**By Craig McCorrisken**

**Advanced Computer Graphics:**

**Individual Coursework Report**

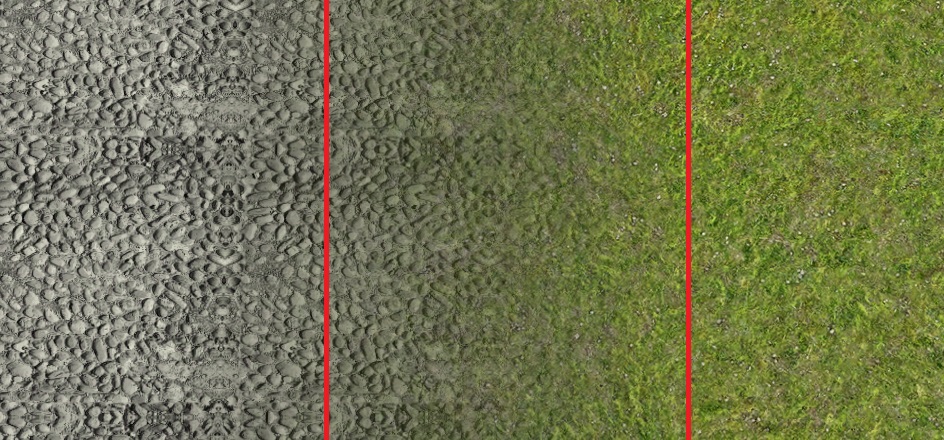
# Brief

The brief for the individual project was to study and develop my knowledge of a 3D graphics effect of my choice and implement the chosen effect in a working project demo. The 3D graphics effect could be anything, however a list of possible topics was given to help inspire students. These included: Supporting (moving or interactive) multiple light sources and types, Texture Blending, Toon shading (with image processing/procedural geometry silhouetting), Implement a custom shader or GLSL image processor, etc.

# Proposed Solution

Out of the list of possible topics available I chose to study and implement texture blending. I chose to study texture blending as it was briefly mentioned during an AGP lab and held my interest enough for me to want to study more about it. For the texture blending demo, I will draw a box in the centre of the screen which is drawn using a shader which has texture blending implemented into it.

To implement an aspect of interactivity into this demo I will allow the user to turn the texture blending on and off with the press of a button. This will allow the user to see the difference between the effects and show them independently. The user will switch between a phong shader and a texture blending shader when the mentioned button is pressed. The user will also be able to use the WASD keys to move the camera around the box and to see the effect from different angles and prove the graphics effect works in a 3D environment.

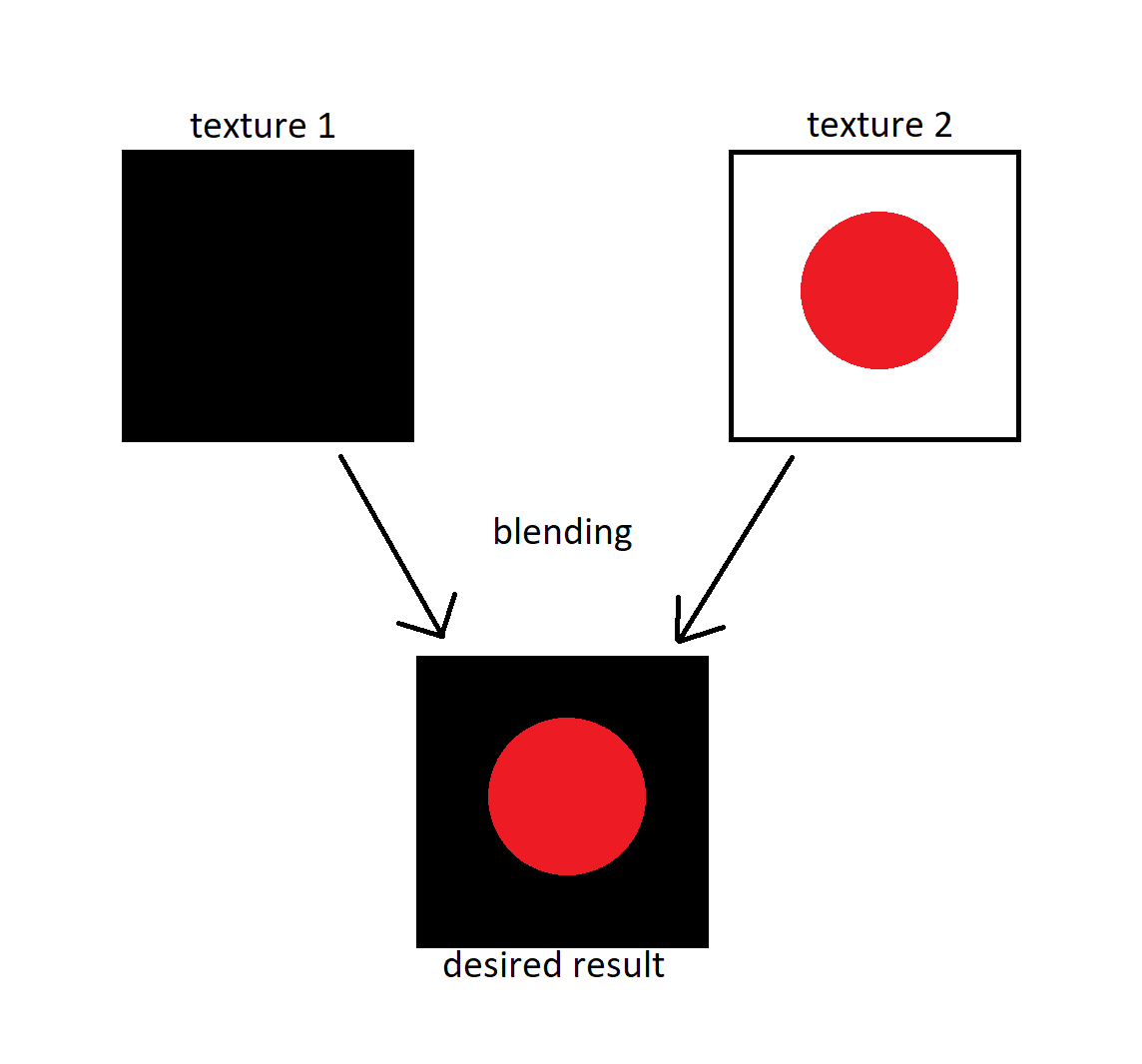


The finished product should look similar to the centre of the above picture. This is a good example of two separate images & how they look after blending them together.

# Concept

Texture blending is a graphical concept in which two separate images are blended together to create a new texture. At its very core, texture blending is taking the colour of each pixel at corresponding locations on two separate textures and blending them together to give the colour of that pixel on the new texture. For example: if you were to take a red pixel from one texture and blend it with a blue pixel from a second texture you would create a new texture which has a purple pixel. The type of purple you would get is dependent on how transparent you made each pixel from each texture, i.e.: the more transparent you make the red pixel, the more blue is visible which will make the new pixel appear a darker shade of purple and vice versa.

Texture blending has many good uses, one of which could be a blending a cobble road surface with a grassy/ moss texture to give the effect of wear & tear/ aged over time. To add a sense of realism, you could affect the alpha value of the moss texture, making it appear darker/thicker in certain areas, i.e. areas it has been walked on or paths have been created.



* A very basic diagram of what you except from texture blending

# Before & after

* Show box using phong & using texture blending (images, brief description)

# Walk through

* Walk through the steps/ method of creating this project
* //edit the below into paragraphs

1. Created a new shader which will handle the texture blending
2. Edited the method for drawing bitmaps to draw pngs (explain why .pngs & not bitmaps necessary)
3. Loaded 2 new textures for blending
4. Walk through binding the textures to the shader ID
5. Drew a box using this new shader & the new textures

# Video

* Insert video link here

# References

<https://answers.unity.com/questions/476294/how-do-you-turn-off-terrain-blending.html>

<https://learnopengl.com/Advanced-OpenGL/Blending>

<https://www.youtube.com/watch?v=szasYykLUA0>

<https://trycolors.com/>

# Implementation:

Firstly, I created 2 new shaders called “textureBlender.frag” & “textureBlender.vert” inside these files I created code based on the multi-texture section of the OpenGL Programming Guide book by Dave Shreiner which takes two separate textures and takes the position & colour of each texel of the image & blends them together to create a new image.

To allow the two images to blend together I had to implement the SDL\_image library which is an extension to SDL itself allows the loading of non-bitmap texture files. This was implemented to allow the loading of texture files which support transparency, namely .PNG files.

I had to modify the loadBitMap function given to us in the labs to allow the loading of all texture files instead of just being used to load Bitmap files.

I initialised 2x .PNG files and blended them together using my newly created texture blending shader and applied this to a box.

# How to Use:

To switch between the texture blending shader & the phong shader for the box, press 3 on your keyboard for texture blending & 4 for the phong shader.

WASD allows the user to move around the level freely.

# Video:

* Insert link