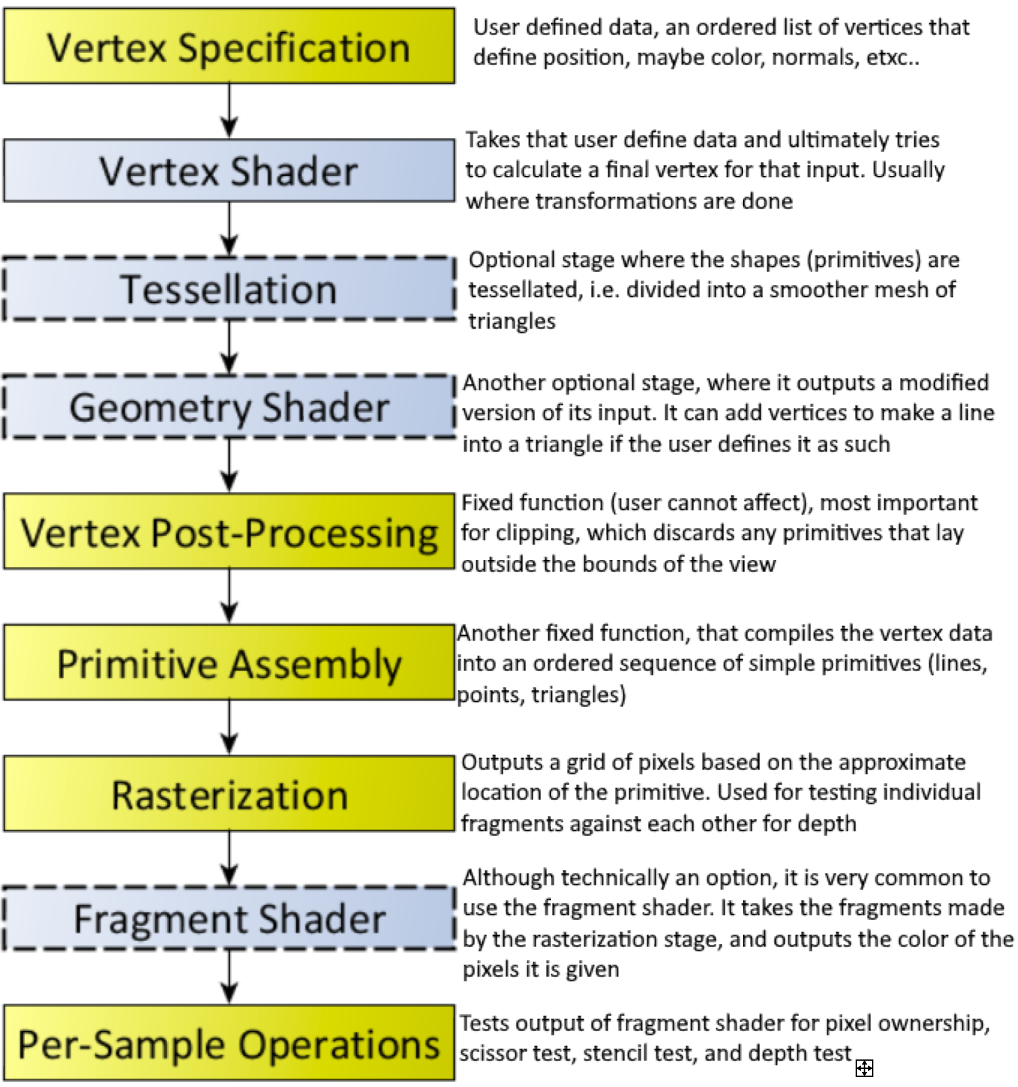
CSC 350 Exam #1, Study Guide

**Rendering Pipeline:**



**Human vision and colors:**

**Rods**:

-cannot distinguish colors

-used for low-light vision

-gray scaling

**Cones**:

-used for color vision

-three types:

-long, medium, short wave length

-thought to be used for R, G, B but are not

**Pupils**:

- black hole located at center of iris

- allows light to pass through into the retina

-size typically 4mm, but can be 2mm in light, or 8mm in dark

-low as 1mm when bright and older, and no more than 5-6 when dark

**Optic Nerve**:

-a paired cranial nerve

-transmits visual information from the retina -> the brain

-uses electrical impulses

-can take a beating, and still work relatively well.

**Adaptability**:

- Iris adjusts its size to let more/less light in depending on surroundings

- if dark, pupils open all the way to let as much light in as possible

-process begins 4 sec after intro to complete darkness

-about 80% done by 30 minutes

-if interrupted by light the whole process starts over again

-if light the pupil restricts to let less light in and protect the retina

**RGBA: Red Green Blue Alpha**

-A three-channel color model, with a fourth channel, alpha used for alpha compositing, which is essentially how transparent each pixel is.

**Key OpenGL Functions:**

**glGenBuffers(n, \*buffers):**

-tells OpenGL to create a buffer object and giving us the ID of said buffer

-A buffer is just a chunk of memory, but an OpenGL buffer actually uses VRAM, not RAM, since this is all being done on the GPU. This is an array of unformatted memory.

**-n:** number of buffers to be created

**-\*buffers:** pointer to a gluint that is dereferenced to return the “name” of the buffer, which is actually just a unique integer that is used to reference the buffer

**glBindBuffer(target, buffer)**

-Binds and actives the buffer we created, to make it so that any drawing comes from this buffer.

- **target:** is the intended use for the buffer, generally we use GL\_ARRAY\_BUFFER, which means that our buffer is intended to be used for vertex attribute data.

- **buffer**: ID of buffer to be bound (activated)

**glBufferData(target, size, \*data, usage)**

-Creates and initializes a Buffers data storage

- **size**: specifies the size (in bytes) of the data store

- **data**: a pointer to the data that will be copied into the buffers data storage.

- **usage**: how our data will likely be used

-**STATIC** = rarely modified, but used a lot.

**glEnableVertexAttribArray(index)**

-Enables a generic vertex attribute array

-**index**: Specifies the index (position) of the attribute we are enabling.

-0 reserved for position

-Other numbers for user-defined attribs such as color, normals, texture coordinates…

**glVertexAttribPointer(index, size, type, normalized, stride, \*pointer)**

-Our way of telling OpenGL what to do/how to interpret the data we sent to it using

glBufferData.

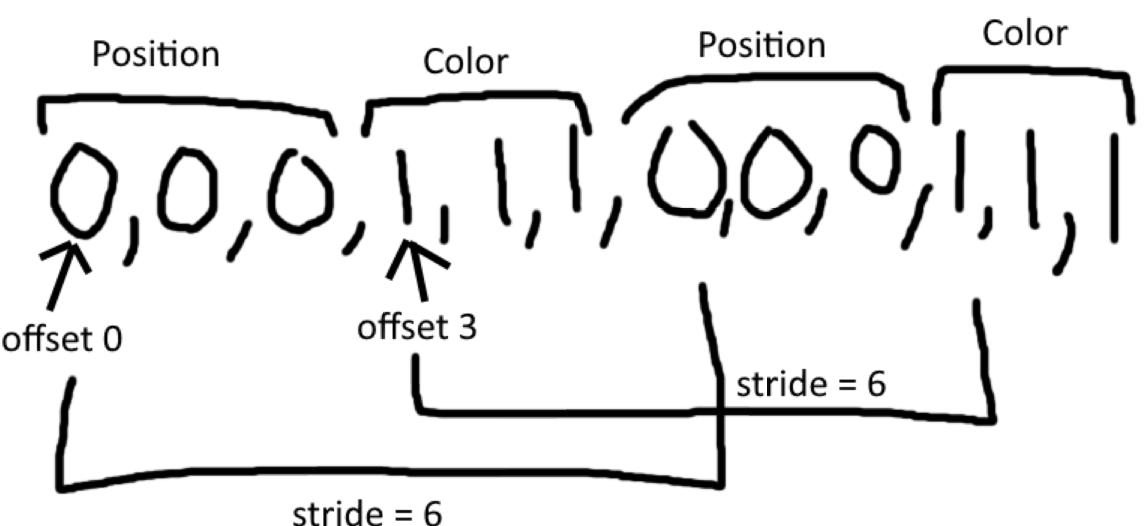
-**index**: the index of the attribute (0) for position (1) for color in our programs

-**size**: the number of elements to be read (must be 1, 2, 3, 4)

-**type**: data type of each element in the array

-**stride**: the byte offset between the first component of the attribute

-**offset**: where in the array to find the first element of the attribute



**glDrawArrays(mode, first, count)**

-This method is what actually renders the vertices from the array data

-**mode**: what kind of primitive shapes are doing to be render, in our cases, we exclusively use GL\_TRIANGLES

-**first**: tells OpenG: where to start getting the vertex data from in the array. Since our position data is at the start of our array data, we use 0.

-**count**: tells OpenGL them number of vertices to render

**glClear(mask)**

-clears buffer to preset values

-depth bit is 1, when we enable the depth test, we check every vertex against 1, if it is less than 1, then it appears, if it is 1 or more, then it doesn’t because it is “deeper”/farther away from the camera.

-GL\_COLOR\_BUFFER\_BIT: indicates that the buffer is enabled to do color writing.

-GL\_DEPTH\_BUFFER\_BIT: indicates the buffer is enabled to use depth bit

-color buffer is default set to 0, 0, 0, 0, black.

-depth buffer is default set to 1, and we test against that to see which fragment gets drawn

**glFlush()**

-this force all issued commands to be ran by the hardware in some amount of time. While the amount of time is not guaranteed, it is guaranteed to be finite. This essentially just forces OpenGL to flush the graphics rendering pipeline.

**Shaders:**

-Shaders are compiled on the GPU

-**Uniform:**

**-**similar to the keyword const.

-means that once it is defined it cannot be changed

-in OpenGL uniform variables are defined by the application

-uniform data in OpenGL will be the same throughout each stage of the shader pipeline. It can only be changed at the application level; the shaders cannot change it.

-**in:**

-a storage qualifier that defines linkage from a previous stage in the shader pipeline.

-These define the inputs for the shader

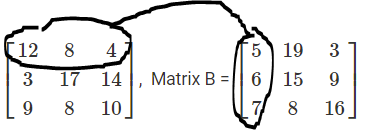
-**out:**

**-**also, a storage qualified, but it defines a link to the next stage in the shader pipeline

**-**These define the outputs for the shader.

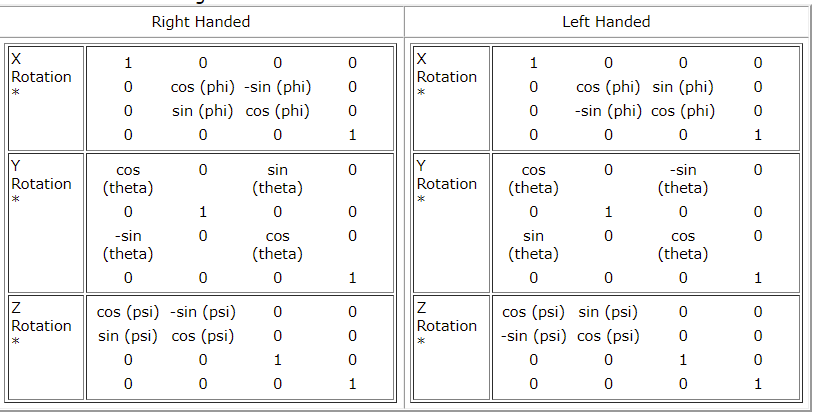
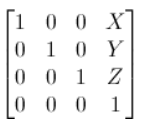
**Chapter 5:**

-**Matrix Multiplication:**



Dot product = (12 \* 5) + (8 \* 6) + (4 \* 7) = 136

-**Translation Matrix:**

-**Rotation Matrices**:

-

-**Scale Matrix:**



-**Vector (x, y, z, w)**

-x: position on x axis

-y: position on y axis

-z: position on z axis

-w: used for mapping 3d -> 2d

-Also, x, y, z are all divided by w

-Allows for transformations such as translation, rotation

-Allows for matrix multiplication to be used

-w = 0 for directions

-w = 1 for positions in Euclidian coordinates

-used to give the illusion of perspective close things big, far things small

**-Parallel vs. Perspective:**

**Parallel:**

**-**Objects that are farther away in the z, do not appear smaller

-center of projection at infinity

-all z cords are equal

-w = 1

**Perspective Projection:**

-Objects that are farther away appear smaller, give illusion of depth

-all points seem to converge as they get farther away (railroad tracks are always parallel, but when you look down them, they seem to merge)

-far: must be > near and defines the far plane that if anything if farther away than that plane, it will not be projected.

-near: must be < far and defines the near clipping plane, if an object is close than near, it will not be displayed.

-fov: the angle of view about the y axis, smaller means things seem more zoomed in, bigger makes things seem farther away.

**Chapter 7, Lighting:**

**Ambient Lighting basics:**

-Indirect light

-the minimum light value anything would have

-simply multiply the ambient light level to everything

-material generally does not matter

**Diffuse lighting:**

**-**the angle of the light affects its intensity

-90 degrees = most intense

-do dot product of normalized surface normal and the light vector (cos theta)

-material matters

**Specular lighting:**

**-**reflective lighting

-changes as the eye moves

-first normalize the camera (eye) vector + the diffuse light vector calculated previously

-take that result and do the dot product of it and the surface normal / 2

**Dot product of Vectors:**

**-**a = (1, 2, 3)

-b = (4, 5, 6)

-So, a b = (1 \* 4) + (2 \* 5) + (3 \* 6) = 32

-to get angle between them:

-a b / (|a| \* |b|)

-|a| = √12 + 22 + 32 = √1+ 4 + 9 = √14

-|b| = √42 + 52 + 62 = √16 + 25 + 36 = √77

- cos( = 32 / (√14 **\*** √77) **=** .912, So, θ = cos-1(.912) = .423 or 24.23