

コンピュータグラフィックス 中間・期末レポート

情報ネットワーク学科

学籍番号 1832053 番

草野史子

1. 目的

WebGL を用いて 3D コンテンツを作成することによって、これまでの講義で学習した 3D デザインの基礎を理解することを目的とする。

2. 装置

3D コンテンツの作成に用いた環境は以下のとおりである。

表 1 作成環境

CPU	Intel Core i7-7500
OS	Windows 10 Home
メモリ	8GB
テキストエディタ	Atom
ブラウザ	Internet Explorer

3. 結果

中間・期末の作品について、以下の URL にて示す。なお、ソースコードは 5.

付録を参照されたい。

①中間作品

<https://fumi477.github.io/webgl.html>

②期末作品

<https://fumi477.github.io/webgl2.html>

4. 参考文献

(ア) Ics.media, "Three.js 入門サイト", <<https://ics.media/tutorial-three/>>, 2020/8/1 参照.

(イ) MITSUE-LINKS, "はじめての「three.js」で 3D 雪だるまを作ろう!", <https://www.mitsue.co.jp/knowledge/blog/frontend/201912/20_0000.html>, 2020/8/1 参照.

5. 付録

各作品のソースコードを以下に示す.

表 2 中間作品 「webgl.html」

```
<!DOCTYPE
PE
html>

<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>中間</title>

  <!-- three.js を読み込む(CDN) -->
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/three.js/99/three.min
.js"></script>
  <script src="OrbitControls.js"></script>

  <script>
```

```

var init = function() {

    var width = 960,
        height = 540;

    // レンダラーを作成
    var renderer = new THREE.WebGLRenderer({
        canvas: document.querySelector('#myCanvas')
    });
    renderer.setSize(width, height);

    // シーンを作成
    const scene = new THREE.Scene();

    // カメラを作成
    const camera = new THREE.PerspectiveCamera(45, width /
height);
    camera.position.set(0, 0, 1000);

    //カメラのコントローラを作成
    const controls = new THREE.OrbitControls(camera);

    //グループを作成
    const group = new THREE.Group();
    scene.add(group);

    //正方形を作成
    const geometry = new THREE.BoxGeometry(60, 60, 60);

    //テクスチャー作成
    var texture1 = new THREE.TextureLoader().load("images/スマイルフ
ェイス 2.png");
    var texture1mat = new
THREE.MeshBasicMaterial({transparent:true});
    texture1mat.map = texture1;

```

```

        var texture2 = new THREE.TextureLoader().load("images/ウインクフ
エイス 2.png");

        var texture2mat = new
THREE.MeshBasicMaterial({transparent:true});
        texture2mat.map = texture2;

//for 文を用いての正方形作成
j = -350
for(var i = 0; i < 8; i++){
    var k = -420;
    for(let l = 0; l <8; l++){
        if(i==4 && l==7){
            const mesh = new THREE.Mesh(geometry, texture2mat);
            mesh.position.x = k;
            mesh.position.y = j;
            group.add(mesh);
        }else{
            const mesh = new THREE.Mesh(geometry, texture1mat);
            mesh.position.x = k;
            mesh.position.y = j;
            group.add(mesh);
        }
        k +=80;
    }
    j += 80;
}

// 平行光源
var directionalLight1 = new THREE.DirectionalLight(0xffffff);
directionalLight1.position.set(1, 1, 1);
// シーンに追加
scene.add(directionalLight1);

update();
// 初回実行
function update() {

```

```
        renderer.render(scene, camera); //レンダリング
        requestAnimationFrame(update);
    }

}

window.addEventListener('load', init);

</script>
</head>
<body>
    <canvas id = "myCanvas"></canvas>
</body>
</html>
```

表 3 期末作品 「weblg2.html」

```
<!DOCTYPE
PE
html>

<html lang="ja">
<head>
  <meta charset="UTF-8">
  <title>最終課題</title>

  <!-- three.js を読み込む(CDN) -->
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/three.js/99/three.min
.js"></script>

  <script>
    var init = function() {

      var width = 960,
          height = 540;

      // レンダーを作成
      const renderer = new THREE.WebGLRenderer({
        canvas: document.querySelector('#myCanvas')
      });
      renderer.setSize(width, height);

      // シーンを作成
      const scene = new THREE.Scene();

      // カメラを作成
      const camera = new THREE.PerspectiveCamera(35, width/height );

      // グループを作成
      const group = new THREE.Group();
```

```

scene.add(group);

// 図形共通テクスチャー
const texture = new
THREE.TextureLoader().load('images/yuki2.png');
var texturemat = new
THREE.MeshBasicMaterial({transparent:true});
texturemat.map = texture;
const geometry = new THREE.SphereGeometry(5, 32, 32);
const material = texturemat

//雪をつくる
for (let i = 0; i < 1000; i++) {
  const mesh = new THREE.Mesh(geometry, material);
  mesh.position.x = (Math.random() - 0.5) * 650;
  mesh.position.y = (Math.random() - 0.5) * 1000;
  mesh.position.z = (Math.random() - 0.5) * 650;
  // グループに格納する
  group.add(mesh);
}

//地面を作成
const floor = new THREE.Mesh(
  new THREE.PlaneGeometry(700, 700),
  texturemat
);
floor.position.y = -60
floor.rotation.x = -Math.PI / 2;
scene.add(floor);

//雪だるまを作成
const head = new THREE.Mesh(

```



```

    new THREE.SphereGeometry(40, 40, 20),
    texturemat
  );
  head.position.x = 0;
  head.position.y = 0;
  head.position.z = 0;

  const body = new THREE.Mesh(
    new THREE.SphereGeometry(50, 50, 50),
    texturemat
  );
  body.position.x = 0;
  body.position.y = -60;
  body.position.z = 0;

  const hat = new THREE.Mesh(
    new THREE.CylinderGeometry(25, 25, 40, 30),
    new THREE.MeshLambertMaterial({ color: 0xffffff })
  );
  hat.position.x = 0;
  hat.position.y = 50;
  hat.position.z = 0;

  const hatline = new THREE.Mesh(
    new THREE.CylinderGeometry(26, 25, 20, 30),
    new THREE.MeshLambertMaterial({ color: 0xe60033 })
  );
  hatline.position.x = 0;
  hatline.position.y = 35;
  hatline.position.z = 0;

  const hat3 = new THREE.Mesh(
    new THREE.CylinderGeometry(40, 40, 5, 30),
    new THREE.MeshLambertMaterial({ color: 0xffffff })
  );
  hat3.position.x = 0;

```

```

    hat3.position.y = 32;
    hat3.position.z = 0;

    const righteye = new THREE.Mesh(
        new THREE.SphereGeometry(5, 25, 40),
        new THREE.MeshToonMaterial({ color: 0x000000})
    );
    righteye.position.x = 15;
    righteye.position.y = 18;
    righteye.position.z = 30;

    const lefteye = new THREE.Mesh(
        new THREE.SphereGeometry(5, 10, 40),
        new THREE.MeshToonMaterial({ color: 0x000000})
    );
    lefteye.position.x = -16;
    lefteye.position.y = 18;
    lefteye.position.z = 33;

    const nose = new THREE.Mesh(
        new THREE.SphereGeometry(5, 30, 20),
        new THREE.MeshLambertMaterial({ color: 0xed9121 })
    );
    nose.position.x = 3;
    nose.position.y = 10;
    nose.position.z = 35;

    //雪だるまのパーツをグループ化
    const yukidaruma = new THREE.Group();

    yukidaruma.add(hat, hatline, hat3, righteye, lefteye, nose, body, head);
    scene.add(yukidaruma);

    // 光源(環境光)
    const ambientLight = new THREE.AmbientLight(0xffffff, 1);

```

```

        scene.add(ambientLight);

        //半球光源
        const light = new THREE.HemisphereLight(0x192f60, 0xffffafa,
1.0);
        scene.add(light);

        update();
        // 初回実行
        function update() {

            // 角度に応じてカメラの位置を設定
            camera.position.x = 600 * Math.sin(Date.now() / 3000);
            camera.position.y = 400;
            camera.position.z = 600 * Math.cos(Date.now() / 3000);
            // 見つめる方向
            camera.lookAt(new THREE.Vector3(0, 25, -15));

            group.position.y -= 0.5;
            renderer.render(scene, camera); //レンダリング
            requestAnimationFrame(update);
        }

    }

    window.addEventListener('load', init);

</script>
</head>
<body>
    <canvas id = "myCanvas"></canvas>
</body>
</html>

```