**Oh Ship**

Requirements Analysis Document

Rae Vroman, Gilad Berman, Zachary Cox

**Table of Contents**

1. Introduction
2. Proposed System
   1. Overview
   2. Functional Requirements
   3. Non-Functional Requirements
   4. System Models
      1. Use case model- **Rae**
         1. Use Case Diagrams
         2. Use Case Descriptions
      2. Object Model- **Zach**
         1. Class Diagram
         2. Class Diagram description
      3. Dynamic Model- **Rae, Zach, Gilad**
         1. Sequence Diagrams
            1. Command Menue Feature

Save Game

Exit Game

View Map

Load Game

Resume Game

Check Inventory

* + - * 1. Item Feature

Use Item

Pick Up Item

Un-Equip Item

Examine Item

Equip Item

Drop Item

* + - * 1. Combat Feature

Win Fight

Surrender Hint

Escape Fight

Surrender Fight

Lose Fight

Receive Damage

Attack Monster

* + - * 1. Navigation Feature

Go to Room

* + - * 1. Puzzle Feature

Examine Puzzle

Solve Puzzle Incorrectly

Ignore Puzzle

Request Hint

Solve Puzzle Correctly

* + 1. User interface- **Zach**
       1. Screen mockups
       2. Navigation

**Section 2: Proposed System**

2.1 Overview

2.2 Functional Requirements

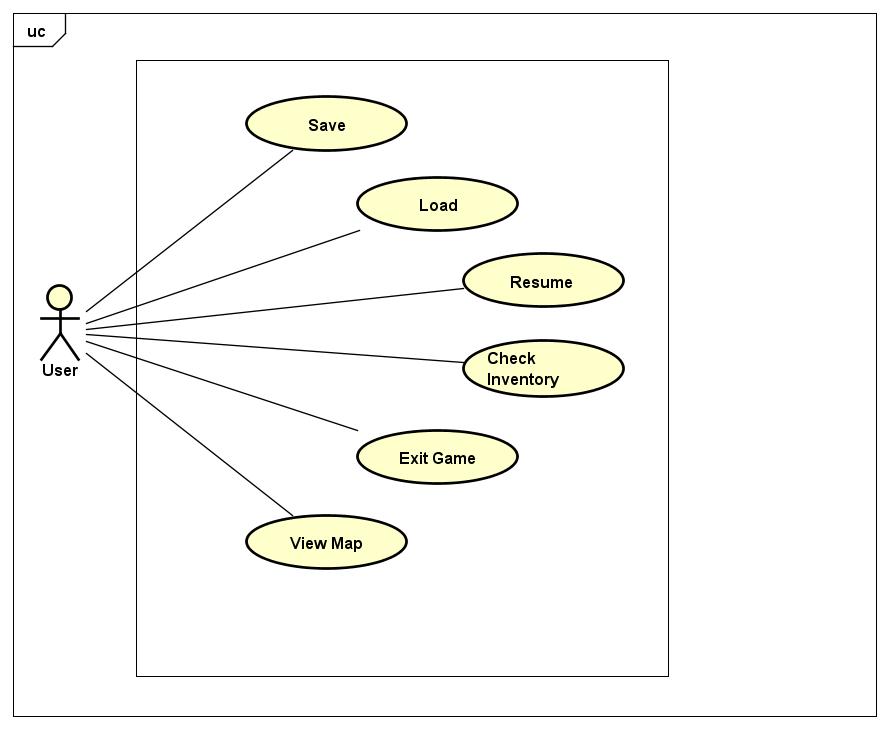
2.3 Non Functional Requirements

**2.4 System Models**

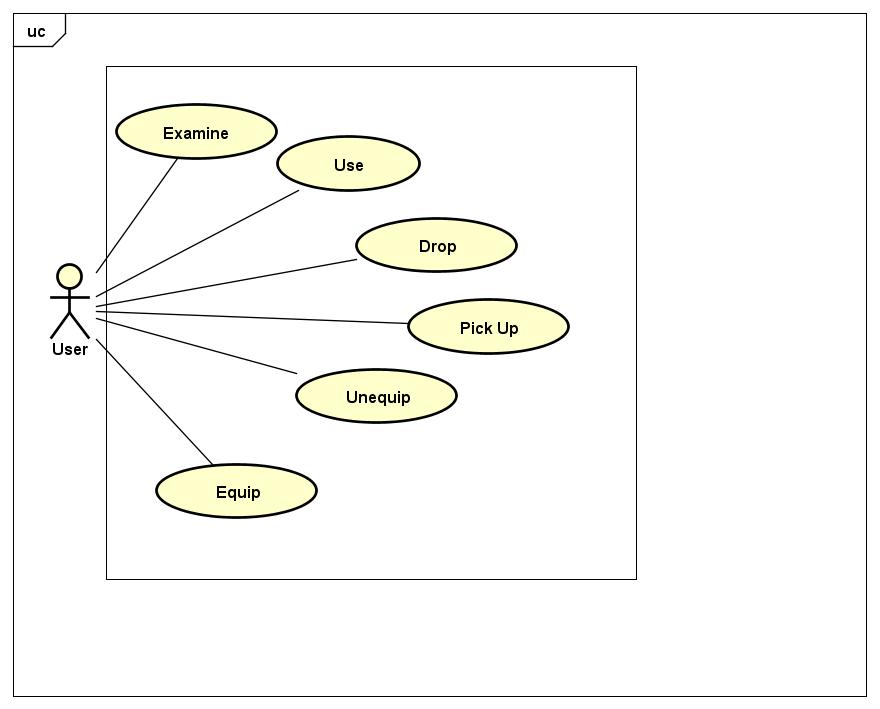
**2.4.1 Use Case Model**

**2.4.1.1 Use Case Diagrams**

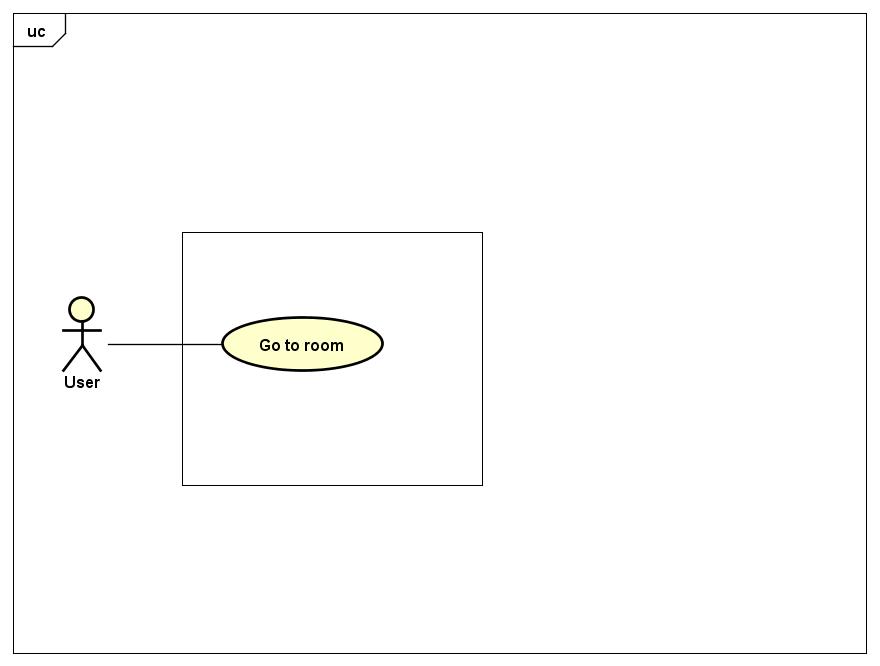
Command Menue Feature



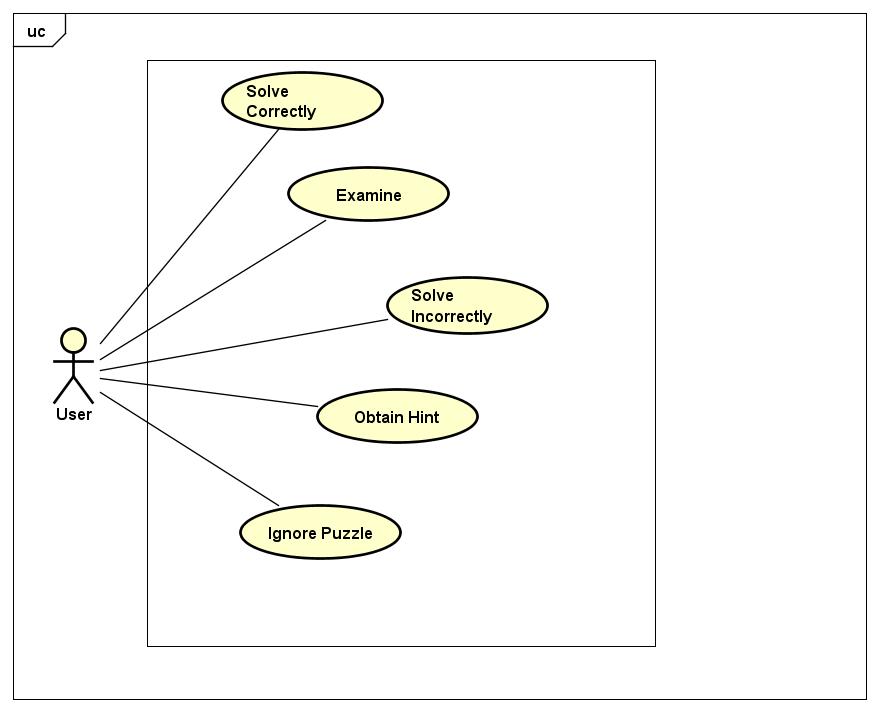
Item Feature



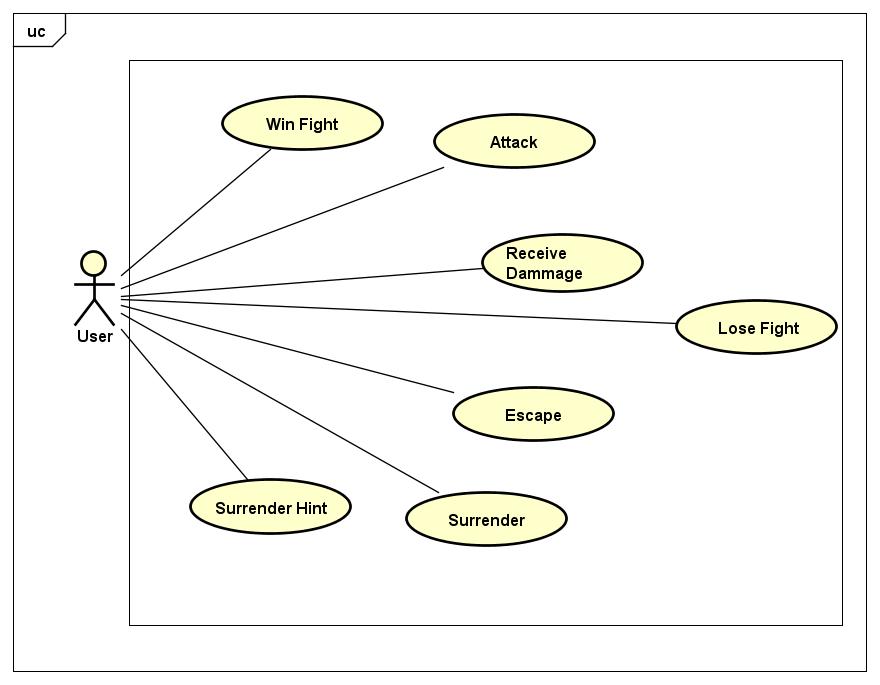
Navigation Feature



Puzzle Feature



Combat Feature



**2.4.1.2 Use Case Descriptions**

**Command Menue**

The user must be able to click a button to pull up the command menue. From there they can choose to save the game, load a game, resume their game, view the map, check their inventory or exit the game.

**Item**

There are many items throughout the game that the user can interact with. They can choose to pick up the item, examine the item, use the item, equip the item, unequip the item or drop the item,

**Navigation**

This game uses a GUI and will therefore have only one way to navigate. The user will click on a part of the map that they wish to navigate to, and if they have been there before, they can go to that room. If they have not been there before, the room they are attempting to travel to must be connected to the room they are currently in, and it must not need a puzzle or fight to get through.

**Puzzle**

For each puzzle, the player must choose to be able to examine the puzzle, ignore it, solve it, solve it incorrectly, or obtain a hint.

**Combat**

Whenever the player is engaged in combat there are many ways they can interact with the system. The player can attack a monster, win the fight, receive damage, lose a fight, escape the fight, get a hint for surrendering the fight, and surrender the fight.

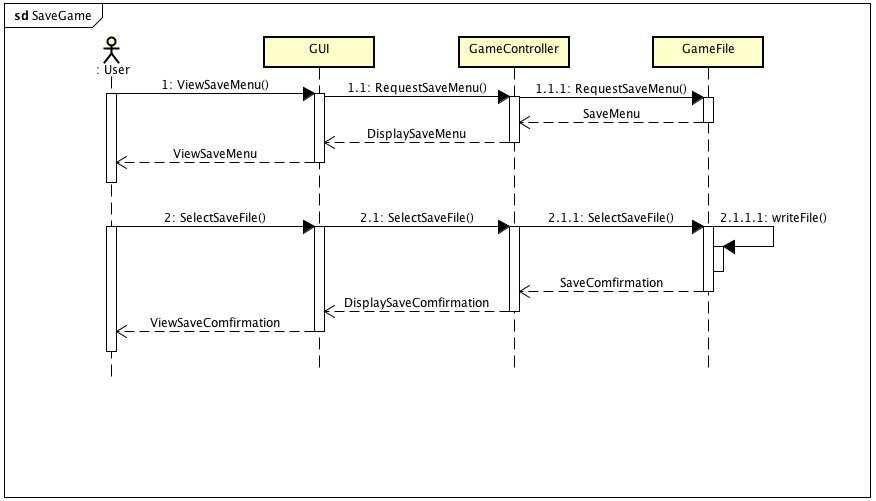
**2.4.2 Object Model**

**2.4.3 Dynamic Model**

**2.4.3.1 Sequence Diagrams**

**3.1.1 Command Menue Feature**

1.1.1 Save Game Diagram

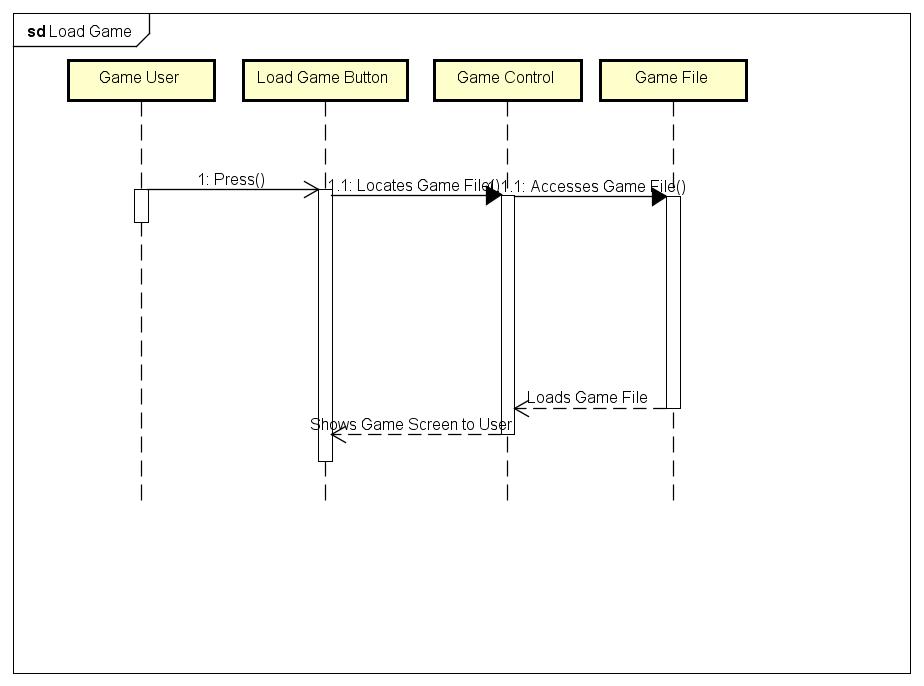


1.1.2 Exit Game Diagram 

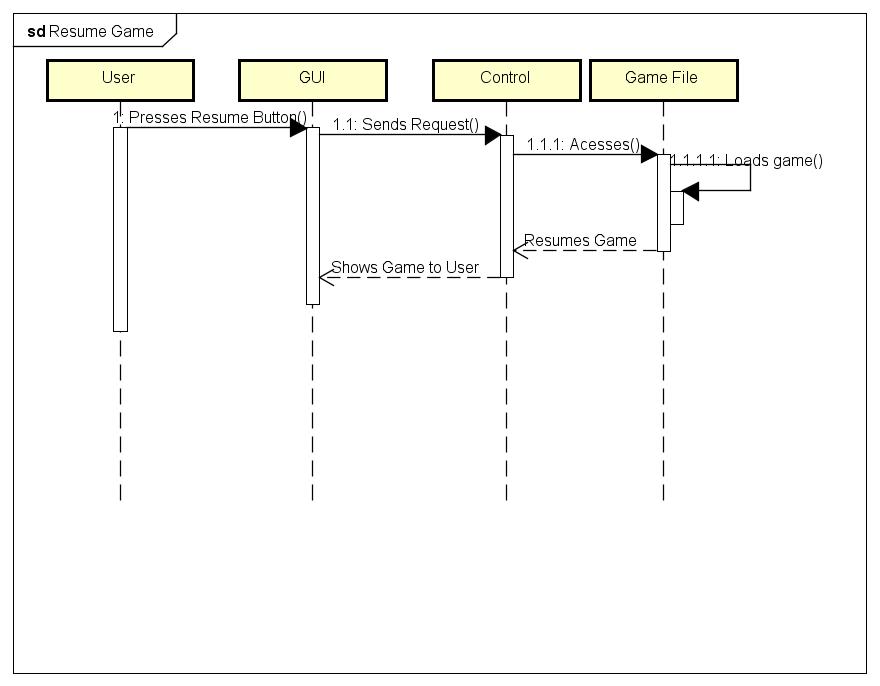
1.1.3 View Map Diagram



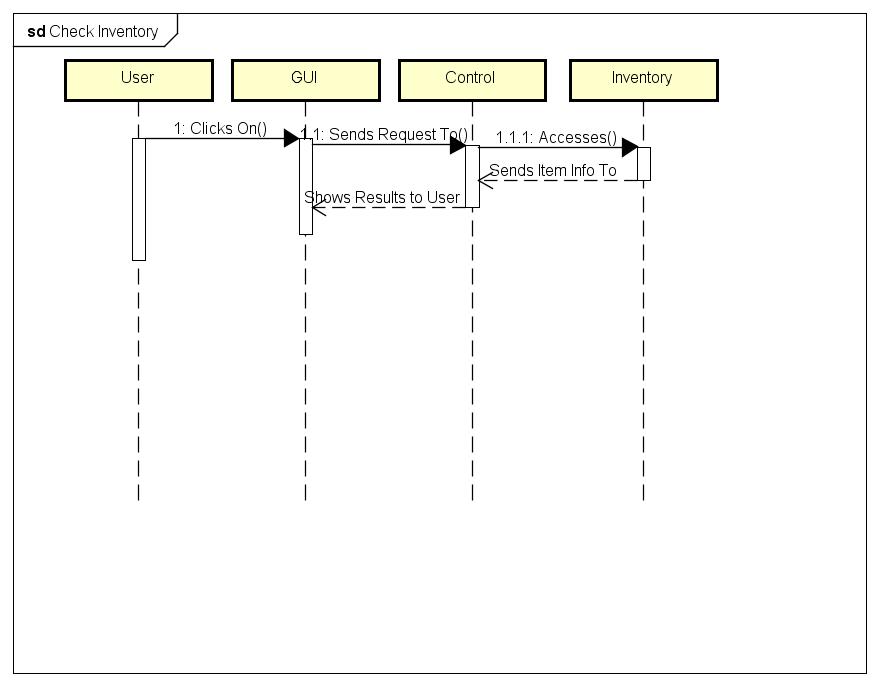
1.1.4 Load Game Diagram



1.1.5 Resume Game Diagram



1.1.6 Check Inventory Diagram



**3.1.2 Item Feature**

1.2.1 Use Item Diagram



1.2.2 Pick Up Item Diagram



1.2.3 Un-Equip Item Diagram



1.2.4 Examine Item Diagram



1.2.5 Equip Item Diagram

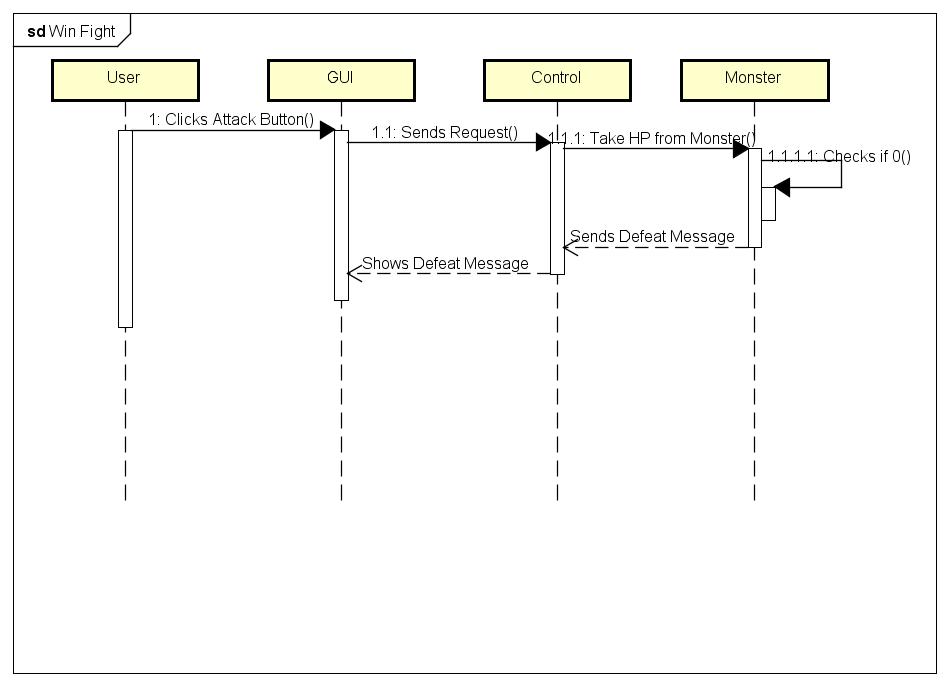


1.2.6 Drop Item Diagram



**3.1.3 Combat Feature**

1.3.1 Win Fight Diagram



1.3.2 Surrender Hint Diagram



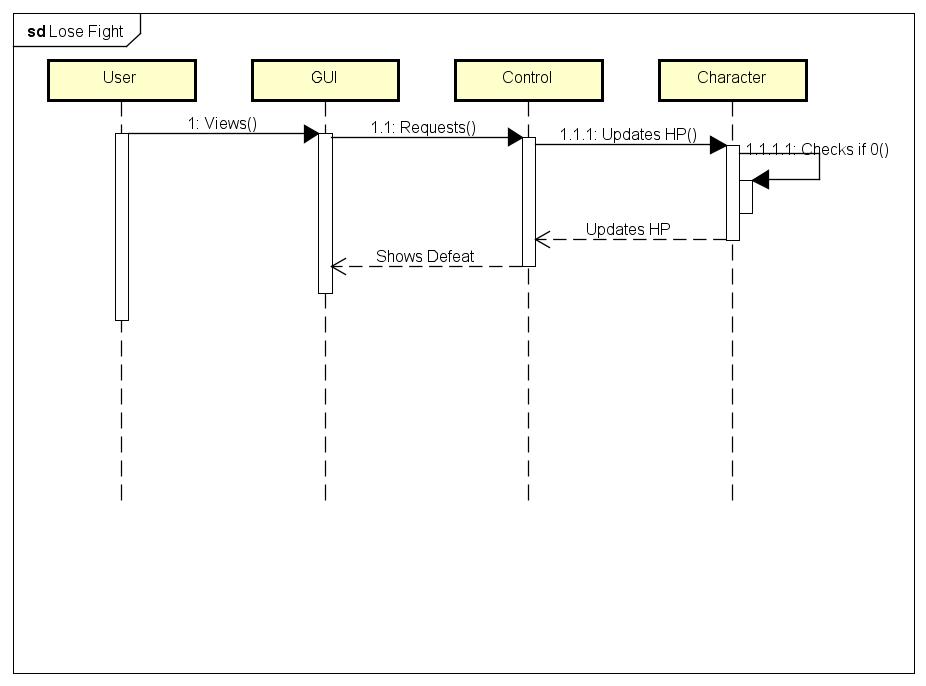
1.3.3 Escape Fight Diagram

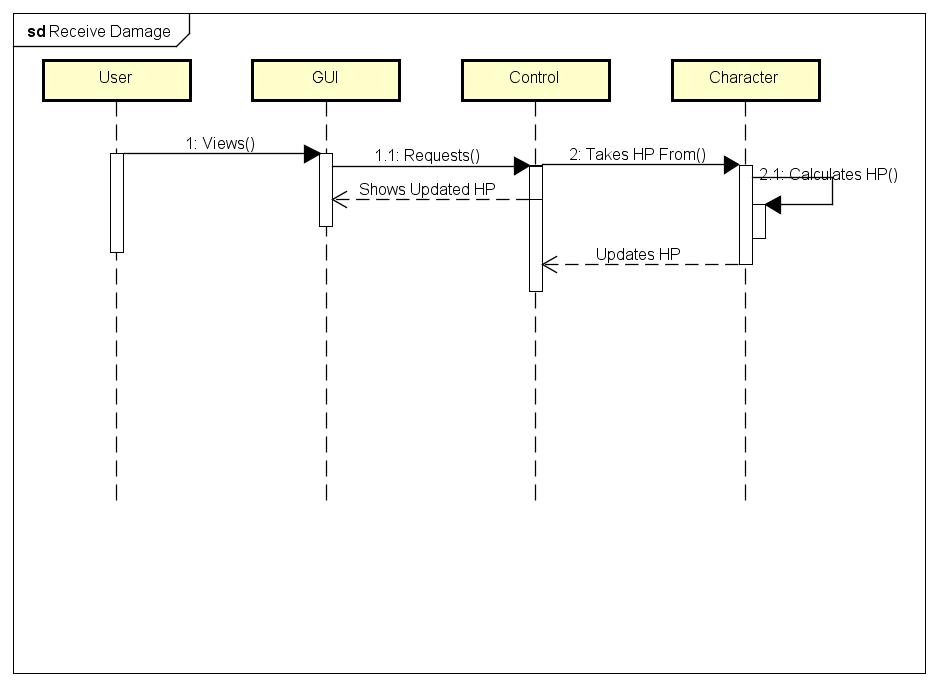


1.3.4 Surrender Fight Diagram

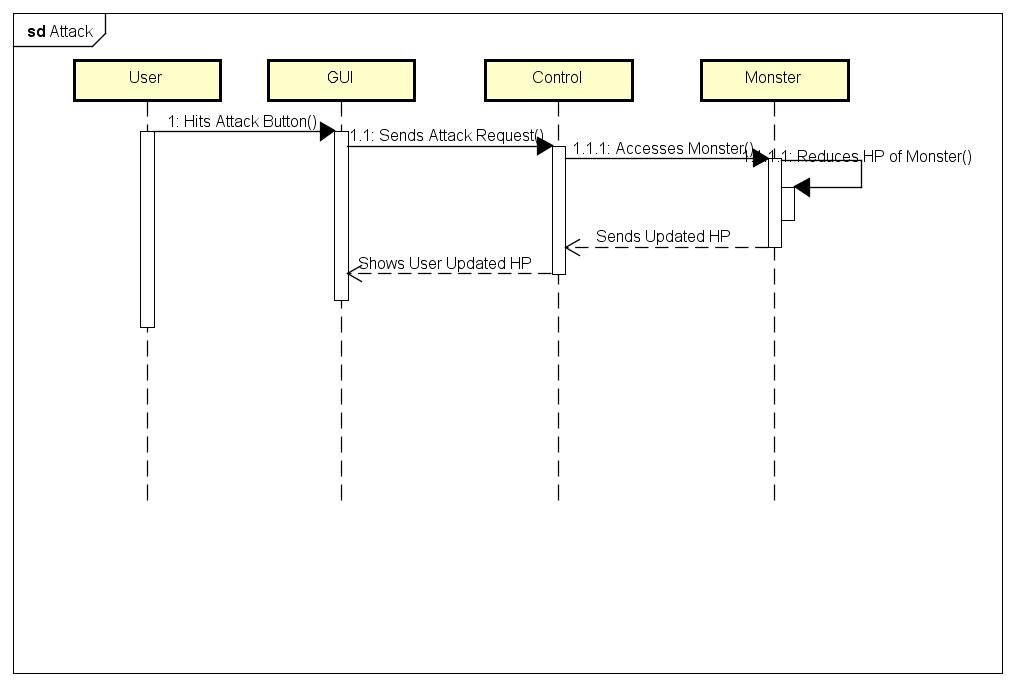


1.3.5 Lose Fight Diagram



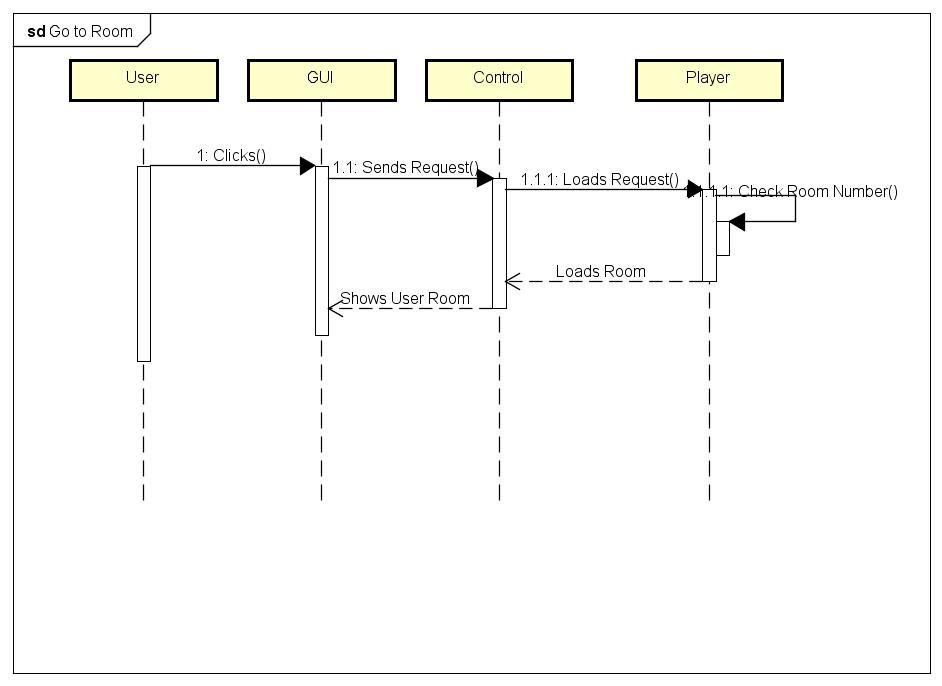
1.3.6 Receive Damage Diagram 

1.3.7 Attack Monster Diagram



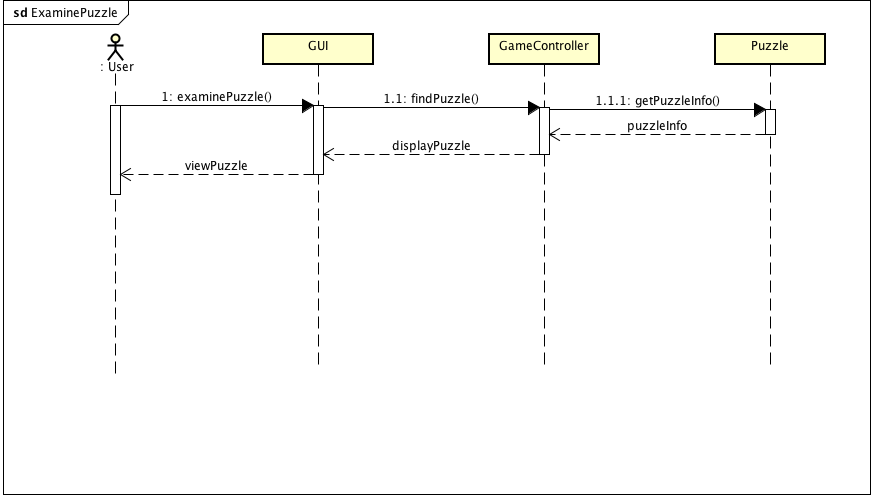
**3.1.4 Navigation Feature**

1.4.1 Go to Room Diagram

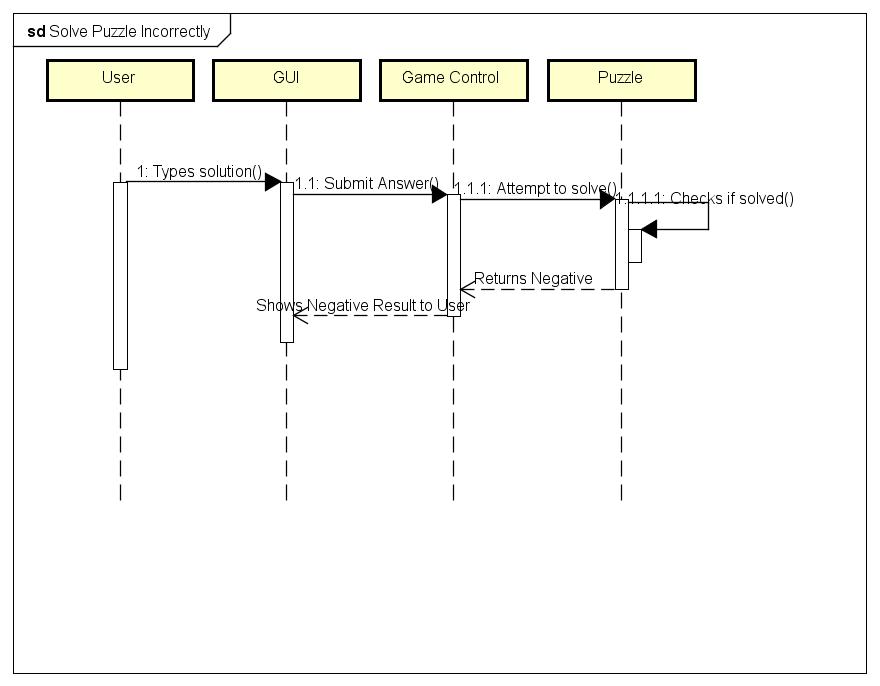


**3.1.5 Puzzle Feature**

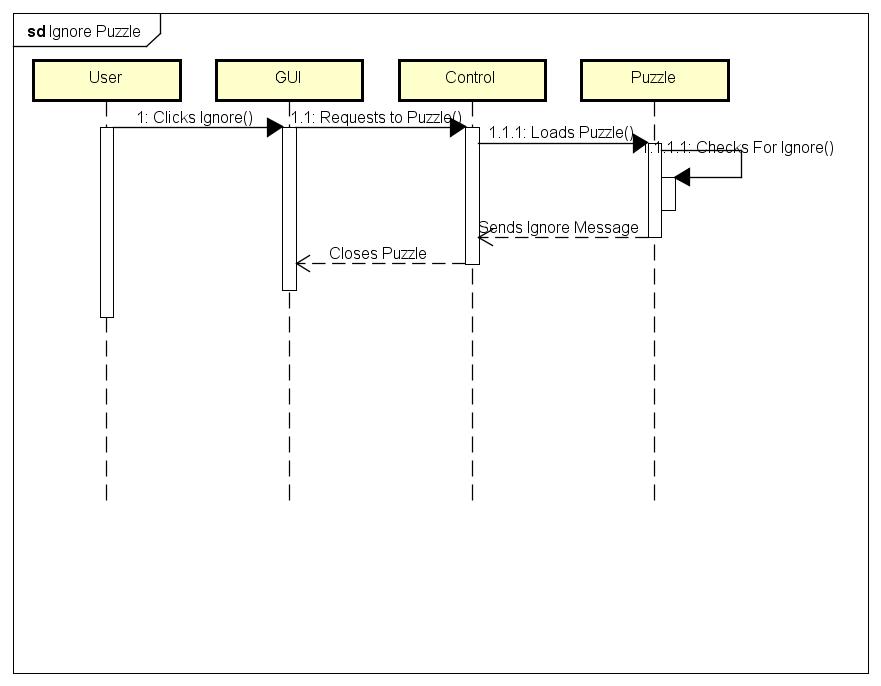
1.5.1 Examine Puzzle Diagram

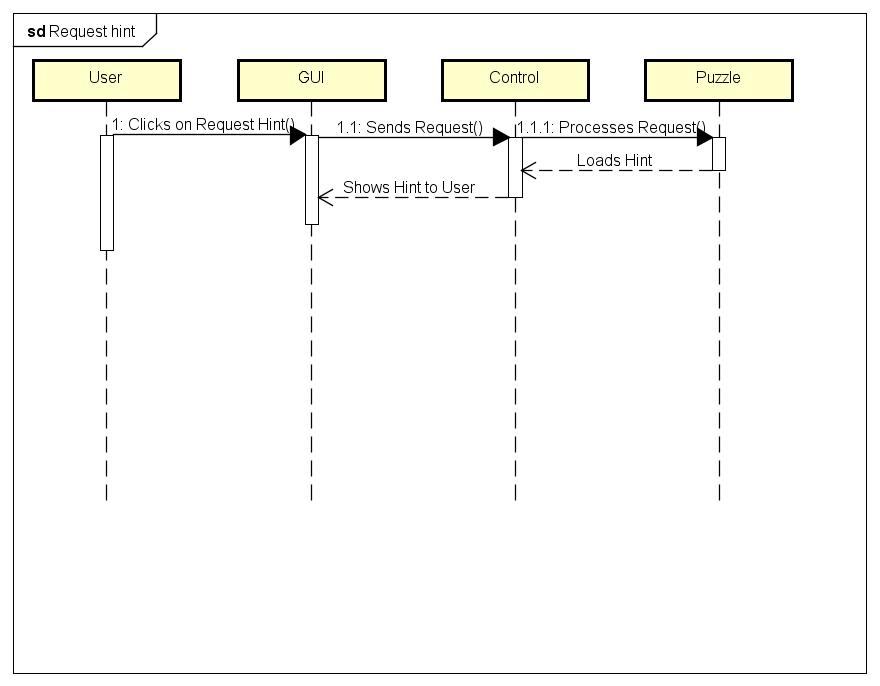


1.5.2 Solve Puzzle Incorrectly Diagram



1.5.3 Ignore Puzzle Diagram



3.5.4 Request Hint Diagram

1.5.5 Solve Puzzle Correctly Diagram

