CSE378: Computer Graphics
Assignment 1 – Catch the fruit

Name: Ahmed Mohamed Ahmed Aly

Id: 18P9313, UEL

GitHub Link:
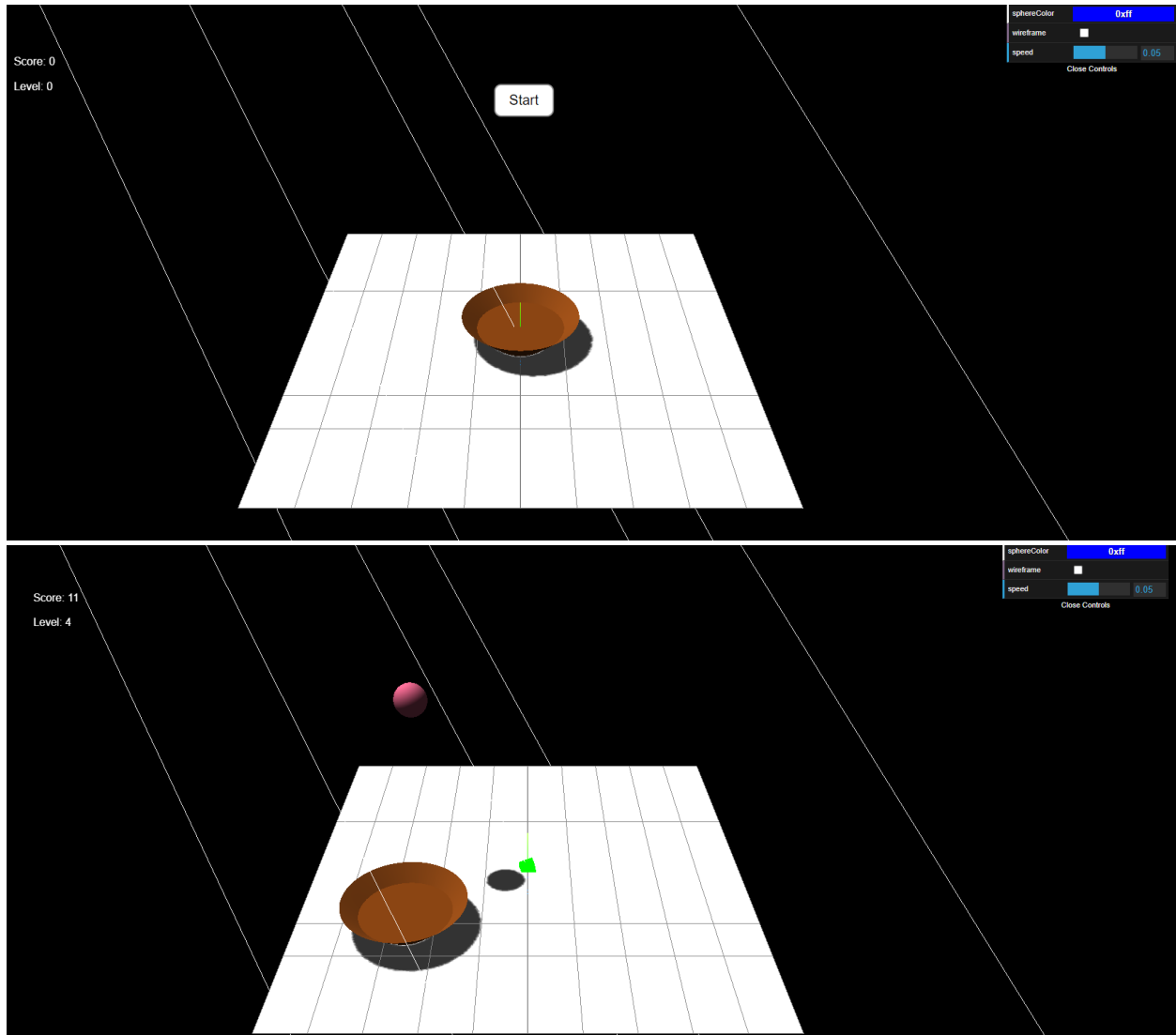
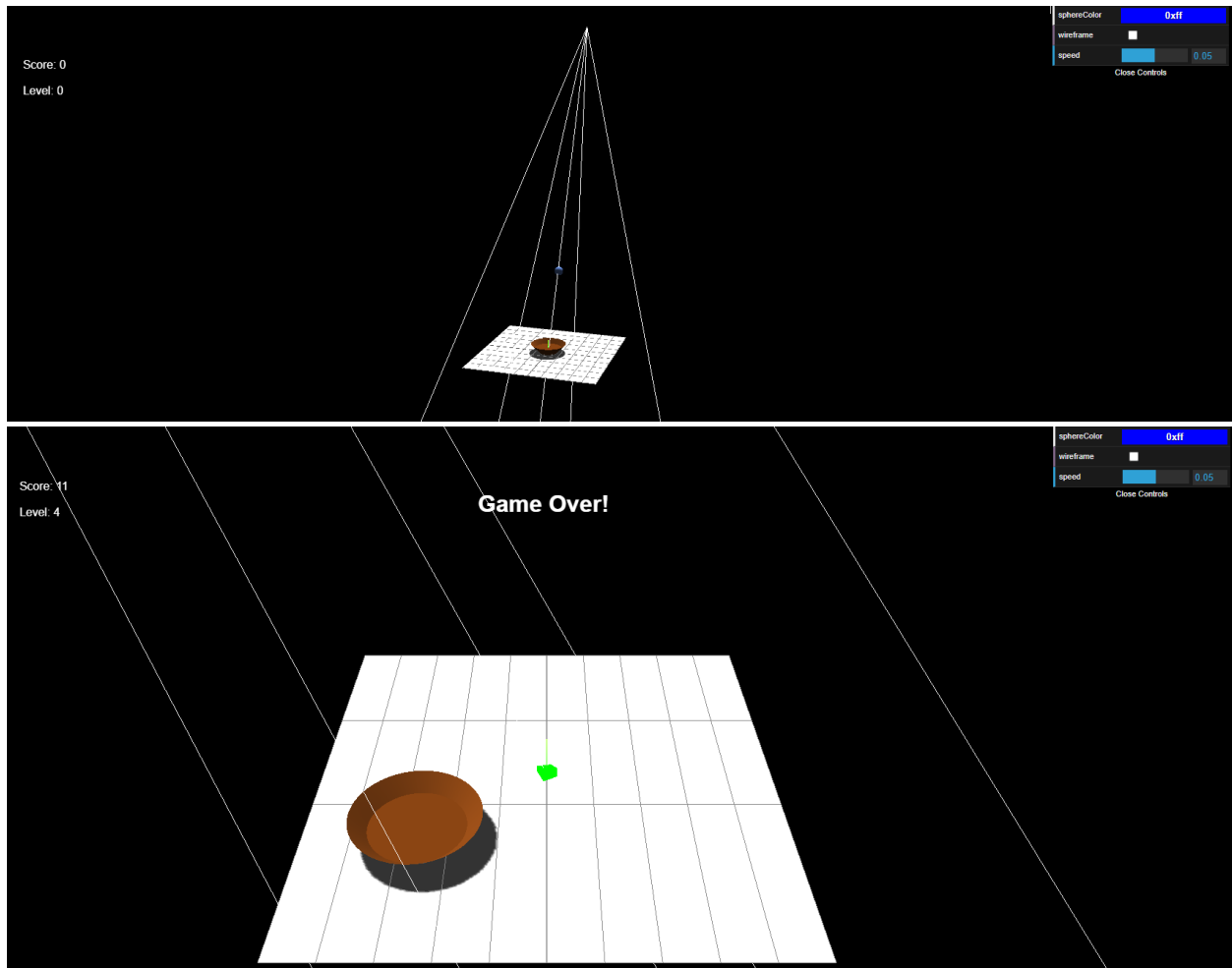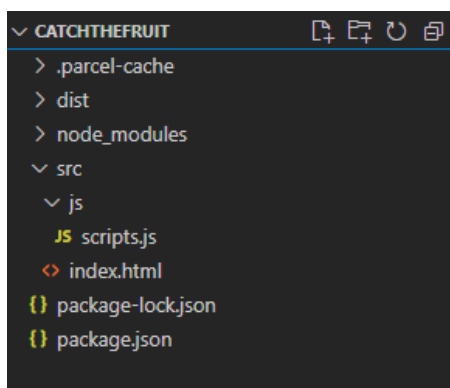github/AhmedElshobaky/catchthefruit

Demo:

Video

Snippets:

Files hierarchy

Breaking down the code

Fruit collector project code consists of:

- A 3D scene using the three.js library and renders it using a WebGL renderer. The scene includes a box, a cylinder that represents a fruit basket, and a sphere that represents the fruit falling form the sky.
- An animation function that gets added to the rendering using

```
renderer.setAnimationLoop( animate );
```

- The animation function updates the positioning of the falling sphere and the positioning of the dish.

```
function animate() {
   updateBasketPosition();
   if(isPlaying){
       updateSpherePosition();
    }
    renderer.render( scene, camera );
}
```

- Update basket positioning function checks on global moveX and moveY which have values of 0.02 and -0.02 (step amount) only if the button is held on, otherwise both are zeros

```
// onkeydown event listener
function onKeyDown(event) {
  if (event.keyCode == 37) {
    moveX = -0.2;
  }
  if (event.keyCode == 39) {
    moveX = 0.2;
  }
  if (event.keyCode == 38) {
    moveZ = -0.2;
  }
  if (event.keyCode == 40) {
    moveZ = 0.2;
    }
}
```

```javascript
// onkeyreleast event listener
function onKeyUp(event) {
  if (event.keyCode == 37) {
    moveX = 0;
  }
  if (event.keyCode == 39) {
    moveX = 0;
  }
    if (event.keyCode == 38) {
    moveZ = 0;
    }
    if (event.keyCode == 40) {
    moveZ = 0;
    }
}
```

```javascript
function updateBasketPosition() {
  positionX += moveX;
  positionZ += moveZ;
  // if the basket is on any edge it stays in the plane constraints
  // collision of basket with the plane edges
  if (positionX < -10) {
    positionX = -10;
  }else if(positionX > 10) {
    positionX = 10;
  }
  if (positionZ < -10) {
    positionZ = -10;
  }else if(positionZ > 10) {
    positionZ = 10;
  }
  basket.position.x = positionX;
  basket.position.z = positionZ;
}
```

- The basket is made up of a cylinder and a circle that represents the bottom of the basket. The cylinder has an inner mesh with a different material to create the illusion of depth.

```javascript
// create fruit basket geometry
const basketGeometry = new THREE.CylinderGeometry( 4, 2, 2, 100,1, true );
const basketMaterial = new THREE.MeshStandardMaterial( { color: 0x8b4513,
wireframe: false } );
const basketInnerMaterial = new THREE.MeshStandardMaterial( { color:
0x8b4513, wireframe: false, side: THREE.BackSide} );

const basketBottom = new THREE.CircleGeometry( 3, 100 );
```

```
const basketBottomMaterial = new THREE.MeshBasicMaterial( { color:
0x8b4513, wireframe: false } );
const basketBottomMesh = new THREE.Mesh( basketBottom, basketBottomMaterial
);
basketBottomMesh.rotation.x = -0.5 * Math.PI;

const basketInner = new THREE.Mesh( basketGeometry, basketInnerMaterial );
const basket = new THREE.Mesh( basketGeometry, basketMaterial );
basket.add( basketInner );
basket.add( basketBottomMesh );
basket.position.set( 0, 1, 0 );
scene.add( basket );
basket.castShadow = true;
```

- An Orbit control is added to the scene to allow the user to rotate and zoom the camera.

```
const orbit = new OrbitControls( camera, renderer.domElement );
```

- A lighting is added to the scene, including an ambient light and a spot light, the spot light is set to cast shadows.

```
const ambiantLight = new THREE.AmbientLight( 0x333333, 1);
scene.add( ambiantLight );

const spotLight = new THREE.SpotLight( 0xffffff, 1);
scene.add( spotLight );
spotLight.position.set( -30, 70, 0 );
spotLight.castShadow = true;
spotLight.angle = 0.3;
```

- dat.gui is added to give the user the ability control the sphere color, speed, wirefram using gui
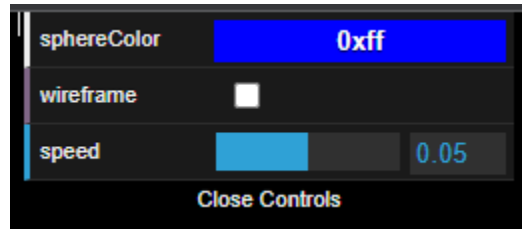
```
const gui = new dat.GUI();

const options = {
sphereColor: 0x0000ff,
wireframe: false,
speed: 0.05
};
gui.addColor( options, 'sphereColor' ).onChange(function(e){
sphere.material.color.set(e);
})

gui.add( options, 'wireframe' ).onChange(function(e){
sphere.material.wireframe = e;
})
gui.add( options, 'speed', 0, 0.1, 0.001 );
```

- Finally, updating the sphere function, it updates the sphere with its new y coordinate and checks if collision occurred with the basket if the y value is at the level of the basket, then it is either gets added to the basket and increment the score or game over if the basket is not under the sphere.

```javascript
var counter = 0;
function updateSpherePosition() {
    sphere.position.y -= options.speed;
    if (sphere.position.y <1 ) {
        // if sphere is in the basket counter increases by 1
        if (sphere.position.x > basket.position.x - 4 &&
            sphere.position.x < basket.position.x + 4 &&
            sphere.position.z > basket.position.z - 4 &&
            sphere.position.z < basket.position.z + 4
            ) {
            scene.remove(sphere);
            sphere.position.x = Math.random() * 20 - 10;
            sphere.position.z = Math.random() * 20 - 10;
            sphere.position.y = 20;
            sphere.material.color.set(getRandomColor());
            scene.add(sphere);
            counter++;
            // increase speed every three fruits collected
            if (counter % 3 == 0) {
                options.speed += 0.01;
            }

        }else{
            // gameover scenario
            isPlaying = false;
            const msg = document.getElementById("gameOver");
            msg.style.display = "block";
            scene.remove(sphere);
        }]
        // update html with new score and level
        document.getElementById("counter").innerHTML = '<p>Score:
'+counter+ ' <br><br> Level: ' + ((options.speed.toFixed(2).slice(2)) - 4)+
'</p>';
    }
}
```

Demo video:

https://github.com/AhmedElshobaky/catchthefruit

References

Three.js. three.js docs. (n.d.). Retrieved January 6, 2023, from https://threejs.org/docs/