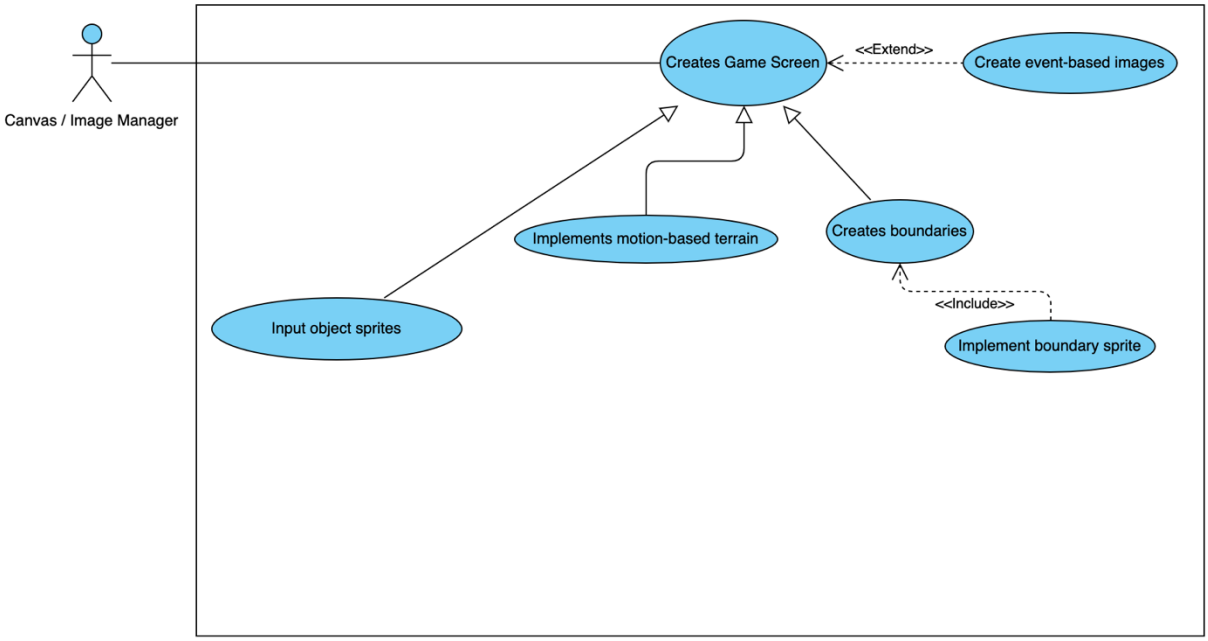
Name **Ronnie Keating** ___ Mark _____/50

1. Brief introduction __/3

My feature for the game is to create sprites that will be implemented into placeholders such as borders (trees and rocks), terrain, and NPC's (nonplayable characters). Some of these sprites, such as water and NPC's, will need motion implemented as well.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

Name: Input object sprites

Summary: Objects such as buildings, rocks, and trees will be implemented throughout the game layout.

Actors: Image manager

Preconditions: Game screen has been created.

Basic sequence:

Step 1: Objects will be placed on the screen.

Exceptions:

Step 1: Objects will have interactions, but also will act as a barrier.

Step 2: They generally cannot move.

Post conditions: Objects stay in position.

Priority: 3*

ID: C01

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

Name: Implement motion-based terrain
Summary: Some terrain can have motion-based movements to make the game look more natural.
Actors: Image manager
Preconditions: Game screen has been created.
Basic sequence:

Step 1: The terrain sprite changes between images every few milliseconds to make it look like it is moving.

Exceptions:

Step 1: Terrain sequences must be consistent.

Post conditions: The terrain looks like motion, such as water.
Priority: 3*
ID: C02
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: Creates boundaries
Summary: Implements boundaries for which any moving character can go. This will include mini boundaries for NPC.
Actors: Image manager
Preconditions: Game screen has been created.
Basic sequence:

Step 1: The terrain is form.
Step 2: Boundary sprites are placed in their positions (could be no sprite).
Step 3: Sprites perform functionality of restricting characters.

Exceptions:

Step 1: Boundaries must be in the same position the entire game, unless to open a new area.

Post conditions: Boundaries stay in their placed position.
Priority: 1*
ID: C03
*The priorities are 1 = must have, 2 = essential, 3 = nice to have.

Name: Create event-based images
Summary: Scenes such as cut and end scenes will need to managed and presented. They only appear every once in a while, so that is why it is an extend object.
Actors: Image manager
Preconditions: Game screen has been created.
Basic sequence:

Step 1: Something triggers an event scene.
Step 2: Objects, sprites, and background are presented on the screen.
Step 3: Scene plays through and returns to player mode.

Exceptions:

Step 1: Event scenes have to be triggered, not random.

Step 2: Finish entire scene before returning to player mode.

Post conditions: Scene is no longer presented.

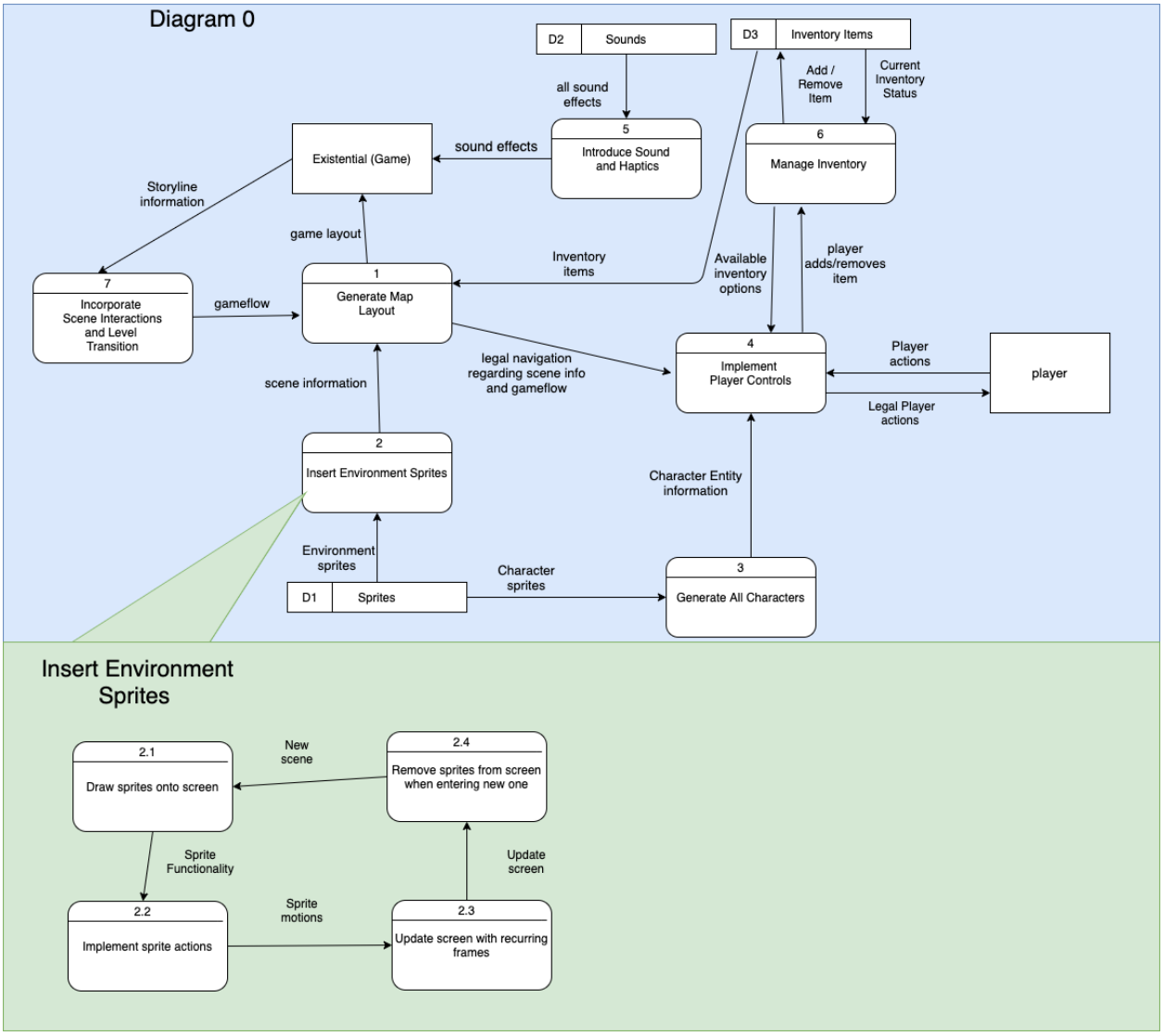
Priority: 2*

ID: C04

*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



Process Descriptions

Draw sprites onto screen:
    WHILE in a current scene
        IF sprite has not been drawn, draw in sprite
    END WHILE
Implement sprite actions:
    IF sprite is a boundary sprite
        keep in position until end of scene
    IF sprite is a motion-based sprite,
        check function for next frame
    IF sprite is not a boundary sprite, but has a limited boundary area
        Check area size
Update screen with recurring frames:
    WHILE in current scene
        IF sprite has motion and on first frame
            Fetch next frame
        ELSE IF sprite is next frame
            Fetch first frame
    END WHILE
Remove sprite from screen when entering new one (scene):
    IF entering into new scene
        WHILE sprites are still on screen
            IF sprite is still printed
                Remove sprite
        END WHILE

4. Acceptance Tests _____9

Test terrain motion:

- Increase speed between each frame
- Keep terrain moving for incremental amounts of time
- Move user closely around terrain

Test sprites:

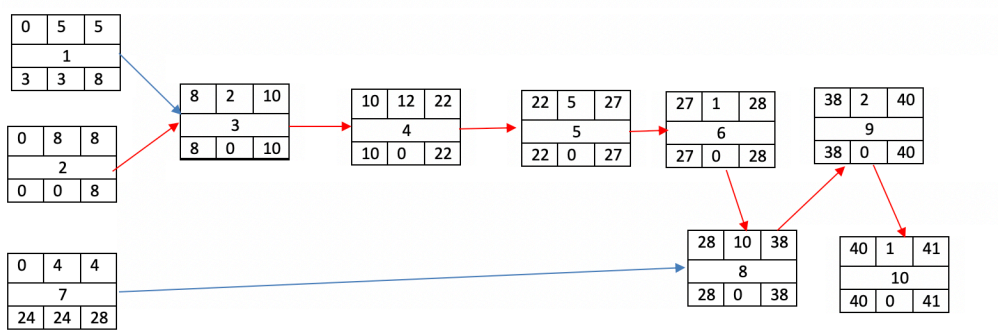
- Make sure sprites follow the correct path and user input
- Increase speed of character direction to see sprite follows
- Place boundary sprites in position and make sure they still function as boundaries
- Run into boundary at different speeds

5. Timeline _____/10

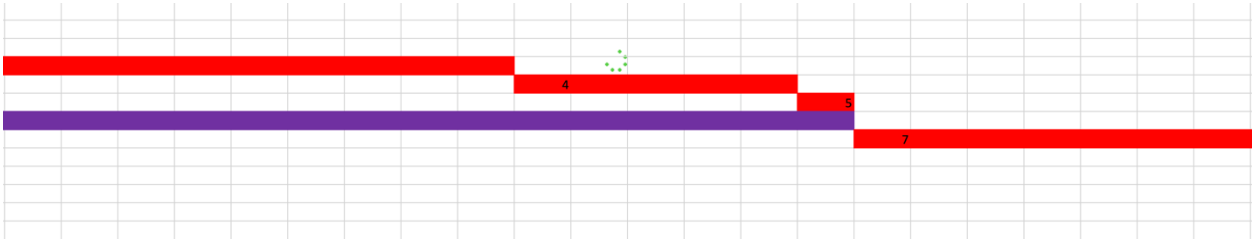
Work items

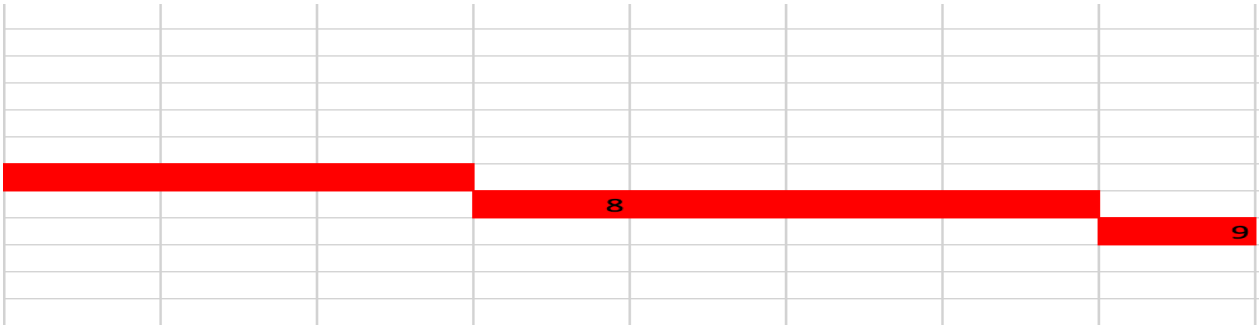
Task	Duration (Hours)	Predecessor Task(s)
1. Design Terrain	5	-
2. Design Sprites	8	-
3. Test Sprite Layout	2	2
4. Implement all sprites and terrain	12	3
5. Test sprites and motion	4	4
6. Document sprites	1	5
7. Design level	4	-
8. Program level	10	7
9. Test level	4	8
10. Document level	1	9

Pert diagram



Gantt timeline


4
5
7


8
9