

Dimensions And Portals V1.20

Peculiar Developer

18th October, 2013

Last update: 18th May 2015

Important

First of all thank you for your interest in the Dimensions and Portals (DAP) Package for Unity Personal & Unity Pro. It is of great importance to follow the initial setup steps in order to make DAP work for both Unity Personal and Unity Pro.

1 Contents

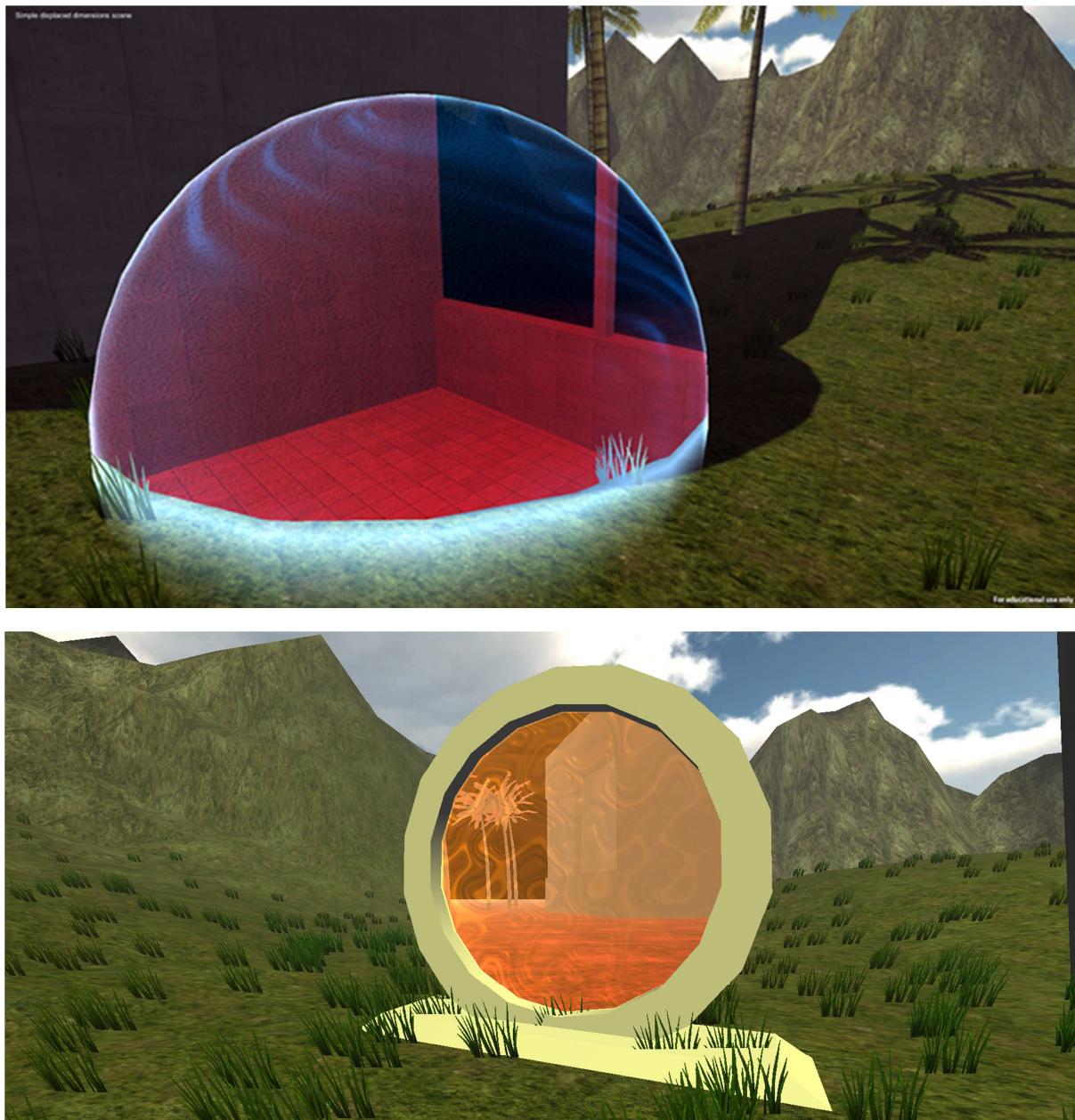
2	Introduction.....	4
2.1	Terminology	4
3	Setup.....	5
3.1	Initial Setup	5
3.1.1	Layers and Tags	5
3.1.2	Physics (Layer Collision Matrix)	7
3.1.3	DimensionsAndPortalsController.....	7
3.1.4	Test run.....	9
3.2	Additional setup for Dimensions	9
3.2.1	Gates	9
3.2.2	Skybox.....	12
3.2.3	Per Dimension Render Settings	12
3.2.4	Light and Special FX.....	13
3.2.5	Oculus Rift Support.....	14
3.3	Additional setup for Portals	14
3.3.1	Portals	15
3.3.2	Scripts	15
4	Programming.....	16
4.1	DimensionsAndPortalsController	16
4.1.1	Important properties (public)	16
4.1.2	Important functions (public)	16
4.2	DimensionalGateController.....	16
4.2.1	Important Functions (public)	16
4.3	Camswitcher.....	17
4.3.1	Important Functions (public)	17
4.4	DAPPPhysics	17
4.4.1	Important Functions (public)	17
5	Known limitations	17
5.1	General.....	17
5.1.1	Nearby gates / portals	17
5.1.2	Layers and Tags	17
5.1.3	Both Gates & Portals	18

5.1.4	Character Controllers.....	18
5.2	Dimensions.....	18
5.2.1	Dimensions	18
5.2.2	Gate collider size	18
5.2.3	Camera's.....	18
5.2.4	Physics.....	18
5.3	Portals.....	18
5.3.1	Portals	18
5.3.2	Glitchy portal look when only one portal is present	19
5.3.3	Transportation	19
5.3.4	Physics.....	19
5.3.5	Thin walls.....	19
5.3.6	Shooting portals partially in air.....	19
5.3.7	Oculus Rift	19
5.3.8	Raycast.....	19
6	Support	20
7	Changelog of Manual.....	Fout! Bladwijzer niet gedefinieerd.
7.1	13 th July 2014	Fout! Bladwijzer niet gedefinieerd.
7.2	30 th June 2014	Fout! Bladwijzer niet gedefinieerd.

2 Introduction

Dimensions And Portals is an asset that allows you to create Dimensional Gates and Portals in Unity. This is available for Unity Personal and Unity Pro and supports both Forward rendering as Deferred rendering (Forward with limitations). This is also available and tested for Unity for Android.

So if you want to create multiple dimensions, dimensional gates, different timelines, special glasses to see ghosts or whatever you can imagine involving the Dimensions and Portals package, you can make it now.



2.1 Terminology

- **Portal:** a portal similar to the ones known from the Valve game Portal and Portal 2
- **Gate:** a 3D shaped gate to another dimension.
- **Dimension:** a virtual dimension based on a layer in Unity to represent another dimension, location, timeline or otherwise.

3 Setup

Please follow the setup so make the package work correctly. Since Unity doesn't allow this level of project editing from script you will have to do a few steps manually. The initial setup consists of 4 steps:

- Make sure you have the Character Controller Asset of Unity imported (see 6.1.4 for details). This should by default be included in the package.
- Setup the Layers and Tags
- Setup the Physics
- Setup the Dimensions and Portals Controller

Later on you can find specific setup items.

Due to changes in the rendering pipeline there is a difference between the Legacy setup, and the new setup divided in Personal and Pro. Please read the appropriate chapters.

3.1 Initial Legacy Setup

3.1.1 Layers and Tags

To make the DAP sample scenes work, you need to add 6 layers in this order:

- **DefaultAlternate**
*This is the default alternate dimension (**Default** is the default dimension)*
- **DefaultAndAlternate**
*This is for any gameobject that exists in both the **Default** and **DefaultAlternate** dimension*
- **DAP_Portal**
Used for the Gates and Portals
- **DAP_DisabledForPlayer**
Used for the Portals to tag an object to be disabled for Portals
- **Player**
Used for the Player in a Portal setup
- **DAP_PortalMask**
Used for the Portals

You can add the layer by clicking on : Edit → Project Settings → Tags

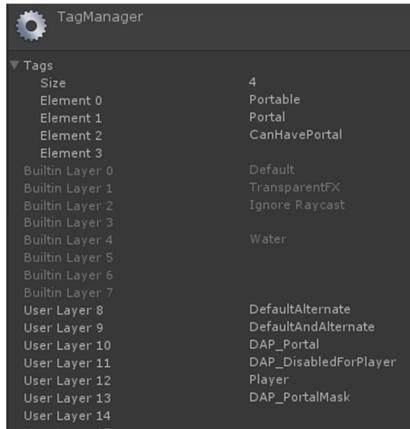
You also need to add 3 tags in this order:

- **Portal**
Used to tag portals and gates
- **Portable**
Used to tag any gameobject that can travel through a portal or gate that allows it.
- **CanHavePortal**
Used to tag any gameobject that can have a portal.

You can add the tags by clicking on : Edit → Project Settings → Tags
(sometimes they are imported automatically)

Important:

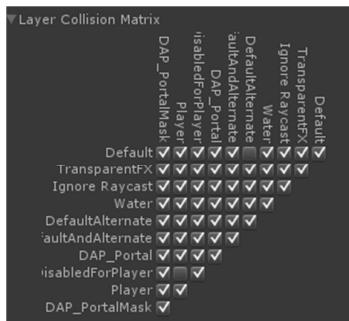
In order for the test scenes to work, the tags and layers must be in the correct order. Please see the screenshot below for the correct order:



3.1.2 Physics (Layer Collision Matrix)

You can edit the physics by clicking on : Edit → Project Settings → Physics

- Uncheck the check that represents **Default** and **DefaultAlternative**.
- Uncheck the check that represents **Player** and **DAP_DisabledForPlayer**.



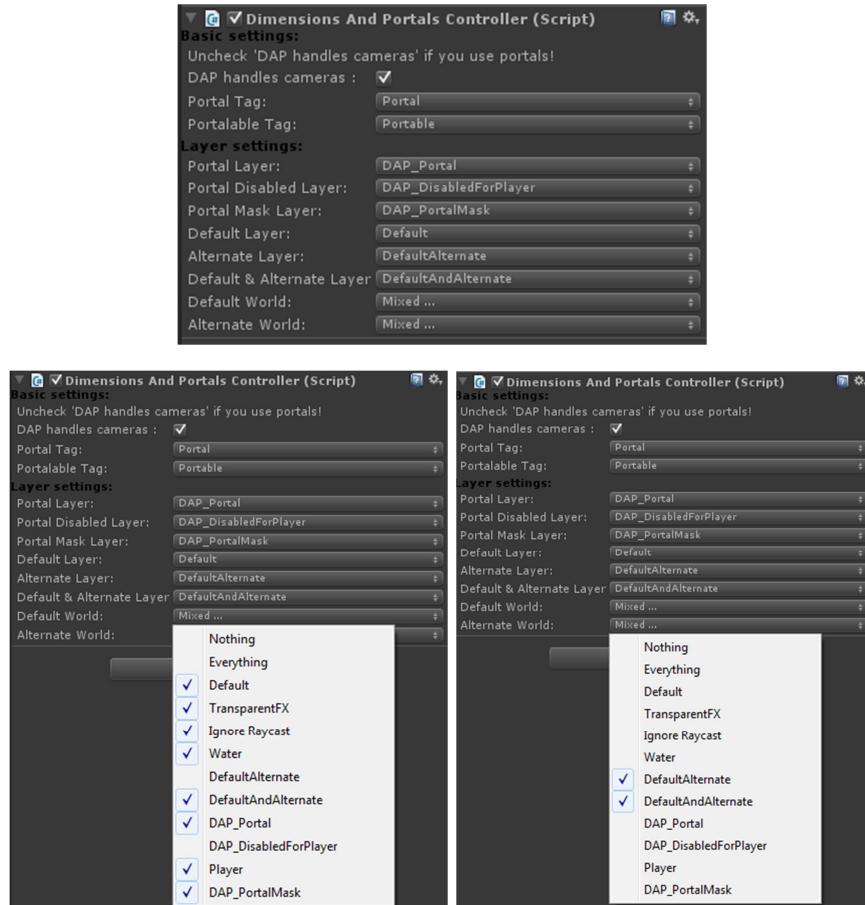
3.1.3 DimensionsAndPortalsController

If you open a test scene (DimensionsAndPortals/Scenes) you will find an object in the scene named **DimensionsAndPortalsController (<Type>)**. You will need to edit/check this for proper usage. For the demo scenes, make sure these are all on their default values.

- **DAP handles cameras:** (default: **Checked**)
Check this option if you want the DAPController to handle the camera settings. This is useful if you want gates you can watch through and mandatory if you have gates you can walk through. **This MUST be disabled if you want to use portals!**
- **Portal Tag:** (default: **Portal**)
The tag that is used for portals.

- **Portable Tag:** (default: **Portable**)
The tag that is used for objects that can pass through portals and gates.
- **Portal Layer:** (default: **DAP_Portal**)
The name that is used to determine the layer of portals and gates. Will give a warning if incorrect.
- **Portal Disabled Layer:** (default: **DAP_DisabledForPlayer**)
Used in the test script to allow the passage of players through portals.
- **Default Layer:** (default: **Default**)
The layer that is used to determine the layer of the default dimension. Only in very special cases would you have to change this.
- **Alternate Layer:** (default: **DefaultAlternate**)
The name that is used to determine the layer of the default dimension. Only in very special cases would you have to change this.
- Default & Alternate Layer:** (default: **DefaultAndAlternate**)
The name that is used to determine the layer of the default dimension. Only in very special cases would you have to change this.
- **Default World:** (default: **All layers minus DefaultAlternate and DAP_DisabledForPlayer**)
- **Alternate World:** (default: **DefaultAlternate and DefaultAndAlternate**)

You can see the default setup below (without the Legacy Setup checkbox):



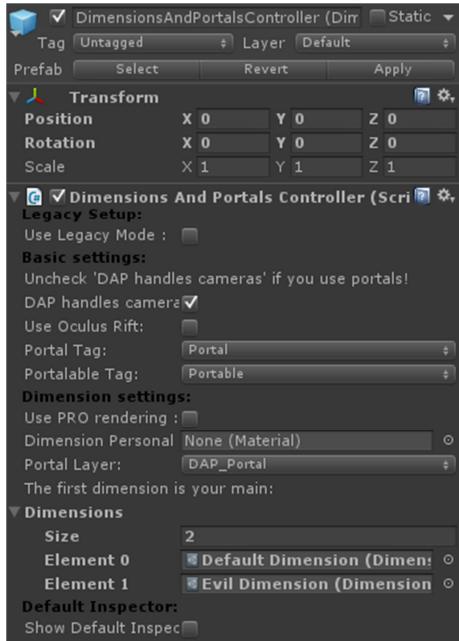
3.1.4 Test run

You can now test the test scenes if they work properly. If not, check if you missed any step.

3.2 Initial Personal Setup

Follow the same steps as in 3.1.1 and 3.1.2

3.3 DAP Controller in Personal (Info)



The DAP Controller is different from the legacy controller. You can see that the **Use Legacy Mode** is unchecked and that there are less options under **Basic Setting**. Furthermore there is a new title called **Dimension settings**. In here you can check if you have PRO rendering and what the **Portal Layer** is. Furthermore there is a material for **Personal Rendering** that for now is a placeholder. When checking **Pro Rendering** this material must be set to the **DimensionPro.mat** since it will change in **Pro Rendering**. Finally you can see the Dimensions that you will have in your scene by **Dimension Config Files**. These must be set prior to rendering your scene. In Personal this will give a warning if you use more than 2 since the automatic dimension handler cannot handle more than that. In pro these can be up to 4-10 depending on your game.

3.4 Initial PRO Setup

Follow the same steps as in 3.1.1 and 3.1.2. Also read 3.2 for details on the use of the DAP Controller in Pro and Personal.

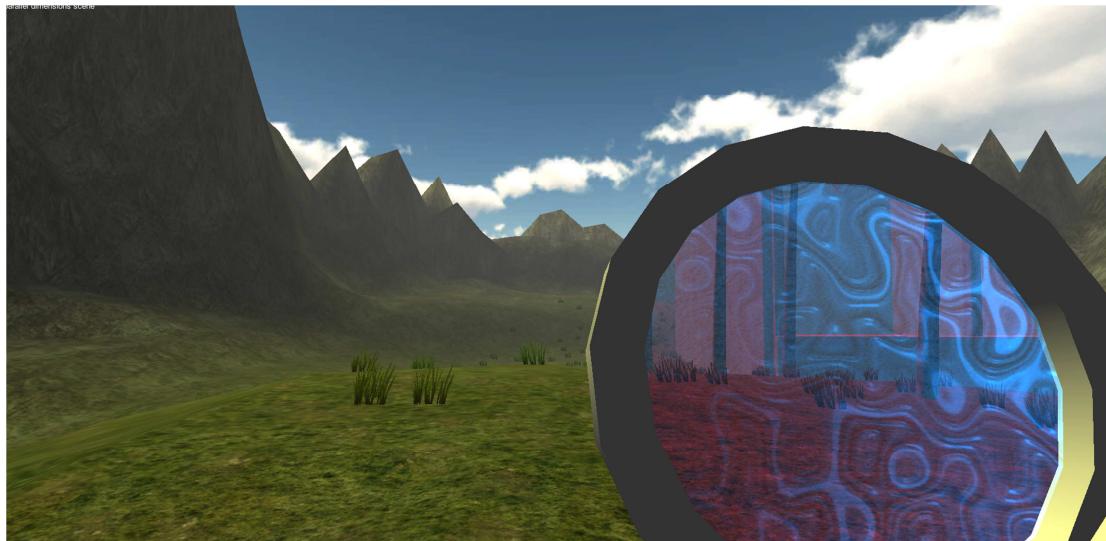
4 In depth setup

4.1 Additional setup for Dimensions

4.1.1 Legacy Gates

Based on the type of scene you are using, you can place gates. These can consist of 2 types: Parallel (or Glass) and Displaced.

4.1.1.1 Parallel Gates

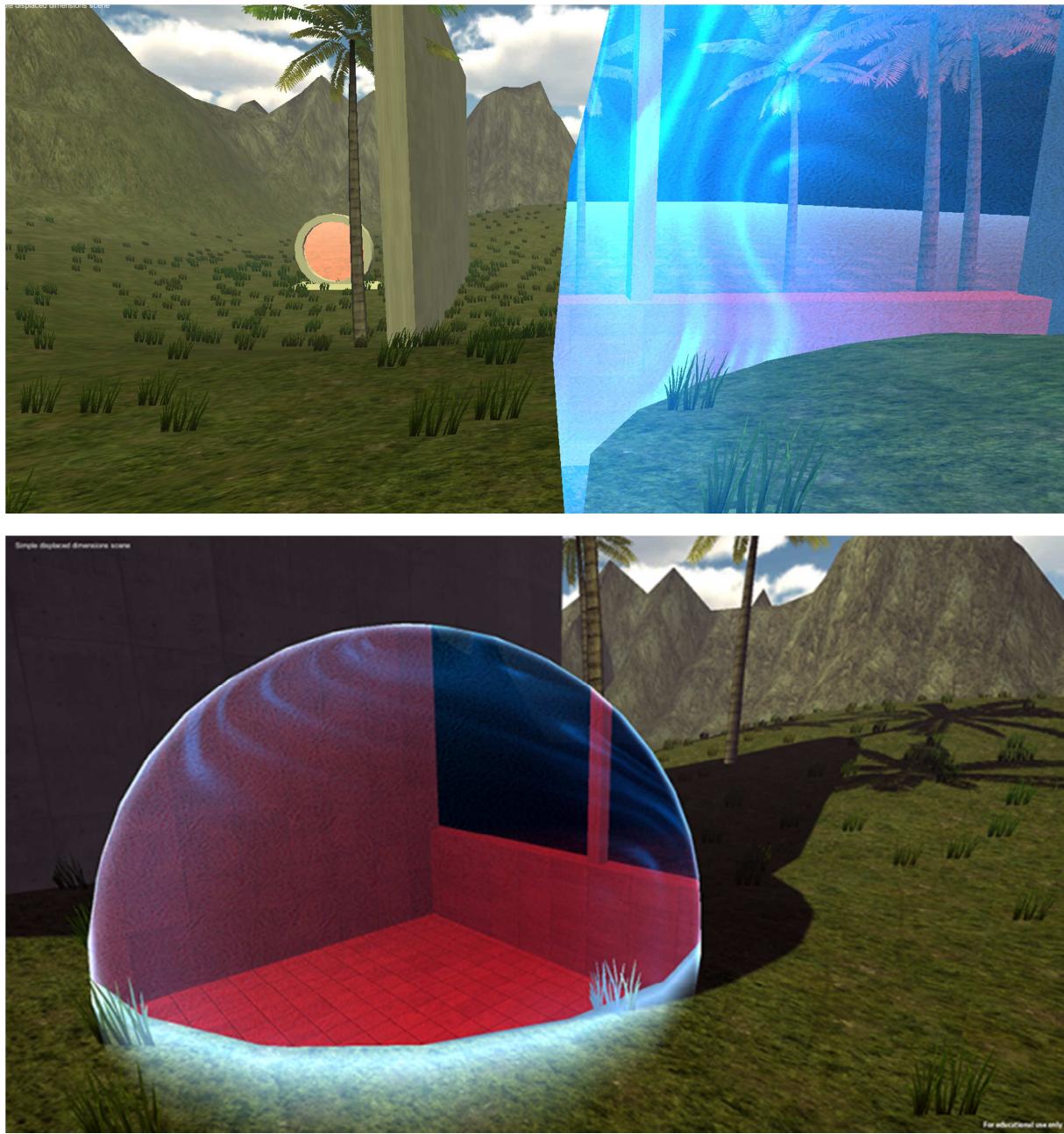


If you only want to use them for viewing, just import the default one, modify one, or create one yourself.

You can do this by Tagging a gameobject as **Portal** and assigning them to the layer **DAP_Portal**. Change the material to the added **Dimension** material. It is also recommended to add an additional material with a portal effect (e.g. blue glow). You could also add the free added **RandomMovement** script to make the portal rotate, oscillate or pulsate. Also add a **DimensionalGateController** to your gameobject. Make sure that **Can Travel Through** is disabled. Setup a **renderOffset** if you have a big gate for better visualization. The **renderOffset** should be about half the size of the closest point from center to the edge of the gate.

If you also want them for travel, you first follow the same steps as a viewing gate. Then you need to check the **Can Travel Through** checkbox of the **DimensionalGateController**. You must also choose a direction for travelling with the **Travel Direction** dropdown. You also need to add a **collider** that is checked as trigger.

4.1.1.2 Displaced Gates



If you only want to use them for viewing, just import the default one, modify one, or create one yourself.

Creating one is similar to a parallel one, however you must also set the **Displaced** field to true and the **Other Gate** field to the appropriate Gate that is the “exit”. If you don’t set an exit, it will disable the renderer. Also use a dropdown to set the **Other Reflects Layer** to the layer the portal would normally be a part of.

If you also want them for travel, you first follow the same steps as a viewing gate. Then you need to check the **Can Travel Through** checkbox of the **DimensionalGateController**. You must also choose a direction for travelling with the **Travel Direction** dropdown. You also need to add a **collider** that is checked as trigger.

If one your gate pairs rotates or pulsates, also check **Match Other Gate Rotation And Size**. Do note that this must only be done for one of the two matching portals!

4.1.2 Pro and Personal Gates

Due to the new render pipeline every portal in theory is displaced. This means that in order to have parallel gates you must place two dimensional gates at the same position. The setup is similar to that of Legacy Gates (4.1.1) with the exception that the travel direction is recommended to set to **Both, Other Reflects Layers** is ignored and that **Points Towards Dimension** comes into play.

The **Points Towards Dimension** config holds the config reference to the dimension this portal looks towards. It is important so set this in Pro and Personal or else your dimensions won't work.

4.1.3 Skybox

4.1.3.1 Legacy

In your **First Person Controller** you have a **Camholder** transform. In there you can find two camera's. The **Normal** and **Alternate Camera**. You can change the skybox here to your liking. The **Normal Camera** renders the skybox for the **Default** dimension while the **Alternate Camera** renders the skybox for the alternate dimension. If you need to access one of these camera's please see **DimensionAndPortalsController** under programming.

4.1.3.2 Pro and Personal

The skyboxes are set via the Dimensional Config Files. If you set the skybox in a custom fashion it will be overwritten.

4.1.4 Per Dimension Render Settings

4.1.4.1 Legacy

It is possible to have per dimension render settings. By adding an **RenderSettingsHandler** to the same transform as your **DimensionAndPortalsController** you can enable this option. This allows per dimension control of most render settings. By checking "Use Current Render Settings" your default dimension render settings will be cloned upon load. If you are good with programming you might be able to switch Render Settings for a single dimension. You can set more render settings in "More Render Settings", however you must call them manually from script.



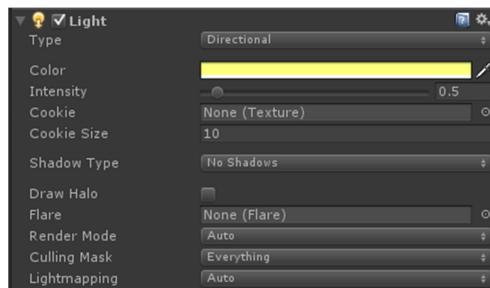
4.1.4.2 Pro and Personal

The render settings are set via the Dimensional Config Files. If you set the render settings in a custom fashion it will be overwritten. In the Dimensional Config Files you can clone the render settings to the config files if you want to.

4.1.5 Light and Special FX

It may be so that you want to render specific light and other fx per dimension. There are two options first option is through Unity self (light only), and the other is through a provided script (Light & FX).

The first option is for Unity Lights. If you select a light you can see the “Culling Mask”. If you want a light to only render in a certain dimension, you must make sure that the Culling Mask of the light is set the correct dimension(s) (Default, DefaultAlternate or DefaultAndAlternate). Do note that this does not work for Lightmapping. Lightmapping will be supported in later versions with the PRO features.



The second option is via a script. For that you need to select the object you want in the scene hierarchy and add a **Depth Switcher** to it. Here you can set the render depths the object is available. Check one or more elements in the **Render In Layers** field to specifically enable and disable the object in certain layers. For ease also set the layer to where the light or Special FX belongs to. This is only recommended for objects/fx/lights/etc. that cannot work with normal culling masks.

4.1.6 *Oculus Rift Support (Legacy only)*

If you are an Oculus Rift (OCR) developer you can use dimensional gates with OCR support. However please follow these steps carefully to setup dimensional gates with OCR support.

OCR for Unity is NOT included in this package. You can obtain that package yourself at <https://developer.oculusvr.com/>. This has many reasons (out of date packages, unwanted for non OCR developers, etc.). DAP will work fine with and without OCR if you include it yourself.

Last tested with OCR for Unity version 0.3.2.

4.1.6.1 *Instructions for the dimensional gate demo scenes*

1. Get the OCR package (OculusUnityIntegration.unitypackage) and import it into your project that also has DAP.
2. Remove the **Normal Camera** and **Alternate Camera** from the**CamHolder** of the **First Person Controller** in your scene.
3. Add two **OVRCameraController** prefabs to the **CamHolder** of the **First Person Controller** in your scene. And set their relative position and rotation to XYZ: 0 0 0.
4. Set the “Follow Orientation” of the **OVRCamera Controller Script** to their own **OVRCameraController** transform for both **OVRCameraController** gameobjects.
5. Set the tag of the first **OVRCameraController** gameobject to “CamNormal” (add if not available).
6. Set the tag of the first **OVRCameraController** gameobject to “CamAlternate” (add if not available).
7. Add to **CameraLeft** and **Camera Right** in the first **OVRCameraController** (CamNormal) a skybox that matches your main dimension.
8. Add to **CameraLeft** and **Camera Right** in the second **OVRCameraController** (CamAlternate) a skybox that matches your alternate dimension.
9. In your **DimensionsAndPortalsController** check the box: “Use Oculus Rift”.
10. Test your scene.

4.2 Additional setup for Portals

Understanding Portals is a quite complicated and has a lot of potential for customization. Besides that there are several approaches on creating and using portals in your game. So I've created the basic Portal Prefabs along with the required components and a few example scripts. So do note that these Portals behave “Valve” Portal like, but differ in some points. How you are going to use these are for yourself to fill in.

Also do note that since the portals are based on the Unity Standard principle, they do contain some “unwanted” effects that can be tackled now by level design and later on perhaps by script in a future update. Also see the limitations.

Use the left and right mouse button to shoot portals.

4.2.1 Portals

4.2.1.1 Prefab

A portal in Dimensions and Portals is a complex object. The prefab consists of a few important components:

- **Visual or Gate**

*This is the portal Gate or Fx. These are: “**PortalGateVisuals**” or “**PortalGate**” in the prefabs. The **PortalGateVisuals** have an intergrated **PortalGate**. **PortalGate** is disabled by default and can be enabled if you want your own portal style.*

- **Camera**

The camera is essential for Unity standard. The basis is cloned of the one in the First Person Controller.

- **Gate Detector**

Detects if the player is in front of the Portal and activates the possibility of “transportation”. Do note that with the standard charactercontroller you cannot seamlessly enter the portal at high speeds.

- **Back Face Mask**

A back face mask to assist the camera for the Unity standard setup.

- **Front Collider**

A collider that prevents you from moving through the object the portal is attached to when you aren't suppose to.

- **Inner Portal**

A box that allows transition from one portal to the other.

4.2.2 Scripts

4.2.2.1 Portals

The root of each portal holds two scripts. The **Portal Controller Script** and the **Simple Portal Transport Controller**. Furthermore the Camera in the portal holds the **Portal Camera Script**. The Gate Detector holds a simple pass through script named **Simple Portal Gate Detector**.

The **Portal Controller Script** and the **Portal Camera Script** are for the portal visuals. The **Simple Portal Transport Controller** and **Simple Portal Gate Detector** are for transporting players through portals. An adapted **Mouselook** script has also been added to the Camholder Transform in the First Person Controller.

It is highly recommended to create your own versions of the transportation scripts. The visuals scripts will be updated by me when necessary.

4.2.2.2 Shooting Portals

A sample script for shooting portals has been added. This is the **Player Portal Controller**. You can check if you want a small Particle FX setup or not and what prefabs to use for the Portals, Camholder and the Camera.

5 Programming

Sometimes you want to manually program certain features, here is a list of each class with the components that are of use. Do note that most Portal Based script functions are private or intended for internal use and therefor have no (usable) accessible script.

5.1 DimensionsAndPortalsController

5.1.1 Important properties (public)

- **rCam** (camera):
The camera that is viewing the default dimension (in FirstPersonController)
- **aCam** (camera):
The camera that is viewing the alternate dimension (in FirstPersonController)
- **activeGates** (List<DimensionGateControllers>) :
A list with the current active gate controllers. Important for camera rendering.

5.1.2 Important functions (public)

- **AddGateToList(DimensionalGateController dimGateController)**
Add a gate to the activeGates list.
- **RemoveGateFromList(DimensionalGateController dimGateController)**
Remove a gate from the activeGates list.
- **IsVisibleFrom(Renderer renderer, Camera camera)**
Check if a certain renderer is visible from a certain camera

5.2 DimensionalGateController

A message call “TransformWentThroughPortal” with a transform is send upon the transportation of a transform through a gate.

5.2.1 Important Functions (public)

- **isConnectedToOther()**
Detect if this is a solitary portal (overlapping scenes) or one that is connected to another one and if it is properly configured.
- **addIgnoreObject(Transform object2ignore)**
Add an object to temporary ignore with OnTriggerEnter

5.3 Camswitcher

5.3.1 Important Functions (public)

- ***listSwitcher()***
Create the list of items that are required to be switched (must be done for each camera!).
- ***IsVisibleFrom(Renderer renderer, Camera camera)***
Check if a certain renderer is visible from a certain camera

5.4 DAPPhysics

5.4.1 Important Functions (public)

- ***DAPPhysics.Raycast(***)***
The normal Physics.Raycast(***) clone that has been adapted to work with Dimensional Gates (not for portals yet!). Can be called in the same way as the original Physics.Raycast(***)�.
- ***(Ray) DAPPhysics.transformRay(Ray ray, Vector3 hit, DimensionalGateController DGC)***
Allows custom transformation of a ray by a dimension based upon a hitpoint. Can be useful for programming AI and such.

6 Known limitations

There are a few limitations to the DAP package. It is important to realize these in the setup of your project.

6.1 General

6.1.1 *Nearby gates / portals in Legacy and Personal*

Due to the technique used, it is not advised to place two gates/portals near each other. DAP tries to determine which gate/portal to render based on the view and proximity of gates/portals. If e.g. two gates/portals are in the same room, fighting over the right view can occur resulting in small glitches. It is advised to make sure your character can't see two gates when standing in the same spot and rotating around. For portals make sure two portals aren't near each other in the same plane. You can't see full visuals of other gates/portals through another gate/portal.

6.1.2 *Layers and Tags*

DAP heavily relies on correct usage of layers. DAP currently uses 6 additional layers above the default ones and requires 3 additional Tags. It is recommended to not interfere with the layers and tags.

6.1.3 Both Gates & Portals

DAP currently doesn't support both portals and gates in the same scene.

6.1.4 Character Controllers

You need the Built-in Character Controllers of Unity 4 in order to let all the demo's to work and all the scripts to compile.

Since many projects already have the Character Controller, as well as many assets also come with their own version (usually a plain copy of the original), the Character Controllers have NOT been added to the package in order to prevent cluttering of scripts.

You can get the Character Controller by using in Unity 4 → Assets → Import Package → Import Character Controller.

6.2 Dimensions

6.2.1 Dimensions

DAP currently supports 2 dimensions in Legacy and Personal. If you want to create the illusion of more than 2 dimensions, you must for now handle that yourself.

Pro supports 4-10 dimensions depending on the amount of layers used per dimension.

6.2.2 Gate collider size

If you have a gate you can walk through in **both** directions (from one dimension to the other and back), you are advised to limit the collider size so that the extends of the collider are smaller than the object going through.

6.2.3 Camera's

All camera's in the scene must have the same rendering path and be part of the first person controller in the same camera holder. This is automatically set up but can be overwritten by other systems.

6.2.4 Physics

Physics for **Portable** objects going through gates are supported for rigidbodies and charactercontrollers.

6.3 Portals

6.3.1 Portals

DAP currently supports 2 portals.

6.3.2 Deferred rendering

Due to changes in Unity 4 → 5 deferred rendering is no temporarily disabled. This will be fixed in later versions.

6.3.3 Glitchy portal look when only one portal is present

This is due to the gate not being disabled when only one portal is active. Feel free to create your own portal back texture and use the Portal Gate when the portal needs to be active.

6.3.4 Transportation

You can only transport Players with the script now. You will have to write your own implementation on transporting objects. Gimbal locking problems can occur when shooting a portal on the ceiling.

6.3.5 Physics

No realistic physics have been implemented (yet). A simple setup that looks a bit like the ones in Valve have been implemented. It is best to create these based upon your own wishes. In the example scripts it is possible to glitch out of the level.

6.3.6 Thin walls

When shooting portals on thin walls, it can be possible to see the Inner Portal of another Portal if you watch through a Portal.

6.3.7 Shooting portals partially in air

Portal shooting is based upon the collider bounds. So when you would have a rotated object, the bounds could lay outside of the mesh, thus making it possible to shoot the portal in the air.

6.3.8 Oculus Rift

There is no Oculus Rift support for Portals

6.3.9 Raycast

The raycast implementation does not work for Portals

7 Support

If you have any questions or suggestions, please use the Unity forum:

- <http://forum.unity3d.com/threads/asset-wip-dimensional-gate-portal-for-unity3d-pro-standard.191527/>
- <http://forum.unity3d.com/threads/released-dimensions-and-portals.226832/>

Asset store update url can be found here: <http://forum.unity3d.com/threads/released-dimensions-and-portals.226832/>.

You can also PM <http://forum.unity3d.com/members/108812-Gobla>, however this will not result in your question or suggestion being treated faster than the forum message.

The site <http://www.peculiardeveloper.com/> will also be updated over time with the information about the asset. However this usually is updated after the Unity Forum.