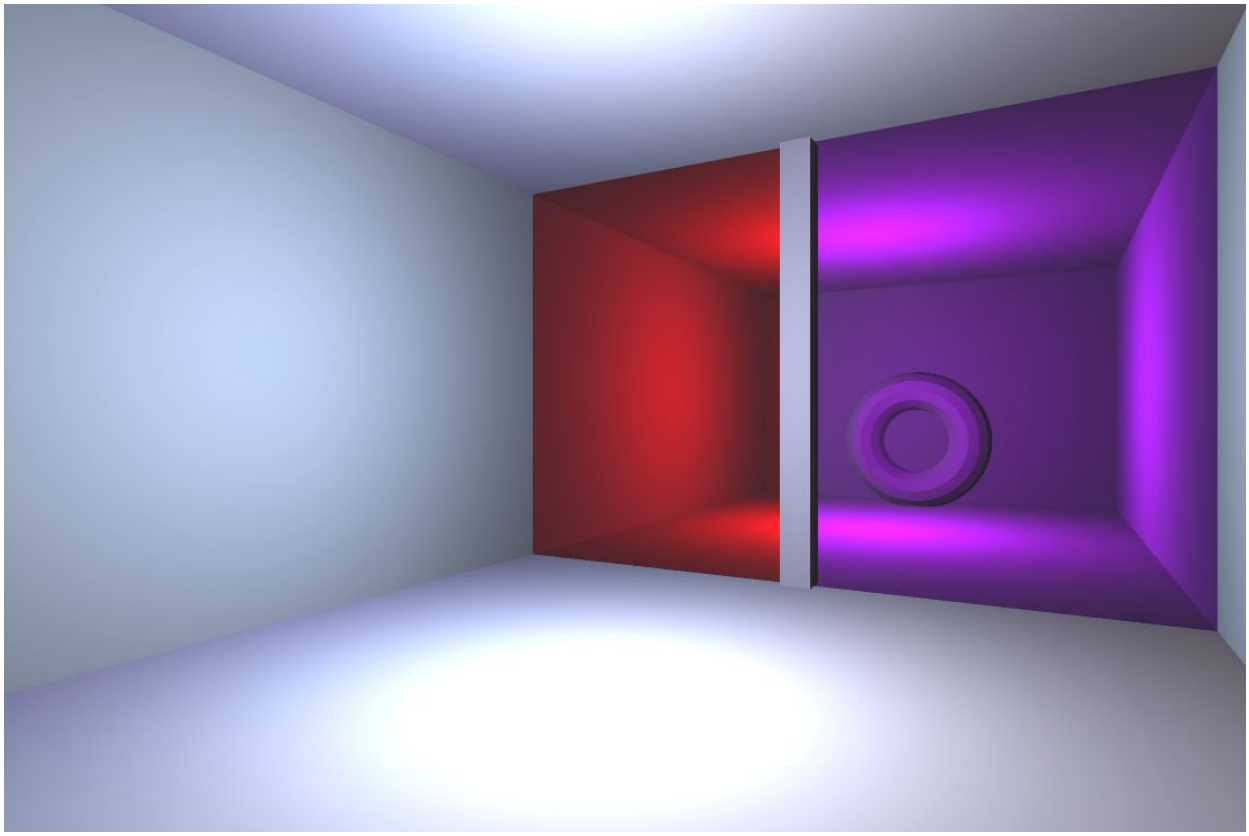


World Portal System

Documentation

Samy Bencherif
SamyBencherif73@gmail.com

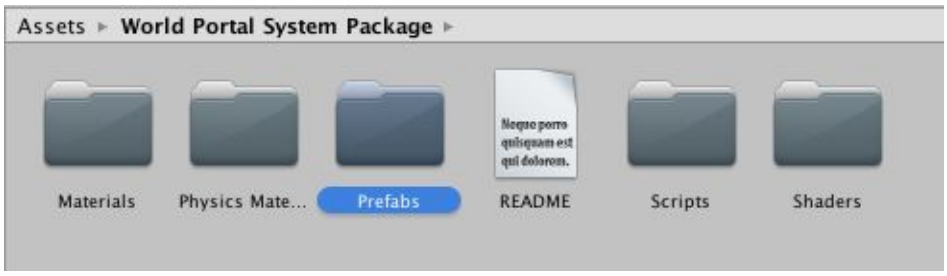
Overview

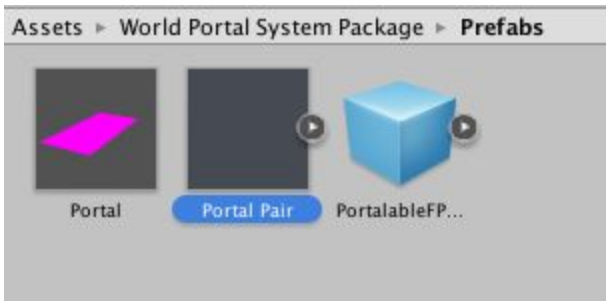
This document will aid you when using the World Portal System for Unity. With it you can create the impression of impossible geometry, have as many portal pairs as you want in a scene, and even make non-rectangular world portals. Additionally you can script with portals!

Contents

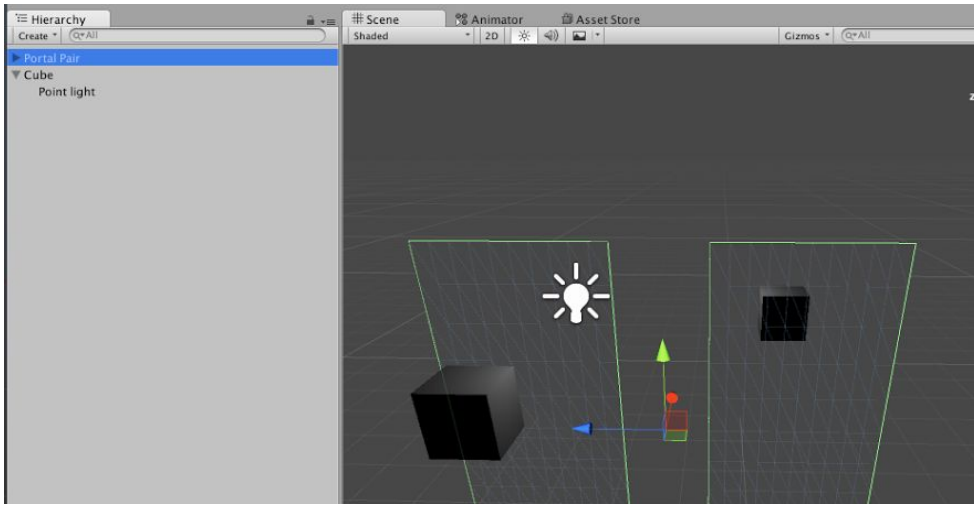
Adding a Portal Pair

To quickly add a basic portal pair (suitable most of the time) find the *Portal Pair* prefab located in the *Prefabs* folder within the package.





Click and drag the prefab into either the **Scene View** or the **Hierarchy.**

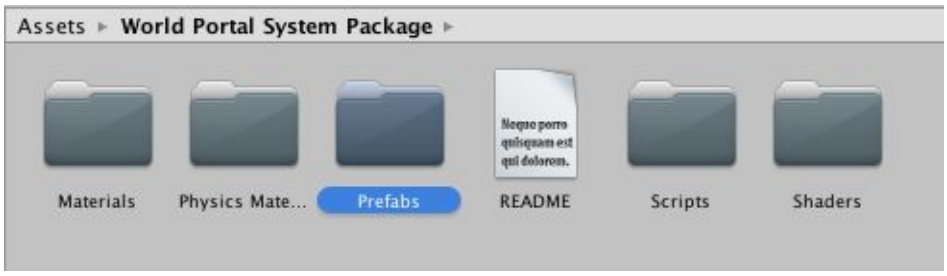


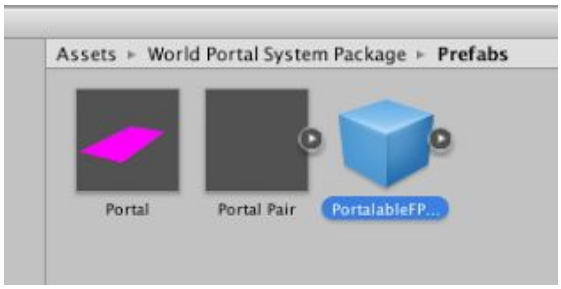
That's all! You have added a portal pair to the scene. You may now position, rotate, and scale the portals.

Note: Please ensure both portals have the same scale and leave the y-value of their scales set to 1. (More about this in Known Design Limitations)

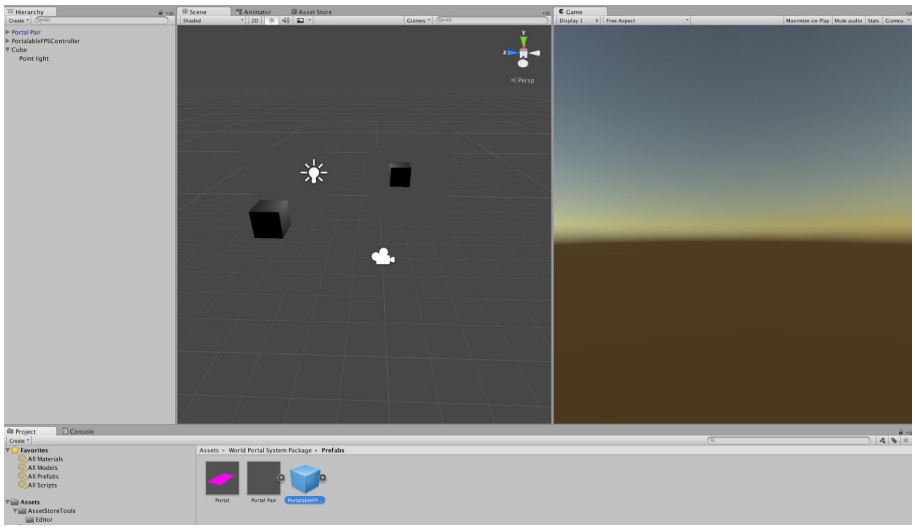
Adding a Portalable Player

To add a portalable player find the *PortalableFPSController* prefab located in the *Prefabs* folder within the package.





Click and drag the prefab into either the **Scene View** or the **Hierarchy.**



Now you can see the player's perspective in the game view. The player will automatically reorient if teleported in a way that rotates the player sideways. The player is using a duplicate of CrossPlatformInputManager's *Mouse X* and *Mouse Y* axis for input.

More Detail About Portals

Dragging in prefabs is not the only way to use this package. For example you can attach the *Portal* script to an object to turn it into a world portal.

Features of the portal script are dependant on other components.

If you want **teleportation** to work you need...

- A Collider (any 3d collider will work but a thin **Box Collider** works best)

If you want the **visual effect** to work you need...

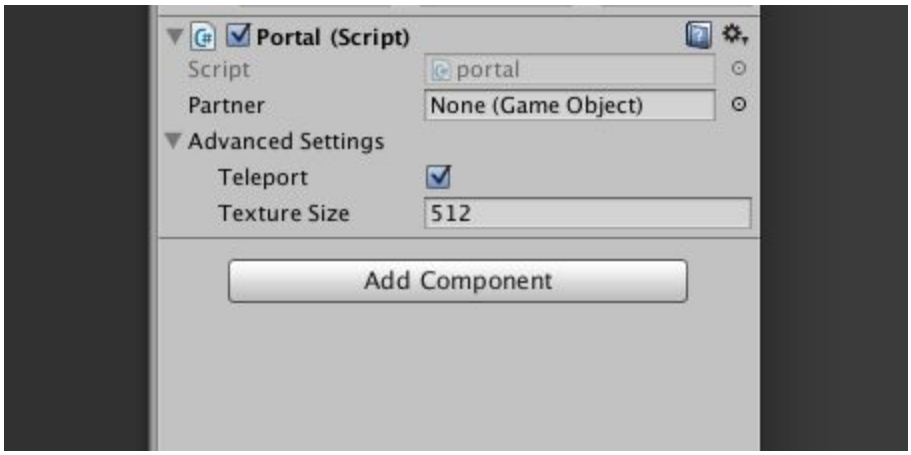
- A Mesh Renderer
  - The Mesh Renderer should be using a material with the FX/Portal shader.
  - The script should set the material automatically. If it does not, set the material to None and it will reload the texture. (More about this in Known Design Limitations)

If you want teleportation to look **seamless** you need...

- To make sure the geometry (attached via Mesh Filter) of the object is flat, has only parallel normals, and is mostly convex 2 dimensionally. To simplify things, here are some shapes that fit the criteria:
  - Circle/Ellipse
  - Any Quadrilateral
  - Any Regular Polygon
- Some that do not:
  - Anything with holes in it
  - A Pointy Star

More Detail About Portals (cont.)

Once you've attached the *Portal* script to an appropriate object, you can configure some settings.



Partner

This is the game object the portal is leading to. This means multiple portals can lead to one location, but a portal can only have one destination. This partner gameObject must also have a *Portal* script and must be enabled.

Teleport

You can use this checkbox to disable teleportation without disabling the trigger on the object.

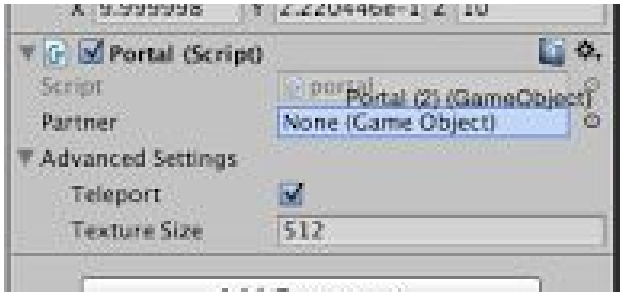
Texture Size

Here you may specify the size of the texture used for the portal effect. The texture is a labeled as a POT. Since two connected Portals must have the same texture size, a portal will either use it's own specified texture size or its partners, depending on which is bigger.

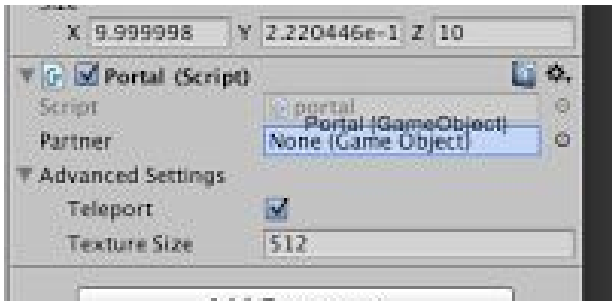
Non-Rectangular Portals

To create a non-rectangular portal pair you need to create your own portal pair without the use of prefabs. It is done like this:

1. Create an object that meets the criteria described in More Detail About Portals
2. Add the *Portal* script to the object
3. Name the object something memorable say *Portal (1)*
4. Duplicate *Portal (1)*, and rename the duplicate to *Portal (2)*
5. Select *Portal (1)* and drag *Portal (2)* into the Partner variable of *Portal (1)*.



6. Select *Portal (2)* and drag *Portal (1)* into the Partner variable of *Portal (2)*.



Scripting

General Information

You can set the variables associated with World Portals programmatically.

This is done the same way as setting any other variable in any other component in Unity, with this general form:

C#

```
MyPortal.GetComponent<portal>().variable = something;
```

Unityscript

```
MyPortal.GetComponent("portal").variable = something;
```

MyPortal is a reference to a World Portal Object

Variable is the name of a public variable

Something is the value to assign to the variable

See More Detail About Portals (cont.) for an explanation of all the public variables.

How to set Partner

C#

```
MyPortal.GetComponent<portal>().partner = MyOtherPortal;
```

Unityscript

```
MyPortal.GetComponent("portal").partner = MyOtherPortal;
```

How to set Texture Size

C#

```
MyPortal.GetComponent<portal>().AdvancedSettings.m_TextureSize = 1024;
```

Unityscript

```
MyPortal.GetComponent("portal").AdvancedSettings.m_TextureSize = 1024;
```

Note: The texture size and teleport variable are children of AdvancedSettings. Also do not forget the "m_" preceding the variable names, which is not visible in the editor.

How to disable Teleportation

C#

```
MyPortal.GetComponent<portal>().AdvancedSettings.m_Teleport = false;
```

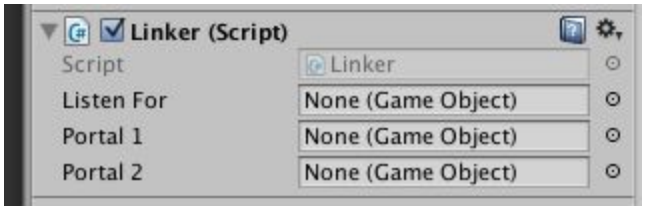
Unityscript

```
MyPortal.GetComponent("portal").AdvancedSettings.m_Teleport = false;
```

Achieving Complex Behavior without Scripting

Those who would like to dynamically manipulate portals without any scripting may use the *Linker* script.

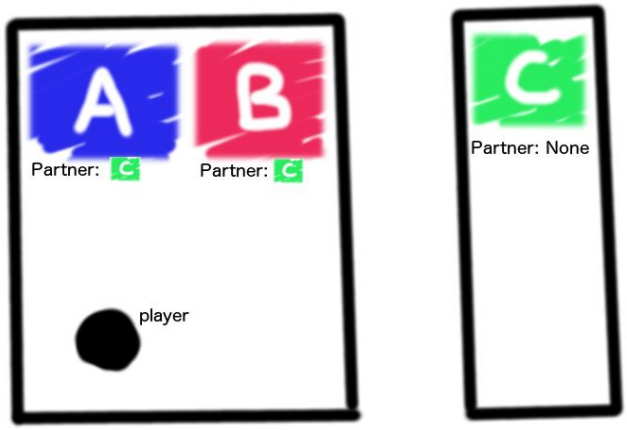
Attach the script to any object with a trigger. When something enters the trigger region, a specified couple of portals will link.



If you would like the script to only listen for a specific object, drag that object into *Listen For*.

Otherwise make sure your trigger area isn't hitting anything with a collider (including floor and walls) or the portals will be linked immediately.

Consider this example scenario

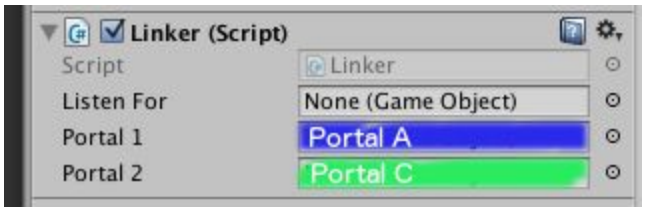


Let's say I want both portals A and B to be linked to C. Now I could set both of their partners to C without a problem, but the problem is that I can't make portal C show the way I came from.

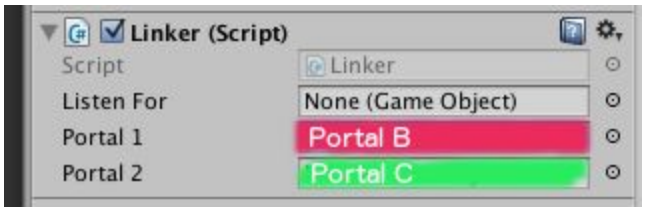
Solving the problem

The goal is to quickly set C's partner to A if I step through A or to set it to B if I step through B.

I will attach a *Linker* to Portal A with the following settings:



I will attach a *Linker* to Portal B with the following settings:



Now the moment I contact either Portal A or B there will be a two way connection between that portal and Portal C.

You can see a live demo that uses this principle here:
http://samybencherif.github.io/Portals-Demo-Scene/

Known Design Limitations

- Does not support recursive portal rendering
- Unity editor has crashed for me while trying...
    - To attach a rigidbody to a portal and throw it through its counterpart.
    - To enter a portal which has a disabled counterpart
- Portals must have the exact same scale as their counterparts
- The y-value of a portal's scale should be set to 1 or you may experience weird graphics bugs and physics objects may get distorted upon teleportation
- If anything goes wrong with a portal while you are editing it, try setting the portal's material to None. This will alert the portal script to update the portal's texture. The material should quickly change to "FX/Portal"
- There is occasional and difficult to notice flickering when entering portals at a specific angle
- Lighting is not transferred through portals
- Shadows created via directional lights are not visible through portals
- These limitations exist, however it is not difficult to work around them as you can see from this live demo and this video demo
    - http://samybencherif.github.io/Portals-Demo-Scene/
    - https://www.youtube.com/watch?v=m-5gdVlY_2s

- Please email me questions or concerns regarding the limitations of this product and/or how to work around them: SamyBencherif73@gmail.com