Name: Dan Blanchette

1. Brief introduction

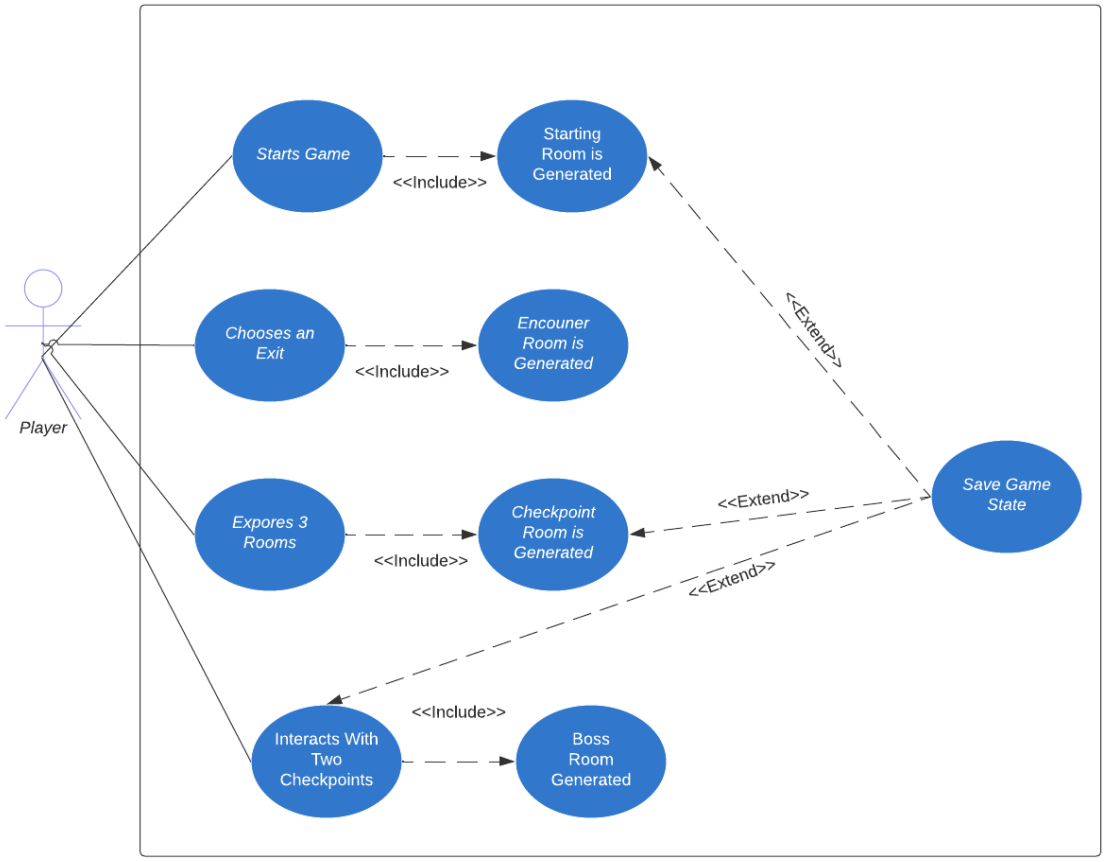
My Feature for D.R.E.A.D (Destroy Ravenous Extraterrestrials and Depart) video game is a room tile randomizer. The game operates in the top-down shooter format and are comprised of multiple rooms which constitute one level of progress for the player.

There will be six configurations for the rooms: one player starting room, three rooms that have paths (qty 1 -3 exits) joined to the previous room by a corresponding cardinal direction from the last transitioned path (North, East, West, or South), a checkpoint room, and a boss arena.

All adjoining rooms will be assigned randomly with a new room (based on the direction the player entered from a previous room). After the player succeeds at three encounters, a checkpoint room will be generated. Once the second checkpoint room has been encountered, the boss arena will be generated. If the player succeeds at the last encounter, it will signify the end of the level and a final exit will be generated for them to interact with.

2. Use case diagram with scenario

Use Case Diagrams



Scenarios

Name: Starts Game
Summary: The player starts the game by pressing the button on the input device
Actors: Player
Preconditions: Player presses start button
Basic sequence:

Step 1: Start button is pushed.
Step 2: Scene transitions from menu to level.
Step 3: Level starting room is generated.

Exceptions:

Step 1: Game state of player progress is saved
Step 2:

Post conditions: Calculated value is displayed.
Priority: 1*
ID: L01

Name: Chooses an exit
Summary: From the starting room, the player moves their character to a room exit (up, down, left, or right)
Actors: Player
Preconditions: Player is in the starting room
Basic sequence:

Step 1: Player moves to one of the exits
Step 2: Based on the chosen direction a new room is generated with one to three exits as options.
Step 3: Player enters thew newly generated room.

Exceptions:

None

Post conditions: Counter keeps track of rooms visited
Priority: 1*
ID: L02

Name: Explores three rooms
Summary: The player increments a room counter for each room the successfully complete. After three rooms have been encountered by the player, a checkpoint room is generated.
Preconditions: Room counter is equal to three rooms.
Basic sequence:

Step 1: Player exits the room.
Step 2: New room is generated, and room counter is incremented.
Step 3: Room counter condition is met, and checkpoint is generated.

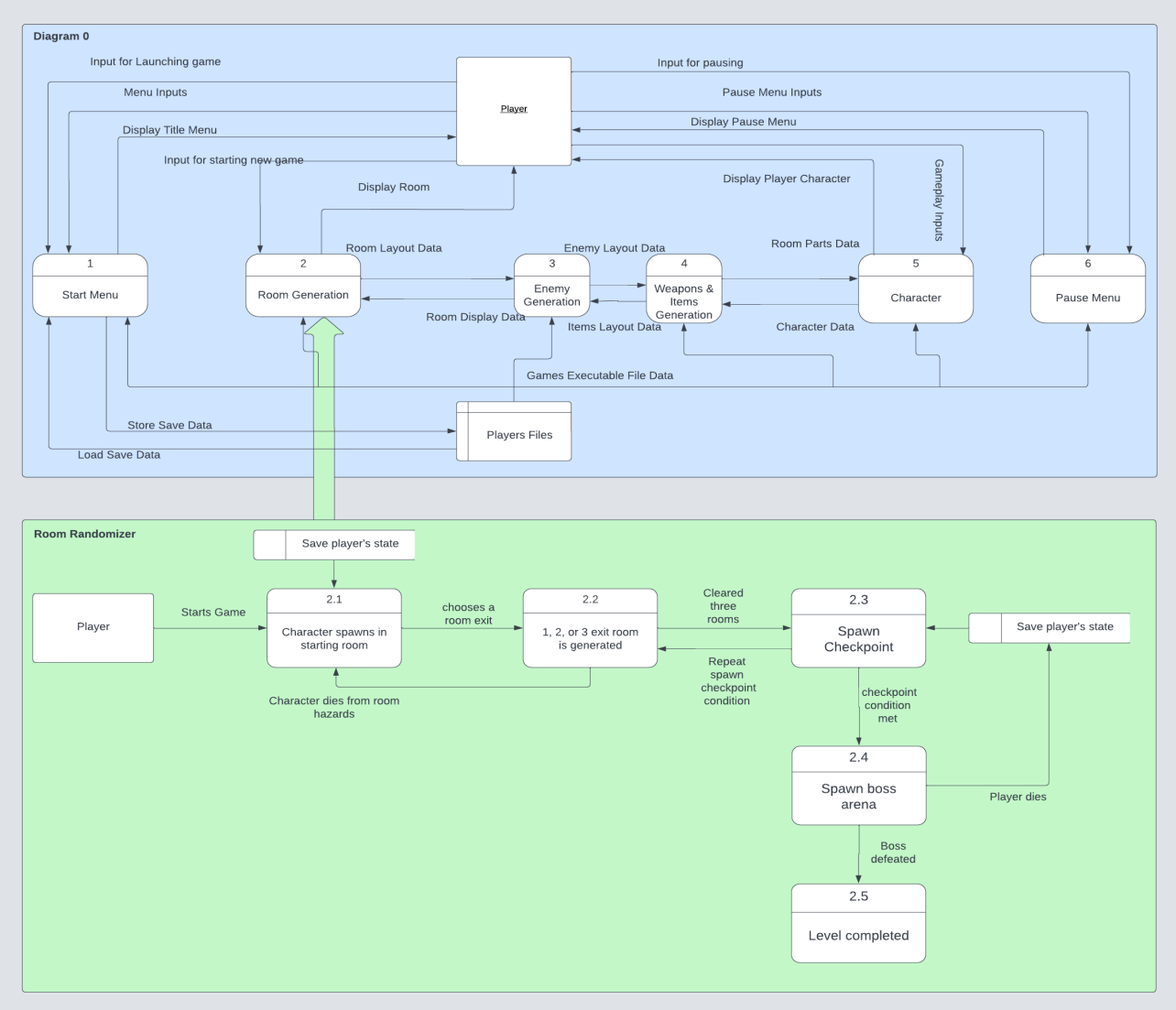
Exceptions:
Step 1: Game state of player progress is saved
Post conditions: Room counter is reset
Priority: 2*
ID: L03

Name: Interacts with two checkpoints
Summary: The player has met the conditions to generate two checkpoint rooms. The criteria are met to spawn the boss room.
Actors: Player
Preconditions: Two checkpoints are used
Basic sequence:
Step 1: Second checkpoint is generated and active.
Step 2: Character moves through a single exit.
Step 3: Boss room is generated.
Step 4: Character enters boss room.
Exceptions:
Step 1: Game state of player progress is saved.
Post conditions: Calculated value is displayed.
Priority: 1*
ID: L04

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

3. Data Flow diagram and Process Description

Data Flow Diagrams



Process Description

Player Presses Start Button	YES			NO
Starting Room is generated	Room with 1 exit generated	Room with 2 exits generated	Room with 3 exits generated	Exception Message Displayed
Move Into the Next Room	Character Has Encounter			Entrance is Sealed off. Character Explores Room
Increment Room Counter	Generate Checkpoint Room			Re-load Saved State
Checkpoint Reached	Increment Checkpoint Counter			Re-load ast Saved State
Generate Boss Room	Encounter Final Boss			Return to Checkpoint Prior to Boss Room

4. Acceptance Tests

Room Generator Feature

RID = Room Identification

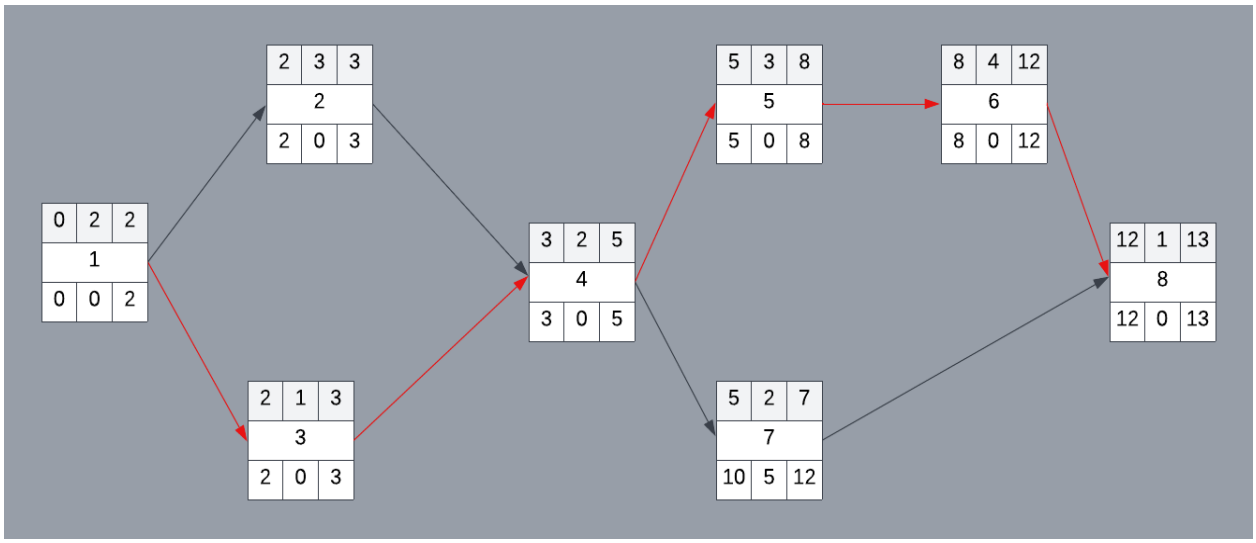
Output	Room Types	Expected Result	Notes
Vector_RNG (1-6)	6	0 < n <= 6	The level selection vector holds the room IDs for level and are randomly selected for generation.
RID: 1,3	1 exit rooms 3 exit room	0 <= 2(n) + 1 <= 1	1 exit and 3 exit rooms generated from the seed will be bound to room ID's 1 and 3. Testing must agree with these values generating the correct rooms within the constraints.
RID: 2,4	2 exit rooms 4 starting room	1 <=2(n) <= 2	Generates a 2 exit. The starting room will be a fixed value guaranteed to be generated (RID:4). Testing must agree with the generated values within the correctly constrained values.
RID: 5	Checkpoints	n%5 = 0	Generates a checkpoint room (RID=5) determined by modulus operation. If the room is generated outside of these constraints, and exception should be thrown.
RID: 6	Boss Room	n%6 = 0	Generates a boss room (RID=6) determined by modulus operation. If the room is generated outside of these constraints, and exception should be thrown.

5. Timeline

Work items

Task	Duration (Hrs)	Predecessor Task(s)
1. Level Design (Conceptual)	2	-
2. Level Implementation	3	1
3. Obstacles/Hazard Design	1	1
4. Sound Implementation	2	2, 3
5. User Documentation	3	4
6. Programming	4	5
7. Testing	2	6
8. Installation of Feature	1	5, 7

PERT diagram



Gantt timeline

	1 hr	2 hr	3 hr	4 hr	5 hr	6 hr	7 hr	8 hr	9 hr	10 hr	11 hr	12 hr	13 hr	14 hr
Task 1: Level Design														
Task 2: Level Implementation		1												
Task 3: Obstacle/Hazard Design		1												
Task 4: Sound Implementation			3											
Task 5: User Documentation					4									
Task 6: Programming								5						
Task 7: Testing					4									
Task: 8: Installation of Feature												7		