



Attempt all the following questions.

Demonstrate with **examples**, **figures**, and/or **equations** whenever possible.

**No of Questions:** 5 in 2 page(s)

**Total Mark:** 75

**Question 1:** (15 marks)

- The pipeline approach to image generation is nonphysical. What are the main advantages and disadvantages of such a nonphysical approach? (5 marks)
- Show that a rotation and a uniform scaling commute. (5 marks)
- Consider a satellite orbiting the earth. Its position above the earth is specified in polar coordinates. Find a model-view matrix that keeps the viewer looking at the earth. Such a matrix could be used to show the earth as it rotates. (5 marks)

**Question 2:** (15 marks)

- The *Phong model* supports three types of material-light interactions: ambient, diffuse, and specular. Explain the three types and identify the four vectors used in the model. (7 marks)
- Show how can the texture be mapped to: (8 marks)
  - Cube
  - Cylinder
  - Cone
  - Sphere

**Question 3:** (15 marks)

You can generate a simple maze starting with a rectangular array of cells. Each cell has four sides. You remove sides (except from the perimeter of all the cells) until all the cells are connected. Then you create an entrance and an exit by removing two sides from the perimeter. Write an application that takes as input the two integers N and M and then draws a **3D**  $N \times M$  maze.



**Question 4:** (15 marks)

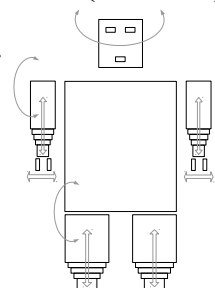
Construct an interactive program that will allow the user to

- position one or more light sources, (4 marks)
- alter the material properties of a unit cube centered at the origin, and (5 marks)
- change the camera position. (6 marks)

**Question 5:** (15 marks)

Robotics is one example in which the parts of the scene show compound motion, where the movement of some objects depends on the movement of other objects. Write an application for rendering

- a humanoid robot with (6 marks)
- two grippers and (3 marks)
- telescopic limbs. (3 marks)
- The application should allow animating the robot using sliders. (3 marks)



*Turnover the page, please.*

## Reference Application

```
1. <!DOCTYPE html>
2. <html>
3. <meta http-equiv="Content-Type"
4. content="text/html; charset=utf-8" >
5. <title>Rotating Square</title>
6.
7. <script id="vertex-shader" type="x-shader/x-vertex">
8. attribute vec4 vPosition;
9. uniform float theta;
10. void
11. main()
12. {
13.     float s = sin( theta );
14.     float c = cos( theta );
15.
16.     gl_Position.x = -s * vPosition.y + c * vPosition.x;
17.     gl_Position.y = s * vPosition.x + c * vPosition.y;
18.     gl_Position.z = 0.0;
19.     gl_Position.w = 1.0;
20. }
21. </script>
22.
23. <script id="fragment-shader" type="x-shader/x-fragment">
24. precision mediump float;
25. void
26. main()
27. {
28.     gl_FragColor = vec4( 1.0, 0.0, 0.0, 1.0 );
29. }
30. </script>
31.
32. <script type="text/javascript"
33. src="../../Common/webgl-utils.js"></script>
34. <script type="text/javascript"
35. src="../../Common/initShaders.js"></script>
36. <script type="text/javascript"
37. src="../../Common/MV.js"></script>
38. <script type="text/javascript">
39. "use strict";
40.
41. var gl;
42. var theta = 0.0;
43. var thetaLoc;
44. var delay = 100;
45. var direction = true;
46.
47. window.onload = function init()
48. {
49.     var canvas = document.getElementById( "gl-canvas" );
50.     gl = WebGLUtils.setupWebGL( canvas );
51.     if ( !gl ) { alert( "WebGL isn't available" ); }
52.
53.     var vertices = [
54.         vec2( 0, 1 ),
55.         vec2( -1, 0 ),
56.         vec2( 1, 0 ),
57.         vec2( 0, -1 )
58.     ];
59.
60.     gl.viewport( 0, 0, canvas.width, canvas.height );
61.     gl.clearColor( 1.0, 1.0, 1.0, 1.0 );
62.
63.     var program = initShaders( gl,
64.         "vertex-shader", "fragment-shader" );
65.     gl.useProgram( program );
66.
67.     var bufferId = gl.createBuffer();
68.     gl.bindBuffer( gl.ARRAY_BUFFER, bufferId );
69.     gl.bufferData( gl.ARRAY_BUFFER, flatten(vertices),
70.         gl.STATIC_DRAW );
71.
72.     var vPosition = gl.getAttribLocation( program,
73.         "vPosition" );
74.     gl.vertexAttribPointer( vPosition, 2, gl.FLOAT,
75.         false, 0, 0 );
76.     gl.enableVertexAttribArray( vPosition );
77.
78.     thetaLoc = gl.getUniformLocation( program, "theta" );
79.
80.     document.getElementById("slider").onchange =
81.     function(event) {
82.         delay = 100 - event.target.value;
83.     };
84.
85.     document.getElementById("Direction").onclick =
86.     function (event) {
87.         direction = !direction;
88.     };
89.
90.     document.getElementById("Controls").onclick =
91.     function( event ) {
92.         switch(event.target.index) {
93.             case 0:
94.                 direction = !direction;
95.                 break;
96.             case 1:
97.                 delay /= 2.0;
98.                 break;
99.             case 2:
100.                 delay *= 2.0;
101.                 break;
102.         }
103.     };
104.
105.     window.onkeydown = function( event ) {
106.         var key = String.fromCharCode(event.keyCode);
107.         switch( key ) {
108.             case '1':
109.                 direction = !direction;
110.                 break;
111.             case '2':
112.                 delay /= 2.0;
113.                 break;
114.             case '3':
115.                 delay *= 2.0;
116.                 break;
117.         }
118.     };
119.
120.     render();
121. };
122.
123. function render()
124. {
125.     gl.clear( gl.COLOR_BUFFER_BIT );
126.     theta += (direction ? 0.01 : -0.01);
127.     gl.uniform1f(thetaLoc, theta);
128.     gl.drawArrays(gl.TRIANGLE_STRIP, 0, 4);
129.     setTimeout(
130.         function () {requestAnimationFrame( render );},
131.         delay
132.     );
133. }
134.
135. </script>
136. </head>
137.
138. <body>
139. <div>
140.     delay 0% <input id="slider" type="range"
141.     min="0" max="100" step="10" value="50" /> 100%
142. </div>
143.
144. <button id="Direction">Change Rotation Direction</button>
145.
146. <select id="Controls" size="3">
147.     <option value="0">Toggle Rotation Direction</option>
148.     <option value="1">Spin Faster</option>
149.     <option value="2">Spin Slower</option>
150. </select>
151.
152. <canvas id="gl-canvas" width="512" height="512">
153. Your browser doesn't support the HTML5 canvas element.
154. </canvas>
155. </body></html>
```

Good Luck  
Dr. Islam ElShaarawy