Fast Shadow Projector tutorial

Welcome to Fast Shadow Projector (FSP) tutorial! Here you will learn the basic steps involved in setting up FSP scripting package, how to use FPS's shadow projector and shadow receiver.

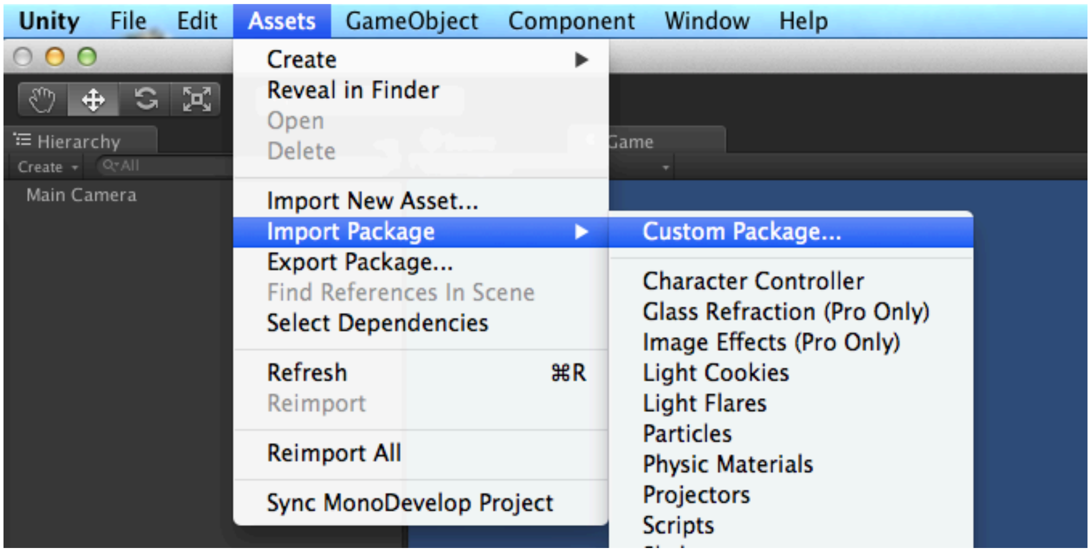
Latest online tutorial version:
http://thegrimworks.com/FastShadowProjector/site/?page_id=5

Supporting tutorial video:
https://www.youtube.com/watch?v=qxj936vDFZY

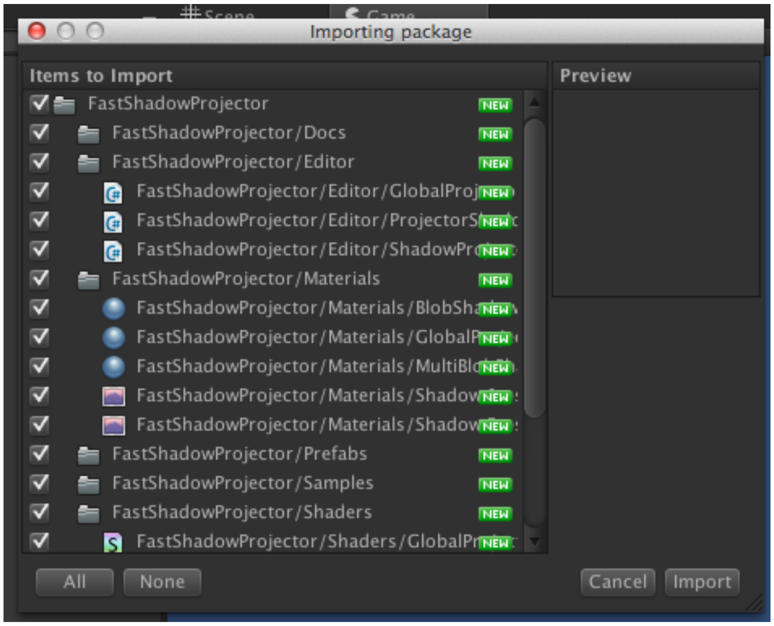
Follow us:
https://www.facebook.com/grimworksgames

Setting up

First download Fast Shadow Projector scripting package from Unity's asset store. To use it, we need to import it into our project. To do this we can either run the **FastShadowProjector.unitypackage** file or from Unity's menu choose: ***Assets -> Import Package -> Custom Package...*** and select **FastShadowProjector.unitypackage**.

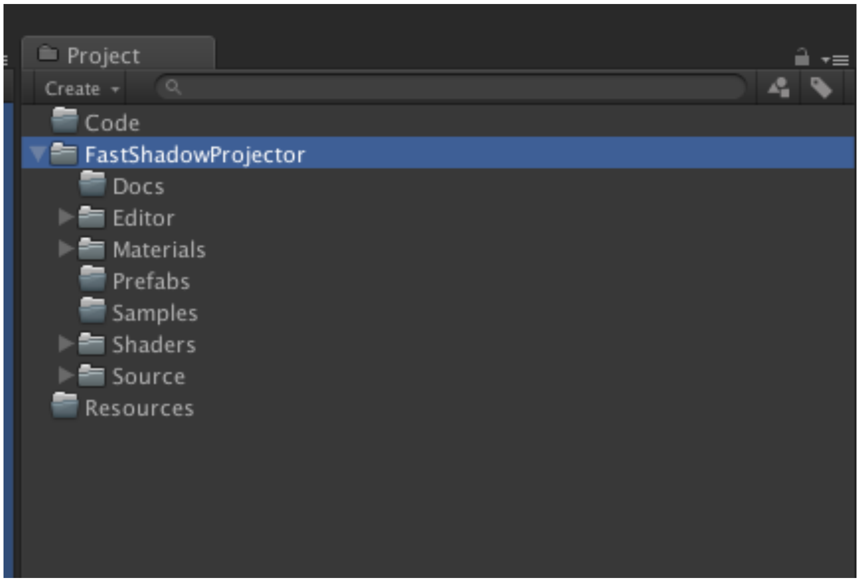


An “Import package” window will pop-up – to continue press “Import” button.



- ☑ FastShadowProjector NEW
- ☑ FastShadowProjector/Docs NEW
- ☑ FastShadowProjector/Editor NEW
- ☑ FastShadowProjector/Editor/GlobalProj NEW
- ☑ FastShadowProjector/Editor/ProjectorS NEW
- ☑ FastShadowProjector/Editor/ShadowPr NEW
- ☑ FastShadowProjector/Materials NEW
- ☑ FastShadowProjector/Materials/BlobSha NEW
- ☑ FastShadowProjector/Materials/GlobalP NEW
- ☑ FastShadowProjector/Materials/MultiBl NEW
- ☑ FastShadowProjector/Materials/Shadov NEW
- ☑ FastShadowProjector/Materials/Shadov NEW
- ☑ FastShadowProjector/Prefabs NEW
- ☑ FastShadowProjector/Samples NEW
- ☑ FastShadowProjector/Shaders NEW
- ☑ FastShadowProjector/Shaders/GlobalPr NEW

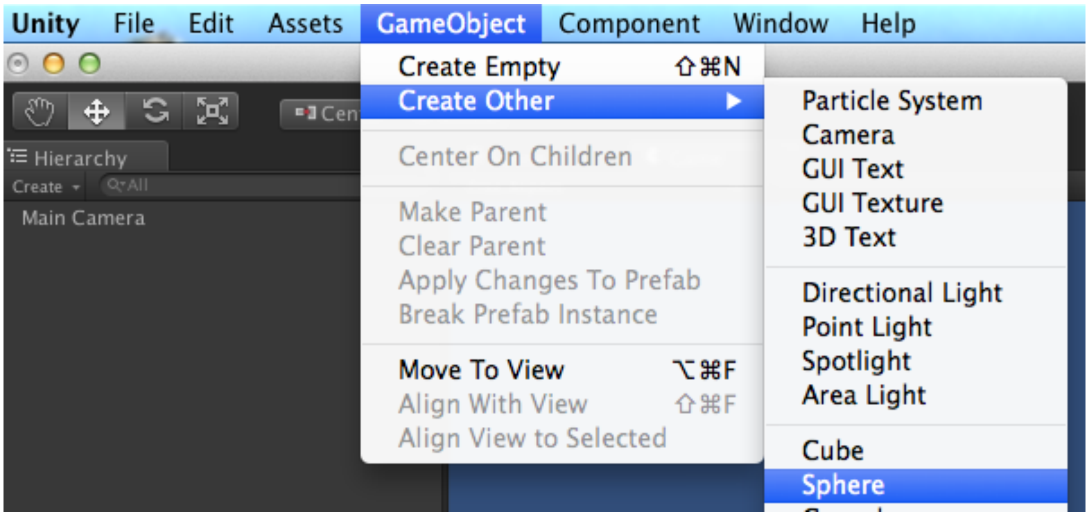
Once importing is done our project files will now include Fast Shadow Projector scripts and support files. The project panel should look something like this:



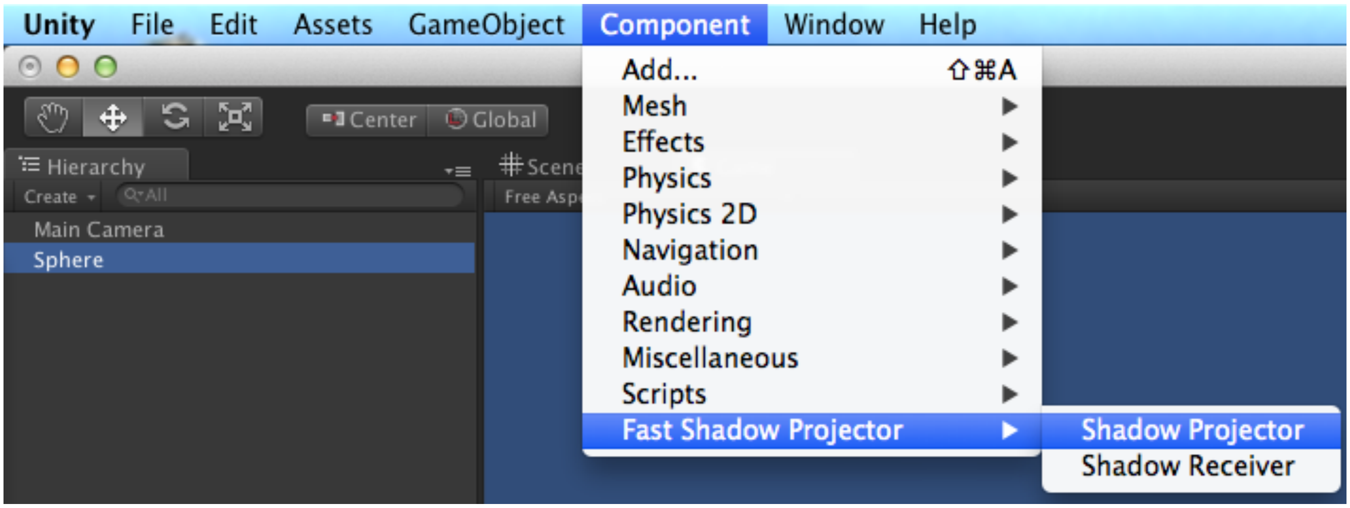
Creating a Shadow Projector

Shadow Projector is what casts a shadow for our GameObjects. For this tutorial we will create a simple sphere that casts a shadow onto a plane. Let's create our sphere:

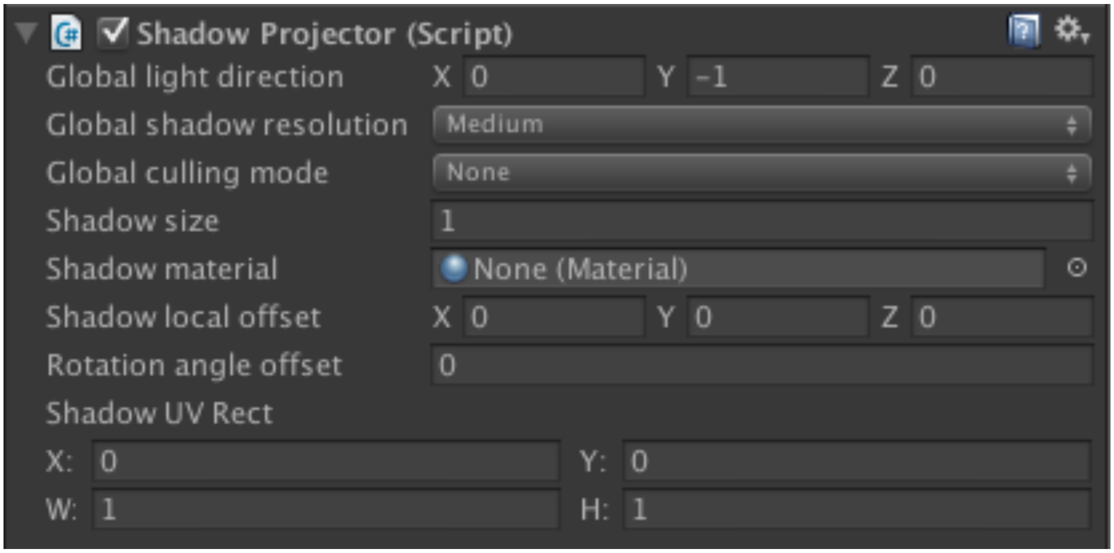
In Unity go to: ***GameObject -> Create Other -> Sphere***



Now let's select our sphere in the Hierarchy panel and add a Shadow Projector by going to: ***Component -> Fast Shadow Projector -> Shadow Projector***



Shadow Projector component settings will now appear in our sphere's Inspector window:

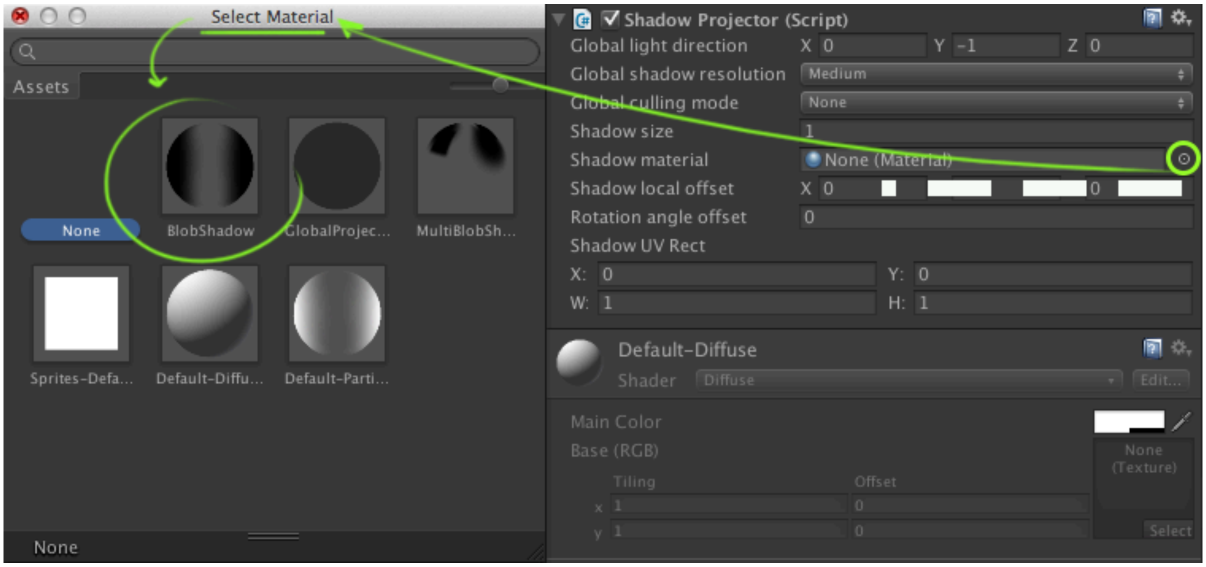


☑ Shadow Projector (Script)

Global light direction X 0 Y -1 Z 0
Global shadow resolution Medium
Global culling mode None
Shadow size 1
Shadow material None (Material)
Shadow local offset X 0 Y 0 Z 0
Rotation angle offset 0
Shadow UV Rect
X: 0 Y: 0
W: 1 H: 1

Notice that the Shadow Material property is "None (Material)". We need to set this otherwise the shadow won't render. To keep this tutorial brief we will only set the Shadow Material option which will give texture to our shadow (all Shadow Projector settings are detailed here).

Let's select the BlobShadow material that comes with Fast Shadow Projector package. To do this click on the circle icon next to "None (Material)" field, this will open "Select Material" window and double click BlobShadow.



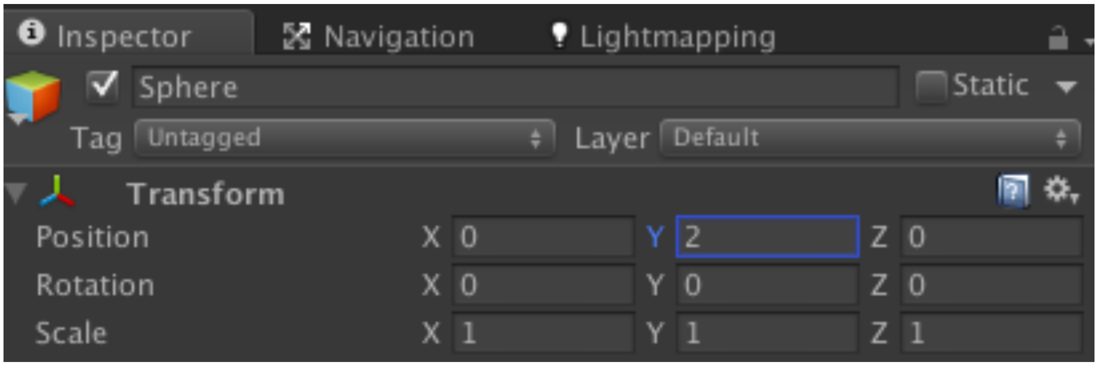
☑ Shadow Projector (Script)

Global light direction X 0 Y -1 Z 0
Global shadow resolution Medium
Global culling mode None
Shadow size 1
Shadow material None (Material)
Shadow local offset X 0 0
Rotation angle offset 0
Shadow UV Rect
X: 0 Y: 0
W: 1 H: 1

Default-Diffuse
Shader Diffuse Edit...

Main Color
Base (RGB) None (Texture)
Tiling Offset
x 1 0
y 1 0
Select

In Inspector window of the sphere, let's change the Y coordinate to 2 so that the sphere "hangs" above ground.



☑ Sphere □ Static

Tag Untagged Layer Default

Transform

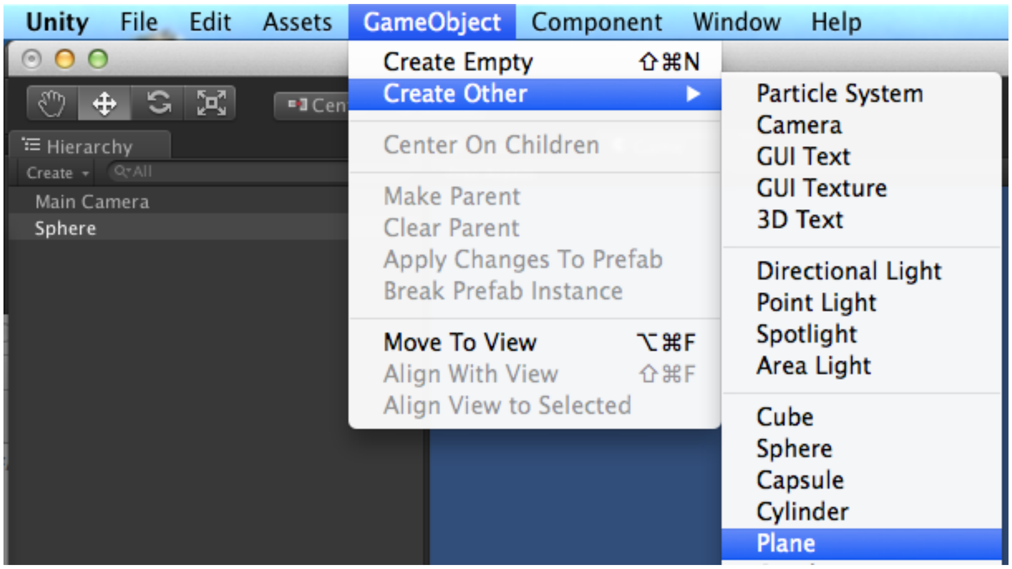
Position	X	0	Y	2	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

Our sphere and Shadow Projector are ready. Now we only need to setup our Shadow Receivers.

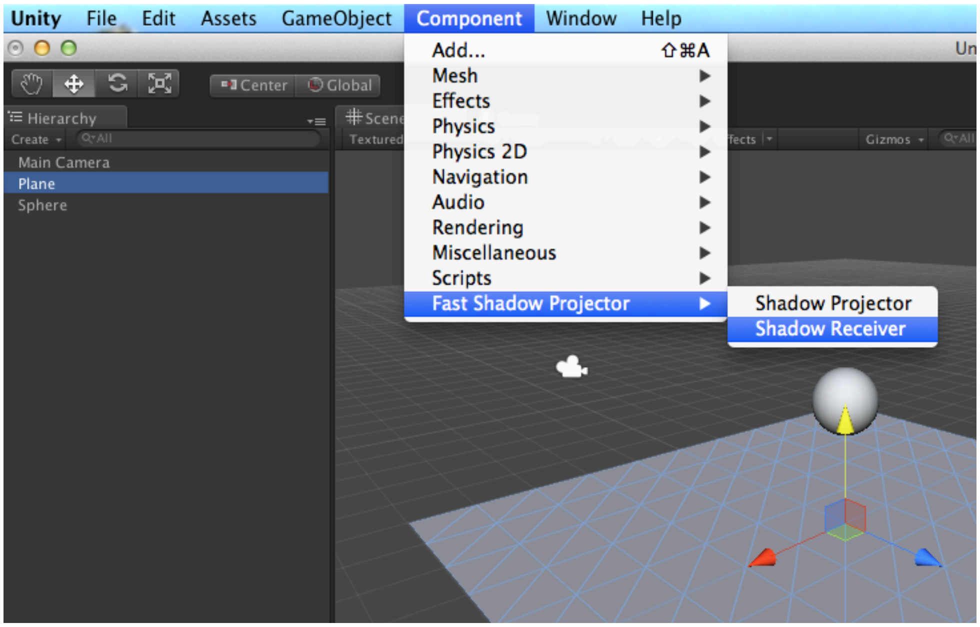
Creating a Shadow Receiver

Shadow Receivers should be added to every GameObject on which shadows should be rendered (street mesh, floor, etc). For this tutorial we will create a plane that is able to receive shadows.

Let's create our plane. Go to: ***GameObject -> Create Other -> Plane***



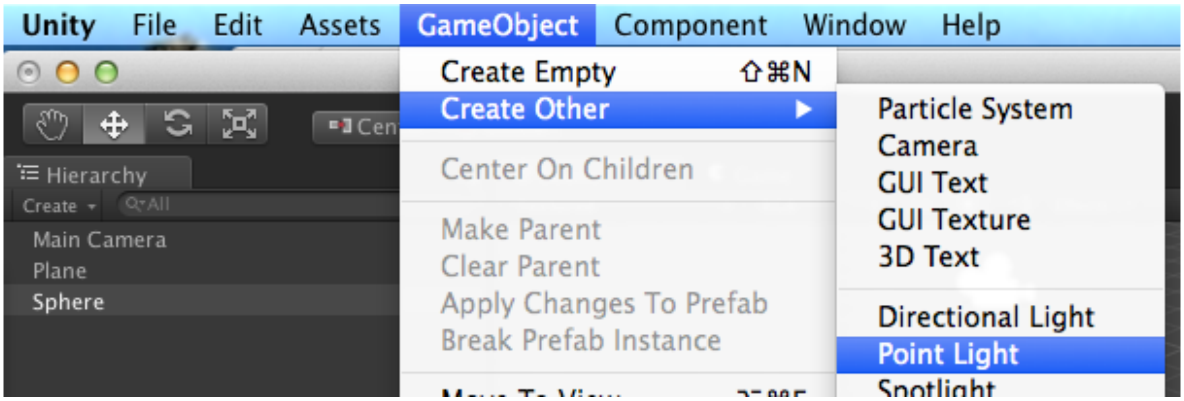
Now let's select our sphere in the Hierarchy panel and add a Shadow Receiver by going to: ***Component -> Fast Shadow Projector -> Shadow Receiver***



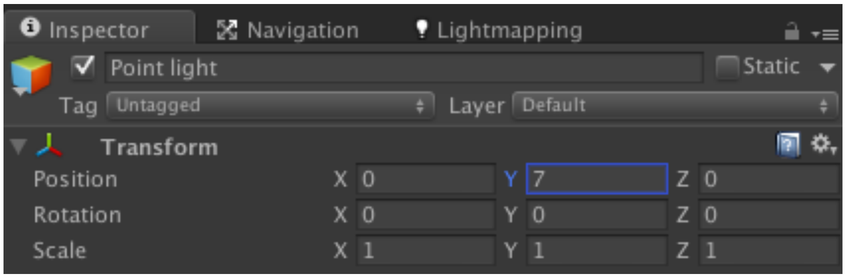
And that's it! We now have a blob shadow being cast from our Sphere onto a plane. Run the project to see the effects.

Optional: add lighting!

To finish off this tutorial, let’s add a Point Light so we can see the objects and shadow in our scene better. Go to: ***GameObject -> Create Other -> Point Light***



In Inspector window of the Poing Light, change the Y coordinate to 7 just to make the scene look better.



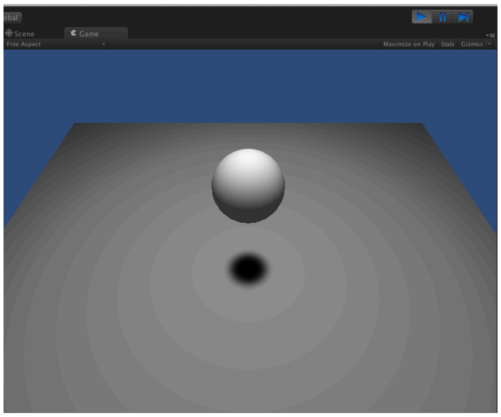
☑ Point light

Tag Untagged   Layer Default

Transform

Position	X	0	Y	7	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

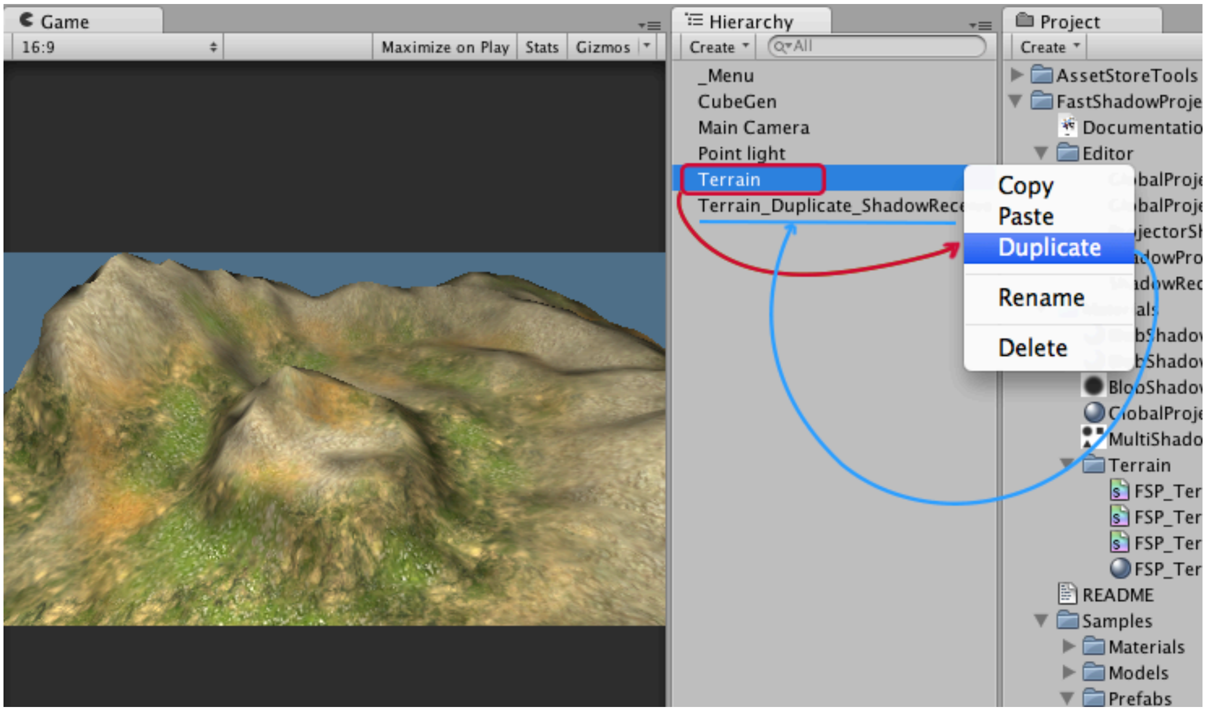
Now let’s run the project to see the results:



Congratulations! You now know how to use the Fast Shadow Projector scripting package for Unity! If you have any questions, please contact us: **support** [AT] **thegrimworks** [DOT] **com**.

Using Fast Shadow Projector with Unity's terrain

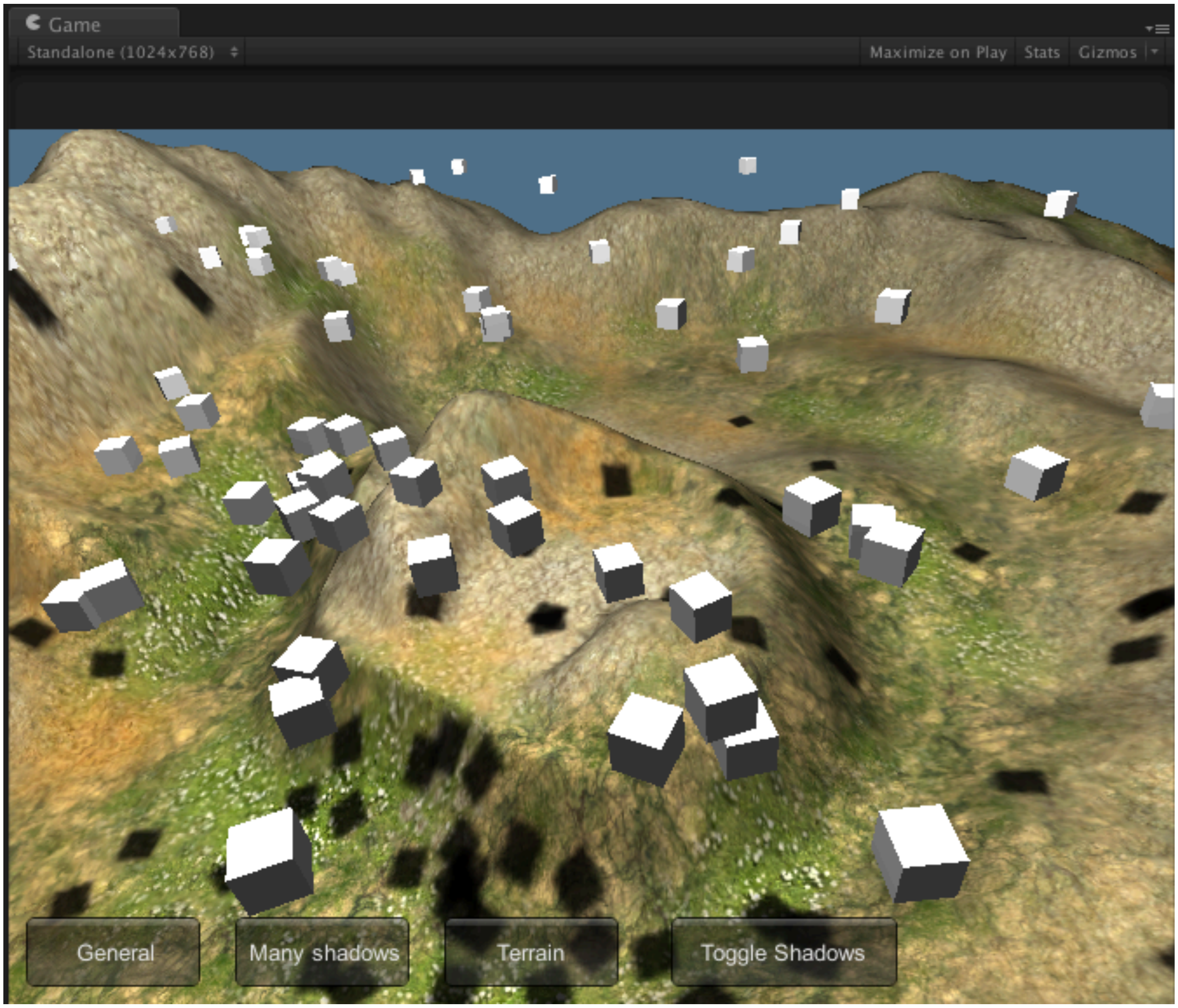
If you are using **Unity 4.0 Pro** or above, you can use Fast Shadow Projector with Terrains generated inside Unity. To enable this functionality the steps are a bit different than using with Mesh Renderers. Once you have your Terrain up and ready, you have to **duplicate** it.



Then select your duplicate and go to: ***Component -> Fast Shadow Projector -> Shadow Receiver***. You will notice that the duplicate terrain became invisible. That is OK – Fast Shadow Projector hides it automatically so that you can continue editing/adjusting your original terrain.

Important note: you can continue texture painting your original Terrain as well as setting/lowering/smoothing the heights, as both of these Terrains are linked regarding their vertex data. Other parameters, such as "Base Map Dist", etc. are NOT linked and different values on original and duplicate terrain may result in shadows suddenly disappearing. Make sure such parameters always match.

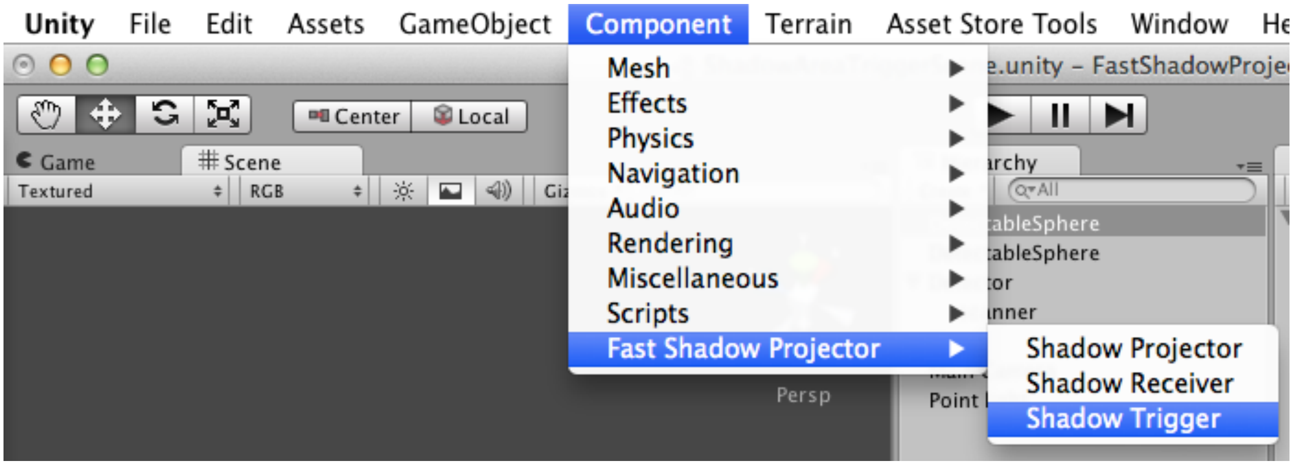
Run your scene and you should see shadows on your terrain:


Game
Standalone (1024x768)
Maximize on Play
Stats
Gizmos
General
Many shadows
Terrain
Toggle Shadows

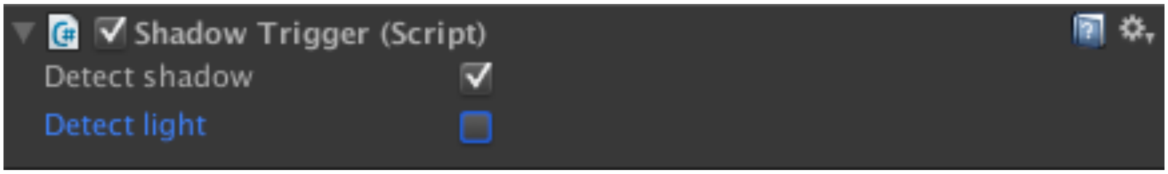
Shadow Triggers

Shadow triggers provide means to determine whether or not a Game Object is inside the area of a projected shadow/light/etc. This may be useful in numerous situations, for example, you might want to dim the direct lighting your Game Object's mesh receives once it enters the shadow area or you want to target an enemy once it enters the Area of Effect of your character.

Shadow Triggers should be added as components to Game Objects that need to receive callbacks when they Enter, Stay or Exit a shadow projected by a Shadow Projector. To add a Shadow Trigger select: ***Component -> Fast Shadow Projector -> Shadow Trigger***



Once you add Shadow Trigger to your Game Object you will notice this component in the Inspector view:



☑ Shadow Trigger (Script)

Detect shadow ☑

Detect light ☐

There are two parameters:

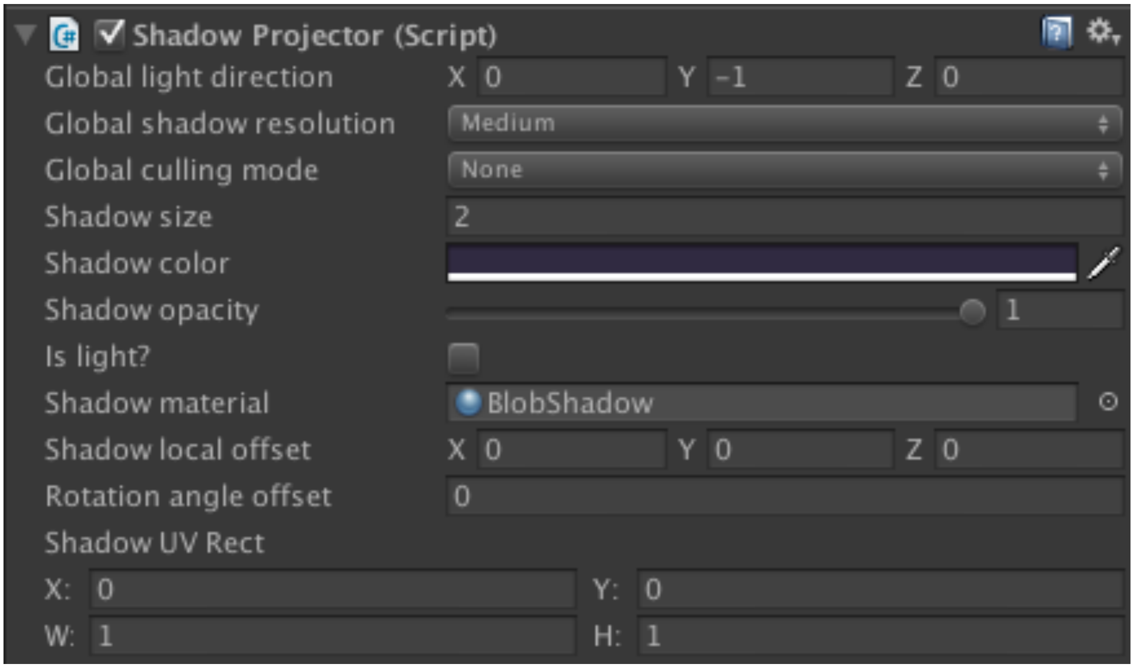
Detect Shadow: if checked, this Game Object will receive callbacks when it Enters, Stays, or Exits the shadow of a Shadow Projector which has the **Is Light** parameter **unchecked**.
Detect light: if checked, this Game Object will receive callbacks when it Enters, Stays, or Exits the shadow of a Shadow Projector which has the **Is Light** parameter **checked**.

Once you've set up your Shadow Trigger in the inspector view, you need to register the Shadow Trigger callbacks you want your Game Object to receive. You can do this by using the following lines:

```
ShadowTrigger shadowTrigger;

void Start () {
     shadowTrigger = GetComponent<ShadowTrigger>();
     shadowTrigger.OnShadowEnter = OnShadowEnter;
     shadowTrigger.OnShadowStay = OnShadowStay;
     shadowTrigger.OnShadowExit = OnShadowExit;
}

void OnShadowEnter() {
     // Your code that handles shadow Enter event...
}

void OnShadowStay() {
     // Your code that handles Stay in shadow events...
}

void OnShadowExit() {
     // Your code that handles Exit event...
}
```

Shadow Projector component details

This is how the Shadow Projector component looks in the Inspector view.



☑ Shadow Projector (Script)

Global light direction X 0 Y -1 Z 0

Global shadow resolution Medium

Global culling mode None

Shadow size 2

Shadow color

Shadow opacity 1

Is light? □

Shadow material BlobShadow

Shadow local offset X 0 Y 0 Z 0

Rotation angle offset 0

Shadow UV Rect

X: 0 Y: 0

W: 1 H: 1

The first three settings are global. Changing a global setting will change that setting for every Shadow Projector existing in the scene.

Global settings

Global light direction: the direction of shadow cast. ***Note:*** this setting implies that there can be only one shadow cast direction for every object at a given time. This setting can be changed at runtime any time, but changing it will affect every Shadow Projector component. This is where most of the performance gain comes from.
Global shadow resolution: This describes the quality of the shadow. For most scenarios setting to ***Medium*** or***High*** is enough. But if you see pixelation in the shadows in large scenes try setting it to ***Very High*** though this comes at a performance price.
Global culling mode: If you are not culling/disabling your objects once they become invisible to the camera, Fast Shadow Projector can at least cull the shadows it's casting. ***Caster bounds*** – culls the shadow once the casting object becomes invisible to the camera (very effective for top-down games). ***Projector bounds*** – culls the shadow if the casting object's bounds intersect with the shadow projection bounds.

Non-global settings

Shadow size: use this setting to adjust the size of the shadow without changing object's scale.
Shadow color: additionally tints the shadow texture.
Shadow opacity: adjusts the translucency of the shadow. Typical usage: lower opacity the higher the object gets from the ground.
Is light?: If not checked all shadows are applied onto receivers using a Multiply type material, if checked – using an Additive material. Additionally, Shadow Triggers use this parameter to determine whether the shadow of this projector should trigger a callback.
Shadow material: sets the shadow's material. It is highly recommended that you use as few materials as possible or even better – use a material with a texture atlas.
Shadow local offset: offsets the point from which the shadow is cast.
Rotation angle offset: offsets the rotation/orientation of the resulting shadow. Typical usage: matching the initial orientation of a cube with it's quad shaped shadow.
Shadow UV Rect: the UV coordinates to use for the shadow's texture. Typical usage: specify texture rect when using a texture atlas.
For more info, please contact us: **support** [AT] **thegrimworks** [DOT] **com**