GameDev.tv Game Jam Starter Kit

Simple Car Racing Project

About

Thank you for downloading the GameDev.tv Game Jam Simple Car Racing Project' Starter Kit!

This kit provides you the basic components you'll need to create a simple arcade racing game, including:

- 3rd person arcade car controller
- Track checkpoint system
- A car model and cones to help you create your track layout

This document will guide you through the setup process and explain everything you need to get started!

Contents

Requirements

- Unity 2021.2 or later
- Cinemachine 2.8 or later
- Input System 1.3 or later

Asset Overview

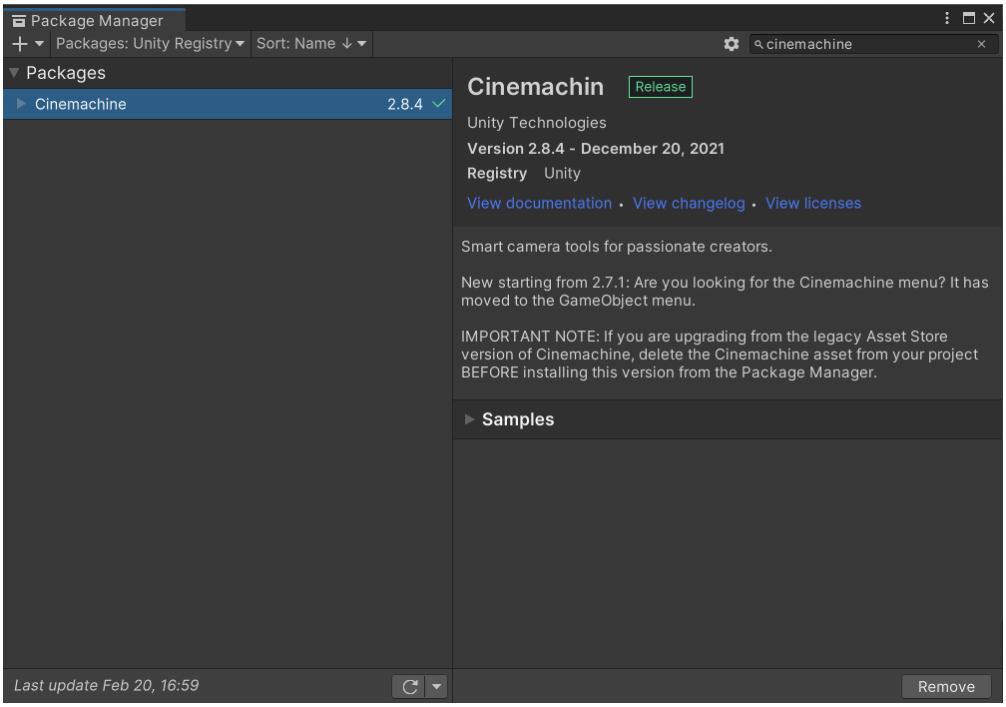
Once imported, you will find everything organized into several folders:

- Art – Includes the; race car model, traffic cone model, and several premade materials
- Input - Contains the input settings and actions for the Unity Input System
- Prefabs – Includes the “Player Car” prefab, which is all set up and ready to go!
- Scenes – Includes the; “Sandbox” demo level, and the TagManager.preset asset (more on that later)
- Scripts – Contains the car controller and checkpoint scripts (more on them later as well)

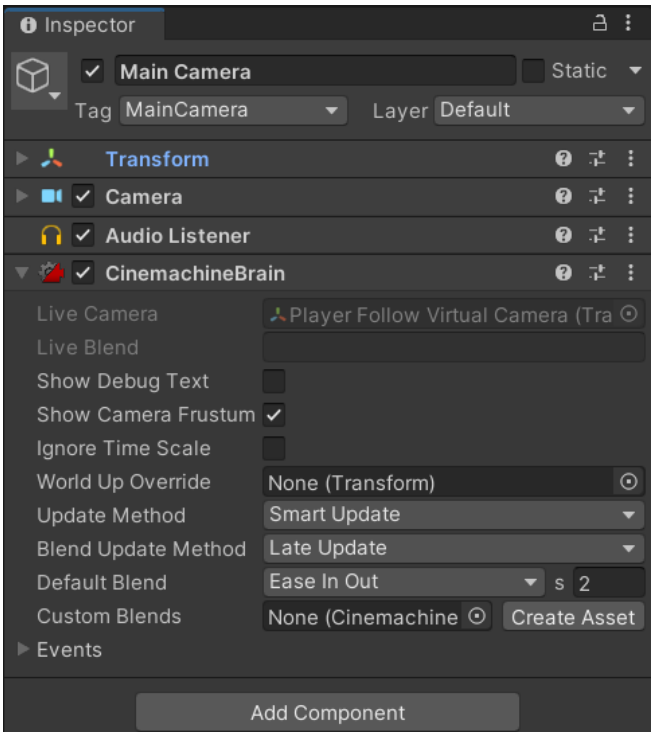
Setup

Setting Up Cinemachine

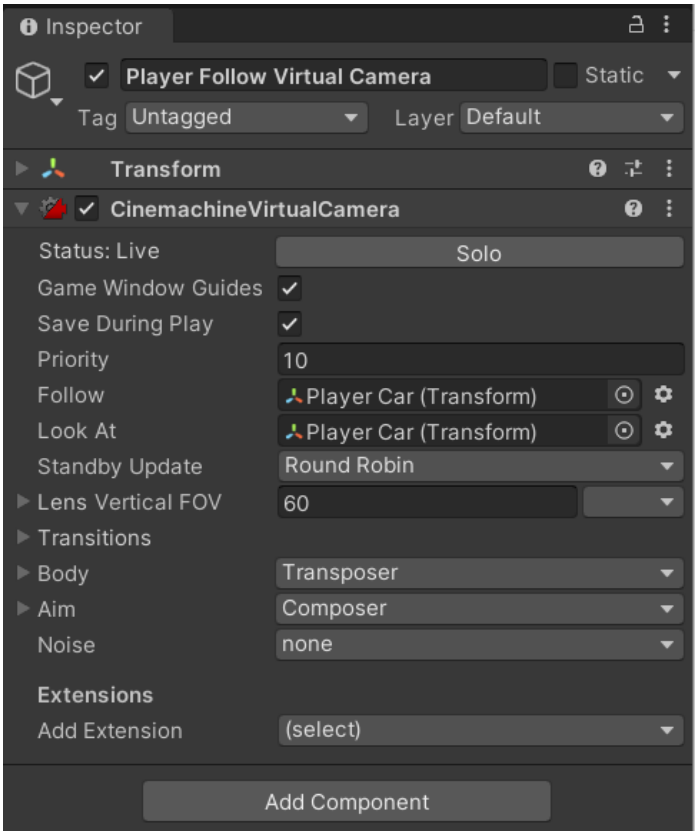
1. Go to; Window > Package Manager
2. Select “Unity Registry” from the Packages drop-down menu
3. Search for the latest version of Cinemachine
4. Press the install button in the bottom right



5. In the Unity hierarchy, navigate to the object containing all of the cameras
6. Check that the “Main Camera” has the Cinemachine component attached (This should already be set up for you)

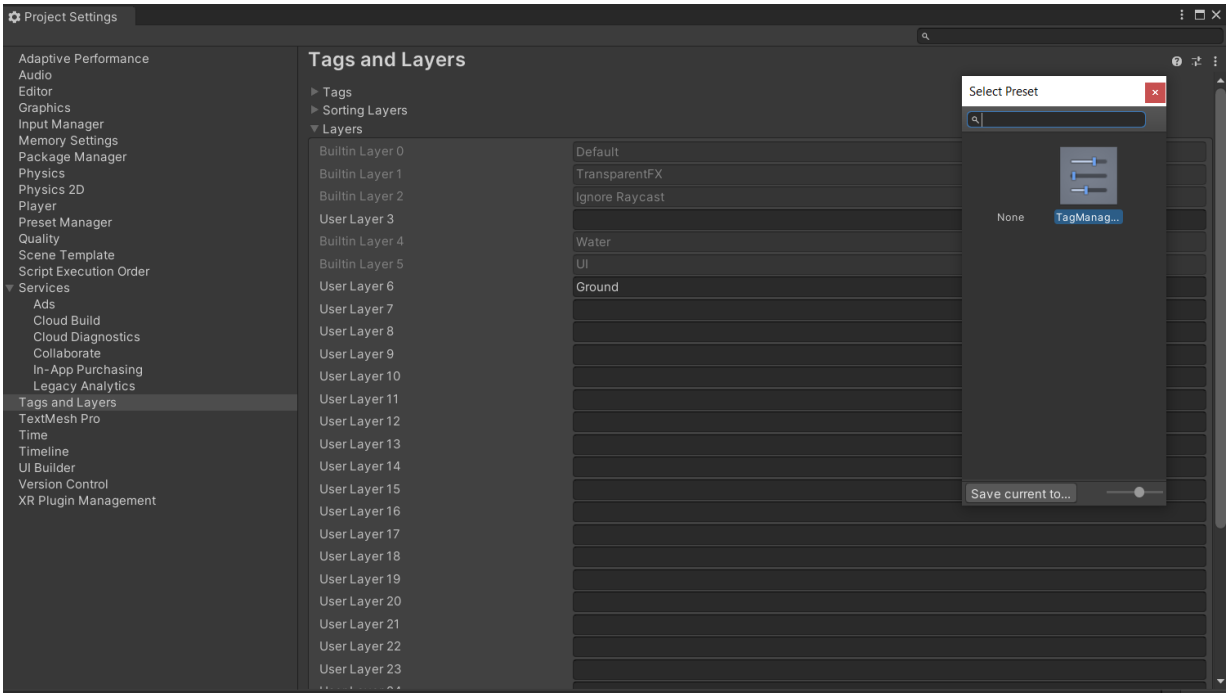


7. Check that the “Player Follow Virtual Camera” has the CinemachineVirtualCamera component attached
8. Set up your camera however you prefer!
   Note: You may want to start by setting the ‘Follow’ and ‘Look At’ parameters to the Player Car GameObject

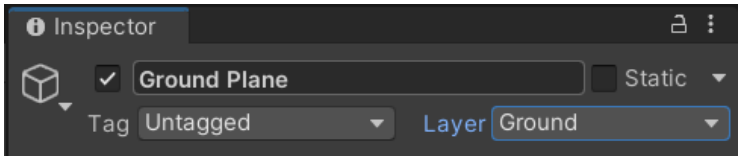


Setting Up Layers

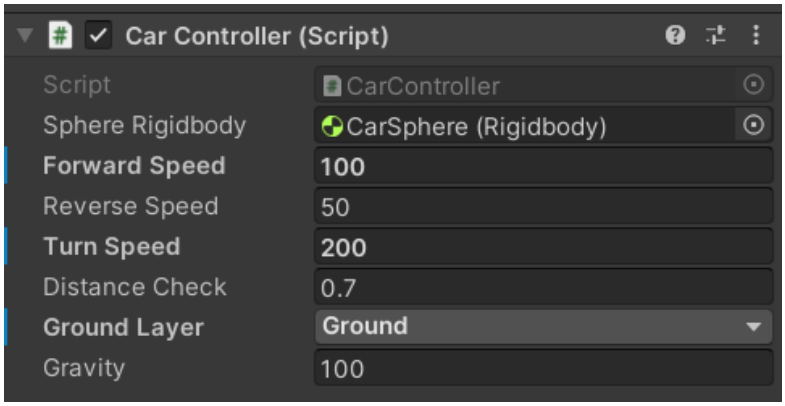
1. Go to; Edit > Project Settings
2. Select “Tags and Layers” from the menu on the left
3. Click on the “Presets” button in the top right (next to the help icon)
4. Select the TagManager.preset
5. This should add a “Ground” layer to slot 6



6. Check that the layers have been correctly set up and connected
    a. The “Ground Plane” GameObject should be assigned to the “Ground” layer (This is a child of the “Environment” GameObject by default)



    b. On the “Player Car” GameObject, the ‘Ground Layer’ parameter of the Car Controller component should be set to the “Ground” layer



Code Overview

All of the code should be fully commented and is organized to be as readable as possible for beginners.

However, here's a brief overview of the included classes.

CarController.cs

This script is attached to the "Player Car" prefab and is responsible for moving the player.

Movement is handled using the new Unity Input system and uses 'Send Message' rather than the more complex event driven system.

The car essentially works by rolling an invisible sphere collider around the track and gives you a nice arcade-style driving experience.
The actual racing car mesh just follows this collider around and is really only there for aesthetics!

CheckpointController.cs

This script is attached to the "Checkpoints" container GameObject.
It manages the checkpoints around the track and stores which checkpoint the player is expected to pass next.

You can either set up the order of these checkpoints manually (via the serialized list in the inspector), or leave that list blank and the order will be automatically generated based on the ordering of the children in "Checkpoints".

Checkpoint.cs

This script simply checks for when the player passes through the checkpoint's trigger collider and tells the CheckpointController script to move on to the next checkpoint.

Need Help?

If you're not already part of our amazing community, why not come and say hi!

- Community Forum
- Discord
- Facebook
- Twitter
- Instagram
- YouTube

We also have a ton of great courses to help you develop your skills in; Unity, Blender, Pixel Art, and more.

Swing by GameDev.tv and supercharge your game development skills today!