Simulacija imerzivnog sportskog prijenosa: gledanje utrke Formule 1 u dograđenoj stvarnosti

Gustetić, Filip

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:168:566318

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-22**

Repository / Repozitorij:

FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repozitory

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 526

SIMULACIJA IMERZIVNOG SPORTSKOG PRIJENOSA: GLEDANJE UTRKE FORMULE 1 U DOGRAĐENOJ STVARNOSTI

Filip Gustetić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 526

SIMULACIJA IMERZIVNOG SPORTSKOG PRIJENOSA: GLEDANJE UTRKE FORMULE 1 U DOGRAĐENOJ STVARNOSTI

Filip Gustetić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 526

Pristupnik:	**Filip Gustetić (0036526151)**
Studij:	Računarstvo
Profil:	Programsko inženjerstvo i informacijski sustavi
Mentorica:	prof. dr. sc. Lea Skorin-Kapov

Zadatak: **Simulacija imerzivnog sportskog prijenosa: gledanje utrke Formule 1 u dograđenoj stvarnosti**

Opis zadatka:

Kroz upotrebu tehnologije dograđene stvarnosti (engl. eXtended Reality, skr. XR), korisnicima se pruža prilika da svoje fizičko okruženje proširе virtualnim elementima. Ova tehnologija sve više nalazi primjene u raznim područjima, kao što je interaktivna 3D vizualizacija simulacije nekog događaja ili procesa. Primjer popularnog natjecateljskog događaja su automobilske utkre, koje gledatelji mogu pratiti i doživjeti na razne načine, poput uživo ili putem 2D ekrana. Primjenom tehnologije XR moguće je ostvariti imerzivan način gledanja utrke. Na primjer, korisnici mogu vidjeti virtualne automobile koji se "natječu" u njihovoj stvarnoj okolini, što potencijalno stvara intenzivniji doživljaj simulirane utrke. Vaš zadatak je osmisliti i implementirati XR aplikaciju koja omogućuje korisnicima prikaz i interakciju s virtualnom reprezentacijom simulacije utrke Formule 1. Simulaciju je potrebno ostvariti na temelju stvarnih podataka prikupljenih pomoću dostupnih aplikacijskih programskih sučelja. Uz prikaz 3D modela, potrebno je u stvarni prostor postaviti virtualne 2D elemente s dodatnim sadržajima vezanim uz utrku. Nadalje, aplikacija treba omogućiti korisniku prelazak iz miješane stvarnosti (u kojoj doživljava virtualnu stazu i bolide kako se po njoj kreću, smještene u stvarni prostor) u virtualnu stvarnost (u kojoj doživljava utrku iz druge perspektive).

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

Uvod 1

1. Imerzivne tehnologije 2

1.1. Virtualna stvarnost 3

1.2. Proširena i miješana stvarnost 4

1.2.1. Miješanje slike 5

1.2.2. Prikaz rezultata miješanja 7

1.2.3. Poravnanje 7

1.3. Dograđena stvarnost 8

1.4. Imerzivne tehnologije u sportskim prijenosima 10

2. Opis aplikacije *Formula XR* 12

2.1. Cilj i motivacija 12

2.2. Korištene tehnologije 14

2.2.1. Meta Quest 3 14

2.2.2. FastF1 15

2.2.3. Visual Studio 2022 16

2.2.4. Unity 16

2.2.5. Meta Interaction SDK 18

3. Izrada aplikacije *Formula XR* 19

3.1. Simulacija utrke 19

3.1.1. Dohvat podataka uporabom FastF1 paketa 19

3.1.2. Razvoj Unity scene 22

3.2. Integracija XR elemenata u aplikaciju 26

3.2.1. Konfiguracija XR scene 26

3.2.2. Interakcija korisnika s virtualnom utrkom 28

3.2.3. Pogled iz perspektive bolida 30

3.2.4. Prijelaz u virtualnu stvarnost 33

4. Prikaz rada aplikacije *Formula XR* 38

5. Analiza aplikacije i moguća poboljšanja 42

5.1. Problemi pri izvedbi 42

5.2. Moguća poboljšanja 44

Zaključak 45

Literatura 46

Sažetak 48

Summary 49

Skraćenice 50

Uvod

Formula 1 (skr. F1) je dinamičan sport u kojem se dvadeset vozača kroz dvadesetak utrka u godini utrkuje za titulu najboljeg vozača godine. Formula 1 je skup sport - izrada svakog od trkaćih automobila, odnosno bolida Formule 1 košta otprilike 16 milijuna eura [25]. Osim za natjecatelje, sport je skup i za organizatore. Cilj Formule 1 kao organizacije je pridobiti što veću publiku kako bi ostvarili što veće profite. Najveći udio navedene publike prati F1 utrke preko televizijskih prijenosa. U 2021. godini, prijenose utrka ukupno je pratilo više od 1.5 milijardi gledatelja [26], a taj je broj od tada još rastao. Kako bi zadržali postojeće i privukli nove gledatelje, organizatori utrka i režiseri TV prijenosa nastoje učiniti prijenos što zanimljivijim pa tako u prijenos prije same utrke ubacuju intervjue s poznatim osobama, prikaz događanja „iza kulisa“ i sl. Sam prijenos utrke obogaćuju virtualnim grafikama koje pomažu gledateljima u praćenju utrka i shvaćanju ključnih podataka koji utječu na razvoj utrke, što je najkorisnije novim gledateljima.

U sklopu ovog rada izrađena je aplikacija nazvana *Formula XR* koja omogućuje praćenje F1 utrke uporabom virtualne reprezentacije staze na kojoj se utrkuju virtualni bolidi. Virtualna utrka temelji se na stvarnim podacima. Zbog toga, aplikaciju je moguće koristiti uz stvarni prijenos te tako dobiti informacije o točnim pozicijama bolida na stazi u svakom trenutku utrke, što dodatno proširuje pogled na utrku jer takve informacije ponekad nisu dostupne u samom prijenosu. Aplikacija je razvijena uporabom tehnologija dograđene stvarnosti (engl. *eXtended Reality*, skr. XR). Dograđena stvarnost je skup tehnologija koje omogućuju imerziju korisnika u kompletnu virtualnu okolinu ili u stvarnu okolinu proširenu virtualnim elementima.

Rad je podijeljen na pet poglavlja. U prvom je poglavlju predstavljena dograđena stvarnost i ostale imerzivne tehnologije povezane s njom te njihova uloga u sportskim prijenosima. U drugom poglavlju detaljno je opisana motivacija i cilj aplikacije *Formula XR*. U ovom su poglavlju također predstavljene tehnologije koje se koriste pri izradi aplikacije. Sam proces izrade aplikacije opisan je u trećem poglavlju, a upute za njeno korištenje u četvrtom. Na kraju rada nalazi se poglavlje koje opisuje probleme pri izvedbi i potencijalna proširenja te zaključak, popis literature i popis skraćenica.

1. Imerzivne tehnologije

U Cambridgeovom rječniku, pridjev imerzivan [1] (engl. *immersive*) opisan je kao privid u kojemu se publika ili igrač osjeća u potpunosti uključeno u određenu aktivnost. Na primjer, cilj glumaca, scenarista, kostimografa i ostalih djelatnika kazališta je stvoriti imerzivnu predstavu koja će gledatelje toliko zanijeti da će zaboraviti da se zapravo nalaze u kazalištu, a ne u mjestu gdje je smještena radnja predstave. Analogna situacija je i s imerzivnim tehnologijama. Imerzivne tehnologije predstavljaju integraciju virtualnog sadržaja s fizičkim svijetom na način koji omogućuje korisniku prirodno i intuitivno iskustvo u pomiješanoj okolini [2]. Slično kao u primjeru s kazališnom predstavom i publikom, cilj imerzivnih tehnologija je zamagliti granicu između stvarne i virtualne okoline te raznim tehnikama i uređajima korisnika u potpunosti „uroniti“ u simuliranu okolinu te pružiti mu osjećaj prisutnosti u njoj [3].

Primjena imerzivnih tehnologija je široka [4]. Koristi se za edukaciju i provođenje raznih treninga u područjima medicine, biologije i sličnih znanosti, ali i u vojsci. Imerzivne tehnologije sve se češće koriste i u industriji zabave, u formi videoigara i filmova. Također, koristi se i u područjima kao što su dizajn, arhitektura, turizam, marketing i još mnogim drugim.

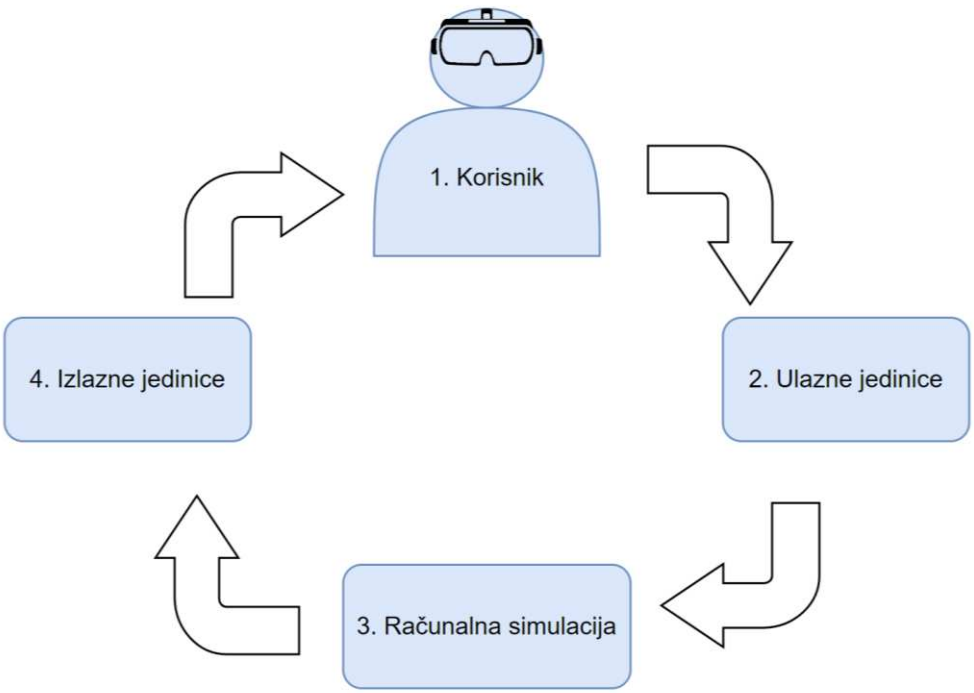
U literaturi, tehnologije virtualne (engl. *Virtual Reality*, skr. VR), proširene (engl. *Augmented Reality*, skr. AR) i miješane engl. *Mixed Reality*, skr. MR) stvarnosti najčešće se spominju kao primjeri imerzivnih tehnologija [2]. Osim navedenih, po definiciji, pod imerzivne tehnologije spada i dograđena stvarnost, ali i tehnologije poput 3D filmova, videozapisa dostupnih u 360 stupnjeva, kina u kojima se sjedala pomiču po potrebi i sličnih tehnologija. Aplikacija izrađena u sklopu ovog rada smatra se aplikacijom dograđene stvarnosti. Kako bi se objasnio pojam dograđene stvarnosti, najprije se treba razumjeti virtualna, proširena i miješana stvarnost.

1.1. Virtualna stvarnost

Virtualna stvarnost podrazumijeva skup računalnih simulacija te raznih ulaznih i izlaznih uređaja kojima je cilj korisnicima stvoriti osjećaj prisutnosti u virtualnom okruženju [4]. Korisnik koji koristi tehnologije virtualne stvarnosti iz svoje je perspektive u potpunosti okružen virtualnim svijetom i raznim virtualnim objektima. Pritom korisnik treba imati što manji dodir sa stvarnosti, odnosno sa stvarnom okolinom i stvarnim objektima u njoj. U idealnom slučaju, dok koristi tehnologije virtualne stvarnosti, korisnik nema nikakav dodir sa svojom stvarnom okolinom.

Za korisnika koji koristi VR tehnologije, figurativno se kaže da se nalazi u zatvorenoj petlji virtualne stvarnosti [4] (slika 1.1). Korisnički pokreti, geste, glasovne naredbe i slični ulazni podaci prate se preko ulaznih uređaja virtualne stvarnosti. Postoje razne vrste ulaznih uređaja, od kojih su najčešći senzori pozicije i senzori pozicije tijela. Senzori pozicije prate položaj korisnika u stvarnome prostoru, a mogu biti optički, mehanički, magnetni i sl. Senzori položaja tijela prate promijene položaja određenog dijela korisnikovog tijela. Primjeri senzora položaja tijela su kontroleri i VR rukavice koje prate položaj ruku korisnika i VR odijela koja mogu pratiti pomake cijelog tijela. Navedenim senzorima može se pratiti tri ili šest stupnjeva slobode (engl. *degrees of freedom*, skr. DoF) [4]. Kada uređaji prate mijenjanje kuta gledanja korisnika, oni prate tri stupnja slobode, a kada se uz mijenjanje kuta gledanja prate i promjene položaja korisnika u prostoru, onda uređaji prate šest stupnjeva slobode. Postoje i drugi senzori kao što su senzori sile i senzori pokreta, poput traka za trčanje na mjestu, no zbog nepraktičnosti i visokih cijena nisu popularni i česti u VR tehnologijama kao ostali navedeni senzori.

Navedeni senzori i uređaji ulazne podatke korisnika šalju računalu na obradu [4]. Računalo obrađuje ulazne podatke i stvara računalnu simulaciju virtualne stvarnosti koju prezentira korisniku preko izlaznih uređaja. Cilj izlaznih uređaja je prenijeti podražaje iz računalne simulacije na korisnika pa tako postoje vizualni, zvučni, haptički i slični izlazni uređaji. Glavna komponenta većine danas najpopularnijih VR uređaja je uređaj sa zaslonom koji se montira na glavu (engl. *head-mounted* display, skr. HMD). HMD je istovremeno i ulazni i izlazni uređaj [5]. Većina HMD-a na sebi imaju zaslon i zvučnike preko kojih korisniku simuliraju virtualnu okolinu, ali i razne senzore i kamere preko kojih prati pokrete korisnika u stvarnome prostoru. Samo računalo koje obrađuje ulazne podatke i stvara računalnu simulaciju virtualne stvarnosti, također se vrlo često nalazi unutar HMD-a.



Slika 1.1 Zatvorena petlja virtualne stvarnosti, modificirano s [4]

1.2. Proširena i miješana stvarnost

Za razliku od virtualne stvarnosti, tehnologije proširene stvarnosti stvaraju okolinu u kojoj su pomiješani stvarni svijet i virtualne informacije [6]. Takve tehnologije nadopunjuju, odnosno proširuju stvarnost tako da u nju dodaju virtualne elemente u stvarnome vremenu tako da oni izgledaju kao dio stvarnog svijeta. U idealnom slučaju, dok koristi uređaje proširene stvarnosti, korisnik ne može raspoznati koji element pripada stvarnome svijetu, a koji virtualnom. Proširena stvarnost može se smatrati sredinom između stvarnog svijeta i virtualne stvarnosti. Uporabom tehnologija proširene stvarnosti, korisnik dobiva novi pogled na određenu situaciju te mu virtualni elementi mogu pružiti pomoć u rješavanju određenih problema.

Definicija miješane stvarnosti često se poistovjećuje s definicijom proširene stvarnosti. Tehnologije miješane stvarnosti stvaraju okolinu u kojoj postoji interakcija u stvarnome vremenu između virtualnih i stvarnih elemenata [7]. Ovakav pristup omogućuje elementima jedne realnosti da direktno reagiraju i odgovaraju na stanje elemenata druge realnosti. Na primjer, tehnologije miješane stvarnosti omogućuju postavljanje virtualnog objekta na stvarnu podlogu, bez da stvarni objekt lebdi u zraku ili propada kroz stvarnu

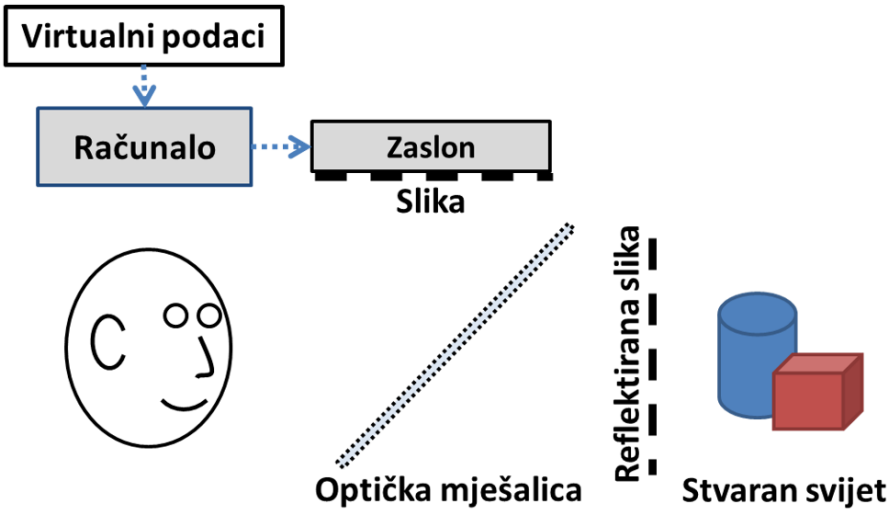
podlogu. Tehnologije miješane stvarnosti omogućuju i interakciju korisnika s virtualnim objektima. Kada koristi AR uređaje, korisnik virtualne objekte može samo promatrati, dok mu MR tehnologije omogućuju pomicanje, okretanje i druge vrste interakcije s virtualnim objektima. Zbog navedenih svojstva, korisniku je još teže razaznati stvarne od virtualnih elemenata.

Unatoč razlikama u definicijama, pojam proširene stvarnosti često se koristi za primjene gdje bi se zapravo trebao koristiti pojam miješana stvarnost. Jedan od razloga zašto se ova dva pojma miješaju je činjenica da je miješana stvarnost relativno nova tehnologija, pogotovo u odnosu na proširenu i virtualnu stvarnost. Zbog toga, kada se naiđe na određenu aplikaciju za koju se tvrdi da je AR aplikacija, ne može se poreći da u njoj ne postoje i MR elementi, odnosno interakcija između stvarnog i virtualnog. U nastavku su opisani glavni izazovi proširene, odnosno miješane stvarnosti.

1.2.1. Miješanje slike

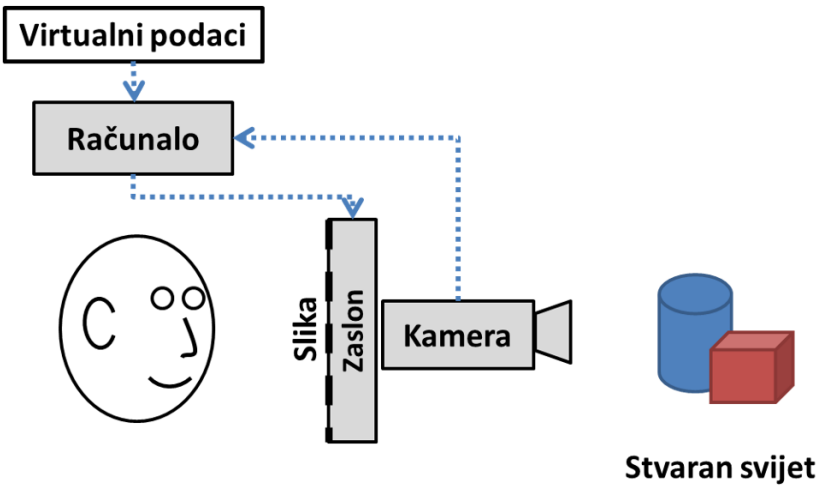
Miješanje slike je pozicioniranje virtualnih objekata u stvarni svijet u stvarnom vremenu [4]. Postoji više tehnika postizanja kvalitetnog miješanja, no danas je najučinkovitije i najpopularnije optičko i video miješanje.

Optičko miješanje je miješanje uz pomoć optičke miješalice, odnosno polu-prozirnog ogledala na kojem korisnik vidi virtualne elemente [4]. Optička miješalica najčešće se nalazi na HMD-u te s jedne strane na nju pada reflektirana slika stvarnog svijeta, a s druge virtualna slika koju generira i projicira računalo. Korisnik koji gleda u optičku miješalicu istovremeno vidi i stvarni svijet i virtualne elemente (slika 1.2). Glavni nedostatak ove tehnike miješanja je osjetljivost na jako svijetlo. U jako osvijetljenim prostorijama, reflektirana slika stvarnog prostora nadjačava generiranu virtualnu sliku te se ona slabo vidi iz korisnikove perspektive. Prednost optičkog nad drugim tehnikama miješanja je jednostavnost. Za optičko miješanje, HMD-u potrebno je samo računalo, zaslon i optička miješalica, što ga čini laganijim u odnosu na uređaje koji koriste druge tehnike miješanja.



Slika 1.2 Optičko miješanje slike, preuzeto iz [4]

Glavna komponenta video miješanja je kamera koja snima stvarnu okolinu i šalje ju računalu na obradu [4]. Računalo snimku okoline proširuje virtualnim elementima i prikazuje korisniku na ekranu (slika 1.3). Za razliku od optičkog miješanja, korisnik koji koristi uređaj s video miješanjem ne vidi stvarnu okolinu direktno, nego preko video snimke zajedno s virtualnim elementima. Ovo je ujedno i jedan od nedostataka video miješanja. Uređaji s kamerom slabe kvalitete mogu uvelike degradirati korisnikovu percepciju stvarne okoline. Također, kod računala sa slabim procesorom može doći do kašnjenja između snimanja video snimke i njenog prikaza korisniku što dodatno može narušiti iskustvenu kvalitetu korisnika. Prednost video miješanja je mogućnost laganog kontroliranja video snimke koja se prikazuje korisnicima jer se u nju jednostavno mogu dodati razni grafički elementi.



Slika 1.3 Video miješanje slike, preuzeto iz [4]

1.2.2. Prikaz rezultata miješanja

Rezultat miješanja se osim na već spomenutim HMD uređajima može prikazati i na pametnim telefonima i tabletima. Pametni telefoni su zbog svoje popularnosti trenutno najkorišteniji vizualni uređaji za prikaz proširene i miješane stvarnosti [8]. Veliki broj ljudi uporabom pametnih telefona koristi i proširenu odnosno miješanu stvarnost bez da su toga svjesni. Primjerice, aplikacije poput Snapchata ili TikToka omogućuju korisnicima proširivanje videozapisa s raznim interaktivnim filterima i virtualnim elementima (slika 1.4). Slika proširene i miješane stvarnosti može se prikazivati i na HUD uređajima (engl. *heads-up displays*). HUD uređaji sliku prikazuju na transparentom zaslonu, a česta primjena im je u vozilima [27]. Primjerice, HUD uređaj može se ugraditi u automobil ispred prednjeg stakla. Takav uređaj omogućuje vozaču da prati navigaciju i slične informacije, dok istovremeno održava fokus na cesti.

Slika 1.4 AR filteri u aplikaciji Snapchat, preuzeto s [9]

1.2.3. Poravnanje

Poravnavanje (engl. *registration*) u AR-u i MR-u je postavljanje virtualnih elemenata u stvarni svijet tako da oni u njemu djeluju što realnije i da je korisniku što teže razaznati radi li se o stvarnom ili virtualnom dijelu okoline. Jedna metoda poravnavanja virtualnih elemenata u stvarnoj okolini oslanja se na uporabi hardvera. AR ili MR uređaji pomoću magnetnih senzora, GPS-a, akcelerometra i sl. uređaja od stvarne okoline grade 3D

koordinatni sustav u koje računalo smješta virtualne elemente [10]. Kvaliteta poravnanja ove metode uvelike ovisi o kvaliteti senzora koji sakupljaju informacije o stvarnoj okolini. Ako se radi o senzorima lošije kvalitete, računalo postavlja elemente na pogrešna mjesta i slika koja se prikazuje korisniku ne izgleda realistično. Druga metoda poravnanja koristi kameru koja kao kod video miješanja snima okolinu i šalje snimku računalu na obradu. Računalo pomoću strojnog učenja analizira okolinu sa snimke i u nju ubacuje virtualne elemente tako da izgledaju što realnije. Ova metoda osjetljiva je na jačinu svjetlosti i veličinu prostora u kojem se koriste te je najučinkovitija u zatvorenim prostorima s odgovarajućom dozom svijetla.

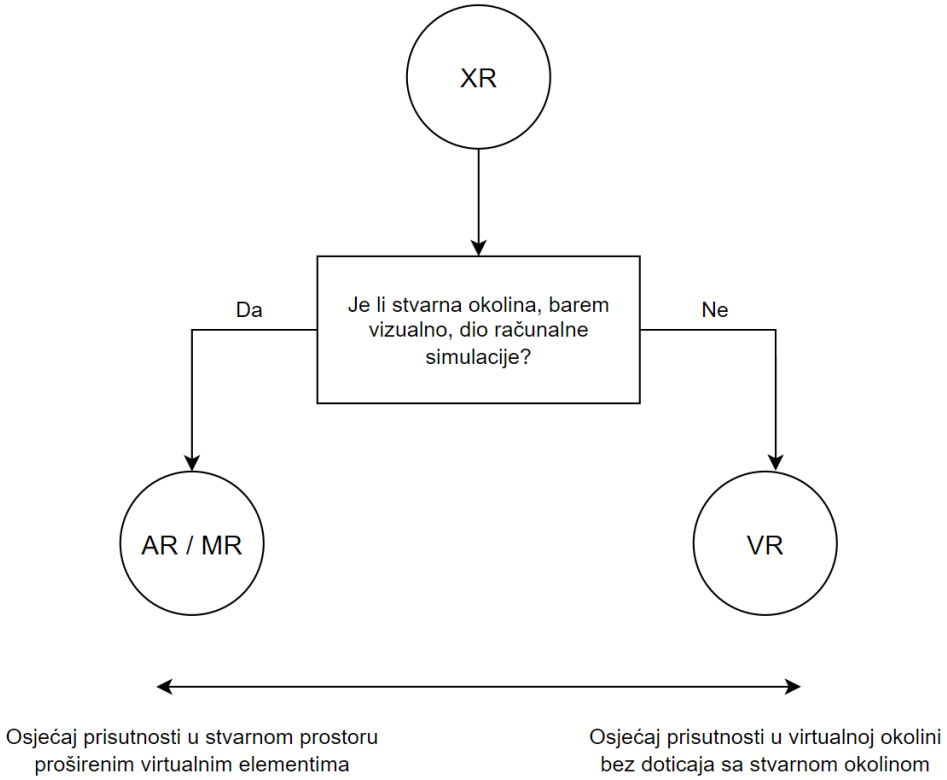
Slijeđenje (engl. *tracking*) je postupak dobivanja orijentacije virtualnih elemenata u stvarnome prostoru i u stvarnome vremenu ovisno o perspektivi i kutu gledanja korisnika [4]. Najjednostavnija, a pritom i visoko kvalitetna metoda slijeđenja je metoda slijeđenja s markerima (engl. *marker based tracking*). Za ostvarivanje ove metode, najprije je potrebno na odgovarajuća mjesta u stvarnu okolinu postaviti markere. Markeri služe kao orijentir AR ili MR kameri te računalo pomoću njih određuje gdje treba postaviti virtualne elemente i kako ih orijentirati. Važno je da su markeri dovoljno veliki da ih računalo može prepoznati sa što veće daljine i iz različitih kutova gledanja. Markeri su uglavnom dvodimenzionalne slike s lako prepoznatljivom vizualnom karakteristikom (npr. QR kodovi).

Slijeđenje se može postići i bez markera. Bezmarkersko slijeđenje (engl. *markerless tracking*) bazira se na dubinskim kamerama i raznim senzorima pomoću kojih računalo prikuplja informacije o okolinu [4]. Računalo potom prati promjene u stvarnoj okolini i promjenu položaja i kuta gledanja AR ili MR uređaja te na temelju prikupljenih informacija postavlja i pravilno orijentira virtualne elemente. Ovakav postupak spada pod SLAM (engl. *simultaneous localization and mapping*) tehnike i zbog velike količine računanja može prouzročiti kašnjenje između snimanja okoline i prikaza virtualnih elemenata na ekranu te tako smanjiti stupanj zadovoljstva korisnika uređajem.

1.3. Dograđena stvarnost

Pojam dograđena stvarnost odnosi se na sve tehnologije koje na bilo kakav način kombiniraju stvaran svijet i virtualne elemente, pružaju korisniku osjećaj prisutnosti u

pomiješanoj okolini te mu omogućuju interakciju s virtualnim elementima [11]. Do sada spomenute imerzivne tehnologije (VR, AR i MR) zajedno se mogu svrstati pod pojam dograđena stvarnost. Na slici 1.5 prikazana je kategorizacija navedenih imerzivnih tehnologija. Pojmom XR nastoji se smanjiti zabuna koja nastaje miješanjem pojmova virtualne, proširene i miješanje stvarnosti, a pogotovo proširene i miješanje. Pod dograđenu stvarnost spadaju i aplikacije koje kombiniraju svojstva virtualne stvarnosti sa svojstvima proširene i miješane stvarnosti, kao što je npr. prelazak iz kompletno virtualne okoline u stvarnu okolinu popunjenu virtualnim elementima.

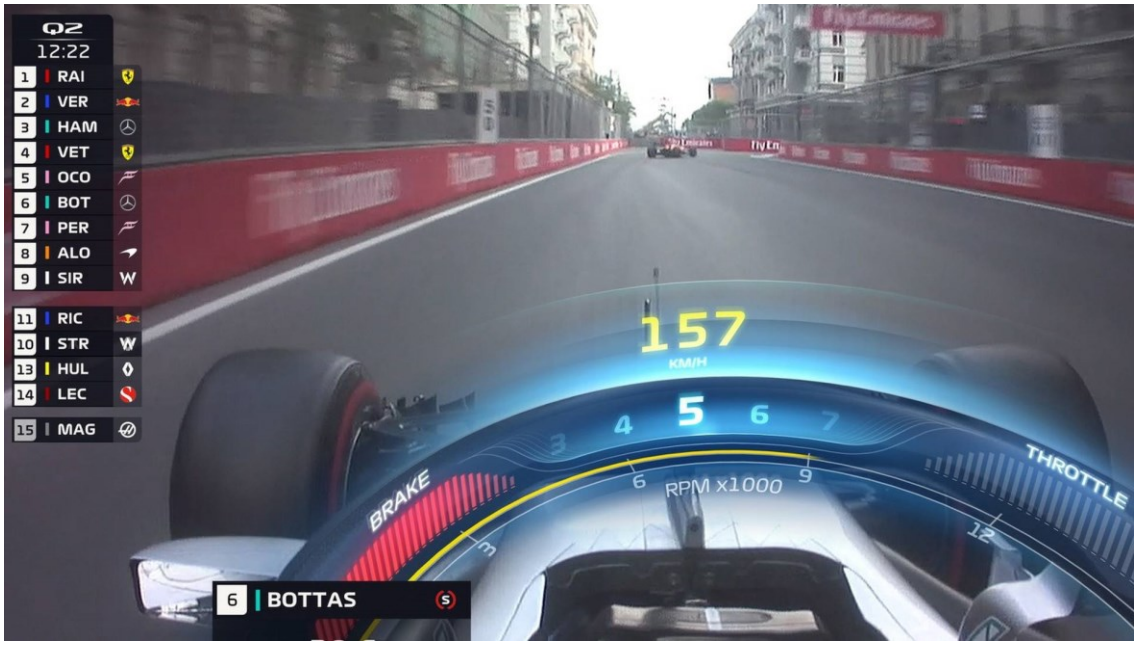


Slika 1.5 Kategorizacija imerzivnih tehnologija, modificirano s [21]

Središnji pojam tehnologija dograđene stvarnosti je imerzija. Bez obzira o kojoj se vrsti ili primjeni XR aplikacije radi, cilj je korisniku zamagliti ili u potpunosti prekriti granicu između stvarne i virtualne okoline te mu virtualnim elementima pružiti novi, prošireni pogled na situaciju u kojoj se nalazi. Za postizanje što bolje imerzije, jasna slika i realistični virtualni elementi često nisu dovoljni sami po sebi. Kako bi se korisniku pružio što veći osjećaj prisutnosti u pomiješanoj stvarnosti, simulaciju je često potrebno obogatiti zvukom, ali i osigurati jasnu orkestraciju virtualnih i stvarnih elemenata ovisno o kontekstu u kojem se korisnik nalazi.

1.4. Imerzivne tehnologije u sportskim prijenosima

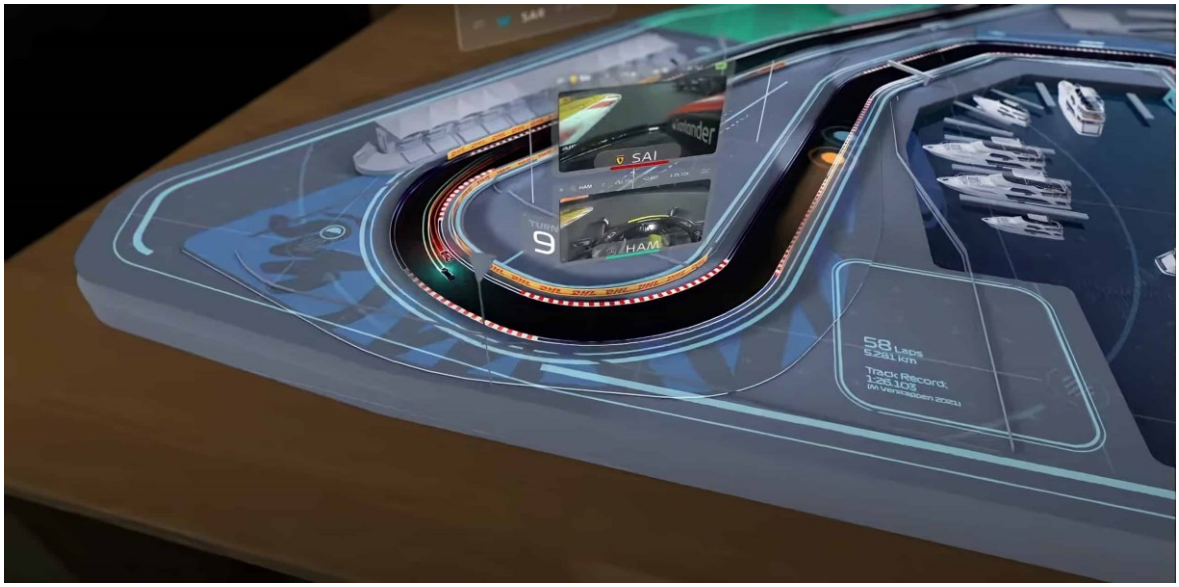
Kao što je već spomenuto, imerzivne tehnologije imaju široku primjenu i pojavljuju se u industriji zabave pa tako i u sportskim prijenosima. Korištenje imerzivnih tehnologija u sportskim prijenosima pruža brojne prednosti obožavateljima sporta, ali i organizatorima i pokroviteljima sportskih događaja [12]. Tehnologijama dograđene stvarnosti obožavateljima određenog sporta može se pružiti imerzivno iskustvo gledanja sportskog događaja. Primjerice, XR tehnologije mogu pružiti obožavateljima osjećaj prisutnosti na stadionu iako sportski događaj gledaju iz svoga doma. Također, virtualnim elementima mogu se prikazati razne statistike i informacije o događaju i tako obogatiti sportski prijenos i učiniti ga lakšim za pratiti. Na primjer, u Formuli 1, virtualni elementi koriste se za prikazivanje brzine F1 bolida u stvarnom vremenu (slika 1.6). Organizatori sportskih događaja koriste imerzivne tehnologije kako bi privukli nove gledatelje i zadržali postojeće. Kako određene sportske događaje prati više milijuna gledatelja, organizatorima je cilj prikazati što više reklama kako bi dobili što veću svotu novca od pokrovitelja. XR tehnologijama reklame se mogu postavljati u prijenos na razna mjesta i na zanimljive načine kako bi privukli pažnju gledatelja.



Slika 1.6 Prikaz brzine F1 bolida u stvarnom vremenu, preuzeto s [13]

Aplikacija Xtadium [14] primjer je XR aplikacije koja omogućuje korisnicima gledanje sportova u dograđenoj stvarnost. Ova aplikacija korisnike smješta na tribinu ili uz sam teren određenog sportskog događaja tako da korisnik iz svoje perspektive vidi sve što bi vidio kada bi se i fizički nalazio na mjestu tog događaja. Također, Xtadium omogućuje praćenje više sportskih događaja istovremeno kao i gledanje događaja s više korisnika koji također koriste aplikaciju. Aplikacija je namijenjena za Meta Quest uređaje, podržava gledanje događaja uživo i na zahtjev (engl. *on-demand*), a najpopularniji sportovi koji se prenose na njoj su tenis, košarka i UFC.

Još jedan primjer XR aplikacije je koncept Johna LePorea [15] koji omogućuje praćenje utrke Formule 1 u stvarnom vremenu koristeći tehnologije dograđene stvarnosti. LePoreova ideja je uporabom virtualnih elemenata prikazati trkaću stazu i F1 bolide na njoj u stvarnom vremenu (slika 1.7) paralelno s prijenosom F1 utrke. Gledatelji bi uporabom ovakve aplikacije dobili uvid o lokaciji svih bolida na stazi, kojeg ne mogu dobiti iz samog prijenosa. Osim toga, zamišljena aplikacija virtualnim elementima prikazuje informacije o utrci kao što su poredak vozača, brzine bolida, broj preostalih krugova i sl. Aplikacija napravljena u sklopu ovog rada inspirirana je LePoreovim konceptom.

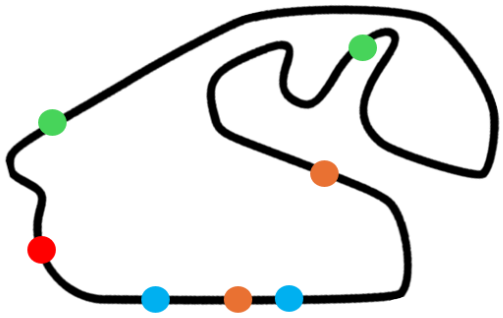


Slika 1.7 LePoreov koncept virtualne staze za praćenje utrke Formule 1, preuzeto s [15]

2. Opis aplikacije *Formula XR*

2.1. Cilj i motivacija

Aplikacija *Formula XR*, inspirirana konceptom Johna LePorea [15], izrađena je kako bi obogatila iskustvo gledanja utrke Formule 1. Prijenos F1 utrka već je sam po sebi veoma informativan jer je proširen raznim virtualnim elementima. Formula 1 je sport vođen podacima [16]. Trkaći inženjeri, vođe F1 momčadi i sami vozači donose odluke na temelju podataka o stanju bolida, njegovog motora i guma te ostalih faktora koji utječu na utrku. Zbog svoje važnosti, navedeni podaci nisu zanimljivi samo djelatnicima F1 momčadi, već i gledateljima koji pomoću njih mogu predviđati što će se iduće dogoditi u utrci. Zbog toga se u prijenosima utrka, raznoraznim virtualnim elementima, gledateljima nastoji prikazati što više relevantnih informacija. Tako u prijenosima postoje raznorazne grafike koje prikazuju poredak vozača u utrci, vremena između vozača, stanje guma trkaćih automobila, vrijeme izgubljeno na promjenu guma i sl. Postoji i grafički prikaz koji točkama prikazuje pozicije F1 trkaćih automobila, odnosno bolida na stazi. Skica jedne takve grafike prikazana je na slici 2.1. Ovakav prikaz jedan je od najboljih prikaza stanja utrke jer pomoću njega gledatelji vide koliki je zapravo razmak između dva bolida na stazi. Mnogim gledateljima, pogotovo onima koji su ne prate sport dugo, vjerojatno je intuitivnije shvatiti koliko jedan bolid zaostaje za drugim kada vide takvu grafiku, nego kada pročitaju da bolid zaostaje, npr. dvije sekunde.



Slika 2.1 Skica grafike koja prikazuje lokaciju bolida tijekom utrke

Kakogod, zbog prostora kojeg zauzima na ekranu, ovakva grafika pojavljuje se samo u određenim dijelovima prijenosa te gledateljima ponekad fali informacija o lokaciji pojedinih bolida na stazi. Aplikacija *Formula XR* napravljena je da poput grafike na slici 2.1 pruža informacije o lokacijama bolida na stazi. Umjesto 2D prikaza, aplikacija *Formula XR* za to koristi 3D model stvarne staze na kojoj se utrkuju 3D modeli F1 bolida. Još jedan problem prijenosa utrka je što režiser prijenosa u jednom trenutku može prikazati samo događanja s jednog dijela staze. Režiser u prvom planu najčešće prikazuje bolide koji se bore za pobjedu ili dijelove staze na kojima se odvija mnogo pretjecanja. Posljedično, režiser ponekad propusti prikazati zanimljive događaje koji se odvijaju na drugim mjestima na stazi. Uporabom aplikacije *Formula XR* korisnik vidi cijelu stazu, odnosno sve dijelove utrke te može pratiti točno one dijelove koji su njemu najzanimljiviji.

Glavni izazov aplikacije *Formula XR* je što vjernije prikazati stvarnu utrku. Ideja aplikacije je da korisnik istovremeno koristi aplikaciju i prati stvarni prijenos na zasebnom ekranu. Zbog toga je potrebno da se virtualni bolidi na virtualnoj stazi gibaju sličnom brzinom kao i stvarni bolid te da se u svakom trenutku nalaze na otprilike istom dijelu staze. Kako lokacije bolida na stazi za vrijeme utrke nisu javno dostupne, aplikacija *Formula XR* koristi se samo za prethodno završene utrke za koje su dostupni podaci o vremenima potrebnim bolidima da odvoze svaki od krugova utrke. Konkretno, u sklopu ovog projekta, aplikacija *Formula XR* koristi se za vizualizaciju utrke Velike nagrade Brazila iz 2021. godine.

Aplikacija *Formula XR* virtualni model staze prikazuje u proširenoj, odnosno miješanoj stvarnosti, kako bi korisnici paralelno mogli pratiti stvarni prijenos utrke. Korisnik virtualnu stazu može smjestiti na bilo koju lokaciju u svojoj stvarnoj okolini. Osim prikaza stanja utrke, aplikacija omogućuje otvaranje jednog ili više dodatnih virtualnih ekrana koji u proširenoj stvarnosti prikazuju pogled na utrku iz perspektive određenog bolida (slična perspektiva kao ona na slici 1.6). Navedena funkcionalnost omogućava korisnicima detaljnije praćenje najzanimljivijih dijelova utrke ili njihovih najdražih vozača. Aplikacija također omogućuje pauziranje i premotavanje utrke te gledanje utrke u usporenom prikazu.

Do sada opisana aplikacija ima samo svojstva proširene i miješane stvarnosti. Kako bi aplikacija opravdala pojam XR u svom nazivu, ona omogućuje prijelaz iz miješane u virtualnu stvarnost. Kao što je prethodno opisano, aplikacija po pokretanju virtualne elemente prikazuje u miješanoj stvarnosti te korisnik je svjestan okoline u kojoj se nalazi dok je koristi. Pritiskom na određeni gumb u sceni, aplikacija omogućuje postepeni prijelaz u virtualnu stvarnost. U virtualnoj stvarnosti aplikacija korisniku pruža osjećaj prisutnosti

na tribini koja se nalazi uz samu stazu. Korisnik tada više nema dodira sa stvarnom okolinom te ne može pratiti stvaran prijenos, ali zato promatra utrku iz perspektive gledatelja na tribini. Ovakva funkcionalnost može biti korisna korisnicima koji namjeravanju posjetiti stvarnu stazu da odaberu koja im tribina najviše odgovara za praćenje utrke.

2.2. Korištene tehnologije

2.2.1. Meta Quest 3

Meta Quest 3 [17] je HMD uređaj za prikaz dograđene stvarnosti (slika 2.2). Razvila ga je tvrtka Meta 2023. godine kao nasljednika Meta Quest 2 uređaja. Meta Quest 2 uređaj i njegovi prethodnici prvenstveno su služili kao uređaji za prikaz virtualne stvarnosti. Novost kod Meta Quest 3 uređaja su tri kamere na njegovoj prednjoj strani. Dvije od tih tri kamera su RGB kamere kojima je cilj što vjernije replicirati ljudski vid. Treća kamera je dubinska kamera i u kombinaciji s ostale dvije, Meta Quest 3 uređaju omogućuju prikaz proširene, odnosno miješane stvarnosti. Uporabom ovih kamera, Meta Quest 3 prati promijene u stvarnoj okolini pa tako i korisnikove ruke te omogućuje interakciju korisnika s virtualnim elementima.

Meta Quest 3, kao i njegovi prethodnici, također može prikazivati i virtualnu stvarnost. Dograđenu stvarnost prikazuje na dva LCD ekrana, svaki namijenjen jednom oku s rezolucijom od 2064 × 2208 piksela [17]. Uređaj može pratiti 3 ili 6 stupnjeva slobode, a opremljen je i sa zvučnicima i mikrofonom. Na samom HMD uređaju nalazi se i računalo koje generira računalnu simulaciju koja se korisnicima prikazuje na zaslonu. Na računalu je instaliran Meta Horizon operacijski sustav koji omogućava korisnicima spajanje na Internet, pohranu podataka, preuzimanje aplikacija, pretraživanje Interneta te druge slične radnje koje ga čine sličnim računalima na pametnim telefonima. Meta Quest 3 dolazi s dva Touch Plus kontrolera pomoću kojih je također omogućeno praćenje korisnikovih pokreta te interakcija s virtualnim elementima.

Izradu XR aplikacija za Meta Quest 3 uređaj olakšava Meta Quest Link[1] aplikacija koja omogućava spajanje uređaja na osobno računalo. Putem ove aplikacije, programeri XR aplikacije izrađene na osobnim računalima mogu testirati direktno na Meta Quest 3 uređaju što ubrzava proces izrade istih. Osim testiranja, Meta Quest Link aplikacija omogućava i pristup podacima na uređaju te ubrzava proces puštanja XR aplikacija u pogon.

Slika 2.2 Meta Quest 3 uređaj, preuzeto s [17]

2.2.2. FastF1

FastF1 [18] je paket u programskom jeziku Python za pristup i analiziranje podataka o održanim utrkama Formule 1. Pomoću ovog paketa mogu se dohvatiti informacije o utrkama kao što su konačni rezultati, telemetrija, pozicije vozača na početku utrke, podaci o gumama, vremenski uvjeti, rasporedi događanja i sl. FastF1 paket, u kombinaciji s Pandas[2] i Matplotlib[3] paketima omogućuje i vizualizaciju dohvaćenih podataka uporabom raznoraznih grafikona. Na službenoj stranici FastF1 paketa i njegovoj GitHub[4] stranici dostupna je detaljna dokumentacija i brojni primjeri korištenja paketa i dohvaćenih

[1] https://www.meta.com/help/quest/articles/headsets-and-accessories/oculus-rift-s/install-app-for-link/

[2] https://pypi.org/project/pandas

[3] https://matplotlib.org

[4] https://github.com/theOehrly/Fast-F1

podataka. U sklopu ovog projekta, paket FastF1 korišten je za dohvaćanje vremena po krugovima i sektorima za vozače u utrci Velike nagrade Brazila 2021. godine.

2.2.3. Visual Studio 2022

Visual Studio [20] je Microsoftova razvojna okolina (engl. *integrated development environment*, skr. IDE) za pisanje i uređivanje programskog koda te razvoj web i mobilnih aplikacija. Visual Studio podržava više programskih jezika od kojih su najpoznatiji C++, C# i Python te sadrži brojne alate koje olakšavaju posao programerima u razvijanju aplikacija, kao što su alat za traženje pogrešaka u kodu (engl. *debugger*), alat za dijeljenje koda uporabom Gita i sl. Dostupni alati i funkcionalnosti okoline Visual Studio mogu se proširiti raznim dodacima (engl. *plugins*). U sklopu ovog projekta, program Visual Studio korišten je za pisanje, uređivanje i testiranje Unity C# skripta i Python skripte za dohvaćanje podataka o F1 utrkama uporabom FastF1 paketa. Za rad s Unityjem i Unity C# skriptama, Visual Studio potrebno je proširiti paketom *Game development with Unity*.

2.2.4. Unity

Unity [19] je platforma za izradu 2D i 3D interaktivnih aplikacija. Predstavljen je 2005. godine za Mac OS X operacijski sustav, no do danas je dostupan i za Windows i za Linux. Unity omogućava programerima efikasan proces izrade interaktivnih igara za raznolike platforme, uključujući osobna računala, pametne telefone, igraće konzole, ali i XR uređaje poput Meta Quest 3 uređaja. Proces izrade interaktivnih aplikacija u Unityju vrlo je intuitivan i jednostavan za naučiti, a temelji se na ubacivanju gotovih 2D ili 3D komponenti u virtualnu scenu te određivanju interakcije između tih komponenti korištenjem C# skripti. Za izradu aplikacije *Formula XR* korištena je Unity verzija 2022.3.11f1.

Sastavni dio Unityja je i Unity Asset Store na kojem se mogu besplatno preuzeti ili kupiti gotove komponente koje su izradili drugi Unity korisnici. Ove komponente mogu znatno ubrzati razvoj aplikacija jer programeri određene funkcionalnosti i komponente mogu jednostavno preuzeti s Unity Asset Storea te fokusirati se na specifične probleme vezane

uz njihovu aplikaciju. U tablici 2.1 naveden su i opisane komponente preuzete s Unity Asset Storea koje su se koristile u procesu izrade aplikacije *Formula XR*.

Tablica 2.1 Popis korištenih komponenti s Unity Asset Storea za izradu aplikacije *Formula XR*

Naziv komponente	Opis komponente
Cartoon Race Track Interlagos[5]	Vjeran 3D model trkaće staze Interlagos.
Low Poly Racing Car[6]	3D model bolida Formule 1.
Bézier Path Creator[7]	Komponenta koja omogućuje izradu Bézierovih krivulja u Unity sceni. Korištena za izradu krivulja na virtualnoj stazi koje bolidi prate.
Meta XR All-in-One SDK[8]	Kolekcija komponenti i C# skripti za ubrzan razvoj XR aplikacija za uređaje kompanije Meta. Detaljnije opisana u potpoglavlju 2.2.5.
Meta XR Interaction SDK OVR Samples[9]	Kolekcija scena u kojima su prikazane funkcionalnosti komponenta koje se nalaze u Meta XR All-in-One SDK paketu.

[5] https://assetstore.unity.com/packages/3d/environments/roadways/cartoon-race-track-interlagos-199223

[6] https://assetstore.unity.com/packages/3d/vehicles/land/low-poly-racing-car-224001

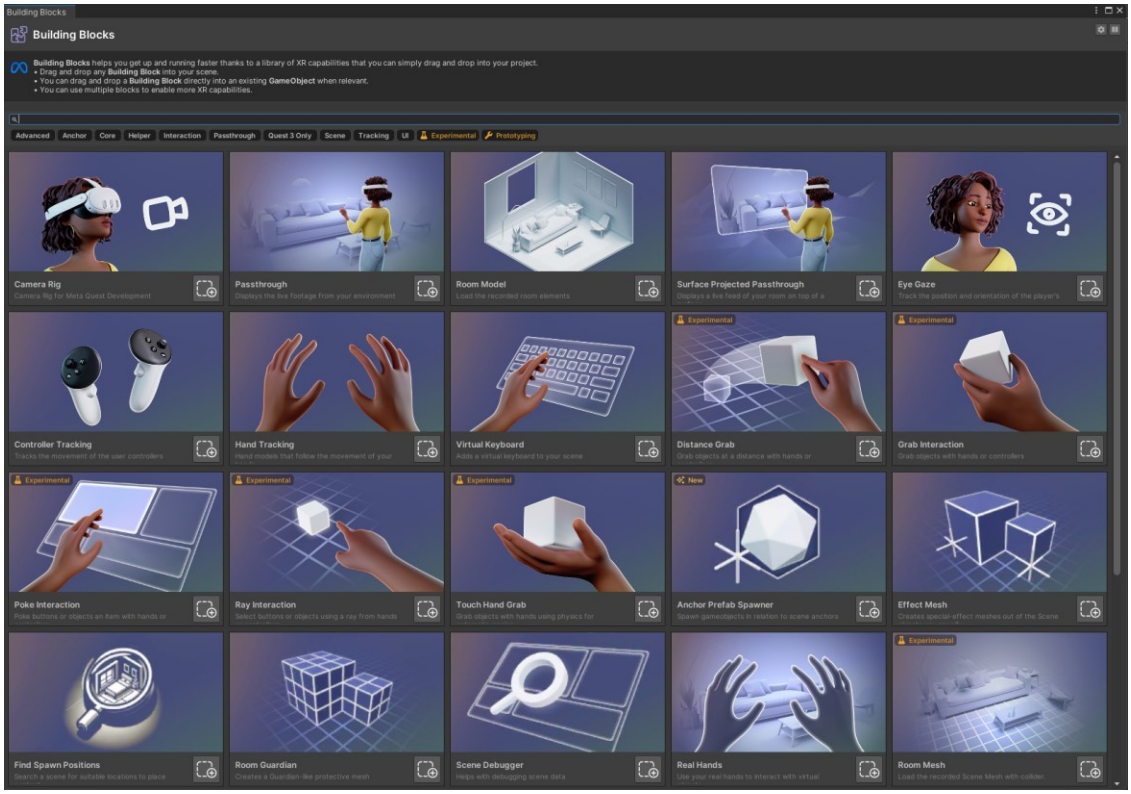
[7] https://assetstore.unity.com/packages/tools/utilities/b-zier-path-creator-136082

[8] https://assetstore.unity.com/packages/tools/integration/meta-xr-all-in-one-sdk-269657

[9] https://assetstore.unity.com/packages/tools/integration/meta-xr-interaction-sdk-ovr-samples-268521

2.2.5. Meta Interaction SDK

Meta Interaction SDK je skup gotovih komponenti, modela, tekstura, animacija i C# skripti za ubrzani razvoj XR aplikacija namijenjenih za uređaje kompanije Meta. Ovaj paket koristi se uz Unity programsku platformu i sposoban je samostalno ispravno konfigurirati postavke Unity projekta. Na ovaj način, paket omogućuje programerima da više vremena posvete problemima vezanim za njihovu specifičnu aplikaciju. Dio Meta Interaction SDK paketa su i Meta Building Blocks.[10] Meta Building Blocks je sučelje (slika 2.3) koje paket nadoda u Unity platformu te omogućuje pregled dostupnih XR komponenti i njihovo jednostavno ubacivanje u Unity scenu. Primjerice, za pružanje korisnicima VR iskustva u virtualnoj prostoriji, potrebno je samo modelirati tu prostoriju i u nju putem Meta Building Blocks sučelja ubaciti komponentu *Camera Rig.* Ova će komponenta konfigurirati sve potrebne postavke i omogućiti pokretanje aplikacije na XR uređaju kompanije Meta.



Slika 2.3 Izbornik za ubacivanje Meta Building Blocks komponenti u Unity scenu

[10] https://developer.oculus.com/blog/accelerate-development-mixed-reality-building-blocks-unity-meta-quest-developers

3. Izrada aplikacije *Formula XR*

Ovo poglavlje podijeljeno je na dvije cjeline koje prate proces izrade aplikacije *Formula XR*. Kao što je navedeno prethodno, primarni cilj aplikacije vjerno je prikazati utrku Formule 1 uporabom virtualnih elemenata poput virtualne staze, virtualnih bolida i sl. Prva cjelina opisuje proces izrade same utrke na programskoj platformi Unity i dohvaćanje podataka za simulaciju stvarne utrke. Rezultat ove cjeline je računalna aplikacija bez ikakvih XR svojstva koja omogućuje gledanje virtualne varijante utrke. Osim same simulacije utrke, aplikacija omogućuje premotavanje, pauziranje i gledanje utrke u usporenom prikazu. U drugoj cjelini, opisan je proces pretvorbe izrađene aplikacije u imerzivnu XR aplikaciju koja je namijenjena za Meta Quest 3 uređaj. Ova cjelina obuhvaća sve XR funkcionalnosti među kojima su i interakcija korisnika s virtualnim elementima, prelazak iz MR varijante u VR varijantu, gledanje utrke iz perspektive bolida i sl.

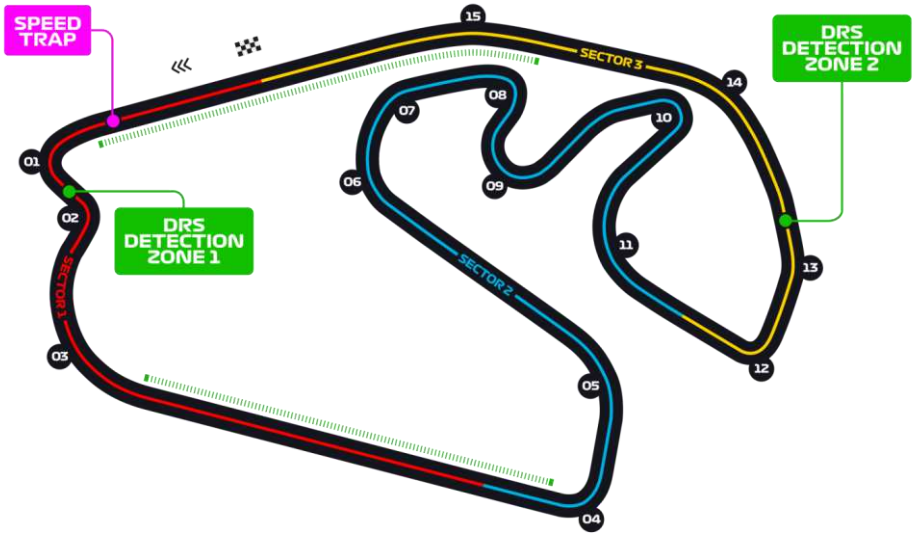
3.1. Simulacija utrke

3.1.1. Dohvat podataka uporabom FastF1 paketa

Kako bi vjerno prikazivala stvarnu utrku, aplikacija *Formula XR* mora se temeljiti na stvarnim podacima. Jedan od izazova izrade ovakve aplikacije je činjenica da podaci o stvarnoj poziciji F1 bolida na stazi u određenom trenutku utrke nisu javno dostupni. Kakogod, dostupno je vrijeme koje je potrebno svakom vozaču da prijeđe bilo koji od krugova utrke. Aplikacija *Formula XR* temelji se upravo na takvim podacima. Aplikacija ne prikazuje vjerno položaj bolida u svakom trenutku utrke, ali osigurava da virtualni bolid započne i završi krug u istom trenutku kao i stvarni. Iako se na ovakav način u simulaciji izgube određeni trenutci utrke kao što su sudari ili poneka pretjecanja, na kraju se utrke, odnosno na kraju svakog kruga, dobi poredak vozača koji je jednak onom u stvarnoj utrci.

Kao što je već navedeno, aplikacija *Formula XR* prikazuje virtualnu varijantu utrke Velike nagrade Brazila iz 2021. godine na stazi Interlagos. Ova utrka odabrana je zato što je jedna od zanimljivijih i upečatljivijih utrka posljednjih nekoliko sezona. U utrci je sudjelovalo

dvadeset vozača iz deset različitih momčadi, koji su prethodnim kvalifikacijama odredili startni poredak u utrci. Na slici 3.1 prikazan je tlocrt staze Interlagos. Interlagos je, kao i sve ostale staze Formule 1, podijeljen na tri sektora (na slici 3.1 označeni crvenom, plavom i žutom bojom). Osim vremena vozača po krugovima, javno je dostupno i vrijeme potrebno pojedinom vozaču da prijeđe jedan od sektora određenog kruga. Sektori su različitih duljina i razlikuju se po broju zavoja. Tako na primjer na stazi Interlagos, sektor 2 ima najviše zavoja te vozači u njemu moraju voziti sporije kako bi uspješno prošli kroz sve zavoje. U aplikaciji *Formula XR* koriste se vremena po sektorima kako bi se što realnije simulirala utrka, odnosno kako bi se postiglo da se bolidi gibaju brže u sektorima s manje zavoja. Osim samih vremena, aplikaciji je potrebna i informacija u kojim je krugovima vozač otišao pred garažu na zamjenu guma ili neispravnih dijelova (dalje u tekstu *pit-stop*). Pomoću ove informacije, aplikacija usmjerava virtualne bolide na *pit-stop* u određenim krugovima.



Slika 3.1 Tlocrt staze Interlagos u Brazilu, preuzeto s [22]

Za dohvat navedenih podataka uporabom paketa FastF1 i programskog jezika Python (kôd 3.1), najprije je potrebno učitati podatke o utrci (redci 5 i 6). Nakon učitavanja podataka, potrebno je dohvatiti vozače koji su sudjelovali u utrci (redak 8) te za svakog vozača i za svaki krug koji je taj vozač odvozio (redci 15 i 25) pročitati vrijeme u sekundama koje je vozaču trebalo da završi krug i svaki sektor u tom krugu (redci 26 - 31). Osim vremena, dohvaćaju se i krugovi u kojima su vozači otišli na *pit-stop* (redci 16 i 33). Dohvaćeni podaci izvoze se u CSV (engl. *comma-separated values*) datoteke kako bi se kasnije

jednostavnije koristili u Unity skriptama koje su pisane u programskom jeziku C#. Dvadeset CSV datoteka nastaje kao rezultat izvođenja skripte. Svaka datoteka sadrži podatke o jednom vozaču (slika 3.2), a nazvana je po kratici prezimena vozača, na primjer datoteka VER.csv sadrži vremena vozača Maxa Verstappena.

```
1   import fastf1
2   import csv
3   import os
4
5   session = fastf1.get_session(2021, 'Brazil', 'R')
6   session.load()
7
8   drivers = session.drivers
9
10  output_folder = 'telemetry'
11  if not os.path.exists(output_folder):
12      os.makedirs(output_folder)
13
14  for driver in drivers:
15      driver_laps = session.laps.pick_driver(driver)
16      box_laps = driver_laps.pick_box_laps('in')
17
18      csv_file = f'telemetry/{driver_laps.Driver.iloc[0]}.csv'
19
20      with open(csv_file, mode='w', newline='') as file:
21          writer = csv.writer(file)
22          writer.writerow(['lap_num', 'lap_time', 's1_time',
23                           's2_time', 's3_time', 'box_lap'])
24
25          for lap in driver_laps.iterlaps():
26              lap_num = lap[1]["LapNumber"]
27              lap_time = lap[1]["LapTime"].total_seconds()
28
29              s1_time = lap[1]["Sector1Time"].total_seconds()
30              s2_time = lap[1]["Sector2Time"].total_seconds()
31              s3_time = lap[1]["Sector3Time"].total_seconds()
32
33              box_lap = lap[0] in list(box_laps.index)
34
35              writer.writerow([lap_num, lap_time, s1_time,
36                               s2_time, s3_time, box_lap])
```

Kôd 3.1 Python skripta za dohvaćanje podataka o utrci uporabom paketa FastF1

lap_num	lap_time	s1_time	s2_time	s3_time	box_lap
1	83.099	25.255	40.85	16.994	FALSE
2	75.675	19.127	39.451	17.097	FALSE
3	74.254	18.781	38.688	16.785	FALSE
4	74.307	18.913	38.321	17.073	FALSE
5	74.172	19.046	38.191	16.935	FALSE
6	87.601	19.163	49.061	19.377	FALSE
7	119.617	25.212	59.672	34.733	TRUE
8	131.737	39.007	59.808	32.922	TRUE
9	147.627	44.815	63.122	39.69	FALSE

Slika 3.2 Dio podataka o vremenima vozača iz jedne CSV datoteke

3.1.2. Razvoj Unity scene

Nakon stvaranja novog Unity projekta, u njega se moraju uvesti podaci dobiveni FastF1 paketom i virtualne reprezentacije staze i bolida preuzete s Unity Asset Storea. Prvi korak nakon uvoza je ubacivanje komponente virtualne staze Interlagos u scenu (slika 3.3). Virtualna staza vrlo je slična stvarnoj. Razlikuje se u ukupnoj duljini i visinskoj razlici pojedinih dijelova staze, što ne predstavlja problem kod razvoja aplikacije *Formula XR*. Osim same staze, dio preuzete komponente je i okolina staze tj. razni neboderi, tornjevi i jezera po kojima je Interlagos prepoznatljiv među obožavateljima Formule 1.

Slika 3.3 Virtualna staza Interlagos u Unity sceni

Sljedeći korak je ubacivanje virtualnih F1 bolida u scenu. U preuzetom paketu virtualni bolidi dolaze samo u tri boje, pa ih je prije ubacivanja u scenu potrebno pobojati promjenom boje materijala u boje momčadi koje su sudjelovale u Formuli 1 2021. godine (slika 3.4). Bolidi trebaju biti postavljeni na startnu rešetku (engl. *starting grid*) poretkom koji su vozači završili kvalifikacije koje su se održale dan prije utrke. Rezultati svih održanih kvalifikacija i utrka Formule 1 mogu se pronaći na službenoj F1 stranici [22]. Nakon postavljanja bolida na startnu rešetku, potrebno je iz scene ukloniti glavnu kameru te na svaki bolid postaviti novu kameru. Nove kamere bit će korištene za prikaz utrke iz perspektive bolida (slika 3.5) te trebaju biti postavljene na bolid tako da je taj pogled što sličniji ekvivalentnom pogledu u stvarnim F1 prijenosima (slika 1.6).

Slika 3.4 Virtualni F1 bolidi u bojama F1 momčadi

Slika 3.5 Pogled iz perspektive bolida u aplikaciji *Formula XR*

Glavna funkcionalnost koja treba biti omogućena u ovoj fazi je gibanje bolida po stazi na temelju podataka dobivenih FastF1 paketom. Kako bi se to omogućilo, najprije je svakom bolidu pridružena jedinstvena oznaka (engl. *tag*) koja je jednaka kratici vozačevog prezimena (npr. oznaka VER za bolid Maxa Verstappena). Zatim je napisana C# skripta *CsvReader* koja za svakog vozača dohvaća CSV datoteke s vozačevim vremenima i sprema ih u strukture podataka preko kojih ih je kasnije jednostavno koristiti u kôdu. CSV datoteke dijele isto ime kao i oznaka pridružena svakom bolidu pa ih je jednostavno dohvatiti preko te oznake.

Gibanje bolida po stazi omogućeno je komponentom Bézier Path Creator, preuzetom s Unity Asset Storea. Komponenta omogućuje crtanje Bézierovih krivulja u Unity sceni. Za svaki od dvadeset bolida, Bézierovim krivuljama nacrtana je putanja koju će bolid pratiti u svim krugovima utrke. Putanja jednog kruga sastoji se od pet različitih Bézierovih krivulja: za svaki od tri sektor jedna krivulja, te po jedna krivulja za prvi i treći sektor u slučaju kada bolid ide na *pit-stop.* Preko komponente Bézier Path Creator može se odrediti duljina svake krivulje. Jednom kada su poznate duljine krivulja, odnosno duljine putanja po kojima se bolidi gibaju te vremena potrebna bolidima da prijeđu te putanje, jednostavno je odrediti brzinu kojom se bolid treba gibati na toj putanji. Brzina se računa kao količnik duljine prijeđenog puta i vremena potrebnog za prijelaz tog puta. Izračunata brzina potom se u *Update* funkciji koristi za izračun točke na krivulji u kojoj se bolid nalazi u svakom trenutku utrke (kôd 3.2). Ovaj izračun u velikoj mjeri olakšavaju unaprijed razvijene funkcije *GetPointAtDistance* i *GetRotationAtDistance* iz paketa Bézier Path Creator. Ovime je omogućeno da virtualni bolid započne i završi krug u istom trenutku kao i stvarni, odnosno da se stvarna utrka vjerno prikaže u virtualnom okruženju.

```
1   void Update()
2   {
3       distanceTravelled += speed * Time.deltaTime;
4
5       transform.position = active.path
6           .GetPointAtDistance(distanceTravelled);
7
8       transform.rotation = active.path
9           .GetRotationAtDistance(distanceTravelled);
10  }
```

Kôd 3.2 *Update* funkcija skripte koja omogućuje gibanje bolida po stazi

Na kraju svakog sektora na virtualnoj stazi postavljen je nevidljivi zid koji služi kao okidač (engl. *trigger*). Kada bolid prođe kroz ovaj zid, skripta dobiva informaciju da je bolid završio prethodni sektor. U tom trenutku skripta dohvaća duljinu idućeg sektora i vrijeme potrebno da bolid prođe taj sektor te računa novu brzinu kojom će se bolid gibati kroz po sektoru. Također, ako se radi o krugu u kojem bolid odlazi na *pit-stop*, skripta ga pravilno usmjerava. Sumirano, jedan će bolid se kroz cijelu utrku gibati po istoj putanji. Razlika je samo u brzini gibanja ovisno o sektoru i broju kruga.

U ovoj fazi simulirana je i procedura početka utrke, omogućeno je premotavanje, pauziranje i gledanje utrke u usporenom prikazu te dodan je pozadinski zvuk. U stvarnoj F1 utrci, nakon što zauzmu svoje mjesto na startnoj rešetci, vozači prate svjetla na semaforu ispred njih. Kada se svih pet svjetala na semaforu upale i potom ponovo ugase, utrka započinje. Sličan sistem je implementiran i u aplikaciji *Formula XR*. U pozadini slike 3.5 nalazi se semafor s pet svjetala. Na pritisak gumba na korisničkom sučelju (opisanom u potpoglavlju 3.2.2), svjetla na virtualnom semaforu počinju se paliti jedno po jedno. Kada su svih pet svjetala upaljena, ostaju upaljena jednu sekundu, a zatim se ugase, čime signaliziraju početak utrke.

Premotavanje, pauziranje i gledanje utrke u usporenom prikazu omogućeno je manipulacijom varijable *Time.timeScale*. Unity koristi vrijeme u raznim izračunima, uključujući animacije, fizičke izračune pa i konkretno izračun brzine gibanja virtualnih bolida po stazi. Manipulacijom *Time.timeScale* može se kontrolirati brzina protoka vremena u Unity aplikacijama te tako ubrzati, usporiti ili zaustaviti izračune i odvijanje akcija u aplikacijama. U aplikaciji *Formula XR*, utrka se može ubrzati tri puta, dvostruko usporiti ili u potpunosti zaustaviti postavljenjem varijable *Time.timeScale* na nula. Za svaku od navedenih akcija napravljena je C# funkcija koja se može pozvati preko gumba na korisničkom sučelju.

Za postizanje što bolje imerzije, u pozadinu aplikacije dodan je zvuk utrke Formule 1. Također, kako aplikacija omogućuje gledanje utrke u VR okolini s virtualne tribine, dodan je zvuk koji imitira Dopplerov efekt prolaska F1 bolida pored tribine. Ovaj zvuk korisnik čuje samo ako utrku prati u VR varijanti iz perspektive navijača na tribini, a postignut je tako što su na stazu, neposredno prije tribine na kojoj se korisnik nalazi, postavljeni dodatni nevidljivi zidovi koji služe kao okidači. Kada bolid prođe kroz takav nevidljivi zid, reproducira se zvuk Dopplerovog efekta prolaska stvarnog F1 bolida pored tribine.

3.2. Integracija XR elemenata u aplikaciju

Do sada napravljena aplikacija omogućuje simulaciju utrke, ali ne sadrži nikakve XR elemente. Ciljevi ove faze su sljedeći:

- omogućiti korisniku gledanje utrke u XR-u na Meta Quest 3 uređaju,
- omogućiti interakciju korisnika i utrke putem korisničkog sučelja,
- virtualnim ekranima pružiti korisniku pogled na utrku iz perspektive bolida i
- omogućiti prijelaz iz dograđene u virtualnu stvarnost pružajući korisniku osjećaj prisutnosti na virtualnoj tribini.

Za jednostavnije ostvarivanje navedenih funkcionalnosti koristi se Meta Interaction SDK paket preuzet s Unity Asset Storea i Meta Building Blocks, tj. gradivni elementi XR aplikacija namijenjenih za uređaje kompanije Meta.

3.2.1. Konfiguracija XR scene

U postojeći Unity projekt najprije je potrebno uvesti pakete Meta XR All-in-One SDK i Meta XR Interaction SDK OVR Samples. Pri samom uvozu Meta XR All-in-One SDK paketa, Unity će samostalno konfigurirati ispravne postavke projekta tako da one odgovaraju za razvoj XR aplikacija i za puštanje aplikacija u pogon na uređaje kompanije Meta. Programer samostalno jedino mora prebacit opciju *Platform* pod *File > Build Settings* na *Android* i odabrati opciju *Oculus* pod *Edit > Project Settings > XR Plug-in Management* za Android operacijske sustave. Konfiguracijom ovih postavki, scena je spremna za razvoj XR aplikacija.

Meta Building Blocks (slika 3.6) je skup gotovih komponenti za ubrzani razvoj XR aplikacija, a rade na principu ubacivanja komponenti u scenu. Aplikacija *Formula XR* koristi sljedeće komponente: *Camera Rig, Passthrough, Controller Tracking, Hand Tracking, Poke Interaction* i *Ray Interaction*. Osim navedenih Meta Building Blocks komponenti, u aplikaciji *Formula XR* korišteni su gotove komponente i C# skripte iz primjera scena paketa Meta XR Interaction SDK OVR Samples.

Komponenta *Camera Rig* središnja je komponenta svih XR aplikacija. Sastoji se od tri virtualne kamere preko kojih se korisniku na HMD-u prikazuje sadržaj scene. Ova

komponenta predstavlja korisnika u Unity sceni. Osim što mu prikazuje sadržaj scene, komponenta prati promjenu kuta gledanja korisnika i promjenu njegova položaja u stvarnom prostoru te analogno pomiče i rotira njegovu reprezentaciju u sceni. *Camera Rig* je središnja komponenta XR aplikacija jer se sve ostale XR komponente povezuju na nju te ne mogu postojati u sceni ako ona ne postoji.

Passthrough komponenta omogućuje korisnicima da uz sadržaj virtualne scene vide i stvarnu okolinu u kojoj se nalaze. Kada bi se u praznu Unity scenu postavila samo komponenta *Camera Rig*, korisnik koji koristi HMD uređaj bi oko sebe vidio samo mrak. Ako se u scenu doda i komponenta *Passthrough,* korisnik bi tada oko sebe vidio stvarnu okolinu, kao da ne koristi HMD uređaj. Ubacivanjem virtualnih elemenata u scenu može se postići da ti elementi iz korisnikove perspektive izgledaju kao dio stvarne okoline. U aplikaciji *Formula XR*, *Passthrough* komponenta je potrebna jer je ideja aplikacije da korisnik paralelno prati stvaran prijenos utrke i utrku na virtualnoj stazi.

Komponente *Controller Tracking* i *Hand Tracking* služe aplikaciji za praćenje Meta Quest 3 kontrolera i korisnikovih ruku. Ideja aplikacije *Formula XR* omogućiti je korisniku interakciju s virtualnim elementima putem kontrolera, ali i bez njih, odnosno samo rukama ako to korisnik preferira. Aplikacija preko stvarnih ruku i kontrolera postavlja njihove virtualne inačice preko kojih je omogućena interakcija korisnika i virtualnih objekta. U jednom trenutku aplikacija dopušta korištenje samo kontrolera ili samo ruku. Čim korisnik barem jednom rukom primi kontroler, aplikacija više ne dopušta interakciju stvarnih ruku i virtualnih objekata sve dok korisnik ne odloži kontroler.

Komponente *Poke Interaction* i *Ray Interaction* proširuju mogućnosti interakcije korisnika i virtualnih elemenata. *Poke Interaction* omogućuje korisnicima da stvarnim rukama pritišću virtualne gumbe na korisničkom sučelju. Ako je u scenu ubačena komponenta *Ray Interaction* iz virtualnih ruku i kontrolera koji su postavljeni preko njihovih stvarnih inačica izlazi zraka pomoću koje korisnici mogu pritiskati gumbe korisničkog sučelja iz daljine ili dohvatiti elemente koji se ne nalaze nužno u njihovom dosegu.

Pri ubacivanju opisanih komponenta u Unity scenu, bitno je jedino da se najprije ubaci komponenta *Camera Rig*, a potom sve ostale komponente proizvoljnim redom. Komponente izrađene u prošloj fazi, odnosno virtualna staza po kojoj se gibaju virtualni bolidi može se trenutačno ukloniti iz scene. Od tih komponenti potrebno je izraditi skup komponenti, odnosno predložak u Unityju (engl. *prefab*), koji će korisnik moći postaviti na proizvoljnu poziciju unutar svoje stvarne okoline. Kada bi se predložak s virtualnom

stazom poravnao sa stvarnim svijetom, staza bi bila dugačka i do nekoliko kilometara. Zbog toga, predložak je potrebno smanjiti oko 1000 puta kako bi cijela virtualna staza s bolidima stala na običan radni stol. Rezultat ovog koraka je scena u kojoj se nalaze samo Meta Building Blocks komponente. Kada bi korisnik pokrenuo ovakvu aplikaciju, oko sebe bi vidio samo svoju stvarnu okolinu i virtualne ruke ili kontrolere u njoj. Cilj idućeg koraka je omogućiti korisniku postavljanje virtualne staze u njegovu stvarnu okolinu i upravljanje utrkom putem korisničkog sučelja.

3.2.2. Interakcija korisnika s virtualnom utrkom

Pri planiranju izrade aplikacije *Formula XR* postojalo je više ideja kako pozicionirati virtualnu stazu na željenu poziciju u stvarnom prostoru. Jedna ideja bila je postavljanje staze korištenjem fizičkih markera poput QR koda. Uporabom određenog softvera, npr. Vuforije,[11] kada bi korisnik pogledao prema markeru, aplikacija bi prepoznala marker i njegovu orijentaciju te na marker „zalijepila“ virtualnu stazu. Iako je ovaj pristup vrlo jednostavan i praktičan jer se virtualnu stazu može jednostavno pomaknuti pomicanjem fizičkog markera, paket Meta Interaction SDK nudi veliki broj mogućnosti koji mogu poslužiti kao alternativa ovom pristupu te nije potrebno posezati za još jednim softverom za rješavanje ovog problema. Drugi pristup bio bi postavljanje staze uporabom virtualne zrake koja izlazi iz virtualnih ruku ili kontrolera. Od ovog pristupa se odustalo jer korisnik u ponekim situacijama ne može prepoznati nalazi li se kraj virtualne zrake iznad stvarnog stola ili ispod njega te ponekad bi se dogodilo da korisnik postavi virtualnu stazu djelomično ispod stola.

Pristup koji je korišten u aplikaciji *Formula XR* omogućuje postavljanje virtualne staze i kontrolerima i rukama. Za postavljanje staze kontrolerima potrebno je pozicionirati desni kontroler na mjesto gdje se želi postaviti stazu i pritisnuti *Trigger* gumb na njemu. Aplikacija pomoću komponenata *Camera Rig* i *Controller Tracking* konstantno prati položaj kontrolera te jednostavno putem skripte može instancirati predložak s virtualnim elementima na toj poziciji. Za postavljena staze rukama, odnosno bez kontrolera, koristi se gotova komponenta za prepoznavanje gesta ruku iz paketa Meta XR Interaction SDK OVR

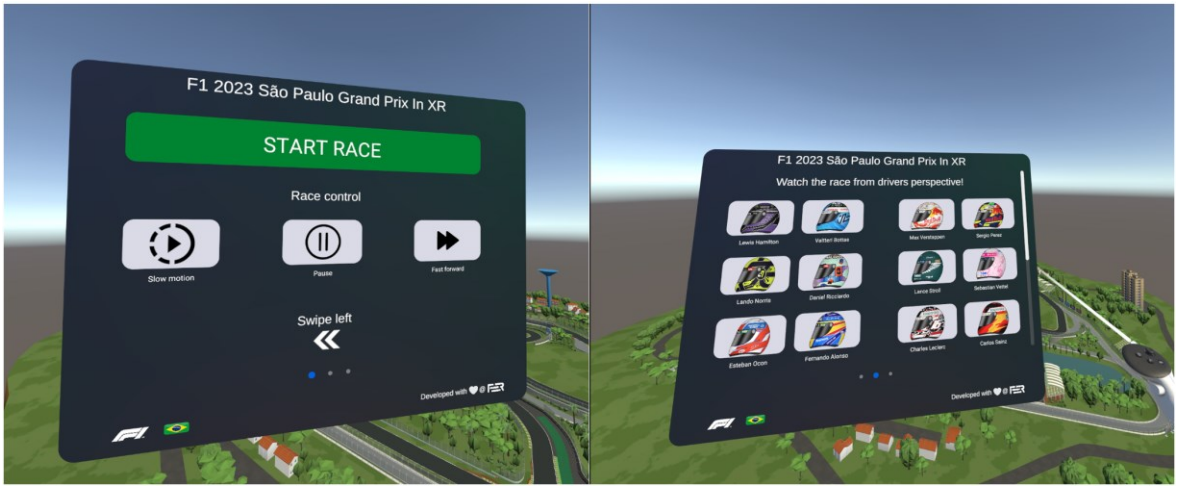
[11] https://developer.vuforia.com/home

Samples. Ova komponenta na temelju pozicije i razmaka između prstiju te njihovom stupnju savinutosti može prepoznati razne geste i pokrete korisnika te inicirati određenu akciju kada prepozna gestu. Ideja postavljanja staze rukama je da korisnik uperi desni kažiprst na poziciju na koju želi pozicionirati stazu te na lijevoj ruci podigne palac prema gore, odnosno napravi tzv. *thumbs-up* gestu. Kada aplikacija prepozna *thumbs-up* gestu na lijevoj ruci, instancirati će virtualnu stazu na mjestu na koje pokazuje korisnikov desni kažiprst. U aplikaciju je dodan i virtualni pano na kojem se nalaze upute u obliku kratkog videozapisa i teksta koje objašnjavaju korisniku kako postaviti virtualnu stazu u svoju stvarnu okolinu.

Osim postavljanja staze u stvaran prostor, aplikacija mora korisniku omogućiti i interakciju sa samom utrkom, odnosno pokretanje, premotavanje i pauziranje utrke te gledanje utrke iz perspektiva bolida i navijača na tribini. Za iniciranje navedenih akcija, izrađeno je korisničko sučelje u obliku lebdećeg virtualnog tableta. Virtualni tablet postoji kao gotova komponenta u jednoj od scena paketa Meta XR Interaction SDK OVR Samples. Na toj komponenti nalaze se skripte koje ju čine vrlo sličnom stvarnom tabletu. Sadržaj na tabletu podijeljen je na nekoliko stranica i može se pomicati (engl. *scroll*) horizontalno i vertikalno. Također, na tabletu se nalaze sve potrebne skripte koje omogućuju korisniku hvatanje, rotiranje i pomicanje tableta po sceni. Korisnik gumbe na tabletu može pritiskati prstima kao i gumbe na stvarnom tabletu ili kontrolerima tako da uperi zraku koja izlazi iz kontrolera prema gumbu na tabletu te pritisne *Trigger* gumb na kontroleru. Dodatno, uporabom *Ray Interaction* komponente, korisnik i bez kontrolera može pritiskati gumbe na daljinu tako da zrakom koja izlazi iz virtualnog kažiprsta nacilja gumb na tabletu te spoji kažiprst i palac, odnosno napravi tzv. *pinch* gestu.

Gotova komponenta tableta modificirana je kako bi odgovarala aplikaciji *Formula XR*. Virtualni tablet pridružen je predlošku sa stazom i bolidima te se pojavljuje u sceni tek kada korisnik postavi predložak u svoju stvarnu okolinu. Sadržaj tableta podijeljen je u tri logičke cjeline, odnosno na tri stranice. Na prvoj stranici nalaze se gumbi za pokretanje utrke te za premotavanje, pauziranje i gledanje utrke u usporenom prikazu (slika 3.6, lijevo). Na drugoj stranici nalazi se po jedan gumb za svakog vozača koji sudjeluje u utrci. Na svakom gumbu je slika kacige određenog vozača, a ispod gumbe njegovo ime i prezime (slika 3.6, desno). Pritiskom na gumb otvara se virtualni pano s pogledom na utrku iz perspektive bolida tog vozača. Treća stranica sadrži gumbe koji iniciraju prijelaz iz dograđene stvarnosti u virtualnu stvarnost u kojoj se korisnik nalazi na virtualnoj tribini.

Na svakom gumbu treće stranice nalazi se slika tribina na koju se korisnik može pozicionirati. Sumirano, nakon što korisnik postavi virtualnu stazu u svojem stvarnom prostoru, sva daljnja interakcija s utrkom omogućena je preko tri navedene stranice virtualnog tableta koji služi kao korisničko sučelje.

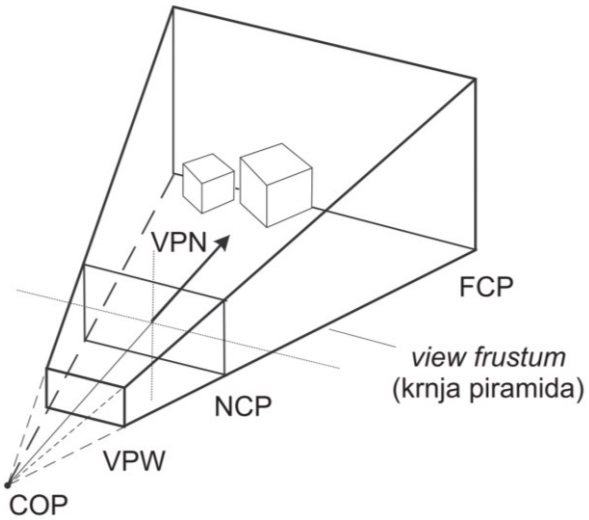


Slika 3.6 Prva i druga stranica virtualnog tableta, prikazane u Unity sceni

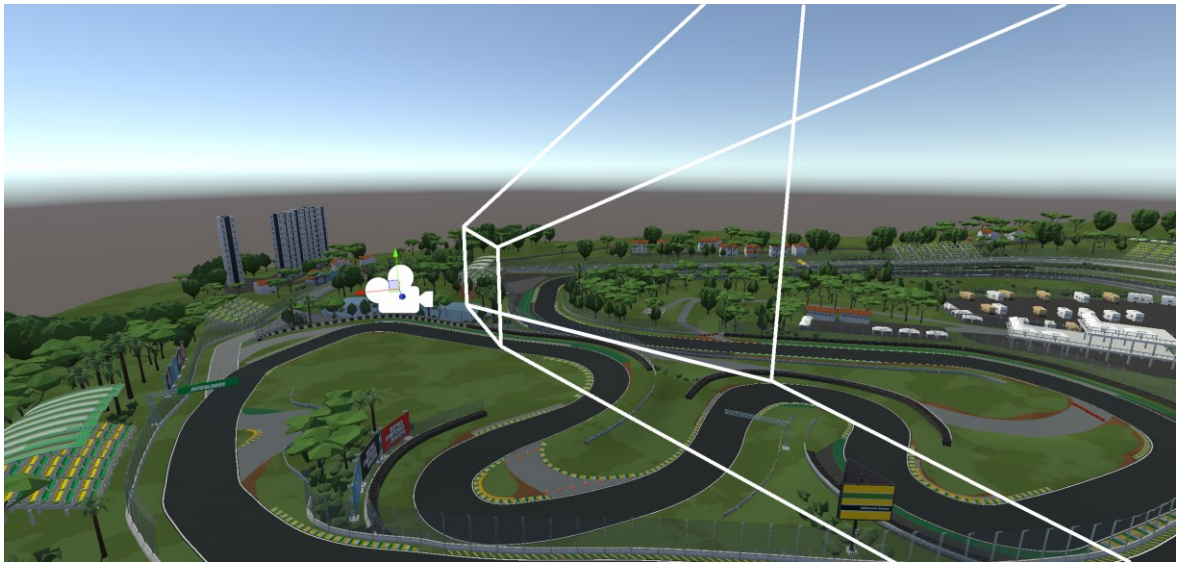
3.2.3. Pogled iz perspektive bolida

Kao što je već objašnjeno, pogled iz perspektive bolida prikazivat će se na virtualnim panoima, sličnim virtualnom tabletu. Za izradu virtualnog panoa može poslužiti isti objekt iz kojeg je izrađen virtualni tablet. S objekta se samo trebaju ukloniti svi gumbi i sav tekst te dodati objekt koji predstavlja virtualni ekran na kojem će se prikazivati utrka iz perspektive F1 bolida. Na virtualni ekran potrebno je dodati komponentu *Render Texture* na koju kamere u Unityju mogu projicirati scenu. Osim ekrana, na virtualni pano dodan je i gumb za uklanjanje panoa te komponenta za prikaz teksta na kojoj će biti ispisano ime i prezime vozača bolida čija se perspektiva prikazuje na ekranu. U prijenosu stvarnih F1 utrka postoje aplikacije koje omogućuju korisniku prikaz više pogleda na utrku istovremeno. Tako na primjer, korisnici mogu gledati utrku iz perspektive više vozača odjednom. Kako bi aplikacija *Formula XR* pružala sličnu funkcionalnost, u njoj je omogućeno otvaranje do četiri virtualnih panoa, odnosno istovremeno praćenje utrke iz perspektive četiri različitih bolida uz naravno i praćenje utrke na virtualnoj reprezentaciji staze Interlagos.

Ideja ove funkcionalnosti je jednostavna. Na svaki virtualni bolid već je od ranije postavljena jedna kamera koja projicira pogled iz bolida. Ideja je putem C# skripte prikazivati pogled kamere koju korisnik odabere na jednom od četiri virtualnih panoa. Međutim, problem je nastao u trenutku kada se virtualna staza smanjila oko 1000 puta kako bi stala u korisnikovu stvarnu okolinu. Naime, prikazivanje pogleda u virtualnu scenu uporabom kamera u Unityju zasniva se na perspektivnoj projekciji u kojoj se scena projicira na projekcijsku plohu [4]. Parametri perspektivne kamere (slika 3.7) su središte projekcije (engl. *center of projection*, skr. COP), projekcijski prozor (engl. *view-plane window*, skr. VPW), normala na projekcijsku plohu u (engl. *view- plane normal*, skr. VPN) te bliska i daleka odrezujuća ploha (engl. *near and far clipping plane*, skr. NCP i FCP). Središte projekcije je točka iz koje se promatra scena, a projekcijski prozor određuje gdje se slika scene projicira. Bliska i daleka odrezujuća ploha zajedno stvaraju krnju piramidu nazvanu *view frustum* te svi objekti koji se nalaze unutar te piramide, iscrtavaju se na ekran. Problem nastaje zato što je u Unityju minimalna udaljenost između središta projekcije i bliske odrezujuće plohe 0.01 Unity jedinica, dok je promjer virtualne staze nedovoljno veći. Iz tog razloga, kada se kamera postavi na virtualni bolid, taj bolid ne upada u *view frustum* jer je preblizu središtu projekcije te se posljedično ne iscrtava na ekranu. Slika 3.8 prikazuje odnos virtualne staze Interlagos te krnje piramide koju čine bliska i daleka odrezujuća ploha (prikazana bijelim linijama ispred kamere). Iako je promjer virtualne staze veći od udaljenosti središta projekcije i bliske odrezujuće plohe, veliki broj virtualnih objekata koji su blizu kamere, odnosno središta projekcije, ne upada u *view frustum* te se ne iscrtava na ekranu. Iz navedenih razloga, nemoguće je prikazivati pogled iz perspektive bolida na ovaj način.



Slika 3.7 Parametri perspektivne kamere, preuzeto iz [4]



Slika 3.8 Odnos *view frustuma* i virtualne staze Interlagos

Kako bi se riješio navedeni problem, u scenu je potrebno ubaciti predložak s virtualnom stazom i bolidima na njoj u njegovoj početnoj, odnosno neumanjenoj veličini. Na takvom predlošku više nema problema s perspektivom kamere jer je promjer virtualne staze uveliko veći od udaljenosti između središta projekcije i bliske odrezujuće plohe te svi objekti prema kojima je okrenuta kamera upadaju u njen *view frustum*. Kako korisnik u svojoj stvarnoj okolini ne bi vidio virtualnu stazu u njenoj početnoj veličini, u Unityju je potrebno izraditi novi sloj (engl. *layer*) i primijeniti ga na sve objekte navedenog predloška. Zatim je potrebno u sceni pronaći glavnu XR kameru koja je u hijerarhiji dijete komponente *Camera Rig*. Pod postavkom *Culling Mask* XR kamere, potrebno je isključiti novo napravljeni sloj. Postavka *Culling Mask* određuje slojeve koje kamera iscrtava te kada se isključi sloj primijenjen na predložak sa stazom u stvarnoj veličini, korisnik ga neće vidjeti u svojoj stvarnoj okolini. Važno je napomenuti kako na kamerama koje su postavljene na virtualne bolide, navedeni sloj mora ostati uključen kako bi kamere mogle iscrtati stazu iz perspektive bolida. Nedostatak ovog pristupa je što aplikacija istovremeno mora simulirati utrku na dvije virtualne staze – onoj početne veličine i na umanjenoj, što može narušiti performanse aplikacije.

Pomoću gumba na drugoj stranici virtualnog tableta, odnosno korisničkog sučelja, mogu se otvoriti i zatvoriti virtualni panoi. Tu funkcionalnost omogućuje C# skripta koja najprije provjerava postoji li pano na kojem se trenutno ne prikazuje utrka. Ako postoji takav pano,

skripta u sceni funkcijom *GameObject.Find* pronađe kameru na bolidu kojeg je korisnik odabrao te preko postavke *Target Texture* kameri dodjeljuje teksturu panoa na koju će kamera iscrtavati pogled iz perspektive F1 bolida. Ako se već na sva četiri panoa prikazuje utrka, skripta će nasumično odabrati jedan pano i na njemu početi prikazivati pogled iz novoizabranog bolida. Skripta je također zadužena za ispisivanje imena i prezimena vozača bolida čija se perspektiva prikazuje u za to predviđeno polje. Slika 3.9 prikazuje virtualnu stazu sa sva četiri aktivna panoa na kojima se vidi utrka iz perspektive F1 trkaćih automobila u Unity sceni.

Slika 3.9 Virtualna staza s panoima koji prikazuju pogled iz perspektive bolida u Unity sceni

3.2.4. Prijelaz u virtualnu stvarnost

Jedna od funkcionalnosti aplikacije *Formula XR* je prijelaz iz miješane stvarnosti, u kojoj korisnik gleda utrku na umanjenom prikazu staze u svojoj stvarnoj okolini, u virtualnu stvarnost koja korisniku pruža osjećaj prisutnosti u potpuno virtualnoj okolini, bez dodira sa stvarnim svijetom. Kada prijeđe u virtualnu stvarnost, korisnik prati utrku s virtualne tribine. Aplikacija *Formula XR* omogućuje prijelaz na šest različitih tribina koje se protežu cijelom dužinom staze, tj. korisniku je omogućeno biranje tribine s koje želi pratiti utrku.

Kako bi se navedena funkcionalnost implementirala, na svaku je virtualnu tribinu potrebno postaviti novu *Camera Rig* komponentu. *Camera Rig* komponente nije moguće umanjiti, stoga ih je potrebno postaviti na tribine virtualne staze koja nije umanjena kako bi stala u stvarnu okolinu korisnika. U sceni ne smije postojati više *Camera Rig* komponenti koje su

istovremeno aktivne. Do sada je u sceni postojala jedna takva komponenta pomoću koje korisnik u miješanoj stvarnosti, tj. u svojoj stvarnoj okolini vidi virtualnu stazu (dalje u tekstu komponenta *MR Camera Rig*). Zbog toga, pri pokretanju aplikacije, nove *Camera Rig* komponente (dalje u tekstu *VR Camera Rig* komponente) moraju biti deaktivirane kako bi aplikacija znala pogled koje kamere treba iscrtavati na ekranu HMD-a. Određena *VR Camera Rig* komponenta bit će aktivirana kada korisnik preko korisničkog sučelja zatraži prijelaz u virtualnu stvarnost. U tom trenutku, aplikacija mora deaktivirati komponentu *MR Camera Rig* kako ne bi došlo do naglog zaustavljanja aplikacije. S *VR Camera Rig* komponenti mora se ukloniti skripta *OVR Manager*. Ova skripta je upravlja *Camera Rig* komponentom te omogućuje iscrtavanje scene na HMD uređaj. U Unity sceni smije postojati točno jedna instanca *OVR Manager* skripte i jednom kada se deaktivira, ne može se više aktivirati bez ponovnog pokretanja aplikacije. Zbog toga, ova skripta je postavljena samo na *MR Camera Rig* komponentu i ne smije se deaktivirati tijekom rada aplikacije.

Prijelaz iz miješane u virtualnu stvarnost treba biti blag kako bi se kod korisnika postiglo što bolje korisničko iskustvo. Kada bi prijelaz bio brz i oštar, poput promjene scene u filmu, kod korisnika bi se mogao izazvati osjećaj dezorijentiranosti i mučnine. Ideja blagog prijelaza je sljedeća: najprije se pomoću objekta na koji je postavljen crni materijal postepeno zatamnjuje kamera na *MR Camera Rig* komponenti. Zatim se deaktivira ta kamera i aktivira kamera na jednoj od *VR Camera Rig* komponenti. Kamera *VR Camera Rig* komponente u početku prikazuje samo crni ekran koji postepeno nestaje, nakon čega kamera prikazuje pogled na stazu s virtualne tribine. Iz perspektive korisnika, ovaj prijelaz izgleda kao postepeno zatvaranje i otvaranje očiju, pri čemu se korisnik iz stvarne okoline premješta u virtualni svijet.

Postepeno zatamnjivanje i osvjetljivanje ekrana postignuto je objektom koji je postavljen preko kamere u Unity sceni. Na objekt je postavljen potpuno crni materijal te na početku prijelaza korisnik iz svoje perspektive vidi samo crni ekran. Na objekt je također dodana napisana C# skripta *CameraFadeScript* koja putem Unity korutine (engl. *corutine*) može mijenjati alfa vrijednost materijala, odnosno njegovu razinu transparentnosti. Mijenjanjem razine transparentnosti objekta koji se nalazi preko kamere, kod korisnika se postiže opisani efekt postepenog otvaranja, odnosno zatvaranja očiju. Korutina *FadeRoutine* (kôd 3.3) napisana je po uzoru na upute YouTube kreatora Valem Tutorilas [23] i Sunny Valley Studio [24]. Skripta materijalu koji se nalazi na objektu mijenja razinu transparentnosti u

redcima 6 i 7, a u retku 10 zaustavlja izvođenje korutine do sljedećeg Unity okvira (engl. *frame*) čime se postiže privid postepenog pojavljivanja ili nestajanje objekta. Proces pojavljivanja ili nestajanja objekta traje tri sekunde. Manipulacijom parametra *aIn* i *aOut* određuje se hoće li se objekt postepeno pojavljivati ili nestajati. a u skripti su napisane i javne metode *FadeIn* i *FadeOut* koje pozivaju korutinu *FadeRoutine* s određenom kombinacijom navedenih parametara za pojavljivanje ili nestajanje objekta.

```
1   private IEnumerator FadeRoutine(float aIn, float aOut)
2   {
3       float timer = 0;
4       while (timer <= fadeDuration)
5       {
6           float tempVal = Mathf.Lerp(aIn, aOut, timer / fadeDuration);
7           componentRenderer.material.SetFloat("_Alpha", tempVal);
8
9           timer += Time.deltaTime;
10          yield return null;
11      }
12
13      Color finalColor = fadeColor;
14      finalColor.a = alphaOut;
15      componentRenderer.material.SetColor("_Color", finalColor);
16  }
```

Kôd 3.3 Korutina *FadeRoutine*

Sama funkcionalnost prijelaza iz miješane u virtualnu stvarnost također se odvija u korutini (kôd 3.4) zato što skripta treba pričekati da se objekt koji je postavljen preko kamere na *MR Camera Rig* komponenti u potpunosti zatamni. Osim poziva *FadeOut* funkcije koja će zatamniti navedeni objekt (redak 3), korutina prije iniciranja prijelaza treba deaktivirati *Passthrough* komponentu (redci 4 i 5). Kao što je već objašnjeno, *Passthrough* komponenta omogućuje XR uređaju da tijekom korištenja aplikacije uz virtualne objekte prikazuje i stvarnu okolinu. U varijanti kada korisnik gleda utrku u virtualnoj stvarnosti, *Passthrough* komponenta nije potrebna jer se kod korisnika pokušava postići što veća imerzija u virtualnoj okolini. Korutina ne deaktivira *Passthrough* komponentu direktno već preko varijable skripte *OVR Manager*. Nakon deaktiviranja ove komponente, skripta mora deaktivirati i kameru na komponenti *MR Camera Rig* (redak 8). Skripta ne smije deaktivirati kompletnu *MR Camera Rig* komponentu jer bi tad deaktivirala i *OVR Manager*

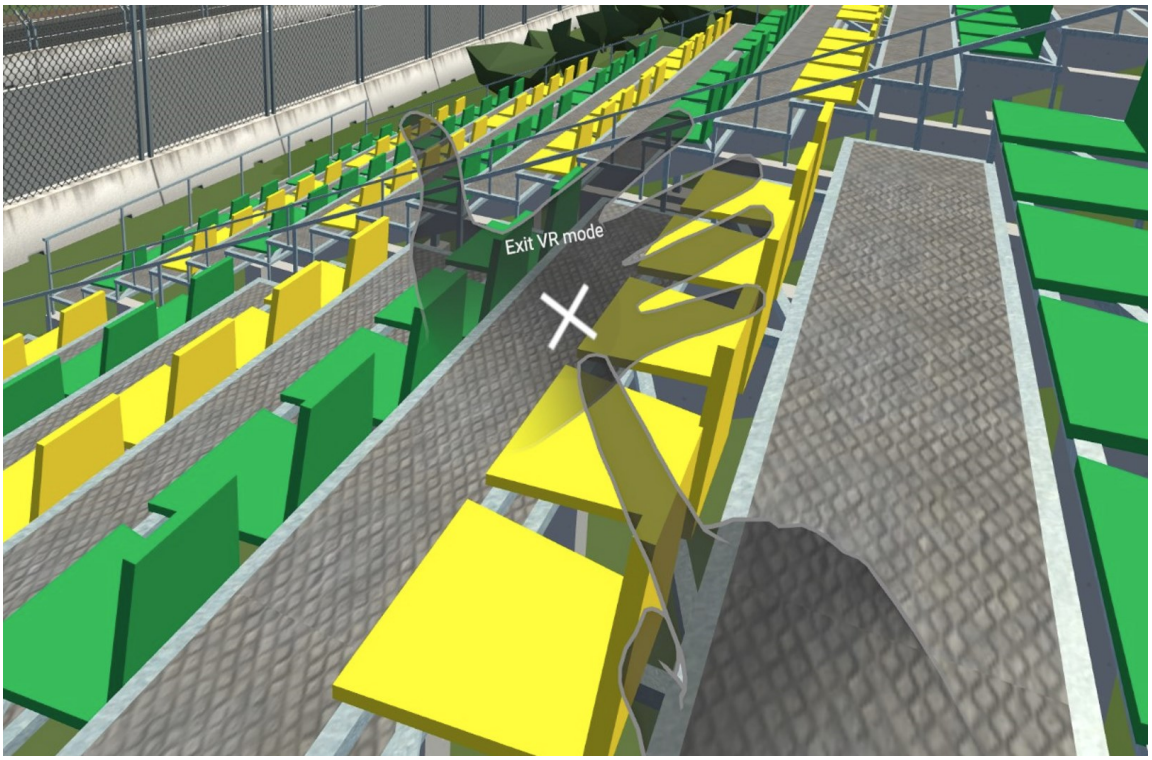
skriptu te aplikacija dalje ne bi funkcionirala. Zadnji korak u korutini je aktivacija jedne od *VR Camera Rig* komponenti (redak 10) čime se završava prijelaz iz miješane u virtualnu stvarnost. Korutinu *SwitchToVr* aktivira druga funkcija u istoj skripti, a tu funkciju korisnik poziva direktno preko gumba na trećoj stranici virtualnog tableta.

```
1   private IEnumerator SwitchToVr(GameObject vrCamera)
2   {
3       mrCameraFader.GetComponent<CameraFadeScript>().FadeOut();
4       mrCamera.GetComponent<OVRManager>()
5               .isInsightPassthroughEnabled = false;
6
7       yield return new WaitForSeconds(3);
8       mrCamera.GetComponent<OVRCameraRig>().enabled = false;
9
10      vrCamera.SetActive(true);
11      activeVrCamera = vrCamera;
12  }
```

Kôd 3.4 Korutina *SwitchToVr* za prijelaz iz miješane u virtualnu stvarnost

Aplikacija *Formula XR* omogućuje i povratak iz virtualne u miješanu stvarnost. Dok se nalazi na virtualnoj tribini, korisnik više ne vidi stvarne kontrolere ili svoje ruke, ali vidi njihove virtualne reprezentacije. U virtualnoj stvarnosti, korisnik također više nema pristup virtualnom tabletu, odnosno korisničkom sučelju sve dok se ne vrati u miješanu stvarnost. Korisnik povratak može inicirati pritiskom *Trigger* gumba na lijevom kontroleru ili uporabom izbornika u lijevom virtualnom dlanu (slika 3.10). Izbornik u dlanu (engl. *palm menu*) također je jedna od već unaprijed postojećih komponenti paketa Meta XR Interaction SDK OVR Samples te je modificirana tako da pokazuje samo jedan gumb koji služi za povratak iz virtualne u miješanu stvarnost. Korisnik izbornik na virtualnom dlanu može otvoriti tako da usmjeri lijevi dlan prema sebi te spoji lijevi kažiprst i palac, odnosno napravi *pinch* gestu. Pritiskom na gumb na izborniku u dlanu ili na kontroleru, aktivira se korutina *SwitchBackToMr* (kôd 3.5) koja pokreće obrnuti postupak od korutine *SwitchToVr*. Korutina najprije u sceni funkcijom *GameObject.Find* pronalazi objekt postavljen preko kamere komponente *VR Camera Rig* i inicira postupak zatamnjivanja ekrana (redci 3 i 7). Zatim, korutina deaktivira kompletnu *VR Camera Rig* komponentu (redak 10) te aktivira *Passthrough* komponentu i kameru komponente *MR Camera Rig*

(redci 13, 14 i 16) čime vraća korisnika u miješanu stvarnost, odnosno u okolinu u kojoj je svjestan svog stvarnog okruženja, ali i F1 utrke prikazane virtualnim elementima.



Slika 3.10 Izbornik u virtualnom dlanu s gumbom za povratak u miješanu stvarnost

```
1   private IEnumerator SwitchBackToMr()
2   {
3       GameObject currentVrCameraFader = GameObject.Find(
4           activeVrCamera.name + "Fader"
5       );
6
7       currentVrCameraFader.GetComponent<CameraFadeScript>().FadeOut();
8       yield return new WaitForSeconds(3);
9
10      activeVrCamera.SetActive(false);
11      activeVrCamera = null;
12
13      mrCamera.GetComponent<OVRManager>()
14              .isInsightPassthroughEnabled = true;
15
16      mrCamera.GetComponent<OVRCameraRig>().enabled = true;
17  }
```

Kôd 3.5 Korutina *SwitchBackToMr* za vraćanje iz virtualne u miješanu stvarnost

4. Prikaz rada aplikacije *Formula XR*

U ovom poglavlju objašnjene su upute za korištenje aplikacije *Formula XR* te su opisane i sumirane sve njene funkcionalnosti. Nakon pokretanja aplikacije na Meta Quest 3 uređaju, korisnik koji koristi uređaj vidi virtualni pano u svojoj stvarnoj okolini. Meta Quest 3 uređaj sliku koju simulira računalo prikazuje tehnikom video miješanja. Kvaliteta slike koju prikazuje uređaj uglavnom je visoka, no može biti zrnata ili pomalo mutna ako se korisnik nalazi u prostorijama s premalo ili previše svjetla. Na virtualnom panou nalaze se upute za postavljanje virtualne staze na kojoj se odvija simulirana utrka, u korisnikovo stvarno okruženje. Korisnik virtualnu stazu može postaviti u svoju okolinu kontrolerima, pritiskom na *Trigger* gumb desnog kontrolera ili rukama, tako da desnim kažiprstom pokaže na koje mjesto želi postaviti stazu i napravi *thumbs-up* gestu lijevom rukom (slika 4.1).

Slika 4.1 Gesta za postavljanje virtualne staze u stvarnu okolinu

Na poziciju na kojoj se nalazi desni kontroler ili na koju korisnik pokazuje desnim kažiprstom, stvorit će se virtualna staza i virtualni bolidi koji su poslagani na startnu rešetku staze. Iznad staze, nalazi se virtualni tablet, odnosno korisničko sučelje koje omogućuje interakciju korisnika sa simuliranom utrkom. Pri stvaranju u sceni, virtualni

tablet uvijek će biti orijentiran prema korisniku. Korisnik virtualni tablet može rotirati i premještati po sceni. Preporučuje se pomaknuti tablet od staze kako ne bi zaklanjao pogled na samu utrku. Na slici 4.2 prikazana je staza s bolidima i tabletom koja se stvori na mjestu koje je korisnik odabrao.

Slika 4.2 Virtualna staza s korisničkim sučeljem postavljena u stvarnu okolinu

Pritiskom gumba *START RACE*, korisnik pokreće proceduru početka utrke. Iznad korisničkog sučelja pojavljuje se semafor s pet svjetala, sličan onome u pozadini slike 3.5. Svjetla se postepeno pale pa potom ugase čime započinje utrka na virtualnoj stazi. Pri paljenju svjetala i početku utrke, reproducira se zvuk paljenja svjetala i kretanja bolida sa startne rešetke. Korisnik tada može šetati oko virtualne staze i promatrati utrku iz različitih kutova. Korisnik može i proći kroz virtualnu stazu kako bi se približio određenom dijelu staze. Za vrijeme trajanja utrke, u pozadini se čuje zvuk vožnje F1 bolida po stazi što kod korisnika može povećati razinu imerzije pa tako i stupanj korisničkog iskustva. Korisnik utrku može i pauzirati, premotati ili usporiti pritiskom na odgovarajuće gumbe korisničkog sučelja. Kako bi pritisnuo gumb na korisničkom sučelju, korisnik se mora približiti virtualnom tabletu i pritisnuti gumb rukom kao da pritišće gumb na stvarnom tabletu.

Alternativno, korisnik može gumb naciljati zrakom koja izlazi iz virtualnih ruku ili kontrolera te napraviti *pinch* gestu ili pritisnuti *Trigger* gumb na jednom od kontrolera.

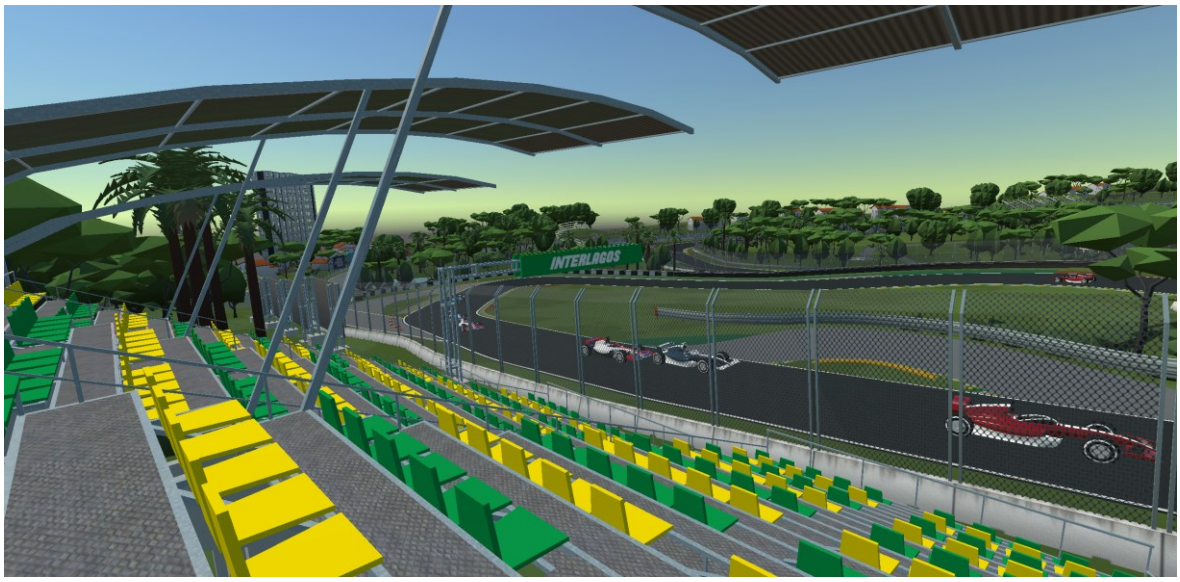
Za mijenjanje stranice na virtualnom tabletu i pristupu drugim funkcionalnostima, korisnik mora postaviti prst na virtualni tablet i pomaknuti ga ulijevo ili udesno tj. napraviti tzv. *swipe* pokret. Isto je moguće i sa zrakom koja izlazi iz virtualnih ruku ili kontrolera. Na drugoj stranici virtualnog tableta nalaze se gumbi za otvaranje virtualnih ekrana na kojima se prikazuje utrka iz perspektive bolida. Korisnik odjednom može otvoriti četiri takva ekrana. Ekrani se, kao i virtualni tablet, stvaraju orijentirani prema korisniku i mogu se premještati po stvarnom prostoru (slika 4.3).

Slika 4.3 Virtualni ekrani koji prikazuju utrku iz perspektive bolida

Na trećoj stranici virtualnog tableta nalaze se gumbi za brzi, ali blagi prijelaz iz miješane stvarnosti u virtualnu stvarnost. U virtualnoj stvarnosti korisnik se nalazi na nekoj od tribina uz stazu i više nije svjestan stvarnog okruženje u kojem se nalazi, već samo virtualne staze i simulirane utrke koja se odvija na njoj. U trenutku kad bolidi prolaze uz tribinu na kojoj se korisnik nalazi, reproducira se zvuk Dopplerovog efekta prolaska stvarnog bolida Formule 1 pored tribine, što potencijalno može povećati imerziju korisnika

u virtualnom okruženju. Na slici 4.4 nalazi se pogled iz perspektive navijača na startno-ciljnoj ravnini, a na slici 4.5 pogled iz perspektive navijača na tribini pored trinaestog zavoja staze Interlagos. Povratak iz virtualne stvarnosti u stvarnost u kojoj je korisnik svjestan virtualne utrke, ali i svoje stvarne okoline, omogućen je pritiskom *Trigger* gumba na lijevom kontroleru ili pritiskom gumba na virtualnom dlanu lijeve ruke. Nakon vraćanja u miješanu stvarnost, korisnik može odabrati novu virtualnu tribinu s koje želi pratiti utrku ili nastaviti pratiti utrku na virtualnoj stazi i na virtualnim ekranima.

Slika 4.4 Pogled iz perspektive navijača na tribini startno ciljne ravnine



Slika 4.5 Pogled iz perspektive navijača na tribini pored trinaestog zavoja staze

5. Analiza aplikacije i moguća poboljšanja

5.1. Problemi pri izvedbi

Najveći izazov u izvedbi aplikacije *Formula XR* je puštanje aplikacije u pogon na Meta Quest 3 uređaj. Kao što je već navedeno, za razvoj aplikacije *Formula XR* između ostalog korištena je i programska platforma Unity te aplikacija Meta Quest Link koja omogućuje spajanje XR uređaja na osobno računalo. Aplikacija Meta Quest Link u kombinaciji s programskog platformom Unity ubrzava razvoj XR aplikacija jer omogućuje njihovo testiranje s platforme direktno na XR uređaju, odnosno programer ne mora trošiti vrijeme na puštanje aplikacije u pogon, već ju samo treba pokrenuti u Unityju. Kada se aplikacija testira na ovakav način, svi izračuni i simulacije odvijaju se na procesoru osobnog računala, dok se na XR uređaju prikazuje samo generirana slika aplikacije. Zbog navedenog, određeni problemi aplikacije mogu se uočiti tek kada se aplikacija pusti u pogon na XR uređaj, odnosno kada se svi često zahtjevni poslovi računanja prebace na procesor XR uređaja koji ima slabije performanse od procesora osobnog računala.

Aplikacija *Formula XR* se tijekom razvoja većinom testirala u Unityju, uporabom Meta Quest Link aplikacije. Aplikacija je prvi put puštena u pogon na Meta Quest 3 uređaj nakon implementacije funkcionalnosti za postavljanje virtualne staze u korisnikovu stvarnu okolinu. Iako je aplikacija do tada funkcionirala gotovo besprijekorno kada se testirala uporabom Meta Quest Link aplikacije, prvi problemi počeli su se nadzirati jednom kada se pustila u pogon na Meta Quest 3 uređaj. Aplikacija je i dalje funkcionirala na uređaju, odnosno prikazivala je simuliranu utrku uporabom virtualnih elemenata. Problem je bio u niskoj rezoluciji tih elemenata i činjenici da je virtualna staza titrala kada bi korisnik pomicao glavu, odnosno HMD uređaj. Uzroci ovih problema nisu utvrđeni i ispravljeni, iako postoji sumnja da do niske rezolucije virtualnih elemenata dolazi kada se aplikacija koristi u miješanoj stvarnosti, odnosno u varijanti u kojoj korisnik uz virtualne elemente vidi i svoju stvarnu okolinu. Kakogod, navedena sumnja nije potvrđena člancima i raznim forumima na Internetu.

Veći problemi pojavili su se kada se finalna verzija aplikacije pustila u pogon na Meta Quest 3 uređaj. Osim prethodno opisanih problema, aplikacija se iznenadno zaustavila

svaki put kada bi korisnik pokušao otvoriti virtualni ekran koji prikazuje utrku iz perspektive bolida. Za uzrok ovog problema smatrala se činjenica da ju u aplikaciji u svakom trenutku aktivna 21 kamera, odnosno jedna kamera na svakom od 20 bolida i jedna kamera koja predstavlja korisnika u aplikaciji. Kako bi se problem pokušao riješiti, aplikacija je prepravljena tako da je u svakom trenutku aktivno što manje kamera. Kamere na bolidima su deaktivirane sve dok korisnik ne zatraži da se kamera uključi i prikaže perspektivu određenog bolida. U ovom slučaju, u bilo kojem trenutku rada aplikacije, moglo je biti aktivno najviše pet kamera, od kojih je jedna za korisnika i četiri na bolidima zato što aplikacija dopušta paralelno praćenje perspektive četiri bolida. Iako je ova funkcionalnost donekle povećala broj sličica koje se iscrtavaju u sekundi (engl. *frames per second,* skr. FPS) i tako poboljšala performanse aplikacije, ona se i dalje zaustavljala svaki put kada bi korisnik pokušao otvoriti virtualni ekran.

Još jedan problem aplikacije je vrlo nizak broj iscrtanih sličica u sekundi kada se aplikacija koristi u VR varijanti, odnosno u varijanti u kojoj korisnik prati utrku iz perspektive navijača na tribini. Uzroci ovog problema utvrđeni su postepenim uklanjanjem raznih komponenti iz Unity scene sve dok broj iscrtanih sličica u sekundi ne poraste na odgovarajuću razinu. Kao uzroci problema utvrđene su Bézierove krivulje i napisana C# skripta *RacingLineFollower* koji u kombinaciji omogućuju gibanje virtualnih bolida po stazi. Paket Bézier Path Creator, korišten za crtanje Bézierovih krivulja, posljednji je put ažuriran 2019. godine te nije u potpunosti prilagođen za novije verzije Unity programske platforme, što potvrđuju i recenzije drugih korisnika na Unity Asset Storeu. Skripta *RacingLineFollower* u svojoj *Update* funkciji (kôd 3.2) obavlja izračune za promjenu pozicije bolida u svakom trenutku rada aplikacije. Ova skripta postavljena je ukupno na 40 bolida, 20 na bolidima koji se nalaze na umanjenom prikazu staze koju korisnik vidi u svojoj stvarnoj okolini i 20 na bolidima staze originalne veličine koju korisnik vidi kada koristi aplikaciju u VR varijanti. Veliki broj instanci ove skripte povlači i veliku količinu računanje u *Update* funkciji koja se poziva jedanput za svaku sličicu (engl. *frame*) koju Unity iscrta, što posljedično negativno utječe na ukupan broj sličica koje Unity iscrta u jednoj sekundi. Za rješenje ovog problema, potrebno je pronaći novi, ažuran paket sličan paketu Bézier Path Creator kojim će se u scenu postaviti Bézierove krivulje koje će pratiti virtualni bolidi. Također, skriptu *RacingLineFollower* potrebno je optimizirati i promijeniti tako da ne obavlja veliku količinu izračuna za svaku Unity sličicu, što bi trebalo povećati ukupan broj sličica iscrtanih u sekundi kada se aplikacije koristi na Meta Quest 3 uređaju.

5.2. Moguća poboljšanja

Svakako najveće i prioritetno najvažnije poboljšanje aplikacije *Formula XR* je rješavanje problema navedenih u prethodnom potpoglavlju tj. uspješno puštanje aplikacije u pogon na Meta Quest 3 uređaj. Nakon rješavanja navedenih problema, aplikacija se može i sadržajno poboljšati. Kao što je već navedeno, Formula 1 je sport vođen podacima te kako bi korisnicima bilo što lakše i zanimljivije pratiti prijenos utrke, sam prijenos može se proširiti raznim relevantnim informacijama o utrci. Tako se npr. u aplikaciju *Formula XR* može dodati virtualni pano koji u tabličnom prikazu prikazuje poredak vozača u utrci i razmak u sekundama između svakog vozača. Na tom tabličnom prikazu mogu se prikazati još poneke informacije o utrci, npr. koja je starost guma koje koriste vozači, koliko su puta otišli na *pit-stop* i sl. Također, u aplikaciju se mogu dodati slični virtualni panoi koji se pojavljuju na određeni događaj. Na primjer, pano koji se pojavljuje kada jedan od vozača postavi najbrži krug utrke ili pano koji obavijesti korisnika kada dođe do sudara bolida i kada sigurnosno vozilo izađe na stazu.

Aplikacija *Formula XR* prikazuje utrku od početka do trenutka kada posljednji bolid završi sve krugove utrke. Stvaran prijenos F1 utrke prikazuje i poneke informacije prije same utrke, kao što su podaci o stazi na kojoj se održava utrka i poredak vozača ostvaren na kvalifikacijama održanim dan prije utrke. U aplikaciji *Formula XR*, poredak vozača u utrci mogao bi se prikazati na virtualnom panou, a za prikaz informacija o stazi Interlagos, mogla bi se iskoristiti XR svojstva aplikacije i virtualni model staze. Na primjer, aplikacija bi na virtualnom modelu staze mogla različitim bojama označiti sektore staze ili zavoje na kojima se odvija najviše pretjecanja te na virtualnom panou ispisati najzanimljivije trenutke na tom zavoju kroz povijest Formule 1. Također, nakon završetka utrke, aplikacija bi mogla virtualnim elementima prikazati ceremoniju proglašenja vozača i momčadi koji su pobijedili u utrci.

Još jedno potencijalno proširenje aplikacije je prikaz utrke u VR varijanti iz perspektive vozača u virtualnom bolidu. Na ovaj bi način aplikacija pružila korisniku osjećaj prisutnosti u bolidu koji se kreće po stazi i do 300 km/h. Kakogod, ova bi funkcionalnost kod ponekih korisnika mogla izazvati mučninu i vrtoglavicu te prije njene implementacije trebalo bi provesti korisničku analizu kako bi se utvrdilo kako korisnici reagiraju na nju.

Zaključak

Jedna od primjena imerzivnih tehnologija, pogotovo podskupa tehnologija koje spadaju pod pojam dograđena stvarnost, je proširivanje sportskih prijenosa uporabom virtualnih elemenata. Raznorazne aplikacije virtualnim elementima proširuju pogled navijača na sport koji prati, pružajući mu uvid u informacije koje potencijalno nisu prikazane u samom televizijskom prijenosu. Ovakve aplikacije mogu pružiti korisniku osjećaj prisutnosti na samom mjestu održavanja sportskog događaja, ali i pomoći organizatorima u oglašavanju i postavljanju reklama u prijenos uporabom virtualnih elemenata.

U sklopu ovog rada izrađena je aplikacija nazvana *Formula XR* koja simulira održanu utrku Formule 1 u dograđenoj stvarnosti uporabom virtualnog modela trkaće staze i trkaćih automobila. Aplikacija se temelji na stvarnim podacima utrke Velike nagrade Brazila iz 2021. godine i može se koristiti paralelno uz stvarni prijenos te utrke. Koristeći aplikaciju korisnici vide virtualnu reprezentaciju staze u svojem stvarnom prostoru te ona im omogućuje pregled kompletne utrke i detaljno praćenje dijelova utrke koji u tom trenutku potencijalno nisu prikazani na stvarnom prijenosu. Aplikacija je obogaćena pozadinskim zvukom F1 utrke, omogućuje korisnicima premotavanje i pauziranje utrke te gledanje utrke iz perspektive određenog F1 bolida. Funkcionalnost koja opravdava pojam XR u imenu aplikacije je prijelaz iz miješane u virtualnu stvarnost. Za razliku od miješane stvarnosti, u virtualnoj stvarnosti korisnik više nije svjestan svojeg stvarnog okruženja, već utrku prati iz perspektive navijača na virtualnoj tribini staze.

Aplikacija je izrađena na programskoj platformi Unity, uporabom C# skripti napisanih u programu Visual Studio i gotovih komponenti preuzetih s Unity Asset Storea. Za dohvat stvarnih podataka o utrci Velike nagrade Brazila korišten je programski jezik Python i paket FastF1, a paket Meta Interaction SDK je unaprijed pripremljenim XR komponentama ubrzao razvoj aplikacije. Iako je namijenjena Meta Quest 3 uređaju, aplikacija *Formula XR* trenutno se zbog optimizacijskih problema ponekih C# skripti i zastarjele komponente preuzete s Unity Asset Storea može pokrenuti samo kroz Unity uređivač, pri čemu se slika putem aplikacija Meta Quest Link prikazuje na Meta Quest 3 uređaju.

Literatura

[1] Cambridge Dictionary. Poveznica: https://dictionary.cambridge.org/dictionary/english/immersive; pristupljeno 5. svibnja 2024.

[2] A. Pavithra, J. Kowsalya, S. Keerthi Priya, T. Kiruba Nandhini (2020) An Emerging Immersive Technology-A Survey. International Journal of Innovative Research and Growth. International Journal of Innovative Research in Technology (IJIRT).

[3] Y. N. Lee, L. Shan, C. Chen (2013) System Development of Immersive Technology Theatre in Museum. Virtual, Augmented and Mixed Reality. Systems and Applications, str. 400 - 408.

[4] I. S. Pandžić, T. Pejša, K. Matković, H. Benko, A. Čereković, M. Matijašević (2013) Virtualna okruženja: Interaktivna 3D grafika i njene primjene. Zagreb: Element.

[5] C. Anthes, R. J. García-Hernández, M. Wiedemann, D. Kranzlmüller (2016) State of the Art of Virtual Reality Technology. Leibniz Supercomputing Centre, Bavarian Academy of Sciences.

[6] R. T. Azuma (1997) A survey of augmented reality. Presence: Teleoperators & Virtual Environments, 6, 4 str. 355 - 385.

[7] W. Hoenig, C. Milanes, L. Scaria, T. Phan, M. Bolas, N. Ayanian (2015) Mixed Reality for Robotics. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[8] S. Lebow (2022) AR/VR headsets can't compete with smartphones. eMarketer. Poveznica: https://www.emarketer.com/content/ar-vr-headsets; pristupljeno 12. svibnja 2024.

[9] J. Leatham (2019) Snap Partner Summit Launches New AR Updates for Snapchat. ARPost. Poveznica: https://arpost.co/2019/04/10/snap-partner-summit-new-ar-features-snapchat/; pristupljeno 12. svibnja 2024.

[10] Y. Wu, W. Che, B. Huang (2021) An Improved 3D Registration Method of Mobile Augmented Reality for Urban Built Environment. International Journal of Computer Games Technology.

[11] S. Alizadehsalehi, A. Hadavi, J. C. Huang (2020) From BIM to extended reality in AEC industry. Automation in Construction.

[12] Hermes (2023) 3 Examples of how XR is Transforming the Sports Industry. Medium. Poveznica: https://iamhermes.medium.com/3-examples-of-how-xr-is-transforming-the-sports-industry-c72f3290ece7; pristupljeno 14. svibnja 2024.

[13] S. Mitchell (2018) F1 is going to experiment more with new TV graphics. Motor1.com. Poveznica: https://uk.motor1.com/news/241607/f1-experiment-new-tv-graphics/; pristupljeno 14. svibnja 2024.

[14] Xtadium. Poveznica: https://xtadiumvr.com/; pristupljeno 14. svibnja 2024.

[15] MAAKE (2024.) A Glimpse into the Future of XR Technology in Formula One and Beyond. Poveznica: https://www.maake.com.au/a-glimpse-into-the-future-of-xr-technology-in-formula-one-and-beyond/; pristupljeno 14. svibnja 2024.

[16] J. Shapiro (2023) Data Driven at 200 MPH: How Analytics Transforms Formula One Racing. Forbes. Poveznica: https://www.forbes.com/sites/joelshapiro/2023/01/26/data-driven-at-200-mph-how-analytics-transforms-formula-one-racing/; pristupljeno 17. svibnja 2024.

[17] Meta Quest 3. Poveznica: https://www.meta.com/quest/quest-3/; pristupljeno 19. svibnja 2024.

[18] FastF1. Poveznica: https://docs.fastf1.dev/; pristupljeno 1. ožujka 2024.

[19] Unity. Poveznica: https://unity.com/; pristupljeno 19. svibnja 2022.

[20] Microsoft Visual Studio. Poveznica: https://visualstudio.microsoft.com/; pristupljeno 19. svibnja 2022.

[21] P. A. Rauschnabel, R. Felix, C. Hinsch, H. Shahab, F. Alt (2022) What is XR? Towards a Framework for Augmented and Virtual Reality. Computers in Human Behavior.

[22] Formula 1. Poveznica: https://www.formula1.com/; pristupljeno 1. ožujka 2024.

[23] Valem Tutorials (2021) Smooth Scene Fade Transition in VR. YouTube. Poveznica: https://www.youtube.com/watch?v=JCyJ26cIM0Y; pristupljeno 10. svibnja 2024.

[24] Sunny Valley Studio (2023) Unity VR Stop Head Clipping with Fade-to-Black Effect. YouTube. Poveznica: https://www.youtube.com/watch?v=YRjfmblMj8Q; pristupljeno 10. svibnja 2024.

[25] J. Mayne (2024) How much does an F1 car cost in 2024? Key parts, history of most expensive Formula 1 cars. The Sporting News. Poveznica: https://www.sportingnews.com/us/formula-1/news/f1-car-cost-key-parts-most-expensive/sbvxis3e0fveuzxevx56b470; pristupljeno 15. lipnja 2024.

[26] Formula 1 (2022) Formula 1 announces TV, race attendance and digital audience figures for 2021. Poveznica: https://www.formula1.com/en/latest/article/formula-1-announces-tv-race-attendance-and-digital-audience-figures-for-2021.1YDpVJIOHGNuok907sWcKW; pristupljeno 15. lipnja 2024.

[27] Z. Bedlekova (2017) What is a Head-up Display a.k.a. HUD. Poveznica: https://www.sygic.com/blog/2017/what-is-a-head-up-display-a-k-a-hud; pristupljeno 20. lipnja 2024.

Sažetak

U ovom radu predstavljene su imerzivne tehnologije. Opisane su osnovne karakteristike virtualne, proširene, miješane i dograđene stvarnosti te njihova primjena u prijenosima sportskih događaja. Uporabom navedenih tehnologija, u sklopu ovog rada izrađena je aplikacija *Formula XR* koja na temelju stvarnih podataka o utrci Formule 1, simulira utrku na virtualnoj stazi koju korisnik vidi u svojoj stvarnoj okolini. Korištenjem aplikacije *Formula XR* gledatelji Formule 1 dobivaju informacije o utrci koje često nisu dostupne u televizijskim prijenosima. Osim što omogućuje korisniku da prati utrku na virtualnoj stazi u svojoj stvarnoj okolini, aplikacija *Formula XR* omogućuje korisnicima prijelaz u virtualnu stvarnost u kojoj prate utrku na tribini virtualne staze iz perspektive navijača. Rad prati proces izrade aplikacije *Formula XR* te u njemu su navedene i opisane tehnologije korištene za njenu izradu. Na kraju rada opisane su upute za korištenje aplikacije te problemi pri izvedbi i potencijalna proširenja.

Ključne riječi: dograđena stvarnost, virtualna stvarnost, Formula 1, Meta Quest 3, Unity

Summary

This thesis focuses on immersive technologies, and describes the basic characteristics of Virtual, Augmented, Mixed, and eXtended reality, as well as their application in broadcasts of sporting events. Integrating these technologies, an application called *Formula XR* has been developed in the scope of this thesis. The application simulates a race based on real Formula 1 race data and displays it on a virtual track which users can see in their real environment. By using this application, Formula 1 viewers receive information about the race that is often not available in television broadcasts. In addition to allowing users to follow the race on a virtual track in their real environment, the *Formula XR* application enables users to transition into virtual reality, where they can watch the race from the perspective of a fan in the stands of the virtual track. The thesis presents the development process of the *Formula XR* application and describes the technologies used for its creation. Instructions for using the application are described at the end, along with issues encountered during implementation and potential future improvements.

Keywords: eXtended Reality, Virtual Reality, Formula 1, Meta Quest 3, Unity

Skraćenice

F1	*Formula 1*	Formula 1, razred automobilskih utrka
XR	*Extended Reality*	dograđena stvarnost
VR	*Virtual Reality*	virtualna stvarnost
AR	*Augmented Reality*	proširena stvarnost
MR	*Mixed Reality*	miješana stvarnost
DOF	*Degrees Of Freedom*	stupnjevi slobode
HMD	*Head-Mounted Display*	zaslon montiran na glavu
SLAM	*Simultaneous Localization And Mapping*	istodobna lokalizacija i mapiranje
IDE	*Integrated Development Environment*	integrirana razvojna okolina
CSV	*Comma-Separated Values*	zarezom odvojene vrijednosti
COP	*Center Of Projection*	središte projekcije
VPW	*View-Plane Window*	projekcijski prozor
VPN	*View-Plane Normal*	normala na projekcijsku plohu
NCP	*Near Clipping Plane*	bliska odrezujuća ploha
FCP	*Far Clipping Plane*	daleka odrezujuća ploha
FPS	*Frames Per Second*	broj iscrtanih sličica u sekundi