



UNIVERZITET U NIŠU  
ELEKTRONSKI FAKULTET  
KATEDRA ZA RAČUNARSTVO



React 360 frejmwork za razvoj VR aplikacija  
- Sistemi virtuelne i proširene realnosti -

Studenti:

Julije Kostov 1026

Darko Stevanović 946

Mentor:

prof. dr Aleksandar Milosavljević

Niš, avgust 2020. god.

## Sadržaj

1	Uvod .....	3
2	Frejmwork React 360 .....	5
	2.1 Struktura aplikacije .....	5
	2.2 2D Interfejsi.....	7
	2.3 3D Objekti .....	10
	2.4 Native moduli .....	12
	2.5. Audio.....	16
3	Zaključak .....	17
4	Literatura .....	18

# 1 Uvod

Termini *virtuelna realnost* i *proširena realnost* su danas u vremenu rapidnog napretka tehnologije veoma često korišćeni. Iako to jesu dva slična termina, gde čak i tehnologije koje se njima bave imaju neke zajedničke tačke u nastavku će se autori rada potruditi da objasne značenja ova dva *srodna* koncepta kao i da razjasne razliku između njih.

**Virtuelnu realnost** (eng. virtual reality) karakteriše potpuno *preuzimanje vida* klijenta čime se ostvaruje utisak prisustva nekoj potpuno drugoj realnosti. Najpoznatiji uređaji koji se koriste u te svrhe su *HTC Vive* i *Oculus Rift*.



Slika 1. HTC Vive [1]

LCD, ili OLED ekrani unutar ovakvog headseta imaju postavljena takva sočiva da klijentu u potpunosti okupiraju vid. Ovaj tip realnosti se koristi za igrice, videe u 360 stepeni recimo. Oprema za proširenu realnost je danas tako napravljena da najčešće koristi tzv. 6 stepeni slobode zahvaljujući spoljašnjim senzorima i na taj način ne samo da može da detektuje kretanje u određenom smeru već i pokrete

koje klijent pravi.

Pored ovako ozbiljne opreme postoji i oprema bazirana na mobilnim telefonima, kao što je *Google Daydream View*, ili *Oculus Go*. Ipak njihove mogućnosti su mnogo manjeg raspona jer svoje sposobnosti baziraju samo na telefonu. Za tako orijentisanu opremu se kaže da ima 3 stepena slobode, odnosno da može detektovati kretanje, ali ne i pokrete.

**Proširena realnost** (eng. augmented reality) sa druge strane, ne odvodi korisnika u neku drugu realnost, već oplemenjuje postojeću. Korisnik sa opremom ima uvid u to šta se dešava oko njega, ali može videti još neka dodatna (proširena) dešavanja. Tehnologija je prilagođena slobodnom pokretu. Oprema koja se koristi u ovom slučaju je uglavnom predstavljena kao prozirne naočare, kao što je, na primer, *Microsoft HoloLens*.



Slika 2. Microsoft HoloLens [2]

Jedna od aplikacija koja je idealna za demonstriranje mogućnosti proširene realnosti, a ujedno i aplikacija za kojom je postojalo planetarno interesovanje je *Pokemon Go* aplikacija. Ona nastaje



Slika 3. Primer pokemona u proširenoj realnosti [3]

2016. godine i biva pre svega namenjena mobilnim uređajima, odnosno *Android* i *iOS* platformama. Korisnici su mogli preko aplikacije da već u svom mestu traže pokemone, hvataju ih, organizuju borbe i treniraju ih.

U okviru ovog rada autori će govoriti o *React 360* frejmvorku koji se koristi za kreiranje virtuelne realnosti. Aplikacija nad kojom će biti demonstrirane njegove mogućnosti će biti virtuelna galerija. Kako je ovo aplikacija koja se izrađuje kao studentski projekat pridodat joj je i edukativni karakter tako da korisnik pored mogućnosti ispitavanja slike ima i mogućnost odgovaranja na pitanja u vezi sa slikom i čitanja detaljnih informacija o njoj samoj. Prilikom razvoja, aplikacija je testirana na računaru i mobilnom telefonu.

## 2 Frejmwork React 360

*React 360* predstavlja frejmwork za kreiranje *virtuelne realnosti*. Kao što se i iz samog imena može zaključiti frejmwork je baziran na *React* biblioteci čija je namena razvoj korisničkih interfejsa na veb-aplikacijama. Jedna od glavnih prednosti ovog frejmworka, *React 360*, je kombinovanje 2D i 3D entiteta. Kako se navodi u zvaničnoj dokumentaciji, osnove frejmworka su slične onima koje se koriste u *Oculus*-u i *Facebook*-ovim VR aplikacijama. [4]

Na samom početku je bitno napomenuti da pored *React 360* frejmworka postoji i biblioteka *React VR*. Ove dve tehnologije se neretko u govoru koriste naizmenično, iako nisu sinonimi. Godine 2017. *Facebook* i *Oculus* su odlučili da kreiraju javaskript biblioteku čija bi namena bila kreiranje 3D realnosti. *Oculus* je nastavio sa razvojem C++ frejmworka koji je danas poznat pod nazivom *React 360* je se njegov API razlikuje od prethodno pomenutog *React VR*-a.

### 2.1 Struktura aplikacije

Za razvoj aplikacije korišćen je editor *Visual Studio Code* na *Linux* platformi. Za kontrolu verzija aplikacije korišćen je sistem za kontrolu verzije *Git*, uz pomoć *Github*-a.

Aplikaciju je moguće generisati korišćenjem *react-360-cli* interfejsa, što su učinili i autori ovog rada. Pomenuti interfejs je neohodno instalirati globalno, ali da bi mogao da funkcioniše neophodno je imati instaliran i *Node* na računaru.

Sama aplikacija se izvršava u pretraživaču.

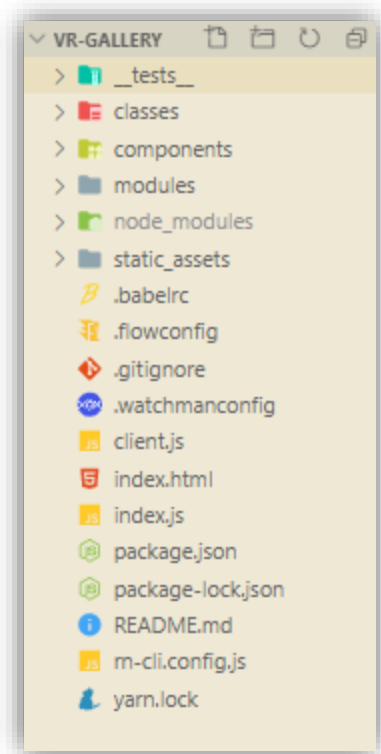
Nakon što je postavka završena, aplikaciju je moguće generisati komandom:

```
(1) react-360 init <ime-aplikacije>
```

i pokrenuti je sa:

```
(2) npm start
```

Struktura aplikacije generisane korišćenjem komandnog interfejsa je data na slici u nastavku:



Slika 4. Struktura foldera aplikacije

Svaka React 360 aplikacija se sastoji iz dva dela: *same React aplikacije i koda koji pretvara React komponente u 3D elemente na ekranu*. [5] Ovaj način razmišljanja je uzet iz *React Native*-a, a ispostaviće se da to nije jedina stvar koja je pozajmljena iz pomenutog frejmworka. Ideja sa razdvajanjem koda se zasniva na načinu funkcionisanja pretraživača. Naime, pretraživači tokom izvršavanja koriste samo jednu nit (*eng. thread*) tako da bi bilo koji zastoј u aplikaciji onemogućio njeno renderovanje. Izvršavanje koda aplikacije u zasebnom kontekstu omogućava renderovanje konstantno aktivnim velikom brzinom. Konkretno, u samoj aplikaciji, fajl koji ima tu namenu i automatski je generisan pomoću *react-360-cli* alata je *client.js*

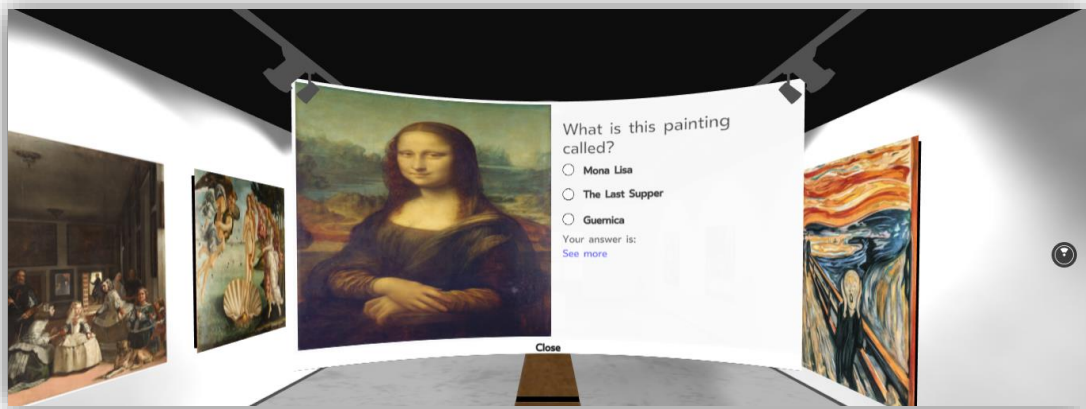
Kao što je već pomenuto, jedna od glavnih prednosti React 360 frejmworka je kreiranje 2D interfejsa u 3D prostoru o čemu će biti više reči u nastavku. Nakon toga biće govora o samim 3D objektima, Native Modulima i Audio komponenti frejmworka. Sve pomenute karakteristike frejmworka su našle svoju primenu u aplikaciji virtuelna galerija.

## 2.2 2D Interfejsi

Za kreiranje 2D interfejsa u React-u 360 koriste se tzv. *površine* (eng. surface). Osnove površina baziraju se *Oculus* tehnologiji i kao jednu od glavnih karakteristika navode mogućnost razmišljanja u pikselima kao merama razdaljine, a ne u fizičkim dimenzijama, što jeste slučaj kada se radi o 3D prostorima. Dakle, površine su definisane širinom i visinom u pikselima, kao i gustinom piskela u fizičkom prostoru.

Same površine se dele na cilindrične i ravne u zavisnosti od svog oblika.

**Cilindrična površina** (eng. Cylinder Surface) je takva da se stiče utisak gledanja u stranu cilindra. Taj cilindar, usled tehničkih postavki samog frejmvorka, ima poluprečnik osnove od četiri metra. Veoma je koristan kada je bitno prikazati sadržaj koji je sve vreme ispred korisnika, nezavisno od smera njegovog kretanja.



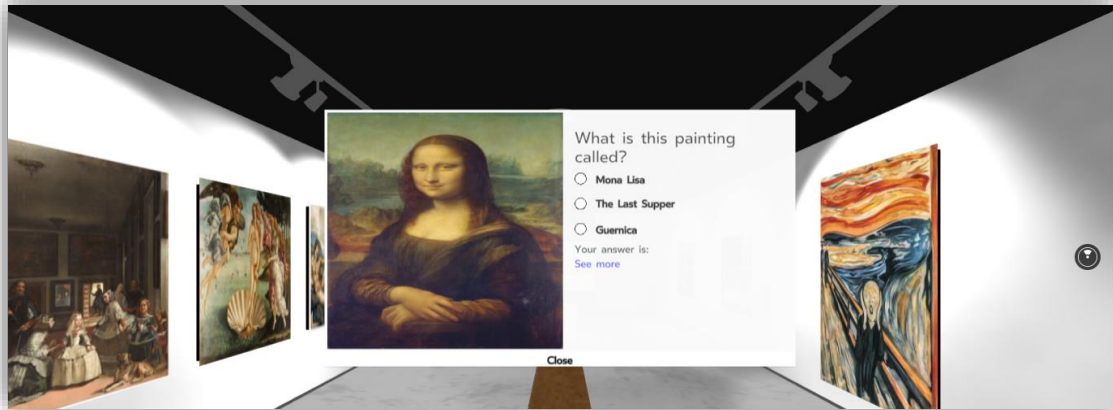
Slika 5. Prikaz modala na cilindričnoj površini

Ravna površina (eng. Flat Surface), kako i samo ime nalaže, nema taj cilindrični, zakrivljeni oblik. Takođe, razlikuje se od prethodnog tipa površine po tome što se ne nalazi sve vreme ispred korisnika; te ukoliko je to jedan od zahteva aplikacije neophodno je to posebno iskoristiti. Kako njena pozicija nije uvek nužno ispred korisnika onda postoje dodatna funkcija koja upravo služi pozicioniranju površine u prostoru. Ta funkcija se naziva *setAngle* i sa objašnjenjem parametara data je na isečku 1.

```
myFlatSurface.setAngle(  
  yawAngle, // ugao koji pozicionira površinu levo-desno  
  
  pitchAngle, // ugao koji pozicionira površinu gore-dole
```

```
rollAngle // ugao koji rotira površinu
);
```

Isečak 1. setAngle funkcija



Slika 6. Prikaz modala na ravnoj površi

Na isečku 2 se može videti kako se generiše jedna ravna površina i koji su koraci zaduženi za to. Interesantno je napomenuti da se za renderovanje koristi funkcija *renderToSurface* koja kao parametre dobija komponentu koja će se izrenderovati na površini, propertije koji će joj biti ubrizgani, kao i površinu na kojoj će se renderovanje dogoditi.

```
open(imageInfo) {
  if (this._surface) {
    return;
  }

  // Kreiranje površine
  this._surface = new Surface(800, 500, Surface.SurfaceShape.Flat);

  // Pozicioniranje
  this._surface.setAngle(0, 0);

  // Renderovanje
  this._render = this._r360Instance.renderToSurface(this._r360Instance.create
Root('ExplainedImage', { imageInfo, width: clientWidth }), this._surface);
}
```



Bitno je reći i da je moguće jednovremeno izrenderovati više površina. Za takav slučaj se koriste ravne površine jer ih je moguće pozicionirati na različitim lokacijama.

Kada se koriste površine treba imati na umu da se tada koristi CSS tehnika (*eng. skr.* Cascading Styling Sheet) raspoređivanja elemenata koja se naziva *Flexbox*. Element koji se iscertava na površini ne mora nužno biti slika, već to može biti i neka druga komponenta, kao što je dugme, ili tekst recimo. Kako bi se rasporedili elementi u okviru komponente koja ih potencijalno sadrži koristi se *Flexbox*. On je po difoltu postavljen u *View* komponenti koja se tipično koristi kao kontejnerska komponenta.

```
<View style={styles.questionWrapper}>
  <Text style={styles.question}>
    {this.props.imageInfo.question}
  </Text>

  {this.props.imageInfo.answers.map((answer, index) => (
    <View style={styles.answerWrapper} key={index}>
      <VrButton
        style={this.getButtonStyles(index)}
        onClick={() => {
          this.setState({ selectedAnswer: index });

          const sound =
            index === this.props.imageInfo.correctAnswer
              ? 'success.wav'
              : 'error.wav';

          AudioModule.playOneShot({
            source: asset(sound),
          });
        }}
      </VrButton>
      <Text style={styles.answerText}>{answer}</Text>
    </View>
  ))}
</View>
```

Isečak 3. Korišćenje View komponente

Na sve ove komponente je moguće primeniti CSS stilove, ali u React Native notaciji. Tako da je React Native dokumentacija stilizovanja ovde bila u potpunosti korišćena kada je bilo potrebe za stilizovanjem.

```
export default StyleSheet.create({

  scaledStyle: {
    transform: ([{ scaleX: 0.6 }, { scaleY: 0.6 }])
  },

  outerWrapper: {
    flexDirection: 'column',

    width: 400,
    backgroundColor: 'blue',
  },

  explainWrapperRow: {
    opacity: 0.9,
    flexDirection: 'row',
    backgroundColor: 'white',
    borderColor: 'white',
    borderWidth: 5,
  },

  poll: {
    flexDirection: 'column',
    backgroundColor: 'white',
    opacity: 0.95,
    width: 390,
  }
  .
  .
  .
```

Isečak 4. Postavljanje stilova

## 2.3 3D Objekti

Kao što je već pomenuto, u React-u 360 moguće je renderovati i 3D objekte. Ono što jeste ograničavajuća karakteristika frejmworka je da se 3D objekti ne mogu generisati u samom frejmworku. Ujedno je to i razlog zašto frejmwork nema primitivne 3D elemente kao što su sfera, ili kocka. Ukoliko postoji potreba da se u projekat uključi neki 3D objekat neophodno je da je on

prethodno bio generisan u nekom drugom alatu za modelovanje. Za korišćenje 3D objekata se koristi komponenta *Entity*.

```
<Entity
  style={{
    transform: [
      { scale: 1 },
      { translate: [this.state.x, 0, this.state.z] },
    ],
  }}
  source={{
    gltf2: asset('scene.gltf'), // scena koja se postavlja
  }}
/>

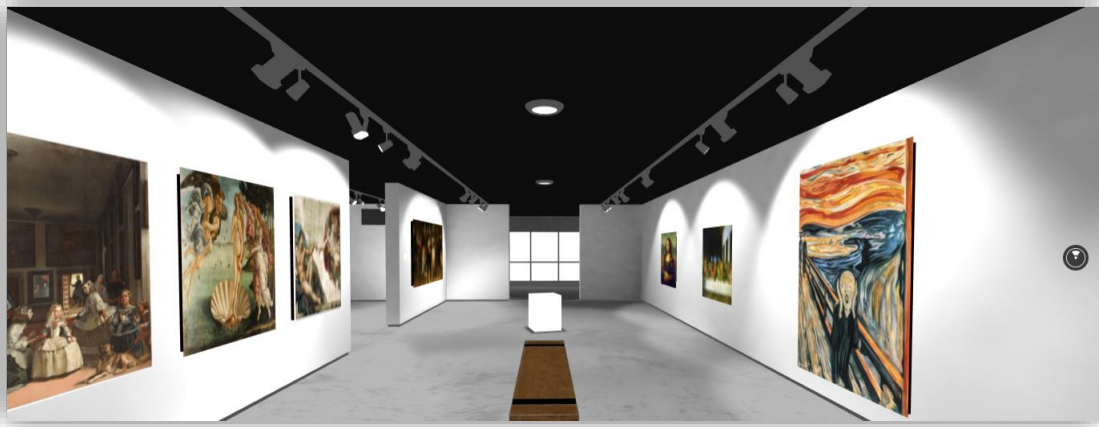
.
.
.

const customLocation = new Location([0, -1, 0]);
r360.renderToLocation(r360.createRoot('vr_gallery'), customLocation);
```

Isečak 5. Dodavanje 3D entiteta u aplikaciju

Za razliku od 2D interfejsa koji se postavljaju na površine, 3D objekti se postavljaju na *lokacije* (eng. location). One predstavljaju tačku u 3D prostoru gde će se *zakačiti* entitet. Bitno je, takođe, napomenuti da kada se radi o mernim jedinicama za lokaciju govori se u metrima, ne u pikselima kao kod površina.

Za potrebe projekta virtuelne galerije autori rada su kupili 3D model sobe koje su iskoristili za postavljanje slika.



Slika 7. 3D model sobe

## 2.4 Native moduli

Native moduli omogućavaju komunikaciju između React aplikacije i glavne niti koja renderuje aplikaciju i time omogućava korišćenje funkcionalnosti koje su dostupne u okruženju pretraživača. Primeri korišćenja native modula su dobijanje informacija o okruženju, manipulacija okruženjem, dobijanje informacija o kontrolerima, itd. Native moduli se kreiraju za runtime code (`client.js`) i potrebno ih je registrovati pre njihovog korišćenja od strane React aplikacije [6].

Za kreiranje Native modula potrebno je kreirati klasu koja nasleđuje `Module` klasu iz `react-360` okruženja. Na isečku 6 prikazan je primer native modula koji se koristi u projektu.

```

class MoveButtonsModule extends Module {

  constructor(ctx) {
    super('MoveButtonsModule');

    this._ctx = ctx;
  }

  _setAngle(horizAngle, vertAngle) {
    if (!!this._surface) {
      this._surface.setAngle(horizAngle, vertAngle);
    }
  }

  _setInstance(r360Instance) {
    this._r360Instance = r360Instance;

    const clientWidth = this._r360Instance._eventLayer.clientWidth;
    const clientHeight = this._r360Instance._eventLayer.clientHeight;

    this._surface = new Surface(clientWidth, clientHeight,
    Surface.SurfaceShape.Flat);
    this._surface.setAngle(0, 0);
    this._surface.setRadius(0.1);
    this._surface.setDensity(5000);

    this._r360Instance.renderToSurface(this._r360Instance.createRoot('MoveButtons
    ', { width: clientWidth, height: clientHeight }), this._surface);
  }
}

```

Isečak 6. Primer native modula

Module se registruje u client.js fajlu prilikom kreiranja React instance. Nakon registrovanja native modula njega je moguće koristiti u okviru React koda. Na isečku 7 prikazano je korišćenje native modula u okviru React koda.

```

import React from 'react';
import {
  NativeModules,
} from 'react-360';

const { ExplainedImageModule } = NativeModules;

export default class ExplainedImage extends React.Component {
  .
  .
  .
  render() {
    return (
      .
      .
      .
      <VrButton
        onClick={() => ExplainedImageModule.close()}
        style={styles.closeButton}
      >
        <Text style={styles.closeText}>Close</Text>
      </VrButton>
    </View>
    );
  }
}

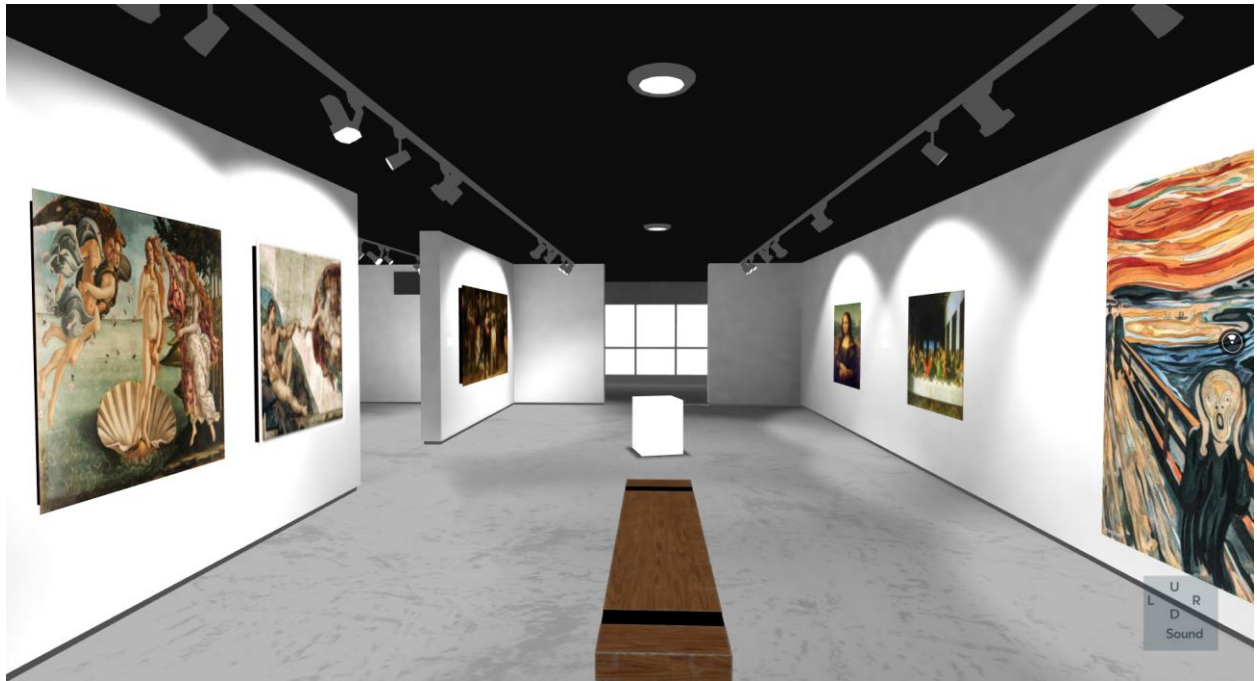
```

Isečak 7. Korišćenje native modula u okviru React koda

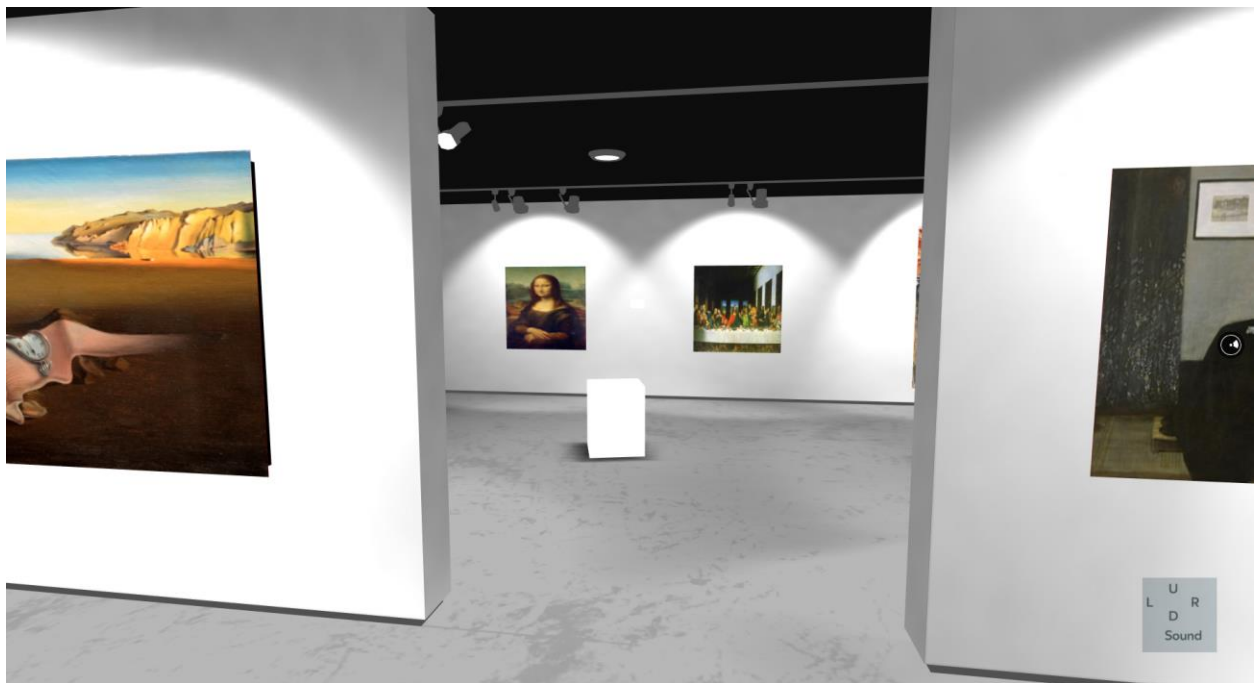
React kod može iz native modula da pročita vrednost neke konstante, da pozove funkciju native modula i da pozove funkciju koja vraća određeni odgovor React kodu.

U okviru projekta native moduli su korišćeni za implementaciju kretanja kroz galeriju i renderovanje modula za različite veličine ekrana. Kretanje je implementirano tako da kad se pritisne dugme na tastaturi native module to registruje i menja trenutne koordinate x i z. Takođe dodati su dugmići na ekranu za kretanje ukoliko se koristi uređaj koji nema tastaturu. React kod na svakih 50ms proverava vrednosti tih koordnata i kad dođe do promene radi translaciju kompletne scene. Drugi pristup za implemetaciju kretanja bio je pomeranjem kamere, ali problem kod tog pristupa je to što se surface renderuje u odnosu na 0, 0, 0 (x, y, z) kooridnate kamere pa je potrebna dodatna logika za izračunavanje pozicije surface-a preko 0, 0, 0 kooridnata i trenutne pozicije kamere.

Na slikama 8 i 9 prikazano je kretanje kroz sobu. U donjem desnom uglu vide se dugmići za kretanje ukoliko uređaj nema tastaturu.



Slika 8. Početna lokacija



Slika 9. Kretanje po sobi

Drugi primer korišćenja native modula u okviru projekta je renderovanje surface-a za različite veličine ekrana uređaja. U ovo slučaju native module se koristi kako bi React kod pročitao informacije o pretraživaču, njegovu širinu i visinu kako bi se skalirao surface.

## 2.5. Audio

U React 360 aplikaciju je veoma jednostavno moguće dodati i audio komponentu. U virtualnoj galeriji, koja sve vreme služi kao primer, postoji zvuk okruženja (*eng.* environmental) koji sve vreme ide u pozadini, kao i kraći zvuk koji se okida prilikom manipulacije kontrolama na anketi sa pitanjima u vezi sa slikama.

```
const { AudioModule } = NativeModules;

AudioModule.playOneShot({ // jednokratno puštanje zvuka
  source: asset(sound),
});

handleSound() {
  !this.state.hasSound
    ? AudioModule.playEnvironmental({ // pokretanje zvuka atmosfere
      source: asset('environmental.mp3'),
      volume: 0.5,
    })
    : AudioModule.stopEnvironmental(); // zaustavljanje zvuka atmosfere

  this.setState({hasSound: !this.state.hasSound});
}
```

Isečak 8. Dodavanje zvuka

Da bi se koristila audio komponenta neophodno je prvo uključiti Audio modul u projekat, a potom koristiti API koji on obezbeđuje.



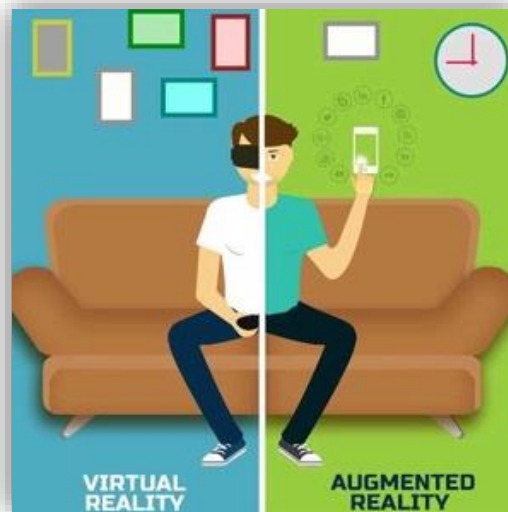
### 3 Zaključak

U okviru ovog rada data je paralela između virtualne i proširene realnosti govoreći o njihovim osnovnim karakteristikama. Nakon toga je čitalac uveden u tehnologiju koja se koristila za razvoj aplikacije čija je namena kreiranje virtualne galerije. U toku rada govoreno je o najbitnijim karakteristikama korišćene tehnologije, uz primere korišćenja, kao i slike rezultata primene tog koda.

Na kraju rada može se zaključiti da je React 360 alat koji se relativno jednostavno može savladati. Daje mogućnost kreiranja pravih 3D aplikacija koje se izvršavaju u pretraživaču. Dodatnu olakšavajuću okolnost predstavlja i činjenica da je baziran na trenutno najpopularnijoj biblioteci za razvoj korisničkih interfejsa na veb-aplikacijama, *React-u*.

Utisak autora je da bi ipak bilo poželjno da se za neke ozbiljnije aplikacije, sa većom kompleksnošću, koriste neke druge VR/AR tehnologije zbog jednostavnosti React 360 frejmvorka. Ta jednostavnost indukuje i glavni nedostatak alata, a to je otežavanje implementacije nekih fičera, iz veoma prostog razloga – nepostojanja mogućnosti implementacije. Kao što je to primer sa kretanjem kroz galeriju u ovom slučaju, ili nemogućnost kreiranja sopstvenih 3D objekata. React 360 je takav da je dobar za sklapanje već postojećih objekata, ali ne i za ručno pravljanje istih od početka.

Svakako, sa sigurnošću se može reći da je odličan za početnike i entuzijaste prema virtualnoj realnosti i da može podstaknuti developere da se zainteresuju za neke druge tehnologije koje su na višem nivou kompleksnosti.



Slika 10. Virtualna i proširena realnost [6]

## 4 Literatura

- [1] HTC Vive review, <https://www.theverge.com/2016/4/5/11358618/htc-vive-vr-review>, avgust 2020
- [2] HoloLens 2 pricing and options, <https://www.microsoft.com/en-us/hololens/buy>, avgust 2020
- [3] Wikipedia, strana Pokemon Go, [https://en.wikipedia.org/wiki/Pok%C3%A9mon\\_Go#/media/File:Pok%C3%A9mon\\_Go\\_AR\\_Mode,\\_Dec\\_2017.png](https://en.wikipedia.org/wiki/Pok%C3%A9mon_Go#/media/File:Pok%C3%A9mon_Go_AR_Mode,_Dec_2017.png), avgust 2020
- [4] What is React 360, <https://facebook.github.io/react-360/docs/what-is.html>, avgust 2020
- [5] Runtime, <https://facebook.github.io/react-360/docs/runtime.html>, avgust 2020
- [6] Native modules, <https://facebook.github.io/react-360/docs/native-modules.html>, avgust 2020
- [7] Virtual Reality vs Augmented Reality – What’s The Difference, <https://virtualrealitiespop.com/virtual-reality-vs-augmented-reality-whats-the-difference-a650a4b3895f>, avgust 2020