

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6523

**USPOREDBA ISKUSTVENE KVALITETE PRILIKOM
KORIŠTENJA DETEKCIJA ŠAKE NASPRAM KORIŠTENJA
KONTROLERA U VIRTUALNOJ STVARNOSTI**

Bruno Bernik

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6523

**USPOREDBA ISKUSTVENE KVALITETE PRILIKOM
KORIŠTENJA DETEKCIJA ŠAKE NASPRAM KORIŠTENJA
KONTROLERA U VIRTUALNOJ STVARNOSTI**

Bruno Bernik

Zagreb, lipanj 2020.

ZAVRŠNI ZADATAK br. 6523

Pristupnik: **Bruno Bernik (0036510347)**

Studij: Računarstvo

Modul: Računarska znanost

Mentor: doc. dr. sc. Mirko Sužnjević

Zadatak: **Usporedba iskustvene kvalitete prilikom korištenja detekcija šake naspram korištenja kontrolera u virtualnoj stvarnosti**

Opis zadatka:

Tržište virtualne stvarnosti raste velikom brzinom posljednjih godina te se na tržištu pojavio i veliki broj različitih sustava za praćenje kontrola korisnika. Jedna od najnovijih metoda je praćenje pokreta ruku korisnika putem posebnih kamera ugrađenih u naočale za virtualnu stvarnost. Jedan od sustava koji podržavaju taj način interakcije u virtualnoj stvarnosti je i Oculus Quest. Vaš zadatak je proučiti sustav za virtualnu stvarnost Oculus Quest i za njega razviti aplikaciju za testiranje tipičnog podigni i postavi zadatka (engl. pick and place task). Aplikacija treba podržavati tri veličine oblika za podizanje i postavljanje kao i mjerenje vremena za odrađivanje zadatka. Dodatno, aplikacija treba podržavati interakciju korisnika putem kontrolera kao i putem praćenja pokreta ruku kroz kamere. Koristeći razvijenu aplikaciju potrebno je napraviti subjektivnu studiju iskustvene kvalitete te statistički obraditi dobivene rezultate. Svu potrebnu literaturu i uvjete za rad osigurat će Vam Zavod za telekomunikacije.

Rok za predaju rada: 12. lipnja 2020.

Sadržaj

Uvod.....	1
1. Korištenje Oculus Questa	3
1.1. Potrebna oprema.....	3
1.2. Upute za uporabu.....	3
2. Opis Oculus Questa	6
2.1. Sustav za praćenje ruku	11
3. Eksperiment	14
4. Rezultati eksperimenta	16
4.1. Vrijeme potrebno za provođenje eksperimenta.....	16
4.2. Utjecaj osvjetljenja	18
4.3. Zaklonjenost kamere.....	19
4.4. Prednosti ruku i kontrolera.....	20
4.5. Nedostatci ruku i kontrolera.....	20
Zaključak	22
Literatura	23
Sažetak	25
Summary	26
Privitak	27
HandTrackingGrabber.cs	27
CollisionMP2.cs.....	28
CollisionSP2.cs	28
CollisionVP2.cs	29
StartTimeV.cs	30
StartTimeS.cs.....	30
StartTimeM.cs	31

Timer.cs.....	31
---------------	----

Uvod

Virtualnom stvarnošću mogu se smatrati prostor i događaji koji su virtualni, odnosno nisu stvarni. Pojam virtualna stvarnost najčešće se veže za simulacije koje su nastale pomoću računala ili drugih sličnih naprava [1]. U virtualnoj stvarnosti korisnik može čuti, vidjeti, a u nekim slučajevima čak i osjetiti ili omirisati virtualne simulacije. U idealnom slučaju, kada bi čovjek bio u virtualnoj stvarnosti, ta stvarnost bi trebali biti toliko stvarna da korisnik ne bi mogao sam raspoznati radi li se o virtualnoj stvarnosti ili realnosti. Također, u idealnoj virtualnoj stvarnosti, korisnik bi se trebao moći nesmetano kretati u tom virtualnom prostoru, uz ograničenja tog virtualnog svijeta. Također svaki podražaj, doživljaj ili događaj koji se dogodi u virtualnoj stvarnosti trebao bi na korisnika imati jednak utjecaj kao kada bi se isto to dogodilo u realnosti, a kada korisnik izađe iz virtualne stvarnosti i vrati se u realnost, na njemu ne bi trebao ostati nikakav utjecaj toga virtualnog svijeta u kojem je bio.

Virtualna stvarnost današnjice je daleko od upravo opisane idealne virtualne stvarnosti, no ipak postoji. Kroz povijest se pokušalo približiti virtualnoj stvarnosti na različite načine no najstariji uređaj koji je uvelike sličan virtualnim uređajima kakvi danas postoje konstruiran je 1956. godine [2]. Od tada do danas razvili su se mnogi slični uređaji i načini za postizanje virtualne stvarnosti. Virtualna stvarnost se postizala preko niza senzora koji bi pratili pokrete tijela (kao npr. senzori položaja, senzori orijentacije, senzori pokreta, senzori sile itd.), te bi se zatim podaci dobiveni od tih senzora obradili (najčešće na računalu) i zatim bi se prikazali korisniku preko zaslona, zvučnika, slušalica, simulatora mirisa i sl. Virtualni uređaj koji će se koristiti u ovom radu podatke o položaju korisnika dobiva preko 4 kamere koje se nalaze na naočalama za virtualnu stvarnost i preko 2 kontrolera koje korisnik koristi.

Cilj rada je opisati uređaj za virtualnu stvarnost Oculus Quest (Sl. 1) i usporediti korištenje ruku i korištenje kontrolera koristeći navedeni uređaj za zadatak tipa „uzmi i postavi“ (engl. „*Pick and place*“). Razlog provođenja usporedbe je upravo Questovo svojstvo pružanja mogućnosti korištenja ruku u virtualnoj stvarnosti jer je Quest prvi Oculus uređaj koji to omogućuje.

Kako i korištenje ruku i korištenje kontrolera ima svoje prednosti, cilj rada je dobiti podatke na temelju kojih bi bilo moguće usporediti navedena dva načina korištenja Questa i na temelju kojih bi bilo moguće odrediti koje su prednosti jednog ili drugog načina. U radu će se provesti i opisati eksperiment kojim će se podaci za usporedbu korištenja ruku i korištenju kontrolera s Oculus Questom za jednostavan zadatak tipa „uzmi i postavi“ (engl. „*Pick and place*“). Osim toga bit će riječ i o Oculus Questu, načinu korištenja Oculus Questa i o nekim svojstvima koje sustav za praćenje ruku pruža.



Sl. 1 Izgled Oculus Questa [3]

1. Korištenje Oculus Questa

U poglavlju će se opisati koje su specifikacije potrebne kako bi se koristio Oculus Quest i što je potrebno napraviti za njegovo korištenje. Nakon toga opisat će se postupci pri razvoju ove aplikacije.

1.1. Potrebna oprema

1. uređaj Oculus Quest, 2. minimalno 2.0+ GHz procesor¹, 3. minimalno 2 GB memorije s nasumičnim pristupom (engl. *Random Access Memory*, skraćeno *RAM*)², 4. Windows 10 (64 bitnu verziju)/ macOS Sierra 10.10+³, 5. Unity razvojno okruženje⁴ (za Unity razvojnu okolinu potrebna je verzija 2017.4.16 ili novija, a preporučene verzije su 2017.4 LTS, 2018.4 LTS, 2019.3) [4]

1.2. Upute za uporabu

U ovome potpoglavlju opisat će se kako koristiti Oculus Quest. Ako bi htjeli koristiti Quest za igranje igara i zanimaciju dovoljne su upute „Korištenje Questa u privatne svrhe“. Ako bi htjeli razvijati aplikacije za Quest tada je potrebno napraviti sve korake navedene u odjeljku „Korištenje Questa u razvojne svrhe“. Na kraju dan je i kratak opis postupaka koji su bili potrebni u razvoju ove aplikacije, no oni se za svakog pojedinca mogu razlikovati.

Korištenje Questa u privatne svrhe

Za korištenje Oculus Questa potrebno je instalirati „Oculus“ aplikaciju. Aplikacija je dostupna na Android 6.0+ i Apple iOS 10+ uređajima i računalima. Za korištenje

¹potrebno je jedino ako se Quest želi koristiti u razvojne svrhe, što će biti objašnjeno u potpoglavlju 1.2.

²potrebno je jedino ako se Quest želi koristiti u razvojne svrhe, što će biti objašnjeno u potpoglavlju 1.2.

³potrebno je jedino ako se Quest želi koristiti u razvojne svrhe, što će biti objašnjeno u potpoglavlju 1.2.

⁴potrebno je jedino ako se Quest želi koristiti u razvojne svrhe, što će biti objašnjeno u potpoglavlju 1.2.

aplikacije, potrebno je napraviti Oculus račun. Račun je moguće napraviti putem Gmail ili Facebook računa.

Nakon izrade računa i instalacije aplikacije, potrebno je povezati Oculus Quest s aplikacijom. Kroz potrebne korake za povezivanje vodi nas sama aplikacija. Nakon toga za korištenje Questa više nije potrebna aplikacija te se Quest može koristiti sam po sebi.

Korištenje Questa u razvojne svrhe

Opisan postupak za korištenje Questa u razvojne svrhe bit će vezan za Unity razvojnu okolinu. Prvo je potrebno napraviti sve što je navedeno u prethodnom odjeljku („Korištenje Questa u privatne svrhe“).

Zatim je u mobilnoj aplikaciji potrebno omogućiti korištenje Questa u razvojne svrhe što omogućavamo klikom na „Settings“ > „More Settings“ > „Developer Mode“. Nakon toga potrebno je napraviti neke promjene u Unityu.

U Unity razvojnom okruženju potrebno je unijeti „Oculus Integration“ sredstvo (engl. *asset*) u projekt. To je Oculusovo besplatno sredstvo (engl. *asset*) koje omogućuje korištenje Questa u razvojne svrhe, a dostupno je na „Asset Storeu“. „Oculus Integration“ sadrži i neke predloške (engl. *prefab*) i scene koje je moguće odmah pokrenuti kako bi se lakše i brže upoznali s mogućnostima koje „Oculus Integration“ nudi.

Ako se želi napraviti potpuno nova scena, potrebno je napraviti nekoliko promjena u postavkama. To se može učiniti tako da se u traci s alatima klikne na „File“ > „Build Settings“ > „Player Settings“ > „Android Settings“. U djelu „Android Settings“ potrebno je 1.otići na XR Settings i u listi koja se zove „Virtual Reality SDK“, kliknuti znak plus i onda kliknuti na „Oculus“, 2.otići na „Other Settings“ i u listi „Graphics APIs“ kliknuti na „Vulkan“ i zatim na znak minus zbog toga što XR ne podržava Vulkan sučelje, 3.otići na „Minimum API Level“ i za vrijednost postaviti 28.

Zatim je potrebno promijeniti platformu u android. To se može učiniti tako da se u traci s alatima klikne na „File“ > „Build Settings“. Tamo je potrebno kliknuti na Android ikonu i u izborniku „Texture Compression“ odabrati „ASTC“ i zatim „Switch Platform“.

Također potrebno je kreirati identifikacijski broj aplikacije, a to je broj koji povezuje Oculus uređaj s aplikacijom koju se razvija. Taj broj dobiva se odlaskom na stranicu <https://dashboard.oculus.com/>. Da bi se pristupilo stranici korisnik mora biti ulogiran sa

svojim Oculus računom. Na stranici se nalazi gumb „Create New App“. Klikom na njega pojavljuje se izbornik u koji je potrebno unijeti ime i platformu za koju je aplikacija namijenjena. Nakon toga na stranici će se prikazati „App ID“ i to je identifikacijski broj aplikacije koji je potreban. Sada se može vratiti u Unity i otići na alatnu traku gdje je potrebno kliknuti na „Oculus“ > „Platform“ > „Edit Settings“. Prikazat će se „Inspector“ prozor u kojem se nalaze 2 prazna polja („Oculus Rift“ i „Oculus GO/Quest or Gear VR“) i u njih je potrebno unijeti identifikacijski broj aplikacije. Još uvijek u istom „Inspector“ prozoru potrebno je odznačiti kvačicu „Use Standalone Platform“.

Kada je to odrađeno, može se spojiti Quest s računalom i otići na „File“ > „Build Settings“ > i u kliznom izborniku „Run Device“ odabrati Oculus Quest uređaj. Zadnji korak je kliknuti na „Build And Run“ i time se scena učitava na virtualni uređaj.

Postupci navedeni u prethodnom odjeljku „Korištenje Questa u razvojne svrhe“ su nužni kako bi se izrađena aplikacija mogla koristiti na Oculus Questu, dok se postupci za izradu scene u razvojnom okruženju za aplikaciju mogu razlikovati. Prvi korak je stvaranje novog projekta u Unity razvojnom okruženju.

Sljedeći važan korak bio je dodavanje Oculusovog paketa „Oculus Integration“ koji omogućuje komunikaciju *headseta* s kontrolerima i sadrži razne predloške (engl. *prefab*) i skripte koje omogućuju osnovne radnje u virtualnoj stvarnosti kao što su hodanje i mogućnost da korisnik vidi virtualni svijet oko sebe. Samim unosom Oculusovih predložaka (engl. *prefab*) u projekt, oni nemaju nikakvu funkciju. Na primjer unosom *OVRHandPrefab*, u sceni se pojavljuje lijeva ili desna ruka no s njom se ne može utjecati na ostale stvari u sceni. Kako bi ju mogli koristiti potrebno je napisati skriptu koja bi joj omogućila neku funkcionalnost.

Sve skripte koje su korištene u ovom projektu su napisane u C# programskom jeziku. Skripte koje sam napisao bit će u potpunosti dodane u Pravitku, a vezane su uz interakciju kontrolera i ruku s određenim predmetima u aplikaciji (kao npr. kako da aplikacija prepozna je li korisnik uhvatio neki predmet ili nije), uz interakciju između predmeta i predmeta (kao npr. kako aplikacija prepoznaje je li pravi predmet na pravom mjestu) i uz prikaz proteklog vremena (kao npr. kada aplikacija prestaje brojati vrijeme, a kada počinje).

2. Opis Oculus Questa

Uređaj Oculus Quest sastoji se od *headseta* i 2 kontrolera. Pošto je *headset* bežičan, to omogućuje korištenje Questa čak i vani, odnosno izvan prostorije no takav način korištenja nije preporučljiv i moguće je da će doći do oštećenja kamere na *headsetu*. Unatoč tome, Quest je svejedno prenosiv i može se prenositi iz prostorije u prostoriju. Sva oprema potrebna za korištenje Questa može se staviti u kutiju dimenzija 35.5 x 22 x 12.5 cm.

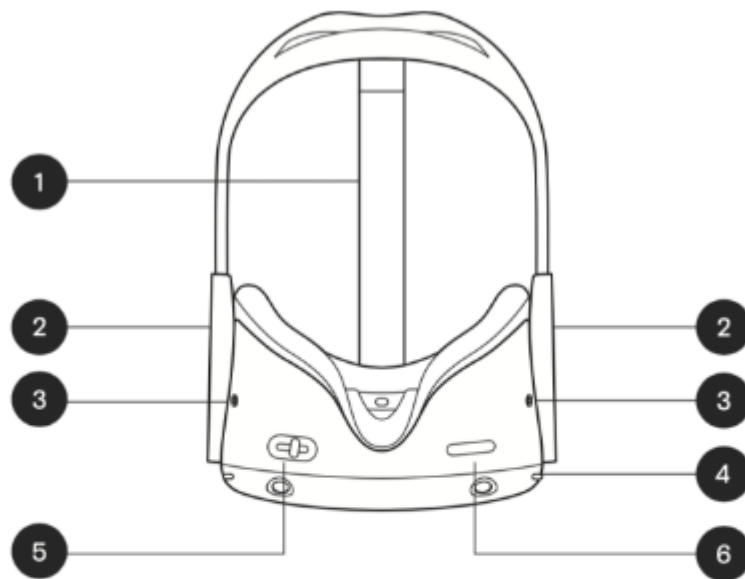
Bežičnost Oculus Questa također omogućuje nesmetano kretanje u prostoru za razliku od nekih dosadašnjih VR uređaja koji moraju biti povezani žicom za računalo. Problem kod žičnih uređaja je bio što je maksimalan radijus korisnikovog kretanja bio određen duljinom žice. Također, korisnik je morao voditi računa i o položaju žice u smislu da se korisnik nije mogao okretati više puta uzastopno oko svoje osi u istom smjeru jer bi se žica zaplela oko korisnika (i time bi se smanjio radijus u kojem se korisnik može kretati ili bi čak moglo doći do iščupavanja žice sa uređaja). Pažnja na žicu bi korisniku mogao biti velik problem zbog toga što u virtualnom svijetu korisnik ne vidi žicu što znači da bi morao misliti na njen položaj u prostoru što pak umanjuje korisnikovo iskustvo jer ga ta žica „povezuje“ sa stvarnošću, a cilj virtualne stvarnosti je upravo da se korisnika odmakne od prave realnosti i pruži mu novo iskustvo.

Osim navedenog, za korištenje Questa, nisu potrebni ni senzori kao što jesu kod mnogih drugih VR uređaja. To je velika prednost Questa jer senzori zauzimaju prostor, potrebno je mnogo utora za utičnice, a i mora se paziti da između senzora i korisnika ne bude nikakvih drugih predmeta. Quest zamjenjuje senzore svojim ugrađenim kamerama, LED lampama i koristeći IMU (inertial measurement unit, naprava koja služi za praćenje ubrzanja i položaja tijela i kutne brzine [5]). Također služi se prilagođenom hardverskom arhitekturom i naprednim algoritmima za računalni vid koji pomoću podataka služe za pamćenje i prepoznavanje prostora i međusobnog položaja objekata u prostoriji [6]. Do potrebnih podataka se dolazi preko 3 vrste senzora koji su ugrađeni u Oculusov hardver. Podaci koji se prikupljaju su: 1. podaci za akceleraciju i brzinu rotacije koju se dobiva od IMU-a u *headsetu* koji pomažu u praćenju pozicije i orijentacije kontrolera i *headseta*, 2. slikovni podaci iz kamere na *headsetu* koji pridonose u stvaranju 3D mape sobe, 3. podaci za prepoznavanje kontrolera preko infracrvenih LED lampi ugrađene u kontrolere.

Kupnjom Oculus Questa, osim *headseta* i 2 kontrolera, dobivaju se i punjač i utor za naočale. Utor za naočale služi kako bi se Quest mogao koristiti i s naočalama. Potrebno je jedino nataknuti utor u stražnju stranu Questa, a ako ga želimo maknuti, dovoljno ga je povući i utor će se odvojiti od *headseta*. Utor čvrsto stoji na Questu i nije potrebno brinuti se o njegovom ispadanju.

Headset

Oculus Quest *headset* sastoji se od (Sl. 2): 1. 4 kamere koje prate položaj kontrolera i korisnika. Kamere se nalaze s vanjske strane Questa, stoga je potrebno biti oprezan da ne bi došlo do njihovih oštećenja. Jedan od načina na koje bi se kamere mogle oštetiti jest da budu izložene velikom utjecaju sunca i baš zbog toga se ne preporučuje korištenje Questa izvan zatvorene prostorije, 2. tipke za paljenje i gašenje Questa, 3. utora za spajanje slušalica i mikrofona na *headset*. Oculus Quest koji ima ugrađene slušalice u *headset* je u razvoju, ali još nije dostupan na tržištu, 4. tipku za pojačavanje i stišavanje zvuka, 5. tipku za podešavanje razmaka leća u *headsetu*. To olakšava upotrebu za ljude s različitim širinama vidnog polja, 6. senzora koji očitava nosi li korisnik *headset* na glavi. U slučaju da ne, *headset* ulazi u stanje mirovanja, 7. utor za USB-type C kabel.

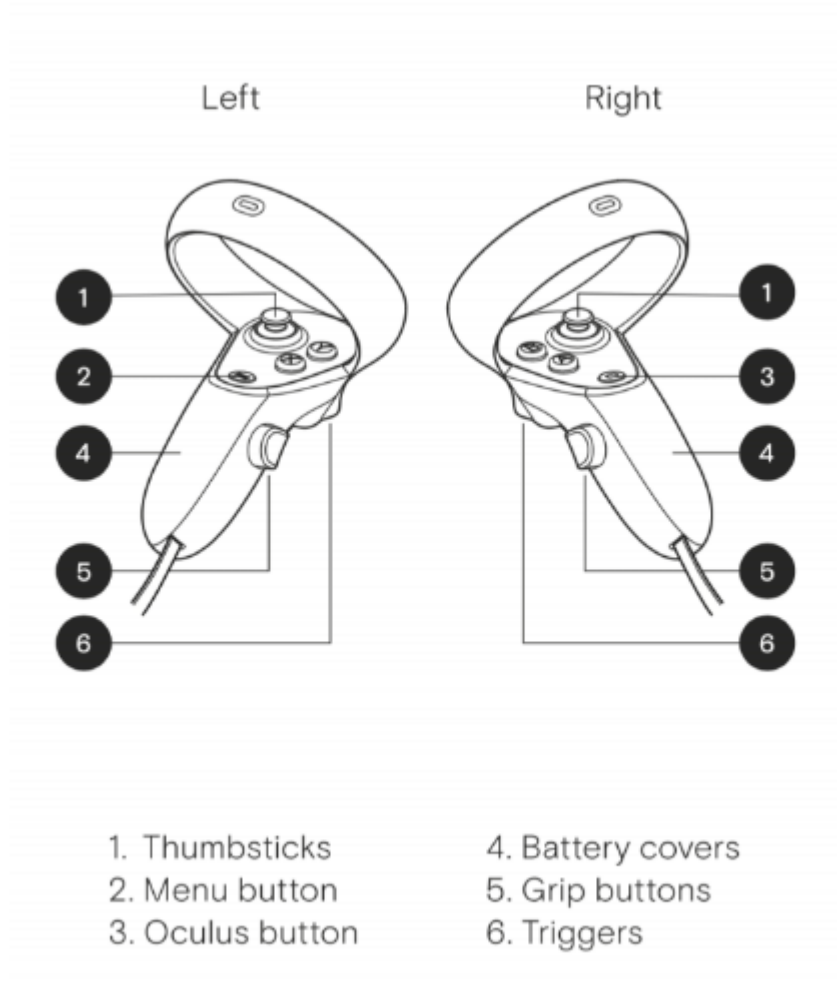


- | | |
|--------------------|------------------|
| 1. Top strap | 4. Power/standby |
| 2. Side straps | 5. Lens spacing |
| 3. Headphone jacks | 6. Volume |

Sl. 2 Prikaz komponenti Oculus Quest *headseta* [7]

Kontroleri

Svaki kontroler ima 6 gumba (Sl. 3), uključujući i jedan klizni. Za napajanje kontrolera potrebna je po jedna AA baterija za svaki kontroler. Svaki kontroler ima i pojas za ruku koji osigurava da kontroler ne bi ispao iz ruke.



Sl. 3 Prikaz gumba na kontrolerima [8]

Granice sigurnosti

Pošto je Quest bežičan, omogućuje kretanje unutar cijele prostorije, no vjerojatno će u prostoriji biti drugih predmeta koji bi mogli zasmetati dok je korisnik u virtualnoj stvarnosti. Kako bi se to izbjeglo Quest omogućuje postavljanje prostornih granica unutar kojih bi se mogao koristiti Oculus Quest. Postavljanje granica je proces označavanja dijelova prostorije unutar kojih će korisnik biti u virtualnoj stvarnosti. Proces je jednostavan, a odvija se tako da se kontrolerom označe granice oko kojih će se željeti kretati. (Sl. 4)



Sl. 4 Označavanje sigurnosne granice [9]

Oblik granice je proizvoljan. Kod postavljanja granica potrebno je nositi *headset* no kroz njega se u tom procesu može vidjeti stvarni prostor oko korisnika.

Za svaku prostoriju moguće je postaviti različite granice. Quest zapamti svaku granicu za svaku prostoriju što znači da je proces postavljanja granice potrebno obaviti samo jednom po prostoriji. Ako se korisniku ne sviđa kako je postavio granice ili je npr. preuredio sobu, moguće je preoblikovati granicu. Granicu je moguće preoblikovati proizvoljan broj puta. Svaki put kada se približimo granici, Quest će nas upozoriti tako da oko cijele granice iscrta virtualni zid i ako bi ga dotakli, postao bi crvene boje što označava izlazak iz sigurnosne granice (Sl. 5).



Sl. 5 Prikaz upozorenja prilikom prelaska sigurnosne granice [10]

Ako korisnik cijelim tijelom izađe izvan granice, Quest će prestati projicirati virtualno okruženje i omogućiti pogled na stvarni prostor oko njega.

2.1. Sustav za praćenje ruku

Kako je praćenje ruku novost koju pruža Oculus Quest, u potpoglavlju će se opisati neke od mogućnosti koje sustav pruža. Quest pomoću svoje četiri kamere prati položaje ruku i zbog toga na rukama nije potrebno nositi nikakvu dodatnu opremu. To omogućava korisniku korištenje ruku u druge svrhe kao što su namještanje *headseta* ili položaja leća u *headsetu* ili stavljanje slušalica i sl.

Proces koji je potreban za rad sustava za praćenje ruku, može se podijeliti u 3 faze [11]. Prva faza je prepoznavanje ruku. To se postiglo neuronskim mrežama i dubokim učenjem u kojem se Questu prikazivao velik broj ruku različitih veličina i izgleda na temelju kojih je Quest naučio prepoznavati ruke. U drugoj fazi, nakon što je preko svoje 4 kamere dobio slike na kojima je prepoznao ruke, Quest iscrtava ključne točke na njima. Te točke će biti kostur na temelju kojih će se u trećoj fazi iscrtati ruke u virtualnom svijetu u kojem je korisnik. U trećoj fazi na temelju ključnih točaka na rukama, pomoću algoritama Quest oko tih točaka iscrtava oblike ruku koje se prikazuju korisniku u virtualnom okruženju. Quest pruža i određene mogućnosti vezane za sustav za praćenje ruku, a neke od njih su: 1.

mijenjanje veličine ruku, 2. interakcija sa stvarima izvan dosega, 3. povećavanje i smanjivanje dosega ruku, 4. različiti načini za interakciju s predmetima. [12]

Veličina ruku

Projekcija ruku u virtualnoj stvarnosti može varirati jer Quest omogućuje mijenjanje veličine ruku. Zadanu (engl. *default*) veličinu ruku moguće je povećati ili umanjiti za određeni faktor kako bi postale iste veličine kao i korisnikove vlastite ruke.

Interakcija sa stvarima izvan dosega

Quest omogućuje interakciju s objektima izvan dosega ruku. To se ostvaruje metodom zrake (engl. *ray*), odnosno iz svake virtualne ruke izlazi jedna zraka. Ako se zraka sudari s određenim objektom, to označava da je predmet u dosegu i da je moguća interakcija s istim.

Povećanje i smanjivanje dosega ruku

Još jedna od mogućnosti je povećanje osjetljivosti (engl. *sensitivity*) pokreta rukama. To znači da je moguće podesiti za koliko će se ruka pomaknuti u virtualnom svijetu u odnosu za koliko se pomaknula u stvarnom svijetu. Na primjer moguće je namjestiti da ako osoba pomakne ruku za 10 stupnjeva izvan virtualnog svijeta da se u virtualnom svijetu ruka pomakne za 100 ili 1 ili isto za 10 ili za bilo koliko stupnjeva u istom smjeru.

Načini za interakciju s predmetima

Postoje 2 načina za interakciju s objektima:

Dodir (engl. *Touch*)

Na ovaj način se interakcija s objektima u virtualnoj stvarnosti postiže kao što se postiže i u stvarnosti. Drugim riječima, potrebno je predmet dodirnuti odnosno približiti se predmetu dok ga se ne dotaknemo rukom i tek je onda moguće činiti s njim što nas je volja.

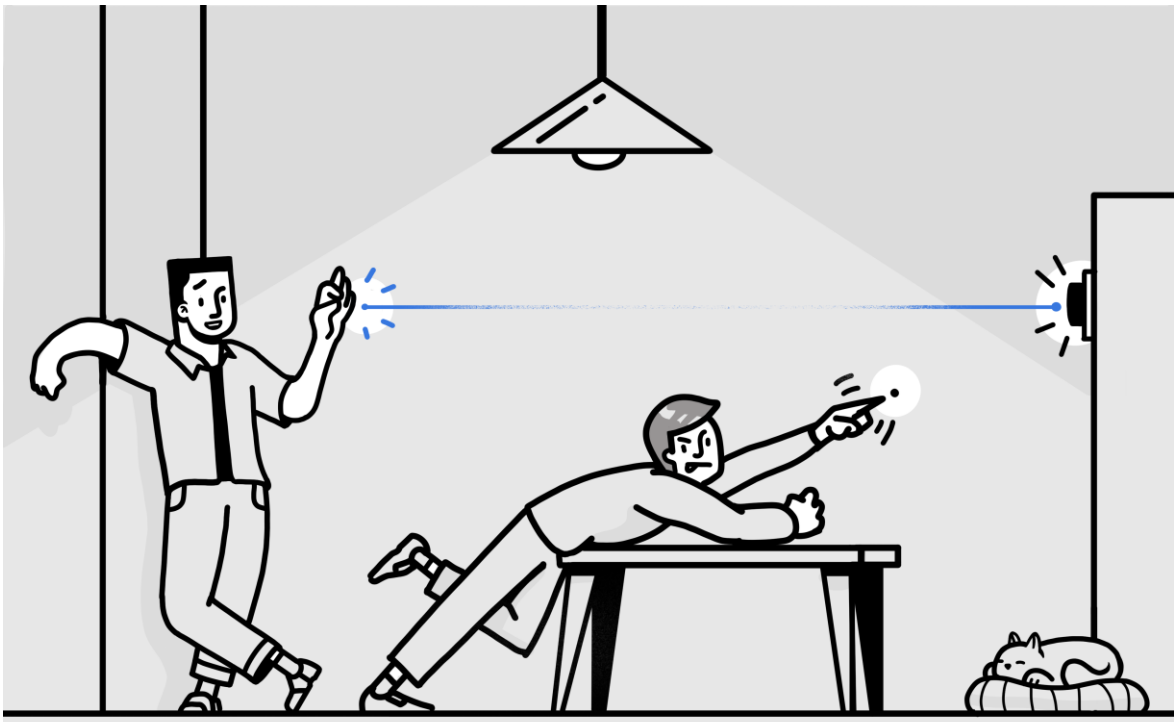
Stisak (engl. *Pinch*)

Stiskom je moguće biti u interakciji i s objektima koji su blizu i s objektima koji su daleko. Stisak se koristi tako da se spoje vrhovi prstiju palca i kažiprsta. (Sl. 6)



Sl. 6 Prikaz korištenja stiska [13]

Stisak se može koristiti na dva načina. Prvi način zahtjeva od korisnika da se približi do objekta dok ne nastane kolizija između ruke i objekta te se zatim spoji palac i kažiprst i time je omogućena interakcija s objektom. Drugi način je pomoću zrake (engl. *ray*) iz ruku tako da se zraka poklopi s objektom i tada je spajanjem palca i kažiprsta moguća interakcija s objektom. Prikaz korištenja metode dodira i metode stiska nalazi se na slici. (Sl. 7)



Sl. 7 Prikaz interakcije metodom stiska pomoću zrake i dodirrom [14]

3. Eksperiment

Opis eksperimenta

Eksperiment će započeti stvaranjem korisnika na sredini sobe. Korisnik će morati otići do postolja s kockama na početku sobe te će morati prenijeti kocke na odgovarajuće postolje na drugom kraju sobe.

Također tijekom eksperimenta broji se vrijeme. Vrijeme će početi kada korisnik podigne prvu kocku, a prestat će kada korisnik posloži sve kocke na odgovarajuća postolja. Vrijeme će se brojati u minutama i sekundama i tijekom cijelog eksperimenta će biti vidljivo korisniku.

U eksperimentu će se testirati prenošenje jedne kocke i više kocaka istovremeno, prenošenje u različitim osvjetljenjima prostorije i prenošenje ovisno o broju prekrivenih kamera na *headsetu*. U prenošenju jedne ili više kocaka istovremeno analizirat će se vremenski rezultati dobiveni testiranjem, a testiranje će se provesti 8 puta korištenjem ruku i 8 puta korištenjem kontrolera.

U testiranju pod utjecajem osvjetljenja i prilikom zaklonjenosti kamera, neće se analizirati vrijeme nego povećanje i smanjenje performansi u tim uvjetima. Testiranje pod utjecajem osvjetljenja će se provesti 3 puta: 1. pod minimalnim utjecajem osvjetljenja (pošto se eksperiment neće moći izvoditi po noći, provesti će se kada utjecaj sunca bude minimalan kao npr. pod mračnim nebom i spuštenim zavjesa i uklonit će se svi drugi izvori svjetlosti u prostoriji. Doduše soba je okružena staklenim zidovima stoga će neki izvori svjetlosti ipak postojati), 2. pod normalnim osvjetljenjem (tijekom dana kada je zadovoljena vidljivost u prostoriji), 3. pod velikim osvjetljenjem (pod velikim utjecajem sunca i drugih izvora svjetlosti).

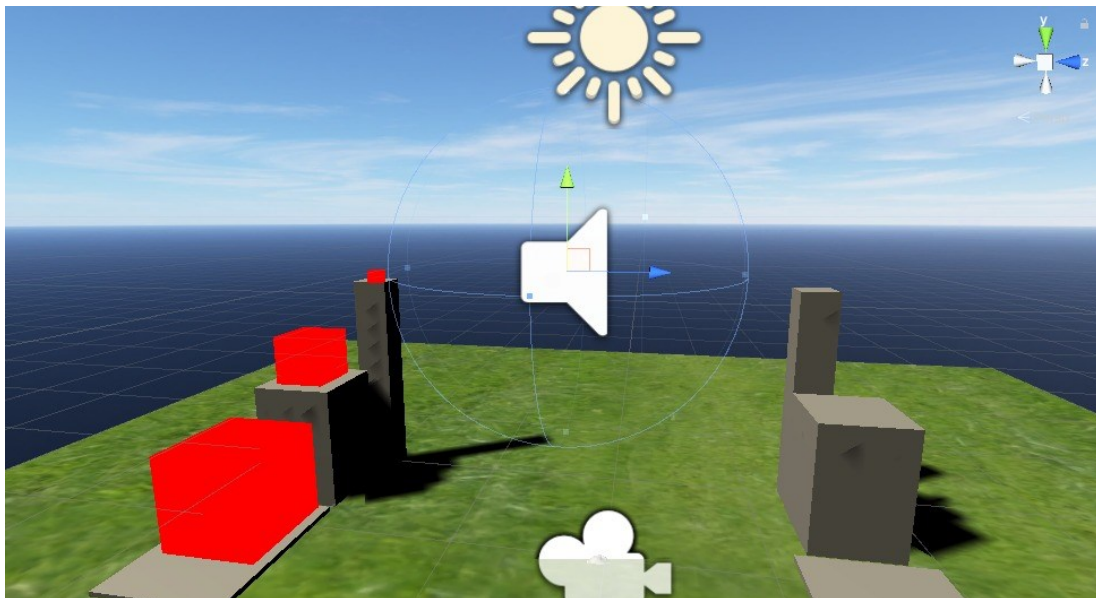
Testiranje performansi zaklanjanjem kamera testirat će se zaklanjanjem jedne kamere, zaklanjanjem 2 kamere, zaklanjanjem 3 kamere i zaklanjanjem 4 kamere. Zaklanjanje 1 kamere će se provesti jednom za svaku kameru, znači 4 puta. Zaklanjanje 2 kamere će se provesti 5 puta (zaklanjanjem dvije gornje, zaklanjanjem 2 donje, zaklanjanjem 2 lijeve, zaklanjanjem 2 desne i zaklanjanjem 2 dijagonalne kamere). Zaklanjanje 3 kamere će se

provesti jednom i zaklanjanje 4 kamere će se provesti jednom. Testiranje zaklanjanjem kamera će se provesti tako da se crnom krpom prekriju odabrane kamere.

Opis prostorije

Bitni elementi u prostoriji: 1. kocke (1 mala, 1 srednja, 1 velika), 2. postolja (2 mala, 2 srednja, 2 velika). Jedno malo, jedno srednje i jedno veliko postolje stoji na početku prostorije i jedno malo, jedno srednje i jedno veliko na kraju prostorije. Na početku eksperimenta, kocke se nalaze na postoljima na početku prostorije. Najveća kocka se nalazi na najmanjem postolju, srednja kocka na srednjem postolju i najmanja kocka na najvećem postolju.

Prostorija će biti otvorena odnosno oko nje neće biti zidovi nego će biti okružena nebeskim svodom (engl. *Skybox*) izgleda plavog neba. (Sl. 8)



Sl. 8 Prikaz prostorije za testiranje

Cilj eksperimenta

Cilj je usporediti korištenje ruku s korištenjem kontrolera u virtualnoj stvarnosti. Mjerit će se lakoća i brzina izvođenja i usporediti utjecaj osvjetljenja i zaklonjenosti kamera na praćenje kontrolera i praćenje ruku. Sustav za praćenje ruku će se usporediti s kontrolerima za svaku vrstu ispitivanja, a na kraju će se popisati sve prednosti i svi nedostaci jednog i drugog.

4. Rezultati eksperimenta

Zbog epidemioloških razloga ispitivanje nije bilo provedeno na skupini ljudi nego sam testiranje provodio samostalno. Tijekom izrade aplikacije, dakle prije nego što je aplikacija bila u potpunosti gotova i spremna za ispitivanje, morao sam isprobavati korištenje i kontrolera i sustava za praćenje ruku, no problemi i rezultati koje sam postigao tijekom izrade aplikacije neće biti uzete u obzir u završnom testiranju. Svi rezultati u ovom poglavlju vezani su uz rezultate koji su postignuti tek kada je aplikacija bila gotova, a provodili su se na način koji sam opisao u prethodnom poglavlju.

4.1. Vrijeme potrebno za provođenje eksperimenta

Testiranje prenošenjem najviše 1 kocke istovremeno

Najkraće vrijeme koje je bilo potrebno za provedbu testiranja koristeći kontrolere je 15 sekundi, a najduže 45 sekundi.

Najkraće vrijeme koje je bilo potrebno za provedbu testiranja koristeći ruke je 17 sekundi, a najduže 1 minuta i 8 sekundi.

Testiranje prenošenjem najviše 2 kocke istovremeno

Najkraće vrijeme koje je bilo potrebno za provedbu testiranja koristeći kontrolere je 10 sekundi, a najduže 26 sekundi.

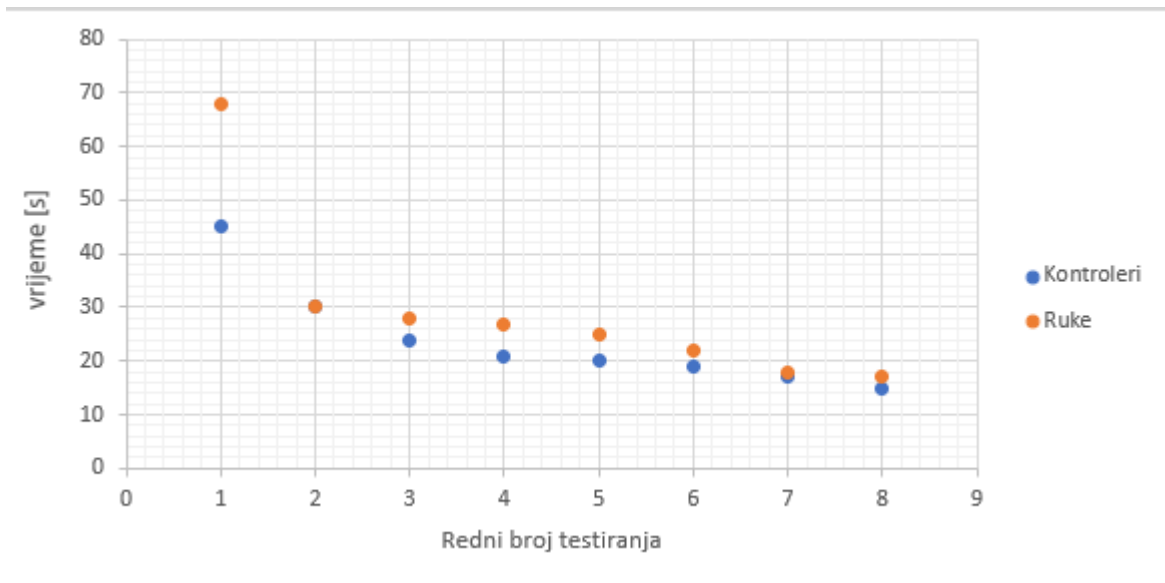
Najkraće vrijeme koje je bilo potrebno za provedbu testiranja koristeći ruke je 15 sekundi, a najduže 46 sekundi.

Ostala vremena provođenja testiranja korištenjem kontrolera prenoseći najviše 1 kocku istovremeno su: 17 sekundi, 19 sekundi, 20 sekundi, 21 sekunda, 24 sekunde i 30 sekundi, što čini prosječno vrijeme od 23.875 sekundi.

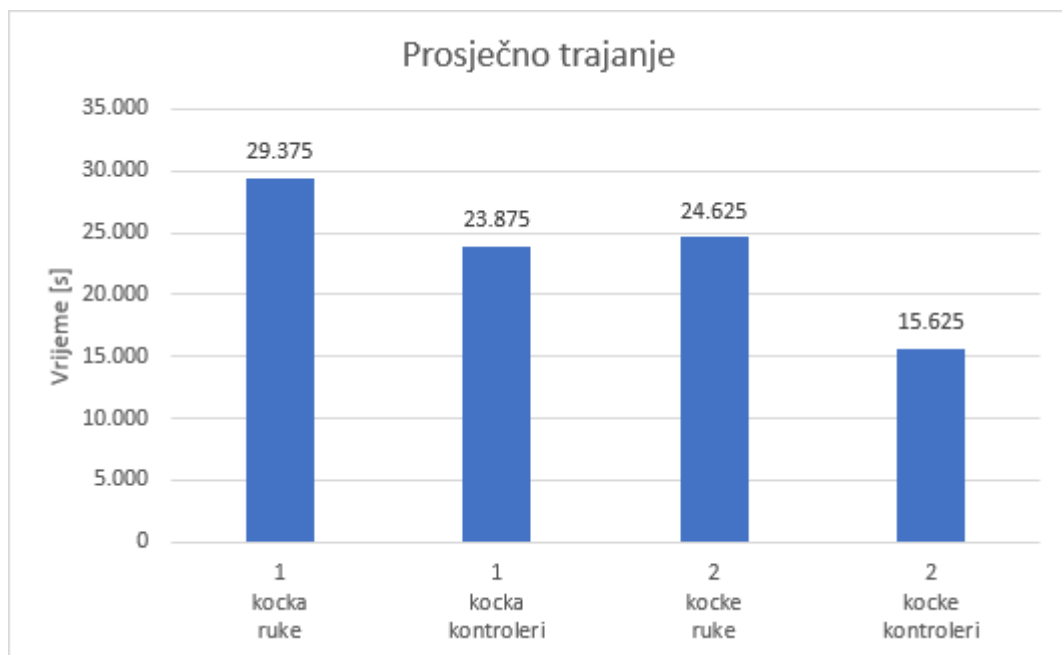
Ostala vremena provođenja testiranja korištenjem kontrolera prenoseći najviše 2 kocke istovremeno su: 11 sekundi, 11 sekundi, 12 sekundi, 15 sekundi, 17 sekundi i 23 sekunde, što čini prosječno vrijeme od 15.625 sekundi.

Ostala vremena provođenja testiranja korištenjem ruku prenoseći najviše 1 kocku istovremeno su: 18 sekundi, 22 sekundi, 25 sekundi, 27 sekundi, 28 sekundi i 30 sekundi, što čini prosječno vrijeme od 29.375 sekundi.

Ostala vremena provođenja testiranja korištenjem ruku prenoseći najviše 2 kocke istovremeno su: 18 sekundi, 20 sekundi, 23 sekundi, 23 sekundi, 24 sekundi i 28 sekundi, što čini prosječno vrijeme od 24.625 sekundi. (Sl. 9) (Sl. 10)



Sl. 9 Prikaz trajanja svakog testiranja za ruke i za kontrolere



Sl. 10 Prikaz prosječnog vremena za ruke i kontrolere ovisno o količini istovremeno prenošenih kocaka

ANALIZA REZULTATA

Usporedba za prenošenje najviše 1 kocke istovremeno

U najboljem slučaju cilj se brže postiže korištenjem kontrolera za 11.8%, a u najgorem za 33.8%. Njihova prosječna vremena se razlikuju za 18.7% u korist kontrolera.

No odstupanje najgoreg rezultata od drugog najgoreg za kontrolere je 33.33% dok za ruke iznosi 55.88%. Razlog tomu je što prilikom jednog testiranja korištenjem ruku, iako sam otpustio stisak, kocka mi je i dalje ostala u ruci umjesto da se odvoji i nisam ju mogao otpustiti neko vrijeme. Većina vremena mi je otišla na otpuštanje kocke i zato je najgori slučaj iznosio 68 sekundi, što je znatnije povećalo prosjek. Unatoč tome, čak i ako se taj rezultat zanemari, po ostalim rezultatima vidi se da kontroleri još uvijek postižu bolji rezultat u ovoj kategoriji.

Usporedba za prenošenje najviše 2 kocke istovremeno

U najboljem slučaju cilj se brže postiže korištenjem kontrolera za 33.33%, a u najgorem za 43.48%. Njihova prosječna vremena razlikuju se za 36.5% u korist kontrolera.

Kontroleri su i u ovoj kategoriji imali bolji rezultat. Razlog tomu je što, kod sustava za praćenje ruku, ako dođe do preklapanja ruku ili ruke budu međusobno vrlo blizu, Quest će imati poteškoća prepoznajući nalazi li se kocka u ruci i u nekim slučajevima će ispustiti kocku te će ju korisnik opet morati podići zbog čega se dodatno gubi vrijeme.

4.2. Utjecaj osvjetljenja

Ispitivanje nije pokazalo različite rezultate pri testiranju s različitim osvjetljenjem. Pod osvjetljenjem u ovom eksperimentu najvećim djelom mislim na sunce kojem je eksperiment bio izložen. Testiranje je bilo provedeno i kada je sunce bilo orijentirano točno prema/u prostoriju, a i kada ga uopće nije bilo i još k tome bez ikakvih drugih izvora svjetlosti u prostoriji. Treba napomenuti da soba ima staklene zidove, stoga nikada nije mogao biti potpuni mrak u prostoriji te ispitivanje nije bilo provedeno u takvom uvjetu.

4.3. Zaklonjenost kamere

Svaki utjecaj zaklonjenosti jednako se odnosio i na sustav za praćenje ruku i na kontrolere. Zbog toga ako se u ostatku potpoglavlja spomenu kontroleri, jednako će se odnositi i na ruke i obrnuto. Quest jako dobro podnosi zaklonjenost te se utjecaj osjeti tek kod zaklonjenosti sve četiri kamere dok zaklanjanje 3 ili manje kamere ima minimalnih ili gotovo nikakvih utjecaja.

Zaklonjenost 1 kamere

Zaklonjenost 1 kamere primjetno uopće ne utječe na praćenje ruku ili kontrolera. Neovisno koja od kamere se zakloni, i praćenje kontrolera i praćenje ruku odvija se jednako dobro kao i sa sve 4 kamere. Pretpostavljam da zaklanjanje 1 kamere ipak utječe na performanse zbog razloga kojeg ću opisati u sljedećem odjeljku („Zaklonjenost 2 kamere“), ali to je toliko mali utjecaj da ga nisam uspio primijetiti.

Zaklonjenost 2 kamere

Čak i kod zaklanjanja 2 kamere ne dolazi do bitnijeg smanjenja performansa. Utjecaj je isti i na praćenje kontrolera i na praćenje ruku.

Ako se prikriju 2 dijagonalne kamere, učinak je skoro pa ne postojeći kao i kod zaklanjanja samo 1 kamere. Ako se prikriju 2 lijeve, 2 desne, 2 gornje ili 2 donje kamere, slabljenje performansi je još uvijek jako malo i primjećuje se tek kod pomaka ruke prema kutu između otprilike 170 do 180 stupnjeva. Kako kamera ne može pratiti ruke iza leđa, pomicanjem ruku više od 180 stupnjeva od vidnog polja, ruke nestanu. Kada su sve 4 kamere u funkciji, nestajanje je brzo i glatko no kada su 2 kamere prikrivene, pomicanjem ruku prema granici vidnog polja, ruke počinju treperiti i nestajati pa se opet pojavljivati (kao da *headset* ne zna bi li ih trebao prikazati ili ne). Također treperenje se događa samo na granicama vidnog polja kod kojih se kamere prekriju (2 gornje ili 2 dolje ili 2 lijeve ili 2 desne kamere). Zbog tog razloga pretpostavljam da i prekrivanje samo jedne kamere ima utjecaj na praćenje ruku ili kontrolera, ali je utjecaj neprimjetno mali.

Zaklonjenost 3 kamere

Utjecaj je prividno isti kao i kod zaklonjenosti 2 kamere. Vjerojatno je praćenje ipak malo više otežano u odnosu na zaklonjenost 2 kamere, ali to nije bilo primjetno.

Zaklonjenost 4 kamere

Kada se zaklone sve 4 kamere, uređaj ne može funkcionirati, aplikacija se zaustavlja i ispisuje se poruka da su sve kamere prikrivene i da Quest ne može pratiti položaje ruku.

4.4. Prednosti ruku i kontrolera

Prednosti ruku

Jedna od prednosti ruku je što su ruke cijelo vrijeme slobodne. Nije potrebno voditi brigu o kontrolerima što stvara veći osjećaj virtualne stvarnosti. Ako u nekom trenutku nije potrebno koristiti ruke u virtualnoj stvarnosti, mogu se koristiti za opipavanje stvari oko korisnika kako bi se korisnik uvjerio da oko njega nema ništa što bi htio dok je u virtualnoj stvarnosti. Iako Quest upozorava korisnika ako se približi sigurnosnoj granici, ipak je dobro imati i mogućnost samoprovjere, zbog nekih nepredviđenih situacija kao što može biti ulazak kućnog ljubimca unutar korisnikove virtualne granice ili druge osobe koja nije svjesna raspona korisnikove granice za virtualnu stvarnost.

Prednosti kontrolera

Kontroleri su bolje detektirali koliziju i hvatanje objekata od sustava za praćenje ruku. Za razliku od sustava za praćenje ruku, korištenjem kontrolera nije bilo upitno hoće li se pritiskom gumba za hvatanja predmet doista uhvatiti ili neće. Jednom kada je objekt bio uhvaćen, bilo je sigurno da će se moći odnijeti do cilja bez ispadanja.

Također, tijekom nošenja kocke, kocka je bila prilično stabilna. Nije se zakretala niti naginjala i dobro je podnosila brze okretaje tijelom. Kod brzih rotacija za 180 stupnjeva, kocka bi se također zarotirala bez ikakvog prividnog kašnjenja dok kod sustava za praćenje ruku to nije bio slučaj.

4.5. Nedostatci ruku i kontrolera

Nedostatci ruku

Tijekom testiranja *headset* je teže detektirao položaj ruku nego položaj kontrolera. Dodatno, u nekim situacijama sustav nije mogao prepoznati drži li se kocka ili ne što je prouzročilo ispadanju kocke iz ruke usred nošenja. To je prouzročilo nepotreban gubitak vremena što je jedan od razloga zbog čega su kontroleri dali bolje vremenske rezultate.

Također, kod brzih pokreta, sustav nije mogao detektirati hvatanje objekata. Naprimjer kada bih se žurno sagnuo i brzo zamahnuo prema kocki i uhvatio ju (u smislu da sam napravio radnje koje bi trebale uhvatiti kocku), sustav nije prepoznao radnju hvatanja i kocka je ostala na mjestu. To je bio još jedan od razloga zbog čega je sustav za praćenje ruku imao slabije rezultate tijekom testiranja jer testiranje nije ovisilo o mojoj brzini hvatanja, nego o brzini detekcije hvatanja.

Nedostatci kontrolera

Pošto se ruke koriste za držanje kontrolera, ne mogu se koristiti za ništa drugo kao npr. za skidanje ili popravljavanje položaja *headseta* na glavi ili stišavanje zvuka, stavljanje slušalica i sl., a također uporaba kontrolera za interakciju u virtualnom okruženju smanjuje utjecaj i doživljaj virtualne stvarnosti.

Zaključak

U ovome testiranju kontroleri su pokazali bolje rezultate. Razlog boljeg postizanja rezultata jest preciznost i točnost koje uporaba kontrolera pruža za razliku od sustava za praćenje ruku. Problem kod korištenja ruku bio je što se katkada dolazi do krive procjene sustava o interakciji između ruku i drugih objekata. Također, korištenje ruku je pokazalo određenu razinu nesigurnosti pri brzim kretnjama. Problemi su se stvarali kod brzog pokreta rukama prema drugim objektima i hvatanju tih objekata, kada sustav ponekada ne bi prepoznao da sam rukom uhvatio objekt nego bi ruka samo prošla kroz njega. Uz to, premda rijetko, čak i pri normalnim kretnjama, sustav za praćenje ruku imao je određenu dozu kašnjenja kao na primjer kada bi se predmet podigao tek oko sekunde nakon sam stvarno podigao objekt što može biti zbunjujuće i frustrirajuće.

S druge strane, ruke su mi pružile veći osjećaj virtualne stvarnosti i da se nije mjerilo vrijeme, frustracije vjerojatno ne bi ni bilo. Također pošto se ne trebaju koristiti kontroleri, ruke su slobodne stoga ih se može koristiti za druge stvari kao npr. ako korisniku nešto padne ili želi popraviti *headset* i slično. Na primjer, iako Quest ima svoje sigurnosne granice, nisam se osjećao sasvim sigurno. Virtualna stvarnost mi je relativno nova stvar pa bih možda sigurnost stekao s vremenom, ali tijekom ovog eksperimenta mnogo sigurnije sam se osjećao koristeći vlastite ruke umjesto kontrolera, jer sam se lakše mogao uvjeriti da sam u sigurnosnom prostoru i ako dođe do pada puno bih se lakše snašao bez kontrolera u ruci.

Uzimajući sve navedeno u obzir, ja bih osobno radije odabrao sustav za detekciju ruku jer iako nije precizan kao kontroler, ipak nije puno u zaostatku. Dodatno, upravo preciznost kontrolera sve čini vrlo jednostavnim dok se s rukama treba malo pomučiti što daje mnogo veći osjećaj realnosti.

Literatura

[1] Pandžić I. S., Pejša T., Matković K., Benko H., Čereković A., Matijašević M., „VIRTUALNA OKRUŽENJA: INTERAKTIVNA 3D GRAFIKA I NJENE PRIMJENE“, 1. izdanje, Zagreb: ELEMENT, 2011.

[2] Pandžić I. S., Pejša T., Matković K., Benko H., Čereković A., Matijašević M., „VIRTUALNA OKRUŽENJA: INTERAKTIVNA 3D GRAFIKA I NJENE PRIMJENE“, 1. izdanje, Zagreb: ELEMENT, 2011.

[3] Poveznica:

https://www.google.com/search?q=oculus+quest&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiRub_HjLTpAhWxsaQKHT_QDHwQ_AUoAXoECA0QAw&biw=1242&bih=568#imgrc=qP0AWvDTJkh_PM, pristupljeno 15. svibnja 2020.

[4] *What's the Oculus mobile app, and where can I get it?* Poveznica: <https://support.oculus.com/322769838595481/>; pristupljeno 11. lipnja 2020

[5] *Inertial measurment unit*, Poveznica:

https://en.wikipedia.org/wiki/Inertial_measurement_unit?fbclid=IwAR03_zt4XwkumgxOcojMP7FdTwYIWyb2axwcqOLOhFXkaJUN_OIITRKXNz4, pristupljeno 11. lipnja 2020.

[6] Joel Hesch, Oskar Linde, Anna Kozminski, AR/VR | COMPUTER VISION, 22.

kolovoza 2020., *Powered by AI: Oculus Insight*, Poveznica:

<https://ai.facebook.com/blog/powered-by-ai-oculus-insight/>, pristupljeno 11. lipnja 2020.

[7] Poveznica:

https://www.google.com/search?q=oculus+quest+controllers&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiix62exfDpAhWOqaQKHfSZBHYYQ_AUoAXoECAwQAw&biw=1366&bih=576#imgrc=qeKAZx9LKuQdKM&imgdii=cMx8fCgmhsCjVM; pristupljeno 11.

lipnja 2020.

[8] Poveznica:

https://www.google.com/search?q=oculus+quest+controllers&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiix62exfDpAhWOqaQKHfSZBHYYQ_AUoAXoECAwQAw&biw=1366&bih=576#imgrc=cMx8fCgmhsCjVM&imgdii=qeKAZx9LKuQdKM; pristupljeno 11.

lipnja 2020.

- [9] What It's Like To Get An Oculus Quest | Unboxing & Setup, 6. lipnja 2019.,
Poveznica: <https://www.youtube.com/watch?v=GKXZXb9h1HM>; pristupljeno 11. lipnja 2020.
- [10] What It's Like To Get An Oculus Quest | Unboxing & Setup, 6. lipnja 2019.,
Poveznica: <https://www.youtube.com/watch?v=GKXZXb9h1HM>; pristupljeno 11. lipnja 2020.
- [11] 11. listopada 2019., *Hand Tracking Deep Dive: Technology, Design, and Experiences*,
Poveznica: <https://www.youtube.com/watch?v=gpQePH-Ffbw&fbclid=IwAR1x9R2EUVaXg-hG2jnzvIeYhR9JnRil9TTnjKkvATimMQhJScBwdQGqYg>, pristupljeno 11. lipnja 2020.
- [12] *Interactions*. Poveznica: <https://developer.oculus.com/design/hands-design-interactions/>, pristupljeno 11. lipnja 2020.
- [13] *Best Practices*. Poveznica: <https://developer.oculus.com/design/hands-design-bp/>; pristupljeno 11. lipnja 2020.
- [14] *Interactions*. Poveznica: <https://developer.oculus.com/design/hands-design-interactions/>, pristupljeno 11. lipnja 2020.
- [15] Pandžić I. S., Pejša T., Matković K., Benko H., Čereković A., Matijašević M.,
„VIRTUALNA OKRUŽENJA: INTERAKTIVNA 3D GRAFIKA I NJENE PRIMJENE“,
1. izdanje, Zagreb: ELEMENT, 2011.
- [16] Pandžić I. S., Pejša T., Matković K., Benko H., Čereković A., Matijašević M.,
„VIRTUALNA OKRUŽENJA: INTERAKTIVNA 3D GRAFIKA I NJENE PRIMJENE“,
1. izdanje, Zagreb: ELEMENT, 2011.
- [17] Pandžić I. S., Pejša T., Matković K., Benko H., Čereković A., Matijašević M.,
„VIRTUALNA OKRUŽENJA: INTERAKTIVNA 3D GRAFIKA I NJENE PRIMJENE“,
1. izdanje, Zagreb: ELEMENT, 2011.
- [18] Pandžić I. S., Pejša T., Matković K., Benko H., Čereković A., Matijašević M.,
„VIRTUALNA OKRUŽENJA: INTERAKTIVNA 3D GRAFIKA I NJENE PRIMJENE“,
1. izdanje, Zagreb: ELEMENT, 2011.

Sažetak

NASLOV: Usporedba iskustvene kvalitete prilikom korištenja detekcija šake naspram korištenja kontrolera u virtualnoj stvarnosti

Virtualna stvarnost je pojam za računalnu simulaciju u kojoj je cilj korisniku pružiti doživljaj virtualnog okruženja [15]. Toj stvarnosti može se pristupiti do određene granice, a kroz povijest ljudi su raznim uređajima pokušavali sve više proširivati tu granicu [16]. Do danas se razvio uređaj za virtualnu stvarnost Oculus Quest kojeg će se u ovome radu opisati.

Oculus Quest jedan je od novijih uređaja za virtualnu stvarnost koji je uveo sustav za praćenje ruku. Quest je bežičan uređaj što znači da se može koristiti neovisno o računalu. Za njegovo korištenje potrebna su samo 2 kontrolera i *headset* ili čak samo *headset* ako se koristi sustav za praćenje ruku. Za praćenje položaja korisnika ne zahtijeva nikakve vanjske senzore što je, uz bežičnost, velika prednost u odnosu na neke dosadašnje uređaje za virtualnu stvarnost.

Pošto je sustav za praćenje ruku relativno nova stvar, u ovom radu provesti će se eksperiment čiji je cilj usporediti korištenje kontrolera sa sustavom za praćenje ruku. Eksperiment je tipa „uzmi i postavi“ (engl. „*pick and place*“). U eksperimentu prenosit će se 3 kocke različite veličine i svaku je potrebno staviti na svoje mjesto. Cilj je prenijeti kocke u što kraćem vremenu. Rezultat dobiven korištenjem kontrolera analizirat će se i uspoređivati s rezultatima dobivenim korištenjem sustava za praćenje ruku.

Ključne riječi: virtualna stvarnost, Oculus Quest, detekcija ruku, kontroleri, *headset*

Summary

TITLE: Comparison of users experience by use of hand tracking against use of controllers in virtual reality

Virtual reality is term for computer simulation which goal is to provide user with sense of virtual environment [17]. We can access to that kind of reality untill we reach some point but through history people have been pushind that limit [18]. Untill today device for virtual reality has benn made, the one which I will describe in this work.

Oculus Quest is one of newer devices for virtual reality whice has brought in system for hand tracking. Quest is wireless device which means it can be used undependently of a computer. For its use only 2 controllers and one headset is needed or even just headset if hand tracking is used. It requires no external sensors for tracking the user which is, togeher with wirelsseness, big advantage compared to some previous devices for virtual reality.

Since hand tracking is relatively new thing, in this work an experiment will be held which goal is to compare use of a controler with hand tracking. Name of experiment is "Pick and Place". In the experiment 3 cubes of different sizes will be brought and each of them will be needed to put to the required place. The goal is to place cubes as short as possible. The results from using controllers will be compared to the ones got by use of hand tracking.

Key words: virtual reality, Oculus Quest, hand tracking, controllers, headset

Privitak

U privitku se nalaze sve skripte koje sam napisao za aplikaciju.

HandTrackingGrabber.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using OculusSampleFramework;
public class HandTrackingGrabber : OVRGrabber
{
    private OVRHand hand;
    public float pinchTreshold = 0.7f;
    protected override void Start()
    {
        base.Start();
        hand = GetComponent<OVRHand>();
    }
    public override void Update()
    {
        base.Update();
        CheckIndexPinch();
    }
    void CheckIndexPinch()
    {
        float pinchStrength =
hand.GetFingerPinchStrength(OVRHand.HandFinger.Index);
        bool isPinching = pinchStrength > pinchTreshold;
        if (!m_grabbedObj && isPinching &&
m_grabCandidates.Count > 0)
            GrabBegin();
        else if (m_grabbedObj && !isPinching)
            GrabEnd();
    }
}
```

HandTrackingGrabber omogućuje sustavu za praćenje ruku interakciju s drugim objektima.

CollisionMP2.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CollisionMP2 : MonoBehaviour
{
    public static int broj = 0;

    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.name == "kocka_Velika")
        {
            ++broj;
        }
    }

    private void OnTriggerExit(Collider other)
    {
        if (other.gameObject.name == "kocka_Velika")
        {
            --broj;
        }
    }
}
```

CollisionMP2 koristi se kako bi aplikacija prepoznala nalazi li se velika kocka na malom postolju.

CollisionSP2.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CollisionSP2 : MonoBehaviour
{
```

```

public static int broj = 0;

private void OnTriggerEnter(Collider other)
{
    if (other.gameObject.name == "kocka_Srednja")
    {
        ++broj;
    }
}

private void OnTriggerExit(Collider other)
{
    if (other.gameObject.name == "kocka_Srednja")
    {
        --broj;
    }
}
}

```

CollisionSP2 koristi se kako bi aplikacija prepoznala nalazi li se srednja kocka na srednjem postolju.

CollisionVP2.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CollisionVP2 : MonoBehaviour
{
    public static int broj = 0;

    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.name == "kocka_Mala")
        {
            ++broj;
        }
    }

    private void OnTriggerExit(Collider other)
    {

```

```

        if (other.gameObject.name == "kocka_Mala")
        {
            --broj;
        }
    }
}

```

CollisionVP2 koristi se kako bi aplikacija prepoznala nalazi li se mala kocka na velikom postolju.

StartTimeV.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class StartTimeV : MonoBehaviour
{
    public static int start = 0;

    private void OnTriggerExit(Collider other)
    {
        if (other.gameObject.name == "kocka_Mala")
        {
            start = 1;
        }
    }
}

```

StartTimeV.cs koristi se kako bi aplikacija prepoznala kada da počne brojati vrijeme.

StartTimeS.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class StartTimeS : MonoBehaviour
{
    public static int start = 0;
}

```

```

private void OnTriggerExit(Collider other)
{
    if (other.gameObject.name == "kocka_Srednja")
    {
        start = 1;
    }
}
}

```

StartTimeS.cs koristi se kako bi aplikacija prepoznala kada da počne brojati vrijeme.

StartTimeM.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class StartTimeM : MonoBehaviour
{
    public static int start = 0;

    private void OnTriggerExit(Collider other)
    {
        if (other.gameObject.name == "kocka_Velika")
        {
            start = 1;
        }
    }
}

```

StartTimeM.cs koristi se kako bi aplikacija prepoznala kada da počne brojati vrijeme.

Timer.cs

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class Timer : MonoBehaviour
{
    public Text counterText;
}

```

```

private string x;

public float seconds = 0f, minutes = 0f;

void Start()
{
    counterText = GetComponent<Text>() as Text;
}

void Update()
{
    if (CollisionVP2.broj == 1 && CollisionSP2.broj == 1
&& CollisionMP2.broj == 1)
    {
        counterText.text = x;
    }
    else
    {
        if (StartTimeV.start == 1 || StartTimeS.start ==
1 || StartTimeM.start == 1)
        {

            seconds += Time.deltaTime;
            if ((int)seconds == 60)
            {
                minutes = minutes + 1f;
                seconds = 0f;
            }
            counterText.text = minutes.ToString("00") +
":" + seconds.ToString("00");
            x = minutes.ToString("00") + ":" +
seconds.ToString("00");
        }
    }
}
}

```

Timer.cs služi za mjerenje vremena.