Name: Gabriel Jones Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My champion feature will be the Goals designated for the enemies to attack. These are game objects that the enemies will pathfind to by default. The player will have to manage the goals, and make sure they do not become destroyed, as this would result in a loss for that level. As such, each level must have at least one goal. Each goal will have hp values, and conduct a proximity check per game cycle. If the right conditions are met, the tower may attack enemies within range.

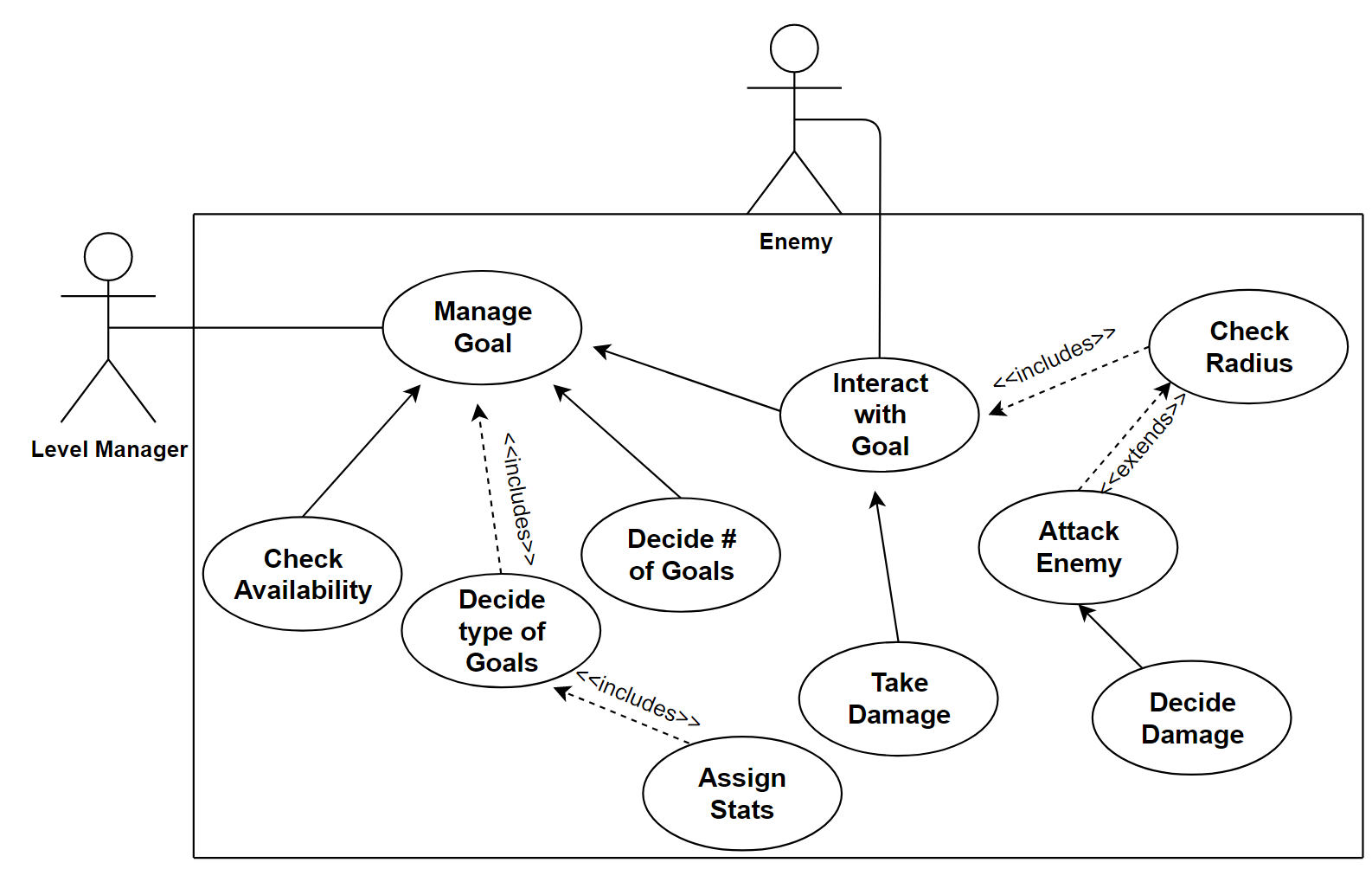
I am also going to be responsible for all game menuing, HUD creation, and UI elements. This will include objects such as the following:

* Main menu title screen
* Options menu
* Credits menu
* Display in game that shows various important player information, such as:
  + Player hp/Mana/Boss hp and statistics/Goal hp/Available spells and summons…etc
* Separate “Hover over”/Popup menu that details current/next wave information

## Use case diagram with scenario \_\_14

[Use the lecture notes in class.

### Use Case Diagrams



### Scenarios

**Name:** Goal Manager

**Summary:** The Goal Manager takes data from the Level Manager and creates/instantiates types of goals relevant. Also allows for enemy interaction with goals

**Preconditions:** Level Manager has been initialized

**Basic sequence:**

**Step 1:** Accept type/name of level loaded and game difficulty

**Step 2:** Manager takes this information and decides number of goals to spawn

**Step 3:** Manager now decides type of goals to spawn

**Step 4:** Manager instantiates goals with correct init data

**Basic sequence 2 (goal damaging enemy)**

**Step 1:** Goal object checks radius for enemy game objects every game update

**Step 2:** Goal attacks enemy if within predetermined range

**Exceptions:**

**Step 1:** Radius is checked, but no enemy is found within range

**Post conditions:** Correct goal is initialized for the level, and enemy goal interaction results in damage being dealt to the proper targets.

**Priority:** 2\*

**ID:** G01

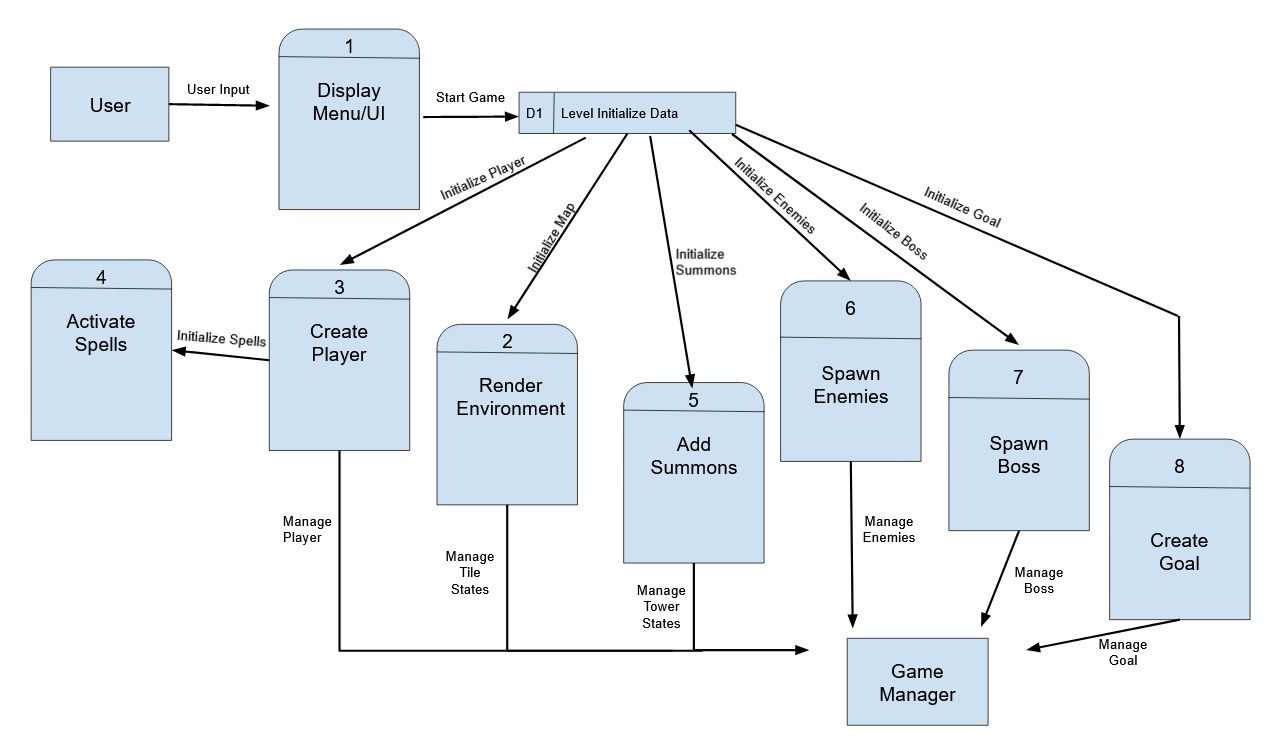
\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

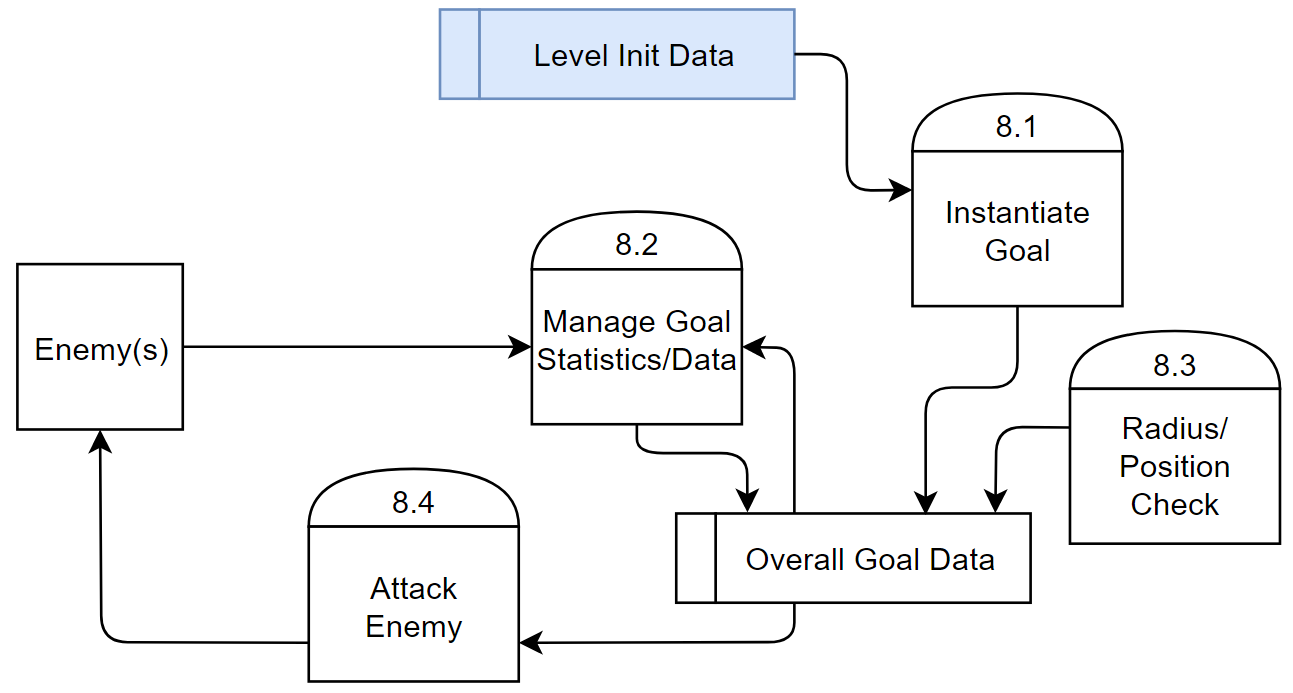
## Data Flow diagram(s) from Level 0 to process description for your feature \_\_\_\_\_\_\_14

[Get the Level 0 from your team. Highlight the path to your feature]

Example:

### Data Flow Diagrams





### Process Descriptions

Instantiate Goal(s):

Once Level Init Data has been calculated, some of this information will be sent to this process. This process will use this data to determine which types of goal to create, as well as the number.

Radius/Position Check:

This process will be a continuous loop called every game update per instance of a goal. The process will check for enemies in range of a predetermined radius and then determine with a raycast if they are within line of sight or not visible to the goal.

Manage Goal Statistics/Data:

Once ‘Overall Goal Data’ has been created, this process can determine and change various data values of the goal in question. This will include goal hp, status, animation changes, resistances, …etc.

Attack Enemy:

If the ‘Radius/Position Check’ process returned meaningful data, then this process should take place. ‘Attack Enemy’ will determine attack type and damage if necessary.

## Acceptance Tests \_\_\_\_\_\_\_\_9

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

Feature will be run per level to ensure correct number and type of goals are assigned.

Enemies will be spawned at various angles around a goal. This should ensure enemies are able to damage the goal regardless of level positioning.

Goal will have its bonus condition met for further testing. Animation and other damaging abilities should activate. Any enemy placed around the goal should be recognized by the goal and an action – most like an attack - should take place. In addition, any stationary enemy placed around the goal outside of range should not be targeted by the goal.

## Timeline \_\_\_\_\_\_\_\_\_/10

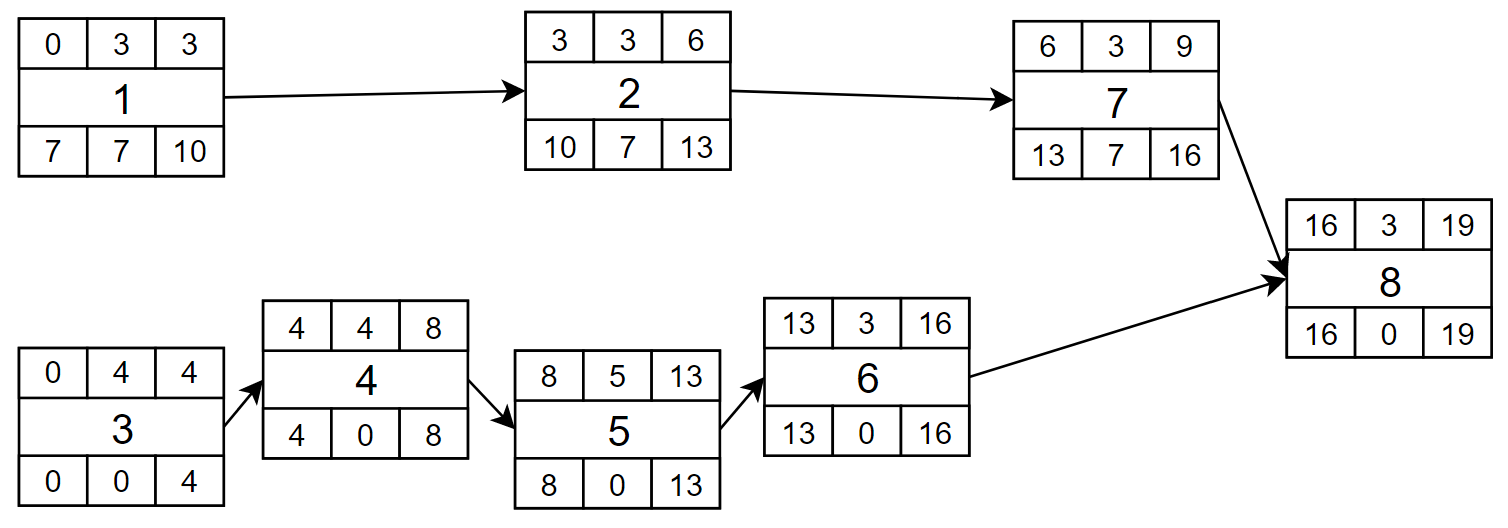
[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. Menu creation | 3 | - |
| 2. UI Creation | 3 | 1 |
| 3. Goal Manager Script Coding | 4 | - |
| 4. Goal Superclass | 4 | 3 |
| 5. Goal Subclasses | 5 | 3,4 |
| 6. Testing/Balancing | 3 | 3,4,5 |
| 7. Hud Elements | 3 | 1,2 |
| 8. Integration Checks | 3 | All Previous |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 |