

计算机图形学

唐慧

影像科学与技术实验室

东南大学计算机学院

corinna@seu.edu.cn

2.1: WebGL相关技术

- 例子：画一个三角形
 - 每个程序包含（至少）两个文件
 - HTML 文件和 JavaScript 文件
- HTML
 - 描述页面
 - 包括应用
 - 包括着色器
- JavaScript
 - 图形绘制

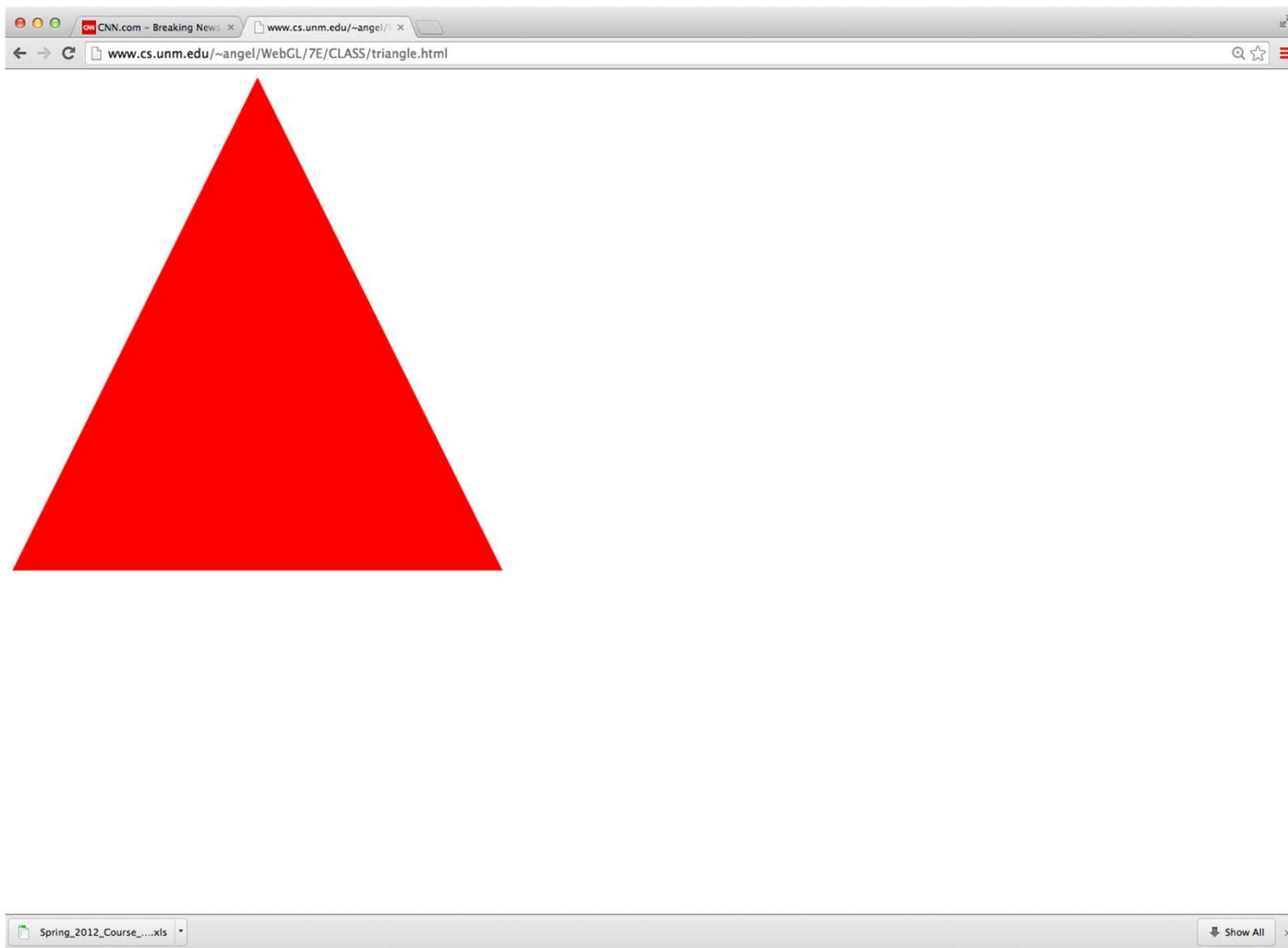


在WebGL中编程

- 可以在任何最新的浏览器中运行WebGL
 - Chrome
 - Firefox
 - Safari
 - IE
- 代码用JavaScript编写
- 在浏览器内部运行 JS
 - 运用本地资源



举例: triangle.html



例子代码

```
<!DOCTYPE html>
<html>
<head>
<script id="vertex-shader" type="x-shader/x-vertex">
attribute vec4 vPosition;
void main(){
    gl_Position = vPosition;
}
</script>
<script id="fragment-shader" type="x-shader/x-fragment">
precision mediump float;
void main(){
    gl_FragColor = vec4( 1.0, 0.0, 0.0, 1.0 );
}
</script>
```



HTML 文件 (续上)

```
<script type="text/javascript" src="../Common/webgl-utils.js"></script>
<script type="text/javascript" src="../Common/initShaders.js"></script>
<script type="text/javascript" src="../Common/MV.js"></script>
<script type="text/javascript" src="triangle.js"></script>
</head>
<body>
<canvas id="gl-canvas" width="512" height="512">
Oops ... your browser doesn't support the HTML5 canvas element
</canvas>
</body>
</html>
```



JS 文件

```
var gl;  
var points;
```

```
window.onload = function init(){  
    var canvas = document.getElementById( "gl-canvas" );  
    gl = WebGLUtils.setupWebGL( canvas );  
    if ( !gl ) { alert( "WebGL isn't available" );  
    }  
}
```

// Three Vertices

```
var vertices = [  
    vec2( -1, -1 ),  
    vec2( 0, 1 ),  
    vec2( 1, -1 )  
];
```



JS文件（续上）

```
// Configure WebGL
//
gl.viewport( 0, 0, canvas.width, canvas.height );
gl.clearColor( 1.0, 1.0, 1.0, 1.0 );

// Load shaders and initialize attribute buffers

var program = initShaders( gl, "vertex-shader", "fragment-shader" );
gl.useProgram( program );

// Load the data into the GPU

var bufferId = gl.createBuffer();
gl.bindBuffer( gl.ARRAY_BUFFER, bufferId );
gl.bufferData( gl.ARRAY_BUFFER, flatten(vertices),
gl.STATIC_DRAW );
```



JS文件 (续上)

```
// Associate our shader variables with our data buffer
```

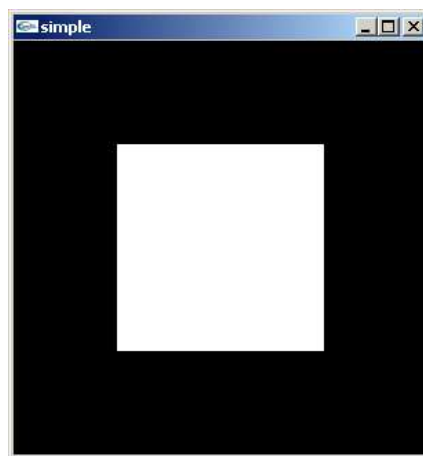
```
    var vPosition = gl.getAttributeLocation( program, "vPosition" );  
    gl.vertexAttribPointer( vPosition, 2, gl.FLOAT, false, 0, 0 );  
    gl.enableVertexAttribArray( vPosition );  
    render();  
};
```

```
function render() {  
    gl.clear( gl.COLOR_BUFFER_BIT );  
    gl.drawArrays( gl.TRIANGLES, 0, 3 );  
}
```



练习

- 下载triangle.html 和 triangle.js 到本地，然后运行
- 编辑这两个文件，并且
 - 改变颜色
 - 显示多个三角形（至少2个）
- 实现下面的效果：



JavaScript 小贴士

JavaScript (JS) 是Web语言

- 所有浏览器都可以运行JS 代码
- JavaScript 是解释性的、面向对象的语言

参考书

- Flanagan, JavaScript: The Definitive Guide, JavaScript权威指南, O'Reilly/机械工业出版社
- Crockford, JavaScript, The Good Parts, O'Reilly
- 互联网教程



JS小贴士

○ JS 慢吗?

- 浏览器中的JS 引擎现在越来越快
- 这个不是图形学的关键问题，因为一旦我们把数据导入到GPU，我们就不需要管是怎么导入的了。

○ JS 是很大的语言

- 我们不需要用全部
- 只选择我们想用的部分
- 不要试图让你的代码像 C 或者 Java



JS小贴士

- 非常少的数据类型：
 - 数字
 - 字符串
 - 布尔
- 只有一种数值类型：32位 float
 - `var x = 1;`
 - `var x = 1.0; // same`
 - potential issue in loops
 - 对于等于的判断有两种：`==` and `===`
- 动态类型 (Dynamic typing)



作用域(Scoping)

- 与其他语言不同
 - 函数作用域(Function scope)
- 在函数体内变量被*hoisted*
 - can use a variable before it is declared
- 在JS中函数是首要类对象



Hoisting

```
1  //代码段1-----
2  var myvar = '变量值';
3  console.log(myvar); // 变量值
4  //代码段2-----
5  var myvar = '变量值';
6  (function() {
7    console.log(myvar); //变量值
8  })();
9  //代码段3-----
10 var myvar = '变量值';
11 (function() {
12   console.log(myvar); // undefined
13   var myvar = '内部变量值';
14 })();
```



Hoisting

```
> //代码段3-----  
var myvar = '变量值';  
(function() {  
  console.log(myvar); // undefined  
  var myvar = '内部变量值';  
})();
```

undefined

```
> //代码段3 Hoisting-----  
var myvar = '变量值';  
(function() {  
  var myvar;//undefined  
  console.log(myvar); // undefined  
  myvar = '内部变量值';  
})();
```

undefined



JS 数组

- JS 数组是对象
 - 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



特定类型的数组

JS has typed arrays that are like C arrays

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

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

通常来说，我们倾向工作于标准JS数组，只有当我们需要向GPU发送数据的时候才转换为特定类型的数组，通过MV.js中的函数



最简方法

- 我们只会用核心的JS和HTML
 - no extras or variants
- No additional packages
 - CSS
 - JQuery
- 专注于图形
 - examples may lack beauty

