**Advanced Computer Graphics:**

**Individual Coursework Report**

**By Craig McCorrisken**

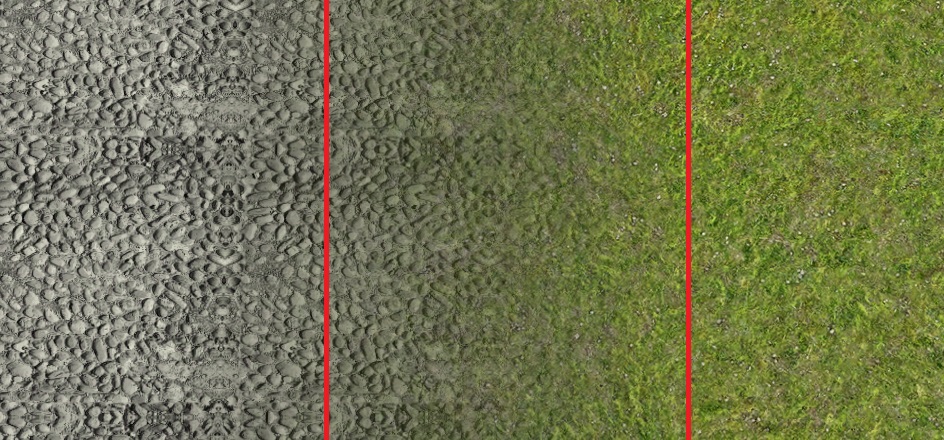
# Brief

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. Pressing the number keys “3, 4, 5 & 6” will switch the texture which is being used as the effect texture and change the look of the box, this will be used to show how the effect works using different effect textures, while keeping the same base texture.

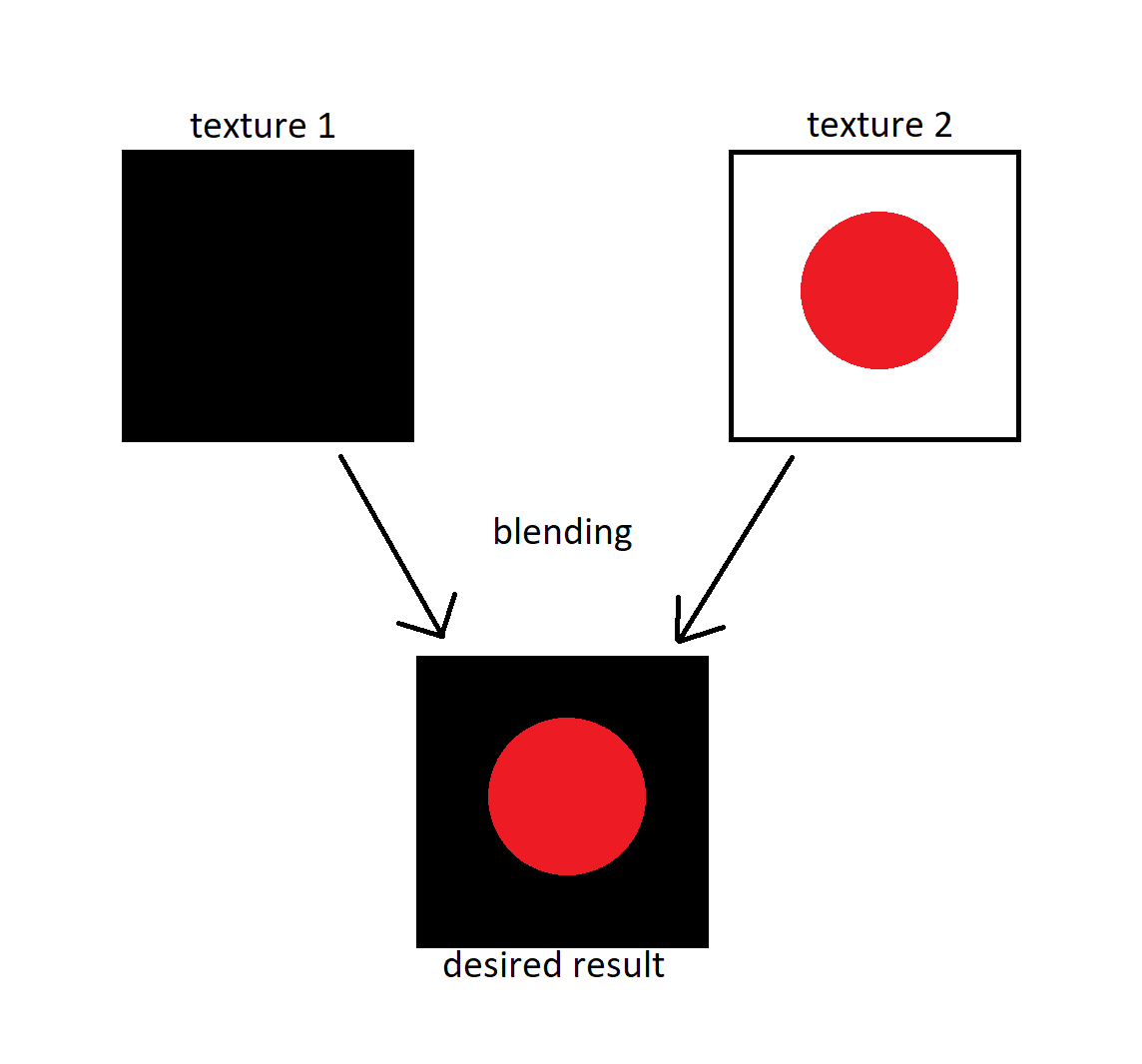


* The finished product should look similar to the centre of the above picture. This is a good example of two separate images & how they look after blending them together.

# Concept

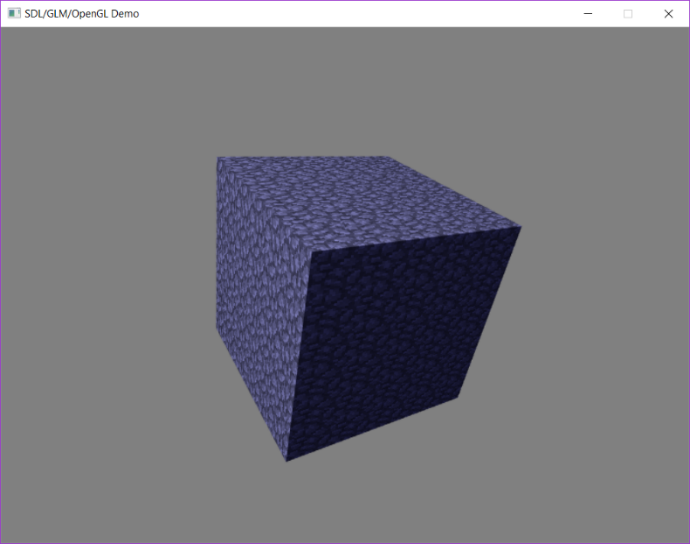
Texture blending is a graphical concept in which two separate images are blended together to create a new texture. At its very core, texture blending is taking the colour of each pixel at corresponding locations on two separate textures and blending them together to give the colour of that pixel on the 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.

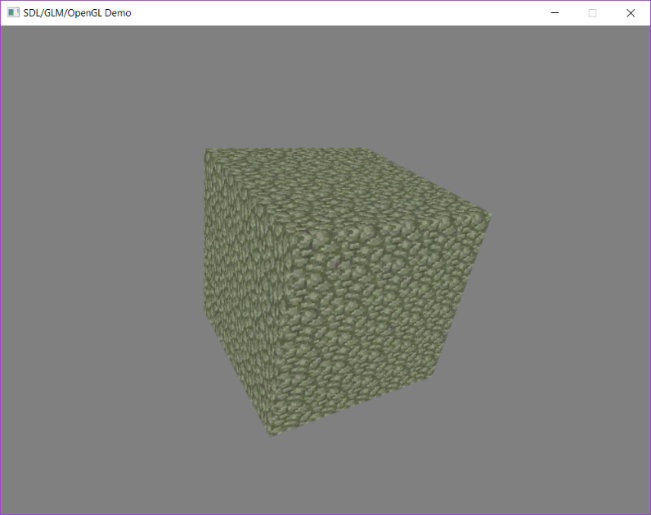


* A very basic diagram of what you except from texture blending

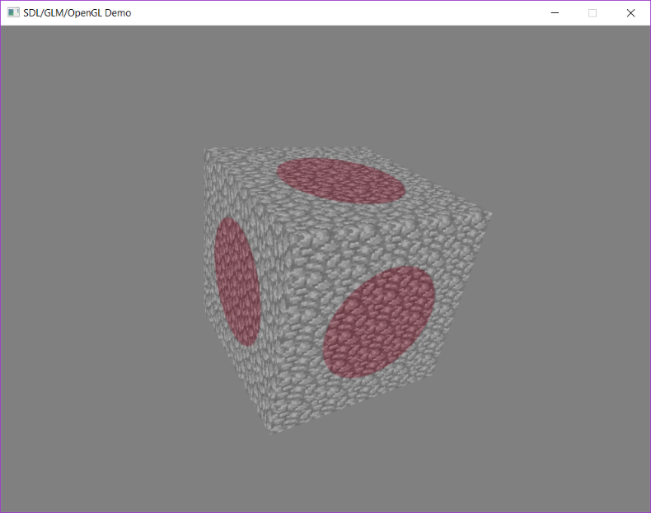
# Before & after



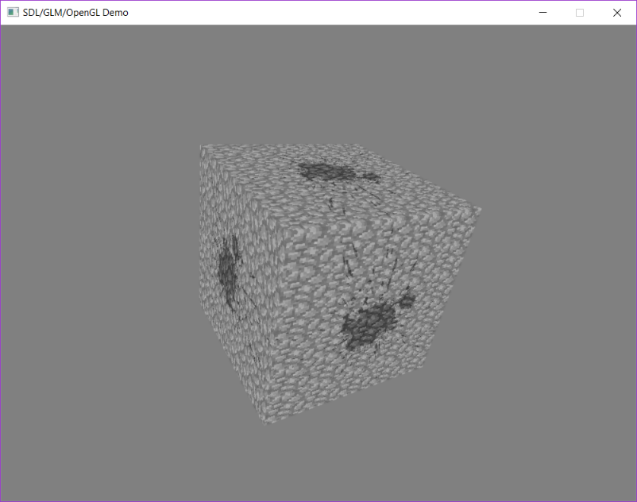
* A 3D cube drawn using a basic Phong lighting shader.



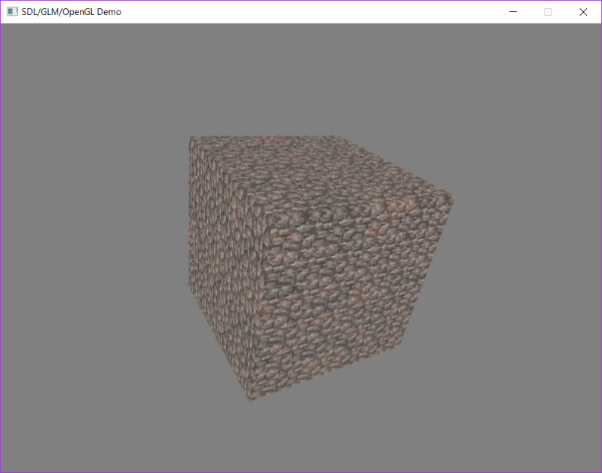
* The same 3D cube as above, however drawn without using the phong lighting model. The shader which is used to draw the cube this time does not consider the specular, ambient or diffuse lighting which the phong model uses to give an object depth & realism. Unlike when drawn using the last shader, the cube this time has a green colour to it, this comes from texture blending. The original texture used to texture the box now has an effect texture drawn on top of it to give the appearance of moss.



* This time the base texture is blended with a red circle which has a transparent box around it. This was done to show how fully transparent sections of an effect image blended with the base image will not change the colour of that texel, only texels with an alpha value greater than 0 will affect the image.



* Another example of a transparent background with a basic image in the centre. This time I have used this texture as it contains more detail & the splatter effect shows how this can be used in game, for example: a player being punched and the blood hitting a wall.



* Finally, I have included this image as a final demonstration, this time of two images where no section is fully transparent. I have included this in my project to show how a texture can be completely changed by using texture blending, not only can texture blending be used to add a small effect to an image, but it can also be used to completely change how an object looks.

# Walk through

Firstly, I created a new shader which would be used to perform texture blending. My first attempt at texture blending was done in the original phong shader, however I had some issues getting this to work, therefor I moved the texture blending to a separate shader which would only, draw the scene and match the view from the c++ program.

Next, I researched the “Using Textures” section of Dave Shreiner’s book OpenGL Programming Guide, specifically the “Using Multiple Textures” subsection which provided me with a lot of help in programming this 3D effect. I implemented this code into my new shader and with some help and some code from the phong shader, I had a working texture blending shader.

Next, I had to modify the way images were loaded. This had to be done as the only texture file formats I could load at the beginning of this project were Bitmaps, however during testing I could not get these bitmaps to show any form of transparency and google searches supplied me with differing opinions on whether bitmaps could support transparency or not. This left me with no option, but to modify the method which loads in images to support the loading of PNG’s which I was certain supported transparency.

After this, I loaded multiple textures which had the .png file extension into the game and draw a box on the screen using these textures. I then created a way to switch between these textures while in game and passed these images into the shader to blend 2 images together into a new image. This was the final step in getting texture blending to work.

# How to Use

Texture blending is automatic. when you load up the program, the first thing you see is a cube which is textured using two separate images blended together. However, the below list while explain how to show the shader off a little more as well as move and change to a different shader

|  |  |  |
| --- | --- | --- |
| Number | Key | Effect |
| 1 | W | Move forward |
| 2 | A | Move left |
| 3 | S | Move backwards |
| 4 | D | Move right |
| 5 | 1 | Draw box using Phong Shader |
| 6 | 2 | Draw box using Texture Blending Shader |
| 7 | 3 | Add the moss effect to the cobblestone box |
| 8 | 4 | Add the red circle effect to the cobblestone box |
| 9 | 5 | Add the brick effect to the cobblestone box |
| 10 | 6 | Add the splatter texture effect to the cobblestone box |
| 11 | , | Turn left |
| 12 | . | Turn right |
| 13 | R | Move up |
| 14 | F | Move down |

# 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/>

https://www.sitepoint.com/gif-png-jpg-which-one-to-use/