Sadržaj

Uvod

Razvoj tehnologija virtualne stvarnosti je doveo do njenog čestog korištenja u edukativne svrhe. Korisniku pruži mogućnost kretanja u tri i u šest stupnjeva slobode i kretanje u prostoru te je savršena za razvoj edukativnih usluga, primjera i igara. Pomaže u razumijevanju teških koncepata zbog svoje izrazite sposobnosti vizualizacije i uronjenosti.

Videoigre predstavljaju jednu od najbrže rastućih industrija te se svake godine proizvode i razvijaju igre u svrhu edukacije, zabave i razbibrige. Pomoću njih se može stvoriti skoro bilo koji scenarij koji se može zamisliti te su zbog toga postale jako popularne među mlađim generacijama i interes sa različite dijelove razvoja i dizajna igara raste iz godine u godinu. Jedan od tih dijelova je grupiranje igrača u meč u višekorisničkim igricama. Cilj ovog rada je kombinirati tehnologiju virtualne stvarnosti i razvoj videoigra sa tehnologijom *Unity* i razviti aplikaciju temeljenu na miješanoj stvarnosti za pomoć pri učenju o algoritmima za spajanje igrača u mečeve u videoigrama.

Rad je podijeljen u 4 glavna poglavlja, poglavlje 1 *Pregled literature* prolazi kroz sve bitne tehnologije korištene u radu. *Virtualna stvarnost* unutar koje se opisuje povijest i glavne funkcionalnosti virtualne stvarnosti i njeno korištenje u radu. *Sustavi za spajanje igrača u višekorisničkim igrama* opisuju glavne ideje i teoriju iza algoritama za spajanje igrača i daju kratki opis njihovih karakteristika. Poglavlje 2, *Opis formalnog modela* predstavlja sve potrebne zahtjeve funkcionalnosti koje rad i aplikacija trebaju imati. Poglavlje 3, *Korištene tehnologije* opisuje sve tehnologije i alate korištene u razvijanju rada. Poglavlje 4, *Izrada obrazovne aplikacije* opisuje u detalje proces razvijanja i funkcionalnosti finalne aplikacije kroz potpoglavlja koja opisuju sve važne scene u radu.

1. Pregled literature

1.1. Virtualna stvarnost

Virtualna stvarnost (kratica VR) je relativno nova tehnologija koja simulira pokrete glave, tijela i ruku u virtualnom 3D okruženju. Definirana je kao virtualno okruženje koje se iskusi pomoću računalnih stimulacija osjeta . Popularizacija termina i prvi razvoj tehnologija koje prethode današnjim VR sustavima je počeo kasnih 1980-ih . VR je danas brzo rastuće i popularno tržište sa mnogo visoko kvalitetnih uređaja kao *Oculus Quest 2*, *Valve Steam Index*, *HTC Vive* i mnogi drugi . Jedan prikaz VR uređaja i virtualne stvarnosti se može vidjeti na Slici 1.1. U ovom radu koristiti će se razvojna okolina *Unity game engine* koji ima mnoge ugrađene VR funkcionalnosti za lakši razvoj VR aplikacija. Skup paketa i knjižnjica koji će se koristiti se zove XR i može se proširiti na ostale tipove simuliranih stvarnosti. XR uspješno emulira gore navedene vrste VR uređaja i kontrolera te se aplikacija može uspješno razvijati za sve uređaje.

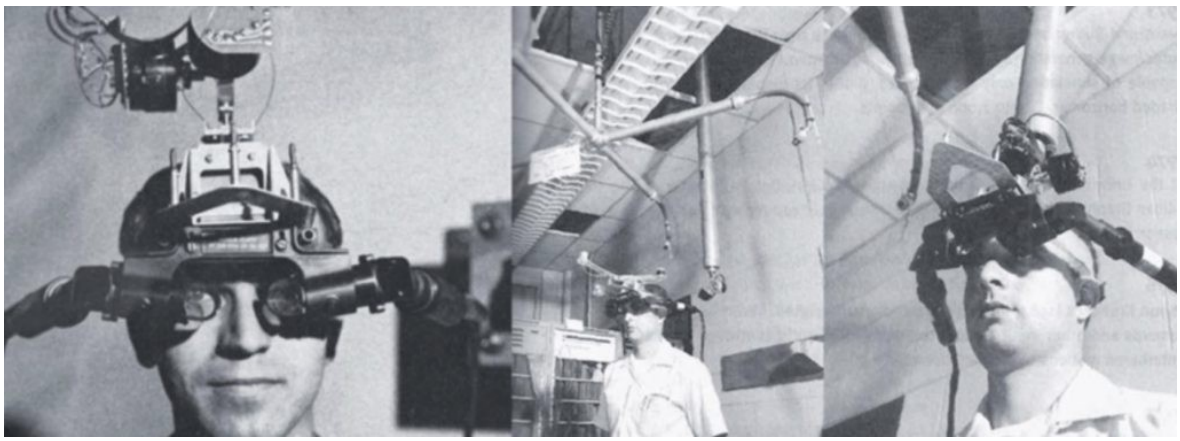
Slika 1.1 Umjetnički prikaz Virtualne Stvarnosti (preuzeto iz)

1.2. Kratka povijest virtualne stvarnosti

Iako je termin „virtualna stvarnost" (*engl. Virtual Reality)* popularizirao Jaron Lanier, osnivač VPL (*Visual Programming Lab)* Research , rani primjeri tehnologija vezanih uz VR dosežu do 1950-ih godina. Kinematograf Morton Heilig je izumio uređaj *Sensorama* koja istovremeno stimulira sva osjetila. *Sensorama* je prikazana u Slici 1.2 Također je patentirao prvi *Head-Mounted display* (HMD) zvan *Telesphere Mask* Heiligovi izumi nemaju mogućnost interakcije sa okolinom niti praćenje pokreta glave, no predstavljaju važnu podlogu za daljnje razvijanje VR tehnologija .

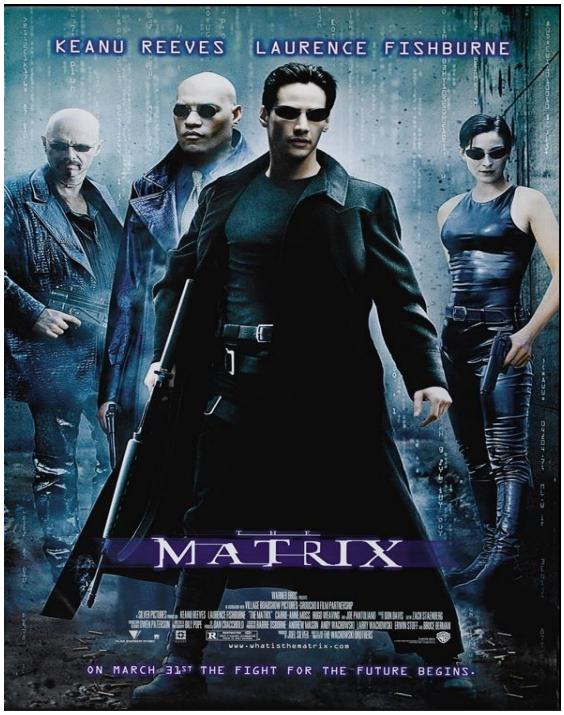
Slika 1.2 Sensorama (preuzeto iz)

Koncepti i izumi iz 1960-ih koji će ilustrirati praćenje pokreta su „*Headsight",* prvi motion-tracking HMD, „*The Ultimate Display"* i *„Sword of Damocles"*. Posljednja dva je dizajnirao pionir Ivan Sutherland i posebno su nam važni jer koncept „*The Ultimate Display"* obuhvaća većinu modernih stavaka virtualne stvarnosti dok „*Sword of Damocles"* je prvi HMD dizajniran za VR/AR upotrebu prikazan u Slici 1.3 .



Slika 1.3 Sword of Damocles (preuzeto iz)

Krajem 1980-ih VR se počinje komercijalizirati sa spomenutim Jaron Lanierom i VPL-om koji su proizveli „*Dataglove"* i „*EyePhone*" te NASA sa svojim projektom „*Project VIEW"* koji simulira dodir. Unatoč uspjesima NASA-e, tijekom devedesetih je dosta uređaja vezanih za VR propalo kao SEGA VR headset i Nintendo Virtual Boy no zahvaljujući popularnosti filmova kao „*The Matrix",* prikazan na Slici 1.4 *i „Lawnmover man"* funkcionalna i komercijalna VR rješenja su ostala poželjna te se područje istraživanja nije napustilo .



Slika 1.4 Plakat za *The Matrix* (preuzeto iz)

„Moderno” doba VR-a počinje oko 2010 godine sa prototipom Oculus Rifta koji je prvi moderni VR headset . Od tada do danas izašlo je mnoštvo samostalnih VR HMD-eva kao Google Cardboard, Oculus Quest 1 i 2, Steam Valve Index, HTC Vive i ostali koji vode trenutnu modernu VR revoluciju. Moderni primjer VR uređaja je prikazan na slici 1.5

Slika 1.5. HTC Vive XR Elite (preuzeto iz)

1.3. Glavne VR tehnologije

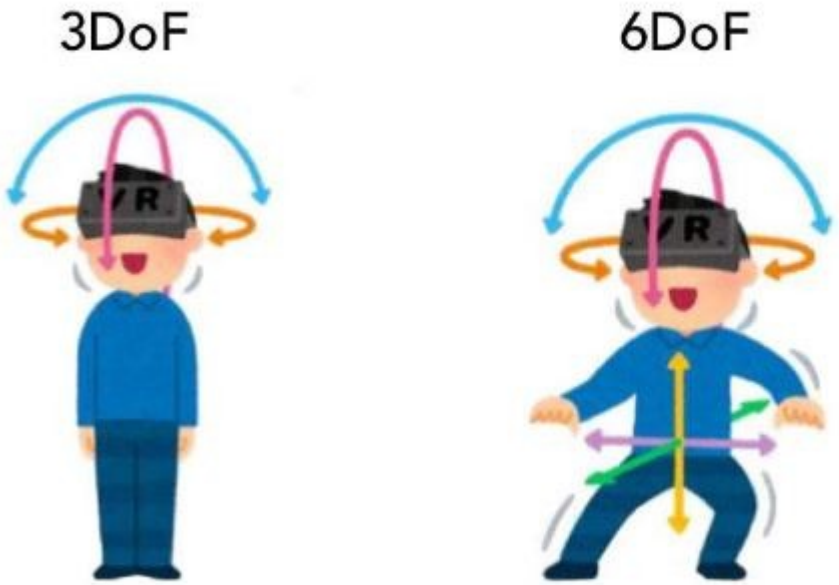
Virtualno okruženje se stvara pomoću mnogo kamera, kontrolera i senzora pokreta. Omogućava se interakcija sa virtualnom generiranom okolinom te se stvara 180 stupnjeva vidno polje za korisnika koje se projicira na zaslon koji korisnik nosi na glavi (*Head Mounted Display ili HMD)* prikazan na Slici 1.6. Praćenje pozicije korisnika i objekata unutar okoline se obavlja na par različitih načina . Kretanje, brzina i rotacija korisnika se mjeri pomoću akcelerometra i žiroskopa, senzori i kamere na HMD-u se koriste za određivanje 3D koordinata u prostoru te se pomoću računalnog vida i markera na objektima sve to sažima u sliku ili okvir koja se prikazuje korisniku. Nakon prikaza slike, često čak i tijekom prikaza se odmah počinje izračunavati sljedeći okvir za daljnji prikaz.

Slika 1.6. Head Mounted Display (preuzeto iz)

Ovakav izračun slike se odvija konstantnom količinom zvanom broj okvira u sekundi (engl. *Frame rate, FPS*). Ljudi ne mogu percipirati razlike iznad 150 FPS-a no broj okvira ispod 60 u virtualnoj stvarnosti stvara nelagodu i vrtoglavicu kod velikog broja ljudi tako da je standard 90-120 FPS-a za većinu uređaja za virtualnu stvarnost .

Stupnjevi slobode (engl. *Degrees of freedom*) se koriste za opisivanje slobode pokreta glave i tijela korisnika. Važni su za određivanje pozicije i orijentacije korisnika za što bolje VR iskustvo. Tri stupnja slobode (3DoF) se koriste kada se prati lokacija i rotacija samo glave tj. Headseta. Ako se prati i pozicija tijela te orijentaciju glave i tijela zasebno tada uređaj ima šest stupnjeva slobode (6DoF) . *Merge VR, Oculus GO i Google Cardboard* su primjeri 3DoF uređaja dok *Oculus Rift, Oculus Quest i HTC Vive* su primjeri 6DoF uređaja. Razlika između stupnjeva slobode je ilustriran na Slici 1.7.

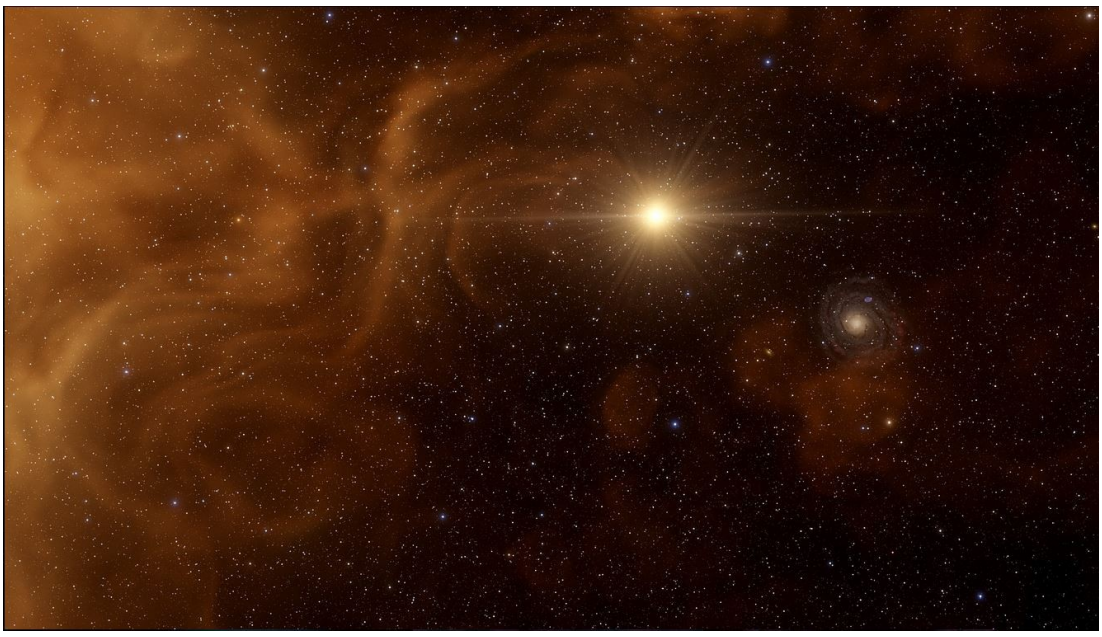
Slika 1.7: Stupnjevi slobode (preuzeto iz)



1.4. Obrazovne VR igre i aplikacije

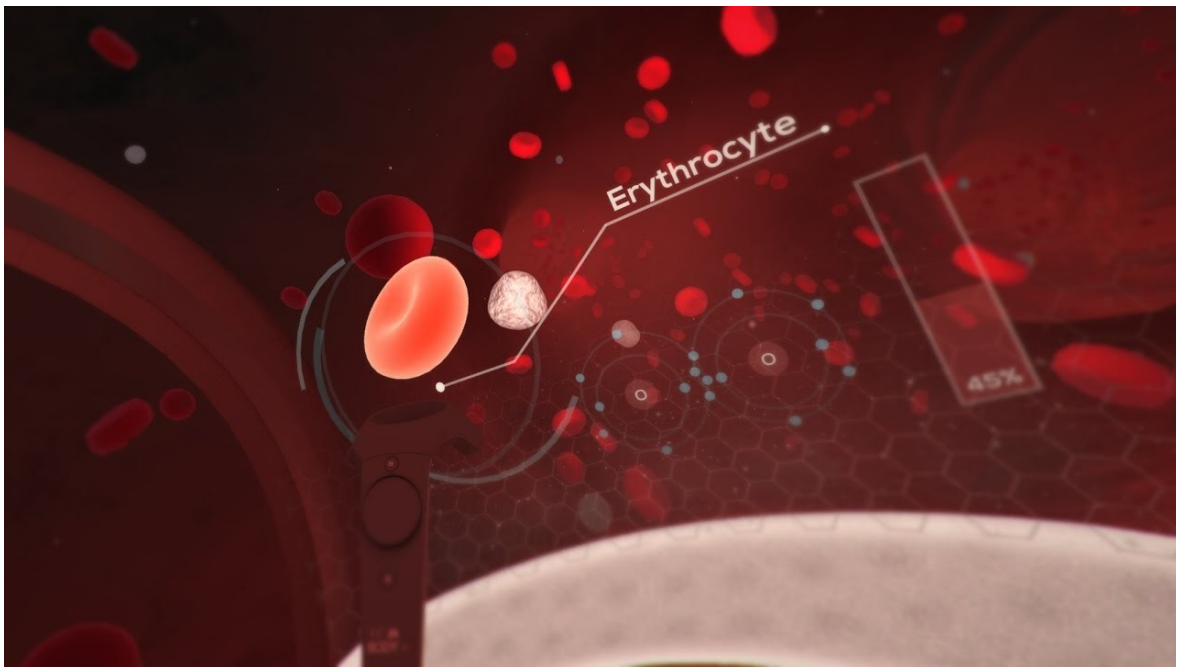
Virtualna stvarnost pruža dodatni sloj stvarnosti koji obični ekrani nemaju. Virtualno iskustvo zbog toga dopušta bolju vizualizaciju, prednosti kod učenja i rješavanje problema koji su bliže stvarnima te tako pruža dobru podlogu za razvoj edukacijskih aplikacija za učenje i produbljivanje znanja pogotovo kod ljudi koji imaju poteškoće s razvojem vještina. Pošto se ovaj rad bavi razvojem obrazovne aplikacije navesti će se primjeri obrazovnih igara i aplikacija kao motivacija za pisanje rada.

Space Engine je realistični virtualni simulator svemira koji se može pokrenuti u VR . Dopušta istraživanje cijelog svemira koje ima veličinu milijardi svjetlosnih godina pomoću svemirskog broda i kontrole toka vremena. Odlična je igra za zabavljat se, ali služi i kao savršena podloga za učenje astronomije, fizike i svemirskih fenomena u sitne detalje. Snimka zaslona jednog dijela svemira iz *Space Engine-a* je prikazan na Slici 1.8.



Slika 1.8 Snimka zaslona iz *Space Engine* (preuzeto iz)

The Body VR je edukacijska igra u virtualnoj stvarnosti koja omogućava igraču da uđe u ljudsko tijelo i istražuje svaku stanicu zasebno . Igrač može pratiti krvne stanice kroz krvotok i učiti o kemijskim i biološkim procesima u detaljnom okruženju virtualne stvarnosti. Jedan dio igre u krvnim žilama je prikazan na Slici 1.9



Slika 1.9. Snimka zaslona iz The Body VR (preuzeto iz)

1.5. Sustavi za spajanje igrača u višekorisničkim igrama

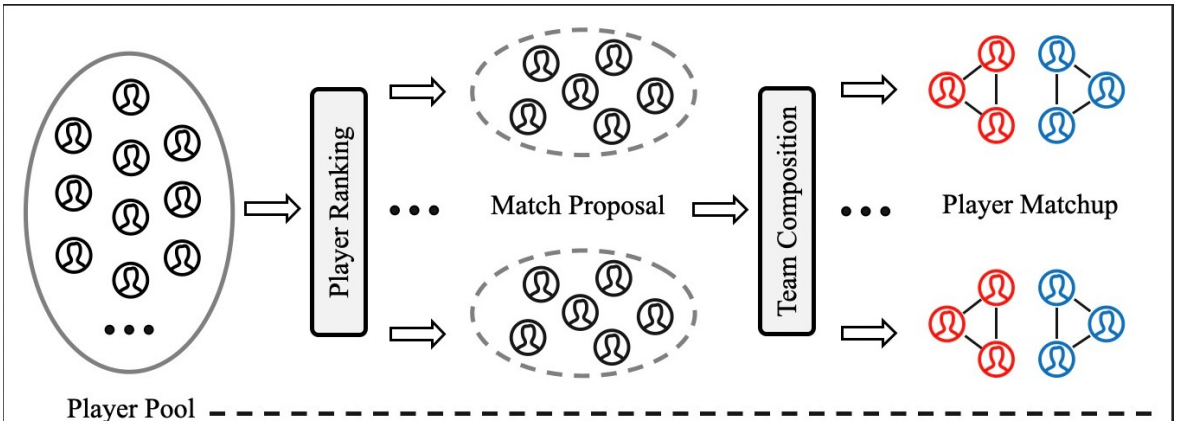
Grupiranje igrača u mečeve ovisno o njihovoj vještini i sposobnosti je glavni dio kompetitivnih igara sa više igrača. Takvo grupiranje se pojavilo u prvim višekorisničkim igricama kao *Doom* ili *Quake* no igrači su tada morali od oka procijeniti vještinu igrača i ručno graditi skupine igrača sa sličnim vještinama. Algoritmi za grupiranje igrača su postojali u raznim 1v1 igrama kao šah gdje bi igračeva vještina bila izražena brojem ovisno o omjeru pobjeda ili gubitaka tzv. *Elo system* ili poboljšani *Glicko system* no igre koji koriste takve sisteme za grupiranje je teško prenijeti na igre sa mnogo više faktora i više igrača od šaha. Jednostavno računanje ocjene vještine se može vidjeti na Slici 1.10.

Slika 1.10: Jednostavni prikaz Elo sistema

Sami sistemi izračunavanja broja koji predstavlja vještinu su izvan dometa ovog rada no valja ih spomenuti kao najvažniji faktor u svim algoritmima za grupiranje. Vrijeme čekanja je isto iznimno važan parametar koji se treba minimizirati za sve igrače. Duže čekanje igrača za meč dovodi do frustracije dok nepažljivo i brzo uparivanje može dovesti do osjećaja beznačajnosti ako je jedan tim mnogo vještiji od drugih. Vrijeme čekanja ovisi o kontekstu igre npr. *League of Legends* ima prosječno trajanje ispod 60 sekundi no među

najviše rangiranim igračima čekanje može trajati čak 45 minuta dok kod MMO igara je normalno čekati 10 minuta za pronalazak grupe. Važno je spomenuti da vrijeme ima značajan efekt na druge faktore i granice odlučivanja, ako je igrač čekao dugo vremena treba ga se što prije staviti u meč čak i ako on nije najbolje moguće rješenje . To znači da vrijeme služi kao neka dinamična granica po kojoj treba određivati strogost ostalih parametara. Kašnjenje obilaska (ping) je isto bitan za minimizaciju mrežnih artefakata kod dugog čekanja i boljeg korisničkog iskustva u meču . Upravo ova tri faktora čine glavni dio simulacije algoritma ovog projekta.

Glavna ideja grupiranja je probati osigurati da je šansa pobjede svake strane što više jednaka i time stvori pošteni meč za sve igrače. Osnovna struktura i ideja grupiranja je prikazana na Slici 1.11. Treba spomenuti sistem *Trueskill,* koji je izumio Microsoft 2007 godine i koji koristi Bayesovsku statistiku, kao prvi moderni algoritam za grupiranje igrača koji se koristi i danas u raznim iteracijama .



Slika 1.11: Struktura grupiranja u mečeve (preuzeto iz)

2. Opis formalnog modela

Aplikacija za grupiranje igrača mora ispuniti neke osnovne zahtjeve u programskom rješenju. U ovom poglavlju će biti nabrojani i opisani, a sama implementacija će se razjasniti u poglavlju 4.

Glavne funkcionalnosti aplikacije:

- Interaktivni glavni izbornik za kretanje po VR aplikaciji
- Prikaz standardiziranih kontrola i upute za VR iskustvo
- Kratka lekcija o algoritmima za grupiranje igrača
- Kviz za provjeru znanja
- Objekt koji imitira karakteristike igrača koji traži meč
- Simulacija algoritma za jednostavno grupiranje igrača
- Vizualizacija samog postupka grupiranja

2.1. Grupiranje igrača u mečeve

Zbog općenitosti modela u zadatku, programsko rješenje će se fokusirati na osnovne faktore koji su sačinjeni u svakom sustavu za grupiranje igrača, nekakva mjera ili aproksimacija igračeve vještine, najčešće *Elo*, kašnjenje obilaska od igrača prema serveru, poznatije kao *ping* i vrijeme provedeno tražeći meč. Ostali faktori ovise o vrsti, kontekstu i različitim sustavima u specifičnoj igri pa ih je teže razraditi na općenitiji način. Unatoč tome, model će imati opisane neke dodatne faktore kao razni klasni sustavi u igrama, dodatni indikator devijacije pouzdanosti (*reliability deviation ili RD ukratko)* i omjer pobijeđenih/izgubljenih igara za svakog igrača. Detaljnija razrada grupiranja igrača faktora i njihov utjecaj na algoritam slijede u sljedećim paragrafima.

Daleko najvažniji faktor je *Elo*, on će biti jednostavno simuliran kao broj uz svakog igrača koji svaki korisnik može promijeniti. Algoritam za grupiranje će uzimati *Elo* u obzir i izbjegavati velike razlike u vještini igrača osim u slučaju velikog vremenskog čekanja, no u pravom primjeru umrežene igre velike razlike u vještini igrača (više od 200 *Elo-a* npr.)

se često izbjegavaju. Osnovna ideja modela je da algoritam uzima osnovnu, promjenljivu, granicu i pokušava spojiti igrače u meč, ako ne uspije će, s prolaskom vremena, povećavati tu granicu dok ne uspije pronaći meč.

2.2. VR specifikacije

Aplikacija treba podržavati različite VR uređaje, primarno *Oculus Quest.* Korisniku treba opisati i prikazati kontrole te razjasniti što svaki gumb na Slici 2.1 radi. Također treba imati konzistentan način kretanja po sceni. VR je jako pogodan za vizualizaciju algoritma grupiranja i ideja je korisniku pomno opisati i pokazati kako svaki dio algoritma funkcionira.

Slika 2.1: Oculus Quest 2 kontroleri (preuzeto iz)

2.3. Glavni izbornik i korisnička interakcija

Izbornik treba podržavati standardnu VR interakciju sa sučeljem (zrake koje izlaze iz virtualnih ruku u interaktivne gumbe npr.) i sučelje treba moći biti dostupno u bilo koje vrijeme za prijelaz u drugu scenu ili izlazak iz aplikacije.

2.4. Algoritam grupiranja

Algoritam grupiranja treba biti napravljen više sa fokusom na vizualizaciju pojedinih dijelova nego na optimizaciju i uspješnost samog algoritma. Treba podržavati

standardno 1v1 grupiranje i grupiranje velikih timova (npr. 5v5). Sve informacije trebaju biti dostupne igraču za učenje raznih principa grupiranja.

2.5. Kviz znanja

Na kraju aplikacije treba biti dostupan i kratka provjera znanja koja će ispitati korisnika i ocijeniti ga ovisno o uspješnosti. Svako pitanje će imati 4 ponuđena odgovora sa jednim točnim odgovorom i igrač će imati 30 sekundi za odabir odgovora. Kviz treba imati dovoljno pitanja da ispita većinu viđenog iskustva u aplikaciji.

3. Korištene tehnologije

Unity game engine je softverski program za izradu i razvoj videoigara za različite platforme. Razvila ga je tvrtka *Unity Technologies* . Koristi se za razvijanje 2D, 3D, interaktivne simulacije te virtualnu i proširenu stvarnost. Izrada igara je olakšana mnogim ugrađenim sučeljima i funkcionalnostima koje omogućavaju programeru da relativno jednostavno, preko grafičkog sučelja osnovano na komponentama, izgradi svoje programsko rješenje. Dodatne funkcionalnosti se lagano dodaju pomoću ugrađene podrške za stvaranje skripti sa programskim jezikom C# koji je objektno i komponentno orijentiran jezik. *Unity* također nudi svoju vlastitu trgovinu tzv. *Unity Asset Store* koji ima besplatne i plaćene unaprijed napravljene softverske pakete koje su napravili drugi korisnici. (napisati koju verziju imam i to).

XR Interaction Toolkit je interakcijski sustav za virtualnu stvarnost razvijen od strane *Unity Technologies* . XR služi kao osnovni okvir za VR funkcionalnosti. Dopušta kretanje kroz scenu, korisnička sučelja za VR, bacanje i hvatanje objekata, kamera specijalizirana za VR i ostale funkcionalnosti. Također osigurava točno mapiranje kontrola za većinu popularnih VR headsetova.

VR Interaction Framework je paket skinut sa *Unity Asset Storea* koji sadržava mnogo izrađenih primjera za osnovne VR funkcionalnosti . Skripte za kontrolu igrača, korisnička sučelja i ostalo koje su spremne za upotrebu. U sklopu ovog projekta su korištene kao podloga za daljnji VR razvoj.

Microsoft Visual Studio je integrirano razvojno okruženje (engl. *IDE Integrated Development Environment)* . Ima ugrađenu potporu za C# i Unity projekte i u sklopu projekta se koristi za razvijanje skripti. Razvila ga je tvrtka Microsoft te uključuje mogućnosti uređivača koda i uklanjanje grešaka u kodu zajedno sa dodatnim grafičkim i interaktivnim sučeljem.

Oculus Quest i *Oculus Quest 2* su HMD uređaji za virtulnu stvarnost prikazani na Slici 3.1 koji su primarno korišteni za građenje i testiranje ove aplikacije (aplikacija bi trebala raditi na većini VR uređaja). Oba uređaja pružaju samostalan doživljaj virtualne stvarnosti i dolaze sa svim ugrađenim senzorima i kamerama. Za upravljanje i praćenje

ruku imaju dva kontrolera sa gumbima i gljivicom te se pomoću njih mogu imitirati akcije kao hvatanje i bacanje objekata .

Slika 3.1. Oculus Quest 2 (preuzeto iz)

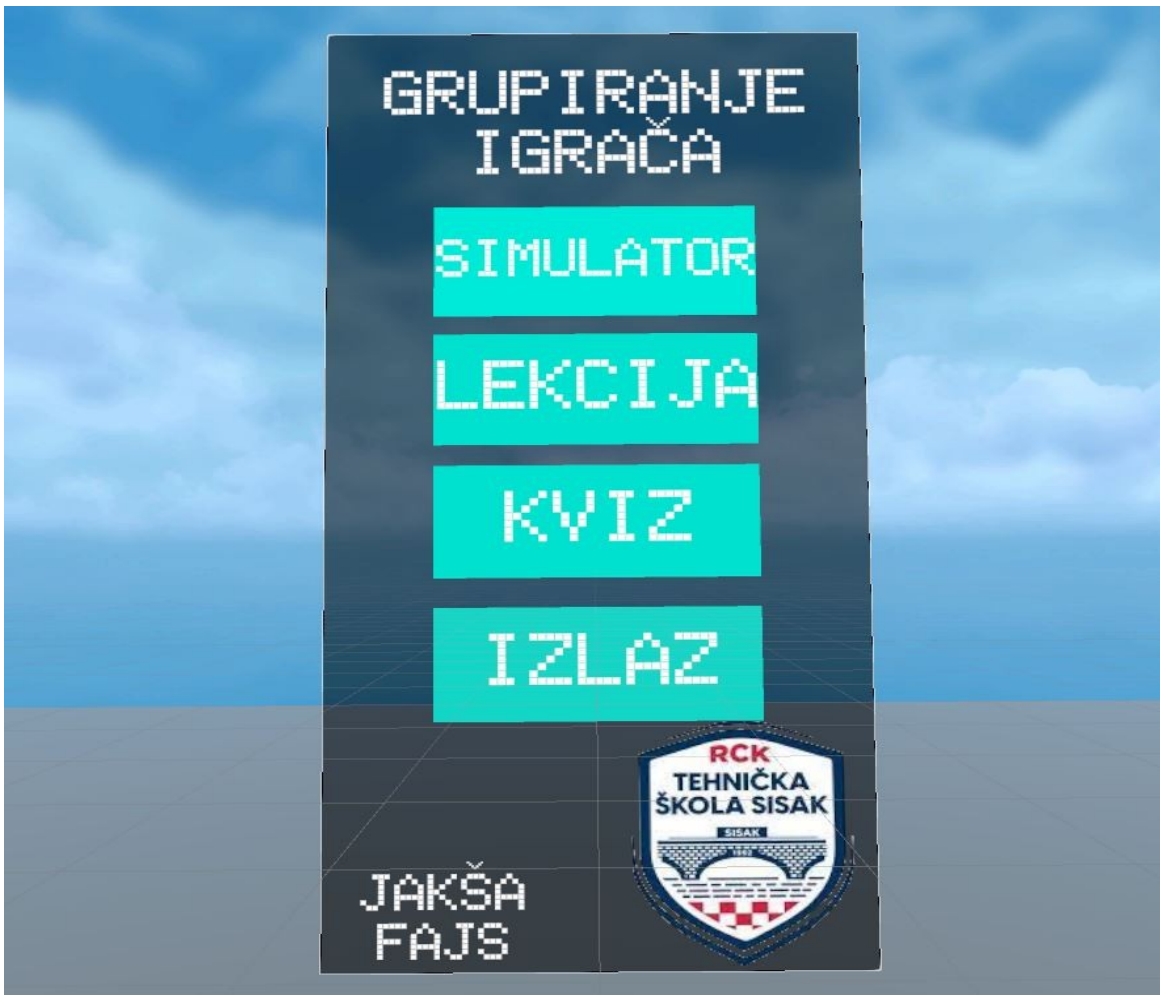
4. Izrada obrazovne aplikacije

U sklopu ovog poglavlja će biti opisana ideja i koncept iza razvijenog projekta. Ova se aplikacija radi u suradnji sa RCK Tehnička škola Sisak, primarni fokus je na vizualizaciji raznih dijelova sustava za spajanje igrača u meč. Glavna scena projekta glumi generaliziranu verziju predvorja (engl. *lobby)* neke igre sa više igrača. Neki od inspiracija su *Among Us ,* prikazan na slici 4.1*, Counter Strike: Global Offensive* itd. Pošto se pokušava odrediti neka struktura koja vrijedi za sve navedene igre, a i igre sa više igrača općenito, koristiti će se faktori koji se mogu primijeniti na sve igre neovisno o njihovom specifičnom kontekstu. Spomenuti faktori su procjena vještine igrača (*Elo*) , kašnjenje obilaska (*ping)* i vrijeme provedeno tražeći meč. Algoritam za grupiranje ima mogućnosti grupiranja različitih brojeva igrača u timu i ima dva glavna načina grupiranja. Projekt naravno nije umrežen već simulira igrače, spajanje igrača u pretraživanje za meč i sam algoritam za spajanje. U sljedećim potpoglavljima ću opisati svaku od tih funkcionalnosti i načine na koji su ostvareni. Scena za spajanje igrača u meč nije jedina u aplikaciji nego ima još par scena koje će se također postupno razraditi kroz sljedeća potpoglavlja. Korisnik se po spomenutim scenama može kretati pomoću funkcionalnosti interaktivnog glavnog VR izbornika.

Slika 4.1: Grupiranje igrača u igri Among Us (preuzeto iz)

4.1 Scena glavnog izbornika

Prva scena s kojom se korisnik susretne je glavni meni aplikacije. On nudi opcije za putovanje po drugim scenama pomoću izbornika. Napravljen je pomoću *VR Canvas* komponente za grafičko sučelje Također ima i upute za korištenje VR-a i ostale specifikacije. Glavni izbornik se nalazi i u drugim scenama i ispunjava funkcionalnost interaktivnog izbornika i prikaza VR kontrola korisniku. Pritiskom *Simulator* se učitava simulator za spajanje igrača u meč. Gumb *Lekcija* učitava kratku lekciju o algoritmima za grupiranje, gumb *Kviz* će korisnika dovesti do kviza znanja te gumb *Izlaz* će izaći iz aplikacije. Učitavanje drugih scena se radi pomoću *SceneLoader* skripte te prikaz dodatnih izbornika za VR kontrole se prikazuje pomoću *Instructions* skripte. Korisnik se ne može kretati po sceni, ali može okretati glavu. Na Slici 4.2. se može vidjeti glavni izbornik i njegove opcije.



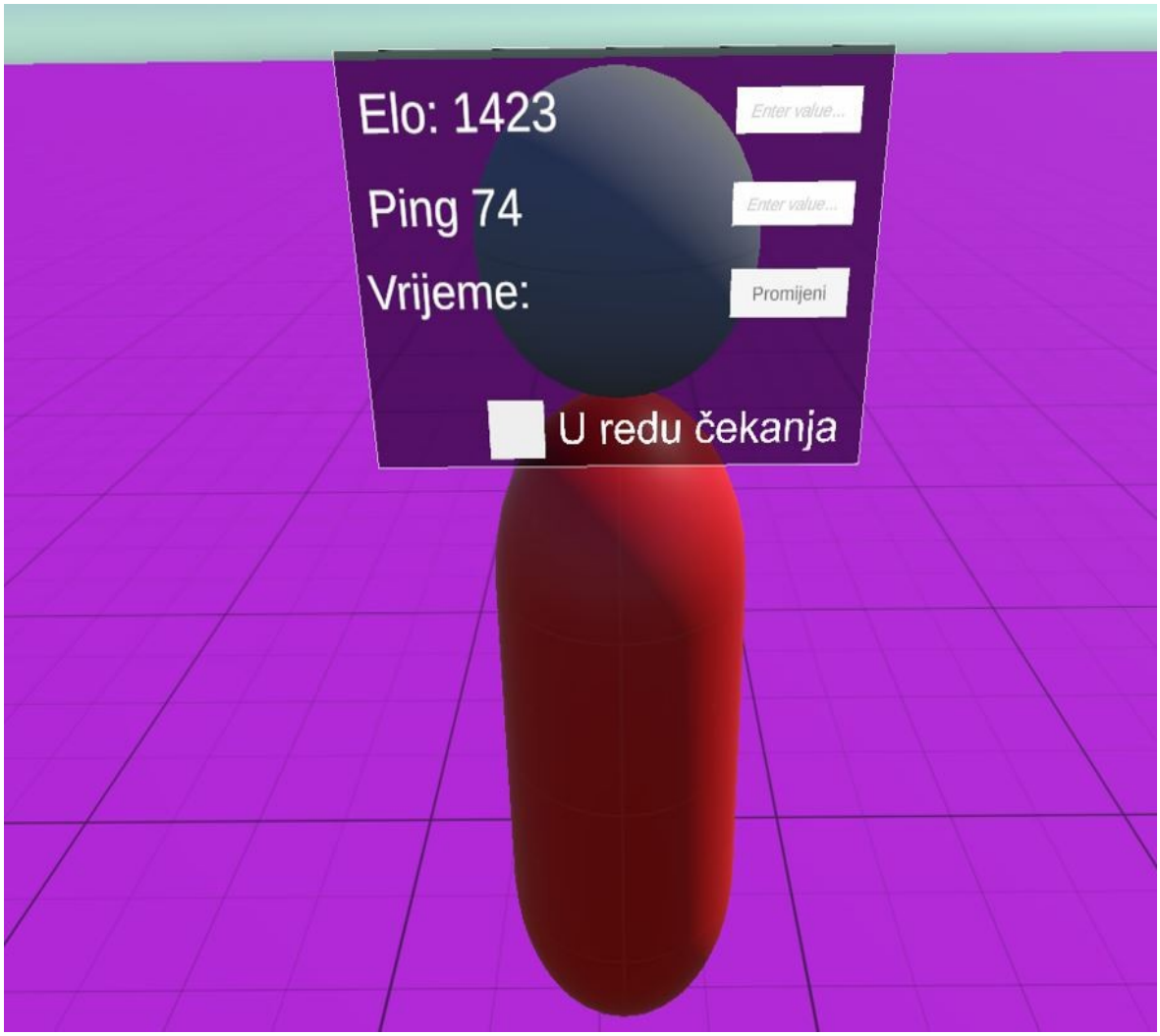
Slika 4.2: Glavni izbornik

4.2 Simulator spajanja igrača u mečeve

Simulator spajanja igrača u meč sadrži sve potrebno za simuliranje algoritma za spajanje igrača u meč. Prva stvar koju će igrač vidjeti je velika platforma sa crvenim natpisom *Red čekanja*. Općenito u aplikaciji crvena boja znači da je nešto isključeno dok zelena boja znači da je uključeno. Ispred platforme je par panela sa s kojima korisnik može rukovati i koje prikazuju informacije o igračima, algoritmu i ostalim zanimljivostima. Na slici 4.3. se može vidjeti glavni podij kojeg korisnik vidi kada se pokrene glavna scena.

Slika 4.3: Queue i kontrole

Pritiskom na gumb *Stvori novog igrača* se stvara nova instanca ispitne lutke koja predstavlja igrača. Lutka na glavi ima sučelje koje predstavlja vrijednosti svake lutke kao *Elo, mrežno kašnjenje i vrijeme provedeno u traženju meča.* Korisnik može mijenjati sve vrijednosti pomoću gumba *Promijeni* stisnuti gumb *U redu čekanja* što će staviti lutku u algoritam za grupiranje kada se i on pokrene. Lutka uspješno imitira bitne karakteristike igrača u višekorisničkoj igri. Neke od C# skripti za opisivanje igrača su *Player Script* koji kontrolira postavljanje i promjenu vrijednosti te pisanje u *Canvas* sučelje. Također ima skripta *Queue Toggle* koji kontrolira promjenu boje lutke i postavljanje u algoritam za grupiranje. Lutka je prikazana na Slici 4.4.

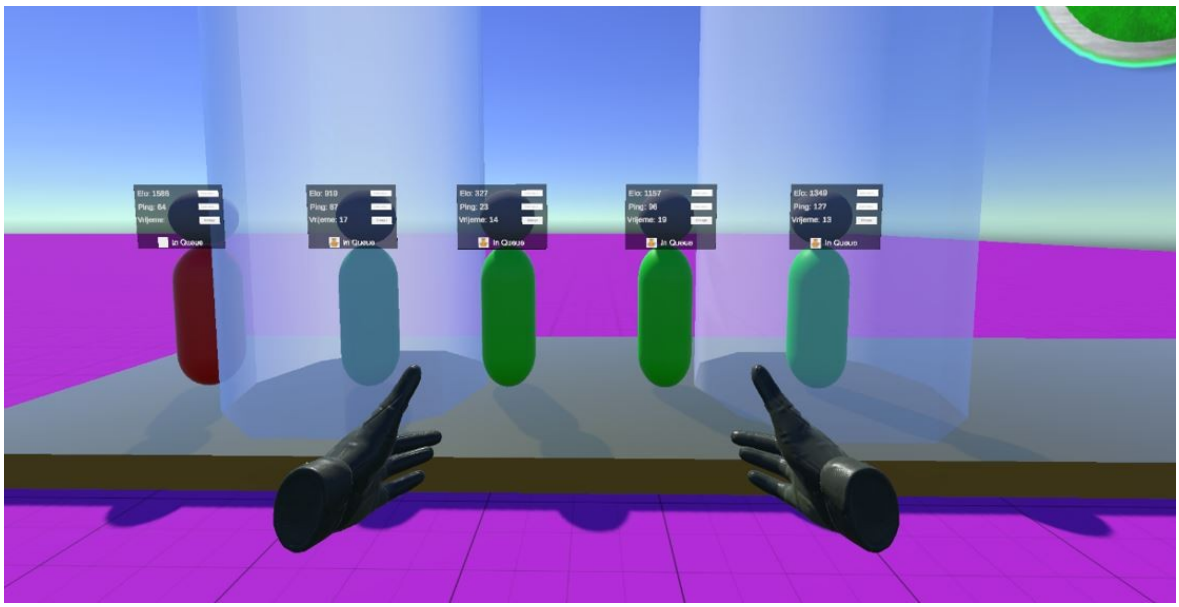


Slika 4.4: Lutka koja imitira igrača

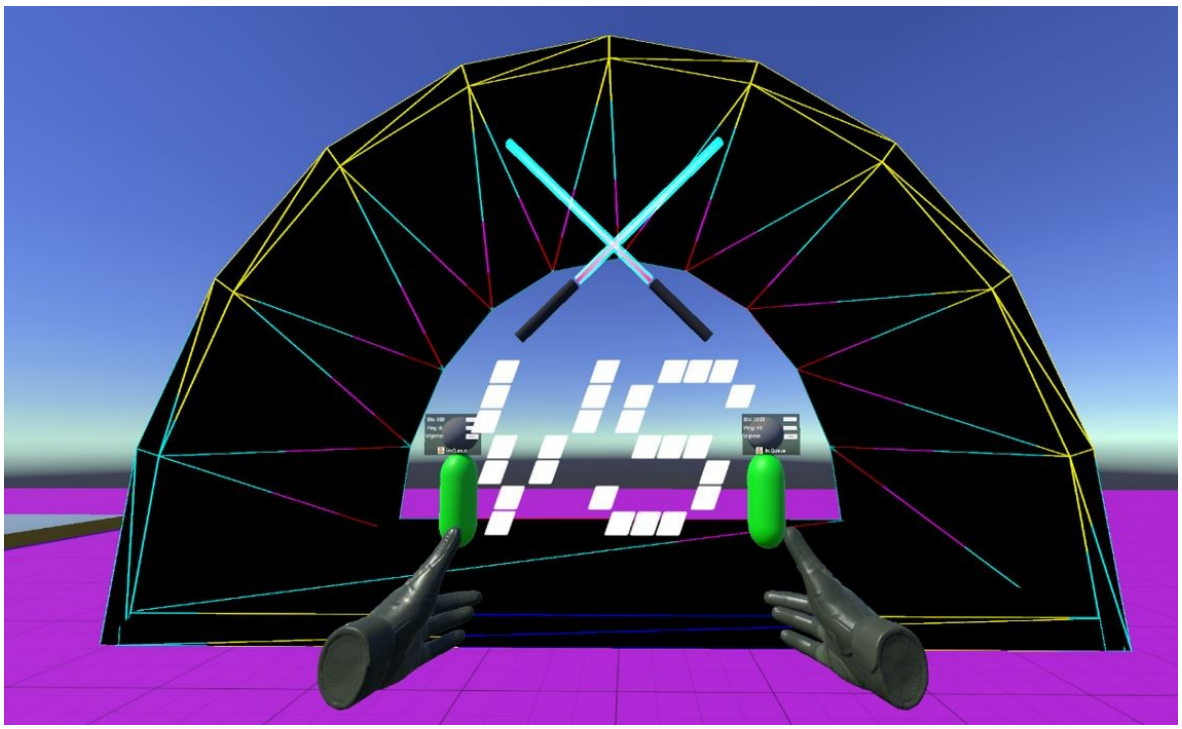
Pritiskom na gumb *Započni Red* se svi aktivni igrači pokušavaju grupirati pomoću skripte *Pokreni Red.* Stvoreni mečevi će biti veličine 1vs1, grupirati će samo dva igrača po meču. Grupiranje je podijeljeno u tri faze: kvalificiranost, provjera i grupiranje. Faza kvalifikacije uspoređuje vrijednosti svakog igrača i uparuje ih ovisno o trenutnim granicama postavljenima u algoritmu. Granice se dinamično mijenjaju ovisno o vremenu provedenom tražeći meč. Faza provjere pokušava pronaći najbolji par (ili više ovisno o postavkama) za svakog igrača. Igrač ima svoju listu potencijalnih parova i algoritam odabire najbolji odabir i ovisno o procijenjenoj kvaliteti meča će ih odmah upariti ili će pričekati određeni broj ciklusa u slučaju da se pojavi bolji kandidat ili se jedan od parova upari sa nekim drugim. Kada sve bude gotovo, velike zrake svjetla će osvijetliti dva odabrana igrača i faza grupiranja će stvoriti simulirani teren koji će instancirati i staviti ta

dva igrača. Tamo korisnik može odrediti tko će pobijediti ili izgubiti i gledati kako to utječe na vrijednosti igrača te ih može vratiti u algoritam grupiranja.

Na Slici 4.5 se može vidjeti jedan primjer grupiranja igrača, trenutno je 5 igrača učitano od kojih 4 traži meč. Nakon nekog vremena provedeno, algoritam je odlučio grupirati igrače 2 i 5. Igrači 3 i 4 nastavljaju tražiti meč. Iznad igrača 2 i 5 se prikaže zraka svijetla koja pokazuje da su grupirani i ubrzo će početi meč. Na slici 4.6 možemo vidjeti taj meč i korisnik može odabrati tko će pobijediti taj meč hodajući do igrača.

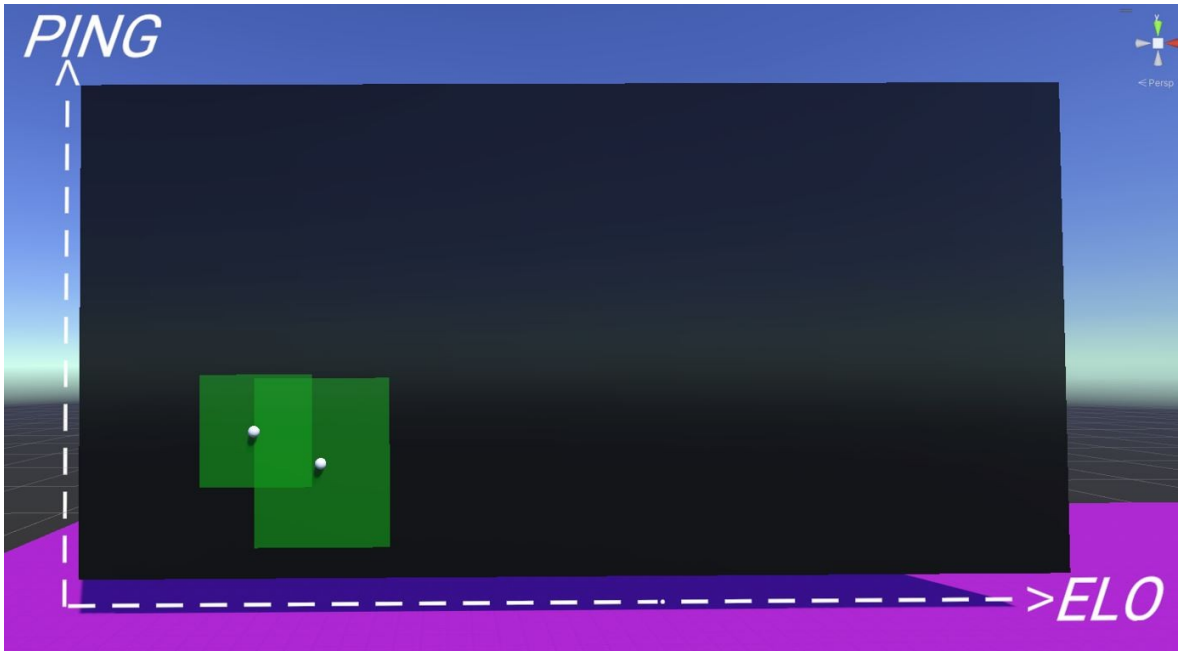


Slika 4.5: Primjer grupiranja igrača



Slika 4.6: Meč između igrača 2 i 5

Iza korisnikove originalne lokacije se može vidjeti jedno veliko platno koje će pomoći pri vizualizaciji igrača koji traže meč. Kada se igrač pojavi na platformi, manja kopija njega će se pojaviti na platnu i staviti će se na lokaciju ovisno o njegovim odabranim faktorima. Bolje rečeno, igrač se postavi na dvodimenzionalni x i y graf. Na tom platnu se može vidjeti granica odlučivanja svakog igrača i njezin postupni rast ovisno o vremenu provedenom u traženju meča. Granica je originalno crvena što znači da algoritam nije pronašao par za igrača. Kada drugi igrač uđe u granicu odlučivanja, obojat će se zeleno i time označiti da postoji mogući par za meč. Vizualizacija se kontrolira pomoću skripte *Qvisualizer* i ona prati sve trenutne vrijednosti i njihove promjene i stavlja ih na graf da korisnik može vidjeti promjenu granica u stvarnom vremenu.



Slika 4.7: Vizualizacija sa dva igrača

Većina objašnjenja je napisana s kontekstom da se radi o 1vs1 igri kao šah, no igre s više igrača mogu imati mnogo veće timove ili veći broj timova. Veći broj timova je trivijalno riješiti sa par dodataka algoritmu no veći broj igrača u timu npr. 5 protiv 5 stvara mnogo veći problem u uparivanju. Ako igra ima dovoljno igrača, prethodno opisana metoda će u razumnom vremenu pronaći dovoljno igrača za puni meč no što ako trenutno igru igra malo igrača ili ih igra puno ali na velikim razlikama u vještini. Algoritam će neke igrače zadržati jako dugo. Popravak ovog problema predstavlja drugi način grupiranja igrača u meč tzv. Naivni način grupiranja. On je također implementiran u sučelju za algoritam te predstavlja alternativni način grupacije gdje ne gleda kvalifikacije igrača nego naivno i brzo uzme potreban broj igrača i pokušava ih najbolje rasporediti u dva tima. U algoritmu se može podesiti broj igrača po timu koji mora biti više od jedan. Kada se promijeni način grupacije postavljeno je na 5 igrača po timu, no korisnik može odabrati proizvoljan broj. Kada pronađe dovoljno igrača u redu čekanja. algoritam prvo uzme dva najbolja igrača i stavi ih u suprotne timove kao „kapetane”. Nakon toga izračunava različite kombinacije igrača i pokušava ih posložiti po timovima tako da prosječni omjer vještine na svakom timu bude što bliži. Postoji granica za omjer vještine između oba tima koju korisnik može promijeniti koja određuje kada će algoritam prestati i započeti meč. Čim algoritam uspije pronaći kombinaciju koja je unutar te granice on će uzeti igrače i poslati ih u meč. Za razliku od prijašnjeg algoritma, naivni algoritam je mnogo brži i

brzopleto spaja igrače u mečeve no zaista je naivan jer samo zato jer je prosječni omjer vještine između blizak ne znači da je meč zapravo pošten za oba tima. Odabir algoritma grupiranja ovisi o kontekstu igre u kojoj se nalazimo i ako je igra jako kompetitivna bolje je koristiti sporiji algoritam, ali ako igra ima jako puno igrača i jako puno mjesta po timu i nije potrebno imati ravnotežu vještina naivni algoritam skoro kompletno eliminira frustraciju čekanja za meč.

Naivno grupiranje igrača se može uključiti sa gumbom *Promijeni Tip Reda* i on se izvodi pomoću skripte *NaiveMatchmaker*. Kada je algoritam za grupiranje u naivnom načinu korisnik dobije opcije za postavljanje broja igrača u timu i granice algoritma.

4.3. Kviz za provjeru znanja

Kviz je implementiran u zasebnoj sceni kojoj se može pristupiti preko glavnog izbornika. Igrač se nalazi u školskoj učionici i kviz će se prikazati na ploči preko *Canvas* elemenata. Kviz se sastoji od 7 pitanja sa četiri ponuđena odgovora i samo jedan je točan. Korisnik ima 30 sekundi za odgovoriti na svako pitanje i na kraju će dobiti ocjenu ovisno o tome na koliko je pitanja odgovorio točno. Učionica se vidi na Slici 4.8.

Slika 4.8: Ucionica za kviz

4.4. Nedostatci i poboljšanja

Aplikacija se može poboljšati sa dodatnim proširenjima u algoritmu za grupiranje. Dodatak faktora za grupiranje kao klase, prijašnji mečevi itd. Algoritam se također može i optimizirati ovisno o bitnijim faktorima i može se dodati podrška za neravnopravne timove npr. 1v4 ili 2v5 no to zahtijeva dodatak kontekstno ovisnih informacija. Dodatne vizualizacije drugih faktora umjesto samo kašnjenje obilaska i *Elo-a* bi se moglo dodati. Proširenje aplikacije na računanje i objašnjenje *Elo* sistema bi isto bilo lagano pošto su teme srodne i trebalo bi samo dodati scenu za vizualizaciju *Elo-a* koji korisnici onda mogu odmah staviti u algoritam za grupiranje.

Zaključak

Obrazovne igre u svrhu učenja i usavršavanja vještina je novi i inovativni način razvijanja znanja. Korisnici se zabavljaju te istovremeno će steći nova znanja koja će im pomoći u životu. Korištenje virtualne stvarnosti poboljšava taj osjećaj jer je korisnik uronjen u virtualno iskustvo i može posvetiti sva osjetila na učenje. Predstavljanje teških i kompliciranih scenarija i koncepata nikada nije bilo lakše nego pomoću edukacijskih igara.

Ovaj rad je razvio jednu takvu aplikaciju za pomoć pri učenju o algoritmima za spajanje igrača u mečeve u videoigrama Aplikacija sadrži dovoljno scenarija za generaliziranu igru da korisnik shvati i nauči osnovne koncepte grupiranja te nudi dodatnu vizualizaciju za svaki korak. Aplikacija grupira igrače po faktorima i prikazuje to korisniku te provjerava naučeno znanje pomoću kratkog kviza točno/netočno.

Aplikacija se može proširiti sa dodatnim načinima grupiranja, faktorima i vizualizacijama za još detaljniji pregled spajanja igrača u mečeve. Pomoću tehnologije virtualne stvarnosti aplikacija bi trebala biti bogato iskustvo koje korisnika uranja u scenarij bolje nego aplikacija bez virtualne stvarnosti.

Literatura

[1] “Virtual Reality” Merriam-Webster.com Dictionary https://www.merriam-webster.com/dictionary/virtual%20reality;

[2] Virtual Reality Society, History of Virtual Reality https://www.vrs.org.uk/virtual-reality/history.html;

[3] GlobalData Technology, History of Virtual Reality: Timeline https://www.verdict.co.uk/history-virtual-reality-timeline/;

[4] Fortune Bussiness Insights, Virtual Reality Market Size, Share & COVID-19 Impact Analysis, https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-market-101378;

[5] Morelo, D. Meet the Sword of Damocles, the First VR Headset in the World https://vrsource.com/meet-the-sword-of-damocles-the-first-vr-headset-in-the-world-17233/;

[6] IMDB, The Matrix, https://www.imdb.com/title/tt0133093/;

[7] Wiltz C, The Story of Sega VR: Sega's Failed Virtual Reality Headset https://www.designnews.com/electronics-test/story-sega-vr-segas-failed-virtual-reality-headset;

[8] Meta.com Oculus Quest 2 https://www.meta.com/quest/quest-pro/;

[9] Shibata T. Head Mounted Display, Displays, Volume 23, Issues 1–2, 2002, Pages 57-64, https://www.sciencedirect.com/science/article/abs/pii/S0141938202000100;

[10] Koutek M., Scientific visualization in virtual reality: Interaction techniques and application development, https://repository.tudelft.nl/islandora/object/uuid:f2ae49ae-d01b-4374-b406-b4c1810ea364;

[11] IrisVR, The Importance of Frame Rates, https://help.irisvr.com/hc/en-us/articles/215884547-The-Importance-of-Frame-Rates;

[12] S. Pandžić I. Sužnjević M. Osnove Virtualnih okruženja – Virtualna Stvarnost (Uvod) https://www.fer.unizg.hr/_download/repository/OVO-P09_Virtualna_stvarnost[2].pdf;

[13] Google VR, Degrees of freedom https://developers.google.com/vr/discover/degrees-of-freedom;

[14] Space Engine https://spaceengine.org/;

[15] The Body VR https://thebodyvr.com/;

[16] Unity Technologies, Unity game engine https://unity.com/;

[17] Microsoft, TrueSkill ranking system, https://www.microsoft.com/en-us/research/project/trueskill-ranking-system/?from=https://research.microsoft.com/en-us/projects/trueskill/&type=exact;

[18] Cornell University, An Analysis of Skill-Based Matchmaking and the Elo Rating System in Video Games, https://blogs.cornell.edu/info2040/2022/09/25/an-analysis-of-skill-based-matchmaking-and-the-elo-rating-system-in-video-games/;

[19] GDC, Skill, Matchmaking, and Ranking Systems Design, https://www.youtube.com/watch?v=-pglxege-gU;

[20] M. Xu, Y. Yu and C. Wu, "Rule Designs for Optimal Online Game Matchmaking," 2021 40th Chinese Control Conference (CCC), Shanghai, China, 2021, pp. 1055-1062, doi: 10.23919/CCC52363.2021.9549977. https://ieeexplore.ieee.org/abstract/document/9549977/authors#authors;

[21] Graepel T., Herbrich R., Ranking and Matchmaking, Game Developer Magazine, https://www.microsoft.com/en-us/research/wp-content/uploads/2006/10/Game-Developer-Feature-Article-Graepel-Herbrich.pdf;

[22] Glickman M., Glicko System http://www.glicko.net/glicko.html;

[23] Q. Deng, H. Li, K. Wang, Z. Hu, R. Wu, L. Gong, J. Tao, C. Fan, and P. Cui. 2021. Globally Optimized Matchmaking in Online Games. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21). Association for Computing Machinery, New York, NY, USA, 2753–2763. https://dl.acm.org/doi/10.1145/3447548.3467074;

[24] XR Interaction Toolkit Manual https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.3/manual/index.html;

[25] VR Interaction Framework https://assetstore.unity.com/packages/templates/systems/vr-interaction-framework-161066;

[26] Microsoft visual Studio https://visualstudio.microsoft.com/;

[27] HTC Vive https://www.vive.com/us/;

[28] Inner Sloth, Among Us https://www.innersloth.com/games/among-us/;

Popis slika