Details for Computer Graphic Scene.

Libraries/external dependencies required:

CMAKE

* Assimp (Asset importer) (http://assimp.sourceforge.net/main\_downloads.html)
* FreeImage http://freeimage.sourceforge.net/download.html
* GLEW (http://glew.sourceforge.net/)   
  Download ZIP and take GL Folder from ‘include’ and

http://glew.sourceforge.net/install.html

* GLFW http://www.glfw.org/
* GLM https://sourceforge.net/projects/ogl-math/
* Application manager
* Shader manager
* Model Manager
* Procedural generation manager?
* Model class
* Camera class
* Lighting class

Key Features I want to implement after the basic set up is done.

* Optimisation – Scene Graph/ Frustum Culling.
* Procedural generation.
* Animated models being loaded in.

**Application manager**

This class will contains the methods/ variables required to create the window. The 4 main methods required for this is:

1. **Initialize -** This method will create the window and set all the initial properties; such as hiding mouse or screen position. Also need to open the correct channels for the shaders. 0 = vertex positions, 1 = texture information etc.
2. **Load –** This method will load the content that is in the scene. For example the models, lighting information, cameras being used etc.
3. **Update –** This method should be updated very frequently and should contain any information for any matrix updates, lighting colour changes etc.
4. **Render –** This method should render everything in the scene.

**Shader manager**

This class will handle the creation, deletion, linking and compiling of Opengl Shaders.

AREADY PRIMITIVE VERSION DONE WRITE METHODS USED HERE.

**Model Manager**

This class manages the importing of models.

PRIMITIVE CLASS ALMOST DONE.

**Model Class**

This class contains information regarding the model.

The main parts are:

Model – imported from model manager.

Material class

This class contains vector information for the model, such as:

* Ambient, diffuse and specular colour
* Shininess value
* Set and get accessors.

Matrix class – Or should data in this class be part of the model class instead?

This class will contain the matrix information for the model.

* Transform position (mat4).
* Normal position (mat3).
* Scale function.
* Rotate function.
* Translate function.

**Camera class**

The first implementation of a camera for the scene will be a free camera. Other camera types can be added later once more on the scene has been developed.

* Position
* Look direction
* Up direction
* Field of view
* Near and far distances.
* Aspect ratio.

**Lighting class**

This class will contain the variables for spot lights, point lights and directional lights. To start off though it will be have the directional lighting in it.

* Direction
* Light colour