ASSIGNMENT 3

SEG 2105 - INTRODUCTION TO SOFTWARE ENGINEERING

Fall 2021

School of Engineering and Computer Science

University of Ottawa

Course Coordinator: Miguel Garzon

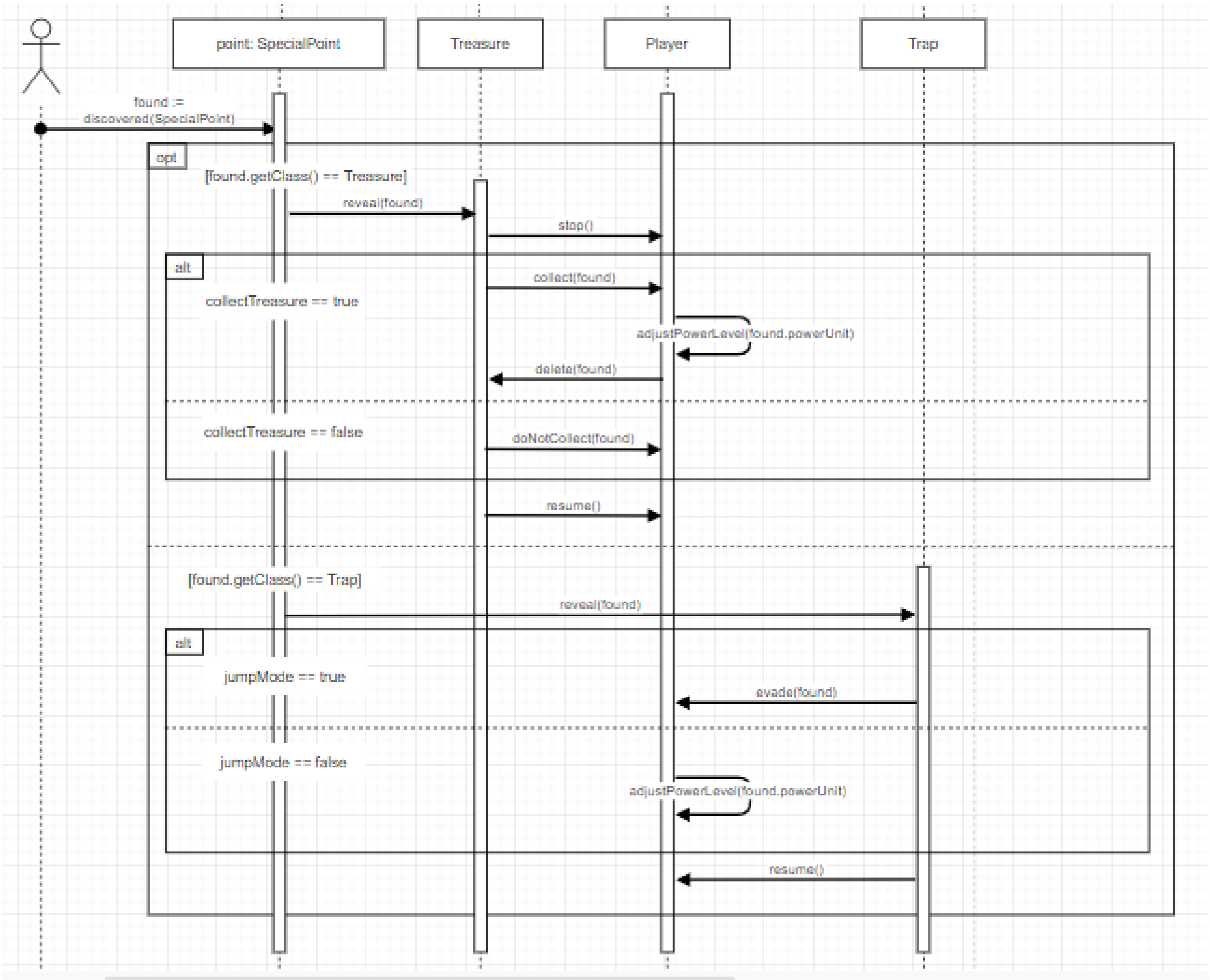
Teaching Assistant: TBD

Jayden Lachhman, 8791694

Submission date: 2021/12/04

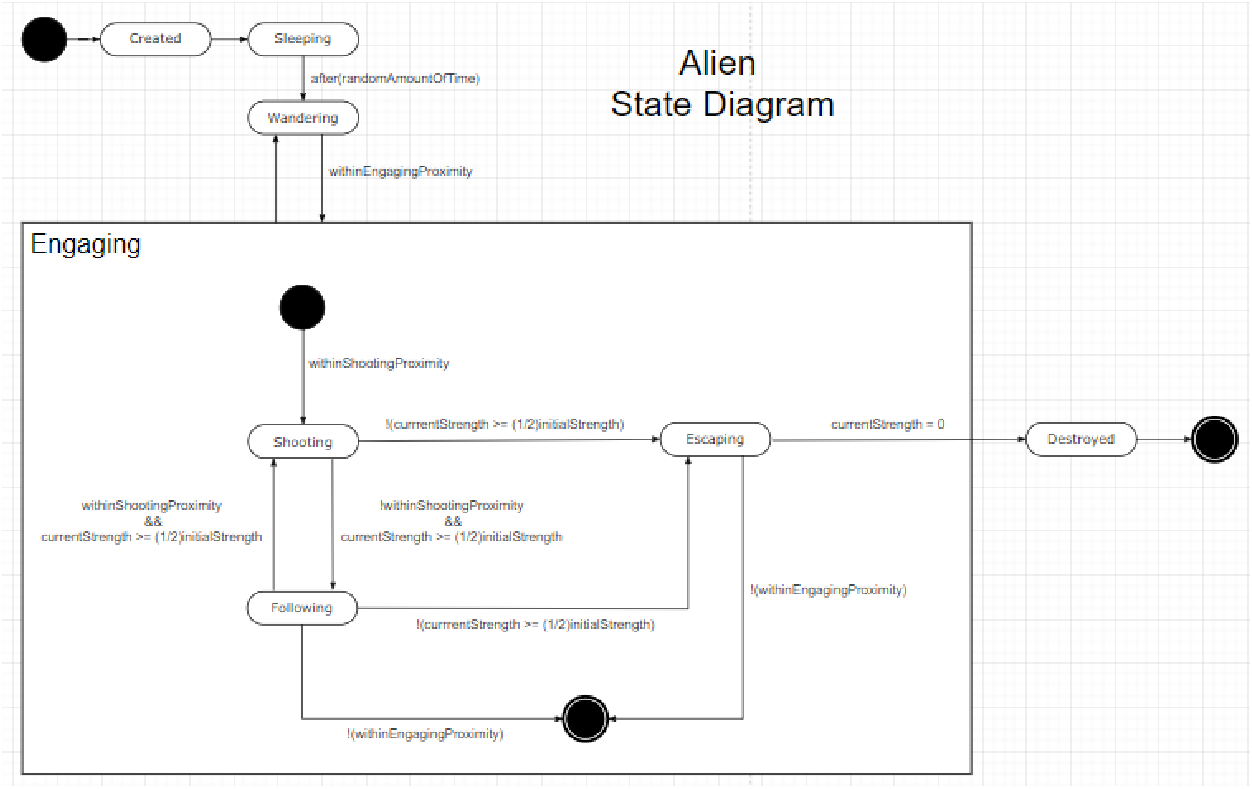
PART 1: CLASS DIAGRAM

PART 2: PLAYER-TREASURE-TRAP SEQUENCE DIAGRAM



One of the important implementation considerations made was that of when/if the SpecialPoint object should be removed from the level. Because, assuming the SpecialPoint object is of type Treasure, the problem description stated that a player could *choose* to collect the treasure. Thus, it should remain at its location in perpetuity assuming the player chooses to leave it such that they may return to it later to gain power units after potentially taking damage from other enemies. The treasure object in the diagram therefore cannot have a strict lifetime, since its lifetime depends on the player's decision, and thus I made it so that the object persists regardless. On the other hand, the problem description did not mention if booby traps could be dismantled or destroyed, by the player or by the game after damaging a player. Hence, I assumed booby traps would persist in perpetuity regardless of if a player evades them or not, so the player always has the potential to be hurt by them.

PART 3: ALIEN STATE DIAGRAM


Alien
State Diagram
Created
Sleeping
after(randomAmountOfTime)
Wandering
withinEngagingProximity
Engaging
withinShootingProximity
Shooting
!(currentStrength >= (1/2)initialStrength)
Escaping
currentStrength = 0
Destroyed
withinShootingProximity
&&
currentStrength >= (1/2)initialStrength
!withinShootingProximity
&&
currentStrength >= (1/2)initialStrength
!(withinEngagingProximity)
Following
!(currentStrength >= (1/2)initialStrength)
!(withinEngagingProximity)

PART 4: FUNCTIONAL/NON-FUNCTIONAL REQUIREMENTS

Example of a well-written requirement:
“The Online Banking System shall allow the Internet user to access her current account balance in less than 5 seconds.”

Functional Requirements:

1. The game shall allow the user to choose whether or not their avatar collects discovered treasure.
2.

Non-functional Requirements
(4) (2 quality)

PART 5: USE-CASES & USE-CASE DIAGRAM WITH DESCRIPTION

Git Repo Link: https://github.com/SEG2105-uottawa/Assignment3_8791694