

Reprodukcija prostorskega zvoka: Web Audio API

Tjaša Mlakar¹, Jaka Sodnik¹, Franc Policardi^{1,2}

¹Univerza v Ljubljani, Fakulteta za elektrotehniko,
Tržaška cesta 25, 1000 Ljubljana

²Raziskovalni Inštitut za nove tehnologije in energetiko (R.I.N.T.E.) d.o.o.,
Stritarjeva 6a, 4000 Kranj, Slovenija

E-pošta: tm9075@student.uni-lj.si, jaka.sodnik@fe.uni-lj.si, franc.policardi@lucami.fe.uni-lj.si

Abstract. *This article discusses localization and reproduction of spatial sound and its rendering in web browsers. The main part presents the complex Web Audio API tool and its interfaces and associated attributes for creating spatial sound by processing sound sources, while the practical part describes and presents the implementation of the tool in a simple user interface for the purpose of testing its functionality.*

1 Uvod

Študija o zaznavanju zvoka pri ljudeh združuje znanje in znanstvene metode psihologije in psihoakustike. Psihoakustika je veja psihofizike, ki vključuje preučevanje fizične zgradbe človeškega ušesa, kanale za širjenje zvoka in njihove prenosne funkcije ter človeško subjektivno zaznavanje zvoka in njegovo interpretacijo [1].

1.1 Lokalizacija prostorskega zvoka

Človeški slušni sistem je sposoben določiti točno lokacijo izvora zvoka. Ta proces imenujemo lokalizacija in je bistvenega pomena za orientacijo v prostoru, komunikacijo z več osebami hkrati, vožnjo z avtomobilom, itd. Ko govorimo o položaju izbranega vira zvoka v prostoru, običajno definiramo njegov položaj glede na položaj poslušalca. Za določitev te relacije je najprimernejši tridimenzionalni koordinatni sistem z določitvijo glave poslušalca v izhodišču le-tega. V koordinatnem sistemu definiramo vodoravni kot (oz. azimut) razsežnosti med -180° in 180° , navpični kot (oz. zenit ali elevacijo) razsežnosti med -90° in 90° ter razdaljo med izvorom in poslušalcem, ki se nanaša na radialno razdaljo od središča glave oziroma izhodišča koordinatnega sistema [1].

Določanje lokacije vira zvoka se začne v samem uhlju [2]. Na vhodu v sluhovod se združi veliko število zvočnih valov, ki ustrezajo različnim potem širjenja, le-te pa seštevajo neposredni oz. neodbit zvok in zvok, ki se odbija od različnih površin in delov telesa. Kombinacija teh vrst zvočnih valov vsebuje informacije o smeri posameznega zvočnega vira in jo je mogoče opisati kot izvor, ki prihaja iz določenega azimuta, elevacije in razdalje glede na poslušalca [1].

Ena najstarejših teorij na področju lokalizacije zvoka je teorija dupleksa, ki opredeljuje medzvočne časovne razlike (ang. Inter-aural Time Difference, ITD, op. p.) in razlike med ušesnimi nivoji (ang. Inter-aural Level Difference, ILD, op. p.) kot dva glavna binavralna znaka za določanje lokacije izbranega vira zvoka. ITD se nanaša na razliko v dolžinah poti, ki jih mora zvok prepotovati od izvora do posameznega ušesa zaradi ločenosti le-teh v prostoru. ILD se nanaša na razliko amplitude zvočnih valov, ko dosežejo levo ali desno uho. Zvok, ki prihaja z desne strani telesa bo intenzivnejši v desnem ušesu kot v levem in obratno. Uho, ki je bolj oddaljeno od vira zvoka je zasenčeno z glavo in tako prispe v uho posledično manj energije kot v uho, ki je bližje izvoru [1].

Lokalizacija zvočnih virov ni odvisna le od binavralnih znakov. Velik pomen za pravilno lokalizacijo imajo frekvenčni pasovi, prav tako pa je natančnost lokalizacije odvisna od glasnosti in trajanja zvočnih signalov. Veliko je odvisno tudi od tega, ali oseba sedi, stoji ali se giblje. Dandanes je na splošno ugotovljeno, da sta oblika in velikost uhljev, glave, ramen in drugih delov človeškega telesa ključnega pomena za natančno lokalizacijo zvoka. Le-ti delujejo kot individualno prilagojeni filtri, ki se razlikujejo od osebe do osebe in specifično spreminjajo frekvenčni spekter vseh zvočnih signalov. Gre za funkcije prenosa, povezane z glavo (ang. Head Related Transfer Function, HTRF, op. p.). Posamezen HTRF predstavlja posameznikov frekvenčni odziv levega in desnega ušesa in ga je skoraj nemogoče posplošiti na širšo populacijo [1].

1.2 Reprodukcija in upodabljanje prostorskega zvoka

Prostorska zvočna reprodukcija je proces poustvarjanja vtisa resničnega ali namišljenega zvočnega okolja, vključno z zaznanimi prostorskimi informacijami, kot so lokacija predmetov in lastnosti prostorske akustike.

Prostorski zvok je bil prvotno pomemben za glasbeno (re)produkcijo, nato pa se je preselil na številna področja, vključno z video igrami, zasloni uporabniškega vmesnika ter virtualno in obogateno resničnostjo [3].

Najpogostejša naprava za reprodukcijo zvoka je zvočnik, elektroakustični pretvornik, ki proizvaja zvok na podlagi električnega vhodnega signala. V primeru prostorskega zvoka sta za reprodukcijo binavralnih razlik (slušne percepcije z obema ušesoma, op. p.) in artefaktov potrebna vsaj dva zvočnika. Vsako uho sprejema signale iz vsakega zvočnika in na posamezno uho ni mogoče predvajati le enega kanala [1].

Reprodukcija, ki temelji na slušalkah, se bistveno razlikuje od konfiguracije katerega koli zunanjega zvočnika, saj lahko neposredno napaja vsako uho z individualnim zvočnim signalom. Oba signala vsebujeta pravilne ITD, ILD in medzvočne spektralne razlike. HRTF oz. HRIR (Head Related Impulse Response) je torej mogoče neposredno uporabiti za reprodukcijo prostorskega zvoka prek slušalk. Še vedno je največji problem individualnost HRIR, ki onemogoča posploševanje za širok spekter poslušalcev ali povzroči občutno zmanjšanje kakovosti reprodukcije [1].

2 Web Audio API

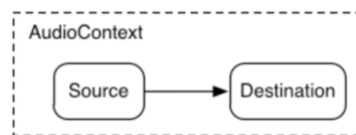
Čeprav je zvok na spletu podprt že dolgo, je bil do nedavnega precej primitiven in ga je bilo potrebno dostaviti prek vtičnikov, kot sta Flash in QuickTime (večpredstavnostni programski platformi, ki omogočata pretakanje zvoka in videa ter lahko zajemata vnos miške, tipkovnice, mikrofona in kamere, op. p.). Uvedba zvočnega elementa `<audio>` v HTML5 je bila zelo pomembna, saj je omogočala osnovno pretočno predvajanje zvoka, pa vendar ni bila dovolj zmogljiva za obvladovanje kompleksnejših zvočnih aplikacij [4].

Za uvedbo zmožnosti obdelave zvoka, ki jo najdemo v sodobnih namiznih aplikacijah za produkcijo zvoka ter v sodobnih zvočnih sistemih za igre, ponuja Javascript programska knjižnica Web Audio API zmogljiv in vsestranski sistem za nadzor zvoka na spletu. Razvijalcem omogoča izbiro zvočnih virov, dodajanje posameznih učinkov zvočnim izvorom, ustvarjanje zvočnih vizualizacij, uporabo prostorskih učinkov in še veliko več [5].

2.1 Osnovni pojmi

Web Audio API vključuje upravljanje zvočnih operacij znotraj elementa zvočnega konteksta (AudioContext, op. p.), ki je bil zasnovan tako, da omogoča modularno usmerjanje. Osnovne zvočne operacije se izvajajo z zvočnimi vozlišči (AudioNode, op. p.), ki so med seboj povezana v graf usmerjanja zvoka (Slika 3). Modularna zasnova zagotavlja prilagodljivost za ustvarjanje kompleksnih zvočnih funkcij z dinamičnimi učinki [5]. Prav tako omogoča poljubne povezave med različnimi zvočnimi vozlišči. Vsako vozlišče ima lahko več poljubnih vhodov (razen izvirnega, ki vhodov nima, ter ciljnega, ki ima samo enega) in izhodov.

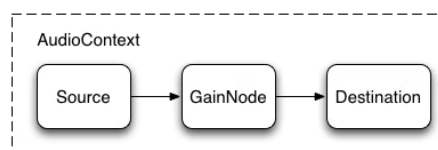
V najpreprostejšem primeru se izvorni vir usmeri neposredno na izhod (Slika 1).



Slika 1: Usmerjanje izvornega zvoka neposredno na izhod.

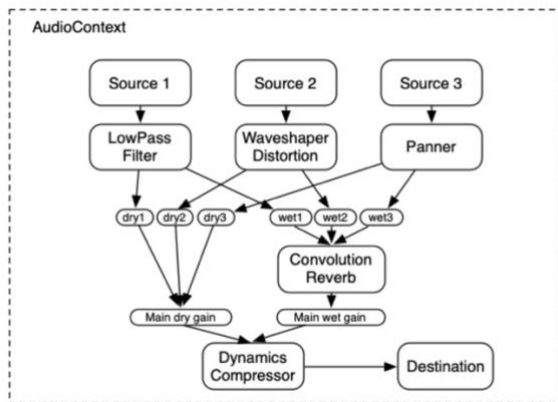
Zvočna vozlišča so povezana v verige (oz. mreže) z vhodi in izhodi. Običajno se veriga prične z enim ali več viri – Web Audio API omogoča uporabo neomejenega števila zvočnih virov. Le-ti so lahko izračunani matematično in predstavljajo periodično valovno obliko, kot je sinusni val, ki povzroči konstanten ton, lahko pa so uvoženi kot posnetki iz zvočnih oz. video datotek. V Web Audio API-ju se zvok nalaga v element AudioBuffer. Osnovni pristop za nalaganje zvočnih datotek je uporaba XMLHttpRequest, ki omogoča nalaganje mono ali stereo datotek v več audio formatih, kot so WAV, MP3, AAC in OGG [4]. Privzeta frekvenca vzorčenja, uporabljena v orodju je 44.1 kHz, zato je najvišja frekvenca uporabljenih zvočnih datotek, ki jo je še mogoče natančno predstaviti, 22 kHz. Pri uporabi Web Audio API-ja je potrebno upoštevati, da je izbira ustreznega zvočnega formata kakor tudi obravnavanje visokih frekvenc odvisno od kompatibilnosti s ciljnim oz. trenutno uporabljenim brskalnikom.

Izhode izvornih vozlišč je mogoče povezati z drugimi vhodi, ki neposredno spreminjajo ali mešajo obstoječe zvočne tokove v nove. Pogosta manipulacija je množenje zvočnih tokov z izbrano vrednostjo, da ti postanejo glasnejši ali tišji (GainNode, op. p.) (Slika 2). Ko je zvok dovolj obdelan za predvideni učinek, ga povežemo z vozliščem cilja (BaseAudioContext.destination, op. p.), ki je zadolženo za pošiljanje zvoka v zvočnike ali slušalke [4].



Slika 2: Množenje vhodnega zvoka za namen povečanja ali zmanjšanja glasnosti.

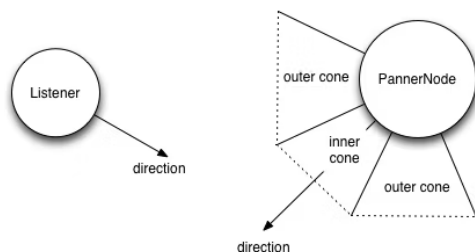
Tipičen potek dela v Web Audio API-ju sledi petim korakom: ustvarjanje zvočnega konteksta, ustvarjanje virov znotraj zvočnega konteksta, ustvarjanje vozlišč učinkov, namenjenih obdelavi izvorov, izbira končne destinacije zvoka, povezovanje virov s posameznimi filtri do cilja (Slika 3) [5].



Slika 3: Primer grafa usmerjanja zvoka s tremi izvori in vrsto modifikacij pred izhodom.

2.2 Prostorski zvok v Web Audio API-ju

Web Audio API omogoča tudi nadzor nad prostorsko umestitvijo zvoka, ki deluje na podlagi modela vir-poslušalec. Pri tem so zapletene prostorske umestitve ustvarjene s pomočjo elementa premikanja (PannerNode, op. p.), za katerim stoji širok nabor matematičnih operacij, ki zagotavljajo učinek 3D zvoka v prostoru. Vmesnik PannerNode definira objekt za obdelavo zvoka, ki predstavlja lokacijo, smer in obnašanje signala zvočnega vira v simuliranem fizičnem prostoru z uporabo desnosučnega kartezičnega koordinatnega sistema. Z uporabo elementa premikanja je mogoče zvočni tok prostorsko razporediti oz. postaviti v prostor relativno na poslušalca (AudioListener, op. p.) [5].



Slika 4: Model vir-poslušalec (PannerNode-Listener) v kontekstu prostorskega zvoka knjižnice Web Audio API.

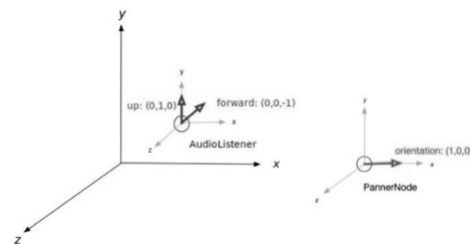
2.2.1 AudioListener

Vmesnik AudioListener predstavlja položaj in orientacijo osebe, ki posluša obdelan ali neobdelan zvočni vir. Pri ustvarjanju instance objekta je potrebno določiti parametre *positionX*, *positionY* ter *positionZ* za lokacijo poslušalca v 3D koordinatnem sistemu. Parametri *forwardX*, *forwardY* in *forwardZ* predstavljajo smer, v katero je poslušalec obrnjen v istem kartezičnem koordinatnem sistem. Poleg tega je potrebno določiti še parametre *upX*, *upY* in *upZ*, ki predstavljajo koordinatni položaj vrha poslušalčeve glave (Slika 5) [4].

2.2.2 PannerNode

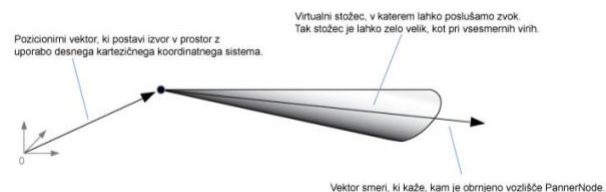
Ob ustvarjanju instance objekta PannerNode mu je potrebno določiti pomembne parametre za namen določanja orientacije in položaja v prostoru. Element PannerNode ima vedno točno en vhod in en izhod. Vhod je lahko mono ali stereo, izhod pa je le stereo – reproduciranje prostorskega zvoka ni mogoče brez vsaj dveh zvočnih kanalov.

Objekt PannerNode ima vektor pozicije, ki definira lokacijo izvora v prostoru in vektor orientacije, ki predstavlja smer v katero se zvok projecira. Vektorja sta definirana kot [*positionX*, *positionY*, *positionZ*] in [*orientationX*, *orientationY*, *orientationZ*] (Slika 5) [4].



Slika 5: Diagram koordinatnega sistema s prikazanimi atributi AudioListener in PannerNode.

Poleg tega ima vmesnik PannerNode virtualni zvočni stožec (Slika 4, Slika 6), ki predstavlja, kako je zvok usmerjen glede na širino izvora. Zvok je lahko vsesmeren in slišen kjerkoli, ne glede na njegovo orientacijo, lahko pa je usmerjen in slišen le, če je obrnjen proti poslušalcu. Če je zvok usmerjen proti poslušalcu, je potemtakem glasnejši, kot če je usmerjen stran od poslušalca. *ConeInnerAngle* ("inner cone", Slika 4, op. p.) opisuje kot virtualnega stožca znotraj katerega ni izgube glasnosti zvoka. *ConeOuterAngle* ("outer cone", Slika 4, op. p.) opisuje kot virtualnega stožca, izven katerega se bo glasnost zmanjševala za konstanto vrednost, definirano z lastnostjo *coneOuterGain*. *ConeOuterGain* opisuje količino zmanjšane glasnosti zunaj stožca in varira med 0 in 1. V primeru, da je vrednost nastavljena na 0, zunaj stožca ni slišati nobenega zvoka [4].

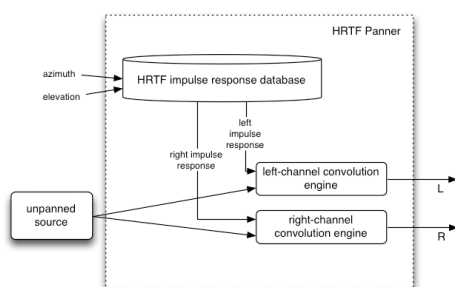


Slika 6: Predstavitev parametrov vozlišča PannerNode.

Lastnost *panningModel* je vrednost, ki določa, kateri algoritem za umestitev v prostor bo uporabljen za postavitev zvočnega vira v 3D prostor. Možni vrednosti sta *equalpower* in *HRTF*. Prvi predstavlja algoritem enake moči in je privzeta vrednost atributa. Slednji upodablja stereo izhod višje kakovosti ter uporablja

konvolucijo z izmerjenimi impulznimi odzivi človeških subjektov, posnetih pri različnih azimutih in višinah. Zagotavlja bolj zaznavno prostorski zvok kot model *equalpower*. Posledično je način *HRTF* zahtevnejši in zahteva več procesorske moči zaradi nenehnega izvajanja konvolucije med vhodnimi podatki in nizom impulzov *HRTF* (Slika 7), poleg tega pa procesira več podatkov zaradi uporabe večih parametrov za reprodukcijo prostorskega zvoka kot način *equalpower* [4].

Zvoki, ki so poslušalcu bližje so glasnejši od tistih, ki so od poslušalca bolj oddaljeni. Lastnost *distanceModel* določa, kako se glasnost zvoka spreminja glede na razdaljo od poslušalca. Na izbiro je linearni, inverzni ter eksponentni model. Vsi modeli na podlagi lastnosti *rolloffFactor*, *maxDistance* in *refDistance* po ustrezni enačbi izračunajo zmanjšanje glasnosti, ki ga povzroči razdalja. *RolloffFactor* je vrednost, ki opisuje, kako hitro se zmanjša glasnost, ko se vir odmika od poslušalca in jo uporabljajo vsi modeli razdalje. Lastnost *maxDistance* predstavlja največjo razdaljo med virom zvoka in poslušalcem, po kateri se glasnost izvora ne zmanjša več. Vrednost je uporabljena le v linearnem modelu. *RefDistance* predstavlja referenčno razdaljo, pri kateri začne učinkovati zmanjšanje glasnosti. To vrednost uporabljajo vsi modeli razdalje [4].



Slika 7: Diagram, ki prikazuje osnovni postopek umestitve zvoka v prostor z uporabo *HRTF*.

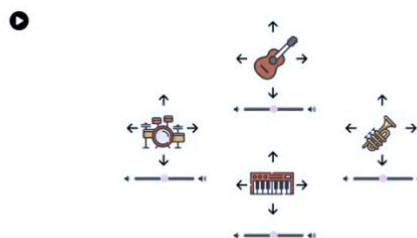
3 Demonstracija uporabe knjižnice Web Audio API

Za demonstracijo upodabljanja prostorskega zvoka s pomočjo programske knjižnice Web Audio API smo ustvarili enostaven vmesnik, ki uporabniku ponuja usklajeno predvajanje štirih zvočnih tokov štirih glasbenih instrumentov (brenkalo, trobilo, tolkalo in klaviatura) ter njihovo premikanje v prostoru. Gre za posnemanje virtualnega ansambla, kjer lahko uporabnik nadzoruje posamezen instrument oz. modificira njegovo glasnost in pozicijo v prostoru.

Za potrebe uporabniškega upravljanja oz. interakcije smo s pomočjo osnovnih HTML elementov oblikovali preprost uporabniški vmesnik, kjer uporabnik nadzoruje predvajanje in ustavljanje zvoka ter premike posameznih instrumentov na zaslonu (Slika 8). Vsakemu od štirih

definiranih instrumentov pripadajo kontrole premika in drsnik, s katerim lahko uporabnik uravnava glasnost posameznega zvočnega toka.

Web Audio API vmesnik temelji na konceptu zvočnega grafa, ki je sestavljen iz posameznih vozlišč. Vhodna vozlišča definirajo posamezni zvočni vhodi štirih instrumentov, povezanih z enim samim gumbom za predvajanje, s čimer je zagotovljena usklajenost posameznih virov za zagotavljanje sinhronosti v trenutku, ko se zvočni viri združijo v eno melodijo. Vsak zvočni tok je predstavljen s svojim zvočnim vozliščem *MediaElementAudioSourceNode* ter v procesu reprodukcije prostorskega zvoka povezan s svojo instanco vozlišča *PannerNode*, kar omogoča njegovo kontroliranje neodvisno od preostalih elementov.



Slika 8: Uporabniški vmesnik za nadzorovanje virtualnega ansambla.

Pri ustvarjanju zvočnega konteksta smo definirali poslušalca in njemu pripadajoče attribute pozicije in orientacije v koordinatnem sistemu. Poslušalca smo umestili na sredino zaslona za simuliranje osebe, ki gleda v prostor z instrumenti. Za začetne pozicije posameznih zvočnih virov smo definirali polovico višine ter širine zaslona. Definirali smo tudi meje premikanja, s čimer smo omejili gibanje oz. možnost premika posameznega instrumenta izven mej zaslona. Za vsak posamezen instrument smo ustvarili instanco objekta *PannerNode*, ob inicializaciji pa smo instancam dodelili tudi vse pripadajoče potrebne attribute (Slika 9). Za atribut *panningModel* smo uporabili model *HRTF*, za pojenjanje intenzitete zvoka med oddaljevanjem od poslušalca linearni model, instrumente smo pozicionirali na sredino zaslona s priročno uporabo vgrajenih HTML lastnosti *window.innerWidth* ter *window.innerHeight*, orientacijo instrumentov pa smo nastavili tako, da so instrumenti obrnjeni proti poslušalcu.

Samo modifikacijo zvočnih virov lahko uporabnik izvede s pomočjo premika drsnika glasnosti instrumenta, pri čemer se spremeni vrednost instance elementa *GainNode*, pripadajoče instrumentu. V primeru, da glasnost zniža na minimalno vrednost, doseže učinek, kot da instrumenta v virtualnem ansamblu ni. S pritiskom na posamezne kontrole premikanja se instrumenti vizualno premikajo po zaslonu, prav tako pa se s pomočjo translacij oz. akcij, ki jih sproži uporabniški vnos

spreminjajo atributi objekta `pannerNode`, kar neposredno vpliva na uporabnikovo percepcijo prostorskega zvoka.

V procesu modifikacije zvočnih virov pred povezovanjem zvoka s ciljnimi izhodnimi vozliščmi smo posamezen zvočni tok povezali s pripadajočim objektom `GainNode` za nastavljanje glasnosti ter objektom `PannerNode` za predstavitev prostorskega zvoka. Nato smo obdelan zvočni vir vezali na objekt `BaseAudioContext.destination` in ga združili s preostalimi zvočnimi viri s pomočjo česar smo dobili eno izhodno vozlišče za potrebe predvajanja izhodnega zvoka.

```
const panner = new PannerNode(audioCtx, {
  panningModel: "HRTF",
  distanceModel: "linear",
  positionX: window.innerWidth/2,
  positionY: window.innerHeight/2,
  positionZ: 300,
  orientationX: 0.0,
  orientationY: 0.0,
  orientationZ: -1.0,
  refDistance: 1,
  maxDistance: 10000,
  rolloffFactor: 10,
  coneInnerAngle: 100,
  coneOuterAngle: 130,
  coneOuterGain: 0.3,
});
```

Slika 9: Definiranje posameznih atributov instance objekta `pannerNode` za posamezen inštrument.

3.1 Zahteve pri delu s knjižnico Web Audio API

Pri uporabi knjižnice Web Audio API je najbolj časovno potratna in najzahtevnejša naloga reprodukcije prostorskega zvoka nastavljanje parametrov elementa `PannerNode` v skladu z zvočnim virom, ki ga želimo upodobiti. Zavedati se moramo, da imajo različni zvočni viri različno usmerjenost zvočnih valov. V konkretnem primeru so bili uporabljeni inštrumenti, katerih usmerjenost je širša, kot na primer pri zvočnikih osebnega računalnika ali mobilnega telefona.

V spletnih aplikacijah, obogatenih z zvokom, je časovni zamik med uporabniškim vnosom (pritisek tipke, premik miške itd.) in slišanim zvokom izredno pomemben. Ta časovni zamik imenujemo zakasnitev oz. latenca in je posledica več dejavnikov. Večja kot je zakasnitev, manj zadovoljujoča bo uporabniška izkušnja. Zakasnitev je še posebno moteča pri glasbeni produkciji in pri igranju iger, kjer lahko daje vtis, da zvoki zaostajajo ali da se igra ne odziva. Pri glasbenih aplikacijah zakasnitev vpliva na ritem, pri igranju pa na natančnost igranja. Razumna zakasnitev je lahko od 3-6 milisekund do 25-50 milisekund, odvisno od aplikacije. Pri uporabi programske knjižnice Web Audio API jo lahko zmanjšamo z zmanjšanjem kompleksnosti obdelave zvoka (z manj uporabe učinkov in filtrov), z zmanjšanjem količine zvočnih podatkov, obdelanih v vsakem ciklu, z zmanjšanjem velikosti vmesnega pomnilnika itd.

Za zagotavljanje varnosti in preprečevanje zlorab je brskalnik Google, ki smo ga uporabljali tekom implementacije programske rešitve za demonstracijo uporabe knjižnice, s politiko CORS (Cross-Origin

Resource Sharing, op. p.) omejil dostop do virov, ki prihajajo iz domen, ki so različne od domene, s katere je bil poslan zahtevek za dostop. Za nemoteno predvajanje zvočnih datotek v brskalniku smo implementacijo uporabniškega vmesnika in pripadajočih zvočnih ter ostalih datotek prenesli v strežniško aplikacijo ter tako zadostili pogojem Googlove CORS politike.

4 Zaključek

Reprodukcija prostorskega zvoka v spletu je konkreten zalogaj vsakega razvijalca s stališča manipuliranja posameznih vrednosti atributov za njegovo najboljše upodabljanje. Razvijalec mora imeti dobro znanje psihoakustike ter poznati značilnosti posameznih zvočnih virov realnega sveta do te mere, da lahko njihovo obnašanje predstavi v skladu s tem, česar je uporabnik vajen v resničnem življenju.

Zaznavanje kakovosti zvoka in zakasnitve je subjektivno, zato je vključitev resničnih uporabnikov in zbiranje njihovih povratnih informacij ključnega pomena za razumevanje, kako dobro Web Audio API deluje v posamezni aplikaciji. V skladu s tem bi bilo smiselno preizkusiti natančnost oz. kvaliteto delovanja knjižnice s končnimi uporabniki, pri tem pa uporabiti katerega izmed standardiziranih testov oz. metodologij, ki so bile razvite za ocenjevanje različnih vidikov zaznavanja zvoka.

Literatura

- [1] J. Sodnik and S. Tomažič, *Spatial auditory human-computer interfaces*, 1st ed. Springer Publishing Company, Incorporated, 2015
- [2] F. A. Everest and K. C. Pohlmann, *Master handbook of acoustics*, 5th ed. New York: McGraw-Hill, 2015.
- [3] R. Menzies-Gow, "Brief introduction to spatial audio reproduction," *VAAE*, Oktober 2018. [Na spletu]. Dostopno: <https://generic.wordpress.soton.ac.uk/vaae/2017/12/31/brief-introduction-to-spatial-audio-reproduction/>, [Dostopano: januar 2023].
- [4] W3C, Web Audio API, [Na spletu]. Dostopno: <https://www.w3.org/TR/webaudio/>, [Dostopano: januar 2023].
- [5] Web APIs | MDN, Web Audio API - Web APIs: MDN, [Na spletu]. Dostopno: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API, [Dostopano: januar 2023].