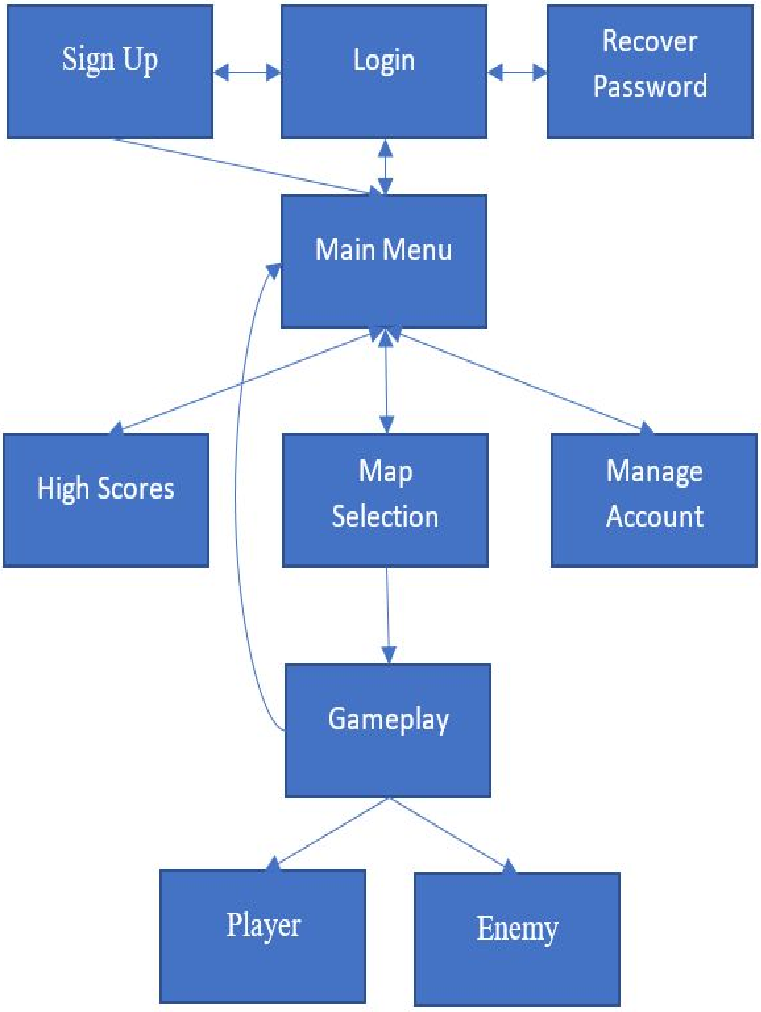
Obsidian Ark

Team 27 Incremental Testing and Regression Testing

Yi Dou, Danny Zou, Jordan Stoddard, Devon Lee, James Hawk

1.
Component Classification

1.1
Component Diagram



Component Details

Login

- Is called by the main menu component and recover password component.
- Calls the main menu component, sign up, and the recover Password component.
- Gets username and password and checks database for user.

Sign Up

- Is called by the login component
- Calls the main menu and login component
- Gets new username and password and creates a new account in the database

Recover Password

- Is called by the login component.
- Calls the login component.
- Gets user email and sends information to reset password.

Main Menu

- Is called by the login, high scores, map selection, manage account components, and gameplay components.
- Calls the login, high scores, map selection, and manage account components.
- Gets the button the user presses and calls correct component

High Scores

- Is called by the main menu component.
- Calls the main menu component.
- Gets all scores from database and displays them

Manage Account

- Is called by the main menu component.
- Calls the main menu component.
- Gets new information from user and updates database

Map Selection

- Is called by the main menu component.
- Calls the main menu component and the gameplay component.
- Gets the map the user selects and loads that map

Gameplay

- Is called by the map selection component.
- Calls the main menu, player, and enemy component.
- Gets locations of players and enemies and updates map accordingly

---

Player

- Is called by the gameplay component.
- Gets user control input and moves character/attacks accordingly

---

Enemy

- Is called by the gameplay component.

1.2

Incremental Testing Form

We followed a top-down incremental testing procedure for our app. Since we were building a game, we wanted to test each component from the start, as if we were users progressing through the app's UI and gameplay. The primary game states branched off from one central state (Menu), which made it easier to understand the flow of the app and which components interact with each other. Finally, the primary gameplay was tested in a similar fashion - the player and enemy components were called from the gameplay component. Due to the parent/child nature of our components, we decided that a top-down testing procedure would be the best choice for the app.

2.
Incremental and Regression Testing

Module: Login Component

Incremental Testing

Defect No.	Description	Severity	Solution
1	Empty credentials should not be valid	3	Implement input validity checks
2	Incorrect credentials should not be valid	3	Implement retrieval function to compare saved account data to input

Regression Testing

Defect No.	Description	Severity	Solution
1	Retrieving data from local file system throws an error	3	Implement socket data transfers for reading account info

Module: Sign Up Component

Incremental Testing

Defect No.	Description	Severity	Solution
1	Creating a new account should write the account info to the local system	3	Implement a file writer through node.js

Regression Testing

Defect	Description	Severity	Solution

No.			
1	Sending data to the local file system throws an error	3	Implement socket data transfers for writing account info

Module: Main Menu Component

Incremental Testing

Defect No.	Description	Severity	Solution
1	My Account button should be able to navigate to account	3	Edit code to make sure game states could change

Regression Testing

Defect No.	Description	Severity	Solution
1	Switching between states multiple times throws an error	3	Update a variable that had the same name as a function

Module: High Score Component

Incremental Testing

Defect No.	Description	Severity	Solution
1	The high score page should be scrollable	1	Implement arrow-key scrolling

Regression Testing

Defect No.	Description	Severity	Solution
1	The high score state	2	Implement button to

	should be able to navigate to the menu state		allow user to return to the menu

Module: Map Selection Component

Incremental Testing

Defect No.	Description	Severity	Solution
1	Carousel should be able to show map previews on left and right of focused map	2	Implemented left and right map previews
2	Carousel should be able to move left and right	3	Implemented left and right movement on mouse click

Regression Testing

Defect No.	Description	Severity	Solution
1	Switching between the Menu and Select states throws an error	3	

Module: Gameplay Component

Incremental Testing

Defect No.	Description	Severity	Solution
1	Entities should collide with the world bounds	1	Implement bounds collision for all entities
2	Camera should scroll with player	2	Attach camera scrolling to player movement controls

Regression Testing

Defect No.	Description	Severity	Solution
1	Exiting the current game session throws errors in the Menu	3	Update state resets and declarations

---

Module: Player Component

Incremental Testing

Defect No.	Description	Severity	Solution
1	Creating a separate player component should retain input controls from Gameplay component	2	Update Gameplay input handler to call the player component
2	Player melee attack sprite should appear in the direction clicked by the mouse	1	Implement directional attacks using the mouse pointer’s relative position and window quadrants
3	Player melee attacks should affect enemies	3	Implement collision detection between player attacks and enemies

Regression Testing

Defect No.	Description	Severity	Solution
1	Accessing player fields from outside its component throws an error	2	Update player object variable accesses in relevant components

Module: Enemy Component

Incremental Testing

Defect No.	Description	Severity	Solution
1	Game should have multiple individual enemies	1	Implement enemy prototype object (Mob) to allow multiple instances
2	Enemies should follow the player	2	Implement enemy movement based on player position
3	Enemies should collide with the player	1	Implement collision detection to stop the enemy movements

Regression Testing

Defect No.	Description	Severity	Solution
1	Enemies hit by the player should be removed from the game	2	Implement internal enemy function to correctly remove sprite from the game