

IZRADA VR IGRE POMOĆU A-FRAME RADNOG OKVIRA

VR GAME DEVELOPMENT USING A-FRAME FRAMEWORK

Sanja Duk¹, Damir Šehić²

¹Tehničko veleučilište u Zagrebu, Zagreb, Hrvatska

²Tehničko veleučilište u Zagrebu, Zagreb, Hrvatska, Student

Sažetak

Rad se bavi problematikom prikaza sadržaja u virtualnoj stvarnosti te skupom tehnologija i uređaja koji su za to potrebni. Opisana je metodologija prikaza virtualne stvarnosti u internetskom pregledniku. Pokazano je kako je navedeno moguće implementirati korištenjem standardnih web tehnologija kao što su HTML5 i JavaScript. Uz standardne web tehnologije korišten je WebVR API-a koji je još u eksperimentalnoj fazi. Njime se upravlja pomoću A-Frame radnog okvira. Rezultat je razvijena igra čije su komponente i funkcionalnosti prikazane u radu.

Ključne riječi: virtualna stvarnost, WebGL, WebVR, A-Frame

Abstract

This paper is based on displaying content in virtual reality and the set of technologies and hardware that are needed. Example of virtual reality is described by implementation in web browser. Possibility of implementation using standard web technologies such as HTML5 and JavaScript has been presented. Along with standard Web technologies, WebVR API has been used, which is still in the experimental phase. A-Frame framework has been used in order to manipulate with WebVR. Game whose components and functionality are shown in this paper, has been developed as a result.

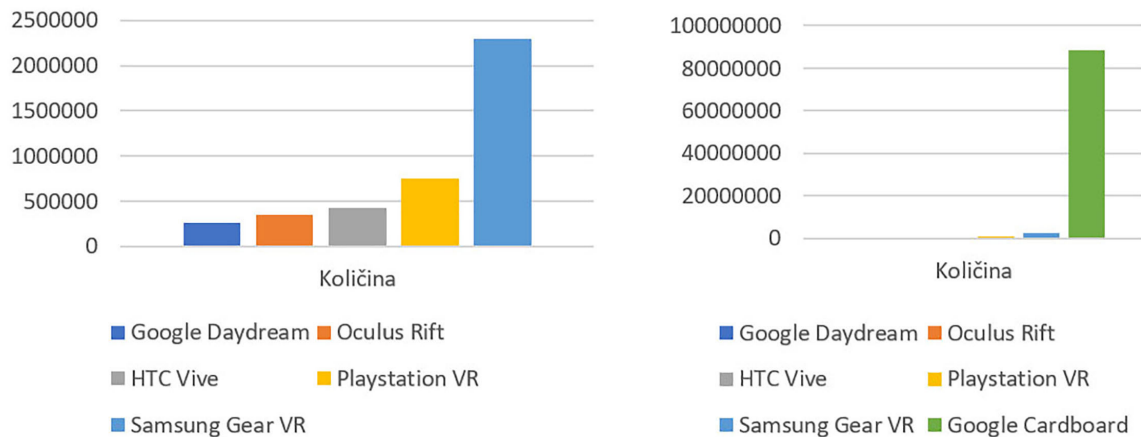
Keywords: virtual reality, WebGL, WebVR, A-Frame

1. Uvod

1. Introduction

Iako se tehnologija virtualne stvarnosti (eng. virtual reality – VR) razvija još unazad 30-ak godina, tek je nedavno postala dostupnija prosječnim korisnicima zbog cijene kao i ostalih tehničkih razloga. Radi široke primjene koja sažeto uključuje razne simulacije, edukaciju i zabavu, pri čemu se stvara duboki osjećaj da se korisnik nalazi na drugom mjestu u već u idućem trenutku, virtualna stvarnost ima velik potencijal. Obzirom da je internet gotovo svima pristupačan, virtualna stvarnost u web pregledniku omogućava širokoj masi da jednostavno isprobaju tu novu vrstu interakcije. WebVR aplikacijsko programsko sučelje (eng. application programming interface – API) omogućava virtualnu stvarnost u web pregledniku, no obzirom da je to samo eksperimentalni API internetskog preglednika, poželjno je koristiti neki radni okvir koji će olakšati cijeli proces izrade VR aplikacije, za što se u ovom radu koristi A-Frame.

Pravila igre koja je razvijena u svijetu virtualne stvarnosti su sljedeća: igrači započinju igru njenim učitavanjem na web adresi te prelaze u VR način rada na uređaju. Na ekranu se pojavljuje objekt kojeg taj igrač kretanjem u virtualnoj stvarnosti mora pronaći. Po pronalasku, klikom na objekt igrač potvrđuje pronalazak te mu se generira novi objekt za potragu. Ovaj ciklus se ponavlja tri puta nakon čega se igraču dodjeljuje novi zadatak potrage. Igrači se po prostoru mogu kretati klikom na zelene kugle čime se događa teleportacija igrača. Pobjednik je onaj igrač koji prvi pronađe sve zadane objekte.



Grafikon 1: Popularnost VR hardvera u 2016. godini – usporedba popularnosti sa i bez Google Cardborda [7]

Chart 1: VR hardware popularity in 2016 - comparison with and without Google Cardboard [7]

2. Elementi virtualne stvarnosti

2. Virtual reality elements

Virtualna stvarnost je računalna tehnologija kojom se korisniku pruža osjećaj da se nalazi u nekom drugom, virtualnom svijetu. To se čini kombinacijom softvera i posebnog hardvera čiji su glavni elementi stereoskopski ekran, hardver za praćenje pokreta, razni uređaji za prijenos ulaznih informacija te stolni i mobilni uređaji. Ova se tehnologija počela razvijati krajem 80.-ih godina prošlog stoljeća, no tehnologija tog doba nije još bila dovoljno razvijena da bi se razvio zadovoljavajući proizvod. Glavni element virtualne stvarnosti je trodimenzionalni prikaz okruženja koji stvara osjećaj dubine. Hardver za virtualnu stvarnost upotrebljava stereoskopski 3D ekran. To je uređaj koji se stavlja na glavu ili ispred očiju (eng. head mounted display – HMD) kako bi se stvorio osjećaj dubine. Stvaranju iluzije potrebni su pomaci i zakrivljenja slika koje se predstavljaju oku – za svako oko po jedna slika. Zadaća softvera je da generira sliku koja je u skladu s time pripremljena, i to minimalno 60 a idealno 120 puta u sekundi (120 FPS). Svako kašnjenje u aplikaciji poželjno je izbjegavati kako se ne bi uništio doživljaj ili kako ne bi došlo do mučnine korisnika. [1]

Drugi korak u zavaravanju mozga da se nalazi negdje drugdje je praćenje pokreta glave uz osvježavanje slike na ekranu ovisno o pokretu u stvarnom vremenu. Hardver koji se koristi za to su kombinacije žiroskopa, akcelerometra te se u nekim slučajevima koristi i magnetometar. [2]

Praćenje pokreta je jednako važno kao i dobar ekran, jer svako kašnjenje u pokretu uništava doživljaj te izaziva mučninu. [1]

Korisnik sa virtualnom stvarnošću može komunicirati pomoću nekoliko vrsta ulaznih uređaja.

To mogu biti:

- igrači kontroleri
- senzori za praćenje pokreta ruku
- bežični kontroleri za ruke

Platforme za stvaranje VR iskustava dijele se na tri grupe:

- nativni kompleti za razvoj softvera
- razne igrače platforme (eng. game engine) kao što je primjerice Unity3D
- najnoviji web preglednici

VR iskustva za web se razvijaju pomoću WebGL-a (eng. Web Graphics Library) i JavaScripta. Prednost web preglednika nad igračom platformom je ta što su web preglednici višeplatformski te postoji mogućnost korištenja web infrastrukture. [1] [6]

Najveći problem kod VR je bolest putovanja odnosno VR mučnina. Virtualne vožnje koje se polako kreću puno manje utječu na mučnine, naspram simulacija pokreta velike brzine. Bolja opcija za kretanje je korištenje teleportera, to jest da korisnik pomoću kontrolera pokaže gdje se želi pomaknuti i tamo se teleportira, što je implementirano u razvijenoj igri. [3]

3. Uređaji za virtualnu stvarnost

3. *Virtual reality hardware*

VR uređaji mogu se podijeliti na tri kategorije:

- uređaji za entuzijaste
- mobilni VR uređaji
- uređaji za masovnu distribuciju

Uređaji za entuzijaste su najskuplji i najkvalitetniji. Najpoznatija dva su HTC VIVE i Oculus Rift.

Oba uređaja imaju odličan ekran sa 1080x1200 točaka po oku te odličnu detekciju orijentacije glave. Mobilni VR uređaji poput Samsung GearVR i Google Daydreama su uređaji koji za ekran koriste novije pametne telefone, postavljaju ih u držače i imaju posebne leće. Skupina uređaja za masovnu distribuciju se svodi na Google Cardboard i verzije Google Cardboarda koje proizvode druge tvrtke. To je kartonska kutija sa utorom za mobitel, dvije leće te gumbom za interakciju. Glavna prednost nad uređajima za entuzijaste mu je cijena, prenosivost i jednostavnost uporabe. Aplikacije za takve uređaje moraju biti jednostavnije zbog performansi uređaja. [2]

4. WebVR programsko sučelje

4. *WebVR API*

Od ranije je poznato da web preglednici mogu prikazivati 3D grafiku u realnom vremenu pomoću WebGL-a, što je API standard za iscrtavanje 3D grafike pomoću JavaScripta. WebVR je eksperimentalni JavaScript API koje pruža podršku za VR uređaje u web pregledniku. WebVR detektira dostupne VR uređaje, provjerava mogućnosti uređaja, detektira orijentaciju i poziciju uređaja te prikazuje slike na uređaju sa željenim brojem slika u sekundi. [5] [6]

On proširuje preglednikovu mogućnost time što automatski radi obradu slike za VR umjesto da ju programer mora sam implementirati, te dodaje i nova JavaScript sučelja za praćenje pozicije i orijentacije glave korisnika. [5] [6]

Prednosti VR aplikacija na webu su:

- Web VR aplikacije nije potrebno skinuti pa instalirati jer se otvaraju u internetskom pregledniku
- podržane su na različitim vrstama uređaja (mobilni, tabletni, prijenosnici, stolni uređaji)
- pojednostavljen je razvoj korištenjem tehnologija koje su poznate većini programera – HTML5 i JavaScript
- većina alata je otvorenog koda
- jednostavno se postavljaju i dijele

Spojeni uređaji za virtualnu stvarnost detektiraju se pomoću funkcije `getVRDisplays`, nakon čijeg poziva je potrebno dohvatiti parametre uređaja za pojedino oko.

```
var vrDisplay
navigator.getVRDisplays().
then((displays) => {
  if (displays.length > 0) {
    vrDisplay = displays[displays.
length - 1]
  }
})
var leftEye = vrDisplay.
getEyeParameters("left")
var rightEye = vrDisplay.
getEyeParameters("right")
```

Programski isječak 1 *Detekcija spojenih VR uređaja*
Code 1 *Detection of connected VR devices, usage of*
getEyeParameters functions

```
var frameData = new VRFrameData()
function onVRFrame() {
  vrDisplay.requestAnimationFrame(on
VRFrame)
  vrDisplay.getFrameData(frameData)
  // ... logika za iscrtavanje slike
za svako oko
  vrDisplay.submitFrame()
}
```

Programski isječak 2 *Praćenje pokreta glave*
Code 2 *Head movements tracking*

Funkcija `onVRFrame` se poziva za svako iscrtavanje slike na VR uređaj. Objekt `frameData` sadrži vrijednosti `leftProjectionMatrix`, `leftViewMatrix`, `rightProjectionMatrix`, `rightViewMatrix` i najzanimljivije svojstvo `pose`. `Pose` je također objekt koji sadrži vrijednosti `angularAcceleration`, `angularVelocity`, `linearAcceleration`, `linearVelocity`, `orientation` te `position` koje služe za praćenje pokreta glave te pomoću kojih se korisnik može orijentirati unutar aplikacije.

5. A-FRAME radni okvir

5. A-FRAME framework

Jedan od popularnih radnih okvira za izradu VR iskustva na webu jest A-Frame. Razvila ga je Mozilla koja je začetnik WebVR projekta. A-Frame podržava većinu VR uređaja poput VIVE-a, Rifta, GearVR-a, Daydreama, Cardboarda te se može koristiti i za proširenu stvarnost [8].

Kao alat za pregled i uređivanje scena koristi se A-Frame inspektor koji omogućava da se pomoću njegovog uređivača mijenjaju veličine, pomiču ili rotiraju entiteti na sceni. Inspektor ubrzava razvoj jer pruža mogućnost pregleda kako će scena izgledati sa željenim promjenama bez ikakvih promjena u kodu.

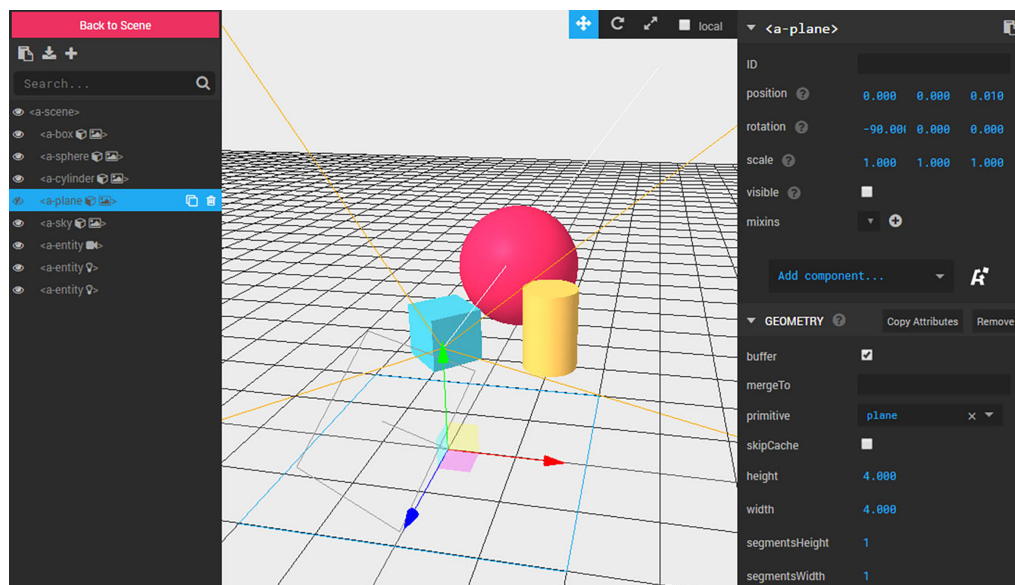
Element `<a-scene>` je globalni korijenski element u kojem se nalazi svi entiteti.

Scena nasljeđuje klasu `Entity` sa svim njenim svojstvima, što znači da čeka da se učitaju svi elementi djeca prije nego što započne ciklus iscrtavanja. Scena se brine o grupiranju WebVR i `three.js` koda, postavlja kameru, `renderer`, `render loop`, početno osvjetljenje, učitava `webvr-polyfill` i dodaje gumb za popunjavanje ekrana i ulazak u VR mod.

```
<!DOCTYPE html>
<html>
  <head>
    <script src="aframe.js"></script>
  </head>
  <body>
    <a-scene>
      <a-box></a-box>
    </a-scene>
  </body>
</html>
```

Programski isječak 3 Minimalni kod za pokretanje A-Framea
Code 3 Minimal A-Framea code

Primitivi su odličan alat za početnike jer na jednostavan način omogućavaju izradu i manipulaciju osnovnih 3D modela. To su pojednostavljeni najčešće korišteni entiteti. Osnovni A-Frame primitivi su `box`, `cone`, `cylinder`, `plane`, `sphere`, `ring`, `torus`, `image`, `sky`, `light`, `text`, `assets`, `assetItem`. Primitivima je jednostavno mijenjati poziciju, rotaciju, veličinu te boju odnosno teksturu, kroz svojstva `position`, `rotation`, `height/width` te `color/src`.



Slika 1: A-Frame inspector

Figure 1: A-Frame inspector

```
<a-box color="red" width="3"></a-box>
```

Programski isječak 4 A-Frame primitiv – kocka

Code 4 A-Frame primitiv - cube

Entiteti su objekti koje je moguće proširiti pomoću komponenti kako bi im se definirao izgled, ponašanje i funkcionalnost. Svi primitivi su implementirani kao entiteti. Geometry komponenta daje osnovni oblik entitetu. A-Frame sadrži ugrađene geometrije box, circle, cone, cylinder, dodecahedron, octahedron, plane, ring, sphere, tetrahedron, torus, torusKnot ali je moguće dodati i vlastitu posebnu geometriju. Material komponenta daje izgled entitetu - boju, prozirnost ili teksturu. Light komponenta označava entitet kao izvor svjetlosti. Svjetlo u sceni utječe na izgled svih drugih entiteta koji nemaju podešen shader:flat. Postoji nekoliko vrsta izvora svjetla u A-Frameu a to su ambient, directional, hemisphere, point i spot.

```
<a-box color="red" width="3"></a-box>
<a-entity geometry="primitive: box;
width: 3"
material="color: red"></a-entity>
```

Programski isječak 5 Usporedba kocke kao primitiva i entiteta

Code 5 Cube as primitive, cube as entity

Svakom entitetu na sceni moguće je dodati teksturu odnosno sliku koja će se aplicirati po njegovoj plohi.

```
<a-box src="slika.png"></a-box>
```

Programski isječak 6 Korištenje teksture na kocki

Code 6 Textured cube

Za interakciju sa entitetima kroz upravljanje klikom služi komponenta cursor. Ona sluša klikove miša, detektira koji je entiteti prvi u ravnini u kojoj se gleda i pokreće posebne događaje.

6. Manipulacija JavaScriptom

6. JavaScript DOM manipulation

Scenu i sve entitete unutar scene moguće je kontrolirati JavaScriptom. A-Frame modificira HTML svih entiteta tako da im proširuje neke DOM API-je. Preporučeno je enkapsulirati sav

JavaScript kod unutar A-Frame komponentata što omogućava modularizirani kod te lakše razumijevanje aplikacije kroz HTML.

```
AFRAME.registerComponent('log', {
  schema: { type: 'string' },
  init: function () {
    console.log(this.data)
  }
})
<a-scene log="Hi Polytechnic">
  <a-box log="Hi Design"></a-box>
</a-scene>
```

Programski isječak 7 Enkapsulirani JavaScript kod u A-Frame komponenti i njegova primjena

Code 7 Encapsulated JavaScript code in A-Frame components and its application

Kroz JavaScript je moguće i kreirati entitet te ga dodati u scenu ili kao dijete element nekog drugog entiteta. Za to se koriste obične funkcije .createElement() i .appendChild(). Manipulacija entiteta se izvršava pomoću funkcije .setAttributes().

```
entityEl.setAttribute('position', {x:
1, y: 2, z: -3})
```

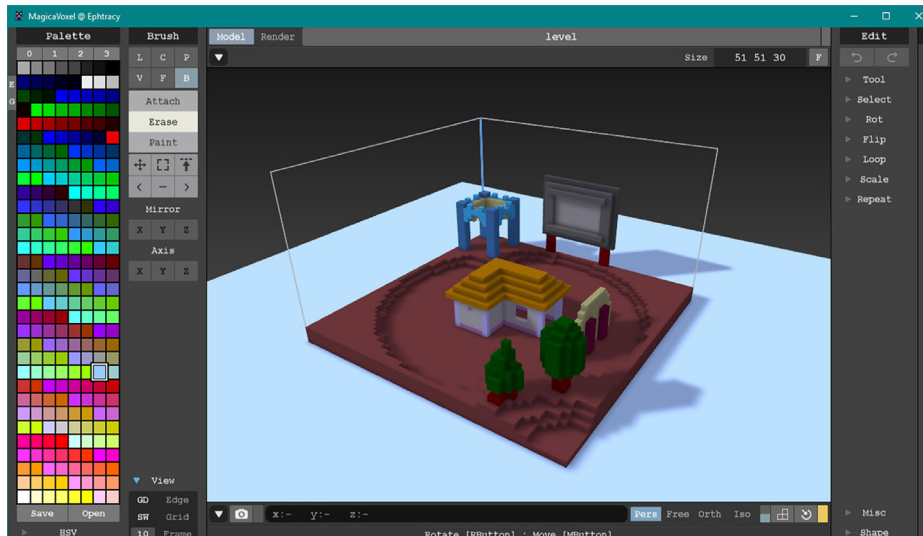
Programski isječak 8 Korištenje setAttributes funkcije

Code 8 Function setAttributes

7. Komunikacija između serverske i klijentske aplikacije

7. Server and client application communication

Komunikacija se izvršava u stvarnom vremenu pomoću web socketa, odnosno pomoću programske biblioteke socket.io. Za pokretanje web socketa u node.js serverskom dijelu aplikacije potrebno je nakon inicijalizacije servera pozvati socket.io i proslijediti mu referencu na http servis. U glavnoj index.html datoteci koja se servira klijentu, zajedno s popratnim JavaScript skriptama, sadržana je A-Frame scena s dodatcima poput pozadinske slike ili 3D modela prostora po kojoj se igrači kreću. Format 3D modela prostora je .bake.ply, izrađen pomoću programa MagicaVoxel te ima zapečene (eng. baked in) sjene zbog optimizacije. Unutar scene je također definiran entitet player, koji unutar sebe sadrži kameru i kursor.



Slika 2: Sučelje alata MagicaVoxel

Figure 2: MagicaVoxel interface

8. Praćenje elemenata scene i igrača

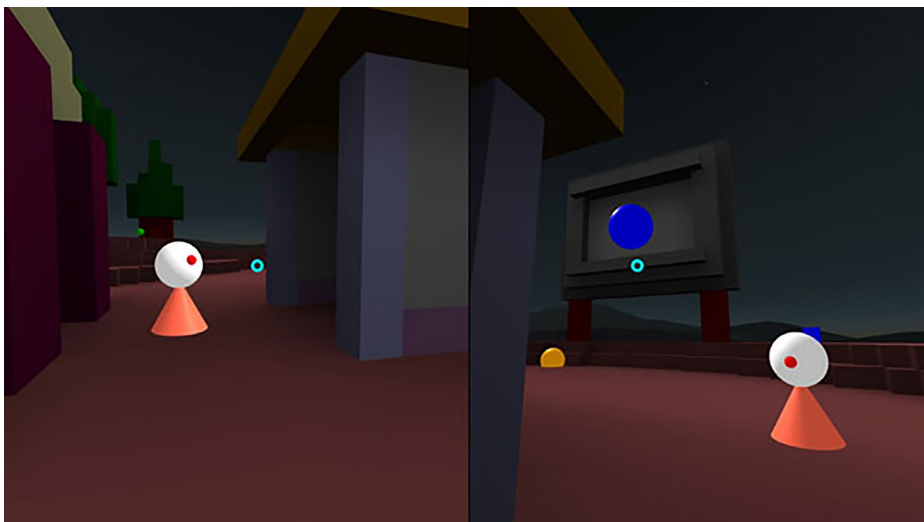
8. Scene and player elements detection

Scena aplikacije sadrži prethodno navedene statične elemente kao što je prostor u kojem se igrači kreću, no sadrži i elemente koji se dinamički mijenjaju kao što su liste točaka koje su dostupne za teleportaciju ili trenutni zadatak. Dodavanje elementa izvršava se manipulacijom DOM-a Javascriptom, koristeći funkcije poput `document.createElement()` i `element.appendChild(newElement)`. Prilikom svake izmjene liste brišu se stare točke i poziva se render funkcija za iscrtavanje novih. Logiku iscrtavanja je moguće optimizirati tako da se ne brišu sve točke i onda ponovo crtaju nove, nego da se brišu i crtaju samo promjene između prošlog i trenutnog stanja.

```
const { onPlayerMove } = require('../player-actions')
export const renderNode = (nodesEl, node, socket, PLAYER_ID) => {
  const newNode = document.createElement('a-entity')
  const nodeProperties = {mixin: 'node', position: node.coordinates, 'change-color-on-hover': 'color: magenta', positionId: node.id}
  Object.keys(nodeProperties).forEach((key) => {window.AFRAME.utils.entity.setComponentProperty(newNode, key, nodeProperties[key]) })
  newNode.addEventListener('click', (e) => onPlayerMove(e, socket, PLAYER_ID))
  nodesEl.appendChild(newNode)
}
```

Programski isječak 9 Render-node.js

Code 9 Render-node.js



Slika 3: Prikaz protivničkog igrača iz perspektive oba igrača

Figure 3: Showing the opponent from the perspective of both players

Kretanje igrača je implementirano kao teleportiranje, a ostvaruje se klikom na zelene kugle u prostoru. U assets direktoriju na serverskoj strani, definirana je lista s koordinatama svih točaka za teleportaciju, dodan im je identifikator i opcija `isDisabled` koja je postavljena na `false`. Ta lista se učitava u `reduceru` i postavlja se u inicijalno stanje u `nodes property`. Na serveru se oslušuju promjene na `nodes propertyu`, te se na svaku promjenu šalje poruka `nodes` preko `web socketa` s trenutnim stanjem tog `propertya`.

```
const store$ = createStore()
store$.pluck('nodes').
distinctUntilChanged()
  .subscribe((s) => io.emit('nodes',
s))
store$.pluck('nodes').
distinctUntilChanged()
  .subscribe((s) => io.emit('nodes',
s))
store$.pluck('objectives').
distinctUntilChanged()
  .subscribe((s) =>
io.emit('objectives', s))
```

Programski isječak 10 Praćenje promjena stanja

Code 10 Monitoring State Changes

Za detekciju orijentacije igrača dohvaća se vrijednost atributa `rotation` na `camera elementu`. Orijehtacija se dohvaća periodički. Kako bi se dobila realna animacija potrebno je postaviti vrlo mali interval od nekoliko puta u sekundi. Na računalima to ne predstavlja problem po pitanju performansi, no na mobilnim uređajima učestalo detektiranje i slanje orijentacije te provjera pristiglih poruka o igračima i često ponovno iscrtavanje uvelike utječe na performanse. Navedeno je moguće optimizirati povećanjem intervala te dodatnom animacijom od početne do konačne točke kretanja.

```
setInterval(() => {
  socket.
emit('playerChangeOrientation', {
  playerId: PLAYER_ID,
  rotation: cameraEl.
getAttribute('rotation') })
}, 100)
```

Programski isječak 11 Dohvaćanje orijentacije igrača

Code 11 Get player orientation

9. Zaključak

9. Conclusion

Prihvaćenost virtualne stvarnosti kao tehnologije za širu javnosti je trenutno u stadiju ranog usvajanja. WebVR u svojoj eksperimentalnoj fazi često mijenja svoj API da bi se mogao dati dobar kritički osvrt na implementaciju, ali mogućnosti koje WebVR otvara web programerima te trenutne performanse koje trenutno osigurava pokazuju veliki potencijal tehnologije.

Korištenjem radnog okvira razvijen je proizvod u kojemu su savladane tehnološke prepreke, a mučnine se gotovo niti ne osjete. WebVR u dijelu web programiranja će vjerojatno omogućiti usku specijalizaciju kroz radne okvire poput A-Framea čiji se API neće mijenjati zbog promjena u WebVRu, već će promjene biti samo u pozadini. Ovime se nastavlja trend potrebe za konstantnim obrazovanjem mladih razvojnih programera i učenjem novih tehnologija.

6. REFERENCE

6. REFERENCES

- [1] Parisi, Toni: „Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile“, O'Reilly, 2015.
- [2] Wikipedia, WebVR, 2017., <https://en.wikipedia.org/wiki/WebVR>, (18.06.2017.)
- [3] Betsy Mason, Virtual reality has a motion sickness problem, 07.03.2017., <https://www.sciencenews.org/article/virtual-reality-has-motion-sickness-problem> (18.06.2017.)
- [4] Kayla Matthews, Virtual Reality Still Has 5 Big Problems to Overcome, 08.06.2016, <http://www.makeuseof.com/tag/virtual-reality-still-5-big-problems-overcome/> (18.06.2017.)
- [5] WebVR, <https://webvr.info/developers/> (18.06.2017.)
- [6] Martin Splitt, WebVR is the next frontier, 22.03.2017., <https://www.youtube.com/watch?v=39MiUkGSEks> (18.06.2017.)

- [7] Martin Splitt, Exploring WebVR, 2017.
https://docs.google.com/presentation/d/1TrqucJ_0nTjnHDtm22WMOq7ULupPSMWCGI9cyeUY5A/edit#slide=id.g2207ea7285_0_24, (18.06.2017.)
- [8] A-Frame, <https://aframe.io/docs/0.5.0/introduction/>, (18.06.2017.)
- [9] WebVR Editor's Draft, 06.06.2017.,
<https://w3c.github.io/webvr/spec/1.1/>,
(18.06.2017.)

AUTORI · AUTHORS



Sanja Duk

Predavač je na Tehničkom veleučilištu u Zagrebu. Rođena je u Zagrebu gdje završava V. gimnaziju. 2008.-te godine završava diplomski studij na Fakultetu elektrotehnike

i računarstva Sveučilišta u Zagrebu te stječe titulu diplomiranog inženjera elektrotehnike. Trenutno je na poslijediplomskom doktorskom studiju na Fakultetu elektrotehnike i računarstva, u polju Računarstvo i računarske znanosti. Područja interesa su web tehnologije, sigurnost informacijskih sustava i baze podataka.

Korespondencija

sduk@tvz.hr

Damir Šehić

Student je Tehničkog veleučilišta u Zagrebu.

Korespondencija

damir.sehic@tvz.hr