DX1219: PRACTICAL 2

The aim of this practical is to build upon your knowledge from practical 1 and add in textures as well as supporting multi textures.

1. Some of your codes might differ if you have done the extra stuff. It is recommended that you create a new shader file. Remember to update the shader name and the material.
2. In order to make the code more self-explanation I have change the code to the following

```
struct vertexData {
    float4 position: POSITION;
    float2 uv: TEXCOORD0;
};

struct vertex2Fragment {
    float4 position: SV_POSITION;
    float2 uv: TEXCOORD0;
};
```

3. I changed the vertex shader to this

```
vertex2Fragment MyVertexShader(vertexData vd) {
```

And the fragment shader to this

```
float4 MyFragmentShader(vertex2Fragment v2f) : SV_TARGET{
```

vertexData will be the struct that is send from the **CPU** to **GPU** to the **vertex shader**.

vertex2Fragment will be the struct that send from **vertex** shader to the **fragment** shader

4. Update the vertex shader to change the return type to the new struct and update the function parameter to **vertexData** too.

```
vertex2Fragment MyVertexShader (vertexData vd)
{
    vertex2Fragment v2f;
    v2f.position = UnityObjectToClipPos(vd.position);
    v2f.uv = vd.uv;
    return v2f;
}
```

5. We will now add in texture to the material, similar to how we are able to change the color, we are able to change the texture accordingly.
    a. Add this to the properties to allow the adding of texture in the material.

```
Properties
{
    _mainTexture ("Texture", 2D) = "white" {}
    _tint ("Tint", Color) = (1,1,1,1)
}
```

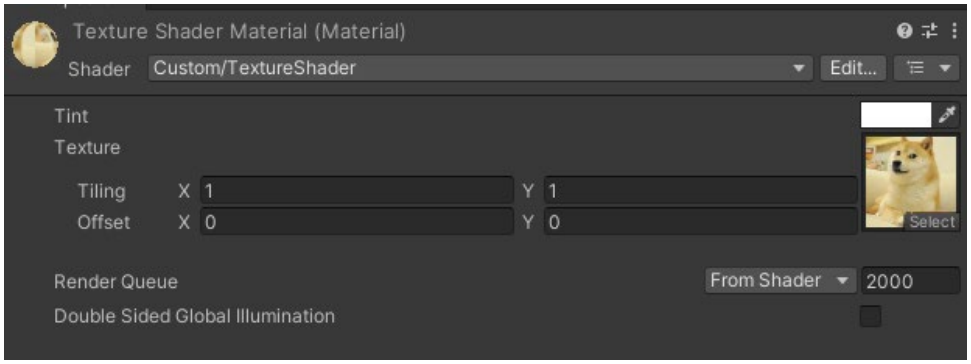
    b. **sampler2D** is used to bound to a texture unit

```
uniform float4 _tint;
uniform sampler2D _mainTexture;
```

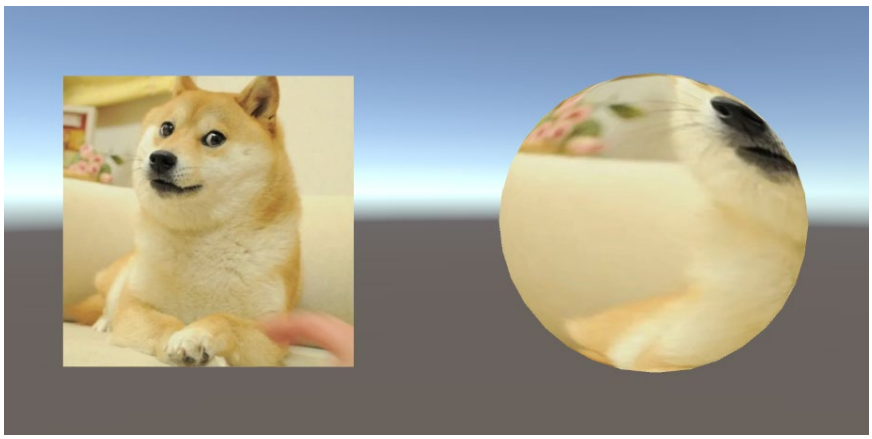
    c. We can make use the sampler2D and the tex2D function to look up the pixel based on the texture and the uv value. Every pixel will have a slight different uv unless they share the same uv value.

```
float4 MyFragmentShader(vertex2Fragment v2f) : SV_TARGET
{
    return tint * tex2D(_mainTexture, v2f.uv)
}
```

6. Add a texture to the material.



7. Congratulation, you have texture and color shader. This should be the scene currently.

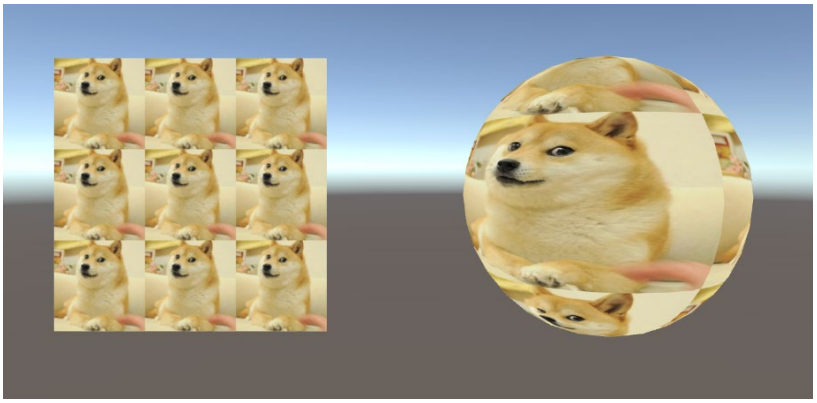


8. Now let do make use of the offset and tiling. Add the following line. It is **IMPORTANT** to add in _ST, although it refers to Scale and Translation due to Unity backwards compatibility. However, it is important for **tiling and offset**.

```
uniform float4 _tint;
uniform sampler2D _mainTexture;
uniform float4 _mainTexture_ST;
```

Add the following to the vertex shader. The tiling vector is used to scale the texture hence it is **(1,1)** by default. We could do it both in the vertex or fragment shader however it would make sense to do it in the vertex shader since we are just scaling each vertex and it will automatically be interpolated in the fragment shader. But if you wish to control the uv more then you will need to do it in the fragment shader. The tiling is set in the .xy portion.

```
v2f.uv = vd.uv * _mainTexture_ST.xy;
```



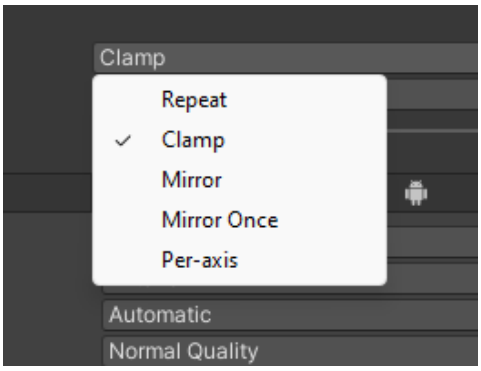
9. Now let work with the offset. The offset portion is stored in the .zw portion of the variable. Remember to add it after scaling.

```
v2f.uv = vd.uv * _mainTexture_ST.xy + _mainTexture_ST.zw;
```

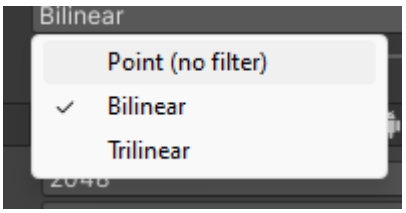
10. We can make use of a boilerplate in UnityCG.cginc, TRANSFORM_TEX that does the same thing as our above code so we can subsitute our code like this.

```
v2f.uv = TRANSFORM_TEX(vd.uv, _mainTexture);
```

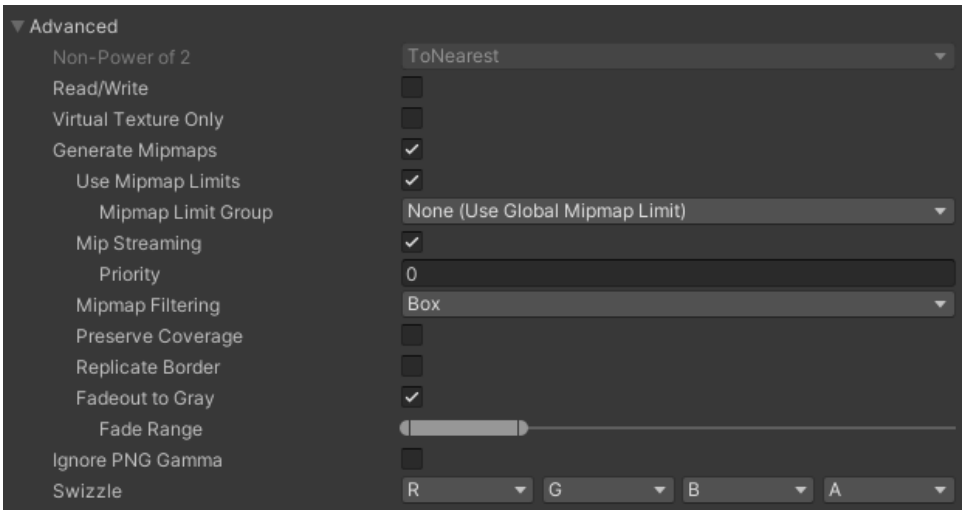
11. Select the texture try changing the texture **wrap** type to see how it will change.

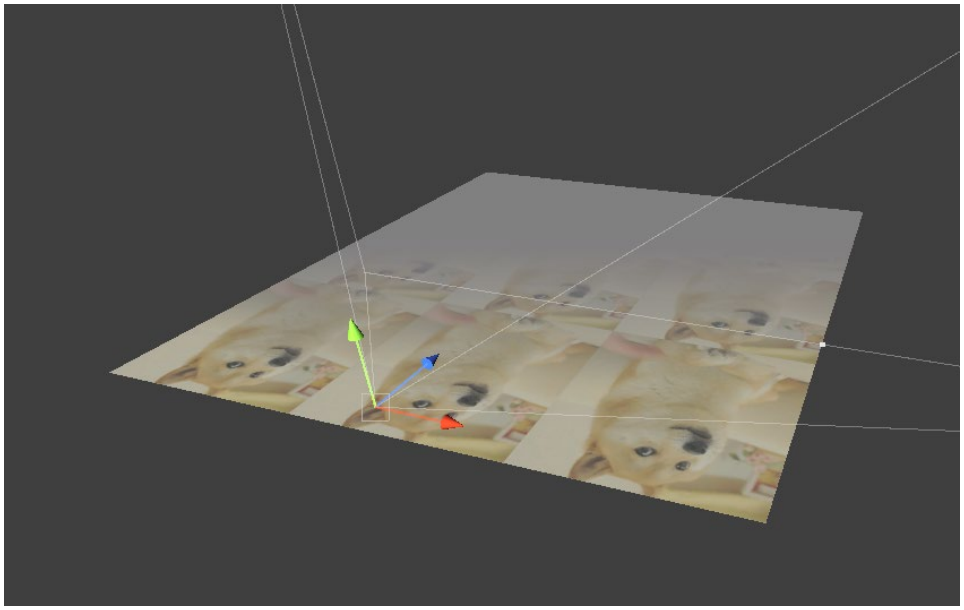


12. Try out changing the **filtering** setting to see if there is any changes.



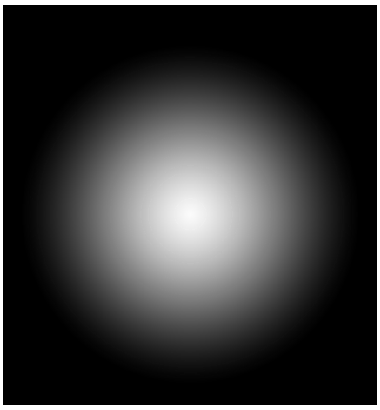
13. Trying out the mipmaps feature as well. Create a large plane with a huge scale set it flat on the ground. Select **Advanced** and play around with the features. Try turning on and off the mipmap by checking and unchecking **Generate Mipmaps**. Try setting the **Fadeout to Gray** and view with the camera. Trying out change the Ansio Level as well. Anisotropic filtering makes the texture look better when viewed at a shallow angle.





You should see something like this.

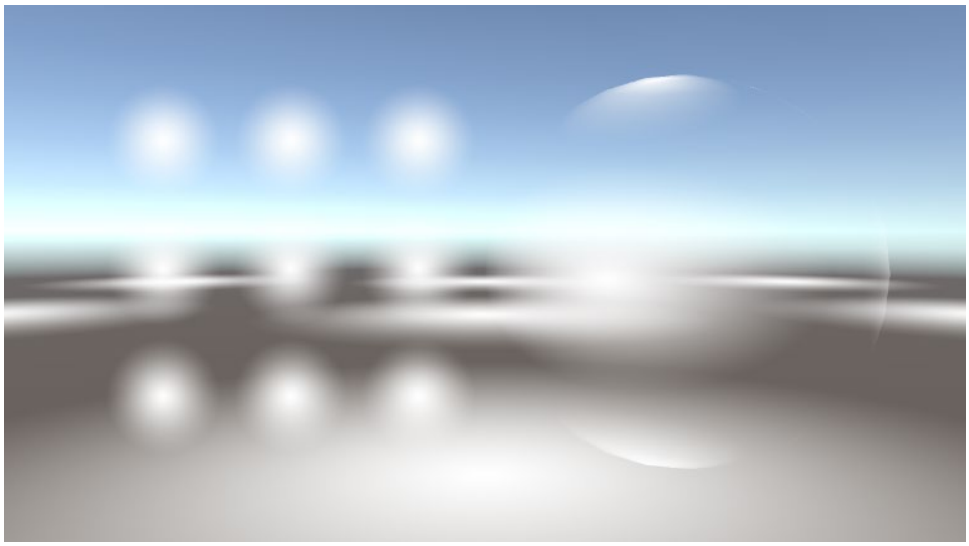
14. Now, let try adding in **alpha** to work for both color using tint and texture. I have provided 2 textures. The first texture is a smiley face texture with transparency at the sides. So it is an direct 1 or 0 alpha value type. The second texture is a circle texture that fades off so there is a gradual change in the alpha value.



15. Add the following to the sub shaders

```
SubShader{
    Tags {"Queue" = "Transparent" "RenderType" = "Transparent"}
    Blend SrcAlpha OneMinusSrcAlpha
```

This will set the sub shader that renders geometry as part of the Transparent render queue and we set the **RenderType** as Transparent which can used for Transparent, Particle, Font objects. Although it is not necessary, it can still work without it. We are also setting the blending operation to enable blending for the default render target to use the source alpha value and 1 – source alpha for the destination factor which work for regular alpha blending.



https://docs.unity3d.com/Manual/SL-Blend.html

Try mixing the other blend types and see what effects could be created

16. Try adding this to the properties

```
_alphaCutoff("Alpha Cutoff", Range(0, 1)) = 0.5
```

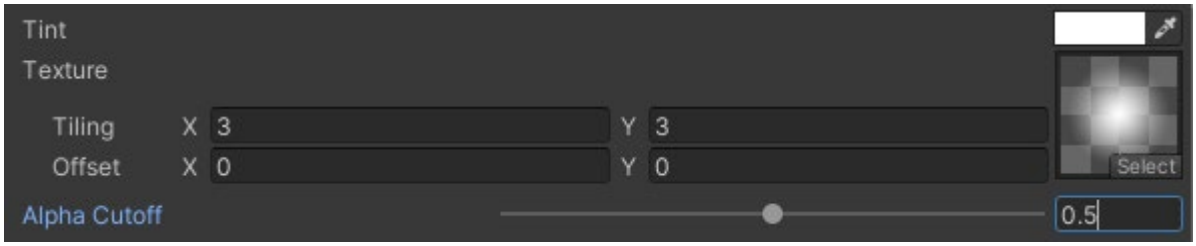
And adding this uniform to the shader

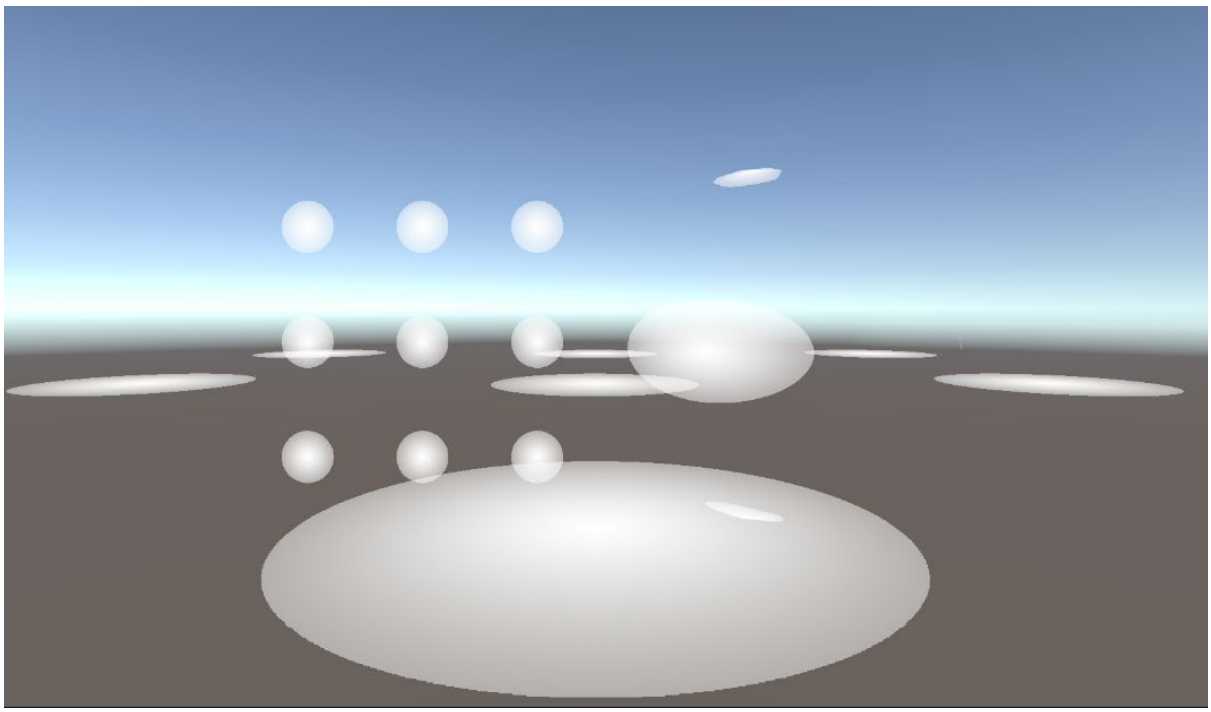
```
uniform float _alphaCutoff;
```

17. Modifiy the fragment shader to make use of the _alphaCutoff uniform.

```
float4 result = _tint * tex2D(_mainTexture, v2f.uv);
clip(result.a - _alphaCutoff);
return result;
```

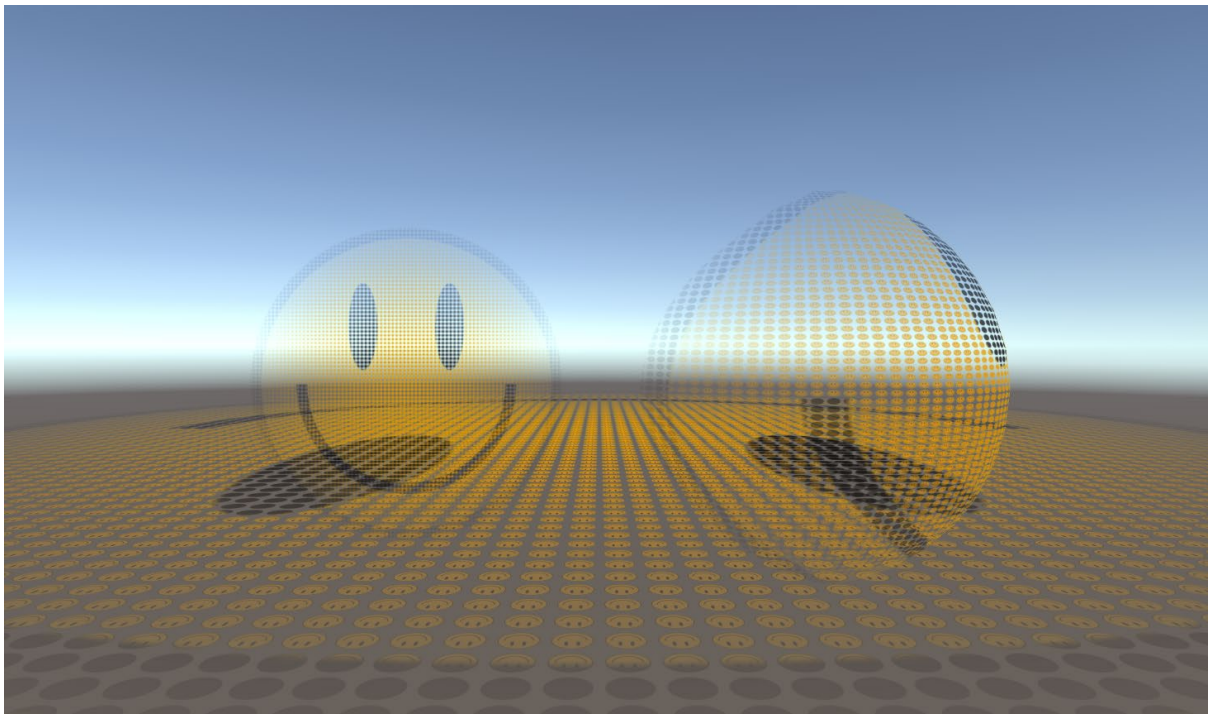
The clip function discards the pixel if the value inside is less than zero.





Now, you have learned how to use texture and add blending to it.

Try creating this effect yourself.



Hint: I did the tiling inside the code rather than using unity texture tiling

Hint: I used 4 difference texture to generate this.

These are 2 different effects

(Answer will not be given so that you can use for your assignment)