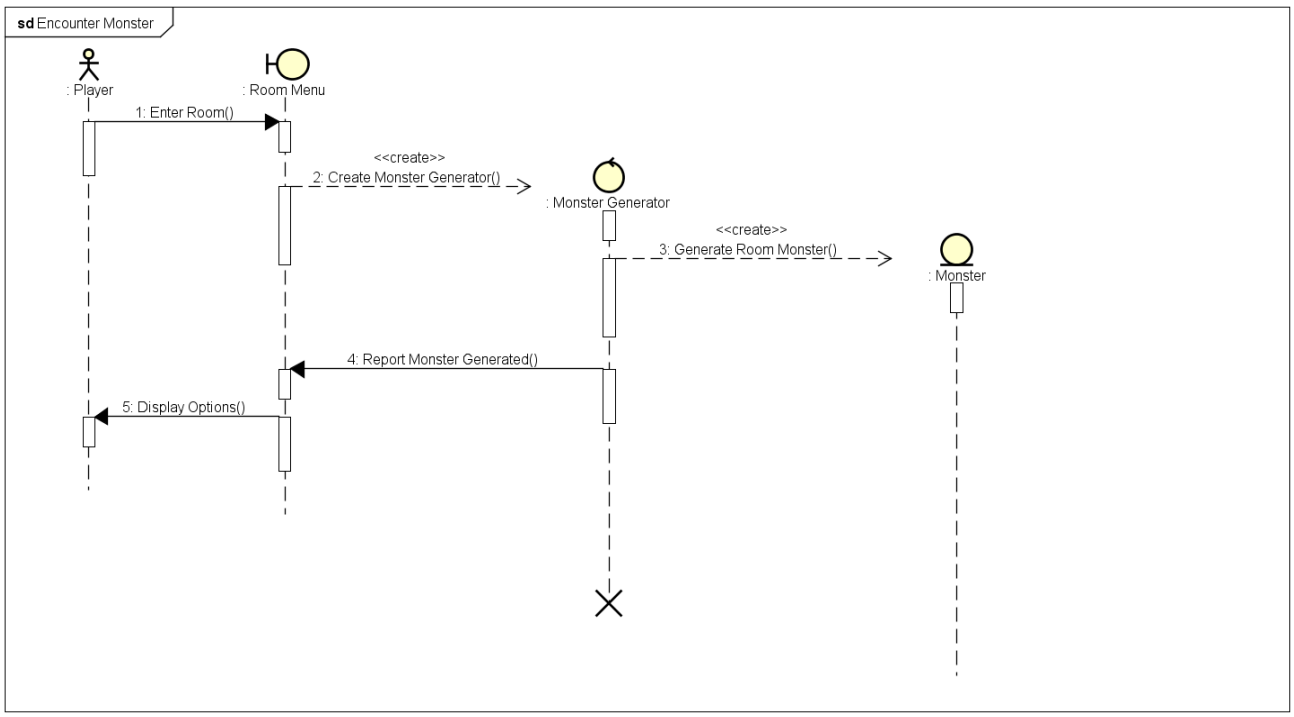


Team Avengers

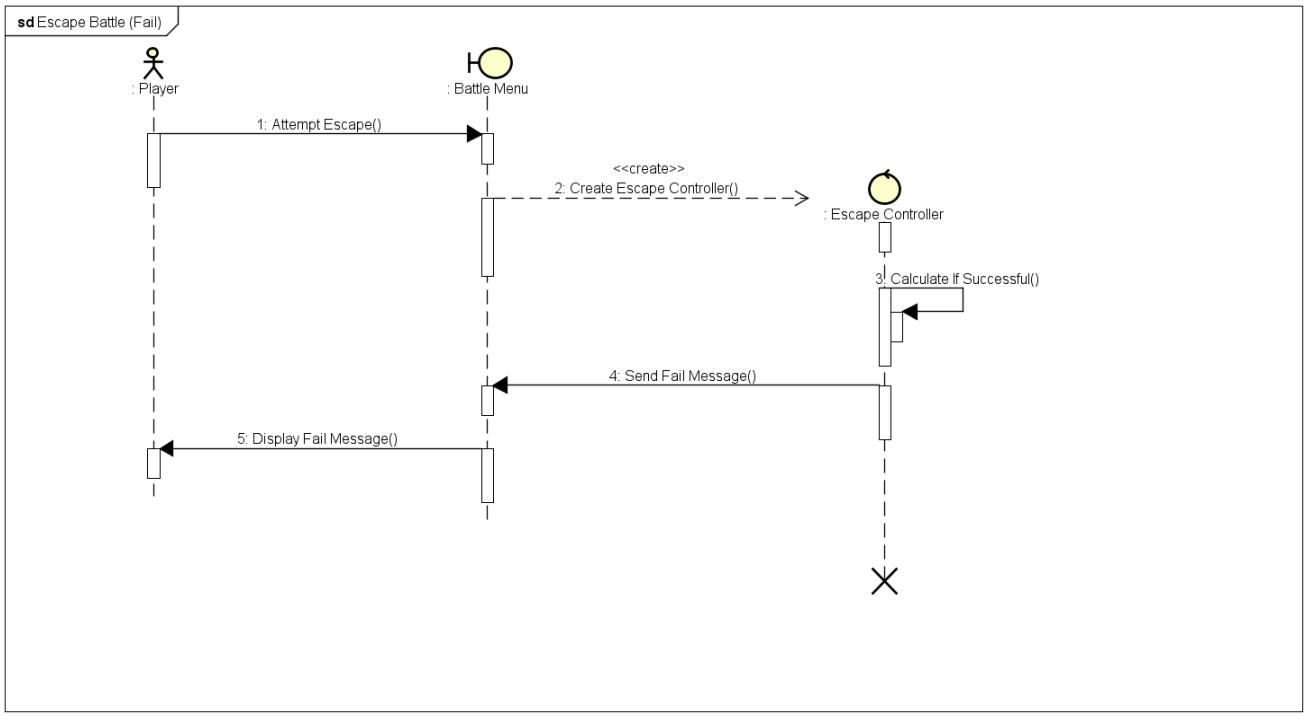
Monster

(Aaron Knobloch)

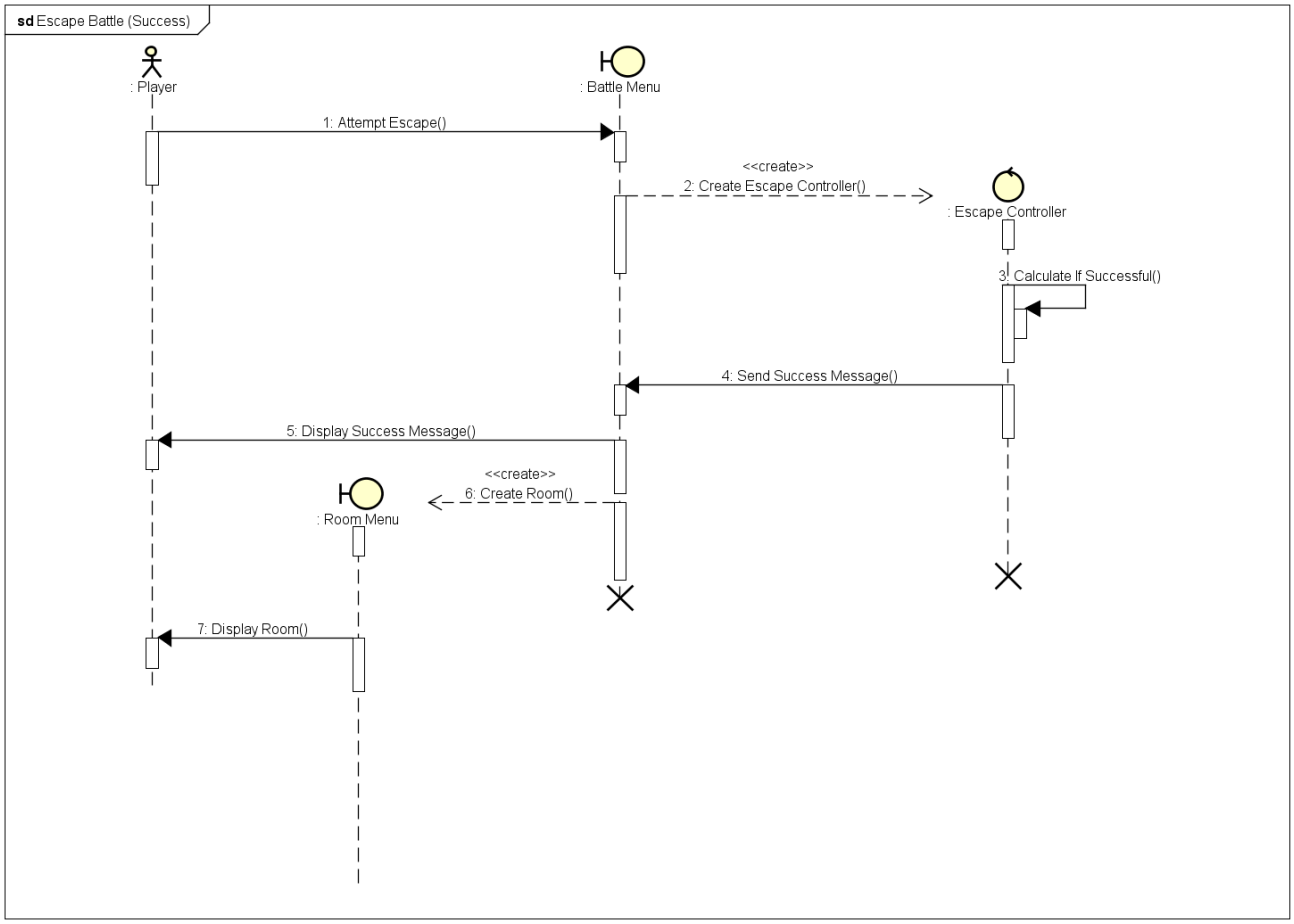




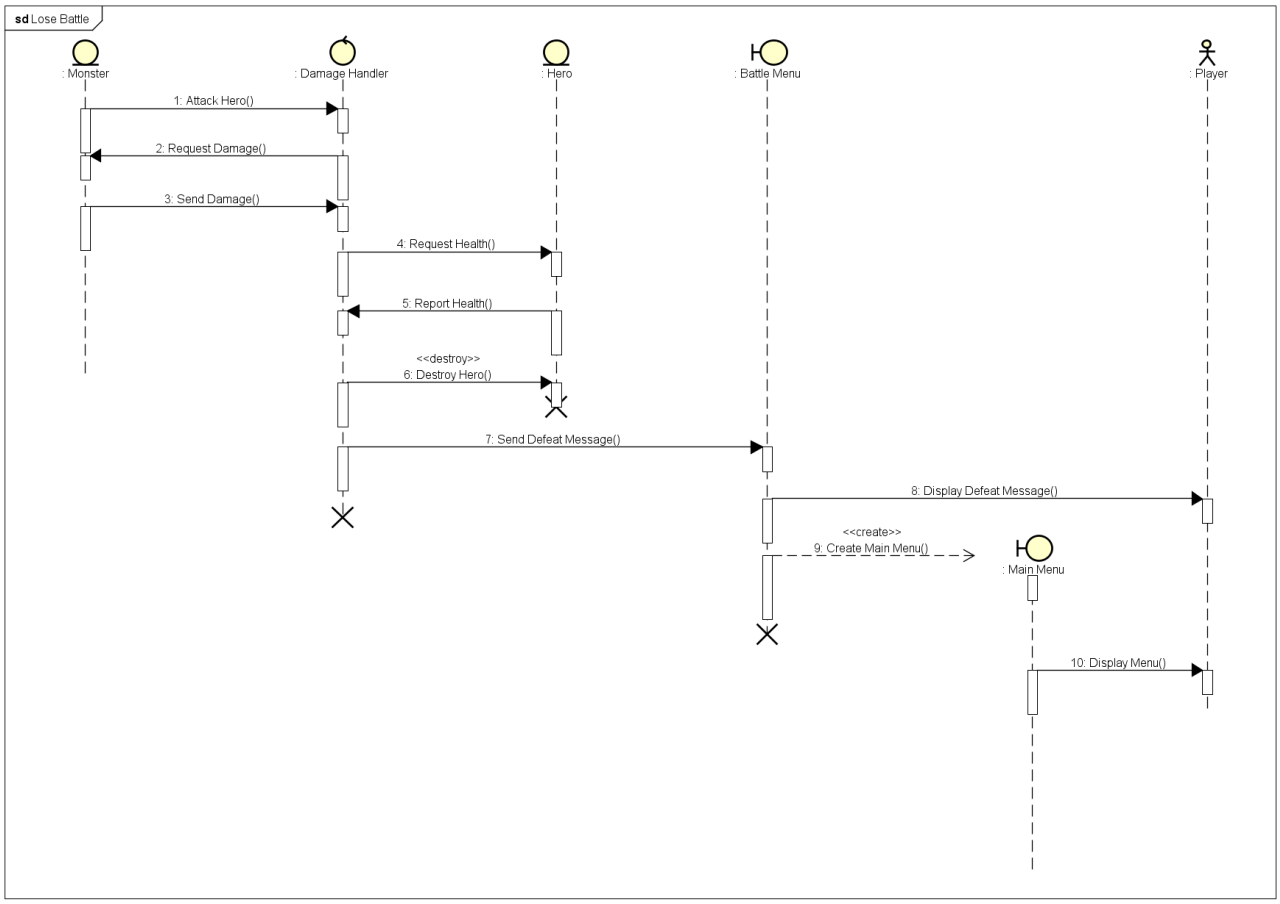
Team Avengers



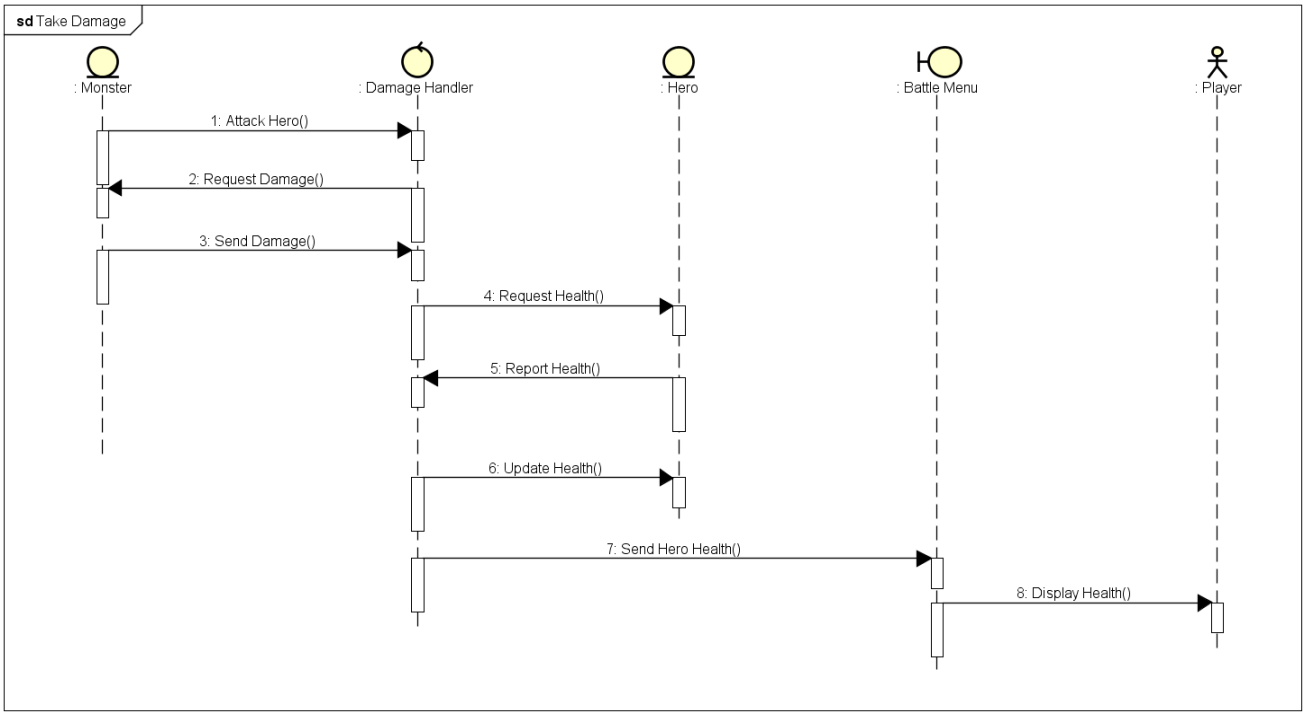
Team Avengers



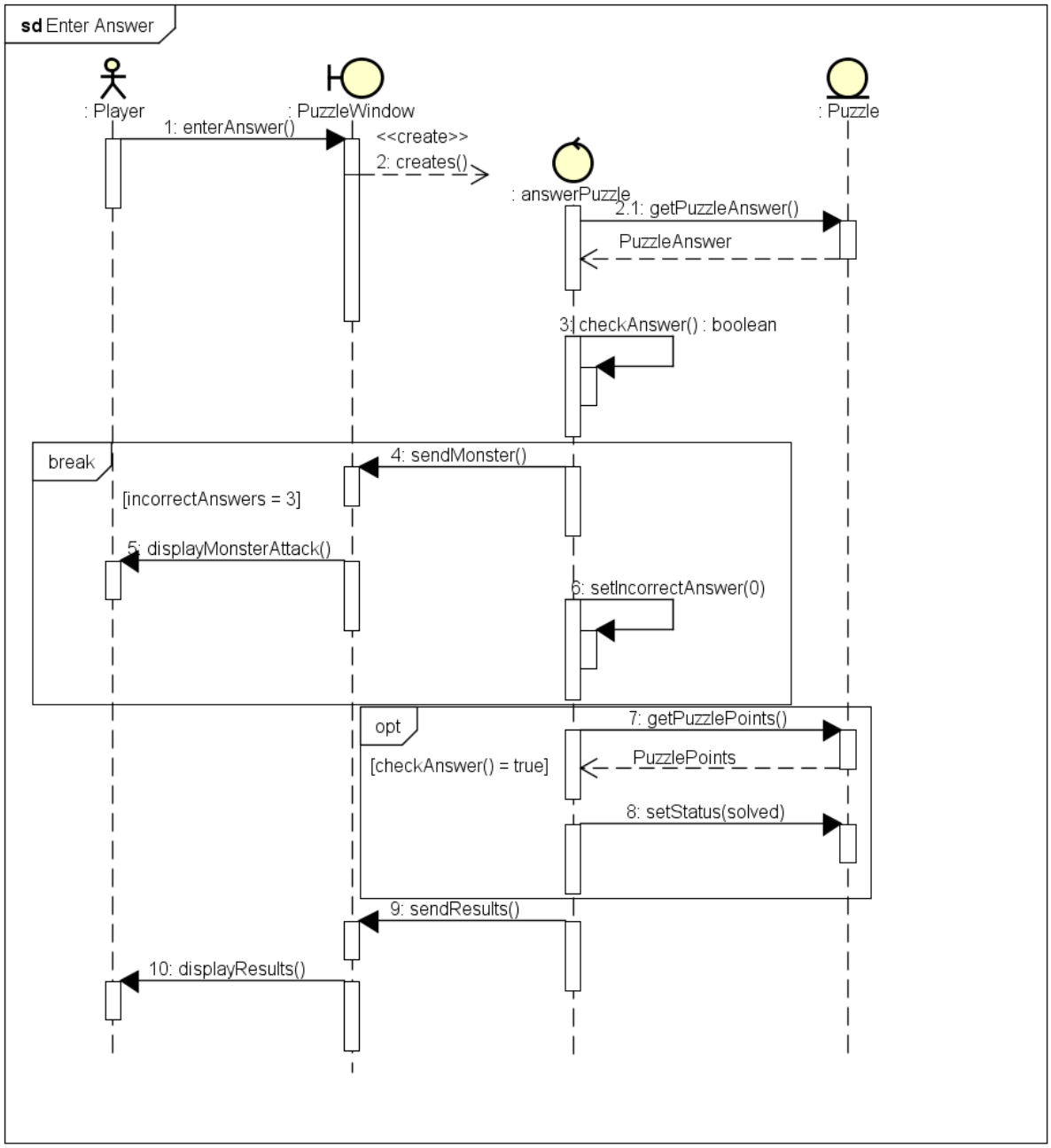
Team Avengers



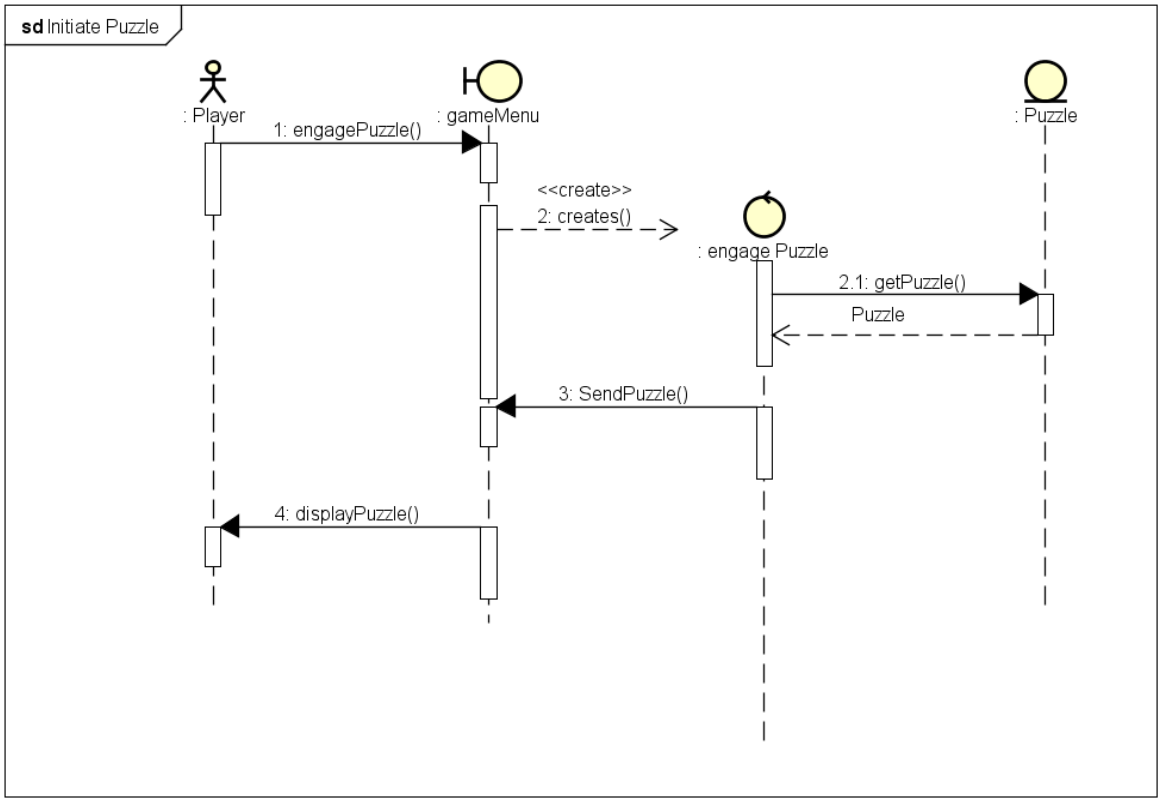
Team Avengers

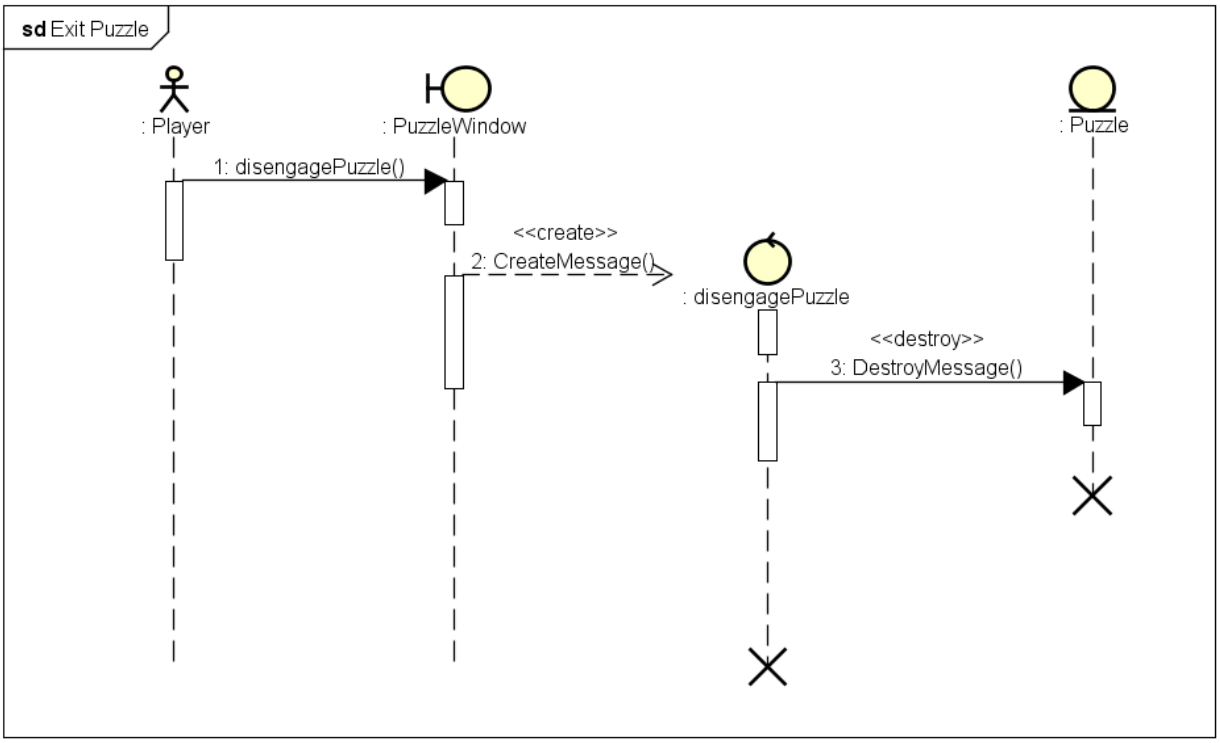


Team Avengers

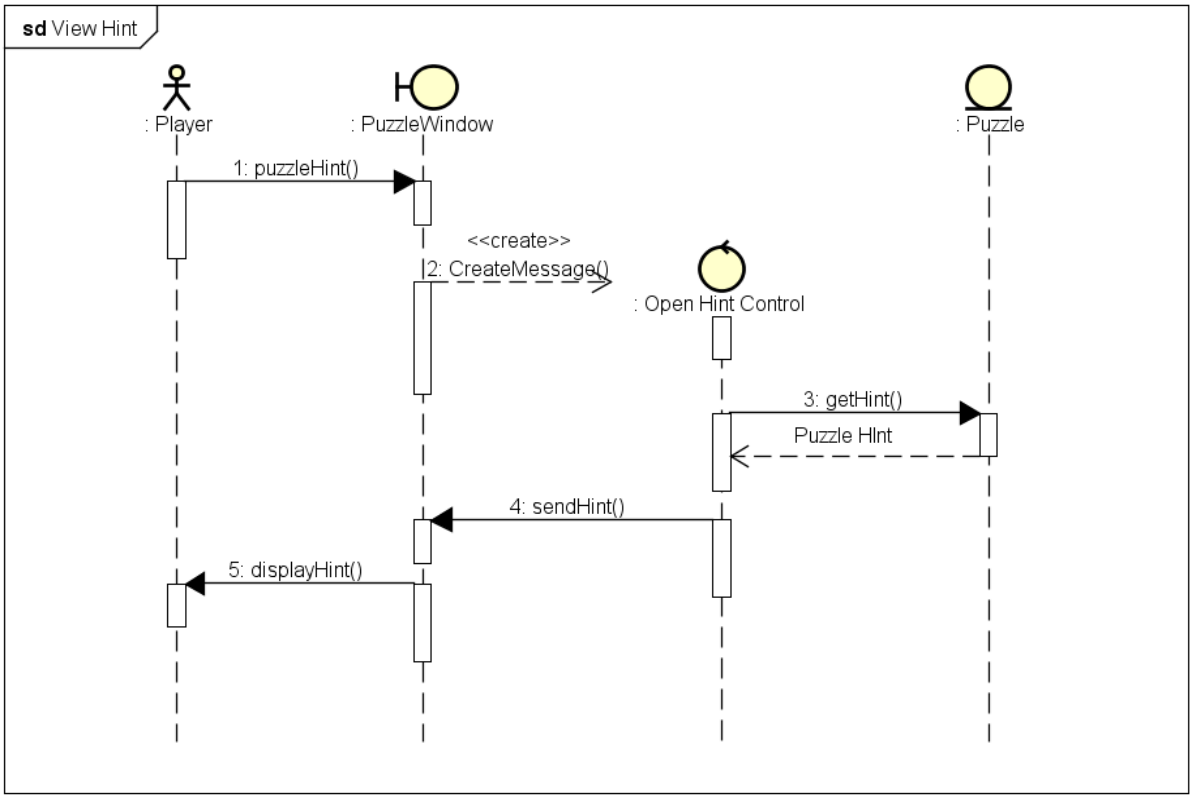
Puzzle (Caleb Sears)	

Team Avengers

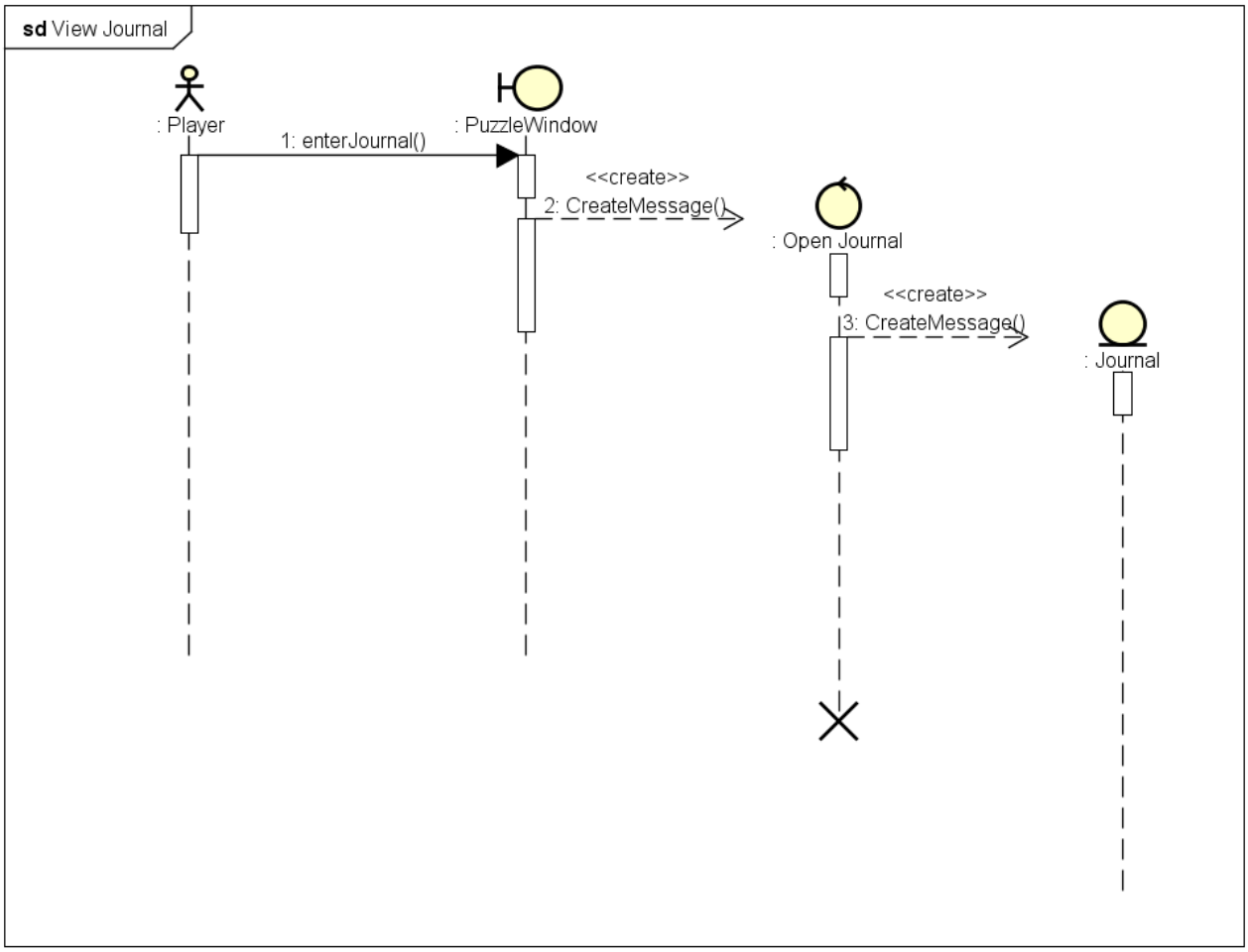




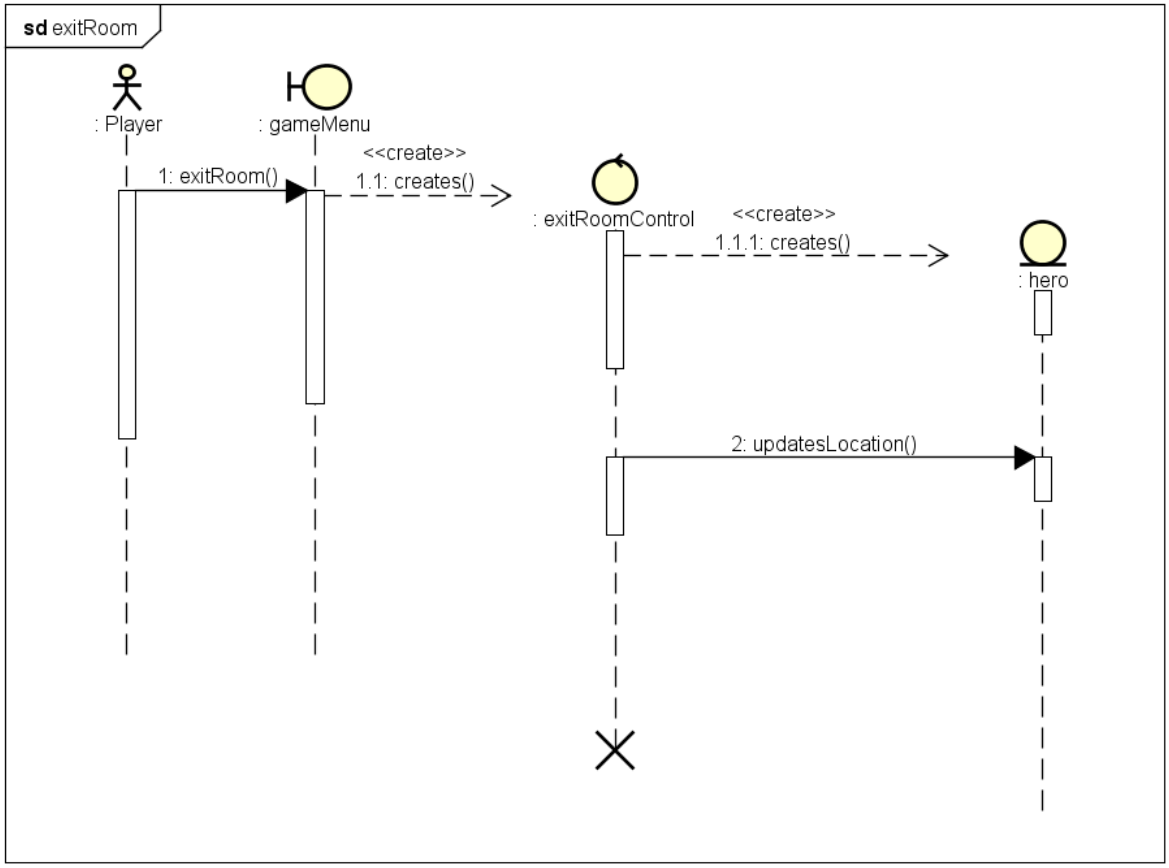
Team Avengers



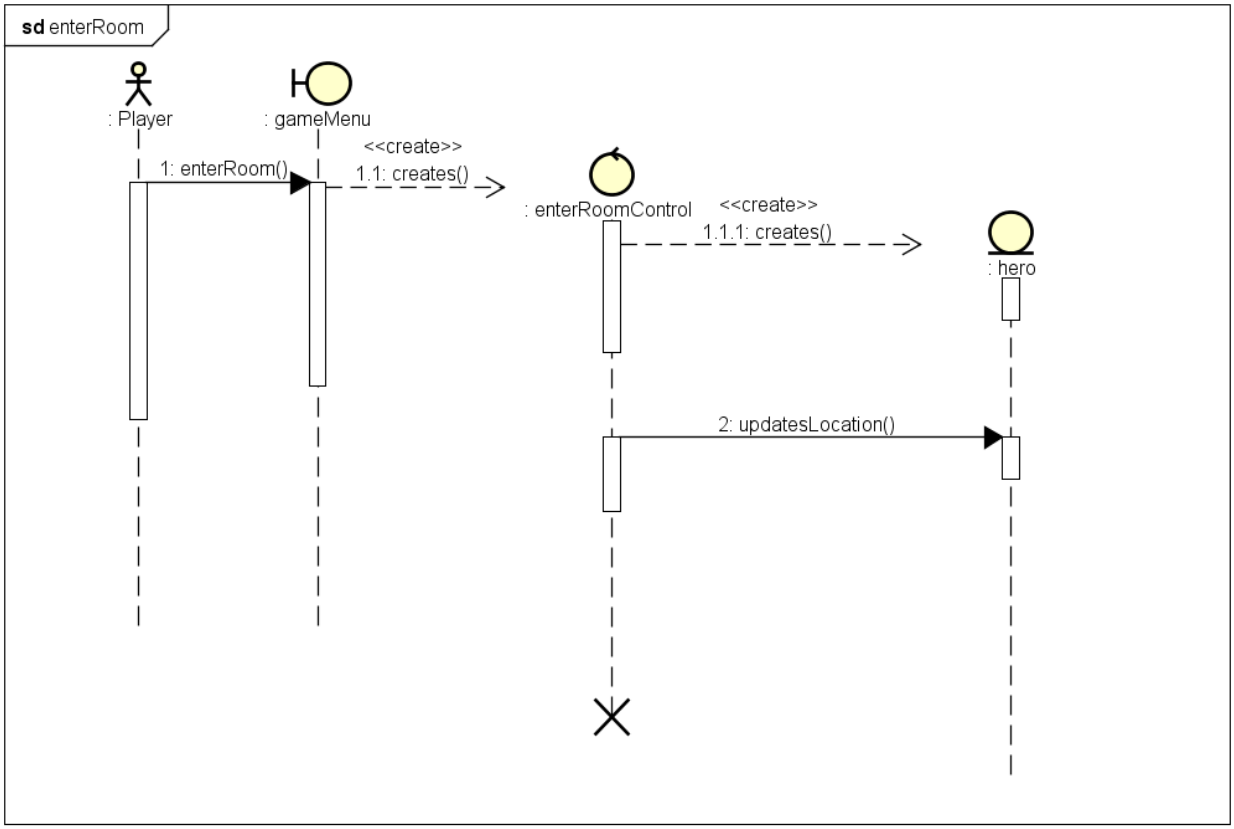
Team Avengers



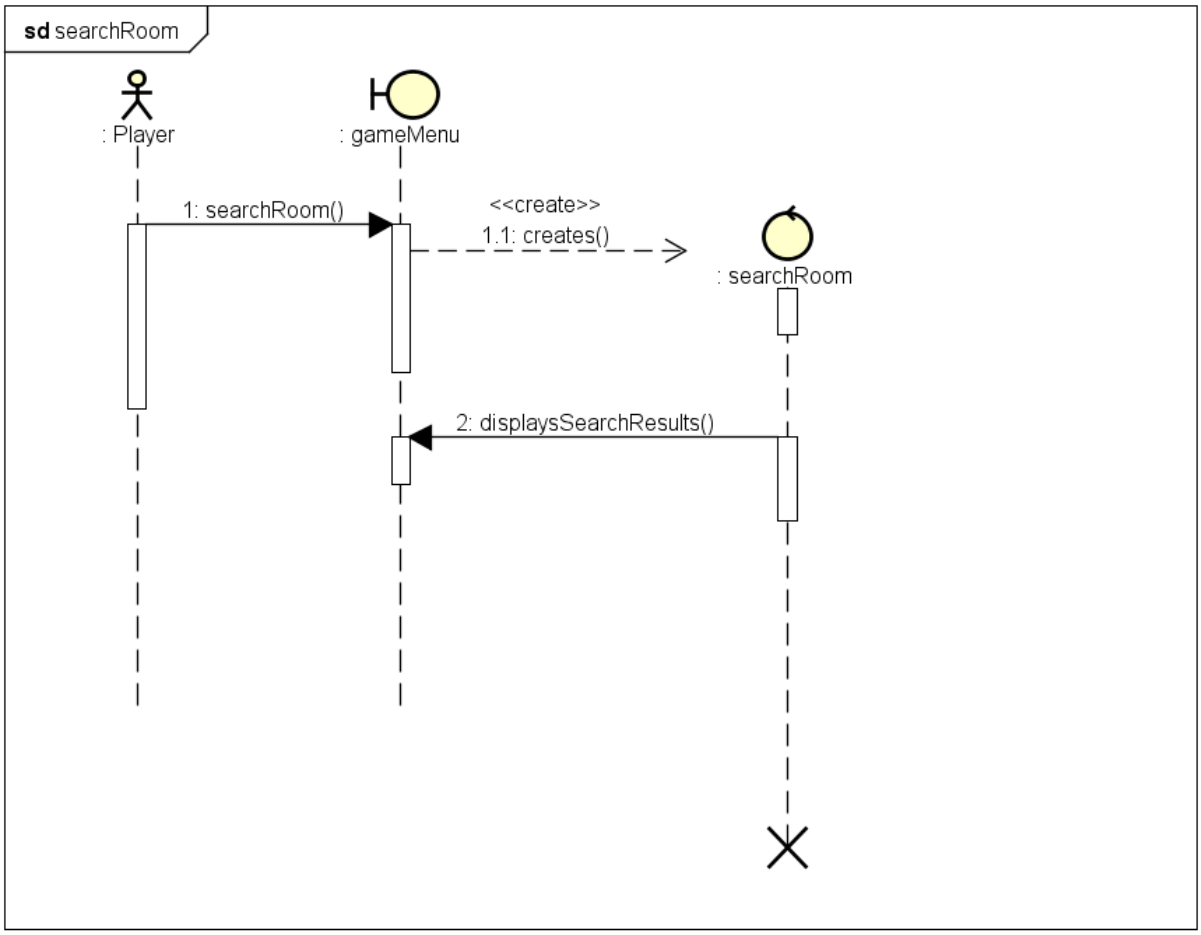
Team Avengers

Room (Estephanie Gonzalez)	

Team Avengers



Team Avengers



2.4.3. User interface (Estephanie Gonzalez)

2.4.3.1. Navigation

Game commands include:

1. *Exit*
2. *Load Game*
3. *Open Inventory*
4. *Save Game*
5. *Show Score*
6. *Start Game*
7. *Select Attack*
8. *Enter Room*
9. *Attempt Escape*
10. *Attack Hero*
11. *Send Damage*
12. *Enter Answer*
13. *Engage Puzzle*
14. *Disengage Puzzle*
15. *Puzzle Hint*
16. *Enter Journal*
17. *Enter Room*
18. *Exit Room*
19. *Search Room*