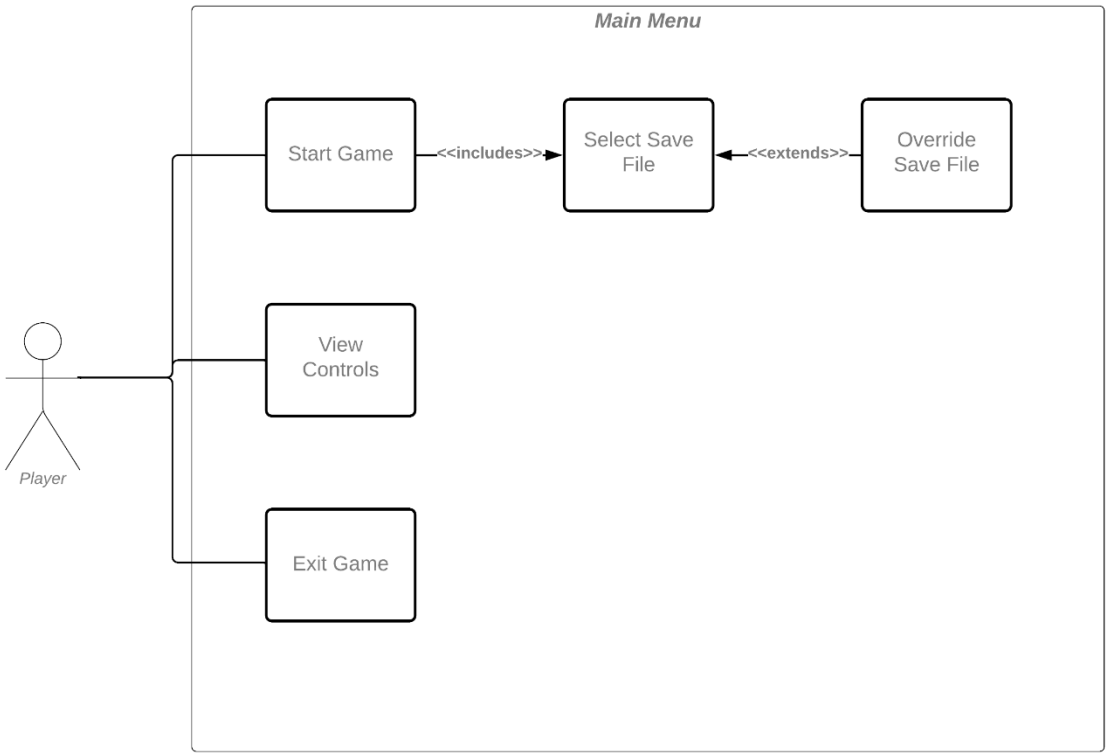
1. Introduction

My feature for the game Dungeon Jump will be the implementation of the user interface and game menus. The game will open to a main menu, from which users can start the game, view the controls, or exit the game. Once in the game, users will be able to access the pause menu, from which the controls are also visible, and they can choose to resume, save, or exit the game. The user interface itself will be visible while the user is playing the game. It will display the current Health Points of the player as well as the details of the instance they are currently in, and of any NPCs/locations they are facing.

2. Use Case Diagrams



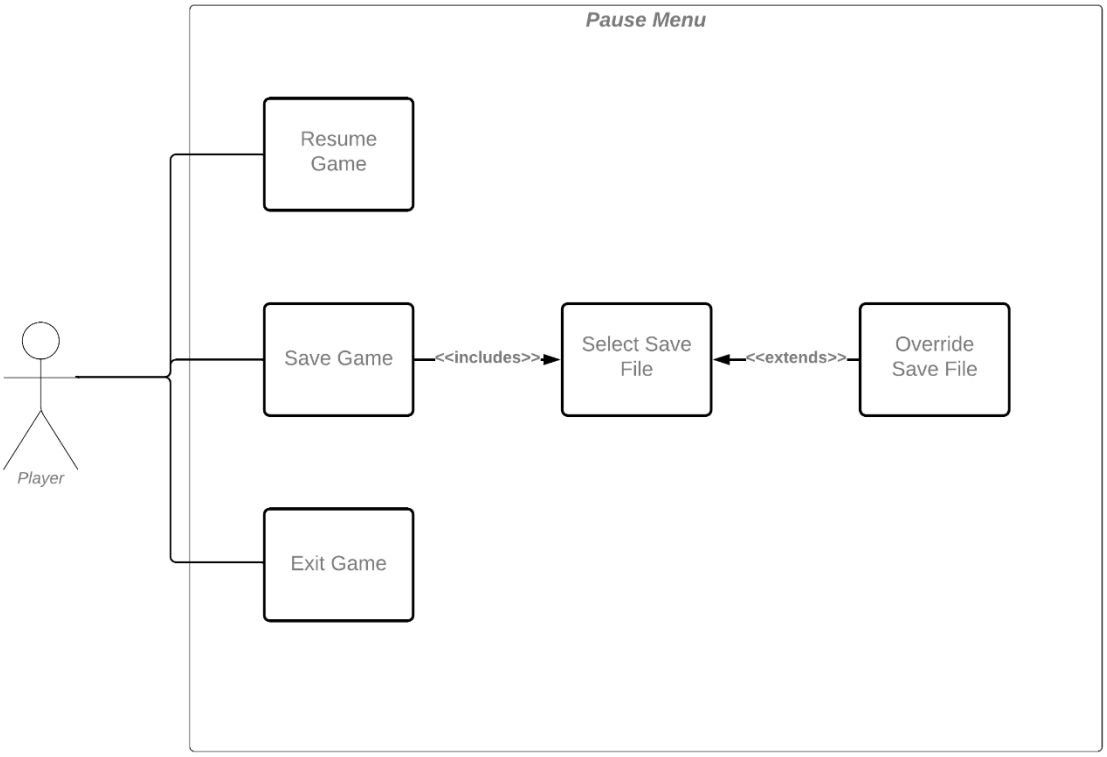
Scenarios

Name: Start Game
Summary: The player selects the option to start the game from the main menu.
Actors: Player
Preconditions: The game is running.
Basic sequence:
Step 1: The player selects the option to start the game.
Step 2: The player selects the save file location to use for the game.
Exceptions:
Step 2: The player selects a save file location that is already in use. They override the save file that was already there.
Post conditions: The game is loaded from the selected save file.

Priority: 1
ID: START

Name: View Controls
Summary: The player selects the option to view the controls from the main menu.
Actors: Player
Preconditions: The game is running.
Basic sequence:
Step 1: The player selects the option to display the controls.
Post conditions: The game controls are displayed on screen.
Priority: 3
ID: CNTRLS

Name: Exit Game
Summary: The player selects the option to exit the game.
Actors: Player
Preconditions: Either the main menu or pause menu are active.
Basic sequence:
Step 1: The player selects the option to exit the game.
Post conditions: The game application exits.
Priority: 1
ID: EXIT

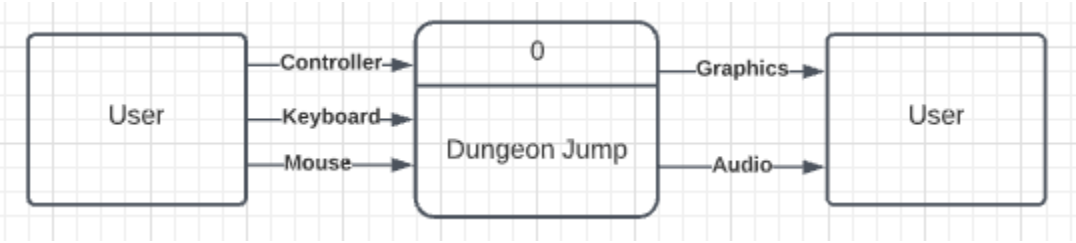


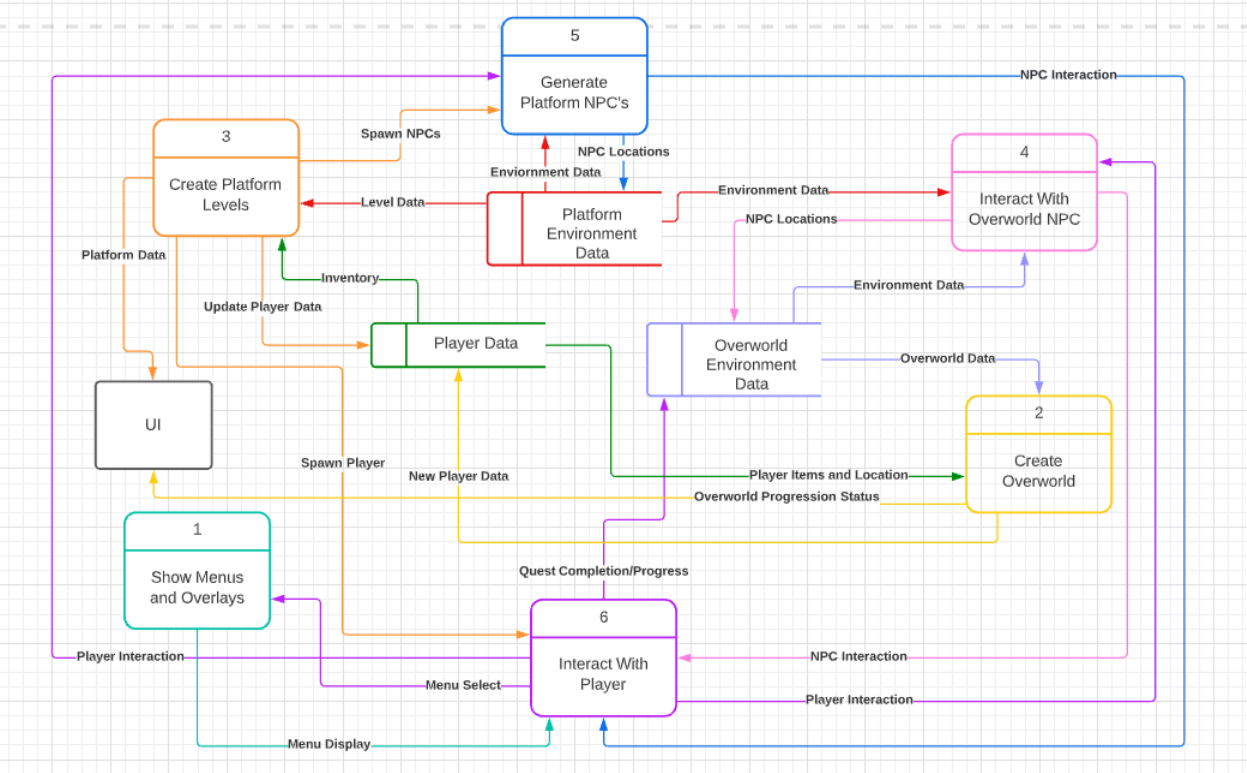
Scenarios

Name: Resume Game
Summary: The player selects the option to resume the game from the pause menu.
Actors: Player
Preconditions: The game is paused.
Basic sequence:
Step 1: The player selects the option to resume the game.
Post conditions: The game hides the pause menu and shows the current game screen.
Priority: 1
ID: RESUME

Name: Save Game
Summary: The player selects the option to save the game from the pause menu.
Actors: Player
Preconditions: The game is paused.
Basic sequence:
Step 1: The player selects the option to save the game.
Step 2: The player selects the save file location to save the game in.
Exceptions: The player selects a save file location that is already in use. They override the save file that was already there.
Post conditions: The data for the current game is saved in the specified save file.
Priority: 1
ID: SAVE

3. Data Flow Diagrams





My process is 1: Show Menus and Overlays.

Process Descriptions

Main Menu:

Player Selection:

CASE player selects option:

START GAME: opens save file options

IF file is empty

Create new file

ELSE

Ask to override or load

IF override

Create new file

ELSE

load

VIEW CONTROLS: opens controls

EXIT GAME: exits game

DEFAULT: do nothing

END CASE

Pause Menu:

Player Selection:

CASE player selects option:

```
                RESUME GAME: resume game
                SAVE GAME: opens safe file options
                        IF file is empty
                                Save to new file
                        ELSE
                                Ask to override
                                IF override
                                        Save to present file
                                ELSE
                                        Do not save
                        DEFAULT: do nothing
        END CASE

        Inventory:
                IF item is obtained
                        Display sprite

In game:
        Health:
                IF health changes
                        Display value
                        IF value is at threshold for sprite change
                                Update HP sprite

        Player Input:
                IF pause input
                        Display Pause Menu
```

4. Acceptance Tests

Button Testing

There is not much room for “breaking” the UI and menus, given the small amount of use cases. This test script will merely press every button possible in the main and pause menus as the input and ensure the appropriate post conditions are met as the output.

HP Generation

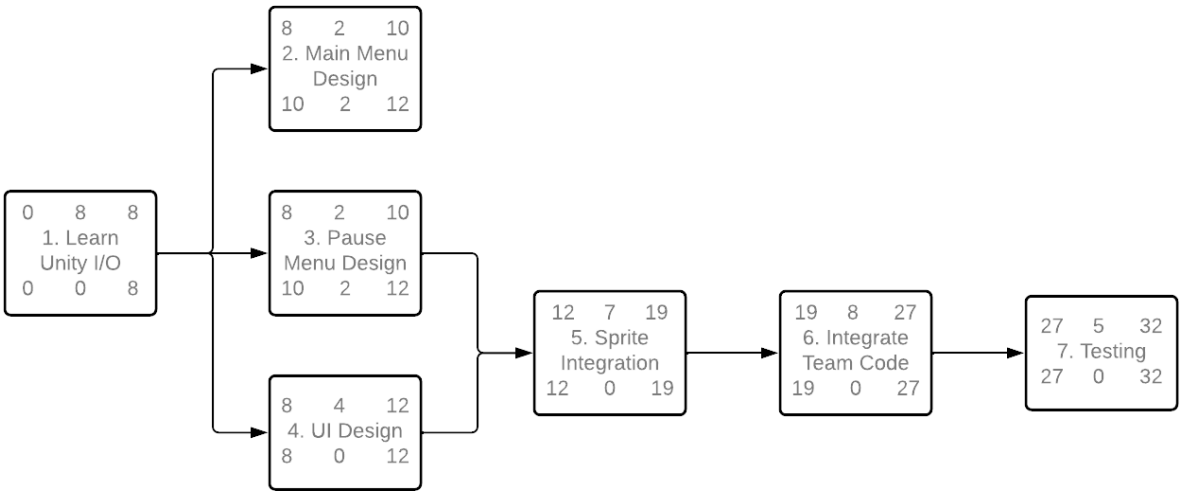
This test script will use as an input a series of random numbers to set the user’s HP to. The output should be the HP sprites on the UI matching the set numbers, and numbers that are not appropriate for the user HP (too high or too low) should be discarded, so the expected output would be no change from the previous value.

5. Timeline

Work Items

#	Task	Duration (Hrs)	Predecessor Tasks
1	Learn Unity I/O	8	-
2	Main Menu Design	2	1
3	Pause Menu Design	2	1
4	UI Design	4	1
5	Sprite Integration	7	3, 4
6	Integration w/ Team Code	8	5
7	Testing	5	6

Pert Diagram



Gantt Timeline