

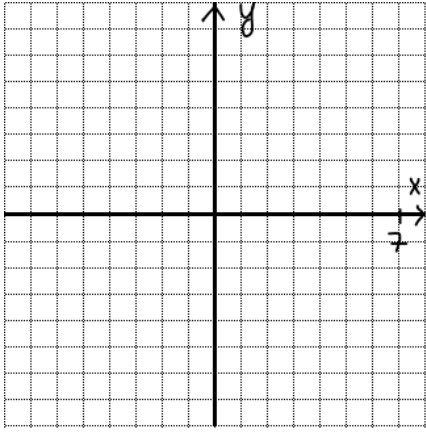
- i. Birden fazla yüzey ya da kenar seçmek için kullanılır _____
- ii. Bir yüzeyin ya da kenarın kopyasını oluşturup çekmek için kullanılır _____
- iii. Bir yüzeyi, istediğimiz iki noktadan geçen bir doğruya göre ikiye ayırmak için kullanılır _____
- iv. Animasyon geliştirilirken üzerinde değişiklik yapıp setlenen framelere _____ frame denir.
- v. Mayada fiziksel olayların animasyonu yapılırken kullanılır _____

	3.b) Herhangi bir küpün seçtiğiniz bir yüzüne yalnız move işlemi uyguladığınızda oluşacak şekli sola, sırasıyla extrude + move uyguladığınızda oluşacak şekli sağa anlaşılar bir şekilde çiziniz. (5P)	
	3.c) Herhangi bir küpün seçtiğiniz bir yüzüne yalnız scale işlemi uyguladığınızda oluşacak şekli sola, sırasıyla extrude + scale uyguladığınızda oluşacak şekli sağa anlaşılar bir şekilde çiziniz. (5P)	
	3.d) Herhangi bir küpün seçtiğiniz bir yüzüne scale + rotate işlemi uyguladığınızda oluşacak şekli sola, sırasıyla extrude + scale + rotate uyguladığınızda oluşacak şekli sağa anlaşılar bir şekilde çiziniz. (5P)	

4.a) Aşağıda gerekli kısımları verilen OpenGL kodu koşulduğunda oluşacak grafik çizim nesnesini boş bırakılan koordinat uzayına çiziniz (koordinat uzayında verilen birimlere ve koddaki köşe konumlarına göre tam ve doğru konumlarda, boyutta ve şekilde olacak şekilde çiziniz, diğer durumlarda puan verilmeyecektir). (8P)

4.b) Aşağıda gerekli bazı kısımları verilen *Three.js* uygulaması web tarayıcısı üzerinde çalıştığında *THREE.BoxGeometry()* ile oluşturulan küp nesnesi üzerinde nasıl bir animasyon gerçekleştirilir. Tüm animasyonu doğru bir şekilde özetleyecek netlikte kısaca açıklayınız. (Örn. <ekranda 180° bir açıya kadar dönen bir küp şeklinde animasyon gerçekleştir> şeklinde) (9P)

```
glBegin(GL_POLYGON);
    glVertex2f(0.0, 0.0);
    glVertex2f(0.0, 3.0);
    glVertex2f(4.0, 3.0);
    glVertex2f(5.0, 1.5);
    glVertex2f(4.0, 0.0);
glEnd();
glFlush();
```



```
// -- kamerayı konumlandır
camera = new
THREE.PerspectiveCamera(70,width/height,1,1000 );
camera.position.z = 200;

// -- sahne nesnesi oluştur
scene = new THREE.Scene();

// bir küp nesnesi ve malzeme kaplaması oluştur
var geometry = new THREE.BoxGeometry(28,28,28);
var material = new THREE.MeshNormalMaterial( { overdraw:
0.5 } );

// -- geometri nesnesi ve malzeme kaplamasını birleştir,
sahneye // -- yerleştir
mesh = new THREE.Mesh( geometry, material );
scene.add(mesh);
...

function animate() {
    requestAnimationFrame( animate );

    if(mesh.scale.x < 2){
        mesh.scale.x += 0.01;
        mesh.scale.y += 0.01;
    }
    else {
        mesh.scale.x = 0.5;
        mesh.scale.y = 0.5;
    }
    renderer.render( scene, camera );
}
```

4.c) Aşağıdaki ifadeleri doğru veya yanlış (D/Y) olarak değerlendiriniz. (8P)

- #__# OpenGL, 3D nesneleri tanımlamak için yüksek-seviyede komutlar içermez. Bunun yerine; nokta, doğru ve poligon gibi alt-seviye geometrik primitif nesneleri içerir.
- #__# OpenGL 'in donanım-bağımsız olmasının bir nedeni, pencere işlemlerini (ekranda bir pencere oluşturmak gibi) yapan ya da kullanıcıdan girdi alan herhangi bir komutunun bulunmaması, bu işlemler için GLUT gibi hazır kütüphanelerin kullanılmasıdır.
- #__# OpenGL masaüstü bilgisayar merkezli bir grafik uygulama programlama arayüzüdür (Direct3D gibi). OpenGL ES ise mobil cihazlar, video oyun konsolları gibi gömülü sistemler için geliştirilmiş bir 2B/3B uygulama geliştirme arayüzü ve grafik işleme dilidir.
- #__# WebGL uygulamaları 3D grafikler oluşturmak için <canvas> elementi, Javascript betik dilini ve GLSL shading dilini kullanmaktadır.