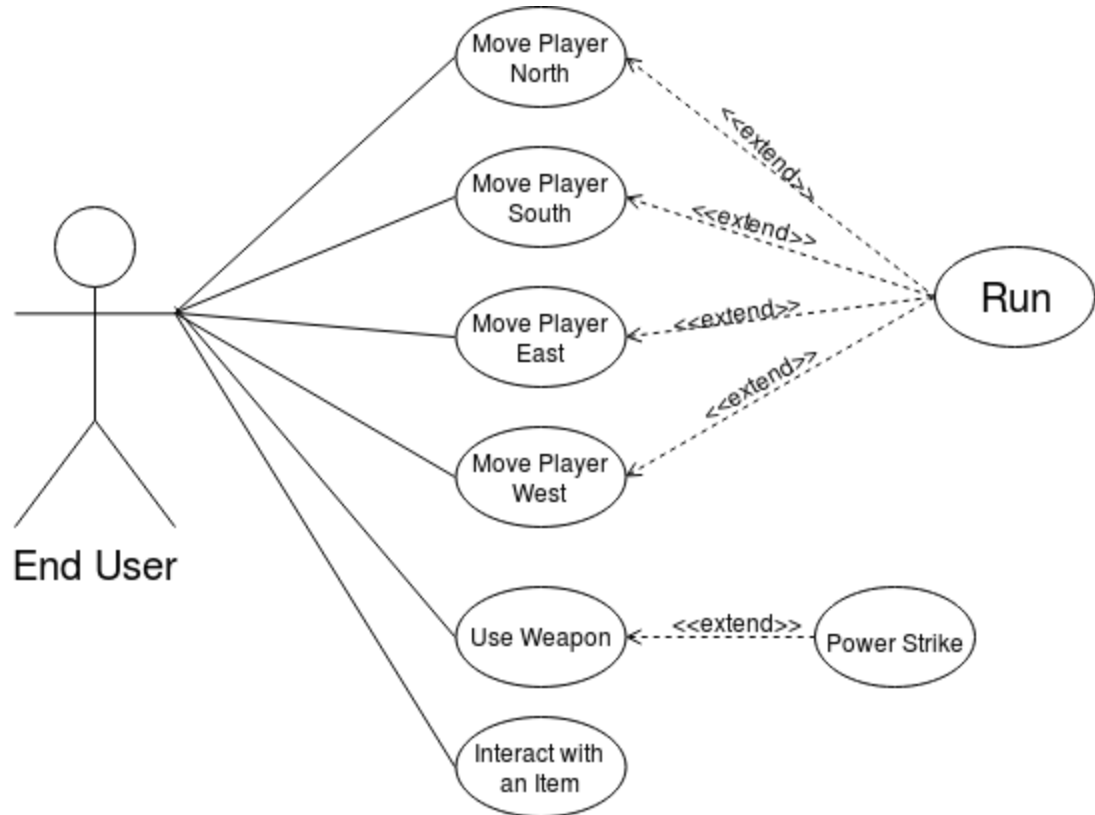


1. Brief introduction _/3

I will be championing the Player class; loading new/saved players, managing health points(HP), moving the player, and using inventory/weapons.

2. Use case diagram with scenario _14

Use Case Diagrams



Scenarios

Name: End User to Player Controls

Summary: Shows how the End User can move the player on the game screen, use weapons, and interact with items

Actors: End User

Preconditions: Game scene and objects have been initialized; the game has been started

Basic sequence(order can be varied):

Step 1: Move N, S, E, or W)

Step 3: Use Weapon

Step 4: Interact with an item

Exception 1 - Running

Step 1: Push a specified key

Step 2: Move in desired direction

Exception 2 - Power Strike

Step 1: Push a specified key

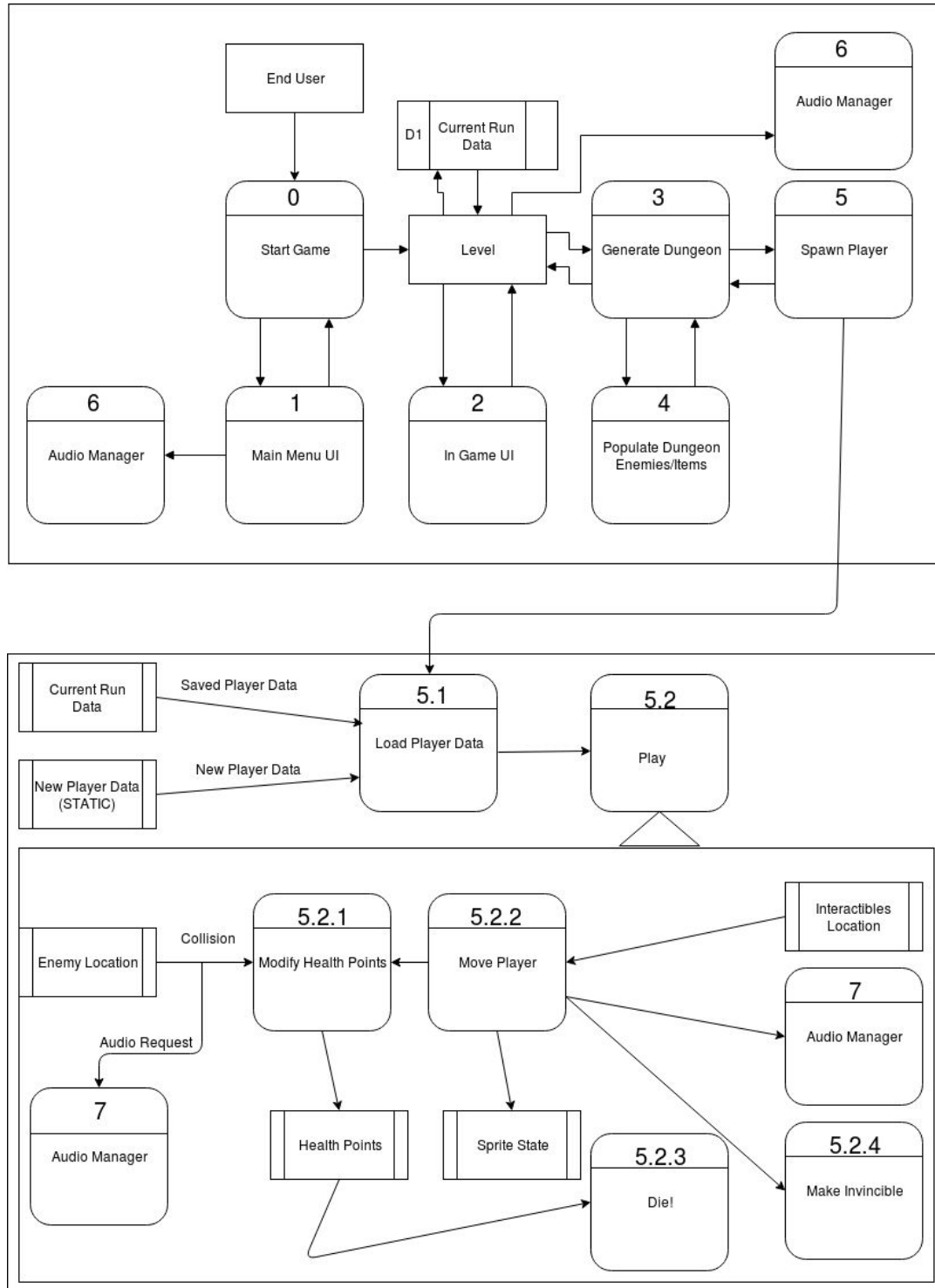
Step 2: Use weapon

Post conditions: Send information to Game manager about interactions with Enemies and Items.

Priority: 1 for main use cases, 3 for extends

3. Data Flow diagram(s) from Level 0 to process description for your feature ____14

Data Flow Diagrams



Process Descriptions

Spawn Player

Load player data

IF new game THEN load default data

ELSE IF saved game THEN load saved data

Move Player

IF moving in different direction THEN change Sprite State

IF contact with solid object THEN do not move players location

ELSE IF contact with intractable

THEN do something and move player

IF intractable increases HP

IF HP <100

THEN increase HP

IF intractable makes invincible THEN Make Invincible

IF sound required THEN call Audio Manager

ELSE IF contact with Enemy

IF not invincible

decrease HP

IF HP is zero

Die

4. Acceptance Tests _____9

Test Movement - Make sure that the Player only moves on the game scene when there is not a solid object in the way.

-Randomly move Player throughout 100 randomly generated levels without enemies.

-Make sure that at the end of the test the player is still in the game scene

Test Health Point

-Write a test function that attempts to add and subtract one thousand integer values between 0 and 100 from the Players HP.

-Make sure that the Player's HP stays below 100

Test Sprite State

-Randomly move the player through 100 randomly generated levels.

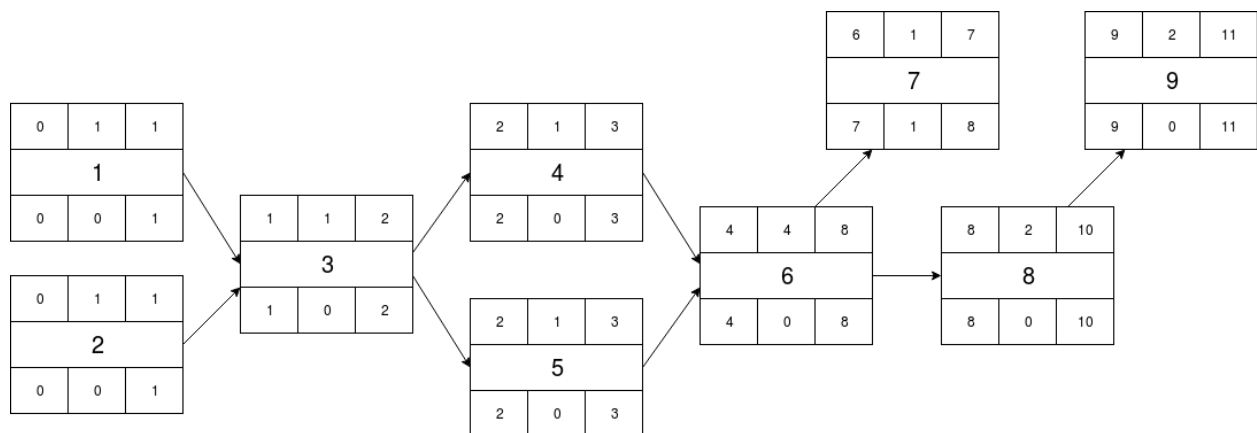
-At random intervals, check Sprite State against expected Sprite State (based on interaction with solid objects, Enemies, or Interactables.

5. Timeline ____/10

Work items

Task	Duration (PWks)	Predecessor Task(s)
1. Create UML Model	1	-
2. Create Class and Sequence Diagrams; Gantt Timeline	1	-
3. Write Skeleton Code/Public Headers	1	1,2
4. Write Acceptance Test plan	1	3
5. Write classes that will be inherited by Enemy Class	1	3
6. Write remaining core features	4	4,5
7. Update Gantt Chart	1	6(started)
8. Finish coding any missing features	2	6
9. Test Release Candidate Make last-minute changes Help other team members	2	8

Pert diagram



Gantt timeline

