

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

ManeuVR

Tehnička dokumentacija

Verzija 1.0

Studentski tim: Ivan Karlović
Danijel Bajlo
Tin Blažević
Filip Husnjak
Luka Mesarić
Marko Mijolović
Karlo Miličević
Maja Radočaj

Nastavnik: prof. dr. sc. Željka Mihajlović

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

Sadržaj

1.	Opis razvijenog proizvoda	3
2.	Tehničke značajke	4
2.1	Tragovi	4
2.2	Sudari i pravila igre	5
2.3	Umjetna inteligencija	5
2.4	Čestični efekti	9
2.5	Zvuk	10
2.5.1	Pronalaženje i izrada zvukova	10
2.5.2	Zvuk motora	10
2.5.3	Zvuk sudara	10
2.5.4	Pozadinski zvukovi	11
2.5.5	Implementacija zvukova u Unityju	11
2.6	Modeliranje	12
2.6.1	Modeliranje u Blenderu	12
2.6.2	Uvoz modela u Unity	15
2.6.3	Smanjenje broja poligona preuzetih modela	15
2.7	Fizikalni model motora	17
2.8	Gibanje motora	18
2.9	Kamere	19
2.10	HTC Vive	20
2.11	Integracija VR tehnologije	20
2.12	Kontrole	21
2.13	Prikaz vodenih površina	22
2.14	Iscrtavanje malih površina voda stajaćica	22
2.15	Iscrtavanje oceana	23
2.16	Naknadna obrada (engl. <i>post processing</i>)	24
2.17	Grafičko korisničko sučelje	25
2.18	Dizajniranje mape	26
2.19	Verzioniranje projekta	27
3.	Upute za korištenje	28
4.	Literatura	29

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

Tehnička dokumentacija

1. Opis razvijenog proizvoda

ManueVR računalna je igra namijenjena korištenju u VR okruženju. Igrač poprima ulogu jednog od motorista na tropskom otoku. Krećući se po otoku, motoristi iza sebe ostavljaju trag. Dotakne li bilo koji od motorista taj trag, igra je za njega završena. Cilj je ostati zadnji motorist u igri.

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

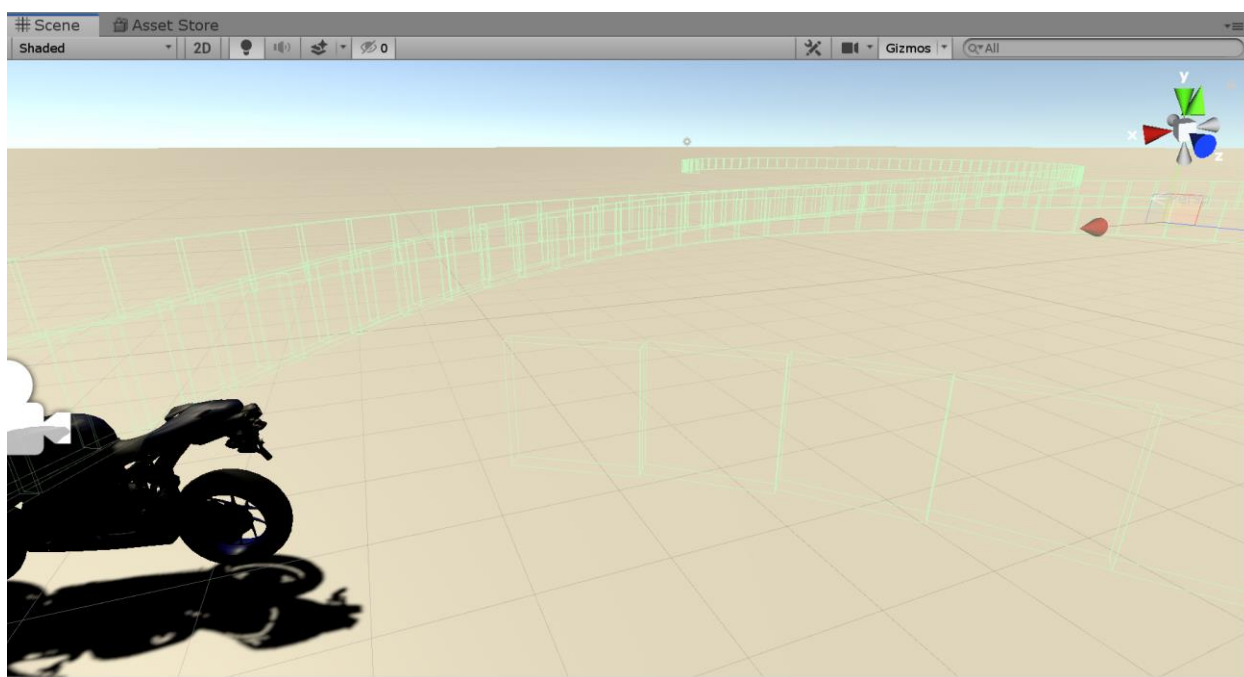
2. Tehničke značajke

2.1 Tragovi

Ključan dio igre su tragovi koje igrači ostavljaju za sobom. Sudar s tragom znači kraj igre za igrača.

Generatori tragova su implementirani kao stroj stanja zbog raznih uvjeta koji utječu na to jesu li aktivni ili ne. Na primjer, tragovi se periodički isključuju kako bi nastali uski prolazi kroz koje igrač može proći. Isto tako tragovi se ne generiraju ako se igrač kreće unazad, ali se odmah aktiviraju kada se počne kretati prema naprijed. Sami tragovi podijeljeni su na 2 dijela, sudarače i grafički prikaz, koji se generiraju odvojeno.

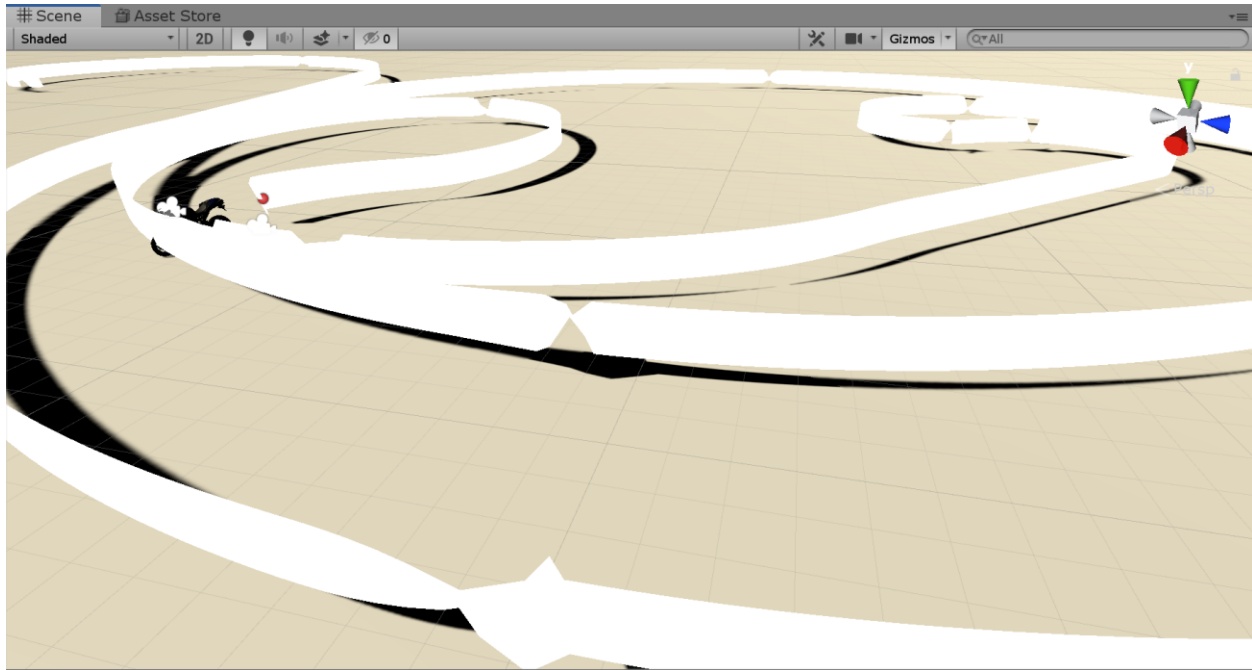
Za generiranje sudarača (slika 2-1) prati se određena točka na motoru i pozicija zadnjeg postavljenog sudarača. Kada njihov razmak prijeđe određeni prag, generira se novi sudarač. Njegova pozicija se određuje na temelju pozicije prethodnog sudarača i vektora prema motoru. Novopostavljeni sudarač orijentiran je prema motoru.



Slika 2-1: Generiranje sudarača traga

Za generiranje grafičkog dijela traga prvotno je korištena komponenta *Trail Renderer* ugrađena u Unity, međutim ta komponenta je oblikovana da bude univerzalna i nakon ograničavanja stupnjeva slobode samo na rotaciju oko y-osi, trag se počeo uvijati (slika 2-2). Napravljeno je vlastito rješenje, slično onome za sudarače, ali s puno manjim korakom kako bi stvaranje traga izgledalo glatko. Iz točke koja se promatra dobivaju se dva vrha translacijom te točke u smjeru pozitivne i negativne globalne y-osi koji se zatim spajaju u trokute s prethodnim vrhovima. Na žičanu mrežu koja se dobije, dodaje se željeni materijal koji određuje konačni izgled traga. Rezultat je da naš vlastiti trag izgleda identično kao i ugrađeni, ali se ne uvrće.[5]

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020



Slika 2-2: Uvrtanje ugrađenih tragova

2.2 Sudari i pravila igre

Na stanje igrača, kao i okidanje zvukova te dodatnih vizualnih efekata utječu sudari s drugim objektima.

Ograničenje platforme Unity je što samo jedna skripta može uhvatiti i obraditi događaj sudara koji se generira kada se on dogodi. Rješenje tog problema je oblikovni obrazac promatrač (engl. *observer*) koji je već ugrađen u sam jezik C#. Posebne skripte hvataju i obrađuju sudare za svaki tip sudarača i definiraju događaje na koje se ostale skripte mogu pretplatiti nakon čega više skripti može reagirati na isti sudar, svaka na svoj način.

Za sve sudarače postoji tri definirana događaja koji se okidaju ovisno o brzini kretanja tijela, što je korisno za okidanje različitih zvukova udaraca. Za glavni sudarač motora čiji sudari utječu na stanje igrača u igri, dodatno se definira događaj koji se okida kada dođe do sudara sa bilo kojim objektom koji je označen (engl. *tag*) kao trag. Za sudarače kotača dodatno se pamti jesu li prizemljeni ili ne, jer ne bi imalo smisla da kotači ostavljaju trag prašine ako su u zraku.

Dodatno se za svakog igrača dodatno promatra koliki mu je pomak u određenom vremenu. Posebni objekti promatraju poziciju igrača u diskretnim koracima i računaju pomak od početne pozicije. Kada njihovo vrijeme promatranja istekne, provjerava se je li ukupni pomak veći od praga i igrač biva izbačenim iz igre ako taj uvjet nije zadovoljen.

2.3 Umjetna inteligencija

Umjetna inteligencija bitan je dio igara za jednog igrača. Ona bi trebala predstavljati izazov koji će biti zanimljiv igraču, ali opet ne smije biti previše zahtjevno.

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

Zbog želje za eksperimentiranjem kao podloga za ostvaranje umjetnih igrača odabran je relativno nov alat Unity ml-agents[4]. Radi se o paketima za Unity i Python koji služe za strojno učenje. Alat je zamišljen tako da ga netko bez velikog znanja u tom području može koristiti, ali je otvorenog koda za istraživače i druge vrste naprednih korisnika. Implementirani su algoritmi i dodatne tehnike za *Reinforcement learning* uz mogućnost imitacijskog učenja. Ideja je da se Unity okoliš koristi za prikupljanje opažanja na temelju kojih agent koji živi unutar okoliša donosi odluke kojima pribavlja određenu nagradu, ovisno o kvaliteti odluke koju je donio. Za imitacijsko učenje, čovjek igra igru u ulozi agenta, dok posebna skripta pamti opažanja i odgovarajuće odluke čovjeka, što će se koristiti na početku treninga kako bi agent brže naučio željeno ponašanje.

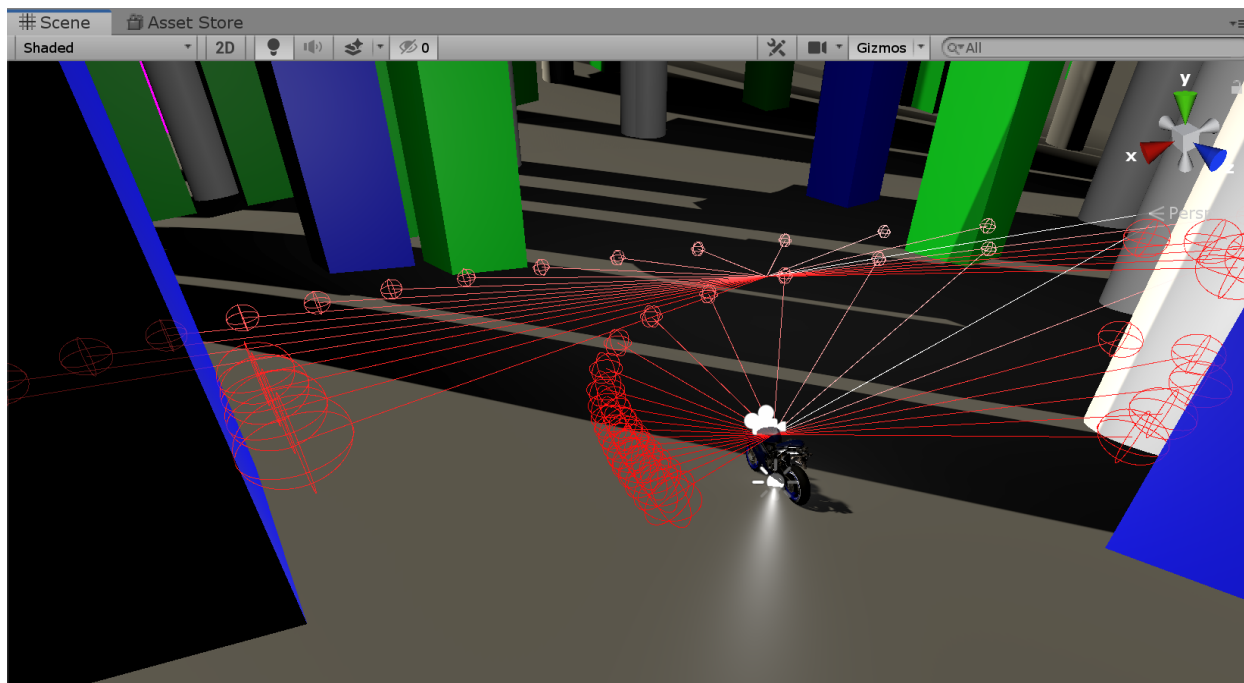
Tijek ostvarivanja umjetne inteligencije na ovaj način je sljedeći:

1. Stvaranje okoliša za učenje
2. Implementacija razreda Agent i Academy
3. Definiranje nagrade
4. Skupljanje demonstracija
5. Trening (učenje)
6. Promatranje naučenog
7. Integracija u konačnu igru

Kao okoliš za učenje proceduralno se generira podloga korištenjem više slojeva OpenSimplex šuma, na koju se nasumično raspoređuje određen broj objekata koji imaju određene Unity oznake (engl. *Unity tags*).

Agent kao osjetila ima 2 senzora (slika 2-3) koji ispaljuju po 24 zrake te informacije o tipu i udaljenosti pogođenih objekata normaliziraju i spremaju u vektor opažanja. Drugi senzor je povišen u odnosu na prvi kako bi mogao pogodi objekte koji su više udaljeni. Moguće je dodati i bilo kakva druga opažanja koja opisuju stanje okoliša ili agenta ali ne treba pretjerivati jer redundantni ulazi usporavaju učenje. Ovdje se koristi lokalna brzina i lokalna rotacija oko y- i x-osi. Agent ima mogućnost donošenja 3 paralelne odluke odjednom, a to su promjena akceleracije, skretanje i kočenje.

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

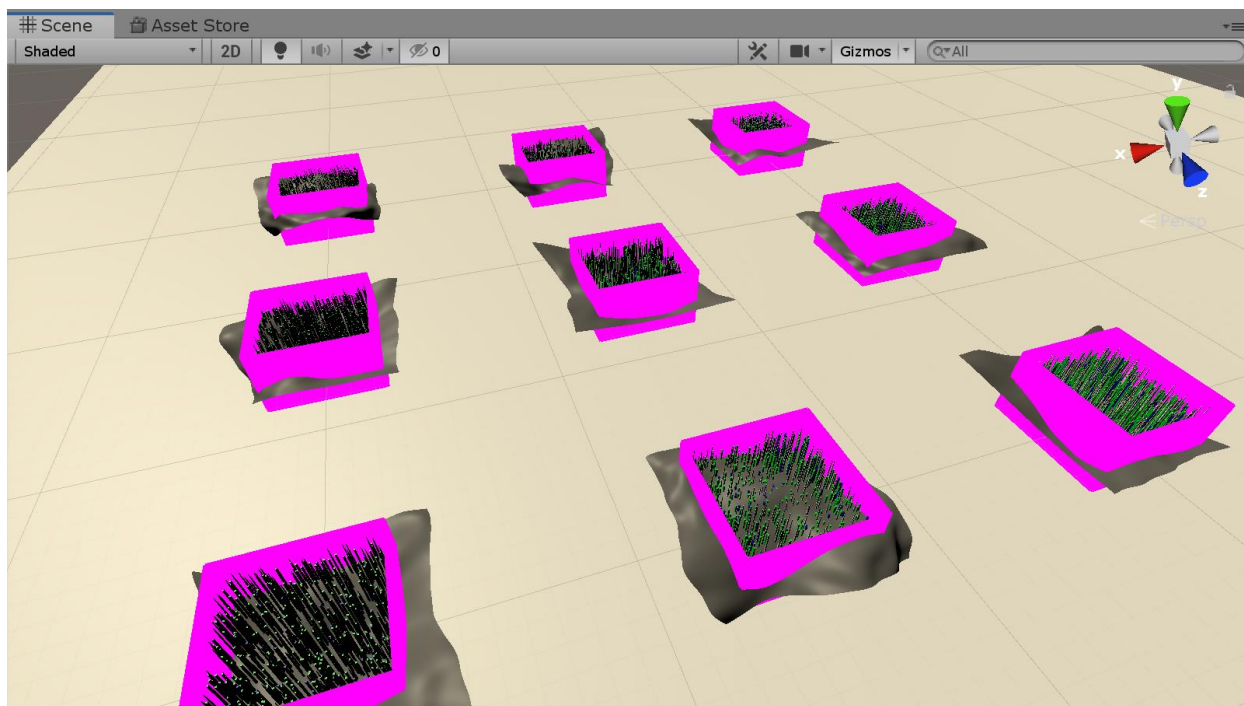


Slika 2-3: Senzori agenta

Nagrada je ono što oblikuje ponašanje agenta. Ovisno o željenom ponašanju, potrebno je definirati adekvatnu nagradu. Osnovna ideja je da agent nauči izbjegavati prepreke što je duže moguće. Prema tome, osnovni pokušaj nagradnog signala linearno raste u vremenu, a sudar s preprekom donosi negativnu nagradu. Ovo je rezultiralo time da je agent naučio stajati na mjestu. Ispostavilo se da je nagradu potrebno pomnožiti omjerom trenutnog i maksimalnog broja okretaja kao bi agent bio prisiljen kretati se. Negativne nagrade za određenu akciju kao što je kočenje rezultiralo je time da je agent odbijao izvršavati takve akcije. Ipak je dovoljno što se broj okretaja pri kočenju smanjuje. Čak se ispostavilo korisnim u slučaju kočenja nagradu pomnožiti koeficijentom malo većim od 1. Također, dodana je nagrada za skretanje kako bi agent bio potaknut na istraživanje. Dodatna mogućnost nagrade je kada agent prođe pored prepreke, dobije nagradu nakon što se dovoljno udalji, šti za cilj ima bolje iskorištavanje prostora.

Za treniranje je potrebno definirati konfiguraciju treninga, odnosno parametre algoritma za učenje, kao i parametre modela neuronske mreže koji će se generirati. Nakon toga, uz skupljene demonstracije, trening može početi. Implementiran je najjednostavniji oblik treninga u kojem je svaki agent u zasebnom okolišu i neovisan je o ostalima. Postavljeno ih je 9 u scenu za trening kako bi učenje bilo brže (slika 2-4), a vrijeme simulacije je ubrzano 100 puta. Generirani model u našem slučaju ima 2 skrivena sloja po 128 neurona, a kao algoritam za treniranje je odabran Soft Actor Critic (SAC). Ispostavilo se da je pola sata do sat vremena demonstracija dovoljno za uspješno učenje. Isto tako, ispostavilo se da je za nekakvo osnovno izbjegavanje prepreka dovoljno 200 000 epizoda, čije izvršavanje traje 7 do 8 sati.

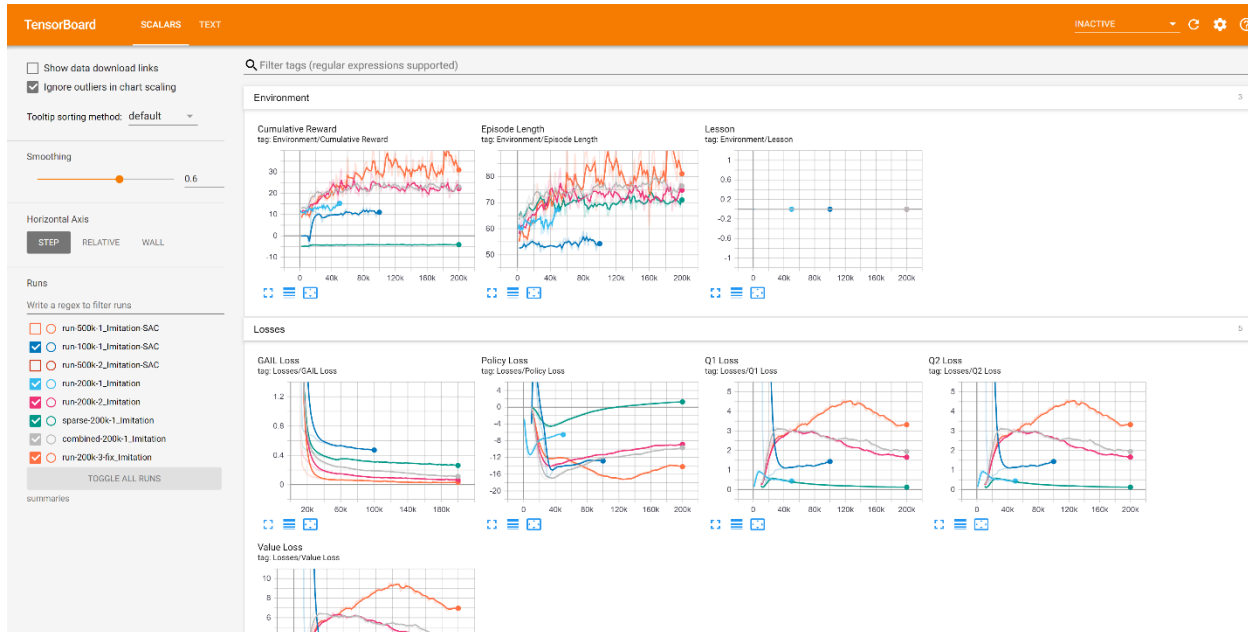
ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020



Slika 2-4: 9 odvojenih okoliša za trening

Budući da je Python paket ml-agents implementiran povrh alata Tensorflow, statistike treninga moguće je pratiti u realnom vremenu koristeći alat TensorBoard (slika 2-5). Pokazatelji uspješnog treninga su smanjivanje entropije odluka ako je prostor odluka diskretan, rast kumulativne nagrade, rast trajanja epizode i smanjivanje gubitaka. Ukoliko naučeno ponašanje nije zadovoljavajuće, potrebno je ponovno krenuti od definiranja nagrada ili mijenjati parametre treninga i modela.

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020



Slika 2-5: Statistike uspješnih i manje uspješnih treninga

Agenti su uspješno naučili izbjegavati prepreke, unatoč malenom broju epizoda treninga. Optimalan bi bio trening od 1 do 3 milijuna epizoda, ali to bi bilo vremenski previše zahtjevno s obzirom da je za trening korišten obični laptop. U posebnim slučajevima, kao što je na primjer bivanje u neposrednoj blizini granice mape, agentu vidljivo nedostaje iskustva jer ne donosi smislene odluke.

2.4 Čestični efekti

Bez čestičnih efekata, svijet unutar igre često izgleda suhoparno i umjetno. Dodavanjem traga prašine, dima motora i sličnih stvari u igru se unosi dodatni realizam.

U ovom je projektu dodano nekoliko čestičnih efekata:

- Trag prašine iza motora
- Ispušni plinovi motora
- Čestice koje nastaju kada se igrač izgubi igru
- Lišće u zraku
- Efekt ubrzanja

Trag prašine motora postavljen je na točke kontakta kotača s podlogom, a pojavljuje se samo ako se igrač kreće brže od određene brzine. Ispušni plinovi motora su slično konfigurirani, tj. gustoća dima ovisi o brzini igrača.

Lišće je napravljeno tako da se lagano pomiče i rotira u svim smjerovima oko igrača, svaki pojedini list polako nastane ni iz čega da bi samo nakon par sekundi isčeznuo u ništavilo.

Efekt ubrzanja temelji se na čestoj simulaciji *warp* pogona u filmovima – mnogo izduženih linija koje iz središta ekrana idu prema rubovima – samo je taj efekt ovdje relativno diskretan kako bi igrač i dalje

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

vidio svijet pred sobom. Javlja se samo kada se igrač približi maksimalnoj brzini.

Čestice koje nastaju kada igrač izgubi igru eksplodiraju iz igrača te se istovremeno smanjuje motor. Glavna uloga ovih čestica, osim da signaliraju isključenje igrača iz igre, jest da maskiraju uklanjanje modela igrača iz svijeta.

Osim čestičnih efekata, u igri se nalazi i lens flare na glavnoj kameri. Lens flare gotov je resurs s Unity Asset Storea.

2.5 Zvuk

Zvuk je jedan od najvažnijih elemenata videoigre koji omogućava udubljivanje u svijet prikazan na ekranu. On daje težinu događajima te prenosi puno povratnih informacija o igračevim akcijama. Značenje mnogih zvukova u videoigramu igrači podsvjesno razumiju čak i ako nikada nisu igrali tu igru. Za naš projekt nismo trebali izmišljati neke nove zvukove, jer je većina interakcija simulacija događaja iz stvarnog života, čiji zvukovi su većini već poznati. Veća sloboda izbora bila je prisutna kod nekih apstraktnijih zvukova poput pozadinske glazbe ili zvuka koji predstavlja da je korisnik izgubio.

2.5.1 Pronalaženje i izrada zvukova

Danas na internetu postoje brojne biblioteke raznovrsnih besplatnih zvukova posebno namijenjene videoigramu. Ipak, mi smo odlučili napraviti većinu zvukova sami kako bismo projektu dali jedinstvenost i stvorili svoj vlastiti zvuk. Za snimanje i uređivanje zvuka korišteni su alati Audacity te FL Studio 12 demo.

2.5.2 Zvuk motora

S obzirom da se igra temelji na vožnji motocikla, prvi i najvažniji zvuk bio je zvuk motora. Kao izvor korištena je besplatna snimka paljenje i vožnje stvarnog motocikla¹. Iz te snimke uzet je segment zvuka motora duljine 0.375 sekunda, čija visina zvuka bila je približno konstantna. Segment je u sebi sadržavao pozadinski šum vjetra koji je bio vrlo primjetan pri brzom ponavljanju zvuka. Taj šum uklonjen je koristeći nisko-propusni filter u alatu FL Studio. Iz istog izvora uzet je i zvuk paljenja motora koji se čuje na početku igre.

Kao zvuk kočenja motocikla korišteno je kočenje automobila², jer motocikli ne proizvode neki distinktan zvuk pri kočenju, no igrač ipak očekuje povratnu informaciju, čak i ako je tehnički nerealna. Jedan od problema kod izrade zvuka bio je to što brzina motocikla nije smjela biti velika te promjene brzine nisu smjele biti nagle, kako ne bi naškodile igraču zbog korištene VR tehnologije. Zato kočenje zvuči naglo, ali u stvarnosti motor samo polagano usporava.

2.5.3 Zvuk sudara

Drugi najvažniji zvuk u igri gdje se motocikli sudaraju s okolinom jest zvuk sudara. Implementirano je nekoliko različitih zvukova sudara koji se mogu čuti ovisno o okolnostima kolizije. U slučaju sudara pri niskoj brzini čuje se tihi sudar koji nije samo slabijeg volumena, već drukčije teksture zvuka u usporedbi s glasnim sudarom. Kao izvor zvuka korišteno je nekoliko slojeva snimke udarca od

¹ <https://www.youtube.com/watch?v=qV6R0HWHssM>

² https://www.youtube.com/watch?v=Q3X_Q4Bkiqc

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

drveni stol, koji je dosta sličan zvuku tvrde plastike kojom bi bio prekriven motor. U tom zvuku većinom su zastupljene niske frekvencije te zvuči dosta organsko i čisto, jer predstavlja blagi udarac ili lupkanje motocikla o prepreku. Svi izrađeni zvukovi sudara sastoje se od više slojeva snimaka koji nisu savršeno poravnati u vremenu, čime se stvara dojam da se sudarilo nešto što sastavljeno od mnogo dijelova i različitih materijala koji ne reagiraju istovremeno na isti podražaj. Zvuk glasnog sudara namjerno je napravljen da bude glasan, neugodan i pretjeran, čime se na najbrži način dojavljuje igraču da se dogodilo nešto loše i nepoželjno. Glasni sudar sastoji se od mnogih slojeva koji se mogu podijeliti u dvije grupe: niskofrekventni zvukovi koji predočavaju snagu udarca te visokofrekventni zvukovi koji dočaravaju zvuk lomljenja i raspadanja. Kako bi se sudar učinio još snažnijim, slojevi niskofrekventnih zvukova su duplicirani i sniženi za pola tona čime je stvorena disonantnost. Kao šlag na kraju, korištenjem alata FL Studio na zvuk je stavljena distorzija koja ga je učinila još glasnijim i opasnijim.

Zvuk pada motocikla zamišljen je dvodijelno. Prvi dio predstavlja udarac motocikla o tlo, sličan zvuku sudara, dok je drugi dio rezultat otpuštanja amortizera koji uzrokuju viši, metalni zvuk. Ispostavilo se da se najrealističniji zvuk pada dobije snimanjem stvarnog pada nekog objekta. S obzirom da je mapa prekrivena pijeskom i travom, zvuku pada motocikla dodani su zvukovi poput gužvanja vrećice, kako bi dočarali podlogu na koju pada.

2.5.4 Pozadinski zvukovi

Za ambijentalne zvukove i glazbu korištene su već postojeće besplatne snimke, jer se izradom originalnih ne bi stvorila bitna razlika. Odabrana je napeta glazba koja odgovara kompetitivnoj naravi igre, no njoj je suprotstavljen smirujući ambijentalni cvrkut ptica. To proturječje zrcali i naglašava apsurd same ideje natjecateljske vožnje motora po uskršnjem otoku. S obzirom da se radnja odvija na otoku, dodan je i pozadinski šum valova koji se može čuti kada je igrač blizu obale.

Izvori:

Pjesma: <https://www.youtube.com/watch?v=NeNhgV1vftw&feature=youtu.be>

More: <https://www.youtube.com/watch?v=aUeCnyQi3bQ>

Ptice: https://www.youtube.com/watch?v=9LfEyxL_f-U

2.5.5 Implementacija zvukova u Unityju

Zvuk u alatu Unity radi na principu izvora (engl. *Audio Source*) i slušatelja (engl. *Audio Listener*). Jedini slušatelj je model samog igrača, koji je ujedno i izvor većine zvukova. Reprodukcijski zvuk odskoči kao rezultat neke pobude lako je ostvarena pretplatom na događanja, koja onda uzrokuju poziv odgovarajućih funkcija.

Zvuk motora postavljen je da se reproducira u petlji, a u svakom okviru (engl. *Frame*) poziva se *update* metoda u kojoj se, između ostalog, računa visina tona zvuka. Ton se povisuje ovisno o brzini motocikla, a prelaskom određenih pragova ponovo snižava kako bi se imitirala promjena brzine. Isto je moralo biti onemogućeno pri negativnoj akceleraciji kako ne bi zvučalo kao da motor ubrzava pri usporavanju. Motor je zamišljen kao četverotaktni pa se visina zvuka također modulira ovisno o taktu. Na zvuk motora utjecaj još imaju jedva primjetna randomizacija te stanje motocikla relativno podlozi. Ako leti u zraku tada se zvuk povisi kako bi se stvorila iluzija bržeg okretanja kotača uslijed manjeg trenja.

Svim zvukovima sudara glasnoća ovisi o brzini pri kojoj se odvija sudar, a visina zvuka ima male nasumične varijacije kako ne bi uvijek zvučali isto. Na temelju brzine također se odlučuje hoće li se reproducirati zvuk tihog ili glasnog sudara.

Pozadinska glazba svira u petlji otpočetak te je uređivanjem osigurano da prijelaz s kraja nazad na

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

početak pjesme ne zvuči neprirodno. Isto vrijedi i za cvrkut ptica.

Neki zvukovi morali su se odviti u kronološkom slijedu, kao paljenje motora pa zvuk motora, ili se reproducirati neprekinuto prije omogućavanje ponovnog ponavljanja, npr. kočenje. To je ostvareno pomoću ko-rutina (engl. *Coroutine*). Na početku igre pokrene se ko-rutina u kojoj se započne reprodukcija zvuka paljenja motora te čeka na završetak reprodukcije. Tek nakon toga pokreće se zvuk motora koji se nadalje reproducira u petlji. Kočenje se obrađuje u funkciji *update*, stoga nije prikladno tamo započinjati reprodukciju zvuka, jer se ona poziva u svakom okviru. Umjesto toga, poziva se ko-rutina u kojoj se onemogućiti ponovna reprodukcija sve do završetka trenutne.

Reprodukcija zvuka valova kad je igrač blizu obale ostvarena je koristeći okidače (engl. *Trigger*). To je jedini zvuk kojem kao izvor nije postavljen motocikl.

Na sredini otoka postavljena je nevidljiva mreža u obliku siluete otoka kojoj je upravljač sudara (engl. *Collision Handler*) postavljen da se ponaša kao okidač. Stoga se motocikl neće sudariti sa sferom, već prolaskom kroz njen plašt okinuti okidač. Izlaskom iz mreže pokreće se reprodukcija zvuka valova, a ponovnim ulaskom se prekida.

S obzirom da svi motocikli u obitelji pripadaju istoj hijerarhiji, dovoljno je bilo samo dodati zvukove za jedan motor i bili su dostupni svima.

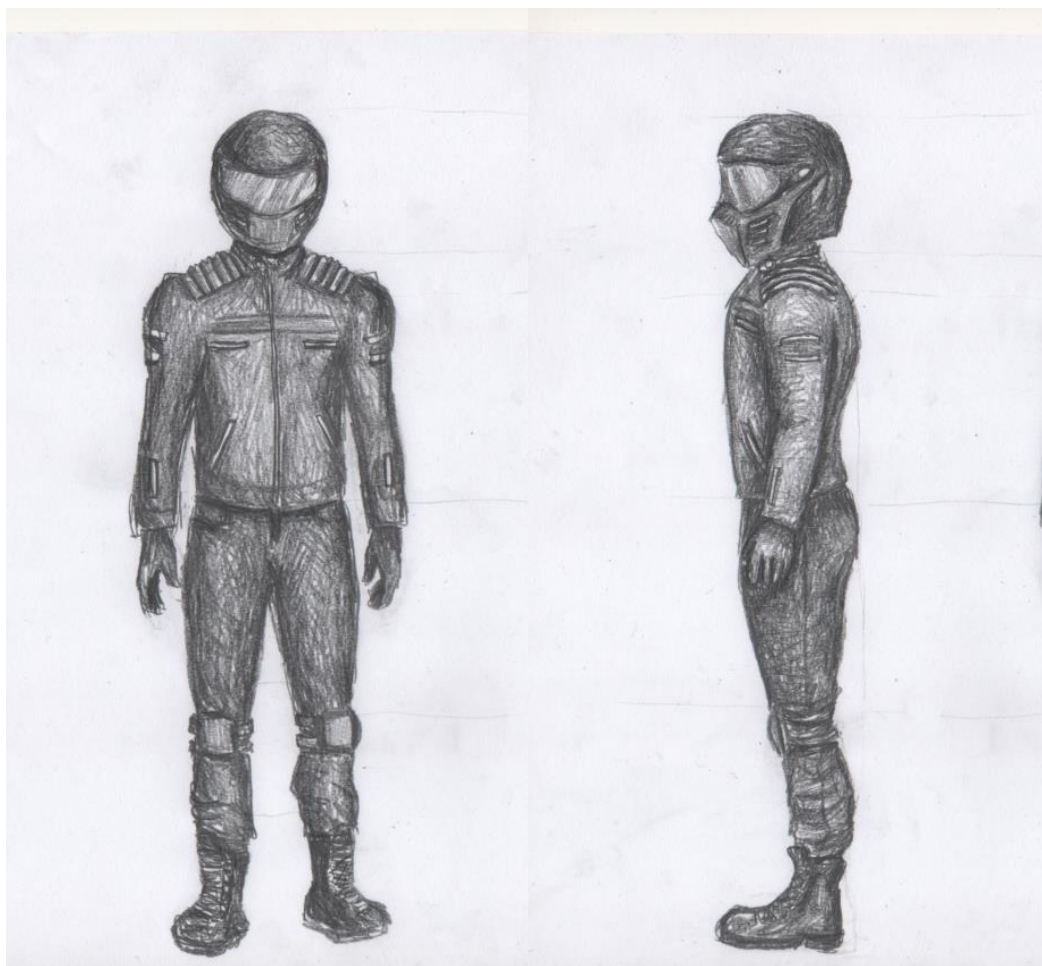
2.6 Modeliranje

U 3D računalnoj grafici, modeliranje je proces razvoja matematičke reprezentacije površina objekata u tri dimenzije pomoću specijaliziranih programskih alata. Rezultat tog procesa je 3D model. Modeli su osnovni gradivni elementi scene video igrice i prva stvar koju igrač može vidjeti. Upravo zato je od ključne važnosti birati modele prikladne za potrebu igrice. Pritom treba paziti da oni budu vizualno privlačni igračima, ali i minimalno komplicirani kako ne bi utjecali na performanse igre. Iako se mnogo gotovih modela već može pronaći na raznim online trgovinama resursa (engl. *Asset stores*), za potrebe ovog projekta modelirani su vozači motora u programskom alatu Blender.

2.6.1 Modeliranje u Blenderu

Prije početka rada, potrebno je napraviti skicu lika (slika 2-6) kako bi se olakšao proces modeliranja.

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020



Slika 2-6: Skica igrivog lika

Na temelju gotove skice koja se unosi u Blender započinje se oblikovanje modela. Ovisno o traženoj razini detalja, to može potrajati od jednog dana do nekoliko tjedana ili mjeseci. U ovom slučaju, modeliranje motorista potrajalo je otprilike jedan tjedan. Svaki komad odjeće poput jakne, hlača ili kacige predstavlja zaseban objekt te je na taj način omogućeno buduće nadograđivanje modela, npr. modeliranje različite odjeće koju igrač može odabrati za svoj lik.

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020



Slika 2-7: Model u Blenderu

U Blenderu (slika 2-7) su također dodani materijali za različite objekte odjeće. Na taj način dodane su boje, a poslije i refleksije na sjajnim predmetima poput kacige. Osim materijala, napravljen je i digitalni kostur (engl. *rigging*). Stvaranje kostura omogućuje nam da pomicanjem odabranih kostiju deformiramo model te da ga tako stavljamo u položaje koje želimo (slika 2-8). Tako je motorist je prije izvoza u Unity stavljen u položaj vožnje motora.



Slika 2-8: Model s podešenim kosturom-

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

2.6.2 Uvoz modela u Unity

Nakon što je motorist uspješno modeliran, izvozi se iz Blendera u .fbx formatu i potom uvozi u Unity (slika 2-9). Podešavanjem sjenčara (engl. *shader*) daje se sjaj i refleksija određenim materijalima poput materijala kacige i vizira. Vozač motora postavlja se u hijerarhiji scene kao dijete objekta motora. Na taj način se pomicanjem motora kroz scenu miče i motorist.



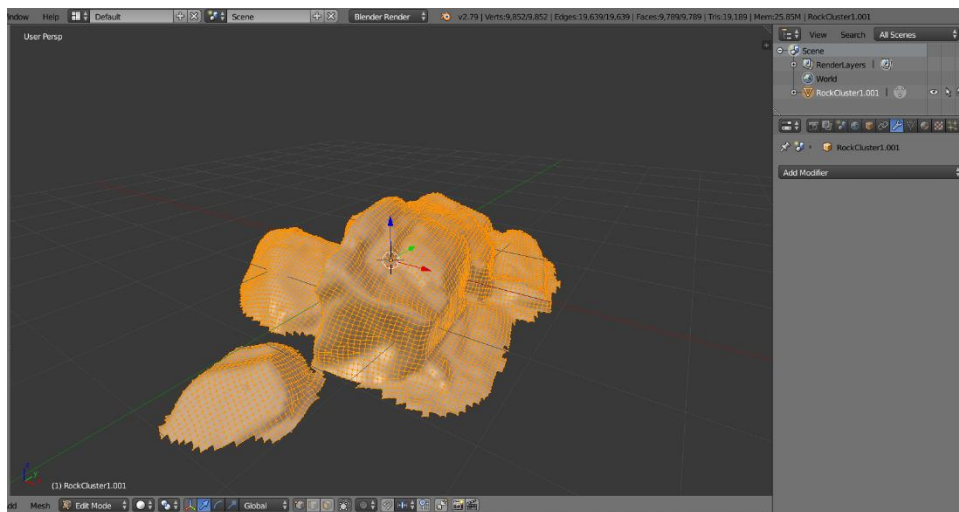
Slika 2-9: Model na motoru

2.6.3 Smanjenje broja poligona preuzetih modela

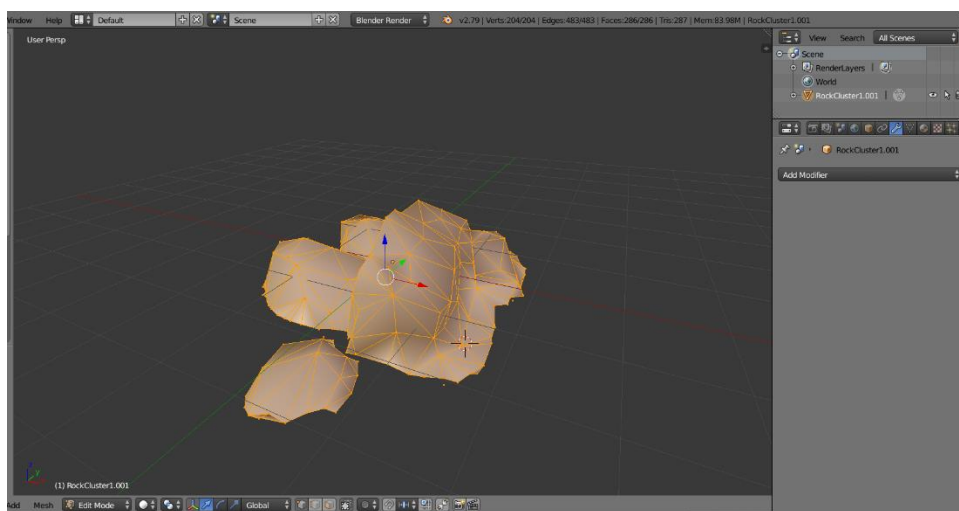
Većina modela korištenih u ManeuVR igrici besplatno su preuzeti sa online trgovina resursa. Međutim, neki od njih previše su detaljni (imaju prevelik broj poligona, slika 2-10) što ruši performanse igre. Taj problem moguće je riješiti i bez višesatne potrage za prikladnijim modelima.

Problematične modele potrebno je jedan po jedan uvesti u Blender i nad njima primjeniti modifikator desetkovanja (engl. *Decimate*) koji smanjuje broj poligona spajajući više susjednih vrhova u jedan. Tako izmjenjeni model treba izvesti iz Blendera i ponovno uvesti u Unity. Na taj način smo znatno smanjili broj prikazanih poligona u sceni te uvelike povećali performanse igre (slika 2-11).

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020



Slika 2-10: Model s 30 000 trokuta



Slika 2-11: Model s 300 trokuta

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

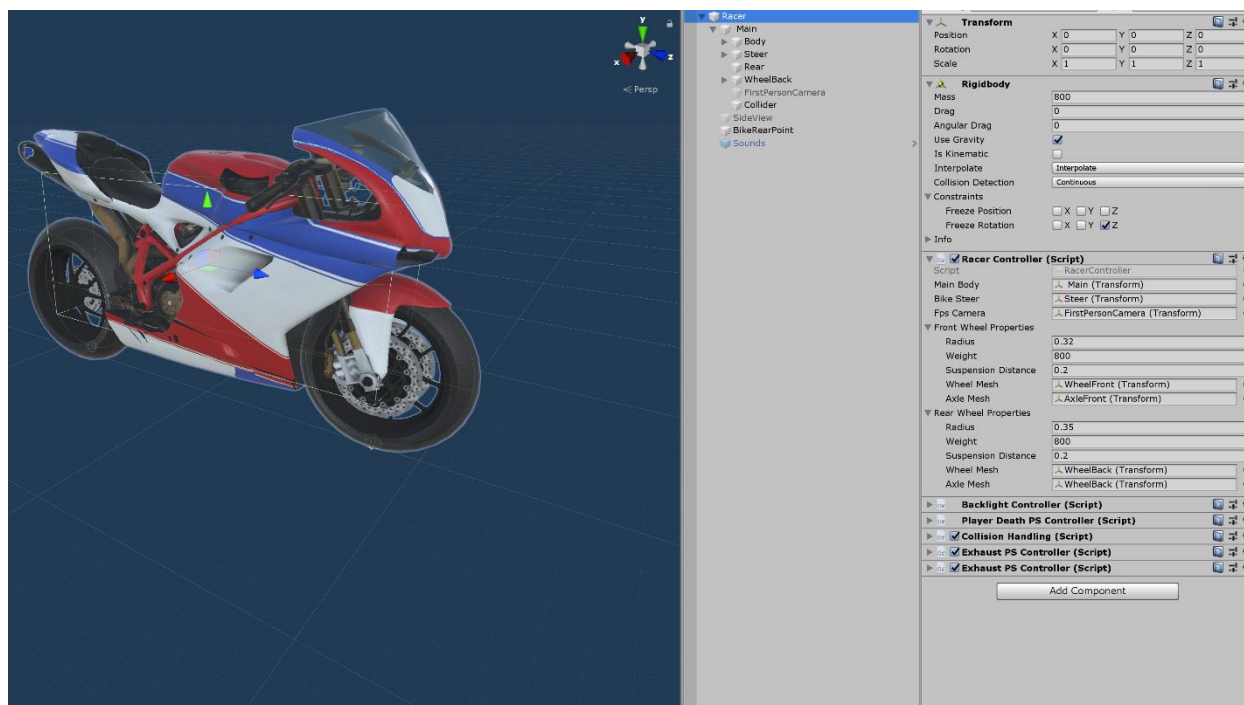
2.7 Fizikalni model motora

Fizika je glavni element igre koji utječe na igrivost i prividni osjećaj stvarnosti. Ovisno o tipu igre fizika može biti vrlo komplicirana i potpuno oponašati stvaran svijet, dok je u drugim igrama fizika prilagođena kako bi igračima igra bila što pristupačnija i zabavnija.

Fizikalni model motora (slika 2-12) sastoji se od korijenskog objekta koji ima svojstva krutog tijela i dva kotača. U Unityju je većina fizike krutog tijela već implementirana u sklopu fizikalnog pokretača (engl. *physics engine*), a objekt poprima svojstva krutog tijela dodavanjem komponente *Rigidbody*. Neki od parametara koje je moguće podesiti za kruto tijelo, a bitno utječu na ponašanje objekta u igri su: masa, gravitacija i inercija.

Jedan od problema koji se javlja prilikom vožnje motora je održavanje ravnoteže, odnosno bočno nagnjanje. Pošto je igra zamišljena u virtualnoj stvarnosti to bi predstavljalo preveliki izazov za igrača. Moguć način zaobilaska ovog problema je simulacija reverznog njihala čime bi se odgovornost održavanja ravnoteže prebacila s igrača na igru. Ipak, pokazalo se da postoje situacije gdje bi ova metoda bila neprikladna pa je u ovom projektu odabran drugačiji pristup.

Problem bočnog nagnjanja riješen je uz pomoć hijerarhijske organizacije objekata. Korijenski objekt motora sadrži komponentu *Rigidbody*, a zamrznuta mu je rotacija oko z osi kako bi se onemogućilo njegovo bočno nagnjanje. Njemu je dodan objekt koji ne sadrži komponentu *Rigidbody* i predstavlja vizualni model motora. Pošto vizualni model nema svojstva krutog tijela onda nema ni ograničenje na rotaciju oko z osi. Na ovaj način problem održavanja ravnoteže je nestao zbog toga što je uklonjeno bočno nagnjanje objekta sa svojstvima krutog tijela, a time i sve bočne sile koje bi narušavale ravnotežu. Nagnjanje motora je ostvareno isključivo podešavanjem lokalne rotacije vizualnog modela oko z osi. Ovim postupkom smo na vrlo jednostavan način dobili potpunu fiziku krutog tijela bez bočne rotacije.



Slika 2-12: Model motora

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

2.8 Gibanje motora

Korijenskom objektu motora dodana su dva objekta od kojih svaki predstavlja jedan kotač i ima ugrađenu komponentu *WheelCollider* koja u Unityju predstavlja sudarač za kotače. To je posebna komponenta s ugrađenom detekcijom sudara namijenjena za vozila. Ukoliko je postavljena kao podkomponenta krutom tijelu, onda se ponaša poput stvarnog kotača. Postoje brojni parametri koje je moguće podešavati, a neki od njih su: masa kotača, radijus kotača, tvrdoća amortizera, trenje. Komponenta *WheelCollider* daje korisne informacije poput trenutnog broja okretaja u minuti te pozicije sudara kotača s nekim drugim objektom. Pokretanje kotača, a ujedno i motora ostvaruje se primjenom momenta sile (engl. *torque*) nad ovom komponentom.

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

2.9 Kamere

Postoje tri pozicije kamere (slika 2-13): prvo lice, treće lice i bočni pogled. Pozicija kamere mijenja se dinamički pomoću skripte. Translacija se provodi bez interpolacije, dok se rotacija kamere linearno interpolira prema odgovarajuće kutu čime se smanjuje utjecaj naglih pokreta kamere prilikom doskoka ili sudara.



Slika 2-13: Pogledi kamere

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

2.10 HTC Vive

HTC Vive (slika 2-14) je uređaj za prikaz virtualne stvarnosti. Razvile su ga tvrtke HTC i Valve Corporation. Sastoji se od naočala koje služe za prikaz virtualne stvarnosti, dvije bazne stanice za praćenje igrača u prostoru i dva kontrolera.



Slika 2-14: HTC Vive

2.11 Integracija VR tehnologije

Unity pruža jednostavan API za integraciju i korištenje VR tehnologije. Podržava više nativnih softverskih biblioteka, a u ovom projektu je korištena biblioteka OpenVR. Ipak, ova biblioteka nije dovoljna za sve funkcionalnosti HTC Vive tehnologije. Na skladištu imovine (engl. *asset store*) može se pronaći dodatak pod nazivom SteamVR.

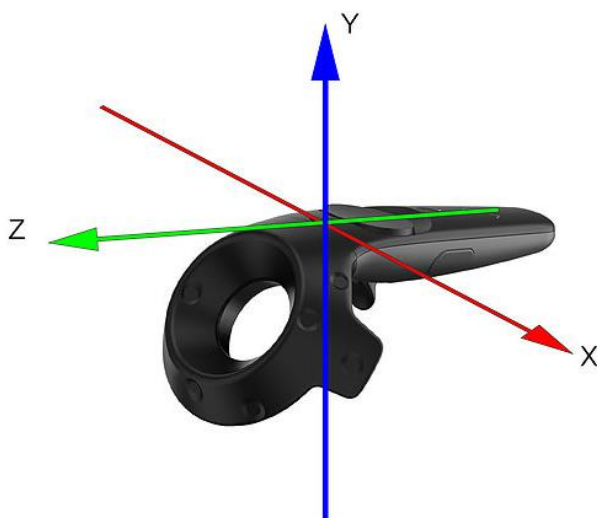
Steam VR je priključnica (engl. *plugin*) za Unity, a služi kao most između Unityja i HTC Vive uređaja. Omogućava tri važne dodatne funkcionalnosti: učitavanje 3D modela za prikaz pozicije kontrolera u sceni, prepoznavanje pritiska tipki na kontroleru i njihovo pridruživanje određenom događaju te određivanje položaja korisnikove ruke pri korištenju kontrolera.

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

2.12 Kontrole

SteamVR podržava više različitih vrsta kontrolera te je stoga povoljan za razvoj računalnih igara. Sastoji se od skupova akcija koje se pomoću posebne aplikacije povezuju s tipkama na kontroleru (slika 2-15). Time se dobiva razina apstrakcije koja znatno olakšava kasniju prilagodbu igre drugoj vrsti kontrolera.

Za ovaj projekt bilo je potrebno definirati samo jednu dodatnu akciju, a to je kočnica. Ostale kontrole motora ostvarene su preko pozicije i rotacije kontrolera u prostoru. Gas predstavlja horizontalna vrtnja desnog kontrolera oko z osi, skretanje ostvareno pomoću pozicije kontrolera duž x osi, dok je vožnja motora na stražnjem kotaču ostvarena pomoću y osi.



Slika 2-15: HTC Vive upravljač

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

2.13 Prikaz vodenih površina

Iscrtavanje (engl. rendering) i simulacija realističnih vodenih površina je jedan od temeljnih problema u području računalne grafike i izrade računalnih igara. Kroz povijest računalne grafike, razvijeno je mnoštvo metoda iscrtavanja vodenih površina. U izradi naše igre, koristili smo dvije različite metode za dva različita cilja:

- Iscrtavanje malih površina voda stajaćica
- Iscrtavanje oceana

2.14 Iscrtavanje malih površina voda stajaćica

Simulacija malih površina voda stajaćica napravljena je pomoću animiranja UV koordinata koje se koriste za uzorkovanje (engl. sampling) normalne mape.[1] UV koordinate se animiraju koristeći mapu toka – teksturu koja za svaku točku na površini vode sadrži dvodimenzionalni vektor. Time je napravljen temelj prikaza vodenih površina jer su one većinom određene vektorom normale koji se koristi za izračunavanje osvjetljenja. Na taj osnovni efekt dodaju se efekt zamućenja vode koji je modeliran eksponencijalno opadajućom maglom te efekt refrakcije koji je temeljen na uzorkovanju dubinske mape pomoću ofseta (engl. offset) koji je dobiven iz normalne mape. Taj pristup nije fizikalno opravdan, ali u praksi izgleda iznenađujuće dobro i ima dobre performanse. Rezultat je prikazan na slici 2-16.



Slika 2-16: Prikaz malih površina voda stajaćica

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

2.15 Iscrtavanje oceana

Prethodno prikazan pristup animiranja tekstura nije prikladan za prikaz velikih valova koji se pojavljuju pri simulaciji velikih vodenih površina poput oceana. Za iscrtavanje oceanskih površina koristi se sjenčar koji pomiče točke na vodenoj površini u vertikalnom smjeru tako da provodi linearnu superpoziciju više neovisnih valnih funkcija. Valna funkcija koja je korištena zove se Gerstner valna funkcija koja nosi ime fizičara koji ju je otkrio – Franz Josef Gerstner. Gerstner valna funkcija je jako popularna u interaktivnoj računalnoj grafici zbog odličnih performansi i vizualne kvalitete. [2]

Unityjev ugrađeni sustav razine detalja (engl. level of detail) je pružao neprihvatljive performanse za prikaz velike površine oceana pa smo zbog toga prešli na oceanski sustav Crest koji ima napredan sustav razine detalja visokih performansi [3]. Navedeni sustav bilo je potrebno konfigurirati za naše potrebe. Proveli smo konfiguraciju simulacije valova, konfiguraciju postavki vodene pjene te konfiguraciju dubine vode koja služi za razrjeđivanje valova i generiranje pjene u ovisnosti o dubini scene. Za simulaciju valova nastavili smo koristiti prethodno opisanu metodu superpozicije više Gerstner valnih funkcija. Rezultat nakon konfiguracije prikazan je na slici 2-17.



Slika 2-17: Prikaz oceana

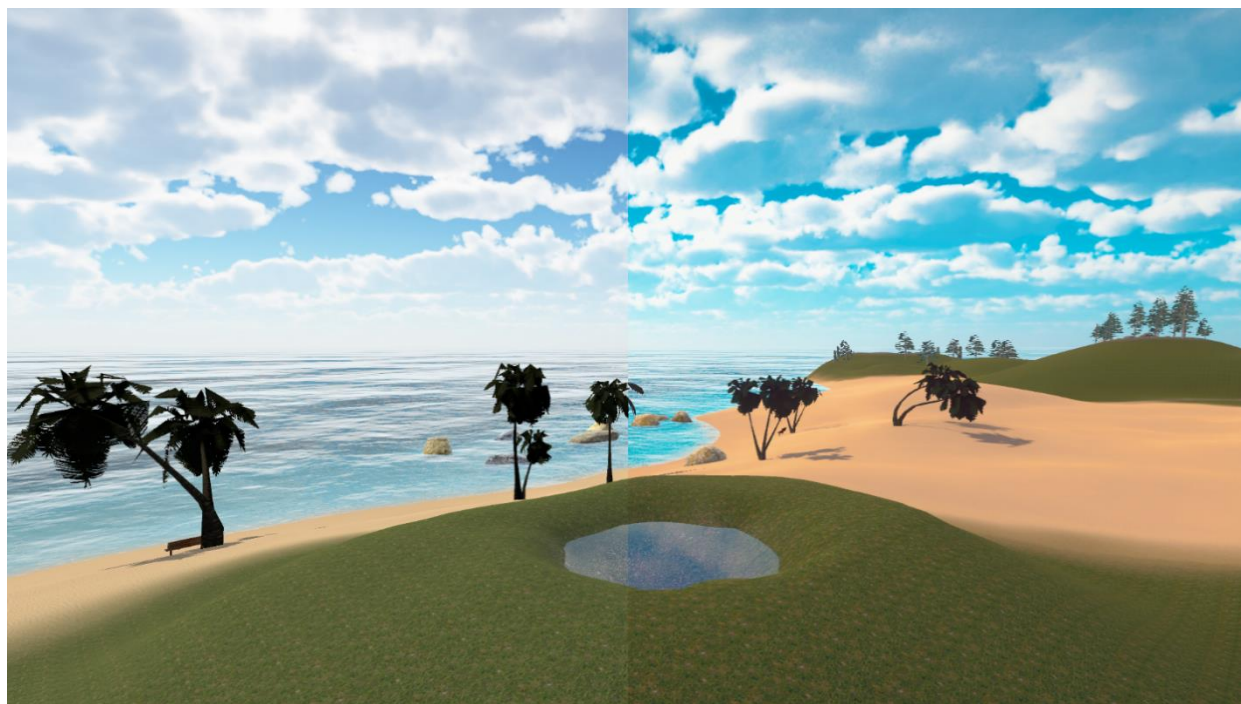
ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

2.16 Naknadna obrada (engl. *post processing*)

Za potrebe naknadne obrade koristili smo poprilično standardan skup efekata:

- Okluzija okoline (engl. screen space ambient occlusion) – za interaktivnu simulaciju ambijentalnog osvjetljenja
- Sjaj (engl. bloom) – za simulaciju artefakta kamera iz stvarnog svijeta u kojem jako svjetlo prodire granice drugih objekata
- Vinjeta (engl. vignette) – za usmjeravanje igračeve pozornosti u centar prikazane scene
- Stupnjevanje boja (engl. colour grading) – za postizanje željenog izgleda finalne scene

Stupnjevanje boja smo napravili tako da smo iz Unityja izveli sliku izgleda scene prije stupnjevanja boja te ju uveli u program DaVinci Resolve. U njemu smo podesili željene parametre boja kao što su nijansa (engl. hue), zasićenje (engl. saturation) i vrijednost (engl. value). Inspiracija nam je bila tzv. „narančasto-tirkizni izgled“ (engl. „orange and teal look“). Nakon što je postignut željeni izgled, izvedena je tablica za pretraživanje (engl. lookup table) – tekstura koja se potom koristi za stupnjevanje boja u Unityju. Rezultat je prikazan na slici 2-18. Lijeva polovica slike prikazuje scenu prije stupnjevanja boja, a desna polovica slike scenu nakon stupnjevanja boja.

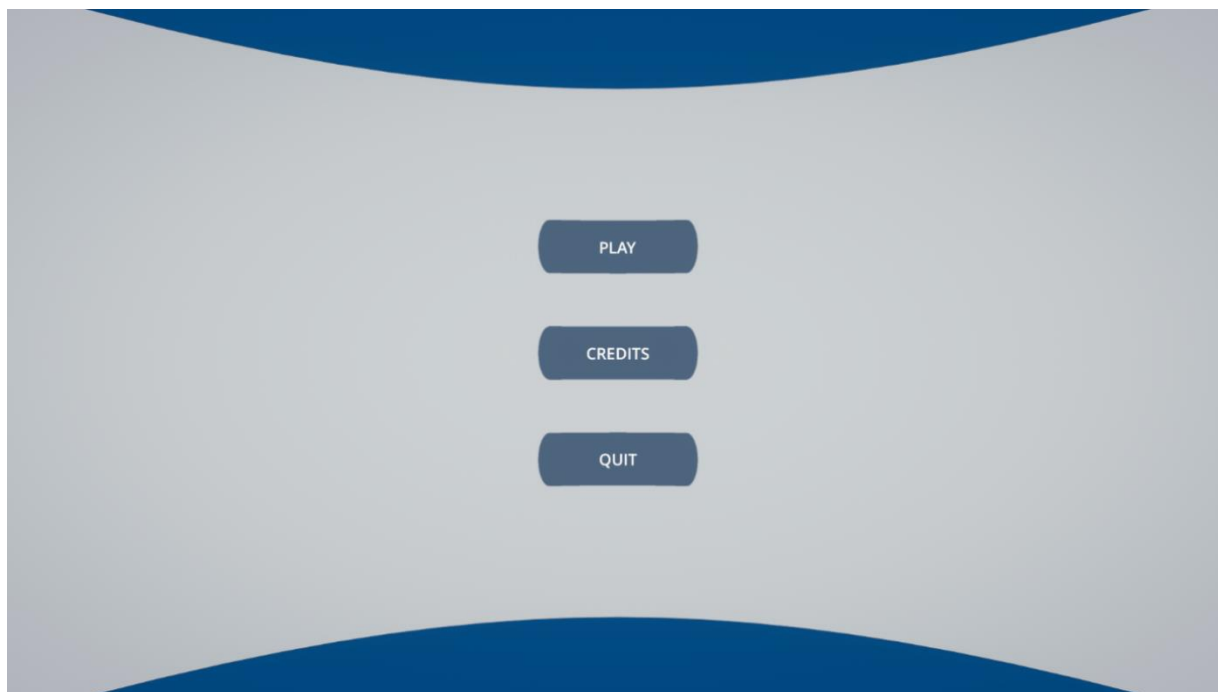


Slika 2-18: Scena prije i poslije stupnjevanja boja

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

2.17 Grafičko korisničko sučelje

Kako je naša igra izrađena za platforme virtualne stvarnosti kojima je cilj što realističnija iluzija prikaza stvarnog svijeta, koristili smo minimalan skup komponenata grafičkog korisničkog sučelja. Glavni izbornik koji je prikazan na slici 2-19 napravljen je pomoću standardnih Unity komponenti za stvaranje grafičkog korisničkog sučelja. Svaka komponenta grafičkog korisničkog sučelja je postavljena kao dijete posebnog objekta platna (engl. canvas). Glavni elementi korisničkog sučelja koji su zaslužni za samu interakciju s korisnikom su gumbi. Korisnik s njima vrši interakciju pomoću VR kontrolera koji služe kao laseri za selektiranje gumba. Kada je korisnik fokusiran na gumb ili ga klikne, pruža mu se povratna informacija (engl. feedback) u obliku zvučnog i vizualnog efekta. Pošto je ovo igra za virtualnu stvarnost, platno se nalazi u trodimenzionalnom prostoru te mu je dodana skripta koja ga zakrivljuje te pridodaje predodžbi trodimenzionalnog prostora.



Slika 2-19: Glavni izbornik

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

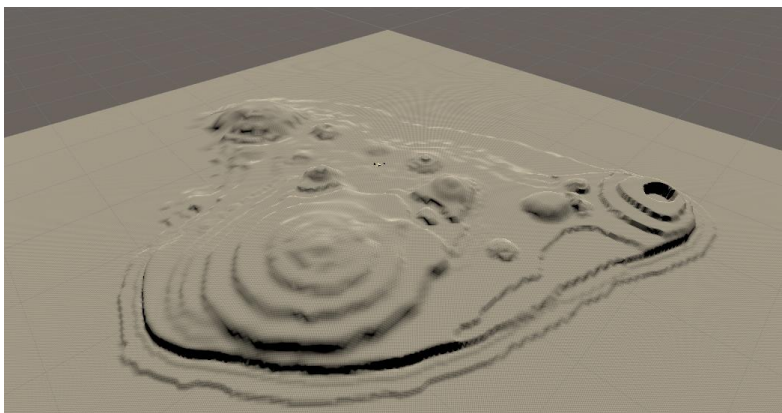
2.18 Dizajniranje mape

Mapa je dizajnirana po uzoru na Uskršnji otok (slika 2-20) koji je približno oblika jednakokraničnog trokuta. Dok je u stvarnosti jedna stranica dugačka otprilike 20 km, u našoj igri otok je umanjen približno 30 puta te je ta stranica dugačka oko 600 metara, čime dobivamo igrivu površinu od ~160 000 m². Nadmorske visine pojedinih dijelova otoka skalirane su približno 15 puta kako bi se u igri istaknula elevacija terena i time ga se učinilo zanimljivijim.



Slika 2-20: Topografija Uskršnjeg otoka [6]

Cijeli teren izrađen je ručno bez korištenja gotovih visinskih mapa (engl. *height map*). Modeliran je koristeći *Unity Terrain Toolbox* i pripadne alate. Cijeli teren stvara se izdizanjem dijelova ravnine (slika 2-21). Zbog tog ograničenja nije moguće izrađivati tunele ili slične kompleksne strukture.



Slika 2-21: Početak izdizanja terena i stvaranja osnovnog oblika otoka

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

Za dodavanje teksture (engl. *Terrain Layer*) na teren korišten je kist (engl. *brush*) naziva *Paint Texture*. Tekstura terena pohranjuje se kao više PNG slika (engl. *splat*) u kojima tri boje (crvena, zelena i plava – RGB sustav) označavaju pozicije na terenu na kojima se mora prikazati tekstura koja je mapirana na tu boju. U različitim slikama se na isti kanal mapiraju različite teksture. U našem projektu aktivno se koriste četiri takve slike. Također je korištena maksimalna razina detalja zbog čega svaka slika ima 2048x2048 piksela te 21,3 MB.

Svi gotovi resursi (engl. *asset*) dobavljeni su iz raznih repozitorija na Internetu, ponajviše *Unity Asset Store* i *Turbosquid*. Od stotinjak pronađenih potencijalnih resursa u igri je iskorišteno njih približno 40. Na mapu (slika 2-22) je postavljeno preko 200 instanci tih resursa, ponajviše crnogoričnog drveća i palmi, desetak vrsta kamenja te mnogo kamenih skulptura ljudskih figura karakterističnih za Uskršni otok, zvanih moai.



Slika 2-22: U potpunosti dovršena mapa

2.19 Verzioniranje projekta

Za verzioniranje projekta korišten je *Git*. Za pohranu nekih vrsta datoteka korišteno je proširenje *Gita* pod nazivom *Git Large File Storage (LFS)*. Inicijalno je projekt postavljen na platformu *GitHub*. Ubrzo nakon početka rada na projektu uočeno je da je *GitHub* smanjio ograničenje veličine repozitorija s 5 GB na 1 GB te je sustav zaključao projekt u trenutku kada smo dosegli 1,5 GB. Iz tog je razloga cijeli projekt preseljen na platformu *GitLab* koja veličinu repozitorija ograničava na 10 GB. U konačnoj verziji projekt ima 5,2 GB [7].

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

3. Upute za korištenje

Pokretanjem izvršne datoteke otvara se glavni izbornik (Slika 2-19). Sa upravljačima VR sustava odabire se jedna od ponuđenih opcija. “Play” pokreće igru, “Credits” prikazuje imena svih developera, a “Quit” izlazi iz igre.

Odabirom opcije “Play” započinje igra. Upravljači VR sustava moraju biti postavljeni u horizontalan položaj (gornji dio upravljača prema unutra). Rotacijom upravljača prema naprijed motor ubrzava, a rotacijom prema unazad usporava. Skretanje lijevo ili desno postiže se povlačenjem onog upravljača koji je na strani na kojoj želite skrenuti prema sebi, dok suprotni odmičete od sebe (na isti način kako se kontrolira bicikl ili motor). Povlačenjem oba upravljača iznad glave motor se propinje na zadnji kotač.

Igra se pobjeđuje tako da korisnik ostane posljednji igrač u igri. Ukoliko igrač miruje duže od 6 sekundi ili se zaleti u trag drugih igrača (ili vlastiti) igra je za tog igrača završena.

ManeuVR	Verzija: 1.0
Tehnička dokumentacija	Datum: 24/01/2020

4. Literatura

- [1] Alex Vlachos. Water flow in Portal 2. ACM SIGGRAPH, 2010.
- [2] GPU Gems. Chapter 1: Effective water simulation from physical models.
<https://developer.nvidia.com/gpugems/gpugems/part-i-natural-effects/chapter-1-effective-water-simulation-physical-models>
- [3] Crest ocean system: User guide.
<https://github.com/crest-ocean/crest/blob/master/USERGUIDE.md>
- [4] Juliani, A., Berges, V., Vckay, E., Gao, Y., Henry, H., Mattar, M., Lange, D. (2018). Unity: A General Platform for Intelligent Agents. *arXiv preprint arXiv:1809.02627*. <https://github.com/Unity-Technologies/ml-agents>.
- [5] Wikipedia, *Hairy ball theorem*, 6.1.2020., https://en.wikipedia.org/wiki/Hairy_ball_theorem, pristupano 11.12.2019.
- [6] Wikipedia, Easter Island map, prosinac 2011.,
https://commons.wikimedia.org/wiki/File:Easter_Island_map-en.svg, pristupano 20.1.2020.
- [7] GitLab repozitorij projekta, sječanj 2020., <https://gitlab.com/ManeuVR/ManeuVR>