Name\_\_Noah Rieth\_\_\_\_\_\_\_\_\_ Mark \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/50

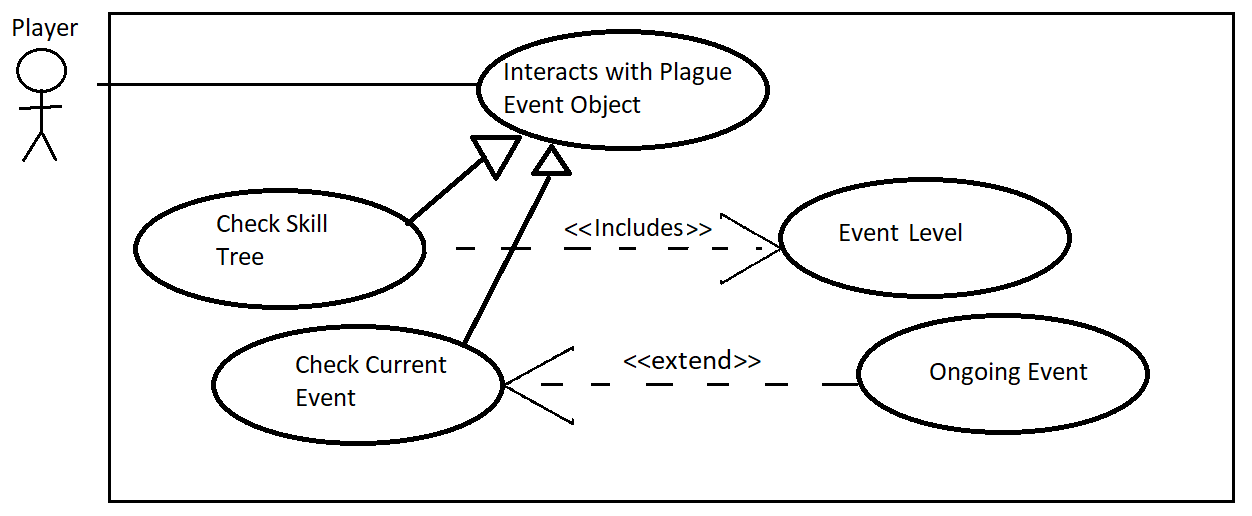
## Brief introduction \_\_/3

My feature of the Exodus video game is to manage “Plague Event” object creation and this object’s child objects during its lifetime. An active Plague Event will alter regular gameplay via a variety of changes, depending on which type of Plague Event was activated. The types of changes these Plague Events make will reflect the biblical plagues on Egypt while under Pharaoh’s rule, such as Darkness, Hail, Blood, Frogs, etc.

A Plague Event can be instantiated when particular conditions are met in the base game. These conditions are as follows: the player interacts with a collider object that is linked to a Plague Event, There is no Plague Event going on already, and the level (i.e. difficulty) of the Plague Event can be determined.

## Use case diagram with scenario \_\_14

### Use Case Diagrams



### Scenarios

**Name:** Plague Event Created

**Summary:** The Player triggers a Plague Event

**Actors:** The Player.

**Preconditions:** The player collides with a game object that triggers PlagueEventCreation().

**Basic sequence:**

**Step 1:** PlagueEventCreation() searches the skill tree/inventory for the triggered event type and finds the level of the current Event.

**Step 2:** PlagueEventCreation() checks the CurrentEventStatus to see that there is not currently an event going on.

**Step 3:** PlagueEventCreation() calls a function to instantiate a Plague Event based on the parameter returned in step one.

**Exceptions:**

**Step 1:** The CurrentEventStatus is already set, meaning that there is already another event going on.

**Step 2:** The initial trigger is ignored and no Plague Event gets instantiated.

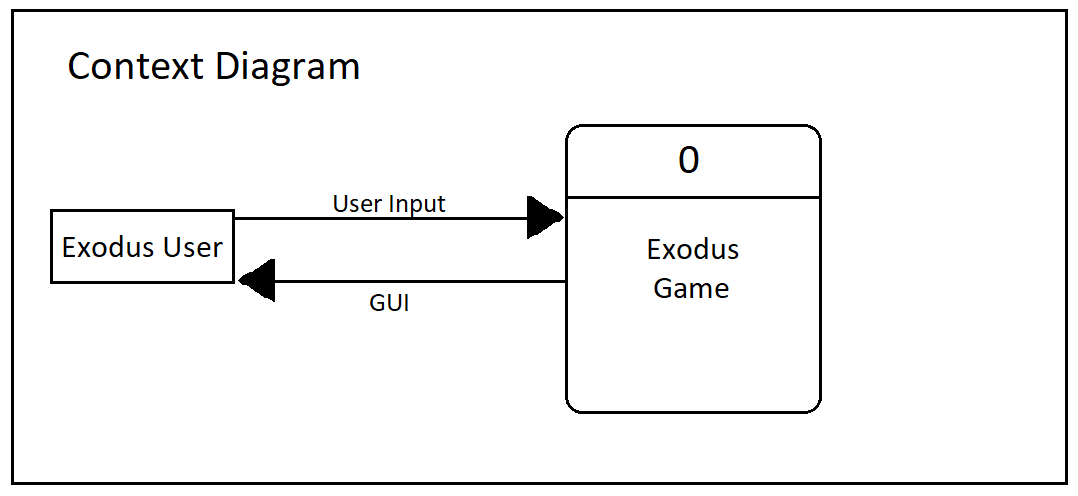
**Post conditions:** There is a Plague Event occurring.

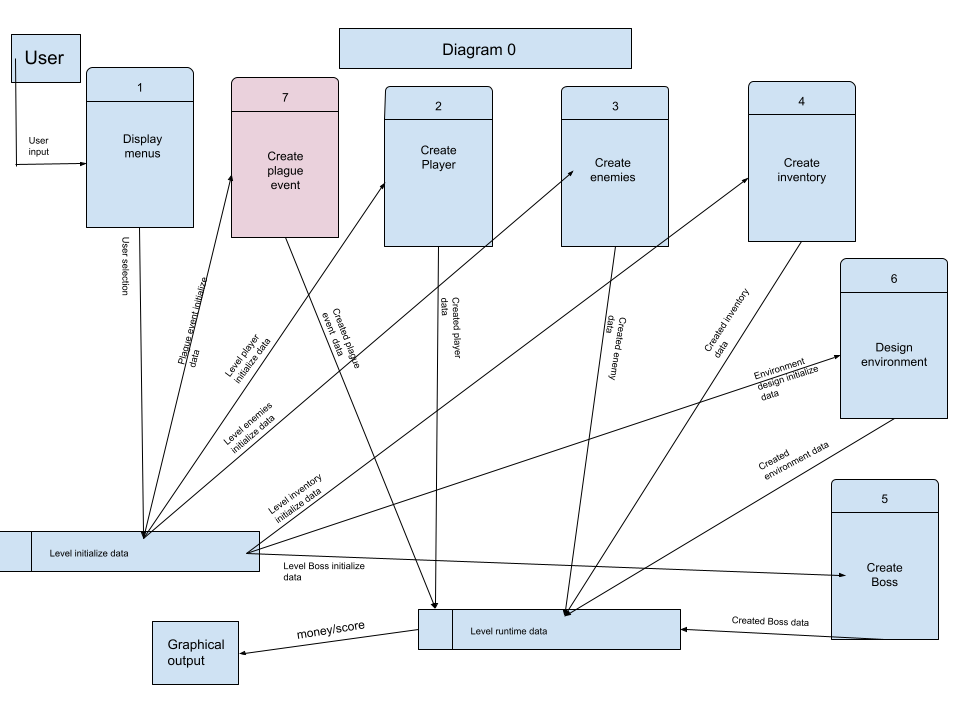
**Priority:** 2

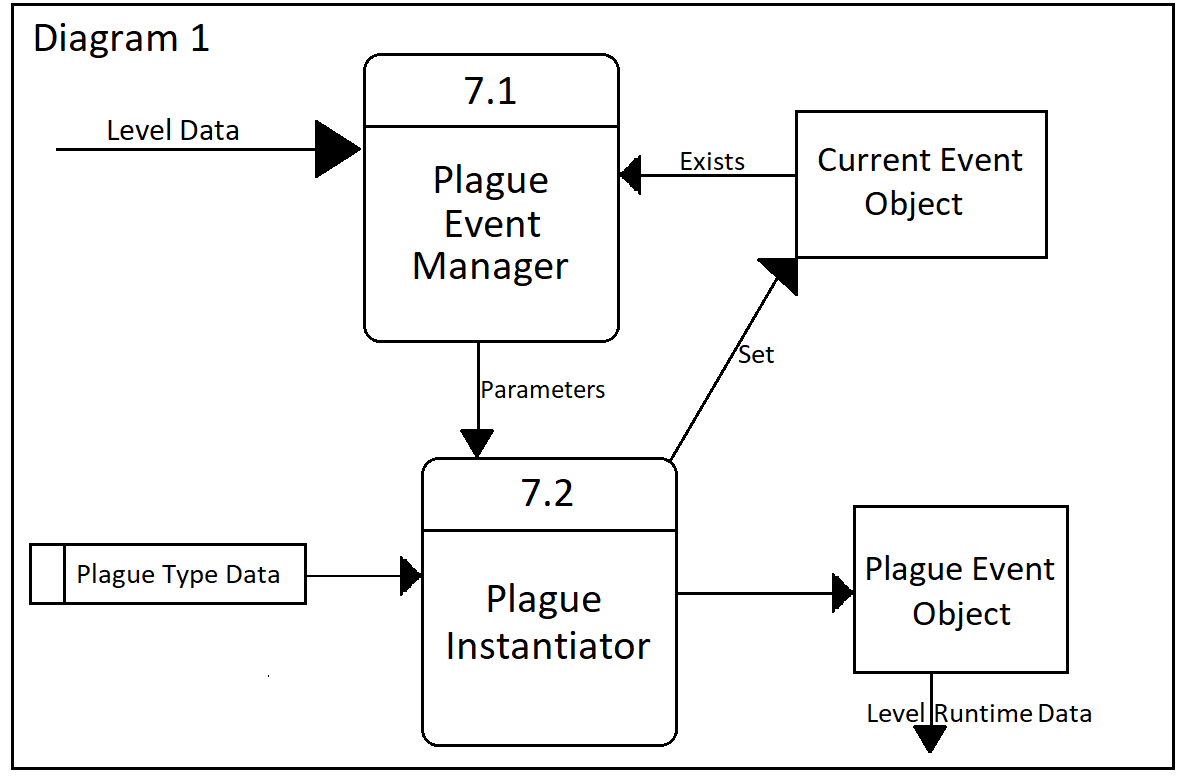
**ID:** C01

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

### Data Flow Diagrams







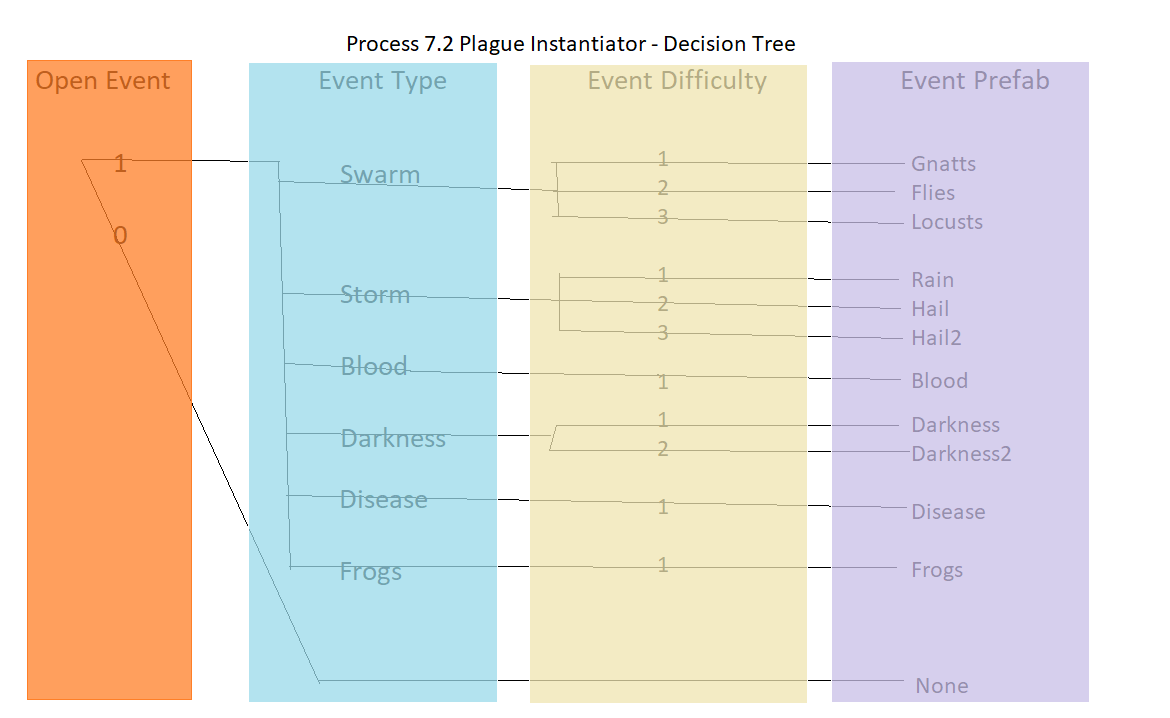
### Process Descriptions

7.1 Plague Event Manager:

This process will be waiting until it receives data from an OnTriggerEnter function saying that an eligible collider has been activated. Upon activation, this process will attempt to access the Current Event Object. If it is set to NULL, then the process will find relevant data about the Player’s skill-tree/inventory (which it received from the calling process) to come up with parameter data about the next Plague event which will be Instantiated by process 7.2. If the Current Event Object is already set to an event, the process will send a 0 parameter to process 7.2.

7.2 Plague Instantiator

This process has to use the parameters it was sent to look through a decision tree to decide which Plague Event Object will be instantiated, according to the decision tree below.



After Instantiating an event prefab, this process will find the Current Event Object, and set it to the game object which it just created. If there was no event instantiated, then the process will end without updating the Current Game Object.

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

To test the Plague Event feature, I manually insert 6 Colliders somewhere into the game, each one with a unique tag corresponding to the event types: swarm, storm, blood, darkness, disease, and frogs. There will be 3 categories of tests.

1. The game has just begun and there is no ongoing event.

For each collider: Run the player into the collider, confirm that the correct event prefab was instantiated.

* 1. For each collider: Run the player into the collider, confirm that no change has occurred to the active event, reset the game.

1. Set any Event Prefab as active, and start the game.

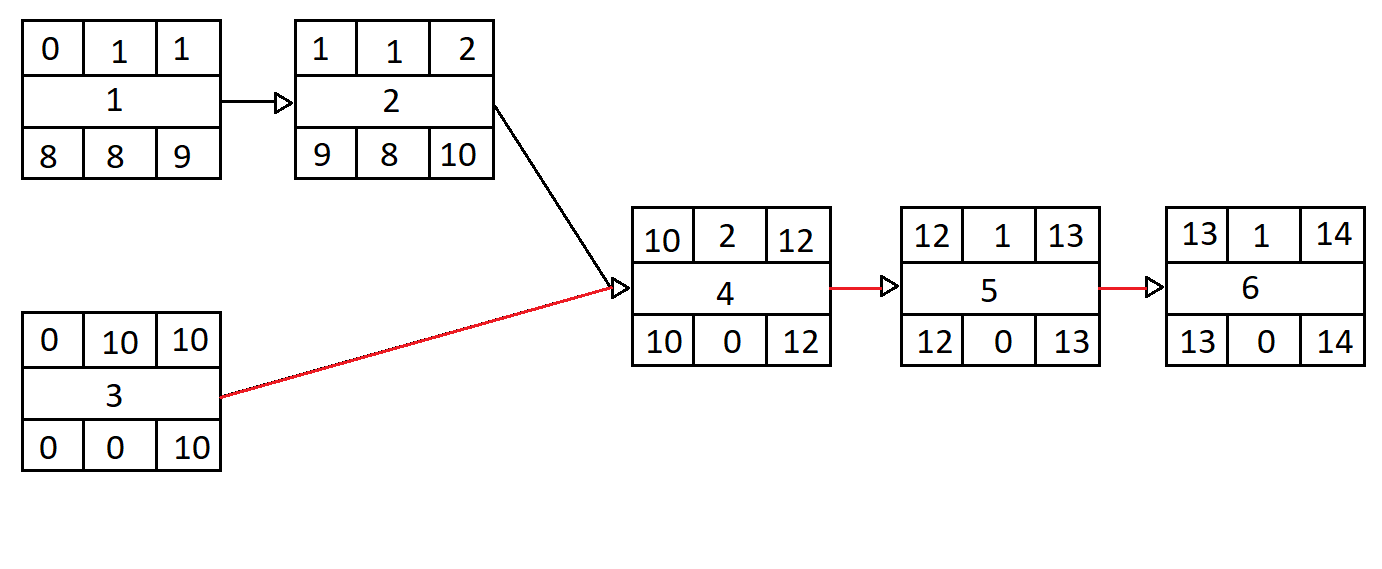
For each collider: Run the player into the collider, confirm that no change has occurred to the active event, reset the game.

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

### Work items

|  |  |  |
| --- | --- | --- |
| Task | Duration (Hours) | Predecessor Task(s) |
| 1. Create Plague Activation Colliders | 1 | - |
| 2. Write Plague Event Manager | 1 | 1 |
| 3. Create Plague Prefabs | 10 | - |
| 4. Write Plague Instantiator | 2 | 2, 3 |
| 5. Acceptance Testing | 1 | 4 |
| 6. Implement into Game | 1 | 5 |

### Pert diagram



### Gantt timeline

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  | 2,3 |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |  | 4 |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |