E6: Rotate by a Quaternion

Course: IGME 309 – Real Time Simulations for Games II

Golisano College of Computing and Information Sciences

School of Interactive Games and Media

Rochester Institute of Technology

Due: Check in MyCourses

Deliverable: Mesh.cpp file (single file, unzipped)

**Objective:**

**The objective of this exercise is for students to understand the concept of rotating a geometric shape by applying the same rotation to each vertex individually using quaternions, rather than rotating the entire shape through matrix transformations. This exercise is designed to highlight the distinction between local and global coordinates and demonstrate how vertex-level transformations can be applied to achieve the same rotational effect as a global transformation.**

**By completing this exercise, students will:**

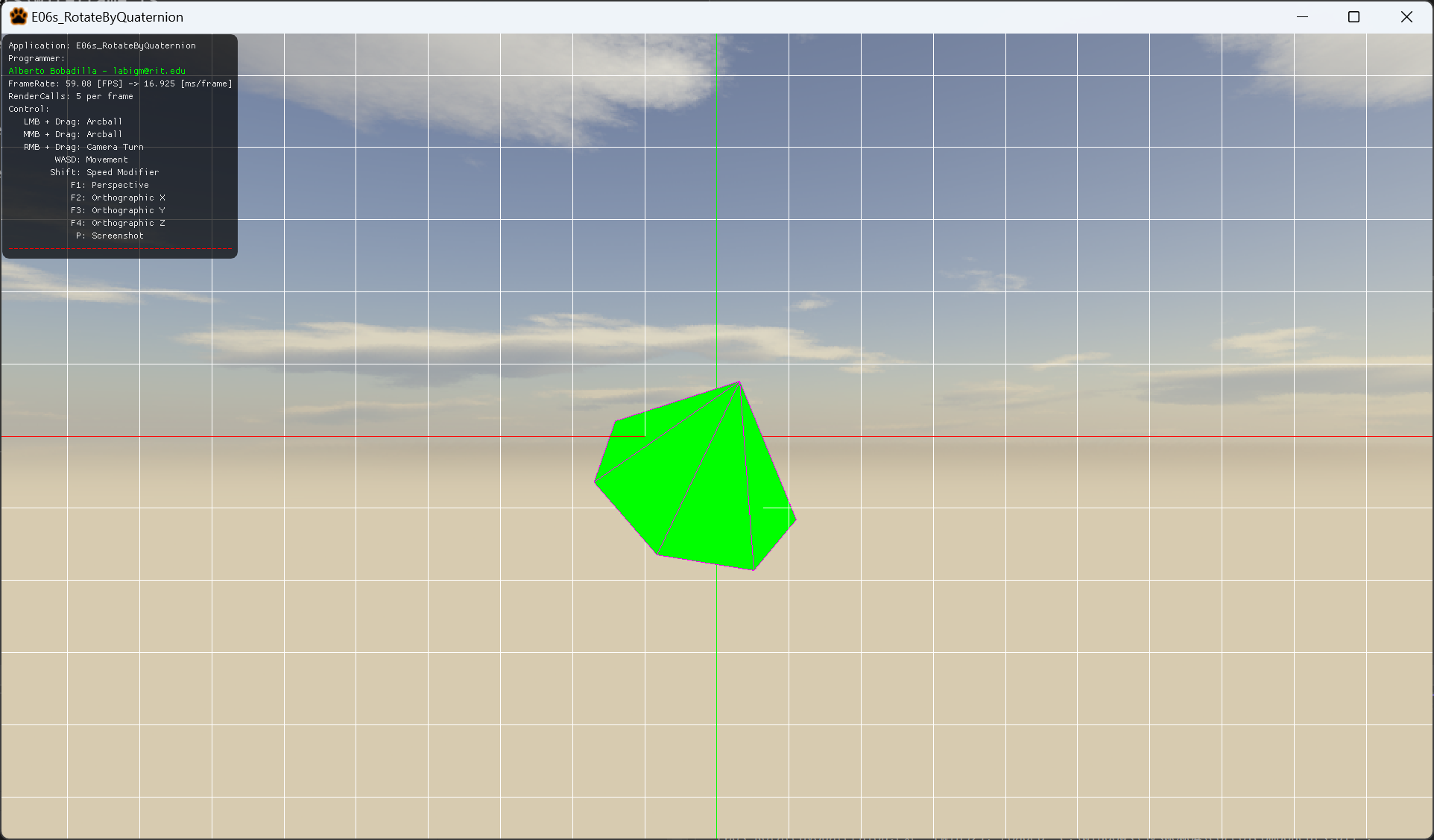
* **Learn how to use quaternions to rotate individual vertices of a 3D shape, ensuring that each vertex undergoes the same rotational transformation.**
* **Understand the differences between local and global coordinate systems, and how applying transformations locally (to vertices) versus globally (to the entire object) can affect the result.**
* **Gain hands-on experience with quaternion math and its application in 3D graphics and geometry transformations.**
* **Explore the advantages of using quaternions for rotation, including avoiding gimbal lock and achieving smooth interpolations.**
* **Gain a deeper understanding of the relationship between object transformations and the local vs. global frame of reference in computer graphics and simulation.**

**This exercise will help students appreciate the flexibility and efficiency of transforming individual vertices within their local coordinate system and how it can be used for more complex, efficient object manipulation in 3D environments.**

**Instructions:**

This exercise follows lecture D6

1. In the root of the repository look for the example execution under \_Binary. It will look like this:



1. For this exercise you will recreate what happened on the project C05B\_RotatePointClouds\_Pt2, you will take a list of points and rotate it. In this case you will rotate it 45 degrees over all 3 axis at once. Your code will be executed in the file Mesh.cpp in the function GenerateCone. As this is the only file you will modify this is the only fine you need to submit in your delivery.
   1. Remember to use glm::anglAxis is the easiest way to create a quaternion given the information you have.
   2. glm::rotate will also return a vector3 instead of a matrix4 if used appropriately, this means, using quaternion and the vector you have to rotate by that quaternion. It may imply a bit of research on your end on how to use this function.
2. All your code will be coded in the mesh.cpp file in said function so this is the only file you need to submit to the dropbox in MyCourses, please do not zip this file.

