

Tasks:

1. Start from a working version of Lab2.
2. The Ogre 1.10 notes might help a bit with some of the Ogre code we don't go through in class – or ask.
3. I attempted to simplify our architecture and have uploaded this as a lab3_start.zip (in perforce/ssuge_solutions) – I'd recommend you get this...
4. Pay attention to our design decisions in class
 - a. Multiple vs. One of each component type
 - b. Location of enum(?) for component type
 - c. Creating Components via GameObject or externally.
5. **(20 points)** GameObject class
 - a. Ogre::SceneNode that's properly initialized
 - b. (Unique) name
 - c. Map of Components
 - d. createMeshComponent (or addComponent)
 - e. setPosition, setOrientation and setScale wrapper methods.
 - f. Clean up everything in the destructor!
6. **(10 points)** Component class
 - a. Abstract getType method:
 - b. getOwner method
 - c. Constructor that takes a GameObject
7. **(10 points)** MeshComponent class
 - a. derived from Component
 - b. Create an Ogre Entity and attach it to the SceneNode
 - c. Take a file-name and whatever else you need in the constructor
 - d. Clean up everything in the destructor!
8. **(20 points)** Test program
 - a. Create a GameObject and add a MeshComponent to it (I'd suggest "Sinbad.mesh")
 - b. Make sure you destroy this GO on shutdown.
9. For this lab, let's try submitting it on blackboard. But...
 - a. Only submit .h and .cpp files you changed / created during this lab (likely game_object.h/.cpp, component.h, mesh_component.h/.cpp, application.h/.cpp)
 - b. Submit it compressed (.zip, .7z, .rar)
 - c. **-5 points** for submitting too much / not enough / not compressing.