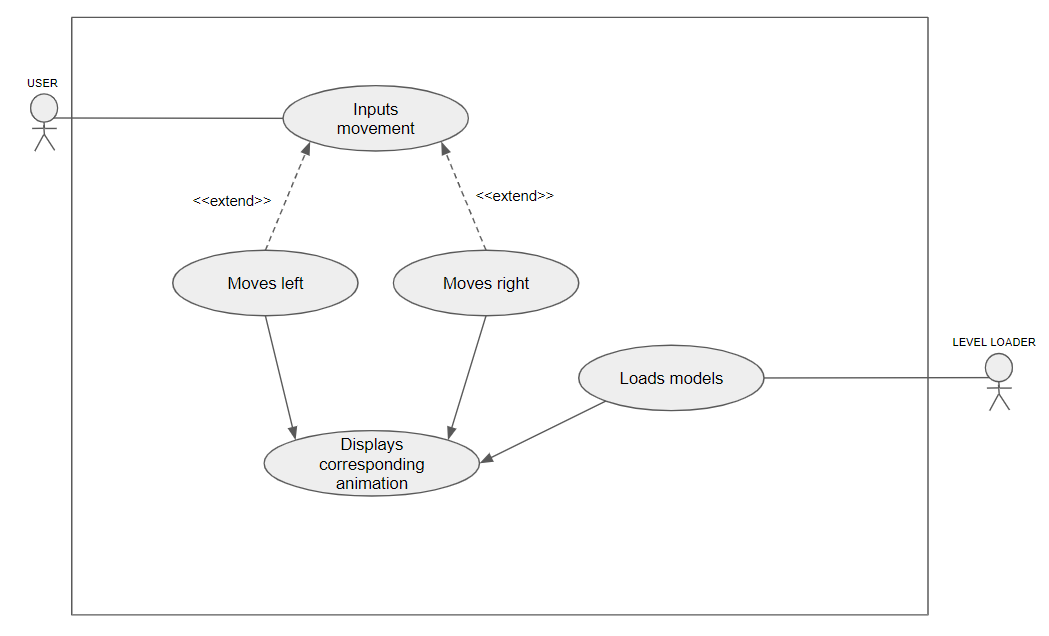
Name: Joshua Dempsey\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

My feature includes the loading of object and environment assets such as the main player character model. My process also evaluates and displays animations depending on what model is in question and what outside stimulus has triggered an event.

## Use case diagram with scenario \_\_14

### Use Case Diagram



### Scenarios

**[You will need a scenario for each use case]**

**Name:** loads model

**Summary:** A request to load the level’s models has been issued from the Level loading process

**Preconditions:** Level Load request

**Basic sequence:**

**Step 1:** determine what level is to be loaded

**Step 2:** fetch corresponding level’s models

**Step 3:** load models into game environment

**Step 4:** Display level model’s initial animations

**Exceptions:**

**Step 1:** failure to load requested models: failed models replaced with placeholder models and unfulfilled requests logged

**Post conditions:** level’s models.

**Priority:**1\*

**ID:** C01

**Name:** moves left

**Summary:** The player has issued a request to move left to the player character model

**Preconditions:** Level has been loaded, model has been loaded

**Basic sequence:**

**Step 1:** determine and evaluate that a left movement request has been input

**Step 2:** determine how extreme movement command was

**Step 3:** load corresponding left movement animation, rotating rudder, wheel, and sails

**Step 4:** Display animation to user

**Exceptions:**

**Step 1:** both left and right are pressed at the same time: ignore animation request, do not display rotation animations

**Step 2:** A certain amount of time elapsed with no movement request: Display corresponding Idle animation

**Post conditions:** Corresponding Animation is displayed to user.

**Priority:** 3\*

**ID:** C02

**Name:** moves right

**Summary:** The player has issued a request to move right to the player character model

**Preconditions:** Level has been loaded, model has been loaded

**Basic sequence:**

**Step 1:** determine and evaluate that a right movement request has been input

**Step 2:** determine how extreme movement command was

**Step 3:** load corresponding right movement animation, rotating rudder, wheel, and sails

**Step 4:** Display animation to user

**Exceptions:**

**Step 1:** both left and right are pressed at the same time: ignore animation request, do not display rotation animations

**Step 2:** A certain amount of time elapsed with no movement request: Display corresponding Idle animation

**Post conditions:** Corresponding Animation is displayed to user.

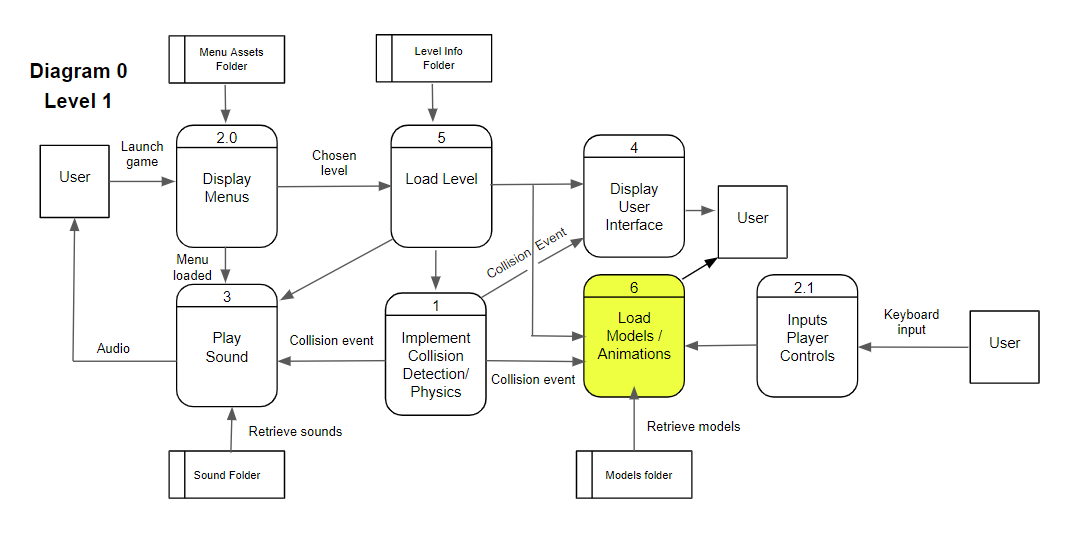
**Priority:** 3\*

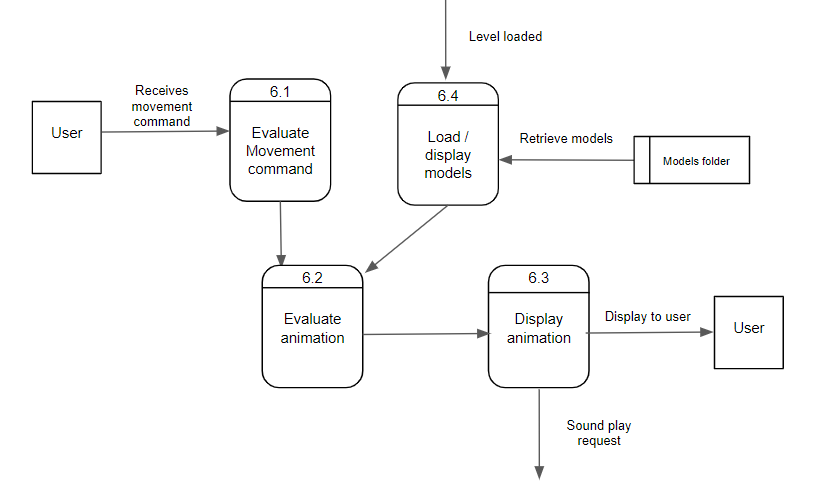
**ID:** C03

\*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

Evaluate Movement command:

Confirm that a movement command has indeed been entered, diagnose whether movement was a left or right press

Evaluate animation:

If movement command was diagnosed as a left request

Queue up corresponding animation for rotating model left

If movement command was diagnosed as a right request

Queue up corresponding animation for rotating model right

Display animation:

Confirm animation request

Begin animation display to user

Request sound play for corresponding animation

Load / display models:

After a level load request has been confirmed, retrieve required models from model database

Display corresponding models and queue up their initial animations

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

**Example for loading models feature**

Stress test to load multiple models concurrently 100 times and log output

The output file will log the model name, location, size, and success or failure of each load request

**Example for display animation feature**

Loop and continuously run all model animations concurrently to examine stress on game’s performance. Write to a file recording number of models playing animations and current game framerate.

**Example for evaluating movement command feature**

Send 1000 movement requests of random left, right, and unidentified movement events. Test the evaluation of them and write original request and evaluation to a file to compare that movement events are being diagnosed correctly

**Example for evaluating animation feature**

Send 1000 animation requests to models and test their evaluation. Write what request was made and what animation was played and compare to confirm accuracy that the animations being displayed are the correct evaluations.

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

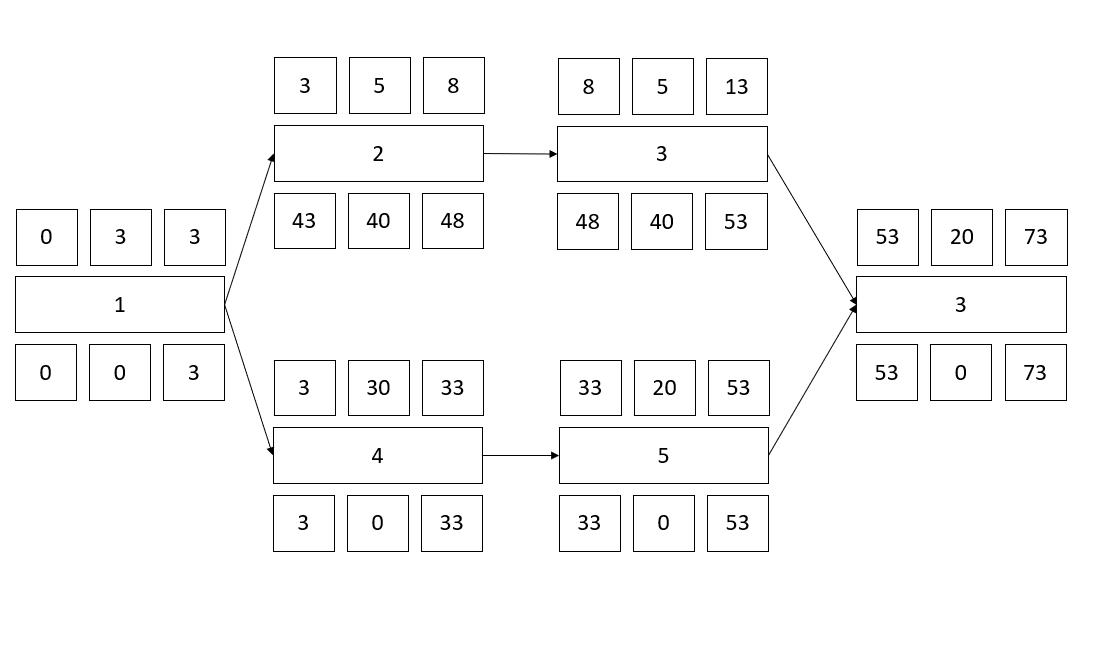
[Figure out the tasks required to complete your feature]

Example:

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (PWks) | Predecessor Task(s) |
| 1. logo creation | 3 | - |
| 2. player character modeling | 5 | 1 |
| 3. player character animation | 5 | 2 |
| 4. object / environment assets model | 30 | 1 |
| 5. object / asset animations | 20 | 2, 4 |
| 6. Testing | 20 | 1,2,3,4,5 |

### Pert diagram



### Gantt timeline

