CSE 333/533: Computer Graphics
Lab 3: Binding Variables

Instructor: Ojaswa Sharma , TAs: Vishwesh Vhavle , Aadit Kant Jha

Due date: 23:59, 4 September 2023

Introduction

Binding variables in GLSL (OpenGL Shading Language) is a crucial step in the shader programming pipeline, enabling communication between the application code and the shader programs running on the GPU.

Story: *Vikram, a brilliant but perpetually drowsy student, found himself in quite a bind when his Computer Graphics TA assigned him the task of implementing three cubes that smoothly transitioned through a cyclic color sequence. Unfortunately, Vikram had slept through all his classes, leaving him clueless about binding variables in shaders and implementing color shading.*

Desperate to salvage his grade, Vikram has sought help. With determination, he has reached out to you, his knowledgeable friend, who must now help him fix his code. Vikram is very close to success. It seems he has implemented passing variables using color attributes, which is giving him the wrong output in this case. You must help him implement passing variables using color uniforms.

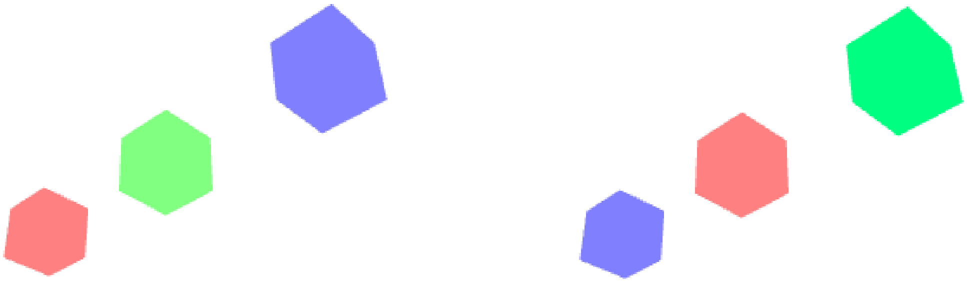
Attributes

Attributes in computer graphics describe per-vertex data, specifying properties like position or shading data for each vertex in 3D models. These values vary across vertices and help shape the geometry. Attributes are vital for vertex shaders, passed from the CPU to GPU, affecting vertex processing independently.

Uniforms

Uniforms are global shader variables, holding constant values for all vertices or fragments during shader execution. They ensure consistent data across vertices or fragments and are used for critical parameters such as transformation matrices and lighting properties, shared across the entire rendering process.

Deliverables



You have been given code to render 3 cubes and we have already implemented the timed coloring. Implement the cube color shading in `cube.cpp` to pass variables using uniforms instead, subsequently update the vertex shader at `shaders/vshader.vs`. Upload the zip file of **patched code** and **2 images of** the **output**.

Name the zip file as lab03_<name>_<roll number>.zip
Example: lab03_vishwesh_2020156.zip

References

https://www.opengl.org/documentation/
https://www.khronos.org/opengl/wiki/Rendering Pipeline Overview
https://www.khronos.org/registry/OpenGL-Refpages
https://www.glfw.org/documentation.html
https://www.khronos.org/opengl/wiki/Framebuffer

Note: Your code should be written by you and be easy to read. You are NOT permitted to use any code that is not written by you. (Any code provided by the instructor/TA can be used with proper credits within your program). Theory questions need to be answered by you and not copied from other sources. Please refer to IIIT-Delhi's Policy on Academic Integrity here.