Assignment 2 & 3: 3D Primitives (part 1 and 2)

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

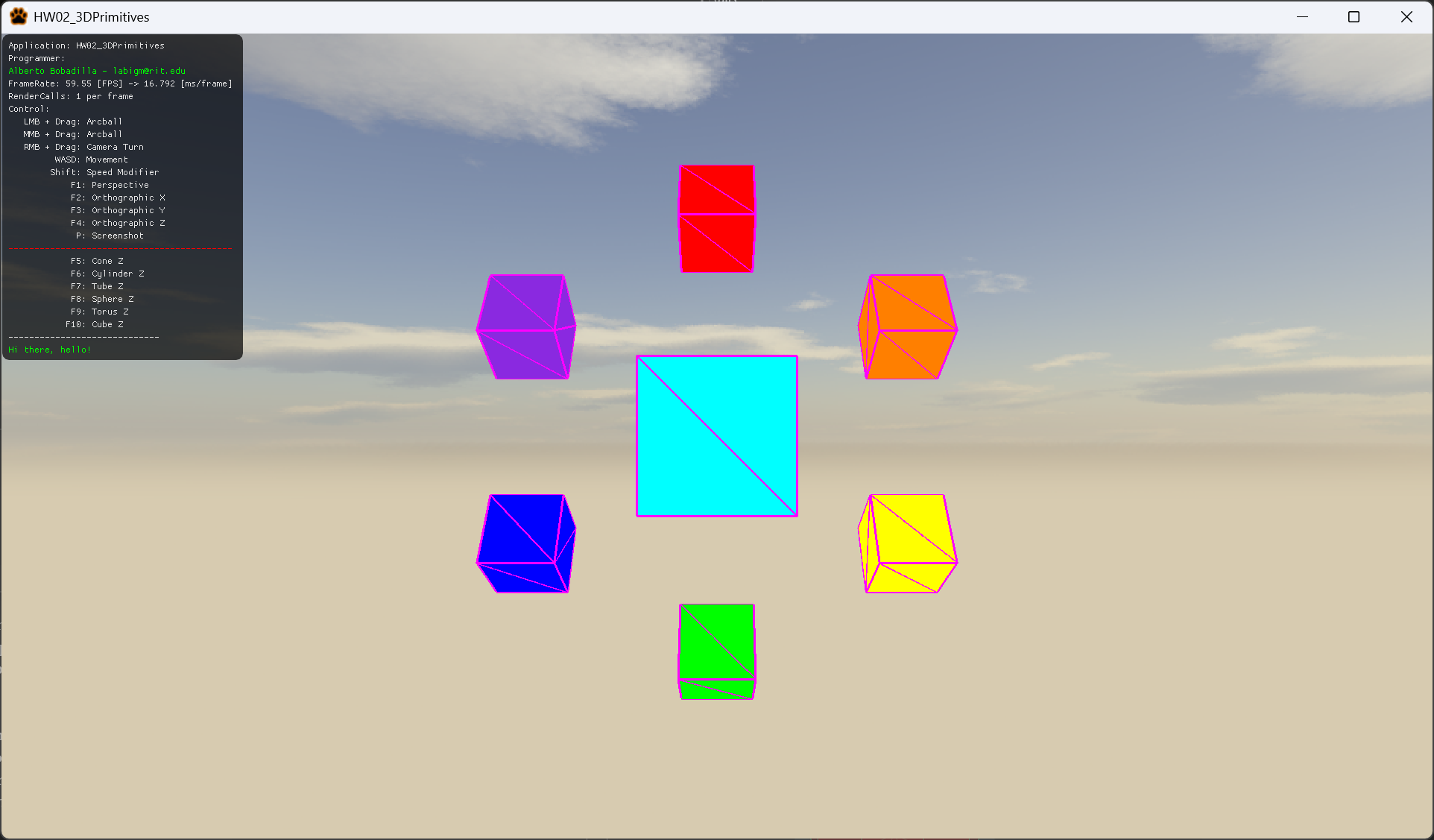
Deliverable: Unzipped MyMesh.cpp only

## Getting Started:

Video description: <https://www.youtube.com/watch?v=SvLDIILQBNs>

In the class repository, I’ve included a solution under the \_Binary folder.

The starter code will give you this out of the box:



## Objective

Implement five procedural 3D shape generators using modern OpenGL practices. These will simulate primitive mesh functionality similar to older glut calls and will support debugging and prototyping by generating test geometry at runtime.

## Background

Legacy OpenGL (pre-3.0) provided easy functions for creating 3D primitives. While deprecated, the utility of rapidly generating placeholder shapes remains relevant. You will recreate this functionality by building your own procedural mesh generators for five distinct primitives.  
  
Each method should dynamically build vertex buffer data, be fully procedural (support variable size and subdivisions), and center the shape at the origin. You are responsible only for the geometry generation; the starter code handles compilation and memory cleanup unless you use custom memory logic.

## Part 1: Assignment A02 – Basic Shapes

* Cone(float radius, float height, int subdivisions, vector3 color);
* radius of the base of the cone
* height of the cone.
* subdivisions (refer to the cylinder’s subdivisions)
* color of the shape



* Cylinder(float radius, float height, int subdivisions, vector3 color);
* radius of the base of the cylinder.
* height of the cylinder.
* subdivisions is how many sides the base have, if this is 3 the cylinder has a triangular base, if it has 360 the base is a circle, etc.
* color of the shape



* Tube(float outerRadius, float innerRadius, float height, int subdivisions, vector3 color);
* outherRadius is the radius of the outermost part of the tube
* innerRadius is the radius of the innermost part of the tube (the whole if you will)
* height of the tube
* subdivisions (refer to the sylinder’s subdivisions)
* color of the shape

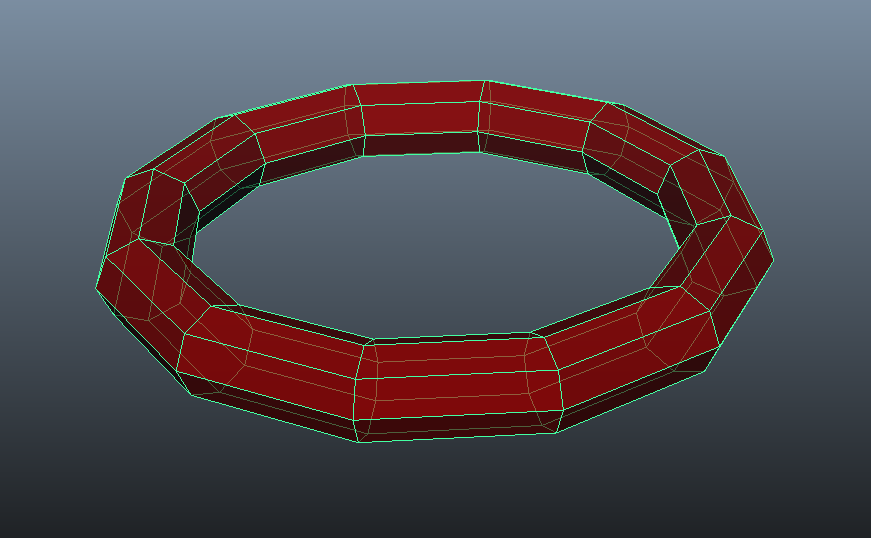


## Part 2: Assignment A03 – Advanced Shapes

* Sphere(float radius, int subdivisions, vector3 color);
* radius, how big your sphere is.
* subdivisions
* color of the shape



* Torus(float innerRadius, float outerRadius, int subdivisions, vector3 color);
* innerRadius is the size of the radius going from the center to the innermost part of the donut.
* outerRadius is the size of the radius going from the center to the outermost part of the donut.
* Subdivisions is the number of subdivisions that goes around the torus.
* Color of the shape.



## Requirements

* All shapes must be centered at the origin.
* All shapes must support parameterized size and subdivisions.
* All shapes must be procedural. Hard-coded geometry will not receive credit.
* Provided code handles memory management and rendering.
* Sphere subdivision methods may vary; any recognizable sphere is acceptable.
* Include in-code comments where relevant, especially for complex math.
* Partial credit will be awarded for incomplete or incorrect implementations with explanation.

## Tips

* Tube is like a cylinder but requires two radii.
* Sphere implementation can vary; focus on logic, not visual perfection.
* Use subdivision logic consistently (e.g., for circular bases).
* Use the Cube(float size, vector3 color) example as reference.

## Controls in Starter Code

* WASD: Move camera
* Right-click (hold): Adjust view angle
* Middle-click (drag): Rotate object in world space
* F4–F10: Switch active primitive

## Submission Instructions

* Submit your Mesh.cpp file containing your logic (no full solution).
* Upload to:  
   - Dropbox: A2 – 3D Primitives pt1  
   - Dropbox: A3 – 3D Primitives pt2
* Push your work to your private fork on Git. (Recommended for backup/disputes.)

## Grading Rubric pt1

|  |  |  |
| --- | --- | --- |
| Shape | Points | Notes |
| Cone | 33.33% | Must support subdivisions and be centered at origin |
| Cylinder | 33.33% | Subdivisions must affect base shape |
| Tube Grading Rubric pt2 | 33.33% | Requires outer/inner radius logic |
| Sphere | 50% | Flexibility in subdivision method, should appear round |
| Torus | 50% | Should loop around to form a ring; consider both radii |

## Penalties

* 0%: Code does not compile or crashes immediately
* -10%: Memory leaks (if custom memory is used)
* -10%: Inadequate or missing comments
* -10%: Hardcoded or non-procedural geometry
* No credit: Submissions that include full VS solution or exceed file size