Tutorial Questions Week 4

2.9 We saw that a fundamental operation in graphics systems is to map a point (x, y) that lies within a clipping rectangle to a point (xs, ys) that lies in the viewport of a window on the screen. Assume that the two rectangles are defined by the viewport specified by glViewport(u, v, w, h); and a viewing rectangle specified by xmin ≤ x ≤ xmax, ymin ≤ y ≤ ymax. Find the mathematical equations that map (x, y) into (xs, ys).

Answer:

We can solve this problem separately in the x and y directions. The transformation is linear, that is xs = ax + b, ys = cy + d. We must maintain proportions, so that xs in the same relative position in the viewport as x is in the window, hence

.

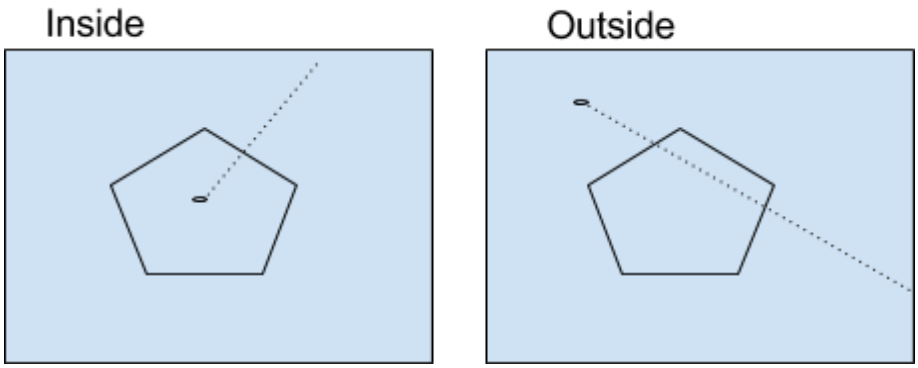
$$\frac{x - x_{min}}{x_{max} - x_{min}} = \frac{x_s - u}{w},$$

$$x_s = u + w\frac{x - x_{min}}{x_{max} - x_{min}}.$$

$$y_s = v + h\frac{x - x_{min}}{y_{max} - y_{min}}.$$

2.11 In practice, testing each point in a polygon to determine whether it is inside or outside the polygon is extremely inefficient. Describe the general strategies that you might pursue to avoid point-by-point testing.

Answer:
You can determine whether or not you are inside the object by checking the number of vertices you cross in one direction. If you get a even number of crossed vertices-> you're outside the object, odd number= inside the object.



2.14 In OpenGL we specify polygons using lists of vertices. Why might it be better to define polygons by their edges? Hint: Consider how you might represent a mesh efficiently.

Answer:

Every vertex for polygon is stored as 72 values in total in the shader.
If saving it as faces and reference to the vertices so it gets faster 48 values while reading them out

When we store every vertex for a polygon for a cube it is 72 values in total (store it like this in the shader)
Instead we save the faces and reference to the vertices so it gets faster 48 values(while reading them out)

Answer the following JavaScript related questions.
a. Do you need to compile JavaScript code in order to execute it?
Answer: No.

b. Variable type: Do you need to declare the type of a variable before using it? Why?
Answer: No, it's just "Var". JS itself determine which kind of variable it is.

c. Variable scoping: What are global variables? How are they declared? What are the problems with using globals? d. How are errors gracefully handled in JavaScript?

Answer: Variables accessible from the entire file. Only global variables in the virtual machine. Really dangerous, global to more than your program which means that another js file can use the same variable name and make your program crash.

e. Does JavaScript support objects?

Answer: Yes, it's object oriented.

f. What is the relationship between canvas and WebGL?

Answer: The canvas is where WEBGL renders the graphic. Which is done by giving context to the canvas with a webgl reference.