

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4530

**INTERAKCIJA GESTAMA U VIRTUALNOJ
SCENI POMOĆU UREĐAJA LEAP MOTION**

Stella Coumbassa

Zagreb, lipanj 2016.

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

Zagreb, 16. ožujka 2016.

ZAVRŠNI ZADATAK br. 4530

Pristupnik: **Stella Coumbassa (0036476863)**
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: **Interakcija gestama u virtualnoj sceni pomoću uređaja Leap Motion**

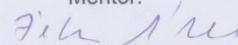
Opis zadatka:

Proučiti funkcionalnosti, mogućnosti i ograničenja uređaja Leap Motion. Napraviti pregled mogućnosti ostvarivanja gesti uz uporabu jedne ili dvije ruke te proučiti mogućnosti očitavanja gesti uz uporabu navedenog uređaja. Razraditi primjere prepoznavanja gesti dobivenih uređajem Leap Motion. Načiniti ocjenu i usporedbu 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.

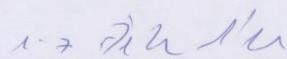
Zadatak uručen pristupniku: 18. ožujka 2016.
Rok za predaju rada: 17. lipnja 2016.

Mentor:



Prof. dr. sc. Željka Mihajlović

Predsjednik odbora za
završni rad modula:



Prof. dr. sc. Siniša Srblijić

Djelovođa:


Doc. dr. sc. Tomislav Hrkać

Zahvaljujem se svojoj mentorici, prof. dr. sc. Željki Mihajlović, na svoj pruženoj pomoći i savjetima tijekom studija te pri izradi ovoga rada.

Zahvaljujem se također svojoj obitelji na neizmjernoj podršci, strpljenju i dugogodišnjem poticaju pri 'potrazi za izgubljenim vremenom'.

Naposljetku, zahvalila bih se naravno i svojim bliskim priateljima na sudjelovanju u svim čudnovatim zgodama kroz koje smo zajedno prošli, kao i razumijevanju za moje katkad neuobičajene dosjetke.

Sadržaj

1. Uvod.....	1
2. Leap Motion.....	2
2.1 Građa i osnovni princip rada.....	2
2.2 Funkcionalnosti.....	5
2.2.1. Koordinatni sustav	5
2.2.2. Praćenje objekata u vidnom polju	6
2.3 Nedostaci i problemi pri korištenju.....	8
3. Geste.....	10
3.1 O gestama	10
3.2 Znakovni jezik.....	10
3.3 Geste pri radu s Leap Motion uređajem.....	12
4. Programska implementacija.....	14
4.1 Integracija Unity 3D pogona i Leap Motion kontrolera	14
4.2 Izrada aplikacije	15
4.2.1. Glavni izbornik	15
4.2.2. Pohranjivanje gesti.....	17
4.2.3. Interaktivno učenje znakovne abecede	18
4.2.4. Prikaz rezultata	21
4.3 Problemi i moguća poboljšanja	22
5. Zaključak	23
6. Literatura	24

1. Uvod

U današnjem svijetu sve više se može zamjetiti brz i značajan razvoj različitih tehnologija interakcije između čovjeka i računala, te kontrole pokreta. Doživljaj virtualnog svijeta kakav je prošlih desetljeća bio dio samo mašte i filmova, napokon postaje stvarnost (*Slika 1-1*). Sve veći je broj uređaja koji cijelo iskustvo digitalnog svijeta dovode na potpuno novu razinu, omogućujući korisnicima da se osjećaju kao da su zaista dio druge stvarnosti. Jedan od takvih uređaja je i Leap Motion. Ovaj mali uređaj omogućuje korisnicima interakciju s digitalnim svjetom pomoću pokreta ruku, te cijelo iskustvo čine usporedivim s onim u stvarnom svijetu.



Slika 1-1 Primjer kontrole pokretom iz filma Minority Report

U ovom radu razradit će se interakcija gestama u virtualnoj sceni pomoću ovog uređaja. Pritom će se u drugom poglavljju razraditi osnove Leap Motion kontrolera, te njegove funkcionalnosti i nedostaci. Kako se pri radu s kontrolerom većinom koriste geste, treće poglavlje obradit će pojam gesti, i njihovu upotrebu pri radu s Leap Motion uređajem. Naposljetku, četvrto poglavlje će demonstrirati implementaciju interakcije gestama u virtualnoj sceni pomoću kontrolera, na temelju aplikacije izrađene u sklopu ovog rada.

2. Leap Motion

Leap Motion™ je periferni uređaj, razvijen od strane američke kompanije *Leap Motion, Inc.*, čiji je cilj korisnicima omogućiti interakciju računalom na što intuitivniji način – pokretima ruku [1]. Ovaj, zaista malen, ali u svakom pogledu fascinantan uređaj, radi na principu praćenja pokreta ruku, te korisniku nudi mogućnost da iskusi digitalni svijet na potpuno novoj razini, čineći cjelokupno iskustvo prirodnijim od onog pri korištenju konvencionalnih sučelja, kao što su tipkovnica i miš.

Dakako, Leap Motion nudi interakciju s digitalnim sadržajem ne samo pri standardnom korištenju računala, već i u virtualnoj te proširenoj stvarnosti, te omogućuje korisniku da pritom koristi ruke na isti način na koji bi ih inače koristio i u stvarnom svijetu [2]. Upravo iz tog razloga, pojedinci koji se odluče na korištenje ovog uređaja napokon imaju mogućnost doživjeti i postati dio uzbudljivog svijeta i stvarnosti, kakva se donedavno mogla vidjeti uglavnom u filmovima.



Slika 2-1 Leap Motion kontroler

2.1 Građa i osnovni princip rada

Leap Motion je kontroler poprilično jednostavne građe i malih dimenzija (točnije 80mm x 30mm x 11.25mm) s obzirom na neizmjerne mogućnosti koje nudi. Kontroler pritom sadrži dva optička senzora (kamere), te tri infracrvene LED-ice. Navedene LED-ice mogu se uočiti na uređaju kada god je on upaljen, odnosno spojen na

računalo [3][4]. Leap Motion također ima mogućnost praćenja ruku do 200 slika u sekundi te relativno veliki interaktivni 3D prostor od 0.2265 m^3 , u kojem se očekuje mogućnost praćenja objekata [4].



Slika 2-2 Dijelovi Leap Motion kontrolera

Pri radu, kontroler učitava podatke dobivene od senzora u svoju lokalnu memoriju, te ih nakon potencijalnih potrebnih prilagodbi šalje putem USB sučelja do *Leap Motion Service*-a, odgovarajućeg softvera zaduženog za samo praćenje pokreta i pravilnu interpretaciju dobivenih podataka. Točnije, *Leap Motion Service* vrši obradu dobivenih slika, da bi se naposljetku dobila 3D rekonstrukcija onoga što uređaj zaista “vidi” [4].

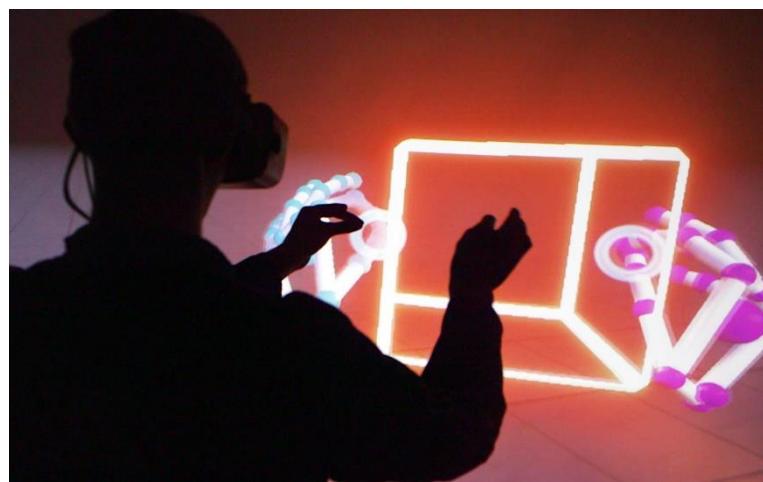
Iz podataka dobivenih navedenom rekonstrukcijom interpretira se, odnosno klasificira, informacija o objektima pronađenima u vidnom polju uređaja, te se pretpostavlja i sama pozicija zaklonjenih objekata. Očekivani objekti u slučaju *Leap Motion kontrolera* su najčešće ruke, prsti ruku te alati koje korisnik drži u rukama, primjerice olovka. Dobivene rezultate *Leap Motion Service* šalje transportnim protokolom do

Leap Motion upravljačke ploče te izvornih i web klijentskih biblioteka. Biblioteke dobivenu informaciju potom organiziraju u samo aplikacijsko programsko sučelje (API) [4].

Nadalje, Leap Motion uređaj može se koristiti u radu s desktop računalom (*Slika 2-3*) te u virtualnoj stvarnosti, i to u kombinaciji s uređajem Oculus Rift (*Slika 2-4*). U ovom radu naglasak je stavljen na korištenje Leap Motion kontrolera u radu s desktop računalom, pa se funkcionalnosti uređaja promatraju upravo s tog stajališta, iako je većina njih gotovo jednaka i u slučaju integracije s uređajem Oculus Rift.



Slika 2-3 Leap Motion u kombinaciji s desktop računalom



Slika 2-4 Leap Motion u kombinaciji s uređajem Oculus Rift

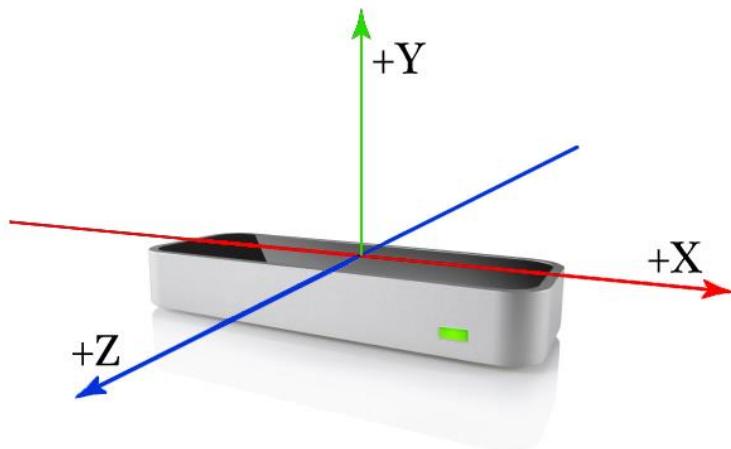
2.2 Funkcionalnosti

Same funkcionalnosti Leap Motion kontrolera najbolje se mogu opisati kroz mogućnosti koje nudi njegovo aplikacijsko programsko sučelje, odnosno API.

Kao što je već ranije rečeno, Leap Motion kontroler ima mogućnost prepoznavanja i praćenja ruku, prstiju te alata, te omogućava raspoznavanje i razlikovanje različitih pokreta, gesti i pozicija ruku. Kako bi se mogle raspoznati pozicije objekata u vidnom polju uređaja te interpretirati podaci, svakako je potrebno definirati koordinatni sustav uređaja [5].

2.2.1. Koordinatni sustav

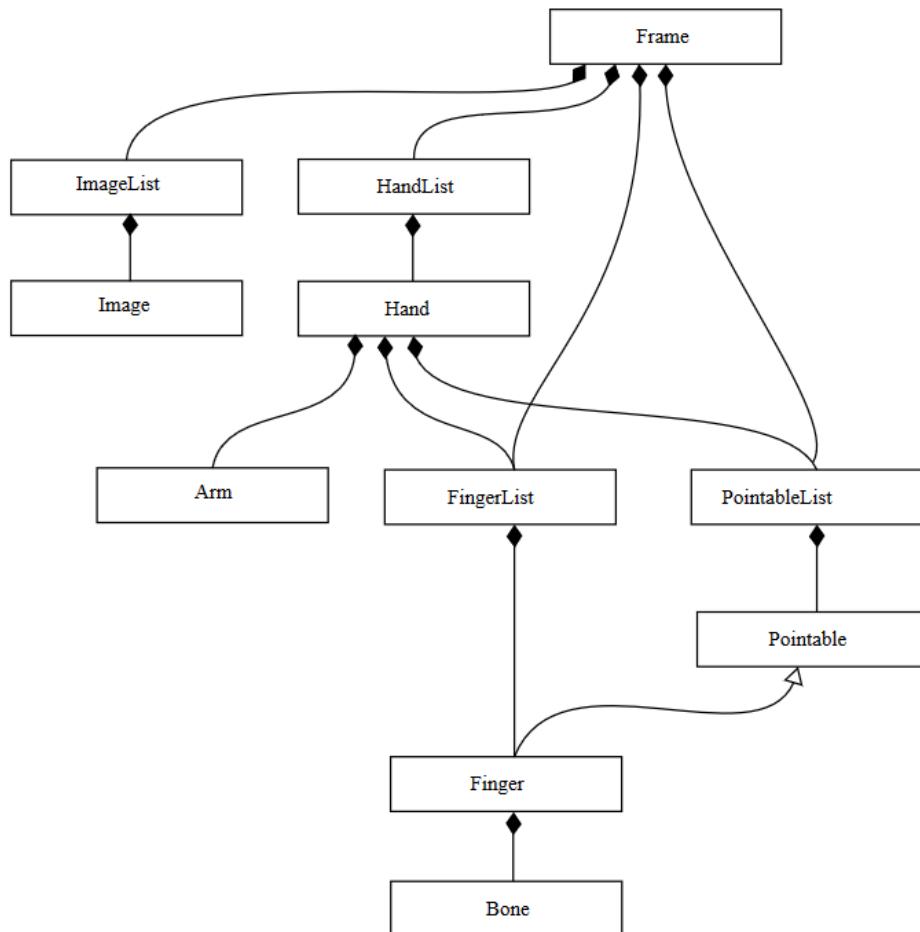
Sustav Leap Motion kontrolera koristi desni koordinatni sustav, čije je ishodište smješteno u središtu gornjeg dijela kontrolera. Pritom je y-os sustava postavljena okomito, dok se x-os i z-os koordinatnog sustava nalaze u horizontalnoj ravnini s obzirom na gornji dio uređaja (*Slika 2-5*). Aplikacijsko programsko sučelje same udaljenosti mjeri u milimetrima, dok se za kuteve očekivano koriste radijani [5].



Slika 2-5 Koordinatni sustav

2.2.2. Praćenje objekata u vidnom polju

Samo praćenje objekata u vidnom polju Leap Motion-a omogućeno je pomoću skupa razreda i funkcija koje nudi Leap Motion API. Za svaki značajniji praćeni objekt postoji i odgovarajući razred. Pri tome, svim praćenim objektima, kao što su prsti ili pak ruke, može se pristupiti preko takozvanih okvira (engl. *Frame*). Svaki od spomenutih okvira sadrži informaciju o izmjerenoj poziciji i ostalim parametrima svih detektiranih objekata. Stoga ni ne čudi što se upravo istoimeni razred, odnosno *Frame* nalazi na vrhu hijerarhije modela praćenja (*Slika 2-6*). Okvir (*Frame*) u suštini sadrži liste svih ruku i prstiju praćenih u odgovarajućem okviru [5].



Slika 2-6 Model praćenja

Od važnijih razreda koji čine model podataka Leap Motion-a svakako je dobro spomenuti:

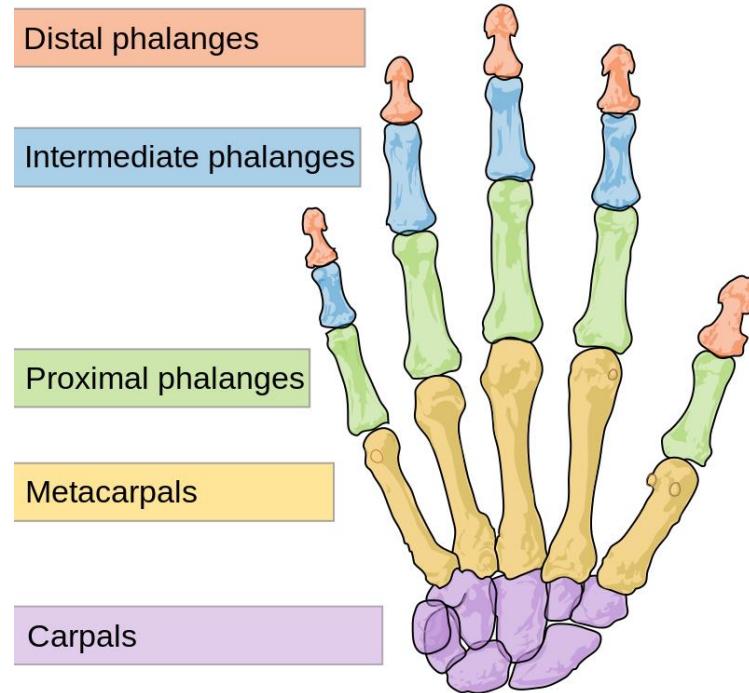
- *Hand*
- *Pointable*
- *Finger*
- *Bone*

Hand razred, kao što mu i samo ime asocira predstavlja ruku. Točnije, on opisuje poziciju i orijentaciju ruke, prati pokrete ruke kroz okvire te sadrži listu prstiju koji pripadaju upravo toj ruci. Dobro je spomenuti i da Leap Motion sadrži interni model ljudske ruke. To je izrazito važno jer omogućuje praćenje ruke, čak i kad ona nije u potpunosti vidljiva. Model ruke također uvijek može definirati pozicije svih pet prstiju ruke. Ukoliko u vidnom polju pojedini prst nije jasno vidljiv, na temelju pozicije ruke i internog modela može se izračunati najvjerojatnija pozicija navedenog prsta, čime se izbjegavaju potencijalne komplikacije pri korištenju kontrolera [5].

Pointable pruža podatke i karakteristike vezane uz detektirani prst ili alat [5].

Finger, koji nasljeđuje *Pointable*, sadrži za podatke za svaki prst ruke, poziciju njegovog vrha te smjer. Dakako, razlikuju se i tipovi prstiju, tako da ne može doći do pogrešnog očitavanja dva istovrsna prsta na jednoj ruci [5].

Bone predstavlja dio prsta, odnosno ovim objektom opisana je pozicija i orijentacija svake kosti prsta. Sveukupno se može razlikovati četiri kosti u svakom prstu (*Slika 2-7*) [5]. Pozicije određenih kostiju prsta mogu biti jako korisne za slučaj prepoznavanja i razlikovanja sličnih gesta ruku, o čemu će biti riječ kasnije u radu.



Slika 2-7 Kosti ruke

Naposljetu, dobro je spomenuti i razred ponuđen od strane Leap Motion aplikacijskog programskog sučelja, koji omogućuje povezivanje sa Leap Motion uređajem – *Controller*. Preko njega se mogu dohvati okviri koji sadrže sve bitne praćene podatke [5].

2.3 Nedostaci i problemi pri korištenju

Dakako, kod uređaja baziranih na praćenju pokreta, kao što je Leap Motion, uvijek se javljuju i određeni nedostaci i problemi uslijed korištenja.

Kako je kod Leap Motion kontrolera praćenje objekata u vidnom polju ostvareno pomoću optičkih senzora i infracrvenih LEDica, u pravilnom raspoznavanju i praćenju objekata komplikacije mogu nastati ukoliko osvjetljenje okoline nije optimalno. To je primjerice slučaj pri prevelikoj prisutnosti infracrvene svjetlosti u vidnom polju uređaja,

na primjer od Sunca. Infracrvena svjetlost dakako ima negativan utjecaj na performanse uređaja [6]. Na svu sreću Leap Motion kontroler može detektirati preveliku količinu infracrvene svjetlosti, te korisnika upozoriti na istu.

Nadalje, ukoliko su neki dijelovi ruke zaklonjeni, prepoznavanje pozicije istih često nije ispravno, što je zapravo očekivano, s obzirom na činjenicu da uređaj može samo prepostaviti pozicije dijelova ruku koje ne vidi.

Leap Motion kontroler također nailazi na problem pri razlikovanju sličnih gesti i položaja ruku. Kako uvjeti rada kontrolera ne mogu uvijek biti idealni, lako se može dogoditi da kontroler pogrešno raspoznaće točan položaj ruke i prstiju na ruci, te i samu orientaciju ruke. Upravo iz tog razloga se može dogoditi da dođe do pogreške prilikom raspoznavanja.

Također, u slučaju da u nekom trenutku u interaktivnom 3D prostoru postoji više ruku, te su one spojene, postoji mogućnost da Leap Motion neće razaznati da je u polju više od jedne ruke, već će se ponašati kao da je riječ o samo jednoj. Slična situacija je i ona u kojoj je više prstiju ruke spojeno. Analogno, tada bi se dva spojena prsta lako mogla interpretirati kao samo jedan.

3. Geste

S obzirom na činjenicu da je naglasak ovog rada na upotrebi gesti pri radu s Leap Motion uređajem, svakako je potrebno i razraditi što zapravo predstavlja sam pojam geste, te kako se geste mogu koristiti u slučaju Leap Motion kontrolera.

3.1 O gestama

Geste predstavljaju oblik neverbalne, odnosno bezglasne komunikacije, pri kojoj se različitim pokretima i položajem tijela vrši prijenos konkretnih poruka i informacija. Geste se mogu koristiti umjesto, ili pak u kombinaciji s govorom. Također, one omogućuju prijenos emocija i misli tijekom uobičajene komunikacije [7]. Iako se sam pojam geste često može asociратi s gestama ruku, geste se mogu formirati i pokretima drugih dijelova tijela.

Različite geste iako se mogu povezati i s različitim kulturama i običajima. Određena gesta u različitim kulturama može imati potpuno drugačije značenje. Tako gesta koja u jednoj kulturi označava pohvalu, u drugoj se može smatrati teškom uvredom i izrazom nepristojnosti [7].

Kao jedan od najboljih primjera komunikacije u kojoj je naglasak na gestama ruku, dobro je izdvojiti znakovni jezik te znakovnu abecedu. Iz tog razloga je i znakovna abeceda odabrana za demonstraciju prepoznavanja gesti Leap Motion kontrolerom, o čemu će biti riječ nešto kasnije.

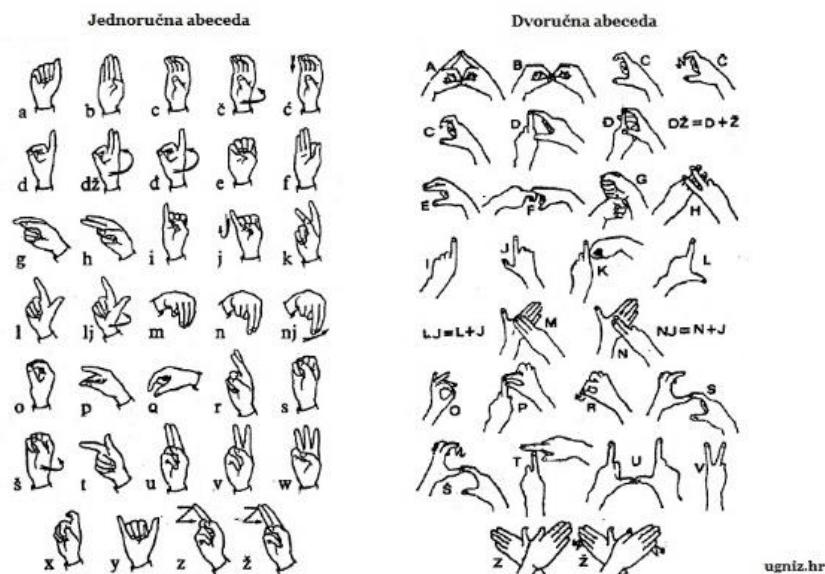
3.2 Znakovni jezik

Znakovni jezik predstavlja jezik u kojem se poglavito kao sredstvo komunikacije koriste ruke [8]. Pritom se pomoću specifičnih položaja i pokreta ruku i tijela nastoji kreirati smisao riječi. Iako se u komunikaciji znakovnim jezikom uvelike koriste ruke,

za uspješnu komunikaciju često se uključuju i držanje tijela i glave te izrazi lica [9]. Ovaj oblik komunikacije često se koristi u zajednicama osoba s oštećenjem sluha, no može se koristiti i u slučaju nemogućnosti reproduciranja govora od strane pojedinaca [8].

Općenito, znakovni jezik uglavnom nije univerzalan, odnosno znakovni jezik se može razlikovati od zemlje do zemlje, iako često postoje sličnosti između različitih znakovnih jezika. Nadalje, svaki znakovni jezik ima svoju gramatiku i potrebno ga je učiti kao bilo koji drugi strani jezik. Tako očekivano postoji i hrvatski znakovni jezik, pri čijem se korištenju kombinira očitavanje s usana i istovremeno korištenje pojedinih znakova koji slijede red riječi na hrvatskom jeziku [9].

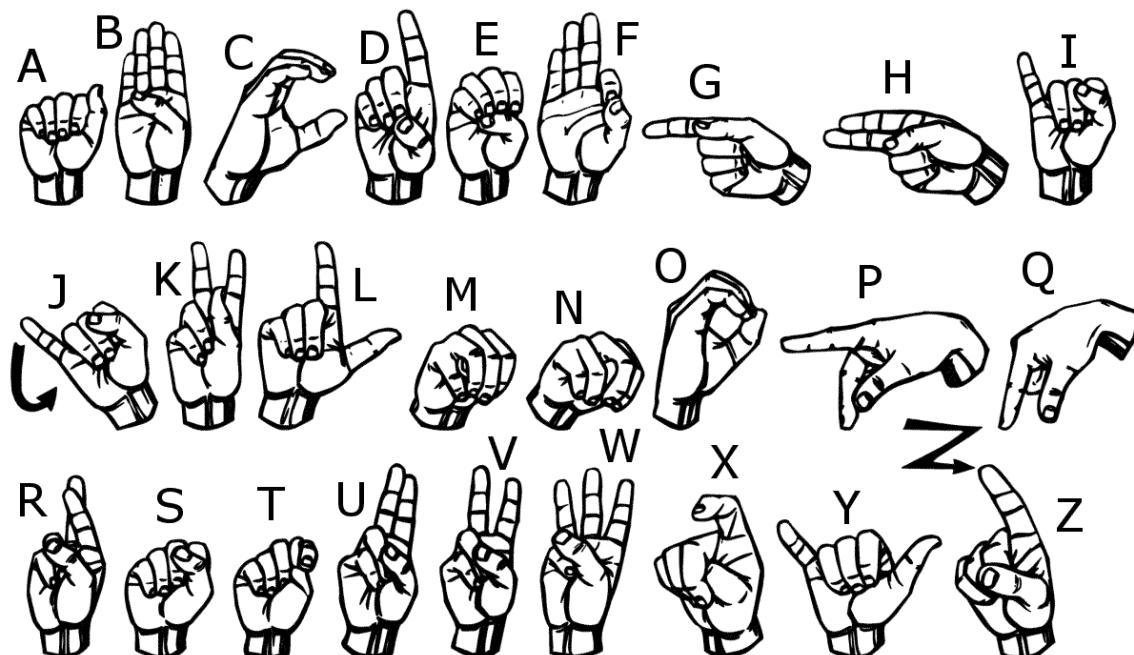
Kod znakovnog jezika je svakako dobro spomenuti i znakovnu abecedu. U hrvatskom znakovnom jeziku, a i većini ostalih stranih znakovnih jezika, može se razlikovati jednoručna i dvoručna abeceda znakovnog jezika (*Slika 3-1*).



Slika 3-1 Primjer jednoručne (lijevo) i dvoručne (desno) abecede hrvatskog znakovnog jezika

U obje vrste abecede pojedina slova definirana su konkretnim položajima prsti ruku. Pritom se kod dvoručne abecede promatraju položaji prstiju obje ruke, dok je kod

jednoručne abecede važan položaj prstiju samo jedne ruke. Hrvatska jednoručna abeceda zapravo se može smatrati verzijom američke (internacionalne) abecede [9].



Slika 3-2 Američka (internacionalna) abeceda znakovnog jezika

3.3 Geste pri radu s Leap Motion uređajem

Iako Leap Motion uređaj radi na principu praćenja pokreta ruku te ima sposobnost razlikovanja različitih pozicija i pokreta ruku, njegovo aplikacijsko programsko sučelje u slučaju rada na desktop računalu ima ugrađene svega četiri geste koje se mogu prepoznati. Za raspoznavanje navedenih gesti Leap Motion API nudi razred *Gesture* koji predstavlja gestu, odnosno pokret koji se prepoznaće. Pri samom raspoznavanju Leap Motion prati pokrete ruke ili ruku u vidnom polju te ako prepozna konkretni uzorak specifičan za određenu gestu, pohranjuje ju u listu gesti pridruženih trenutnom okviru. Dakako, prepoznata gesta ne mora biti duljine jednog okvira. Naime, ako je

riječ o dinamičkoj gesti, odnosno gesti koja predstavlja jedan cijeli pokret, ona se ne može prikazati pomoću samo jednog okvira, pošto jedan okvir predstavlja jedan položaj ruke. U tom slučaju više okvira koji sadrže konkretnu gestu ima pohranjenu gestu u listi svojih gesti. Kako bi se znalo koji točno okviri sadrže gestu, kod dinamičkih gesti gotovo se uvijek definira početak i kraj geste, koji se mogu detektirati primjerice na temelju brzine i položaja ruke u određenom trenutku [5]. Četiri ugrađene geste koje Leap Motion API ima mogućnost prepoznati su redom:

- *Circle* – kružni pokret prsta
- *Swipe* – ravan pokret otvorenog dlana s jedne prema drugoj strani
- *Screen Tap* – pokret pritiska ekrana, usmjeren prema naprijed
- *Key Tap* – pokret pritiska tipke, usmjeren prema dolje

Ukoliko korisnik želi uključiti druge geste u rad s kontrolerom ili pak kreirati svoje vlastite geste, potrebno je samostalno osmisliti i implementirati način njihovog raspoznavanja, ili koristiti gotova rješenja dostupna na Internetu, primjerice u sklopu vanjskih biblioteka. Primjer jednog takvog postupka, koji se koristio za raspoznavanje pokreta pri implementaciji aplikacije vezane uz ovaj rad, bit će obrađen u sljedećem poglavljiju.

4. Programska implementacija

Kako bi se demonstriralo i razradilo način i primjere raspoznavanja gesti pomoću uređaja Leap Motion, u sklopu ovog rada izrađena je i odgovarajuća aplikacija. S obzirom na činjenicu da pojam korištenja gesti često asocira na upotrebu znakovnog jezika, ideja iza navedene aplikacije bila je proizvesti interaktivan i jednostavan način na koji bi korisnik aplikacije imao priliku naučiti osnove znakovnog jezika, čime bi se ujedno i pridonijelo smanjenju jezičnih barijera.

Međutim, kako je već ranije spomenuto u radu, kompletan znakovni jezik često se ne sastoji samo od gesti ruku, već je popraćen i očitavanjem s usana, te kombinacijom gesti ruku s ostalim dijelovima tijela. To predstavlja i svojevrsnu prepreku za korištenje kompletног znakovnog jezika u slučaju Leap Motion uređaja, pošto je njegova temeljna uloga upravo praćenje pokreta ruku. Stoga je kao predmet raspoznavanja u aplikaciji odabrana abeceda znakovnog jezika.

4.1 Integracija Unity 3D pogona i Leap Motion kontrolera

Leap Motion uređaj se u ovom slučaju koristio u kombinaciji s radom na računalu, a upravo zbog jednostavnosti izrade igara koju pruža, za izradu aplikacije odabранo je Unity razvojno okruženje.

Pritom je za uspješnu izradu aplikacije, osim Unity 3D pogona, bilo potrebno i s Interneta dohvatiti Leap Motion SDK v2, odnosno verziju Leap Motion softvera namijenjenu razvoju i praćenju pokreta na desktop računalu. Također, Unity 3D sam po sebi nema automatski ugrađene alate kojima bi se moglo izravno pristupiti razredima i funkcijama koje nudi Leap Motion aplikacijsko programsko sučelje. Stoga je u projekt za aplikaciju bilo potrebno uključiti i dodatni paket sa sredstvima koja omogućavaju korištenje podataka iz Leap Motion uređaja pri izradi aplikacije (*Unity Assets for Leap Motion v2*), kao i gotove modele, scene i skripte koje značajno olakšavaju cijeli process izrade [5].

Na taj način, sam proces izrade nema značajnije razlike u odnosu na onaj uobičajeni. Programiranje ponašanja pojedinih objekata u sceni standardno se svodi na pisanje skripti, a pri izradi aplikacije navedene skripte pisane su u C# programskom jeziku. Nadalje, pri samoj izradi scene je potrebno pripaziti na postavljanje Leap Motion kontrolera na ispravnu poziciju u odnosu na kameru, kako bi se ruke praćene od strane kontrolera našle na ispravnoj poziciji pri korištenju aplikacije.

4.2 Izrada aplikacije

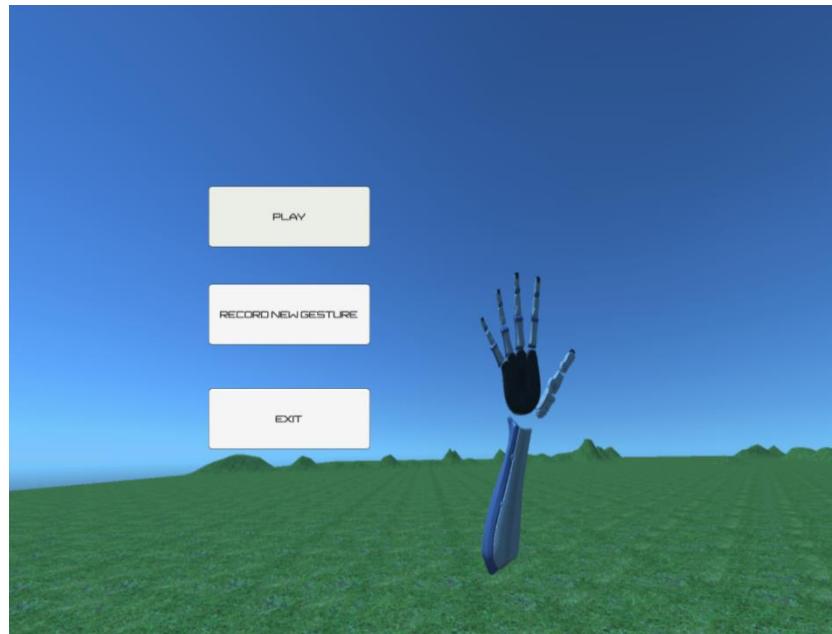
Kao što je već ranije spomenuto u ovom poglavlju, aplikacija izrađena u sklopu ovog rada radi na principu prepoznavanja slova znakovnog jezika. Međutim, jedan od glavnih izazova pri izradi ove aplikacije je dakako bila implementacija prepoznavanja i razlikovanja znakova, pošto Leap Motion sam po sebi može raspoznavati samo četiri geste, od koje niti jedna nije primjenjiva na raspoznavanje slova abecede znakovnog jezika. Stoga je ovom programskom proizvodu, osim prepoznavanja gesti, dana i funkcionalnost pohranjivanja proizvoljnih korisnikovih gesti. Na taj način definiraju se i pohranjuju nove geste, čije se prepoznavanje odvija u glavnom dijelu aplikacije.

Aplikacija se može podijeliti na četiri dijela: glavni izbornik, dio za pohranjivanje gesti, dio za interaktivno učenje te prikaz rezultata.

4.2.1. Glavni izbornik

Glavni izbornik predstavljen je scenom *Main menu* (*Slika 4-1*). To je ujedno i scena koja se pojavljuje na početku rada aplikacije, te preko nje korisnik odlučuje kako želi dalje postupiti. Dakle, u glavnom izborniku korisnik može odlučiti želi li interaktivno učiti abecedu znakovnog jezika (gumb *Play*, odnosno *Igra*), kreirati svoje vlastite geste (gumb *Record new gesture*, odnosno *Snimanje nove geste*), ili pak samo izaći iz aplikacije (gumb *Exit*). Na temelju gumba koji pritisne mišem, korisnika se prebacuje

u iduću odgovarajuću scenu ili se završava rad aplikacije, ukoliko je korisnik pritisnuo gumb *Exit*.



Slika 4-1 Glavni izbornik

Prijelaz u novi dio aplikacije na pritisak gumba ostvaren je skriptom *Menu* (hrv. *Izbornik*).

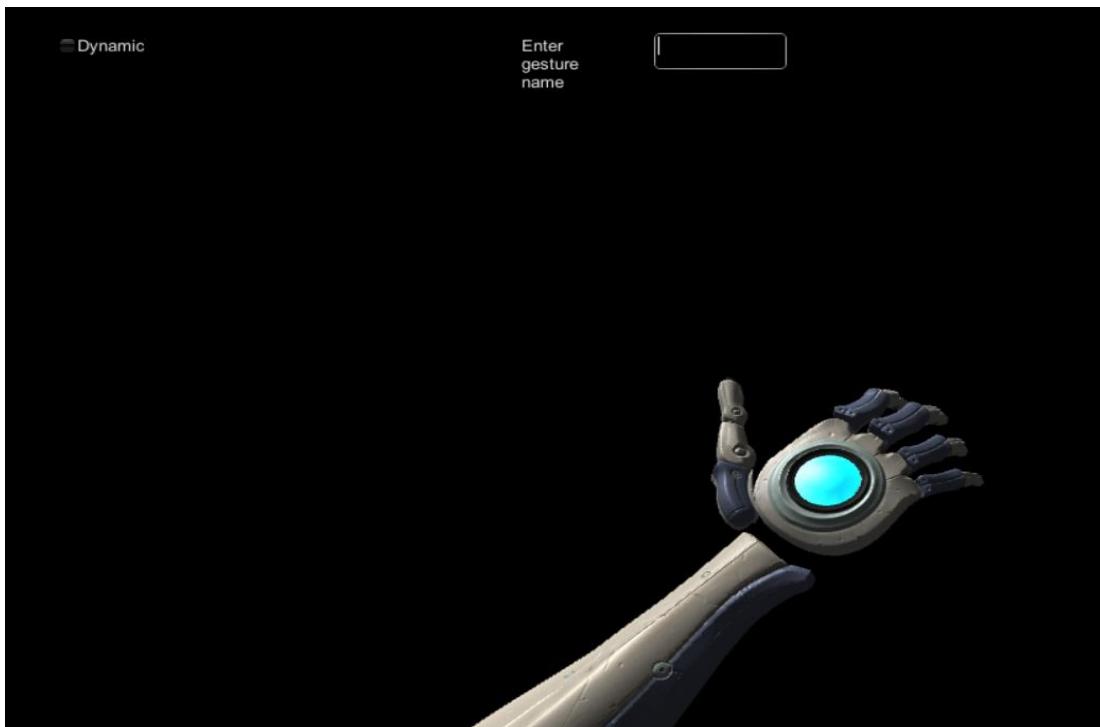
Primjer preusmjeravanja korisnika u drugi dio aplikacije pri pritisku gumba *Play* (hrv. *Igra*) pozivom nove scene dan je sljedećim isječkom koda:

```
public void OnPlay()
{
    SceneManager.LoadScene("Gestures");
}
```

Nadalje, prije opisa dijela za raspoznavanje gesti, dobro je prvo opisati dio aplikacije za snimanje i pohranjivanje gesti. Naime, na njemu se kasnije bazira i prepoznavanje gesti.

4.2.2. Pohranjivanje gesti

Dio aplikacije zadužen za snimanje, odnosno pohranu gesti, napravljen je u sceni *Record and Playback* (Slika 4-2). Navedena scena i pripadajuće skripte, preuzete su iz paketa sa sredstvima za korištenje Leap Motion kontrolera, koji je bio spomenut na početku poglavlja.



Slika 4-2 Snimanje geste

Nad scenom su napravljene određene modifikacije, pošto je originalna verzija omogućavala snimanje i reproduciranje gesti. Modifikacijom se onemogućilo reproduciranje gesti, ali se zadržao postupak snimanja kojim se ostvaruje pohrana važnih informacija o gesti. Također, omogućen je unos naziva geste, te odabir je li gesta statička ili dinamička, odnosno je li njeno trajanje duljine jednog okvira ili više. Naime, u ovom slučaju statickim gestama smatraju se geste pri kojima je bitan samo položaj prstiju ruke i same ruke, te je za njihovo prepoznavanje relevantan samo

jedan okvir. Pri takvim gestama očekuje se da ruka miruje. S druge strane dinamičke geste predstavljaju cijeli pokret, te se stoga protežu kroz više okvira.

Snimanju gesti nadodano je i odbrojavanje, kako bi se korisnik mogao pripremiti na izvođenje geste, i znati trenutak kada snimanje započinje. Snimanje geste korisnik započinje pritiskom tipke 'N' na tipkovnici, nakon čega započinje odbrojavanje. Pritiskom tipke 'S' korisnik označava završetak snimanja geste.

Također, snimanje geste temelji se na serijalizaciji primjerka okvira (engl. *Frame*). Serijalizacijom se podaci pohranjeni u okviru pohranjuju kao niz bajtova u .bytes datoteku. Pohranjenim podacima kasnije se može pristupiti deserijalizacijom istih datoteka [5].

Za pohranu podataka okvira, korištena je metoda iz razreda *LeapRecorder*, koji je dostupan u sklopu paketa preuzetog s Interneta. Primjer pohrane okvira prikazan je sljedećim isječkom koda:

```
/** Saves the recorded frames to a file, overwriting an existing file. */
public string SaveToNewFile(string path) {
    if (File.Exists(@path)) {
        File.Delete(@path);
    }

    FileStream stream = new FileStream(path, FileMode.Append, FileAccess.Write);
    for (int i = 0; i < frames_.Count; ++i) {
        byte[] frame_size = new byte[4];
        frame_size = System.BitConverter.GetBytes(frames_[i].Length);
        stream.Write(frame_size, 0, frame_size.Length);
        stream.Write(frames_[i], 0, frames_[i].Length);
    }

    stream.Close();
    return path;
}
```

4.2.3. Interaktivno učenje znakovne abecede

Ovaj dio aplikacije radi na način da se slučajnim odabirom odabere slovo za koje je korisnik dužan napraviti odgovarajuću gestu, te je ostvaren u sceni *Gestures* (hrv.

Geste). Korisniku je pritom dan i primjer geste koju treba napraviti. Ukoliko je napravljena gesta ispravna, korisniku se daje novo slovo na ‘pogađanje’, te mu se povećava broj osvojenih bodova. U igri također postoji i ograničeno vrijeme u kojem korisnik treba napraviti što više gesti, a ukoliko to želi, korisnik može i preskočiti pojedinu gestu te prijevremeno izaći iz igre. Naravno, pri izvođenju geste korisnik može i vidjeti model svoje ruke u sceni. Model ruke također je preuzet iz paketa Assets.



Slika 4-3 Izgled igre

Međutim, kako bi se uopće moglo prepoznati je li ostvarena gesta ispravna, odnosno o kojoj je gesti riječ, potrebno je ostvariti raspoznavanje gesti.

Raspoznavanje gesti je u ovom slučaju napravljeno samo za statičke geste, odnosno geste duljine jednog okvira, te je ostvareno skriptom *RecognizeGestures.cs*. Skripta radi na način da u svakom trenutku igre provjerava podatke okvira koje dobiva od Leap Motion kontrolera.

Za svaki okvir se provjerava sadrži li on položaj ruke koji bi se mogao poistovjetiti s nekom od prethodno snimljenih gesta, a sve prethodno snimljene geste pohranjene su u obliku .bytes datoteka.

Kako bi se trenutni okvir mogao usporediti s onim koji je prethodno pohranjen, potrebno je deserijalizirati pohranjeni okvir, čime se ujedno dobiva njegova kopija. Za trenutni okvir i onaj pohranjeni, odnosno deserijalizirani okvir, računa se razlika u položajima ruku. Prvo se provjerava sadrže li oba okvira jednak broj ruku. Ukoliko to nije slučaj, sa sigurnošću se može reći da nije riječ o istoj gesti. Nadalje, ako pak broj ruku je jednak, uspoređuje se broj ispruženih prstiju ruku, te naposljetu i položaj svake kosti ruke u odnosu na normalu dlana. Ukoliko postoji razlika između ruku, ona će biti definirana brojčanom vrijednošću. Tako se primjerice razlika položaja određene kosti ruke u odnosu na normalu dlana računa kao razlika vektorskih skalarnih produkata vektora smjera kosti i vektora normale dlana. Primjer računanja razlike prikazan je sljedećim programskim isječkom:

```
//check relative position of every finger (every bone in finger) to hand normal
    for (int i = 0; i < 5; i++)
    {
        Bone modelBone;
        Bone currentBone;

        foreach (Bone.BoneType boneType in
        (Bone.BoneType[])Enum.GetValues(typeof(Bone.BoneType)))
        {
            modelBone = modelHand.Fingers[i].Bone(boneType);
            currentBone = currentHand.Fingers[i].Bone(boneType);

            float modelBoneNormal = compareBonePositionToHandNormal(
                modelBone.Direction.Normalized,
                modelHand.PalmNormal.Normalized);

            float currentBoneNormal = compareBonePositionToHandNormal(
                currentBone.Direction.Normalized,
                currentHand.PalmNormal.Normalized);

            totalSumOfDifference += Mathf.Abs(currentBoneNormal -
            modelBoneNormal);
        }
    }
```

Na taj način se može zaključiti i da je gesta koju je korisnik napravio, odnosno trenutna gesta, najvjerojatnije ekvivalentna pohranjenoj gesti s najmanjom razlikom, naravno, ukoliko je ta razlika dovoljno mala.

Raspoznavanje gesti napravljeno je za većinu slova internacionalne jednoručne znakovne abecede te slova 'A' i 'B' hrvatske dvoručne znakovne abecede. Dobro je i napomenuti da uređaj nema većih problema pri raspoznavanju dvoručnih gesti, ukoliko prepozna da se u njegovom vidnom polju nalaze dvije ruke. U tom slučaju položaji ruku za obje ruke obrađuju se na isti način kao i za samo jednu.

4.2.4. Prikaz rezultata

Pažljivi čitatelj mogao je primijetiti kako se u prethodnom potpoglavlju spomenulo i bodovanje gesti. Upravo to bodovanje gesti dio je rezultata koji se ispisuje na kraju igre.

Nakon isteka vremena igre ili prijevremenog prekida igre, korisnika se preusmjerava na konačnu scenu naziva *Score* (hrv. *Rezultat*), koja prikazuje uspjeh korisnika u igri (*Slika 4-4*). Ovdje se korisniku prikazuje broj osvojenih bodova, kao i broj ispravno napravljenih gesti od ukupnog broja traženih gesti.

U ovom trenutku korisnik ima mogućnost izaći iz igre, ili se pak vratiti na glavni izbornik.



Slika 4-4 Rezultat

4.3 Problemi i moguća poboljšanja

Kao što je ranije spomenuto u poglavlju, aplikacija trenutno radi samo sa statičkim gestama, te bi jedna od mogućnosti nadogradnje zasigurno bila i implementacija prepoznavanja dinamičkih gesti. Samo prepoznavanje dinamičkih gesti prepostavljalo bi svojevrstan izazov, iz razloga što je potrebno definirati početak i kraj određene dinamičke geste. Iz tog razloga, raspoznavanje slova 'J' i 'Z' nije implementirano jer ona pripadaju dinamičkim gestama.

Nadalje, trenutno prepoznavanje gesti često nailazi na probleme pri razlikovanju sličnih gesti, te se može dogoditi da dođe do slučajne zamjene jedne geste drugom. Primjerice, položaj ruke za znak 'S' često se pogrešno interpretira kao slovo 'E', upravo iz razloga što su geste u ovom slučaju relativno slične. Svojevrstan problem također predstavlja prepoznavanje slova kod kojih je više prstiju zaklonjeno. U tom slučaju Leap Motion može samo prepostaviti položaje zaklonjenih prstiju, te je pojava pogreške očekivana, a ponekad se uopće ne može niti prepoznati slovo. To je primjerice slučaj kod slova 'G' i 'H'. Preciznije raspoznavanje svakako bi se moglo ostvariti efikasnijim načinom raspoznavanja i učenja gesti, na primjer upotrebom dubokih neuronskih mreža.

Međutim, dobro je napomenuti i da ponekad i sam uređaj pogrešno detektira položaj ruke, te se u tom slučaju može utjecati samo na njegove uvjete rada. Ovaj problem već je opisan na kraju drugog poglavlja.

5. Zaključak

Unatoč pokojem nedostatku ili pak komplikaciji pri radu, Leap Motion se zaista može smatrati uređajem koji cijelo iskustvo interakcije čovjeka s računalom dovodi na novu razinu. Preciznost praćenja pokreta koju nudi ovaj uređaj, otvara nova vrata razvoju aplikacija te cijelom doživljaju interakcije s virtualnim svijetom, čineći ju bliže onom sa stvarnim svijetom nego ikada prije, dok uporaba gesti dobiva potpuno novu dimenziju.

S druge strane, razvoj aplikacije koja zahtjeva upotrebu novih, neugrađenih gesti, može se vrlo brzo zakomplikirati. Sami proces zaista preciznog raspoznavanja i razlikovanja sličnih gesti, kompleksniji je nego što se to možda na prvi pogled čini, a povećanje broja gesti, često znači i povećanje mogućnosti za pogrešku. Također, u obzir se moraju uzeti i vanjski čimbenici, odnosno uvjeti rada, na koje se ne može uvijek utjecati. No, navedeni problemi ne bi se trebali smatrati kamenom spoticaja, već novim prostorom za napredak i razvoj ovog područja, kao i pomicanje granica svijeta kakvog poznajemo.

6. Literatura

[1] Wikipedia, Leap Motion

https://en.wikipedia.org/wiki/Leap_Motion, travanj 2016

[2] Leap Motion, About Us

<https://www.leapmotion.com/about-us>, lipanj 2016

[3] SparkFun, Leap Motion Teardown

<https://learn.sparkfun.com/tutorials/leap-motion-teardown#introduction>, svibanj 2016

[4] Leap Motion, How Does the Leap Motion Controller Work?

<http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>, svibanj 2016

[5] Leap Motion Developer, Documentation

<https://developer.leapmotion.com/documentation/index.html?proglang=current>, svibanj 2016

[6] Thomas Winther, The Leap Motion's sensitivity to lighting

<https://thomaswinther.no/techtalk/is-the-leap-motion-sensitive-to-lighting>, svibanj 2016

[7] Wikipedia, Gesture

<https://en.wikipedia.org/wiki/Gesture>, lipanj 2016

[8] Wikipedia, Sign Language

https://en.wikipedia.org/wiki/Sign_language, svibanj 2016

[9] Dodir, Načini komunikacije

<http://www.dodir.hr/hzj.php#.V1w99jW52CA>, svibanj 2016

[10] Stranica GNU tima za hrvatske prijevode

<http://www.gnu.org/server/standards/translations/hr/>, lipanj 2016

Sažetak

Naslov: Interakcija gestama u virtualnoj sceni pomoću uređaja Leap Motion

U ovom radu obrađena je interakcija gestama u virtualnoj sceni pomoću uređaja Leap Motion. Rad započinje upoznavanjem čitatelja s uređajem Leap Motion, njegovim funkcionalnostima i nedostacima. Nadalje, u radu se opisuje pojам geste te daje uvid u znakovni jezik kao način komunikacije koji koristi geste. Također, rad objašnjava i upotrebu gesti u sklopu Leap Motion uređaja. Naposljetu, opisuje se aplikacija čiji je cilj demonstrirati postupak raspoznavanja gesti, te se čitatelju daje uvid i u način ostvarenja navedenog postupka. Konačno, opisuju se i problemi koji nastaju pri radu aplikacije i raspoznavanju gesti, kao što su razlikovanje sličnih gesti i raspoznavanje u slučaju zaklonjenih dijelova ruku.

Ključne riječi: Leap Motion, geste, praćenje pokreta, znakovni jezik

Abstract

Title: Interaction in Virtual Scenes by Gestures Using Leap Motion Device

This paper describes interaction in virtual scenes by gestures using Leap Motion device. The thesis begins with the introduction to the Leap Motion device, as well as its functionalities and flaws. Furthermore, reader is introduced to the concept of gesture and sign language, as a way of communication that uses gestures. The usage of gestures in combination with Leap Motion device is also briefly described. Finally, the thesis describes the application developed as a part of this thesis. The app demonstrates the process of gesture recognition, while the insight in how the process is realized is provided in the paper. Finally, a quick overview of problems that arise during the usage of application is given. Some of the problems are the discernment, as well as the recognition during tracking of not completely visible hands.

Keywords: Leap Motion, gestures, motion tracking, sign language