

Getting Started

Introduction

Thank you for purchasing our 'Light-Weight Top Down Camera' If you have any issues or questions don't hesitate to contact us at support@modungames.com.

Installation

Input System

Note: This information can also be found here: [Unity Input System documentation](#)

Before installing the 'Light-Weight Top Down Camera' make sure Unity's new Input System package has been added to your project. This can be done via the Package Manager. To open this window, go to Window -> Package Manager (see Figure 1). Make sure preview packages are shown by going to Advanced -> Show preview packages (see Figure 2). Finally, find and select the Input Package in the list on the left and click 'Download' and then 'Import' to add the package to your project.

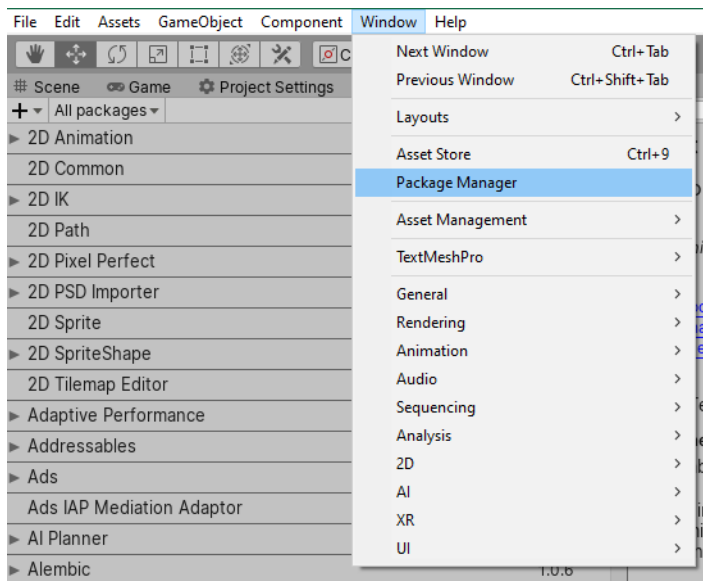


Figure 1: Opening the Package Manager window.

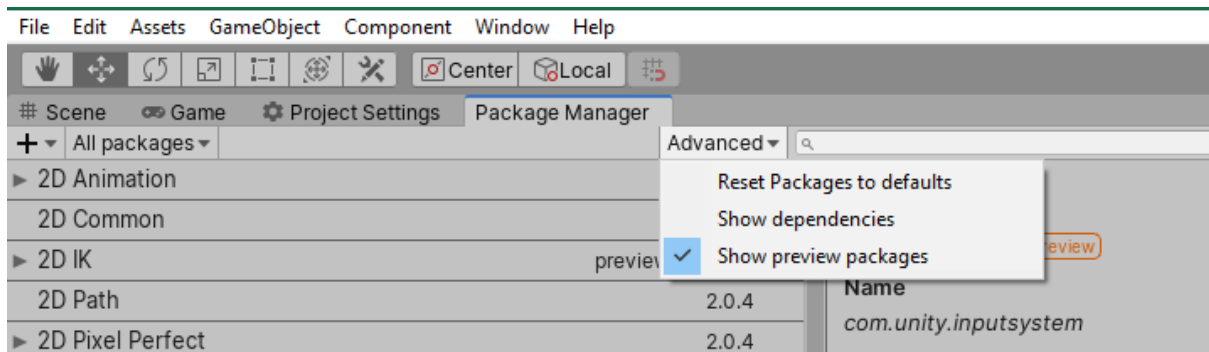


Figure 2: Enable showing preview packages.

Enabling the new input system

To use the new input system, it needs to be enabled in your Project Settings: Edit -> Project Settings -> Player. Change 'Active Input Handling' to 'Input System Package (New)'.

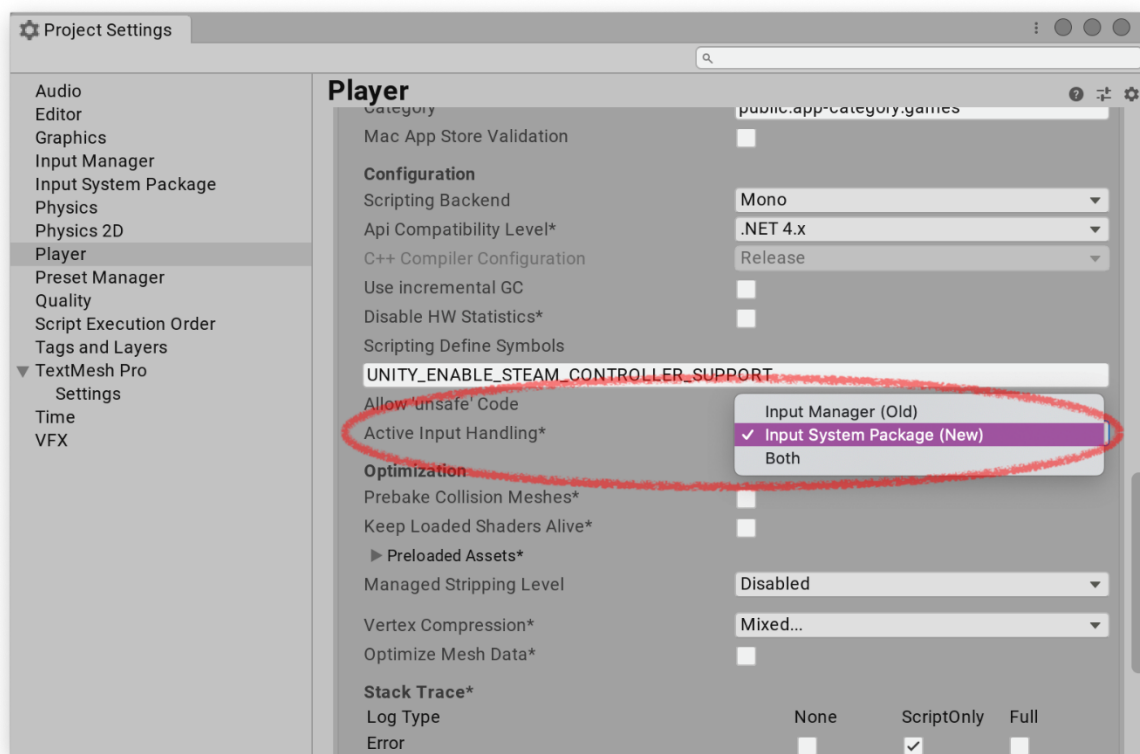


Figure 3: Enabling the new Input System.

Camera

The 'Light-Weight Top Down Camera' package can be found in your purchased assets in the Asset Store window. If you're using 2019.3 or newer you can also find the package in the Package Manager under 'My Assets'. Find the package, download it and import it.

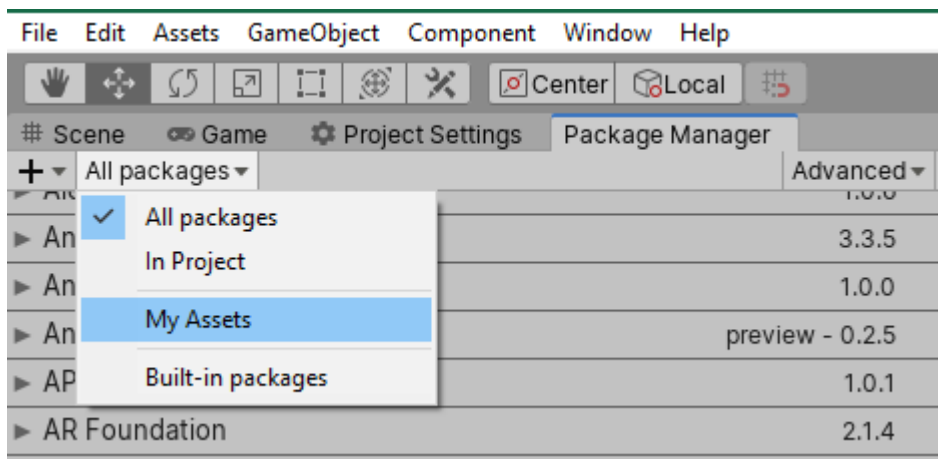


Figure 4: The 'My Assets' section of the Package Manager.

Usage

To use the 'Light-Weight Top Down Camera' you'll have to add the 'Camera Controller'- component to a camera as shown in Figure 5.

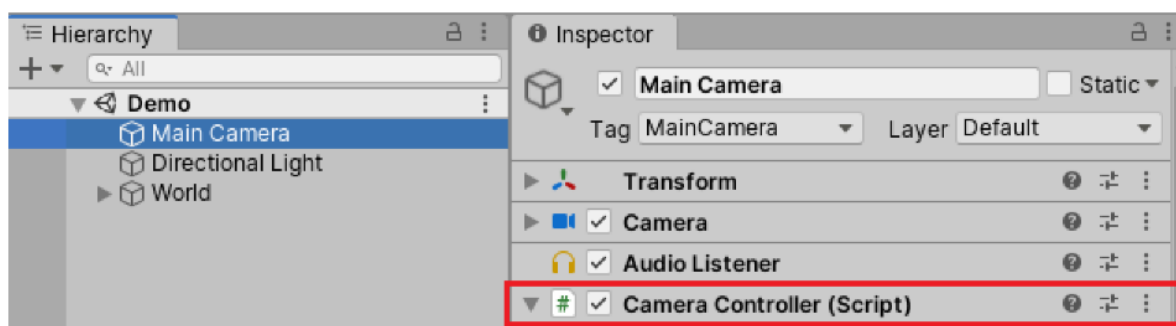


Figure 5: The 'Camera Controller'- component needs to be added to the Camera game object.

Figure 6 shows the settings available in the Camera Controller. The settings can be divided into four parts: General Settings, Panning, Zooming and Rotating.

General Settings

The **minimum and maximum camera bounds** are two points of a cuboid (3D rectangular box) that define the bounds in which the camera can move.

The **Update Method** defines which method updates the camera position, i.e. the Update or LateUpdate method.

The last two settings are useful when the camera needs to be moved while the game is paused by setting `Time.timeScale = 0`. **Calculate Delta Time** tells the Camera Controller to calculate delta time itself instead of relying on Unity's `Time.deltaTime`. **Ignore Time Scale** tells the Camera Controller to ignore `Time.timeScale = 0`, meaning that the camera will still move even when the game is paused using this method.

☒
Camera Controller (Script)

General Settings

Minimum Camera Boui X Y Z
Maximum Camera Boui X Y Z
Update Method
☒ Calculate Delta Time
☒ Ignore Time Scale

Panning

Button Panning

☒ Button Panning
Speed

Edge Panning

☒ Edge Panning
Speed
Inner Border
Outer Border

Grip Panning

☐ Grip Panning
Grip Pan Speed

Zooming

FOV Zooming

☐ Use FOV Zooming
Speed
FOV Zoom Smoothing
Minimum FOV
Maximum FOV

Vertical Translation Zooming

☒ Use Vertical Translation Zooming
☒ Allow Scroll Zooming
Speed
Speed (Scroll)
Vertical Translation Zc

Rotating

☒ Rotate Horizontally
☒ Rotate Vertically
Rotate Speed
Horizontal Space
Vertical Space

Figure 6: The Camera Controller settings.

Panning

The Camera Controller supports 3 different panning methods:

1. Button panning: using WASD and arrows by default to move the camera.
2. Edge panning: moving the cursor to the edge of the screen to move the camera.
3. Grip panning: clicking and moving the cursor to move the camera.

Each method can be enabled or disabled according to your game's needs or user preferences. Each method also uses their own **Speed** setting which defines how fast the camera is being moved by an interaction. Lastly, for edge panning the area in which the camera is being moved can be defined. The **Inner Border** setting defines the number of pixels inside of the game window that will move the camera. The **Outer Border** setting defines the number of pixels outside of the game window that will move the camera.

Zooming

The Camera Controller supports two ways of zooming: FOV zooming and vertical translation zooming (default). Both methods can be enabled or disabled to your game's needs or user preferences.

FOV zooming will modify the camera's FOV. The speed with which the value is being modified can be changed with the **Speed** setting. The **FOV Zoom Smoothing** setting defines a timespan over which the zooming animation occurs to make the user experience a bit smoother. Lastly, you can define a **minimum and maximum FOV** for the camera.

Vertical Translation Zooming moves the camera along the vertical axis. This can be done with buttons (ZX) or the scroll wheel. The scroll wheel can be disabled separately using the **Allow Scroll Zooming** setting. Both use a different **Speed** setting but use the same smoothing timespan.

Rotating

The Camera Controller supports rotating both horizontally (QE) and vertically (CV). Both methods can be enabled or disabled according to your game's needs or user preferences. Both methods use the same rotation **Speed** which defines how fast the rotation will occur. Lastly, the Space can be defined for the rotation. This defines how the rotation occurs: relative to Self or the World.

Input and key bindings

The Camera Controllers comes packaged with an action mapping defining a default key binding. Double-clicking the InputActions asset will open the Input System's action map window (see Figure 7). The key bindings (blue and pink lines) can be changed. This means the WASD can for example be changed to IJKL or this key mapping can be added as an extra key binding, much like the arrow keys. The actions expect certain value types though so there might be cases where adding different keys or input controllers cannot be done without changing the code. If you've made any changes don't forget to save the changes!

The actions (green lines) cannot be changed unless the code is manually modified.

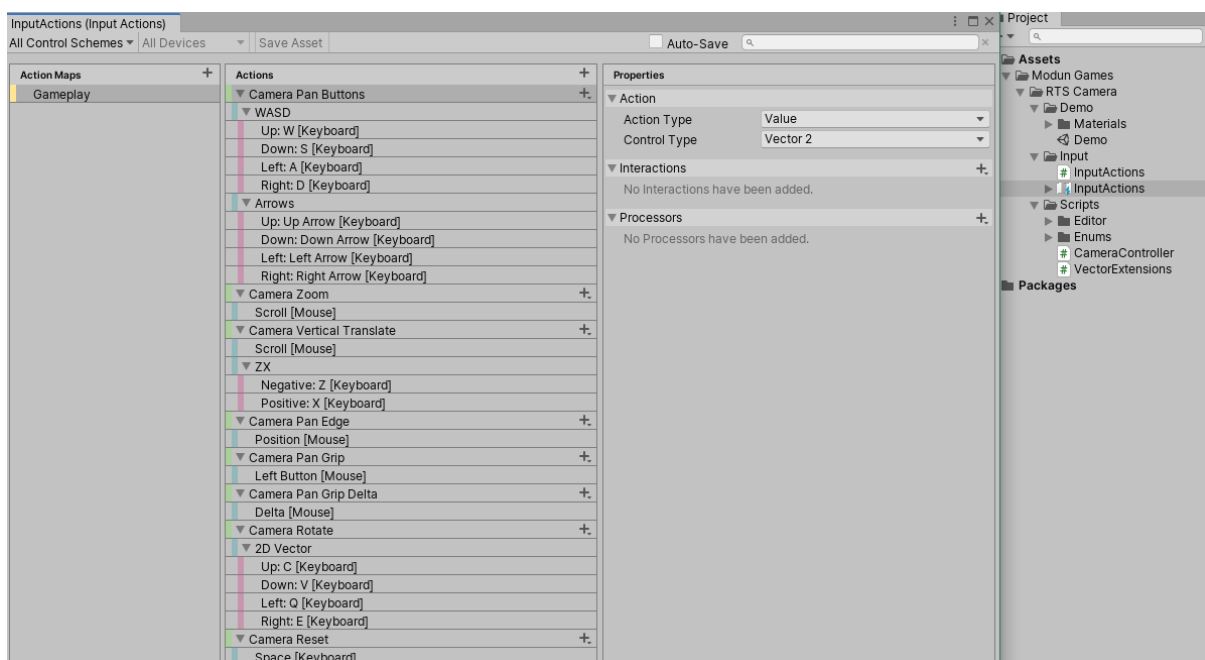


Figure 7: Default key bindings.