

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 1037

**Sustav za mjerenje motorike
glave i vrata primjenom uređaja
Oculus Rift**

Ivan Hakštok

Zagreb, lipanj 2015.

Zagreb, 2. ožujka 2015.

DIPLOMSKI ZADATAK br. 1037

Pristupnik: **Ivan Hakštok (0036459827)**
Studij: Računarstvo
Profil: Programsko inženjerstvo i informacijski sustavi

Zadatak: **Sustav za mjerenje motorike glave i vrata primjenom uređaja Oculus Rift**

Opis zadatka:

Upoznati se s postojećom literaturom iz područja primjene virtualne stvarnosti u evaluaciji motoričkih sposobnosti glave i vrata. Upoznati tehnologije koje će se koristiti u izradi diplomskog rada, a odnose se na uređaj Oculus Rift i Unity razvojno okruženje za računalne igre. Na temelju ovih tehnologija, oblikovati i programski ostvariti sustav za mjerenje motoričkih sposobnosti glave i vrata, koji ispitanika vodi kroz seriju motoričkih testova. Testovi se odnose na pozicioniranje, dohvaćanje i praćenje raznih virtualnih objekata, a završavaju sa svrhovitim računalnom igrom (engl. serious game) koja objedinjuje karakteristike različitih testova. Strukturirati odgovarajući repozitorij podataka u datotečnom sustavu, u koji se snimaju vremenske serije kutova gibanja glave ispitanika i cjelokupnog stanja virtualne scene tijekom provedbe definirane serije testova. Sučelje za supervizora sustava treba omogućiti unos osnovnih informacija o ispitaniku, supervizoru i mjerenju, te omogućiti kontrolu supervizoru nad tijekom izvođenja testova.

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: 2. ožujka 2015.

Rok za predaju rada: 30. lipnja 2015.

Mentor:



Prof. dr. sc. Željka Mihajlović

Djelovođa:



Doc. dr. sc. Igor Mekterović

Predsjednik odbora za
diplomski rad profila:



Prof. dr. sc. Krešimir Fertalj

Sadržaj

1.. Uvod	1
2.. Značaj mjerenja pokretljivosti glave i vrata	2
2.1.. Primjena virtualne stvarnosti pri mjerenju	2
3.. Oculus Rift	4
4.. Testovi sposobnosti kretanja	6
4.1.. Određivanje područja pokretljivosti	6
4.2.. Test statičkog pozicioniranja	10
4.3.. Test skokovitog pozicioniranja	10
4.4.. Test praćenja	11
5.. Analiza podataka	13
6.. Programska implementacija	16
6.1.. Aplikacija za provođenje testova	17
6.1.1.. Radno okruženje Unity	17
6.1.2.. Oblikovanje aplikacije	18
6.1.3.. Scena za postavljanje parametara	19
6.1.4.. Zajednički elementi scena koje koriste Oculus Rift	21
6.1.5.. Scena za određivanje područja pokretljivosti	22
6.1.6.. Scena testa statičkog pozicioniranja (<i>Zero test</i>) . .	22
6.1.7.. Scena testa skokovitog pozicioniranja (<i>Step test</i>) .	23
6.1.8.. Scena testa praćenja (<i>Tracking test</i>)	24
6.1.9.. Igra	25

6.2.. Aplikacija za analizu	28
6.2.1.. Oblikovanje aplikacije	28
6.2.2.. Baza podataka	29
6.2.3.. Rad aplikacije	29
7.. Rezultati	34
8.. Zaključak	36
Literatura	37
A.. Shema Oculus Rift uređaja	41

1. Uvod

Bol u vratu koja rezultira smanjenom pokretljivošću vrata je česta pojava u ljudskoj populaciji, te od bolova u vratu godišnje pati i do 50% populacije. Uzroci boli se kreću od privremenih, kao što su istegnuće mišića, do trajnih trzajnih ozljeda vrata, nastalih kao posljedica nesreće. Bol u vratu može dovesti do pogoršanja u obilježjima pokreta vrata kao što su područje pokretljivosti (ROM, eng. *range of motion*), preciznost repositioniranja i izdržljivost vratnih mišića [1].

U posljednje vrijeme, razvojem dostupnih tehnologija virtualne stvarnosti i HMD (*head-mounted display*) uređaja, pojavila se mogućnost analize kretanja vrata korištenjem upravo tih tehnologija. Precizno je moguće odrediti vektor položaja i rotacije glave, te samim time i rotaciju i nagib vrata, te je iz tih podataka moguće odrediti značajke koje bi ukazivale na eventualne potencijalne pokrete koji uzrokuju bol.

Cilj ovog rada je razvoj sustava za objektivnu procjenu poteškoća u kretanju vrata i glave koristeći tehnologiju Oculus Rift. Sustav bi trebao predstavljati poboljšanje u odnosu na postojeće metode. Objektivnom procjenom mogla bi se točnije procijeniti razina boli i ozljede te samim time omogućiti razlučivanje između različitih skupina ljudi s poteškoćama u kretanju vrata s obzirom na stupanj ozljede.

2. Značaj mjerenja pokretljivosti glave i vrata

Uspostavljanje objektivne i standardizirane metode radi utvrđivanja postojanja problema s kretanjem u području glave i vrata predstavlja velik problem npr. pri određivanju odštete kod osiguravajućih tvrtki [2]. Pokreti i stabilnost glave ovise o sustavu kontrolnih mehanizama između osjetila na glavi (npr. ravnoteža) i vratnih mišića. Kod ljudi s kroničnom boli u vratu te kod ljudi s trzajnim ozljedama vratne kralježnice postoji smanjena mogućnost kontrole pokreta u odnosu na zdrave ljude [3].

Problem kod mjerenja i utvrđivanja problema s kretanjem je u tome što je teško odrediti pouzdanost testa. Testovi su ograničeni tehnologijom, te nemogućnost pokrivanja ili mjerenja unutar cijelog područja pokretljivosti glave rezultira u nepouzdanim rezultatima. Međutim, s razvojem preciznog, efikasnog i lako dostupnog uređaja kao što je Oculus Rift, moguće je lakše ostvariti neke od testova za analizu pokretljivosti, te samim time dobiti pouzdaniji sustav za analizu kretanja.

2.1. Primjena virtualne stvarnosti pri mjeranju

Nekoliko prijašnjih radova temeljilo se na korištenju virtualne stvarnosti za objektivizaciju ozljeda i pokretljivosti vrata [1, 3, 4]. Virtualna stvarnost

omogućava da se osoba postavi u virtualnu scenu u kojoj je moguće ostvariti interakciju s objektima i samom scenom, te je također moguće prikupiti informacije o kretanjama glave osobe (što u ovom slučaju samo predstavlja rotiranje kamere unutar scene), kao i informacije o stanju svih objekata unutar scene. Ispitaniku je moguće zadati jednostavne zadatke u sceni čija je svrha ispitivanje sposobnosti kretanja vrata i glave.

Ovaj rad pokušava ostvariti sustav sličan sustavima opisanima u navedenoj literaturi, ali s naglaskom na lako dostupne tehnologije Unity i Oculus Rift, koje iako primarno namijenjene korištenju u video igrama, imaju mogućnosti koje ih čine korisnima u drugim područjima, među kojima je i zdravstvo. Laka dostupnost i pristupačnost omogućava znatna poboljšanja u dizajnu sustava.

3. Oculus Rift

Uređaj Oculus Rift je HMD (eng. *Head-mounted display* - naglavni ekran) uređaj za prikaz virtualne stvarnosti razvijen od strane tvrtke Oculus VR. Oculus Rift omogućuje precizno praćenje položaja glave s malom latencijom (ispod 20 milisekundi), te ostvaruje prikaz pomoću OLED ekrana s malom perzistencijom, što eliminira pojave kao što je zamućivanje pokreta (eng. *motion blur*). Uređaj u sebi sadrži žiroskop, akcelerometar i magnetometar. Te tri komponente povezane su s mikrokontrolerom koji putem USB veze prenosi podatke o poziciji i rotaciji glave (detaljna shema nalazi se u prilogu A).

Oculus Rift je prvi put najavljen 2012. godine, te je od tada prošao kroz par inačica. U trenutku pisanja ovog rada, konačna javna inačica uređaja još nije puštena u prodaju, no kao pripremu za tu inačicu, te za razvoj aplikacija koje koriste Oculus Rift, proizvedene su razvojne inačice (eng. *Development Kit*).

Prednost Oculus Rift-a je mogućnost jednostavne integracije s razvojnim okruženjem kao što je Unity. Moguće je lako dohvaćati i spremati podatke poslane preko USB veze, te slati povratnu informaciju, tj. sam prikaz na dva ekrana unutar Oculus-a pomoću HDMI veze.

U trenutno najnovijoj inačici Oculus-a, svaki ekran ima rezoluciju od 960*1080, te može osvježavati sliku brzinom od 90Hz. Ukupna zajednička veličina oba ekrana je 7 inča, a odvojenost ekrana služi da bi se onemogućilo da jedno oko vidi prikaz namijenjen drugom oku. Također,

Oculus omogućuje polje pogleda (FOV - *Field of view*) do 100 stupnjeva, s 6 stupnjeva slobode kretanja (rotacije glave oko x, y ili z osi te kretanje glave po x, y ili z osi)[5].

Praćenje lokacije glave je u novim inačicama uređaja ostvareno koristeći infracrvenu kameru, međutim u ovom radu nema potrebe za praćenjem lokacije glave, već je samo bitna rotacija.

4. Testovi sposobnosti kretanja

Sposobnost kretanja odražava se u nekoliko čimbenika, kao što su brzina kretanja, trzaji, oscilacije, glatkoća pokreta i vrijeme reakcije. Da bi se dobila kompletna slika o kretanju, potrebno je ostvariti testove pogodne za analizu tih čimbenika. U ovom radu ostvarena su tri testa:

- test statičkog pozicioniranja
- test skokovitog pozicioniranja
- test praćenja

Tri testa su ostvarena unutar virtualne stvarnosti, te je također osmišljena i igra koja obuhvaća aspekte više testova. U suštini, svi testovi se zasnivaju na usmjeravanju pogleda prema objektu koji se nalazi na ekranu unutar područja pokretljivosti ispitanika, te se razlikuju u lokacijama i uzorcima kretanja objekta. Moguće je spremirati podatke o testu te kretanjima objekta i smjeru pogleda ispitanika.

4.1. Određivanje područja pokretljivosti

Područje pokretljivosti može se protumačiti kao granica unutar koje je dozvoljeno izvršavati testove, s namjerom da ispitanik uspije proći sve testove. Osnovni način određivanja područja pokretljivosti je kretanje ispitanika glavom u četiri smjera - gore, dolje, lijevo i desno. Ispitanik bi se trebao kretati u ta četiri smjera koliko mu pokretljivost vrata i glave to do-

pušta. Na ovaj način dobijemo četiri ekstrema, po jedan na negativnom i pozitivnom dijelu x i y osi.

Sljedeći korak je interpolacija cjelokupnog područja pokretljivosti na temelju dobivenih ekstrema. Najjednostavniji način za provođenje testova i izračun bilo bi pravokutno područje omeđeno s četiri pravca:

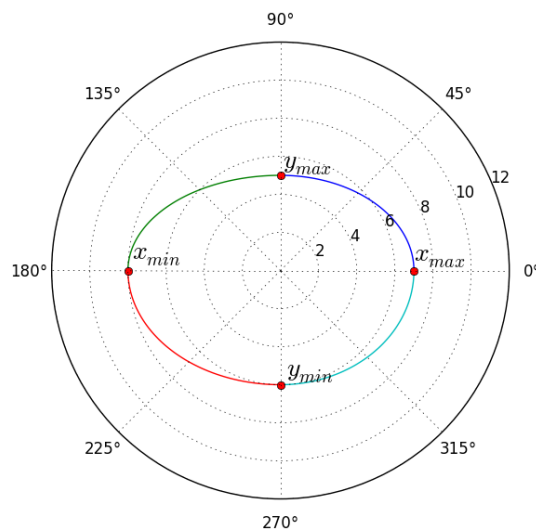
$$y = y_{max}$$

$$y = y_{min}$$

$$x = x_{max}$$

$$x = x_{min}$$

Unutar ovako definiranog pravokutnika, moguće je jednostavno odbaciti neke dijelove testova koji bi npr. slučajnim generiranjem lokacije završili izvan pravokutnika. Međutim, takav oblik nije vjerodostojni prikaz područja pokretljivosti.



Slika 4.1.: Primjer četiri segmenta u polarnom koordinatnom sustavu

Jedan od oblika koji bi bio točniji s obzirom na realan oblik područja pokretljivosti glave je elipsa. U ovom slučaju, budući da imamo informacije o četiri ekstrema područja pokretljivosti, možemo generirati četiri eliptična segmenta (jedan unutar svakog kvadranta). Svaka dva segmenta imala bi jednu zajedničku točku (jedan ekstrem). Međutim, javlja se problem odsijecanja dijelova testa koji se nalaze izvan elipse. Zbog toga je bolje prijeći iz kartezijevog koordinatnog sustava u polarni koordinatni sustav.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \quad (4.1)$$

$$x = r(\theta) \cos \theta,$$

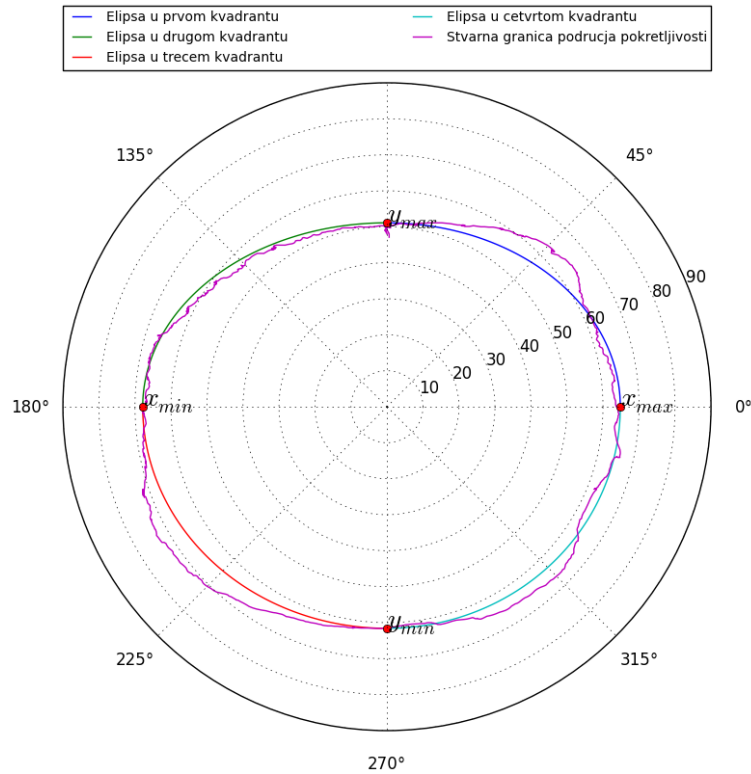
$$y = r(\theta) \sin \theta,$$

$$r(\theta) = \frac{ab}{\sqrt{(a \sin \theta)^2 + (b \cos \theta)^2}} \quad (4.2)$$

Preko parametarskog zapisa varijabli x i y , možemo iz jednadžbe 4.1 dobiti jednadžbu 4.2, kao parametarsku jednadžbu polumjera elipse u ovisnosti o kutu, gdje su a i b polumjeri elipse po x , odnosno y osi. Uvrštavanjem četiri različite kombinacije parova ekstrema po x i y osi, dobivamo četiri segmenta, jedan unutar svakog kvadranta. Na slici 4.1 može se vidjeti primjer jedne granice područja pokretljivosti. Ovakva četverodijelna elipsa se pokazala kao vrlo dobra aproksimacija tog područja, kao što se može vidjeti na slici 4.2.

Još jedna pogodnost načina rada u polarnom koordinatnom sustavu je jednostavnost generiranja lokacije unutar područja pokretljivosti stvaranjem nasumičnih vrijednosti. Umjesto stvaranja uređenog para (x, y) te zatim provjere i odbacivanja lokacije koja se ne nalazi unutar područja pokretljivosti, moguće je generirati par (θ, r) gdje je r nasumični broj između 0 i $r(\theta)$.

Problem kod ovakvog generiranja lokacije je translacija s dvodimenzionalnog prostora u trodimenzionalni. Način koji se primjenjuje u ovom



Slika 4.2.: Usporedba elipse sa stvarnim područjem pokretljivosti

radu je translacija dvodimenzionalnog vektora s ravnine $z = z_c$ na sferu konstantne udaljenosti od promatrača. Pretvorba se vrši prema formuli:

$$\begin{aligned}
 x &= r(\theta) \cos(\theta) \\
 y &= r(\theta) \sin(\theta) \\
 \vec{o}_{2D} &= x\hat{i} + y\hat{j} \\
 \vec{o}_{3D} &= \vec{o}_{2D} + z_c\hat{k} \\
 \hat{o}_{3D} &= \frac{\vec{o}_{3D}}{\|\vec{o}_{3D}\|} \\
 \vec{o}_S &= c \cdot \hat{o}_{3D}
 \end{aligned} \tag{4.3}$$

Gdje \vec{o}_{2D} predstavlja lokaciju na ravnini $z = z_c$, \vec{o}_{3D} predstavlja tu istu lokaciju u trodimenzionalnom prostoru, \hat{o}_{3D} predstavlja normalizirani oblik prethodnog vektora, a \vec{o}_S predstavlja konačni vektor lokacije na sferi polumjera c sa središtem u lokaciji promatrača. \hat{i} , \hat{j} i \hat{k} su jedinični vektori u x, y i z smjeru. Takav vektor, zbog istog mjesta izvorišta, možemo usporediti s vektorom pogleda promatrača, te dobiti podatke o razlici u kutu

pojedinačno po osima.

4.2. Test statičkog pozicioniranja

Kod testa statičkog pozicioniranja, ispitaniku se prikazuje objekt u sredini područja pokretljivosti. Ispitanik tada treba zadržati pogled na objektu neko određeno vrijeme (par sekundi). Pri tome se mjere manje oscilacije u kretanju glave i vrata.

Sredina područja pokretljivosti bi trebala predstavljati neutralnu točku, u kojoj nema naprezanja niti po jednoj osi (odnosno, nema fleksije i rotacije vrata).

Izmjerene oscilacije i greške u smjeru x i y osi koriste se da bi se utvrdila stabilnost kretanja vrata. Moguće je analizirati spektar dobivenih oscilacija, te taj spektar oduzeti od spektara dobivenih kasnijim mjerenjima radi dobivanja "filtriranih" podataka iz kojih je moguće preciznije izvući neke značajke.

4.3. Test skokovitog pozicioniranja

Test skokovitog pozicioniranja sastoji se od stvaranja objekata na lokacijama unutar područja pokretljivosti. Lokacije mogu biti nasumične ili unaprijed zadane. Ispitanik treba u što kraćem vremenu doći do sljedećeg objekta te zadržati pogled na njemu kratko vrijeme, čime se izbjegava pogodak kao rezultat nasumičnih kretnji ili prelijetanja.

Iz podataka dobivenih ovim testom moguće je izvući sljedeće objektivne značajke [6, 7]:

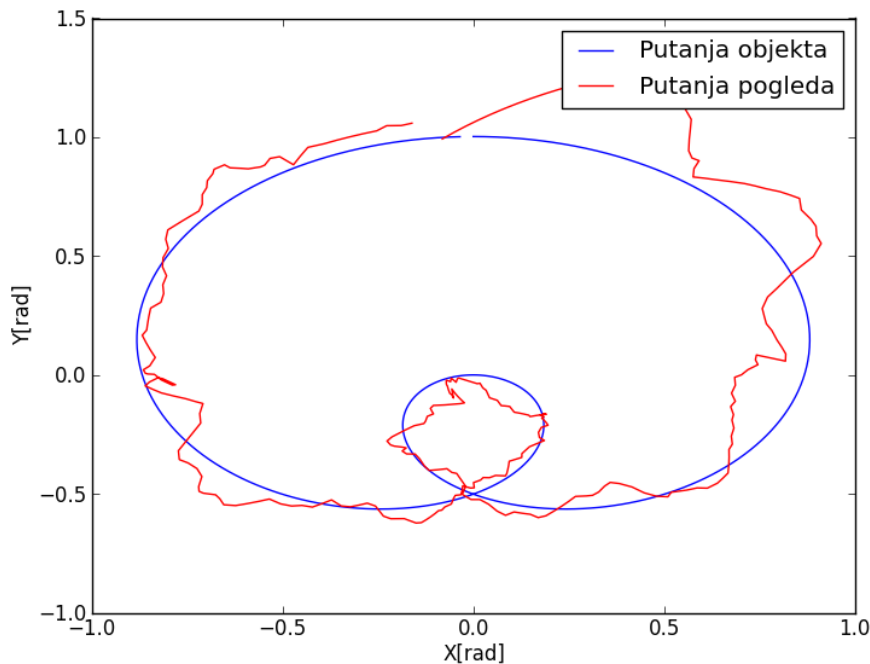
1. *Vrijeme reakcije* je vrijeme od pojavljivanja objekta do početka kretanja. Početak kretanja se može definirati kao trenutak u kojem je brzina kretanja veća od određenog postotka maksimalne brzine.
2. *Maksimalna brzina* je najveća brzina kretanja glave ispitanika ostvarena pri usmjeravanju pogleda na pojedini objekt na ekranu.
3. *Srednja brzina* je prosječna brzina od početka kretanja do pogotka.
4. *Postotak vremena do vrha* je vrijeme od početka kretanja do ostvarenja maksimalne brzine izraženo u postocima.
5. *Broj vrhova brzine* je broj lokalnih ekstrema brzine od početka kretanja do pogotka, te služi kao mjera glatkoće pokreta. Može se definirati kao broj promjena predznaka u vrijednostima akceleracije.
6. *Indeks trzaja (Jerk index)* je također mjera glatkoće pokreta dobivena iz treće derivacije kretanje po vremenu.
7. *Oštrina repozicioniranja i sklonost repozicioniranja* su mjere koje ukazuju na količinu prelijetanja ili podlijetanja cilja. Oštrina je srednja vrijednost greške ciljanja (prelijetanje ili podlijetanje), dok je sklonost repozicioniranja standardna devijacija greške.

Indeks trzaja je mjera koja normalizira trzaj s obzirom na vrijeme trajanja pokreta, te se računa prema formuli 4.4, gdje J predstavlja vektor vrijednosti trzaja duljine n , u trajanju kretanja vremena t na udaljenosti D .

$$C_j = \sqrt{\frac{1}{2} \sum_{i=1}^n J_i^2 \frac{t^5}{D^2}} \quad (4.4)$$

4.4. Test praćenja

Test praćenja sastoji se od toga da ispitanik pogledom prati objekt na ekranu koji se kreće po nekoj unaprijed zadanoj putanji. Putanja se može skalirati tako da se cijela nalazi unutar područja pokretljivosti, a moguće je mijenjati i brzinu kretanja te broj ponavljanja. Putanje su u



Slika 4.3.: Primjer usporedbe putanje objekta i pogleda promatrača

stvari funkcije zadane u parametarskom obliku - kao ulaz primaju varijablu t te vraćaju par vrijednosti (x, y) . Jedan primjer je skup funkcija 4.5:

$$\begin{aligned} x &= \cos t \\ y &= \sin 2t \end{aligned} \tag{4.5}$$

Kod praćenja, bilježi se vektor pogleda ispitanika te vektor objekta. Iz toga je moguće izvući kutove u x i y smjeru te izračunati grešku u tim smjerovima. Na temelju greške moguće je napraviti test koji u slučaju prevelike greške prilagođava brzinu kretanja objekta.

Ovakav test omogućava provjeru kontinuiranog kretanja. Pad brzine ili veća greška bi se tada javili u područjima gdje bol otežava kretanje vrata, a također se može pratiti i da li te u kojem trenutku pogled odlazi izvan područja pokretljivosti.

5. Analiza podataka

Kao rezultati testova dobivene su dvije vremenske serije vektora: jedan za pogled ispitanika i jedan za lokaciju objekta. Da bi se provodila analiza nad tim vektorima, potrebno ih je prebaciti u kutove u x i y smjeru prema sljedećoj formuli:

$$\begin{aligned}\hat{n} &= x\hat{i} + y\hat{j} + z\hat{k} \\ \alpha &= \arccos x/z \\ \beta &= \arccos y/z\end{aligned}\tag{5.1}$$

Gdje je \hat{n} početni vektor, α kut u x smjeru, te β kut u y smjeru. Vektor \hat{n} je normaliziran, što znači da je $x^2 + y^2 + z^2 = 1$.

Iz dobivenih kutova za vektore pogleda i lokacije objekta moguće je izvući neke osnovne značajke kao što su relativna i apsolutna razlika u kutu, te derivacije: brzina (prva derivacija), ubrzanje (druga derivacija) te trzaj (treća derivacija). Budući da se radi o diskretnim uzorcima, derivaciju je potrebno dobiti na sljedeći način:

$$\begin{aligned}d\alpha_i &= \frac{\alpha_{i+1} - \alpha_i}{\Delta t}, & \forall i \in \langle 1, N-1 \rangle \\ d^2\alpha_i &= \frac{d\alpha_{i+1} - d\alpha_i}{\Delta t}, & \forall i \in \langle 1, N-2 \rangle \\ d^3\alpha_i &= \frac{d^2\alpha_{i+1} - d^2\alpha_i}{\Delta t}, & \forall i \in \langle 1, N-3 \rangle\end{aligned}\tag{5.2}$$

Gdje je Δt vremenski interval uzorkovanja, a N broj uzoraka kuta. Budući da je brzina izvođenja simulacije konstantnih 60fps (*frames per se-*

cond - slika u sekundi), a podaci se snimaju jednom po slici, vremenski interval je konstantan i jednak $1/60$, odnosno $\Delta t \approx 0.0167$. Uzevši to u obzir, moguće je preurediti jednadžbe za derivacije u oblik koji ne zahtijeva međurezultate:

$$\begin{aligned} d\alpha_i &= \frac{\alpha_{i+1} - \alpha_i}{\Delta t}, & \forall i \in \langle 1, N-1 \rangle \\ d^2\alpha_i &= \frac{\alpha_{i+2} - 2\alpha_{i+1} + \alpha_i}{\Delta t^2}, & \forall i \in \langle 1, N-2 \rangle \\ d^3\alpha_i &= \frac{\alpha_{i+3} - 3\alpha_{i+2} + 3\alpha_{i+1} - \alpha_i}{\Delta t^3}, & \forall i \in \langle 1, N-3 \rangle \end{aligned} \quad (5.3)$$

Podaci o brzini, ubrzanju i trzaju su korisni jer možemo vidjeti izglađenost kretanja, kao i značajke poput vremena reakcije, maksimalne i prosječne brzine te broj lokalnih ekstrema brzine o kojima je već bilo riječi u poglavlju 4.3.

Još jedna korisna stvar koju je moguće izvući iz vremenske serije promjene kuta je spektar. Spektar omogućuje da uvid u udio pojedinih frekvencija u ukupnom signalu. Kod ljudi, većina bitnih informacija nalazi se ispod granice od 10Hz [7]. Pri računanju spektra, iz vremenske serije realnih vrijednosti dobiju se kao rezultat kompleksne vrijednosti, koje se mogu transformirati u amplitudu i fazu. Budući da se radi o signalu u diskretnoj domeni, za dobivanje spektra potrebno je preći iz vremenske u frekvencijsku domenu pomoću diskretne Fourierove transformacije [10]. Transformacija se vrši prema sljedećoj formuli:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N}, \quad k \in \mathbb{Z} \quad (5.4)$$

Vektori vrijednosti koje ulaze u transformaciju mogu biti raznih duljina. Kraće vrijednosti rezultiraju točnijim prikazom spektra kod tranzijenata, no ne daju dobru informaciju o ukupnom spektru. Zbog toga je potrebno odrediti takvu veličinu vektora vrijednosti koje se predaju transformaciji,

a da rezultira dobrim prikazom tranzijenata, te ukupnog spektra.

Budući da za obradu cijelog skupa podataka uzimamo više vektora vrijednosti, rezultat je zapravo funkcija u trodimenzionalnom prostoru, gdje x-os predstavlja početni uzorak, y-os frekvenciju dok se na z-osi nalazi vrijednost amplitude za neku frekvenciju dobivene iz vektora duljine N koji započinje na uzorku na mjestu čija je vrijednost na x-osi. Ovakva trodimenzionalna funkcija nije kontinuirana već diskretna, jer su vrijednosti na x i y-osima diskretne.

Kod diskretne Fourierove transformacije, moguće je dobiti neželjene pojave zbog diskontinuiteta na početku i kraju signala, što se događa jer je jedna od pretpostavki za izvršavanje same transformacije ta da je signal periodičan. Zbog toga je potrebno izgladiti krajeve signala da bi se uklonile neželjene pojave, a sačuvao spektar. U tu svrhu koristi se takozvana funkcija prozora (eng. *Window function*). Jedan primjer takve funkcije je Hammingov prozor, dan formulom 5.5:

$$w(n) = \alpha - \beta \cos\left(\frac{2\pi n}{N-1}\right) \quad (5.5)$$

Ovisno o odabranim pozitivnim vrijednostima α i β , vektor duljine N se skalira množenjem s Hammingovim prozorom, te se zbog svojstva funkcije krajevi skaliraju na vrijednosti koje smanjuju diskontinuitet.

6. Programska implementacija

Implementacija sustava za praćenje glave i vrata napravljena je kao dvije zasebne aplikacije: aplikacija za provođenje testova i snimanje podataka o kretanju te aplikacija za analizu podataka dobivenih prethodnom aplikacijom. Obje aplikacije izrađene su na operacijskom sustavu Windows 7. Aplikacija za provođenje testova razvijana je u okruženju Unity koristeći programski jezik C#, dok je aplikacija za analizu razvijana u okruženju Visual Studio 2012, također koristeći programski jezik C#. Za izradu grafičkih modela korišten je alat Blender. U idućim poglavljima bit će objašnjene pojedinosti kod implementacije obje aplikacije te tijekom izrade programskog rješenja. Ključne faze izrade su:

- postavljanje uređaja Oculus Rift i pripadnih alata
- postavljanje radne okoline za izradu prve aplikacije korištenjem Unity okruženja
- izrada aplikacije za provođenje testova i implementacija testova
- postavljanje radne okoline za izradu druge aplikacije korištenjem Visual Studio 2012 okruženja
- izrada baze podataka
- izrada aplikacije za analizu podataka

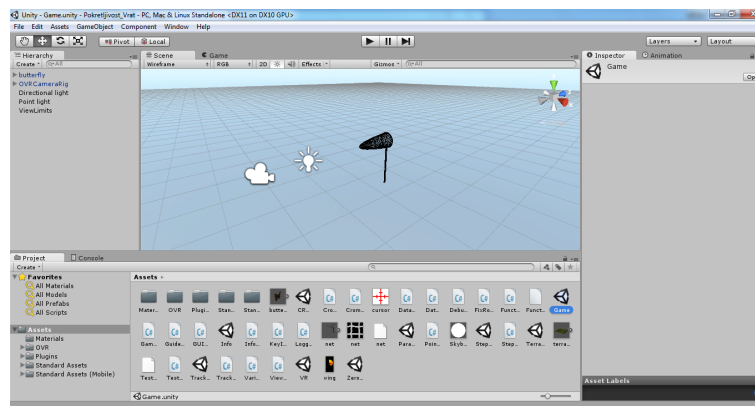
Razlog za izradu dvije odvojene aplikacije je rasterećenje provođenja testova od zahtjevnih proračuna koji bi mogli utjecati na brzinu izvođenja aplikacije. Da bi se izbjegle neželjene pojave kao što je trzanje slike,

aplikacija za provođenje testova radi minimalan broj potrebnih funkcija za provođenje testova i spremanje rezultata, što omogućuje da se izvršava na gotovo konstantnih 60 slika u sekundi.

6.1. Aplikacija za provođenje testova

6.1.1. Radno okruženje Unity

Radno okruženje korišteno za izradu aplikacije za provođenje testova i implementaciju testova je Unity. Unity je okruženje primarno namijenjeno izradi video igara, prvi put izdana 2005. godine. Unity ima podršku za mnoge platforme te se sve češće upotrebljava u industriji video igara, pogotovo kod manjih nezavisnih proizvođača.



Slika 6.1.: Radno okruženje Unity

Rad u Unity-u zasniva se na stvaranju scena, gdje svaka scena sadrži svoje objekte te skripte koje upravljaju ponašanjem objekata. Unity ima veliki broj ugrađenih razreda za rad s grafičkim objektima, zvukom, te fizikom objekata, sudarima, tkaninom, česticama. Pristup tim razredima ostvaruje se preko javnog sučelja.

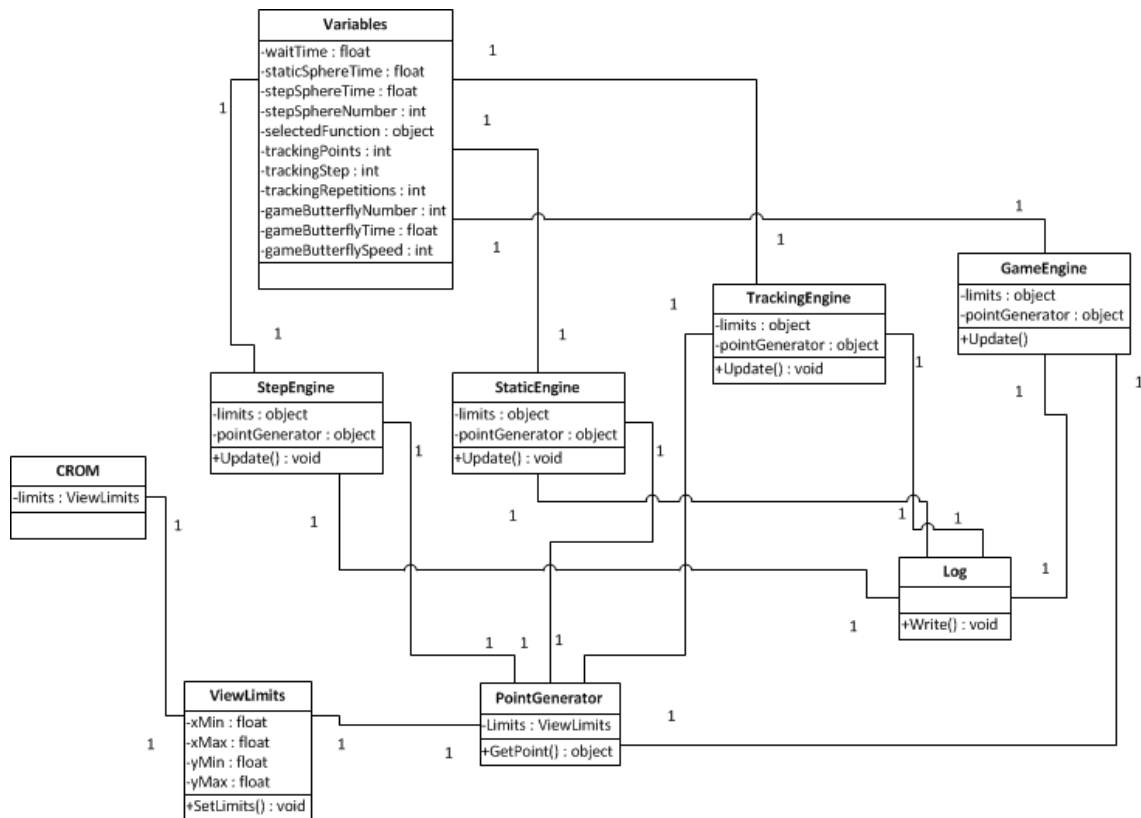
Unity u sebi ima ugrađenu podršku za uređaj Oculus Rift, što je čini prikladnom za korištenje u izradi sustava u sklopu ovog rada [8]. Za potrebu ove aplikacije, napravljeno je 6 scena:

- scena za postavljanje parametra testova
- scena za mjerenje područja pokretljivosti (CROM - cervical range of motion)
- scena za provođenje testa statičkog pozicioniranja
- scena za provođenje testa skokovitog pozicioniranja
- scena za provođenje testa praćenja
- scena svrhovite igre

6.1.2. Oblikovanje aplikacije

Aplikacija je dizajnirana na način da svaka scena sadrži razred koji služi kao upravljački sustav scene. Unutar svakog pojedinog razreda postoji funkcija `Update` koja se poziva jednom po slici, dakle u prosjeku oko 60 puta u sekundi. Ti upravljački razredi pozivaju metode iz dva statička razreda - `Variables` i `Log`, te sadrže referencu na razred `PointGenerator`. Razred `Variables` služi za spremanje parametara dok razred `Log` služi za spremanje podataka.

Razred `PointGenerator` služi sa generiranje nasumičnih točaka koje se nalaze unutar područja pokretljivosti. Podaci o području pokretljivosti su pak sadržani unutar razreda `ViewLimits`, te jedino scena za mjerenje područja pokretljivosti sadrži referencu na taj razred. Scena za test praćenja također sadrži referencu na razred `FunctionGenerator`, koji služi za generiranje putanje testa praćenja, te u sebi sadrži pozive na funkcije iz vanjske `.dll` datoteke.



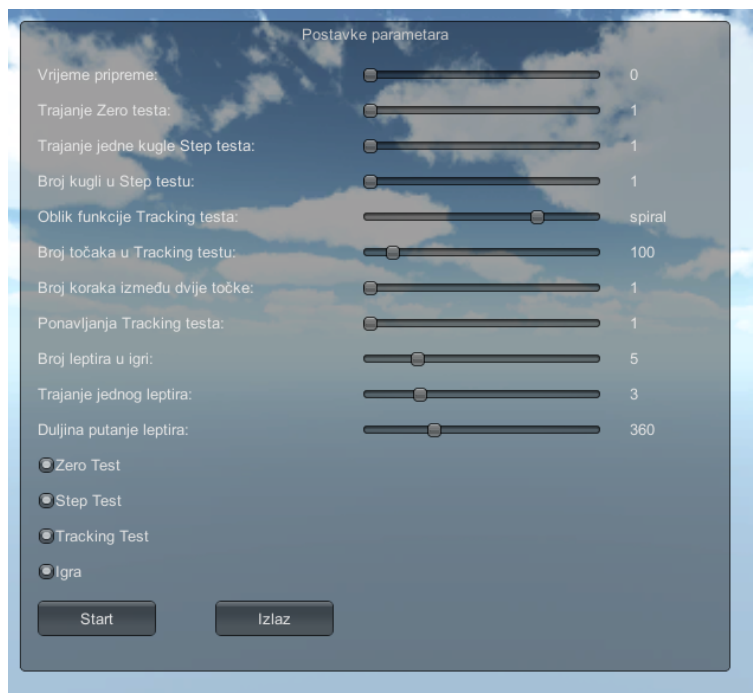
Slika 6.2.: Dijagram razreda

6.1.3. Scena za postavljanje parametara

Prva izrađena scena je scena za postavljanje parametara. Parametri koji se mogu podesiti vidljivi su na slici 6.3. Za svaki parametar moguće je preko klizećeg izbornika odabrati vrijednost u unaprijed zadanom rasponu. Također je moguće i isključiti odnosno uključiti pojedine testove.

Promjenjivi parametri su sljedeći:

- *Vrijeme pripreme* - vrijeme prije početka izvođenja bilo kakvog testa, dok se ispitanik ne prilagodi.
- *Trajanje Zero testa* - vrijeme u kojem je potrebno gledati u objekt u Zero testu (statičko pozicioniranje).
- *Trajanje jedne kugle Step testa* - vrijeme u kojem je potrebno zadržati pogled na jednoj kugli u Step testu (skokovito pozicioniranje).
- *Broj kugli u Step testu* - broj kugli koje se pojavljuju na ekranu



Slika 6.3.: Scena za postavljanje parametara

unutar Step testa.

- *Oblik funkcije Tracking testa* - izbor unaprijed zadane funkcije prema kojoj se generiraju točke unutar Tracking testa (praćenje).
- *Broj točaka u Tracking testu* - broj točaka koje se generiraju za odabranu funkciju.
- *Broj koraka između dvije točke* - broj potrebnih koraka (odnosno frame-ova) da se prijeđe is točke u sljedeću točku. Veća vrijednost rezultira sporijim kretanjem.
- *Ponavljanja tracking testa* - broj ponavljanja funkcije Tracking testa.
- *Broj leptira u igri* - ukupan broj leptira koji se generiraju na ekranu tijekom igre.
- *Trajanje jednog leptira* - vrijeme u kojem je potrebno pogled usmjeriti na leptira.
- *Duljina putanje leptira* - broj točaka za koje se generira putanja leptira.

Po završetku postavljanja parametara, nastavlja se s određivanjem područja pokretljivosti koje je osnovica za provođenje ostalih testova. Sve vrijednosti parametara predaju se u statički razred `Variables` te svaki sljedeći test može iz njega dohvatiti one parametre koji su mu potrebni.

6.1.4. Zajednički elementi scena koje koriste Oculus Rift

Scene u kojima je potreban Oculus Rift su scena za određivanje područja pokretljivosti, test statičkog pozicioniranja, test skokovitog pozicioniranja, test praćenja te igra. Te scene imaju neke zajedničke elemente koji se uglavnom odnose na kretanje glave i bilježenje podataka.

Unutar svake od navedenih scena, postoji unaprijed izgrađeni (eng. *prefab*) objekt kamere. Kretanje kamere kontrolira se pomoću uređaja Oculus Rift. Taj objekt kamere zapravo sadrži dvije kamere - jedna za lijevo i jedna za desno oko. Njihov međusobni razmak i orijentacija odgovara razmaku i orijentaciji ljudskih očiju.

Takva unaprijed izrađena kamera dolazi kao dio razvojnih alata uređaja Oculus Rift (*Oculus Rift Standard Development Kit*). Osim samih pozicija i orijentacije kamere, taj već unaprijed napravljeni objekt sadrži i programski kod koji služi za povezivanje uređaja i napravljenog objekta, kao i programski kod koji omogućuje dohvaćanje podataka o lokaciji i rotaciji.

Još jedan zajednički element za navedene scene je razred `Log` koji služi za pisanje podataka unutar datoteke. Taj razred realiziran je kao statički te sadrži metode za zapis vremena i podataka. U datoteku mogu biti zapisana dva tipa podataka - vektori pogleda i objekta, te informativne oznake. Za oba tipa podataka obavezno se bilježi vrijeme u kojem su zabilježeni, što se koristi unutar aplikacije za analizu.

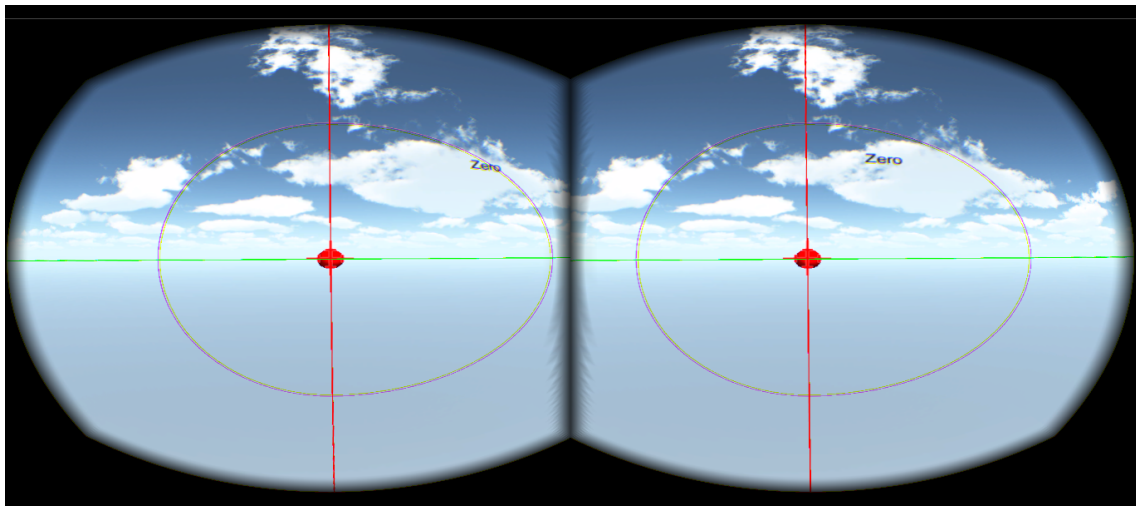
6.1.5. Scena za određivanje područja pokretljivosti

Određivanjem područja pokretljivosti stvara se granica unutar koje se izvršavaju svi testovi. Područje pokretljivosti određuje se tako da ispitanik glavom napravi maksimalan otklon u četiri smjera - gore, dolje, lijevo, desno. Time se odrede četiri maksimalna kuta pomoću kojih se interpolira područje pokretljivosti.

Područje pokretljivosti sprema se u poseban razred koji je zadužen za generiranje točaka unutar njega, te skaliranje točaka koje se nalaze izvan područja pokretljivosti.

6.1.6. Scena testa statičkog pozicioniranja (*Zero test*)

Kod testa statičkog pozicioniranja, ispitanik mora u nekom vremenu zadržati pogled na objektu. Duljina tog vremena učitava se iz statičkog razreda *Variables* unutar kojeg su spremljeni parametri.



Slika 6.4.: Scena testa statičkog pozicioniranja

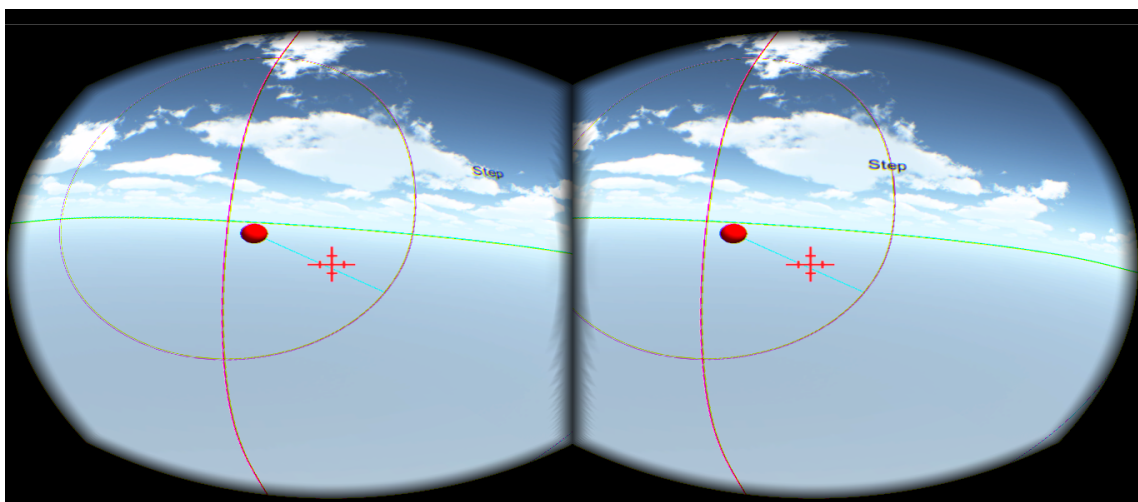
Područje pokretljivosti nema neku ulogu kod ovog testa, no kugla se na početku nalazi u središtu područja pokretljivosti. Iako je područje pokret-

ljivosti određeno elipsama od kojih svaka ima dva fokusa, kao centar se uzima točka od koje su izmjereni otkloni u sva četiri smjera, tj. neutralna pozicija glave.

Ovim testom mjere se manji trzaji i greška pri pozicioniranju glave i vrata, te su takve greške uglavnom manje od polumjera kugle koju je potrebno gledati, pa se zbog toga vrijeme ne pokreće ponovno u slučaju da pogled ispitanika ode izvan područja kugle.

6.1.7. Scena testa skokovitog pozicioniranja (*Step test*)

Kod skokovitog pozicioniranja, cilj je testirati vrijeme reakcije te brzinu i preciznost repositioniranja. Test je realiziran na način da ispitanik pogled usmjerava prema kugli koja se pojavljuje na ekranu, te zadržava pogled na kugli dok ona ne nestane. Parametri kao što su trajanje vremena pojavljivanja kugle i broj kugli koje se pojavljuju na ekranu se učitavaju iz razreda *Variables*.



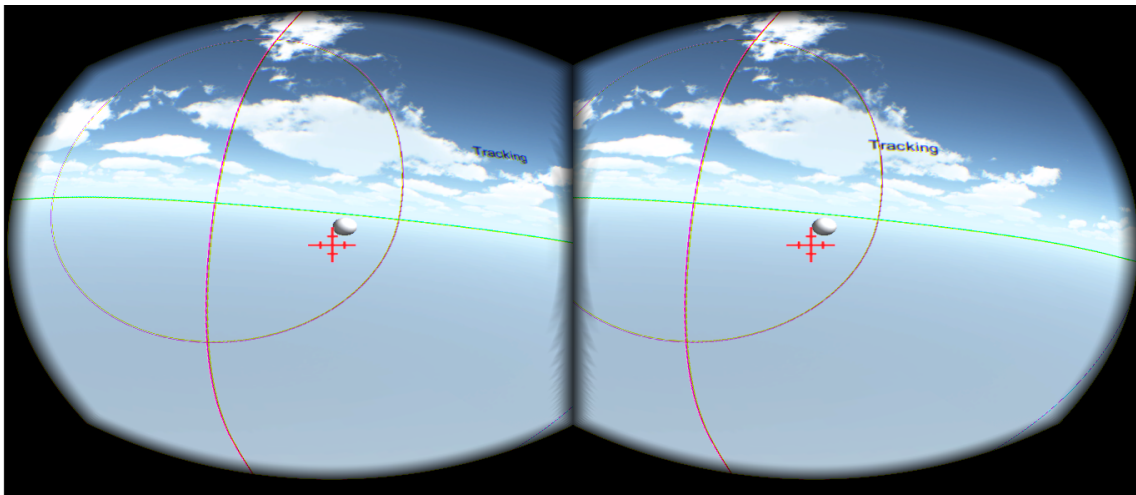
Slika 6.5.: Scena testa skokovitog pozicioniranja

Kugle se generiraju unutar područja pokretljivosti tako da se generiraju dvije slučajne varijable koje predstavljaju polarne koordinate, te pretodno prikazane kugle nestaju. Na dovoljno velik broj kugli, slučajne varijable rezultiraju ravnomjernom raspodjelom kugli unutar četiri kvadranta, te na različitim udaljenostima od neutralne pozicije.

Ispitanik bi trebao prema svakoj novo stvorenoj kugli pogled usmjeriti što je brže moguće, te se time ispituje brzina i preciznost repositioniranja.

6.1.8. Scena testa praćenja (*Tracking test*)

Kod testa praćenja, ispitanik pogledom prati kuglu koja se kreće unutar područja pokretljivosti prema nekoj unaprijed zadanoj putanji. Iz razreda `Variables` uzimaju se parametri broja točaka koje se generiraju prema putanji, broj slika koje su potrebne za prijelaz iz jedne točke u sljedeću, funkcija putanje te broj ponavljanja.



Slika 6.6.: Scena testa praćenja

Moguće funkcije putanje za odabir u sceni s parametrima učitavaju se iz vanjske biblioteke funkcija (*dll - Dynamic Link Library*). Pomoću

mehanizma refleksije (sposobnosti programa da obavlja operacije nad vlastitom strukturom prilikom izvođenja) dohvaća se popis funkcija iz biblioteke koje ispunjavaju kriterije potrebne za ispravan rad:

- funkcija kao parametar prima jednu varijablu tipa broja s pomičnim zarezom (float), te
- funkcija vraća dvije vrijednosti s pomičnim zarezom koje predstavljaju koordinate, odnosno uređeni par (x, y)

Nakon što je funkcija odabrana, izračunava se broj točaka zadanih parametrom broja točaka, na jednakim udaljenostima. Tako primjerice, ako je zadana funkcija krug, a parametar broja točaka postavljen na pet, rezultat bi bio pravilni peterokut. U kombinaciji s parametrom broja slika potrebnih za prijelaz u sljedeću točku, iz jednostavnih funkcija može se dobiti više različitih putanji.

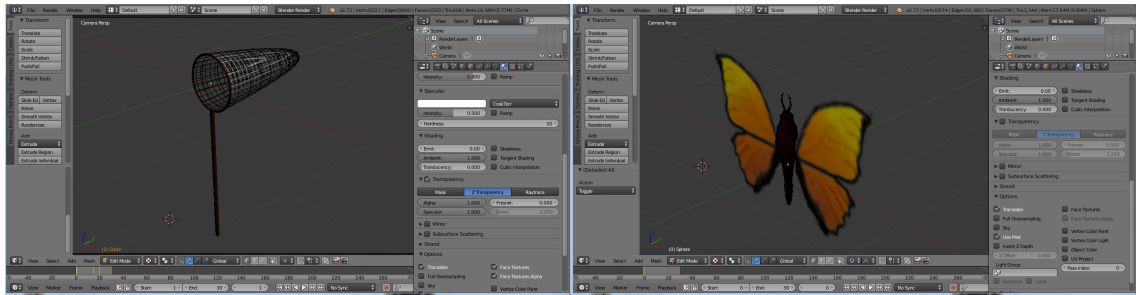
Dobivene točke potrebno je skalirati da se nalaze unutar područje pokretljivosti. To je realizirano na način da se traži najdalja točka od centra koja je izvan područja pokretljivosti. Zatim se za tu točku određuje kut u odnosu na x-os, te se izračuna polumjer područja pokretljivosti za taj kut. Omjer udaljenosti točke te dobivenog polumjera predstavlja faktor skaliranja, koji je veći od 1. Udaljenost svih točaka dijeli se s tim faktorom, pri čemu njihov kut ostaje očuvan.

6.1.9. Igra

U okviru ovog rada, potrebno je bilo ostvariti igru koja obuhvaća aspekte više testova. U tu svrhu, osmišljena je igra lova na leptire. Ispitanik mora postići što veći rezultat, a pri igranju se još uvijek prikupljaju podaci o kretanju. Igra je kao scena stavljena na kraj ostalih testova, što znači da se uvijek izvršava posljednja.

Prvi korak pri izradi igre bio je izrada trodimenzionalnih modela leptira i mreže. Modeli su napravljeni koristeći alat Blender3D, te zatim uvezeni u Unity okruženje. Materijali kao što su tekstura krila ili mreže napravljeni

su korištenjem alata Gimp.



Slika 6.7.: Izrada modela

Kao što je već rečeno, igra treba obuhvaćati više testova, te je potrebno na neki način ostvariti bodovanje za uspješno odigranu igru. Igra je zato osmišljena na sljedeći način:

- leptir se pojavljuje na nekoj lokaciji na ekranu
- ispitanik pogledom mora otići do leptira u određenom vremenu, u suprotnom leptir nestaje
- ispitanik mora pogledom pratiti leptira dok on ne bude uhvaćen
- igra završava kada nema više leptira

Sustav bodovanja trebao bi biti takav da se nagrađuje što preciznije izvođenje zadatka. Stoga se bodovi ostvaruju na sljedeći način:

- za dolazak pogledom do leptira, ispitanik dobije do 1000 bodova, ovisno o brzini
- za praćenje leptira, ispitanik dobije bodove ovisne o kutu između pogleda i leptira - što je kut manji, broj bodova je veći

Točne formule za određivanje bodovanja zadane su na sljedeći način:

$$p_{pogodak} = \lfloor 1000 \cdot (1 - \frac{t_{trenutno} - t_{pocetno}}{t_{max}}) \rfloor, \quad t_{trenutno} < t_{pocetno} + t_{max}$$

$$p_{pracenje} = \lfloor \frac{10}{|\alpha|} \rfloor, \quad \alpha < 5$$

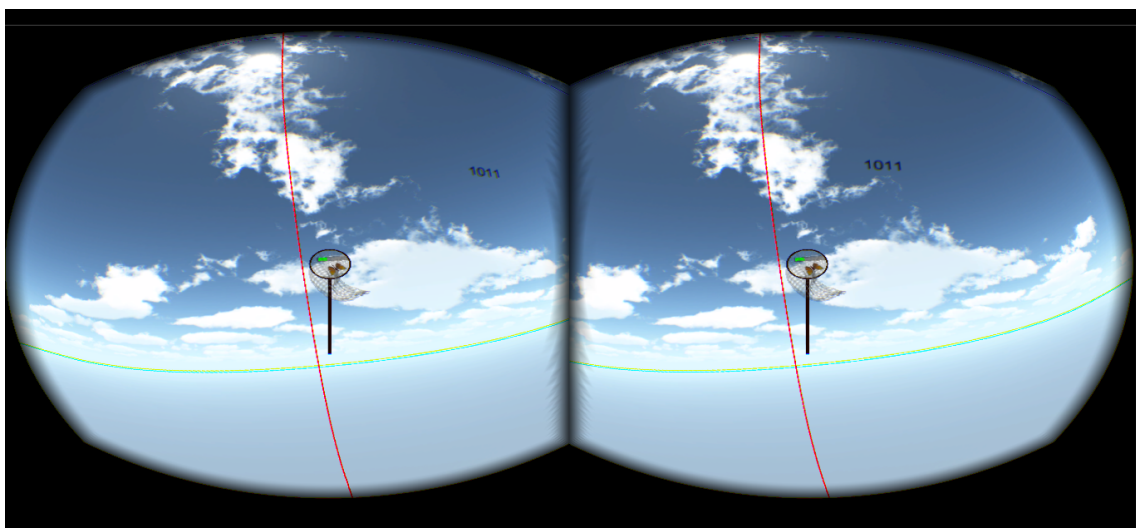
(6.1)

Bodovi za praćenje dodjeljuju se frekvencijom 60Hz, razlika kuta između pogleda i leptira (α) mora biti manja od 5 stupnjeva, te vrijeme za dolazak pogledom na leptira ne smije isteći. U protivnom se dobije 0 bodova. Putanja za praćenje se određuje nasumično tako da se generiraju četiri točke, te se zatim nad tim točkama konstruira Bezierova krivulja. Bezierova krivulja je parametarska krivulja sa formulom:

$$\mathbf{B}(t) = (1 - t)^3\mathbf{P}_0 + 3(1 - t)^2t\mathbf{P}_1 + 3(1 - t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3, \quad 0 < t < 1 \quad (6.2)$$

Gdje su \mathbf{P}_i točke, a t parametar koji se računa na temelju broja proteklih slika od početka praćenja i parametra brzine. Na početku praćenja je $t = 0$ dok je na kraju praćenja $t = 1$.

Bodovi ovdje samo predstavljaju mjeru ukupne preciznosti, te je moguće ispitanika provesti kroz iste postavke igre više puta da se vidi postoji li rast ili pad preciznosti i brzine ovisno o oporavku ili pogoršanju ozljede.



Slika 6.8.: Scena igre

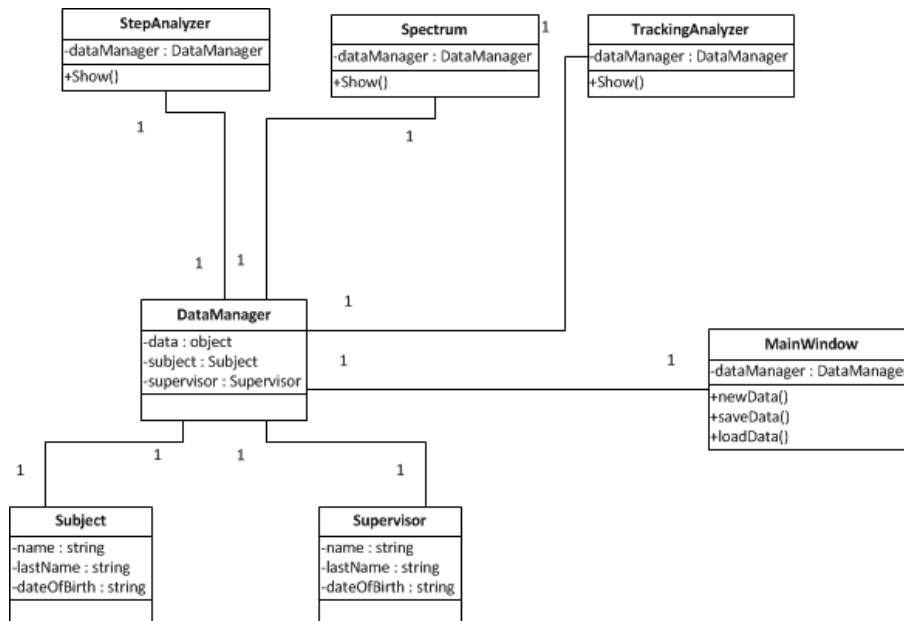
6.2. Aplikacija za analizu

Aplikacija za analizu ostvarena je koristeći razvojno okruženje Visual Studio 2012 te programski jezik C#. Također su korištene biblioteke OxyPlot (za iscrtavanje grafova) i Math.NET (za napredne matematičke operacije)[11, 12]. Ostvarena je i baza podataka korištenjem sustava SQLite, a rezultati analize se spremaju kao datoteke u .xml formatu.

6.2.1. Oblikovanje aplikacije

Aplikacija za analizu podataka ima jednostavan dizajn. Analiza svakog pojedinog testa vrši se u posebnom prozoru, te svaki prozor predstavlja svoj zasebni razred. Svaki taj razred ima referencu na statički razred `DataManager`, koji služi za pohranu podataka i informacija. Razred `DataManager` također sadrži reference na razrede koji predstavljaju ispitnika, odnosno supervizora, zbog potrebe za kasnijim spremanjem tih podataka u bazu.

Glavni prozor sadrži i izbornike pomoću kojih je moguće obavljati rad s

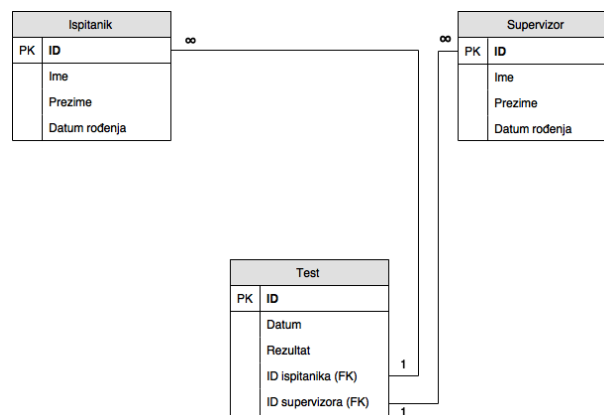


Slika 6.9.: Dijagram razreda aplikacije za analizu

datotekama i bazom podataka. Rad sa bazom podataka se, slično kao i rad s grafovima te matematičkim operacijama, obavlja pozivima funkcija vanjske biblioteke.

6.2.2. Baza podataka

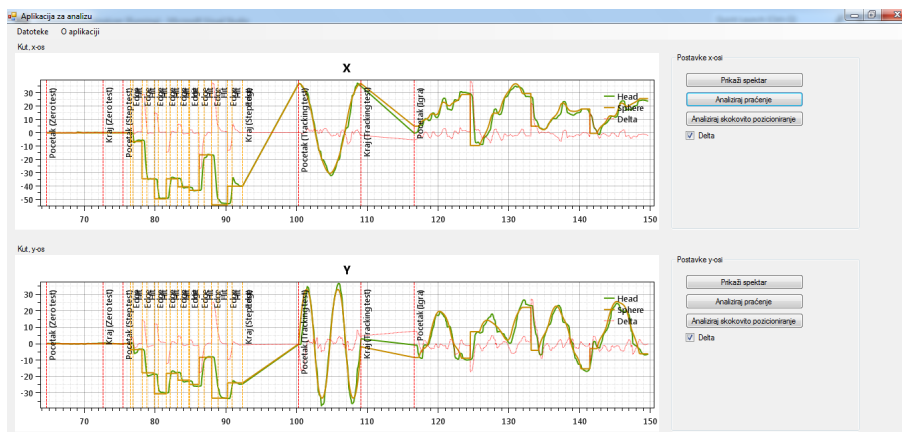
Baza podataka služi kako bi se pohranili rezultati prijašnjih analiza. Cjelokupni rezultati se ne spremaju u bazu, već u .xml datoteku, čiji put se zatim zapisuje u bazu. To može dovesti do problema ako se npr. datoteka obriše ili premjesti. Dijagram baze podataka prikazan je na slici 6.10.



Slika 6.10.: Dijagram baze podataka

6.2.3. Rad aplikacije

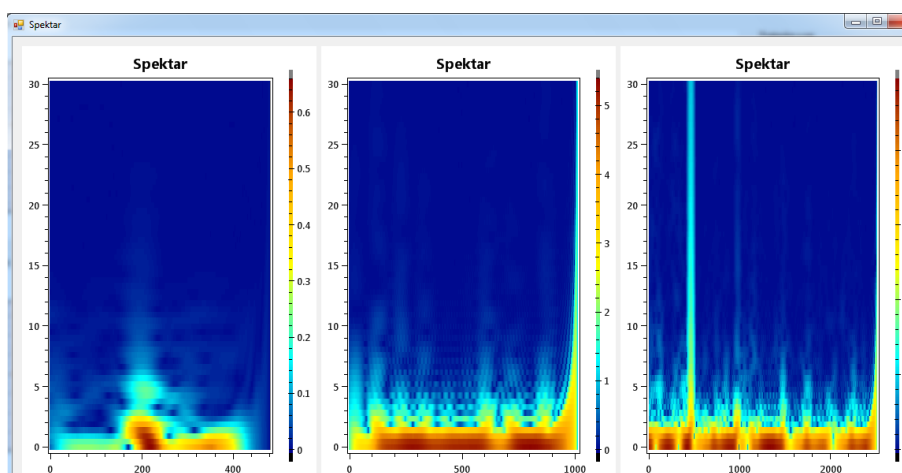
Glavni dio aplikacije je prozor za prikaz grafova putanje objekta i pogleda. Grafovi se generiraju prilikom učitavanja datoteke nastale kao izlaz iz Unity aplikacije za provođenje testova. Izgled grafičkog sučelja može se vidjeti na slici 6.11.



Slika 6.11.: Grafičko sučelje aplikacije za analizu

Dodatak OxyPlot sadrži mogućnost da se pojedini graf smanji, poveća, te pomiče u lijevo, u desno, gore ili dole. To omogućuje da centriramo pogled na određeni raspon koji želimo detaljnije promotriti. U prozoru su prikazana dva grafa, po jedan za x- i y-os. Na svakom grafu prikazane su putanje objekta te pogleda, a moguće je uključiti prikaz razlike između dvije putanje (*Delta*).

Ovisno o testovima koji su provedeni u fazi testiranja, moguće je izvlačiti značajke iz putanje glave. Značajke izvučene iz pojedinog testa opisane su u poglavlju 4.



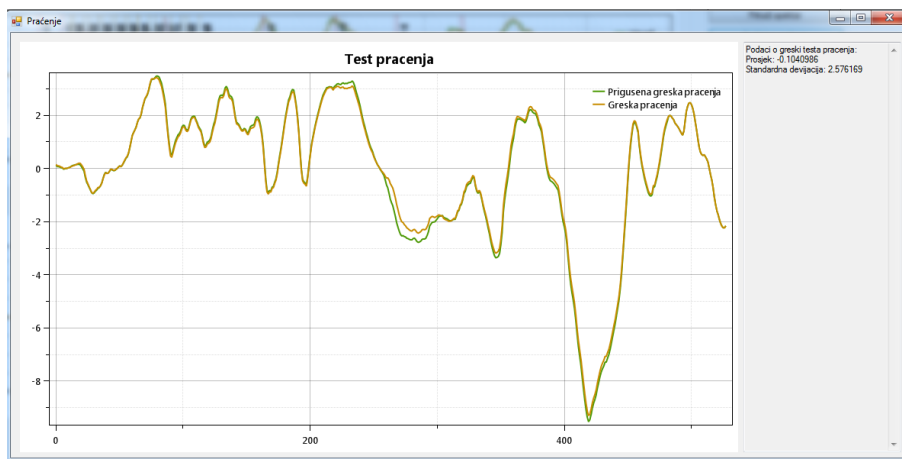
Slika 6.12.: Prikaz spektra

Osim detaljnog prikaza grafova putanje objekta i glave, aplikacija pruža još mogućnosti za analizu testova. Aplikacija je dizajnirana na način da se značajke zapišu u .xml datoteku tek nakon otvaranja prozora za određenu analizu. Prva takva mogućnost je prikaz spektra. Spektar služi za prikaz frekvencijskog sadržaja određenog signala, u ovom slučaju kretanja glave. Budući da je postojala prevelika razlika u amplitudi nižih i viših frekvencija, spektar je prikazan koristeći logaritamsku skalu. Primjer jednog dobivenog spektra može se vidjeti na slici 6.12. Prikazan je po jedan spektar za svaki od tri testa. Crvena boja predstavlja veću amplitudu, dok plava boja predstavlja manju amplitudu. Mogu se uočiti porasti amplituda viših frekvencija u trenutcima kada se glava ispitanika brže kreće, npr. pri pojavljivanju kugle u testu skokovitog pozicioniranja, prikazanom u sredini.

Kod testa praćenja, analiza se svodi na analizu greške praćenja. Ako je test praćenja proveden, onda je u aplikaciji za analizu moguće otvoriti prozor za test praćenja u kojem se računa prosječna greška, standardna devijacija, te se prikazuje graf greške praćenja u slučaju kada oduzmemo trzaje dobivene u testu statičkog pozicioniranja. Trzaji se oduzimaju na način da se izračuna spektar rezultata oba testa, te se spektri oduzmu. Promjene su uobičajeno minimalne, jer trzaji kod testa statičkog pozicioniranja uglavnom sadrže više frekvencije, a kako je već bilo navedeno kod opisa spektra, više frekvencije imaju znatno manje amplitude od nižih. Međutim razlika se još uvijek može primjetiti, što se može vidjeti na slici 6.13.

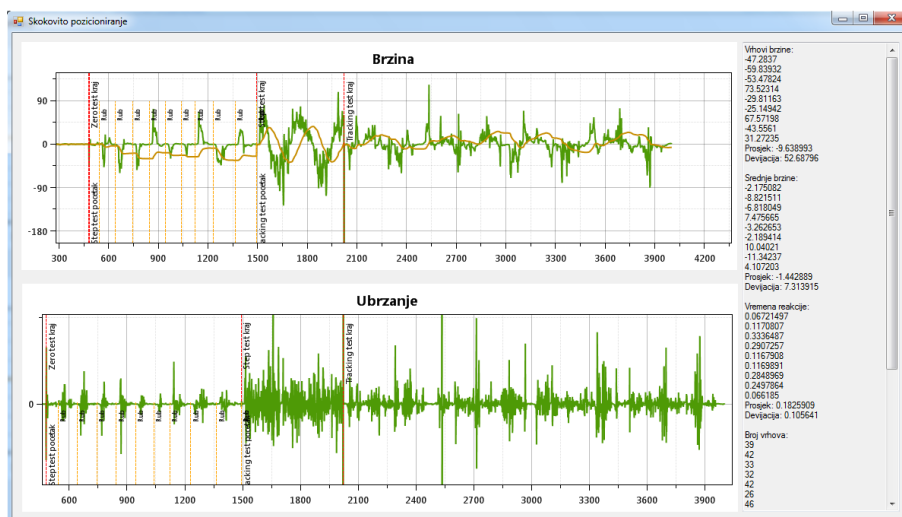
Pri implementaciji diskretne Fourierove transformacije za izračun vrijednosti spektra, korištena je duljina vektora od 256 uzoraka, da bi se dobio pouzdan frekvencijski prikaz uz očuvanje tranzijentnih vrijednosti, te je također korišten Hammingov prozor za uklanjanje neželjenih pojava nastalih diskontinuiranim vrijednostima na početku i kraju vektora.

Zadnja preostala analiza je analiza testa skokovitog pozicioniranja. Iz



Slika 6.13.: Analiza greške testa praćenja

signala se izvlače značajke navedene u poglavlju 4. Budući da test skokovitog pozicioniranja obuhvaća veći broj pokreta glave (zbog pojave većeg broja objekata na ekranu), potrebno je izračunati svaku značajku za svaku kretnju. Za svaku značajku je također izračunata srednja vrijednost i standardna devijacija.



Slika 6.14.: Analiza testa skokovitog pozicioniranja

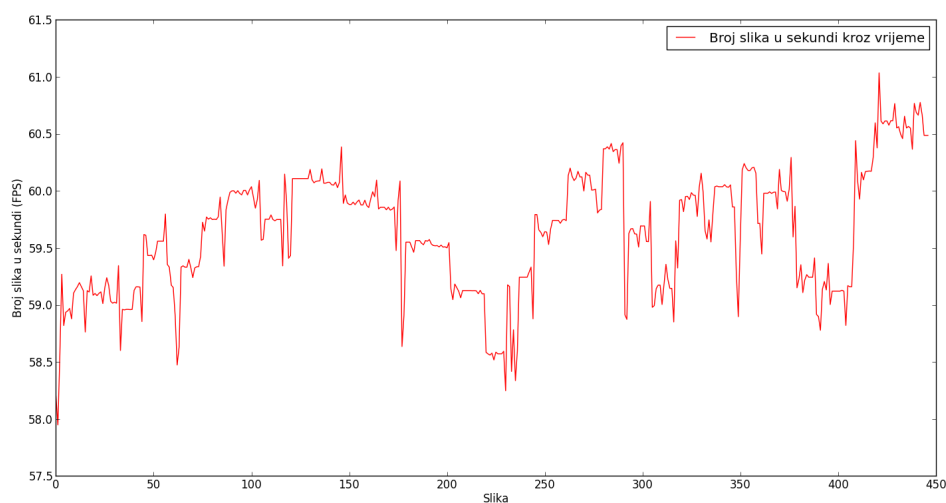
Kao što se može vidjeti, iz testa skokovitog pozicioniranja moguće je izvući najveći broj značajki. U prozoru za analizu tog testa također je os-

tvaren prikaz grafova brzine i akceleracije (prve i druge derivacije). Kao što se može vidjeti na slici 6.14, moguće je uočiti vremenske periode u kojima dolazi do kretanja glave prema objektu.

Aplikacija u konačnici omogućuje spremanje rezultata analize u bazu te u obliku .xml datoteke, kao i spremanje grafova u .png formatu. Bitno je napomenuti da se sve značajke računaju odvojeno za x i y-os, što omogućuje lokaliziranje problema s kretanjem na fleksiju ili rotaciju vrata.

7. Rezultati

Uređaj Oculus Rift pokazao se kao pouzdan pri mjerenju pokretljivosti glave. Kašnjenje uređaja je neznatno, a rezolucija dovoljna da se objekti nesmetano i jasno prikazuju na ekranu. Aplikacija za provođenje testova se bez problema zadržava na približno 60 slika u sekundi, dok jedini pad u performansama nastaje pri prelasku iz jednog testa u drugi, no to je pak vezano uz način na koji okruženje Unity ostvaruje prijelaz između dvije scene. Unutar jednog testa, broj slika u sekundi je gotovo konstantan. Broj slika u sekundi unutar izvršavanja jednog testa može se vidjeti na slici 7.1, gdje je broj slika u sekundi izračunat u prozoru od jedne sekunde.



Slika 7.1.: Broj slika u sekundi

Kod aplikacije za analizu, dobivene vrijednosti svake od značajki su približno jednake očekivanim vrijednostima. Očekivane vrijednosti utemeljene su na rezultatima dobivenima unutar prijašnjih radova koji su se bavili ovom tematikom, što je znak da su postupci izračuna danih značajki dobro implementirani, te su time sami rezultati pogodni za eventualnu daljnju analizu i objektivizaciju kretanja glave i vrata.

8. Zaključak

Uređaji za prikaz virtualne stvarnosti koji su lako dostupni, poput uređaja Oculus Rift, omogućavaju razvoj i primjene takve tehnologije za mnoga područja koja imaju praktičnu uporabu u svakodnevnom životu. Jedno takvo područje je i mjerenje pokretljivosti glave i vrata radi utvrđivanja postojanja boli ili ozljede vrata i vratne kralježnice.

Kako se može vidjeti u ovom radu, implementacija postojećih metoda mjerenja pokretljivosti nije pretjerano zahtjevna, zbog velike razine podrške koja danas postoji za rad s trodimenzionalnim sustavima i igrama, te podrške koja postoji za integraciju takvih sustava s perifernim uređajima, bilo upravljačkim uređajima ili uređajima za prikaz. Mjerenja dobivena implementiranim testovima su točna, te rezultati dobiveni analizom uvelike odgovaraju rezultatima dobivenima sličnim sustavima u prošlosti.

U budućnosti postoji mnogo mjesta unaprijeđenje sustava za mjerenje pokretljivosti vrata i glave. Dio napretka temelji se na daljnjem razvoju tehnologija za prikaz virtualne stvarnosti, te radu na nekim aspektima kao što su smanjenje latencije, veća rezolucija i bolje preslikavanje virtualne scene na prikaz uređaja, iako su i u ovom trenutku oni već na visokoj razini.

Literatura

- [1] Sarig Bahat H, et al., *Interactive cervical motion kinematics: Sensitivity, specificity and clinically significant values for identifying kinematic impairments in patients with chronic neck pain*, Manual Therapy (2014), <http://dx.doi.org/10.1016/j.math.2014.10.002>
- [2] Ćosić K, Slamić M, Popović S, Rac Z, *VR Based Adaptive Stimulation in Estimation of Whiplash Injuries*, Proceedings of the Vision, Modeling, and Visualization Conference 2002, Erlangen, Germany (2002), 35-42
- [3] Kramer M et al., *Reliability of a new virtual reality test to measure cervicocephalic kinaesthesia*, J Electromyogr Kinesiol (2008), doi:10.1016/j.jelekin.2008.05.005
- [4] Kristjansson E, Hardardottir L, Asmundardottir M, Gudmundson K, *A new clinical test for cervicocephalic kinesthetic sensibility: "The Fly."*, Arch Phys Med Rehabil (2004), 85:490-5
- [5] Specifications, 2015, *The All New Oculus Rift Development Kit 2 (DK2) Virtual Reality Headset | Oculus Rift - Virtual Reality Headset for 3D Gaming*, <https://www.oculus.com/dk2/>, 14.05.2015.
- [6] Sarig Bahat H, Weiss PL, Laufer Y, *The effect of neck pain on cervical kinematics, as assessed in a virtual environment*, Arch Phys Med Rehabil, 2010, 91:1884-90
- [7] Sjölander P et al., *Sensorimotor disturbances in chronic neck pain - Range of motion, peak velocity, smoothness of move-*

ment, and repositioning acuity, Manual Therapy, 2008, 13:122-131
<http://dx.doi.org/10.1016/j.math.2006.10.002>

[8] Unity, 2015, *Unity - Game engine, tools and multiplatform*,
<http://unity3d.com/unity>, 29.05.2015.

[9] Unity Scripting Reference, 2015, *Unity - Scripting API*,
<http://docs.unity3d.com/ScriptReference/>, 29.05.2015.

[10] Ramirez R W, *The FFT: Fundamentals and Concepts*, Prentice Hall
PTR, 1984

[11] OxyPlot, 2015, *OxyPlot*, <http://oxyplot.org/>, 16.06.2015.

[12] Math.NET, 2015, *Math.NET*, <http://www.mathdotnet.com/>,
16.06.2015.

Sustav za mjerenje motorike glave i vrata primjenom uređaja Oculus Rift

Sažetak

Objektivizacija pokretljivosti vrata i glave je problem koji se javlja ponajviše kod određivanja postojanja boli ili ozljeda vrata i vratne kralježnice. U ovom radu prikazana je ideja iza sustava koji bi služio za mjerenje pokretljivosti glave i vrata uz korištenje virtualne stvarnosti preko uređaja Oculus Rift. Također su prikazani načini analize podataka dobivenih mjerenjem.

U sklopu rada, ostvarena je i programska implementacija takvog sustava, te aplikacije za analizu. Unutar programske podrške implementirani su testovi za mjerenje pokretljivosti, kao i metode za dohvaćanje bitnih značajki iz podataka dobivenih mjerenjem. U radu je prikazana struktura programske podrške, kao i njena funkcionalnost.

Ključne riječi: Virtualna stvarnost, Oculus Rift, Unity, pokretljivost glave i vrata, trzajne ozljede

System for measurement of head and neck motorics based on Oculus Rift device

Abstract

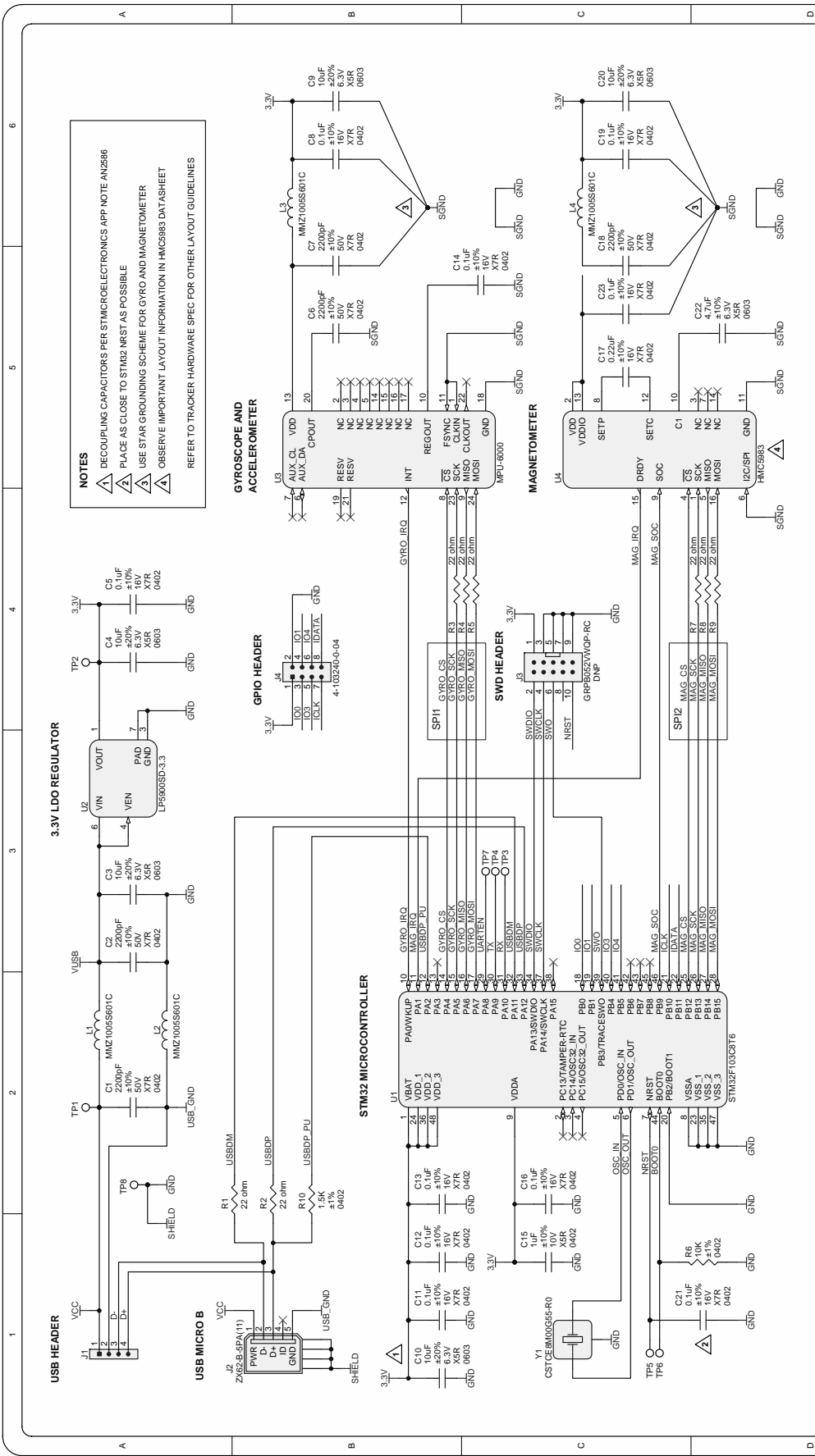
Objectivization of head and neck mobility is a problem that appears when determining the existence of pain or injuries in the neck or cervical spine. In this paper, an idea is presented for a system that could be used for measuring the motorics of the head and neck by using virtual reality with the Oculus Rift device. The methods of analyzing the given measurement data are also presented.

As a part of this paper, a software implementation of a such system is also created, together with an analysis application. In this software, tests for measuring motorics have been implemented, along with the methods for getting important factors from the measurement data. In the paper, the design of the software has been shown, along with its functionality.

Keywords: Virtual reality, Oculus Rift, Unity, head and neck motorics, whiplash injuries

Dodatak A

Shema Oculus Rift uređaja



DESCRIPTION		OCULUS TRACKER		ENGINEERING	DRAWING		APPROVED	DATE
NOTES		FINAL RELEASE		NIRAV PATEL	L. BAINBRIDGE			29 JAN 2013
		SIZE	TYPE	DOCUMENT NUMBER	APPROVED DATE		VERSION	SHEET
		B	SCH	N/A			2	1 of 1

