CGS – Final Study Guide

# Day I

* Know It Cold: Working with Hexadecimal color values
* Know It Cold: Plotting Pixels on the Frame Buffer / Back Buffer.
* What is Blitting? How does it work? What is its history?
* What are pixels? How do they relate to the Raster? How can they be stored?
* What is Gamma? How is it implemented? Be able to perform luminance conversions.
* Know the history of the raster display. How does it compare to a vector display?
* Though they look similar, how do most modern 2D games differ from classic 2D games?

# Day II

* Know It Cold: Slope-Intercept
* Know It Cold: Vectors & Gradients
* Know It Cold: Implicit & Parametric Line Equations
* What are the various Line Drawing Algorithms? How do they work?
* What is linear interpolation? What are some of its many applications?
* What are Octants? How do they impact line drawing?

# Day III

* Know It Cold: Normalized Device Coordinates
* Know It Cold: Barycentric Space & Coordinates
* Know It Cold: Interpolation – Linear & Barycentric
* Know It Cold: Vector3 & Matrix3x3 operations
* What are some of the pros & cons of 3D Cartesian vs. Screen/Pixel Space?
* What is Rasterization? Why do we need it?
* What are Shaders? What advantages do they give us?
* Review the triangle drawing algorithms & what they all have in common.

# Day IV

* Know It Cold: Local Space, World Space, View Space, Projection Space, NDC Space.
* Know It Cold: Vector4 & Matrix4x4 operations.
* Know It Cold: Orthographic vs. Perspective Projection
* Know It Cold: Homogenous Coordinates.
* What is the perspective divide? When is it carried out and why?
* What is Global Space? How does it differ from World Space?
* What is the viewing frustum? What components define its make up? When are they applied?

# Day V

* Know It Cold: The Z Buffer Algorithm
* Know It Cold: UV coordinates & Texels
* Know It Cold: Texture Mapping Triangles
* What is the Painter’s Algorithm? Why was it used early on? How can it fail?
* Review the advantages of the Depth Buffer. Does the Painter’s Algorithm have any?
* What are Texture Coordinates? How does a Pixel differ from a Texel?
* What does it mean to “sample” a texture? What is the process during Rasterization?
* Understand the key differences between a Linear vs. an Exponential Z buffer.

# Day VI

* Know It Cold: Perspective Correct Interpolation
* Know It Cold: Bi-Linear Texture Filtering
* Know It Cold: MipMaps – Generation, Storage and Retrieval
* Know It Cold: Clipping primitives against the near plane
* What is the difference between Affine texture mapping and Perspective Correct?
* Why does Mip-Mapping exist? What is it designed to combat?
* What is texture filtering? What are the various types of filters? How do they improve quality?
* Understand the rudimentary way we chose a specific Mip-Level in lab 6.
* Why do we need primitive clipping? Why does the space we choose to clip in matter?

# Day VII

* Know it Cold: The Dot Product and Cross Product
* Know it Cold: Applying lighting calculations through the vertex shader (Gouraud)
* Know it Cold: Applying lighting calculations through the pixel shader (Phong)
* Understand surface normals, their purpose and application.
* What are the different types of reflection models?
* What are the fundamental types of light sources? How are they implemented mathematically?
* What is an emissive surface? Can all surfaces be approximated with one simple formula?
* How does attenuation impact our lighting calculations?

# HARDWARE

* Know It Cold: The Core Interfaces required to initialize Direct3D11
* Know it Cold: How to Load vertices into VRAM and describe them to the 3D pipeline
* What are the key advantages of hardware rendering? Disadvantages?
* What is a video driver? How do they work with the API and why are they necessary?
* What is a Constant Buffer? How do you update its contents efficiently?
* How are shaders implemented in hardware? What is HLSL and what is its basic syntax?