



Mikor használjunk hardvergyorsított Webes térképet?

<https://gaborfarkas.github.io>
<https://github.com/GaborFarkas>
gfarkas@gamma.ttk.pte.hu



Farkas Gábor
tanársegéd
PTE TTK FI

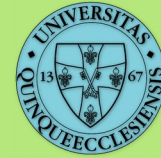
Térképrajzolás a Weben



- 3 különböző módszer
 - DOM (D3, OpenLayers 2, Leaflet)
 - Canvas (OpenLayers, Leaflet)
 - WebGL (Mapbox GL JS, Deck.gl, Tangram, OpenLayers)



DOM



OpenStreetMap Edit History Export

GPS Traces User Diaries Copyright Help About GFarkas

Edit feature

Point Line Area

University Grounds

All fields

Name
Budapesti Műszaki és Gazdaságtudományi Egyetem

Multilingual name
Deutsch
Technische und Wirtschaftswissenschaftliche Universit...

Multilingual name
English
Budapest University of Technology and Economics

Multilingual name
magyar
Budapesti Műszaki és Gazdaságtudományi Egyetem

Operator
Unknown

Address
123 Street Unit
City Postcode

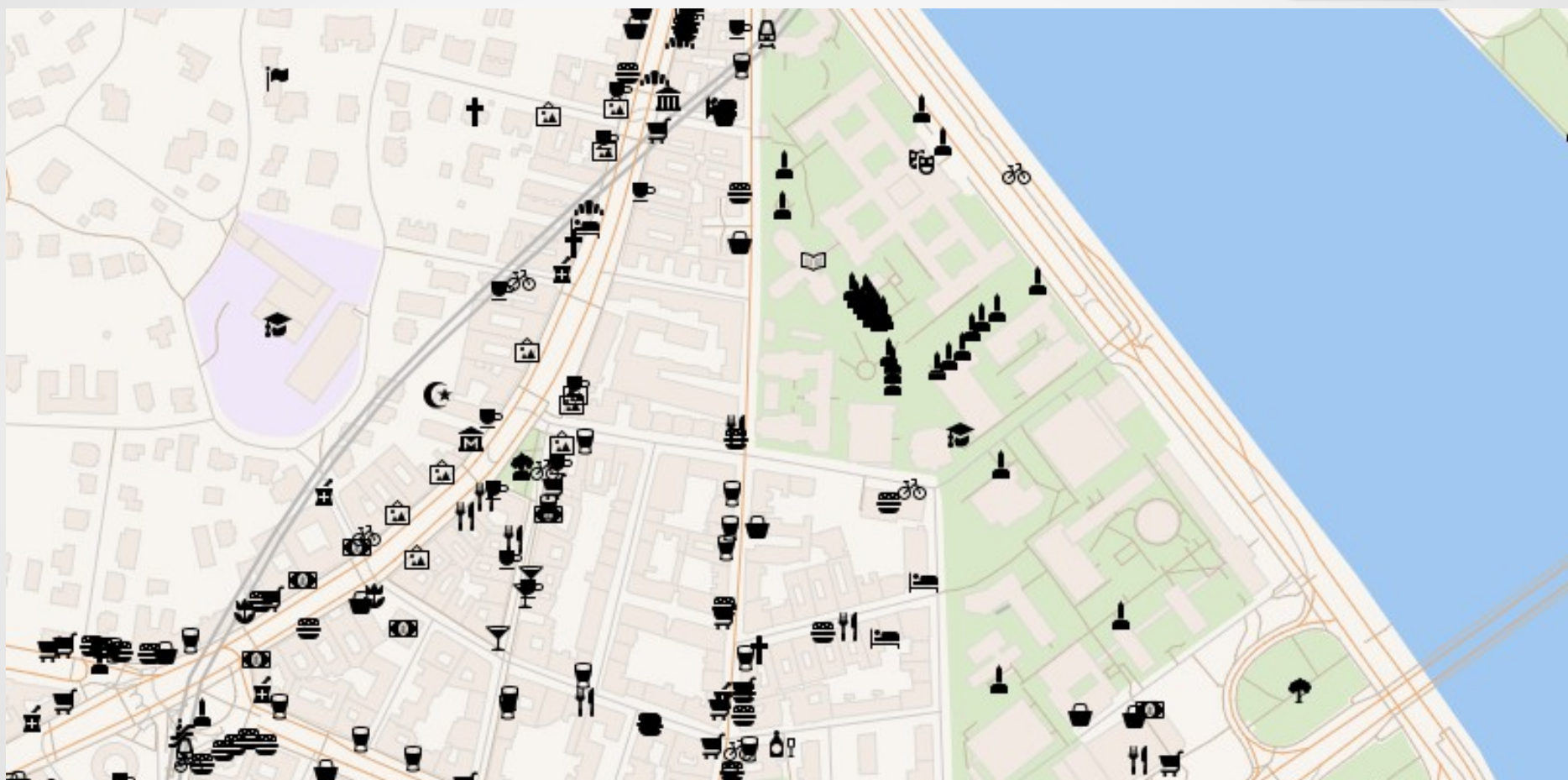
Internet Access
Yes, No, Wifi...

SSID (Network Name)

View on opensstreetmap.org

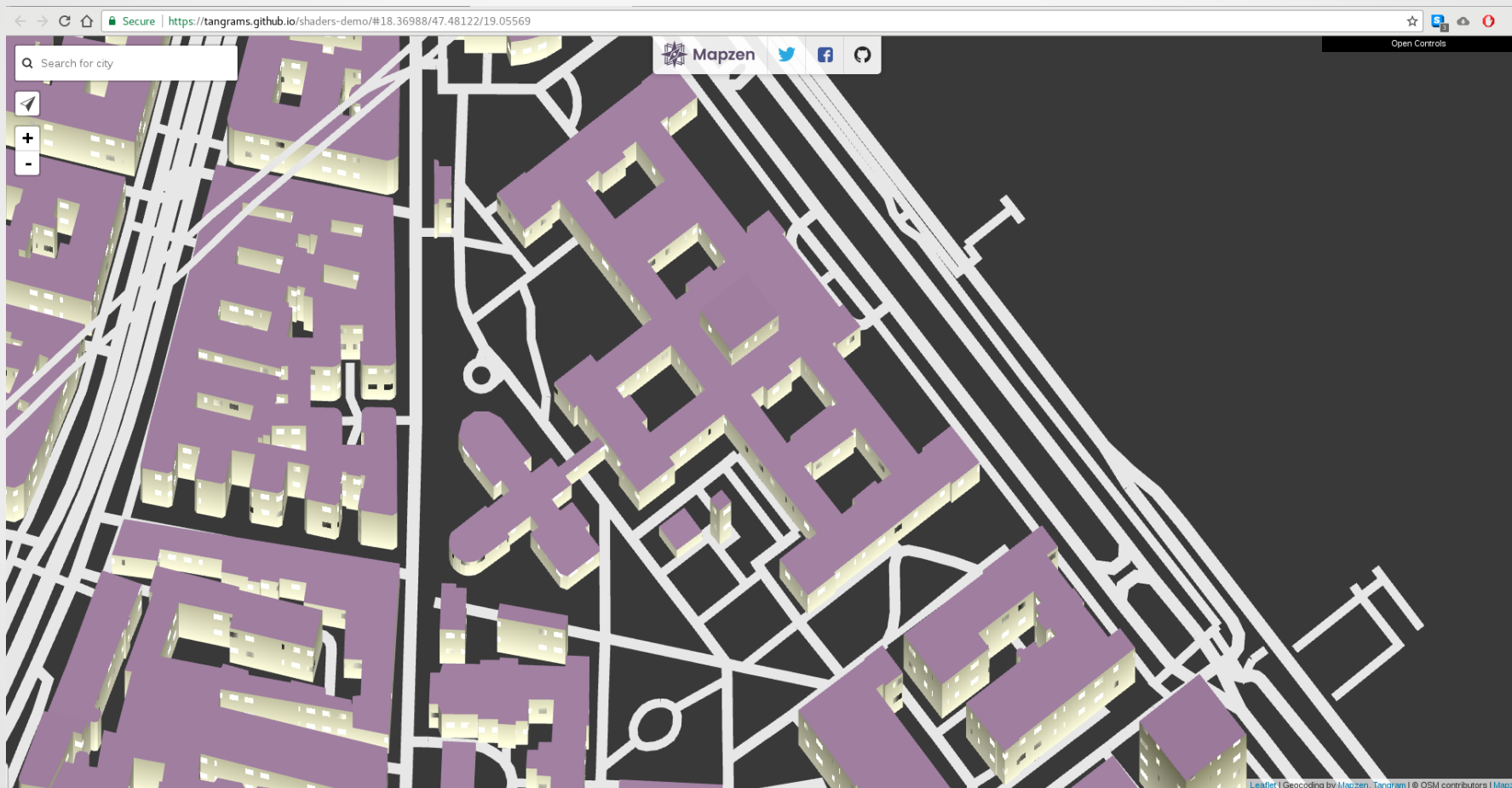
2017. november 24.

Canvas



2017. november 24.

WebGL

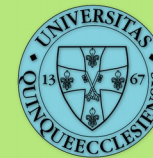


2017. november 24.

- Kirajzolás kiválóan párhuzamosítható folyamat, tehát
 - Minél jobban szétszedjük annál több mag tud számolni egyszerre
 - Nem kell magas órajel, csupán tömérdek processzor
 - GPU
- Minél több feladatot lát el a GPU, annál gyorsabb



Hardveres gyorsítás



← → ↻ 🏠 Chromium | chrome://gpu

Note: To properly save this page, select the "Webpage, Complete" option in the Save File dialog.

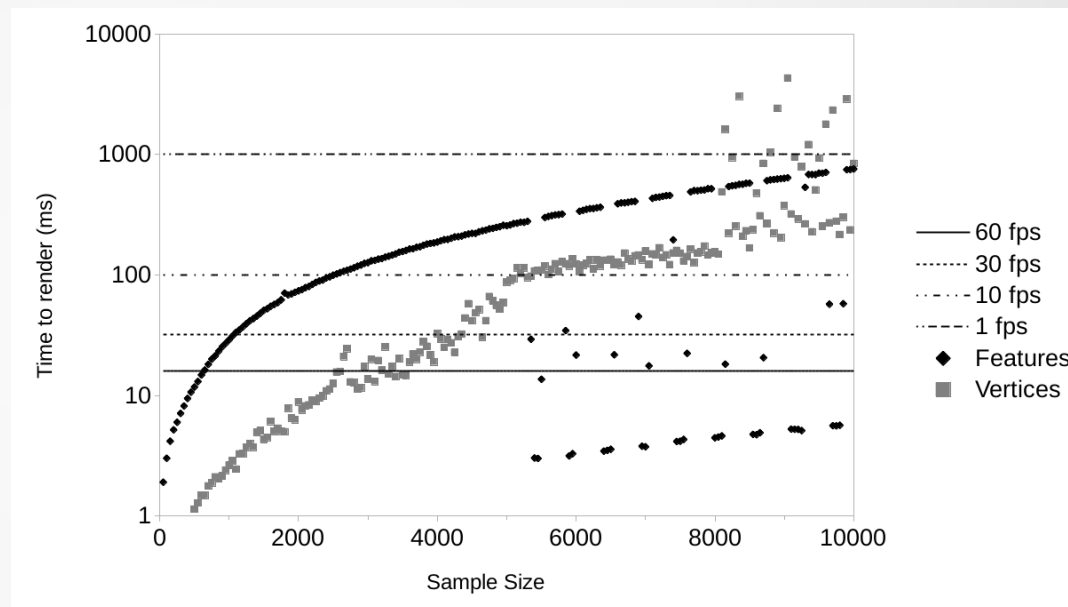
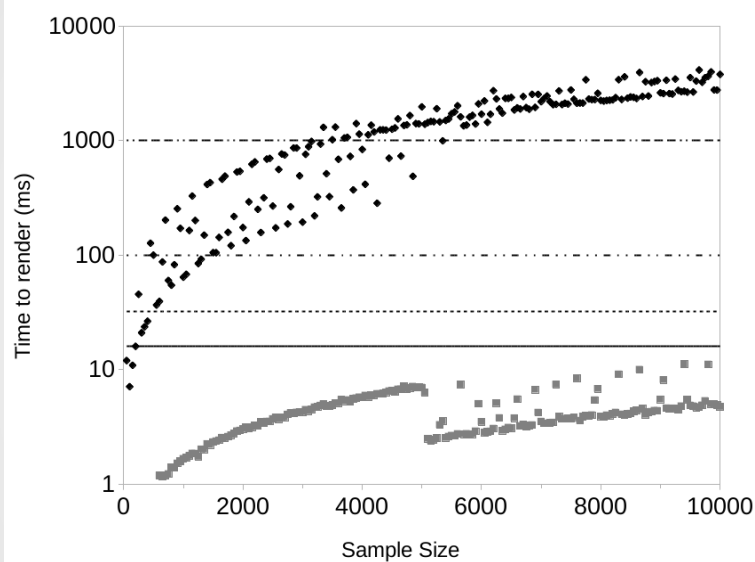
Graphics Feature Status

- Canvas: **Hardware accelerated**
- CheckerImaging: **Disabled**
- Flash: **Hardware accelerated**
- Flash Stage3D: **Hardware accelerated**
- Flash Stage3D Baseline profile: **Hardware accelerated**
- Compositing: **Hardware accelerated**
- Multiple Raster Threads: **Enabled**
- Native GpuMemoryBuffers: **Software only. Hardware acceleration disabled**
- Rasterization: **Software only. Hardware acceleration disabled**
- Video Decode: **Hardware accelerated**
- Video Encode: **Hardware accelerated**
- WebGL: **Hardware accelerated**
- WebGL2: **Hardware accelerated**

Driver Bug Workarounds

- adjust_src_dst_region_for_blitframebuffer
- clear_uniforms_before_first_program_use
- count_all_in_varyings_packing
- decode_encode_srgb_for_generatemipmap
- disable_framebuffer_cmaa
- disable_post_sub_buffers_for_onscreen_surfaces
- msaa_is_slow
- scalarize_vec_and_mat_constructor_args

DOM vagy Canvas?



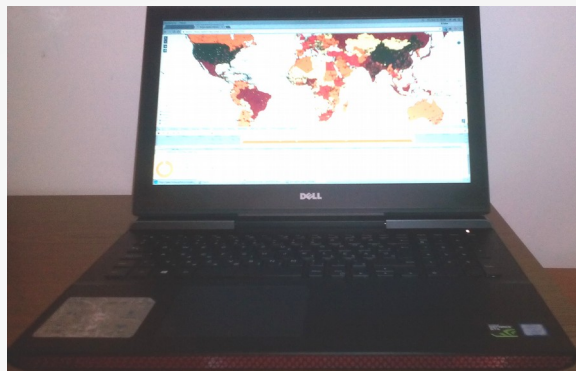
Canvas vagy WebGL?



```
JS Untitled-1
1 clippedLayer.on('precompose', function (evt) {
2   var ctx = evt.context;
3   ctx.save();
4   ctx.beginPath();
5   ctx.rect(20, 20, 100, 100);
6   ctx.clip();
7 });
8 clippedLayer.on('postcompose', function (evt) {
9   evt.context.restore();
10 });
11
```

```
JS Untitled-2
7 'attribute vec2 a_position;',
8 'uniform vec2 u_resolution;',
9 'void main() {',
10 '    vec2 pixelPos = a_position / u_resolution * 2.0 - 1.0;',
11 '    gl_Position = vec4(pixelPos * vec2(1, -1), 0, 1);',
12 '}',
13 ].join('');
14 clippedLayer.on('precompose', function (evt) {
15   var context = evt.glContext;
16   var gl = context.getGL();
17   var program = gl.createProgram();
18   var vertexShader = gl.createShader(gl.VERTEX_SHADER);
19   gl.shaderSource(vertexShader, vertexShaderSource);
20   gl.compileShader(vertexShader);
21   gl.attachShader(program, vertexShader);
22   var fragmentShader = gl.createShader(gl.FRAGMENT_SHADER);
23   gl.shaderSource(fragmentShader, fragmentShaderSource);
24   gl.compileShader(fragmentShader);
25   gl.attachShader(program, fragmentShader);
26   gl.linkProgram(program);
27   context.useProgram(program);
28   var resolutionLocation = gl.getUniformLocation(program, 'u_resolution');
29   gl.uniform2f(resolutionLocation, context.A.width, context.A.height);
30   var positionLocation = gl.getAttribLocation(program, 'a_position');
31   gl.enable(gl.STENCIL_TEST);
32   gl.colorMask(false, false, false, false);
33   gl.stencilOp(gl.KEEP, gl.KEEP, gl.REPLACE);
34   gl.stencilFunc(gl.ALWAYS, 1, 255);
35   var buffer = gl.createBuffer();
36   gl.bindBuffer(gl.ARRAY_BUFFER, buffer);
37   gl.bufferData(gl.ARRAY_BUFFER, new Float32Array([
38     20, 20, 120, 20, 20, 120, 120, 120
39   ]), gl.STATIC_DRAW);
40   gl.enableVertexAttribArray(positionLocation);
41   gl.vertexAttribPointer(positionLocation, 2, gl.FLOAT, false, 0, 0);
42   gl.drawArrays(gl.TRIANGLE_STRIP, 0, 4);
43   gl.bindBuffer(gl.ARRAY_BUFFER, null);
44   gl.deleteBuffer(buffer);
45   gl.colorMask(true, true, true, true);
46   gl.stencilFunc(gl.NOTEQUAL, 0, 255);
47 });
48 clippedLayer.on('postcompose', function (evt) {
49   var context = evt.glContext;
50   var gl = context.getGL();
51   gl.disable(gl.STENCIL_TEST);
52 });
53
```

Canvas és WebGL számokban



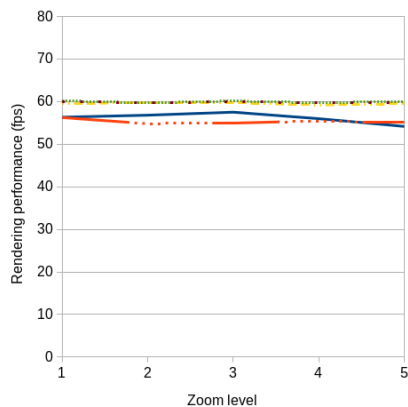
- CPU: Intel i7 7700HQ
- GPU1: NVIDIA GeForce GTX 1050Ti
- GPU2: Intel HD Graphics 630



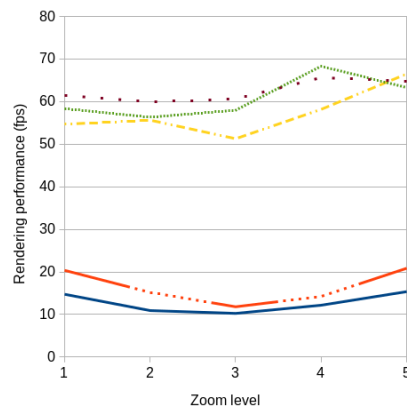
Canvas és WebGL számokban



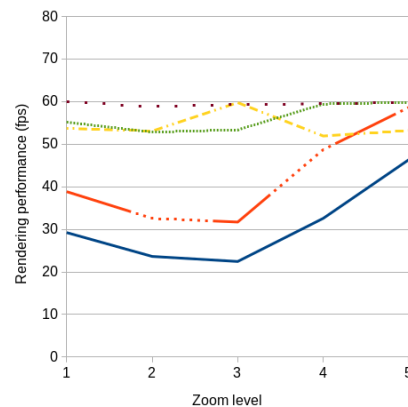
WebGL
animation



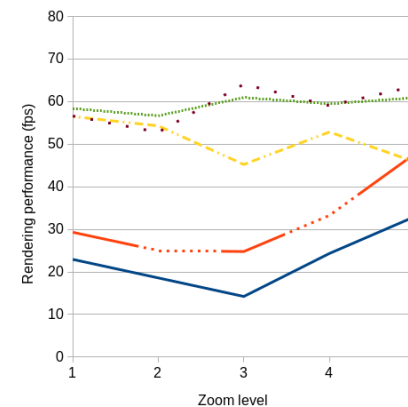
WebGL
draw



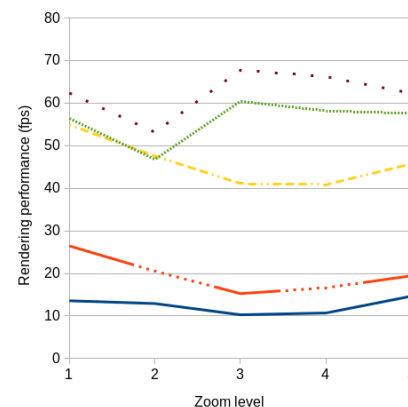
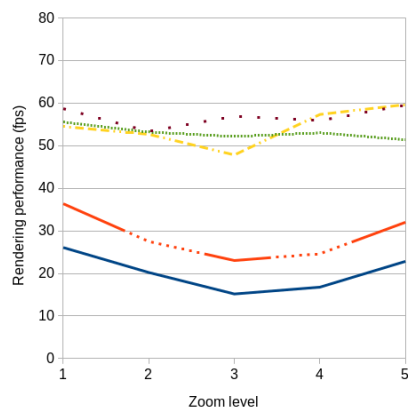
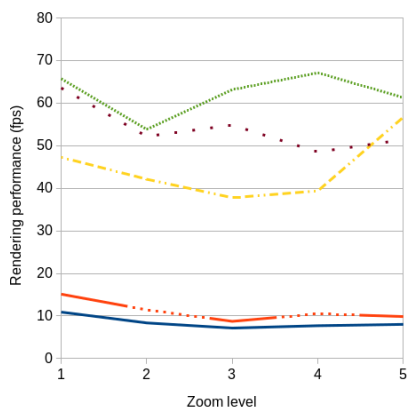
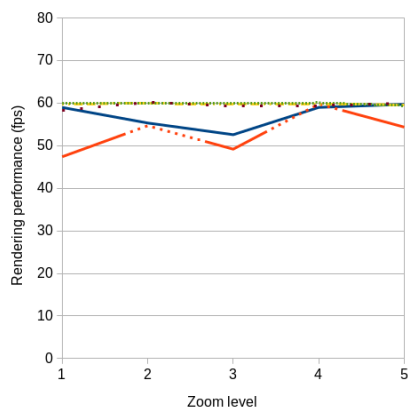
Canvas
animation



Canvas
draw



NVIDIA



Intel

Mi a baj a poligonokkal?



Poly	WebGL	AVG & FPS	%	Canvas	AVG & FPS	%
Total	2838.4	283.84	100.00	372.7	37.27	100.00
Scripting	2827.8	3.523111612	99.63	222.4	26.83123155	59.67
Rendering	1.2		0.04	1.2		0.32
Painting	2.9	W Caching	0.10	2.9		0.78
Other	6.3	54.03	0.22	146.5		39.31
		18.50823617				
Prepare	2713.1		95.59	120.3		32.28
Compose	94.7		3.34	86.1		23.10
	ol.render.webgl.PolygonReplay.triangulate_	1865.3	65.72	CanvasRenderingContext2D	46.8	12.56
	ol.render.webgl.PolygonReplay.processFlatCoordinates_	300.9	10.60			
	ol.render.webgl.PolygonReplay.classifyPoints_	131.9	4.65			
	ol.render.webgl.LineStringReplay.drawPolygonCoordinates	308	10.85			

-

Köszönöm a figyelmet!

