Highway Racer 2

Thank you for purchasing **Highway Racer 2**. This documentation will guide you through the process of adding, modifying, and customizing various aspects of the game, including player vehicles, traffic vehicles, levels, roads, scoring systems, multipliers, game modes, menus, and more.

Package Contains

12 player vehicles, 5 traffic vehicles, 1 main menu scene, and 3 gameplay scenes

Desert, town, and country environment

User friendly editor scripts and editor windows to create & use your own content

Scene managers and observers

Photon integration

All necessary scripts, models, materials, textures, sound FX, etc...

Video Tutorial Series for Creating & Configurating Player Cars, Traffic Cars, New Levels, Edit Prices, etc...

You can access this tutorial series from **this** link. I'll keep the playlist updated.

Leading Features

Endless Curved Road Management: Unlike the original, Highway Racer 2 supports endless curved roads, adding a new level of challenge and excitement to your racing experience (You can still use the straight roads if you wish).

Enhanced Vehicle Control: Much more stable and responsive vehicle handling, designed for faster reactions and smoother gameplay.

Advanced Vehicle Customization: Deeper customization options, allowing for more detailed modifications to your vehicles.

Improved Traffic Vehicles: More stable traffic system, ensuring a seamless and immersive gameplay experience.

Customizable Vehicle Camera: Adjust the vehicle camera settings to your liking, with more options for fine-tuning your perspective.

New Environments and Roads: Completely redesigned roads with new environments including the main menu scene.

New Editor Features: Much more faster scene setups in a few seconds with HR_SceneManager that allows you to create brand new scenes for main menu and gameplay menu.

Object Placement on Curved Roads: You're not restricted to use demo assets comes with the asset. You can use your own content on the roads and reskin the game easily.

Optimization: Improved optimization on the traffic vehicles and road management, ensuring a smooth performance across all supported platforms.

Realistic Car Controller Pro Integration: Powered by the latest version (V1.52.1) of Realistic Car Controller Pro, included for free with Highway Racer 2.

Platform Compatibility: Fully tested and ready to deploy on PC, Mac, Linux, Android, iOS, WebGL.

Game Modes: Choose from three different game modes, including FPS, TPS, and top camera perspectives.

Lane-Switching Traffic: Dynamic traffic with lane-switching capabilities.

Vehicle Unlocks and Upgrades: Unlock and upgrade cars, with easy editor configuration for prices and upgrades.

Comprehensive Editor Tools: Take advantage of user-friendly editor scripts and windows to create and manage your content.

Photon Integration Addons: Engage in 1 V 1 overtaking races with full Photon integration for a competitive multiplayer mode.

Detailed Documentation: Access full PDF documentation, covering vehicle and level creation, optimization, menu configuration, and Photon integration.

If you're enjoying Highway Racer 2, we would greatly appreciate it if you could take a moment to leave a review. Your feedback helps us improve and reach more users.

For any issues or support requests, please contact me directly via email before leaving a review. I'm committed to resolving any problems as quickly as possible to ensure the best experience for you.

Before Releasing Your Game and About Intellectual Property Violating

- **Originality Matters**: Avoid using the same screenshots that appear on the Asset Store page.
- **Unique Descriptions**: Craft your own game description rather than using the one provided on the Asset Store page.
- **Custom Branding**: Refrain from using the logo included with the package; create a unique game name and icon.
- **Personalized Reskin**: It's important to reskin the game with your own content. Avoid using the same assets provided in the package to maintain originality. The demo assets included are provided to help you understand how the system works in the simplest way possible.

Before Importing the Package

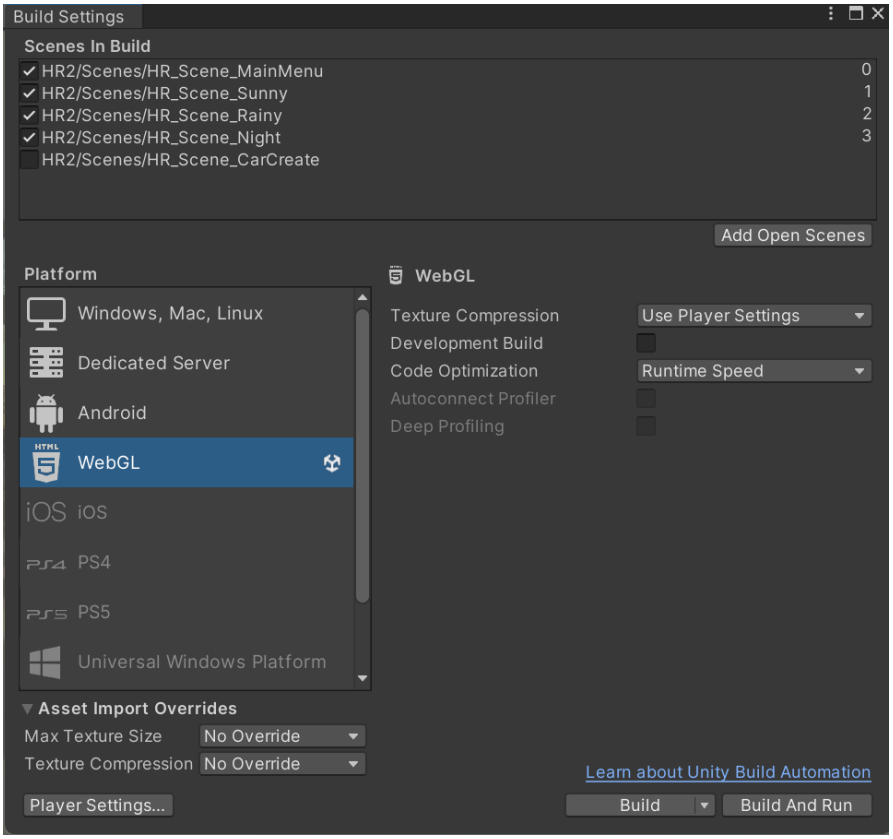
This project was developed using **Unity 2022.3.15f1**. To ensure full compatibility, please use Unity **2022.3.15f1** or a more recent version. I strongly recommend using the latest **Long-Term Support (LTS)** version of Unity for stability and reliability. Avoid using beta or the very latest non-LTS versions, as they may introduce unexpected issues.

Before integrating this package, ensure that you are working in a new project without any pre-existing settings that could conflict with the package.

After Importing the Package

Check for Errors: After importing the package, check your console for any errors. If you encounter an "unreferenced script" error, this likely indicates a corrupted import. In this case, delete the imported files and re-import the package. Note that successful imports occur in 99% of cases.

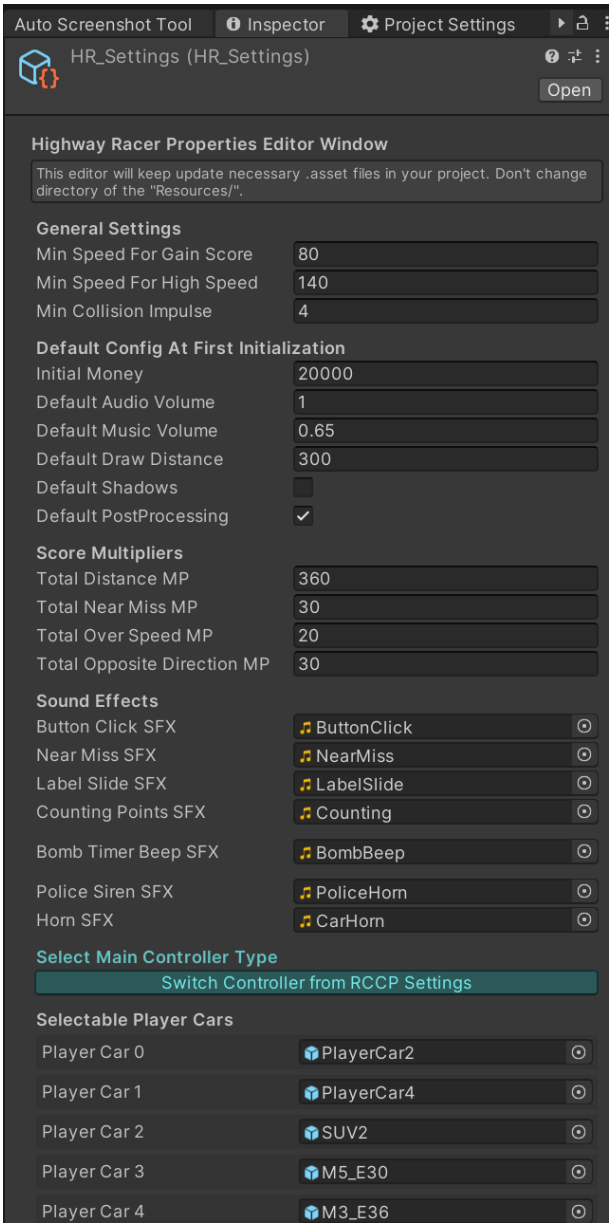
Review Build Settings: Navigate to **File → Build Settings** to review your build settings. Ensure that your scenes, build target, and player settings are correctly configured. The package comes with default settings, but it's important to verify that they align with your project's requirements.



Before making any modifications, hit play and thoroughly test the game. Ensure that every mode, every car, and every upgrade functions as expected. It's crucial to confirm that your project is working correctly before making any changes.

HR_Settings (General Settings)

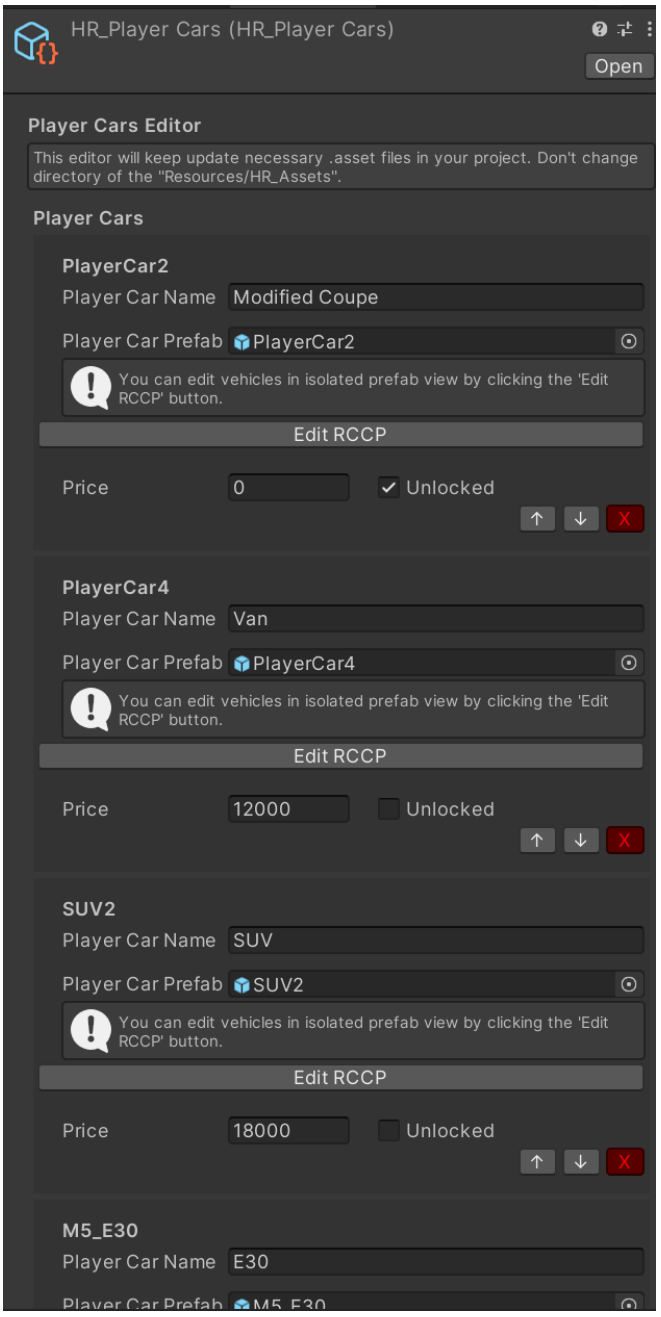
All global settings are managed through a single editor script. You can access these settings by navigating to **Highway Racer → HR_Settings** in the top bar menu.



This panel allows you to customize various aspects of the package without needing to modify any scripts. For testing purposes, you can set the "**Initial Money**" to higher value. Just remember to disable it before publishing or releasing your game.

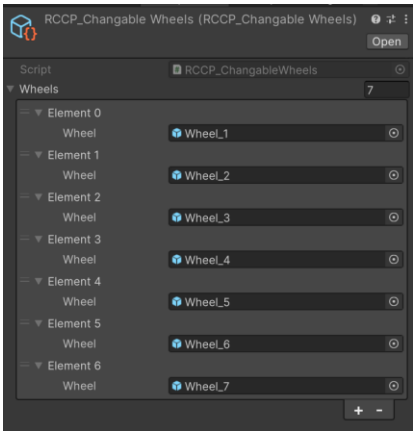
Configure Player Cars

To edit the current player cars and their prices, click on “**Configure Player Cars**.” You can access this feature either from the General Settings panel or directly from the toolbar by navigating to **Highway Racer → Configure Player Cars**.



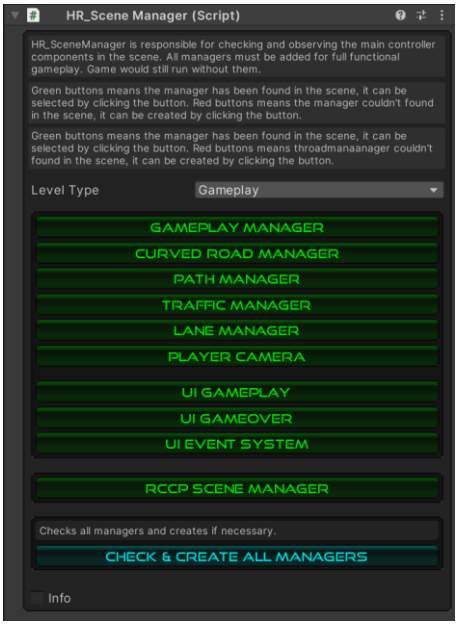
Configure Upgradable Wheels

To edit the current upgradable wheels and their prices, click on "**Configure Upgradable Wheels**." These wheels are essentially models, so ensure that your custom models have the same sizes, with correct pivots and axes, just like the wheels provided in the package. You can access "**Configure Upgradable Wheels**" from the **General Settings** panel or directly from the toolbar by navigating to **Highway Racer → Configure Upgradable Wheels**.

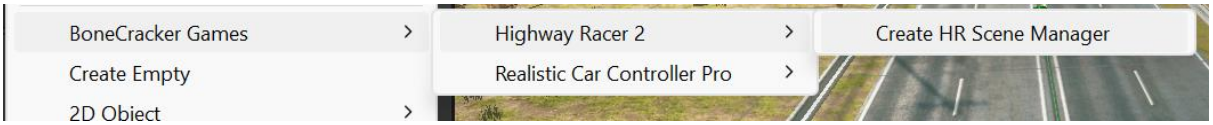


HR_SceneManager

The **HR_SceneManager** serves as the main manager for the scene and can be utilized in both the main menu and gameplay scenes. By selecting the level type, it will automatically observe and manage all submanagers within the scene. The **HR_SceneManager** will notify you of the scene setup and any missing managers. If a submanager is missing, clicking on it will either select or create it automatically.



You can create the **HR_SceneManager** by right-clicking in the hierarchy panel and navigating to **BCG → HR2 → Create HR SceneManager**.



Adding / Editing Player Cars

All player cars in the game are controlled by **Realistic Car Controller Pro**. If you want to add your own car, you must set it up according to the Realistic Car Controller Pro documentation. Additionally, you can watch the tutorial video for step-by-step instructions on how to add a new vehicle.

Highway Racer 2 Create Player Vehicle Tutorial Video

In the Scenes folder, you'll find a scene named "**HR_Scene_CarCreate**". Open this scene to build your car. Once you're satisfied with the results, you can add the car to your game.

To add your car to the game, follow these four simple steps:

1. **Create Your Car Controller**: Set up your car according to the **Realistic Car Controller Pro** (RCCP) documentation.

2. **Add the HR_Player Script**: The **HR_Player** script will be automatically added to your car.

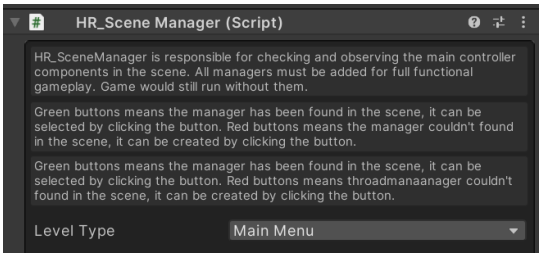
3. **Create and Save the Prefab**: Use the button on the **HR_Player** script to create and save a prefab of your car by dragging the root object into the "**Prefabs/PlayerCars**" folder.

4. **Add the Prefab to the Player Cars List**: Use the button on the **HR_Player** script to add your prefab to the player cars list.

Adding New Scenes

All the necessary handlers can be found in the "**Scenes / Scene Setups**" folder. For a quick setup, you can simply drag and drop them into your scene. However, creating a new scene and manually placing these handlers is generally unnecessary.

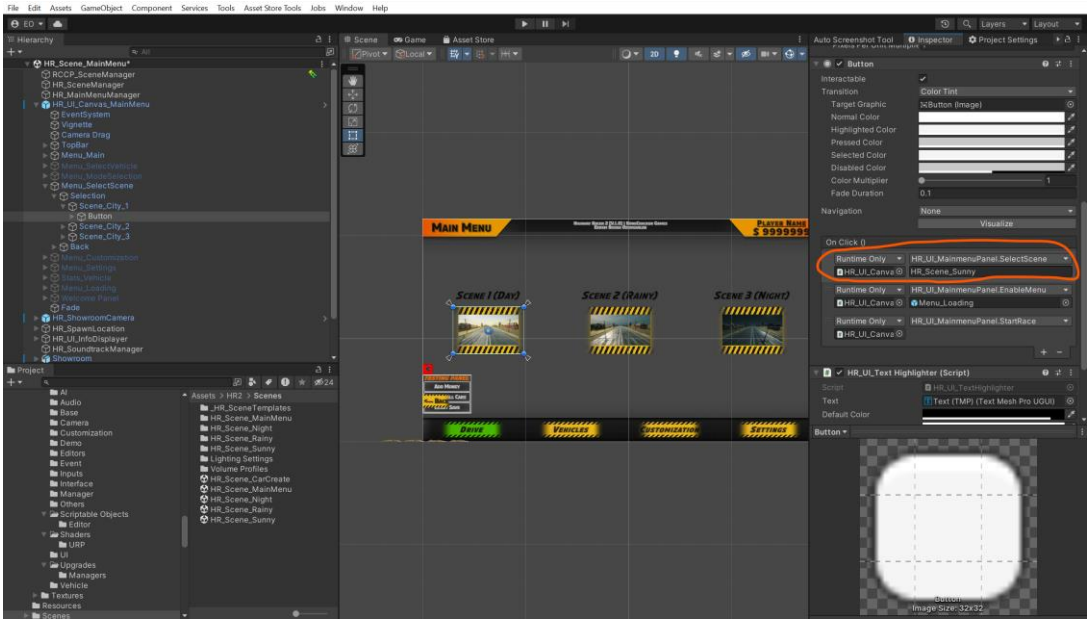
The **HR_SceneManager** in your scene will handle this for you. Just add the **HR_SceneManager** to your scene, and it will automatically create the necessary submanagers within a few seconds.



The main menu and gameplay scenes use different submanagers, so it's important to select the correct level type in the **HR_SceneManager**. This ensures that the appropriate submanagers are activated for each scene.

UI Scene Selection Buttons in the Main Menu Canvas

The UI buttons on the main menu are configured to use scene names. When a player selects a level, the UI button triggers a method in the manager script to save the scene selection. To ensure everything works correctly, double-check that each UI button is assigned the correct target scene name. If the scene name is incorrect, the button may load the wrong scene.

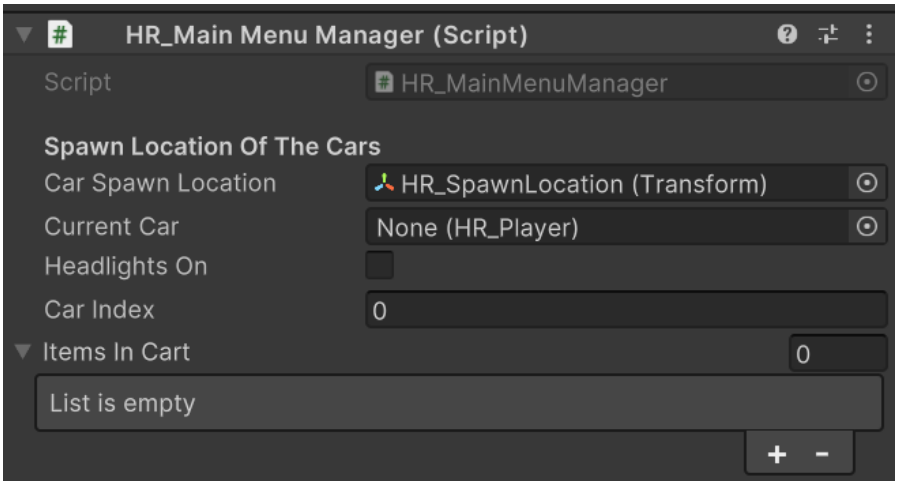


How the Main Menu Scene Works?

The **HR_MainMenuManager** in the main menu scene handles all aspects of scene management. This manager is responsible for a wide range of tasks, including:

- Spawning, enabling, and disabling player vehicles
- Handling vehicle purchases, locking, and unlocking
- Saving and loading the selected scene
- Managing items in the cart

The **HR_MainMenuManager** ensures smooth operation of the main menu and its interactions with other game elements.

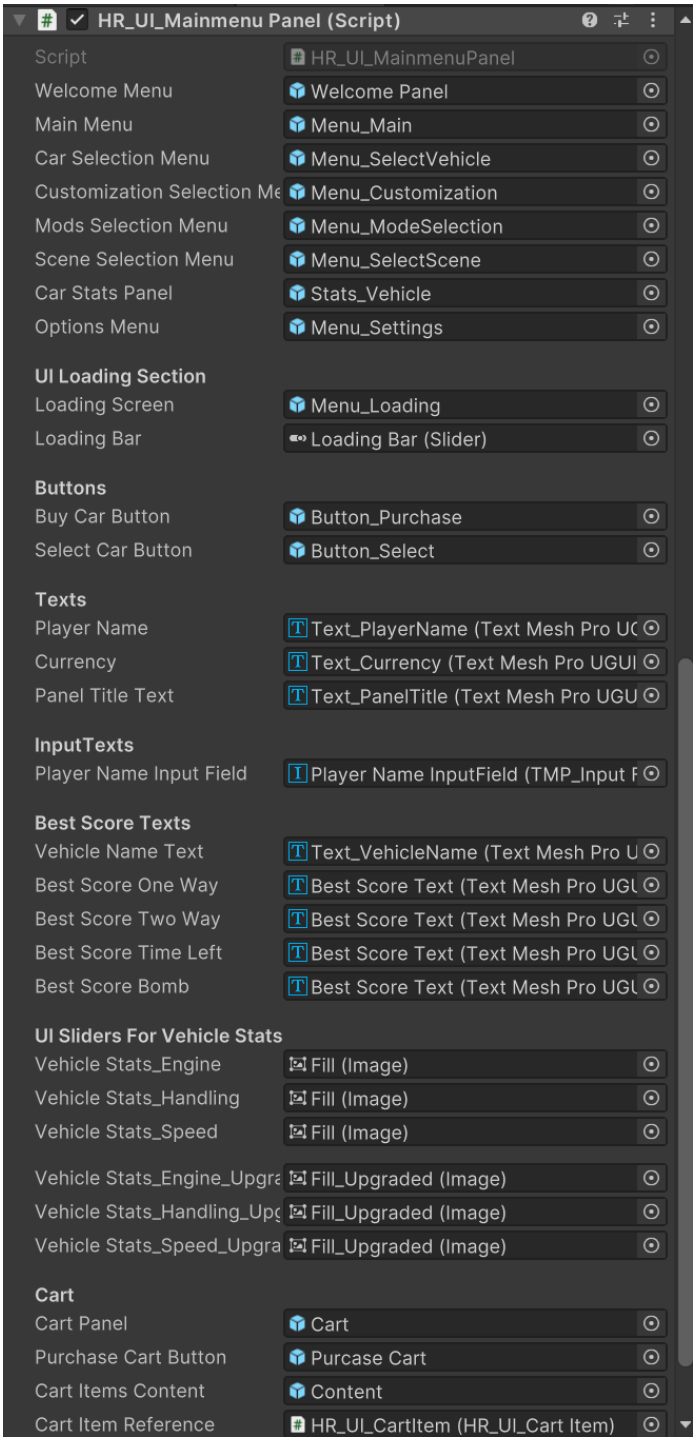


The **HR_UI_Canvas_MainMenu** in the main menu is the central hub for all UI elements, including buttons, sliders, toggles, and dropdowns. These elements are linked to methods within the **HR_UI_MainmenuPanel** script, which is attached to the **HR_UI_Canvas_MainMenu**.

When a player interacts with any UI element, the **HR_UI_Canvas_MainMenu** triggers the corresponding method in the **HR_UI_MainmenuPanel** script. This script then communicates with the **HR_MainMenuManager** by calling the appropriate methods.

It's important to note that UI elements do not directly call methods in the **HR_MainMenuManager**. Instead, all interactions are routed through the **HR_UI_MainmenuPanel** for proper handling and management.

For example, when a player selects a vehicle, the action first triggers a method in the **HR_UI_MainmenuPanel** script. The **HR_UI_Canvas_MainMenu** then relays the command to the **HR_MainMenuManager**, which handles the specific task.



The main menu scene also includes the **HR_ShowroomCamera**, which orbits around the player vehicle. Players can interact with the camera to rotate and view the vehicle from different angles, enhancing the showroom experience.

How the Gameplay Scenes Work?

Gameplay scene management is handled by the **HR_GameplayManager**. This manager is responsible for various critical functions, including:

- Spawning player vehicles
- Setting audio volume
- Applying game modes
- Listening to player events

In addition to the **HR_GameplayManager** script, gameplay scenes also include the following managers:

- **HR_CurvedRoadManager**: Creates and manages the curved roads within the scene.
- **HR_PathManager**: Calculates the navigation paths used by both player and traffic vehicles.
- **HR_LaneManager**: Creates and manages the lanes associated with the main path, which are utilized by player and traffic vehicles.

- **HR_TrafficManager**: Creates and manages the traffic vehicles throughout the scene.

These managers work together to ensure a smooth and dynamic gameplay experience.

Optimization On Levels

All models in the game are designed with a minimal number of materials to maximize efficiency. The scenes utilize pooled traffic cars and roads, ensuring that resources are used efficiently, and performance remains high.

Additionally, none of the traffic car lights rely on pixel lighting, which helps to further optimize performance. These lights are used solely for visual effects like halos and lens flares, contributing to a visually appealing experience without compromising on performance.

Local Saved Files

Highway Racer 2 (HR2) uses the **PlayerPrefs** registry data to save various player-related information, including settings, cars, upgrades, high scores, and achievements. This ensures that player progress and customizations are preserved across sessions.

HR_API

Highway Racer 2 (HR2) provides a comprehensive API for managing various game functionalities, including currency management, vehicle unlocking, high scores, player options, and more. Below is a summary of the available API methods:

Highway Racer 2 (HR2) provides a comprehensive API for managing various game functionalities, including currency management, vehicle unlocking, high scores, player options, and more. Below is a summary of the available API methods:

Currency Management

- **Get Current Currency:** HR_API.GetCurrency()
  Retrieves the player's current money balance.

- **Add Currency:** HR_API.AddCurrency(int add)
  Increases the player's money by the specified amount.

- **Consume Currency:** HR_API.ConsumeCurrency(int consume)
  Decreases the player's money by the specified amount.

Vehicle Management

- **Get Unlocked Vehicles:** HR_API.UnlockedVehicles()
  Returns a list of owned vehicles as an integer list.

- **Lock Vehicle:** HR_API.LockVehicle(int index)
  Locks the specified vehicle.

- **Unlock Vehicle:** HR_API.UnlockVehicle(int index)
  Unlocks the specified vehicle.

- **Check Vehicle Ownership:** HR_API.OwnedVehicle(int index)
  Checks if the specified vehicle is owned by the player.

High Scores

- **Get High Scores:** int[] scores = HR_API.GetHighScores()
  Retrieves high scores as an integer array.

Player Information

- **Set Player Name**: HR_API.SetPlayerName(string newPlayerName)
  Sets the player's name.

- **Get Player Name**: HR_API.GetPlayerName()
  Retrieves the player's name as a string.

- **Check First Gameplay**: HR_API.IsFirstGameplay()
  Checks if this is the player's first gameplay session.

Scene Management

- **Load Main Menu**: HR_API.MainMenu()
  Loads the main menu scene.

- **Restart Scene**: HR_API.Restart()
  Restarts the current scene.

Audio and Visual Settings

- **Set Master Audio Volume**: HR_API.SetAudioVolume(float volume)
  Sets the master audio volume.

- **Get Master Audio Volume**: HR_API.GetAudioVolume()
  Retrieves the master audio volume.

- **Set Music Volume**: HR_API.SetMusicVolume(float volume)
  Sets the music audio volume.

- **Get Music Volume**: HR_API.GetMusicVolume()
  Retrieves the music audio volume.

- **Set Draw Distance**: HR_API.SetDrawDistance(float distance)
  Sets the maximum draw distance for the game.

- **Get Draw Distance**: HR_API.GetDragDistance()
  Retrieves the maximum draw distance.

Gameplay Statistics

- **Set Total Played Time:** HR_API.SetTotalPlayedTime()
  Saves the total played time.

- **Get Total Played Time:** HR_API.GetTotalPlayedTime()
  Retrieves the total played time.

Mobile Controller Settings

- **Set Controller Type:** HR_API.SetControllerType(int controllerIndex)
  Sets the mobile controller type.

- **Get Controller Type:** HR_API.GetControllerType()
  Retrieves the mobile controller type.

Graphics Settings

- **Toggle Shadows:** HR_API.SetShadows()
  Toggles shadows on or off.

- **Get Shadows Option:** HR_API.GetShadows()
  Retrieves the current shadows option.

- **Toggle Post Processing:** HR_API.SetPP()
  Toggles post-processing effects on or off.

- **Get Post Processing Option:** HR_API.GetPP()
  Retrieves the current post-processing option.

Game Reset

- **Reset Game:** HR_API.ResetGame()
  Resets the game and deletes all saved data.

Contact

If you need any assistance, feel free to contact me at **BoneCrackerGames@gmail.com**.
Please include your invoice number in the email for faster support.