Car UI Complete Pack - Unity Asset

Description

Car UI Complete Pack is a dedicated package for mobile car games, providing essential UI elements to enhance user experience.

It includes a wide selection of pre-made buttons, virtual steering wheel, parking sensors, a GPS UI, Performance Tracker and a speedometer, all ready for use and customization.

What the package includes:

- Sprites: Essential images for various UI elements, such as buttons, icons, and more.

- Prefabs: Pre-made buttons, steering wheels, parking sensors, GPS, and speedometer, ready to be used in your scenes.

- Demo Scene: A demonstration scene showcasing how the UI elements can be used in a game context.

- Scripts: Example scripts illustrating how the package works and how to integrate UI elements into your game.

** The scripts are provided only as examples for correct usage; modifications can be made to adapt them to your game logic.*

Installation Instructions

1. Importing the Package into Unity:

- Open your Unity project.

- Go to Window > Package Manager > Car UI Complete Pack > Download

- In the pop-up window, check all the elements you want to import (usually, all are selected by default) and click Import

2. Adding UI Elements to Your Project:

- After import, the UI elements will appear under the Assets folder in Car UI Complete Pack.

- Drag the desired prefabs into your scene to add and position them as needed.

- Use the demo scene to see how elements are organized and how they work in context.

3. Configuring Scripts:

- Example scripts are already set up to illustrate UI functionality.

- If you want to integrate them into your game logic, you can modify them as needed.

- Ensure that you have correctly set references to objects in the scene (such as buttons, steering wheels, sensors, GPS, and speedometer).

4. Testing:

- After adding elements to your scene, click Play to test interactions with buttons, the steering wheel, GPS, speedometer and parking sensors.

- Adjust their sizes, positions, colors and behaviors according to your game's requirements.

Usage Guide

1. Adding Prefabs to the Scene:

- After importing the package, find the pre-made buttons in CarGame Pack UI/Prefabs.

- Drag a button into the scene from the Assets folder.

- Each button is pre-configured with a UI Button component, making it functional immediately.

- Customize the button's size, style, and colors in the Inspector.

- Link the button to the corresponding functions in your game (e.g., accelerating or braking the car).

2. Adding a Virtual Steering Wheel:

- You can find steering wheel in Car UI Complete Pack/Prefabs/Steering Wheel.

- Drag into your scene to add it.

- Steering wheel is set up with a script that controls rotation based on the player's movements.

- If you want to integrate it into a vehicle control system, link the steering wheel script to the vehicle control script.

- Customize the steering wheel size and position according to your game's needs.

3. Configuring Parking Sensors:

- Parking sensors are included in Car UI Complete Pack/Prefabs/Parking Sensor.

- Place the sensors near the car to simulate a parking warning system.

- Each sensor is configured to display visual feedback when the vehicle gets too close to an obstacle.

- Adjust sensor behavior by modifying the parameters in the associated script.

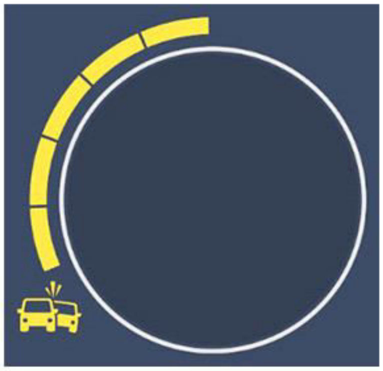
4. GPS UI:

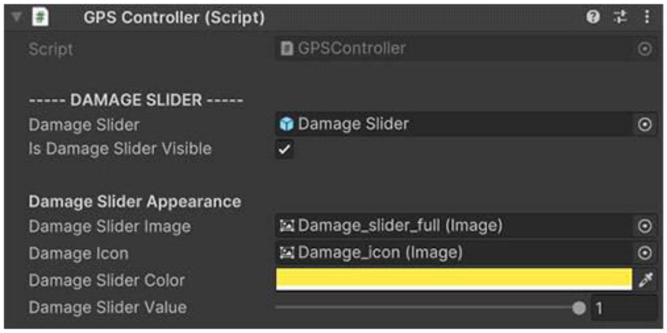
- The GPS UI is included in Car UI Complete Pack/Prefabs/GPS.

- Drag the GPS prefab into your.

- Configure the elements in the Inspector.

**They must be configured while the code is not running, so that they remain saved.*





GPS Controller (Script)

Script	GPSController
----- DAMAGE SLIDER -----	
Damage Slider	Damage Slider
Is Damage Slider Visible	✓
Damage Slider Appearance	
Damage Slider Image	Damage_slider_full (Image)
Damage Icon	Damage_icon (Image)
Damage Slider Color	🟨
Damage Slider Value	1

Step 1: Declare Variables for Components

```
private GPSController gpsController;
```

Step 2: Find the Components in the Scene

```
gpsController = FindObjectOfType<GPSController>();
```

Step 3: Initialize and Use the Components

```
// Set value for the DAMAGE component

gpsController.DamageSliderValue = 1f;
```

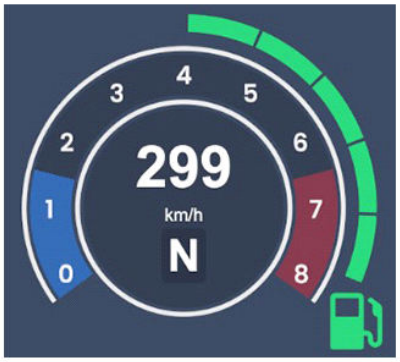
5. Speedometer:

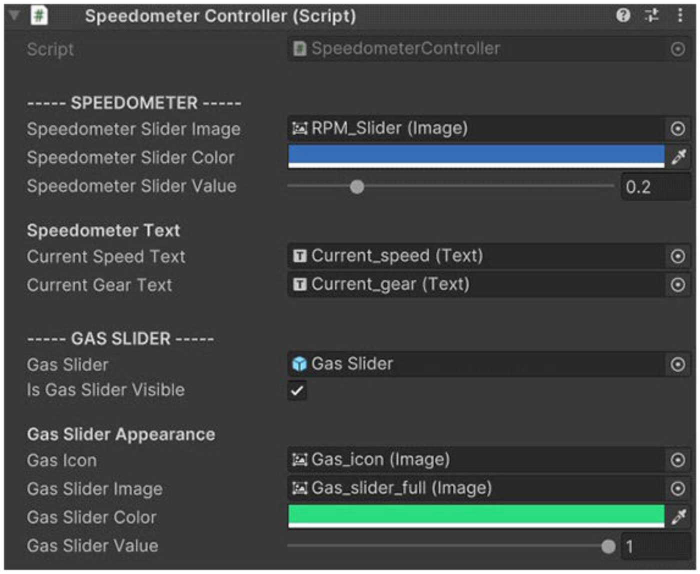
- The GPS UI is included in Car UI Complete Pack/Prefabs/Speedometer.

- Drag the GPS prefab into your.

- Configure the elements in the Inspector.

**They must be configured while the code is not running, so that they remain saved.*





Speedometer Controller (Script)

Script	SpeedometerController
----- SPEEDOMETER -----	
Speedometer Slider Image	RPM_Slider (Image)
Speedometer Slider Color	🟦
Speedometer Slider Value	0.2
Speedometer Text	
Current Speed Text	Current_speed (Text)
Current Gear Text	Current_gear (Text)
----- GAS SLIDER -----	
Gas Slider	Gas Slider
Is Gas Slider Visible	✓
Gas Slider Appearance	
Gas Icon	Gas_icon (Image)
Gas Slider Image	Gas_slider_full (Image)
Gas Slider Color	🟩
Gas Slider Value	1

Step 1: Declare Variables for Components

```
private SpeedometerController speedometerController;
```

Step 2: Find the Components in the Scene

```
speedometerController = FindObjectOfType<SpeedometerController>();
```

Step 3: Initialize and Use the Components

```
// Set values for the SPEEDOMETER components

speedometerController.SpeedometerSliderValue = 0.2f;

speedometerController.currentGearText.text = "N";

speedometerController.currentSpeedText.text = "0";


// Set value for the GAS component

speedometerController.GasSliderValue = 1f;
```

6. Testing and Customization:

- After adding the UI elements, click Play to verify interactions.

- Buttons and steering wheel respond immediately to player actions, GPS provides navigation, speedometer displays real-time speed, and parking sensors offer feedback when the car approaches obstacles.

- Customize each UI element from the Inspector: change colors, sizes and behaviors to match your game's style and theme.

Compatibility

This package is designed for use in Unity and provides a wide range of UI elements (buttons, steering wheel, parking sensors, GPS, and speedometer) that are easy to integrate into any Unity project.

✉

For any questions or inquiries, feel free to contact us at:

info@dont-studio.com