WebGL: based on openGL old
rendering interactive 2D and 3D graphics within web browser

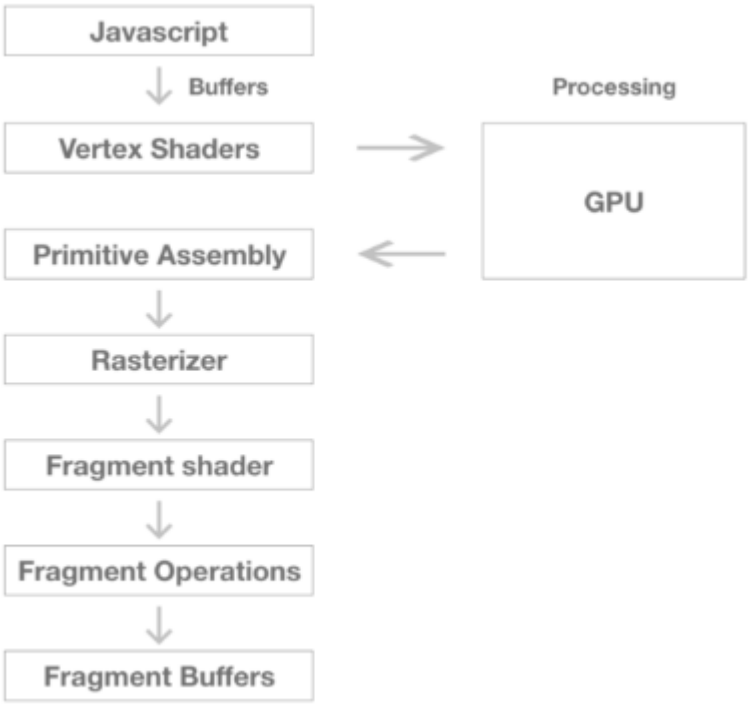
Js API

Integration with <Canvas> of html ,
Main program code in js

const canvas = document.querySelector("#glcanvas");
const gl = canvas.getContext("webgl");

Shaders (that run on GPU) - vertex or fragment- opengl ES shading language (ESSL)- can be done with “3js” , “spline” (frameworks based on webgl)

3D: Vertex - primitive - fragment - pixel



Unreal Engine does not have native support for WebGL
Solution : Unreal Engine project -> HTML5 export option (has its limits) -> Webgl for rendering (need optimisation and adjustments)

Wonder Interactive plugin

Overall performance:

1) Complexity of project
2) Divice and Browser

Epic is working on WebGL exporter for Unreal Engine (only for simple implementation)

HTML5 solutions -> at the limit -> development not headed toward commercial use

Unsuitability -> large-scale projects -> with data rich, high fidelity models.

On Client Side ->

WebGL relies completely on client browser and hardware
Data downloads first -> wait time
Clients storage -> large datasets
Data security -> Additional steps and time
WebGL extensions on client side helps

Low-level graphics -> not so possible to create high-fidelity 3D models

Restrictions on
-size of model
-memory limits

Device Pixel Ratio (DPR) is the ratio between the physical pixel density (can be seen) of a device and its logical pixel density (how many fit into each cm/inch -> depends on setting).

As of Unreal Engine 4.24, support for the HTML5 platform has been migrated out of the engine to a public Platform Extension that can be maintained and improved by community members. Documentation for the HTML5 Platform Extension is hosted alongside the GitHub repository.

Community-supported HTML5 platform extension for Unreal Engine 4

Chatgpt:

Unreal Engine 5 does not have built-in support for exporting projects directly to HTML5

Unreal Engine 4, however, does provide HTML5 support

Unreal Engine 5.2 documentation has no webgl,html5,

Unreal Engine 4 with HTML5 direct packaging option without any other modules or plugins,

Demo game running excellent,
-Controls,
-Lights,
-Physics,

Engine Core compiled to-> Web Assembly (allows browser to run binary code) -> similar performance as native app

Each frame on canvas of html, CPU,GPU usage are low

Step up game with:
Multiple Light sources,
High resolution texture and reflection,

60fps and 1080p resolution with CPU,GPU around 30 to 40%

Next game:

Larger map size,
Foliage.
Volumetric light,
Screen Space reflection,

Limitation started ;
Project smooth but visual effects (some) are removed,

Volumetric light and some reflections are missing,

Drawbacks:

Massive download size,(near 1GB - 100s MB per project)

-So only small game,product or design demo,

Engine Features are limited:

Scene with realistic foliage with high texture
Engine cant support extremely high texture material -> reverted to ugly texture.

Not all projects can be run,it needs to support HTML5 from first -> struck at packaging itself,

From version 4.24 unreal stops supporting support for html5 and changed it to extension,

Download unreal engine source code from github,and combine it with HTML5 extension,and recompile engine editor - pain in the ass

Developer community now has to take care of this instead,

If we are not using latest features like ray-tracing,
Download older unreal engine to support HTML5.