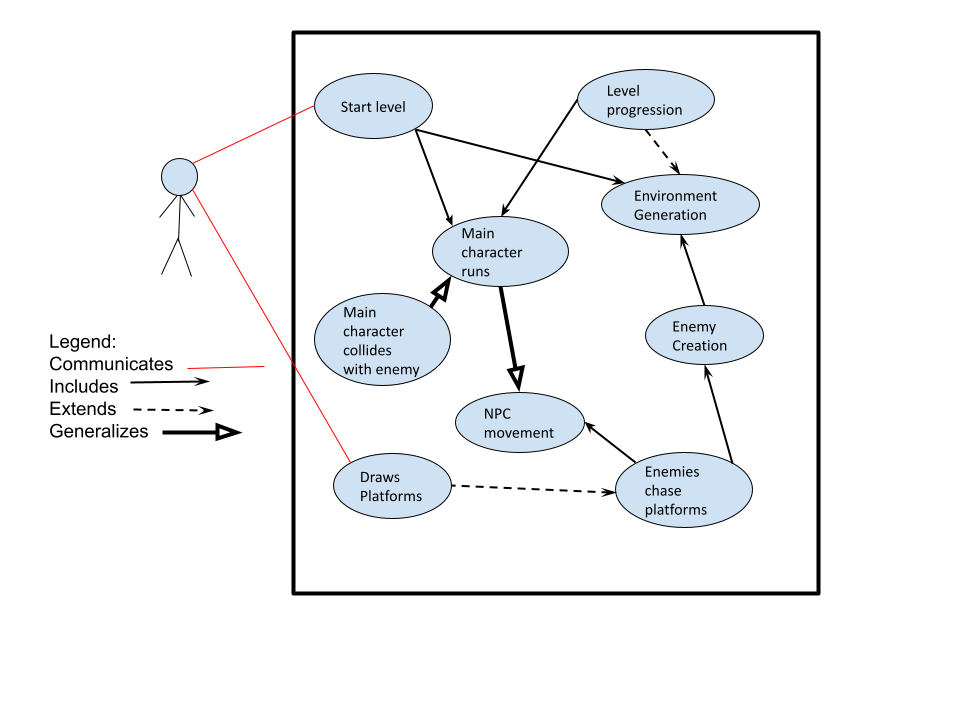
Name: Nathan Nguyen Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

## Brief introduction \_\_/3

I will be championing the NPCs within Lake Runner. Specifically this pertains to enemy and player sprite animations as well as enemy behavior movement and interactions. In our game, users do not control the player, but rather they manipulate the environment around the main character while he simply runs, jumps, falls, or otherwise moves constantly across the level. Additionally, Enemies and dangers will be on the level, and when the users draw new platforms, those enemies will need to react to the changing environment. I will be implementing those behaviors.

## Use case diagram with scenario \_\_14

Global use case included in group documents, not shown here.

note: I am using dotted lines to designate extends because it effectively communicates a difference between includes and extends.

### Scenarios

**Name:** NPC initialization and behavior.

**Summary:** The player starts the game, causing the player to run and enemies to appear.

**Actors:** Player

**Preconditions:** Level has been selected

**Basic sequence:**

**Step 1:** Player begins level.

**Step 2:** The Environment is generated, Enemies are generated, and the main character begins to run.

**Step 3:** Enemies move around the stage colliding with the player causing other game interactions.

**Exceptions:**

**Step 1:** Player draws a platform, causing enemies to chase the new platform.

**Step 2:** Player progresses in the level, causing more of the level to be generated.

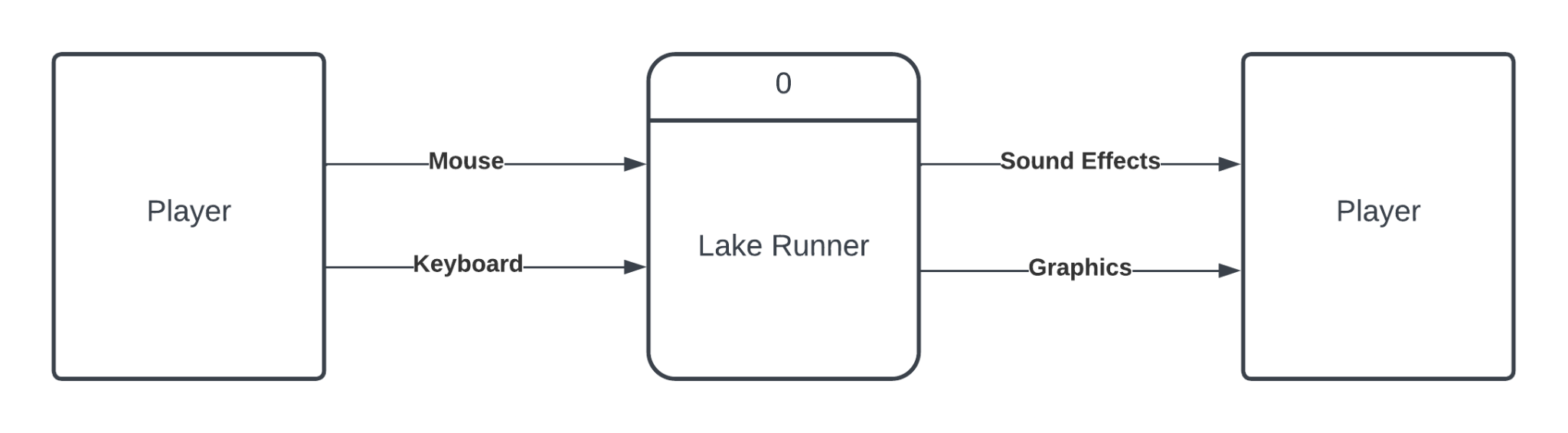
**Post conditions:** Calculated value is displayed.

**Priority:** \*1

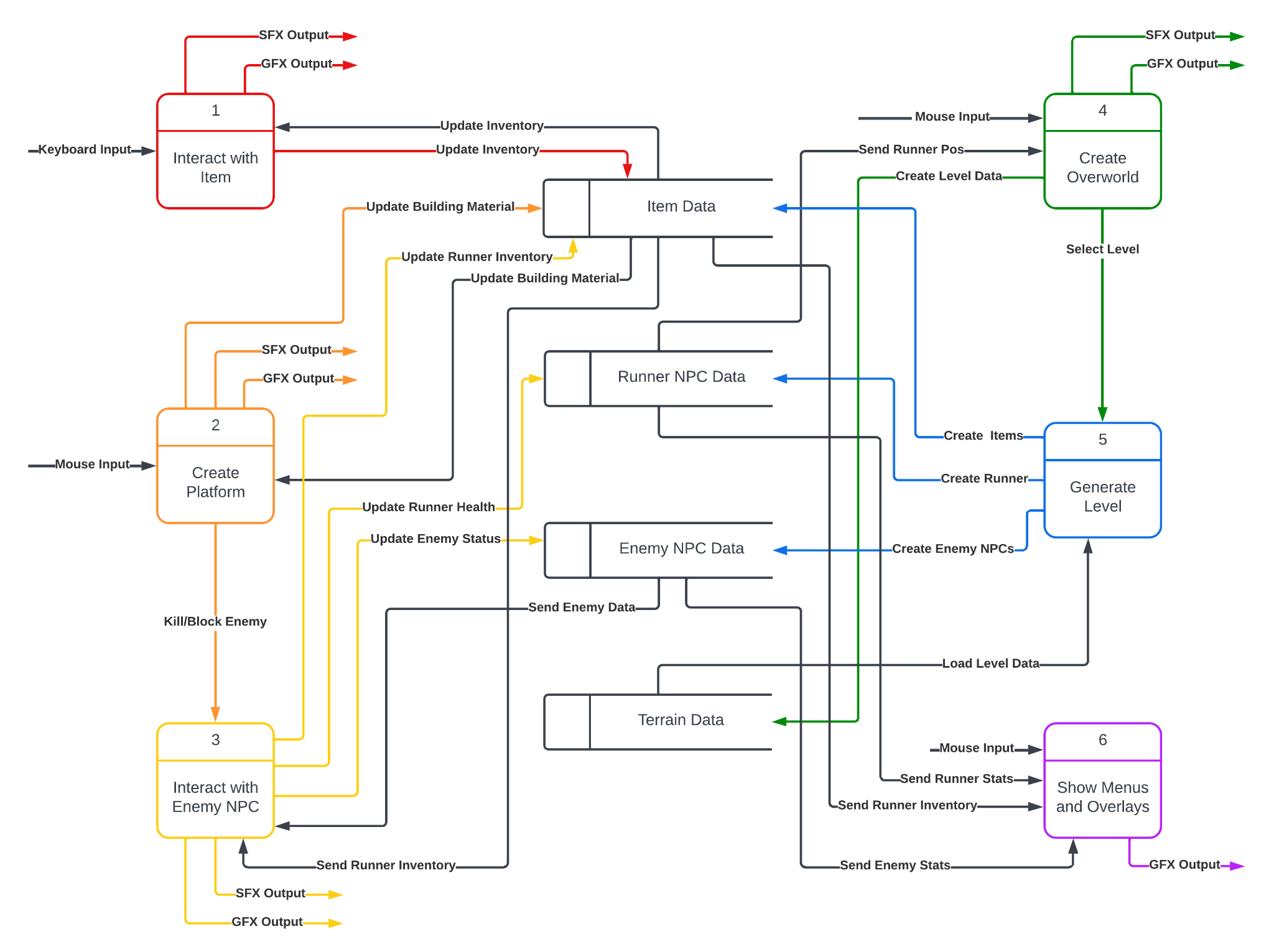
**ID:** C3

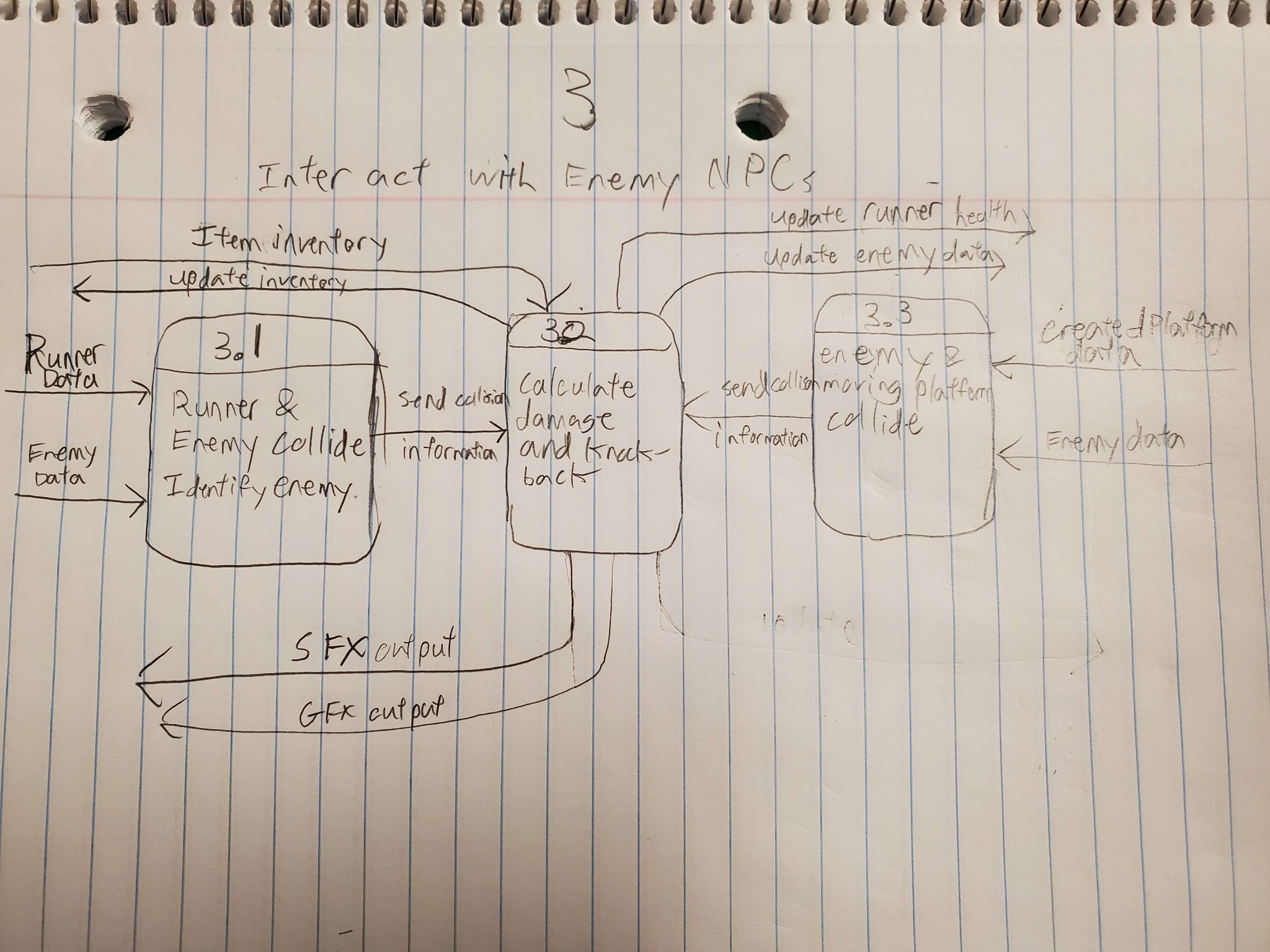
\*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

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

Here is our context diagram

**Data Flow Diagrams:** This is our Diagram 0. My features involve process number 3, and its connections, highlighted in Yellow.



Seeing as Umbrello will not run on my laptop, Diagram 3 is hand-drawn.

### Process Descriptions:

3.1: If runner position and enemy position overlap, determine if the enemy is a crab, coconut, or crocodile. Also identify which direction the runner collided with the enemy from (above, below, front, back).

3.2: If the collision was between the runner and a crab and the player did not collide with the crab from below, check the player’s item inventory and remove a random item, add that item to the crab’s inventory.

If the collision was with the runner and a coconut, and the runner did not collide from above, check the runner’s health, subtract 1hp, and knock the runner back while killing any momentum and destroying the coconut. If the collision was from above, destroy the coconut and bounce the player upward.

If the collision was with a crocodile and not from behind, check the player’s health and subtract 2 hp. If the collision was from behind, cause the runner to jump high.

If the collision was with a platform and an enemy, and the collision was from above, damage the enemy according to the enemy health and size of the platform.

Play sound effects and animations accordingly.

3.3: Whenever a new moving platform collides with an npc, determine the health of the npc, the size of the platform, and the direction which the platform collided with the npc from.

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

Enemy NPC Behaviors:

Place 100 enemies in a 10x10 grid near where three new platforms will be placed. Two of the new platforms should appear simultaneously to see how the AI reacts. the other platform should be placed such that enemies are above, below, inside of, and/or trapped by the platform.

Test should be run three times, with enemies stationary, speeding up, and at max speed. The test will be considered successful if after 30 seconds, the velocity of each enemy varies by less than 5% (no glitchy shaking) and the position of the enemy is within 50% the platform’s length from the center of the platform, or the enemy is obstructed from getting to the platform.

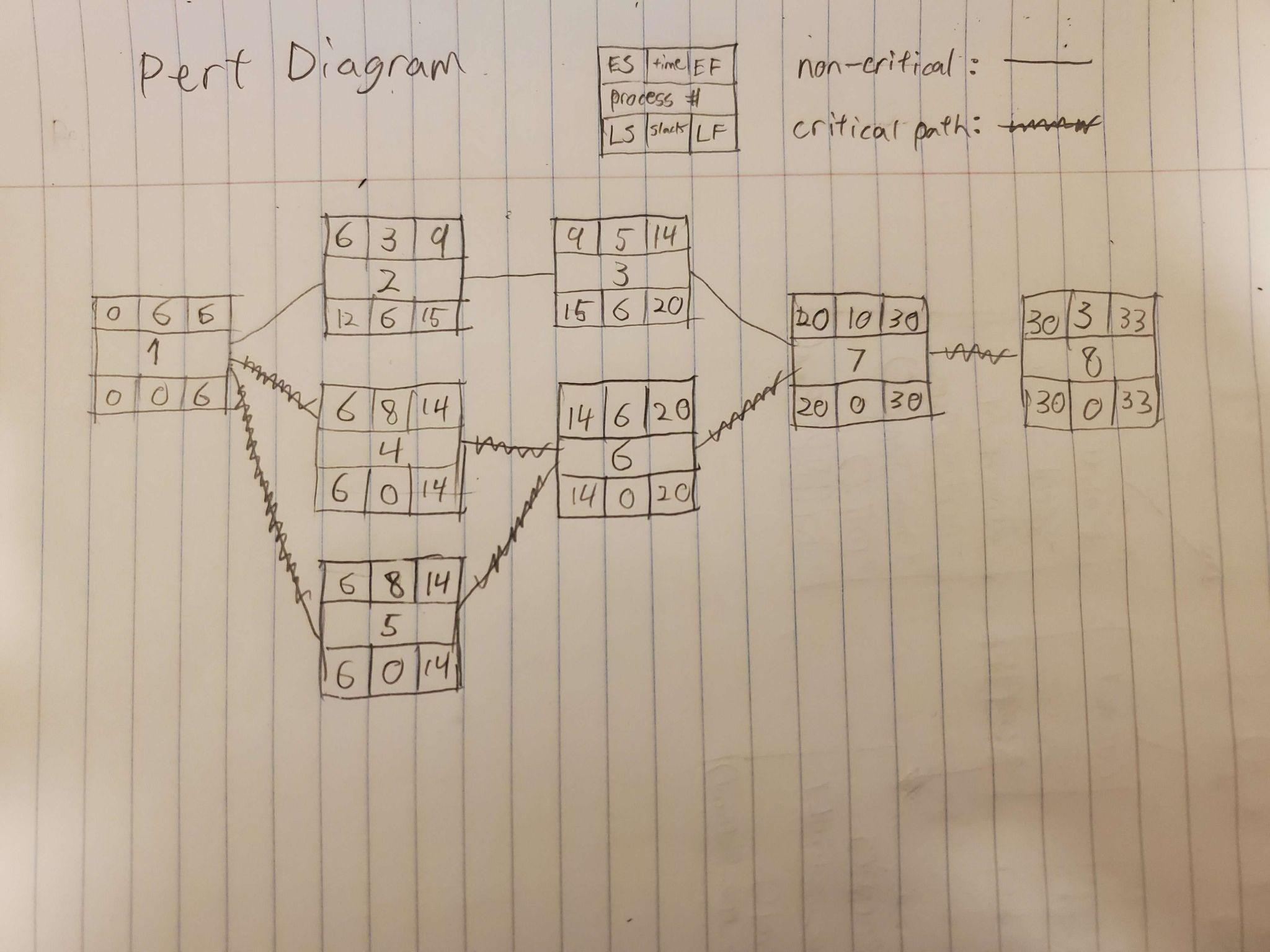
Main character behavior test: Ten platforms will be placed in various positions in front of the main character, some of the platforms will be too tall to jump over, some will be just tall enough, some will be slanted at angles which the main character cannot walk up, and some platforms will be too short to merit jumping over. The test will specify the desired position on the level for the main character to arrive at when it encounters the various platforms. If the player character is not within an acceptable distance from the desired location, the test will be considered a failure.

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

### Work items

| Task | Duration (PWks) | Predecessor Task(s) |
| --- | --- | --- |
| 1. Requirements Collection | 6 | - |
| 2. Find Entity sprites | 3 | 1 |
| 3. Implement Animations | 5 | 2 |
| 4. Create Enemy AI | 8 | 1 |
| 5. Create Player AI | 8 | 1 |
| 6. NPC Interactions With Platforms | 6 | 4,5 |
| 7. Testing and rework | 10 | 3,6 |
| 8. Project Demoing | 3 | 7 |

### Pert diagram



### Gantt timeline

