

WebGL (2.0)

- OpenGL **ES** 3.0 za spletne brskalnike (2017)
 - https://en.wikipedia.org/wiki/OpenGL_ES#OpenGL_ES_3.0
 - <https://www.khronos.org/registry/webgl/specs/latest/2.0/>
- Risanje na <canvas>
- Podprti praktično vsi moderni brskalniki
 - Tudi za mobilne naprave
- Prednost: 3D vizualizacija za najbolj širok nabor naprav
 - <https://caniuse.com/webgl2>

Chrome	Edge*	Safari	Firefox	Opera	IE
			2-24		
			1 2 25-41		
4-42		3.1-10	1 42-44		
3 43-55	12-18	4 10.1-14.1	1 5 45-50	10-42	
56-106	79-106	15-16.0	51-105	43-90	6-10
107	107	16.1	106	91	11
108-110		16.2-TP	107-108		

Chrome for Android	Safari on iOS*	Samsung Internet	Opera Mini*	Opera Mobile*	UC Browser for Android	Android Browser*	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
	3.2-11.4									
	4 12-14.8	4-6.4								
	15-16.0	7.2-17.0		12-12.1		2.1-4.4.4				
107	16.1	18.0	all	64	13.4	107	106	13.1	13.18	1 5 2.5

Razlike med WebGL 2.0 in OpenGL 3/4

- OpenGL ES 3.0
<https://registry.khronos.org/OpenGL-Refpages/es3.0/>
 - Večinoma enak kot OpenGL 3/4, manj funkcionalnosti
- `glDrawArrays` → `gl.drawArrays`
- ~~`GL_DOUBLE`~~
- Precision Qualifiers (v senčilnikih):
 - `precision highp/mediump/lowp float;`
- Različica v senčilnikih: `#version 300 es`

precision highp/mediump/lowp

- Priporočljivo nastavljanje natančnosti spremenljivk v senčilnikih

Qualifier	Floating Point Range	Floating Point Magnitude Range	Floating Point Precision	Integer Range	
				Signed	Unsigned
highp	As IEEE-754 $(-2^{126}, 2^{127})$	As IEEE-754 $0.0, (2^{-126}, 2^{127})$	As IEEE 754 relative: 2^{-24}	$[-2^{31}, 2^{31}-1]$	$[0, 2^{32}-1]$
mediump (minimum requirements)	$(-2^{14}, 2^{14})$	$(2^{-14}, 2^{14})$	Relative: 2^{-10}	$[-2^{15}, 2^{15}-1]$	$[0, 2^{16}-1]$
lowp (minimum requirements)	$(-2, 2)$	$(2^{-8}, 2)$	Absolute: $2^{-8} / 2^{-9}$ signed/unsigned	$[-2^8, 2^8-1]$	$[0, 2^9-1]$

Implementacija

- Canvas:

```
<body>
  <div id="containerDiv">
    <canvas id="myCanvas" width="400" height="400">Your web browser does not support Canvas.</canvas>
  </div>
</body>
```

- Inicijalizacija (javascript):

```
$(document).ready(function() {

    var gl = null;
    var canvas = document.getElementById("myCanvas");
    var msg = "";
    try {
        gl = canvas.getContext("webgl2");
    } catch (e) {
        msg = "Error creating WebGL context: " + e.toString();
    }

    if (!gl) {
        alert(msg);
        throw new Error(msg);
    }

    gl.clearColor(0.2, 0.2, 0, 1);
    ...
});
```

Implementacija

- Senčilniki:

```
<head>
  <script id="simple-vs" type="x-shader/x-vertex">
    #version 300 es
    precision mediump float;
    uniform mat4 PVM;
    layout(location = 0) in vec3 VertexPosition;

    void main(){
      gl_Position=PVM * vec4(VertexPosition,1);
    }
  </script>
  <script id="simple-fs" type="x-shader/x-fragment">
    #version 300 es
    precision mediump float;
    void main(){
      gl_FragColor=vec4(1,0,0,1);
    }
  </script>
</head>
```

- Prevajanje:

```
var program = gl.createProgram();

var vs = gl.createShader(gl.VERTEX_SHADER);
gl.shaderSource(vs, $("#simple-vs").text());
gl.compileShader(vs);
gl.attachShader(program, vs);
var shaderInfo = gl.getShaderInfoLog(vs);
if (shaderInfo.length > 0) {
  alert("VShader Info: " + shaderInfo);
}
...
```

Implementacija

- Nalaganje podatkov na GPU):
 - Ne pozabiti na VAO!

```
bufferTrikotniki = gl.createBuffer();
vaoTrikotniki = gl.createVertexArray();
gl.bindVertexArray(vaoTrikotniki);
gl.bindBuffer(gl.ARRAY_BUFFER, bufferTrikotniki);
gl.bufferData(gl.ARRAY_BUFFER, new Float32Array([ 0, 0, 0, 0.5, 0.0, 0.0, 0.5, 0.5, 0.0 ]), gl.STATIC_DRAW);
gl.enableVertexAttribArray(0);
gl.vertexAttribPointer(0, 3, gl.FLOAT, false, 3 * 4, 0);
```

Implementacija

- Izris:

```
setInterval(drawLoop, 33);  
  
function drawLoop() {  
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);  
    ...  
    gl.bindVertexArray(vaoTrikotniki);  
    gl.drawArrays(gl.TRIANGLES, 0, 3);  
    ...  
    var e = gl.getError();  
    if (e) {  
        alert("Draw error: " + e);  
    }  
}
```

Delo z matrikami

- <http://glmatrix.net/> omogoča podobno notacijo kot GLM
- Alternativa: <https://humbletim.github.io/glm-js/>

```
var matrix = glmatrix.mat4.create();
glMatrix.mat4.identity(matrix);
...
var matrix2 = glmatrix.mat4.create();
glMatrix.mat4.identity(matrix2);
...
var matrix3 = glmatrix.mat4.create();
glMatrix.mat4.identity(matrix3);
...
```

```
var M = glmatrix.mat4.multiply(
  glmatrix.mat4.create(),
  matrix,
  glmatrix.mat4.multiply(
    glmatrix.mat4.create(),
    matrix2,
    matrix3
  )
);
```

```
gl.uniformMatrix4fv( gl.getUniformLocation(program, "PVM"), false, PVM);
```


Dodatna literatura

- https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Getting_started_with WebGL
- <https://msdn.microsoft.com/en-us/library/dn621085%28v=v.s.85%29.aspx>
- <https://www.khronos.org/registry/webgl/specs/2.0/>
- https://www.khronos.org/webgl/wiki/Main_Page
- <https://registry.khronos.org/OpenGL-Refpages/es3.0/>
- <https://shaderific.com/glsl.html>
 - Primerjava števila vseh funkcij od OpenGL:
 - <https://www.opengl.org/sdk/docs/man2/> - OpenGL 2
 - <https://www.opengl.org/sdk/docs/man4/> - OpenGL 4
- **Pazite, da je dokumentacija za WebGL 2.0 in ne samo WebGL (1.0, stara različica)**

WebGL 1.0 – zgodovina

- Starejša različica standarda
 - Temelji na OpenGL ES 2.0
 - Je še sploh uporaben?
- Podprt na še več brskalnikih: <https://caniuse.com/webgl>
- Ni podprto:
 - ~~Non-Power of Two Texture Support~~
 - ~~3D textures~~
- Programska koda je bolj podobna OpenGL 2.x
- Razlike v kodi
 - Ni podprto: GenVertexArrays, in/out v senčilnikih, ...
 - Vhod v senčilnik oglišč
 - in vec3 VertexPosition → attribute vec3 VertexPosition;
 - layout(location=0) in ~~attribute~~ vec3 in_Position;
 - V kodi JavaScript: canvas.bindAttribLocation(program, 0, "in_Position")
 - Prenos podatkov med senčilnikom oglišč in fragmentov
 - in/out vec3 LightIntensity → varying vec3 LightIntensity
 - VAO ne obstaja (~~glBindVertexArray~~, ~~glCreateVertexArray~~)
 - bindBuffer, vertexAttribPointer,... je potrebno vsakič klicati med izrisom