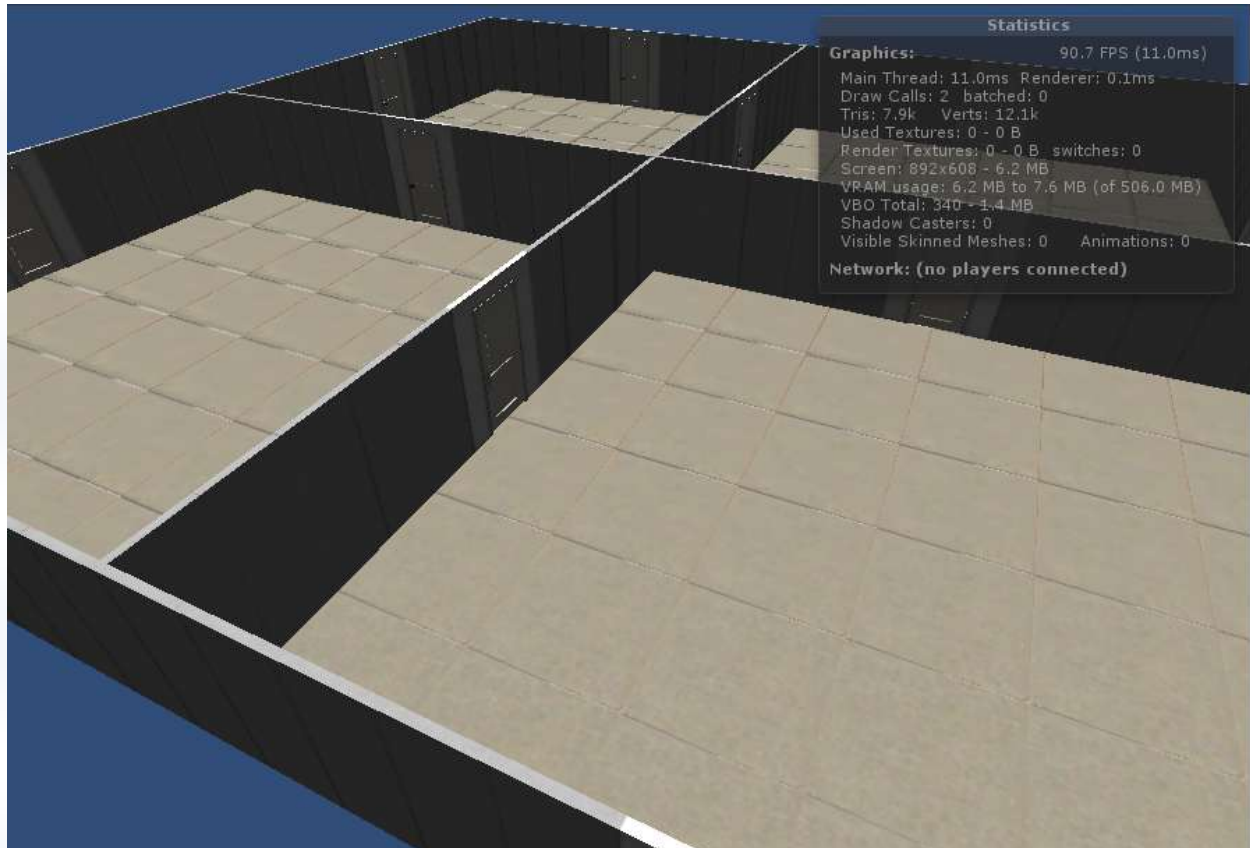


Draw Call Minimizer

Version 1.3



Contents

- Introduction
- How To Use
- What is to come
- Contact Information

Introduction

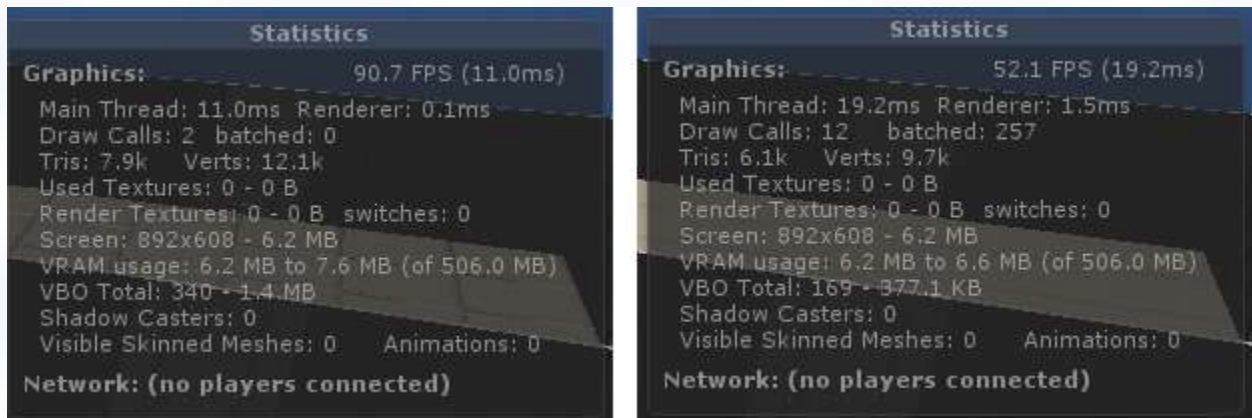


Figure 1: Left using optimized script, right without using script. Note the rendering time and CPU time

In short, what Draw Call Minimizer is, is a tool that combines meshes and textures together to make one draw call instead of multiple ones.

The reason that this becomes helpful is because as more and more objects are on the screen, the more draw calls get made to the GPU. As more draw calls get made to the GPU, the list grows and grows, and it becomes a large bottleneck for performance.

Every frame, the CPU sends a draw call for every object that is being rendered to the GPU. Now if you have 100s of objects being rendered, that is 100s of objects being sent to the GPU every frame, which is time that could be spent doing other things such as AI code etc.

Now, if you were to combine everything, and turn all of your meshes into one giant mesh, and all of your textures into one giant texture, that removes that bottleneck. The CPU needs to send one draw call now instead of 100s of them, and the time is now spent doing other things, such as that AI code, or just bumping up the frame rate.

How to Use

Using the Draw Call Minimizer is very easy, and creating an easy experience was one of the main focuses of development. There are only a few short steps to get this tool in your project and up and running and the next few pages are going to show you how.

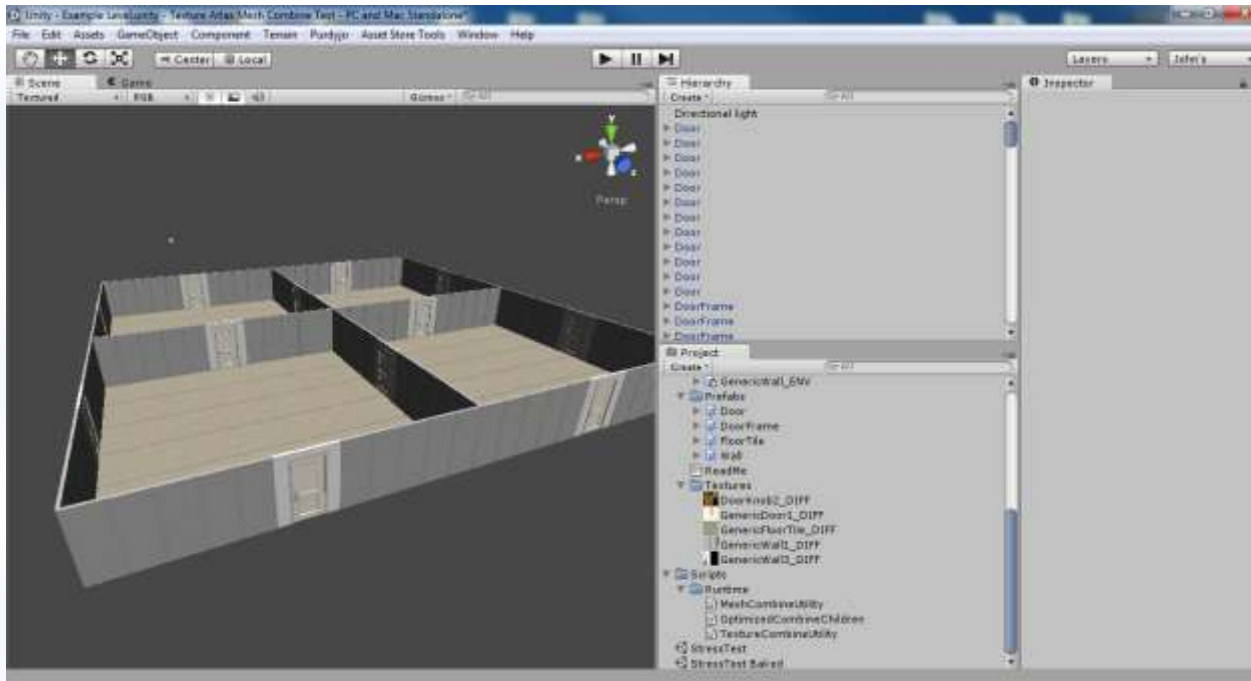


Figure 2: Example, a basic level

The picture above is a very simple level. It uses about 5 different meshes and 5 different textures. When run, the level has about 12 draw calls, and Unity dynamically batches about 257 of them. This dynamic batching, while it reduces rendering time, it takes up CPU time.

Draw Call Minimizer batches everything right at the start of the level, so instead of having 12 draw calls and 257 batched, it cuts it down to 2 draw calls and 0 batched. There is less rendering time AND less CPU time.

The way that this script works, is that you have a parent object that everything being combined is childed under. You attach the script to the parent object, fiddle with the settings if you need to, and play the game. All of the combining is done at run time so no extra assets are created.

Now before I continue ranting, I am going to show you step by step how to get this thing going.

Now the first thing that I want to make sure you guys know, is that if you want to use this tool properly, you need to make sure that every texture being used by the combined meshes are set to Read/Write Enabled.

The script needs this to gather the data to combine all of the textures together.

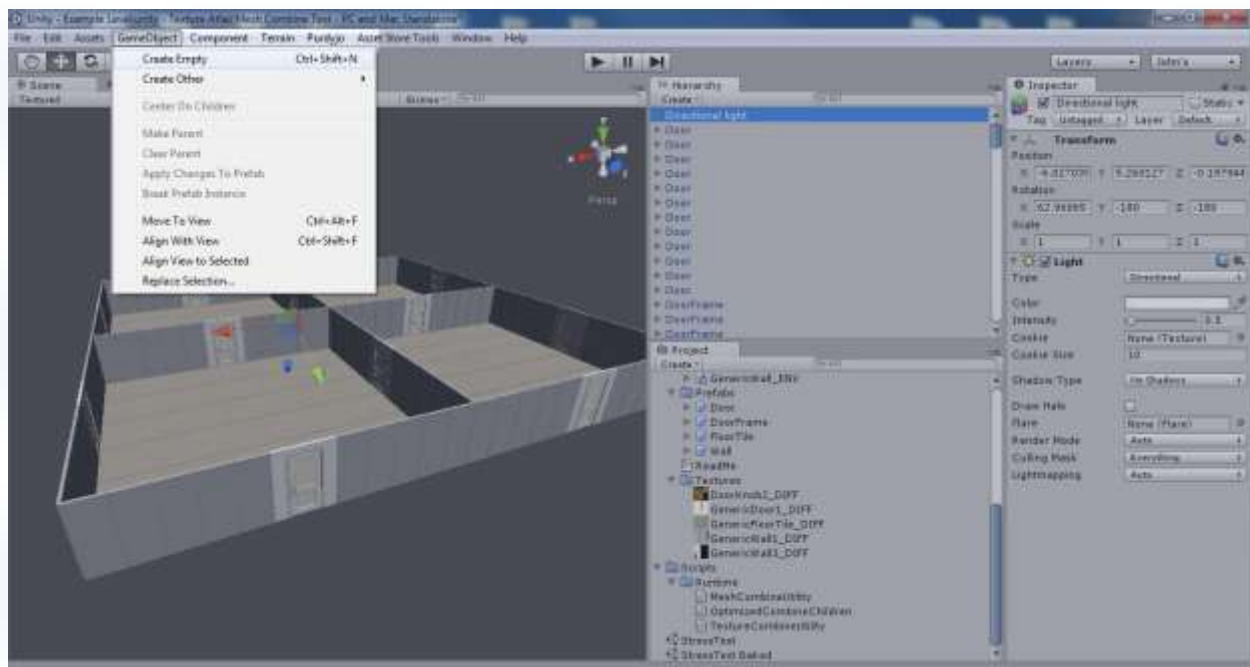
Another thing to remember is that the Unity default shader and texture is not accessible, and the script will fail if it is used. A work around is to create basic texture to place on everything that is not currently completed (A basic 1 x1 white pixel texture will do).



Alright, so now that this is under control let's begin.

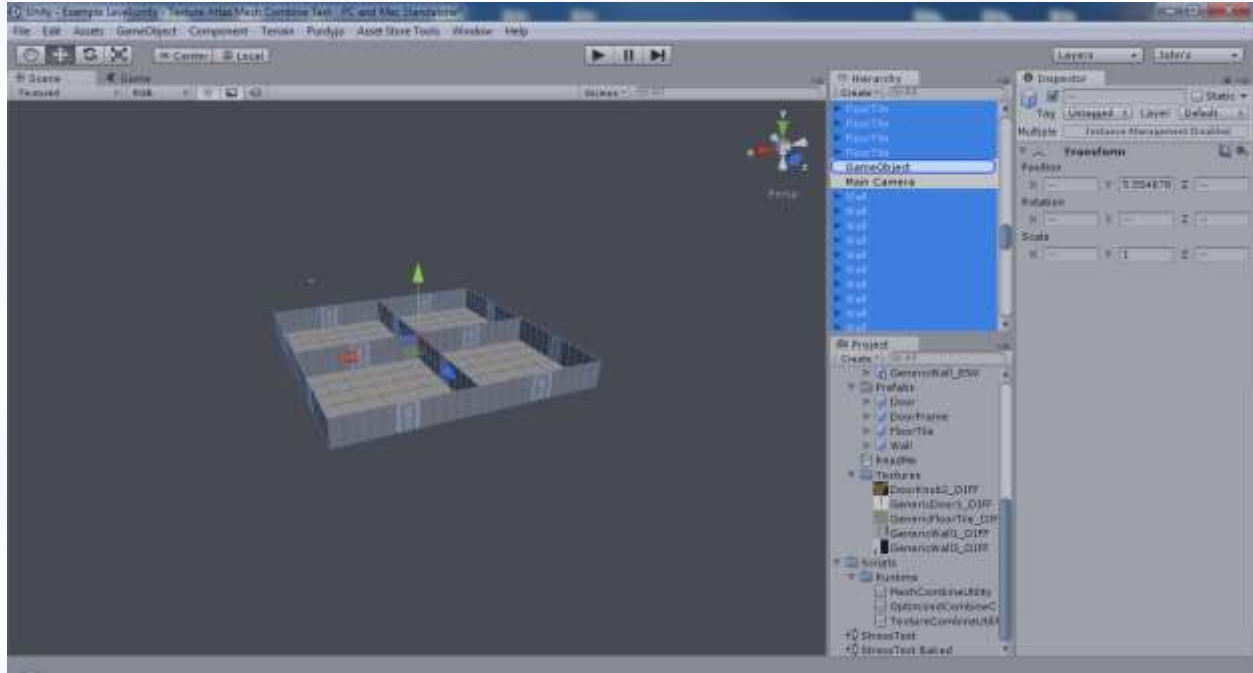
Step 1: Create an empty GameObject

Go to GameObject > Create Empty



Step 2: Child everything under the new GameObject

So basically all you want to do now is select everything that you want combined together and drag it onto the newly created GameObject to child it.



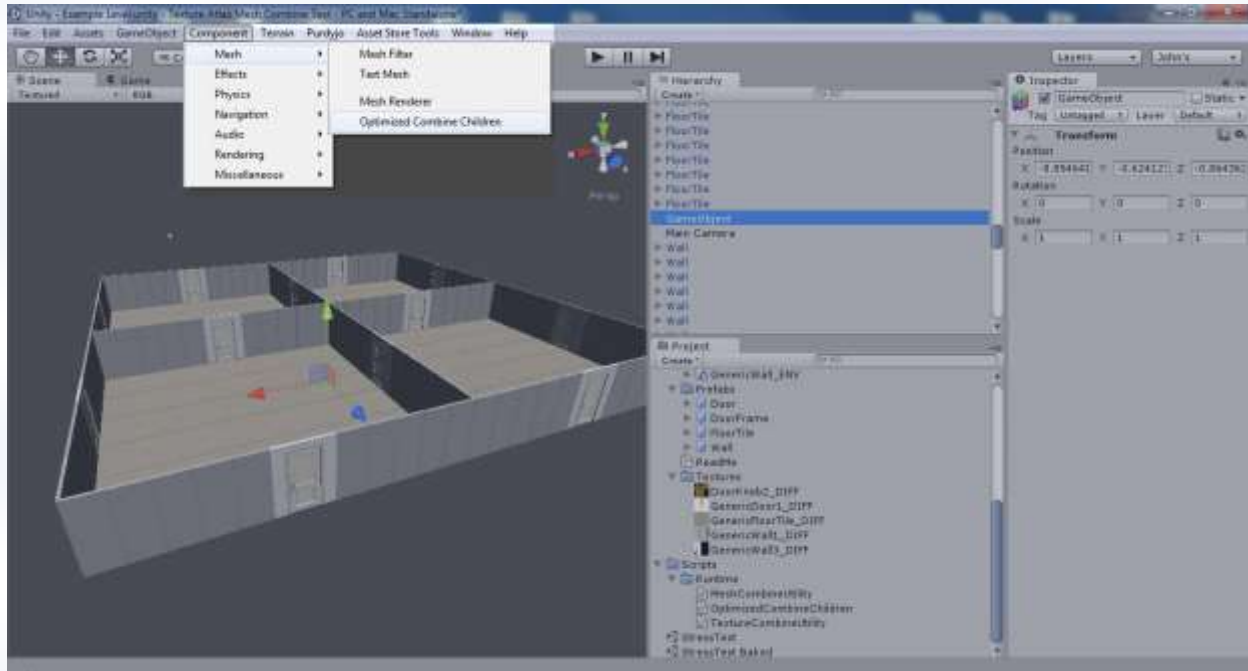
Something that you can keep in mind is that if you want to take advantage of the occlusion culling, combining everything into one mesh is not the way to go.

Since occlusion culling takes the bounding box of the mesh to determine if something is visible, one giant mesh and one giant bounding box makes the occlusion culling useless.

Combining each room in a building on the other hand still takes the occlusion culling into account and you also still get the optimization bonuses from this tool.

Step 3: Add Optimized Combine Children to the new GameObject

The script being used is called `OptimizedCombineChildren`, and you can either drag it from the project, or go to the menu **Component > Mesh > Optimized Combine Children**



You might notice that the way this script runs is similar to the Unity's **Combine Mesh** tool. That is because the very first version of this software was an optimized version of it.

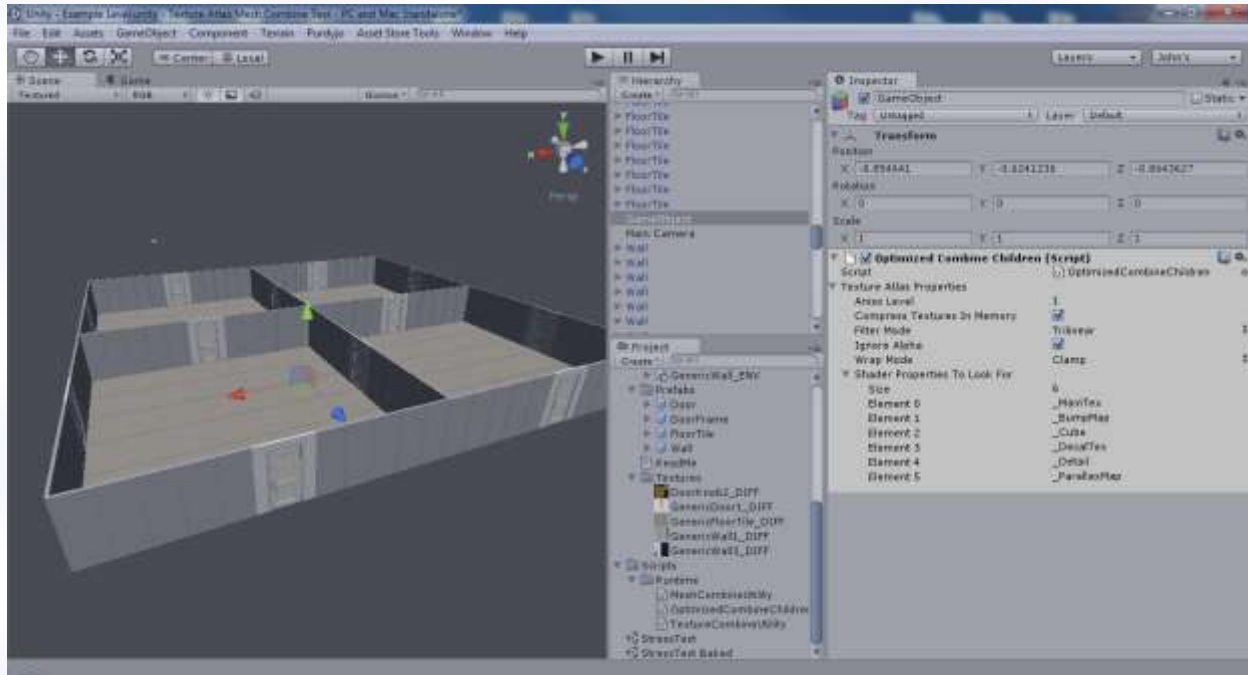
The difference between the two is that Unity's **Mesh Combine** tool only combines meshes and not textures. The performance increase from doing that is not very large, and there are still multiple draw calls that can be taken out.

Draw Call Minimizer on the other hand combines all meshes and creates as few textures and shaders as possible.

This is useful and an example of it being so is that if you use only **Diffuse** shaders, and combine everything under that one parent, it will make one giant texture atlas and one giant mesh, effectively making one draw call.

Step 4: Checking out the settings

Draw Call Minimizer also has settings that you can fool around with, most of which being for the texture atlas being created. Once you add the script to the parent object you will see the settings like so:

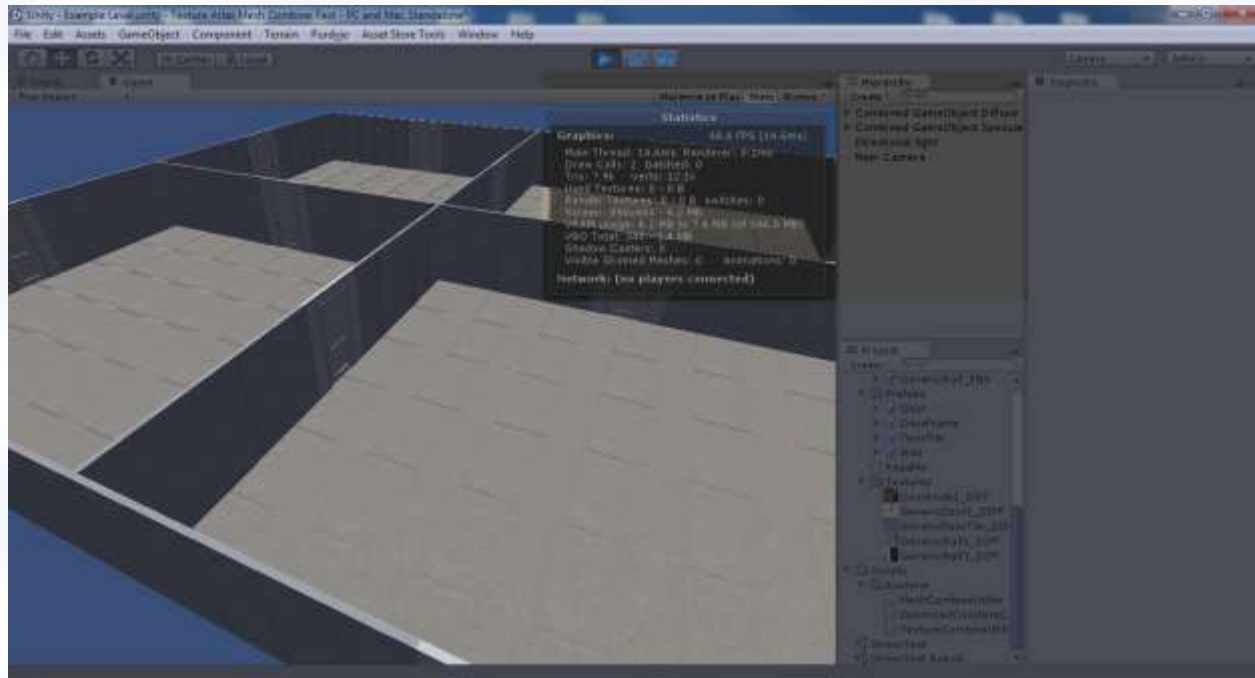


Texture Atlas Properties

- **Aniso Level**
 - The Anisotropic filtering level. The higher it is, the better it looks at an angle from a distance(Reduces blur)
- **Compress Textures In Memory**
 - I give you that option. Compressing the texture atlas is what takes up the most time with the tool, and if you are ok with a larger texture in memory, want a shorter run time, or just don't want a slight loss of quality, you can turn it off
- **Filter Mode**
 - The filtering mode used to blend mip-maps. (Point, Bilinear, Trilinear)
- **Ignore Alpha**
 - If you have alpha in your textures but don't need it, you can just tell the tool to ignore it and save performance and reduce the size of the texture atlas
- **Wrap Mode**
 - The wrap mode for your texture atlas. It really doesn't make a difference, since if you are using moving textures, including it in the texture atlas will just mess it up
- **Shader Properties to Look For**
 - This is a nice feature that was just added in 1.3. All Shaders are now supported, even custom ones. The only thing you need to do is tell it what texture properties you added to the shader and it will take care of the rest. The properties used in all of Unity's textures are included automatically, and you can remove the ones not needed to shave off some time that this tool takes to run.

Step 5: Run your game

That is all it takes to get this up and running. Once you run the game you will notice new combined objects and reduced draw calls.



Contact Information

If you have any questions at all, whether it is about why something does what, or where something is, or when a new feature is being added, do not hesitate to contact me.

I am giving you the most common places that I check, so that if you have a question I can answer it as soon as I get it.

Email: johnjrpurdy@gmail.com

Twitter <https://twitter.com/#!/purdyjo>

Draw Call Minimizer Blog Page:

<http://purdyjotut.blogspot.ca/p/draw-call-minimizer.html>