

به نام خدا

پروژه گرافیک کامپیوتری

تیر ماه 1399

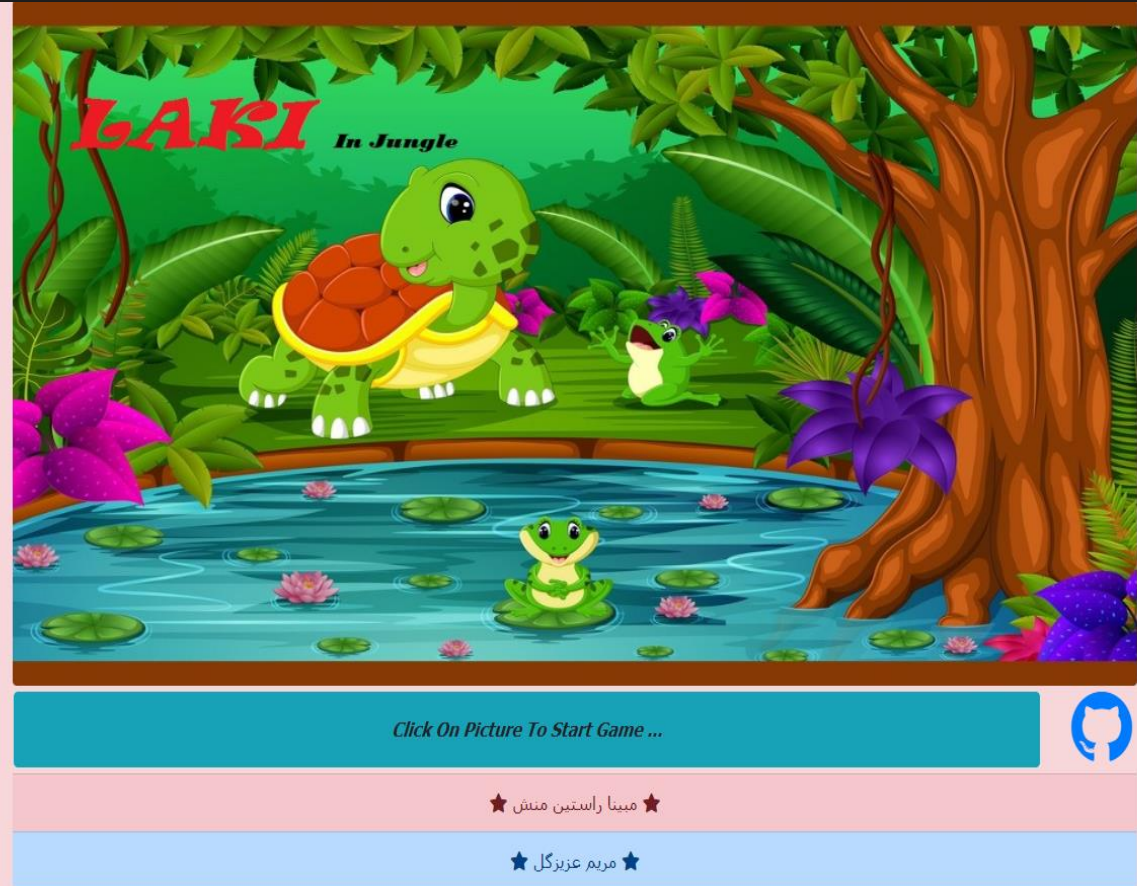
LAKI In Jungle

Mobina Rastinmanesh & Maryam Azizgol



Start page

- ✓ برای طراحی صفحه شروع از HTML, CSS, Bootstrap استفاده شده است.
- ✓ شما با کلیک روی آیکون گیت هاب به صفحه پروژه ما در گیت هاب هدایت میشوید.
- ✓ با کلیک بر روی عکس به بازی وارد میشوید...



Number of donuts eaten 0 Time 54

Open Controls

♥ 100 %

```
<h3 class="d-inline">Number of donuts eaten <span class="badge badge-warning badge-pill " id="message">0</span></h3>
<h3 class="d-inline ">Time <span class="badge badge-danger badge-pill " id="time">0</span></h3>
<div class="float-right rounded" style="width: 100px ; height: 50px ; font-size: larger ;background-color: palevioletred" >

  <i class="fas fa-heart"></i>
  <span id="live" style="font-size: larger">100</span>
  <span style="font-size: larger">%</span>
</div>
```

این بخش برای مشخص کردن تعداد دونات خورده شده و زمان باقی مانده و مقدار جانی که لاک پشت دارد است .
زمان 60 ثانیه است و جان لاک پشت برای شروع 100% است در هنگام برخورد باهر درخت 25% از جان او کم شده و صدایی شنیده میشود و درخت پاک میشود ...
لاک پشت در واقع 4 جان دارد و وقتی جان آن تمام شود میبازید و به صفحه game over هدایت میشوید.
و اگر 60 ثانیه زمان شما تمام شود modal به شما نمایش داده میشود که تعداد دونات های خورده شده را میبینید و میتوانید با again مجددا بازی کنید.

time is over

10 🍌

again

Modal

```
<div id="myModal" class="modal">
  <div class="modal-dialog modal-lg modal-dialog-centered modal-dialog-scrollable">
    <div class="modal-content px-5" style="height:400px; ">

      <!-- Modal Header -->
      <div class="modal-header text-danger">
        <h4 class="modal-title"> time is over </h4>
      </div>

      <!-- Modal body -->
      <div class="modal-body ">

        <span id="eaten" class="text-success"></span>
        <i class="fas fa-thumbs-up"></i>
      </div>

      <!-- Modal footer -->
      <div class="modal-footer">
        <a href="project.html" class="btn btn-success">again</a>
      </div>
    </div>
  </div>
</div>
```

timer

```
var seconds = 60;
var el = document.getElementById('time');

function incrementSeconds() {
    if(seconds) {
        seconds -= 1;
        el.innerText = seconds;
    }
    if(seconds==0){
        el.innerText="finish";
        var modal = document.getElementById("myModal");
        document.getElementById('eaten').innerHTML = tedad;

        modal.style.display = "block";
    }
}

var cancel = setInterval(incrementSeconds, 1000);
```

پس از اتمام زمان (0)
در کادر مربوطه finish نوشته
میشود و modal نمایش داده
میشود...

دونات ها هاي چرخنده ☺

```
/ *****  
var pinkdonut='texture/yami.jfif';  
var browndonut='texture/brown.jpg';  
var colorfulldonut='texture/colorfull.jpg';  
var ndonut='texture/bahbah.jpg';  
var bluedonut='texture/blue.jfif';  
var reddonut='texture/red.jfif';
```

ما در اینجا 6 نوع دونات با رنگ و طمع مختلف داریم ☺
که هر کدام texture مخصوص خود را دارند .
هر کدام به طور رندوم در یک قسمت از زمین قرار گرفته اند..
و همه آنها به متغیر donut اضافه شده اند..
به طور مثال اینجا کد موقعیت یابی یک مدل از دونات هارا میبینیم.

```
for (var j = 0; j < 50; j++) {  
    var r=torus1.clone();  
    r.position.x = Math.random() * 2000 - 1000;  
    r.position.y = -44;  
    r.position.z=Math.random() * 2000 - 1000;  
    donut.add(r);  
}
```


دونات ها هاي چرخنده ☺

```
donut = new THREE.Object3D();
```

```
var textureTorus1 = new THREE.TextureLoader().load(pinkdonut);  
var textureTorus2 = new THREE.TextureLoader().load(browndonut);  
var textureTorus3 = new THREE.TextureLoader().load(colorfulldonut);  
var textureTorus4 = new THREE.TextureLoader().load(bluedonut);  
var textureTorus5 = new THREE.TextureLoader().load(ndonut);  
var textureTorus6 = new THREE.TextureLoader().load(reddonut);  
var Torusgeometry = new THREE.TorusGeometry(6, 3, 15, 20 );
```

```
var TorusboxMaterial1 = new THREE.MeshPhongMaterial({shininess: 100, map: textureTorus1});  
var torus1 = new THREE.Mesh( Torusgeometry, TorusboxMaterial1 );  
var TorusboxMaterial2 = new THREE.MeshPhongMaterial({shininess: 100, map: textureTorus2});  
var torus2 = new THREE.Mesh( Torusgeometry, TorusboxMaterial2 );  
var TorusboxMaterial3 = new THREE.MeshPhongMaterial({shininess: 100, map: textureTorus3});  
var torus3 = new THREE.Mesh( Torusgeometry, TorusboxMaterial3 );  
var TorusboxMaterial4= new THREE.MeshPhongMaterial({shininess: 100, map: textureTorus4});  
var torus4 = new THREE.Mesh( Torusgeometry, TorusboxMaterial4 );  
var TorusboxMaterial5= new THREE.MeshPhongMaterial({shininess: 100, map: textureTorus5});  
var torus5 = new THREE.Mesh( Torusgeometry, TorusboxMaterial5 );  
var TorusboxMaterial6= new THREE.MeshPhongMaterial({shininess: 100, map: textureTorus6});  
var torus6 = new THREE.Mesh( Torusgeometry, TorusboxMaterial6 );
```

```
torus1.scale.set(0.7,0.7,0.7);  
torus2.scale.set(0.7,0.7,0.7);  
torus3.scale.set(0.7,0.7,0.7);  
torus4.scale.set(0.7,0.7,0.7);  
torus5.scale.set(0.7,0.7,0.7);  
torus6.scale.set(0.7,0.7,0.7);
```

کد ساخت دونات هايمان

چرخش تمام دونات ها

```
function rotat() {  
    var t = (Date.now() / 700);  
    for (var j = 0; j < 300; j++) {  
        if(donut.children[j]!=null) {  
            donut.children[j].rotation.x = t * 2;  
            donut.children[j].rotation.y = t / 2;  
            donut.children[j].rotation.z = t;  
        }  
    }  
    controls.update();  
}
```

برخورد با دونات ها (خوردن دونات ها)

```
function checkdonut() {  
    for(var i=0; i < 300; i++) {  
        var donutObj = donut.children[i];  
        var carObj1 = cubecar;  
        var firstBB1 = new THREE.Box3().setFromObject(donutObj);  
        var secondBB1 = new THREE.Box3().setFromObject(carObj1);  
        var collision1 = firstBB1.intersectsBox(secondBB1);  
        if(collision1) {  
            donut.remove(donut.children[i]);  
            console.log('COLLISION donut');  
            tedad +=1 ;  
            appendText(tedad);  
        }  
    }  
}  
  
var loopdonut = new THREE.PhysicsLoop(30);  
loopdonut.add(checkdonut);  
loopdonut.start();
```

در اینجا برای تشخیص برخورد تمام دونات ها چک میشود...
اگر برخوردی صورت گرفته باشد :
اولا دونات حذف میشود (خورده میشود)
دوما در کنسول collision donut چاپ میشود.
و سومایکی به تعداد دونات ها خورده شده اضافه میشود

درخت ها

```
var objectLoader = new THREE.ObjectLoader();

objectLoader.load("model/tree/tree-un.json", function (obj) {
    var cubeGeometry = new THREE.CubeGeometry(7,50,10);
    var wireMaterial = new THREE.MeshBasicMaterial({color: 0x000000, wireframe: true, visible: false});
    cubetree = new THREE.Mesh(cubeGeometry, wireMaterial);
    obj.add(cubetree);
    obj.scale.set(2,2,2);

    for (var i = 0; i < 150 ; i++) {
        obj2=obj.clone();

        obj2.position.x = Math.random() * 2000 - 1000 ;
        obj2.position.y = -65;
        obj2.position.z = Math.random() * 2000 - 1000;
        obj2.rotateY(-Math.random() * 150 - 27);

        trees.add(obj2);
    }
});

scene.add(trees);
```

درخت ها json هستند که از سایت clara.io دانلود شده اند...

برای تشخیص برخورد دور درخت ها یک مکعبی در نظر گرفته شده است که دیده نمیشود و در واقع برخورد با آنها صورت میگیرد...

150 تا درخت داریم که موقعیت آنها در صحنه به صورت رندوم است. و همین طور چرخش رندوم دارند که همه شان به یک جهت نباشند... و همه شان به trees اضافه شدند و در نهایت به صحنه add شدند...

برخورد با درخت ها

```
var joon=100 ;
function checktree() {
    for(var i=0; i < 150; i++) {
        var treeObj = trees.children[i];
        var carObj1 = cubecar;
        var firstBB1 = new THREE.Box3().setFromObject(treeObj);
        var secondBB1 = new THREE.Box3().setFromObject(carObj1);
        var collision1 = firstBB1.intersectsBox(secondBB1);
        if(collision1) {

            var listener = new THREE.AudioListener();
            camera.add(listener);
            audioLoader = new THREE.AudioLoader();
            music2 = new THREE.PositionalAudio(listener);
            audioLoader.load('music/1.mp3', function (buffer) {
                music2.setBuffer(buffer);
                music2.setRefDistance(15);
                music2.play();
            });
            trees.add(music2);

            joon=joon-25;
            trees.remove(trees.children[i]);
            console.log('COLLISION tree');
            document.getElementById('live').innerHTML = joon;
            if(joon ==0){
                go2NewUrl('game_over.html', 0.2);
            }
        }
    }
}

var looptree = new THREEEx.PhysicsLoop(30);
looptree.add(checktree);
looptree.start();
```

در صورت برخورد با درخت :

1. صدایی پخش میشود
2. از جون لاک پشت 25% کم میشود
3. درخت پاک میشود
4. در کنسول collision tree چاپ میشود
5. مقدار جان در بالای صفحه آپدیت میشود
6. اگر مقدار جان او 0 شود به صفحه game over منتقل میشود.

پرنده ها (Tween)



سه پرنده داریم که json هستند ...

و در صحنه حرکت میکنند ...

با استفاده از Tween .

و مقصد و جهت انها رندوم است .

با کلید ⬆ حرکت انها مجددا شروع میشود .

```
function onDocumentKeyUp(event) {  
    if (event.keyCode == 38) {  
        tweenbird();  
        tween1.start();  
        tween2.start();  
        tween3.start();  
    }  
}
```

```

objectLoader2 = new THREE.ObjectLoader();
objectLoader2.load("model/bird/bird.json", function (obj) {

    obj.scale.set(0.05,0.05,0.05);
    bird1=obj.clone();
    bird2=obj.clone();
    bird3=obj.clone();

    bird1.position.z = Math.random() * 1000 - 500;
    bird1.position.y = -8;
    bird1.position.x =Math.random() * 1000 - 500;
    bird1.rotateY(-Math.random() * 150 - 27);
    scene.add(bird1);

```

بخش اول کد برای مثال برای پرنده اول
که مکان اول او هم رندوم است

```

// Randomize target
function tweenbird() {

    target1.x = Math.random() * 1400 - 700;
    target1.y = -35;
    target1.z = Math.random() * 1200 - 900;
    bird1.rotateY(-Math.random() * 150 - 27);

    target2.x = Math.random() * 1400 - 700;
    target2.y = -35;
    target2.z = Math.random() * 1200 - 900;
    bird2.rotateY(-Math.random() * 150 - 27);

    target3.x = Math.random() * 1400 - 700;
    target3.y = -35;
    target3.z = Math.random() * 1200 - 900;
    bird3.rotateY(-Math.random() * 150 - 27);

}

```

و 3 مدل Tween برای 3 پرنده

```

target1 = {x: 50 , y: -30, z: 50};
target2 = {x: Math.random() * 1000 - 500 , y: -30, z: Math.random() * 1000 - 500};
target3 = {x: Math.random() * 1000 - 500 , y: -30, z: Math.random() * 1000 - 500};

tween1 = new TWEEN.Tween(bird1.position).to(target1, 7000);
tween1.delay(500);
tween1.easing(TWEEN.Easing.Elastic.In);

tween2 = new TWEEN.Tween(bird2.position).to(target2, 7000);
tween2.delay(500);
tween2.easing(TWEEN.Easing.Exponential.InOut);

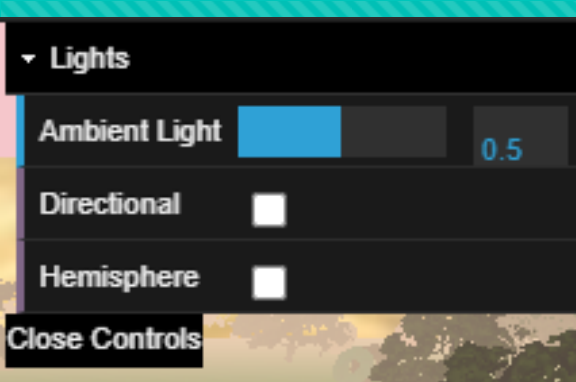
tween3 = new TWEEN.Tween(bird3.position).to(target3, 7000);
tween3.delay(500);
tween3.easing(TWEEN.Easing.Quartic.InOut);

```

مقصد و جهت رندوم

```
scene.fog = new THREE.Fog(0xefd1b5, 0.1, 1200);
```





کنترل کردن نور صحنه

```
function createPanel() {  
  
  var gui = new dat.GUI({width: -300});  
  var lightsFolder = gui.addFolder('Lights');  
  
  props = {  
    'Ambient Light': 0.5,  
    'Directional Light': false,  
    'Hemisphere Light': false,  
  };  
  
  lightsFolder.add(props, 'Ambient Light', 0, 1).step(0.01).onChange(function (value) {  
    ambientLight.intensity = value;  
  });  
  lightsFolder.add(props, 'Directional Light').onChange(btn2);  
  lightsFolder.add(props, 'Hemisphere Light').onChange(btn3);  
  lightsFolder.open();  
}
```

```
function btn2() {  
  toggleVisible(directionalLight);  
}  
  
function btn3() {  
  toggleVisible(hemisphereLight);  
}  
  
function toggleVisible(p) {  
  p.visible = !p.visible;  
}
```


نور ها

```
ambientLight = new THREE.AmbientLight(0xdbdb87, 0.5);
scene.add(ambientLight);

directionalLight = new THREE.DirectionalLight(0xaaaaaa, 1);
scene.add(directionalLight);
directionalLight.visible = false;

hemisphereLight = new THREE.HemisphereLight(0xffffff1, 0x777788, 0.90);
scene.add(hemisphereLight);
hemisphereLight.visible = false;
```

چمن

```
var texture = new THREE.TextureLoader().load('texture/grass_6.jpg');
texture.repeat.set(100, 100);
texture.wrapS= THREE.RepeatWrapping;
texture.wrapT= THREE.RepeatWrapping;
texture.magFilter= THREE.NearestFilter;
texture.minFilter= THREE.LinearMipMapLinearFilter;

var surface = new THREE.PlaneGeometry(2000, 2000);
var surfaceMaterial = new THREE.MeshBasicMaterial({color: 0xcccccc, side: THREE.DoubleSide, map: texture});
var surfaceMesh = new THREE.Mesh(surface, surfaceMaterial);
surfaceMesh.rotateX(Math.PI / 2);
surfaceMesh.position.y = -50;
scene.add(surfaceMesh);
```



Sky box

```
var reflectionCube = new THREE.CubeTextureLoader()  
    .setPath('texture/cube/skybox/')  
    .load(['px.jpg', 'nx.jpg', 'py.jpg', 'ny.jpg', 'pz.jpg', 'nz.jpg']);  
reflectionCube.format = THREE.RGBFormat;  
scene = new THREE.Scene();  
scene.background = reflectionCube;  
scene.fog = new THREE.Fog(0xefd1b5, 0.1, 1200);
```



nx



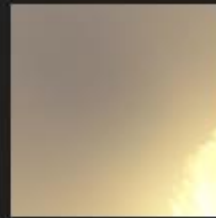
ny



nz



px



py



pz

موزیک حین بازی

```
var listener = new THREE.AudioListener();
camera.add(listener);
audioLoader = new THREE.AudioLoader();
music = new THREE.PositionalAudio(listener);
audioLoader.load('music/oggy.mp3', function (buffer) {
    music.setBuffer(buffer);
    music.setRefDistance(50);
    music.play();
    music.loop = true;
});

scene.add(music);
```


Camera & renderer & scene & control

```
scene = new THREE.Scene();

camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 0.1, 10000);
renderer = new THREE.WebGLRenderer({antialias: true});
renderer.setClearColor(new THREE.Color(0x000000));
renderer.setSize(window.innerWidth, window.innerHeight);

camera.position.x=10;
camera.position.y =-10;
camera.position.z =-10;
camera.lookAt(scene.position);

renderer.setPixelRatio(window.devicePixelRatio);
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);

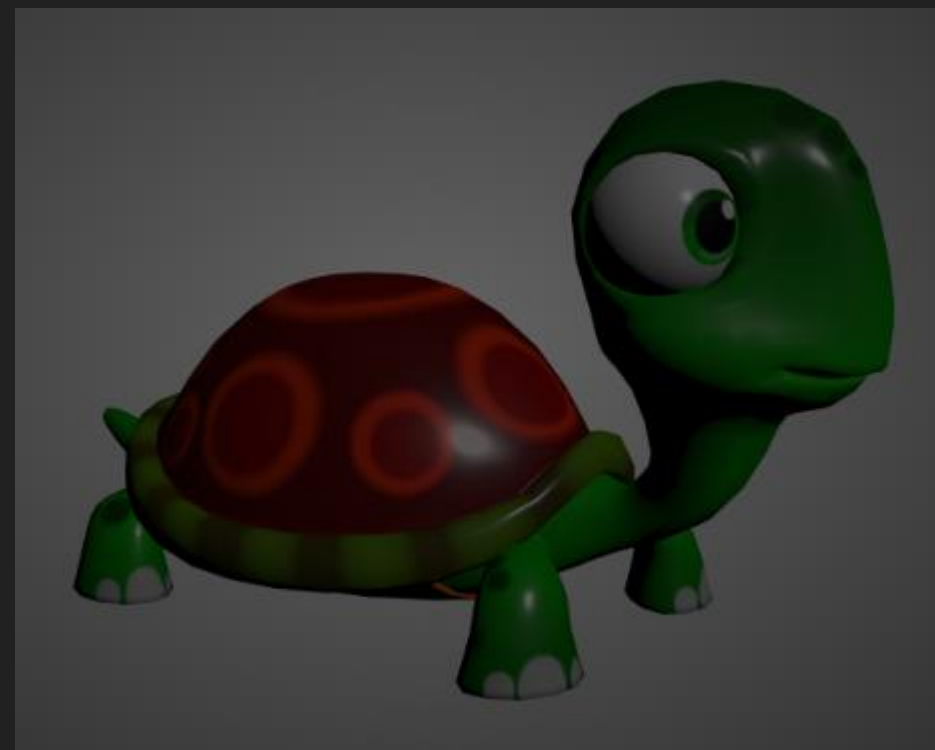
window.addEventListener('keyup', onDocumentKeyUp, false);
window.addEventListener('resize', onWindowResize, false);
```

```
controls = new THREE.OrbitControls( camera, renderer.domElement );
```

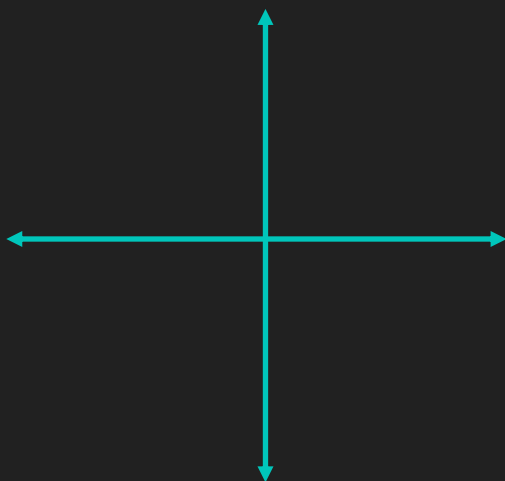
لاک پشت

```
objectLoader2 = new THREE.ObjectLoader();
objectLoader2.load("model/LAK/lak.json", function (obj) {
    var cubeGeometry = new THREE.CubeGeometry(0.5,0.5,1);
    var wireMaterial = new THREE.MeshBasicMaterial({color: 0xffffff, wireframe: true, visible: false});
    cubelak = new THREE.Mesh(cubeGeometry, wireMaterial);
    obj.position.z = 0;
    obj.position.y = -50;
    obj.position.x = 10;
    obj.scale.set(30,30,30);
    lak=obj;
    lak.add(cubelak)
    scene.add(lak);
});
```

ما دور لاک پشت هم یک مکعب
فرضی در نظر گرفته ایم که برخورد
با این مکعب صورت میگیرد...



حرکت لاک پشت



```
var delta = clock.getDelta();
var moveDistance = 100 * delta;
var rotateAngle = Math.PI / 2 * delta;

if (keyboard.pressed(String.fromCharCode(38))) {
    lak.translateZ(-moveDistance);
}
if (keyboard.pressed(String.fromCharCode(40))) {
    lak.translateZ(moveDistance);
}
if (keyboard.pressed(String.fromCharCode(37))) {
    lak.rotateOnAxis(new THREE.Vector3(0, 1, 0), rotateAngle);
}
if (keyboard.pressed(String.fromCharCode(39))) {
    lak.rotateOnAxis(new THREE.Vector3(0, 1, 0), -rotateAngle);
}

if (lak) {
    var relativeCameraOffset = new THREE.Vector3(0, 2, 5);
    var cameraOffset = relativeCameraOffset.applyMatrix4(lak.matrixWorld);
    camera.position.x = cameraOffset.x;
    camera.position.y = cameraOffset.y;
    camera.position.z = cameraOffset.z;
    camera.lookAt(lak.position);
}
```

Game over page

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>game over</title>
  <script type="text/javascript" src="lib/bootstrap.min.js"></script>
  <link rel="stylesheet" href="lib/bootstrap.min.css">
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></script>
</head>
<body >
  <embed src="music/taboot2.mp3" autostart="true" loop="true" hidden="true" ></embed>

  <div >
    <video controls width="100%" height="650" loop autoplay>
      <source src="movie/gameover.mp4" type="video/mp4" >
    </video>
    <a href="start.html" class="btn btn-danger font-weight-bold font-italic d-block py-4">start again</a>
  </div>
</body>
</html>
```

موزیک

ویدیو تابوت 😊

start رفتن به صفحه

تمرین

با زدن دکمه تمرین ها در صفحه نخست به صفحه تمرینات ما که شامل مبحث سایه و کنترل است هدایت میشوید...

Click On Picture To Start Game ...



★ مبینا راستین منش ★

★ مریم عزیزگل ★

تمرین ها

✓ برنامه نویسان :

- مبینا راستین منش 971113017
- مریم عزیزگل 971113069

✓ اساتید :

- استاد سید امیر هادی مینوفام
- استاد فراز سامعی

مبینا راستین منش	مریم عزیز گل
%۵۰	۵۰%