Introduction to Computer Graphics with WebGL

Ed Angel

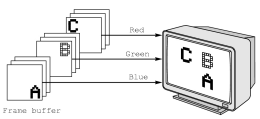
Color

Attributes

- Attributes determine the appearance of objects
  - Color (points, lines, polygons)
  - Size and width (points, lines)
  - Stipple pattern (lines, polygons)
  - Polygon mode
    - Display as filled: solid color or stipple pattern
    - Display edges
    - Display vertices
- Only a few (gl_PointSize) are supported by WebGL functions

RGB color

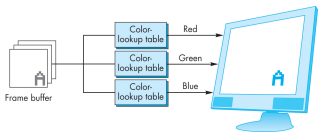
- Each color component is stored separately in the frame buffer
- Usually 8 bits per component in buffer
- Color values can range from 0.0 (none) to 1.0 (all) using floats or over the range from 0 to 255 using unsigned bytes



THE UNIVERSITY of NEW MEXICO

Indexed Color

- Colors are indices into tables of RGB values
- Requires less memory
  - indices usually 8 bits
  - not as important now
    - Memory inexpensive
    - Need more colors for shading



THE UNIVERSITY of NEW MEXICO

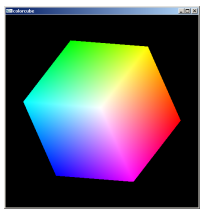
Setting Colors

- Colors are ultimately set in the fragment shader but can be determined in either shader or in the application
- Application color: pass to vertex shader as a uniform variable or as a vertex attribute
- Vertex shader color: pass to fragment shader as varying variable
- Fragment color: can alter via shader code

THE UNIVERSITY of NEW MEXICO

Smooth Color

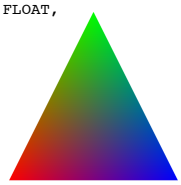
- Default is *smooth* shading
  - Rasterizer interpolates vertex colors across visible polygons
- Alternative is *flat shading*
  - Color of first vertex determines fill color
  - Handle in shader



THE UNIVERSITY of NEW MEXICO

```
var colors = [1, 0, 0, 0, 1, 0, 0, 0, 1];
var cbufferId = gl.createBuffer();
gl.bindBuffer( gl.ARRAY_BUFFER, cbufferId );
gl.bufferData( gl.ARRAY_BUFFER, flatten(colors),
    gl.STATIC_DRAW );

var vColor = gl.getAttribLocation( program, "vColor" )
gl.vertexAttribPointer( vColor, 3, gl.FLOAT,
    false, 0, 0 );
gl.enableVertexAttribArray( vColor );
```



THE UNIVERSITY of NEW MEXICO

Shaders

```
                    //vertex shader
attribute vec4 vPosition;
attribute vec4 vColor;
varying vec4 fColor;

void main(){
    gl_Position = vPosition;
    fColor = vColor;
}
                   //fragment shader
precision mediump float;
varying vec4 fColor;
void main()
{
    gl_FragColor = fColor;
}
```


THE UNIVERSITY of NEW MEXICO

Sending a Uniform Variable

```
// in application

vec4 color = vec4(1.0, 0.0, 0.0, 1.0);
colorLoc = gl.getUniformLocation( program, "color" );
gl.uniform4f( colorLoc, color);

// in fragment shader (similar in vertex shader)

uniform vec4 color;

void main()
{
    gl_FragColor = color;
}
```