

# まとめと総合演習

2015 年度 情報可視化論

陰山 聡

2015.07.14

# 全体のまとめ

# シェーダとシェーディング言語： GLSL

OpenGL シェーディング言語 (OpenGL SL, GLSL) 4.0

GPU を使うための言語

CG ソースコード = OpenGL ソースコード  
+ GLSL(フラグメントシェーダ) ソースコード  
+ GLSL(頂点シェーダ) ソースコード

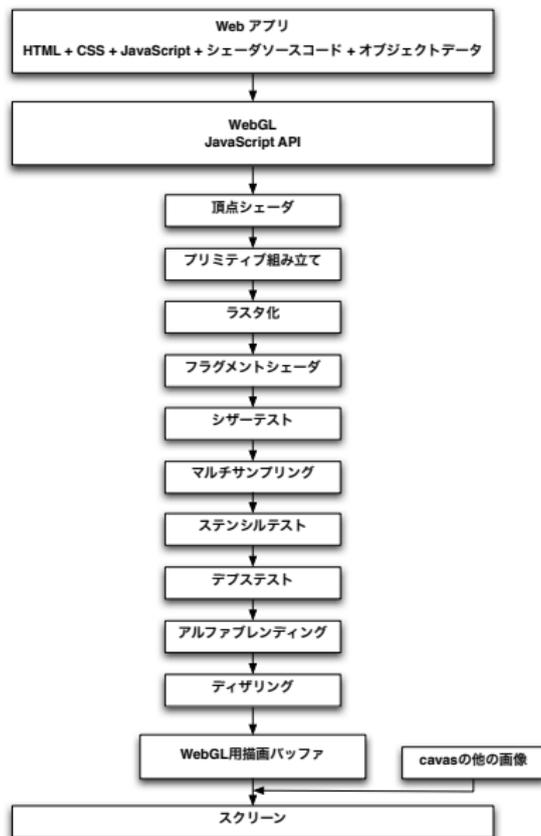
## WebGL とは

WebGL = シェーダを使い、HTML5 の canvas に、JavaScript で 3D CG を書くための API

## WebGL の特徴

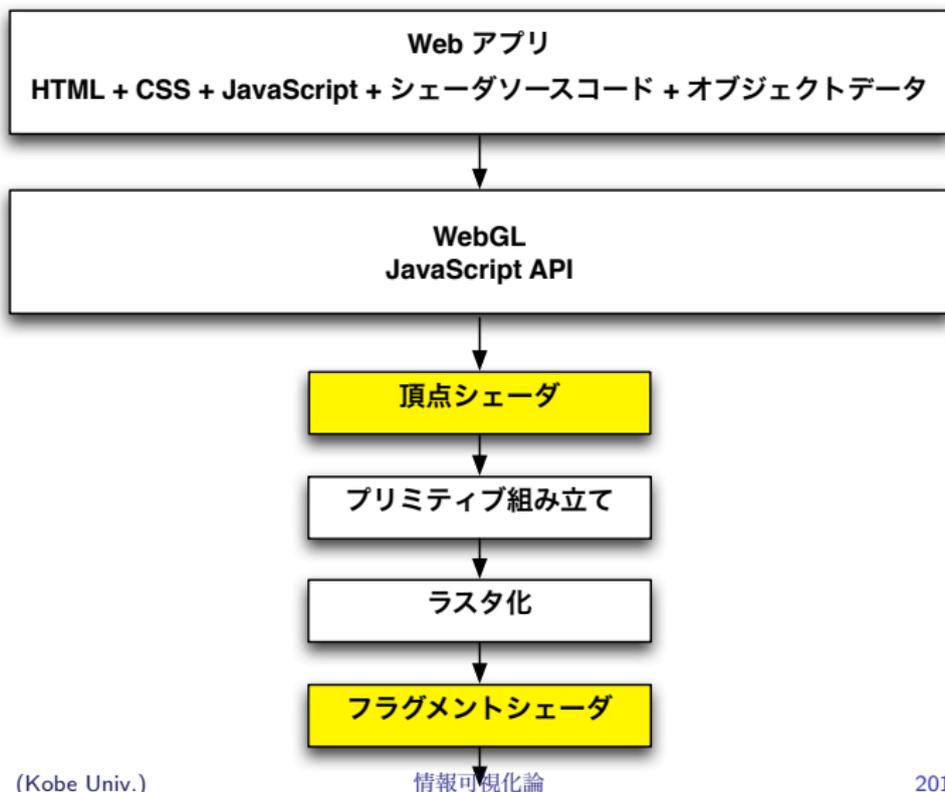
- クロスプラットフォーム
- オープンスタンダード
- Web で GPU を使ったレンダリングが可能
- 開発・利用が容易：プラグイン不要
- ソースコードが見える
- グラフィックス（OpenGL）と UI（ウィンドウ管理やイベント処理）の分離が明白

# WebGL のグラフィックスパイプライン



## シェーダ

頂点シェーダ（バーテックスシェーダ）とフラグメントシェーダ



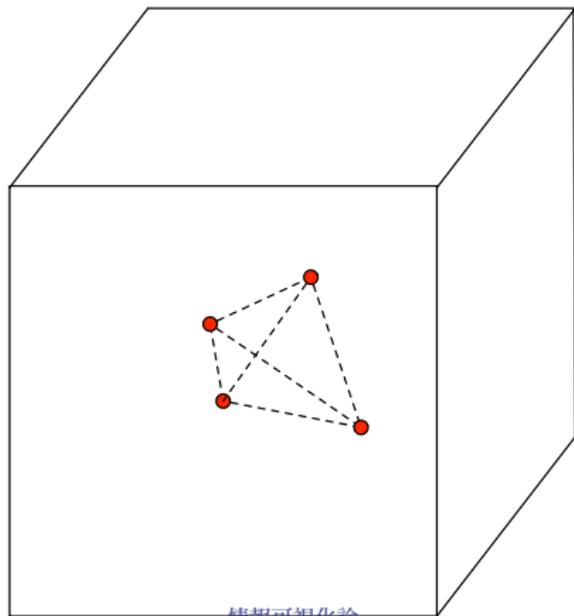
# WebGL アプリケーション

Web アプリ = HTML + CSS + JavaScript

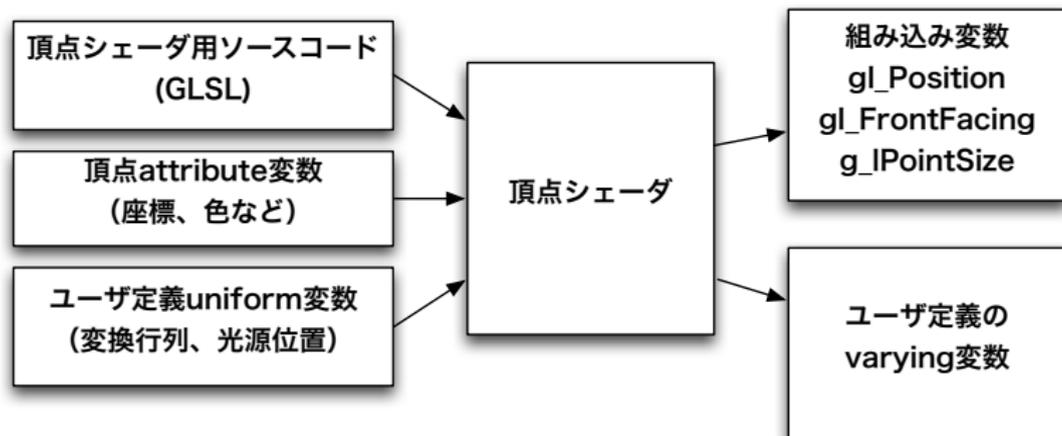
WebGL アプリ = HTML + CSS + JavaScript + シェーダ言語 (OpenGL SL)

## 頂点シェーダ

- 各頂点に対して処理を行う
- 並列処理
- $n$  個の頂点があれば  $n$  個の頂点シェーダプロセッサを同時に実行させる

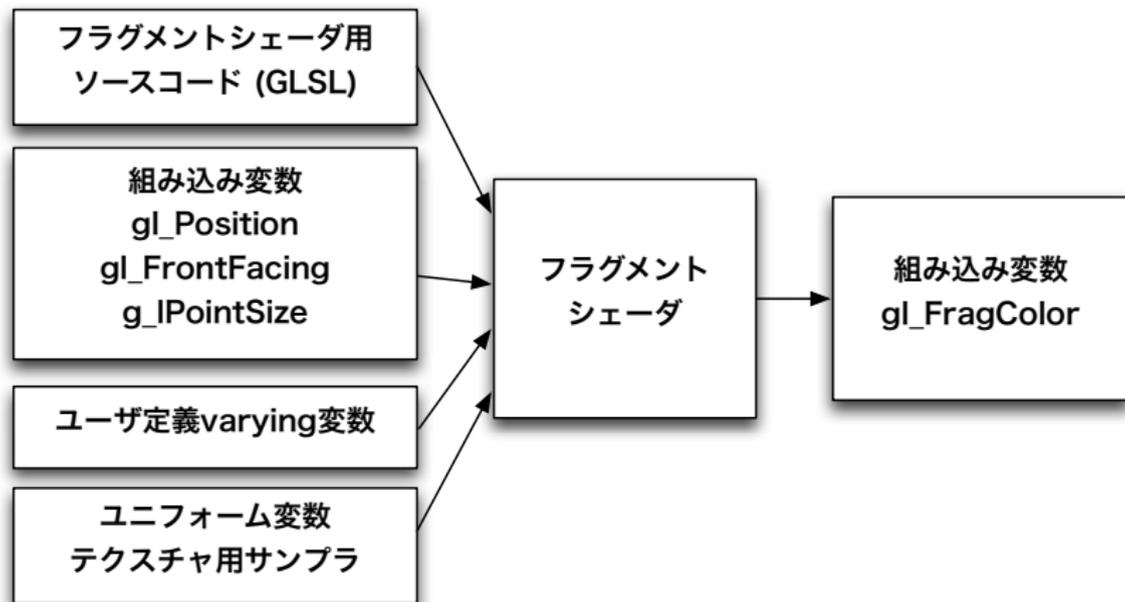


# 頂点シェーダの入出力データ



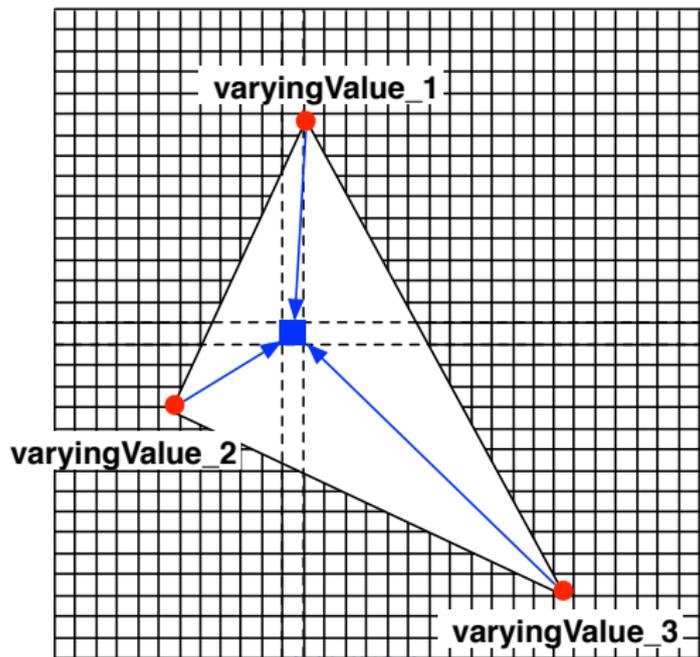
## フラグメントシェーダの入出力

全てのフラグメントで並列処理。シェーディング言語でプログラム。



## varying 変数の補間

- 頂点シェーダ からフラグメントシェーダへは varying 変数を通じて情報を送る。
- 各フラグメントの varying 変数値は自動的に線形補間される。



# WebGL での 3D 描画プログラム

- 頂点データの生成と転送
  - 法線データの生成と転送
  - テクスチャデータの生成と転送
  - 物体の座標変換 (4 行 4 列)
  - 材質 (反射) 特性設定
  - 照明設定
  - 射影変換 (4 行 4 列)
- … 面倒

# 3D CG ライブラリ

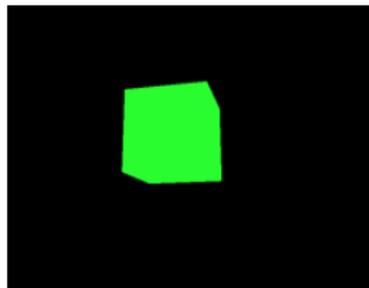
# WebGL 用 JavaScript ライブラリ

## WebGL のラッパ

- Three.js (<http://threejs.org>)
- Away3D TypeScript
- Babylon.js

## Three.js sample

回転する直方体



サンプルコード

`three_js_sample_cube.js`

必要なライブラリ：`three.min.js`

```
<html>
<head>
<title>My first Three.js app</title>
<style>canvas { width: 100%; height: 100% }</style>
</head>

<body>
<script src="js/three.min.js"></script>
<script>
  var scene = new THREE.Scene();
  var camera = new THREE.PerspectiveCamera(75, window.innerWidth/wi
  var renderer = new THREE.WebGLRenderer();
  renderer.setSize(window.innerWidth, window.innerHeight);
  document.body.appendChild(renderer.domElement);
  var geometry = new THREE.CubeGeometry(1,1,1);
  var material = new THREE.MeshBasicMaterial({color: 0x00ff00});
  var cube = new THREE.Mesh(geometry, material);
```

```
scene.add(cube);  
camera.position.z = 5;  
  
var render = function () {  
    requestAnimationFrame(render);  
    cube.rotation.x += 0.01;  
    cube.rotation.y += 0.01;  
    renderer.render(scene, camera);  
};
```

```
render();  
</script>
```

```
</body>  
</html>
```

# 演習

# レポート課題

- 照明とテクスチャマッピング、アニメーションを使った WebGL プログラムを作れ。
- Three.js などのライブラリは使わないこと。
- 提出はメールで。添付ファイルは3つ
- **アーカイブはしないでください。**
  1. レポート PDF ファイル： **ファイル名： report\_05.pdf**
  2. 作成した HTML ファイル： **ファイル名： report\_05.html**
    - ファイル名中のアンダースコア ( \_ ) は半角
    - ファイル拡張子は html とし、htm としない
  3. テクスチャデータ
- 提出先： kageyama.lecture@gmail.com
- レポートの PDF ファイルは1 ページ（表紙はつけない）。
- メールタイトル： **情報可視化論 レポート 5**
- レポートには以下を記述すること
  - 学籍番号と氏名
  - ウェブ公開時の名前（本名またはイニシャル）
  - 何を描いたかという記述
  - 描いた図形のキャプチャ図
- 締め切り： 7/22（水） 23:59