

# Tehnologije navidezne resničnosti za prikaz športnih dogodkov

Nejc Povšič, Matevž Pogačnik

Univerza v Ljubljani, Fakulteta za elektrotehniko

E-pošta: np2758@student.uni-lj.si

## Use of virtual reality technology to display sports statistics

**Abstract.** This paper addresses a mobile application which uses virtual reality technology to display sports matches and statistics. It was designed for the Android mobile operating system with the use of the Unity 3D tool, a graphics engine that allows for very fast and efficient development of graphically intensive games and simulations. The application is created for the ever-present sports market using technology that has taken the field by storm. The application displays a football game in a virtual world in conjunction with some statistics information about both teams and a complete timeline of the event. Interaction is performed with the user's gaze, which controls a cursor that can trigger various actions, such as playing or pausing a video, setting the video player to full screen, selecting a timeline event, etc. This paper explains how the application was created and which methods and technologies were used in its development. It also touches upon the possible future development in the case of a successful productivisation.

## 1 Uvod

Z uporabo navidezne resničnosti smo ustvarili mobilno aplikacijo za gledanje nogometa v virtualnem svetu. Uporabnik je postavljen v umetno ustvarjeno okolje, kjer ima pred seboj zaslon, ki predvaja nogometno tekmo, poleg zaslona pa so postavljeni sezname s podatki o sestavi ekipe in sezname s časovnico tekme, ki vsebuje vse omembe vredne dogodke. Športne statistike za aplikacijo so bile pridobljene od podjetja Sportradar, ki se ukvarja z agregacijo in analizo športnih podatkov. Aplikacija je narejena za operacijski sistem Android z uporabo orodja Unity, igralnega pogona, ki doda vmesni nivo abstrakcije, saj sam poskrbi za računanje v prostoru, izrisovanje in grajenje za določeno platformo, tako da se lahko razvijalec posveti stvarjem, ki se vsebinsko tičejo njegovega projekta. Prav tako ima zelo dobro spisano dokumentacijo in podporo uporabniške skupnosti, zanj pa je ustvarjenih ogromno vtičnikov, ki razvijalcem olajšujejo delo.

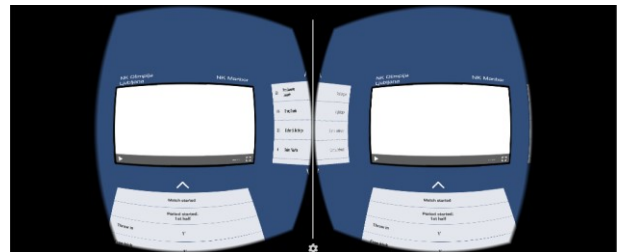
## 2 Uporabljene tehnologije za izvedbo rešitve

### 2.1 Navidezna resničnost

Virtualna ali navidezna resničnost (ang. »Virtual Reality - VR«) je definirana kot računalniška simulacija, pri kateri ima uporabnik občutek, da se nahaja v umetnem

okolju [1]. Kot okno v to okolje uporabimo VR vizir, ki ma vgrajeni dve leči, vsako posebej usmerjeno proti določenemu očesu, naši možgani pa nato iz obeh pogledov sestavijo enotno sliko, ki ustvarja videz, da smo v drugem svetu. Vizir ima lahko zaslon za vsako oko že vgrajena ali pa je narejen tako, da vlogo zaslona prevzamejo zunanje naprave kot so mobilni telefoni, ki jih moramo vstaviti v sam vizir.

Prikaz navidezne resničnosti se od klasičnega prikaza, ki smo ga vajeni, precej razlikuje, kot vidimo na spodnjih slikah 1 in 2. Za razliko od slednjega imamo pri navidezni resničnosti namesto ene kamere, skozi katero opazujemo sceno, sedaj dve. To pomeni, da mora grafična procesna enota celotno sceno izrisati dvakrat in to z večjim vidnim kotom, dvakrat izločiti elemente, ki niso vidni, preslikati tridimenzionalno sceno na dvodimenzionalni zaslon za vsako kamero, na vsaki kameri dodati distorzijo, ki pomaga pri prikazu skozi leče, vse to pa pomeni veliko več dela, ki ga mora računalnik opraviti [2].



Slika 1: Prikaz scene v navidezni resničnosti



Slika 2: Prikaz scene na klasičen način

Da človek zares dobi občutek, da je v navideznem svetu, mora prikaz navidezne resničnosti slediti premikom glave in, če je potrebno, tudi premikom celotnega telesa. Pri tem je zelo pomembna latenca ali zakasnitev, torej čas med sprožitvijo akcije in prejemom odgovora nanjo. Poskrbeti moramo, da je ta čim manjša, da uporabnika ne zmoti in mu ne uniči predstave, saj lahko v skrajnem primeru vodi tudi do simulacijske bolezni (slabost, izguba orientacije) [3].

Da ohranjamo občutek navidezne resničnosti, je zelo pomembna tudi ločljivost zaslona, ki ga opazujemo. Obraz imamo namreč zelo blizu ekrana, zato se lahko

zgodí, da vidimo posamezne slikovne elemente (piksle), kar je pri virtualni resničnosti skrajno nezaželjeno.

Računalnik opravi vse potrebne izračune in izriše sceno na zaslon, a za prikaz potrebujemo neko napravo, da dve različni sliki, ki sta na zaslonu, prikažemo tako, da ju naši možgani lahko zlijejo. Tu pride v poštev že omenjeni VR vizir.

Zelo pomembna komponenta VR vizirjev so leče. Te imajo dve glavni nalogi: dovolj povečati prikaz zaslona, tako da ne vidimo robov, in ustvariti gorišče v neskončnosti [4]. Objekti na zaslonu, ki je za lečo, tako izgledajo oddaljeni dlje, kot dejansko so. Leče tudi popačijo sliko, ki jo dobijo od zaslona, zato ji je treba na zaslonu dodati inverzno popačitev, da jo naše oči vidijo čim bolj resnično [4].

## 2.2 Unity

Unity je platforma za razvoj iger, v katerem lahko na preprost način ustvarjamo tri- ali dvodimenzionalne računalniške igre s pomočjo naj sodobnejših tehnik in tehnologij, ki so na voljo. Ideja Unity-ja je bila, po besedah ustvarjalcev, »demokratizacija razvijanja iger«, kjer bi lahko vsakdo, ne glede na poklic in izobrazbo, ustvarjal računalniške igre [5]. Uporablja ga ogromno število razvijalcev, saj je enostaven za uporabo, ima zelo dobro dokumentacijo, veliko skupnost, je zelo zmogljiv, dobro podprt in razširljiv. Zanj je ustvarjenih mnogo vtičnikov, ki uporabnikom še dodatno olajšajo delo, čeprav jih je večina plačljivih. Z Unity-jem lahko zgradimo igro za 27 različnih platform, med katerimi je tudi Android.

Unity podpira dva programska jezika, v katerih lahko programiramo, C#, ki je uraden programski jezik orodja in jezik, v katerem je logika aplikacije spisana, in Javascript, ki pa ga počasi umikajo. Vse skripte za delovanje v Unity-ju so sestavljene iz dveh glavnih delov, iz funkcije *Start*, ki se zažene takoj, ko je skripta dodana v sceno, in iz funkcije *Update*, ki se požene pri vsakem izrisu nove sličice.

Velika prednost Unity-ja je razširljivost, saj omogoča, da v projekt dodajamo razne vtičnike. Za zeleno delovanje aplikacije smo jih morali nekaj dodati. Med njimi so bili GoogleVR, ki ga je ustvaril Google za potrebe platform Daydream in Cardboard ter je dodal nek nivo abstrakcije, saj je v ozadju poskrbel za prikaz virtualne resničnosti in pomagal pri gradnji za Android operacijski sistem, vtičnik za predvajanje posnetkov Easy Movie Texture, ki omogoča predvajanje videov v prostoru na poljubnem objektu, in vtičnik za interpretiranje oblike JSON v programskem jeziku C#.

## 2.3 Android

Android je mobilni operacijski sistem za pametne telefone, ki je zgrajen na Linuxovem jedru. Prva različica je bila izdana leta 2008 in je bila na voljo le za peščico mobilnih naprav, ker pa je Android odprtokodni sistem, ga je kmalu začelo uporabljati vse več

proizvajalcev mobilnih naprav. Tako je počasi pridobil na popularnosti in je do danes postal najbolj razširjen mobilni operacijski sistem.

## 2.4 Strojna oprema

Za prikaz aplikacije smo potrebovali Android telefon in VR vizir, v katerega smo lahko mobilno napravo vstavili. Uporabljen je bil telefon LG G6, ki vsebuje vse potrebne senzorce in je dovolj zmogljiv, da aplikacijo poganja učinkovito in brez zatikanja. Prikazan je na sliki 3.



Slika 3: Mobilni telefon LG G6

Da virtualna resničnost lahko deluje, mora telefon nekako zaznati svoj položaj in orientacijo v prostoru. Moderne naprave to dosegajo s pospeškomerom in žiroskopom.

Pospeškomer (ang. »accelerometer«) je, kot ime nakazuje, zmožen meriti veličino in smer pospeška ter s tem tudi sile, ki vplivajo nanj. Te so lahko dinamične, kot so recimo sile pri premikanju naprave, ali statične, kot je Zemljina gravitacija. Zmožen je zaznati tudi blage spremembe rotacije naprave, ampak je pri tem zelo nenatančen, potrebuje pa tudi pomoč vgrajenega kompasa [6]. Android zna iz pospeškometra pridobiti veličino pospeška v vseh treh oseh (prišteto skupaj z gravitacijo ali samo linearni pospešek brez gravitacije).

Žiroskop (ang. »gyroscope«) je naprava za merjenje kotne hitrosti naprave, ki je predstavljena v stopinjah ali radianih na neko časovno enoto. S tem omogoča zaznavanje hitrosti rotacije naprave v vseh treh oseh, s čimer lahko zelo točno določimo orientacijo telefona v prostoru.

Za VR vizir smo izbrali preprost VR BOX, ki ima prostor v katerega vstavimo telefon in dve leči, ki jih lahko premikamo naprej, nazaj, levo in desno, da si pravilno nastavimo razdaljo do oces in med ocesoma. Če imamo razdaljo pravilno nastavljen, lahko naši možgani brez problemov sliki združijo v prostorsko celoto. Vizir ne vsebuje nobenih senzorjev, saj vse delo opravi telefon s svojimi senzorcji. Viden je na sliki 4.



Slika 4: VR BOX vizir

### 3 Športni podatki

Športne podatke, uporabljene v aplikaciji, smo pridobili od podjetja Sportradar, ki se ukvarja z urejanjem in analizo športnih podatkov za večino obstoječih športov v okviru katerih potekajo tekmovanja. Za najpopularnejše, kot je npr. nogomet, imajo na voljo več vrst podatkov, kot so celotni urnik turnirjev, statistika ekip, statistika posameznih igralcev, primerjava dveh ekip, statistika tekem, itd.

Največji del aplikacije in njen namen je seveda prenos nogometne tekme. Za potrebe demonstracije naše rešitve je bil izbran posnetek tekme med Mariborom in Olimpijo, ki se je odvijala 8. avgusta 2016.

#### 3.1 Športni trg

Šport je gonilo za ogromno novih tehnologij, predvsem na področju multimedije. V prenosih športnih dogodkov se že nekaj časa uporablja obogatena resničnost (npr. namišljena pot žoge pri nogometu), vedno bolj pa se večja zanimanje tudi za navidezno resničnost.

Pojavili so se že 360 stopinjski posnetki iz tekem za navidezno resničnost, ki gledalca postavijo na sam štadion (tudi za svetovno prvenstvo v nogometu 2018 so za določene tekme na voljo prenosi, namenjeni naglavnim zaslonom). Nekateri so virtualno resničnost uporabili kot orodje za analizo preteklih tekem ali za učenje novih spretnosti [7]. Možnosti za implementacijo navidezne resničnosti je na športnem trgu ogromno in naša rešitev je le ena izmed mnogih, ki so na voljo.

### 4 Rešitev virtualnega prikaza tekem in statistik

#### 4.1 Funkcionalnosti aplikacije

Aplikacija vsebuje samo dve sceni, med katerima lahko menjamo. Po končani inicializaciji in začetnem zaslonu se odpre prva scena, v kateri je v prostoru dodan samo gumb za odpiranje druge scene. V osnovi aplikacija omogoča ogled športne tekme s prikazom statistik tekme, ekip igralcev in seveda rezultat.

Za uporabo aplikacije potrebujemo VR vizir, čeprav deluje tudi brez njega. Trenutno je omogočeno samo rotacijsko sledenje glavi, kar pomeni, da se ne moremo premikati po navideznem prostoru, ampak sceno opazujemo le iz ene točke.

Ker je aplikacija narejena za VR naglavni sistem brez dodatnih gumbov, je krmiljenje zelo omejeno, saj imamo na voljo le pogled. Edini način interakcije z okoljem je usmerjanje pogleda na interaktivne objekte, ki akcijo sprožijo instantno ali z nekim časovnim zamikom. Ko s kazalcem pristanemo na predmetu, ki ima nase vezano neko akcijo, se, če je bilo določeno, pokaže odštevalnik časa. Če takrat, ko se odštevanje konča, še vedno opazujemo isti objekt, se akcija izvede. Ta sistem je za navidezno resničnost brez dodatnih krmilnikov najbolj ustrezen.

Ko aktiviram gumb v prvi sceni, se odpre druga, ki vsebuje prikaz tekme in podatkov. Postavljeni smo v nek prostor, kjer imamo postavljena 4 platna. Na sredini je zgoraj predvajalnik video posnetka, spodaj časovnica, levo in desno pa je prikaz postavitve obeh ekip, kot je prikazano na sliki 5.



Slika 5: Druga scena aplikacije

Tudi tukaj ima zelo veliko vlogo kazalec, saj zoper ga zopet potrebujemo za interakcijo z objekti v okolju. Vsak interaktivni objekt, ki ga pogledamo, povzroči, da se kazalec poveča, kar nakazuje, da je z njim povezana neka akcija. Na sezname so dodane puščice, ki ob pogledu premikajo seznam navzgor ali navzdol, saj je podatkov preveč, da bi jih prikazali naenkrat.

#### 4.2 Arhitektura

Za potrebe aplikacije smo postavili centralen strežnik, kjer so športni podatki zapisani v mongoDB podatkovni bazi. Da smo podatke lahko posredovali do aplikacije, smo ustvarili REST API končne točke, na katere smo potem izvajali klice s protokolom http (ang. »Hypertext Transfer Protocol«). Aplikacija posnetek pridobiva iz strežnika z uporabo protokola HLS (ang. »Http Live Streaming«).

Kot omenjeno, sta v aplikaciji na voljo dve sceni. Prva ima vlogo začetnega menija, kjer lahko trenutno samo zaženemo drugo sceno, a se gumbi lahko prosto dodajajo. Njena hierarhija je zato zelo preprosta, saj vsebuje le pred nastavljen objekt za kamero, globalno luč, prazen objekt, na katerega vežemo skripte in pred nastavljen objekt, ki vsebuje interaktivne gumbe. Pred nastavljeni objekti oz. *prefabi*, so v Unity-ju zelo koristni, saj nam dovolijo, da vstavimo identičen objekt v več scen, ali pa ga vstavimo večkrat v sklopu posamezne funkcije.

Na objekt kamere je seveda vezan tudi kazalec, ki se skupaj s njo transformira, tako da vedno stoji točno na sredini pogleda. Deluje na principu sledenja žarku (ang.

»Ray tracing«), ki je način preverjanja kolizij med žarkom in objekti. Predstavljajmo si ga, kot da iz našega pogleda v smeri kazalca sije žarek, ki seka vse predmete na njegovi poti. Da vpliva na nek predmet, moramo temu dodati komponento za detekcijo trkov in poslušalec dogodkov, ki zajame dogodek, ko se ta sproži.

Zraven kazalca smo dodali tudi objekt za časovno zamaknjeno akcijo. Pri nekaterih objektih nismo želeli, da se akcija izvede takoj, ker lahko uporabnik nenamerno pogled usmeri v objekt. Da bi si olajšali delo, smo dodajanje števca hoteli narediti čim bolj enostavno, saj ga je bilo potrebno dodati na kar nekaj objektov.

Ko se druga scena zažene, se takoj na začetku izvede klic na strežnik za podatke, v skripto, ki skrbi za nalaganje posnetka, pa se nastavi URL naslov video posnetka. Klic na strežnik se izvede v posebni funkciji, ki jo omogoča Unity, imenovani *Coroutine*. Ta poskrbi, da izvajanje ni vezano na posodobitev okvirjev v *Update* funkciji, ampak teče vzporedno in je pri vsaki posodobitvi zmožna nadaljevati od tam, kjer je končala izvajanje. Ker je pridobivanje podatkov iz strežnika asinhrona operacija in nočemo, da nam prekine delovanje aplikacije dokler traja, je uporaba tega tipa funkcije nujna.

Ko športni podatki pridejo do odjemalca, se sezname napolnijo z njimi. Za vsak element se v sceno doda nov pred nastavljen objekt, ki vsebuje skripto za nadzor vseh komponent za prikaz besedila, ki se v Unity-ju imenuje *Text*. Glede na tip podatka (časovnica ali postavitev) smo ustvarili dva pred-nastavljena objekta, ki imata seveda drugačno postavitev.

Video posnetek se med tem prav tako naloži preko spleta na izbrano platno z uporabo Easy Movie Texture vtičnika. Uporaba je zelo preprosta; sam vtičnik nam da dostop do poslušalcev, ki opazujejo ali se je posnetek naložil, ali je prišlo do napake, itd. Javno dostopne so tudi funkcije, ki skrbijo za ustavljanje ali predvajanje posnetka, za premikanje po posnetku in ugotavljanje časa trajanja ter koliko časa je že preteklo pri predvajanju.

### 4.3 Problemi pri razvoju

Pri razvoju rešitve smo naleteli na nešteto težav, ki smo jih morali rešiti. Ena izmed največjih, vsaj v sklopu virtualne resničnosti, je bila učinkovitost aplikacije. Narejena je za mobilne naprave, ki so zelo omejene v procesorski in grafični moči. S tem, ko smo zmanjšali podrobnosti, izključili sence, zmanjšali ločljivost tekstur in uporabili materiale, ki niso odbijali svetlobe, smo občutljivo izboljšali delovanje aplikacije in zmanjšali porabo baterije.

Glavni del aplikacije, prikazovanje tekme na nekem samostojnem zaslonu v okolju, bi bilo na Android napravi zelo težko implementirati, saj Unity privzeto podpira samo prikazovanje video posnetka čez celoten zaslon. Rešitev smo našli v vtičniku Easy Movie

Texture. Z njim je bil prikaz tekme zelo nezahteven proces, saj podpira tudi strujanje preko http protokola.

## 5 Zaključek

V članku je opisana izdelava delujočega prototipa mobilne aplikacije, ki je osnovana na virtualni resničnosti. Namenjena je športnim navdušencem, ki bi radi tekme spremljali na nov način z uporabo najnovejših tehnologij računalniške grafike.

Seveda aplikacija še ni končna. Trenutno je rešitev implementirana samo za točno določeno tekmo, dalo pa bi se jo razširiti, da bi podpirala več tekem in turnirjev. Dodati bi se morale nove končne točke na strežniku in omrežni klici iz naprave, ki bi do teh točk dostopale. Podatki bi se tudi morali sami posodabljeni.

Sam grafični vmesnik je dokaj preprost in vsebuje samo komponente, ki so nujne za delovanje, ker je aplikacija omejena s strojno opremo telefonov, zato je tudi izgled sveta zelo generičen in dokaj pust. Z razvojem mobilnih naprav se bo izboljševala tudi strojna oprema, kar pomeni več procesorske in grafične moči ter posledično več prostih rok pri ustvarjanju virtualnega sveta.

## Literatura

[1] Navidezna resničnost. Dosegljivo: [https://sl.wikipedia.org/wiki/Navidezna\\_resni%C4%8Dnost](https://sl.wikipedia.org/wiki/Navidezna_resni%C4%8Dnost) [Dostopano: 3.4.2018]

[2] Stereoscopic Rendering in WebVR. Dosegljivo: <https://hacks.mozilla.org/2015/09/stereoscopic-rendering-in-webvr/> [Dostopano 6.5.2018]

[3] The Technical Challenges of Virtual Reality. Dosegljivo: <https://iq.intel.com/the-technical-challenges-of-virtual-reality/> [Dostopano 6.5.2018]

[4] VR Lens Basics: Present And Future. Dosegljivo: <https://www.tomshardware.com/news/virtual-reality-lens-basics-vr,36182.html> [Dostopano: 18. 6. 2018]

[5] Unity (game engine). Dosegljivo: [https://en.wikipedia.org/wiki/Unity\\_\(game\\_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)) [Dostopano: 23.5.2018]

[6] D. Avguštin, Analiza tresljajev s senzorji v mobilnih napravah. Dosegljivo: [http://eprints.fri.uni-lj.si/2269/1/Avgu%C5%A1tin\\_D%2D1.pdf](http://eprints.fri.uni-lj.si/2269/1/Avgu%C5%A1tin_D%2D1.pdf) [Dostopano: 28.5.2018]

[7] How virtual reality is transforming the sports industry. Dosegljivo: <https://techcrunch.com/2016/09/15/how-virtual-reality-is-transforming-the-sports-industry/> [Dostopano: 18. 6. 2018]