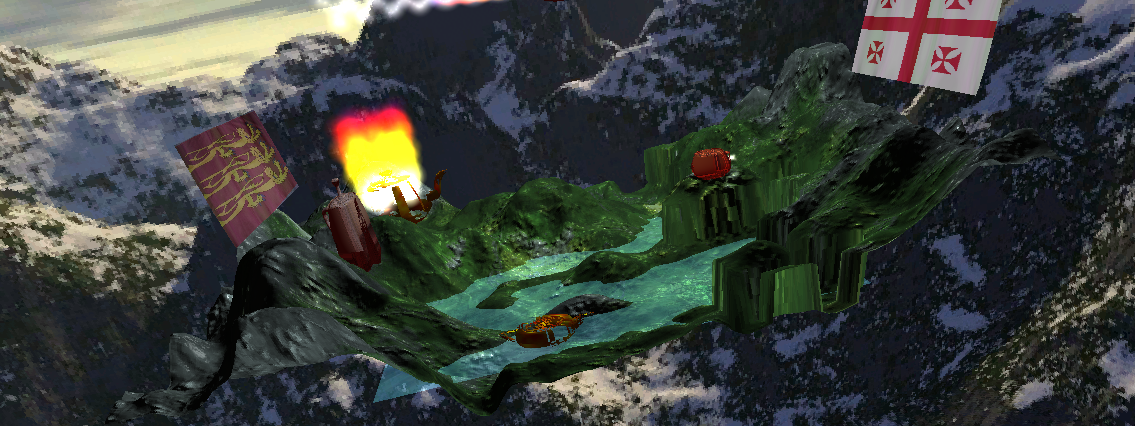
**Teapot Battlefield Effects**

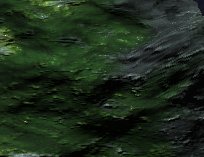
**Sky**

The sky effect is achieved by using the cube mapping technique. Before applying the texture to the model’s surface it is multiplied by the ModelViewProjectionMatrix. By utilising six 2D texture images to portray up, down, left, right, forward, and backwards, the result is seemingly infinite environment.



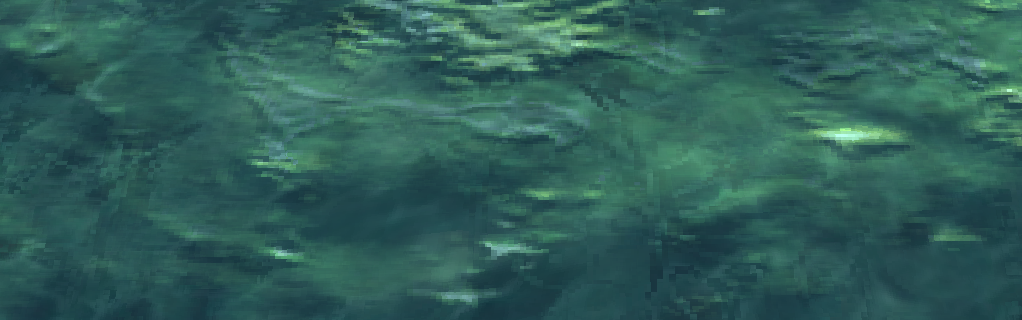
**Terrain**

The terrain is bumped map by using parallax mapping. This effect is reached by (in the vertex shader) taking the view position and light source and transforming them into tangent space. In the fragment shader the height information is read from a height map, and is used to offset the incoming texture coordinate. The corrected texture coordinates are then used to get the colour from the colour map and normal from the normal map.  
I also made various parts of the terrain more interesting and fit in with the rest of my scene by altering the y coordinates of the terrain based on the vertex position.



**Animated Lake Water**

The animated lake water effect is done by displacing the texture coordinates in a semi-random fashion according to a noise map in the fragment shader. The water is also alpha blended with terrain so that you can partially see the terrain below the water.



**Deformed and Animated Teapots**

I have created several deformed and animated teapots, they are:

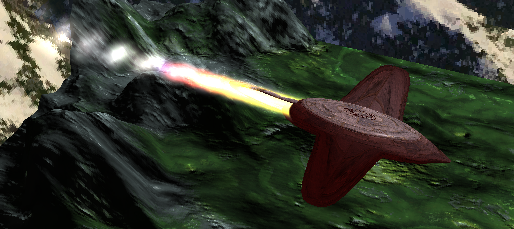
Burning Eroded Teapot

The eroded effect is achieve by setting the pixel to 0 if the x coordinate of the texture (looked up against a noise texture) is less than 0.5, else it is set to 1.  
The burning effect is achieved by implementing a particle system and positioning it relative to the position of the teapot.



Flying Jet Teapot

The flying jet teapot is deformed by using a series of if statements and the smooth step function. It is animated by using the SinTime semantic to move the coordinates of the jet. The jet has an animated tongue of fire and smoke jetted from the tail. This is achieved by using a particle system and positioning/rotating it relative to the jet’s tail. The flying jet teapot also drops a bomb roughly every 120 seconds that causes an explosion upon hitting the terrain.



Robot Teapot

The robot teapot, like the jet teapot, is deformed using a series of if statements and the smooth step function.



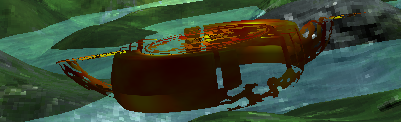
Tank Teapot

The tank teapot is also deformed using a series of if statements and the smooth step function. It is animated using the SinTime and CosTime variable semantics to modify its position, so it moves backwards and forwards in an animation loop. It uses a particle system to represent smoke coming from the exhaust of the tank.



Boat Teapot

I also created an extra deformed teapot. I deformed the top half of the teapot so it was flat, and left the natural curvature of the teapot as it was because it already looked like a boat. I also made the boat-teapot eroded so it looked like a shipwrecked boat; I achieved this by doing what I did in the pixel shader of the burning eroded teapot.  
The teapot is animated, it moves slightly backwards and forwards using the CosTime variable semantic, this helps give the look of a truly abandoned shipwreck.



**Fire, Smoke, and Explosions**

As depicted above the tank has smoke coming from its exhaust, and the jet has a fire and smoke jetted from its tail. These effects are achieved using particle systems. The particle systems are made up of a 100 quads that are billboarded; the particle system is looped, fades the particles out over time and uses the variables, particleHeight, particleSpeed, particleSpread, particleSystemShape, particleShape, and particleSize to define the desired effects.  
The explosion effects incorporated are a little different than the fire and smoke particle systems, by the fact that the particles are emitted in a spherical fashion instead of semi-random.



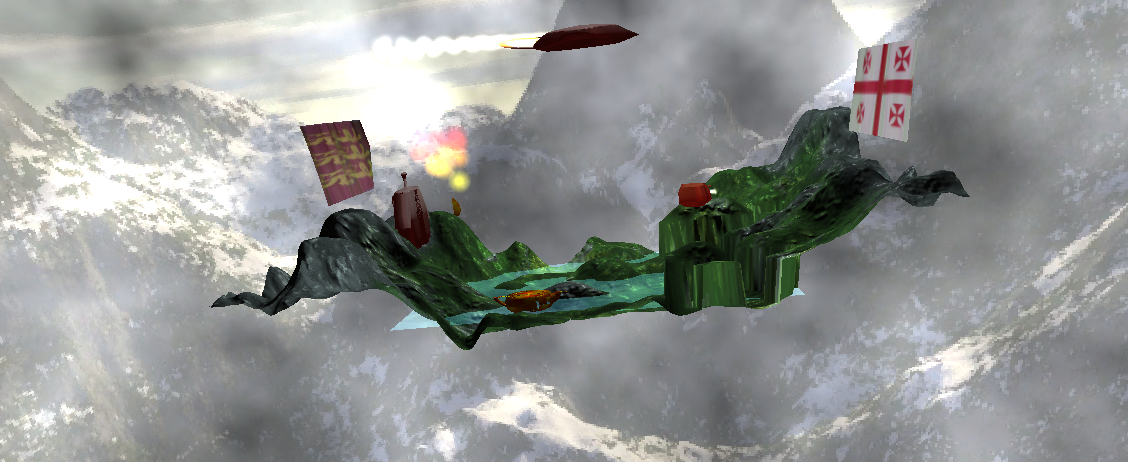
**Animated Flags with Anisotropic Lighting**

There are two animated flag effects in the project, and they both use anisotropic lighting. The flag effect is created by perturbing the vertices according to a sinusoidal wave, which increases in its strength the further away from the beginning the vertex is.  
The anisotropic lighting effect on the flag is achieved by taking the position of the viewer, and the half angle and the source of light. All of these are dotted with the surface normal; this gives a texture lookup to a gradient which is mirrored. The flag’s texture and the anisotropic adjustment are linearly interloped using lerp intrinsic.



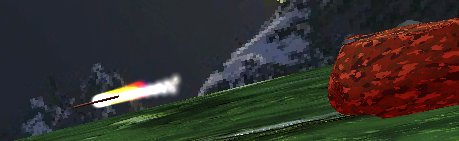
**Fog**

I have added a fog effect to my scene and which is animated and it also surrounds the environment. Most of the work is done in the Pixel Shader; a 3D noise texture is used to displace the coordinates of the texture coordinates based on the cloudSpeed and noiseSpeed variables. It is blended with the skybox so it is still possible to see the environment through the fog slightly.



**Animated tongues of fire jetted from the tail of the flying teapot-jet and missiles**

The animated tongues of fire and smoke jetted from the tail of the flying teapot-jet have already been shown above. There is also a missile in the project, it is fired roughly once every 120 seconds (controlled via the time0\_X variable semantic) from the turret/gun of the tank teapot:



**Flying Missiles and Explosions with Synchronised Behaviour**

The flying missile (from the tank teapot), and the bomb (from the jet teapot) are fired roughly once every 120 seconds in conjunction with the time0\_X variable semantic. In the fragment shaders of the missiles/bombs and corresponding particle systems if the current time is not within the specified time range then the pixel is simply killed via the discard intrinsic. Using this method of control it is very easy to synchronise behaviours between the explosions and the missiles/bombs, this is done by the hit-time of the missile and bomb and synchronising it with the particle system explosion effect.

Just After Impact



Before Impact

