SHADERS

- Scary!
- Weird terminology
  - Primitive assembly
  - Rasterization
  - Zwrite
  - Cull
  - Stencil
- Math and Data
  - Vertices, Fragments
  - Vectors, Matrices, Textures
  - Cross and Dot product
  - Matrix multiplication
OPENGL PIPELINE
HOW TO LEARN?

- Understanding the pipeline?
- How about Unity?
- Please keep trying!

Experiment → Fail → Learn → Repeat
SHADERS

• What is a shader?
  • It is a small program that runs on the GPU
  • Usually written in a high level shader language (e.g. GLSL)
  • Produce images
  • Input: Mesh, Material Data, Lighting Data, and etc.

• common shaders:
  • Vertex Shader: executes once for every vertex
  • Fragment Shader: executes one for every fragment (potential pixel)
SHADERS IN THE GRAPHICS PIPELINE

OpenGL (application software)
SHADERS IN THE GRAPHICS PIPELINE

Diagram showing the pipeline:
- Vertices
- Transf. Vertices
- Connectivity information
- Assembly
- Colored Fragments
- Interpolation
- Fragments
- Raster
VERTEX SHADER APPLICATIONS

- Moving vertices
  - Transformations
  - Morphing
  - Wave motion (e.g., water)
  - Fractals
- Lighting
  - More realistic models
  - Cartoon shaders
FRAGMENT SHADER APPLICATIONS

Per fragment lighting calculations

per vertex lighting

per fragment lighting
Texture mapping

smooth shading  environment mapping  bump mapping
LAST OF THE BUILT-INS

- `isnan(x)` – true is x is not a number (NAN)
- `isinf(x)` – returns true is x is $+\infty$ or $-\infty$
- `floatBitsToInt(x)` – converts floating point values to ints
- `intBitsToFloat(x)` – converts integers to floating points
SHADER LANGUAGES

- HLSL
  - High Level Shading Language
  - Microsoft
  - DirectX
  - Windows, XBox

- GLSL
  - OpenGL Shading Language
  - OpenGL ARB (Architecture Review Board)
  - OpenGL
  - Windows, Mac, Linux, iOS, Android and more

- CG
  - C for Graphics
  - Nvidia
  - DirectX & OpenGL
  - Deprecated but…
  - Thanks to Unity, covers all platforms
UNITY!

- ShaderLab
  - Unity specific shading and material language
  - All shaders written in Unity must be wrapped with ShaderLab
  - May seem like extra work but it’s actually saving you from a lot more work