

Computer Graphics

Jochen Lang

jlang@uottawa.ca

Faculté de génie | Faculty of Engineering

Jochen Lang, EECS
jlang@uOttawa.ca

Buffers in WebGL2

- Set-up in initialization routine
 - generate and bind VAO
 - generate, bind and fill (multiple) VBO for storing vertices and their attributes (e.g., colors, normals, texture coordinates)
 - unbind VAO and possibly go to the next set-up
- Drawing routine
 - bind the desired VAO
 - call draw using data stored in the VBOs of the VAO
 - unbind the VAO
- Clean-up
 - If there is a lot of data in GL, free up the space taken up by the VAO/VBO (see `gl.deleteVertexArrays`)

Using Additional Vertex Attributes

- Adjust the shader program

```
layout (location=0) in vec4 position;
layout (location=1) in vec3 color;
```

- Query the locations for the attributes in your initialization

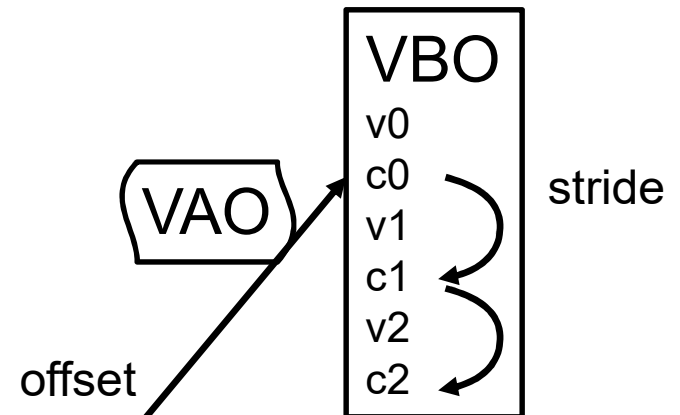
```
locPos = gl.getAttributeLocation( program, "a_vertex");
locColor = gl.getAttributeLocation(program, "a_color");
```

- Use the location

```
gl.vertexAttribPointer(locColor, 3, gl.FLOAT, false,
                        0, 0);
gl.enableVertexAttribArray(locColor);
```

Format of VBO: Interleaved

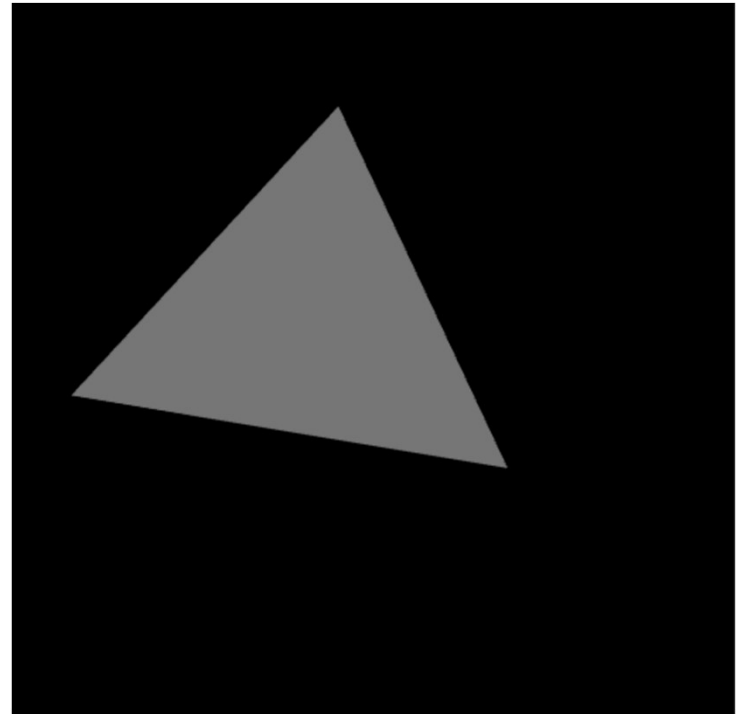
- Interleaved storage in one buffer
 - Keep all information about a vertex together and make use of the stride parameter
 - Here assuming a vertex is stored as three floats



```
gl.vertexAttribPointer(locColor, 3, gl.FLOAT, false,
    Float32Array.BYTES_PER_ELEMENT * 3 )
Float32Array.BYTES_PER_ELEMENT * 3 );
```

Tasks in this Lab

- Run the starter code
- Modify the starter code to use colors from a buffer rather than hardcoded yellow.
 - Change vertex shader
 - Add color information (rgb values) to buffer in javascript
 - Link color attribute from shader to colour in VBO



Summary

- Buffer objects enable data transfer from host program to WebGL API
- Buffer objects hold byte data and can be used flexibly
- Buffer objects can be organized in VAO

