Com S 336  
Fall 2014  
Notes on midterm exam

This will be a 75 minute written test covering everything we’ve done up through the ADS lighting model and Phong shading. Some topics and review questions are listed below. The written homework from the beginning of the course provides some good examples for exam questions on transformations. Some other ideas for review or exam questions are shown in italics.

You may need to write short pieces of Javascript code or GLSL code. You do not have to memorize specific OpenGL functions but you should be familiar with how to use them in loading data and rendering. We will not worry about code to load and compile shaders. You’ll need to be able to write shader code at roughly the level of detail of the examples we’ve done and should be familiar with the basic GLSL types and operations (e.g. mat4, vec3, the normalize function, etc.)

References

Keep in mind that 100% of what we've covered has been applied in examples or homework, so you can always see it in action by studying the code!

Gortler Chapter 1 is a brief overview of the graphics pipeline.  
Teal Book Ch. 3, the first section ("Drawing Multiple Points"), includes some nice illustrations of OpenGL buffers. Also our first code example, GL_example1, includes comments explaining what's going on in detail.  
Gortler Chapters 2 - 5 cover transformation matrices and frames (though not the view and projection matrices).  
Teal Book chapters 3 and 4 talk about transformations at a more elementary level. (Also see all the examples in the transformations directory.)  
In addition, Teal Book Ch. 9 discusses the idea of hierarchical objects (the stuff explained in mathematical terms in Gortler Ch. 5). (Also see the Hierarchy example in the homework3 directory.)  
For the view and projection matrices, see Chapter 7 of the Teal Book. The section "Specifying the Viewing Direction" explains the view point and setLookAt function. The section "Specifying the Visible Range Using a Quadrangular Pyramid" has a nice diagram showing the parameters of a perspective projection using setPerspective or setFrustum. (Also see the comments in the definitions of the view and projection variables in RotatingCube.js, which you can play with.)
There is a terse explanation of lighting in Gortler 14.1 - 14.3. The Teal Book Chapter 8 provides a good explanation of diffuse lighting and normal vectors, but does not discuss the specular component. This page, http://pages.cpsc.ucalgary.ca/~slongay/pmwiki-2.2.1/pmwiki.php?n=CPSC453W11.Lab12 has a good picture and a brief explanation of the Phong reflection model. (Remember the symbols k and L stand for vectors with red, blue, and green component). Maybe the best way to understand the lighting model is to study the shader code in the examples in the lighting directory, though.

Topics

The graphics pipeline:
   (Model -> )
   Vertex processing -> Primitive assembly -> Rasterization -> Fragment processing
   (-> framebuffer)

Role of shaders in the pipeline
- Could you write a vertex shader that uses two adjacent vertices to calculate a normal vector? Explain briefly.
- Could you write a shader that changes the way clipping is done?

OpenGL buffers and vertex attributes
- Describe the process of loading data and connecting data to vertex attributes
- Could the same data be used by more than one shader without reloading the data on the GPU? Explain briefly.
- Given a sample of code such as our original GL_example1.js, modify it so that each vertex of the square has a different color and those colors are interpolated when it is rendered

Linear interpolation
- Suppose you need to scale and bias the range 200 to 400 into the range -5 to 5. Write a formula to do it.
- Create an affine transformation $M$ such that given a coordinate vector $c = [x, y, 0, 1]^T$, $Mc = [x', y', 0, 1]^T$, where $x'$ is equal to $x$ converted from Celsius to Fahrenheit and $y'$ is $y$ converted from Fahrenheit to Celsius. (Write your answer as a the product of a scale matrix followed by a translation matrix.)

Basic color representation
Clip coordinates, clipping volume, depth testing
- Why does clipping potentially create new polygons?
- Describe the z-buffer algorithm using pseudocode

Vertex attributes, “varying” variables
- What is the role a variable declared as attribute?
- What is the role of a varying variable?
- Write a fragment shader that colors each pixel with a greyscale value obtained from the average of the red, green, and blue values of a varying variable called fColor.
- Write a complete shader program (vertex shader and fragment shader) that will darken pixels according to their depth in clip space. Assume that the vertex position and color are attributes, and that the position is given in clip coordinates. Points that are closest to the viewer should have the given color, and points farthest from the viewer should be 50% darker than given.

OpenGL primitives
- Given a set of vertices, show what is rendered using glDrawArrays(GL_LINES, ...), glDrawArrays(GL_TRIANGLE_FAN, ...), etc.
- Indexed rendering
- Given a bunch of vertices, write down the indices that would be used to render it using glDrawElements(GL_TRIANGLES, ...), or b) as a wireframe using glDrawElements(GL_LINES, ...),

Linear transformations as matrices: given a transformation f, what does f do to the basis vectors?
Then M = \[
\begin{bmatrix}
 f(e1) & f(e2) & f(e3)
\end{bmatrix}
\]
- Suppose e1 and e2 are the standard 2d basis vectors, and that 2d linear transformation takes e1 to \([5, 7]^T\) and e2 to \([-1, -3]^T\). Write down a matrix for this transformation.

Using homogeneous coordinates to represent translations
An affine transformation is a linear transformation followed by a translation
- Prove that the 4d matrix for an affine transformation always has \((0 \ 0 \ 0 \ 1)\) in bottom row.
- Suppose M is an affine matrix whose right column is \([x \ y \ z \ 1]^T\), where x, y, and z are nonzero. Under what conditions is it possible that the right column of \(M^{-1}\) is exactly \([-x \ -y \ -z \ 1]^T\)?

2D rotation, translation, scaling
- Given a 2D triangle with vertices \((0, 0), (2, 0), (2, 1)\), sketch the triangle after
  a) translation by \((4, 5)\) followed by rotation by 90 degrees
  b) rotation by 90 degrees followed by translation by \((4, 5)\)
  c) translation by \((4, 5)\) followed by scaling by 2 in the y direction
(show the approximate new coordinates in each case, e.g., on graph paper)
- For each of a, b, c write down the transformation matrix (as a product of standard matrices)

A frame is a set of basis vectors, plus an origin
Coordinate systems used in OpenGL: model, world, eye/view, clip, window, viewport
- What coordinate system are the fragment coordinates in (e.g. the gl_FragCoord variable accessible in the fragment shader)?

Points vs vectors in homogeneous coords
Standard matrices: Translate(), RotateX(), RotateY(), RotateZ(), Scale()
- Suppose an object is centered at \((p_1, p_2, p_3)\) and we want to rotate it through angle \(r\) about its z axis. Write down the necessary matrix in terms of the standard matrices (e.g. \(\text{Translate}()\), etc.) (Don’t multiply them out!)

- Suppose an object is centered at \((p_1, p_2, p_3)\) and we want to rotate it through angle \(r\) about a line through its center, where the line is directed at angle \(\theta\) from the positive y-axis and angle \(\phi\) from the positive x-axis, as in spherical coordinates. Write down the necessary matrix in terms of the standard matrices.

Intrinsic vs. Extrinsic transformations

Inverting a product of standard transformations

An affine transformation consists of scaling by \((1, 2, 1)\), then a translation to \((1, 2, 3)\). Describe the matrix for this transformation, and calculate its inverse. (You don’t have to multiply out the result.)

Changing frames; the view transformation, \(M\) vs \(M^{-1}\)

- Suppose you are working in 2D and your new frame has origin \((5, 10)\) and that its x-axis is given by the vector \([0, 1]\) and its y-axis is given by \([-1, 0]\).
  a) Find the matrix that transforms the standard frame into the second one
  b) What are the coordinates of the point \((1, 2)\) with respect to this new frame? Draw a picture to verify.
  c) Find the matrix \(S\) that gives you the new coordinates of any point with respect to the new frame, e.g. \(S[1, 2, 0, 1]^T\) would be the answer to (b)

- An object is centered at \((2, 3, 4)\). What transformation will make the object 5 times as big without moving its center?

- Suppose some frame \(F'\) is obtained from the world frame by matrix \(A\). What transformation will rotate an object 30 degrees ccw about the z-axis of frame \(F'\)?

- Suppose you are given coordinates for three orthonormal vectors \(x = [x_1 \ x_2 \ x_3 \ 0]^T\), \(y = [y_1 \ y_2 \ y_3 \ 0]^T\) and \(z = [z_1 \ z_2 \ z_3 \ 0]^T\), plus a point \(p = [p_1 \ p_2 \ p_3 \ 1]^T\), that together describe a frame that you would like to use as your camera. Write down the view matrix.

- Write the pseudocode for the function \(\text{LookAt}(\text{eye}, \text{at}, \text{up})\). Assume you have basic functions such as \(\text{RotateX}()\), \(\text{Translate}()\), etc

Orthographic projections, view volume

- Describe what the function \(\text{setOrtho}(\text{left}, \text{right}, \text{bottom}, \text{top}, \text{near}, \text{far})\) does.
- Why do we have to specify near and far clipping planes?

Perspective projection

- What would the arguments to the \(\text{setFrustum}(...)\) function be in order to give you the same matrix as \(\text{setPerspective}(30, 1.5, 2, 100)\)?

Roles of the model, view, and projection matrices

- How do these three matrices relate to some of the coordinate systems we have discussed?
- Describe an example showing why we might want to keep them separate

Normal vectors
How to calculate face normals from vertex coordinates
How to calculate vertex normals from face normals

Lambert shading
- Why do we have to normalize L and N in our lighting calculation?

Phong (ADS) lighting model
- Suppose at a given vertex, L points to the light and N is the normal vector and V points to the view point. Assume all three are normalized. Sketch a picture. Write down the three components of the lighting model as accurately as you can.

Gouraud and Phong shading
- Briefly explain the difference between Gouraud and Phong shading
- Explain the nature of the artifacts that occur using Gouraud shading with a specular component
- Why do we have to normalize after interpolating vectors? Give an example showing that interpolating between two vectors of length 1 does not necessarily give you a vector of length 1.
- Write a complete shader pair for per-fragment shading using a point light. Assume that the light position is given in world coordinates as a uniform variable, and that the model, view, and projection matrices are given as uniform variables.
- What do you have to change in the above if instead the light is given as a direction vector in world coordinates, rather than as a point?
- We normally do lighting calculations in eye coordinates. Could you do it just as well in world coordinates? Explain what might be easier/harder/different.