CPSC 352 Test 2 Information

Coverage: Everything we’ve covered in lectures on chapters 6, 7, WebGL, and Three.js

Format: Short answer and problem solving (understanding terms and algorithms, no programming)

Topics include:

1. Chapter 5: Viewing
   a. Projections: parallel vs. perspective. COP or DOP. Projectors. Orthographic projections.
   b. View frustum; front and back clipping planes
   c. Orthographic projection. One, two, three-point perspective.
   d. Camera API: Eye point, LookAt point, up vector
   e. Projection API: fov, aspect, near, far; uvn frame (VRP, VUP, VPN)
   f. Perspective projections: similar triangles to compute coordinates.
   g. Perspective matrices, divide by w step
   h. How can you use projections to fake shadows?

2. Chapter 6: Lighting and shading
   a. Global vs. local illumination models
   b. Light sources: ambient, distant, point, spot
   c. Umbra, penumbra
   d. Phong illumination model – understand the Phong illumination equation (ambient, diffuse, specular, specular power)
   e. Lambert’s law and Lambertian surfaces
   f. Specular power
   g. Shading using the Phong illumination model
   h. Flat shading
   i. Gouraud and Phong shading – advantages and disadvantages?
   j. Bilinear interpolation

3. WebGL Intro
   a. WebGL ES pipeline
   b. Vertex shaders, fragment shaders
   c. Uniforms, varying variables, samplers
   d. Vertex buffer objects, program objects
   e. WebGL primitives
   f. Polygon triangulation (Van Gogh algorithm)
   g. Depth buffer algorithm for hidden surface removal
   h. Three.js concepts
   i. 3D morphing

4. Chapter 7: Discrete techniques
   a. Texture mapping
      i. Use of texture coordinates
      ii. Adjusting for scale. Nearest vs. linear interpolation
      iii. Mipmaps
      iv. Combining multiple textures
   b. Skybox
   c. Environment (reflection) mapping
   d. Light mapping
   e. Sprites
   f. Bump mapping, normal mapping
   g. Displacement mapping
   h. Billboard
      i. Blending. Problems with depth buffering. Screen-door transparency
   j. Particle systems
   k. Antialiasing
   l. OpenGL’s buffers: Color, depth, stencil. What each is used for. How (conceptually) you would do things like fog, depth of field, etc. with buffers.