

電子教材開発者および対話型3Dコンテンツ 開発者向け講習会 ～WebGLを用いた3Dコンテンツ開発～

附属図書館付設教材開発センター
岡田 義広
(感性融合デザインセンター兼任)
(システム情報科学府担当教員)

平成25年6月27日(木)

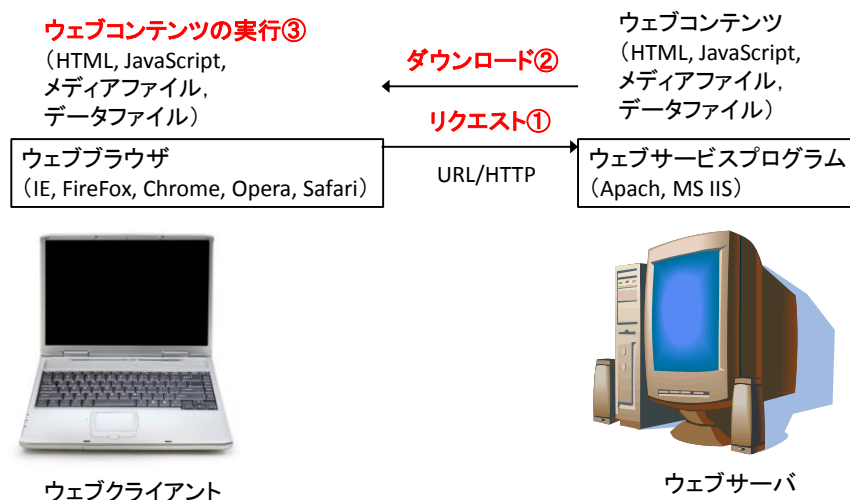
内容

- 附属図書館付設教材開発センターのご紹介
<http://www.icer.kyushu-u.ac.jp/>
- 双方向型のインタラクティブな電子教材の開発について
http://www.icer.kyushu-u.ac.jp/howto_make
- WebGL 講習
 - WebGLとは？, ウェブコンテンツの実行
 - 対応ブラウザ

WebGLを用いたウェブ3Dコンテンツ - WebGLとは? -

- **HTML5 (Hyper Text Markup Language 5)**
 - ウェブコンテンツの新しい仕様
 - マルチメディアや3次元CG(WebGL)のネイティブサポート
 - Canvas要素(2D/3D画像処理)
- **JavaScript**
 - ウェブブラウザで稼働するスクリプト言語(インタプリタ言語)
- **OpenGL**
 - 3次元グラフィックスアプリケーションを開発するための業界標準の関数ライブラリ
 - Xbox等ゲーム開発向けにはMicrosoft Direct3D
- **OpenGL ES(Embedded Systems)**
 - OpenGLのサブセットで携帯端末等の組み込み用
- **WebGL (OpenGL 2.0 or OpenGL ES 2.0のプラットフォーム)**
 - JavaScript用の3次元グラフィックス表示のための関数群

WebGLを用いたウェブ3Dコンテンツ - ウェブコンテンツの実行 -



*URL(Uniform Resource Locator), HTTP(Hyper Text Transfer Protocol)

WebGLを用いたウェブ3Dコンテンツ - 対応ブラウザ -

- Windows PC
 - Firefox, Chrome, Opera?
- Mac PC
 - Safari
- Android端末(Nexus7, Galaxy)
 - Firefox, Chrome, Opera?
- iOS 端末(iPad, iPad mini, iPod, iPhone)
 - 未対応

WebGLを用いたウェブ3Dコンテンツ 閲覧システムの紹介

- three.js
 - <http://mrdoob.github.com/three.js/>
 - WebGLを使い易くするJavaScriptライブラリ
 - 多数の使用例コンテンツが提供されている
- ウェブ3Dコンテンツ閲覧システム
 - http://contsrv.i.kyushu-u.ac.jp/web/WebGL/WebGL_Viewer.zip
 - ダウンロード用
 - http://contsrv.i.kyushu-u.ac.jp/web/WebGL/WebGL_Viewer/Smooth_sample.html
 - サンプル
 - 対応する形状データ: objファイルとmtlファイル

ウェブ3Dコンテンツ閲覧システム

WebGL_View/

形状データファイルとマテリアルファイル

beethoven.mtl, beethoven.obj,
cow.mtl, cow.obj,
CTSkeleton-zup-r2.mtl, CTSkeleton-zup-r2.obj,
bunny.mtl, bunny.obj,

ライセンスファイル

LICENSE,

three.jsファイルと変更ファイル

OBJLoader.js,
MTLLoader.js, OBJMTLLoader.js, OBJMTLLoader_Mod.js,
OBJMTLLoader_Flat.js, OBJMTLLoader_Smooth.js,
stats.min.js, three.min.js, Detector.js,
TrackballControls.js, TrackballControls_Mod.js

サンプルhtmlファイル

Smooth_beethoven.html, Smooth_bunny.html, Smooth_CTSkeleton-zup-r2.html, Smooth_cow.html,
Smooth_sample.html,
Flat_bunny.html, Flat_cow.html, Flat_beethoven.html, Flat_CTSkeleton-zup-r2.html,
Flat_sample.html,

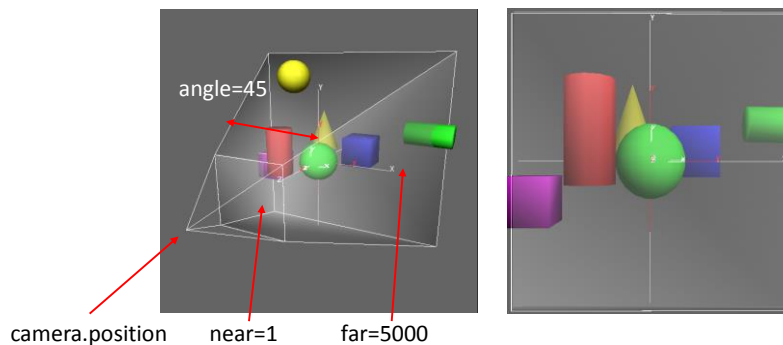
このファイル

list.txt

Perspective : 透視投影

```
71: camera = new THREE.PerspectiveCamera( 45,  
window.innerWidth / window.innerHeight, 1, 5000 );
```

```
72: camera.position.z = 500;
```



Object: 形状データの読み込み

```

115:    // model
116:    var loader = new THREE.OBJMTLLoader();
117:    loader.addEventListener( 'load', function ( event ) {
118:        object = event.content;
119:        object.position.x = 0;
120:        object.position.y = 0;
121:        object.position.z = 0;
122:        object.scale.x = 1.0;
123:        object.scale.y = 1.0;
124:        object.scale.z = 1.0;
125:        object.rotation.x = 0.0;
126:        object.rotation.y = 0.0;
127:        object.rotation.z = 0.0;
128://    object.materials[0].shading = THREE.SmoothShading;
129:        scene.add( object );
130:        objects[nobjects] = object;
131:        nobjects++;
132:    });
133:    loader.load( 'CTSkeleton-zup-r2.obj', 'CTSkeleton-zup-r2.mtl' );

```

アンケート内容

【質問1】

WebGL講習の内容について以下から該当するものの一つの番号をお答えください。

- 1) 内容をよく理解できた。
- 2) 内容をまあまあ理解できた。
- 3) 内容がよく分からなかった。
- 4) 内容が全く分からなかった。

【質問2】

今回の講習内容で不明な点や改善してほしい点、感想などをご記入ください。

【質問3】

その他、教材開発センターで実施を希望する講習内容等がありましたらご記入ください。

```

1:<!DOCTYPE html>
2:<html lang="en">
3:  <head>
4:    <title>three.js webgl - loaders - OBJ MTL loader</title>
5:    <meta charset="UTF-8">
6:    <meta name="viewport" content="width=device-width, user-scalable=no, minimum-scale=
1.0, maximum-scale=1.0">
7:    <style>
8:      body {
9:        font-family: Monospace;
10:       background-color: #000;
11:       color: #fff;
12:       margin: 0px;
13:       overflow: hidden;
14:     }
15:     #info {
16:       color: #fff;
17:       position: absolute;
18:       top: 10px;
19:       width: 100%;
20:       text-align: center;
21:       z-index: 100;
22:       display: block;
23:     }
24:     .a { background: #000; color: #f00; }
25:     .button { background: #999; color: #fff; padding: 0.2em 0.5em; cursor: pointer }
26:     .active { background: #fff; color: #000; }
27:   </style>
28: </head>
29:
30: <body>
31:   <script src="three.min.js"></script>
32:   <script src="TrackballControls_Mod.js"></script>
33:   <script src="MTLLoader.js"></script>
34:   <script src="OBJMTLLoader_Smooth.js"></script>
35:   <script src="Detector.js"></script>
36:   <script src="stats.min.js"></script>
37:   <div id="info">
38:     <a href="http://threejs.org" target="_blank">three.js</a> - WebGL_Viewer -
39:   </p>
40:   <span>Touch operations:</span>
41:   <span id="touch_rotate"><a href="javascript:controls.setState('rotate');">
[Rotate]</a></span>
42:   <span id="touch_zoom"><a href="javascript:controls.setState('zoom');">[ Zoom ]</
a></span>
43:   <span id="touch_pan"><a href="javascript:controls.setState('pan');">[ Pan ]</a>
</span>
44:   <span id="touch_none"><a href="javascript:controls.setState('none');">[ None ]</
a></span>
45:   </div>
46:   <script>
47:     var container, stats;
48:     var camera, controls, scene, renderer;
49:     var mouseX = 0, mouseY = 0;
50:     var mposX = 0, mposY = 0;
51:     var info;
52:     var objects = new Array(200);
53:     var nobjects = 0;
54:     var object, picked_object, directionallight;
55:     var picked_object_material, picked_material;
56:     var windowHalfX = window.innerWidth / 2;
57:     var windowHalfY = window.innerHeight / 2;
58:
59:     var touch_rotate = document.getElementById( "touch_rotate" );
60:     var touch_zoom = document.getElementById( "touch_zoom" );
61:     var touch_pan = document.getElementById( "touch_pan" );
62:     var touch_none = document.getElementById( "touch_none" );
63:

```

```

64:         init();
65:         animate();
66:
67:         function init() {
68:             container = document.createElement( 'div' );
69:             document.body.appendChild( container );
70:
71:             camera = new THREE.PerspectiveCamera( 45, window.innerWidth / window.
innerHeight, 1, 5000 );
72:             camera.position.z = 50;
73:
74:             controls = new THREE.TrackballControls( camera );
75:             controls.rotateSpeed = 1.0;
76:             controls.zoomSpeed = 1.2;
77:             controls.panSpeed = 0.8;
78:             controls.noZoom = false;
79:             controls.noPan = false;
80:             controls.staticMoving = true;
81:             controls.dynamicDampingFactor = 0.3;
82:             controls.keys = [ 65, 83, 68 ];
83:             controls.addEventListener( 'change', render );
84:
85:             info = document.createElement( 'div' );
86:             info.style.position = 'absolute';
87:             info.style.top = '70px';
88:             info.style.width = '100%';
89:             info.style.textAlign = 'center';
90:             info.innerHTML = 'Drag to change the view';
91:             container.appendChild( info );
92:
93:             objectname = document.createElement( 'div' );
94:             objectname.style.position = 'absolute';
95://             objectname.style.top = '0px';
96://             objectname.style.left = '0px';
97://             objectname.style.width = '200px';
98:             objectname.style.backgroundColor = 'black';
99:             objectname.style.fontSize = '24pt';
100:             objectname.innerHTML = 'Object Name';
101:             container.appendChild( objectname );
102:
103:             // scene
104:
105:             scene = new THREE.Scene();
106:             scene.add( camera );
107:
108:             var ambient = new THREE.AmbientLight( 0x101030 );
109:             scene.add( ambient );
110:
111:             directionalLight = new THREE.DirectionalLight( 0xffeedd );
112:             directionalLight.position.set( 0, 0, 1 ).normalize();
113:             scene.add( directionalLight );
114:
115:             // model
116:             var loader = new THREE.OBJMTLLoader();
117:             loader.addEventListener( 'load', function ( event ) {
118:                 object = event.content;
119:                 object.position.x = 0;
120:                 object.position.y = 0;
121:                 object.position.z = 0;
122:                 object.scale.x = 1.0;
123:                 object.scale.y = 1.0;
124:                 object.scale.z = 1.0;
125:                 object.rotation.x = 0.0;
126:                 object.rotation.y = 0.0;
127:                 object.rotation.z = 0.0;
128://             object.materials[0].shading = THREE.SmoothShading;
129:                 scene.add( object );
130:                 objects[nobjects] = object;

```

```

131:         nobjects++;
132:     });
133:     loader.load( 'cow.obj', 'cow.mtl' );
134:     loader.load( 'beethoven.obj', 'beethoven.mtl' );
135:
136:     picked_material = new THREE.MeshPhongMaterial({color: 0xffffffff, specular:
0x888888, shininess:50, ambient: 0xffffffff, emissive: 0x888888 });
137:
138:     renderer = new THREE.WebGLRenderer();
139:     renderer.setSize( window.innerWidth, window.innerHeight );
140:     container.appendChild( renderer.domElement );
141:
142:     // マウスの座標を取得し、-1?+1の範囲に正規化する
143:     document.addEventListener( 'mousemove', onDocumentMouseMove, false );
144:
145:     window.addEventListener( 'resize', onWindowResize, false );
146: }
147:
148: function check_touch_button_state() {
149:     touch_rotate.style.backgroundColor = 'black';
150:     touch_zoom.style.backgroundColor = 'black';
151:     touch_pan.style.backgroundColor = 'black';
152:     touch_none.style.backgroundColor = 'black';
153:     if ( controls.touch_button === -1 ) {
154:         touch_none.style.backgroundColor = 'red';
155:     }
156:     if ( controls.touch_button === 0 ) {
157:         touch_rotate.style.backgroundColor = 'red';
158:     }
159:     if ( controls.touch_button === 1 ) {
160:         touch_zoom.style.backgroundColor = 'red';
161:     }
162:     if ( controls.touch_button === 2 ) {
163:         touch_pan.style.backgroundColor = 'red';
164:     }
165: }
166:
167: function check_picked_object() {
168:
169:     if ( picked_object !== null ) {
170:         picked_object.material = picked_object_material;
171://         picked_object.material.emissive = new THREE.Color(0);
172:         picked_object = null;
173:     }
174:
175:     // マウスが指した座標に存在するオブジェクトを得る
176:     var projector = new THREE.Projector();
177:     var pos = new THREE.Vector3(mouseX, mouseY, -1);
178:     var ray = projector.pickingRay(pos, camera);
179:     var intersects;
180:
181:     for(var i = 0; i < nobjects; i++) {
182:         intersects = ray.intersectObjects(objects[i].children);
183:         // マウスがobjectの上であれば、名前を表示する
184:         if(intersects.length > 0 ) {
185:             picked_object = intersects[0].object;
186://             picked_object.material.emissive = new THREE.Color(0x888888);
187:             picked_object_material = picked_object.material;
188:             picked_object.material = picked_material;
189:             break;
190:         }
191:     }
192: }
193:
194: function check_touch_position() {
195:     if ( controls.touchX === -1 && controls.touchY === -1 ) return;
196:
197:     mposX = controls.touchX;

```



```

198:         mposY = controls.touchY;
199:
200:         mouseX = (controls.touchX / window.innerWidth ) * 2 - 1;
201:         mouseY = (controls.touchY / window.innerHeight) * -2 + 1;
202:
203:         check_touch_button_state();
204:         check_picked_object();
205:     }
206:
207:     function onDocumentMouseMove(e) {
208:         e.preventDefault();
209:
210:         mposX = e.clientX;
211:         mposY = e.clientY;
212:
213:         mouseX = (e.clientX / window.innerWidth ) * 2 - 1;
214:         mouseY = (e.clientY / window.innerHeight) * -2 + 1;
215:
216:         check_touch_button_state();
217:         check_picked_object();
218:     }
219:
220:     function onWindowResize() {
221:
222:         windowHalfX = window.innerWidth / 2;
223:         windowHalfY = window.innerHeight / 2;
224:
225:         camera.aspect = window.innerWidth / window.innerHeight;
226:         camera.updateProjectionMatrix();
227:
228:         controls.handleResize();
229:
230:         renderer.setSize( window.innerWidth, window.innerHeight );
231:     }
232:
233:     function animate() {
234:         requestAnimationFrame( animate );
235:
236:         render();
237:         controls.update();
238:     }
239:
240:     function render() {
241:         check_touch_position();
242:
243:         if ( picked_object != null ){
244:             info.innerHTML='Object picked, Group: ' + picked_object.geometry.name +
, Material: ' + picked_object_material.name;
245:             objectname.style.left = mposX + 'px';
246:             objectname.style.top = mposY + 'px';
247:             objectname.innerHTML = picked_object_material.name;
248:         } else {
249:             info.innerHTML='MouseX: ' + mouseX + ', MouseY: ' + mouseY;
250:             objectname.innerHTML = '';
251:         }
252:
253:         directionalLight.position.set( camera.position.x, camera.position.y, camera.
position.z ).normalize();
254:         renderer.render( scene, camera );
255:     }
256: </script>
257: </body>
258:</html>

```