

Name__ Ryan Buckel_____

Mark _____/50

[**Instructions:** Remove everything that is not a heading below and fill in with your own diagrams, etc.]

1. **Brief introduction** __/3

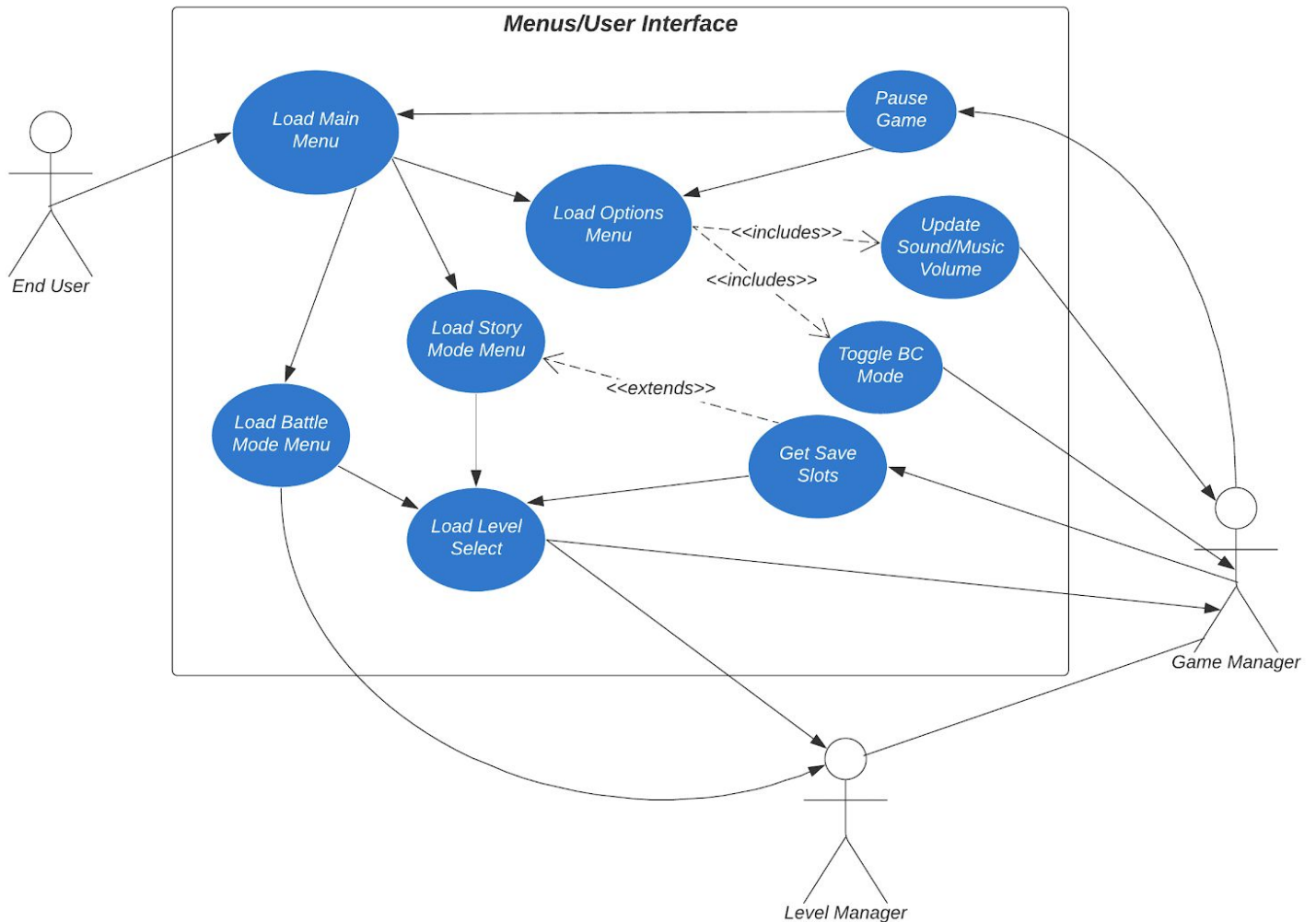
My feature is mainly involved with the User Interface of Super Spicy Gun Game. The UI consists of various menus to choose between all modes of the game with the notable menus being the main menu, pause menu, and an options/settings menu. Each menu has submenus for the user to interact with. The methods implemented into the UI will directly improve “quality-of-life” with respect to configuring the game.

2. Use case diagram with scenario _14

Use Case Diagrams

User Interface Use Case Diagram

Ryan | February 27, 2021



Scenarios

Name: Load Main Menu

Summary: The main menu is loaded when the game is launched, giving options to play certain game modes, change settings, or quit the application.

Actors: End User

Preconditions: End User has launched game.

Basic sequence:

1. If 'Story Mode' is clicked, switch scene to story mode menu.
2. If 'Battle Mode' is clicked, switch scene to battle mode menu.

3. If 'Options' is clicked, switch scene to story mode menu.
4. If 'Quit Game' is clicked, close the application.

Exceptions: None

Post conditions: Any one of the story mode, battle mode, or options menus are displayed, or the application is closed.

Priority: 2

ID: LMM

Name: Load Options Menu

Summary: A list of changeable settings are displayed when the "Options" button is clicked.

Actors: End User

Preconditions: Main menu is the current scene.

Basic sequence:

1. Check for change in options parameters since first load
2. If there is a change, store changed value(s) in temporary storage.
3. If 'Apply Changes' is selected, function calls to appropriate scripts will be called
4. If 'Back' button pressed, go to last visited scene.

Exceptions: if change in any option is too high or too low of a value, change to prefixed min or max value.

Post conditions:

Priority: 2

ID: LOM

Name: Load Story Mode Menu

Summary: The story mode menu is loaded when it is selected from the main menu.

Actors: End User

Preconditions: Main menu is the current scene.

Basic sequence:

1. If user selects 'New Game,' change scene to new game cutscene.
2. If user selects 'Load game,' change scene to Load Game Menu.
3. If user presses 'Back,' change scene to the previous one.

Exceptions: None

Post conditions: The user is displayed with a new menu (either main menu, load game menu, or a cutscene).

Priority: 2

ID: LSM

Name: Load Battle Mode Menu

Summary: Menu that allows the user to input data for how many people are to battle with each other. Also allows to modify select rules to change gameplay characteristics.

Actors: End User

Preconditions: Main menu is the current menu.

Basic sequence:

1. Check to see how many players are battling.
2. Check to see how many rounds there are.
3. Check if a time limit is specified or not.
4. If 'Back' button is pressed, navigate to previous menu.

Exceptions: None

Post conditions: Player and battle modification data is sent to the game manager, and level select screen is loaded.

Priority: 3

ID: LBM

Name: Load Level Select

Summary: When the user has either loaded a save from story mode, started a new game from story mode, or entered necessary battle mode information, a level selection screen will display before the game starts.

Actors: End User, Level Manager

Preconditions: None.

Basic sequence:

1. If a level is clicked on, change scene to the appropriate level.
2. Send data to game manager to differentiate between battle and story mode to accommodate spawning algorithm.
3. If the back button is pressed, go to previous screen.

Exceptions: None.

Post conditions: A level is selected, and control is handed off to game manager/level manager.

Priority: 2

ID: LLS

Name: Get Save Slots

Summary: When the end user wants to load a game, the game manager will attempt to retrieve save data for each slot and display it on the screen.

Actors: Game Manager

Preconditions: 'Load Game' Button is selected.

Basic sequence:

1. Call an accessor function from the game manager to get save data.
2. Configure data accordingly to display on screen.
3. When back button is pressed, go to New/Load Game menu.

Exceptions: If save data is not found, display placeholder 'No Save Data' in the affected slot(s).

Post conditions: Save Data is displayed in each slot, ready to be selected by the user.

Priority: 3

ID: GSS

Name: Pause Game

Summary: Whenever the user needs to stop playing, a button is pressed to freeze the game in place, displaying options to resume game, change settings, or quit game.

Actors: End User, Game Manager

Preconditions: The player is currently playing in a game.

Basic sequence:

1. If 'Resume Game' selected, switch scene to gameplay.
2. If 'Options' selected, switch scene to options menu.
3. If 'Quit Game' selected, switch scene to main menu.

Exceptions: Whenever the user is already in a menu, do not pause game.

Post conditions: Focus is returned either to the game, the options menu, or the main menu.

Priority: 2

ID: PGM

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

Data Flow Diagrams

Diagram 0:

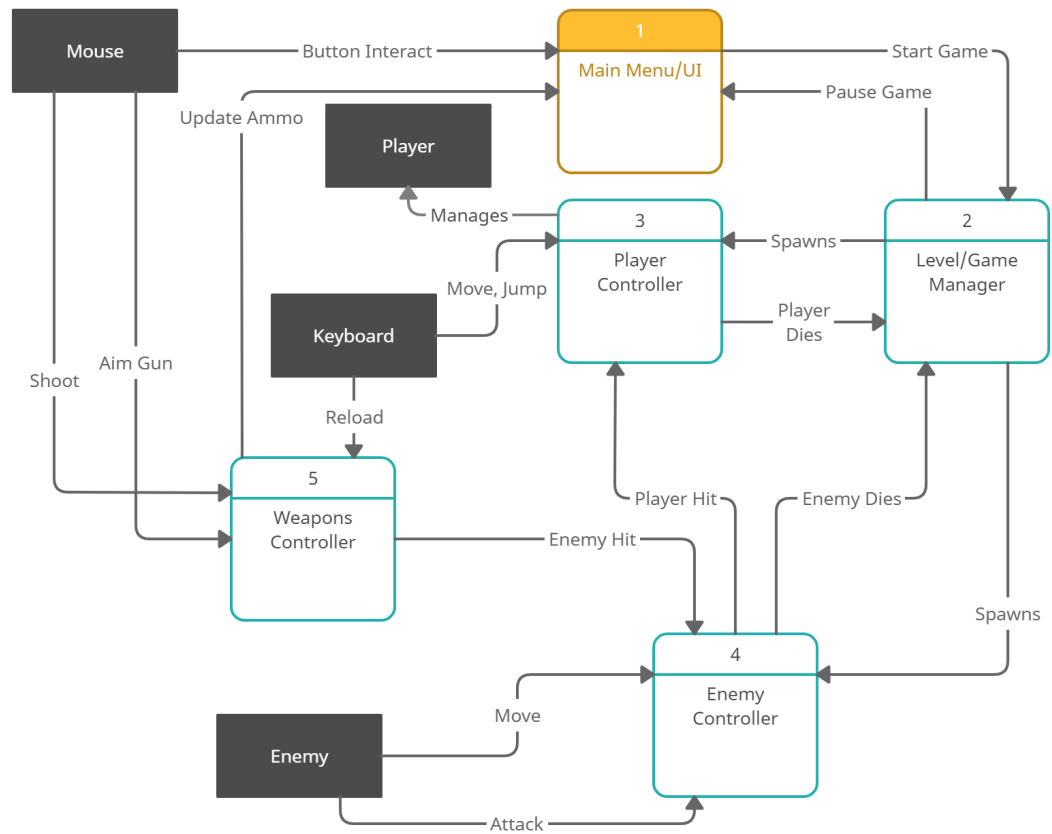


Diagram Level 1: Main Menu/User Interface

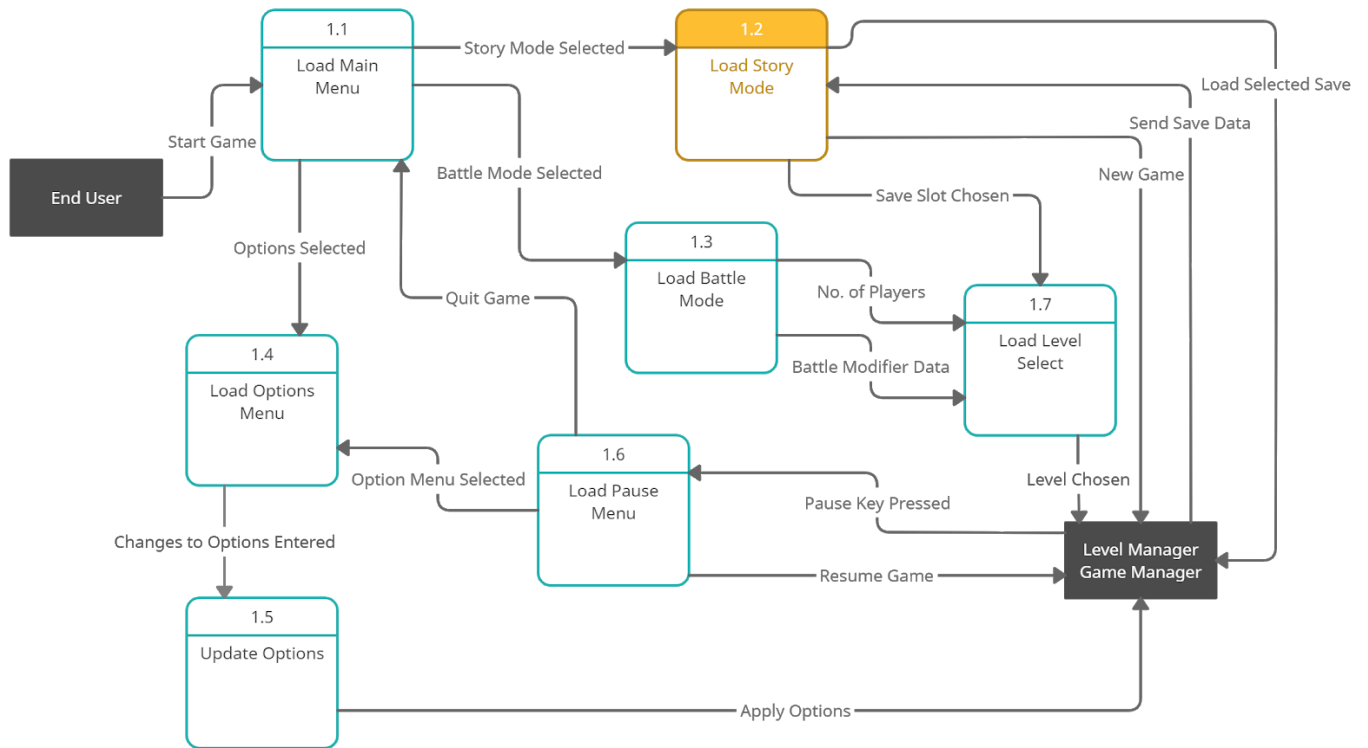
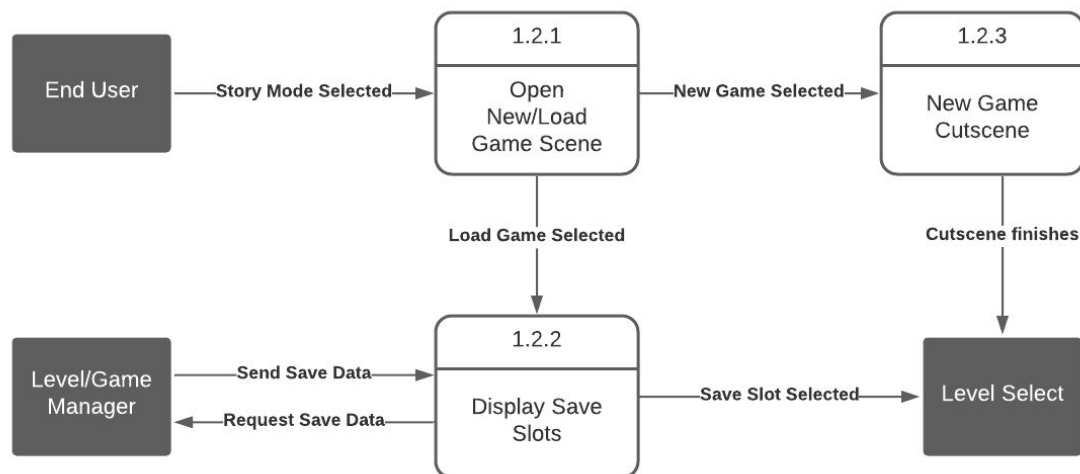


Diagram Level 2: 1.2 - Load Story Mode



Process Descriptions

1.2.2 - Display Save Slots:

```

//Get Save data, load into save slots
FOR each save slot when i < 3, i++:
    FOR each separate data entry j < 2, j++:
        init SaveDataArray[3][2]
        Call Function from Game manager ('GetSaveData()')
        SaveDataArray[i][j] = GetSaveData(i, j)
        DISPLAY Timestamp, In-Game Location in slot
    END FOR
//When save slot is selected
OnEvent('buttonClick'){
    SET active save data
    LOAD GameScene
}

```

4. Acceptance Tests _____9

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

For The Options Menu:

Attempt to write settings to erroneous values (depends on UI implementation)

Possible valid values:

Sound, Music Volume: anything from 0 to 100.

BC Mode: Anything but 'true' or 'false'.

Inputs:

Sound/Music Volume: anything greater than 100 or less than 0.

BC Mode: text garbage

Outputs:

Sound / Music Volume: Reading from Unity audio listener.

BC Mode: In-Game Behavior of enemies and attacks.

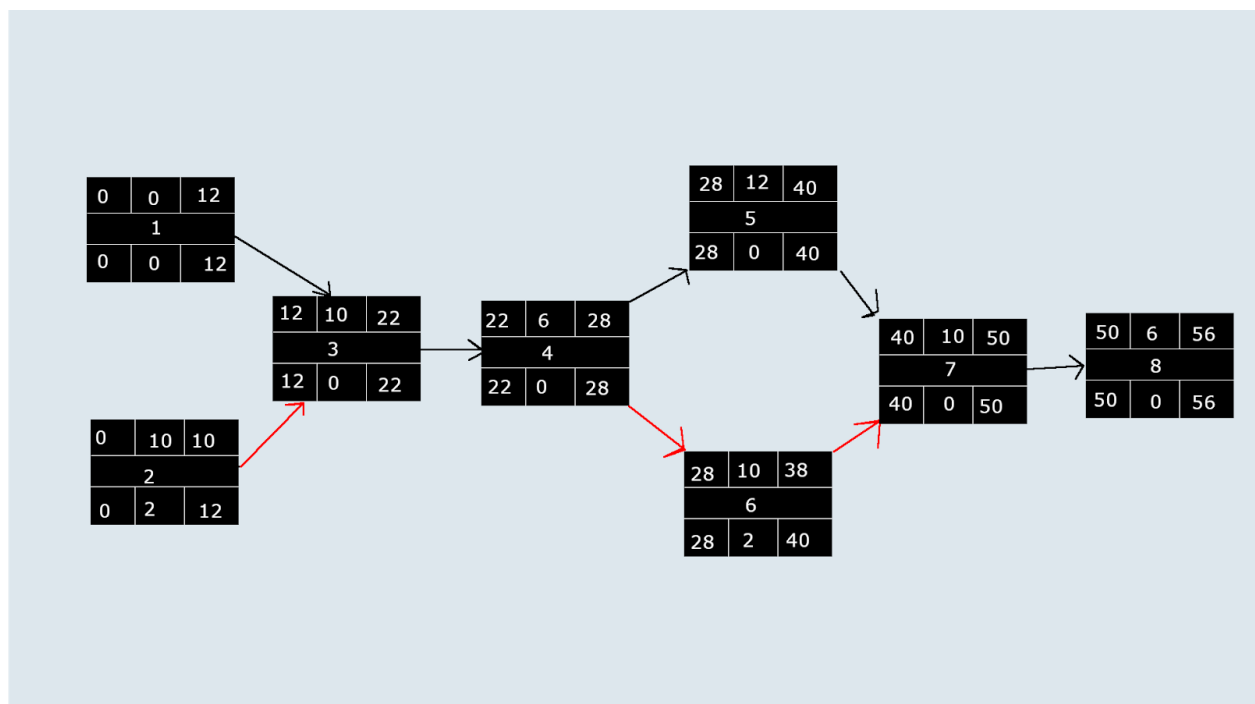
5. Timeline _____/10

Work items

Task	Duration (Hours)	Predecessor Task(s)
1. User Interface Tutorials	5	-

2. Menu(s) Design	6	-
3. Menu(s) Implementation	6	1,2
4. Initial input stress tests	2	3
5. Revise Menu Scripts, Prefabs	6	4
6. Implement In game Overlays	5	4
7. Final Testing, Revisions	3	6
8. Installation	1	7

Pert diagram



Gantt timeline

