REVIEW

Congratulations on reaching the end of the first chapter! I highly encourage you to start up a new project and try to reimplement everything you've learned thus far. This will benefit your learning *so much*. It will reinforce what you know well and bring light to areas you need to spend more time on.

Glossary

Learning the vocabulary can help your understanding and makes communication much more streamlined (and is essential for interviews).

Try to define everything on your own. After you've defined a word, go to the Vocabulary subpage, expand the word you defined, and compare your definition to the one there to see if you missed anything important.

OpenGL:

CONCEPTS

- **Viewport:**
- **Graphics Pipeline:**
- **Shader:**
- **Vertex:**
- **Normalized Device Coordinates (NDC):**
- **Vertex Buffer Object (VBO):**
- **Vertex Array Object (VAO):**
- **Element Buffer Object (EBO):**
- **Uniform:**
- **Texture:**
- **Texture Wrapping:**
- **Texture Filtering:**
- **Mipmaps:**
- **Texture Units:**
- **Vector:**
- **Matrix:**
- **Local Space:**
- **World Space:**
- **View Space:**
- **Clip Space:**
- **Screen Space:**
- **LookAt:**
- **Euler Angles:**

LIBRARIES

- **GLAD:**
- **GLFW:**
- **stb_image:**
- **GLM:**

Vocabulary

Sunday, March 27, 2022 3:13 PM

OpenGL

- A formal *specification* of a graphics API that defines the layout and output of each function.

CONCEPTS

- **Viewport**
  - The 2D window region where we render to.
- **Graphics Pipeline**
  - The entire process vertices have to walk through before ending up as one or more pixels on the screen.
- **Shader**
  - A small program that runs on the graphics card. Several stages of the graphics pipeline can use user-made shaders to replace existing functionality.
- **Vertex**
  - A collection of data that represents a single point.
- **Normalized Device Coordinates (NDC)**
  - The coordinate system your vertices end up in after perspective division is performed on clip coordinates. All vertex positions in NDC in the range `(-1.0, 1.0)` will not be discarded or clipped and end up visible on the screen.
- **Vertex Buffer Object (VBO)**
  - A buffer object that allocates memory on the GPU and stores all the vertex data there for the graphics card to use.
- **Vertex Array Object (VAO)**
  - Stores buffer and vertex attribute state information.
- **Element Buffer Object (EBO)**
  - A buffer object that stores indices on the GPU for indexed drawing.
- **Uniform**
  - A special type of GLSL variable that is global (each shader in a shader program can access this uniform variable) and only has to be set once.
- **Texture**
  - A special type of image used in shaders that is usually wrapped around objects, giving the illusion an object is extremely detailed.
- **Texture Wrapping**
  - Defines the mode that specifies how OpenGL should sample textures when texture coordinates are outside the range `(0.0, 1.0)`.
    - e.g. Repeat, Mirrored Repeat, Clamp to Edge, and Clamp to Border.
- **Texture Filtering**
  - Defines the mode that specifies how OpenGL should sample the texture when there are texels (texture pixels) to choose from. This usually occurs when a texture is magnified.
    - e.g. Nearest Neighbor and Linear.
- **Mipmaps**
  - Stored smaller versions of a texture where the appropriate-sized version is chosen based on the distance from the viewer. These textures are usually only used when a texture is minified.
    - Mipmaps also have their own filtering specifications, e.g. Nearest Neighbor and Linear.
- **Texture Units**
  - Allows for multiple textures on a single shader program by binding multiple textures, each to a different texture unit.
- **Vector**
  - A mathematical entity that defines directions and/or positions in any dimension.
- **Matrix**
  - A rectangular array of mathematical expressions with useful transformation properties.
- **Local Space**
  - All coordinates relative to the object's origin. This is the space an object begins in.
- **World Space**
  - All coordinates relative to a global origin.
- **View Space**
  - All coordinates as viewed from a camera's perspective.
- **Clip Space**
  - All coordinates as viewed from a camera's perspective, but with projection applied. This is the space the vertex coordinates should end up in, as output of the vertex shader. OpenGL does the rest (clipping/perspective division).
- **Screen Space**
  - All coordinates as viewed from the screen. Coordinates range from 0 to screen width/height.
- **LookAt**
  - A special type of view matrix that creates a coordinate system where all coordinates are rotated and translated in such a way that the user is looking at a given target from a given position.
- **Euler Angles**
  - Defined as yaw, pitch, and roll angles that allow us to form any 3D direction vector from these 3 values.

LIBRARIES

- **GLAD**
  - An extension loading library that loads and sets all OpenGL's function pointers for us so we can use all (modern) OpenGL's functions.
- **GLFW**
  - An open source, multi-platform library for OpenGL that provides a simple API for creating windows, contexts and surfaces, receiving input and events.
- **stb_image**
  - Image loading library.
- **GLM**
  - A mathematics library tailored for OpenGL.