

Završni rad

Petar Ivančević

1. kolovoza 2015.

# Sadržaj

|          |                               |          |
|----------|-------------------------------|----------|
| <b>1</b> | <b>Uvod</b>                   | <b>3</b> |
| <b>2</b> | <b>Korištene tehnologije</b>  | <b>4</b> |
| 2.1      | Unity . . . . .               | 4        |
| 2.2      | SketchUp . . . . .            | 4        |
| <b>3</b> | <b>Igrajući objekti</b>       | <b>5</b> |
| 3.1      | Komponente . . . . .          | 5        |
| 3.1.1    | Transformacija . . . . .      | 5        |
| 3.1.2    | Sudarači . . . . .            | 5        |
| 3.1.3    | Kolni sudarači . . . . .      | 5        |
| 3.2      | Mrežne komponente . . . . .   | 6        |
| <b>4</b> | <b>Kotači</b>                 | <b>7</b> |
| 4.1      | Rotiranje kotača . . . . .    | 7        |
| 4.2      | Problemi u zavojima . . . . . | 8        |
| <b>5</b> | <b>Literatura</b>             | <b>9</b> |

# 1 Uvod

Ode treba opisat nešto...

## 2 Korištene tehnologije

### 2.1 Unity

Unity je jedan od najpopularnijih razvojnih platformi za izradu 2D i 3D igrica. Moguće je korištenjem ovog alata napraviti jednu verziju igrice koja će se moći pokretati na računalu, igrajućim konzolama, mobilnim uređajima i web stranicama.

Skripte se mogu pisati u C# ili javascriptu. Preporuka je pisati u C# zbog samog stila pisanja jer se u kratkom periodu dosegne više od pedeset linija kôda i koristi dosta ugrađenih metoda. Za sve ugrađene metoda treba znati koji su argumenti koje primaju, a ako se koristi javascript to se neće moći vidjeti. Framework koji se koristi za pisanje je Monodevelop jer se unity može koristiti na Windows mašinama i na Macintosh mašinama.

Moguće je besplatno koristiti unity dok se ne dosegne prihod od 100,000 dolara. Ukoliko se zaradi ova svota novca preko igre razvijene u unity-u, tada je potrebno kupiti profesionalnu verziju. Trenutna verzija je 5.0, koja nažalost još uvijek ima problema sa linux platformom.

### 2.2 SketchUp

SketchUp je alat za izradu 3D geometrijskih tijela napravljen od strane Google-a. Koristi se većinom u građevinskim obrtima zbog vrlo jednostavnog načina izrade modela. Dvodimenzionalni elementima se jednostavno dodaje treća dimenzija preko gurni/povuci (*eng. Push/Pull*) alata. Ne preporuča se za izradu složenijih modela jer ne pruža dovoljno mogućnosti kao neki drugi alati.

Vozilo korišteno kroz cijelu igru je također izrađen u SketchUp-u. Nakon što se izrade kompletni modeli i njihove animacije, treba se napraviti datoteka koja ima 3ds ekstenziju.

## 3 Igrajući objekti

Svaki objekt koji se može vidjeti u unity-u je **igrajući objekt** (*eng. GameObject*). Kada se napravi bilo koj objekt on treba imati svoju poziciju unutar svijeta. Kako bi se mogla znati njegova pozicija koristimo komponente. Svaki objekt mora imati svoju transformaciju (*eng. Transform*). U suštini igrajući objekti su samo kontejneri koji sadrže komponente.

### 3.1 Komponente

Komponente su kao što je rečeno dijelovi igrajućih objekata. One daju funkcionalnost objektima, te omogućavaju krajnjim korisnicima da razlikuju igrajuće objekte. Neke važnije komponente koje su korištene za izradu igrice su:

- Transformacija
- Sudarači
- Kolni sudarači

#### 3.1.1 Transformacija

Igrajući objekti moraju imati svoju **transformaciju**, inače se neće moći prikazati u svijetu. Transformacija definira širinu, poziciju i orijentaciju objekta. Svaka transformacija ima svoj pivot koji određuje centar, odnosno prema njemu se gledaju širina, pozicija i rotacija. Pivot se može gledati globalno ili lokalno. Globalno gledanje je pozicija pivota gledajući koordinate x,y,z svijeta. Lokalno gledanje pivota je relativna pozicija naspram pozicije objekta.

#### 3.1.2 Sudarači

Dodat opis za sudarače

#### 3.1.3 Kolni sudarači

Unity ima predefinirane elemente za simulirati prave kotače prizemljenih vozila koje zovemo **kolne sudarače**. Prema dokumentaciji unity-a ovi sudarači se ne dodaju kao komponente, već trebaju biti komponente zasebnih igrajućih objekata. Znači za svaki kotač treba napraviti novi igrajući objekt, koji se

zove prazni igrajući objekt (*eng. Empty GameObject*). Doda se komponenta praznom objektu i cijeli objekt se pomakne tako da je centriran sa kotačima.

Najvažnije metode za vožnju su moment motora (*eng. motor Torque*), moment kočnice (*eng. brake Torque*), te kut okretanja (*eng. steer angle*). Moment motora je sila koja djeluje na osovinu kotača izražena u Newton metrima. Predznak sile će odrediti smjer kretanja. Kut okretanja određuje za koliko će se okrenuti model prilikom skretanja, a moment kočnice određuje silu kočenja u Newton metrima. Primjer kôda za kretanje vozila se može vidjeti u ispisu 1

```
for(int i = 0; i < 4; i++)  
    this.wheelsColliders[i].motorTorque = thrustTorque;
```

Ispis 1: Skripta za kretanje vozila

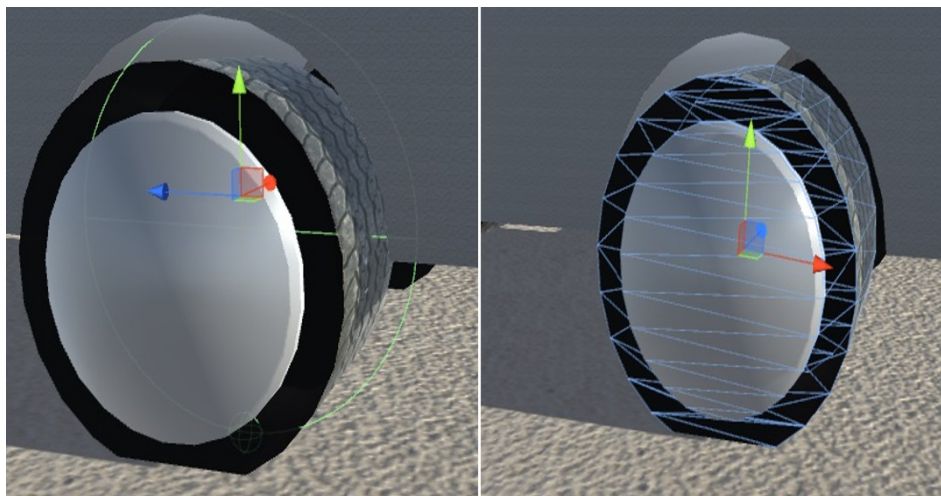
Jedan propust postoji kod ovih sudarača. Naime ukoliko se postave veće brzine kretanja događa se čudna stvar. Model sam počima skretati prema desno.

## 3.2 Mrežne komponente

## 4 Kotači

Kako je navedeno u prethodnom poglavlju kotači se sastoje od više komponenti. Te komponente su mreža (*eng. Mesh*), transformacija, te mrežni prevoditelj (*eng. Mesh Renderer*) koji dopušta korisniku da vidi konačni element. Ono što se koristi za kretanje modela je kolni sudarač (*eng. Wheel Collider*). Mreža se može vidjeti na slici ?? desno, a sudarač na slici ?? lijevo.

Isto tako veoma bitna stvar je povećati masu modela, inače će početi nekontrolirano rotirati na sceni, kao da je upalo u crnu rupu. Vrijednost u konačnici definira kojom silom će gravitacija privlačiti model, odnosno definiramo brzinu. Što je masa veća sporiji je igrajući objekt i obrnuto.



Slika 1: Kotači modela

### 4.1 Rotiranje kotača

Prilikom vožnje automobila za bolju simulaciju potrebno je okretati i kola. Za isprogramirati ovu naizgled jednostavnu radnju više stvari treba unaprijed biti dobro definirano, inače se stvar komplicira. Ukoliko je igrajući objekt loše definiran i pivoti nisu dobro postavljeni na kotačima, zbog mehanike koju unity koristi objekti rotiraju krivo ili treba koristiti naprednije metode koje povećavaju broj linija koda i stvaraju dodatne probleme. Rotacija kotača bi

trebala biti oko njegove osi, te se zato mora postaviti pivot u sredinu kotača. Ovo se obavlja tijekom izrade samog modela, te treba paziti na to prije unosa u unity. Ako se pivot nije centrirao prilikom izrade, onda se treba obavljati popravljavanje. Koraci za popravljavanje:

1. Pronaći objekt (kotač) unutar hijerarhije
2. Napraviti novi prazni objekt na istoj razini kao i kotač
3. Kotač ubaciti u prazni objekt

Sada za okretanje kotača se koristi novi prazni objekt jer je njegov pivot centriran. Ovakav pristup je jako učestalo zbog dizajnera koji ne paze na pivate unutar svojih modela. Skripta za okretanje kola se može vidjeti u ispisu 2:

```
for (int i = 0; i < 4; i++)
    this.tirePivots[i].transform.Rotate(
        Vector3.forward,
        this.speed * move * Time.deltaTime
    );
```

Ispis 2: Skripta za okretanje kola

## 4.2 Problemi u zavojima

Prilikom vožnje vozila dodavanjem samo kolnih sudarača ne bismo dobili potpunu imitaciju pravog vozila. Jedan od problema koji se javlja je preokretanje u zavojima. Ova pojava je sasvim opravdana i nije nikakva greška programa. Što se ustvari događa? Kada vozilo uđe u zavoj i započne skretanje, na njega djeluje centrifugalna sila, podigne se prednji kotač sa tla i kako ne postoji protusila preokrene se.

Za ovaj slučaj postoji više rješenja, a onaj koji se koristi u igrici je sljedeći. Pri skretanju se provjeri koji kotač se podiže od tla i na njega se primjeni sila koja djeluje u smjeru gravitacije. Na ovaj način se smanji utjecaj centrifugalne sile, te samim time teže preokrenuti vozilo.



## 5 Literatura

1. <https://docs.unity3d.com/Manual>
2. <https://docs.unity3d.com/ScriptReference>
3. <http://forum.unity3d.com/threads/how-to-make-a-physically-real-stable-car-with-wheelcolliders.50643/>