Contents

[Section 3 - Development: 2](#_Toc152692962)

[Prototype 1 – Main Menu: 2](#_Toc152692963)

[Description: 2](#_Toc152692964)

[List of Success Criteria: 2](#_Toc152692965)

[Developing the prototype: 3](#_Toc152692966)

[Errors and solutions: 3](#_Toc152692967)

[Final Code and Output: 4](#_Toc152692968)

[Summary: 4](#_Toc152692969)

[Prototype 2 -: 4](#_Toc152692970)

[Description: 4](#_Toc152692971)

[List of Success Criteria: 4](#_Toc152692972)

[Errors and solutions: 4](#_Toc152692973)

[Final Code and Output: 4](#_Toc152692974)

[Summary: 4](#_Toc152692975)

[Prototype 3 -: 4](#_Toc152692976)

[Description: 4](#_Toc152692977)

[List of Success Criteria: 4](#_Toc152692978)

[Errors and solutions: 4](#_Toc152692979)

[Final Code and Output: 4](#_Toc152692980)

[Summary: 4](#_Toc152692981)

[Prototype 4 -: 4](#_Toc152692982)

[Description: 4](#_Toc152692983)

[List of Success Criteria: 4](#_Toc152692984)

[Errors and solutions: 4](#_Toc152692985)

[Final Code and Output: 4](#_Toc152692986)

[Summary: 4](#_Toc152692987)

# Section 3 - Development:

## Prototype 1 – Main Menu:

### Description:

For the first prototype, I will cover the development of the Main menu scene for the game. This will be useful in identifying potential errors, how I would fulfil the success criteria I have set for the main menu and what the impact is for stakeholders. The main menu will be the pivotal function of the game, which will allow for seamless navigation across different scenes, allowing them to customize keyboard and mouse inputs, graphics, and sound settings to meet the users’ requirements, select the game difficulty, learn more about how to play the game, load a previous game save or exit out of the game application.

When the user launches the game, the game will open in Fullscreen mode and the first scene that will be displayed is the main menu scene. The main menu scene has 5 buttons (Start Game, Load Game, Settings, About and Exit). This menu is meant to be easy for users to understand, enabling seamless navigation between scenes.

Using JavaFX allows for the user of FXML and CSS. FXML is a markup language used with JavaFX to define user interfaces, separating the structure of the UI from the application logic. CSS is employed in JavaFX to style the visual components defined in FXML, providing a more maintainable and visually appealing application. To make the development of the main menu GUI, SceneBuilder will be used. It allows developers to design JavaFX user interfaces graphically by providing a drag-and-drop interface for arranging UI components. SceneBuilder generates FXML code, which can be seamlessly integrated into JavaFX applications, streamlining the UI design process This allows for GUIs to be in the correct layout as required, without much coding, reducing development time. Additionally, LWJGL will be utilized for the main menu prototype. However, the main menu won’t utilize the system GPU as the main menu is not resource intensive on the computer, especially since the main menu doesn’t need to process more than one task simultaneously.

### List of Success Criteria:

|  |  |  |
| --- | --- | --- |
| Main Menu Success Criteria | Justification | |
| 1. Must display on a resizable window (640x480 to 1920x1080) or Fullscreen | Ensures usability and accessibility. Resizable window accommodates correct ratio display, and Fullscreen supports compatibility for all displays. | |
| 1. Start Game button to change from the Main Menu scene to the respective scene | Test button functionality to avoid unexpected scenarios. Critical for navigating through scenes in a project with multiple scenes. | |
| 1. Once the User presses the Start game button, they can select 5 difficulties | Enhances overall gaming experience by catering to different skill levels, providing a range of difficulty options. |
| 1. Load Game button to change from the Main Menu Scene to the game save selector scene | Enables seamless continuation of gaming progress, contributing to a user-friendly and enjoyable experience. |
| 1. Settings button to change from the Main Menu scene to the Settings Scene. The user can change their Graphics, Key binds, or sound settings | Enables users to customize their gaming environment, enhancing comfort during gameplay. |
| 1. Graphics settings scene | Offers control over visual aspects, ensuring smooth gameplay on various devices. |
| 1. Keybind setting | Allows users to personalize controls, contributing to a more intuitive gaming experience. |
| 1. Sound setting scene | Enhances immersion, accommodates players in different environments, and caters to those with specific preferences. |
| 1. About button to show information about how to play the game | Provides straightforward game instructions, reducing frustration, and improving enjoyment for players. |
| 1. Exit Button to close the game application | Offers a standard way for users to exit the game, ensuring a seamless conclusion to the gaming experience. |

### Developing the prototype:

First, I created my JavaFX project using the Maven build system archetype. Once I created the project, I had to add the LWJGL 3.3.3 library by adding the config dependencies to the project's pom.xml file. This will be used on a later prototype, but it is better to have all my libraries already incorporated into the project, so it doesn’t need to be added later.

Having incorporated all the required libraries into the project, I proceeded to generate essential files for creating a display window, including ‘**Game.java**’, ‘**MainMenu.java**’, ‘**MainMenuController.java**’, and ‘**MainMenu-view.fxml**’.

Each class has its dedicated function, which I had broken down to make changes to code much easier to understand when having to debug in later stages. Here is a brief description of what each file in the project does:

‘**Game.java**’ – Launches the application by calling the **MainMenu.java** class.

‘**MainMenu.java**’ – Launches the main menu scene by creating a display window and setting the main menu scene.

‘**MainMenuController.java**’ – Declares the main menu objects like buttons and labels. Each object can be given a dedicated function upon a user’s action.

‘**MainMenu-view.fxml**’ – Defines the structural layout of the main menu UI, specifying the arrangement of components like labels and buttons within the scene.

‘**styles.css**’ – Enhances the visual styling of the main menu, such as setting the background colour, text colour, and border styles for a cohesive and appealing design.

A screenshot of a computer

Description automatically generated

Here is the Hierarchical structure of the file system

After creating all the initial files for the project, the creating of the main menu can begin.

### Errors and solutions:

### Final Code and Output:

### Summary:

## Prototype 2 -:

### Description:

### List of Success Criteria:

### Errors and solutions:

### Final Code and Output:

### Summary:

## Prototype 3 -:

### Description:

### List of Success Criteria:

### Errors and solutions:

### Final Code and Output:

### Summary:

## Prototype 4 -:

### Description:

### List of Success Criteria:

### Errors and solutions:

### Final Code and Output:

### Summary: