Modular Complex System Brief

Assessment Task 1

System Purpose

The purpose of this project is to create a modular and flexible UI system in Unity, which can be easily customized, extended, and integrated into various projects. It will support different types of UI elements such as buttons, images, text fields, sliders and toggles. It will handle different screen resolutions and aspect ratios to ensure proper scaling and positioning of UI elements. The UI system will provide an intuitive and customisable layout system to allow the arrangement of UI elements in a visually appealing manner. The system will be optimised for performance to ensure a smooth user experience.

Libraries

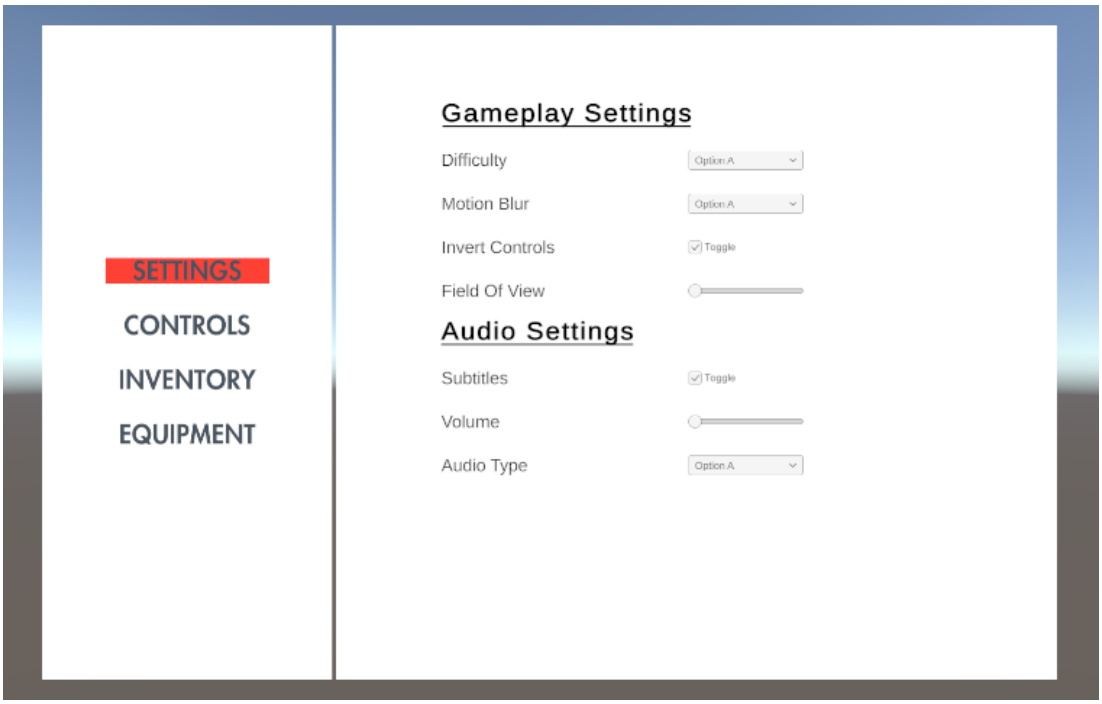
- The UI system will be based on Unity's UI framework, utilizing canvas, panels, and UI components.
- LeanTween will be used to animate various UI elements like pulsing buttons on click and easing in panels
- Depending on time I will implement some particle assets, character assets and other visual components to dress up the UI and possibly have a basic level.

Canvas Structure

For this project I won't be using very many complex mathematical operations and advanced algorithms. The canvas will be structured in a way that utilizes various Horizontal/Vertical Layout Groups, Layout elements, Content Size Fitters and Scroll Rect's. These components will work together to keep the Menu's tabs and content in the correct position and allow for the addition and removal of options. Once the designated screen space fills up the scrolling functionality will allow for the user to easily scroll through all the content they add to the menu. Depending on time, I may try and create my own layout group component with more functionality that could allow the menu to be better customised to the individual's liking or requirements.

Redistribution

The modular complex system I am building will be distributed as a Unity asset package, allowing someone to easily download it to their project and create a simple menu in a timely manner. Will provide basic instructions in a "read me" file to explain how it works.



Modular Design

The menu will be a scriptable object so the user can create different menus with a consistent style, but also give the designer options to customise things like the buttons, fonts, headers, layout functionality and include their own prefabs if they so choose. The scriptable object will be designed to create settings menus, gameplay controls, drag and drop inventories and a battle pass. Apart from these basic use cases the functionality provided should be able to create different UI options it wasn't directly intended for. I would like to implement a custom editor for this component so I can logically display all the functionality of the scriptable object and how it works. Tool tips and correct labelling of all the variables will be essential as the design of this scriptable object requires lots of lists inside lists to store all the required data. This gets messy and difficult to follow in the Unity editor.

How to Integrate

I will save the canvas I setup as a prefab so it can be duplicated, and multiple menus can be created then switched out in the game. The canvas will be set out with two separate sides.

Tab/button side - this side will hold all the buttons or tabs which are Instantiated at runtime. The tab prefab I have provided has a script that changes colour on selection and animates (using LeanTween) when you select it. The tab button script also takes the name of the tab from the scriptable object. The user could create their own button prefab, but they would need include the script to get the correct name allocated to it. This section will be scrollable if the number of buttons exceed the screen space I have allocated. The spacing between buttons and view space can be edited in the inspector if needed.

Panel/Content side – this side will hold all the content of each selected tab and is scrollable if the amount of content fills up the space. All the content data from the scriptable object is spawned at runtime and change panels depending on tab selection. I plan to do all this searching and assigning of data through for loops and if statements, but I would like to implement a better method if I can find a way to make it happen. I will continue to research and talk to others about how this system could be improved as I develop it further.

The "UIScriptableObject" will be where I obtain all the user data for the Menu being created. Most of the data will be stored in Lists which are arranged in a tier system. So, each tab will have a list of headers to be displayed and each header will have a list of options under it. These options can have 'settings' features attached to them like drop down menus, toggles and sliders. They can also have attached text boxes for additional information in the "Controls" menu type. Another option will be to implement a drag and drop UI with customisable amounts of slots. Finally, the last option will be a battle pass structure that will incorporate text and attached images being spawned in the content area.

The "Tab Group" Script will do all the work reading through the data and assigning the correct options and values in the correct location. It must remain on the "Tabs" game object in canvas to operate correctly. Lastly, I will develop the functionality to save all settings and data that requires saving so it doesn't reset each time the menu is created on play.