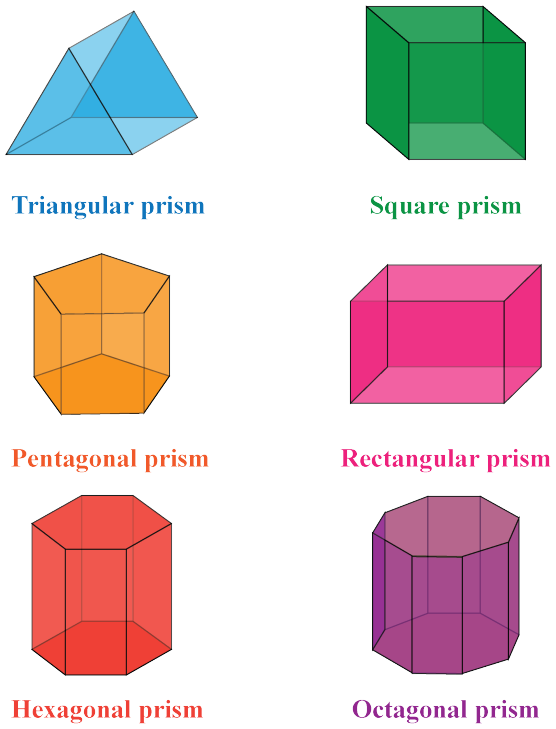
CS7.30 — Computer Graphics

Instructor: Avinash Sharma
Release Date: January 8

Due Date: January 17 23:55
Assignment: 0

Assignment #0 acts as your introduction to OpenGL, which is the most generalized and widely used API for 2D and 3D rendering. In this assignment you will be rendering 3D Solids onto your screen and even move them around!

Maybe we need more than an Obama Prism. Prisms are shapes with two parallel aligned polygons of the same number of sides and have rectangles joining these sides / edges. In this assignment you will be generating these polygons in 3d space and rendering them with the help of OpenGL.



Prisms with different polygon bases

Note: We are going to provide some boilerplate code. However we highly recommend you to make your own boilerplate code using the first few chapters of LearnOpenGL. They give a detailed explanation of how the boilerplate works which will provide a better understanding. We'll be sharing our tutorial code anyway which is derived from LearnOpenGL. You can use that as well if you want to.

Part A: Procedural Generation of the Prism

You will be generating and rendering a prism of given parameters. The number of sides of the polygon in the prism should be passed as a **command line argument**. As this argument increases, the prism will tend to become cylindrical in nature.

1. Code up the prism generation pipeline which should create the vertices and the faces of the prism for any general n sides.
   Hint: Generate one polygon's vertices on a plane and then translate these to get the other polygon's vertices and use these to build all the necessary faces.

2. You will now have the vertices and the faces. Next, color each of the faces of the prism with any color you like (each face should have a unique color - you can also assign a unique color to each vertex of a face).

Part B: Bringing the scene to life with motion

1. **Flying Camera** The fly camera is a popular camera locomotion technique used in blender and the like to navigate the scene. You will be implementing it by binding six keys - W and S for forward and backward, A and D for left and right and Q and E for up and down for the respective camera movement. *Ensure the camera faces the prism's (center) at all times*

2. **Object Translation** Another six keys of your choice should be assigned for 6D (up, down, left, right, towards, away w.r.t to the camera) movement of the prism. *The camera need not follow the prism.*

3. **Prism fashion show** Assign 2 numeric keys to move the camera to any pre-defined position of choice. *But, ensure the camera is still facing the prism*

4. **Prism, let's go for a spin** On press of the key R, the prism should rotate about any single axis (X, Y, Z) of your choice.

5. **Oh! How the turntables?** The turntable effect involves making the camera revolve around the object of interest. Implement this effect by making the camera rotate around the prism on pressing the key T. *The camera should always be facing the prism.*

Bells & Whistles

Additionally, if you are interested you can learn how to place textures on the surfaces. Render the shapes that you made with some good looking textures. Please ensure that the different faces of the final shape should be distinctly visible. We would love to see how creative you guys are.

Other Instructions

1. We would really love to see how well you code. Ensure that your code is modular and clean. **Do not create different animation functions for different prisms**

2. Debug your code and ensure it works well and is complete before your submission.

3. You can have multiple files and library codes. But make sure you submit only your code files, as moodle has a size restriction and we will not accept submission via any other means. *There could be exceptions, if you manage to write that much code :)*

4. Late submissions will not be accepted under any circumstances, so ensure you are on time. The deadline is **23:55 PM on January 17th, 2022**

5. Jokes aside, plagiarism is treated very strictly. You may discuss with others, but just don't copy code and make it hard for us.