Name: Scott Anderson Mark _____/50

1. Brief introduction _/3

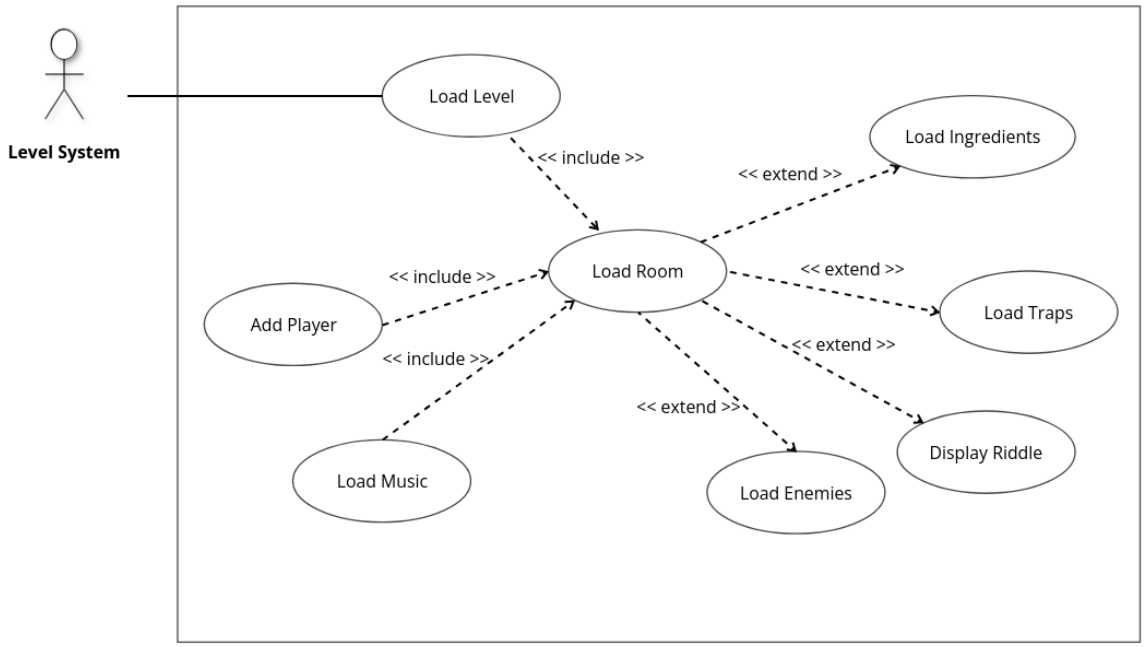
My feature for Pastafarian's Delight is the puzzle/riddle design as well as the level design.

My job is to design the puzzles/riddles that are given to the player. There will be a bank of options for the different types of pasta dishes and the ingredients that go into them. I will come up with riddles for the player to decipher in order to determine the correct ingredient for that room.

Additionally, I will be designing the levels. This will include the layout of the room, where traps are located, and the difficulty of each room. The difficulty will increase based on which level you are on (1, 2, or 3).

2. Use case diagram with scenario _14

Use Case Diagram



Scenarios

Name: Load entered room

Summary: When a room is entered the Level System triggers the Load Room use case. Requiring the room, enemies, traps, ingredients, and the music to be loaded. The Player

is added to the room at the entrance and the riddle is displayed. The riddle for the room is dependent on the pasta dish that is chosen at the start of the game.

Actors: Level System

Preconditions: Pasta dish has been chosen, determines which riddles are to be displayed.

Basic sequence:

Step 1: Load room

Step 2: Load enemies

Step 3: Load traps

Step 4: Load ingredients

Step 5: Add player

Step 6: Display riddle for the room that has been entered

Step 7: Load music

Exceptions:

Step 6: Riddle for the ingredient is based on the pasta dish to be created

Post conditions: The player is now ready to play this room.

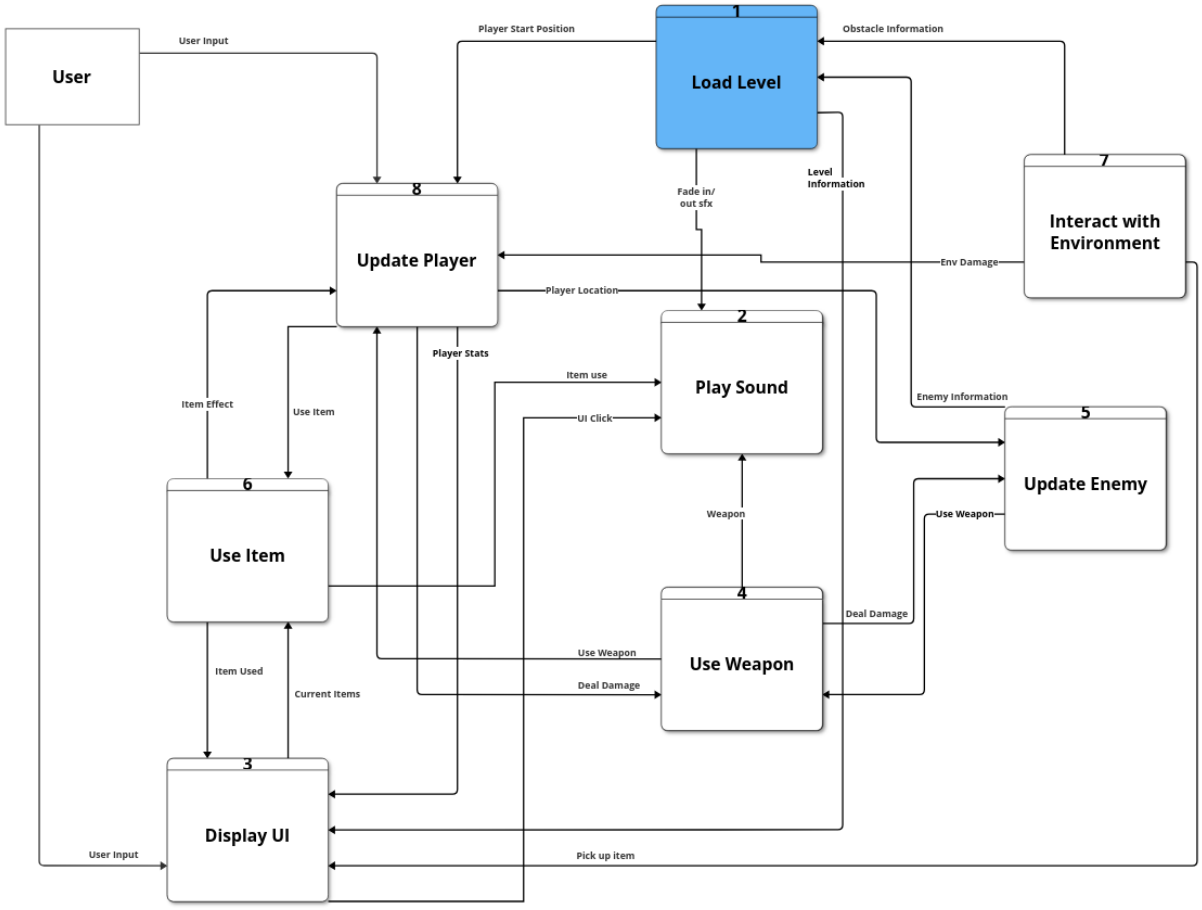
Priority: 1

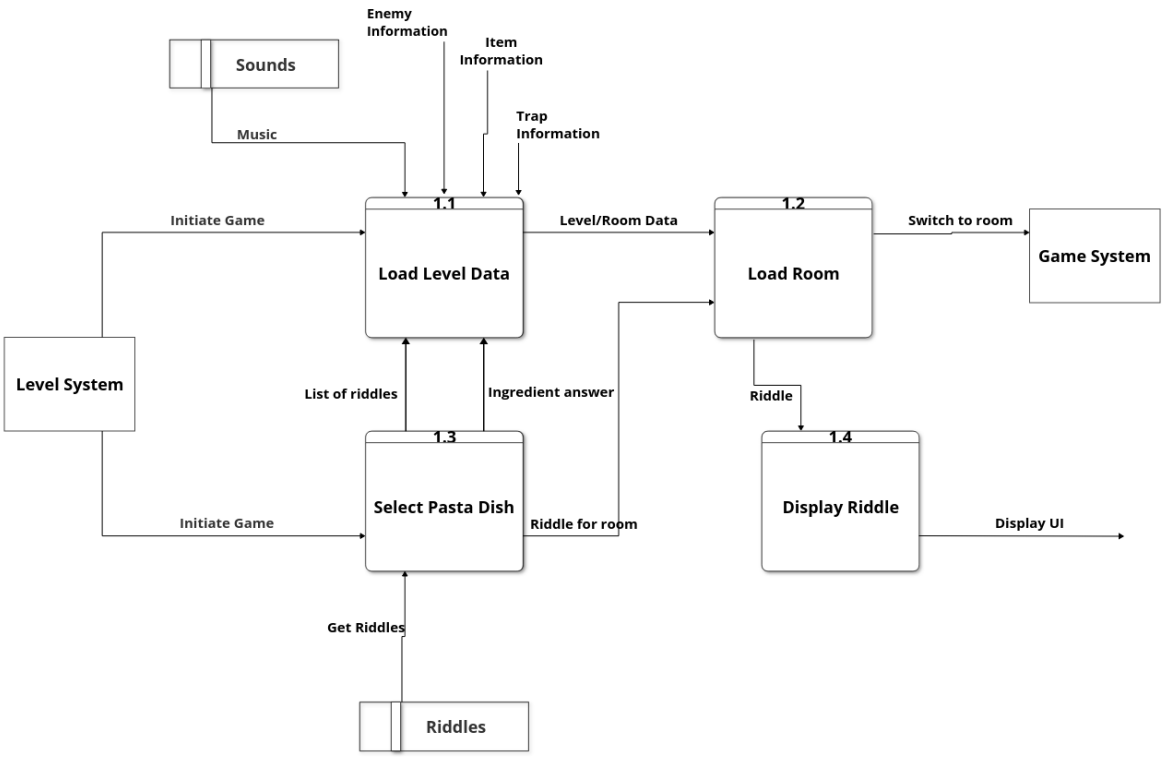
ID: C01

*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

Process description for Load Room (1.1)

Get puzzle information

Save riddle answer

Assign corresponding ingredient riddle and ingredient to each room

Send data to room loading function

Get items

Select 4 ingredients that are not one of the correct ingredients

Select power up for end of room reward

Send data to room loading function

Get enemies

Select enemies for room

Assign enemy starting positions

Send data to room loading function

Get traps

Select traps for room

Assign trap location

Send data to room loading function

Get sound

Load calm music

Load action music

4. Acceptance Tests _____9

Within each room, there will be several default locations for where enemies will spawn and where traps can be placed.

For traps, we will want to make sure that we do not place more than one trap in a specific location. Essentially, we want to make sure that we do not have any race conditions that allows for multiple traps to be in one location.

Similarly with enemies, we will have multiple types of enemies, and we will want to ensure that a room is occupied by different enemy types. Also, that they are not overlapping in their spawning positions.

The same ideas can be applied to the ingredients that are placed in the room. They must all be different and include 1 that is the correct ingredient for the pasta puzzle. And they must not be overlapping in their positions.

5. Timeline _____/10

[Figure out the tasks required to complete your feature]

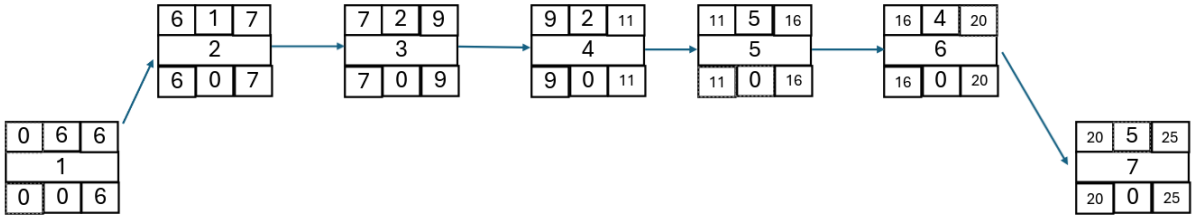
Example:

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Program main level loading system	6	-
2. Design Pasta dishes	1	-
3. Design riddles for ingredients	2	2
4. Implement Pasta puzzle/riddles	2	2, 3
5. Program puzzle picker at start of level	5	4
6. Program puzzle validation system	4	5
7. Testing	5	6

8.		

Pert diagram



Gantt timeline