



# Introduction to Computer Graphics with WebGL

Ed Angel

JavaScript

---

---

---

---

---

---

---



## JavaScript Notes

- JavaScript (JS) is the language of the Web
  - All browsers will execute JS code
  - JavaScript is an interpreted object-oriented language
  - JavaScript is not related to Java but contains elements from many languages
- References
  - Flanagan, JavaScript: The Definitive Guide, O'Reilly
  - Crockford, JavaScript, The Good Parts, O'Reilly
  - Many Web tutorials

---

---

---

---

---

---

---



## Executing JS Code

- In browser
  - embed in HTML page

```
<html>
<script type = text/javascript>
  console.log("hello world");
</script>
</html>
```
- Download an application such as node.js and run inside a window

```
>node myfile.js
```

---

---

---

---

---

---

---

## JS Notes

- Is JS slow?
  - JS engines in browsers are getting much faster
  - Not a key issues for graphics since once we get the data to the GPU it doesn't matter how we got the data there
- JS is a (too) big language
  - We don't need to use it all
  - Choose parts we want to use
  - Don't try to make your code look like C or Java

---

---

---

---

---

---

---

## JS Notes

- Only three native types:
  - numbers
  - strings
  - booleans
- Only one numerical type: 32 bit float
  - `var x = 1;`
  - `var x = 1.0; // same`
  - two operators for equality `==` and `===`
    - `1==1' //true`
    - `1===1' //false`
- Everything else is an object
- Dynamic typing

---

---

---

---

---

---

---

## JS Objects

- There are many ways to define objects in JS including methods similar to Java and C++
- All objects inherit attributes and methods from a prototype which itself may have a prototype
- We will consider arrays as an example and because we need them for our applications

---

---

---

---

---

---

---

## Scope

- Different from other languages
- Function scope
- variables are *hoisted* within a function
  - can use a variable before it is declared
- Note functions are first class objects in JS
  - `x = 1` //legal statement but dangerous
  - use `var x = 1`;
- Also note that both window and canvas are defined outside of our application and are globals in our code

---

---

---

---

---

---

---

## JS Arrays

- JS arrays are objects
  - inherit methods
  - `var a = [1, 2, 3];`
    - is not the same as in C++ or Java
  - `a.length` // 3
  - `a.push(4);` // length now 4
  - `a.pop();` // 4
  - avoids use of many loops and indexing
  - Problem for WebGL which expects C-style arrays

---

---

---

---

---

---

---

## Typed Arrays

JS has typed arrays that are like C arrays

```
var a = new Float32Array(3)
var b = new Uint8Array(3)
```

Generally, we prefer to work with standard JS arrays and convert to typed arrays only when we need to send data to the GPU with the *flatten()* function in MV.js

When working with a 4 x 4 array, WebGL (and all versions of OpenGL) expect to see a sixteen element array with data in column major order

---

---

---

---

---

---

---

## A Minimalist Approach

- We will use only core JS and HTML
  - no extras or variants
- No additional packages
  - CSS
  - JQuery
- Focus on graphics
  - examples may lack beauty
- You are welcome to use other variants as long as others can run them from your URL

---

---

---

---

---

---

---