Name: Andrew Sorensen Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My primary feature will be coding the bosses. This will include their mechanics, stats, behavior/movements (if applicable), phases, and implementing their sprites. The 3 bosses that we plan to add are a main feature of the game as these bosses will come at the end of every level (3 levels, 3 bosses).

Each boss will have their own theming and unique attack patterns, possibly with some randomness thrown in for the sake of replayability. Any randomness in the boss fights will be handled by an internal random number generator. Plans are for boss 3 to be a static enemy while bosses 1 and 2 are either semi-static or fully mobile. Phases within each boss will change a part of the combat once enough damage has been dealt to the boss.

## Use case diagram with scenario \_\_14

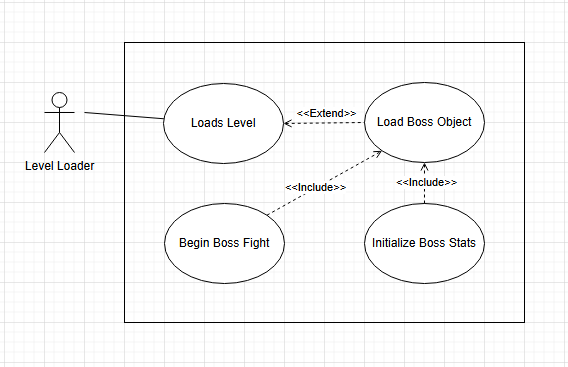
[Use the lecture notes in class.

Ensure you have at least one exception case, and that the <<extend>> matches up with the Exceptions in your scenario, and the Exception step matches your Basic Sequence step.

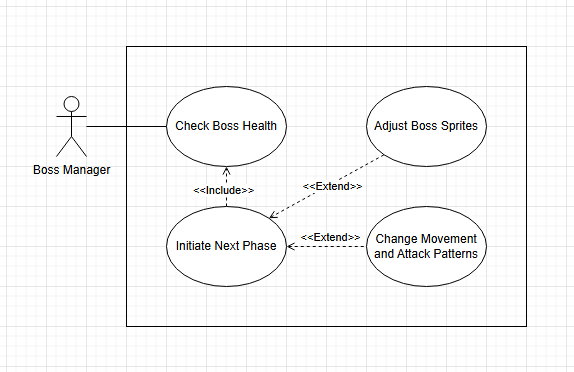
Also include an <<include>> that is a suitable candidate for dynamic binding]

### Use Case Diagrams

Use Case Diagram 1:



Use Case Diagram 2:



### Scenarios

**Name:** Summon Boss

**Summary:** The level loader loads the boss fight

**Actors:** Level Loader

**Preconditions:** The player has progressed enough for the level loader to be activated

**Basic sequence:**

**Step 1:** Load level

**Step 2:** Load boss object as part of loading level

**Step 3:** Initialize boss stat values

**Step 4:** Begin boss fight

**Exceptions:**

**Step 1:** Boss fails to load as part of loading the level: attempt level reload

**Step 2:**

**Post conditions:** Boss is loaded and ready to fight the player character

**Priority:** 1

**ID:** UC1

**Name:** Change Boss-Phase

**Summary:** Progress from current to next Boss-Phase when a certain health threshold is reached

**Actors:** Boss Manager

**Preconditions:** Boss is loaded in and Boss Fight has been initiated

**Basic Sequence:**

**Step 1:** Check Boss Health

**Step 2:** Initiate transition to next Boss-Phase

**Step 3:** Change movement, attack patterns, and sprites as needed

**Exceptions:**

**Step 1:** Boss has not reached health threshold: do nothing

**Step 2:** Boss health is at or below 0: Boss death

**Post Conditions:** Boss’s next phase has begun

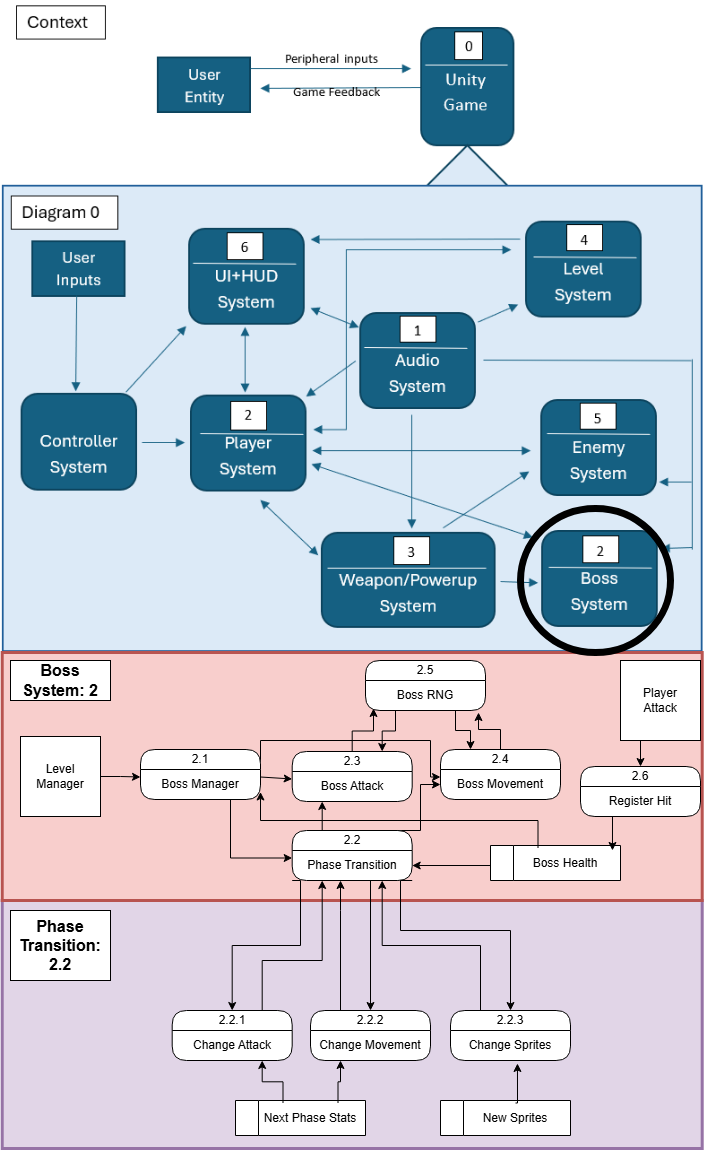
**Priority:** 2

**ID:** UC 2

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

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

### Data Flow Diagrams



### Process Descriptions (WIP)

Assign rooms\*:

WHILE teacher in two places at once OR two classes in the same room

Randomly redistribute classes

END WHILE

**\*Notes**: Yours should be much longer. You could use a decision tree or decision table instead if it is more appropriate.

## Acceptance Tests \_\_\_\_\_\_\_\_9 (WIP)

Run **each** Boss Fight/Encounter 50 times (150 total for all 3 bosses) and note whether all aspects of boss fight trigger.

Expected Behaviors:

* Boss Spawns in and Boss Fight begins upon Boss Level entry
* Boss attacks and movements are truly random (when RNG is added) and does not just loop the exact same attack pattern each time.
  + Some repetition is expected, but each fight should be somewhat unique
* Boss attacks hurt player as expected
* Boss-Phase transition happens at designated health threshold
  + Does not happen too early
  + Does not happen too late
* Boss’s attacks and movements actually change after the phase transition (when prescribed)

## Timeline \_\_\_\_\_\_\_\_\_/10 (WIP)

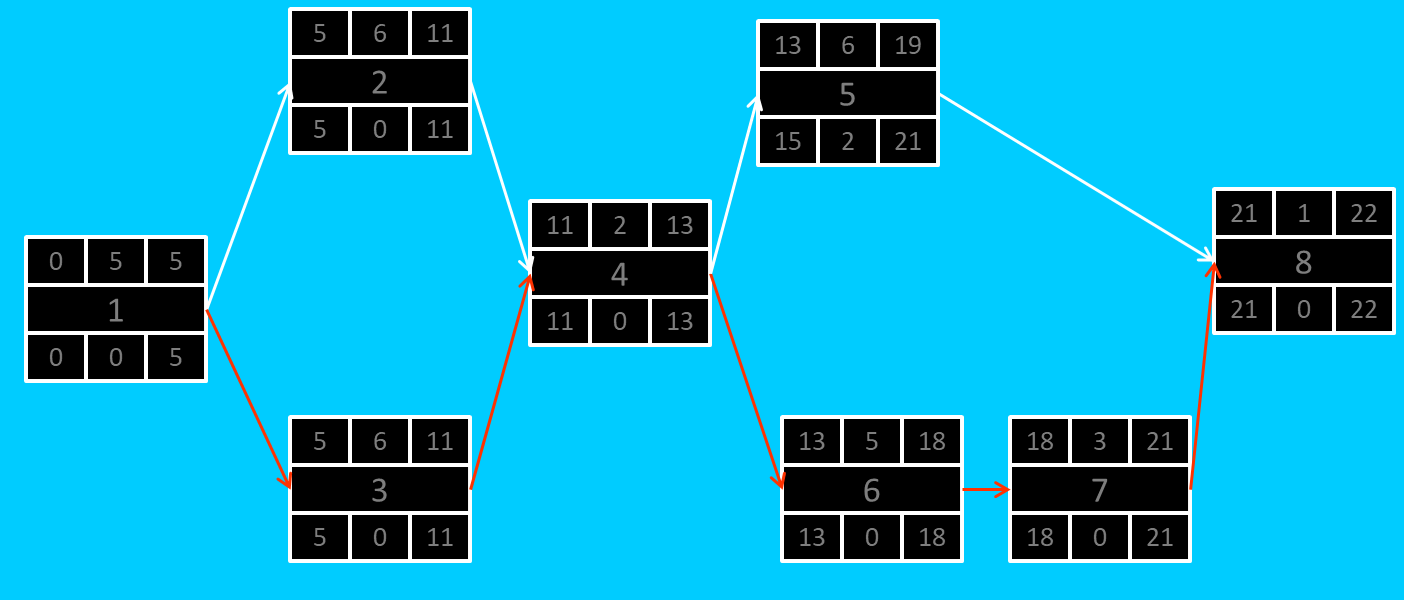
[Figure out the tasks required to complete your feature]

Example:

### Work items

| Task | Duration (PWks) | Predecessor Task(s) |
| --- | --- | --- |
| 1. Requirements Collection | 5 | - |
| 2. Screen Design | 6 | 1 |
| 3. Report Design | 6 | 1 |
| 4. Database Construction | 2 | 2, 3 |
| 5. User Documentation | 6 | 4 |
| 6. Programming | 5 | 4 |
| 7. Testing | 3 | 6 |
| 8. Installation | 1 | 5, 7 |

### Pert diagram



### Gantt timeline

| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 7 |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |