INFO-3111, Michael’s Day 3 lesson plan

* Add the “z” coordinate to the model:
  + Loader needs to load it
  + Shader needs to have it
  + Vertex layout needs to be set up:
* We will also add the normal to the vertex, now.
  + We’ll convert the model, adding normal
  + We’ll load and pass the normal
* We’ll add the VAO code in here, too
  + See cVAOManager
* cMesh (the thing we are drawing)
* Depth (“Z”) buffer, too:
  + glClear() requires GL\_DEPTH\_BUFFER\_BIT added, too
  + glEnable(GL\_DEPTH\_TEST)
  + glCullFace( GL\_BACK );

Details:

* add a cMeshObject class:

#ifndef \_cMeshObject\_HG\_

#define \_cMeshObject\_HG\_

#include <string>

#include <glm/glm.hpp>

#include <glm/vec3.hpp> // glm::vec3

#include <glm/vec4.hpp> // glm::vec4

class cMeshObject

{

public:

cMeshObject();

~cMeshObject();

std::string meshName;

glm::vec3 position; // 0,0,0 (origin)

glm::vec4 wholeObjectColour; // 0,0,0,1 (black)

glm::vec3 orientation; // Euler angles (x, y, and z)

float scale; // 1.0f

bool isWireframe; // false

};

#endif // \_cMeshObject\_HG\_