Name: Michael Crapse Mark _____________________/50

[**Instructions**: Remove everything that is not a heading below and fill in with your own diagrams, etc.]

1. Brief introduction __/3

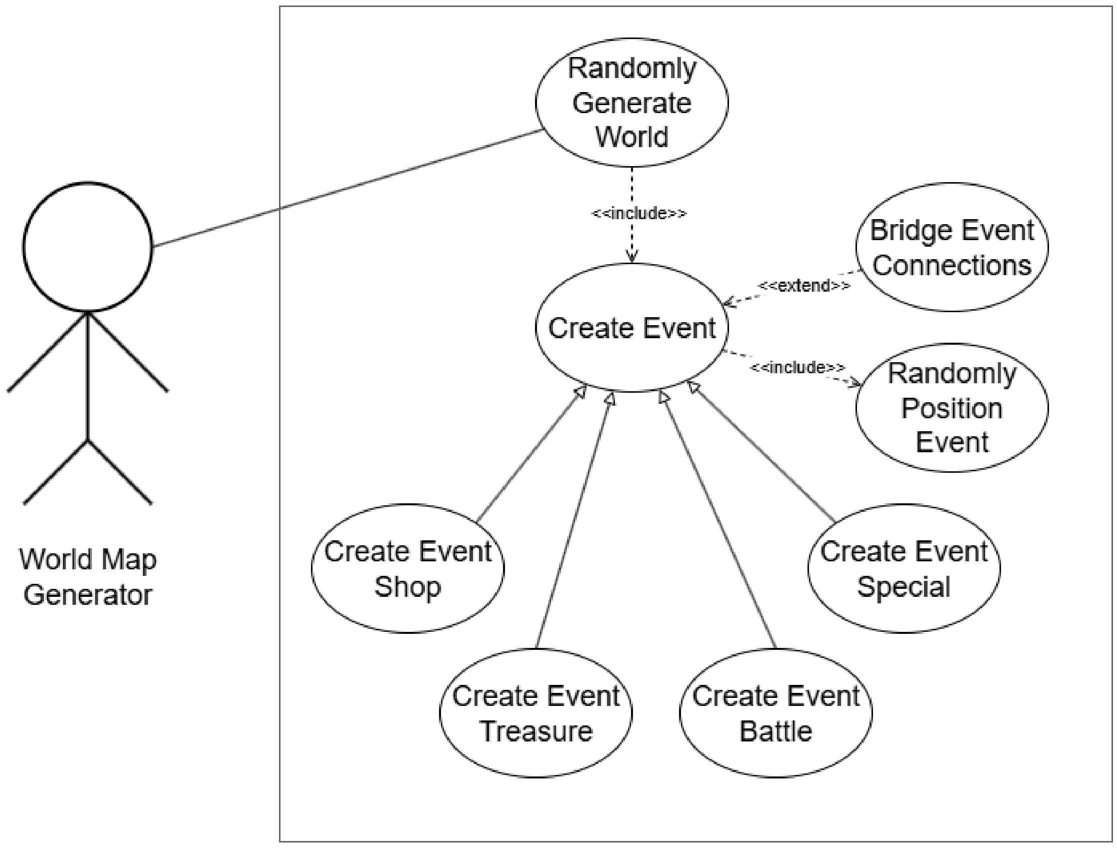
I am implementing the world map, its random generation, and transitions to and from the events in the map.

I will need to connect the work done for the events: treasures, shops, specials, and battles. I will also be working to ensure the user interface menus work with the map.

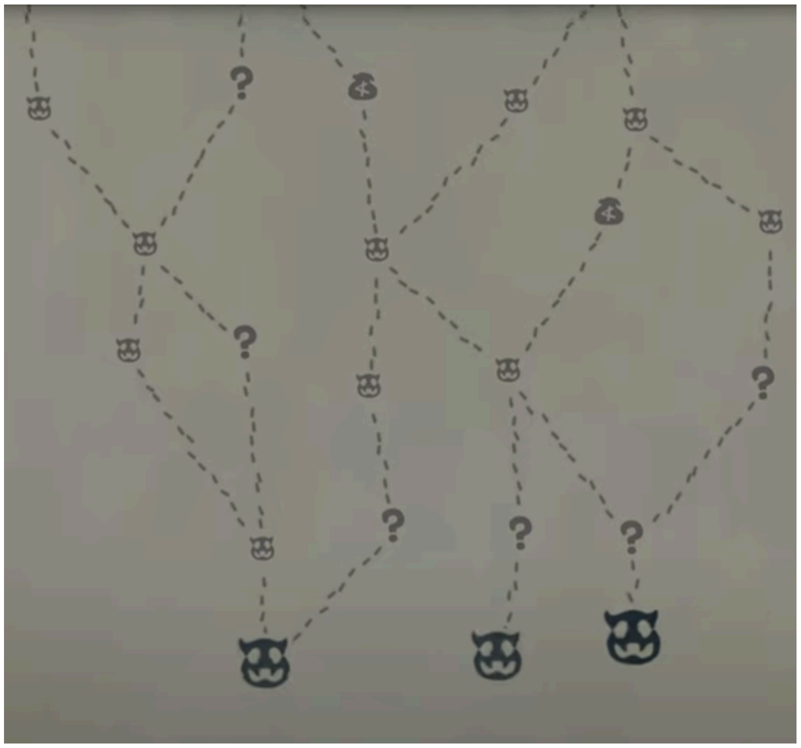
Additionally, I will program the randomization and generation of the world map. This includes the placement of events, the event connections, and the type of events. After map generation, the map will be rendered with its associated events and connections.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenario Concept



Scenarios

Name: Randomly Generate World

Summary: The World Map Generator creates events to generate the map

Actors: World Map Generator

Preconditions: Player has entered the world map for the first time

Basic sequence:

- **Step 1:** Create rows of events from first to final events
- **Step 2:** Randomize event types
- **Step 3:** Randomly position events horizontally
- **Step 4:** Connect event paths to preceding row events

Exceptions:

- **Step 4:** Upon creation of first events: do not connect path to preceding row (there is no preceding row to connect paths to)

Post conditions: World map is generated

Priority: 2

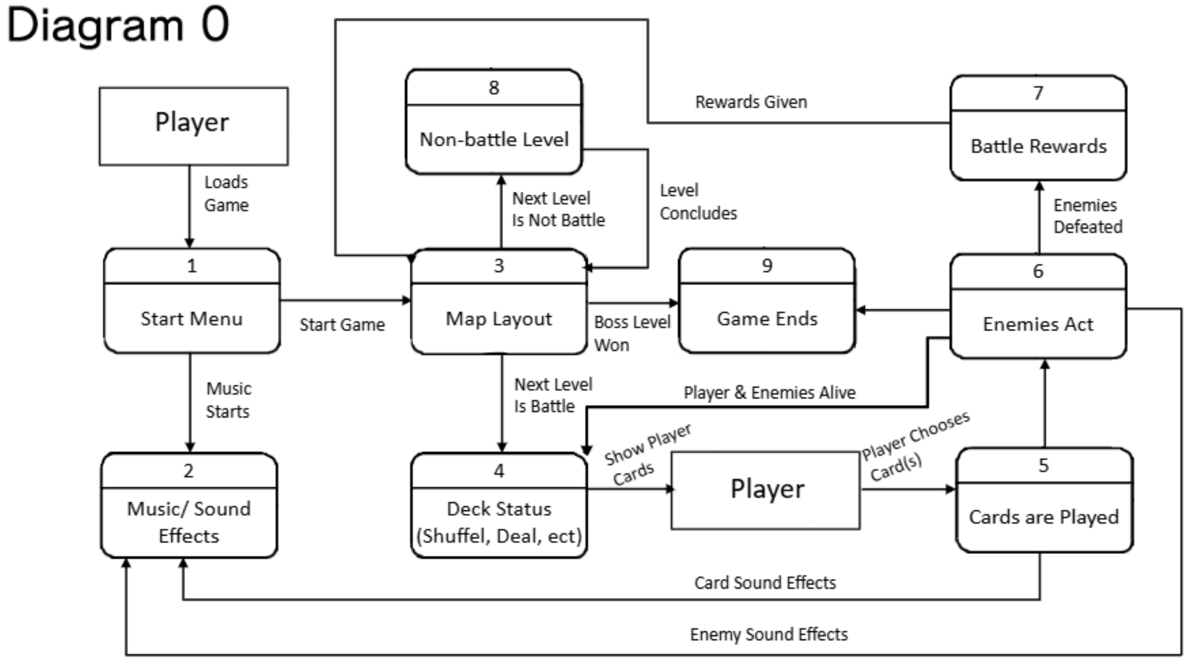
ID: GM1

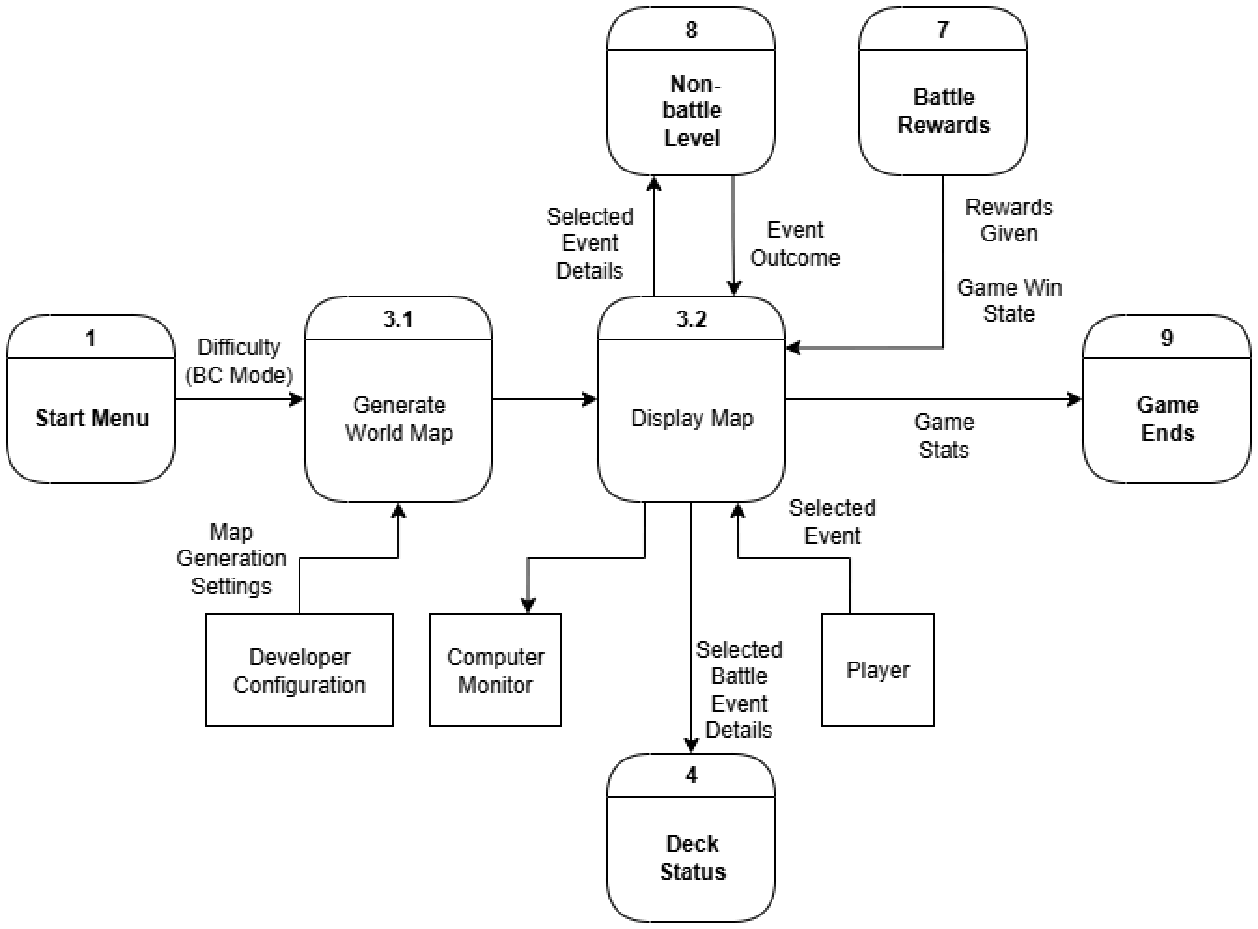
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

3. Data Flow diagram(s) from Level 0 to process description for your feature ______14

Data Flow Diagrams

Diagram 0





Process Descriptions

```
Generate World Map*:
    FOR map row IN max map rows
        FOR event IN map row events
            Create randomize event type
            Position event
            Connect event to previous row event
        END FOR
    END FOR

    Transition to display map state
    Pass the generated world map

Display Map*:
    IF player select battle event
        Prepare the battle deck and hand
        Transition game state to battle event
        Pass information about the battle
    ELSE
        Transition game state to a non-battle event
        Pass information about the event type
        Apply event results
    END IF

    IF transitioned from battle state
        Apply event results
        IF player won final event
            Transition to game ends state
            Pass statistics and game results
        END IF
    END IF

    IF transitioned from any event type
        Disable event
        Enable connected events in the next row
    END IF
```

4. Acceptance Tests ______9

World Map Generator

Due to the random nature of the world map generator, the following tests will be executed 1,000 times per single test run. Tests will also end upon the first occurrence of a failure case. This assures that most randomness is accounted for.

1. **Structural Validity**
   The following will be tested to ensure structural validity:
   - Assure every event has a path to the final event. (No dead ends except final)
   - Every event (except start events) must connect to a previous row.
   - The map must remain acyclic, since it's meant to represent forward-only progression.
   - Each event type must appear at least once (or a specified number of times).
   - The first and final row of events must be battle events

Example statistics for generated maps

Statistic	Value	Pass Tests?	Notes
Dead ends	1	T	The final event should be the only dead end in the map. Any map not having exactly 1 dead end in it is a failed test.
Start event connections	0	T	
Every event after start connects to a previous row	T	T	
Map is cyclic	T	F	Map must be acyclic
At least one of every event type	T	T	
Final non-battle events	0	T	
Starting non-battle events	2	F	

5. Timeline _______/10

[Figure out the tasks required to complete your feature]

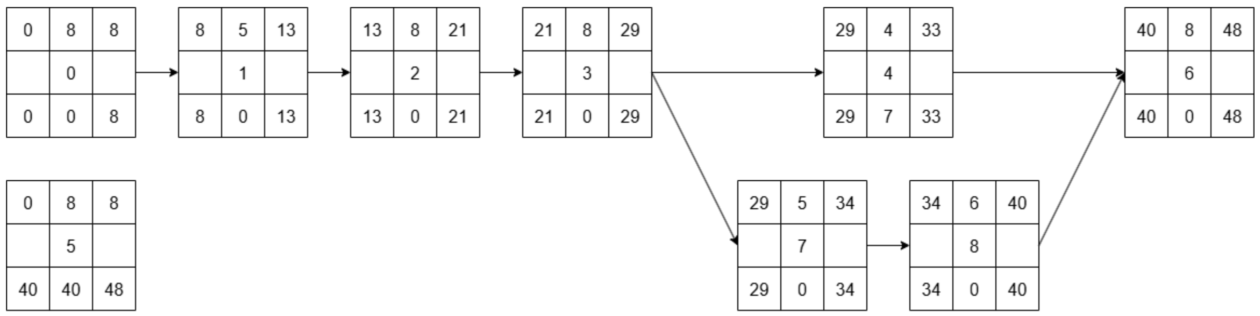
Example:

Work items

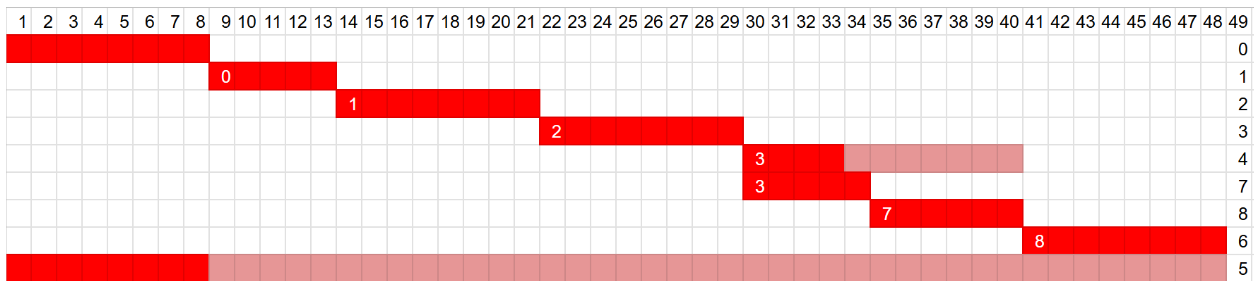
Task	Duration (Hours)	Predecessor Task(s)
0. Map layout	8	-
1. Event Interactions	5	0
2. Surrounding Map GUI	8	1

3. Map Generation	8	2
4. Testing	4	3
5. Artwork	8	-
6. Polish	8	4, 8
7. Documentation	5	1, 3
8. Integration with Project	6	7

Pert diagram



Gantt timeline


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 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
0
1
2
3
4
7
8
6
5
0
1
2
3
3
7
8