

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 868

SIMULACIJA VOŽNJE

Dario Kiramarios

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 868

SIMULACIJA VOŽNJE

Dario Kiramarios

Zagreb, lipanj 2023.

ZAVRŠNI ZADATAK br. 868

Pristupnik: **Dario Kiramarios (1191246601)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentorica: prof. dr. sc. Željka Mihajlović

Zadatak: **Simulacija vožnje**

Opis zadatka:

Proučiti grafički programski paket Unity, a posebice mogućnosti povezivanja s vanjskim napravama za upravljanje vozilom, odnosno volanom, mjenjačem i pedalama. Na modelu cesta ostvariti simulaciju vožnje korištenjem vanjskih naprava za upravljanje simulacijom vožnje. Diskutirati utjecaj različitih parametara. Načiniti ocjenu rezultata i implementiranih algoritama. Izraditi odgovarajući programski proizvod. Koristiti grafički programski paket Unity. Rezultate rada načiniti dostupne putem Interneta. Radu priložiti algoritme, izvorne kodove i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 9. lipnja 2023.

Zahvaljujem prof. dr. sc. Željki Mihajlović na mentorstvu i pomoći pruženoj pri pisanju završnog rada.

Sadržaj

Sadržaj	ii
Uvod	1
1. Osnovne komponente	2
1.1. Model automobila.....	2
1.2. Model kotača	3
1.3. Virtualni amortizeri	3
1.4. Model svijeta	4
2. Programsko ostvarenje	6
2.1. Razred „Brzina“.....	6
2.2. Skripta „Brzinomjer“.....	6
2.3. Skripta „LogiWheelController“.....	8
2.3.1. Upravljanje tipkovnicom.....	11
2.3.2. Upravljanje volanom	12
2.4. Skripta „SceneLoader“	14
2.5. Skripta „CameraSwitch“.....	15
2.6. Skripta „ZvukAuta“.....	17
3. Rezultati	18
3.1. Ocjena rezultata	18
3.2. Razlika u simulaciji vožnje tipkovnicom i Logitech G29 volanom	18
3.3. Diskusija implementiranih algoritama	18
Zaključak	19
Literatura	20
Sažetak	21

Summary	22
Privitak	23

Uvod

Simulacija vožnje je ključni alat u razvoju i istraživanju automobilskih tehnologija. Kroz simulaciju vožnje, moguće je stvoriti sigurno, kontrolirano i realistično okruženje koje omogućava testiranje novih ideja i koncepta prije nego što se primijene u stvarnim vozilima. Ova tehnika ima široku primjenu u područjima kao što su automobilska industrija, obuka vozača, istraživanje sigurnosti prometa i razvoj autonomnih vozila.

Cilj ovog završnog rada je razviti jednostavnu simulaciju vožnje u okruženju Unity, koristeći tipkovnicu ili volan s pedalama i mjenjačem. Simulacija je osmišljena tako da pruža korisnicima približno iskustvo vožnje automobila i ovisno o odabranim parametrima i malo zabavnije te brže iskustvo vožnje.

Unity je popularan alat za razvoj igara i simulacija, poznat po svojoj intuitivnom korisničkom sučelju i snažnim mogućnostima za stvaranje realističnih virtualnih svjetova. Kroz integraciju sa stvarnim kontrolerima, poput Logitech G29 volana s pedalama i mjenjačem, simulacija vožnje postaje još uvjerljivija i korisnici mogu doživjeti realistične senzacije vožnje.

U okviru ovog završnog rada, detaljno će se opisati arhitektura simulacije vožnje u Unity okruženju, uključujući implementaciju upravljanja putem tipkovnice i volana s pedalama i mjenjačem. Također će se diskutirati utjecaj različitih parametara na sami osjećaj vožnje i provesti evaluacija simulacije kroz testiranje i analizu performansi vožnje.

Kroz ovaj rad, cilj je pružiti korisnicima uvod u realističnu simulaciju vožnje, ali i pružiti platformu za daljnje istraživanje i razvoj u području simulacija vožnje vozila.

1. Osnovne komponente

Simuliranje vožnje je napravljeno u programskom paketu Unity koristeći tipkovnicu i Logitech G29 volan, pedale i mjenjač. Ovo nije u potpunosti realistična simulacija vožnje već je uvod u simulaciju vožnji.

1.1. Model automobila

Za sami model automobila se može preuzeti bilo koji model koji je napravljen za Unity, ali se mora pripaziti da su kotači zasebni dijelovi automobila. To je bitna napomena jer se na tome temelji ovaj cijeli rad tj. kotači moraju biti posebni dijelovi jer svaki ima svoje ponašanje. Na sam auto je povezana skripta „LOGIWheelController“ o kojoj ću kasnije pisati, ali je također povezana komponenta krutog tijela kojom definiramo masu, koeficijent otpora i mnoge druge parametre automobila.

Za ovaj projekt sam odabrao model automobila BMW M3 E46 (slika 1.1) koji je došao „u paketu“ zajedno s paketom *Realistic Car controller*¹. Automobil je jako detaljno napravljen ima puno zasebnih dijelova poput unutrašnjost auta, zasebni farovi, motor koji se vidi kada se hauba podigne i slično, ali to nije bilo bitno za ovaj projekt pa tako nije niti iskorišteno.



Slika 1.1: Model automobila BMW M3 E46

1.2. Model kotača

Kotači su zasebni dijelovi automobila i svaki kotač je neovisan o ostalima. Svaki kotač ima određenu snagu, brzinu, jačinu kočenja, akceleraciju i ponašanje svojih amortizera koji su poveznica između kotača i auta. Svaki kotač ima mrežnu (Mesh) komponentu i komponentu sudarača (Collider). Komponenta sudarača je bitna komponenta za simulaciju vožnje. Putem te komponente se može definirati masa samih kotača, njihova veličina te razne postavke za virtualne amortizere koji povezuju kotače i tijelo automobila.



Slika 1.2: Slika modela kotača

1.3. Virtualni amortizeri

Postoje cijela znanost koja se bavi finim podešavanjem amortizera i proučavanjem koje su postavke optimalne za određeni tip vožnje. U ovom su radu odabrane neke jednostavne vrijednosti koje približno simuliraju stvarni osjećaj vožnje. Za samu tvrdoću opruga i koliko oni ublažuju vožnju automobila su odabrane visoke brojke jer je inače automobil jako

„plesao“ po cesti, a to nije realistično za takav tip automobila. Nakon što su postavljene dovoljno velike brojke za ponašanje virtualnih amortizera i njegove parametre, auto se „umirio“ i imao puno bolji osjećaj upravljanja na cesti. Specifično, postavljeno je da su amortizeri jako „tvrđi“ tj. da se ne komprimiraju previše i samim time se auto ne nagnje ni približno puno kao prije. Primjer upisanih vrijednosti koje su korištene u simulaciji vožnje su prikazani na slici 1.3.

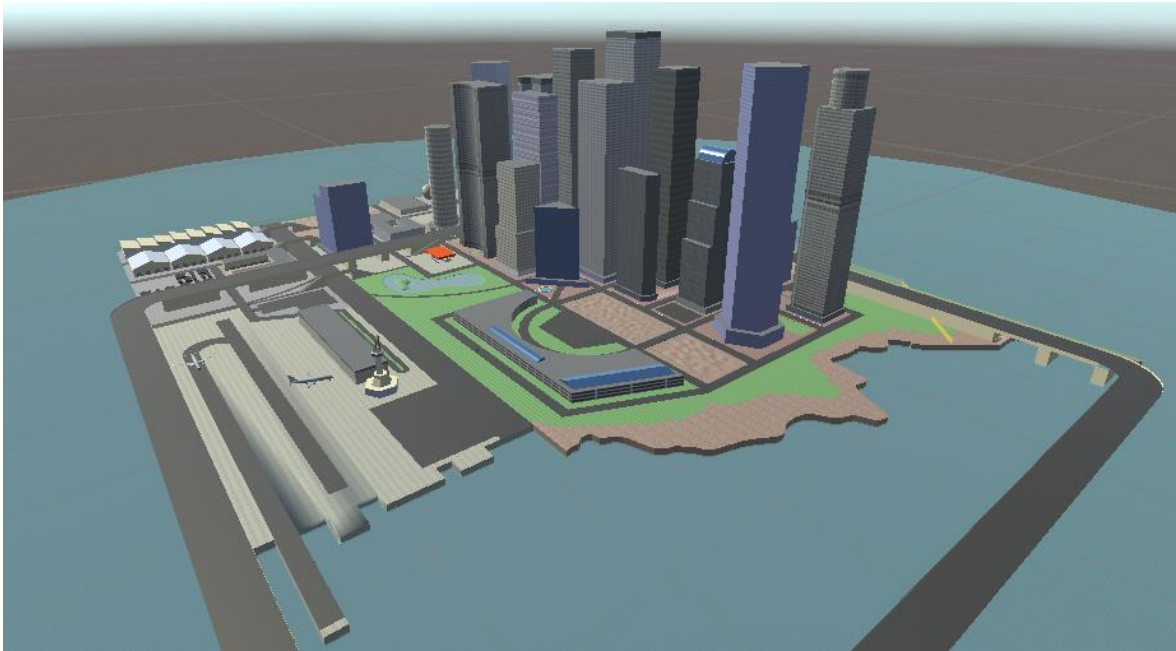


▼ Suspension Spring	
Spring	30000
Damper	3000
Target Position	1
▼ Forward Friction	
Extremum Slip	0.4
Extremum Value	1
Asymptote Slip	0.8
Asymptote Value	0.5
Stiffness	1
▼ Sideways Friction	
Extremum Slip	0.2
Extremum Value	1
Asymptote Slip	0.5
Asymptote Value	0.75
Stiffness	1

Slika 1.3: Vrijednosti parametara virtualnog amortizera

1.4. Model svijeta

Model svijeta dolazi „u paketu“ s *Realistic Car controller*¹ i on je samo učitani i postavljen te ništa nije mijenjano nad njime. Model grada je prikazan na slici 1.4.



Slika 1.4: Slika modela grada

2. Programsko ostvarenje

2.1. Razred „Brzina“

Kako ne bi nastalo zabune riječ Brzina pisano velikim slovom označava trenutnu brzinu u kojoj se nalazi automobil, a kada piše malim slovom brzina onda se referira na sami iznos vektora kretanja automobila.

Razred Brzina je osnovni razred na kojem se temelji ubrzanje i mijenjanje Brzina tijekom vožnje. Razred je jednostavan i svaka Brzina ima svojstvo „gear“ tj. cjelobrojni broj, „minBrzina“ i „maxBrzina“ koji su brojevi s decimalnom vrijednošću koji označavaju minimalnu i maksimalnu brzinu tj. raspon Brzine. To je tako napravljeno kako bi se bolje simulirao osjećaj mijenjanja Brzina i ubrzanja zbog nedostatka stvarnog modela motora koji pokreće automobil. Kada bi se radila puno detaljnija verzija simulacija vožnje, trebao bi se kreirati model motora automobila koji ima klipove koji se pokreću i pomoću kojih se onda računaju stvarni „okretaji motora“ i pomoću kojih se onda kreiraju Brzine automobila.

Ovaj se razred koristi u skripti zvanj „LogiWheelController“ radi boljeg simuliranja karakteristika automobila. Sami razred „Brzina“ se vidi na slici 2.1 .

```
public class Brzina
{
    public int gear;
    public float minBrzina;
    public float maxBrzina;
}
```

Slika 2.1: razred Brzina

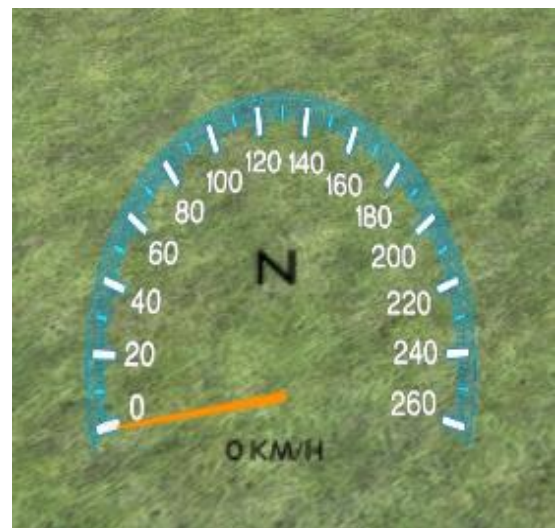
2.2. Skripta „Brzinomjer“

Kako bi se vidjela trenutna brzina automobila na način koji je svim vozačima poznat, koristi se brzinomjer. Na brzinomjeru je upisan raspon od 0 do 260 *km/h* jer to je raspon brzina

kojim se automobil može kretati. Skripta je povezana sa samim automobilom i tako očitava stvarnu brzinu automobila i Brzinu u kojoj se automobil nalazi. Također ta skripta upravlja okretanjem kazaljke koja ukazuje trenutnu brzinu automobila na digitalnom brzinomjeru. Uz to, ispisuje se trenutna brzina s brojkama, ali se i također vidi trenutna Brzina na vrhu brzinomjera. Pri ispisivanju trenutne brzine i bilo kakvim računicama se trebalo dodatno množiti s faktorom 3.6 jer to je faktor kojim se preračunava iz m/s u km/h koji se prikazuju na brzinomjeru. Primjer brzinomjera tijekom vožnje i njegovog prikazivanja brzine, Brzine i kazaljke su prikazane na slici 2.2. Također, ako je auto postavljen da ide unazad („u rikverc“) ili da miruje („u leru“) onda se na brzinomjeru ispisuje „R“ ili „N“ od engleskih riječi „reverse“ i „neutral“ i to je prikazano na slici 2.3 i 2.4. Sliku brzinomjera i strelicu sam preuzeo s githuba³.



Slika 2.2: Primjer brzinomjera pri vožnji



Slika 2.3: Primjer brzinomjera pri stajanju



Slika 2.4: Primjer brzinomjera pri kretanju unatrag

2.3. Skripta „LogiWheelController“

Ovo je glavna skripta na kojoj se zasniva cijela simulacija. U njoj je određeno na koji način se upravlja automobilom tj. upravlja li se Logitech G29 volanom, pedalama i mjenjačem ili tipkovnicom. Ako skripta primijeti da je Logitech G29 spojen, onda je automatski odabran taj način vožnje, no ako volan nije spojen, automobilom se upravlja tipkovnicom. Stoga će ovaj dio objašnjavanja skripte biti razdvojen u 2 dijela, upravljanje tipkovnicom i upravljanje volanom.

Prije toga, općenito o skripti. Skripta u sebi sadrži mnogo varijabli i parametara, a neki od bitnijih su CurrentGear, listaBrzina, acceleration, breakingForce, maxTurnAngle, backLock i multiplier. CurrentGear nam govori u kojoj Brzini se automobil trenutno nalazi i on je u rasponu od -1 do 6 gdje je -1 za ići unatrag, 0 je „ler“, a ostale Brzine odgovaraju njihovoj brojčanoj reprezentaciji. ListaBrzina je lista koja u sebi sadrži objekte Brzina i pomoću kojeg onda znamo koliki je raspon svake Brzine i koliko brzo se automobil može kretati u toj Brzini. Acceleration je zapravo snaga automobila koju korisnik može zadati. Korisnik također može zadati breakingForce koji opisuje koliko „jako“ automobil može usporiti i maxTurnAngle koji predstavlja maksimalni kut rotacije kotača pri okretanju volana. BackLock predstavlja binarnu vrijednost (istina ili laž) za zaključavanje okretanja stražnjih kotača. Naime, skuplji automobili imaju mogućnost da im se stražnji kotači pri nižim brzinama okreću u suprotnom smjeru od smjera okretanja prednjih kotača kako bi im se radijus okretanja auta smanjio. Također imaju mogućnost da im se pri višim brzinama stražnji kotači okreću u istom smjeru kao i prednji što pridonosi stabilnosti automobila pri većim brzinama. To je implementirano u simulaciju kako bi automobil bio „gipkiji“ pri nižim brzinama, a stabilniji na višim brzinama, no ako se to korisniku ne sviđa, može se isključiti na pritisak gumba. Kod za okretanje kotača se nalazi na slici 2.5. Multiplier služi za korekciju i simulaciju osjećaja kada se automobil kreće presporo za Brzinu u kojoj se nalazi npr. ako se automobil kreće 20km/h u 4. Brzini, automobil će usporeno ubrzavati sve do trenutka kada ne ubrza do brzine koja je upisana kao minimalna brzina za tu Brzinu i to je prikazano na slici 2.6. Također u skripti je ograničena maksimalna brzina kretanja automobila u svakoj Brzini i time se također povećava osjećaj realnosti vožnje. Pri kraju općenitog djela koji je zajednički oba načina upravljanja se nalazi primjena ubrzanja i kočenja na sve kotače vozila te rotacija kotača i kod je vidljiv na slici 2.7 i 2.8.

Na samom početku skripte se poziva Start() funkcija koja poziva funkciju *nadodajBrzine* koja kreira listu Brzina i upisuje njihove minimalne i maksimalne brzine. Lista je numerirana od 0 do 8 gdje je 0 zapravo „rikverc“, 1 je „ler“, 2 je prva Brzina, 3 je druga Brzina itd. Dodatno se učita automobil s „GetComponent<Rigidbody>()“ kako bi se kasnije u kodu mogle raditi manipulacije nad automobilom. Nakon toga se poziva Update() funkcija koja ima već opisani zajednički dio i dio koji je zaseban za svaki način kontroliranja automobila. Također se u ovoj skripti nalazi kod koji povezuje stvarni volan i volan automobila u igrici. Time se ostvaruje povezanost okretanja stvarnog volana s rotacijom volana automobila tijekom simulacije. Ako se koristi Logitech G29, onda se korelacija 1:1, no ako se koristi tipkovnica i tipke „A“ i „D“ onda je efekt simulacije manje realan.

```
//ako su zaključani stražnji kotači, otključaj ih i obrnuto
if (Input.GetKey(KeyCode.L))
{
    backLock = !backLock;
    if (backLock)
    {
        BR.steerAngle = 0;
        BL.steerAngle = 0;
    }
}

if (!backLock) //ako se stražnji kotači smiju okretati
{
    if (auto.velocity.magnitude * 3.6f > 25.0f)
    {
        BR.steerAngle = currentTurnAngle * 0.1f;
        BL.steerAngle = currentTurnAngle * 0.1f;
    }
    else
    {
        BR.steerAngle = -currentTurnAngle * 0.3f;
        BL.steerAngle = -currentTurnAngle * 0.3f;
    }
}
```

Slika 2.5: Kod za okretanje stražnjih kotača ako je opcija backLock uključena te uključivanje/isključivanje opcije BackLock

```

//da ne ide brze od max brzine za tu Brzinu
if (auto.velocity.magnitude * 3.6f >= listaBrzina[CurrentGear + 1].maxBrzina)
{
    acceleration *= 0.0f;
} //ako auto ide presporo za Brzinu
if (auto.velocity.magnitude * 3.6f < listaBrzina[CurrentGear + 1].minBrzina)
{
    acceleration *= 0.3f;
}
else multiplier = 1f;

```

Slika 2.6: Kod za određivanje maksimalne brzine kretanja automobila u određenoj Brzini

```

//ubrzavaj sa svim kotacima
FR.motorTorque = currentAcceleration * 0.5f;
FL.motorTorque = currentAcceleration * 0.5f;
BR.motorTorque = currentAcceleration;
BL.motorTorque = currentAcceleration;

//koci sa svim kotacima
FR.brakeTorque = currentBreakForceFront;
FL.brakeTorque = currentBreakForceFront;
BR.brakeTorque = currentBreakForceBack;
BL.brakeTorque = currentBreakForceBack;

//azuriraj svaki kotac
UpdateWheel(FR, FRTransform);
UpdateWheel(FL, FLTransform);
UpdateWheel(BR, BRTransform);
UpdateWheel(BL, BLTransform);

```

Slika 2.7: Zajednički kod oba načina upravljanja koji primjenjuje odgovarajuće sile na kotače te rotira kotače


```

4 references
void UpdateWheel(WheelCollider col, Transform trans)
{
    //Dohvati trenutno stanje wheel collidera
    Vector3 position;
    Quaternion rotation;

    col.GetWorldPose(out position, out rotation);

    //Postavi stanje kotača
    trans.position = position;
    trans.rotation = rotation;
}

```

Slika 2.8: Kod za rotaciju kotača automobila

2.3.1. Upravljanje tipkovnicom

Upravljanje automobila tipkovnicom se odrađuje tipkama „W A S D“ koje su klasične tipke za kontroliranje na tipkovnici. Dodatne tipke su tipka razmak i tipka „L“ te tipke lijevi „Shift“ i lijevi „Control“. Tipkama „W“ i „S“ se kontrolira ide li automobil naprijed ili unazad i one su ili stisnute ili nisu tj. ili su 0 ili su 1 što nije najbolje za simulaciju vožnji jer uveliko smanjuje realističnost vožnje. Isto vrijedi za tipke „A“ i „D“ za kontroliranje okretanja automobila i pri simulacijama se ova razlika jako očituje u odnosu na upravljanje volanom zbog već navedenog razloga. Primjer koda za pokretanje automobila i zaustavljanje se nalazi na slici 2.9 i 2.10. Tipka „L“ omogućuje ili onemogućuje rotiranje stražnjih kotača, a tipka razmak jače koči i zaustavlja automobil brže nego da pritisćemo tipku „S“.

```

//ako stisnemo W ili S, idi naprijed ili nazad
currentAcceleration = acceleration * Input.GetAxis("Vertical") * multiplier;
currentTurnAngle = maxTurnAngle * Input.GetAxis("Horizontal");

//rotiraj kotace i samim time auto
FR.steerAngle = currentTurnAngle;
FL.steerAngle = currentTurnAngle;

//ako stisnemo space/razmak, koristi "rucnu"
if (Input.GetKey(KeyCode.Space))
{
    currentBreakForceFront = breakingForce * 10;
    currentBreakForceBack = breakingForce * 10;
}
else
{
    currentBreakForceFront = 0f;
    currentBreakForceBack = 0f;
}

```

Slika 2.9: Kod pri kojem se učitava korisnikov pritisak tipki „W A S D“ ili tipke razmak te primjena akceleracije ili usporavanja na automobil te rotacija kotaca

```

//Left shift je za povecati brzinu
if (Input.GetKey(KeyCode.LeftShift) && CurrentGear + 1 <= 6 && cooldownTimer <= 0f)
{
    CurrentGear += 1;
    cooldownTimer = cooldownDuration;
}

//Lijevi control je za smanjiti brzinu
if (Input.GetKey(KeyCode.LeftControl) && CurrentGear - 1 >= -1 && cooldownTimer <= 0f)
{
    CurrentGear -= 1;
    cooldownTimer = cooldownDuration;
}

```

Slika 2.10: Kod pri kojem se učitava korisnikov pritisak tipki lijevi „Control“ ili lijevi „Shift“ kako bi se povećala ili smanjila Brzina

2.3.2. Upravljanje volanom

Upravljanje automobila volanom je pravi način za ovu simulaciju. S volanom, pedalama i mjenjačem se simulira pravi osjećaj vožnje. Kako bi sve moglo funkcionirati i kako bi se mogli dobiti ulazni podatci od volana, treba se uvesti u projekt *Logitech gaming SDK*³. Nakon toga se treba povezati LogitechSDK s projektom tako da se kreira objekt i onda

poveže što se želi učitavati s tim objektom i to je prikazano na slici 2.11. Nakon toga se poziva funkcija *GumbiciNaVolanu* kojom se učitava pritisak gumba na volanu i stanje mjenjača tj. u kojoj se Brzini nalazi mjenjač.

```
LogitechGSDK.DIJOYSTATE2ENGINES rec;  
rec = LogitechGSDK.LogiGetStateUnity(0);  
  
GumbiciNaVolanu(rec);
```

Slika 2.11: Kod za učitavanje stanja volana, pedala i mjenjača

Funkcija *GumbiciNaVolanu* provjerava koji je gumb trenutno pritisnut na volanu ili mjenjaču s napomenom da više gumba odjednom može biti pritisnuto. Ako je kvačilo pritisnuto i mjenjač je u nekoj Brzini, onda će se to prikazati na brzinomjeru i auto će se sukladno Brzini i pokretati, no ako se ne pritisne kvačilo, onda auto neće „ući“ u Brzinu. Isto vrijedi ako mijenjamo Brzine putem papučica na volanu koje služe kao povećavanje ili smanjivanje Brzina za 1 Brzinu. Kako bi taj način mijenjanja Brzina radio, također se mora pritisnuti kvačilo.

Raspon okretanja volana je od -32767 do 32767 no kako je to prevelik raspon, onda je to skalirano i translaticirano na raspon od -1 do 1. Taj raspon je odabran jer -1 predstavlja da je volan u potpunosti okrenut u lijevo, a 1 da je u potpunosti okrenut u desno. Slični postupak je napravljen za pedale gdje je raspon pedala za gas, kočnicu i kvačilo skaliran i translaticiran na raspon od 0 do 1 gdje 0 predstavlja nestisnuto stanje, a 1 u potpunosti stisnuto stanje. Primjer koda za skaliranje i translaticiranje ulaznih podataka volana, pedala i mjenjača je prikazan na slici 2.12. Ostatak koda za upravljanje automobila volanom je praktički identičan dijelu koda za upravljanje tipkovnicom, osim što su zamijenjene vrijednosti učitavanja ulaza s tipkovnice s učitavanjem ulaza s volana.

```
//da raspon volana bude [-1,1], -1 je lijevo, 1 je desno
x0s = rec.lX / 32767f;
//da raspon za gas bude [0,1], 0 je nestisnuto, 1 je stisnuto
GasInput = ((-rec.lY / 32767f) + 1) / 2f;
//da raspon kocnice bude [0,1], 0 je nestisnuto, 1 je stisnuto
BreakInput = ((-rec.lRz / 32767f) + 1) / 2f;
//da raspon kvacila bude [0,1], 0 je nestisnuto, 1 je stisnuto
ClutchInput = ((-rec.rglSlider[0] / 32767f) + 1) / 2f;
```

Slika 2.12: Kod za skaliranje i translatiranje ulaznih podataka volana, pedala i mjenjača

2.4. Skripta „SceneLoader“

Pri samom pokretanju igrice mogu se odabrati dvije opcije, a to su, vožnja po gradu i izlazak iz igrice. Pritiskom na odabrani gumb se pokreće se skripta *SceneLoader* koja je povezana na svaki gumb i učitava se odabrana scena i samim time i simulacija. Izgled početnog ekrana se vidi na slici 2.13.



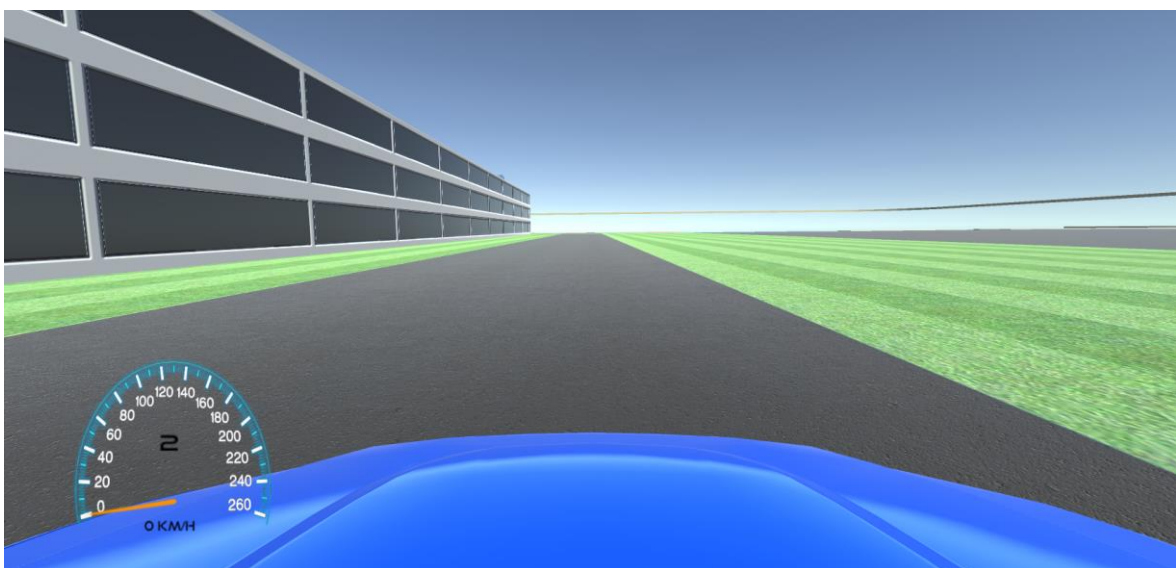
Slika 2.13: Izgled početnog ekrana simulacije

2.5. Skripta „CameraSwitch“

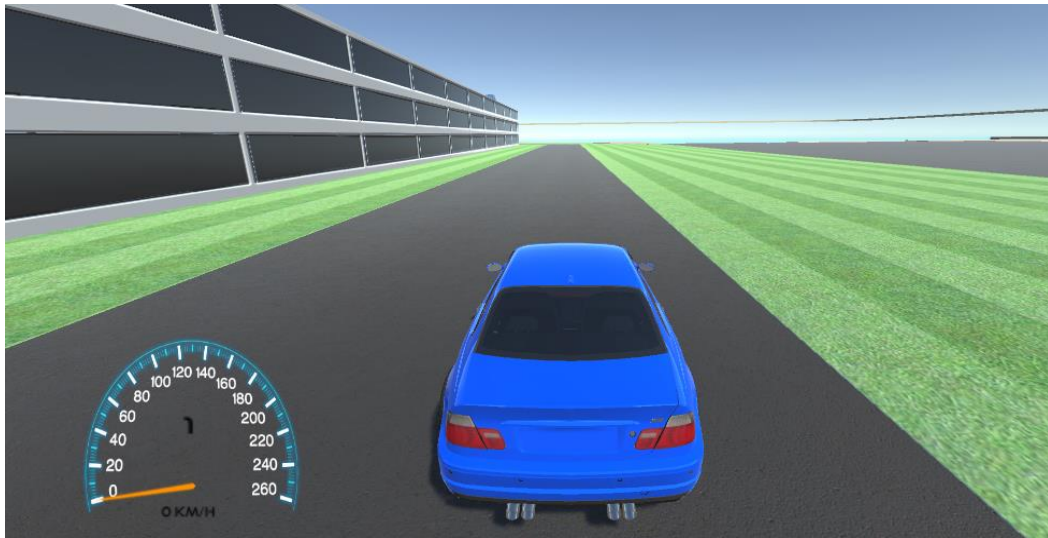
Ova skripta služi za mijenjanje kamere tj. pogleda korisnika tijekom vožnje. Postoje četiri pogleda i to su pogleda unutar kabine automobila, pogleda s haube automobila, pogleda kojim se vidi cijeli automobil tijekom vožnje i pogleda za vožnju unatrag. Pogled se ciklički mijenja na pritisak tipke „C“ na tipkovnici ili na pritisak tipke „X“ na volanu. Primjeri pogleda su na slikama 2.14, 2.15, 2.16 i 2.17.



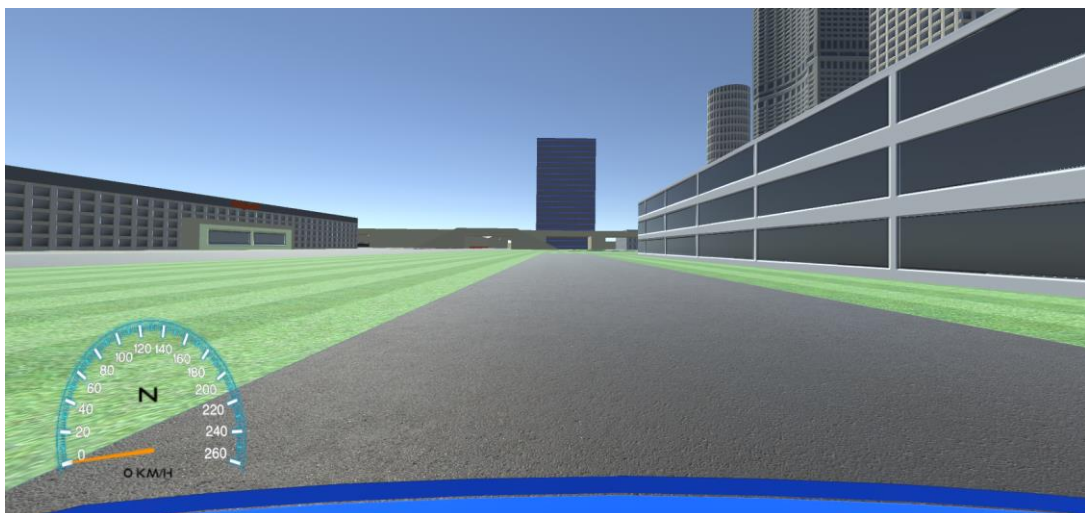
Slika 2.14: Pogled iz unutrašnjosti automobila



Slika 2.15: Pogled s haube automobila



Slika 2.16: Pogled gdje se vidi cijeli automobil



Slika 2.17: Pogled za vožnju unatrag

2.6. Skripta „ZvukAuta“

Kako bi se povećala realnost simulacije uključen je zvuk automobila. Ovo je samo naivni pristup zvuku automobila jer se koristi samo jedan izvor zvuka automobila kojem se mijenja visina ovisno o brzini automobila. Također se visina zvuka mijenja ovisno i o Brzini u kojoj se nalazi. Visina zvuka se određuje na način da ako je automobilu „leru“ ili stoji, visina zvuka je fiksirana na 1, no ako se automobil pokreće, onda visina zvuka raste do maksimalne vrijednosti od 3.5. Kako bi se izračunala visina zvuka, koristila se funkcija *izracunajPitch* koja dobiva trenutnu brzinu automobila i raspon Brzine u kojoj je trenutno te vraća izračunatu visinu zvuka. Primjer koda za skriptu „ZvukAuta“ je prikazan na slici 2.18.

```
void Start()
{
    audioSource = GetComponent();
    audioSource.pitch = defaultPitch;
}

0 references
void Update()
{
    // auto.listaBrzina
    brzina = target.velocity.magnitude * 3.6f;
    if (brzina < 0.01f) brzina = 0.0f;
    if (auto.CurrentGear != 0)
    {
        rasponBrzine = auto.listaBrzina[auto.CurrentGear + 1].maxBrzina - auto.listaBrzina[auto.CurrentGear + 1].minBrzina;

        trenutniPitch = izracunajPitch(brzina, rasponBrzine);

        audioSource.pitch = trenutniPitch;
    }
    else
    {
        audioSource.pitch = 1f;
    }
}

1 reference
float izracunajPitch(float brzina, float rasponBrzine)
{
    float omjer = (3f - 1f) / rasponBrzine; //rasponIzlaza/rasponUlaza
    float izlaznaVrijednost = brzina * omjer + 1f;
    if (izlaznaVrijednost > 3.5f) izlaznaVrijednost = 3.5f;
    return izlaznaVrijednost;
}
```

Slika 2.18: Primjer koda za skriptu „ZvukAuta“

3. Rezultati

3.1. Ocjena rezultata

Ostvarena simulacija je jako dobar uvod u simulaciju vožnje. Dobije se poprilično dobar osjećaj brzine i osjećaj za prostor te vrijeme potrebno za upravljanje automobila. Posebno je zanimljivo kada se počnu mijenjati parametri automobila poput snaga automobila, snage kočenja te ponašanje virtualnih amortizera.

3.2. Razlika u simulaciji vožnje tipkovnicom i Logitech G29 volanom

Korištenje simulacije uveliko je olakšano i realnije ako se koristi volan s mjenjačem i pedalama nego ako se koristi samo tipkovnica. Koristeći volan se dobije puno bolji i realniji osjećaj za upravljanjem automobilom. Sama kontrola automobila puno je „smirenija“ jer korisnik može točno odrediti koliko jako želi stisnuti gas ili kočnicu da automobil ubrza ili uspori ili koliko želi okrenuti volan da automobil skrene. To je nešto što je na tipkovnici nemoguće jer je tipkovnica binarna tj. gumb je ili u potpunosti stisnut ili nije uopće stisnut. Stoga je ciljani način korištenja ove simulacije zapravo uz volan s pedalama i mjenjačem.

3.3. Diskusija implementiranih algoritama

Implementirani algoritmi dobro odrađuju svoji posao, ali pošto je simulacija vožnje jako kompleksna tema tako i algoritmi mogu biti. Napisani algoritmi su savršeni kao uvod u simulaciju vožnje, ali kako bi se ostvario realniji efekt simulacije, trebali bi se koristiti puno napredniji algoritmi, puno složenija fizika te složenije komponente automobila koji se ovdje ne koriste zbog uvodnog pristupa tematici.

Zaključak

U ovom završnom radu uspješno je razvijen uvod u simulaciju vožnje u Unity okruženju, pružajući korisnicima zabavno i realistično iskustvo vožnje. Integracija različitih ulaza za upravljanje omogućila je raznovrstan niz interakcija korisnika, poboljšavajući vjerodostojnost simulacije. Arhitektura simulacije i implementacija upravljanja pokazale su se učinkovitima, omogućujući korisnicima istraživanje i testiranje različitih parametara vozila i reakcija.

Osim toga, evaluacija i analiza performansi provedene u okviru ovog rada potvrdile su sposobnost simulacije da približno dobro prikaže stvarne scenarije vožnje. Rezultati dobiveni testiranjem potvrdili su učinkovitost simulacije i njezin potencijal za daljnje razvijanje i istraživanje u području simulacija vožnje.

Sveukupno, ovaj završni rad je savršena prilika uvoda u simulaciju iskustva vožnje automobila, ali kako bi simulacija bila što realnija trebalo bi još mnogo vremena uložiti kako bi se i najsitniji parametri namjestili.

Literatura

- [1] Realistic Car Controller (RCC),
<https://www.incern.org/2022/04/realistic-car-controller.html>
- [2] Slika brzinomjera i strelica od korisnika TheDeveloper10
<https://github.com/TheDeveloper10/Unity-Speedometer>
- [3] Logitech Gaming SDK,
<https://assetstore.unity.com/packages/tools/integration/logitech-gaming-sdk-6630>
- [4] Sama simulacija te sav popratni kod, skripte i dokumentacija, Dario Kiramarios,
https://gitlab.com/Dario_Kiramarios/simulacija-voznje-fer
- [5] Logitech G HUB,
<https://www.logitechg.com/en-eu/innovation/g-hub.html>
- [6] Logitech Gaming Software,
<https://support.logi.com/hc/en-nz/articles/360025298053-Logitech-Gaming-Software>

Sažetak

Simulacija vožnje

Simulacija vožnje je moćan alat za razvoj i istraživanje automobilskih tehnologija. U ovom završnom radu opisuje se razvoj simulacije vožnje u Unity okruženju, koristeći tipkovnicu, volan s pedalama i mjenjačem. Cilj rada je pružiti korisnicima uvod u realistično iskustvo vožnje automobila te omogućiti testiranje različitih parametara vozila i reakcija vozila na promjene. Kroz integraciju stvarnih kontrolera, simulacija vožnje postaje još uvjerljivija. U radu se detaljno opisuje arhitektura simulacije, implementacija upravljanja te se provodi evaluacija kroz testiranje i analizu performansi. Ovaj rad pruža platformu za daljnje i dublje istraživanje i razvoj u području simulacija vožnje.

Ključne riječi: simulacija vožnje, Unity, volan s pedalama, mjenjač

Summary

Driving simulation

Driving simulation is a powerful tool for the development and research of automotive technologies. This thesis describes the development of a driving simulation in the Unity environment, using a keyboard, a steering wheel with pedals, and a shifter. The objective of this work is to provide users with an introduction to a realistic car driving experience and enable testing of different vehicle parameters and reactions to changes. By integrating real controllers, the driving simulation becomes even more immersive. The thesis provides a detailed description of the simulation architecture, the implementation of controls, and conducts evaluation through testing and performance analysis. This work serves as a platform for further and deeper research and development in the field of driving simulations.

Keywords: driving simulation, Unity, steering wheel with pedals, shifter

Privitak

Preuzimanje simulacije

Simulacija i sav popratni kod, skripte i dokumentacija se mogu preuzeti sa GitLab⁴ profila.

Instalacija programske podrške

Treba instalirati Logitech G HUB⁵ i Logitech Gaming Software⁶ kako bi volan u potpunosti funkcionirao. Nakon toga bi se pokretanjem igrice sve automatski trebalo povezati i raditi bez poteškoća.

Upute za korištenje programske podrške

Dostupne su kratice na tipkovnici:

- tipka Escape – izlazak iz programa
- tipke „W“ i „S“ - pokreću auto naprijed ili nazad
- tipke „A“ i „D“ - rotiraju kotače auta
- tipka razmak - zaustavlja auto
- tipka „Left Shift“ - povećava Brzinu
- tipka „Left Control“ - smanjuje Brzinu
- tipka „C“ - mijenja pogled
- tipka „L“ - zaključava/otključava rotaciju stražnjih kotača automobila

Kontrole na volanu, mjenjaču i pedalama:

- tipka „X“ - mijenja pogled
- tipka kocka - zaključava/otključava rotaciju stražnjih kotača automobila
- desna papučica volana - povećava Brzinu
- lijeva papučica volana - smanjuje Brzinu