ALPHA\_RCRacing / Menus & UI

Architecture/Design Document

**Table of Contents**

1 Introduction.. 3

2 Design Goals.. 3

3 System Behavior.. 3

4 Logical View... 3

4.1 High-Level Design (Architecture of the Entire system)… page 4

4.2 Mid-Level Design of the Menus & UI Module... page 5

4.3 Detailed Class Design of the Menus & UI Module… page 6

5 Process View of the Menus & UI Module... page 7-9

6 Use Case View... page 10

Change History

**Version:** 0.1

**Modifier:** Alexander Achorn

**Date:** 03 / 18 / 2022

**Description of Change:** Module Design Document started.

**Version:** 0.2

**Modifier:** Alexander Achorn

**Date:** 04 / 16 / 2022

**Description of Change:** Added Networking Changes

1. **Introduction**

This document describes the architecture and design for ALPHA\_RCRacing, a game being developed by Inertial Sketch. ALPHA\_RCRacing is a Multiplayer RC Car Game where player compete against each other in a competitive racing track filled with jumps, turns and Power Ups.

The purpose of this document is to describe the architecture and design of the Menus & UI Module application in a way that addresses the interests and concerns of all major stakeholders. For this application the major stakeholders are:

* Developers;
* Project Manager.

1. **Design Goals**

The design priorities for the Menus & UI system are:

* The design should result in a menu that is easy to navigate.
* The design should allow for the reuse of common widgets.
* The design should be easily understood and adjusted by all team members

1. **System Behavior**

The Menus & UI module is built from a single HUD Class InGameUI and a series of Widgets:

* MainMenu
* PauseMenu
* VolumeMenu
* UsernameMenu
* ControlsMenu
* LocalRaceMenu
  + NetworkRaceMenu (inherits from LocalRaceMenu)
  + NetworkMapMenu (is another NetworkRaceMenu)
* FinishScreen

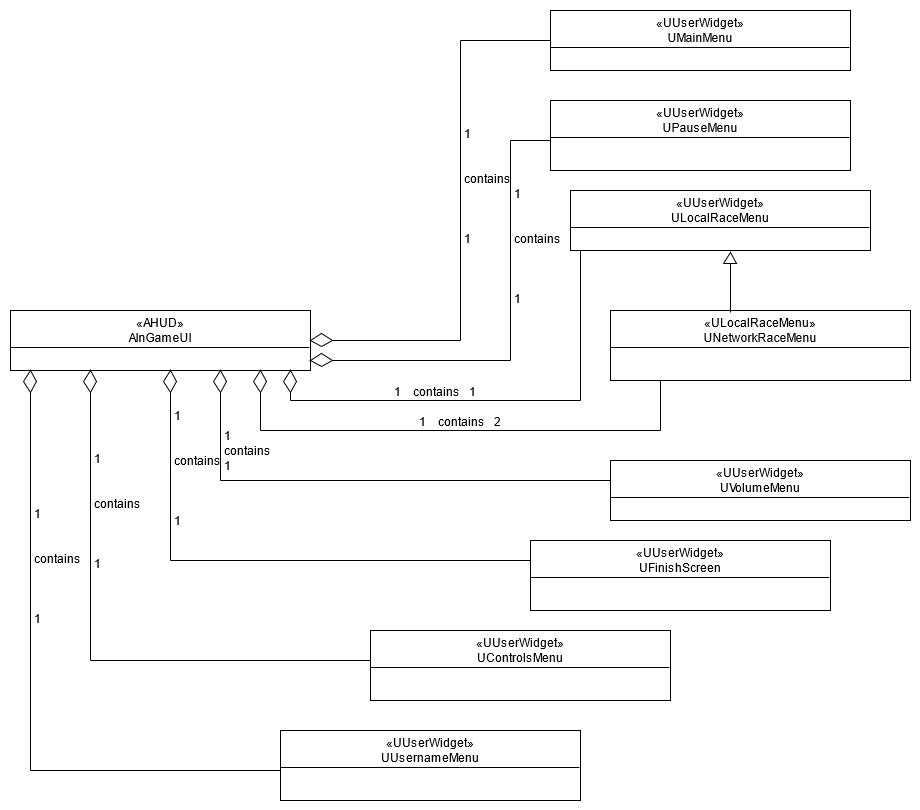
This will allow for the single master HUD class to add and remove the various UI elements from the Viewport. This also makes it far easier to reuse Widgets like the Volume, Controls and Username at any point.

1. **Logical View**

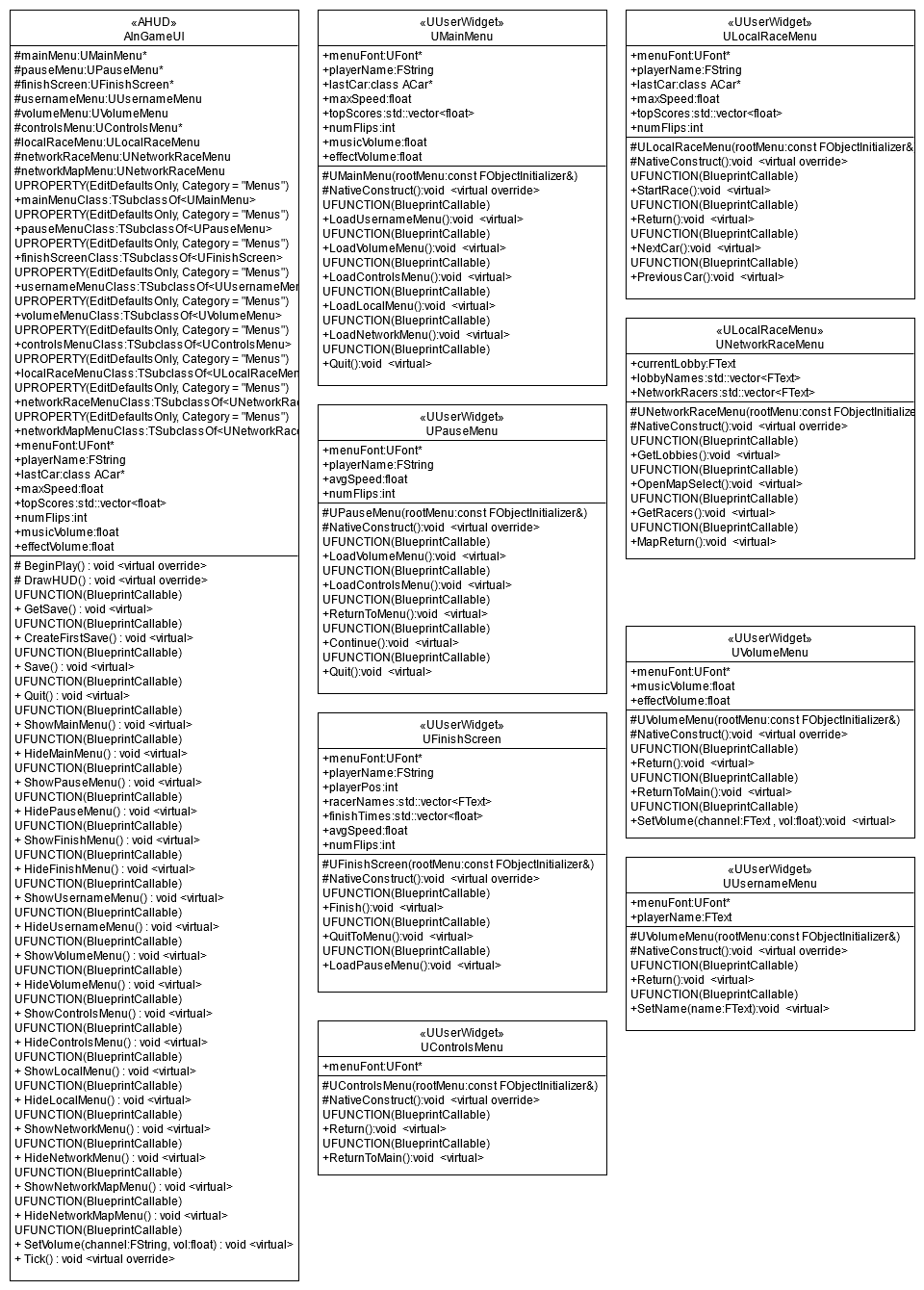
The logical view describes the main shared components, attributes and switches of the system. This includes modules, the static relationships between modules, and their dynamic patterns of interaction.

In this section the modules of the system are first expressed from a macro perspective and progressively goes to a micro perspective to view the detailed sequences and components.

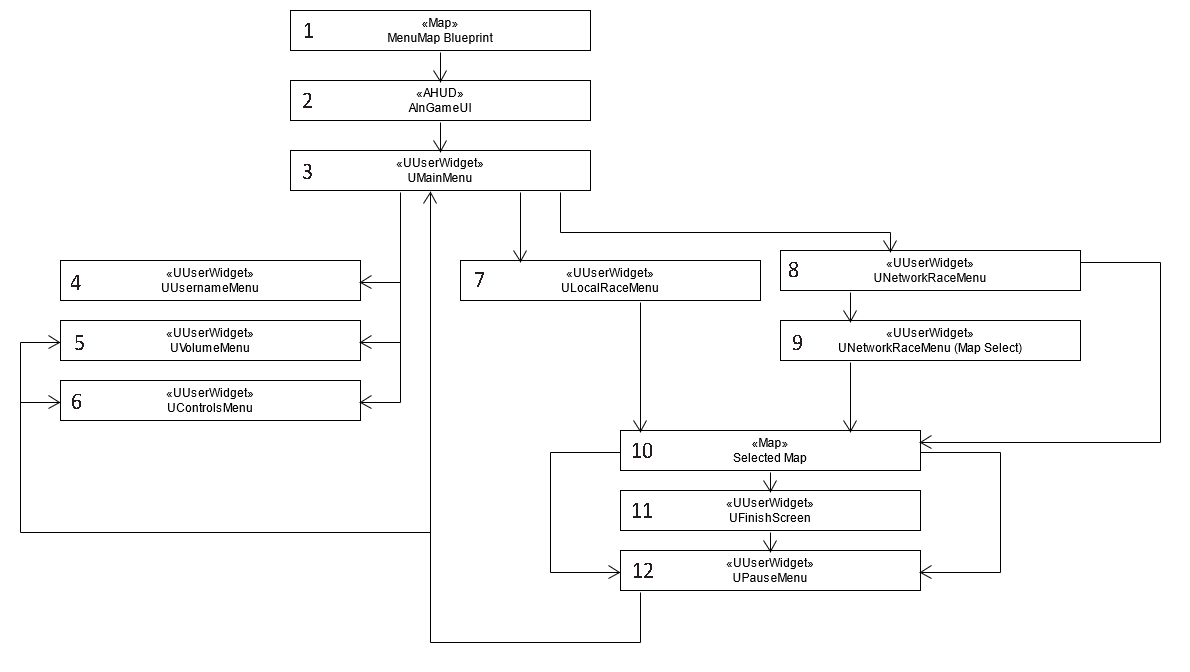
* 1. **High-Level Design (Architecture of the Entire system)**
* RacingPawn System is the main system and handles the car movements, current power up and various effects.
* PowerUp System knows its mesh and collision sphere as well as its cooldown and despawn timer. Its children have unique effects.
* UI System is used to indicate crucial information like the player’s current power up.
* Menu System handles the selection of car, map and handles the race creation.
* InputComponents system handles the controls for PC, Console and VR.
  1. **Mid-Level Design of the Menus & UI Module**



* 1. **Detailed Class Design of the Menus & UI Module**



1. **Process View of the Menus & UI Module**

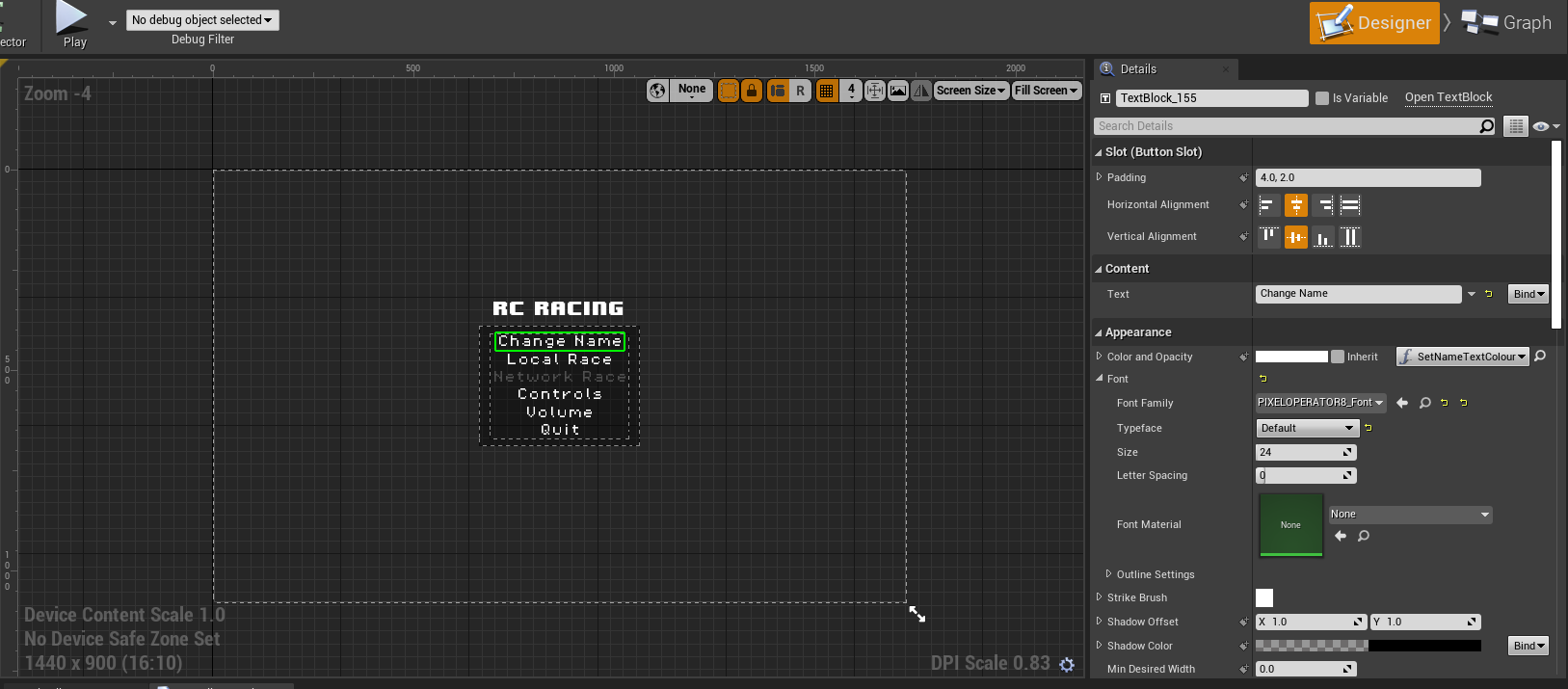
****

The Menu sequence begins at the start of play when the InGameUI class is initialized and the MenuMap is loaded. Next the MenuMap will call ShowMainMenu to open the Main Menu. The Main Menu will check for a saved file and create on if one does not exist. Once the Main Menu is loaded the next part of the sequence is up to the user. They can open any of the submenus. If they proceed to the Local or Network Race menus they will be given the option of which car and map to use. After selecting a map all menus will be hidden. From a map a player can hit pause to open the Pause Menu, or Finish the race to load the Finish Screen. The finish screen can be followed back to the Pause Menu which can proceed back to the Main Menu or to the Volume and Controls submenus.

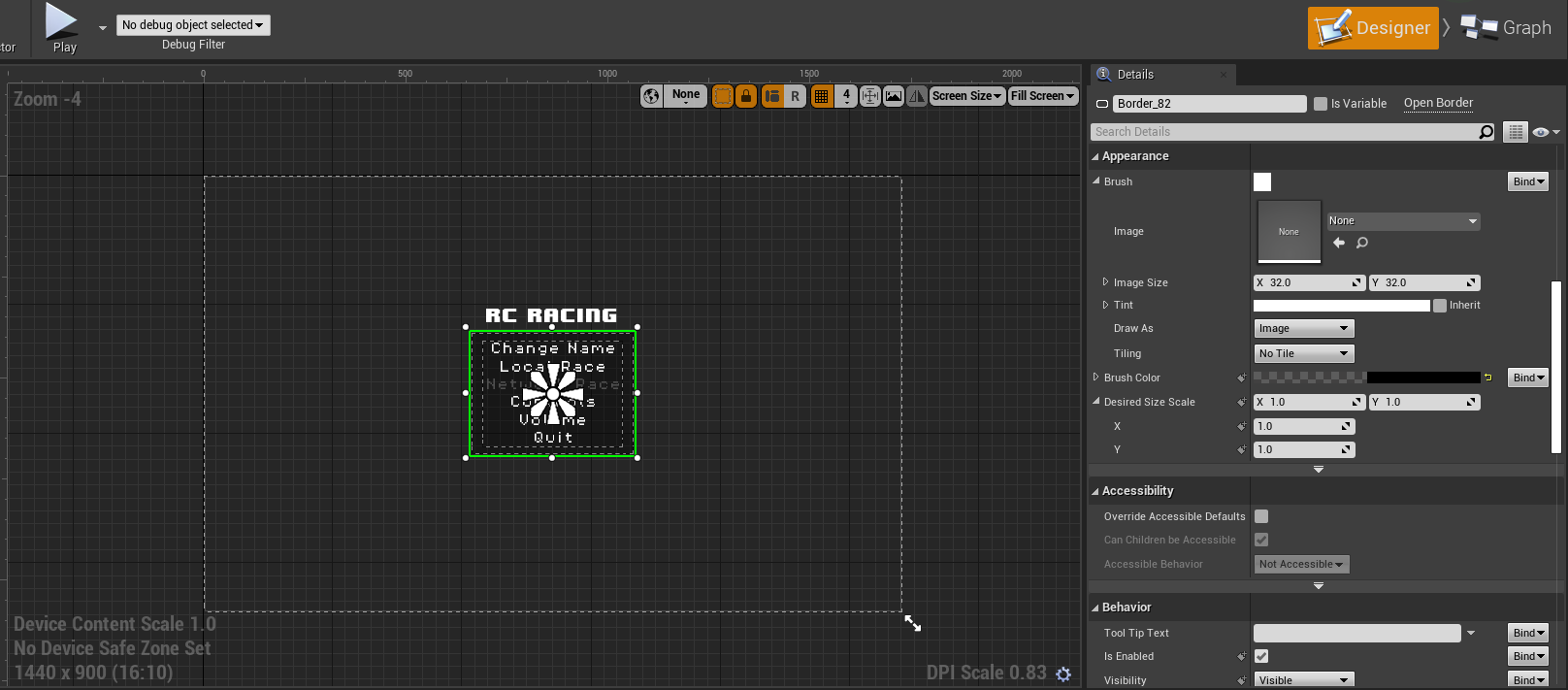
Changing Menu Text Highlights

A screenshot of a video game

Description automatically generated

Editing Menu Text 

Changing Panel Backgrounds



1. **Use Case View**

For this release, all Menus & UI only work with a mouse. In future releases, keyboard and gamepad support should be implemented.

The Menu is designed to facilitate the navigation of local and network races. It’s primary goal being to be easy to use and to save common features like Username, Volume Settings, and last picked car. The widgets for doing this will each use the InGameUI class to call methods to save these settings to a json or other style of save file. Each Widget is configured with Hover, Press, and Release effects to change the text colour to contrast when using buttons. Each also has a Click effect and all these effects are done in blueprint for ease of use by all team members.

The Main menu is loaded at startup when the MenuMap is left as default. The pause menu can be brought up at any time while in a map by pressing backspace or escape if not using the editor.