

# Indexed Meshes in WebGL



CS 418: Interactive Computer Graphics  
Professor Eric Shaffer

# Indexed Face Set

Can be used for offline storage...a file format

Or can be used as an internal data structure

One block of data are the vertices

- Each vertex is a set of 3 coordinates
- Often referred to as the geometry of the mesh

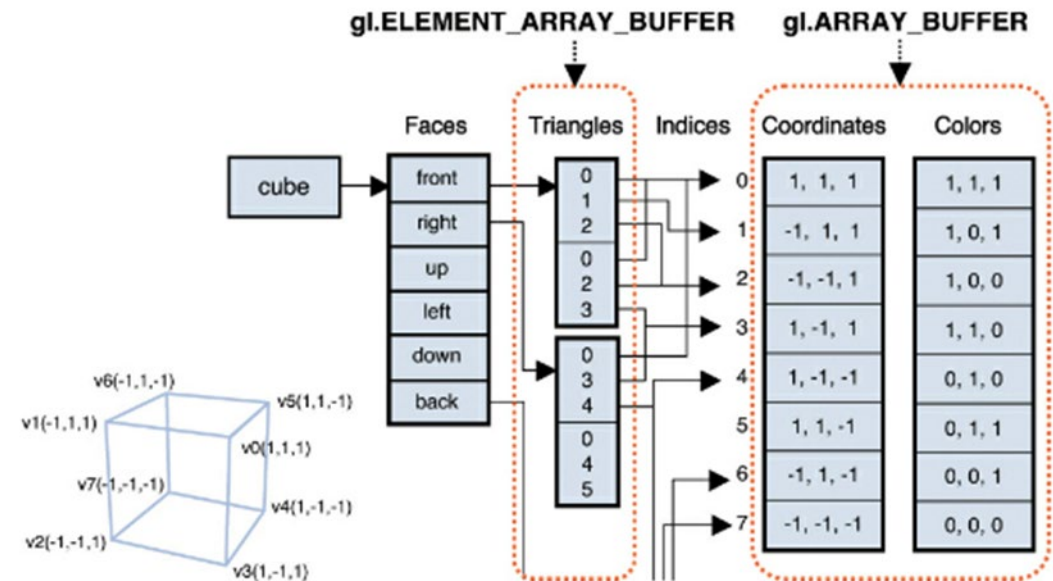
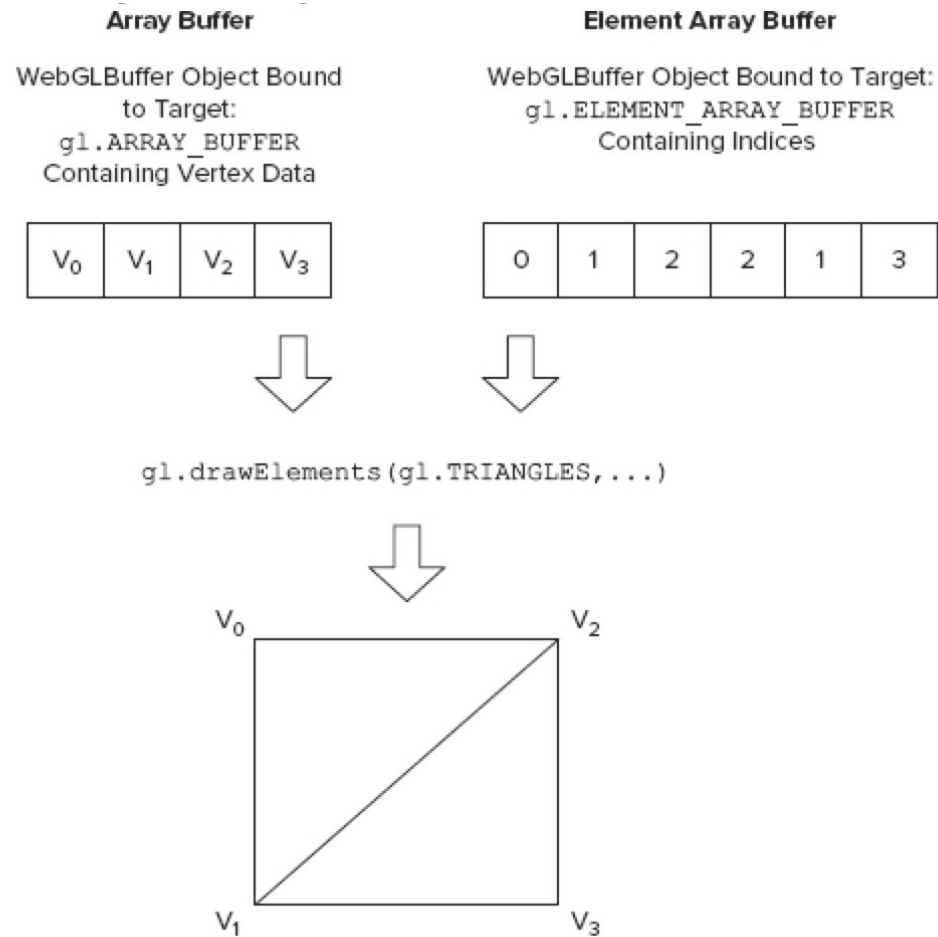
Another block of data is the set of triangles

- Each triangle is set of 3 integers vertex IDs
- The vertex IDs are indices into the vertex block

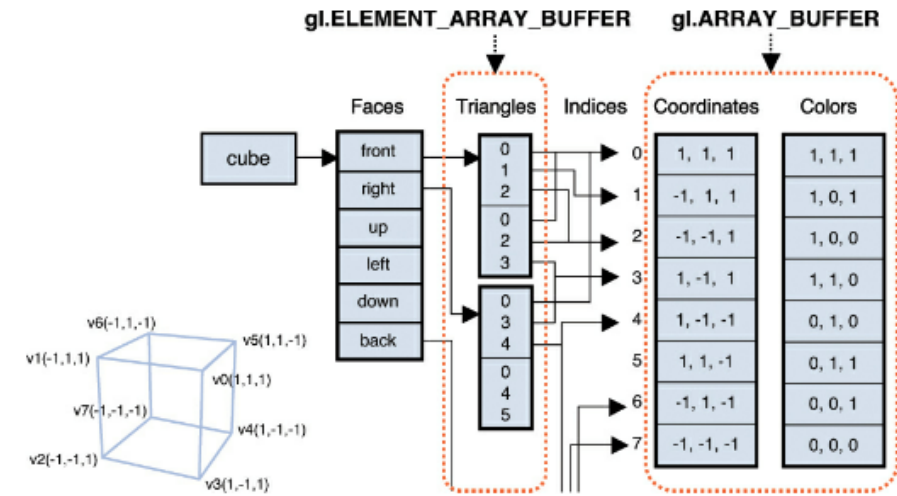
Vertices	Triangles
$x_1 \ y_1 \ z_1$	$v_{11} \ v_{12} \ v_{13}$
$\dots$	$\dots$
$x_v \ y_v \ z_v$	$\dots$
	$\dots$
	$\dots$
	$v_{F1} \ v_{F2} \ v_{F3}$

What are some advantages of this representation?

# WebGL: gl.drawElements() method



# WebGL Indexed Drawing



Bind vertex position buffer to `gl.ARRAY_BUFFER`

```
gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);  
gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(positions), gl.STATIC_DRAW);  
gl.enableVertexAttribArray(positionAttributeLocation);  
gl.vertexAttribPointer(positionAttributeLocation, size, type, normalize, stride, offset);
```

Need to create an index buffer bound to `gl.ELEMENT_ARRAY_BUFFER`

```
gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, indexBuffer);  
gl.bufferData(gl.ELEMENT_ARRAY_BUFFER, new Uint16Array(indices), gl.STATIC_DRAW);
```

At draw time, bind to the index buffer and draw with `gl.drawElements`

```
gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, indexBuffer);  
gl.drawElements(primitiveType, count, indexType, offset);
```

# gl.drawElements primitives

```
void gl.drawElements(mode, count, type, offset);
```

## Parameters

### mode

A `GLenum` specifying the type primitive to render. Possible values are:

- `gl.POINTS`: Draws a single dot.
- `gl.LINE_STRIP`: Draws a straight line to the next vertex.
- `gl.LINE_LOOP`: Draws a straight line to the next vertex, and connects the last vertex back to the first.
- `gl.LINES`: Draws a line between a pair of vertices.
- `gl.TRIANGLE_STRIP`
- `gl.TRIANGLE_FAN`
- `gl.TRIANGLES`: Draws a triangle for a group of three vertices.

# Index Precision

```
void gl.drawElements(mode, count, type, offset);
```

## type

A `GLenum` specifying the type of the values in the element array buffer. Possible values are:

- `gl.UNSIGNED_BYTE`
- `gl.UNSIGNED_SHORT`
- When using the `OES_element_index_uint` extension:
  - `gl.UNSIGNED_INT`

## for WebGL 1

`gl.UNSIGNED_BYTE` with max index of 255

`gl.UNSIGNED_SHORT` where the maximum index is 65,535.

`OES_element_index_uint` extension can be enabled

allows `gl.UNSIGNED_INT` and indices up to 4,294,967,296

# OES\_element\_index\_uint extension

```
gl.getExtension(name);
```

## Parameters

### name

A `String` for the name of the WebGL extension to enable.

## Return value

A WebGL extension object, or `null` if name does not match (case-insensitive) to one of the strings in `WebGLRenderingContext.getSupportedExtensions`.

```
1 var ext = gl.getExtension('OES_element_index_uint');  
2   
3 gl.drawElements(gl.POINTS, 8, gl.UNSIGNED_INT, 0);
```