Name: Brandon Lunney   Total Marks ___/50

1. Brief introduction   ___/3

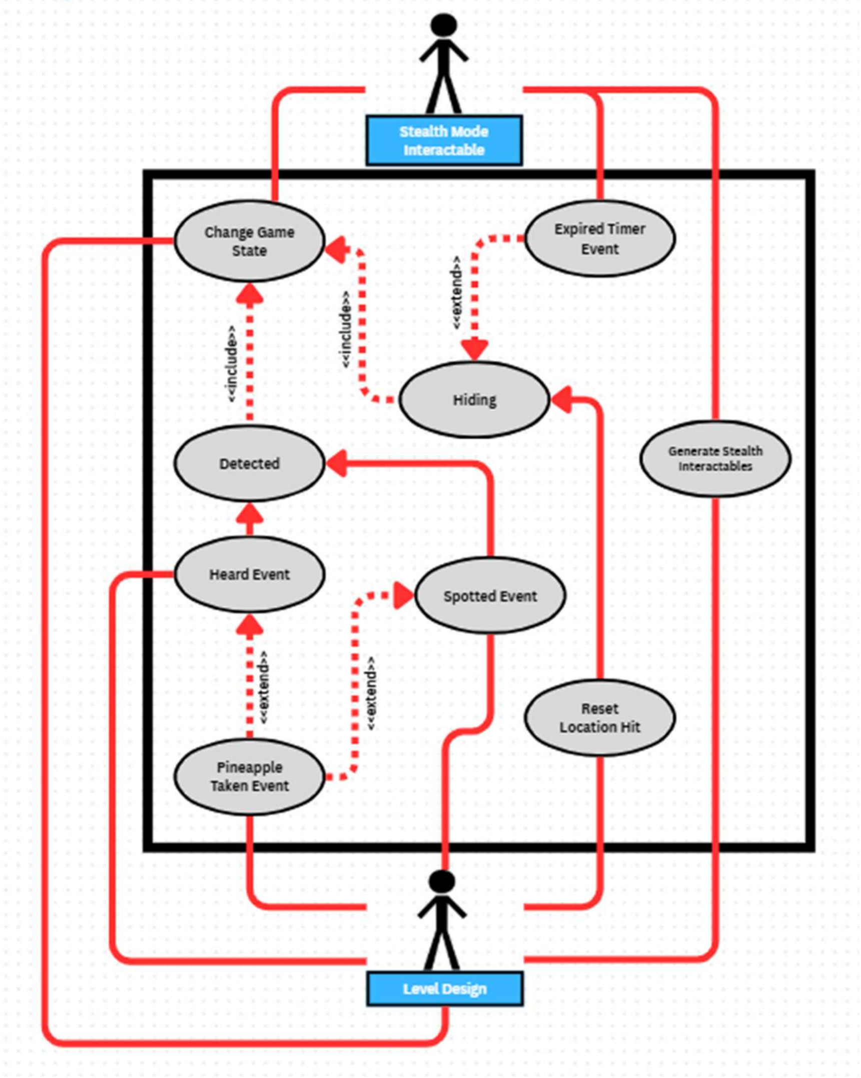
I will be responsible for implementing stealth game mechanics and interactions within the level design mechanism.

The system will rely on being within a specific "sight" range of a detection agent, making sound within "hearing" range of a detection unit, and either resetting state via checkpoint or elapsed time.

I will integrate the connections between systems within the level design for sound traps and the signal system to interact with detection agents within the play space. This will require partnership and coordination with other game systems being designed by other teammates.

2. Use case diagram with scenario   ___/14

Use Case Diagram



Scenario Concept

Scenario 1: **Change Game State to Detected**

Summary: The Game State in the level design is changed to Detected, changing the enemies in range to alert status.

Actors:

- Stealth Mode Interactable
- Level Design

Preconditions:

- Detected event triggered by stealth interactable in the Level Design.

Basic sequence:

1. Detected trigger hit on stealth interactable in level design.
2. Update State Data.
3. If Hiding, send state change to AI System to switch all enemies to Oblivious.
4. Reset timer.
5. Determine alerted enemies.
6. Send effected enemy states to Level Designer.

Exceptions:

2. If pineapple is taken, change all enemy states to alert.

Post conditions: Mode is correct and enemies are set to alert status.

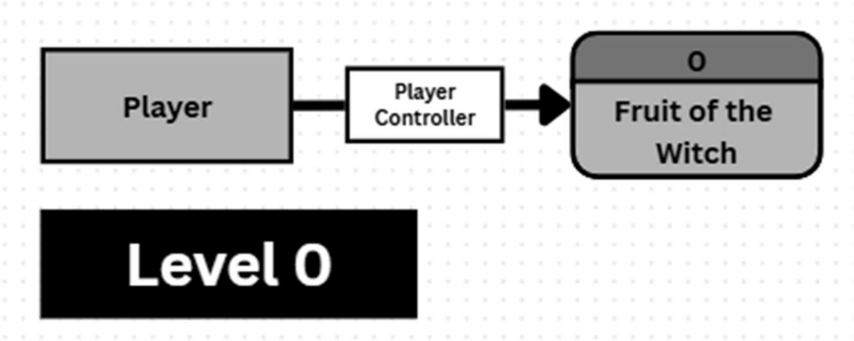
Priority: 1 [Must Have]

ID: GM1

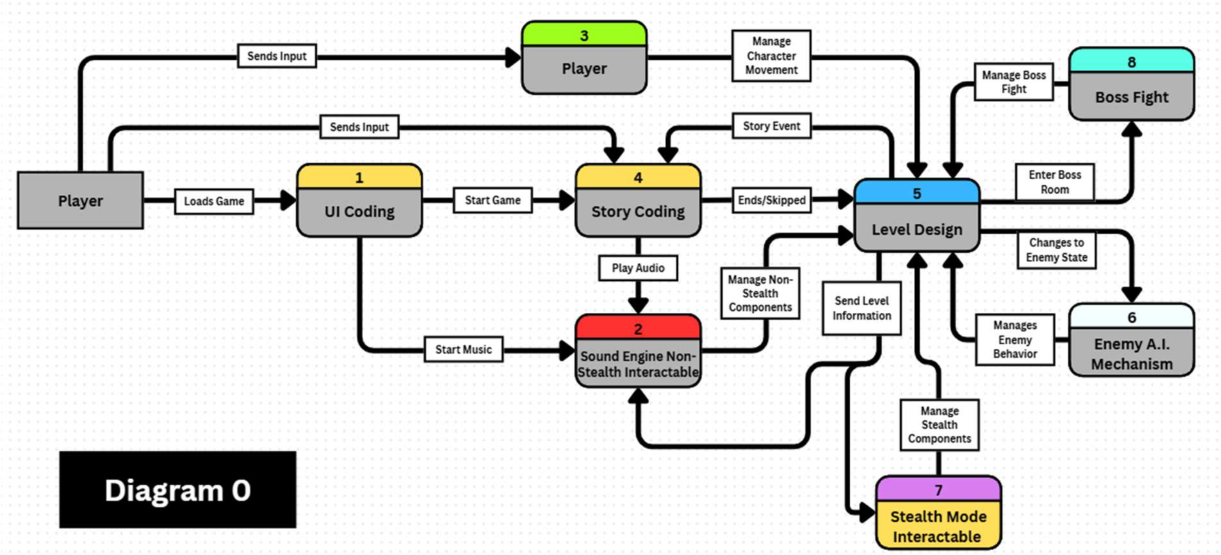
3. Data Flow diagram(s) from Level 0 to process description for your feature

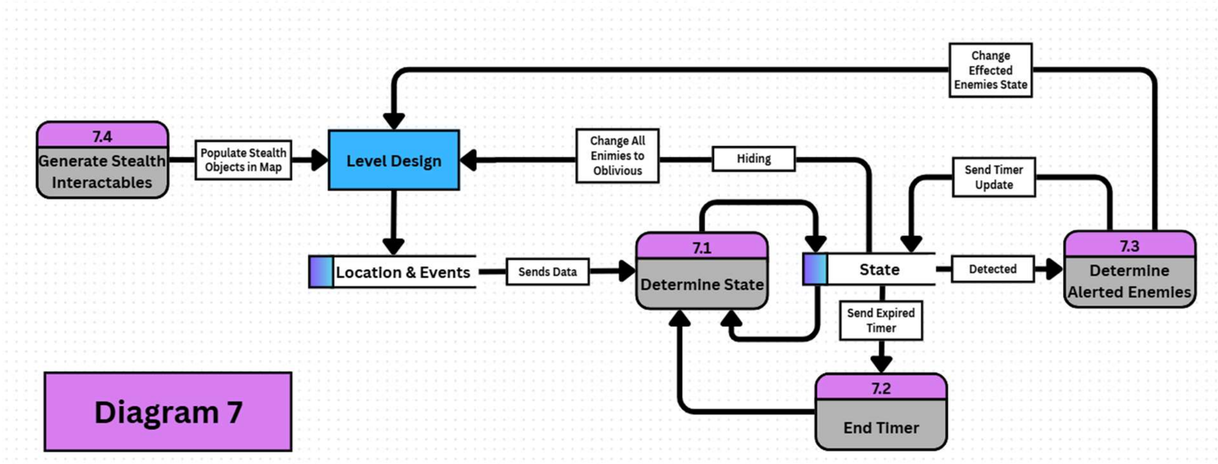
___/14

Context Diagram



Data Flow Diagrams




Change Effected Enemies State
7.4
Generate Stealth Interactables
Populate Stealth Objects in Map
Level Design
Change All Enimies to Oblivious
Hiding
Send Timer Update
Location & Events
Sends Data
7.1
Determine State
State
Detected
7.3
Determine Alerted Enemies
Send Expired Timer
7.2
End Timer
Diagram 7

Process Descriptions

Determine State*:

```
Pull Location & Event data.

IF timer expired OR event data is “hiding” THEN
    Send hiding to state data.
    Send signal to AI SYSTEM to change all enemies to oblivious.
END IF

ELSE IF event data is “detected” THEN
    Send detected state to data.
    Transition state to Determine Altered Enemies.
    Pass Location & Event data.
END  IF
```

Determine Altered Enemies*:

```
SET timer to default time.

FOR enemies IN location data
    IF location is EQUAL to alerted range, THEN
        Place enemy in alerted enemy list.
    END IF

Send altered enemy list to level deisgn
```

Determine End Timer*:

```
Pull timer data from State.

IF timer expired THEN
    Send timer expired to Determine State.
    Transition state to Determine State.
END IF

ELSE IF timer data is reset THEN
    Transition state to Determine State.
END  IF
```

Determine Generate Stealth Interactables*:

```
Pull data from Location & Events
```

```
FOR stealth interactable IN interactable list
    IF stealth interactable EQUAL to requested THEN
        Place enemy in interactable list with location data
    END IF

Send interactable list to level design
```

4. Acceptance Tests __/9

Stealth Management System

This system has three states, two data locations, an internal timer, and three input streams. As such the tests will use these locations as attack vectors for tests to be executed 1,000 times to test validity. If a failure state is found, it will be logged and the test process will terminate for that run.

Testing Structural Validity

These are the tests that will be run:

- Ensure states do not lead to dead ends.
- Validate that the timer behaves and triggers correctly.
- Send junk data to the data locations to ensure proper data sanitization.
- Ensure bad data does not break the state flow if unexpected data is pulled.

Example statistics for generated maps

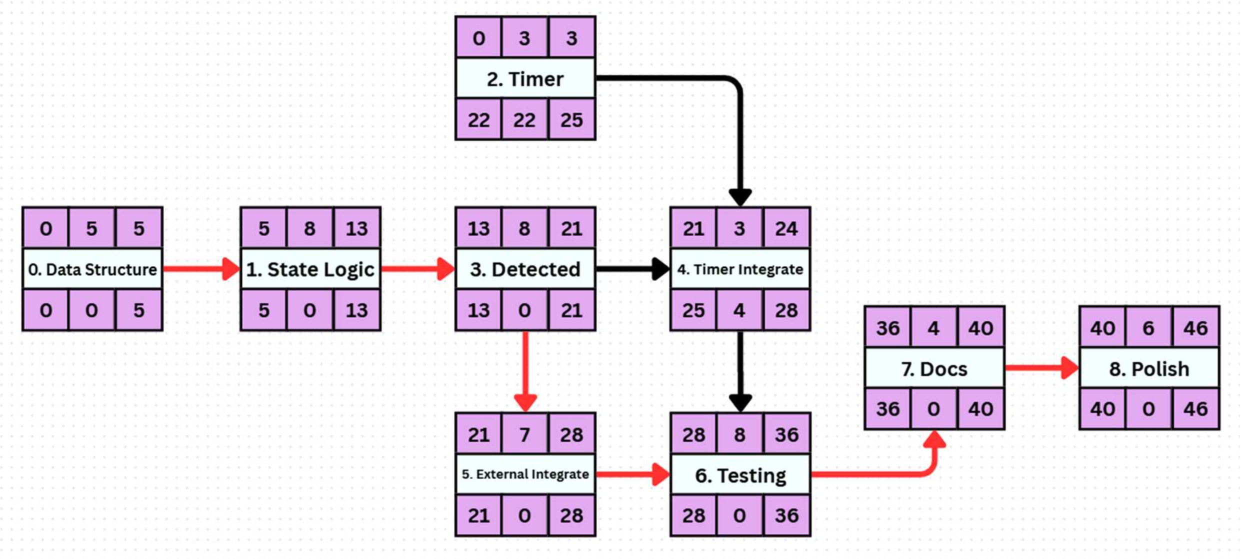
Statistic	Value	Pass Tests?	Notes
Timer Failures	FAST COMMAND	T	Sent repeated commands for reset, end, and start in random sequences. Behavior as expected.
Start event connections	T	T	Tested starting at different states to see if a failure could be reached
Send random junk as input to data locations	RANDOM INPUT	T	Flood the data locations with junk input. No failures detected.
Pull junk data from data	JUNK DATA	F	Had states pull data from poisoned data set to ensure second catch for data corruption, failure detected in 7.3

5. Timeline ___/10

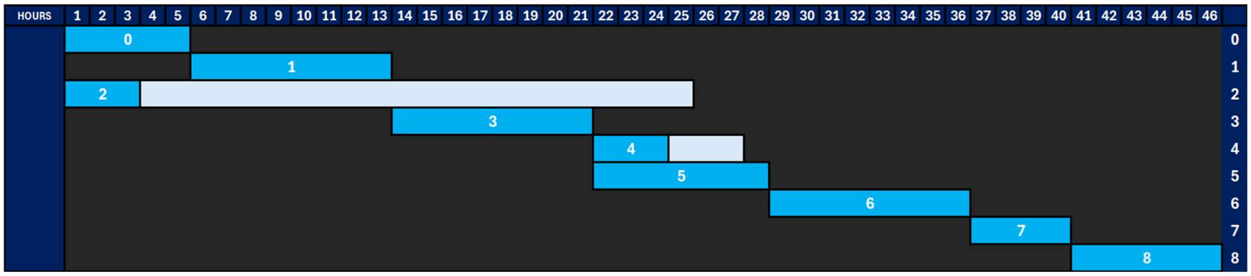
Work items

Task	Duration (HRS)	Predecessor Task(s)
0. Stealth Interactable & Data Structure	5	-
1. Determine State Logic	8	0
2. Background Timer	3	-
3. Determine Alerted Enemies Process	8	1
4. Timer Integration	3	2, 3
5. External Input System Integration	7	3
6. Testing	8	0, 1, 2, 3
7. Documentation	4	0, 1, 2, 3
8. Polish	6	7

Pert Diagram



Gantt Timeline


HOURS