



AD SOYAD :	DEĞERLENDİRME						
NUMARA :	İMZA:	1 →	2 →	3 →	4 →	5 →	T =

Aşağıdaki sorulardan yalnızca 4 tanesini cevaplayınız.

1.

Bilgisayara karşı oynanan bir 3D tank oyununda, bilgisayarı temsil eden tankın hedefe kilitlenmesi için (kullanılacak matris(ler)/vektör(ler) ve bunlar arasındaki işlemler hakkında bilgiler de içeren) bir algoritma öneriniz. (25P)

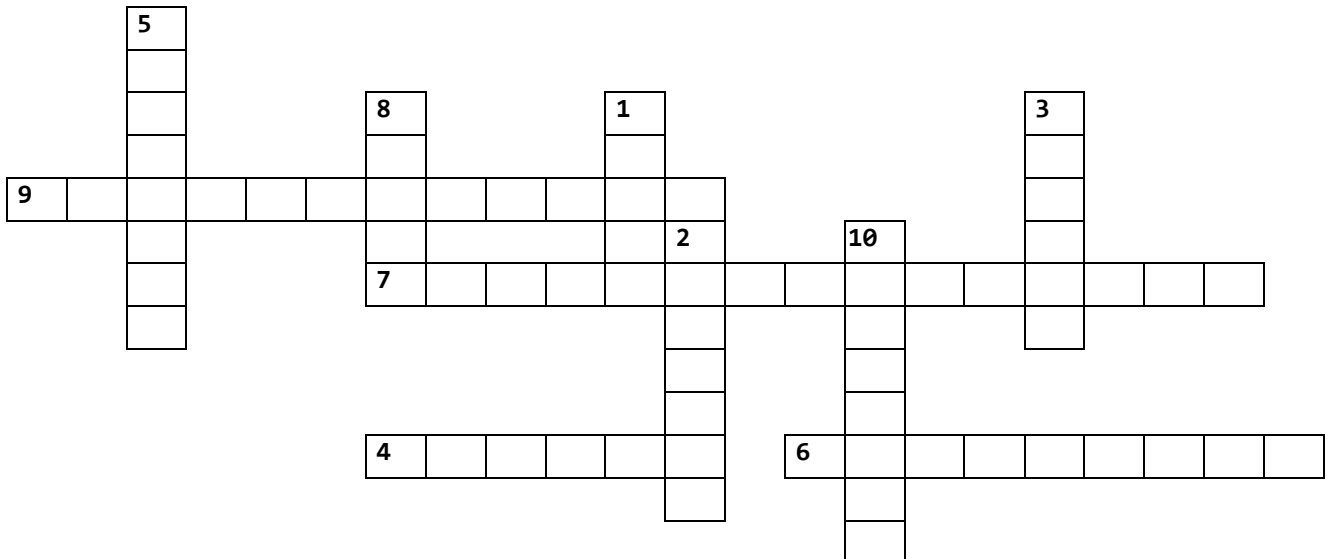
2.

a) Pürüzlü Yüzey Üretiminde yeni normale göre Phong Boyama Modelinin bileşenleri hesaplanırken kullanılan **dot()** ve **max()** fonksiyonlarıyla aynı işi yapacak **myDot()** ve **myMax()** isimli iki fonksiyonun kodunu yazınız. (20P)
float3 vDiffuse = vFinalColor.rgb *
max(0.0f, dot(L, vFinalNormal.xyz)) * 0.8;

b) DirectX uygulamalarında ConstantBuffer ne amaçla kullanılır? (5P)

3. MAYA'da kullanılan toollara/tuşlara ait kısa açıklamaların karşılıklarını ilgili kutucuklara yazınız. (25P)

- 1 Cismi hareket ettirmek için kullanılır.
- 2 Cismi döndürmek için kullanılır.
- 3 Cismi ölçeklemek için kullanılır.
- 4 Birden fazla yüzey ya da kenar seçmek için kullanılır.
- 5 Bir yüzeyin ya da kenarın kopyasını oluşturup çekmek için kullanılır.
- 6 Bir yüzeyi, istediğimiz iki noktadan geçen bir doğruya göre ikiye ayırmak için kullanılır.
- 7 Yüzey sayısını tek seferde artırmak için kullanılır.
- 8 Animasyon geliştirilirken üzerinde değişiklik yapıp setlenen framelere frame denir.
- 9 Setlenen iki frame arasındaki hareketin, aradaki framelere etkisini belirlemek için kullanılır.
- 10 Mayada fiziksel olayların animasyonu yapılırken kullanılır.



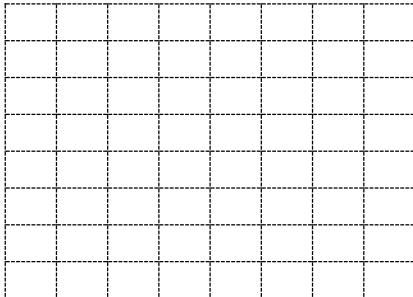
4.

```
void ayarlar(void){
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glOrtho(-3.0, 5.0, -5.0, 3.0, -1.0, 1.0); }
void display(void){
    float radius = 0.5;
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.0, 0.0);
    glBegin(GL_POLYGON);
    glVertex2f(-2,1);
    glVertex2f(0,1);
    glVertex2f(1,0);
    glVertex2f(2,0);
    glVertex2f(2,-1);
    glVertex2f(-2,-1);
    glEnd();
    glColor3f(0.0, 1.0, 0.0);
    glBegin(GL_POLYGON);
    for (float i = 180; i < 360; i++)
        glVertex2f(radius*sin(i) - 1.5 , radius*cos(i) - 1);
    glEnd();
    glBegin(GL_POLYGON);
    for (float i = 180; i < 360; i++)
        glVertex2f(radius*sin(i) + 1.5, radius*cos(i) - 1);
    glEnd();
    glFlush();
}
```

a) OpenGL kütüphanesinin diğer grafik programlama kütüphanelerine kıyasla neden daha fazla tercih edildiğini 5 madde ile açıklayınız. (5P)

b) OpenGL kütüphanesinde önemli bir fonksiyon olan gluLookAt fonksiyonun 9 parametresini açıklayınız (5P)

c) Yukarıda ayarlar ve display fonksiyonları verilen OpenGL kodları koşulduğunda alınacak çıktıyı verilen koordinat sistemine göre aşağıdaki kutuya çizin. (15P)



5.a) Aşağıda gerekli kısımları verilen (shaderlar, buffer object, render fonk.) ve diğer kısımları göz ardı edilebilecek WebGL kodu koşulduğunda web dokümanı üzerinde görselleştirilecek çizim nesnesini boş bırakılan koordinat uzayına çizin. (7P)

```
<html>
<head>
<script id="2d-vertex-shader" type="x-shader/x-vertex">
    attribute vec3 a_position;
    void main() {    gl_Position = vec4(a_pos, 1); }
</script>
<script id="2d-fragment-shader" type="x-shader/x-fragment">
    void main() { gl_FragColor = vec4(1.0, 0.0, 0.0, 1.0); }
</script>
</head>
<body>
<script type="text/javascript">
    ...
    gl.bufferData(
        gl.ARRAY_BUFFER,
        new Float32Array([
            1.0,-1.0,-1.0,1.0,-1.0,-1.0,1.0,-1.0,1.0,0.0,0.0,0.0
        ]),
        gl.STATIC_DRAW );
    ...
    function render() {
        window.requestAnimationFrame(render, canvas);
        ...
        PositionLocation = gl.getAttribLocation(program, "a_pos");
        gl.enableVertexAttribArray(positionLocation);
        gl.vertexAttribPointer(positionLocation,2,gl.FLOAT,false,0,0);
        gl.drawArrays(gl.TRIANGLES, 0, 6);
    }
</script>
</body>
</html>
```

b) Aşağıdaki ifadeleri Doğru veya Yanlış (D/Y) olarak değerlendiriniz. (10P)

- ☐ WebGL 2D veya 3D grafiklerinin görselleştirilmesine olanak sağlayan bir Java ve GLSL temelli bir API'dir.
- ☐ HTML5 ile birlikte herhangi bir güncel tarayıcıda çalışabilir ancak yerel (istemci) kaynaklarının (CPU, GPU vs.) kullanımına izin vermez.
- ☐ Three.js'de **var geometry=new THREE.BoxGeometry(uz_x, uz_y, uz_z)** metodu bir Küre (Sphere) WebGL çizim nesnesinin oluşturulmasını sağlar.
- ☐ Birbirinden bağımsız her bir fragmentin ortama göre renk vb. değerleri vertex shaderlarla hesaplanır.
- ☐ HTML5 destekli web tarayıcıları üzerinde çalışması nedeniyle işletim ve pencere sistemi seviyesinde platform bağımsızlık sağlamaz. Çalışması için Adobe Flash Player gibi gerekli eklentilerin kurulmasına ihtiyaç duyar.

c) WebGL'in DirectX ve OpenGL API'lerinden en temel farkı nedir? Three.js kütüphanesine WebGL uygulamaları geliştirmek için neden ihtiyaç duyarsınız? (Birer cümle ile açıklayınız.) (8P)