**VIZUALIZACIJA ZVUKA**

Projektni rad iz kolegija

„Vizualizacija podataka“

Matija Lekić

1. god. DRD

**OPIS PROJEKTNOG ZADATKA**

Cilj ovog projektnog zadatka je kreirati vizualizaciju frekvencijskog raspona zvučne datoteke u stvarnom vremenu, odnosno, preuzimati vrijednosti frekvencijskih pojaseva za vrijeme reproduciranja zvučne datoteke i prikazivati ih dinamično u obliku stupčastog grafa. Tehnologija izvođenja je JavaScript biblioteka D3.js, koja je pogodna upravo za brzu i jednostavnu izradu vizualizacija za web. Za izradu stupčastog grafa korišten je SVG, vektorski zapis slike, zasnovan na XML jeziku. Zbog uporabe zvučne datoteke u web aplikaciji, potrebno je korištenje i Web Audio API-a, koji je raznolik sustav za obradu zvuka na web-u, podržan u većini preglednika. Za estetsku obradu vizualizacije, korišten je CSS u posebnoj datoteci te ponešto JavaScript-a u zasebnoj datoteci za vremenski prikaz zvučne datoteke.

**TEHNIČKO I PROGRAMSKO RJEŠENJE**

Problem reproduciranja zvučne datoteke na Web-u lako se može riješiti korištenjem Web Audio API-a, koji omogućuje rukovanje audio operacijama unutar elementa *audio context.* Prvo je postavljen *audio* element i pridodan mu je *source*, odnosno, putanja zvučne datoteke koju se želi reproducirati, u ovom slučaju specifična pjesma u Mp3 formatu. Zatim su stvorena dva gumba, jedan za pokretanje i drugi za pauziranje pjesme, a *audio* element im je predan preko *id*-a. Potrebno je doći do podataka o samoj pjesmi, a to je moguće stvaranjem *audio context*-a. Naknadno je dodan već postojeći *audio* element i preoblikovan u *MediaElementSource*, što omogućuje manipulaciju audio zapisa na izvoru te je stvoren *Analyser* preko kojega je moguće dobiti frekvencijske podatke. Na kraju je preusmjeren izlaz *audio* elementa na ulaz u *Analyser* , koji je preusmjeren na odredište, odnosno zvučnike sustava. Određena je minimalna i maksimalna vrijednost *gain*-a, koji će *Analyser* promatrati te će bilježiti vrijednosti isključivo u tom rasponu.

Frekvencijske vrijednosti koje će biti dohvaćene, moraju se privremeno pohraniti negdje dok se ne prikažu i dok ih ne zamijene nove. U tu svrhu stvoreno je polje od 128 članova. Funkcija *getByteFrequencyData* koja je korištena za dohvaćanje frekvencijskih vrijednosti, vraća kao rezultat polje vrijednosti koje imaju raspon od 0 do 255. Ovisno o veličini polja u koje se kopiraju te vrijednosti, *Analyser* odabire dvostruko veću vrijednost za *fft size*, kako bi se inače polje vrijednosti od 1024 člana ravnomjerno rasporedilo na željeni broj odsječaka. Kako stvoreno polje ima 128 članova, tako će i konačni graf sadržavati 128 stupaca. Koristeći već spomenutu funkciju, korisnički stvoreno polje će se popunjavati i dinamički predavati vrijednosti grafu i mijenjati y vrijednost stupaca.

Za prikaz stupčastog grafa korištena je D3 biblioteka i SVG elementi. Stvoreno je onoliko stupaca kolika je veličina korisnički stvorenog polja koje će primiti frekvencijske podatke, a razmak između svakog stupca je korisnički postavljen. Popunjavanje y vrijednosti stupca je, kako je i već rečeno, izvršeno dinamički, kako se osvježava polje frekvencijskih vrijednosti, s time da ovisno o vrijednosti *gain*-a svakog frekvencijskog pojasa nijansa boje ispune pojedinog stupca varira od tamne za niske vrijednosti do svijetle za visoke vrijednosti. Konačno, grafu su dodane x i y os te oznake na osima. Budući da je prethodno definiran minimalni i maksimalni iznos *gain*-a u decibelima kojeg će *Analyser* uzimati u obzir, to je i raspon vrijednosti koji će se prostirati uz y os. Međutim, kako su vrijednosti frekvencijskih pojasa kompresirani sa 1024 na 128 odsječaka, potrebno je izračunati koliko je širok frekvencijski pojas kojeg predstavlja jedan stupac u grafu. Prikladna formula glasi *N \* samplerate/fftsize* , gdje je *samplerate* standardnih 44100 Hz, *fftsize* je dvostruko veći od veličine polja u kojega se kopiraju frekvencijske vrijednosti, odnosno, iznosi 256, dok je *N* jednak 1 jer se promatra raspon jednog stupca. Kada se to uvrsti, dobije se rezultat od približno 170 Hz za jedan stupac. Ako se ta vrijednost pomnoži s brojem stupaca, ukupni raspon grafa na x osi iznosi 22050 Hz. Budući da se radi o frekvenciji, oznake na x osi nisu ravnomjerno raspoređene kao na y osi, već su postavljene na okrugle vrijednosti. Kako bi se vrijednosti *gain*-a pojedinog stupca mogle bolje pratiti, iscrtane su vodoravne linije na razini oznaka na y osi.

Dodatno uređivanje aplikacije kako bi izgled bio prihvatljiv i ugodan oku učinjen je u CSS-u. Također, u CSS-u su uređivani prethodno stvoreni gumbi, osi te linije mreže na pozadini grafa.

**KORIŠTENE TEHNOLOGIJE**

JavaScript (D3.js, SVG)

*-materijali s laboratorijskih vježbi kolegija „Vizualizacija podataka“*

Web Audio API

*http://code.tutsplus.com/tutorials/the-web-audio-api-what-is-it--cms-23735*

*https://developer.mozilla.org/en-US/docs/Web/API/Web\_Audio\_API*

*http://www.html5rocks.com/en/tutorials/webaudio/intro/*

*http://stackoverflow.com/questions/14789283/what-does-the-fft-data-in-the-web-audio-api-correspond-to*

CSS

*http://www.w3schools.com/css*