COMP3320 Introduction to OpenGL

Alex Biddulph

The University of Newcastle, Australia

Based on the work provided at `www.learnopengl.com`

Semester 2, 2019

What is OpenGL?

- A standard, maintained by the Khronos Group [1], specifying how graphics operations should behave
  - Each operation is specified to generate a certain result
  - Graphics card manufacturers are free to implement operations however they please, provided the result is meets the standard
- An API (Application Programming Interface)
  - Operating system agnostic
  - Window system agnostic

[1] The Khronos Group Inc: `www.khronos.org`

What is OpenGL?

- A rendering library
  - An external library is needed to create a window that OpenGL can render on to
- A state machine
  - OpenGL only knows about triangles
  - The current state tells OpenGL how to render those triangles

Common OpenGL Libraries

GLFW [2]: Allows you to:

- Create and manage windows and OpenGL contexts
- Handle keyboard, mouse, and joystick inputs

GLAD [3]: OS-specific library abstracting away from the graphics card's implementation of the OpenGL functions

GLM [4]: OpenGL C++ Mathematics library based on the OpenGL Shading Language (GLSL)

SOIL [5]: Simple OpenGL Image Library - a small C library useful for uploading image textures into OpenGL

ASSIMP [6]: Open Asset Import Library - useful for loading 3D models from various common formats

---

[2] GLFW: `www.glfw.org`
[3] GLAD: `glad.dav1d.de`
[4] OpenGL Mathematics: `glm.g-truc.net/0.9.9/index.html`
[5] Simple OpenGL Image Library: `www.lonesock.net/soil.html`
[6] The Open-Asset-Importer-Lib: `www.assimp.org`

OpenGL Workflow with GLFW and GLAD

1. Initialise GLFW and set OpenGL context version and profile to use
   - We will use OpenGL context version 3.3 and the core profile in these examples
2. Create a window and set its width, height, and title
3. Make the window's context the main context for the current thread
4. Initialise GLAD and set it up to find all of the OpenGL function pointers (this is OS specific)
5. Set up callback functions to handle window resizing and user inputs
6. Set up rendering objects and textures
7. Enter a rendering loop that handles updating the screen
8. Clean up