Name: Ian Wood Mark /50

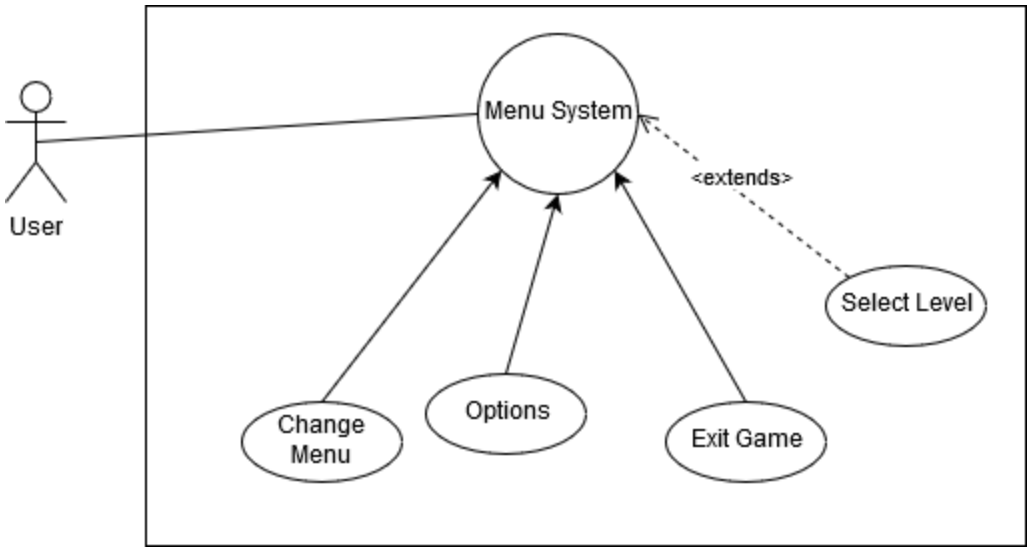
1. Brief introduction __ / 3

I am in charge of the HUD (Heads-Up Display) and the menus within Meteor Beat. The HUD displays information to the player such as how many lives they have remaining, how close they are to the goal, and a combo counter which tracks how on rhythm the user is.

The menus allow the user to select a difficulty mode, a level, and pause the game to go back to the main menu, restart the level, or quit the game.

2. Use case diagram with scenario __ / 14

Menu System Use Case Diagram



Name: Change Menu
Summary: User can switch which menu they're in to another.
Actors: User
Preconditions: User is currently in a menu and not in the core game loop.
Basic Sequence:

Step 1: Accept input of user clicking on button.
Step 2: Button changes scene to the menu that button is connected to.
Step 3: Display new menu.

Exceptions:

Step 1: Options menu during gameplay won't have difficulty option.

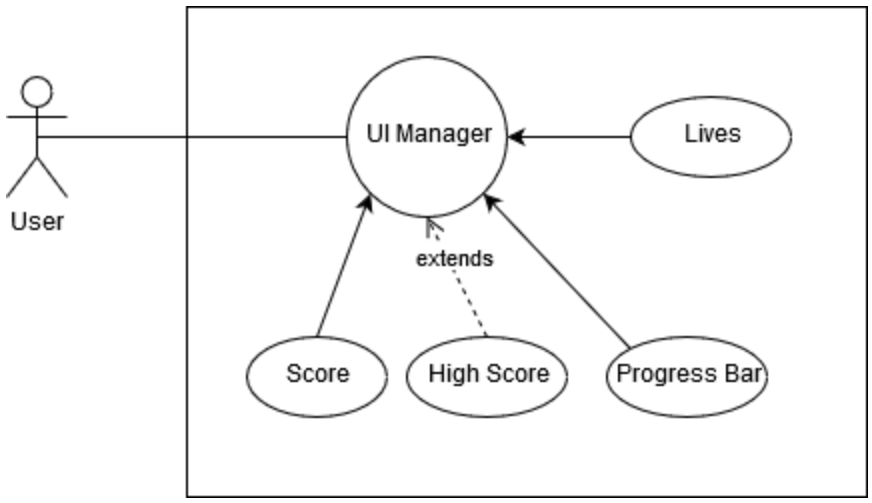
Post conditions: The menu will have changed.
Priority: 2
ID: UI01

Name: Options
Summary: User can change options like volume and difficulty.
Actors: User
Preconditions: User is in the options menu.
Basic Sequence:
Step 1: Accept input of user clicking on button or dragging the slider.
Step 2: Option will be changed.
Exceptions:
Step 1: Options during gameplay will not allow user to edit difficulty.
Post conditions: A game option will have changed.
Priority: 2
ID: UI02

Name: Exit Game
Summary: User can exit the game from the main menu or during gameplay.
Actors: User
Preconditions: User is in the main menu or has paused gameplay.
Basic Sequence:
Step 1: User clicks on "Exit Game" button.
Step 2: Game will close.
Post conditions: The game will have closed.
Priority: 1
ID: UI03

Name: Select Level
Summary: User can select one of the game's levels from this menu.
Actors: User
Preconditions: User has selected the level select option from the main menu.
Basic Sequence:
Step 1: User clicks on a level.
Step 2: Game will load the level using a loading screen.
Step 3: Game will transition to level after level loads.
Post conditions: The game will have loaded the level and begin play.
Priority: 3
ID: UI04

User Interface System Use Case Diagram



Name: Score
Summary: User can see their score during gameplay.
Actors: User
Preconditions: User is currently in the core game loop.
Basic Sequence:

Step 1: User interacts with the game, scoring points.
Step 2: Message is sent to UI element to update score.
Step 3: Display new score.

Exceptions:

Step 1: Options menu during gameplay won't have difficulty option.

Post conditions: The score will have changed.
Priority: 1
ID: UI05

Name: High Score
Summary: User can see the highest score achieved in Meteor Beat.
Actors: User
Preconditions: User is currently in the core game loop and can connect to high score server.
Basic Sequence:

Step 1: On boot up attempt to make connection to high score server.
Step 2: Display high score.
Step 3: If user scores higher, update high score.

Exceptions:

Step 1: If the game cannot connect, do not display the high score UI elements.
Post conditions: If the high score changes, it will be displayed the next time the level loads.
Priority: 3
ID: UI06

Name: Progress Bar
Summary: User can tell how close to the end of the level they are.
Actors: User
Preconditions: User is currently in the core game loop and the game is not paused.
Basic Sequence:
Step 1: As the game progresses, the ship entity should send its Z-coordinate as a message at regular intervals.
Step 2: Based on the percentage of level completed, set progress bar.
Step 3: Update progress bar graphic.
Post conditions: As the player goes through the game, the progress bar will change.
Priority: 2
ID: UI07

Name: Lives
Summary: User can tell how many lives they have left.
Actors: User
Preconditions: User is currently in the game loop and makes a collision.
Basic Sequence:
Step 1: Check that the collision will reduce the number of lives.
Step 2: Send message to Lives UI Element.
Step 3: Update the number of lives displayed.
Step 4: Display new amount of lives.
Exceptions:
Step 1: If the number of lives would be zero, show game over screen.
Post conditions: The lives displayed will change.
Priority: 1
ID: UI08

3. Data Flow diagram(s) from Level 0 to process description for your feature
___ / 14

Diagram 0

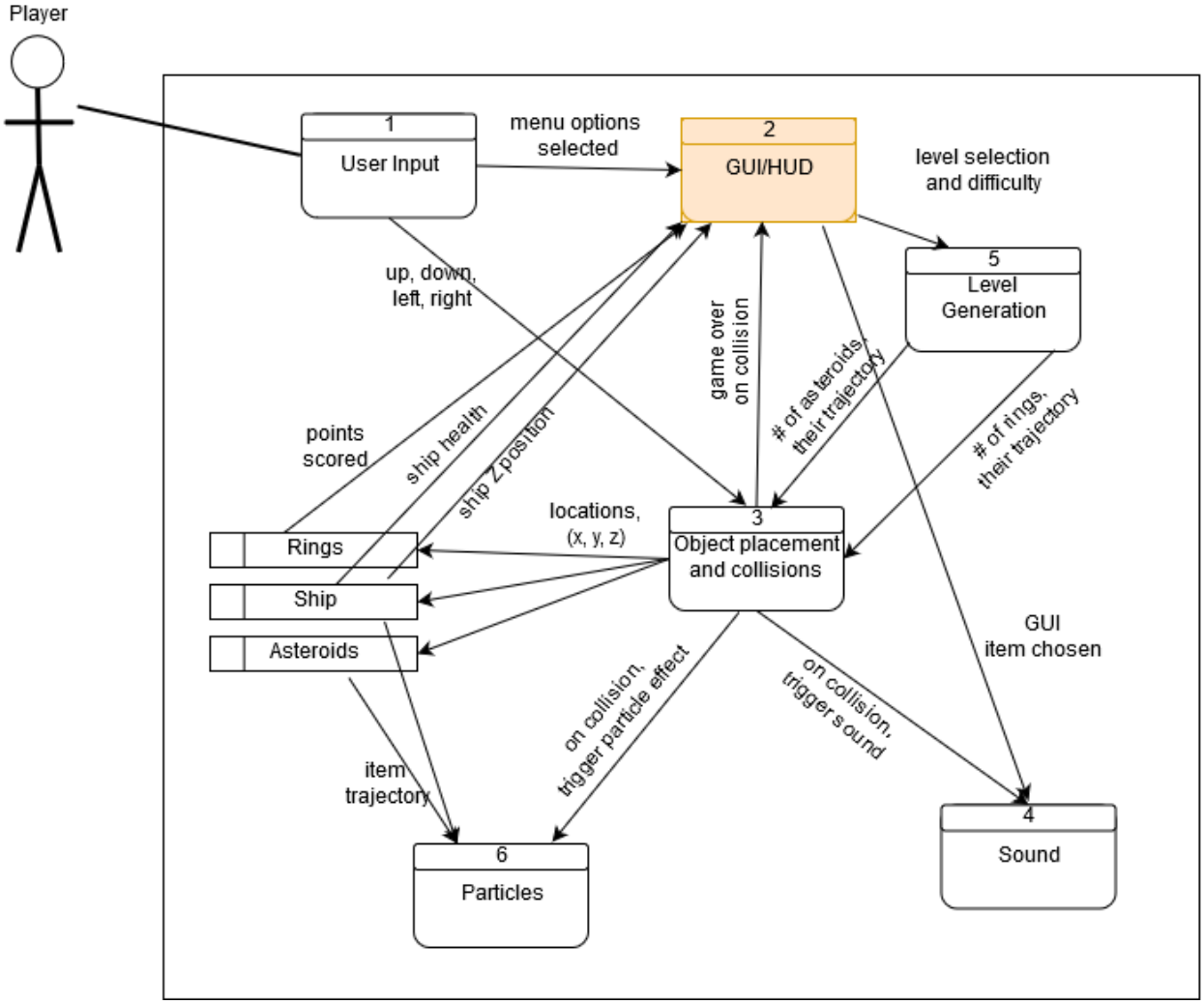
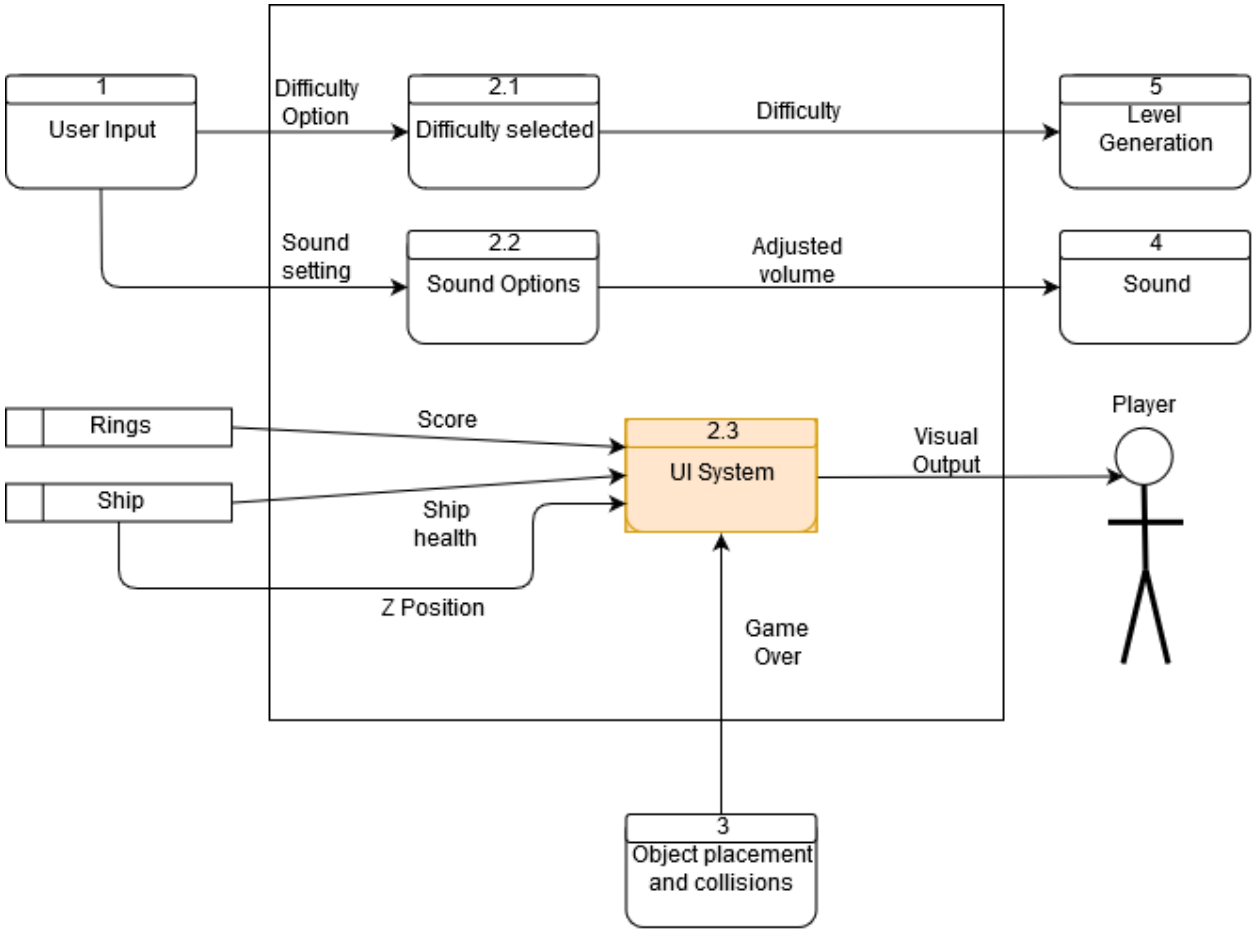


Diagram 1, 2 GUI/HUD

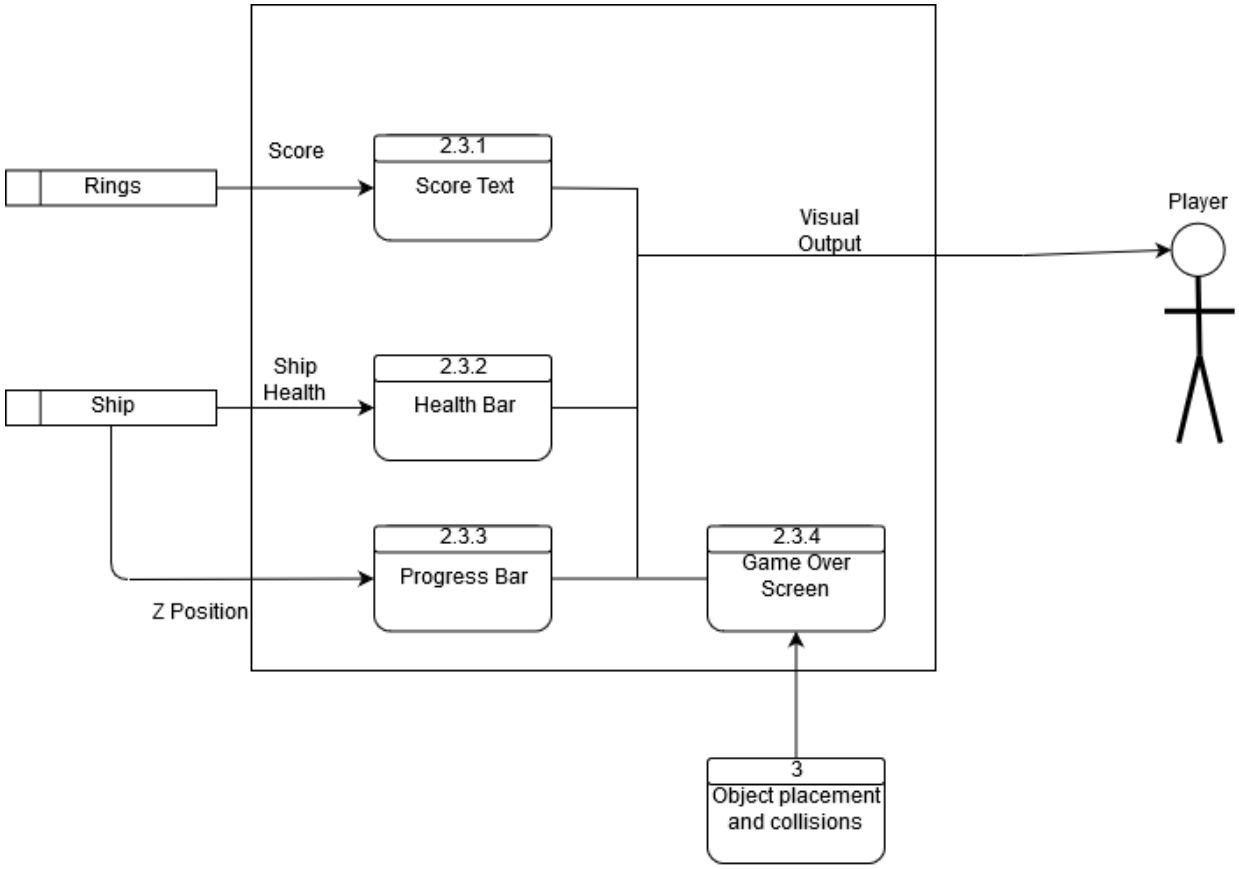


Process Descriptions

2.1 Difficulty Selected: Stores the difficulty the player selected. Uses this when generating the level.

2.2 Sound Options: Modifies how loud the volume should be. This value is sent to the sound manager.

Diagram 2, 2.3 UI System



Process Descriptions

2.3.1 Score Text: Receives message from rings when a collision is made to update the score text.

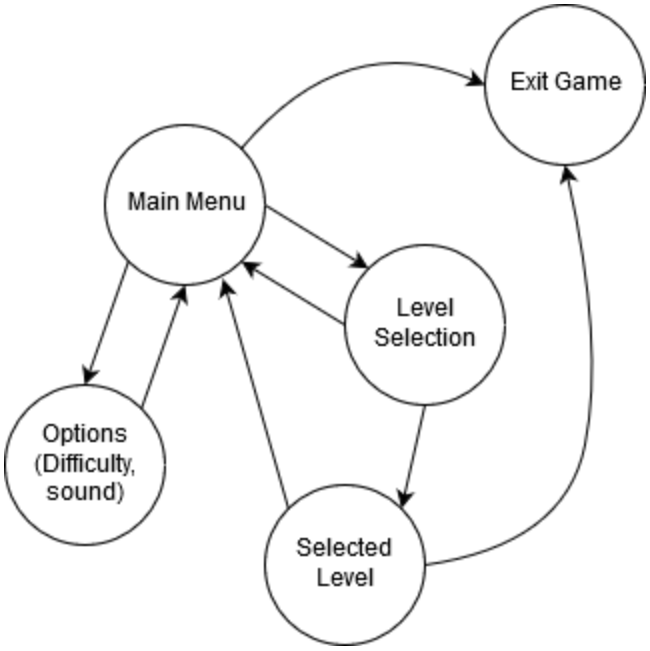
2.3.2 Health Bar: On a collision that causes damage, the ship will send a message to update the health bar visual.

2.3.3 Progress Bar: In regular intervals, the ship will send its Z position and the progress bar will calculate how much the progress bar needs to fill up by.

2.3.4 Game Over Screen: On a collision, if the health of the player .

4. Acceptance Tests ___ / 9

For the menus, I want to ensure that they follow transitions outlined on the graph below:

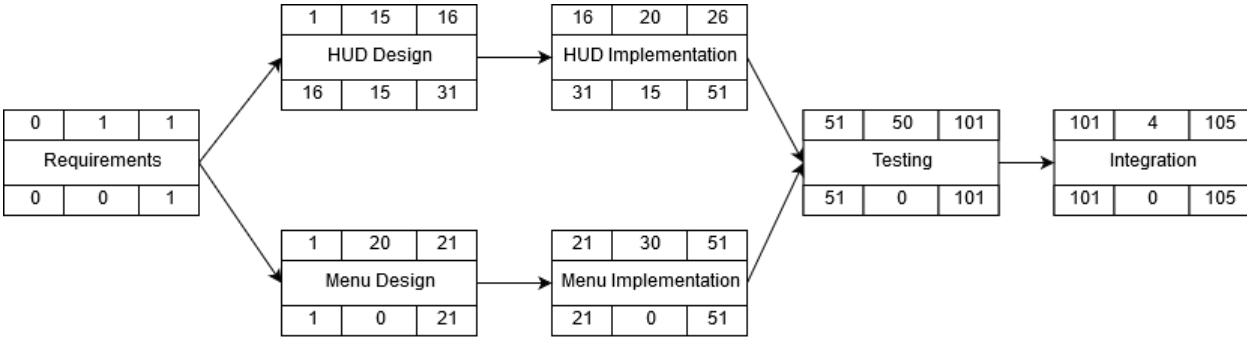


To test the HUD, I will first test that the life counter decreases when the player collides with an object. For the progress bar, I will place the ship at intervals such as 10%, 20%, etc. I will compare the progress bar to this to make sure it is correct. For the combo counter, I will want to make sure that when the counter is reset internally within the program that it displays correctly on the HUD.

5. Timeline __ / 10

Work Items

Task	Duration (PHrs)	Predecessor Tasks
1. Requirements Collection	1	-
2. HUD design	15	1
3. HUD Implementation	20	2
4. Menu Design	20	1
5. Menu Implementation	30	4
6. Testing	50	3, 5
7. Integration	4	6


0 1 1
Requirements
0 0 1
1 15 16
HUD Design
16 15 31
16 20 26
HUD Implementation
31 15 51
1 20 21
Menu Design
1 0 21
21 30 51
Menu Implementation
21 0 51
51 50 101
Testing
51 0 101
101 4 105
Integration
101 0 105


1
2
1
4
5
6