Four Dimensional Portals

Documentation v1.0.0

Four Dimensional Portals is a portal system implementation based on physics contacts modification and render texture. This portal system is only compatible with **Unity 2021 LTS or newer versions** and **does not support the unity built-in character controller component**.

The demo scenes require **Input System** & **Shader Graph** packages to work.

Thank you for purchasing this asset.
Contact email: xuwucontact@gmail.com

Table of Contents

Object Overview

Portal System (MonoBehaviour)

The **PortalSystem** handles portal physics and rendering. The user must ensure that there is one and only one **PortalSystem** active in the scene.

Portal (MonoBehaviour)

Only portals with the same **PortalConfig** can link to each other.

Portal Traveler (MonoBehaviour)

The **PortalTraveler** must be attached to **Rigidbody**. Only rigidbodies that are portal travelers can pass through the portal, otherwise they will be blocked by the portal plane.

Portal Config (ScriptableObject)

Indicates the appearance and physics related settings of the portal.

Portal System Additional Camera Data (MonoBehaviour)

The **PortalSystemAdditionalCameraData** must be attached to a **Camera**. It indicates the portal rendering settings of the camera. If a camera doesn't have this, all portal rendering settings of the camera will use default settings, just like the scene view.

Rigidbody Ghost (MonoBehaviour)

The **RigidbodyGhost** is a clone object to keep objects colliding between portals, it will be automatically generated by the **PortalSystem**, do not add it manually.

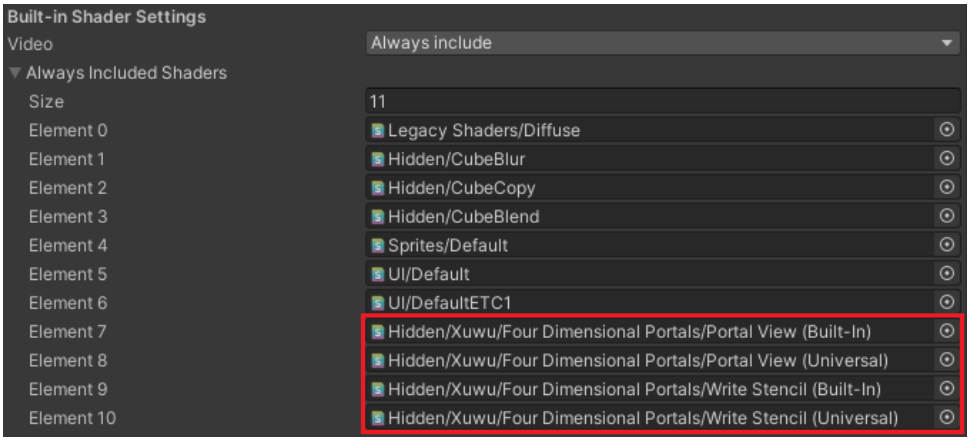
Collider Ghost (MonoBehaviour)

The **ColliderGhost** is a clone object to keep objects colliding between portals, it will be automatically generated by the **PortalSystem**, do not add it manually.

How To Use

Quick Start

1. Open a new unity project and install this package.
2. Under Project Settings > Graphics > Built-in Shader Settings, add the following shaders to the always included shaders.



3. Open a demo scene under **Assets/Xuwu/FourDimensionalPortals/Demo/Scenes/**.
4. Press play to test and see how it works.

Portal & PortalTraveler Setup

Please refer to the demo scenes.

PortalConfig Setup

Please refer to the **PortalConfig** examples under **Assets/Xuwu/FourDimensionalPortals/Demo/PortalConfigs/**.

Custom CharacterController

Please refer to **Assets/Xuwu/FourDimensionalPortals/Demo/Scripts/RigidbodyCharacterController.cs**.

Portal Placement

Please refer to **Assets/Xuwu/FourDimensionalPortals/Demo/Scripts/PortalGun.cs**.

Custom Shader

Please refer to the shadergraph examples under **Assets/Xuwu/FourDimensionalPortals/Demo/Arts/Shaders/**.

Additional Information

Please Note

- Changing transform.position/rotation will always override rigidbody.position/rotation in the next physics sync, **to avoid mistakes you should always use transform instead of rigidbody to set position/rotation** unless you know what you are doing.
- When rendering a portal with an oblique projection matrix, it breaks z-depth and causes problems with effects using z depth (e.g., ssao in urp).
- The OnCollisionXXX callbacks do not work properly when the collider interacts with a portal.
- Make sure the distance between the portal and the placement plane collider is greater than contact offset.