

## &lt;本日の課題&gt;

## 課題 10-1 複数のシェーダを使う

サンプルコード **sample10-1** を実行し、テクスチャマッピング用のシェーダセットと光の表現用のシェーダセットを同時に利用できることを確認しなさい。

**sample10-1 (抜粋)**

## ○シェーダの設定

```

<!-- テクスチャマッピング用のシェーダセット -->
<script id="vertexShader1" type="x-shader/x-vertex">
    ...
</script>
<script id="fragmentShader1" type="x-shader/x-fragment">
    ...
</script>

<!-- 光の表現用のシェーダセット -->
<script id="vertexShader2" type="x-shader/x-vertex">
    ...
</script>
<script id="fragmentShader2" type="x-shader/x-fragment">
    ...
</script>

```

シェーダその1の定義

シェーダその2の定義

## ○シェーダの初期化

```

// メッシュデータを生成
const plane = GL.Mesh.plane({ coords: true }); // テクスチャ座標付きの平面
const cube = GL.Mesh.cube({ normals: true }); // 法線ベクトル付きの立方体

// テクスチャ画像を読み込む
const texture = GL.Texture.fromURL('grass.jpg');

// シェーダプログラムを読み込んで初期化
const shader1 = new GL.Shader('vertexShader1', 'fragmentShader1');
const shader2 = new GL.Shader('vertexShader2', 'fragmentShader2');

```

別のシェーダとして初期化

## ○シェーダの呼び出し

```

// --- 平面の描画 (テクスチャマッピング) ---
gl.pushMatrix();
gl.rotate(90, 1, 0, 0); // 平面を x 軸回転して水平にする
gl.scale(10, 10, 10); // 平面を拡大
texture.bind(0); // テクスチャをバインド
shader1.uniforms({
    texture: 0 // テクスチャユニット 0 をシェーダに渡す
});
shader1.draw(plane); // 平面を描画
gl.popMatrix();

// --- 立方体の描画 (光の表現) ---
gl.pushMatrix();
gl.translate(0, 1, 0); // 立方体を上方向に移動
shader2.draw(cube); // 立方体を描画
gl.popMatrix();

```

シェーダその1の呼び出し (テクスチャ)

シェーダその2の呼び出し (光の表現)

## ○課題 10-2 CG の制作

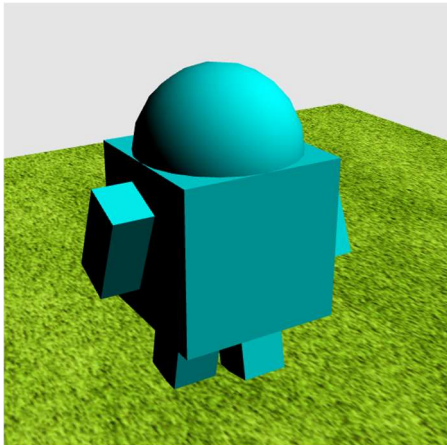
これまで学んだ CG の技術に応用して、WebGL の lightgl.js による CG 作品を制作しなさい。テーマは自由とします。ただし、次の要件を満たすこと。

また、作品のテーマと以下の要件を満たしていることを説明する文章もあわせて提出してください。

### <要件>

- 1) 複数のオブジェクト： 最低3つ以上のオブジェクトを使うこと。
- 2) 光の表現： シェーダによる光の表現を使うこと。(最低1か所以上)
- 3) テクスチャマッピング： テクスチャを適用したオブジェクトを使うこと。(最低1か所以上)
- 4) 視点の操作： マウスなどにより視点を変更できるようにすること。
- 5) アニメーション： オブジェクトに動きが伴うアニメーションを含むこと。(最低1か所以上)
- 6) オブジェクトの操作： キーボードの入力により、オブジェクトを操作できること。(最低1つ以上)

作成例) 手足が動く操作可能なロボット



以上