Name Alex Parenti_____ Mark _____/50

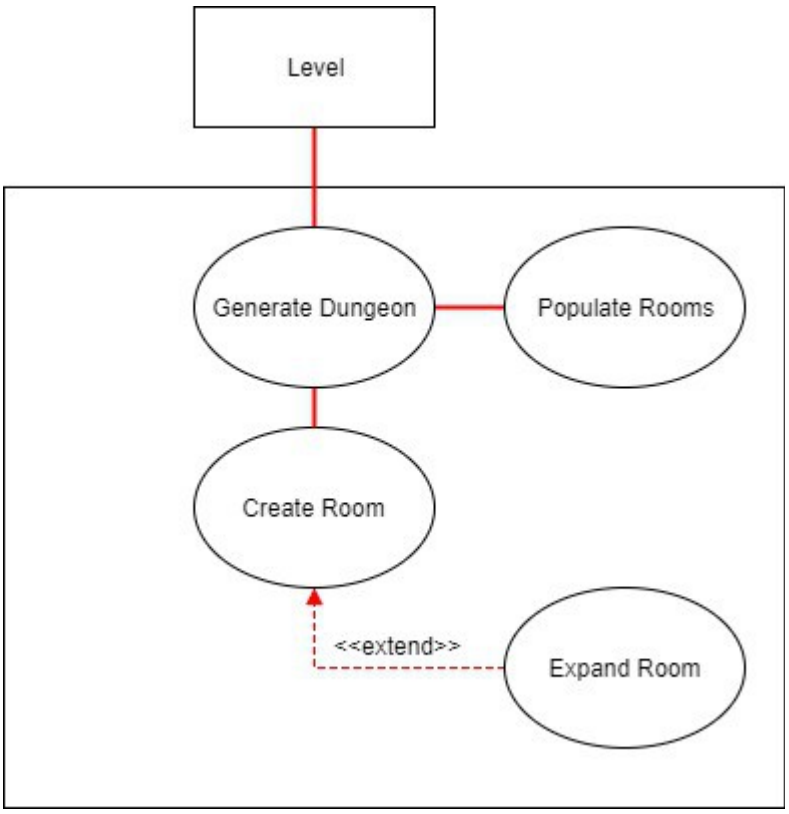
1. Brief introduction __/3

Procedural Dungeon Generation

On levels loading, my code will be responsible for procedurally generating a dungeon layout and populating the dungeon layout with items and enemies provided by the other teams, as well as spawning the player object at a start location, and creating an exit for level completion.

2. Use case diagram with scenario __14

Use Case Diagrams



Scenarios

Name: Generate Dungeon

Summary: The level prompts the dungeon generator to procedurally generate a dungeon for the player to play through

Actors: Level

Preconditions: Seed has been set, level difficulty specified, player object assigned

Basic sequence:

*Expand – means to generate a connecting room with its own exits, adding those exits to a list for further expansion.

Step 1: Generate starting room and exits

Step 2: Expand dungeon through one valid exit, add other exits to a list.

Step 3: Continue expanding dungeon through a single new valid exit until randomly chosen distance to middle is reached.

Step 4: Generate middle room with exits.

Step 5: Continue expanding dungeon through a single new valid exit until randomly chosen distance from middle to end is reached.

Step 6: Continue expanding dungeon from all other exits added to original list.

Step 7: choose and delete some percentage of superfluous rooms that lead to the same end room.

Step 8: Generate dungeon room visuals (doors, decorations, etc.)

Step 9: populate rooms with interacting items.

Step 10: populate rooms with enemies.

Step 11: spawn player in start room.

Exceptions:

Step 1: Attempt to populate rooms with items but no item object has been assigned to dungeon generator: no items will be generated. Debug message displayed “No items assigned to dungeon generator”.

Step 2: Attempt to populate rooms with enemies but no enemy object has been assigned to dungeon generator: no enemies will be generated. Debug message displayed “Boring dungeon. No enemies assigned to dungeon generator”.

Post conditions: Dungeon has been generated. The level is ready for a player to play.

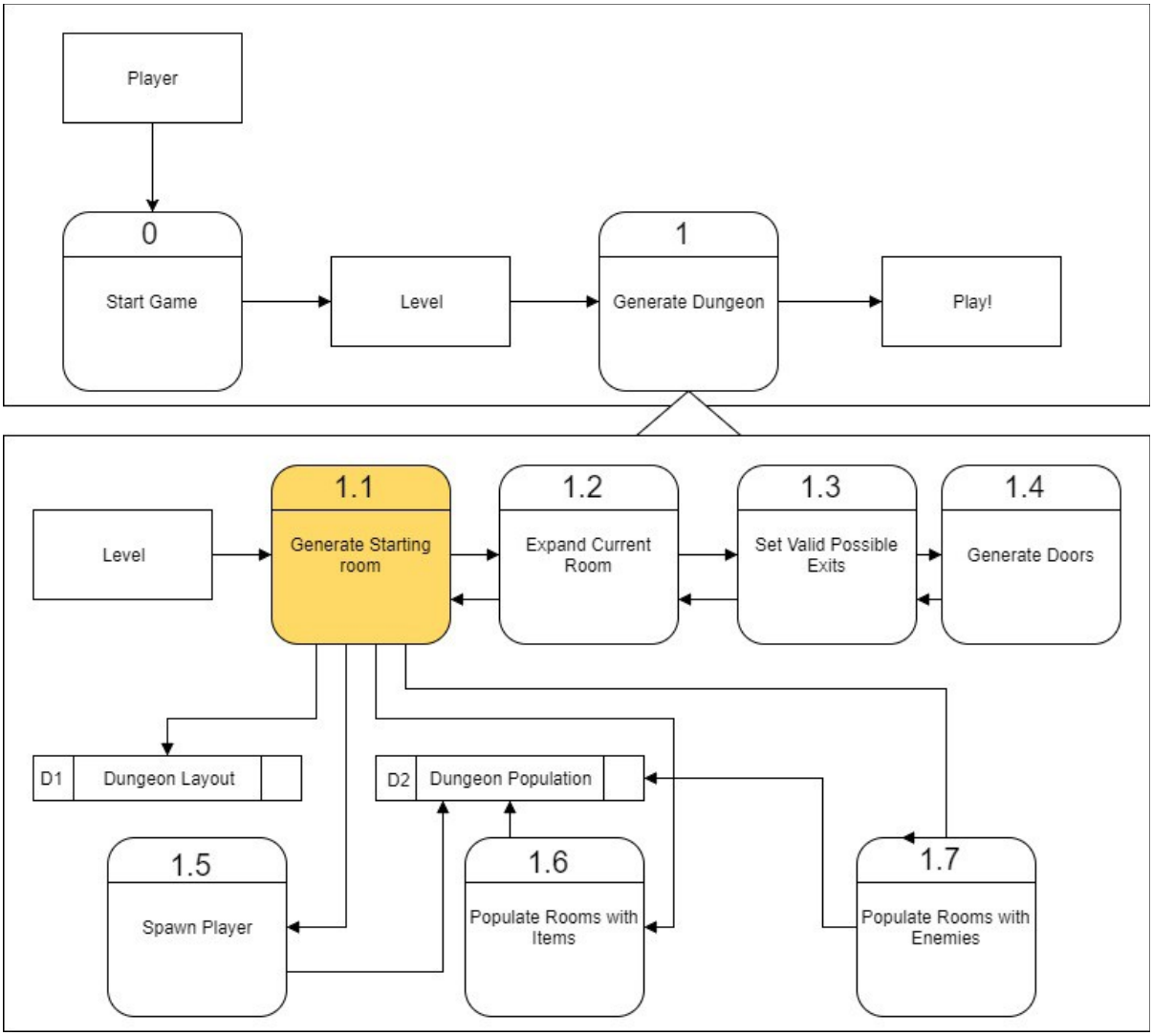
Priority: 2*

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

Generate starting room of a size within a specified range.

Choose points along the top and right side of the starting room (as main path will only expand up and/or to the right.

Specify position along each edge for a valid main path.

Randomly select one of those exits to being the main path branch.

Continue to expand room from starting room.

4. Acceptance Tests _____9

[Describe the inputs and outputs of the tests you will run. Ensure you cover all the boundary cases.]

2 success cases and 6-8 boundary/exception cases

Branch Generator Feature

Run dungeon generator 50 times with input of different seeds and difficulty levels each time.

No rooms should overlap one another.

Room sizes should be distributed between a range of preset values.

Number of expanding branches off the main path of the dungeon should be relative to difficulty value specified for dungeon generation.

Output screenshots of generated dungeon

Seed Control of Dungeon Generator

Run dungeon generator 10 times with input of the same seed and difficulty level each time.

The generated dungeon should be identical each time the generator is ran.

Output screenshots of generated dungeon.

Exception 1 – cluster of rooms at the start of a branch

A branch has a high number of possible paths to reach the end room of the branch.

Possible Solution: increase percentage of superfluous rooms removed for that particular branch.

Exception 2 – cluster of rooms at the end of a branch

A branch has a high number of rooms clustered around the branche's end room.

Possible Solution: increase percentage of superfluous rooms removed from that particular branch. Focusing on rooms nearest branch end room.

Exception 3 – Branch Conflict with main path

An expanding branch reaches a point where all of its exits would conflict with main path if they were to expand.

Possible solution: end branch. Potentially increase size of branches last room if it would fit to add more interest.

Exception 4 – No Valid Room Placement

similar to above. If generator reaches a point where branch expansion hasn't reached preset values but no valid room exists exist.

Possible Solution – end dungeon generation early.

Exception 5 – No Seed or Difficulty values available

If the seed and difficulty isn't sent to the dungeon generator, the dungeon generator will use a random seed of its own generation and a starting difficulty of 0.

Exception 6 – No Items Available

No item object has been specified.

Possible solution – Have a generic object be used as a placeholder, showing where an item would be located. Debug print a message warning that no item object was specified.

Exception 7 – No Enemies Available

No enemy object has been specified.

Possible solution – have a generic enemy object used as a placeholder, showing where an enemy would have been spawned. Debug print a message warning that no enemy object was specified.

Exception 8– No Player Available

No player object has been specified.

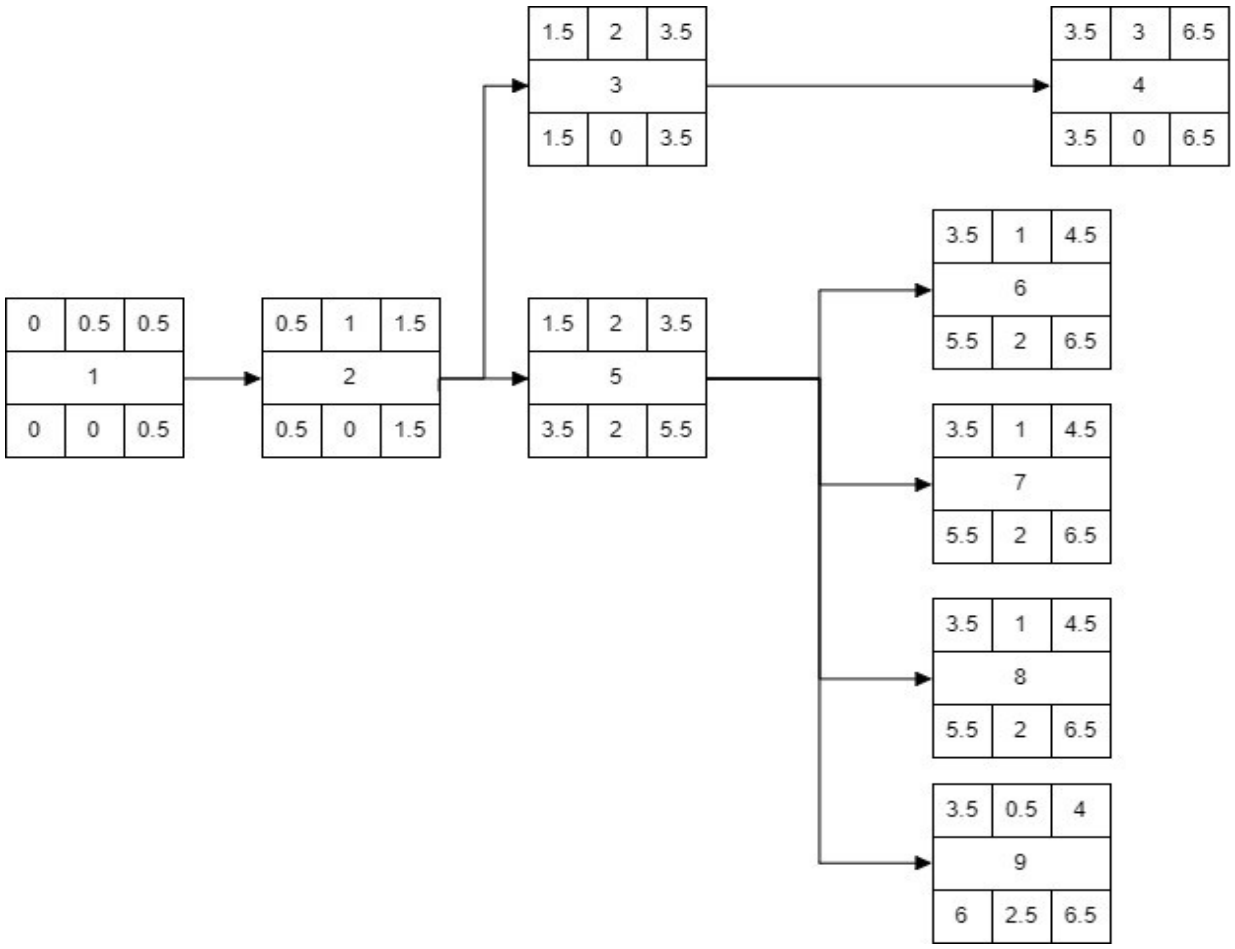
Possible solution – as with missing item or enemy objects, spawn a generic object to use as a player placeholder, showing where player would have spawned. Debug print a message warning that no player object was specified.

5. Timeline _____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Define range of room sizes and ratios for dungeon sizes per difficulty	0.5	-
2. Specify data structure for rooms and doors	1	1
3. setup room generation algorithms	2	2
4. get initial main path of rooms generating (from start to middle to end)	3	3
5. get branching rooms to generate	2	2
6. generate visuals for rooms	1	5
7. Create physics colliders around rooms	1	5
8. Populate rooms with enemies & items	1	5
9. Spawn player object in start room, and exit object in end room.	0.5	5

Pert diagram



Gantt timeline