**Content**

[1. Zapocet 2](#_Toc22571361)

[2. History of graphics on web 4](#_Toc22571362)

[1. What is WebGL? 4](#_Toc22571363)

[2. What is rendering? 4](#_Toc22571364)

[3. What is rasterization? 4](#_Toc22571365)

[4. What are pixels and what is PPI (pixel per inch) 5](#_Toc22571366)

[5. What is OpenGL ES SL 5](#_Toc22571367)

[3. Set of steps when writing a WebGL app 6](#_Toc22571368)

[3.1.1. Vertices 6](#_Toc22571369)

[3.1.2. Arrays 6](#_Toc22571370)

[3.1.3. Indices 7](#_Toc22571371)

[3.1.4. Buffers 7](#_Toc22571372)

[3.1.5. Mesh 8](#_Toc22571373)

[3.1.6. Shader Programs 8](#_Toc22571374)

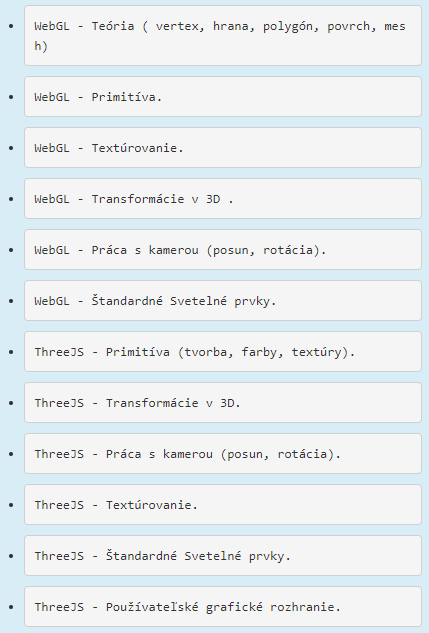
[4. WebGL – Graphics Pipeline / rendering pipeline 11](#_Toc22571375)

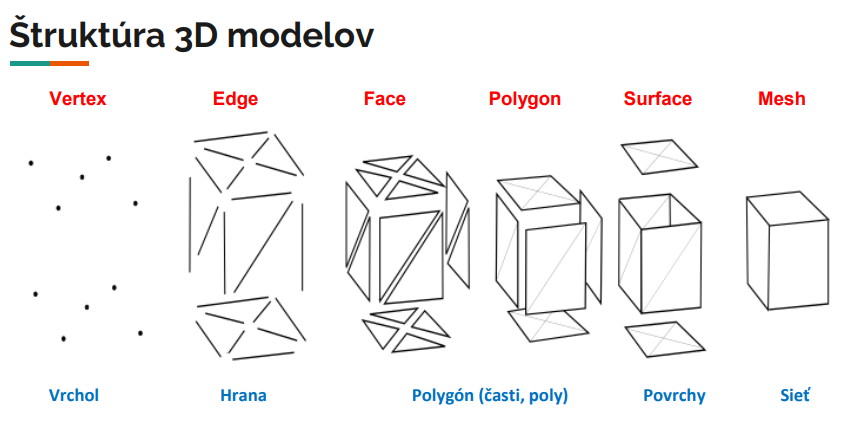
[5. WebGL - Triangles 12](#_Toc22571376)

[1. Shader 12](#_Toc22571377)

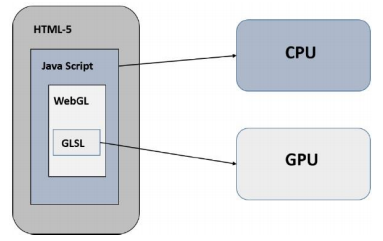
[6. 13](#_Toc22571378)

1. Zapocet





1. History of graphics on web
2. Java applications – Applets and JOGL (Java binding for OpenGL) – require JVM to run.
3. Stage3D API provided by Adobe (Flash, Air) – 2D and 3D capabilities on web browsers as well as IOS + Android. Proprietary software, it was not used as web standard
4. WebGL – 2011, can run without JVM, it’s completely controlled by web browser.
   1. What is WebGL?

WebGL (**Web Graphics Library**) is the new standard for **3D** graphics on the Web, It is designed for the purpose of rendering **2D** graphics and interactive 3D graphics. It is a **JavaScript** **API** that can be used with **HTML5**. WebGL code is written within the **<canvas> tag of HTML5**. It is a specification that allows Internet browsers access to **Graphic Processing Units (GPUs)** on those computers where they were used.

* 1. What is rendering?

Rendering is the process of generating an image from a model using computer programs. In graphics, a virtual scene is described using information like **geometry**, **viewpoint**, **texture**, **lighting**, and **shading**, which is passed through a render program. The output of this render program will be a digital image.

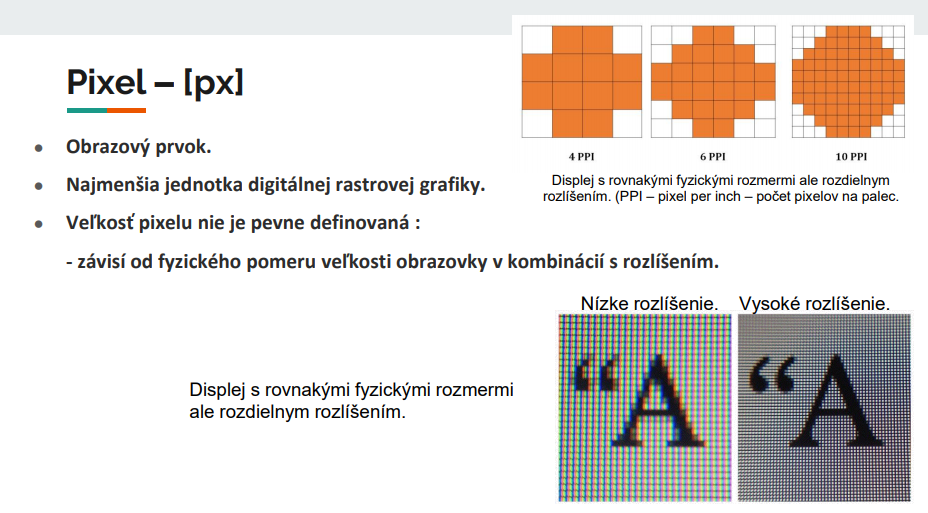
* Software Rendering − All the rendering calculations are done with the help of CPU.
* Hardware Rendering − All the graphics computations are done by the GPU (Graphical processing unit).
* Server-based rendering
* Client-based rendering
  1. What is rasterization?

Rasterization is the process by which most modern display systems turn electronic data or signals into projected images, such as video or still graphics.

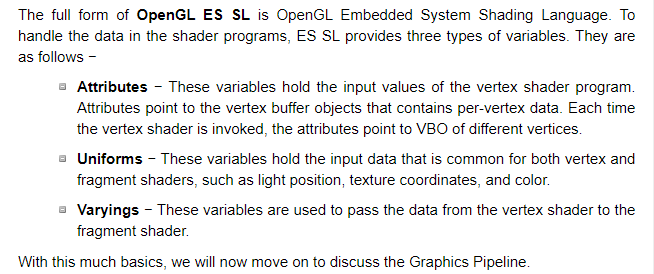
The origin of image rasterization dates back to the early days of television technology. In the mid-twentieth century, televisions typically consisted of cathode ray tube (CRT) monitors, which scanned lines across their display screens that gradually accumulated into complete images.

Rasterized graphics are often compared with image vectors. While rasterization is typically a process of compiling scan lines or pixels on a bitmap, in contrast, vectors incorporate mathematical functions in order to create images based on geometric shapes, angles and curves.

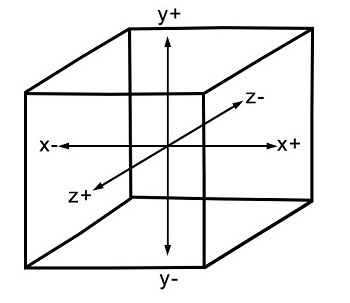
* 1. What are pixels and what is PPI (pixel per inch)



* 1. What is OpenGL ES SL

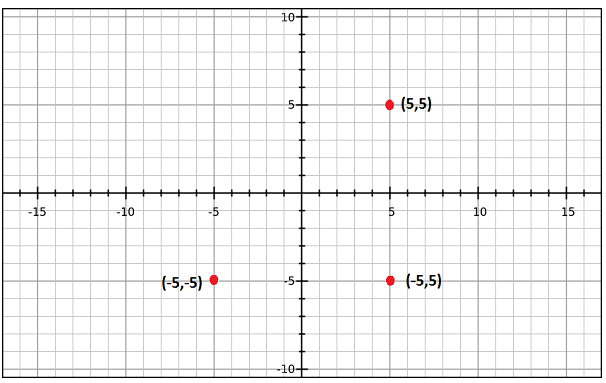


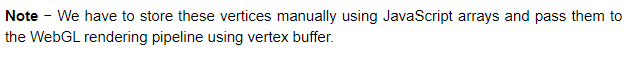
1. Set of steps when writing a WebGL app

WebGL is mostly a low-level **rasterization** API rather than a 3D API. To draw an image using WebGL, you have to pass a vector representing the image. It then converts the given vector into pixel format using OpenGL SL and displays the image on the screen.

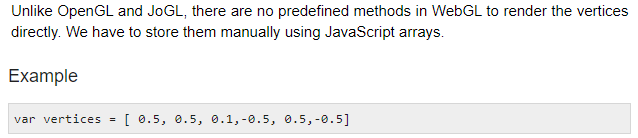
Coordinate System

* + 1. Vertices

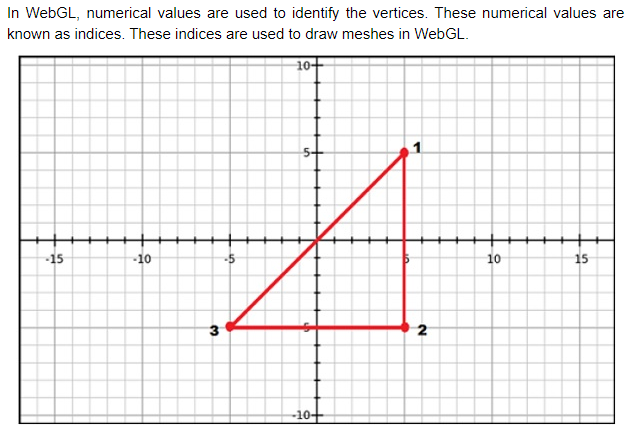
Generally, to draw objects such as a polygon, we mark the points on the plane and join them to form a desired polygon. **A vertex is a point which defines the conjunction of the edges of a 3D object**. It is represented by three floating point values each representing x, y, z axes respectively



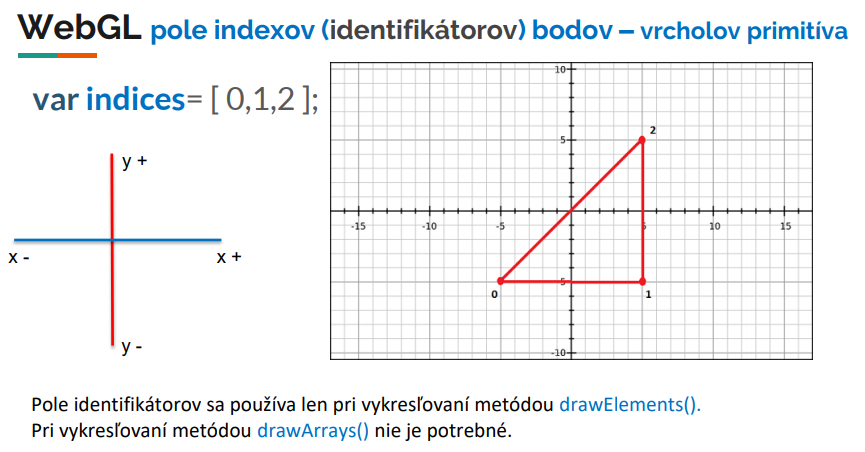
* + 1. Arrays



* + 1. Indices







* + 1. Buffers

Buffers are **the memory areas of WebGL that hold the data**.

* **drawing buffer**
* **frame buffer -** portion of graphics memory that hold the scene data. This buffer contains details such as width and height of the surface (in pixels), color of each pixel, depth and stencil buffers.
* **vetex buffer –** stores data about vertices
* **index buffer.** – stores data about indices

After storing the vertices into arrays, we pass them to WegGL graphics pipeline using these Buffer objects.

* + 1. Mesh
    2. Shader Programs

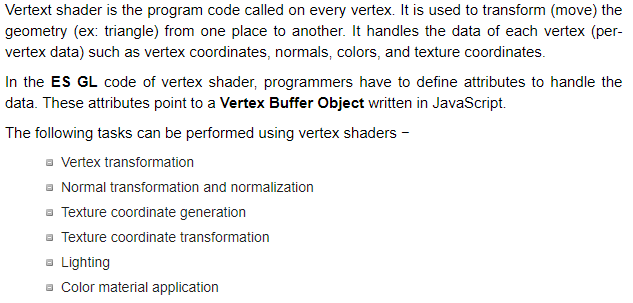
We normally use triangles to construct meshes. Since WebGL uses GPU accelerated computing, the information about these triangles should be transferred from CPU to GPU which takes a lot of communication overhead.

WebGL provides a solution to reduce the communication overhead. Since it uses ES SL (Embedded System Shader Language) that runs on GPU, we write all the required programs to draw graphical elements on the client system using shader programs (the programs which we write using OpenGL ES Shading Language / GLSL).

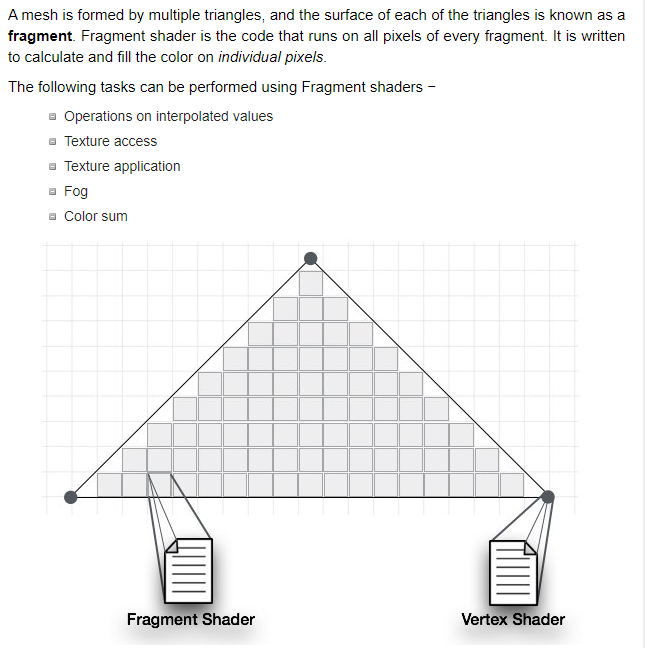
These shaders are the programs for GPU and the language used to write shader programs is GLSL. In these shaders, we define exactly how vertices, transformations, materials, lights, and camera interact with one another to create a particular image.

In short, it is a snippet that implements algorithms to get pixels for a mesh. We will discuss more about shaders in later chapters. There are two types of shaders − Vertex Shader and Fragment Shader.

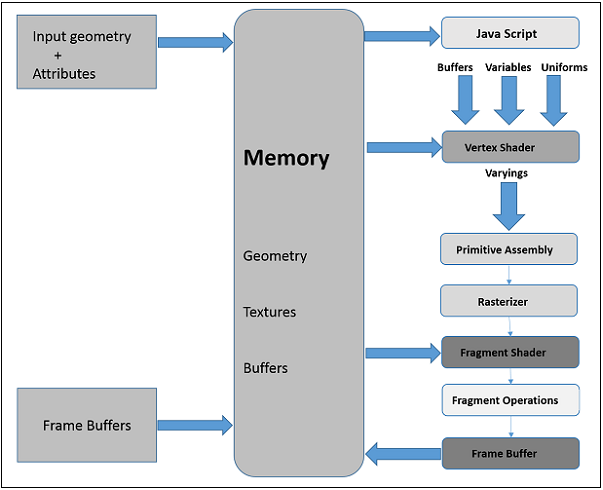
* + - 1. Vertex Shader

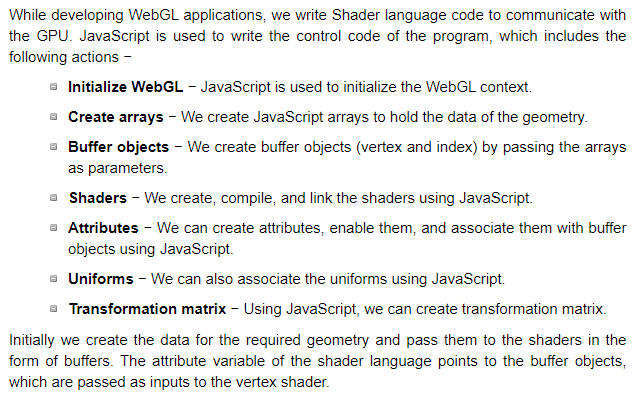


* + - 1. Fragment Shader



1. WebGL – Graphics Pipeline / rendering pipeline





1. WebGL - Triangles

Graphics – OpenGL and DirectX in C#, C++ and Java. ?! – Advanced ☺

Everything is made from polish apples or triangles.

**Rasterization**

Graphics apps – buffers

**Color** **buffer** gl.COLOR\_BUFFER\_BIT

**Depth** **buffer** gl.DEPTH\_BUFFER\_BIT

**Fragment** **Shader** – Give each pixel a color, from the information provided from the rasterizer and uniforms.

**Vertex** **Shader**

* 1. Shader

In computer graphics, a shader is a type of computer program that was originally used for shading (the production of appropriate levels of light, darkness, and color within an image), but which now performs a variety of specialized functions in various fields of computer graphics special effects, or does video post-processing unrelated to shading, or even performs functions unrelated to graphics at all.

Shaders calculate rendering effects on graphics hardware with a high degree of flexibility. Most shaders are coded for a graphics processing unit (GPU), though this is not a strict requirement. Shading languages are usually used to program the GPU rendering pipeline, which has mostly superseded the fixed-function pipeline that only allowed for common geometry transformation and pixel-shading functions; with shaders, customized effects can be used. The position, hue, saturation, brightness, and contrast of all pixels, vertices, and/or textures used to construct a final image can be altered on the fly using algorithms defined in the shader, and can be modified by external variables or textures introduced by the program calling the shader.