SVEUČILIŠTE U ZAGREBU FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 3567

Virtualna Mala dvorana Koncertne Dvorane Vatroslava Lisinskog

Luka Kunić

Zagreb, lipanj 2014.

SVEUČILIŠTE U ZAGREBU FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA ODBOR ZA ZAVRŠNI RAD MODULA

Zagreb, 11. ožujka 2014.

ZAVRŠNI ZADATAK br. 3567

Pristupnik: Luka Kunić (0036468102) Studij: Računarstvo Modul: Računarska znanost

Zadatak: Virtualna Mala dvorana Koncertne Dvorane Vatroslava Lisinskog

Opis zadatka:

Proučiti korisnička sučelja koja omogućuju kretanje virtualnim prostorima. Proučiti tehnike interakcije objektima u takvom okruženju. Razraditi model Male dvorane Koncertne dvorane Vatroslava Lisinskog. Ostvariti programsku implementaciju koja omogućuje prikaz virtualnog prostora te kretnje tim prostorom i interakciju s objektima. Koristiti različite tehnike i tehnologije korisničkog sučelja, Načiniti ocjenu ostvarenih rezultata.

Izraditi odgovarajući programski proizvod. Rezultate rada načiniti dostupne putem Interneta. Radu priložiti algoritme, izvorne kodove i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Zadatak uručen pristupniku: 14. ožujka 2014. Rok za predaju rada: 13. lipnja 2014.

Mentor:

Fron A M Prof.dr.sc. Željka Mihajlović

Dielovođa:

Doc.dr.sc. Tomislav Hrkać

Predsjednik odbora za završni rad modula:

Prof.dr.sc. Siniša Srbljić

SADRŽAJ

1.	Uvod	1					
2.	Virtualni prostor	2					
	2.1. Primjene	3					
	2.2. Stvaranje virtualnog prostora	4					
3.	Kretanje virtualnim prostorom						
	3.1. Fizičke ulazne jedinice	5					
	3.2. Izravno upravljanje	6					
	3.3. Virtualne ulazne jedinice	8					
	3.4. Slobodno kretanje	8					
	3.5. Upravljanje glasovnim naredbama	9					
4.	Programska implementacija	11					
	4.1. Korišteni alati	12					
	4.2. Model dvorane	12					
	4.3. Glasovne naredbe	14					
	4.4. Slobodno kretanje dvoranom	17					
	4.5. Kretanje glasovnim naredbama	19					
5.	Rezultati	21					
	5.1. Preciznost glasovnog upravljanja	21					
	5.2. Usporedba mikrofona	22					
	5.3. Usporedba hrvatskog i engleskog jezika	23					
6.	Zaključak	24					
7.	Literatura	25					

1. Uvod

Posljednjih godina, tehnologije proizvodnje grafičkog sklopovlja napreduju velikom brzinom, što omogućuje izradu sve bržih i efikasnijih grafičkih procesora. Kao posljedica tog napretka, danas je moguće izraditi vrlo kompleksne virtualne prostore, u stvarnom se vremenu njima kretati i na različite načine upravljati objektima u prostoru.

Najraširenija primjena virtualnih prostora je u računalnim igrama, za prikaz virtualnih svjetova, a mnogo su korisnije primjene u arhitekturi i simulatorima kojima se mogu simulirati situacije iz stvarnog svijeta. Prednost takvih simulacija je mala cijena u odnosu na stvarni svijet te sigurni i kontrolirani uvjeti, zbog čega se koriste kao zamjena za skupe ili opasne radnje. Također, virtualni prostori mogu se koristiti i u situacijama kad nije praktično ili moguće kretati se stvarnim prostorom. Primjerice, ukoliko korisnik nema izravan pristup nekom prostoru, može se izraditi virtualna replika tog prostora koja korisniku omogućuje kretanje i interakciju s objektima.

Način interakcije s virtualnim prostorom i objektima ovisi o njihovoj svrsi i o okruženju u kojem se koriste. Ponekad nije praktično virtualnim prostorom upravljati pomoću tipkovnice i miša, već je jednostavnije koristiti alternativne načine upravljanja, primjerice glasovne naredbe ili uređaj s ekranom osjetljivim na dodir.

U ovom su radu razmotrene neke tehnike kretanja virtualnim prostorima i načini interakcije s objektima u virtualnom okruženju. Izrađen je model Male dvorane Koncertne Dvorane Vatroslava Lisinskog te su implementirana dva načina kretanja tim prostorom — upotrebom tipkovnice i miša za slobodno kretanje prostorom i upotrebom glasovnih naredbi na hrvatskom i engleskom jeziku.

2. Virtualni prostor

Virtualni prostor ne nalazi se u fizičkom svijetu već u memoriji računala. To može biti replika stvarnog prostora koji postoji u fizičkom svijetu ili novi, izmišljeni prostor. Bez obzira postoji li prostor u stvarnom svijetu ili ne, računalni model pruža veliku slobodu kretanja i interakcije s objektima u tom prostoru. Korištenje virtualnih prostora omogućuje rješavanje problema iz stvarnog svijeta na siguran, kontroliran i jeftin način, ali tehnologija još nije dovoljno napredovala kako bi korisnik potpuno mogao uroniti u virtualni prostor. Kako bi se postigla virtualna stvarnost, potrebno je pobuditi sva osjetila korisnika i pronaći jednostavne i intuitivne načine interakcije s virtualnim objektima prilagođene domeni primjene i okolini u kojoj će se koristiti.



Slika 2.1: Primjer virtualnog prostora

Nedostatak osjećaja dodira prilikom interakcije s objektima jedan je od većih problema prilikom postizanja virtualne stvarnosti. Korištenjem uobičajenih metoda za upravljanje virtualnim objektima, kao što su tipkovnica i miš, korisnik ne može dobiti povratnu informaciju o dodiru prilikom interakcije s objektima u prostoru. U raznim simulatorima koriste se replike stvarnih uređaja kako bi korisnik imao dojam da upravlja tim uređajima (npr. kabina aviona u simulatoru za pilote), a uz to se koriste motori koji simuliraju vibracije i hidraulički sustavi koji simuliraju G-sile.

Pomoću naglavnih uređaja za prikaz virtualne stvarnosti (engl. *head-mounted displays*) kao što je primjerice *Oculus Rift*, svako oko dobiva svoju sliku i pojačava se dojam trodimenzionalnosti virtualnog prostora koji se prikazuje. Uređaj prati pomake glave i koristi dobivene podatke kako bi pomicao očište virtualne kamere kojom se gleda virtualni prostor. Takvi uređaji nisu još dovoljno razvijeni da bi pružili realističnu sliku virtualnog prostora, prvenstveno zbog malih rezolucija ekrana.

2.1. Primjene

Virtualni prostori najrašireniju primjenu imaju u računalnim igrama, a upravo zbog industrije igara i želje za što kompleksnijim i kvalitetnijim prikazima znatno su napredovali grafički procesori koji prikazuju virtualne prostore. Napredak tehnologije omogućilo je otvaranje novih grana korištenja virtualnih prostora, kao što su simulacije u stvarnom vremenu, računalne animacije i interaktivne vizualizacije.

Virtualni prostori vrlo su bitni za simuliranje određenih radnji iz fizičkog svijeta koje je skupo, teško ili opasno izvesti u stvarnosti, a napretkom tehnologije te simulacije postaju sve vjernije i bliže stvarnom svijetu. Upotrebom simulacija ne nastoji se samo prikazati virtualni prostor, već pružiti korisniku potpun osjećaj virtualne stvarnosti. Primjerice, simulatore koriste piloti kako bi se pripremili prije nego što prvi puta lete u pravim avionima, padobranci kako bi savladali potrebne tehnike na siguran način prije prvog skoka, a vozači Formule 1 kako bi se upoznali s novim stazama prije utrke.

U arhitekturi i dizajnu, interaktivni virtualni prikaz objekata velik je napredak u odnosu na statičke skice na papiru. Lakše je i jeftinije rano uočiti nedostatke i unositi izmjene na projektu jer autor može po želji upravljati virtualnim objektom i uočiti detalje koje bi inače uočio tek prilikom izrade objekta.

Virtualni prikaz prostora vrlo je bitan i u filmskoj industriji za izradu detaljnih scena čija bi izrada u stvarnom svijetu bila teška ili nemoguća, te za proširivanje stvarnih prostora kako bi se doimali znatno veći i kompleksniji.

2.2. Stvaranje virtualnog prostora

Za izradu virtualnih prostora koriste se alati za 3D modeliranje, kao što su primjerice Autodesk 3ds Max i Maya, Blender i drugi. Osim tih alata, prilikom modeliranja prostora iz stvarnog svijeta, mogu se koristiti 3D skeneri koji snimaju prostor u tri dimenzije i pohranjuju podatke o prostoru u obliku oblaka točaka (engl. point cloud). S obzirom na veliki broj točaka skeniranog modela, potrebno ga je pretvoriti u model poligonima korištenjem alata za 3D modeliranje kako bi bio pogodan za interakciju u realnom vremenu. Zatim je potrebno omogućiti korisniku kretanje tim prostorom i interakciju s objektima. To je lako postići upotrebom pogonskih alata (engl. engines) koji nude mogućnost izgradnje cijele 3D scene i programska sučelja za programiranje interakcije s virtualnim objektima. Jedan od takvih alata je Unity, koji će biti korišten za kretanje virtualnom dvoranom u praktičnom dijelu ovog rada.

3. Kretanje virtualnim prostorom

Interakciju s objektima i kretanje virtualnim prostorom moguće je ostvariti na različite načine. Različite tehnike upravljanja pogodne su za različite radnje unutar virtualnog prostora:

- Fizičke ulazne jedinice upravljanje pokretima u virtualnom prostoru pomoću fizičkih uređaja spojenih na računalo
- Izravno upravljanje pokreti korisnika u stvarnom svijetu izravno se preslikavaju na pokrete u virtualnom prostoru
- Virtualne ulazne jedinice korisnik upravlja pokretima u virtualnom svijetu pomoću virtualnih ulaznih naprava koje su iscrtane na ekranu i ne postoje u fizičkom svijetu

U nastavku su opisani najčešći načini upravljanja te njihove prednosti i mane.

3.1. Fizičke ulazne jedinice

Fizičke ulazne jedinice mogu biti jednostavne poput tipkovnice i miša ili složenije poput volana za simulaciju vožnje i čitave kabine aviona za simulaciju leta. Pogodne su za precizno upravljanje virtualnim objektima, ali je teško izravno preslikati naredbe koje se šalju preko ulazne jedinice u virtualni prostor kojim upravljamo. Zbog toga neke radnje, koje je jednostavno izvesti u fizičkom svijetu, može biti izuzetno zahtjevno izvesti u virtualnom okruženju. Osim toga, jednostavnije ulazne jedinice ne nude taktilnu povratnu informaciju korisniku tijekom upotrebe, a i komplicirane su za korištenje ukoliko korisnik nosi naglavni uređaj za prikaz virtualnog prostora, budući da ih ne može vidjeti.

Upravljanje tipkovnicom i mišem jedan je od najraširenijih načina upravljanja virtualnim prostorima, pogotovo u računalnim igrama. Mišem korisnik upravlja smjerom gledanja, a tipkovnicom upravlja brzinom i smjerom kretanja. Kretanjem u virtualnom prostoru može se upravljati i samo mišem, tako da se klikom na objekt u prostoru korisnik približi odabranom objektu. Upravljanje tipkovnicom i mišem pogodno je i za pregledavanje modela u alatima za 3D modeliranje zbog jednostavnosti i brzine kretanja kroz prostor, no zbog ograničenosti kretanja miša u dvije dimenzije po površini stola, takav način upravljanja nije prirodan. Alternativan način upravljanja su posebni 3D miševi koji omogućuju kretanje u sve tri dimenzije istovremeno, te tako olakšavaju rad u alatima za 3D modeliranje.



Slika 3.1: 3D miš za upravljanje pogledom u alatima za 3D modeliranje

Postoje i specijalizirani uređaji koji su pogodni samo za kretanje određenim vrstama virtualnih prostora. Primjer takvih uređaja su volani ili kabine raznih vozila, pomoću kojih korisnik može upravljati simulacijama vožnje na posve prirodan način. Takvi uređaju nisu praktični za kretanje u virtualnim prostorima u kojima se ne simulira vožnja, ali u odnosu na tipkovnicu i miš, nude znatno preciznije i prirodnije upravljanje vozilima u virtualnom prostoru.

3.2. Izravno upravljanje

Izravno upravljanje u virtualnom prostoru podrazumijeva preslikavanje pokreta iz stvarnog svijeta u virtualni, pri čemu su potrebni odgovarajući senzori, kao što su *Microsoft Kinect* ili *Leap Motion Controller*. *Kinect* sadrži kameru, senzor dubine i mikrofon, pomoću kojih prepoznaje pokrete, izraze lica i prima glasovne naredbe. Takvim uređajem moguće se na prirodniji način kretati u virtualnom prostoru, jer uređaj prepoznaje pokrete i geste koji se izravno preslikavaju u virtualni prostor. Nedostatak *Kinecta* je mali domet senzora od 2-5m, što otežava kretanje većim virtualnim prostorima, kao i nepreciznost senzora koji još nisu dovoljno razvijeni kako bi savršeno prepoznali pokrete korisnika. *Leap Motion Controller* prepoznaje geste koje korisnik izvodi rukama, uz pomoć infracrvenih kamera kojima prati prostor iznad uređaja. Prepoznaje pokrete prstima i razne geste slične onima koje prepoznaju ekrani osjetljivi na dodir, ali ima iznimno kratak domet, manji od 1m.



Slika 3.2: Leap Motion Controller, uređaj za prepoznavanje pokreta rukom

Kretanje virtualnim prostorom koristeći prirodne pokrete tijela vrlo je intuitivno te korisnik ne mora razmišljati o nekim radnjama koje su u stvarnosti automatske. Jedan korak u stvarnom svijetu, očitan senzorom za praćenje pokreta, može odgovarati točno jednom koraku u virtualnom prostoru, zbog čega se pojačava dojam stvarnosti virtualnog prostora. Nedostatak takvog načina upravljanja je usko područje unutar kojeg senzori prate pokrete korisnika. Virtualni prostori mogu biti vrlo veliki i nije se moguće kretati cijelim virtualnim svijetom koristeći samo pokrete iz stvarnog svijeta, već su potrebni alternativni načini kretanja, kao što su geste i druga pomagala.

Geste omogućuju kretanje pokretima ruku i tijela, ali pokreti se više ne prenose izravno u virtualni prostor, već svaka gesta pokreće unaprijed definiranu radnju u virtualnom prostoru. Primjerice, korisnik može pokrete rukom koristiti za određivanje smjera kretanja ili za odabir objekta u prostoru kojem se želi približiti. Ovakav način upravljanja i dalje je intuitivniji od upotrebe fizičkih ulaznih jedinica, ali nije dovoljno precizan ni intuitivan kao što je upravljanje izravnim preslikavanjem pokreta u virtualni prostor. Uz geste, korisnik može koristiti i dodatna pomagala, kao što su naglavni uređaji za prikaz opremljeni senzorima pokreta. Upotrebom takvih uređaja korisnik pomacima glave može odrediti smjer gledanja u virtualnom prostoru, što znatno olakšava kretanje upotrebom gesti.

Bez obzira na nedostatke, uređaji za praćenje pokreta korisnika nude znatno prirodniji način upravljanja objektima i kretanja virtualnim prostorom te će za određene primjene u budućnosti biti dobra alternativa fizičkim ulaznim jedinicama.

3.3. Virtualne ulazne jedinice

Ulazne jedinice za upravljanje pokretima ne moraju postojati u stvarnom svijetu kako bi ih korisnik mogao koristiti. Ulazne jedinice mogu postojati u samom virtualnom prostoru ili mogu biti dio grafičkog korisničkog sučelja aplikacije. Virtualne ulazne jedinice mogu biti tipke iscrtane na ekranu, polja za unos parametara ili cijele upravljačke ploče smještene u virtualnom prostoru. Korisnik njima može upravljati koristeći ekran osjetljiv na dodir, tipkovnicu i miš ili neke druge jednostavne vanjske ulazne jedinice.

Upotreba virtualnih ulaznih jedinica može biti komplicirana jer korisnik ne može dobiti jednake povratne informacije od virtualnog uređaja kakve može dobiti od stvarnog uređaja. Također, teško je implementirati velike virtualne ulazne jedinice zbog ograničenja u veličini korisničkog sučelja.

3.4. Slobodno kretanje

Slobodno kretanje virtualnim prostorom može se poistovjetiti s letom. Korisnik se može kretati u svim smjerovima, kao da leti kroz virtualni prostor, pri čemu su bitni smjer pogleda te smjer i brzina kretanja.

Smjer pogleda određuje što će biti u vidnom polju kamere prilikom kretanja virtualnim prostorom. Smjerom pogleda može se upravljati upotrebom vanjskih ulaznih jedinica, praćenjem pokreta i gestama ili upotrebom naglavnih uređaja za prikaz sa senzorima pokreta. Odabir tehnike upravljanja smjerom pogleda ovisi o području primjene i potrebe za preciznošću s jedne strane, a jednostavnošću i brzine upravljanja s druge. Također, smjer pogleda može biti jednoznačno određen točkom ili objektom koji se nalazi u virtualnom prostoru, tako da kamera uvijek gleda u smjeru te točke ili objekta, bez obzira na trenutni položaj.

Smjer i brzina kretanja virtualnim prostorom određeni su smjerom i iznosom vektora kretanja, a korisnik upotrebom ulaznih jedinica, pokreta ili gesti upravlja parametrima tog vektora. Ukoliko je smjer kretanja određen smjerom u kojem gleda kamera, korisnik će se kretati usmjeravanjem kamere u željenom smjeru kretanja i određivanjem brzine. Moći će se kretati samo naprijed, a pri svakoj promjeni smjera gledanja će se promijeniti i smjer kretanja. Premda takav način kretanja ima svoje primjene, poželjnije je da smjer kretanja i smjer pogleda budu neovisni. U tom slučaju, korisnik će imati slobodu kretanja u svim smjerovima bez obzira na smjer u kojem gleda, što je prirodnije i bliže kretanju u stvarnom svijetu.

3.5. Upravljanje glasovnim naredbama

U određenim situacijama nije praktično koristiti fizičke ulazne jedinice poput tipkovnice i miša za kretanje virtualnim prostorom i interakciju s virtualnim objektima. Potrebno je naći alternativne načine upravljanja koji su precizni, brzi i jednostavni za korištenje, a glasovno upravljanje može biti vrlo dobro rješenje za određene primjene.

Velika prednost glasovnog upravljanja je jednostavnost korištenja, budući da korisnik ne upravlja sustavom ručno, već zadaje glasovne naredbe koje sustav prepoznaje, pri čemu se izvodi jedna ili više radnji koje odgovaraju dobivenoj naredbi. Uz to, naredbe su riječi i rečenice iz prirodnog jezika, čime se dobiva velika sloboda upravljanja u odnosu na ulazne jedinice koje imaju ograničen broj upravljačkih elemenata. Svaka rečenica prirodnog jezika je potencijalna glasovna naredba, što daje gotovo beskonačan broj naredbi koje se mogu zadati sustavu kojim se upravlja.

Što bi bile radnje koje se pokreću glasovnim upravljanjem? One mogu biti

jednostavni pokreti kojima se korisnik kreće u virtualnom prostoru i upravlja objektima, analogno kretanju upotrebom raznih ulaznih jedinica. Problem takvog načina upravljanja je nedovoljna preciznost, zbog čega ne bi bio praktičan za korištenje. Za takve pokrete, bilo bi prikladnije koristiti druge načine upravljanja kao što su fizičke ulazne jedinice ili geste. S druge strane, jedna glasovna naredba može se koristiti za pokretanje većeg broja povezanih radnji, što je znatno efikasnije od ručnog pokretanja zasebnih radnji.

4. Programska implementacija

Praktični zadatak ovog rada bio je izraditi 3D model Male dvorane Koncertne Dvorane Vatroslava Lisinskog te implementirati različite načine kretanja i interakcije s objektima u tom virtualnom prostoru. Model je izrađen alatom *Blender*, na temelju tlocrta i fotografija slobodno dostupnih na web stranicama dvorane Lisinski. Interakcija s modelom implementirana je upotrebom alata *Unity* i programskog jezika C#, a za slušanje glasovnih naredbi napisana je aplikacija koja koristi *Speech.Recognition* programsko sučelje unutar .*NET* radnog okvira.



Slika 4.1: Model Male dvorane Koncertne Dvorane Vatroslava Lisinskog

U nastavku su pobliže opisani dijelovi implementacije programskog rješenja i alati korišteni za njihovu izradu.

4.1. Korišteni alati

Model dvorane izrađen je upotrebom *Blendera*, besplatnog skupa alata otvorenog koda koji, među ostalim, sadrži alate za izradu 3D modela, animaciju, simulaciju fizike, praćenje pokreta i iscrtavanje. Budući da je otvorenog koda, *Blender* nudi programsko sučelje za pisanje skripti u programskom jeziku Python kojima je moguće dodavati vlastite alate (engl. *add-ons*) i unaprijediti postojeće. To ga čini vrlo moćnim sustavom, zbog čega u velikom broju primjena može biti dostojna zamjena komercijalnim rješenjima kao što su *Autodesk 3ds Max* i *Maya*. Dostupan je na http://www.blender.org.

Za izradu aplikacije koja sluša glasovne naredbe korištena je radna okolina *Visual Studio 2012* tvrtke *Microsoft* uz .*NET*, njihov radni okvir za izradu aplikacija za operacijski sustav *Windows*. U sklopu tog radnog okvira uključen je programski paket *Speech.Recognition*, koji sadrži razrede i metode za slušanje i prepoznavanje govornog jezika.

Za implementaciju kretanja i interakcije s objektima u virtualnom prostoru dvorane, korišten je pogonski alat *Unity*. Glavna namjena tog alata je izrada računalnih igara, a omogućuje jednostavnu izradu aplikacija za više različitih platformi. Alat je komercijalan, ali postoji i besplatna inačica koja nudi većinu funkcionalnosti, dovoljnih za izradu jednostavnih aplikacija i igara. Više informacija o alatu i besplatna verzija nalaze se na http://www.unity3d.com.

4.2. Model dvorane

Dvorana je izrađena na temelju tlocrta i fotografija dostupnih na web stranicama dvorane Lisinski. Prvo su prema tlocrtu napravljeni obrisi same dvorane, pozornica i gledalište, nakon čega je bilo potrebno u gledalište dodati sjedala. Pošto se u dvorani nalazi oko 300 sjedala, model sjedala izrađen je zasebno, a zatim je upotrebom funkcije *Link* dodan u scenu. Takav način dodavanja modela u scenu omogućuje da svaka izmjena na vanjskom modelu odmah bude vidljiva na svim objektima tog modela u sceni.

Zatim je bilo potrebno odrediti pozicije i orijentacije sjedala u gledalištu. Nakon što su sva sjedala bila točno pozicionirana, u scenu su dodani prazni objekti (engl. *empty*), pozicionirani i orijentirani po sjedalima, kako bi se omogućilo dinamičko odlučivanje o tome koja sjedala će biti prikazana u konačnoj aplikaciji. Svakom praznom objektu koji predstavlja sjedalo pridijeljeno je odgovarajuće ime, kako bi se pojednostavio odabir sjedala prilikom pozicioniranja kamere na željenu poziciju upotrebom glasovnih naredbi. U scenu je dodan još jedan prazan objekt koji je postavljen kao hijerarhijski roditelj svim sjedalima, što omogućuje brz prelazak preko svih sjedala u dvorani prilikom dinamičkog postavljanja.



Slika 4.2: Izrada modela dvorane u Blenderu

Na kraju, u model dvorane dodani su ostali objekti koji nisu bitni za konstrukciju, nakon čega su modelu sjedala i svim objektima u modelu dvorane pridijeljene teksture. Teksture su izrađene upotrebom alata *GIMP*, a sastoje se od tri komponente: difuzne, spekularne i mape normala. Prije dodavanja tekstura, objekte je bilo potrebno razmotati (engl. *UV unwrap*), odnosno prikazati 3D objekt u 2D prostoru kako bi se poligonima objekta mogla pridijeliti 2D tekstura. Teksture se dodaju upotrebom materijala (engl. *materials*) u *Blenderu*, a za kombiniranje komponenata tekstura koristi se uređivač čvorova (engl. *node editor*) koji omogućuje kreiranje foto-realističnih materijala.

Završeni model zatim je pohranjen u format .FBX, koji sadrži model i njegove materijale, a može sadržavati i kostur i animacije modela. Taj format se može jednostavno učitati u Unity, zbog čega je odabran prilikom izrade ovog rada.

U Unity pogonu izrađena je aplikacija *MDKretanje* koja sadrži model Male dvorane. Prilikom pokretanja aplikacije, na pozicije praznih objekata koji predstavljaju sjedala postavljaju se odgovarajući modeli sjedala, što je pokazano u isječku koda 4.1. Korisniku je omogućeno slobodno kretanje prostorom dvorane upotrebom tipkovnice i miša, a uz pomoć zasebne aplikacije opisane u nastavku, moguće je i postavljanje kamere na pozicije sjedala upotrebom glasovnih naredbi.

Kod 4.1: Funkcija za generiranje sjedala

```
void generiraj()
{
    GameObject newSjedalo;
    foreach (Transform prazan in transform)
    {
        newSjedalo = (GameObject) Instantiate(sjedalo,
        prazan.transform.position, prazan.transform.rotation);
        newSjedalo.transform.parent = prazan;
    }
}
```

4.3. Glasovne naredbe

Za detekciju glasovnih naredbi izrađena je aplikacija *MDSpeechRecognition* koja sluša govor korisnika te u njemu pokušava prepoznati moguće naredbe. Ukoliko aplikacija prepozna naredbu, prosljeđuje ju aplikaciji *MDKretanje*, koja omogućava kretanje virtualnom dvoranom. U nastavku je opisana aplikacija *MDSpeechRecognition* te način komunikacije s aplikacijom *MDKretanje*.

Aplikacija MDSpeechRecognition napisana je u jeziku C# koristeći programski paket za prepoznavanje govora Speech.Recognition radnog okvira .NET. Komponenta tog paketa koja omogućava prepoznavanje govora je razred SpeechRecognitionEngine koji sluša govor primljen pomoću mikrofona i dojavljuje rečenice koje je prepoznao u obliku niza znakova. Kako bi



Slika 4.3: Sučelje aplikacije MDSpeechRecognition

SpeechRecognitionEngine prepoznao riječi govora, potrebno mu je dodati jednu ili više gramatika koje sadrže popise mogućih riječi i rečenica. Gramatika se definira u obliku XML datoteke koja se učitava pomoću razreda *Grammar*. U isječku koda 4.2 prikazano je kako se pokreće slušanje govora korištenjem razreda *SpeechRecognitionEngine* i *Grammar*.

```
// zapocni slusanje vise naredbi
recognizer.RecognizeAsync(RecognizeMode.Multiple);
```

Gramatiku čine riječi i pravila. Riječi su definirane izgovorom i zapisom, a pravila se sastoje od jedne ili više riječi koje je moguće prepoznati kao cjelinu. Pravila su hijerarhijski strukturirana, zbog čega je moguće umetati jedno pravilo unutar drugog. Korijensko pravilo definirano je atributom *root*, a da bi se riječ ili rečenica definirana nekim pravilom mogla prepoznati, potrebno je umetnuti to pravilo u korijensko pravilo. Pravilo može sadržavati izbor riječi i eksplicitno zadane riječi točnim redoslijedom, pri čemu neke riječi mogu biti i opcionalne. Primjer definicije gramatike prikazan je u isječku koda 4.3. Definirano je korijensko pravilo brojevi. Pravilo se sastoji od opcionalne riječi 'broj' i izbora (oznaka 'one-of') jednog od dva broja. Svaka riječ unutar pravila ima zadan zapis (atribut 'sapi:display') i standardni izgovor (riječ unutar oznaka 'token'). Kako bi dodali riječi engleskog jezika u gramatiku, potrebno je odrediti samo standardni izgovor riječi, no za dodavanje riječi hrvatskog ili nekog drugog nepodržanog jezika, potrebno je odrediti točan izgovor korištenjem fonema. Takav izgovor definiran je atributom 'sapi:pron', čija vrijednost je niz znakova koji predstavljaju foneme i naglaske. Primjerice, niz "J S1 EH D S2 AX N" odgovara izgovoru riječi "jedan".

Kod 4.3: Primjer pravila gramatike

```
<rule id="brojevi" scope="public">
   <item repeat="0-1">
     <token sapi:display="broj"> number </token>
   </item>
   <one-of>
       <item>
         <token sapi:display="1" sapi:pron="J S1 EH D S2 AX N">
           jedan
         </token>
       </item>
       <item>
         <token sapi:display="2" sapi:pron="D V AA">
           dva
         </token>
       </item>
   </one-of>
</rule>
```

Primjerice, gramatikom prikazanom u isječku prepoznati će se rečenice "jedan" ili "broj dva", a rezultati prepoznavanja koji će se ispisati biti će "1" i "broj 2". Neće se prepoznati druge riječi koje nisu definirane gramatikom, kao ni rečenice koje nisu određene pravilima gramatike. Izrada vlastite gramatike znatno povećava preciznost glasovnog upravljanja jer će svaka prepoznata riječ ili rečenica biti stvarna naredba za koju sustav izvodi odgovarajuće radnje.

Prepoznate glasovne naredbe potrebno je proslijediti aplikaciji *MDKretanje* kako bi se korisniku omogućilo glasovno upravljanje kretanjem u virtualnoj dvorani. Naredbe se šalju u obliku niza znakova upotrebom TCP *socketa* na vlastitu IP adresu (127.0.0.1) po proizvoljno odabranom portu 22222. Aplikacija *MDKretanje* sadrži skriptu koja sluša promet na tom portu, prima naredbe te ih prosljeđuje skripti za pozicioniranje kamere.

Kod 4.4: Slanje naredbi među aplikacijama

```
// Slanje (MDSpeechRecognition)
TcpClient client = new TcpClient("127.0.0.1", 22222);
NetworkStream s = client.GetStream();
StreamWriter sw = new StreamWriter(s);
sw.AutoFlush = true;
sw.WriteLine(message);
// Primanje (MDKretanje)
TcpListener tcpListener = new TcpListener(22222);
tcpListener.Start();
// cekaj poruku...
TcpClient client = tcpListener.AcceptTcpClient();
NetworkStream s = client.GetStream();
```

```
StreamReader reader = new StreamReader(s);
```

msg = reader.ReadLine();

```
ProcessMessage(msg);
```

4.4. Slobodno kretanje dvoranom

U sklopu izrade ovog rada implementirane su dvije metode kretanja virtualnom dvoranom. Prva i jednostavnija metoda je kretanje upotrebom tipkovnice i miša. Ovakav način upravljanja korisniku omogućuje slobodno kretanje virtualnim prostorom dvorane. Slobodno kretanje implementirano je tako da korisnik pomacima miša određuje smjer pogleda virtualne kamere, a tipkama na tipkovnici određuje smjer i brzinu kretanja.

U pogonu Unity, pritisak tipke na tipkovnici ili pomak miša očitavaju se upotrebom razreda Input koji dojavljuje aplikaciji koja tipka je pritisnuta ili u kojem smjeru je pomaknut miš. Također, prilikom izrade aplikacije, na raspolaganju je funkcija Update() koja se poziva prilikom svakog osvježavanja ekrana. U toj funkciji moguće je odrediti koje će se radnje događati prilikom svakog osvježavanja, kao što je očitavanje pritisnutih tipki tipkovnice. U isječku koda 4.5 prikazano je kako se očitavaju neke od pritisnutih tipki, pri čemu se mijenjaju položaj kamere i brzina kretanja u virtualnom prostoru. Upotrebom tipki W, A, S i D, korisnik se kreće naprijed, lijevo, natrag i desno, tipkama E i Q kreće se gore i dolje, a brzinu određuje tipkama + i - na numeričkom dijelu tipkovnice.

Kod 4.5: Primjer očitavanja pritisaka tipki na tipkovnici

```
void Update () {
    //...
    if(Input.GetKey(KeyCode.W)) {
        transform.Translate(0, 0, movementSpeed);
    }
    if(Input.GetKey(KeyCode.A)) {
        transform.Translate(-movementSpeed, 0, 0);
    }
    //...
    if(Input.GetKey(KeyCode.KeypadPlus)) {
        movementSpeed += 0.01f;
        if(movementSpeed > 0.1f)
            movementSpeed = 0.1f;
    }
    //...
```

4.5. Kretanje glasovnim naredbama

Druga metoda kretanja virtualnom dvoranom je upotrebom glasovnih naredbi. Naredbe definirane gramatikom očitavaju se pomoću aplikacije *MDSpeechRecognition* i šalju aplikaciji *MDKretanje*. Naredbe se zatim analiziraju, a na temelju njihova sadržaja korisnik se pomiče na željenu poziciju u virtualnoj dvorani. Glasovnim naredbama korisnik zadaje blok sjedala, broj reda i broj sjedala, pri čemu se pozicionira iznad odabranog sjedala i rotira tako da smjer pogleda bude prema pozornici. Gramatikom je definirana dodatna naredba 'ugasi' koja omogućuje istovremeno gašenje obje aplikacije.

Analiza glasovnih naredbi implementirana je upotrebom stroja stanja, koji je prikazan na slici 4.4 . Glasovne naredbe koje korisnik koristi za kretanje su oblika "pokaži <blok> red <broj> sjedalo <broj>", pri čemu su 'pokaži', 'red' i 'sjedalo' ključne riječi koje mijenjaju stanje, a <blok> i <broj> su oznake skupina riječi kojima se definira željena pozicija. Riječi iz skupine <blok> označavaju blokove sjedala koji se nalaze u dvorani (lijevo, sredina i desno), a brojevi su iz intervala [1, 14].



Slika 4.4: Stroj stanja koji omogućuje glasovno upravljanje

U početnom stanju aplikacije 'INIT' čeka se prva glasovna naredba. Aplikacija naredbu rastavlja na pojedinačne riječi i redom analizira svaku riječ. Ključna riječ 'pokaži' označava početak naredbe i postavlja aplikaciju u stanje 'UPDATE' u kojem čeka naziv jednog od blokova ili jednu od ključnih riječi 'red' ili 'sjedalo'. Ukoliko slijedeća riječ označava blok sjedala, aplikacija prelazi u stanje 'BLOK', postavlja broj reda i broj sjedala na nulu i čeka ključnu riječ 'red' s brojem reda. Nakon toga aplikacija prelazi u stanje 'RED', pohranjuje broj reda i čeka ključnu riječ 'sjedalo' s brojem sjedala. Aplikacija prelazi u stanje 'SJEDALO', pohranjuje broj sjedala i prelazi u stanje 'READY' koje označava da su svi podaci o traženom sjedalu dostupni. Ukoliko se aplikacija nalazi u stanju 'READY', prilikom slijedećeg poziva funkcije *Update()* kamera se pozicionira na mjesto odabranog sjedala i aplikacija se vraća u početno stanje. Pozicija i rotacija kamere interpoliraju se između trenutnog položaja i položaja određenog glasovnom naredbom tako da prijelaz bude što prirodniji.



Slika 4.5: Aplikacija MDKretanje nakon primljene glasovne naredbe

Na slici 4.5 prikazana je aplikacija *MDKretanje* nakon primljene naredbe "pokaži lijevo red 5 sjedalo 4". Posljednja primljena naredba prikazana je u gornjem lijevom kutu prozora aplikacije. Kamera se pozicionirala na odabrano sjedalo i rotirala se u smjeru točke na sredini pozornice.

5. Rezultati

Zadatak ovog rada bio je implementirati različite načine kretanja virtualnom dvoranom. Prva metoda kretanja je jednostavno slobodno kretanje upotrebom tipkovnice i miša. Prednosti ove metode su jednostavnost upotrebe zbog malog broja upravljačkih radnji te sloboda kretanja u svim smjerovima. Nedostaci ovakvog načina upravljanja su nepreciznost prilikom kretanja i sporo pozicioniranje na željeno mjesto u prostoru. Pošto je glavna zadaća napravljene aplikacije pozicioniranje iznad zadanog sjedala u dvorani, upravljanje mišem i tipkovnicom zbog svojih nedostataka nije optimalan izbor načina kretanja. Potrebna je alternativa koja će prije svega omogućiti brzo premještanje između sjedala, tako da upravljanje bude jednostavno i precizno. Upravo zbog toga, izabrano je upravljanje glasovnim naredbama što se pokazalo kao znatno bolja metoda kretanja virtualnom dvoranom.

U nastavku će biti razmotrena preciznost glasovnog upravljanja s obzirom na korištenje različitih mikrofona i s obzirom na zadavanje glasovnih naredbi na hrvatskom i engleskom jeziku.

5.1. Preciznost glasovnog upravljanja

Kako bi se ispitala preciznost glasovnog upravljanja, potrebno ju je prvo definirati. Za potrebe ovog rada, neka preciznost bude određena postotkom ispravno prepoznatih naredbi prilikom glasovnog upravljanja. Aplikacija će se ispitivati skupom naredbi, pri čemu će se mijenjati određeni parametri aplikacije kao što su korišteni mikrofoni i jezik na kojem se zadaju naredbe, a za svaki parametar će se bilježiti broj ispravno prepoznatih naredbi.

5.2. Usporedba mikrofona

Prilikom ispitivanja preciznosti glasovnog upravljanja, korištena su tri različita mikrofona kako bi se provjerio njihov utjecaj na uspješnost prepoznavanja glasovnih naredbi. Korišteni su ugrađeni mikrofon prijenosnog računala, stolni mikrofon i laringofon¹. Ispitivanje je provedeno u serijama od 50 glasovnih naredbi, pri čemu se provjeravala uspješnost prepoznavanja svake izgovorene naredbe. Preciznost je određena kao prosjek rezultata nakon tri serije ispitivanja.

Pokazalo se da je u tihom okruženju upravljanje upotrebom triju mikrofona približno jednako precizno. Rezultati ispitivanja pokazani su tablicom 5.1. Za svaki mikrofon naveden je broj ispravno prepoznatih naredbi po serijama i izračunata preciznost.

Mikrofon	Serija 1	Serija 2	Serija 3	Preciznost (%)
Ugrađeni	43	41	42	84.00
Stolni	39	42	41	81.33
Laringofon	40	44	43	84.66

Tablica 5.1: Rezultati ispitivanja mikrofona u tihim uvjetima

Znatno veća razlika u preciznosti upravljanja među mikrofonima stvorila se u bučnom okruženju. Preciznost stolnog i ugrađenog mikrofona pala je na manje od 60%, dok je preciznost laringofona ostala približno jednaka. Rezultati ispitivanja u bučnom okruženju prikazani su tablicom 5.2.

Tablica 5.2: Rezultati ispitivanja mikrofona u bučnim uvjetima

Mikrofon	Serija 1	Serija 2	Serija 3	Preciznost (%)
Ugrađeni	27	28	24	52.66
Stolni	29	30	28	58.00
Laringofon	43	41	41	83.33

 $^1\mathrm{Mikrofon}$ koji se prislanja na grlo govornika, a glas snima očitavanjem vibracija tijekom govora

5.3. Usporedba hrvatskog i engleskog jezika

Aplikacija *MDSpeechRecognition* omogućuje prepoznavanje naredbi na hrvatskom i engleskom jeziku. Ispitana je preciznost glasovnog upravljanja s obzirom na jezik naredbi, a kao ulazna jedinica korišteni su stolni mikrofon i laringofon. Način ispitivanja bio je sličan kao kod ispitivanja preciznosti mikrofona, a rezultati za hrvatski jezik preuzeti su iz tablice 5.1.

Upotrebom stolnog mikrofona, rezultati prepoznavanja naredbi engleskog jezika bili su nešto slabiji. Ispravno je prepoznato tek nešto manje od 70% izgovorenih naredbi, za razliku od naredbi hrvatskog jezika kojih je prepoznato više od 80%. Upotrebom laringofona, prepoznavanje engleskog jezika se popravilo i rezultati su bili sličniji prepoznavanju naredbi hrvatskog jezika. Preciznost upravljanja naredbama engleskog jezika iznosila je oko 77%. Rezultati ispitivanja prikazani su tablicom 5.3, pri čemu su rezultati upotrebom različitih mikrofona naznačeni oznakama (S) za stolni mikrofon i (L) za laringofon.

Mikrofon	Serija 1	Serija 2	Serija 3	Preciznost (%)
Engleski (S)	35	33	34	68.00
Hrvatski (S)	39	42	41	81.33
Engleski (L)	37	40	39	77.33
Hrvatski (L)	40	44	43	84.66

Tablica 5.3: Rezultati upravljanja naredbama na engleskom jeziku

6. Zaključak

Izrada virtualnih prostora na temelju teško dostupnih ili opasnih stvarnih okruženja omogućuje korisnicima jednostavno kretanje i interakciju s objektima u tim okruženjima. Nove tehnologije već danas omogućuju djelomičan doživljaj virtualne stvarnosti, a postoji veliki potencijal za napredak u tom području.

Korisnicima su na raspolaganju brojne metode kretanja virtualnim prostorom, a metode opisane u ovom radu samo su neke od dostupnih. Kretanje fizičkim ulaznim jedinicama može biti sporo jer zahtjeva unos svakog pojedinačnog pokreta, zbog čega su nekad potrebne alternativne metode kao što je glasovno upravljanje. Primjerice, jednom glasovnom naredbom u implementiranoj aplikaciji korisnik se gotovo trenutno translatira po sve tri osi do pozicije odabranog sjedala te se rotira tako da gleda prema pozornici. Za takvo pozicioniranje upotrebom miša i tipkovnice, korisnik bi trebao puno više vremena i unosa, pri čemu ne bi mogao postići jednaku preciznost.

Model dvorane izrađen za potrebe ovog rada odličan je primjer broja mogućnosti koje pružaju virtualni prostori. Primjerice, ovako implementirana aplikacija može se iskoristiti za prikaz prilikom prodaje ulaznica za koncert u toj dvorani. Upotrebom konfiguracijske datoteke moguće je pridijeliti stanje svakom sjedalu, a kupac bi pri odabiru sjedala mogao vidjeti kako pozornica izgleda s odabranog mjesta u dvorani. S druge strane, dodavanjem virtualnih objekata kao što su zvučnici i reflektori, znatno bi se olakšala organizacija koncerata, jer bi organizatori mogli planirati gledajući virtualnu scenu. Ovi primjeri pokazuju kolike su mogućnosti upotrebe virtualnih prostora, a razvojem tehnologija će ih biti sve više.

7. Literatura

- Mine, M. R., Virtual Environment Interaction Techniques, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.294. 7700&rep=rep1&type=pdf, 17.5.2014.
- Pandžić, I. S., Virtualna okruženja: interaktivna 3D grafika i njene primjene,
 izdanje, Zagreb, Element, 2011.
- 3. Microsoft, System.Speech.Recognition Namespace, http://msdn.microsoft.com/en-us/library/system.speech. recognition.aspx, 22.4.2014.
- Koncertna Dvorana Vatroslava Lisinskog, *Tlocrti dvorane*, http://www.lisinski.hr/category.php?id=2, 5.5.2014.
- Unity Technologies, Unity Documentation, http://unity3d.com/learn/documentation, 8.5.2014.

Virtualna Mala dvorana Koncertne Dvorane Vatroslava Lisinskog

Sažetak

Rad sadrži pregled različitih načina kretanja virtualnim prostorima i interakcije s virtualnim objektima. Opisane su najčešće korištene metode upravljanja kretanjem u virtualnom prostoru: vanjske ulazne jedinice, uređaji za praćenje pokreta koji omogućuju izravno upravljanje pokretima i virtualne ulazne jedinice iscrtane na ekranu.

U praktičnom dijelu rada, upotrebom alata *Blender* izrađen je 3D model Male dvorane Koncertne Dvorane Vatroslava Lisinskog. U pogonskom alatu *Unity* izrađena je aplikacija koja korisniku omogućuje kretanje virtualnom dvoranom koristeći dvije različite metode kretanja. Prva metoda je kretanje virtualnim prostorom upotrebom tipkovnice i miša, pri čemu korisnik upravlja smjerom gledanja te smjerom i brzinom kretanja. Druga metoda omogućuje korisniku odabir sjedala u virtualnoj dvorani glasovnim naredbama, pri čemu se koristi vanjska aplikacija za prepoznavanje govora na hrvatskom i engleskom jeziku.

Ključne riječi: virtualni prostor, 3D, glasovno upravljanje, laringofon, Unity, Blender, C#

Virtual Small hall of the Vatroslav Lisinski Concert Hall

Abstract

The thesis provides an overview of techniques used for movement in virtual environments and interaction with virtual objects. Main user input methods are described: physical input units, motion tracking devices and virtual input devices.

In the practical section of the thesis, a 3D model of the Small hall of the Vatroslav Lisinski Concert Hall was built using *Blender*. An application was made in the *Unity* engine, enabling users to move through the virtual hall using two distinct movement techniques. First technique utilizes the keyboard and mouse to implement free movement, enabling the user to fly through the virtual environment. The second technique enables the user to move to a seat in the hall using voice commands in Croatian and English via an external application.

Keywords: virtual environment, 3D, voice control, throat microphone, Unity, Blender, C#