

WebGL i napredak 3D grafike na internetu WebGL and 3D graphic advances on the Internet

Domagoj Trojko, mag. ing. graph. tech.
Graditeljska škola, Športska 1, 40000 Čakovec
dtrojko@gmail.com

Sažetak: Razvojem web tehnologija, brzine prijenosa podataka kao i napretkom računalnoga hardvera, stvorili su se uvjeti za bogatije korisničko iskustvo na webu. Velikani računalne industrije odlučili su stvoriti radnu grupu koja će razviti tehnologiju za prikaz interaktivnih 3D i 2D grafika visokih performansi koje će se moći prikazati u bilo kojem kompatibilnom internetskom pregledniku. Rezultat radne skupine je WebGL, JavaScript sučelje za programiranje aplikacija, tj. tehnologija koja koristi resurse grafičkoga procesora uređaja na kojem se pokreće i ne zahtjeva instalaciju nikakvih dodataka na uređaj. Tim činom je pokrenuta revolucija koja je omogućila bogata korisnička iskustva u okviru internetskih preglednika. Kako se biblioteka, s obzirom na trenutnu brzinu razvoja web tehnologija, ne može nositi s najnovijim tehničkim zahtjevima, izrađen je WebGL 2.0 standard, te je u procesu uvođenje i najnovijega WebGPU standarda.

Rad s WebGL-om zahtijeva široko i duboko poznavanje programiranja u JavaScriptu i dobro poznavanje matematike, jer WebGL spada u skupinu programskih jezika niske razine pošto mora biti u mogućnosti komunicirati s hardverom uređaja. Stoga se, uskoro nakon pojave WebGL-a, krenulo u izradu biblioteka koje olakšavaju rad s WebGL-om. Najpopularnija biblioteka, u ovom trenutku, smatra se three.js koja znatno olakšava rad s WebGL-om pošto dolazi s rješenjima koja brinu za izradu scena, sjena, tekstura, materijala, matematiku za prikaz 3D modela i ostale stvari za koje WebGL traži od programera da izradi sam.

Ključne riječi: *3D, grafika, WebGL, WebGPU, three.js, JavaScript*

Abstract: With the development of web technology, the speed of data transfer as well as with advances in computer hardware technology, the requirements for a richer user experience on the web are here. Big names in the computer industry have decided to form a working group which will develop the technology for displaying interactive 3D and 2D graphics

of high performance that can be displayed in any compatible browser. The result of the working group is WebGL, a JavaScript API – a technology which uses the resources of the devices GPU and doesn't require any additional plug-ins. This was how a revolution was started which enabled rich user experience through browsers. This library, regarding the evolution speed of web technologies, can't cope with the newest technical requirements, so a WebGL 2.0 standard followed, and the newest WebGPU standard is in the process of implementation.

Working with WebGL implies a wide and deep understanding of JavaScript development as well as good mathematic knowledge, because WebGL is a low-level programming language because it must communicate with the device's hardware. That's why, soon after WebGL was published, libraries were developed which facilitate working with WebGL. Most popular library, at this time, is three.js which makes working with WebGL significantly easier, because it takes care of scene, shadow, texture, material, math and other essential things needed for 3D model display, and WebGL requires developers to do these things by themselves.

Keywords: *3D, graphics, WebGL, WebGPU, three.js, JavaScript*

1. Uvod

WebGL (en. Web Graphics Library) tj. biblioteka grafika za internet, je JavaScript sučelje za programiranje aplikacija (API) koje može renderirati interaktivne 3D i 2D grafike visokih performansi unutar bilo kojeg kompatibilnoga preglednika za internet bez potrebe korištenja dodatnih priključaka (Mozilla developer network, 2023). WebGL je API niske razine gdje je i za najjednostavnije radnje nužno pisati mnogo linija koda, a nije naodmet i poznavanje geometrije, trigonometrije i linearne algebre (HTML5DevConf & IoTaconf, 2014). U tom procesu nužno je učitati, kompilirati i povezati shadere, postaviti varijable koje se prosljeđuju u shadere i matematiku matrica kako bi oblike mogli animirati (Khronos.org, 2011).

WebGL za renderiranje 3D grafike daje poveznicu s grafičkim procesorom (GPU) uređaja na kojem se pokreće aplikacija. Programeri koji koriste WebGL očekuju kako će, jednom programirana, aplikacija raditi podjednako dobro na cjelokupnom spektru hardvera, radilo se o stolnim računalima ili mobilnim uređajima. Pristup u kojem unutar aplikacije ne mora biti predviđen način na koji će pojedini uređaji reagirati na sam kôd, već to za njih radi sam WebGL, pristup je koji pogoduje njegovom raširenom korištenju. On upravlja razlikama među operacijskim sustavima, dostupnim driverima, grafičkim procesorima ili internetskim

preglednicima (Cozzi, 2016).

2. WebGL i razvoj

Tehnologija za prikazivanje 3D grafika u internetskim preglednicima počela je svoj razvoj u eksperimentima zaposlenika Mozille, Vladimira Vukićevića, koji je radio unutar HTML elementa *canvas* i 2006. godine razvio Canvas 3D prototip. Za svoj eksperiment koristio je OpenGL API kako bi ga mogao pokrenuti na internetu. Iz toga procesa neprofitna organizacija Khronos grupa 2009. godine pokrenula je WebGL radnu skupinu u koju su se uključili Apple, Google, Intel, Mozilla i drugi. To je 2011. godine rezultiralo prvom, WebGL 1.0 specifikacijom, specifikacija WebGL 2.0 slijedila je 2017. Obje specifikacije počivaju na OpenGL ES specifikacijama, WebGL 1.0 na OpenGL ES 2.0, a WebGL 2.0 na OpenGL ES 3.0.

Neko vrijeme, najveća prepreka usponu WebGL bio je Adobe Flash, multimedijaska platforma koja je u tom razdoblju bila prisutna na većini PC računala i koja je zahtijevala instaliranje dodatnoga priključka (en. plug-in). Flash je bio ograničen na stolna računala i zahtijevao je znatne računalne resurse, tako da je dolaskom mobilnih uređaja sposobnih za pristupanje internetu polako počeo gubiti na važnosti i primjeni. Steve Jobs je u svojem otvorenom pismu iz 2010. godine (Jobs, 2010) opisao probleme s tehnologijom iza Flasha i otvoreno zagovarao korištenje HTML-a, CSS-a i JavaScripta, specifično HTML5 tehnologije, koja je na kraju Flash u potpunosti istisnula. Od kraja 2020. godine Adobe više ne nudi podršku za Flash (Adobe, 2021).

Napredak računalne tehnologije, grafičkih kartica i mogućnosti internetskih preglednika znatan je kako na mobilnim uređajima tako i na računalima, pa za mnoge mogućnosti WebGL 2.0 nema odgovarajuće odgovore. Od 2017. godine, W3C grupa počela je rad na novom internet standardu za 3D grafiku te je standard nazvala WebGPU. U travnju 2023. godine Google je objavio kako će Chromium preglednici i Google Chrome preglednik dolaziti s omogućenom potporom za WebGPU. U vrijeme pisanja ovoga rada, specifikacija ima status radnoga nacrtu i omogućena je u Chrome, Edge i Opera preglednicima za osobna računala, a za Firefox se može omogućiti intervencijama korisnika. Ne postoji mogućnost korištenja na nijednom mobilnom pregledniku (Deveria, 2023).

3. Što je potaknulo razvoj 3D grafike na webu?

Kao i većina stvari u današnjem svijetu, razvoj 3D grafike na webu potaknuo je razvoj

tehnologije. No, taj razvoj nije došao samo iz jednog smjera, a i nije sav razvoj bio isključivo uvjetovan napretkom tehnologije.

Prvo, i najočitije, napredak samoga računalnog hardvera bio je presudan za poticanje napredaka na internetu. Najbitnija stvar, kod prikazivanja 3D grafike, svakako je grafički procesor tj. GPU. Gotovo svaki veliki napredak u razvoju grafičkih procesora, pratio je i novi vizualni pristup izradi web stranica (Peddie, 2022). Treće i četvrto doba GPU-a se događa u razdoblju intenzivnog razvoja Web 2.0, peto doba se poklapa s razvojem naprednih, responzivnih rasporeda vizualnih elemenata, dok se u šestom razdoblju intenzivno razvija 3D grafika na webu i animacije .

Drugi napredak odnosi se na brzinu pristupa internetu, koji je nužan za prikaz kompleksnih grafika u vremenu prihvatljivom za prosječnoga korisnika. Povećanjem brzine interneta, razvojni programeri za web su počeli istraživati granice mogućih prikaza i pomicati ih. Prema dostupni podacima, brzina interneta u SAD-u je od kraja 2007. godine i brzine od 3.67 Mbp/s, do početka 2017. godine porasla na 18.75 Mbp/s, što predstavlja peterostruko povećanje (Petroc, 2023). U posljednje tri godine, od 2020. godine do 2023.godine, globalna brzina interneta povećana je s 24.8 Mbp/s do 46.8 Mbp/s (Armstrong, 2023), dok je 2021. godine u Japanu postignuta brzina prijenosa podataka od 319 (Tbit/s) (FutureTimeline.net, 2021). S takvim brzinama prijenosa podataka, web stranice ne moraju više prikazivati optimizirane, statične fotografije za što brži prikaz, pošto veći dio ciljane publike s lakoćom može otvoriti stanice s kompleksnijim multimedijским sadržajem, uključujući video i 3D grafiku.

Internetski preglednici softver su koji je u konstantnoj fazi razvoja. Dolaskom tehnologija koje danas znamo pod nazivom HTML5 i CSS3, a i ubrzanim razvojem JavaScript programskoga jezika kojeg prati sve veći broj biblioteka i razvojnih okružja, korisnici, kao i developeri, očekuju da se internetske stranice očitavaju brzo. To internetskim preglednicima predstavlja veliki izazov pošto je na internetu dostupna ogromna količina podataka i informacija. Ono što bi nekad moralo biti kompilirano u zasebnu aplikaciju, danas se u internetskim preglednicima kompilira tzv. tehnikom Just-In-Time, gdje se JavaScript funkcije kompiliraju točno u onom trenutku kad to web stranica ili aplikacija zahtjeva. Za takve procese preglednici stalno unapređuju svoje strojeve za skripte. Također razvoj se fokusira i na strojeve za renderiranje. U tom području je napravljen najveći napredak te je omogućen rad sa sve kompleksnijim 2D i 3D grafikama (Howse, 2020).

Razlozi napretka koji se ne tiču napretka same tehnologije svakako su tržišna natjecanja i

zahtjevi korisnika. Tržište na internetu raste nevjerojatnom brzinom i već je neko vrijeme nemoguće ignorirati internet kao medij promocije i prodaje. Sasvim je razumljivo da u tom tržišnom natjecanju, nastaju internetske stranice s inovativnim i naprednim rješenjima koje zatim druge tvrtke pokušavaju kopirati ili nadmašiti.

Korisnici također postaju sve zahtjevniji te traže imerzivna iskustva koja im se nude u ostalim medijima, kao i korisničko iskustvo koje je zanimljivo, zabavno i intuitivno.

4. Kako se implementira 3D grafika

Kod prikazivanja 3D grafike, u svakom slučaju veliku ulogu je igrala sve veća dostupnost softvera za 3D modeliranje, poput Blendera, Maya, Cinema 4D. Kako je dostupan sve većem broju korisnika koji lako uče te programe pomoću javno dostupnih multimedijalnih sadržaja, sve veći broj je 3D modela koji su besplatno dostupni svakom tko ih želi koristiti u svojim projektima.

Mnogo 3D animacija i manipulacija može se izraditi već pomoću CSS jezika, poput animacija u kojima se manipulira objektima u 3D prostoru. Pomoću CSS-a se objektima može rotirati, promijeniti veličina ili pozicije. Već pomoću tih tehnika se mogu izraditi impresivne 3D animacije (Shukla, 2023).

Ukoliko se neki 3D efekti ne mogu postići pomoću CSS jezika, većina se može postići pomoću JavaScripta, tj. WebGL API. Ipak, WebGL je težak za korištenje. Svi 3D modeli sačinjeni su od trokuta, kojih je nekad nekoliko tisuća, a nekad milijuna. Ukoliko se želi iscrtati jedan trokut u 3D prostoru pomoću WebGL API, potreban je veći broj linija kôda, a kad se trebaju dodati perspektiva i osvjetljenje, te se zatim sve animirati, programerski zadatak postaje znatno teži. Razvoj 3D iskustva u nativnom WebGL kôdu naporan je posao koji od programera traži znatnu količinu znanja JavaScripta, kao i poznavanje geometrije i matematičkih principa. Zsigurno se na internetu ne bi pojavljivale nove tehnike i zadivljujuća 3D iskustva kad bi jedini alat za izradu istih bio WebGL. Stoga su se uskoro pojavile biblioteke koje olakšavaju rad s WebGL poput BabylonJS, PlayCanvas i Three.js.

4.1. Osnove WebGL tehnologije

Tehnologija koja stoji iza WebGL nešto je kompleksnija nego uobičajene web tehnologije pošto je osnovna namjena korištenje resursa grafičke kartice uređaja. Niska razina programskoga jezika znači da lakše i brže može obavljati mnoštvo kalkulacija koje su potrebne za renderiranje kompleksnog 3D sadržaja.

Korištenjem JavaScript jezika, za svaki trokut u 3D modelu, određuje se gdje će on biti stvoren u prostoru, kako će izgledati njegova boja, tekstura, sjene i sl. Tada se te informacije prosljeđuju u GPU, koji te informacije procesira i vraća pogled na scenu. Takav proces se naziva cjevovod renderiranja (Caballero, 2011).

U tom procesu prvo se izrađuju nizovi vrhova (eng. vertex array) koji sadrže informacije o položaju pojedinoga vrha 3D objekta u prostoru. Te podatke preuzima GPU koji ih šalje u međuspremnik vrhova iz kojeg ih propušta kroz shader vrhova. Shader vrhova programer može izraditi sam ili koristiti onaj koji mu je dostupan u WebGL biblioteci. Nizovi indeksa, rezultat toga procesa, upravljaju kasnijim spajanjem tih vrhova u trokute (Caballero, 2011).

Rasterizator uzima svaki od tih trokuta. Za svaki trokut smješten van zaslona izrezuje ga i zanemaruje, a od ostalih vidljivih dijelova izrađuje raster prema podacima zapisanim u shaderu vrhova i te podatke primjenjuje na površinu svakoga trokuta (Caballero, 2011).

Dijelom procesa koji je najosjetljiviji na performanse uređaja upravlja shader fragmenata. On uzima generirane podatke pretvorene u piksele te za svaki piksel daje podatke o boji i dubini i zapisuje ih u međuspremnik fragmenata. U tom procesu se izračuni rade za svaki piksel posebno, a mogu se provoditi i operacije poput mapiranja tekstura i osvjetljenja. Međuspremnik fragmenata posljednji je dio cjevovoda renderiranja. Za shader fragmenata, kao i za shader vrhova, programer može sam napisati kôd ili koristiti onaj koji postoji u WebGL biblioteci (Caballero, 2011).

Kod 1. Kod za upravljanje učitanim teksturama u WebGL

```
function handleLoadedTexture(texture) {
    gl.pixelStorei(gl.UNPACK_FLIP_Y_WEBGL, true);
    gl.bindTexture(gl.TEXTURE_2D, texture);
    gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE,
texture.image);
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, gl.LINEAR);
    gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER,
gl.LINEAR_MIPMAP_NEAREST);
    gl.generateMipmap(gl.TEXTURE_2D);

    gl.bindTexture(gl.TEXTURE_2D, null);
}
```

Izvor: GitHub – korisnik gpjt

4.2. Najpopularnija WebGL biblioteka Three.js

Kako ovakav pristup traži mnogo programerskoga, ali i matematičkoga, znanja, u većini slučajeva gdje se koristi WebGL u izradi web stranica ili aplikacija razvojni programeri koriste neku od već spomenutih biblioteka. Three.js je, trenutno, najpopularnija biblioteka web developere i izradu AR/VR sadržaja (three.js, 2024).

Popularnost three.js proizlazi iz toga što programiranje u WebGL-u iziskuje mnogo vremena ukoliko se želi izraditi nešto konkretno što ima primjenu na web stranici. Three.js dolazi s rješenjima koja brinu za izradu scena, sjena, tekstura, materijala, matematiku za prikaz 3D modela i ostale stvari za koje WebGL od programera traži da izradi sam. U članku (Caballero, 2011) objavljenom na web stranici za web developere preglednika Opera, detaljno je prikazano koliko više linija koda je potrebno za izradu 3D modela Mjeseca u WebGL-u (GitHub - gpjt, 2011) nego kad se izradi pomoću neke biblioteke (Github - philogb, 2011).

Kod three.js, ukoliko se želi izraditi jednostavan geometrijski 3D oblik, potrebno je prvo učitati three.js. Zatim je u HTML-u potrebno otvoriti canvas element u kojem se zatim ucrtava oblik. Za renderiranje se učitava WebGLRenderer. Sljedeći korak je stvaranje kamere kojoj se trebaju pridodati nekoliko elemenata poput, područja pogleda, aspekta, prostora ispred kamere koji će biti renderiran. Na taj način se stvara krnji stožac koji predstavlja scenu unutar koje se iscrtavaju objekti. Također se mora izraditi scena, definirati geometrija objekta i definirati boja objekta. Zadnji korak za izradu objekta je izrada mreže kojoj se definira geometrija, materijal, pozicija, orijentacija i veličina u sceni. Sve ove radnje pretpostavljaju pisanje jedne do pet linija jednostavnoga kôda. Kad se tom objektu doda animacija poput rotacije i osvjetljenje, moguće je vidjeti 3D prirodu objekta (three.js, 2024).

Kod 2. *Kod za očitavanje svih svojstava aplikacije u three.js*

```
var gl = app.gl,  
    program = app.program,  
    scene = app.scene,  
    canvas = app.canvas,  
    camera = app.camera,  
    //get light config from forms  
    lighting = $id('lighting'),  
    ambient = {
```

```
r: $id('ambientR'),  
g: $id('ambientG'),  
b: $id('ambientB')  
},  
direction = {  
  x: $id('lightDirectionX'),  
  y: $id('lightDirectionY'),  
  z: $id('lightDirectionZ'),  
  
  r: $id('directionalR'),  
  g: $id('directionalG'),  
  b: $id('directionalB')  
};
```

Izvor: GitHub – korisnik philogb

5. Zaključak

Napretci u brzini GPU-a, brzini prijenosa podataka na internetu, kao i sve zahtjevniji korisnici i sve veća bitka na tržištu, ubrzali su i unaprijedili primjenu 3D grafike na web stranicama. Glavni alat za razvoj te web tehnologije je WebGL, alat koji je svoju prvu inačicu doživio 2011. godine, drugu 2017. godine, a nedavno se počela ograničeno primjenjivati i WebGPU tehnologija, nasljednik WebGL-a .

WebGL je programski jezik niske razine koji mora komunicirati s hardverom uređaja na kojem se pokreće, bio on stolno ili prijenosno računalo ili mobilni uređaj. Zadaća WebGL-a je što učinkovitije iskoristiti grafički procesor uređaja za prikazivanje 2D i 3D grafike na tom uređaju i to bez potrebe za instaliranjem dodatnih priključaka. Još jedna velika zadaća WebGL-a je upravljanje razlikama koje postoje između različitih operacijskih sustava, različitih grafičkih kartica ili internetskih preglednika.

Rad u WebGL-u zahtjeva veliko programersko znanje, kao i znanje matematike, i potrebno je uložiti mnogo vremena i ispisati mnogo linija kôda kako bi se napravile najjednostavnije stvari u web pregledniku. Zato su se razvile biblioteke koje znatno olakšavaju rad s WebGL-om i programiranje u njemu. Trenutačno najpopularnija biblioteka je three.js koja je veoma popularna kod web developera koji izrađuju web stranice i aplikacije koje zahtijevaju rad s 3D objektima.

3D grafika na webu je u veoma uzbudljivoj fazi s velikim napretcima i znatnim brojem novih web stranica koje miču granice onoga što je danas moguće u prikazu 2D i 3D grafike. Neke druge tehnike, poput CSS animacija, također pridonose tom napretku, ali najbitniji alat je i dalje WebGL.

Literatura

1. Adobe, 2021. *Adobe Flash Player End of Life*. [Mrežno] Available at: <https://www.adobe.com/products/flashplayer/end-of-life.html> (09.02.2024).
2. Armstrong, M., 2023. *Statista*. [Mrežno] Available at: <https://www.statista.com/chart/31075/global-average-internet-download-speed/> (12.02.2024).
3. Caballero, L., 2011. *An Introduction to WebGL - Part 1*. [Mrežno] Available at: <https://dev.opera.com/articles/introduction-to-webgl-part-1/> (15.02.2024).
4. Cozzi, P., 2016. *WebGL Insights*. Boca Raton: Taylor & Francis Group.
5. Deveria, A., 2023. *WebGPU | Can I Use*. [Mrežno] Available at: <https://caniuse.com/webgpu> (10.02.2024).
6. FutureTimeline.net, 2021. *Japan sets new Internet speed record*. [Mrežno] Available at: <https://www.futuretimeline.net/blog/2021/07/19-future-internet-speed-predictions.htm> (12.02.2024).
7. GitHub - gpjt, 2011. *GitHub*. [Mrežno] Available at: <https://github.com/gpjt/webgl-lessons/blob/master/lesson11/index.html> (15.02.2024).
8. Github - philogb, 2011. *GitHub Gist*. [Mrežno] Available at: <https://gist.github.com/philogb/818017> (15.02.2024).
9. Howse, B., 2020. *The 2020 Browser Battle*. [Mrežno] Available at: <https://www.anandtech.com/show/16078/the-2020-browser-battle-surfing-with-speed> (13.02.2024).
10. HTML5DevConf & IoTaconf, 2014. *HTML5DevConf May 2014: Nick Desaulniers, Mozilla: Raw WebGL*. [Mrežno] Available at:

https://www.youtube.com/watch?v=H4c8t6myAWU&t=472s&ab_channel=HTML5DevConf%26IoTaconf (07.02.2024).

11. Jobs, S., 2010. *Steve Jobs - Thoughts on Flash*. [Mrežno] Available at: <https://newslang.ch/wp-content/uploads/2022/06/Thoughts-on-Flash.pdf> (08.02.2024).
12. Khronos.org, 2011. *Getting Started - WebGL Public Wiki*. [Mrežno] Available at: https://www.khronos.org/webgl/wiki/Getting_Started (07.02.2024).
13. Mozilla developer network, 2023. *WebGL: 2d and 3D graphics for the web - Web APIs*. Available at: https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API (06.02.2024).
14. Peddie, J., 2022. *The History of The GPU - Eras and Environment*. Cham: Springer.
15. Petroc, T., 2023. *Statista*. [Mrežno] Available at: <https://www.statista.com/statistics/616210/average-internet-connection-speed-in-the-us/> (12.02.2024).
16. Shukla, M., 2023. *The Rise of 3D Graphics in Web Design*. [Mrežno] Available at: <https://manoj-shu100.medium.com/the-rise-of-3d-graphics-in-web-design-techniques-and-tools-fd5e05ef3084> (14.02 2024).
17. three.js, 2024. *Fundamentals - three.js manual*. [Mrežno] Available at: <https://threejs.org/manual/#en/fundamentals> (15.02 2024).