

Name Hongxi Zhu

Mark                                 /50

### 1. Brief introduction   /3

My feature for the Galleon Game is design and implementation of overworld enemies and AI of enemies.

When player enter the game world, my job is to generate enough enemies in world randomly. I need to ensure that the positions of these generations must not be too gathering or more evacuated. I should try to avoid players from encountering some difficult enemies after completing a difficult battle, or the player wandered on the map for a few minutes and did not meet an enemy.

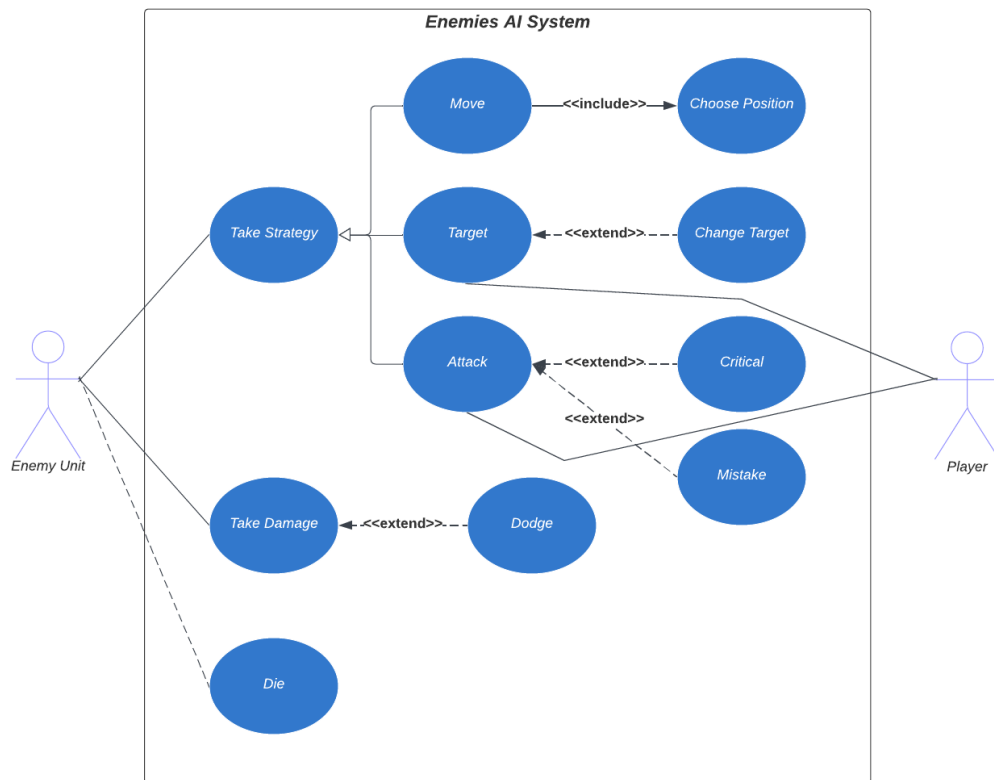
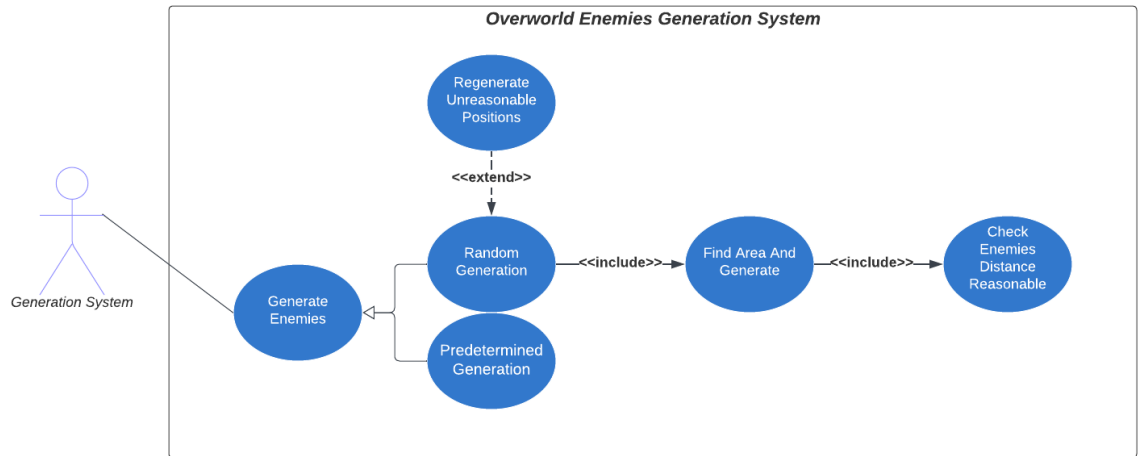
Moreover, I am responsible for implementing AI scripts for every enemy in the world. The AI scripts will include the following features: the randomization of unit position, the logic of choosing the attacking target, strategy of attacking or moving, etc.

## 2. Use case diagram with scenario \_14

### Use Case Diagrams

#### AI Specialist Features Diagram

Zhu, Hongxi (zhu1692@vandals.uidaho.edu)



## Scenario-Overworld Enemies Generation System

**Name:** Generate Enemies

**Summary:** The System Generate Enemies following rules.

**Actors:** Generation System

**Preconditions:** Game has started, and the Player enter the World.

**Basic sequence:**

**Step 1:** Check the World and Player status.

**Step 2:** Check Predetermined Generation Position Data.

**Step 3:** Keep generating enemies until the level settings are reached.

**Step 4:** Check whether the distance between all enemies conform to the rules.

**Exceptions:**

**Step 1:** Throw exception and return Main Menu.

**Step 2:** Generate the predetermined enemies.

**Step 3:** Clean all generation and re-generate.

**Step 4:** Re-generate specific enemies.

**Post conditions:** An event is triggered.

**Priority:** 1\*

**ID:** GE01

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

## Scenario-Enemies AI System\_ Take Strategy

**Name:** Take Strategy

**Summary:** Every Enemy has an attribute determine how often they take a strategy. When they take strategies, there are three choices.

**Actors:** Enemy unit

**Preconditions:** It's time for next strategy.

**Basic sequence:**

**Step 1:** Check the enemy status.

**Step 2:** Consider strategy weights to make decisions.

**Case Move:** Choose a right position to move.

**Case Target:** Find a right player unit as current target. But the target may be changed by some influence factor.

**Case Attack:** Attack current target, the damage value may be critical or zero because of mistake.

**Exceptions:**

**Step 1:** Stop to next strategy.

**Step 2:** Make a strategy again, if still failed, stop.

**Post conditions:** Enemy unit act as strategy.

**Priority:** 1\*

**ID:** AI01

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

### Scenario-Enemies AI System\_ Take Damage

**Name:** Take Damage

**Summary:** Reduce HP when player unit attack enemy unit.

**Actors:** Enemy unit

**Preconditions:** Enemy unit receive attack message from player unit.

**Basic sequence:**

**Step 1:** Check the enemy HP and status.

**Step 2:** Determine whether take damage according to the dodge rate.

**Exceptions:**

**Step 1:** Take damage directly.

**Step 2:** Take damage directly.

**Post conditions:** Reduce enemy unit HP.

**Priority:** 1\*

**ID:** AI02

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

### Scenario-Enemies AI System\_ Die

**Name:** Die

**Summary:** Destroy enemy unit if its HP is lower than zero.

**Actors:** Enemy unit

**Preconditions:** enemy unit HP is lower than zero.

**Basic sequence:**

**Step 1:** Check the enemy HP and status.

**Step 2:** Destroy enemy unit.

**Exceptions:**

**Step 1:** Destroy enemy unit directly.

**Step 2:** Destroy enemy unit directly.

**Post conditions:** Destroy enemy unit.

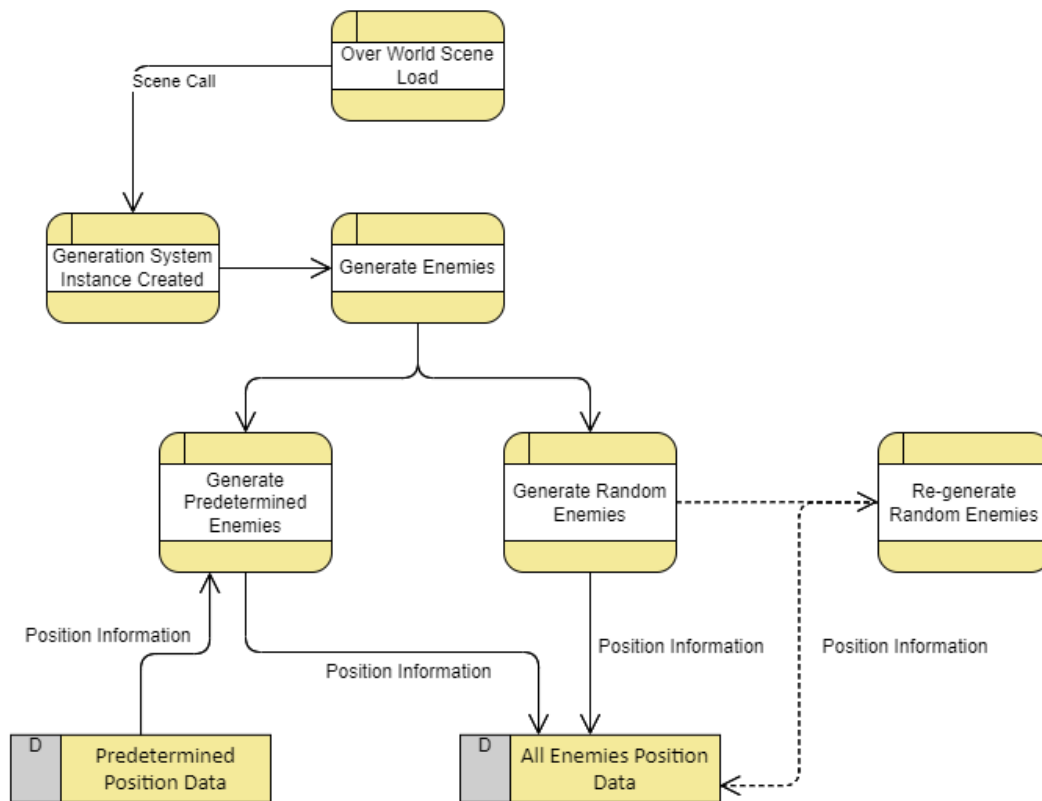
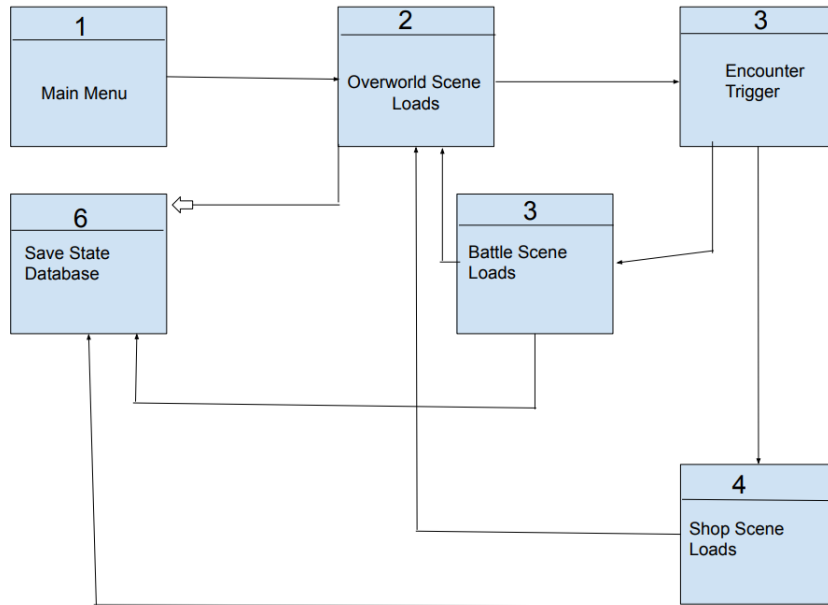
**Priority:** 1\*

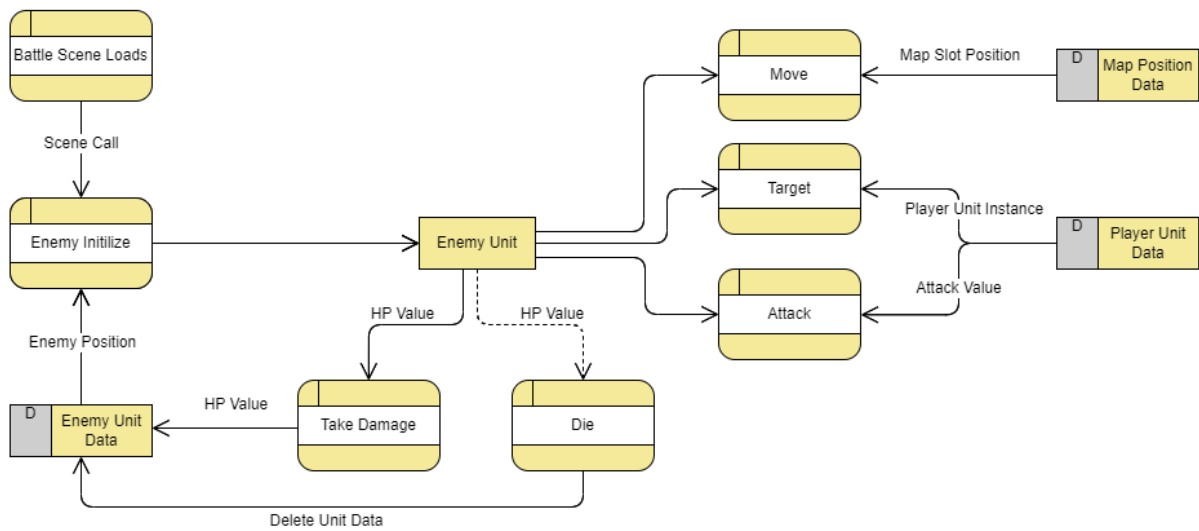
**ID:** AI03

\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

### 3. Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_14

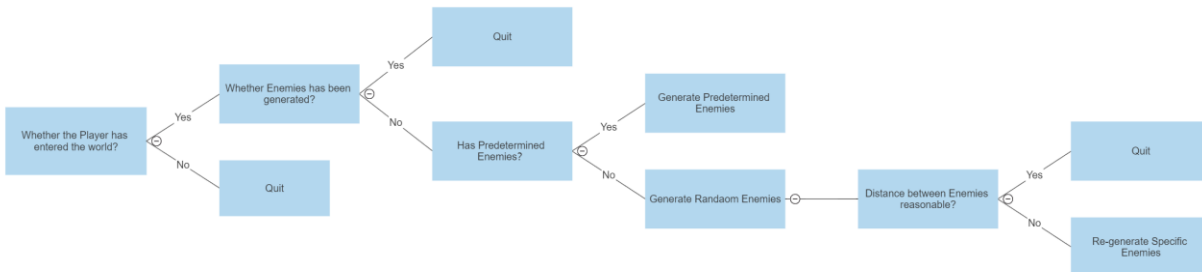
#### Data Flow Diagrams



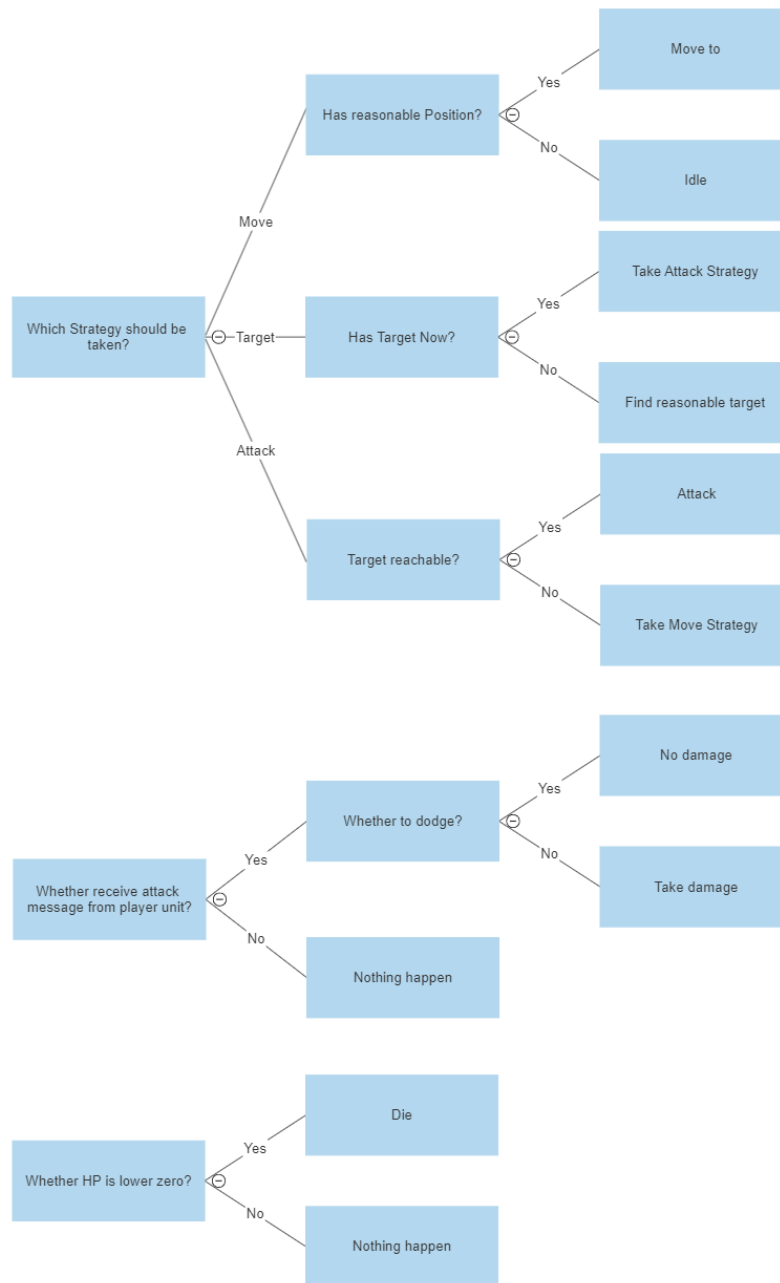


## Process Descriptions

The process description for Overworld Generation Enemies is displayed below in a decision tree.



The process description for Enemy AI is displayed below in a decision tree.



#### 4. Acceptance Tests \_\_\_\_\_9

##### Overworld Enemies Generation:

##### Test Case: Generate Enemies at Random Positions Within the Map

Input conditions: Map size is 1000\*1000, generate 30 enemies.

Expected output: 30 enemies at random positions within the map.

Steps:

1. Create a map with a size of 1000\*1000.
2. Trigger the event to generate 30 enemies.

Assertions:

1. The generated enemy count is 30.
2. Each enemy's position is within the map boundaries.

The output file will have the following characteristics:

- Enemies number: 30
- Enemies' positions
- Whether every enemy is within the map

Run feature 1000 times sending output to a file.

## **Enemy AI:**

### **Test Case: Enemy Strategy and Taking Damage from Player Attack**

Input conditions: The enemy's current health points are 100, it's located at a specific position on the map. The enemy can choose to either move, target or attack. The player's attack value is 100 and it will initiate an attack to enemy. The enemy act first.

Expected output: The enemy makes a strategy and takes damage from player then die.

Steps:

1. Set the enemy's current state, including its position and health points.
2. Trigger the enemy's decision-making process.
3. Simulate the player's attack.

Assertions:

1. The enemy's position or target has changed, or the HP value of the player has changed.
2. The enemy has dead.

The output file will have the following characteristics:

- Enemy's information including position, target, HP value.



- Player's information about HP value.

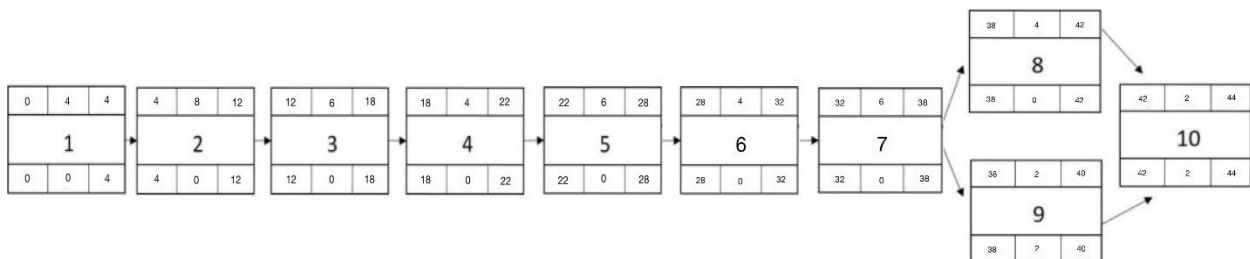
Run feature 1000 times sending output to a file.

## 5. Timeline \_\_\_\_/10

### Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Requirements Collection	4	-
2. Basic Knowledge Study	8	1
3. Framework Design	6	2
4. Database Construction	4	3
5. Overworld Enemies Generation Programming	6	4
6. Enemy Class Design	4	5
7. Enemy AI Programming	6	6
8. Documentation	4	7
9. Testing	2	7
10. Release	2	8, 9

### Pert diagram



## Gantt timeline

