Computer Graphics **Test 1 Review**

Here is a list of problems/topics covered in test 1:

- How is vertex interpolation done? For example, given the coordinates of two points, how does the graphics system figure out the location of all the points in between these two vertices? How will this method be applied to interpolation of color along a line or on a geometric primitive such as a triangle?

- Given a minimum WebGL program, for example your homework 1 program, understand the precise meaning of every single statement in the program, both the Javascript program and vertex and fragment shader programs.
  - Understand the relationship among a variable defined in a vertex shader (or fragment shader), the variable in the Javascript program that references the location of that variable in the shader program,
  - Understand how the buffer on the GPU is created and linked to ARRAY_BUFFER, and how the actual data is passed to the buffer and being used to send to the vertex shader for processing.

- Be able to write the Javascript code to draw a 2D shape
  - Know the WebGL drawing primitive elements used in gl.drawArrays;
  - Know how to generate the vertex positions, and to send the vertex positions to vertex shader;
  - Know how to compute the proper transformation matrix, and to send the matrix to vertex shader;
  - Know how to send vertex color to vertex shader and later pass onto fragment shader for color interpolation;
  - Know how to add new data to buffer already created on GPU
  - Be able to apply recursion to tessellate triangle

- Shader programs
  - What is the main purpose of a vertex shader program?
  - What is the main purpose of a fragment shader program?
  - When to use “attribute”, “uniform”, and “varying” qualifiers to declare variables in vertex shader and fragment shader

- Event handle:
  - Be able to write Javascript code segments to add event handler to various events, including mouse clicks, keyboard entry
  - How to convert mouse click positions to the internal representations/values of the corresponding positions

- Vector Analysis - Refer to problems in homework 3 and class notes on vector analysis

- Transformation
  - Understand how to write the transformation matrix given verbal transformation description. For example what will be the transformation matrix if we would like to translate a figure **a** units along the x axis and **b** units along the **z** axis, or if we would like to flip the figure along the **x** axis.
    - translate, rotate, scale, and shear transformation operation and the corresponding matrices
    - how to use modelviewmatrix during composite transformations
    - how to use push and pop operations with modelviewmatrix stacks
  - Given a simple geometric shape in terms of the definitions of the vertices, and a transformation matrix, be able to derive the transformed shape by computing the transformed vertices
  - Be able to draw figures using transformations, modelviewmatrix, and push and pop operations
  - Understand and perform world window to viewport transformation
  - Zooming and panning using ortho

- Animation
  - Be able to write code to perform simple animation
  - What specifically does the “requestAnimFrame” function do?
  - Use setTimeOut in conjunction with “requestAnimFrame” to control the speed of animation.