Information Visualization

W06: Exercise - Coordinate Systems and Transformations

Graduation School of System Informatics

Department of Computational Science

Naohisa Sakamoto, Akira Kageyama

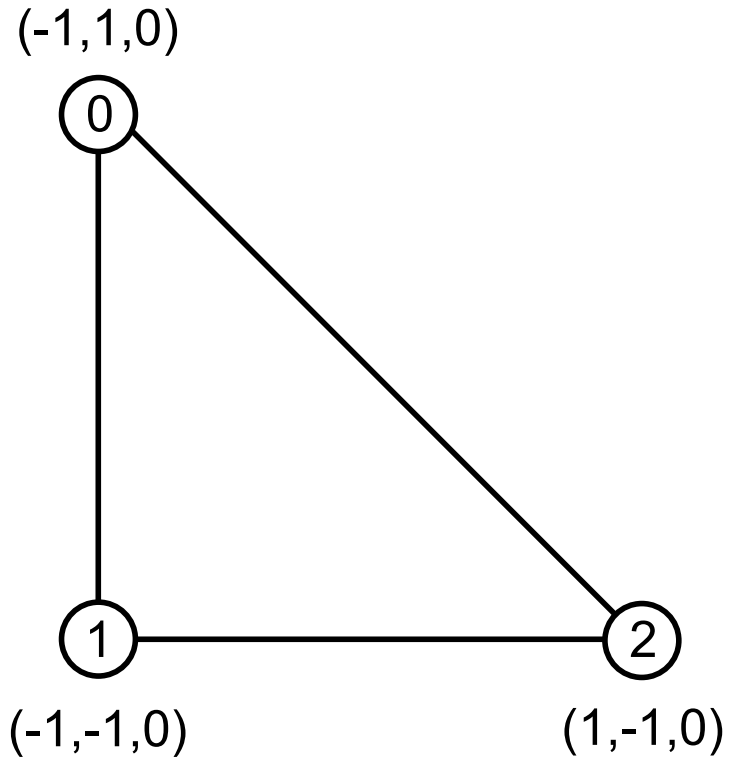
Apr.26, 2017

Schedule

- W01 4/11 Guidance
- W02 4/12 Setup
- W03 4/18 Introduction to Data Visualization
- W04 4/19 CG Programming
- W05 4/25 Rendering Pipeline
- W06 4/26 Coordinate Systems and Transformations
- W07 5/09 Shading
- W08 5/10 Shader Programming
- W09 5/16 Visualization Pipeline
- W10 5/17 Data Model and Transfer Function
- W11 5/23 Scalar Data Visualization 1 (Isosurface Extraction)
- W12 5/24 Implementation of Isosurface Extraction
- W13 5/30 Scalar Data Visualization 2 (Volume Rendering)
- W14 5/31 Implementation of Volume Rendering
- W15 6/06 Student Presentations

Drawing a triangle geometry

- A triangle geometry has three vertices and a connectivity list.




```
var vertices = [
    [-1, 1, 0], // v0
    [-1,-1, 0], // v1
    [ 1,-1, 0]  // v2
];

var faces = [
    [0,1,2] // f0: v0-v1-v2
];
```

Drawing a triangle geometry

- THREE.Geometry

```
var v0 = new THREE.Vector3().fromArray( vertices[0] );
var v1 = new THREE.Vector3().fromArray( vertices[1] );
var v2 = new THREE.Vector3().fromArray( vertices[2] );
var id = faces[0];
var f0 = new THREE.Face3( id[0], id[1], id[2] );

var geometry = new THREE.Geometry();
geometry.vertices.push( v0 );
geometry.vertices.push( v1 );
geometry.vertices.push( v2 );
geometry.faces.push( f0 );
```

Drawing a triangle geometry

- Assign a color to each face.

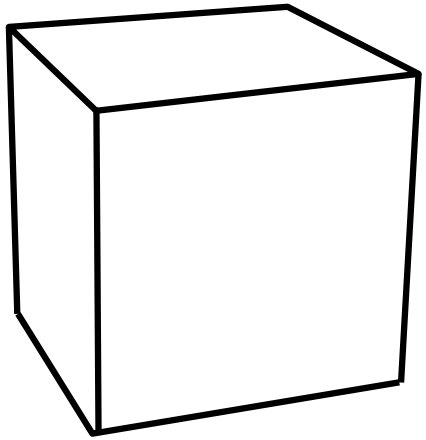
```
var material = new THREE.MeshBasicMaterial();
material.vertexColors = THREE.FaceColors;
geometry.faces[0].color = new THREE.Color( 1, 0, 0 );
```

- Assign a color to each vertex

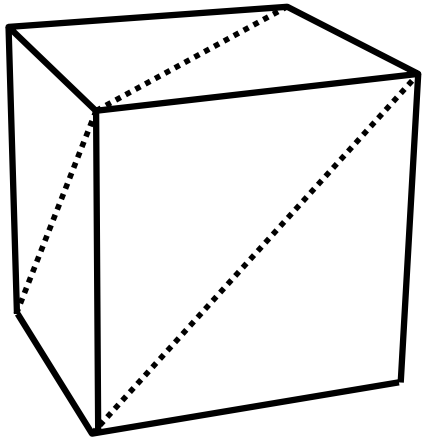
```
var material = new THREE.MeshBasicMaterial();
material.vertexColors = THREE.VertexColors;
geometry.faces[0].vertexColors.push(new THREE.Color(1,0,0));
geometry.faces[0].vertexColors.push(new THREE.Color(0,1,0));
geometry.faces[0].vertexColors.push(new THREE.Color(0,0,1));
```

Task 1

- Draw a cube by using THREE.Geometry



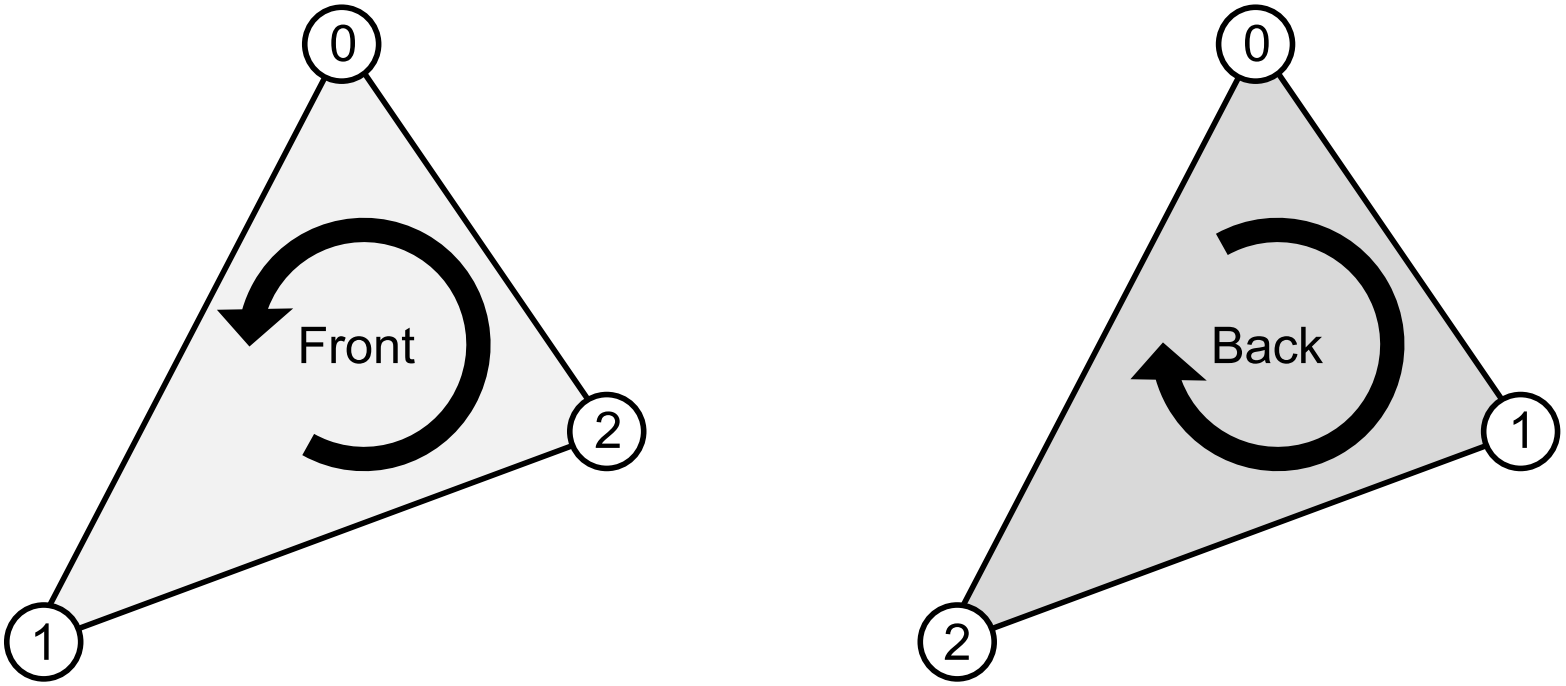
THREE.BoxGeometry



THREE.Geometry

Note

- Front and back faces
  - Depend on the order of the vertices in a triangle



Hint

- Normal Vector
  - Required for lighting (shading)

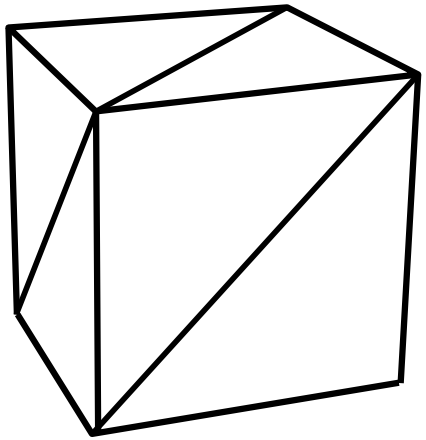
```
// Normal vectors for each face are automatically computed.
geometry.computeFaceNormals();
```

  - Both side rendering

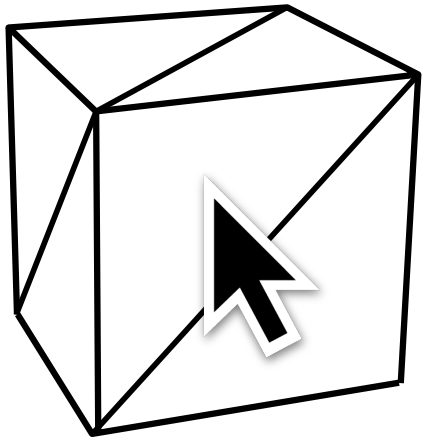
```
// Front: THREE.FrontSide (default)
// Back: THREE.BackSide
// Both: THREE.DoubleSide
metrial.side = THREE.DoubleSide
```

Task 2

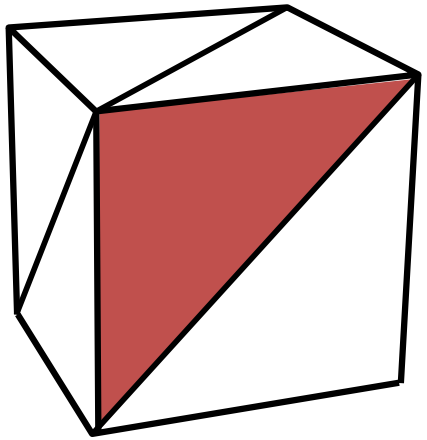
- Implement mouse picking for the triangle faces on the rotating cube.



1. Draw a rotating cube



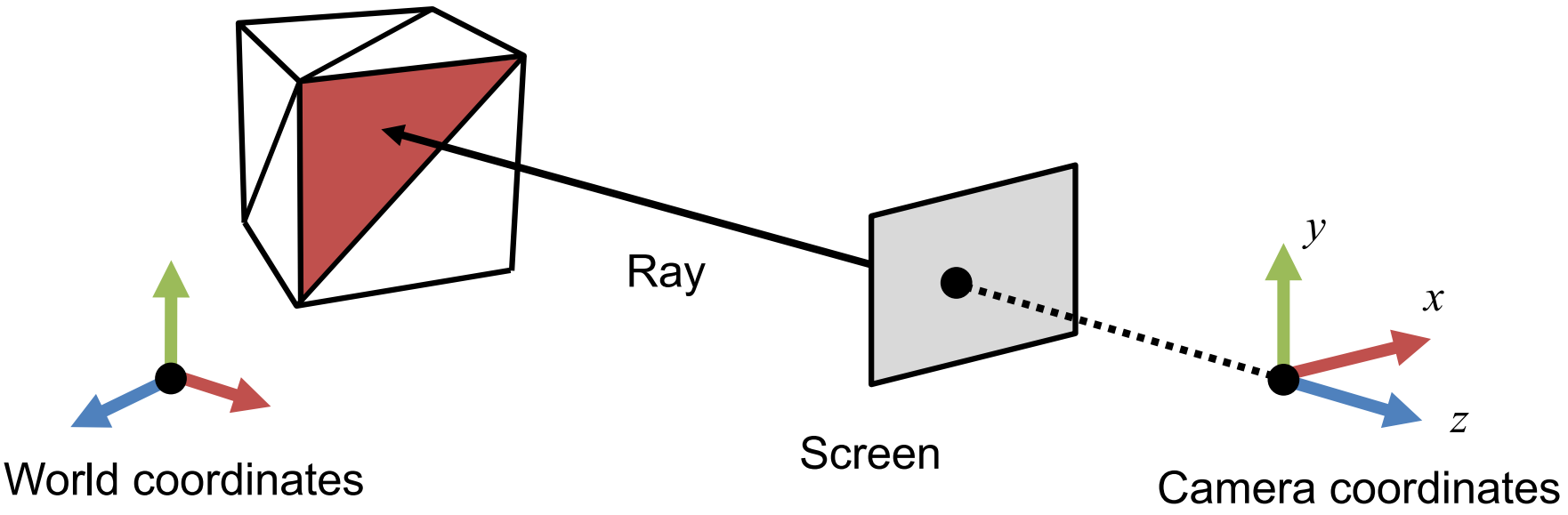
2. Click on a face



3. Change the color of the face

Hint

- Mouse picking with ray casting
  - Estimate a ray in world coordinate system from the clicked point on the screen.
  - Check ray intersections with the triangle faces.



Hint

- First of all, add a mouse down event.

```
document.addEventListener( 'mousedown', mouse_down_event );
function mouse_down_event( event )
{
    // Mouse picking
    // ...
}
```

Hint

- THREE.Raycaster

```
var raycaster = new THREE.Raycaster( origin, direction );
var intersects = raycaster.intersectObject( triangle );
if ( intersects.length > 0 )
{
    intersects[0].face.color.setRGB( 1, 0, 0 );
    intersects[0].object.geometry.colorsNeedUpdate = true;
}
```

Hint

- Clicked point in window coordinates

```
var x_win = event.clientX;
var y_win = event.clientY;
```

- Window coordinates to NDC

```
var vx = renderer.domElement.offsetLeft;
var vy = renderer.domElement.offsetTop;
var vw = renderer.domElement.width;
var vh = renderer.domElement.height;

var x_NDC = 2 * ( x_win - vx ) / vw - 1;
var y_NDC = -( 2 * ( y_win - vy ) / vh - 1 );
```

Hint

- NDC to world coordinates

```
var p_NDC = new THREE.Vector3( x_NDC, y_NDC, 1 );
var p_wld = p_NDC.unproject( camera );
```

- Origin and direction of the ray

```
var origin = ...;
var direction = ...;
```

Polling

- Take the poll
  - Student ID Number
  - Name
  - URL to Task 1
  - URL to Task 2