Thermite3D Manual

Resource System

The Ogre graphics engine (upon which Thermite is built) provides an advanced resource management system. Thermite builds upon this system and so many of the resource types recognised by Ogre are automatically supported by Thermite. In addition, Thermite extends some of these resource types and adds new ones.

Where are resources/how are they found?

Figure XXX illustrates some of the available resource types. In this figure, resource types which are part of Ogre have a red background, while those which have been added for Thermite have a green background. An arrow from one resource type to another indicates that the latter is in some way dependent upon the former. It is something of a simplification but is useful to aid understanding:



An explanation of each of the resource types is as follows:

GPU Program Source: These are files containing OpenGL GLSL, Direct3D HLSL, or NVidia Cg program source code. They typically intended to form part of a vertex, fragment (pixel), or geometry program. The same program code may be shared between multiple programs, for example, a Perlin noise function might be used to perform displacement in the vertex shader or procedural texturing in the fragment shader. These files will typically be written by a programmer or technical artist.

Note: The terms 'GPU program' and 'shader' are often used interchangably in graphics literature. A shader is a program which runs on the GPU, but it can also refer to the hardware unit which executes the program.

GPU Program: These connect shader source code with other information (such as the entry point, supported GPU profiles, default paramaters, etc) into a complete GPU program. Again, they are typically written by a programmer or technical artist.

Texture Data: This is typically images which are applied to the surface of objects, but it can also be used as a way of passing arbitrary data to GPU programs (e.g. normal maps, lookup tables, etc). Texture data is most commonly created by artists.

Material Script: These pull together GPU programs with texture data to create the final look of a surface. Ogre material scripts have a lot of advanced properties including supporting inheritance and level of detail. More information about Ogre's material scripts can be found in the Ogre manual:

http://www.ogre3d.org/docs/manual/manual\_14.html

This also gives further details about the shaders and GPU programs mentioned previously.