

ŠOLSKI CENTER VELENJE  
ELEKTRO IN RAČUNALNIŠKA ŠOLA  
Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ SAŠA REGIJE

RAZISKOVALNA NALOGA

**OPTIMIZACIJA PROMETNEGA TOKA S  
POMOČJO RAČUNALNIŠKE SIMULACIJE**

Tematsko področje: Promet

Avtor:

Timotej Kompare

Mentor:

Aleš Spital

Velenje, 2024

Raziskovalna naloga je bila opravljena na Šolskem centru Velenje, Elektro in računalniški šoli.

Mentor: Aleš Spital

Datum predstavitve: marec 2024

## **KLJUČNA DOKUMENTACIJSKA INFORMACIJA**

ŠD Elektro računalniška šola Velenje, 2023/2024

KG promet / analiza / optimizacija / krožno križišče

AV KOMPARE, Timotej

SA SPITAL, Aleš

KZ 3320 Velenje, SLO, Trg mladosti 3

ZA Elektro računalniška šola Velenje

LI 2024

## **IN OPTIMIZACIJA PROMETNEGA TOKA S POMOČJO RAČUNALNIŠKE SIMULACIJE**

TD Raziskovalna naloga

IJ sl

JI sl / en

AI Moja raziskovalna naloga sistematično obravnava kompleksni problem optimizacije prometnega toka z izčrpnim pregledom naprednih metod računalniške simulacije na konkretnem primeru krožnega križišča iz lokalnega okolja. Osredotoča se na analizo vpliva odločitev voznikov na dinamiko prometnega toka, pri čemer se osredotoča na uporabo konceptov teorije iger in modeliranja vedenja udeležencev v prometu. Poleg tega poglobljeno preučuje paradokse in kontraproduktivne ukrepe prometne politike, pri čemer se izvaja iskanje najbolj optimalnih rešitev z uporabo matematičnih konceptov in računalniških algoritmov. Cilj raziskave je razviti inovativne rešitve, ki temeljijo na celovitem razumevanju udeležencev v prometu in prometno infrastrukturo. Z uporabo teh je namen predvsem izboljšati učinkovitost prometnih sistemov, kar posledično omogoči boljše načrtovanje in upravljanje prometnih omrežij v prihodnosti, hkrati pa prispeva k ustvarjanju bolj trajnostnega, varnega in učinkovitega prometnega okolja.

## **KEY WORDS DOCUMENTATION**

ND Elektro in računalniška šola Velenje, 2023/2024

CX traffic / analysis / optimization / roundabout

AU KOMPARE, Timotej

AA SPITAL, Aleš

PP 3320 Velenje, SLO, Trg mladosti 3

PB Elektro in računalniška šola Velenje

PY 2024

## **TI OPTIMIZATION OF TRAFFIC FLOW USING COMPUTER SIMULATION**

DT Research work

LA SL

AL sl/en

AB My research assignment systematically deals with the complex problem of traffic flow optimization with a comprehensive review of advanced computer simulation methods on the concrete example of a roundabout from the local environment. It focuses on the analysis of the impact of driver decisions on the dynamics of traffic flow, focusing on the application of game theory concepts and modeling the behavior of road users. In addition, it deeply examines the paradoxes and counterproductive measures of transport policy, where the search for the best optimal solution is carried out using mathematical concepts and computer algorithms. The goal of the research is to develop innovative solutions based on a comprehensive understanding of traffic participants and the transport infrastructure. By using them, the main purpose is to increase the efficiency of transport systems, which later enables better planning in the management of transport networks in the future, and at the same time contributes to the creation of a more sustainable, safe and efficient transport environment.

## KAZALO VSEBINE

<b>1</b>	<b>UVOD</b> .....	<b>1</b>
<b>2</b>	<b>PREGLED VSEBIN</b> .....	<b>2</b>
2.1	RAZVOJ PROMETA .....	2
2.2	PROBLEMATIKA PROMETNIH ZASTOJEV .....	3
2.2.1	NAČINI ZA ZMANJŠANJE PROMETNIH ZASTOJEV .....	3
<b>3</b>	<b>BRAESSOV PARADOKS</b> .....	<b>5</b>
3.1	PRIMERI IZ REALNEGA ŽIVLJENJA .....	5
3.1.1	ODSTRANITEV ZVEZDNEGA IGRALCA IZ KOŠARKAŠKE EKIPE .....	5
3.1.2	PROJEKT OBNOVE AVTOCESTE V SEOULU .....	6
3.1.3	ZAPRTJE 42ND STREET NA MANHATTNU .....	7
3.2	ZAKAJ PRIHAJA DO TEGA POJAVA? .....	7
3.3	GAME THEORY .....	8
3.3.1	ZAPORNIŠKA DILEMA .....	9
3.3.2	NASH EQUILLIBRIUM .....	10
3.4	SIMULACIJA IN ANALIZA PARADOKSA .....	11
3.5	POTEK IZDELAVE SIMULACIJE .....	18
3.5.1	USTVARJANJE OMREŽJA S POVEZOVANJEM VOZLIŠČ .....	19
3.5.2	RAČUNANJE NAJKRAJŠE POTI MED VOZLIŠČI .....	19
3.5.3	VIZUALIZACIJA OMREŽJA .....	22
<b>4</b>	<b>VPLIVI NA PRETOČNOST PROMETA</b> .....	<b>24</b>
4.1	POJAV NAMIŠLJENE PROMETNE ZAMAŠITVE .....	24
4.1.1	ZAKAJ PRIDE DO NAVIDEZNE ZAMAŠITVE? .....	24
4.1.2	POENOSTAVLJENA RAZLIČICA TAKŠNEGA POJAVA .....	25
4.2	EFEKT SHOCKWAVE .....	25
4.2.1	REŠITVE, KI PRIPOMOREJO K OPTIMIZACIJI .....	26

4.2.2	ANALIZA EFEKTA .....	26
4.2.3	IZRAČUN PRETOČNOSTI SEMAFORIZIRANEGA KRIŽIŠČA .....	27
4.2.4	PRETOČNOST AVTOMATIZIRANEGA PROMETA.....	28
<b>5</b>	<b>ANALIZA PRETOČNOSTI DVOPASOVNEGA KROŽNEGA KRIŽIŠČA V LOKALNEM OKOLJU .....</b>	<b>30</b>
5.1	RAZVOJ NAJVEČJEGA KROŽNEGA KRIŽIŠČA V ŠALEŠKI DOLINI..	30
5.2	TEHNIČNA ANALIZA KROŽNEGA KRIŽIŠČA.....	32
5.2.1	KRAK 1 – CELJSKA CESTA .....	32
5.2.2	KRAK 2 – ŠALEŠKA CESTA .....	33
5.2.3	KRAK 3 – CESTA FRANTIŠKA FOITA .....	34
5.2.4	KRAK 4 – ŽAROVA CESTA .....	34
5.3	MERJENJE GOSTOTE IN OPAZOVANJE PROMETA .....	35
5.4	IZDELAVA ANKETE NA TEMO PRETOČNOSTI IN ZADOVOLJSTVA S KROŽNIM KRIŽIŠČEM.....	36
5.4.1	VKLJUČEVANJE V KROŽNO KRIŽIŠČE NA PRIMERU DANE SITUACIJE .....	37
5.5	PREUČEVANJE VPLIVOV NA PRETOČNOST PROMETA V DANEM OMREŽJU.....	38
5.5.1	SLEDI BRAESSEVEGA PARADOKSA .....	39
5.5.2	NENADNE ZAMAŠITVE PROMETNEGA TOKA .....	40
5.5.3	POJAV EFEKTA SHOCKWAVE.....	42
5.6	TEORETIČNE REŠITVE ZA POVEČANJE PRETOČNOSTI KROŽIŠČA	43
5.6.1	REKONSTRUKCIJA KROŽIŠČA.....	43
<b>6</b>	<b>IZDELAVA SIMULACIJE PROMETNEGA OMREŽJA NA KONKRETNEM PRIMERU IZ LOKALNEGA OKOLJA .....</b>	<b>46</b>
6.1	NAČRTOVANJE IZDELAVE SIMULACIJE .....	46
6.2	IZBIRA ORODJA ZA IZDELAVO SIMULACIJE.....	47

6.2.1	KAJ JE UNITY IN ZAKAJ SE UPORABLJA?.....	47
6.3	3D IZDELAVA PROMETNEGA OMREŽJA.....	48
6.4	OSNOVNE MEHANIKE PROMETNIH UDELEŽENCEV.....	50
6.4.1	LINEARNO PREMIKANJE VOZILA OD TOČKE A DO TOČKE B..	50
6.4.2	INTERPOLACIJA TOČK NA KRIVULJI.....	52
6.4.3	BÉZIERJEVA KRIVULJA.....	53
6.4.4	PRETVARJANJE KRIVULJE V POT Z OGLIŠČI.....	63
6.4.5	RAČUNANJE POZICIJE NA DANI KRIVULJI.....	64
6.4.6	SAMODEJNO SLEDENJE VOZIL PO NAČRTOVANI POTI.....	66
6.4.7	SAMODEJNO ZAVIRANJE IN POSPEŠEVANJE VOZIL.....	68
6.5	USTVARJANJE OMREŽJA POTI.....	69
6.5.1	DODAJANJE VMESNIH SENZORJEV ZA VARNO VKLJUČITEV V PROMET.....	71
6.5.2	NADGRADNJA OMREŽJA Z UPORABO SEMAFORIZIRANE SIGNALIZACIJE.....	73
<b>7</b>	<b>REZULTATI.....</b>	<b>75</b>
7.1	MERITVE GOSTOTE PROMETA.....	75
7.2	ANALIZA ODGOVOROV ANKETE.....	77
7.2.1	POZNAVANJE CESTNO PROMETNIH PREDPISOV.....	77
7.2.2	ZBRANOST V PROMETU.....	79
7.2.3	PRETOČNOST PROMETA.....	83
7.3	SIMULACIJA.....	85
<b>8</b>	<b>RAZPRAVA.....</b>	<b>87</b>
<b>9</b>	<b>ZAKLJUČEK.....</b>	<b>89</b>
<b>10</b>	<b>ZAHVALA.....</b>	<b>90</b>
<b>11</b>	<b>VIRI IN LITERATURA.....</b>	<b>91</b>

## KAZALO SLIK

Slika 1: Primer zaporniške dileme [14] .....	9
Slika 2: Grafični prikaz enostavnega cestnega omrežja .....	11
Slika 3: Grafični prikaz prometnega omrežja skozi katerega pelje 400 vozil .....	12
Slika 4: Prikaz nadgradnje prometnega omrežja z dodatno povezavo .....	13
Slika 5: Prikaz najhitrejša poti pri 325 vozilih .....	14
Slika 6: Prikaz najhitrejša poti pri 400 vozilih .....	15
Slika 7: Grafični prikaz omrežja pri gostoti 450 vozil .....	16
Slika 8: Grafični prikaz omrežja brez dodatne povezave s 450 vozili .....	17
Slika 9: Prikaz preobremenitve omrežja brez dodatne povezave .....	18
Slika 10: Prikaz enostavnega omrežja za namene računanja najkrajše poti.....	20
Slika 11: Križišče v Velenju pred izgradnjo krožnega križišča (Foto: Milan Marič) ....	31
Slika 12: Dvopasovno krožno križišče v Velenju iz ptičje perspektive [18].....	31
Slika 13: Krak 1 – Celjska cesta [17] .....	32
Slika 14: Krak 2 – Šaleška cesta [17] .....	33
Slika 15: Krak 3 – Foitova cesta [17] .....	34
Slika 16: Krak 4 – Žarova cesta [17] .....	35
Slika 17: Prva dana situacija za vključitev v krožno križišče [23] .....	37
Slika 18: Druga dana situacija za vključitev v krožno križišče [23] .....	38
Slika 19: Nepravilno zapuščanje krožišča direktno iz notranjega pasu [20].....	41
Slika 20: Prikaz pravilne razvrstitve in signalizacije pri zapuščanju krožišča [20] .....	42
Slika 21: Načrt rekonstrukcije krožišča – turbo [19].....	44
Slika 22: Načrt rekonstrukcije krožišča – flower [19].....	45
Slika 23: Skica enostavnega krožnega križišča .....	49
Slika 24: 3D model krožnega križišča .....	49
Slika 25: Prikaz zelene interpolacije v obliki krivulje, in nelinearno.....	53
Slika 26: Primer cestnega omrežja [22].....	54
Slika 27: Prikaz linearne povezave dveh točk v koordinatnem sistemu.....	56
Slika 28: Tri linearno povezane točke .....	57
Slika 29: Prikaz kvadratne Bézierjeve krivulje .....	58
Slika 30: Štiri linearno povezane točke v zaporednem vrstnem redu.....	59
Slika 31: Prikaz dveh kvadratnih krivulj med štirimi točkami .....	60



Slika 32: Prikaz Bézierjeve krivulje tretje stopnje .....	61
Slika 33: Sestavljena Bézierjeva krivulja tretje stopnje .....	62
Slika 34: Sestavljena Bézierjeva krivulja označena znotraj Unity programskega okolja .....	62
Slika 35: Primerjava med zgoščenimi in ne zgoščenimi točkami na krivulji.....	63
Slika 36: Več izrisanih poti, ki sestavljajo prometno omrežje znotraj programa Unity.	69
Slika 37: Primer, kjer se avtomobil ne more vključiti v krožno križišče [23].....	72
Slika 38: Prikaz dominacije določene relacije znotraj simulacije .....	74
Slika 39: Prikaz delovanja simulacije s semaforizirano signalizacijo .....	74

## **KAZALO ENAČB**

Enačba 1: Izračun razdalje glede na pospešek.....	27
Enačba 2: Linearna interpolacija med dvema točkama .....	51
Enačba 3: Linearna Bézierjeva krivulja.....	55
Enačba 4: Kvadratna Bézierjeva krivulja .....	56
Enačba 5: Bézierjeva krivulja tretje stopnje .....	58

## **KAZALO GRAFOV**

Graf 1: Podatki o številu vozil v ponedeljek med 14.00 in 15.00 .....	76
Graf 2: Gostota prometnega toka v različnih časovnih obdobjih .....	77
Graf 3: Prikaz rezultatov ocene poznavanja cestno prometnih predpisov.....	78
Graf 4: Prikaz rezultatov izbire prometnega pasu v krožišče na dani situaciji.....	79
Graf 5: Ocena zbranosti voznikov v prometu.....	80
Graf 6: Pogosto pogovarjanje voznikov s sopotniki.....	81
Graf 7: Pripravljenost voznikov na speljevanje pri rdeči luči .....	82
Graf 8: Število povzročiteljev prometnih nesreč .....	83
Graf 9: Rezultati izbire prometnega pasu iz druge dane situacije .....	84
Graf 10: Rezultati mnenja glede pretočnosti Velenjskega krožnega križišča .....	85
Graf 11: Povprečna čakalna doba vozil za vključitev v krožišče .....	86

## **KAZALO KODE**

Koda 1: Funkcija algoritma dijkstra .....	22
Koda 2: Prikaz uporabe funkcije Lerp() na osnovnem primeru .....	52
Koda 3: Funkcija za izračun točke na krivulji, glede na razdaljo.....	65
Koda 4: Funkcija za izračun točke glede na odstotek celotne poti.....	66
Koda 5: Funkcija za računanje rotacijske smeri na krivulji .....	67
Koda 6: Spreminjanje pozicije in rotacije objekta na krivulji.....	68
Koda 7: Sprožilec za preverjanje križanja poti.....	70
Koda 8: Funkcija za izračun najbližje točke na poti glede na pozicijo objekta.....	71

## 1 UVOD

V raziskovalni nalogi sem se odločil analizirati različne pojave, ki vplivajo na promet in potovalni čas posameznikov in prometno omrežje kot celoto. Raziskoval sem vpliv ljudi in človeškega faktorja na pretočnost prometa. Namreč poznavanje teh razlogov je ključno za razvoj boljših in učinkovitejših prometnih sistemov in oblikovanje strategij za optimizacijo prometne infrastrukture.

Glavni namen raziskovanja je bila optimizacija prometnega toka na cestnem omrežju, specifično na primeru lokalnega dvopasovnega krožnega križišča v Velenju, ki je znano po nizki pretočnosti v prometni končnici.

### HIPOTEZE:

1. Širjenje prometnih povezav poveča pretočnost prometa in posledično zmanjša potovalni čas od točke A do B.
2. Optimizacija pretočnosti lokalnega prometnega krožnega križišča je možna brez rekonstrukcije.
3. Ustvarimo lahko računalniško simulacijo poljubnega prometnega omrežja, ki omogoči prikaz optimiziranega prometnega toka brez človeškega vpliva.

## **2 PREGLED VSEBIN**

Promet predstavlja ključni vidik vsakdanjega življenja sodobne družbe, ki je kljub svojim prednostim pogosto zaznamovan s težavami in izzivi. Vsakodnevno se soočamo z različnimi vidiki prometnih težav, med katerimi so zastoji ena izmed najbolj običajnih in motečih pojavov. Ti ne le vplivajo na naše časovno načrtovanje, temveč imajo tudi ekonomske, okoljske in socialne posledice.

### ***2.1 RAZVOJ PROMETA***

Promet je bil ključni dejavnik človeškega razvoja skozi celotno zgodovino človeštva. Od prvih preprostih prevoznih sredstev, kot so bili konji in vozovi, do današnjih visokotehnoloških avtomobilov in pametnih prometnih sistemov, je promet doživel izjemno razvojno pot, ki je temeljito spremenila način, kako se premikamo, komuniciramo in živimo. [1]

V zadnjih desetletjih smo priča hitremu razvoju tehnologije, ki spreminja način, kako se premikamo in upravljamo s prometom. Pametni prometni sistemi, povezana vozila, avtonomna vozila in aplikacije za deljenje prevoza so samo nekatere inovacije, ki oblikujejo prihodnost prometa. Pametni prometni sistemi uporabljajo napredne tehnologije, kot so senzorji, kamere, umetna inteligenca, za spremljanje prometa, upravljanje prometnih tokov, zmanjšanje zastojev in izboljšanje varnosti na cestah. Povezana vozila omogočajo komunikacijo med vozili in infrastrukturo, kar omogoča boljše upravljanje prometa in preprečevanje nesreč. Avtonomna vozila pa obetajo popolnoma novo dobo mobilnosti, kjer bodo vozila sposobna voziti samodejno brez človeškega posredovanja. [1]

## **2.2 PROBLEMATIKA PROMETNIH ZASTOJEV**

Prometni zastoji, ki jih običajno označujemo kot "traffic jams" ali preprosto "zamaški", predstavljajo enega izmed najbolj običajnih, motečih in razširjenih problemov sodobnega urbanega življenja. Čeprav so se pojavili kot posledica povečanega števila vozil v prometu, so njihove posledice precej kompleksne in večplastne. [1] [11]

Prvi in najopaznejši učinek prometnih zastojev je izguba časa. Vsakodnevno se na milijone ljudi po vsem svetu znajde ujetih v prometnih konicah, kar povzroča zamudo pri njihovih dnevnih opravkih, v službi, šoli in pri drugih pomembnih dejavnostih. Ta izguba časa ni le moteča, temveč ima tudi resne ekonomske posledice. Po podatkih raziskav porabijo vozniki v nekaterih mestih do 100 ur letno v zastojih, kar povzroči milijarde izgubljenih dolarjev zaradi zamud pri delu in drugih poslovnih dejavnostih. [1] [11]

Poleg časovne izgube prometni zastoji povzročajo tudi povečano onesnaževanje okolja. Vozila, ujeta v zastoj, porabijo več goriva in oddajajo več toplogrednih plinov, kar prispeva k poslabšanju kakovosti zraka in podnebnim spremembam. Poleg tega prometni zastoji povečujejo tudi tveganje za prometne nesreče, saj lahko dolgotrajno čakanje in nepredvidljivo premikanje vozil povzročita frustracijo in povečata stopnjo nevarnosti na cesti.

Zaradi teh in drugih razlogov je zmanjšanje prometnih zastojev postalo ključna prioriteta za mestne oblasti, urbaniste, prometne inženirje ter druge strokovnjake. Obstaja več pristopov k reševanju te problematike, vključno z izboljšanjem prometne infrastrukture, uvedbo pametnih prometnih sistemov, spodbujanjem trajnostnih načinov prevoza, kot sta kolesarjenje in uporaba javnega prevoza ter spremembo vedenja voznikov. [1]

### **2.2.1 NAČINI ZA ZMANJŠANJE PROMETNIH ZASTOJEV**

Eden izmed učinkovitih načinov za zmanjšanje prometnih zastojev je izboljšanje prometne infrastrukture. To vključuje gradnjo dodatnih prometnih pasov, razširitve cest in avtocest, izboljšanje križišč ter uvedbo naprednih prometnih signalizacij. Poleg tega

lahko uvedba pametnih prometnih sistemov, ki temeljijo na tehnologiji in podatkih, pomaga pri boljšem upravljanju prometa in optimizaciji pretoka vozil na cestah. [1] [11]

Poleg infrastrukturnih ukrepov je pomembno tudi spodbujanje alternativnih načinov prevoza. Investicije v javni prevoz, kolesarske poti in pešpoti ter spodbujanje souporabe vozil lahko zmanjšajo odvisnost od individualnega avtomobilskega prevoza in posledično zmanjšajo obremenitev cest. [1]

Ključnega pomena je tudi ozaveščanje in izobraževanje voznikov o pomembnosti defenzivne vožnje, spoštovanja prometnih pravil ter sprejemanja alternativnih načinov prevoza. S spremembo vedenja voznikov lahko bistveno zmanjšamo število prometnih nesreč ter izboljšamo pretok prometa na cestah. [1] [11]

### **3 BRAESSOV PARADOKS**

Braessov paradoks je pojav, ki opisuje nenavaden pojav v teoriji omrežij, kjer dodajanje nove poti v omrežje lahko poslabša celotno učinkovitost omrežja. Poimenovan je po nemškem matematiku Dietru Braessu, ki je leta 1968 kot prvi raziskoval ta pojav. [3]

V osnovi gre za to, da lahko dodajanje nove poti med dvema točkama v omrežju poveča celotno količino prometa in s tem poveča čas potovanja za vse udeležence. To se zgodi zaradi vpliva različnih faktorjev na potek prometa, kot so čas potovanja, prometni tokovi in interakcije med udeleženci. [3] [4]

Tipičen primer bi bil, če bi v mestu obstajale tri poti med dvema točkama, pri čemer bi bila ena cesta zelo zasedena. Na prvi pogled bi se zdelo smiselno dodati četrto pot, da bi razbremenili zasedenost na preostalih poteh; vendar pa se lahko zgodi, da bi dodana pot spodbudila več prometa, kar bi na koncu povečalo zastoje in čas potovanja na vseh poteh. [3] [4]

#### ***3.1 PRIMERI IZ REALNEGA ŽIVLJENJA***

Pojav tega paradoksa lahko zasledimo tudi v drugih panogah, ne le prometu. Primer letih so recimo električna omrežja, komunikacijska omrežja, družbene mreže, ekipni šport, kot je denimo košarka, itd.

##### **3.1.1 ODSTRANITEV ZVEZDNEGA IGRALCA IZ KOŠARKAŠKE EKIPE**

Primer Braessovega paradoksa na področju košarke je moč zaslediti, ko iz ekipe odstranimo zvezdnega igralca. Ta igralec je običajno v ekipi najboljši in posledično zadene največ točk. Zaradi tega se žoga večinoma pretaka preko njega. To si lahko predstavljamo tako, da je koš končna destinacija, igralci pa so vmesne postaje. Relacije

med njimi so podaje, ki jih morajo opraviti. Večina podaj je tako namenjenih temu specifičnemu igralcu, ki naposled spravi žogo v koš. Pojav tega paradoksa je tukaj očiten, saj ob odstranitvi takšnega igralca iz ekipe izgubimo ključno povezavo med omrežjem, v tem primeru igralci, in končno destinacijo, košem, zato je potrebno uporabiti druge povezave, v tem primeru je potrebno vključiti druge igralce in s podajami preko njih priti do zelenega cilja. To sistem razbremeni in posledično porazdeli pretočnost podaj v igri košarke, saj pride do več možnosti pretoka žoge z igrišča do koša.

V tej raziskovalni nalogi se bom posvetil predvsem pojavom tega paradoksa na področju prometa. Teh smo lahko skozi zgodovino zabeležili kar nekaj, vendar sem našel nekaj primerov, kjer je bil prav ta pojav najbolj izrazit.

### **3.1.2 PROJEKT OBNOVE AVTOCESTE V SEOULU**

Cheonggyecheon Restoration Project v Seulu, Južna Koreja, je bil prelomen projekt, ki je spremenil urbano krajino mesta. Projekt je vključeval odstranitev avtocestne povezave, imenovane Cheonggyecheon highway, njeno preoblikovanje v rečni tok in javno rekreacijsko območje. Cilj projekta je bil narediti mesto bolj ekološko in trajnostno ter izboljšati kakovost življenja prebivalcev. [8] [9]

Seul se je soočal s pomembnimi prometnimi težavami, ki so bile posledica hitre rasti prebivalstva, urbanega razvoja, gospodarske rasti in povečane uporabe avtomobilov. Cheonggyecheon highway je bila ena od glavnih prometnih žil mesta, ki je nosila velik del prometa, zato je bilo veliko strahu glede nadomestitve te ključne ceste z javnim vodnim tokom, saj se je domnevalo, da bo to prometne težave v mestu še poslabšalo. [8] [9]

Vendar pa je bil rezultat projekta presenetljiv. Prometni tok se je v regiji dejansko pospešil, s čimer se je zmanjšal čas potovanja. To lahko razložimo s pomočjo Braessovega paradoksa, ki temelji na motivih voznikov. Zaradi zaprtja avtoceste je bilo potrebno promet razpršiti na več drugih cest, kar je posledično zmanjšalo skupni čas potovanja. [8] [9]



Prav ta projekt je tako postal odličen primer Braessovega paradoksa v praksi, saj je pokazal, kako lahko odstranitev ključne prometne poti dejansko izboljša prometno situacijo v mestu, namesto da bi jo poslabšalo. To je jasen primer, kako lahko premišljena urbanistična strategija in infrastrukturni projekti prispevajo k boljši kakovosti življenja in trajnostnemu razvoju mest. [3] [8] [9]

### **3.1.3 ZAPRTJE 42ND STREET NA MANHATTNU**

V letu 1990 je New York City izvedel eksperimentalno ukrepanje z začasnim zaprtjem 42nd Street na Manhattnu kot del prizadevanj za obnovo mesta. Ta poteza je želela preoblikovati preobremenjeno prometno arterijo v pešcem prijazno območje, s čimer bi izboljšali urbanokrajinsko izkušnjo prebivalcev in obiskovalcev. Kljub začetnim pozitivnim pričakovanjem, pa je zaprtje 42nd Street nevede izpostavilo paradoksalne posledice, podobne Braessovemu paradoksu. Čeprav je bil cilj zmanjšanje prometnega zastoja, so spremembe v prometnih vzorcih povzročile povečane zastoje na okoliških cestah, kar je prometne razmere paradoksalno poslabšalo. [1] [2] [3]

Zaprtje ceste 42nd Street je razkrilo kompleksnost urbanega prometnega sistema in pokazalo, kako lahko navidezno pozitivne spremembe v infrastrukturi vodijo v nepričakovane rezultate. To je pomemben nauk za urbano načrtovanje, ki poudarja potrebo po celovitem premisleku in analizi pred vsako spremembo v prometni infrastrukturi. Kljub dobrim nameram lahko nepremišljeno ukrepanje povzroči večje težave kot koristi, kar nas opominja, da je treba pri načrtovanju upoštevati kompleksnost prometnih sistemov in njihovih interakcij z okoljem. [2]

## **3.2 ZAKAJ PRIHAJA DO TEGA POJAVA?**

Razlog za nastanek tega pojava so predvsem vozniki, ki sebično izbirajo pot, po kateri se bodo peljali, vendar pri tem pogosto ne izberejo najboljše možne poti za celotno omrežje, ampak izberejo tisto, ki njim tisti trenutek najbolj ustreza. To pomeni, da lahko dodatna

cesta ali zmogljivost privabi več prometa, saj se vozniki odločijo za to alternativo, namesto da bi ostali na svojih prvotnih poteh, kar posledično poslabša učinkovitost celotne infrastrukture. [3] [4]

Posledično se promet porazdeli neenakomerno, kar lahko povzroči zastoje in poslabšanje celotnega prometnega omrežja. Kljub temu da bi posamezni vozniki lahko izboljšali svoj čas potovanja z izbiro preostalih poti, to ni mogoče, saj bi to morali storiti prav vsi vozniki, ki se tisti trenutek nahajajo v prometnem omrežju; pa čeprav to za nekatere posameznike pomeni nekaj minut daljšo vožnjo. [3] [4]

### ***3.3 GAME THEORY***

Teorija iger proučuje strateške interakcije med racionalnimi igralci. Osredotoča se na analizo odločitvenih procesov, kjer vsak igralec izbira strategijo, ki naj bi mu prinesla najboljše možne rezultate, ob upoštevanju strategij drugih igralcev. Glavni cilj teorije iger je razumevanje ravnotežij in rezultatov v različnih scenarijih, kar omogoča napovedovanje izidov in oblikovanje učinkovitih strategij v interaktivnih okoljih. [5]



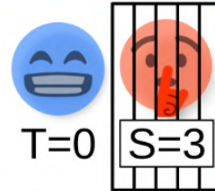

V kontekstu teorije iger lahko Braessov paradoks razumemo kot primer situacije, kjer se vozniki v prometnem omrežju znajdejo v večigralski igri, kjer vsak posamezni igralec, v tem primeru voznik, izbira svojo pot glede na svoj osebni cilj in interes. Vsak voznik si želi doseči svoj cilj, ki je običajno čim hitrejši prihod na ciljno destinacijo. [3]

Toda vsak voznik se odloča le na podlagi svojih trenutnih informacij in osebnih preferenc, ob tem pa nima vedno vpogleda v celotno prometno omrežje. Namesto da bi izbral najboljšo pot za celotno omrežje, sebično izbere pot, ki mu v določenem trenutku najbolj ustreza, kar lahko vodi v neracionalno porazdelitev prometa in poslabšanje celotne učinkovitosti prometnega omrežja. [5] [12]

### 3.3.1 ZAPORNIŠKA DILEMA

Zaporniška dilema (ang. Prisoner's dilemma) je miselni eksperiment teorije iger, kjer dva igralca, ki ne moreta komunicirati med seboj, izbirata med sodelovanjem ali izdajo. Glavna značilnost tega pojava je, da je optimalna strategija vsakega posameznika odvisna od odločitve drugega igralca. [5] [6] [7]

Koncept teorije zaporniške dileme poteka takole. Dva člana kriminalne združbe sta aretirana in zaprta. Vsak od njiju je v osami, brez možnosti komuniciranja z drugim. Policija prizna, da nima dovolj dokazov, da bi ju obtožili zločina, vendar ima zadosti dokazov, da lahko oba zapre za eno leto. Hkrati policija vsakemu zaporniku ponudi možnost, da lahko izda svojega partnerja in bo v tem primeru oproščen krivde, medtem ko bo partner dobil tri leta zapora. V primeru, da oba zapornika izdata drug drugega, bosta oba obsojena na dve leti zapora. Ob tem predpostavimo, da oba zapornika razumeta pravila igre, vsak izmed njiju cilja na najboljši lastni izkupiček ter hkrati ne veda, katero možnost je izbral drugi. [6] [7]

		B	
		B ostane tiho	B priča proti A
A	A ostane tiho	 R=1 R=1	 S=3 T=0
	A priča proti B	 T=0 S=3	 P=2 P=2

Slika 1: Primer zaporniške dileme [14]

Iz analize njunih najboljših možnosti, lahko ugotovimo, da je najboljša izbira za vsakega od njiju izdaja. Vsak izmed njiju ima torej dve možnosti, ostati tiho ali pa izdati drugega. Torej, če se oseba A odloči, da ostane tiho, bi morala oseba B izdati drugega za najboljši lastni izkupiček, namreč ostati svoboden je bolje, kot pa iti v zapor za eno leto. Vendar tudi v primeru, če se oseba A odloči za izdajo, mora storiti enako tudi oseba B, namreč dve leti zaporne kazni sta boljši kot pa tri. Iz tega torej ugotovimo, da je najboljša strategija za osebo B, da se odloči za izdajo, saj to predstavlja najboljši izkupiček, ne glede na to, kaj izbere oseba A. Prav tako enako velja za osebo A pri izbiri njegove strategije.

To vodi do ugotovitve, da kar je najboljše za posameznika, ni nujno, da je najboljše za družbo kot celoto. V našem primeru bi torej najboljši skupni izkupiček bil, da oba ostaneta tiho ter s tem oba prejmeta zaporno kazen enega leta, kar skupno znese dve leti zaporne kazni. Situacija, kjer oba izbereta izdajo, pravzaprav predstavlja najslabši možni izkupiček, saj je skupna zaporna kazen štiri leta, več kot katerikoli drug izid.

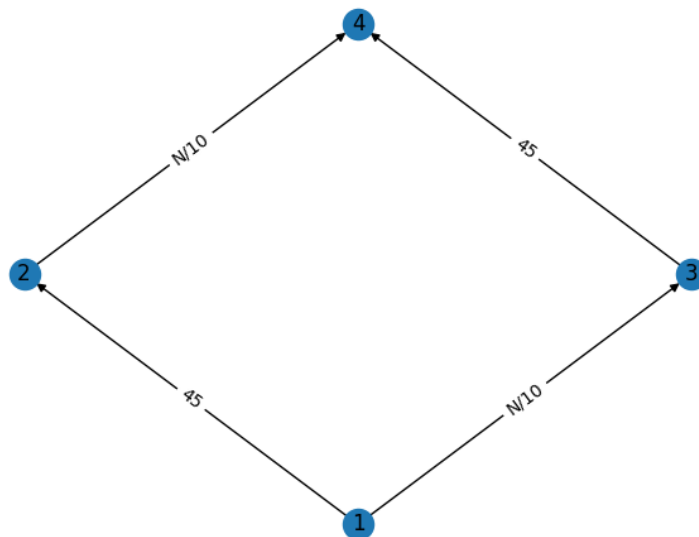
### 3.3.2 NASH EQUILLIBRIUM

Nashovo ravnotežje (ang. Nash Equilibrium) je koncept, ki opisuje situacijo, v kateri noben igralec ne more izboljšati svojega izida z menjavo svoje strategije, če pozna izbrano strategijo drugega igralca. Ta pojav nastane takrat, ko so vsi igralci v »ravnovesju« in nihče izmed njih ne želi spremeniti svoje odločitve, saj zaradi tega ne bi prejel višje nagrade ali pa bi prav zaradi tega prejel celo manjšo nagrado. [12] [13]

V zgornjem primeru je torej strategija za najboljši posameznikov izkupiček izdaja. Kar pomeni, izdaja obeh zapornikov predstavlja Nashovo ravnotežje, pri čemer nihče izmed zapornikov nima razloga, da bi menjal svojo strategijo, saj bi lahko ostal brez zaporne kazni ali pa bi zaporno kazen povečal za eno leto. Ta specifični pojav imenujemo močno Nashovo ravnotežje (ang. Strong Nash Equilibrium), kjer menjava strategije prinese slabši rezultat, ne glede na izbiro nasprotnikove strategije. [12] [13]

### 3.4 SIMULACIJA IN ANALIZA PARADOKSA

Za lažjo predstavo in razumevanje bomo ustvarili poenostavljeno prometno omrežje, kjer se bodo naši vozniki skušali pripeljati od točke A do točke B. Vendar povezava med tema dvema točkama žal ne bo direktna. Točki bosta torej med sabo povezovali dve cestni relaciji iz dveh delov, pri čemer bo en del avtocestna relacija, kjer bo pretočnost prometa ogromna. Za lažjo predstavo in računanje bomo predpostavili, da bo pretočnost neomejena, kar pomeni, da gostota prometa ne bo imela vpliva na sam pretok prometa, medtem ko bo pretok druge cestne relacije odvisen od gostote prometa tisti trenutek na tej relaciji.

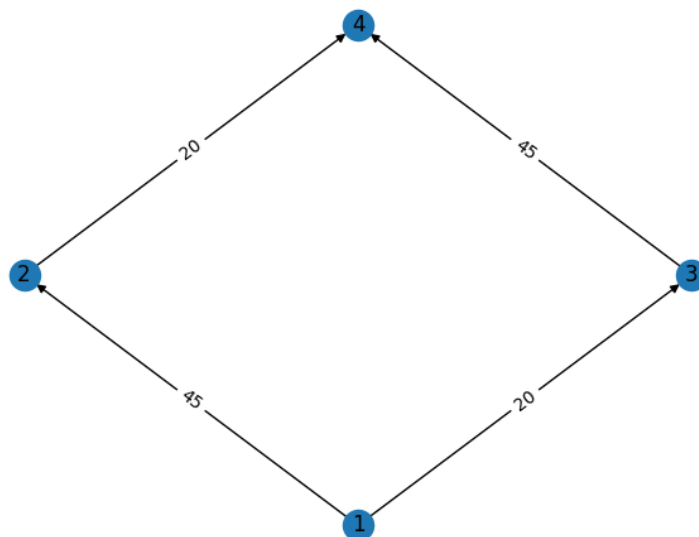


Slika 2: Grafični prikaz enostavnega cestnega omrežja

Naše cestno omrežje je torej sestavljeno iz dveh statičnih delov, (1–2 in 3–4), kjer je čas potovanja neodvisen od gostote prometa in bo vedno 45 minut, in dveh dinamičnih delov (1–3 in 2–4), kjer bo čas odvisen od gostote prometa in bo v našem primeru  $N/10$  minut, pri čemer  $N$  predstavlja število vozil v tistem trenutku na tej relaciji, kar pomeni, da v primeru, ko bo na tej relaciji istočasno deset vozil, bo desetemu vozilu v vrsti ta relacija prištela dodatno minuto potovalnega časa.

Sedaj lahko simuliramo vedenje voznikov v tem prometnem omrežju. Recimo, da skozi simulacijo naenkrat pošljemo 400 voznikov enega za drugim. Vsi želijo prispeti od točke »1« do točke »4«. Možnosti sta tako v tem primeru le dve, ali se bodo vozniki odločili, da bodo najprej vozili po avtocestni relaciji in šele nato po relaciji, odvisni od gostote prometa, ali obratno. Pri tem predpostavimo, da so vsi udeleženci prometa seznanjeni z razmerami na cestah in lahko vnaprej izračunajo svoj potovalni čas do želene destinacije. Recimo, da jim pri tem pomaga navigacija.

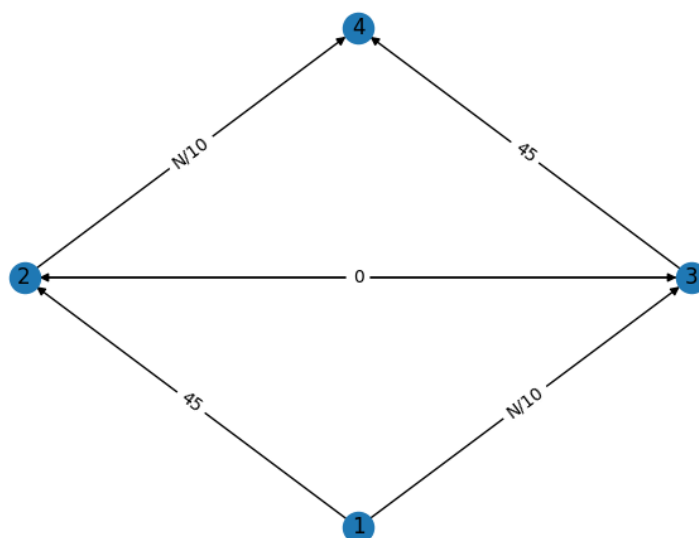
Prvi avtomobil v vrsti bo zaradi tega imel prosto izbiro poti, da bo prišel na želeno destinacijo, saj mu bosta obe poti vzeli vsega skupno 45,1 minute potovalnega časa. Drugi voznik bo tudi lahko izbiral med obema potema, vendar v primeru, da se bo odločil za isto pot, kot se je odločil voznik pred njim, bo njegov potovalni čas znašal 45,2 minute. Vendar, ker vsaka sekunda šteje, bo drugi voznik zato ubral drugo, hitrejšo, pot, ki mu bo vzela vsega skupaj 45,1 minute potovalnega časa. Tretji voznik bo tako imel spet prosto izbiro, saj izbira poti ne bo imela vpliva na njegov končni čas, v obeh primerih bo znašal 45,2 minute. To se bo ponavljalo, dokler se po našem prometnem omrežju ne bo porazdelilo vseh 400 vozil.



Slika 3: Grafični prikaz prometnega omrežja skozi katerega pelje 400 vozil

Če bo vsak voznik izbral, zase, najbolj optimalno pot, se bo gostota prometa porazdelila in s tem bo polovica vozil na relaciji 1–2–4, polovica pa na relaciji 1–3–4, kar pomeni 200 vozil na vsaki relaciji. Skupni potovalni čas zadnjega vozila v koloni bo torej znašal 65 minut.

Vendar se nam 65 minut zdi preveč, zato želimo naše prometno omrežje optimizirati, razširiti, tako da bi potovalni čas zmanjšali. Odkrijemo odličen način, kjer lahko dodamo novo relacijo 2–3, ki za potovanje vzame nič minut. Ker je to povezava, ki ne vzame dobesedno nič potovalnega časa, se nam zdi najboljši način za razširitev našega cestnega omrežja.

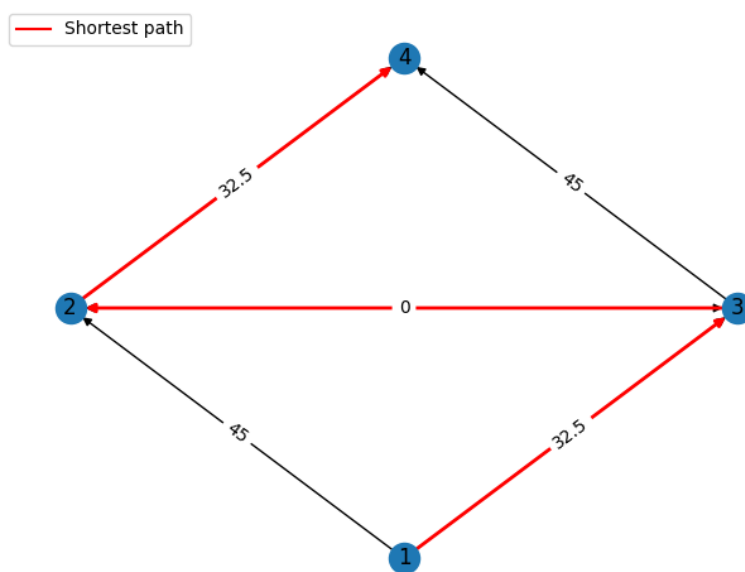


Slika 4: Prikaz nadgradnje prometnega omrežja z dodatno povezavo

Nato ponovno poženemo simulacijo našega prometnega omrežja s 400 vozili, ki jih enega za drugim pošljemo od destinacije »1« do destinacije »4« in ugotovimo, da najhitrejša pot znaša 80 minut. Kar je 15 minut več kakor pri omrežju brez dodatne relacije. Ampak, kako je to možno? Čudežna povezava bi morala potovalni čas zmanjšati in ne povečati.

Ta pojav, kjer širjenje prometnih omrežij z dodatnimi relacijami potovalnega časa ne zmanjša, temveč ga celo poveča, imenujemo Braessov paradoks. Do tega pojava pride zaradi človeške narave, kjer vsak posameznik gleda nase in lastni izkupiček. V novi

simulaciji z dodatno relacijo lahko tako izračunamo potovalni čas za vsako vozilo posebej. Predvidevamo, da bo vsak izmed voznikov izbral sebi najhitrejšo pot. V tem primeru, bo voznik ena izbral relacijo 1–3–2–4, saj bo zaradi izjemno nizke gostote prometa za to potreboval 0,2 minute potovalnega časa. Drugi voznik bo izbral isto pot, saj je tudi zanj najhitrejša pot z relacijami 1–3–2–4, za katero bo porabil 0,4 minute. Ta postopek se bo nadaljeval, saj bo ta relacija najbolj optimalna za vse voznike.

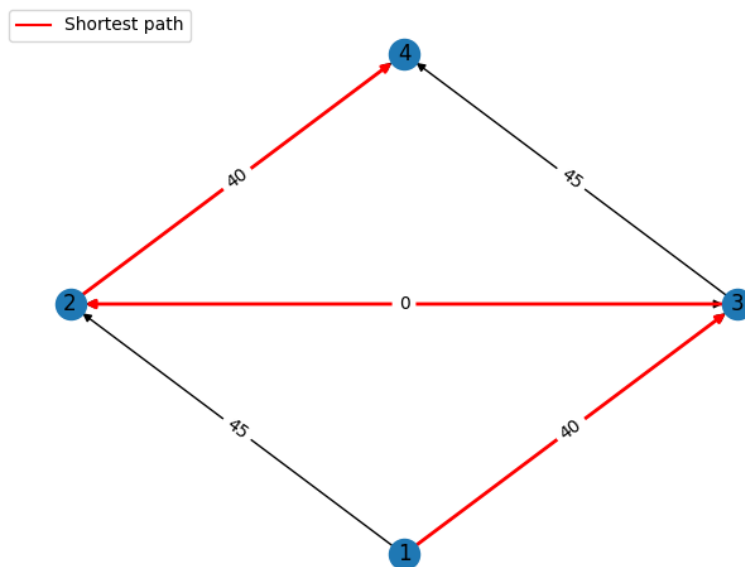


Slika 5: Prikaz najhitrejše poti pri 325 vozilih

Do tega, težko razumljivega pojava, pride pri 325. vozilu, ki izenači potovalni čas predhodnega cestnega omrežja, brez dodatne relacije 2–3. Namreč čas, ki ga vozilo potrebuje za relacijo 1–3 in 2–4 je 32,5 minute, zaradi prenatrpanosti relacije, katere čas je odvisen od gostote prometa  $N/10$ . Skupno to, z dodatno relacijo 2–3, znaša 65 minut potovalnega časa, kar je enako kot pri prometnem omrežju brez dodatne povezave. A vendar se čas, ki ga vozila potrebujejo za prihod na cilj, tukaj ne ustavi. Denimo, če v našo simulacijo pošljemo 326 vozil, bo skupni potovalni čas zadnjega v vrsti znašal 65,2 minute, kar je že 0,2 minute več kot potovalni čas zadnjega v vrsti pri prometnem omrežju brez dodatne relacije, s kar 74 vozili več.



Tukaj lahko zasledimo lastnosti zaporniške dileme, ki se pojavi med vozniki, kjer vsak izmed njih gleda le na lastno dobro in ne na dobrobit celotne družbe. V tem primeru so vsa vozila v Nashovem ravnotežju, pri čemer nihče izmed voznikov ne bo imel boljšega izkupička, četudi menja strategijo, v tem primeru pot, po kateri bo prišel do končne destinacije.

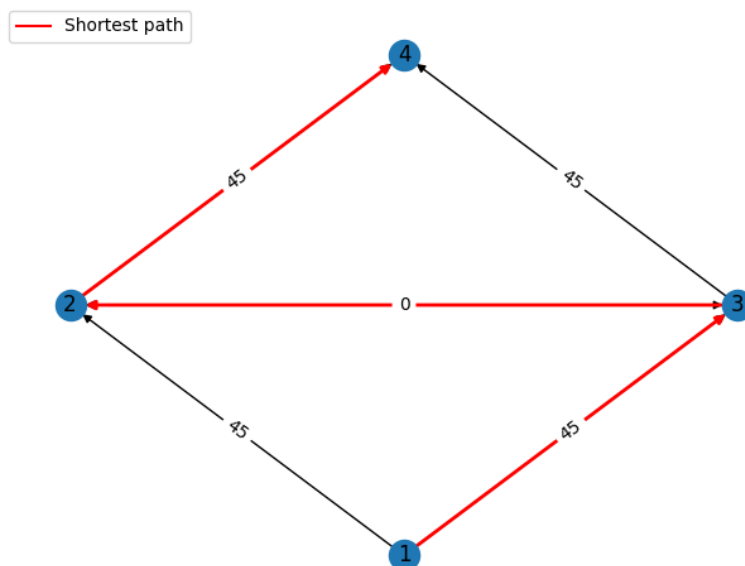


Slika 6: Prikaz najhitrejše poti pri 400 vozilih

Če se sedaj vrnemo na simulacijo s 400 sočasnimi vozili v prometnem omrežju z dodatno relacijo, ugotovimo, da so vsa vozila v Nashovem ravnotežju, saj nihče ne bo profitiral s spremembo izbrane poti, kajti s tem bi si povečal skupni potovalni čas. Recimo, da želi zadnji voznik v vrsti spremeniti potovalno pot, mu pri tem preostaneta še dve možnosti. Prva možnost sta relaciji 1–2–4 ali 1–3–4, za kateri bi potreboval skupno 85 minut, kar je še vedno 5 minut dlje kot za najhitrejšo možno pot in relacijo 1–2–3–4, ki skupno znaša 90 minut potovalnega časa in je kar 10 minut daljša kot najhitrejša pot. To imenujemo močno Nashovo ravnotežje, saj bi vsak, ki bi zamenjal svojo izbrano pot, svoj čas, ki ga porabi za opravljeno pot, podaljšal.

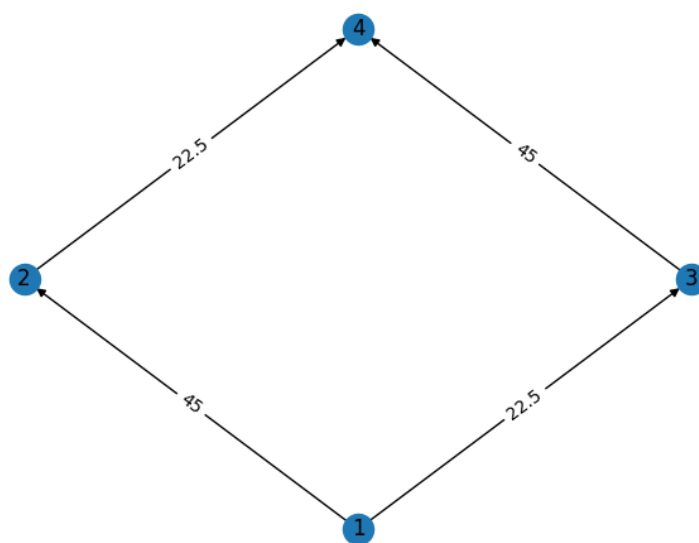
To nam pove, da bo vsak, ki bo svojo potovalno pot želel čimbolj skrajšati, izbral relacijo 1–3–2–4, ki je v tem primeru najhitrejša in znaša 80 minut skupnega potovalnega časa.

Četudi to pomeni, da je skupni potovalni čas še vedno za skoraj četrtino daljši od potovalnega časa v prometnem omrežju, z enakim številom vozil, brez dodatne relacije.



Slika 7: Grafični prikaz omrežja pri gostoti 450 vozil

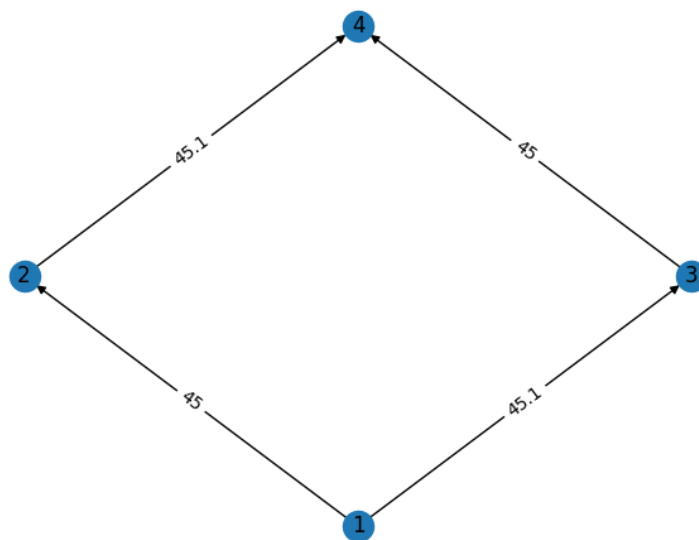
V »optimiziranem« prometnem omrežju dosežemo popolno ravnotežje pri sočasni prometni gostoti 450 vozil, kjer se potovalni čas ustali in doseže svoj maksimum, 90 minut. Torej, ko bosta relaciji 1–3 in 2–4, s svojo največjo pretočno zmogljivostjo, izenačili potovalni čas z relacijama 1–2 in 3–4, v tem primeru 45 minut, dodatno povečanje gostote prometnega toka ne bo imelo vpliva na končni potovalni čas, saj bodo vsi nadaljnji vozniki lahko ubrali pot 1–2–3–4, ki ni odvisna od gostote prometa, zato bo potovalni čas, na tej relaciji, vedno 90 minut.



**Slika 8: Grafični prikaz omrežja brez dodatne povezave s 450 vozili**

Na primeru simulacije prometnega omrežja brez dodatne relacije 2–3 s 450 sočasnimi vozili lahko opazimo, da je skupni potovalni čas zadnjega v vrsti 67,5 minute, kar je, v primerjavi z omrežjem z dodatno relacijo, še vedno 22,5 minute hitreje.

Če sedaj gostoto prometnega toka povečamo na 900 vozil, lahko izračunamo, da bo najhitrejši potovalni čas zadnjega vozila znašal 90 minut, kar se enači s potovalnim časom v prometnem omrežju z dodatno relacijo, a le pri pol manjši gostoti prometa. Kar pomeni, da je za gostoto vozil pri 900 ali manj v našem virtualnem omrežju optimalnejša infrastruktura omrežja brez dodatne relacije kot pa omrežja z njo, pri čemer je končni potovalni čas omrežja brez dodatne relacije, pri gostoti nad 900 vozil, vedno večji od 90 minut, saj bo pri tem omrežju le-ta vedno odvisen od gostote prometa in se bo vzporedno z gostoto vedno spreminjal.



**Slika 9: Prikaz preobremenitve omrežja brez dodatne povezave**

Potovalni čas pri prometnem omrežju brez dodatne relacije 2–3, z 902 sočasnimi vozili, znaša 90,1 minute, kar je nekaj sekund več od omrežja z dodatno relacijo, kjer je pri 902 vozilih potovalni čas še vedno 90 minut. Pri tem torej ugotovimo, da pri velikem povečanju gostote prometa dodatna relacija le izboljša sam potovalni čas voznikov.

### ***3.5 POTEK IZDELAVE SIMULACIJE***

Simulacijo sem zasnoval za namen lažjega računanja potovalnega časa posameznih vozil v našem virtualnem omrežju, pri večji gostoti samega omrežja in za lažjo grafično predstavo. Za izdelavo simulacije sem uporabil Python programski jezik, ki je namenjen predvsem hitremu in enostavnemu pisanju programske kode ter razvoju različnih vrst aplikacij, vključno z računalniškimi programi, spletnimi aplikacijami, znanstvenimi modeli, umetno inteligenco in avtomatizacijo nalog.

### 3.5.1 USTVARJANJE OMREŽJA S POVEZOVANJEM VOZLIŠČ

Za izdelavo svojega omrežja, za namen simulacije in vizualizacije Braessovega paradoksa, sem uporabil knjižnico NetworkX, ki je namenjena analizi, gradnji in vizualizaciji kompleksnih omrežij.

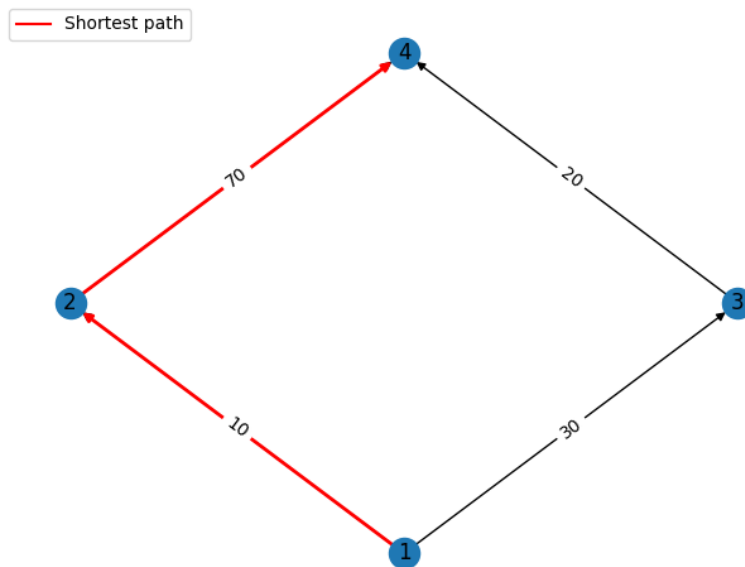
Omrežje sem zasnoval tako, da sem postavil štiri vozlišča, ki sem jih nato povezal z relacijami. Vsaki relaciji sem dodelil utež, ki je v mojem primeru pomenila potovalni čas, porabljen na tej relaciji. Uteži sem tako razdelil na statične in dinamične. Statične so bile vedno konstantne in se z gostoto prometa niso spreminjale, medtem ko so se dinamične spreminjale. Tako sem za vsakim vozilom, ki je prevozilo relacijo, odvisno od gostote prometa, posodobil uteži teh relacij.

### 3.5.2 RAČUNANJE NAJKRAJŠE POTI MED VOZLIŠČI

Načinov za izračun najkrajše poti od točke A do točke B v nekem omrežju je več. Ena izmed teh je možnost, pri kateri vozila nimajo vpogleda v celotno omrežje, vendar le na relacije, ki so povezane s trenutnim vozliščem, na katerem se nahajajo.

Pri tem program začne z najkrajšo dosegljivo relacijo in pri vozlišču preveri naslednje relacije, to počne dokler ne pride v slepo ulico in se vrača po vozliščih nazaj, dokler ne najde prvega niza relacij, ki vodijo do zelenega cilja. Pri tem seveda vedno izbere najkrajšo relacijo, ki je vidna z vozlišča, pri katerem se nahaja.

Prav tu sem se na začetku storil napako, saj sem izračun najhitrejše poti izvedel po principu sosednje relacije z najmanjšo utežjo, kar je mogoče na prvi pogled pravilen postopek, vendar ni primeren za računanje najhitrejše poti od točke A do točke B v celotnem omrežju.



Slika 10: Prikaz enostavnega omrežja za namene računanja najkrajše poti

Na tem konkretnem primeru omrežja z vozlišči in relacijami lahko vidimo, da sta od točke »1« do točke »4« dve različni poti. Od vozlišča »1« lahko našo pot začnemo po eni izmed dveh relacij. Ena meri 10 in druga 30 enot. Če računamo najkrajšo pot, in nimamo vpogleda v celotno omrežje ter s tem nadaljevanja naše poti do ciljne destinacije, bomo izbrali relacijo z utežjo 10, saj je to manj kot utež 30. Na vozlišču »2« nam preostane le ena možnost, ki v tem primeru vodi do naše končne destinacije. Ta utež je vredna 70 enot, kar pomeni, da pot, ki smo jo opravili, znaša skupno 80 enot, in ker smo prišli na cilj, se tukaj naše potovanje zaključuje.

Čeprav ta pot ni najbolj optimalna, ker na začetku izberemo relacijo z utežjo 30, ki na prvi pogled ni najkrajša, vendar nato nadaljujemo do končne destinacije na relaciji z utežjo 20 enot, kar znaša skupno 50 enot; kar je v tem primeru krajša pot kot tista, ki smo jo izračunali po tem principu.

Ker sem moral izračunati najbolj optimalne poti v celotnem omrežju, sem uporabil drugo metodo, in sicer algoritem Dijkstra. To sem storil zato, ker je ta algoritem posebej zasnovan za iskanje najkrajših poti v grafih z utežmi na povezavah, kar je ključno pri simulaciji prometnih omrežij, kjer se želi določiti najbolj optimalna pot glede na čas, razdaljo ali utež.

Postopek algoritma Dijkstra se začne z določitvijo začetnega vozlišča, od katerega se želi najti najkrajšo pot do vseh ostalih vozlišč. Začetno vozlišče ima začetno razdaljo 0, medtem ko so vse ostale razdalje začasno nastavljene na neskončno. Algoritem nato iterativno posodablja razdalje do sosednjih vozlišč, pri čemer upošteva uteži povezav med njimi. Za vsako sosednje vozlišče se izbere najkrajša pot, kar se nadaljuje, dokler ni obiskano vsako vozlišče v grafu ali dokler ni več dosegljivih vozlišč z neskončno razdaljo.

Eden od ključnih konceptov algoritma Dijkstra je ideja izbire najkrajše poti do vsakega vozlišča v vsakem koraku, ob upoštevanju že znanih najkrajših poti do drugih vozlišč. Ta pristop zagotavlja, da se najkrajše poti do vsakega vozlišča postopoma izboljšujejo, dokler ni dosežena končna rešitev.

Algoritem Dijkstra se uporablja v številnih aplikacijah v vsakdanjem življenju, kot so navigacijski sistemi, kjer omogoča iskanje najhitrejših poti med lokacijami, logistika, telekomunikacije in še bi lahko naštevali. Prav zaradi njegove preprostosti, učinkovitosti in uporabnosti ga uvrščajo med enega najpomembnejših algoritmov v računalništvu.

```
def dijkstra_algorithm(graph, start_node):  
  
    unvisited_nodes = list(graph.get_nodes())  
    shortest_path = {}  
    previous_nodes = {}  
  
    # Nastavili bomo vrednost neskončno vseh neobiskanih vozlišč  
    max_value = sys.maxsize  
    for node in unvisited_nodes:  
        shortest_path[node] = max_value  
  
    # Začetno vozlišče označimo z vrednostjo 0  
    shortest_path[start_node] = 0  
  
    # Algoritem se izvaja, dokler ne obiše vse vozlišč  
    while unvisited_nodes:  
  
        # Poišče vozlišče z najmanjšo vrednostjo  
        current_min_node = None  
  
        for node in unvisited_nodes:  
            if current_min_node == None:  
                current_min_node = node  
            elif shortest_path[node] < shortest_path[current_min_node]:  
                current_min_node = node  
  
        # Pridobi sosednja vozlišča  
        neighbors = graph.get_outgoing_edges(current_min_node)  
  
        for neighbor in neighbors:  
            tentative_value = shortest_path[current_min_node] +  
                graph.value(current_min_node, neighbor)  
  
            if tentative_value < shortest_path[neighbor]:  
                shortest_path[neighbor] = tentative_value  
                previous_nodes[neighbor] = current_min_node  
  
        # Označi vozlišče kot obiskano  
        unvisited_nodes.remove(current_min_node)  
  
    return previous_nodes, shortest_path
```

**Koda 1: Funkcija algoritma dijkstra**

### 3.5.3 VIZUALIZACIJA OMREŽJA

Za vizualizacijo te simulacije sem uporabil knjižnico Matplotlib, ki je ena izmed najbolj priljubljenih knjižnic v programskem jeziku Python za ustvarjanje statičnih, interaktivnih in animiranih vizualizacij podatkov.



Matplotlib omogoča ustvarjanje različnih vrst vizualizacij, vključno s črtastimi grafikoni, histogrami, tortnimi grafikoni, površinami, slikami, 3D-grafikoni in še veliko več. Poleg tega knjižnica omogoča tudi natančno prilagajanje vizualnih elementov, kot so barve, legende, osi, oznake in slogi, kar omogoča ustvarjanje vizualizacij po meri in prilagojenih potrebah projekta. Zaradi tega je bil prikaz omrežja, namenjenega simuliranju paradoksa, z uporabo te knjižnice enostaven.

## **4 VPLIVI NA PRETOČNOST PROMETA**

V kontekstu nenehnega razvoja prometnih sistemov in tehnologije je optimizacija pretočnosti prometa postala ključna naloga za izboljšanje učinkovitosti cestnega omrežja in zmanjšanje prometnih zastojev. Tukaj se bom posvetil raziskovanju in razumevanju pojavu nenadnih zastojev, pri čemer bom uporabil simulacije in izračune, da bi identificiral in analiziral dejavnike, ki vodijo do nastanka le-teh. S tem bom prispeval k boljšemu razumevanju dinamike prometnih tokov ter razvoju učinkovitih strategij za zmanjšanje prometnih zastojev in izboljšanju pretočnosti prometa.

### ***4.1 POJAV NAMIŠLJENE PROMETNE ZAMAŠITVE***

Med številnimi izzivi, ki jih prinaša promet, je eden izmed najpogostejših pojavov imenovan namišljena prometna zamašitev (ang. Phantom traffic jam effect). Ta pojav se pogosto pojavlja na avtocestah, kjer se zdi, da se prometni zastoj pojavlja brez očitnega razloga in se nato »raztopi«, ne da bi pri tem prišlo do kakršnih koli nesreč ali ovir na cesti. [11]

#### **4.1.1 ZAKAJ PRIDE DO NAVIDEZNE ZAMAŠITVE?**

Eden od ključnih dejavnikov, ki prispeva k nenadni zamašitvi prometnega toka, je reakcijski čas voznikov. Človeški možgani potrebujejo čas, da zaznajo spremembe v okolju in nanje odreagirajo. Če se na primer vozilo nenadoma ustavi pred vozilom za njim, mora voznik v njem začeti z zaviranjem. Lahko pa se zgodi, da vozniki ne zaznajo spremembe pravočasno ali pa njihova reakcija ni dovolj hitra, kar povzroči verižno reakcijo zaviranja. Tudi majhne zamude pri reakciji enega voznika lahko imajo kaskadne učinke na promet, kar vodi v širjenje zastoja. [11]

Poleg tega k temu pojavu prispevajo tudi napake pri vožnji, ki jih naredijo vozniki. Nepričakovano menjavanje pasov, nepravilno ocenjevanje hitrosti ali razdalje,

nepričakovano zaviranje ali pospeševanje – vse to so napake, ki lahko sprožijo verižno reakcijo in povzročijo zastoje na cestah.

#### **4.1.2 POENOSTAVLJENA RAZLIČICA TAKŠNEGA POJAVA**

Predpostavimo, da imamo primer prometa, ki poteka tekoče. Pri tem bomo zanemarili vse človeške in ostale faktorje, ki bi lahko vplivali na rezultate te analize. Na cesti imamo kolono stotih vozil, pri čemer se vsa gibljejo s hitrostjo 100 km/h. Razdalja med vsakim izmed njih je enaka povprečnemu človeškemu reakcijskemu času. Nenadno zaviranje prvega vozila v koloni zaradi zunanjega razloga, kot je recimo prečkanje divjadi čez cesto, povzroči verižno reakcijo. Zaradi tega vsako naslednje vozilo zavira malenkost bolj, ker se to razlikuje od človeškega instinkta, lahko določimo, da vsako vozilo zavira za 1 % bolj kot vozilo pred njim.

To pomeni, če imamo 100 vozil, vsako izmed njih se premika s 100 km/h in prvo vozilo v koloni zavira, pri čemer zmanjša trenutno hitrost za 50 %. To sproži reakcijo vozil za njim, pri čemer vozilo tik za njim zmanjša hitrost za 51 %, da slučajno ne bi prišlo do trka. V resničnem življenju je to zaviranje med vozili veliko večje in izrazitejše, vendar tukaj to predpostavimo zaradi lažje predstave.

V tem primeru bo po zaviranju prvega vozila za 50 % začetne hitrosti, vozilo, ki je v koloni na 51. mestu, svojo hitrost zmanjšalo za 100 %, kar je 0 km/h.

#### **4.2 EFEKT SHOCKWAVE**

Prehod iz rdeče na zeleno luč na semaforju je situacija, ki jo doživljamo vsak dan na cestah. Ta prehod sproži zaporedje dogodkov, ki lahko vplivajo na učinkovitost prometa. Eden izmed ključnih pojavov, ki se pojavlja pri tem prehodu, je tako imenovani efekt »shockwave«, kjer vozila pospešujejo z zamikom, kar na pretočnost prometa vpliva negativno. [11]

V našem primeru je naš proces prehoda na zeleno luč na semaforju kompleksen proces, ki vključuje več dejavnikov, kot so signalizacija semaforja, odzivnost voznikov in prometne razmere. Ko na semaforju zasveti zelena luč, lahko prvi voznik v vrsti začne z vožnjo, pri čemer mu sledijo ostala vozila. Pri tem prihaja do ključnega problema, saj prihaja med prehodi vozil iz mirujočega stanja v pospešeno enakomerno gibanje do zamika.

#### **4.2.1 REŠITVE, KI PRIPOMOREJO K OPTIMIZACIJI**

Časovnik za odštevanje pred prehodom na zeleno luč je eno izmed najučinkovitejših rešitev za povečanje pretočnosti prometa pri semaforjih. Ta tehnologija omogoča voznikom, da se bolje pripravijo na spremembo svetlobnega signala in začnejo pospeševati že pred prehodom semaforja na zeleno luč, kar zamik v pospeševanju vozil posledično zmanjša.

Poleg tega lahko časovnik za odštevanje voznikom zagotovi tudi večjo varnost in zmanjša stres v prometu. Z jasnim prikazom časa, ki je še preostal do spremembe svetlobnega signala, lahko vozniki bolje ocenijo trenutno situacijo in se ustrezno pripravijo na začetek vožnje. To lahko zmanjša tveganje za nenadne manevre ali nesreče ter prispeva k bolj tekočemu in varnemu prometnemu toku. Skupaj s tehnikami sinhronizacije semaforjev in naprednimi sistemi opozarjanja voznikov lahko časovnik za odštevanje pred prehodom na zeleno luč igra ključno vlogo pri izboljšanju pretočnosti prometa in ustvarjanju boljših prometnih razmer na cestah.

#### **4.2.2 ANALIZA EFEKTA**

Za analizo je predlagan algebraični model razvoja prometa znotraj urbanega odseka cest, ki vključuje modeliranje pojavov, ki jih povzročajo semaforji. Model omogoča rešitev prometne gostote in pretoka brez reševanja diferencialnih enačb, kar bistveno zmanjša računsko obremenitev. [10] [11]

Predstavljamo si mestni odsek ceste, kjer je na koncu semafor, ki ob zeleni luči omogoča prehod skozi križišče. Razumno je domnevati, da predstavljeni odsek ceste predstavlja osnovni segment prometne mreže, kar pomeni, da znotraj odseka ni dovoljenih zunanjih tokov. [10]

Pretočnost dane situacije lahko določimo s številom vozil  $N$ , ki bodo prečkala semaforizirano križišče v določenem časovnem intervalu  $T$ . Za lažji izračun bomo zanemarili dejstvo, da vozniki med seboj ne pospešujejo enakomerno, zato bomo določili pospešek  $a$ , pri čemer nato vozilo doseže maksimalno hitrost  $v$ . Gostota našega prometa bo v tem primeru  $p = 1$ , kar pomeni eno vozilo na en meter.

Sedaj lahko izračunamo število vozil, ki bodo prečkala križišče ob prižigu zelene luči, ob predpostavki povprečnega reakcijskega časa ene sekunde.

#### 4.2.3 IZRAČUN PRETOČNOSTI SEMAFORIZIRANEGA KRIŽIŠČA

Pri  $N$  številu vozil lahko tako izračunamo porabljeni čas za prečkanje vsakega posameznega vozila  $n$ , pri čemer bo čas, pri katerem bo začelo s pospeševanjem, enak vsoti reakcijskih časov vozil pred njim,  $n * 1$  sekunde. Razdalja, potrebna za prečkanje semaforiziranega križišča, pa bo enaka vsoti dolžin celic pred vozilom. V našem primeru bo to torej  $n * s$  [m], kjer je  $s = 1$  m.

Glede na podan pospešek in maksimalno omejeno hitrost, ki je po zakonu voznik ne sme prekoračiti, lahko sedaj izračunamo čas, ki ga bo voznik potreboval za to razdaljo. V primeru, da je skupni porabljeni čas manjši od časa  $t$ , ki nam pove, kako dolgo bo gorela zelena luč, jo bo potem ta avto lahko uspešno prevozil, v nasprotnem primeru bo moral počakati na naslednjo.

**Enačba 1: Izračun razdalje glede na pospešek**

$$D = v * t + \frac{a * t^2}{2}$$

Na podlagi teh informacij lahko izračunamo potovalni čas vsakega izmed vozil v semaforiziranem omrežju, na katerem se nahaja  $N = 50$  vozil, in posledično izračunamo tudi pretočnost danega križišča ob določenih parametrih. Prvo vozilo v vrsti  $n_1$  bo tako imelo skupni potovalni čas iz vsote reakcijskega časa 1 sekunde in časa, porabljenega za prevoženo razdaljo  $n * s$ , ob določenem pospešku. To v našem primeru skupno znaša 2,41 sekunde. Drugo vozilo v vrsti bo tako potrebovalo za prečkanje križišča 3 sekunde in tako dalje, vse do zadnjega, petdesetega v vrsti. Zadnje vozilo,  $n_{50}$ , bo v našem primeru za prečkanje semaforiziranega križišča, porabilo 60 sekund.

Če bi torej zelena luč na semaforju gorela 30 sekund, bi križišče, v danem časovnem intervalu, lahko pod temi pogoji prečkalo 23 vozil, pri čemer bi zadnje vozilo, ki bi uspelo prečkati križišče, za opravljeno pot porabilo 29,78 sekunde.

#### 4.2.4 PRETOČNOST AVTOMATIZIRANEGA PROMETA

Predstavljajmo si svet, kjer so vsi udeleženci prometa, samovozeča vozila. Vsako izmed vozil ima točne informacije o prometu, v katerem je tisti trenutek udeleženo, prav tako pa je omrežje zasnovano tako, da pridobimo skupni najboljši izkupiček vseh udeležencev v prometu in ne le posameznikov.

To pomeni, da bi vsa vozila točno vedela, kdaj se bo na semaforju prižgala zelena luč, prav tako bi vedela, s kakšno hitrostjo se in se bo premikal promet pred njimi. Kar v teoriji pomeni, da bi vsa vozila ob prižigu zelene luči lahko začela pospeševati istočasno, pri čemer se znebimo faktorja človeškega reakcijskega časa, ki vpliva na končni potovalni čas. To torej pomeni, da bi na končni potovalni čas  $n$  vozila v koloni na semaforiziranem križišču vplivala le razdalja ter pospešek vozila pred njim, saj bi, za maksimalni izkupiček, lahko pospeševala enako hitro in pri tem ne bi prišlo do trka.

Če sedaj uporabimo isti primer semaforiziranega križišča kot v zgornjem primeru, kjer smo pri izračunu končnega potovalnega časa upoštevali človeški faktor, lahko izračunamo novi potovalni čas za  $n$  vozila, pri čemer nam tega ni treba upoštevati. Konkretno to pomeni, da bo pri časovnem intervalu  $T = 30$  s na semaforju, 23. vozilo v vrsti še vedno uspelo prečkati križišče, kajti bo za 23 m oddaljenosti od prečkanja križišča

porabilo le 6,78 sekunde, kar je kar 23 sekund manj kot v primeru, kjer je vsak izmed voznikov za speljevanje potreboval 1 sekundo reakcijskega časa. V teoriji bi torej pod temi pogoji križišče lahko prečkalo kar 327 vozil, pri čemer bi 327. vozilo za prečkanje križišča potrebovalo le 29,94 sekunde.

## **5 ANALIZA PRETOČNOSTI DVOPASOVNEGA KROŽNEGA KRIŽIŠČA V LOKALNEM OKOLJU**

Krožna križišča s svojimi številnimi prednostmi postajajo vse bolj priljubljena alternativa običajnim križiščem. Najprej izstopa izjemna prometna varnost, ki jo krožna križišča nudijo, predvsem zaradi tega, ker hitrosti znotraj krožnih križišč niso velike. S svojo obliko, ki vključuje središčni otok, krožna križišča zmanjšujejo število konfliktnih točk in preprečujejo prehitevanje, kar vodi v manjše število prometnih nesreč, pri katerih so posledice običajno manj hude. Poleg tega imajo krožna križišča večjo prepustnost prometa, kar zmanjšuje zastoje in s tem prispeva k manjšemu hrupu ter emisijam. Nižji stroški vzdrževanja, estetski videz ter vloga pri umirjanju prometa v urbanih okoljih, le še dodatno utrjujejo njihovo priljubljenost in uporabnost v sodobnih mestih. S tem postajajo krožna križišča ključen gradnik bolj trajnostne in varne prometne infrastrukture.

Za raziskovanje analize in optimizacije prometa na lokalnem primeru sem si izbral krožno križišče Velenje, na katerem bom raziskoval vplive na pretočnost prometa in jo nato poskusil teoretično optimizirati s pomočjo animacije in simulacije.

### ***5.1 RAZVOJ NAJVEČJEGA KROŽNEGA KRIŽIŠČA V ŠALEŠKI DOLINI***

Največje krožno križišče v Šaleški dolini, ki se nahaja tik pod Velenjskim gradom, je bilo najprej križišče, ki je bilo prometno precej obremenjeno. Zaradi tega so oktobra leta 2009 uradno odprli novo dvopasovno krožno križišče v Velenju.

Prav to križišče je zaradi geografske lokacije prometno najbolj obremenjeno križišče v Velenju, s čimer povezuje mesto z ostalimi pomembnimi regijami. To križišče je glavno vozlišče, ki povezuje promet iz celjske strani proti Šoštanjju in savinjski regiji, hkrati pa tudi s Koroško (smer Slovenj Gradec).





**Slika 11: Križišče v Velenju pred izgradnjo krožnega križišča (Foto: Milan Marič)**

Prevelika pretočnost prometa skozi to križišče, pred tem naj bi mestna občina Velenje naredila raziskavo, kjer so prešteli skoraj 32 tisoč vozil, ki so prečkali omenjeno križišče, je bi bil povod za optimizacijo križišča in posledično povečanja pretočnosti z gradnjo krožnega križišča.



**Slika 12: Dvopasovno krožno križišče v Velenju iz ptičje perspektive [18]**

## **5.2 TEHNIČNA ANALIZA KROŽNEGA KRIŽIŠČA**

Krožno križišče v Velenju je krožno križišče s štirimi kraki, ki povezuje Celjsko cesto, Cesto Františka Foita, Šaleško cesto in Žarovo cesto, pri čemer Celjska in Foitova cesta predstavljata glavno prometno smer, ostali dve pa stransko prometno smer.

### **5.2.1 KRAK 1 – CELJSKA CESTA**

Predstavlja edini uvoz v mesto s celjske strani. Cesta se na uvozu najprej iz enega prometnega pasu, razdeli na tri, pri čemer dva vodita do vključitve v krožno križišče, medtem ko je tretji pas za desne zavijalce. To pomeni, da omogoča prometnim udeležencem zavijanje v prvi izvoz, brez potrebne vključitve v glavni krožni tok.

Glavni problem tega kraka je ta, da je izvoz na Celjsko cesto iz krožnega križišča mogoč neposredno iz notranjega pasu, pri čemer predhodna menjava cestnega pasu znotraj krožnega križišča ni potrebna. Kar ne bi bil problem, če ne bi bili prometni pasovi za uvoz in izvoz preveč strnjeni in je razdalja, na voljo za razvrščanje na zunanji pas, zatem ko se zapeljemo mimo izvoza, premajhna.



**Slika 13: Krak 1 – Celjska cesta [17]**

## 5.2.2 KRAK 2 – ŠALEŠKA CESTA

Predstavlja vhod in izhod iz mesta Velenje. Cesta se pri uvozu iz dveh prometnih pasov razdeli na tri, pri čemer tretji pas predstavlja pas za desne zavijalce, ki lahko krožno križišče obidejo v smeri Šoštanja. Druga dva prometna pasova pa vodita do vključitve v krožni prometni tok.

Izmerjena razdalja tega kraka je približno 100 m, pri čemer je na izvozu dolžina odseka ceste, na katerem se je dovoljeno razvrščati, približno 45 m. Kar pa predstavlja največji problem, saj je lahko v teoriji na tem odseku le 10 vozil na enem prometnem pasu, če upoštevamo, da je povprečna dolžina osebnega vozila približno 4,5 metra. Pri tem je potrebno tudi upoštevati dejstvo, da se lahko v prometu nahajajo tudi vozila, ki so konkretno daljša, kar ta izračun še pomanjša. Ta razdalja je definitivno prekratka, saj je takoj za izvozom semaforizirano križišče, kjer je interval rdeče luči dolg približno 30 sekund. To pa je glede na pretočnost krožnega križišča dovolj, da posledično zamaši tudi glavni krožni tok znotraj krožnega križišča, saj vozila zaradi prenatrpanosti ne morejo pravočasno zamenjati prometnega pasu.



Slika 14: Krak 2 – Šaleška cesta [17]

### 5.2.3 KRAK 3 – CESTA FRANTIŠKA FOITA

Ta krak predstavlja glavno prometno smer in povezuje mesto s Šoštanjem ter Savinjsko regijo. Krak ima tako na uvozu kot na izvozu dva prometna pasova.



Slika 15: Krak 3 – Foitova cesta [17]

### 5.2.4 KRAK 4 – ŽAROVA CESTA

Ta krak je prometno najmanj pretočen, saj povezuje mesto z okolico. Izvoz ima le en prometni pas, medtem ko se uvoz iz enega prometnega pasu razdeli na dva prometna pasova. Pri tem je desni pas namenjen le za tiste, ki bodo krožno križišče zapustili pri prvem izvozu v smeri Celjske ceste.

Glavni problem tega kraka je ta, da se na razdalji približno 6 m od krožnega toka nahaja prehod za pešce in kolesarje. To pomeni, da se na tistem odseku naenkrat lahko v povprečju nahaja le eno vozilo, ne da bi pri tem del vozila ostal znotraj krožnega toka pri izvozu ali na prehodu za pešce na uvozu.

Prehod za pešce in kolesarje je preblizu krožnemu toku, kar je lahko posledično razlog za zamašitev prometnega toka znotraj krožnega križišča, v primeru, da želi na tem izvozu naenkrat krožno križišče zapustiti več vozil in medtem prehod za pešce prečka pešec.



**Slika 16: Krak 4 – Žarova cesta [17]**

### ***5.3 MERJENJE GOSTOTE IN OPAZOVANJE PROMETA***

Za nadaljnjo izdelavo simulacije sem potreboval tudi približne podatke gostote prometa v krožnem križišču. Namen je bil predvsem pridobiti povprečno gostoto pri največji pretočnosti, kar sem sklepal, da bo ob delovnih dnevih. Odločil sem se, da bom v različnih časovnih obdobjih preštel število vozil, ki se bodo v krožno križišče vključevala ali izključevala v intervalu 1 ure.

Za opravljanje meritev sem si izbral ponedeljek in petek ob treh različnih urah, saj sem predpostavljaj, da bo takrat največja pretočnost. Meritve sem opravljal med urama 8.00 in 9.00, med 14.00 in 15.00 ter med 18.00 in 19.00. Meritve sem ponovil štirikrat, in sicer enkrat za vsak krak krožnega križišča. Kar vse skupaj nanese na 8 dni s tremi meritvami po eno uro. Vse meritve so bile opravljene v mesecu januarju.

To je vodilo do ne najbolj točnih podatkov, a tukaj nisem potreboval natančnih podatkov, temveč le približno vrednost gostote pretočnosti na vsakem uvozu in izvozu ter približen odstotek vrednosti gostote prometa vsakega izmed krakov v krožnem križišču glede na gostoto celotnega vozlišča.

#### ***5.4 IZDELAVA ANKETE NA TEMO PRETOČNOSTI IN ZADOVOLJSTVA S KROŽNIM KRIŽIŠČEM***

Za lažjo analizo in preučevanje pretočnosti krožnega križišča ter prometa nasploh, sem izdelal anketo s pomočjo orodja Google Forms, iz katere sem pridobil mnenja in odzive uporabnikov krožnega križišča ter lokalnih prebivalcev.

V anketo sem vključil starost voznikov, koliko časa že imajo vozniški izpit ali so že bili krivec v prometni nesreči in pogostost udeleževanja v prometu. Poleg tega sem v anketo vključil nekaj vprašanj na temo zbranosti v prometu, kjer so anketiranci ocenili njihovo zbranost med vožnjo ter vpliv sopotnikov in stresa na le-to. Hkrati pa so morali oceniti svoje poznavanje cestno prometnih predpisov na lestvici od ena do pet.

Kot dodatek sem anketi dodal dva primera vključevanja vozila v dvopasovno krožno križišče, kjer so lahko izbirali prometni pas, po katerem bi se v promet vključili, če bi bili tisti trenutek sami vozniki tega avtomobila v dani situaciji.

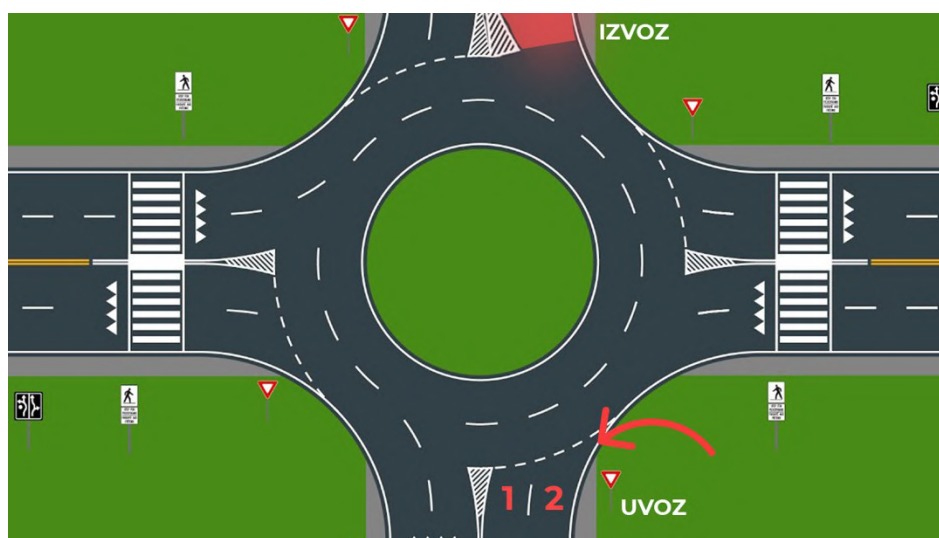
Na koncu pa so lahko izrazili svoje zadovoljstvo z dvopasovnim krožnim križiščem v Velenju in podali svoje mnenje glede pretočnosti le tega.

Vse podatke sem analiziral in preučil s pomočjo Python programskega jezika, kjer sem za napredno analizo podatkov uporabil knjižnico »Pandas«, ki omogoča enostavno in učinkovito delo s podatkovnimi formati.

### 5.4.1 VKLJUČEVANJE V KROŽNO KRIŽIŠČE NA PRIMERU DANE SITUACIJE

V anketo sem vključil dve prometni situaciji dvopasovnega krožnega križišča, kjer so morali anketiranci izbrati, po katerem prometnem pasu bi se vključili, če bi se morali v krožno križišče vključiti kot naslednji v vrsti.

Prva dana situacija je temeljila na preverjanju odločanja izbire prometnega pasu, če na uvozu, v katerem se vključujemo, ni nikogar drugega in imamo prosto pot za varno vključevanje. Na voljo za vključitev imamo dva prometna pasova, označena s številko 1 in 2, krožno križišče pa želimo zapustiti v drugem izvozu. (Slika 17).



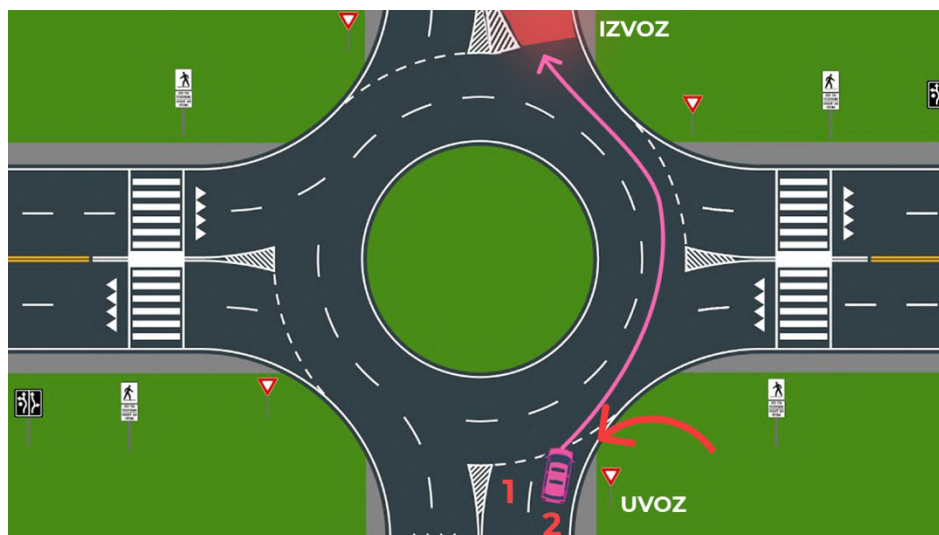
Slika 17: Prva dana situacija za vključitev v krožno križišče [23]

Druga dana situacija dvopasovnega krožnega križišča je zelo podobna prvi, vendar je na uvozu pred nami, na prometnem pasu s številko 2, že eno vozilo, ki prav tako namerava krožno križišče zapustiti v drugem izvozu (Slika 18). Pri tem predpostavljamo, da poznamo njegove namere in vemo, pri katerem izvozu bo zapustil krožišče, kar v resničnem svetu ni možno.

V tej dani situaciji lahko spet izbiramo med dvema različnima možnostma za vključevanje v krožno križišče, kjer je na uvozu pred nami že eno vozilo pripravljeno na

vklučevanje. Izbiramo lahko torej med vključevanjem označenim s številko 1, kjer se v krožni tok vključujemo sočasno z voznikom roza avtomobila prikazanega na sliki, ali pa označenim s številko 2, kjer se bomo v krožni tok vključili za voznikom roza avtomobila na istem prometnem pasu.

Pri tem je potrebno upoštevati, da v primeru, če izberemo možnost označeno s številko 1, ne bomo uspeli izvesti pravočasnega in varnega razvrščanja, saj nam bo to oviral voznik roza avtomobila. Posledično bomo primorani opraviti še en dodaten krog po notranjem pasu krožnega križišča.



Slika 18: Druga dana situacija za vključitev v krožno križišče [23]

## ***5.5 PREUČEVANJE VPLIVOV NA PRETOČNOST PROMETA V DANEM OMREŽJU***

Po raziskovanju vplivov na pretočnost prometa ter opazovanju lokalnega krožnega križišča, lahko sedaj preučim sledi teh teoretičnih vplivov in pojavov na praktičnem primeru.



### 5.5.1 SLEDI BRAESSEOVEGA PARADOKSA

Na tej točki sem opazoval sledi Braessovega paradoksa. Ta pojav sem, med opazovanjem krožnega križišča v Velenju, opazil pri vključevanju vozil v krožišče.

Krožna križišča omogočajo neprekinjen tok prometa skozi križišče in posledično povečajo pretočnost, v primerjavi z navadnim semaforiziranim križiščem. Dodajanje še enega prometnega pasu znotraj krožišča, bi moralo omogočiti večjo pretočnost, saj lahko promet poteka po obeh pasovih, pa vendar temu ni vedno tako.

Namen dvopasovnih krožišč je ta, da v primeru zapuščanja krožišča na prvem izvozu, uporabimo zunanji prometni pas, medtem ko za vse ostale izvoze uporabimo notranji prometni pas in s tem povečamo pretočnost krožišča, saj omogočimo vključitev v promet, po zunanjem prometnem pasu, ostalim voznikom. To določa tudi Zakon o pravilih cestnega prometa.

»Voznik, ki zapelje na krožno križišče, katerega vozišče ima dva ali več prometnih pasov, se razvrsti na notranji prometni pas in na ta način omogoči vključitev v promet na križišču tudi drugim voznikom, razen če zapušča križišče na najbližjem izvozu ali mu prometna ureditev ali prometna situacija tega ne dovoljuje ali mu je zaradi lastnosti križišča (npr. krožna križišča z majhnim premerom) ali lastnosti motornega vozila (npr. tovorno vozilo s priklopnikom) to oteženo.« Kršenje tega člena se lahko kaznuje z globo 40 evrov. [21]

»Preden zapusti krožno križišče, se mora voznik razvrstiti na zunanji prometni pas, razen na izvozu, na katerem je s predpisano prometno signalizacijo dovoljeno zavijanje desno tudi z notranjega prometnega pasu. V primerih, ko razvrščanje iz notranjega na zunanji pas zaradi gostote prometa ni mogoče, je voznik dolžan opraviti najmanj eno kroženje oziroma toliko kroženj, kolikor je potrebnih za razvrstitev.« Kršenje tega člena se lahko kaznuje z globo 40 evrov. [21]

V primeru, da se želi več vozil razvrstiti na notranji pas, ali pa je razdalja med trenutnim uvozom in drugim izvozom zelo majhna, se zna zgoditi, da bo kateri izmed voznikov sebično za izhod iz krožišča, pri ne prvem izvozu, izbral zunanji pas. To bo posledično pomenilo, da bo s tem lahko onemogočil pravočasno razvrščanje vozila na notranjem

pasu, ki prav tako želi krožišče zapustiti pri istem izvozu, kar ga bo prisililo, da opravi vsaj še eno kroženje.

Posledično bo zaradi tega, ker ne bi rad ostal ujet na notranjem pasu, tudi sam za izhod pri drugem izvozu, začel uporabljati zunanji pas za vključitev v krožišče. To bo privedlo do tega, da bo vse več in več ljudi začelo uporabljati zunanji pas, kar pa bo povzročilo zastoje in kolone pri vključitvi krožišča po zunanjem pasu.

Na tej točki omrežje doseže Nashovo ravnotežje. Stanje v katerem nobeno izmed vozil nima razloga za menjavo prometnega pasu, saj s tem ne bi pridobilo ničesar. V primeru, da bi za vključitev izbralo notranji pas, pa obstaja možnost, da v teoriji, ostane v krožišču ujeta za vedno.

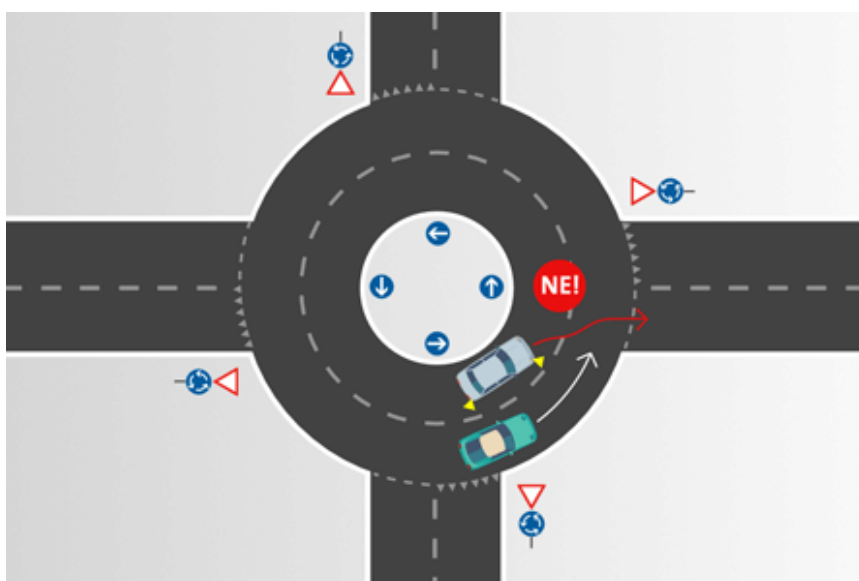
### **5.5.2 NENADNE ZAMAŠITVE PROMETNEGA TOKA**

Vzrokov za pojav nenadne zamašitve prometnega toka je več, vendar v primeru, da eden izmed udeležencev v prometu stori napako (npr. nepravilno razvrščanje na zunanji pas), povzroči spremembo hitrosti vseh ostalih udeležencev na zunanjem pasu, kar lahko privede do popolne zaustavitve prometnega toka. Posledično lahko traja nekaj časa, preden vsi spet pospešijo nazaj do prejšnje hitrosti, da se zamašitev sprostí.

Med opazovanjem prometa na Velenjskem dvopasovnem krožišču, sem nemalokrat opazil popolni zastoj, ki se je nato čez čas sprostil. Največkrat je bil za zastoj razlog v infrastrukturi in ne prometnih udeležencih, kot na primer prehod za pešce na izvozu za Žalsko cesto ali pa prekratka dolžina izvoza na Šaleško cesto.

Za zastoj pa so bile krive tudi napake voznikov, največkrat nepravilno razvrščanje in nepravilna uporaba signalizacije. Nepravilno razvrščanje je bilo še posebej opazno na izvozu za Foitovo in Šaleško cesto, kjer se vozila velikokrat sploh niso razvrstila na zunanji pas, ampak so krožišče zapustila direktno iz notranjega pasu. To, ne le da ni v skladu s cestno prometnimi predpisi, ampak je tudi zelo nevarno, saj lahko privede do trčenja z vozilom na zunanjem pasu, ki misli, da vozilo na notranjem pasu še ne bo zapustilo krožišča.

»V primeru, da krožno križišče zapuščamo na prvem izvozu, ostanemo na zunanjem prometnem pasu, za vse ostale izvoze pa uporabimo notranji prometni pas. Krožno križišče smemo zapustiti le z zunanjega prometnega pasu, razen če prometna signalizacija zahteva drugače. V primeru, da zgrešimo izvoz ali pa se nam iz notranjega pasu ni uspelo pravočasno razvrstiti na zunanji prometni pas, odpeljemo še en krog ter se tako na zunanji pas razvrstimo ob naslednji priložnosti.« Kršenje tega člena se lahko kaznuje z globo 40 evrov. [21]



Slika 19: Nepravilno zapuščanje krožišča direktno iz notranjega pasu [20]

Kot omenjeno, pa je razlog tudi nepravilna uporaba signalizacije, ki povzroča zmedo ostalih udeležencev v krožnem križišču. Največkrat je bilo to možno opaziti tam, kjer so se želela vozila, pri uvozu Celjske ceste, razvrščati iz notranjega na zunanji pas in so pri tem prezkodaj vklopila utripalko. S tem so zavajala vozila, ki so se v krožišče želela vključiti iz Celjske ceste, saj so mislila, da bodo namesto razvrščanja, krožišče zapustila iz notranjega pasu, kar tisti odsek omogoča. Posledično se je vozilo, ki je prezkodaj vklopilo utripalko, začelo razvrščati, hkrati pa se je vozilo na uvozu začelo vključevati. To je vodilo do tega, da sta oba pritisnila na zavoro in precej zmanjšala hitrost, lahko tudi popolnoma zaustavila.



Slika 20: Prikaz pravilne razvrstitve in signalizacije pri zupuščanju krožišča [20]

»Pri zupuščanju krožnih križišč zaradi zagotavljanja pretočnosti in varnosti ostalih udeležencev v prometu vedno pravočasno uporabimo desne utripalke, najboljše takoj ko odpeljemo mimo zadnjega uvoza. S prehitrim vklopom lahko zavedemo voznika, ki se želi vključiti na krožno križišče, kar lahko privede do nevarne situacije, s prepoznim vklopom pa tvegamo nalet vozila z zadnje strani in zmanjšujemo pretočnost krožnega križišča. Pri vključevanju na krožno križišče ni treba vklopiti desnih utripalk. Vklopimo jih le, če je krožno križišče izvedeno tako, da je izvoz, po katerem želimo zapustiti krožno križišče, v neposredni bližini uvoza.« [21]

### 5.5.3 POJAV EFEKTA SHOCKWAVE

Ta pojav sem opazil pri vseh uvozih v krožišče, kjer se je ob začasni sprostitvi krožnega toka, v krožišče vključilo manj vozil, kot bi se jih lahko, če bi z enakim pospeševanjem se začela vozila vključevati istočasno.

Največkrat pa je bilo mogoče ta pojav opaziti na uvozu Žarove ceste, zaradi prehoda za pešce v neposredni bližini krožišča. Prehod za pešce namreč omogoča, da se krožnemu križišču naenkrat lahko približa le eno vozilo na vsakem pasu, nakar prepreči priključek

naslednjega, saj ustavljanje na prehodih za pešce ni dovoljeno. Zato mora naslednje vozilo ustaviti na oddaljenosti, od glavnega krožnega toka, pri približno 13 m. To posledično onemogoči vključitev v krožišče, ki bi lahko bila mogoča, če bi se lahko postavil tik za prvo vozilo.

## ***5.6 TEORETIČNE REŠITVE ZA POVEČANJE PRETOČNOSTI KROŽIŠČA***

Največji problem tega krožišča je, da je razvrščanje, iz notranjega prometnega pasu na zunanega, velikokrat onemogočeno zaradi prevelike gostote prometa na zunanjem prometnem pasu.

### **5.6.1 REKONSTRUKCIJA KROŽIŠČA**

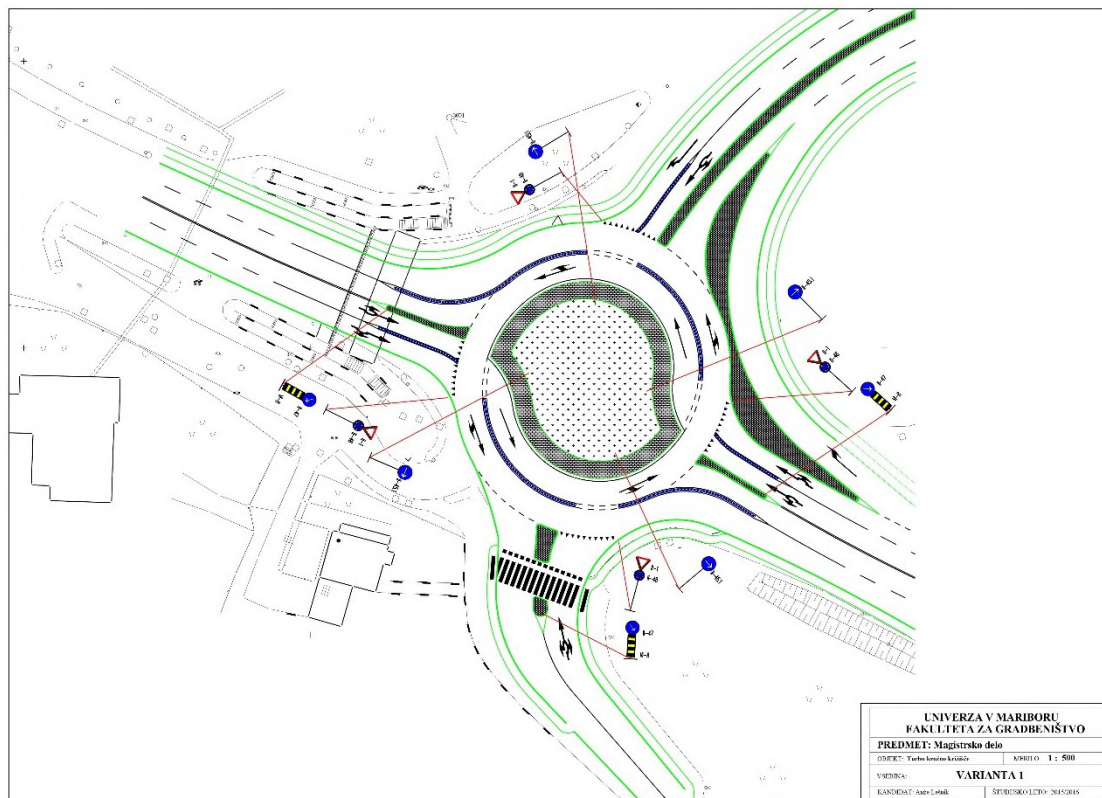
Ena izmed rešitev, bi lahko bila rekonstrukcija krožnega križišča, vendar je to sedaj zelo težko izvedljivo, saj bi bilo potrebno ves promet na tem vozlišču preusmeriti. Poleg tega pa bi to verjetno bil tudi velik finančni zalogaj. Pa si vseeno pogledjmo načine, kako bi lahko krožišče dolgoročno izboljšali.

Med raziskovanjem sem našel dva načrta Velenjskega krožišča, ki sta potencialni rešitvi za optimizacijo pretočnosti prometa na podlagi rekonstrukcije celotnega krožišča. Oba načrta sta bila izdelana v sklopu magistrskega dela, leta 2016. [19]

Prva rešitev je »turbo krožišče«, ki bi bila izvedljiva z minimalnimi posegi v trenutno konstrukcijo krožnega križišča. Glavni razlog za učinkovitost je ta, da so pasovi fizično ločeni, pri čemer menjava prometnih pasov ni dovoljena. Zaradi tega je seveda nujno potrebna pravilna razvrstitev pri uvozu v krožišče, saj kroženje po takšnem krožišču ni mogoče in boste primorani nadaljevati v smeri, ki vam jo izbrana pot določa.

To je tudi izjemno učinkovita rešitev iz vidika varnosti, saj je številno konfliktnih točk minimalno, kar zmanjša možnosti trčenja. Hkrati pa predstavlja velik finančni zalogaj,

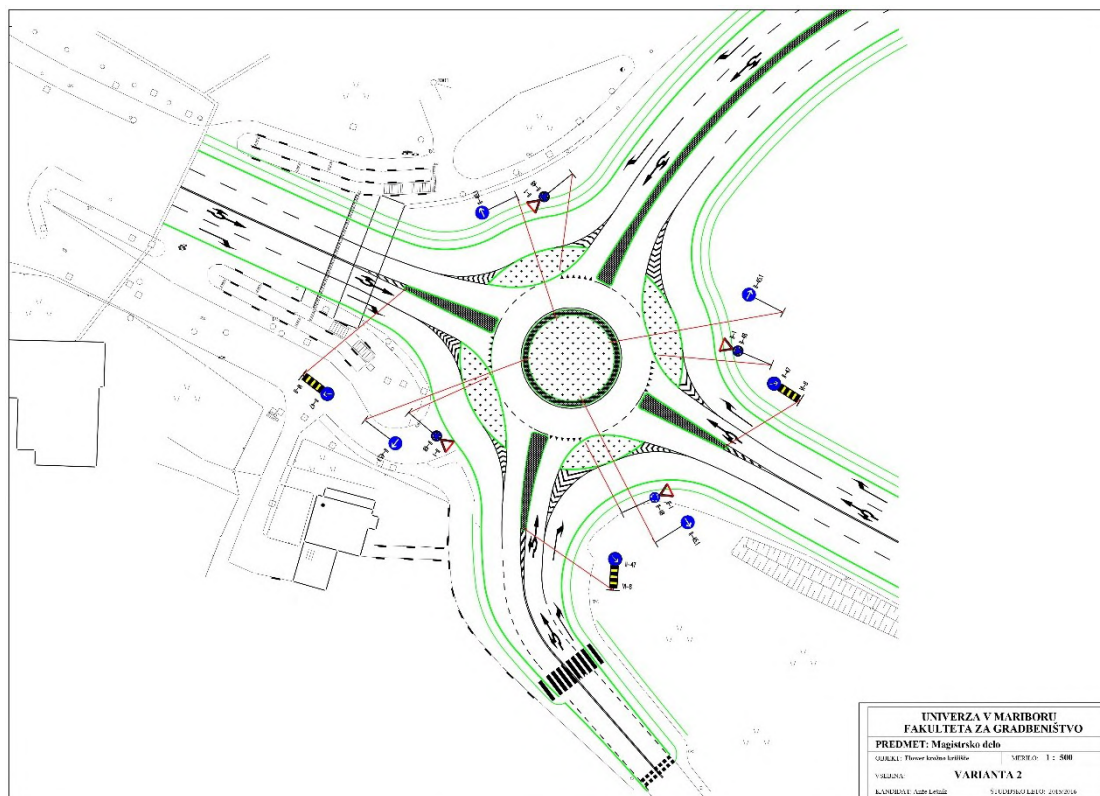
saj bi bilo potrebno rekonstruirati sredinski otok ter zgraditi robnike, ki ločujejo prometne pasove.



Slika 21: Načrt rekonstrukcije krožišča – turbo [19]

Druga rešitev pa bi bilo krožišče »flower«, ki ima poudarek predvsem na pasovih za desne zavijalce pri prvem izvozu. Posledično krožno križišče iz dveh pasov spremeni v enega, kjer se zunanji pas pretvori v obvoz krožišča do prvega izvoza.

Pri izvedbi takšne rekonstrukcije bi bil glavni zalogaj dodajanje novih prometnih pasov ter otoki, ki ločujejo krožišče s pasovi za desne zavijalce.



Slika 22: Načrt rekonstrukcije krožišča – flower [19]

Pri tem je potrebno upoštevati, da ta rešitev onemogoči vstop na parkirišče, med Foitovo in Žarovo cesto, direktno iz krožnega križišča. Vendar je ta še vedno mogoča preko Foitove ceste. Pri tem bi bilo najbrž potrebno dodati še možnost levega zavijanja pri izvozu na Foitovi cesti. Medtem ko bi pri rešitvi »turbo«, ostal dostop do omenjenega parkirišča preko krožnega toka.

## **6 IZDELAVA SIMULACIJE PROMETNEGA OMREŽJA NA KONKRETNEM PRIMERU IZ LOKALNEGA OKOLJA**

Po konkretni analizi največjega krožnega križišča v mojem lokalnem okolju, Velenju, sem se odločil, da bi rad preveril ali je možno pretočnost krožnega križišča povečati s pomočjo računalniške simulacije. Rad bi odkril načine, ki bi pripomogli k izboljšanju pretočnosti in posledično zmanjšali potovalni čas, ki je potreben za prevoz krožnega križišča.

### ***6.1 NAČRTOVANJE IZDELAVE SIMULACIJE***

Preden sem začel z izdelavo simulacije, sem moral določiti ključne stvari, ki so potrebne za uspešno izvedbo računalniške simulacije krožnega križišča. Cilj izdelave računalniške simulacije je predvsem povečati pretočnost krožnega križišča, zato sem najprej izdelal načrt, kako naj bi moje krožno križišče izgledalo. Zaradi tega, ker sem želel simulacijo poenostaviti, sem se odločil, da bom izdelal krožno križišče le z enim voznim pasom in ne z dvema. Razlog za to odločitev je bil predvsem ta, da želim odpraviti problem prenatrpanosti prometnega toka na določeni relaciji, ki posledično onemogoči vključitev drugih uvozov v krožišče. Za optimiziranje takšnega problema, je dovolj simulacija le z enim voznim pasom.

Nato sem se moral odločiti, katere parametre bi rad v simulaciji spreminjal, da bi dosegel čim večjo pretočnost samega krožnega križišča. Najprej bi rad simuliral  $N$  število vozil v določenem krožnem križišču, pri čemer ima vsako izmed vozil vnaprej določeno pot od točke A do B, pri čemer točka A predstavlja uvoz, v katerem se vozilo vključuje v promet in točka B izvoz, v katerem bo vozilo zapustilo krožno križišče.

Vsako izmed vozil v simulaciji, bo torej imelo vnaprej načrtano pot, katero mora prevoziti, pri čemer je potrebno upoštevati tudi druge udeležence v tej preprosti prometni simulaciji. To pomeni, da nobeno izmed vozil ne bo smelo ogrožati ali izsiliti drugih



udeležencev v prometu. Vsako izmed vozil se bo ravnalo po cestno predpisanih pravilih in cilj bo zgolj in samo pripeljati se do končne destinacije. Nato želim še dodati možnost spreminjanja gostote vozil na določenih relacijah. Na primer, da bom lahko določil delež N vozil, ki bodo s svojo potjo začela na določenem uvozu.

Svojo simulacijo bi rad naredil čim bolj modularno, kar bi mi omogočilo enostavno spreminjanje in dodajanje elementov v simulacijo. Poleg tega bi rad omogočil možnost širjenja, saj je moj namen dodati čim več stvari, ki bi lahko pripomogle k izboljšanju rezultatov moje simulacije in posledično simulacijo uporabil tudi na drugih primerih prometnih omrežij v prihodnosti.

## ***6.2 IZBIRA ORODJA ZA IZDELAVO SIMULACIJE***

Moj namen ni le ta, da bi izdelal simulacijo, ki mi bo kot rezultat vrnila podatke, katere bi lahko posledično prikazal s pomočjo grafa, ampak tudi možnost vizualnega vpogleda v delovanje same simulacije, ki bi omogočila lažjo predstavo delovanja simulacije. Zato je bil moj namen izdelati simulacijo, ki jo bomo lahko spremljali v realnem času in bo hkrati razumljiva in enostavna za razumeti. To sem dosegel tako, da sem se odločil za izdelavo 3D simulacije, ki bo hkrati tudi animirana. Za izdelavo takšne simulacije poznamo več orodij, vendar sem se odločil za uporabo orodja Unity, ki je namenjeno predvsem za izdelavo 2D in 3D računalniških iger, kot nalašč za izdelavo moje animirane simulacije v tri-dimenzionalnem okolju.

### **6.2.1 KAJ JE UNITY IN ZAKAJ SE UPORABLJA?**

Unity je vodilna platforma za razvoj iger in interaktivnih 3D in 2D aplikacij, ki združuje zmogljiva orodja za izdelavo in upravljanje digitalnih vsebin. Ta programska oprema je ustvarjena z namenom olajšanja razvoja različnih vrst digitalnih projektov, kot so računalniške igre, simulacije, virtualna resničnost, razširjena resničnost in interaktivne izobraževalne aplikacije.

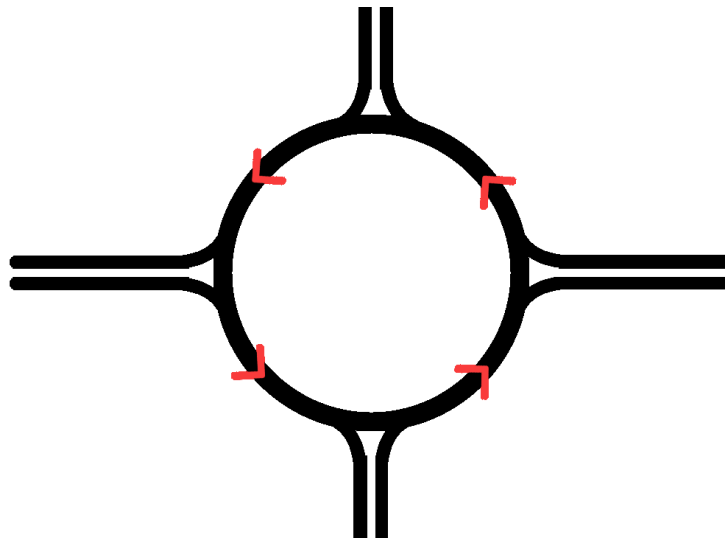
Unity omogoča razvijalcem dostop do različnih orodij, kot so vizualni urejevalniki, animacijski sistemi, fizikalni motorji in podpora za več platform, kar olajša ustvarjanje kompleksnih digitalnih vsebin. S pomočjo Unityja lahko ustvarjalci enostavno združujejo grafiko, zvok, animacije in programiranje v enotno integrirano okolje, kar pospešuje proces razvoja in omogoča hitrejše iteracije.

Platforma Unity je priljubljena med razvijalci zaradi svoje dostopnosti in prilagodljivosti, ki omogoča ustvarjanje raznolikih projektov za različne namene in ciljne skupine. Uporablja se lahko povsod, v izobraževanju, zdravstvu, arhitekturi in drugih panogah za ustvarjanje interaktivnih in vizualno bogatih izkušenj. Skozi leta je Unity postal ključno orodje v industriji digitalnega ustvarjanja, ki omogoča ustvarjanje inovativnih, zabavnih in izobraževalnih vsebin za široko občinstvo po vsem svetu.

Prav zaradi njegove prilagodljivosti in različnih orodij, ki jih ponuja, sem se odločil, da bom svojo simulacijo izdelal z orodjem Unity, ki za razvoj uporablja C# programski jezik.

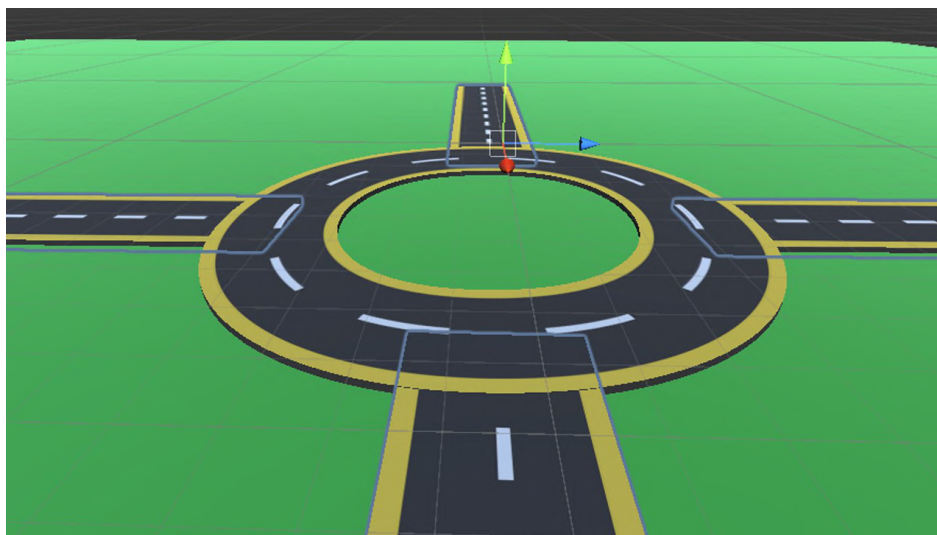
### ***6.3 3D IZDELAVA PROMETNEGA OMREŽJA***

Preden sem začel s programskim delom izdelave simulacije, sem si najprej moral pripraviti 3D okolje v katerem bom izdelal simulacijo. Najprej sem skiciral načrt, kako želim, da bi moja simulacija izgledala. Kot sem že prej omenil, sem načrtal enopasovno krožno križišče s štirimi izvozi.



Slika 23: Skica enostavnega krožnega križišča

Nato sem s pomočjo Blenderja izdelal 3D model krožnega križišča in cest, ki se nanj priključujejo. Mojemu 3D modelu sem dodal teksturo in ga nato uvozil v moj Unity projekt, da sem lahko nadaljeval z izdelavo simulacije.



Slika 24: 3D model krožnega križišča

## **6.4 OSNOVNE MEHANIKE PROMETNIH UDELEŽENCEV**

Ko sem izdelal 3D model prometnega omrežja, je sledil programski del simulacije. Prvi korak pri izdelavi prometnega omrežja, znotraj Unity programskega okolja, je bil ustvarjanje vozil, ki bodo v simulaciji predstavljala udeležence v prometu. To pomeni, da se bodo znala sama, brez nekoga, ki bi jih pri tem nadziral in upravljal, premikati po vnaprej določenih poteh in pri tem smiselno pospeševati ter zavirati, če bo le to potrebno.

Tako sem začel s preprostim samodejnim premikanjem 3D kocke, ki je v mojem primeru predstavljala vozilo, po vnaprej določeni poti. Kako točno sem to izvedel? Tako kot se ljudje v prometu premikamo od točke A do točke B, tako sem zasnoval tudi program, kjer bo vsako vozilo potovalo od začetne do končne točke, pri čemer bo vsaka izmed točk predstavljala točko s tremi vrednostmi: x, y in z os v mojem 3D prostoru. Ampak ker sem se odločil za izdelavo poenostavljenega prometnega omrežja, sem zato določil, da bodo vozila spreminjala svojo pozicijo le glede na dve osi, in sicer x ter z os. S tem sem premikanje omejil na 2D prostor, kjer bodo vozila ves čas skozi simulacijo ohranjala vrednost y, kar pomeni, da v mojem primeru, ne bodo spreminjala višine.

### **6.4.1 LINEARNO PREMIKANJE VOZILA OD TOČKE A DO TOČKE B**

Sedaj pa naslednji korak, premikanje vozila od koordinat prve točke, do koordinat druge točke. To sem storil tako, da sem v metodi *Update()*, ki se izvede poleg vsake sličice v sekundi, spremenil pozicijo vozila. Spreminjanje pozicije vozila, mi je omogočila funkcija *Lerp()*, linearna interpolacija med dvema točkama, ki kot parameter vzame dve točki tipa *Vector3* in decimalno vrednost *t*, ki je vrednost na intervalu  $[0,1]$  in predstavlja vrednost, za kakšen delež naj funkcija interpolira med tema dvema točkama. Funkcija nato vrne rezultat tipa *Vector3*, to so le koordinate v 3D prostoru, ki predstavlja vsoto razdalj med začetno točko a in razliko razdalje med obema točkama, pomnoženo z vrednostjo *t*.

**Enačba 2: Linearna interpolacija med dvema točkama**

$$c = a + (b - a) * t$$

V konkretnem primeru nam vrednost  $t$  predstavlja delež razdalje med začetno točko in trenutno točko, ki pa je zmnožek med hitrostjo linearne interpolacije in časom. To pomeni, da bi radi za vsak časovni interval spremenili razdaljo našega predmeta od začetne točke do vmesne točke, s koeficientom med začetno in končno točko. Če ponazorim to s poenostavljenim primerom, si predstavljajmo, da imamo dve točki, ki sta med seboj oddaljeni 10 m. Recimo, da želimo med tema dvema točkama na časovnem intervalu 1s interpolirati s hitrostjo 1 m/s. To pomeni, da se bo trenutna točka linearne interpolacije vsako sekundo prestavila za 1 m, kar pomeni, da bomo med tema dvema točkama z razdaljo 10 m interpolirali v 10 sekundah, pri čemer se bo vrednost  $t$  vsako sekundo povečala za 1/10 razdalje.

V Unity programskem okolju, to storimo za vsako sličico, ki se v eni sekundi izvede, kjer začetni čas odštejemo od trenutnega, da dobimo čas od začetka izvajanja interpolacije in ga nato pomnožimo s hitrostjo. Nato za vsako sličico izračunamo novo točko med začetno in končno točko, glede na izračunan delež razdalje. Posledično spreminjamo pozicijo nove točke, pri čemer bo gladkost premikanja odvisna od gostote izvajanja metode, kar v našem primeru pomeni od velikosti sličic na sekundo.

```
public class LinearInterpolationExample : MonoBehaviour
{
    public Transform startMarker;
    public Transform endMarker;

    // Hitrost v m/s
    public float speed = 1.0f;

    // Začetni čas
    private float startTime;

    // Razdalja med obema točkama
    private float journeyLength;

    void Start()
    {
        // Čas ob začetku izvajanja
        startTime = Time.time;

        // Izračun razdalje med obema točkama
        journeyLength = Vector3.Distance(startMarker.position,
            endMarker.position);
    }

    // Izvajanje Update() metode vsako sličico na sekundo
    void Update()
    {
        // Razdalja enaka zmnožku razdalje med časom in hitrostjo
        float distCovered = (Time.time - startTime) * speed;

        // Delež trenutne razdalje glede na skupno razdaljo
        float fractionOfJourney = distCovered / journeyLength;

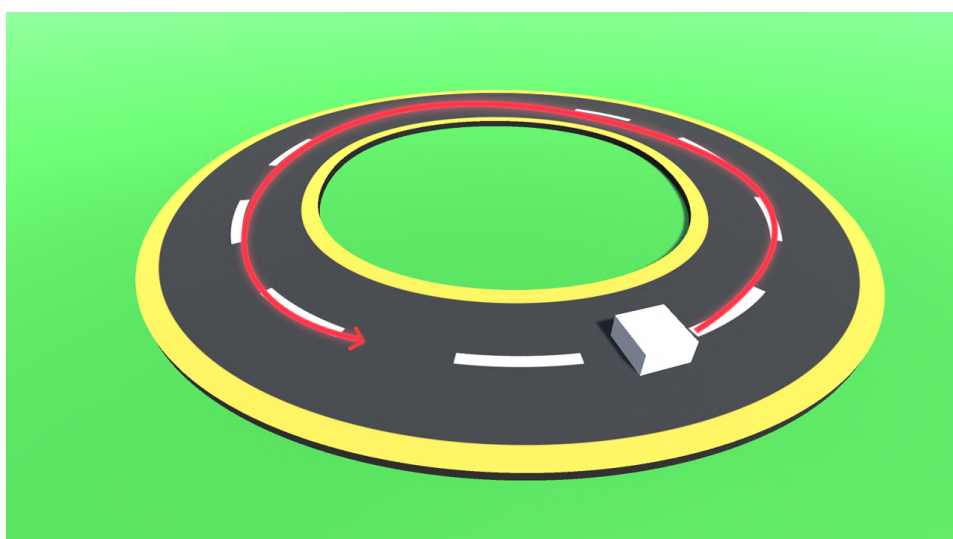
        // Nastavljanje nove pozicije glede na trenutni delež razdalje
        transform.position = Vector3.Lerp(startMarker.position,
            endMarker.position, fractionOfJourney);
    }
}
```

**Koda 2: Prikaz uporabe funkcije Lerp() na osnovnem primeru**

## 6.4.2 INTERPOLACIJA TOČK NA KRIVULJI

Vendar uporaba linearne interpolacije v krožnem križišču ni možna, saj je uporaba le-tega možna le na linearni premici, na kateri ležita obe točki. Tega pa v prometu ne vidimo pogosto, saj ceste v prometnih omrežjih niso linearne, vendar so večinoma, če gledamo iz ptičje perspektive, krivuljastih oblik. Prav zaradi tega, ker krožno križišče ni ravna

premica, med točkami ne moremo linearno interpolirati, saj posledično interpolacija med točkama ne bo v obliki krivulje. Kako torej interpolirati med dvema točkama v obliki kroga ali druge ovalne oblike? Unity ima v svoji zbirki funkcij tudi funkcijo *Slerp()*, ki nam omogoča sferično interpolacijo med dvema vektorjema. Od funkcije *Lerp()* se razlikuje predvsem v tem, da se vektorji namesto kot točke v prostoru, obnašajo kot rotacijske smeri. To pomeni, da je smer vrnjenega vektorja interpolirana s kotom, pri čemer je razdalja odvisna od linearne razdalje med začetno in končno točko.

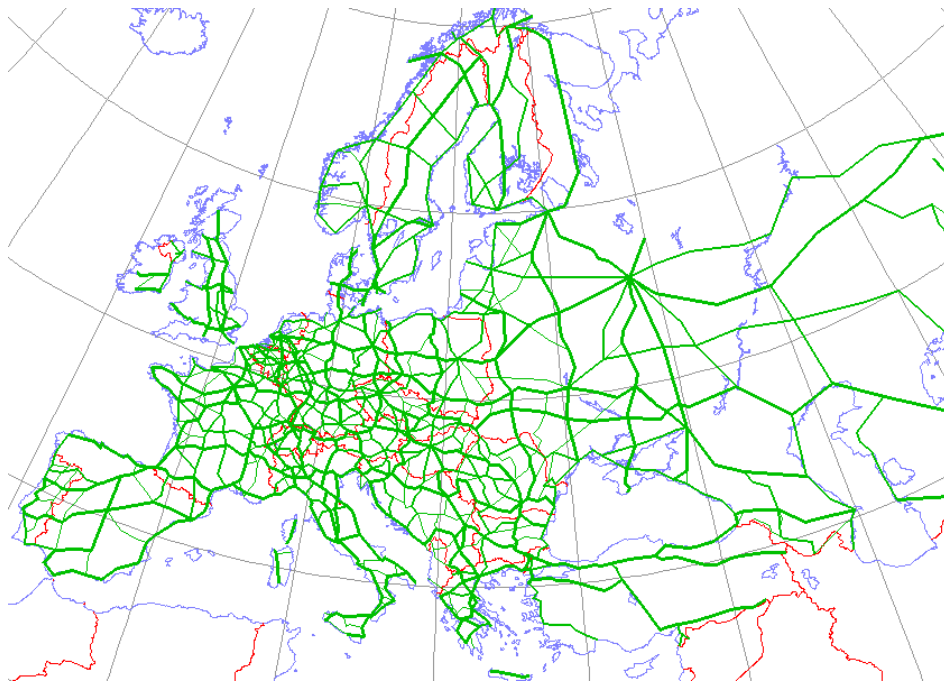


**Slika 25: Prikaz želene interpolacije v obliki krivulje, in nelinearno**

Vendar moramo pri tem upoštevati, da nam to omogoča interpolacijo med dvema točkama le sferično, kar pomeni, da pri interpolaciji v 2D prostoru, bo interpolacija le v obliki kroga in ne kakršnekoli poljubne krivulje.

### **6.4.3 BÉZIERJEVA KRIVULJA**

Prav to pa vodi do naslednje ugotovitve, namreč svet ni sestavljen le iz ravnih črt in pravilnih krogov, povsod bomo lahko zasledili nepravilne oblike in prav te so v prometu najbolj opazne, ker so ceste v oblikah najbolj zahtevnih krivulj.



**Slika 26: Primer cestnega omrežja [22]**

Torej uporaba funkcij *Lerp()* in *Slerp()* bi bila možna, le če bi zasnoval prometno omrežje, ki bi bilo sestavljeno iz cest, ki bi bile ravne ali v obliki kroga. Kar bi sicer bilo dovolj, za izdelavo res enostavne simulacije krožnega križišča, a vendar mi je bil cilj izdelati simulacijo, ki bo dovolj modularna, da se bi jo dalo razširiti. Zato se mi je zdelo najbolj smiselno izdelati simulacijo, kjer bo lahko cestno prometno omrežje poljubne oblike in oblika ter načrt omrežja ne bosta imela vpliva na delovanje same simulacije. Posledično sem zaradi želje, da bi lahko izdelal prometno omrežje s cestami poljubnih oblik, naletel na koncept Bézierjeve krivulje.

Bézierjeva krivulja je parametrična krivulja (parameter  $t$  na intervalu  $[0,1]$ ), uporabljena v računalniški grafiki in sorodnih področjih. Sestavljena je iz diskretnih kontrolnih točk, ki določajo gladko, neprekinjeno krivuljo s pomočjo formule. Običajno je krivulja namenjena približevanju oblike iz resničnega sveta, ki sicer nima matematične predstavitve ali je predstavitev preveč zapletena ali neznana. Uporablja se tudi pri oblikovanju računalniških pisav in pri animaciji.



V vektorski grafiki se Bézierjeve krivulje uporabljajo za modeliranje gladkih krivulj, ki se lahko neomejeno povečajo. Tako nastanejo poti, ki so kombinacije povezanih Bézierjevih krivulj. Poti niso omejene z mejami rasteriziranih slik in jih je zelo enostavno spreminjati.

Torej Bézierjeva krivulja je točno to, kar potrebujem za načrtovanje oblik prometnega omrežja iz resničnega sveta, kjer bodo v večini nastopale krivulje, po katerih se bodo premikala vozila v moji simulaciji. Sedaj moram te krivulje le še izračunati.

Poznamo tri osnovne tipe Bézierjevih krivulj:

- Linearna (brez kontrolnih točk)
- Kvadratna (ena kontrolna točka)
- Kubična (dve kontrolni točki)

Kontrolne točke se uporabljajo za definiranje poti krivulje med dvema končnima točkama z določanjem njene smeri, ukrivljenosti in oblike nasploh. Postavitve in manipulacija kontrolnih točk omogočata prilagajanje, oblikovanje in ustvarjanje gladke in natančne krivulje. Pri čemer v bistvu kontrolne točke delujejo kot sidra ali ročaji, ki usmerjajo obnašanje krivulje in omogočajo učinkovito doseganje zelenih oblik in vzorcev.

Pa si pogledajmo delovanje Bézierjeve krivulje s pomočjo kontrolnih točk na konkretnem primeru, kjer želimo med seboj povezati dve točki. To lahko storimo, tako da najprej izračunamo linearno Bézierjevo krivuljo med tema dvema točkama.

### Enačba 3: Linearna Bézierjeva krivulja

$$\mathbf{B}(t) = (1 - t)t\mathbf{P}_0 + t\mathbf{P}_1, 0 \leq t \leq 1$$

$\mathbf{B}(t)$  v tem primeru predstavlja točko na tej krivulji z neko vrednostjo  $t$  na intervalu  $[0,1]$ , ki je izračunana glede na dve končni točki  $\mathbf{P}_0$  in  $\mathbf{P}_1$ . Krivulja v tem primeru ni krivuljaste oblike, ampak je ravna črta ter predstavlja linearno interpolacijo med dvema točkama.



**Slika 27: Prikaz linearne povezave dveh točk v koordinatnem sistemu**

Če sedaj v koordinatni sistem dodamo še tretjo točko, ki nam bo predstavljala kontrolno točko, lahko s pomočjo te točke izračunamo kvadratno Bézierjevo krivuljo.

#### **Enačba 4: Kvadratna Bézierjeva krivulja**

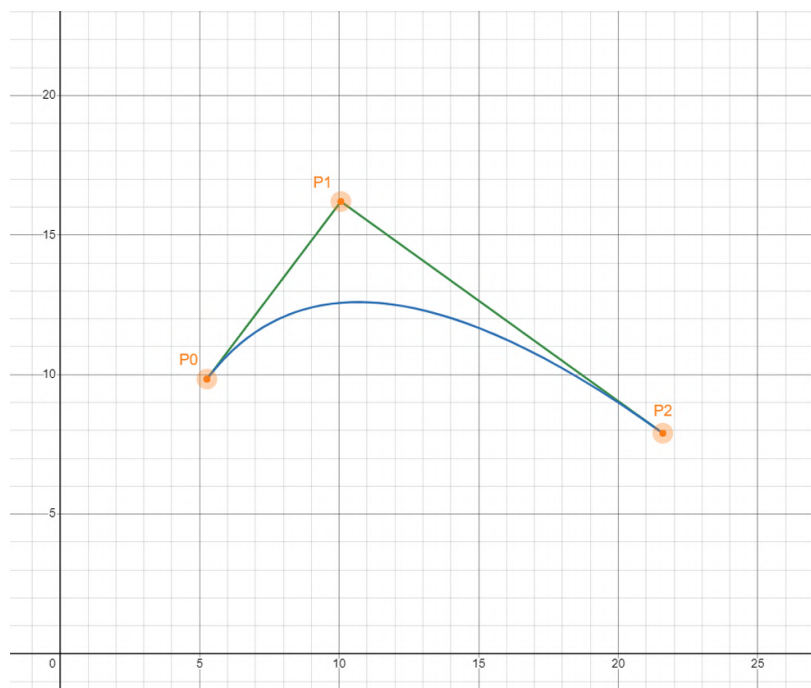
$$\mathbf{B}(t) = (1 - t)^2 \mathbf{P}_0 + 2(1 - t)t \mathbf{P}_1 + t^2 \mathbf{P}_2, 0 \leq t \leq 1$$

Iz enačbe lahko razberemo, da je Bézierjeva krivulja druge stopnje le interpolacija med dvema linearnima krivuljama. To lahko ponazorimo, tako da med sabo povežemo vse tri točke v zaporednem vrstnem redu. Povežemo jih s pomočjo enačbe linearne krivulje (Enačba 1), kjer povežemo točki  $P_0$  in  $P_1$  ter  $P_1$  ter  $P_2$ , pri čemer nam točka  $P_1$  predstavlja kontrolno točko. Tako dobimo dve linearni Bézierjevi krivulji, ki nam ponazarjata dve linearni interpolaciji med dvema točkama.



**Slika 28: Tri linearno povezane točke**

$B(t)$  v tem primeru predstavlja vrednost  $t$  na intervalu  $[0,1]$ , pri čemer namesto interpolacije med končnima točkama  $P_0$  in  $P_2$ , funkcija najprej interpolira med obema linearnima povezama  $P_0$  in  $P_1$  ter  $P_1$  in  $P_2$ , ki predstavljata dve linearni Bézierjevi krivulji. Nato pa interpolira med tema dvema linearnima krivuljama za vsako vrednost  $t$ . Tako dobimo kvadratno Bézierjevo krivuljo med točkama  $P_0$  in  $P_2$ .



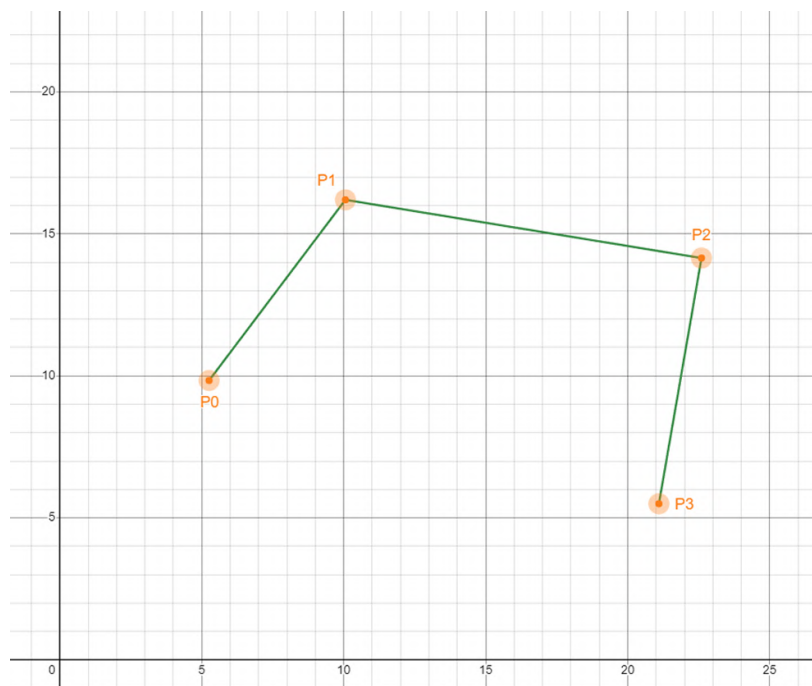
**Slika 29: Prikaz kvadratne Bézierjeve krivulje**

Glavna razlika med kvadratno in kubično Bézierjevo krivuljo je ta, da ima kvadratna krivulja manj kontrolnih točk, kar pomeni, da je kubična krivulja bolj prilagodljiva in vsestranska, saj omogoča širok razpon oblik krivulj. Ravno zato sem se odločil, da bom v svoji simulaciji uporabljal krivulje tretje stopnje, saj želim čim večjo natančnost in prilagodljivost.

**Enačba 5: Bézierjeva krivulja tretje stopnje**

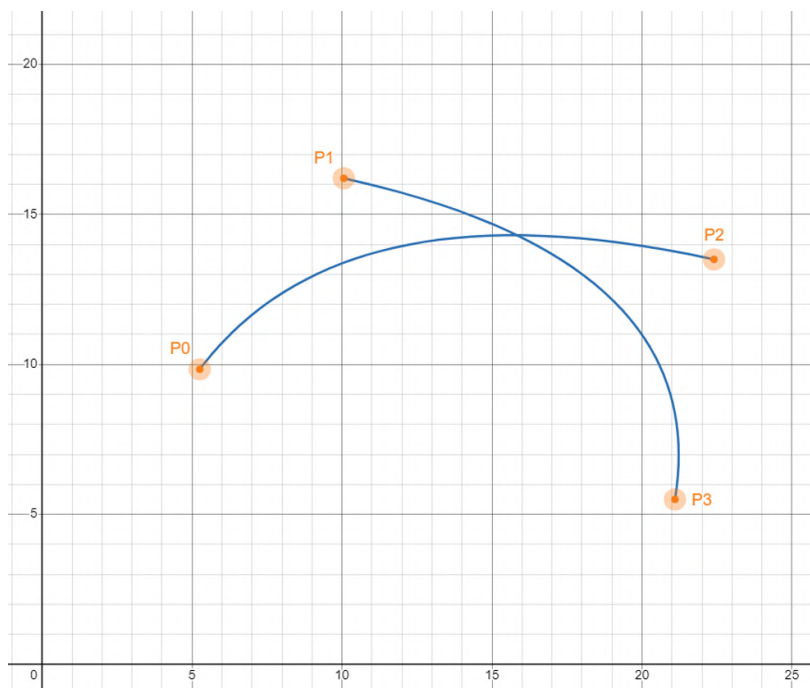
$$\mathbf{B}(t) = (1 - t)^3 \mathbf{P}_0 + 3(1 - t)^2 t \mathbf{P}_1 + 3(1 - t) t^2 \mathbf{P}_2 + t^3 \mathbf{P}_3, 0 \leq t \leq 1$$

Za prikaz izračuna Bézierjeve krivulje tretje stopnje potrebujemo poleg dveh končnih točk (Slika 27) še dve dodatni kontrolni točki, ki bosta uporabljeni za računanje ukrivljenosti krivulje, ki jih lahko nato med sabo v zaporednem vrstnem redu najprej linearno povežemo, s pomočjo linearne Bézierjeve krivulje.



**Slika 30: Štiri linearno povezane točke v zaporednem vrstnem redu**

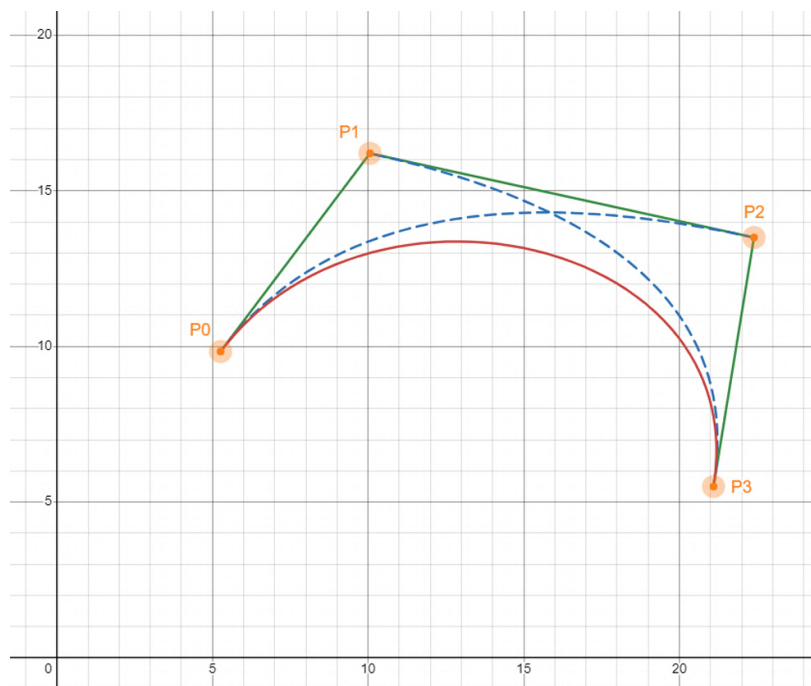
Tako smo sedaj linearno povezali, za vsako vrednost  $t$  na intervalu  $[0,1]$ , točki  $P_0$  in  $P_1$ ,  $P_1$  in  $P_2$ ,  $P_2$  in  $P_3$  ter  $P_2$  in  $P_3$ , kjer točki  $P_1$  in  $P_2$  predstavljata kontrolni točki, točki  $P_0$  in  $P_3$  pa končni točki. Sedaj lahko opazimo vzorec dveh krivulj druge stopnje, pri čemer lahko izračunamo kvadratno krivuljo med točkama  $P_0$  in  $P_2$ , kjer  $P_1$  predstavlja kontrolno točko in  $P_1$  in  $P_3$ , kjer kontrolno točko predstavlja točka  $P_2$ . Te točke lahko sedaj med seboj povežemo s pomočjo kvadratne Bézierjeve krivulje in tako dobimo dve krivulji druge stopnje.



**Slika 31: Prikaz dveh kvadratnih krivulj med štirimi točkami**

Vrednost  $B(t)$  sedaj dobimo tako, da za vrednost  $t$ , na intervalu  $[0,1]$  interpoliramo med dobljenima kvadratnima krivuljama.

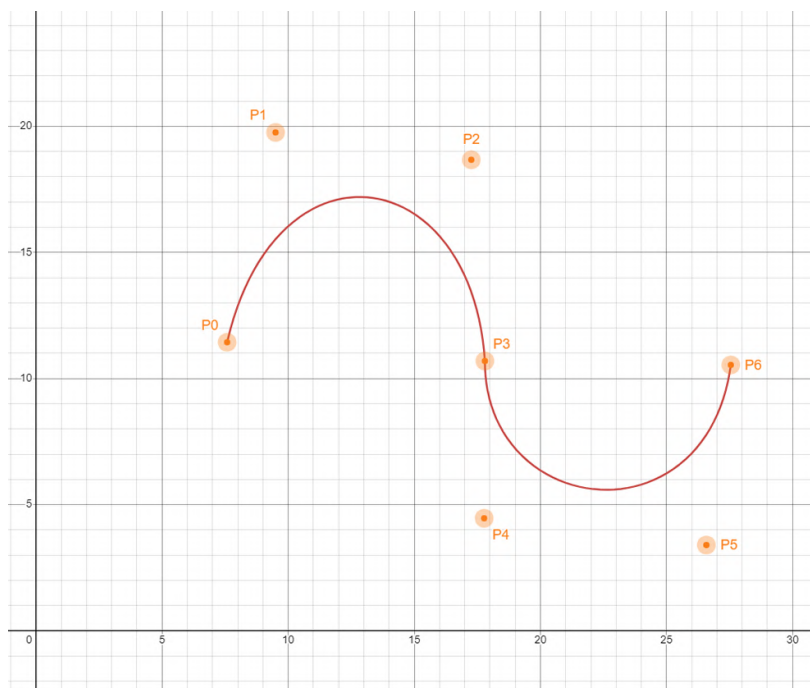
Kot rezultat dobimo krivuljo med točkama  $P_0$  in  $P_3$ , pri čemer je odvod krivulje tretje stopnje enak koeficientu premice, ki povezuje obe krivulji druge stopnje v točki  $B(t)$ .



**Slika 32: Prikaz Bézierjeve krivulje tretje stopnje**

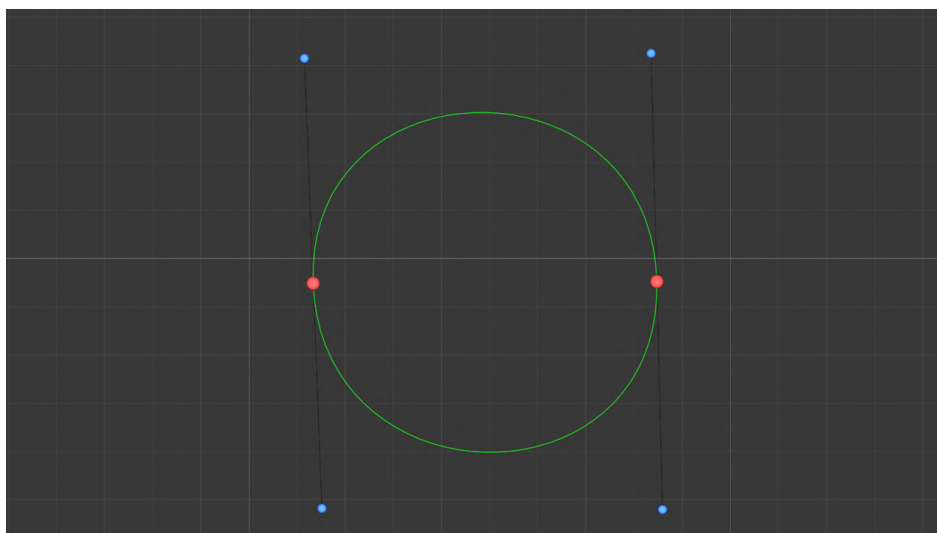
Sedaj lahko s pomočjo Bézierjevih krivulj sestavljamo kompleksne poljubne krivulje, ki tvorijo različne poti, podobne takšnim, kot jih poznamo v resničnem svetu. To lahko storimo tako, da dodamo novo točko  $P_6$ , katero želimo v obliki krivulje povezati s točko  $P_3$ . Zato dodamo vmes še dve kontrolni točki  $P_4$  in  $P_5$ , ki bosta usmerjali ukrivljenost te nove krivulje. Posledično dobimo sestavljeno Bézierjevo krivuljo tretje stopnje, ki povezuje točke  $P_0$ ,  $P_3$  in  $P_6$  s pomočjo kontrolnih točk  $P_1$ ,  $P_2$ ,  $P_4$  in  $P_5$ .

Posledično lahko tako sestavljamo krivulje v neskončnost, prav tako pa lahko tvorimo zaprte ali odprte krivulje, pri čemer je zaprta krivulja sklenjena in začetna točka  $P_0$  predstavlja tudi končno točko. Takšno obliko krivulje bom potem uporabil za izdelavo sklenjenega dela krožnega križišča, medtem ko bodo uvozi in izvozi, ki se nanj priključujejo, ostali odprte nesklenjene krivulje.



**Slika 33: Sestavljena Bézierjeva krivulja tretje stopnje**

Znotraj Unity projekta sem na osi y označil točke, katere želim, da s pomočjo Bézierjevih krivulj tretje stopnje tvorijo sestavljeno pot. S tem sem narisal krivuljo v obliki kroga, ki bo predstavljala začetno pot, za prometne udeležence v simulaciji, kateri morajo slediti.



**Slika 34: Sestavljena Bézierjeva krivulja označena znotraj Unity programskega okolja**

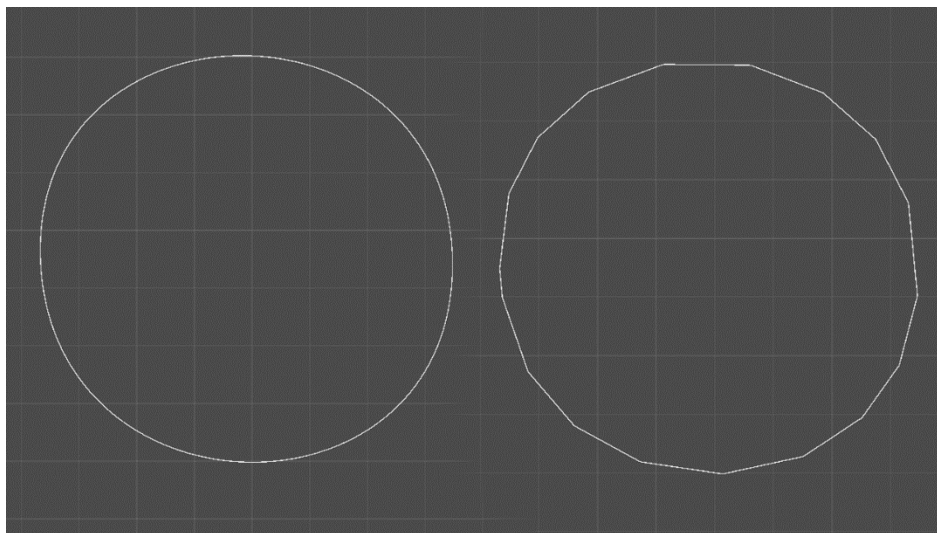


#### 6.4.4 PRETVARJANJE KRIVULJE V POT Z OGLIŠČI

Bézierjeve krivulje so matematično precej kompleksen koncept, zato bi sprotno računanje krivulj tretje stopnje, za vsakega izmed prometnih udeležencev v moji simulaciji, pri vseh vrednostih  $t$  na intervalu  $[0,1]$ , zahtevalo kar precej procesorske moči. To bi naposled lahko vodilo k neuravnoteženemu delovanju simulacije, posledično tudi zrušenju celotnega sistema, saj bi se računalniška simulacija preobremenila.

Zato krivuljo pred začetkom simulacije pretvorimo v pot z oglišči. To pomeni, da na krivuljo postavimo zgoščene točke, ki so nato med seboj linearno povezane. Namreč v teoriji lahko krog tako zelo povečamo, da dobimo raven odsek obsega. Enako lahko storimo tudi s krivuljami, vendar gostoto točk omejimo, da simulacije ne preobremenimo.

Tako lahko po potrebi spreminjamo koeficient gostote, ki v našem primeru predstavlja  $N$  število točk na eni enoti dolžine ali obsega celotne krivulje. To v praksi pomeni, koliko vrednosti  $t$  bomo uporabili za izris krivulje. Opazimo lahko, da večja kot je gostota točk, bolj je ukrivljenost gladka, natančna.



Slika 35: Primerjava med zgoščenimi in ne zgoščenimi točkami na krivulji

#### 6.4.5 RAČUNANJE POZICIJE NA DANI KRIVULJI

Računanje pozicije na premici v koordinatnem sistemu je precej enostavno. Koordinate točke dobimo tako, da v funkcijo premice  $f(x)$  vstavimo vrednost  $x$ , funkcija pa nam vrne vrednost  $y$ . Enako lahko storimo pri vseh ostalih funkcijah kot so, kvadratna, eksponentna, racionalna funkcija, polinom itd. To je mogoče zaradi tega, ker ima vsak  $x$  svojo vrednost  $y$  in hkrati nobena izmed vrednosti  $y$  nima več kot ene vrednosti  $x$ . Posledično je zato interpolacija po takšnih funkcijah precej enostavna, saj povečujemo ali manjšamo vrednost  $x$ , nato nam funkcija vrne vrednost  $y$ , in mi lahko pridobimo neskončno mnogo točk na dani funkciji.

To pa ni vedno nujno za poljubno krivuljo v koordinatnem sistemu, saj je mogoče, da ima ena vrednost  $y$  več vrednosti  $x$ , kar pomeni, da ne moremo izračunati točne pozicije s pomočjo funkcije. Zato sem za računanje pozicije na krivulji uporabil drug način.

Vse točke, ki smo jih dobili pri pretvarjanju krivulje v pot z oglišči, lahko shranimo v en seznam. Pri tem bodo vse točke razvrščene po vrsti, od začetne točke na Bézierjevi krivulji, do končne točke. Pri čemer, bomo izračunali skupno razdaljo vseh linearnih razdalj med sosednimi točkami, ki smo jih dodali na to krivuljo. S tem bomo dobili skupno dolžino te naše poti. Nato bomo za izračun posamezne točke lahko uporabili funkcijo, ki bo kot parameter zahtevala delež razdalje na tej krivulji na intervalu  $[0,1]$ , kot rezultat pa bomo dobili točko, ki je od začetne točke oddaljena za zahtevan delež celotne razdalje poti z oglišči.

```
// Točka glede na prepotovano razdaljo
public Vector3 GetPointAtDistance (float dst) {
    float t = dst / length;
    return GetPointOnInterval (t);
}

// Pridobljena točka glede na vrednost 't', kjer je 0 začetek poti in 1 konec
poti.
public Vector3 GetPointOnInterval (float t) {
    var data = CalculatePercentOnPathData (t);
    return Vector3.Lerp (GetPoint (data.previousIndex), GetPoint
(data.nextIndex), data.percentBetweenIndices);
}
```

**Koda 3: Funkcija za izračun točke na krivulji, glede na razdaljo**

Iz zgornjega primera programske kode (Koda 3), lahko opazimo, da funkcija za izračun točke glede na vrednost  $t$  na intervalu  $[0,1]$ , za pridobljene podatke kliče drugo funkcijo *CalculatePercentOnPathData()*, ki kot rezultat vrne nov objekt s podatki dveh točk, ki obdajata želeno vrednost parametra  $t$  glede na celotno pot. To izračunamo tako, da začnemo pri sredinski točki celotne poti in preverimo, če se ta točka nahaja pred ali po vrednosti  $t$ . Nato razdaljo v tisti smeri, v kateri se vrednost  $t$  nahaja, razpolovimo in ta postopek ponavljamo, dokler ne najdemo dveh točk, ki delež celotne poti pri vrednosti, obdajata.

```
// Za vrednost 't' med 0 in 1, izračuna indeksa dveh vozlišč, ki obdajata to
vrednost t.
// Izračuna tudi, kako daleč je t med tema dvema vozliščema kot odstotek med 0
in 1.
TimeOnPathData CalculatePercentOnPathData (float t) {

    int prevIndex = 0;
    int nextIndex = NumPoints - 1;
    int i = Mathf.RoundToInt (t * (NumPoints - 1)); // Začetna točka

    // Začne pri sredinski točki in preveri, če t leži levo ali desno od te
točke.
    // Nadaljuje z razpolavljanjem razdalje med indeksi, dokler ne najde prave
točke.
    while (true) {
        // t leži levo
        if (t <= times[i]) {
            nextIndex = i;
        }
        // t leži desno
        else {
            prevIndex = i;
        }
        i = (nextIndex + prevIndex) / 2;

        if (nextIndex - prevIndex <= 1) {
            break;
        }
    }

    float abPercent = Mathf.InverseLerp (times[prevIndex], times[nextIndex],
t);
    return new TimeOnPathData (prevIndex, nextIndex, abPercent);
}
```

**Koda 4: Funkcija za izračun točke glede na odstotek celotne poti**

#### 6.4.6 SAMODEJNO SLEDENJE VOZIL PO NAČRTOVANI POTI

Sedaj, ko sem uspel raziskati način, kako ustvariti poljubno krivuljo, jo pretvoriti v točke, ki so med seboj linearno povezane in izračunati posamezne točke glede na delež celotne razdalje, lahko med temi točkami interpoliram, da bodo objekti sledili tej poti.

Tako sem v okolju Unity ustvaril vozilo, ki bo zaenkrat v namene poenostavljanja le v obliki kocke. Nato sem načrtal sklenjeno krivuljo, po kateri želim da poteka simulacija vozila. Vozilo sem nato sprogramiral tako, da v zanki sledi načrtani poti. Vozilu lahko spreminjamo različne parametre, kot so začetna pozicija (ang. spawn point), hitrost, pot,

kateri mora slediti in smer. Nato za vsako sličico na sekundo izračuna razdaljo glede na hitrost in čas in spremeni pozicijo vozila v točko, ki jo vrne funkcija *GetPointAtDistance()*, pri trenutni razdalji.

Pri tem sem naletel na težavo, saj namreč premikanje po načrtovani poti ne izgleda naravno. To pa zaradi tega, ker pri spreminjanju pozicije ohranja rotacijo, kar posledično pomeni statično premikanje po poti. Zato sem moral vozilom v simulaciji poleg spreminjanja pozicije dodati še spreminjanje rotacije, ki deluje na podoben način. Glede na razdaljo, za določeno točko na krivulji, objekt orientira v smeri normale te krivulje.

```
public Quaternion GetRotationAtDistance (float dst) {
    float t = dst / length;
    return GetRotation (t);
}

// Izračuna rotacijo, ki usmeri objekt v določeni točki vzdolž normale
krivulje
public Quaternion GetRotation (float t) {
    var data = CalculatePercentOnPathData (t);

    Vector3 direction = Vector3.Lerp (localTangents[data.previousIndex],
    localTangents[data.nextIndex], data.percentBetweenIndices);

    Vector3 normal = Vector3.Lerp (localNormals[data.previousIndex],
    localNormals[data.nextIndex], data.percentBetweenIndices);

    return Quaternion.LookRotation (MathUtility.TransformDirection (direction,
    transform, space), MathUtility.TransformDirection (normal, transform,
    space));
}
```

#### Koda 5: Funkcija za računanje rotacijske smeri na krivulji

Tako lahko sedaj v programski kodi, ki je del vsakega objekta, v mojem primeru vozila v simulaciji, dodam še sprotno računanje in spreminjanje rotacije za vsako sličico v sekundi. To storim tako, da se nastavi nova rotacijska vrednost objekta v metodi *Update()*, poleg nove pozicijske vrednosti, za vsako spremembo razdalje na krivulji.

```
void Update()
{
    if (pathCreator != null)
    {
        distanceTravelled += speed * Time.deltaTime;

        transform.position =
            pathCreator.path.GetPointAtDistance(distanceTravelled,
            endOfPathInstruction);

        transform.rotation =
            pathCreator.path.GetRotationAtDistance(distanceTravelled,
            endOfPathInstruction);
    }
}
```

**Koda 6: Spreminjanje pozicije in rotacije objekta na krivulji**

#### 6.4.7 SAMODEJNO ZAVIRANJE IN POSPEŠEVANJE VOZIL

Ko sem uspešno izdelal samodejno sledenje vnaprej določeni poti, pri čemer se vozila tudi pravilno obračajo v smeri vožnje, je bil moj namen dodati programsko logiko vsem prometnim udeležencem, ki bi preprečila morebitno trčenje z ostalimi udeleženci.

To sem dosegel tako, da sem na sprednji del vozila dodal senzor, ki bo zaznaval oddaljenost vozil pred njim. Senzor v tem primeru predstavlja le neviden objekt, kateremu lahko spreminjam dolžino, ki bo deloval kot prag, kje mora vozilo pričeti z zaviranjem. Razlog, da sem senzor dodal le na sprednji del in ne okoli celotnega vozila je ta, ker vozilo ne spreminja smeri vožnje. Vožnja bo vedno potekala naravnost, v smeri krivulje, saj vozilo sprti spreminja rotacijsko smer glede na smer krivulje.

Deloval bo kot sprožilec, ki bo preverjal ali se senzor prekriva s kakšnim drugim vozilom ali ne. V primeru, da se, bo izračunal razdaljo med voziloma. Če bo razdalja enaka dolžini senzorja, pomeni, da se vozilo premika z enako hitrostjo kot vozilo pred njim, pri čemer bo dolžina senzorja predstavljala varnostno razdaljo med vozili. V primeru, da bo razdalja med voziloma manjša kot dolžina senzorja, bo vozilo začelo z zaviranjem, dokler ne bo razdalja enaka ali večja kot dolžina senzorja.

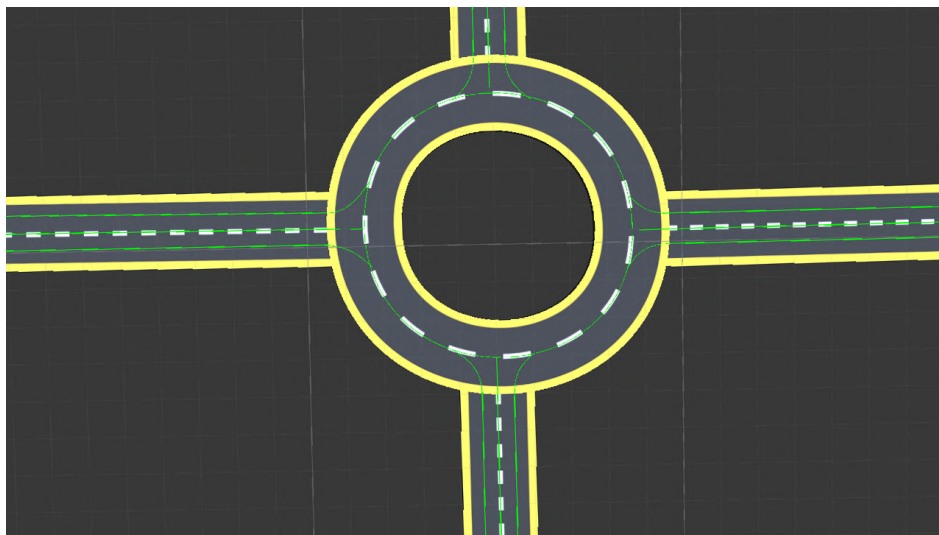
To bo izračunano tako, da se bo izračunal odstotek razdalje med voziloma glede na dolžino senzorja. Če bo odstotek razmerja dolžin večji od 1, pomeni, da ima vozilo

spredaj dovolj prostora in lahko začne s pospeševanjem. V primeru, da bo razmerje enako, vozilo ne bo niti zaviralo, niti pospeševalo, ampak bo ohranjalo enakomerno hitrost. V primeru, da bo ta vrednost manjša kot 1, pomeni, da je razdalja med voziloma premajhna in bo vozilo začelo z zaviranjem.

## ***6.5 USTVARJANJE OMREŽJA POTI***

S pomočjo Bézierjevih krivulj, lahko ustvarjamo poljubne krivulje, ki so lahko tako sklenjene kot neskljenjene, poleg tega pa jih lahko med sabo sestavljamo v neskončno dolge poti. Vendar prometno omrežje ni le ena pot, je skupek več možnih poti.

Zato moram tudi moji simulaciji omogočiti, da lahko prometni udeleženci menjajo med potmi, katerim bodo sledili. Konkretno na primeru krožnega križišča, bo ena pot predstavljena kot glavni krožni tok, pri čemer bo vsak izvoz vseboval še dve dodatni poti, eno za vključitev in drugo za izključitev iz krožnega prometa.



**Slika 36: Več izrisanih poti, ki sestavljajo prometno omrežje znotraj programa Unity**

V krožnem križišču mora tako vozilo v simulaciji priti iz ene poti, ki predstavlja uvoz, preko druge poti, ki predstavlja krožni tok, do tretje poti, ki predstavlja izvoz. Prestaviti

se mora od točke A do točke B, pri čemer se točki ne nahajata na isti krivulji. Naš model vozila je potrebno nadgraditi, da ne bo sledil le eni določeni poti, vendar bo znal sam zamenjati pot, ko bo to potrebno z namenom uspešnega prihoda do zelenega cilja.

Zato sem celotno pot, od točke A do B, zapisal v obliki seznama zaporednih poti, po katerih se mora vozilo peljati. Nato sem na mesta, kjer se poti med seboj križajo, v mojem primeru krožnega križišča, se v nobeni izmed točk ne sekajo tri ali več poti, postavil sprožilce, ki bodo zaznali, ko bo vozilo prečkalo križanje teh poti. To sem storil tako, da sem model vozila sprogramiral tako, da je za vsako sličico preverjal ali se dotika sprožilca na tem križišču. V primeru, da se, začne slediti naslednji poti.

```
public void OnTriggerEnter(Collider other)
{
    // Preveri, če se dotika sprožilca z oznako "PathIntersection"
    if (other.gameObject.tag == "PathIntersection")
    {
        Car.nextPoint = RoadNetwork.GetNextPoint(Car.currentPath,
        Car.nextPath);

        if (Car.nextPoint == other.GetComponent<PathIntersection>().point)
        {
            // Zamenjanje trenutne poti z naslednjo potjo
            Car.currentPath = Car.nextPath;
        }
    }
}
```

#### Koda 7: Sprožilec za preverjanje križanja poti

Nato sem, v primeru, da je vozilo prečkalo sprožilec, ki nakazuje spremembo poti, izračunal najbližjo točko nove poti glede na trenutno pozicijo vozila. To pa zaradi tega, ker bi drugače vozilo prestavilo na začetek poti, saj bi se le sledenje novi poti začelo pri začetni točki in ne nujno pri točki, na kateri se nahaja vozilo. Zato je potrebno izračunati točko, ki je tisti trenutek menjanja poti, najbližje vozilu, in sledenje poti začeti od N točke na poti.



```
// Računanje najbližje točke glede na pozicijo objekta
TimeOnPathData CalculateClosestPointOnPathData (Vector3 localPoint) {

    float minSqrDst = float.MaxValue;
    Vector3 closestPoint = Vector3.zero;
    int closestSegmentIndexA = 0;
    int closestSegmentIndexB = 0;

    for (int i = 0; i < localPoints.Length; i++) {
        int nextI = i + 1;
        if (nextI >= localPoints.Length) {
            if (isClosedLoop) {
                nextI %= localPoints.Length;
            } else {
                break;
            }
        }

        Vector3 closestPointOnSegment = MathUtility.ClosestPointOnLineSegment
            (localPoint, localPoints[i], localPoints[nextI]);
        float sqrDst = (localPoint - closestPointOnSegment).sqrMagnitude;
        if (sqrDst < minSqrDst) {
            minSqrDst = sqrDst;
            closestPoint = closestPointOnSegment;
            closestSegmentIndexA = i;
            closestSegmentIndexB = nextI;
        }
    }

    float closestSegmentLength = (localPoints[closestSegmentIndexA] -
        localPoints[closestSegmentIndexB]).magnitude;

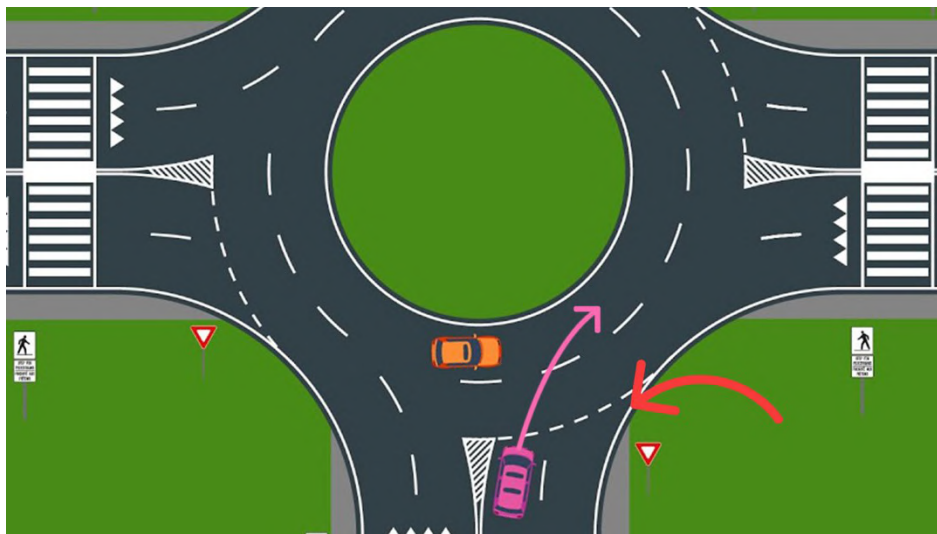
    float t = (closestPoint - localPoints[closestSegmentIndexA]).magnitude /
        closestSegmentLength;

    return new TimeOnPathData (closestSegmentIndexA, closestSegmentIndexB, t);
}
```

**Koda 8: Funkcija za izračun najbližje točke na poti glede na pozicijo objekta**

### 6.5.1 DODAJANJE VMESNIH SENZORJEV ZA VARNO VKLJUČITEV V PROMET

Za varno vključevanje v promet, specifično vključevanje v krožna križišča, moramo preveriti ali imamo za to dovolj časa, da to opravimo varno, ne da bi ob tem povzročili morebitni trk z drugimi udeleženci v prometu. Pri tem pa ni dovolj, da spremljamo promet, ki se odvija pred nami, ampak tudi znotraj samega krožnega križišča.



Slika 37: Primer, kjer se avtomobil ne more vključiti v krožno križišče [23]

Na zgornjem primeru (Slika 37) lahko opazimo, da se voznik roza avtomobila ne more varno vključiti na notranji pas, saj se v tem trenutku na notranjem pasu že nahaja voznik oranžnega avtomobila. Vendar ga sedaj v moji simulaciji še ne bi zaznal, saj se ne nahaja pred njim. V moji simulaciji bi torej voznik roza avtomobila pričel z vključevanjem, nakar bi zaznal vozilo pred njim in se posledično zaradi prihajajočega trka ustavil sredi krožnega križišča, pri čemer bi oviral pretočnost prometa. Prav tega pa jaz nisem želel, zato sem dodal vmesne točke, lahko jih imenujemo senzorje, pred vključitvijo v krožni promet.

Senzorji bodo tako delovali kot trikotni stop znak v resničnem življenju, kjer se moramo udeleženci v prometu po potrebi zaustaviti, saj po prednosti cesti že poteka promet. Ko bo avtomobil v moji simulaciji prišel do senzorja, bo ta preveril ali se na odseku krožnega loka, med izvozom in uvozom v krožišče pri trenutnem kraku, na katerem se avtomobil nahaja, nahaja še kakšno drugo vozilo. V primeru, da se, se bo avtomobil primoran ustaviti in počakati, dokler ne bo imel proste poti. V nasprotnem primeru se bo lahko varno vključil v krožni promet.

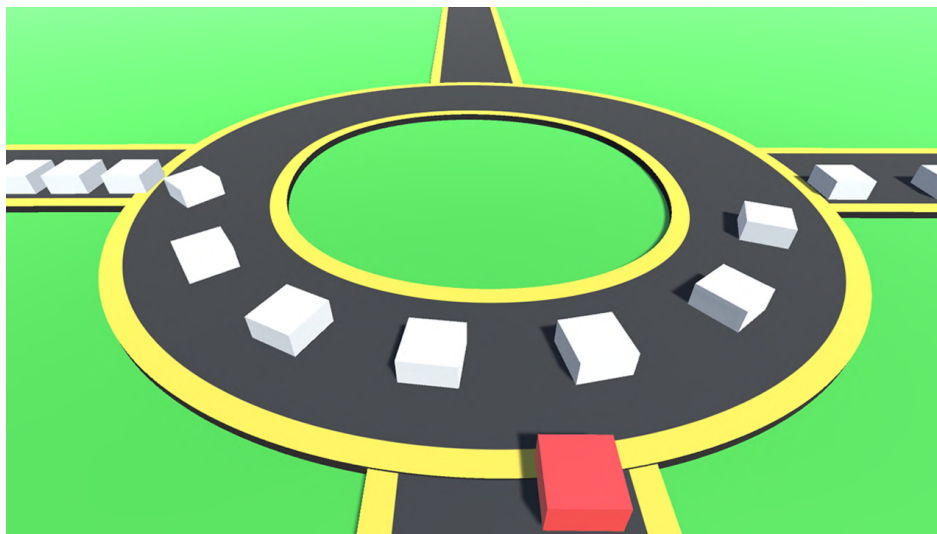
## 6.5.2 NADGRADNJA OMREŽJA Z UPORABO SEMAFORIZIRANE SIGNALIZACIJE

Za poskus preprečevanja zamašitve krožnega križišča, sem se odločil, da v simulacijo dodam semaforje, ki bodo signalizirali dovoljenje za vključitev v krožno križišče. To bi moralo tok na določenih relacijah zaustaviti, če bi bilo to potrebno, s čimer bi se krožni tok sprostil, kar bi omogočilo vključitev v krožišče drugih uvozov, katerih relacija ni med dominantnimi.

Odločil sem se za uporabo enostavnega primera semaforja, ki bo imel le dve stanji: rdečo in zeleno luč. Semafor sem postavil na vsak priključek v krožno križišče in ga povezal s senzorjem pred vstopom v krožni tok. Senzor sem nato posodobil, da poleg preverjanja prometnega toka znotraj krožišča, preveri tudi luč na semaforju.

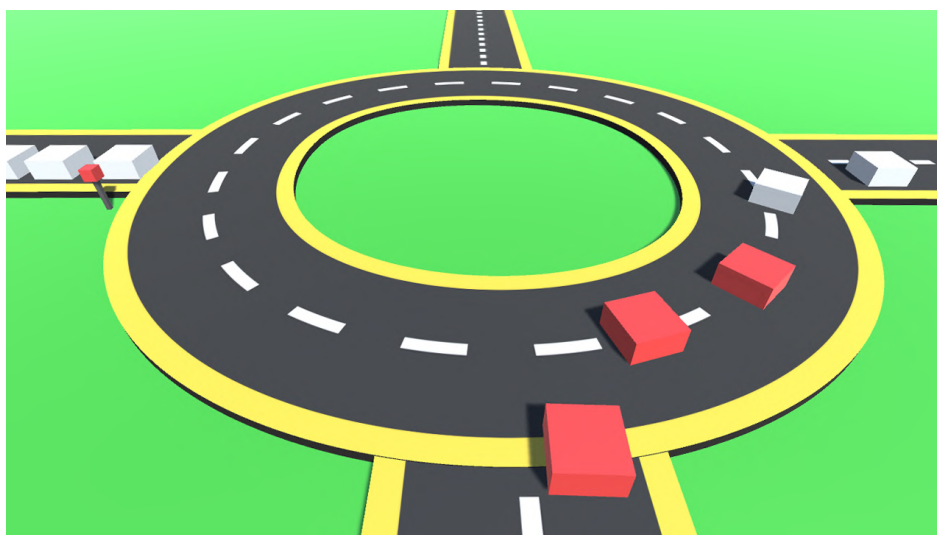
Nato sem senzorjem dodal še osveževanje časa, ki bo sproti merilo čas čakanja za vstop v krožišče. To sem storil z namenom, da bi lahko nastavil največji dovoljen prag, pri katerem je vozilo v stanju čakanja na sprostitev toka. S tem sem upal, da bi lahko povečal pretočnost skozi vse uvoze.

To v praksi pomeni, da bi v primeru, kjer bi dominantna relacija preprečila vključitev drugih udeležencev v promet, to začasno zaustavili in s tem sprostili promet na drugih uvozih, kjer se je pred krožiščem naredila več metrov dolga kolona.



**Slika 38: Prikaz dominacije določene relacije znotraj simulacije**

V zgornjem primeru (Slika 38), se tako rdeče vozilo ne bo moglo vključiti v krožno križišče, dokler vozilo znotraj križišča tega ne bo zapustilo pri izvozu, na katerem se sedaj rdeče vozilo nahaja in s tem omogočilo rdečemu vozilu dovolj časa, da se vključi.



**Slika 39: Prikaz delovanja simulacije s semaforizirano signalizacijo**

## **7 REZULTATI**

Moja raziskava je bila namenjena predvsem analizi in optimizaciji prometnega toka s pomočjo računalniških algoritmov za namen preučevanja in simuliranja različnih vplivov na promet, njegovo pretočnost in udeležence v njem. Vsi rezultati temeljijo na opravljeni anketi, meritvah gostote prometa lokalnega krožišča ter izdelani računalniški simulaciji.

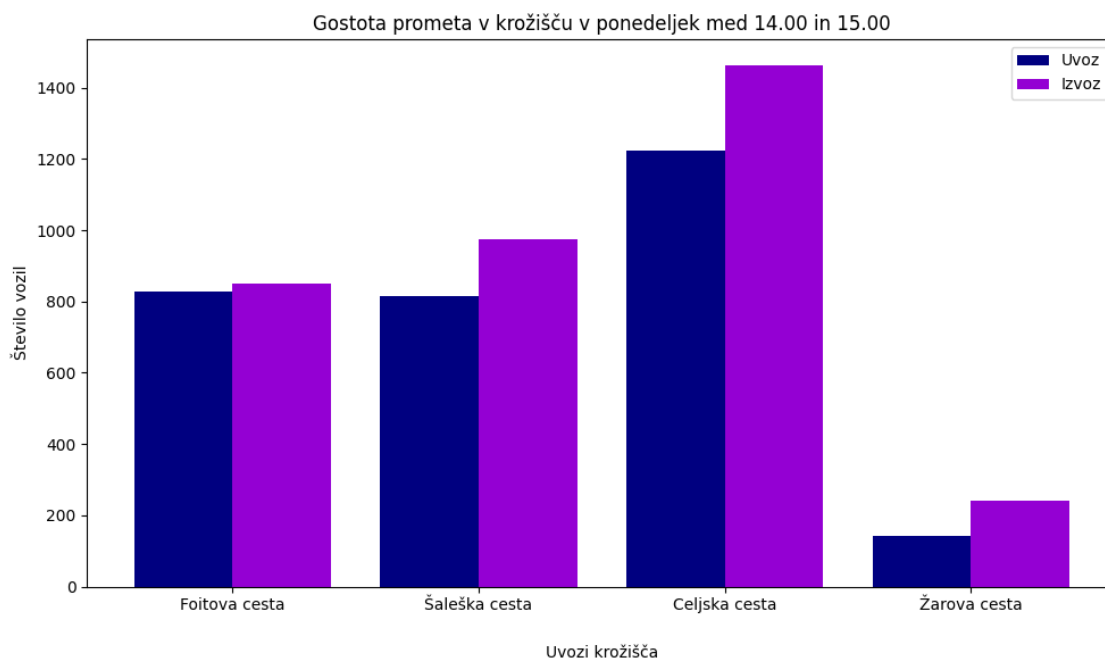
### ***7.1 MERITVE GOSTOTE PROMETA***

Meritve sem opravljal v ponedeljek in petek ob treh različnih urah, na vseh krakih krožnega križišča. Zanimala me je predvsem največja gostota prometa na določenem časovnem intervalu ene ure, ki bi jo lahko nato uporabil v simulaciji.

V ponedeljek med 14.00 in 15.00 se skozi krožišče v povprečju pelje približno 3000 vozil, če upoštevamo meritve, ki so bile narejene pri uvozu v krožišče. Meritve posameznih krakov namreč niso bile izvedene v istem časovnem obdobju, zato prihaja do odstopanja med številom vozil, ki so se ob določenem časovnem intervalu vključevali, in številom vozil, ki so iz križišča odhajali.

Namen izvajanja teh meritev je bil, poleg merjenja gostote števila vozil, tudi razmerje med vsemi uvozi in izvozi, saj me je zanimalo kateri kraki so najbolj obremenjeni. Tako lahko denimo iz ponedeljkovih meritev, ob drugi uri, izračunamo razmerje obremenjenosti vsakega izmed krakov.

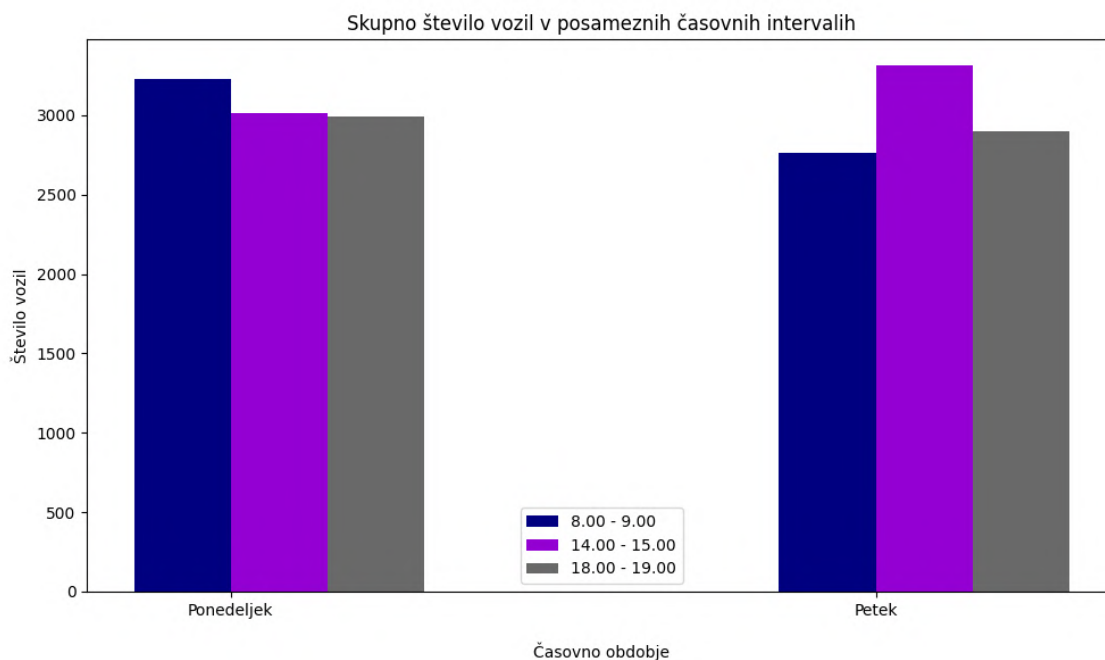
Iz meritev, ki so bile storjene na ponedeljek, med drugo in tretjo uro popoldan, lahko ugotovimo, da je najbolj obremenjen krak Celjske ceste, skozi katerega, na danem intervalu, se v povprečju pelje približno 41 % celotne gostote prometa. Najmanj obremenjen pa je, pričakovano, krak Žarove ceste s približno 6 % gostote celotnega prometa krožnega križišča.



**Graf 1: Podatki o številu vozil v ponedeljek med 14.00 in 15.00**

Zatem, ko sem izračunal približno gostoto prometa na posameznem kraku, sem potreboval še največje število vozil, ki se v določenem intervalu peljejo skozi krožišče. To sem izračunal tako, da sem primerjal število vozil, ki se v krožišče vključujejo, v šestih različnih časovnih obdobjih.

Iz teh rezultatov ugotovimo, da je bila največja gostota prometa v petek ob dveh popoldne in sicer približno 3300 vozil.



**Graf 2: Gostota prometnega toka v različnih časovnih obdobjih**

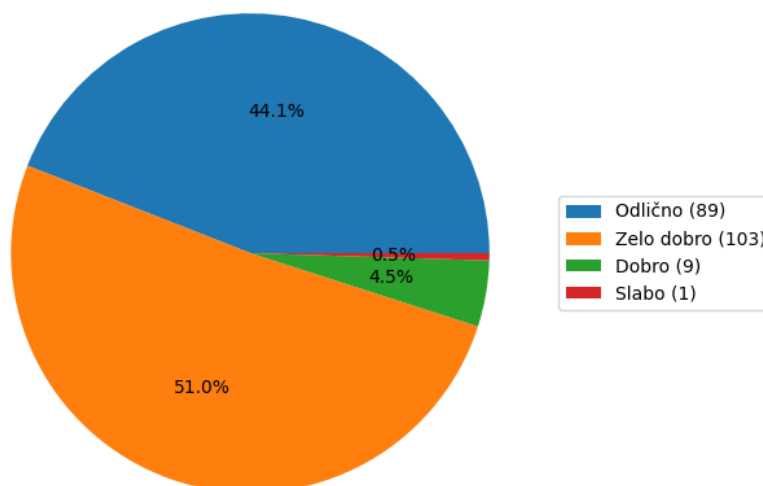
## ***7.2 ANALIZA ODGOVOROV ANKETE***

Na podlagi ankete, na katero je odgovorilo kar 212 anketirancev, od katerih je veljavni vozniški izpit imelo 202 anketirancev, sem naredil analizo rezultatov. Pri tem sem pri analizi rezultatov upošteval le odgovore tistih, ki so imeli vozniški izpit. V rezultatih bom predstavil analizo rezultatov ankete, ki sem jih razdelil v tri kategorije z podvprašanji.

### **7.2.1 POZNAVANJE CESTNO PROMETNIH PREDPISOV**

Na podlagi vprašanja, kjer so morali anketiranci podati oceno njihovega poznavanja cestno prometnih predpisov, pri čemer 1 predstavlja zelo slabo poznavanje in 5 odlično, sem ugotovil, da je kar 44 % vseh anketirancev, ki menijo, da je njihovo poznavanje cestno prometnih predpisov odlično. Poleg tega je več kot polovica anketirancev mnenja, da je njihovo poznavanje predpisov zelo dobro.

Kako bi ocenil svoje znanje cestno prometnih predpisov?



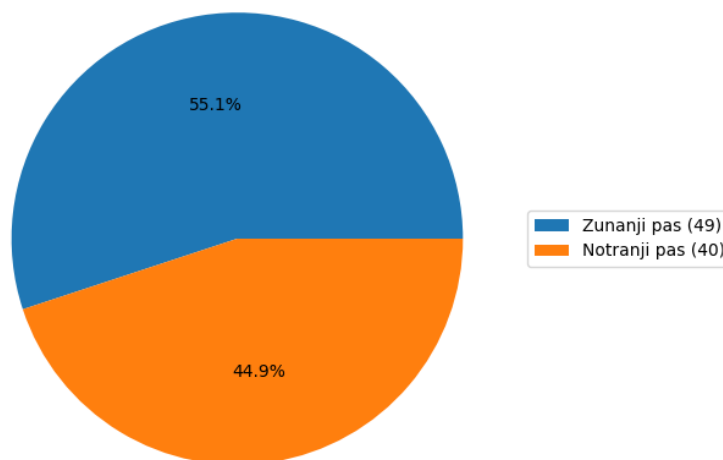
**Graf 3: Prikaz rezultatov ocene poznavanja cestno prometnih predpisov**

Če izluščimo vse, ki so mnenja, da je njihovo poznavanje predpisov odlično, lahko to preverimo na konkretnem primeru vključevanja v dvopasovno križišče, kjer so morali anketiranci v dani situaciji (Slika 17) izbrati prometni pas, po katerem se bodo vključevali v promet, če bi krožišče zapustili pri drugem izvozu. Pri tem bi morali, če bi se ravnali po cestno prometnih predpisih, izbrati notranji pas, saj le-ta omogoča največjo pretočnost krožišča, kjer se lahko hkrati v promet vključujejo tudi drugi udeleženci.

A vendar temu ni tako, saj bi jih kar 55 % izbralo pas označen s številko 2, ki v dani situaciji prikazuje zunanji pas. To pomeni, da se jih večina ne ravna po prometnih predpisih, pri čemer je njihovo odlično poznavanje le-teh pod vprašanjem.



Izbira prometnega pasu pred vključitvijo v dvopasovno krožišče



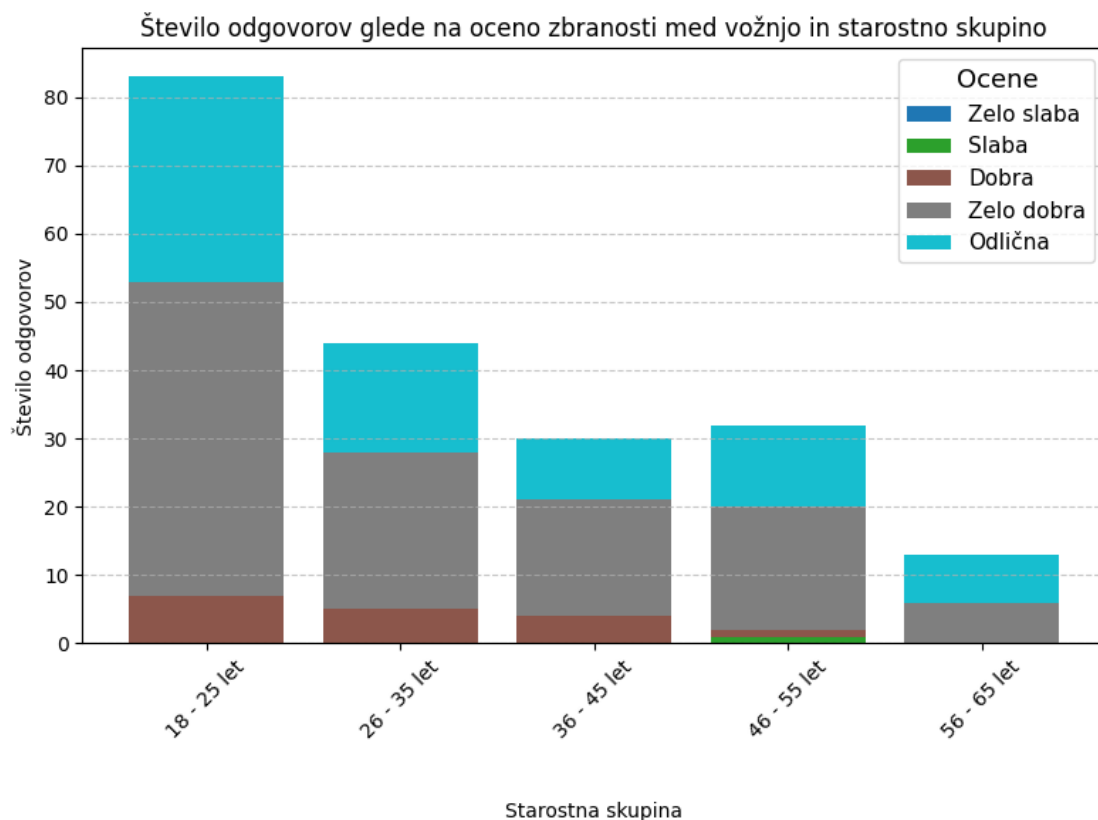
Graf prikazuje le anketirance, ki so svoje znanje CPP ocenili z odlično

**Graf 4: Prikaz rezultatov izbire prometnega pasu v krožišče na dani situaciji**

### 7.2.2 ZBRANOST V PROMETU

Največ anketirancev je na vprašanje, kjer so morali oceniti svojo zbranost med vožnjo, ocenilo, da je med vožnjo njihova zbranost zelo dobra ali odlična. Poleg tega pa jih 9 % meni, da je njihova zbranost dobra oziroma slaba. Največji odstotek teh je v starostni skupini med 36 in 45 let.

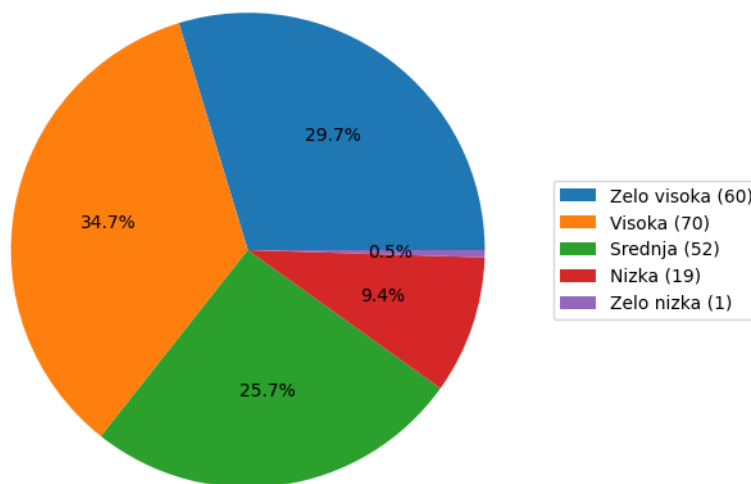
Zanimivo, da so čisto vsi anketiranci v starostni skupini med 56 in 65 leti mnenja, da je njihova zbranost zelo dobra ali odlična, od tega jih je bilo kar 54 %, ki so ocenili odlično zbranost med vožnjo v prometu.



**Graf 5: Ocena zbranosti voznikov v prometu**

Pa vendar se jih, od tega kar 54 % anketirancev, med vožnjo pogosto pogovarja s sopotniki, kar lahko posledično konkretno zmanjša zbranost v prometu. Od tega se kar 30 % anketirancev med vožnjo vedno pogovarja s sopotnikom, če je le-ta prisoten.

Pogostost pogovarjanja s sopotniki med vožnjo

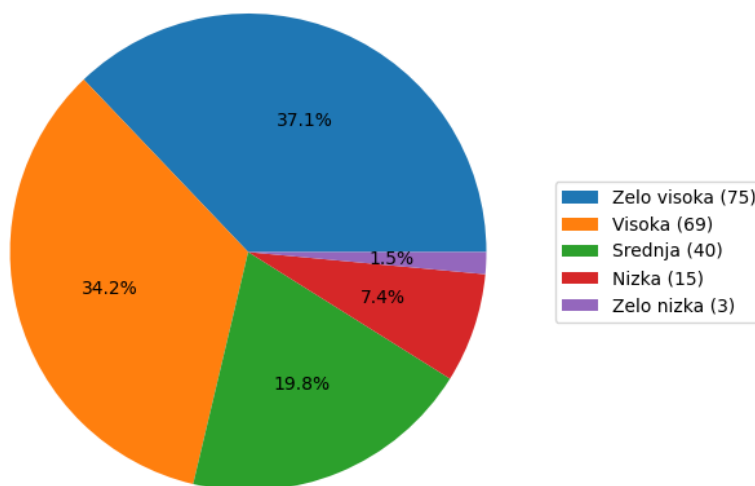


**Graf 6: Pogosto pogovarjanje voznikov s sopotniki**

Pri tem jih je kar 71 % ocenilo, da imajo visoko pripravljenost na speljevanje pri rdeči luči. Od tega pa jih je 37 % mnenja, da je njihova pripravljenost ob čakanju izredno visoka, kar naj bi pomenilo, da pozornost posvečajo le temu, kdaj se bo rdeča luč spremenila v zeleno, da bodo lahko, kar se da, hitro speljali.

Vsi tisti, ki imajo pri tem nizko ali zelo nizko pripravljenost, 9 % vseh anketirancev, pa so razlog, da je pojav efekta »shockwave« toliko bolj opazen.

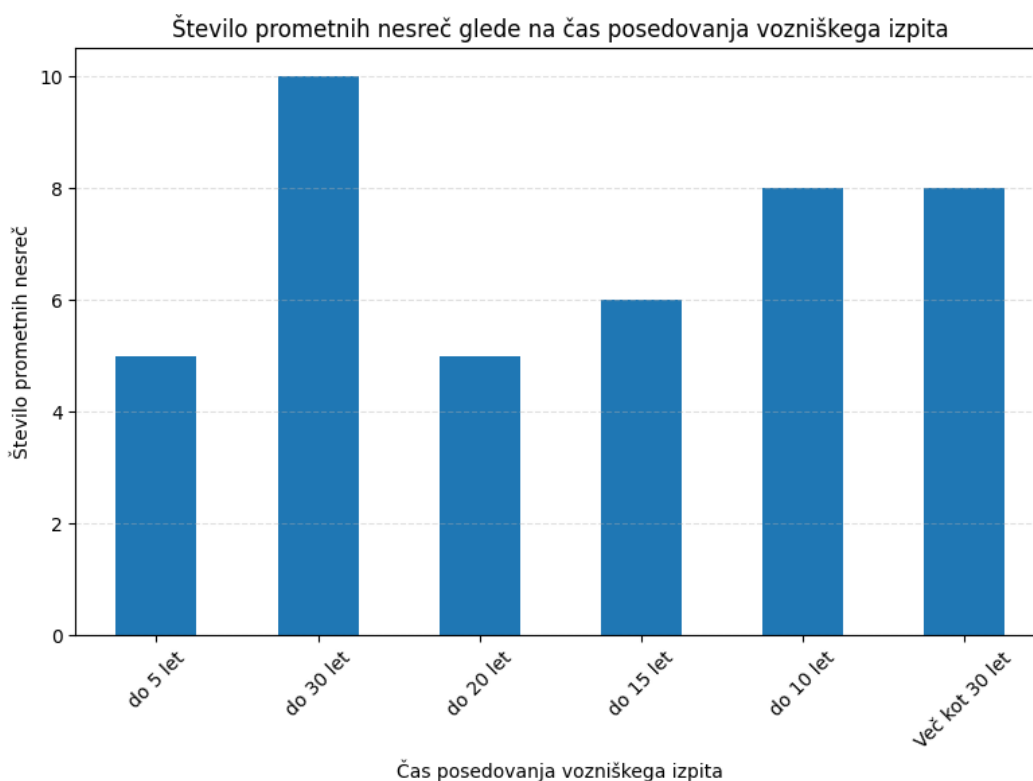
Nivo pripravljenosti na speljevanje pri rdeči luči



Graf 7: Pripravljenost voznikov na speljevanje pri rdeči luči

Pri tem me je zanimalo tudi, če so bili že kdaj krivi za nastanek prometne nesreče, saj je največkrat razlog za nastanek le-teh, pomanjkanje zbranosti voznikov v prometu. Rezultati so zelo zanimivi, saj na podlagi odgovorov, nihče izmed tistih, ki so še voznik začetnik, ni bil povzročitelj nesreče do sedaj, pa čeprav bi lahko pričakovali, da bi lahko bilo krivo pomanjkanje izkušenj.

Od vseh anketirancev jih je bilo kar 21 %, od tega največ tistih, kar 2,4 %, ki imajo vozniški izpit med 20 in 30 let.



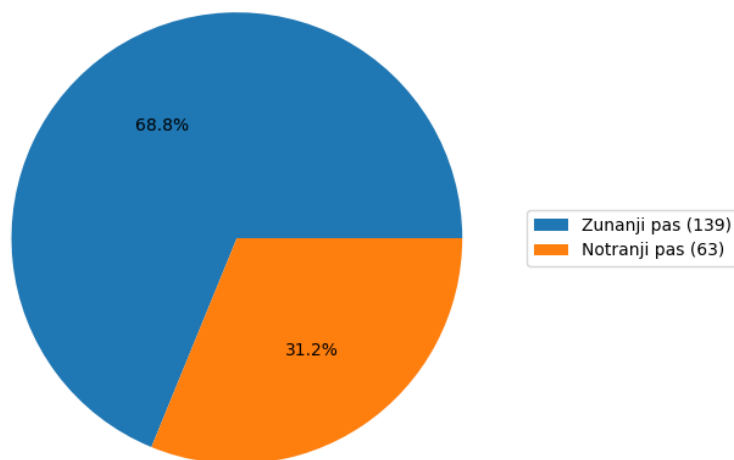
Graf 8: Število povzročiteljev prometnih nesreč

### 7.2.3 PRETOČNOST PROMETA

Na primeru dane situacije iz resničnega življenja (Slika 18), kjer se želimo vključiti v dvopasovni krožni promet, ko se pred nami že vključuje vozilo, medtem ko oba želita zapustiti krožišče v drugem izvozu, sem anketirancem postavil vprašanje, kateri pas bi izbrali za vključitev.

Kar 69 % bi se jih v tej situaciji razvrstilo na zunanji pas, da bi se izognili tega, kjer bi jih, pri razvrščanju na zunanji pas iz notranjega, oviral avtomobil, s katerim bi se vključili v promet. To nam na realnem primeru prikaže značilnosti Braessovega paradoksa, kjer jih naposled večina začne uporabljati zunanji vozni pas, pa čeprav to posledično pomeni manjšo pretočnost in povečanje potovalnega časa celotnega omrežja.

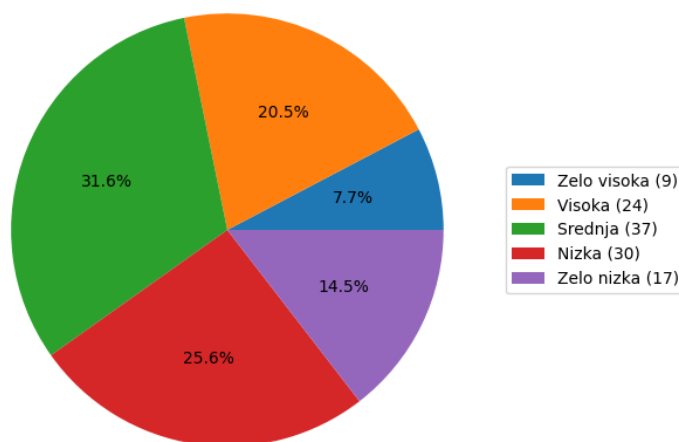
Izbira prometnega pasu pred vključitvijo v dvopasovno krožišče, kjer se spredaj že vključuje vozilo



**Graf 9: Rezultati izbire prometnega pasu iz druge dane situacije**

Hkrati pa je zanimivo dejstvo, da je kar 40 % anketirancev, ki bi pri prvi dani situaciji (Slika 17), kjer se v krožno križišče vključujejo sami z namenom izhoda na drugem izvozu, izbrali vključitev preko zunanjega prometnega pasu, mnenja, da ima krožno križišče v Velenju nizko pretočnost. Pa čeprav je to eden izmed ključnih razlogov za zmanjšanje pretočnosti dvopasovnih krožnih križišč.

Ocena pretočnosti dvopasovnega krožišča v Velenju



Graf prikazuje le anketirance, ki bi se, v dvopasovno krožišče, vključevali po zunanjem pasu za izhod na drugem izvozu

**Graf 10: Rezultati mnenja glede pretočnosti Velenjskega krožnega križišča**

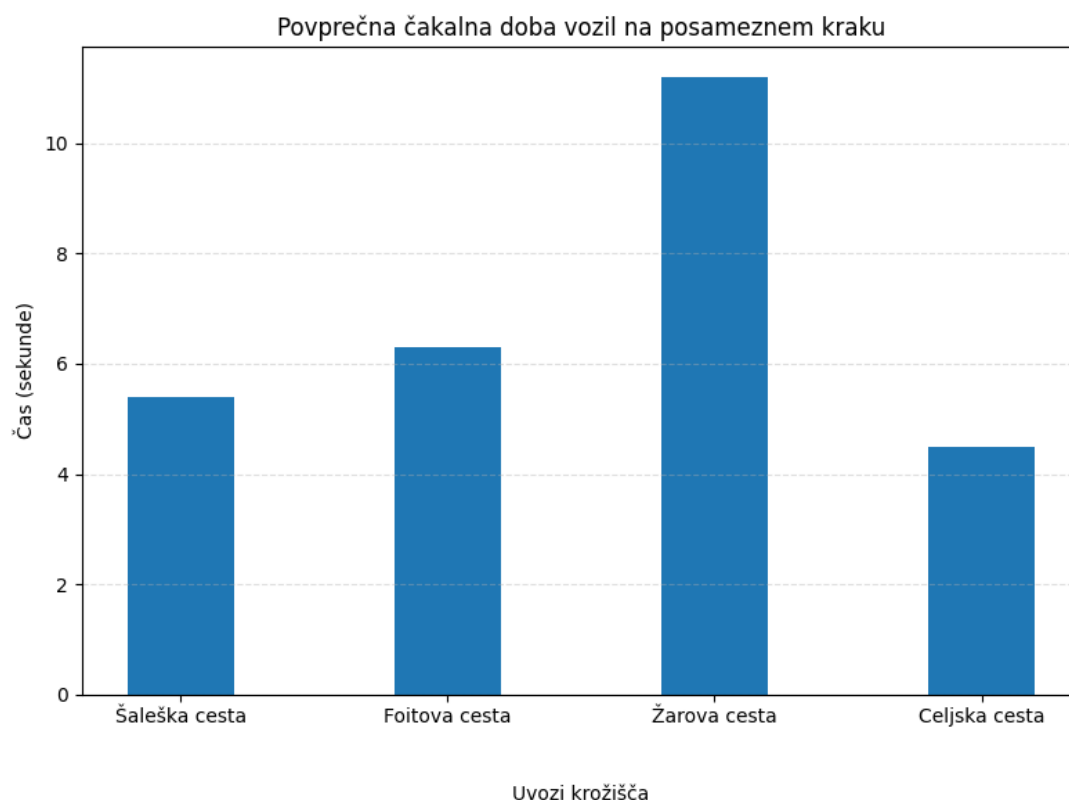
### **7.3 SIMULACIJA**

S pomočjo izdelane simulacije, sem nato lahko simuliral pretočnost prometa skozi dano krožno križišče, na podlagi izmerjenih podatkov gostote prometa krožnega križišča v Velenju. Simulaciji sem lahko določil začetno vrednost N števila vozil, pri katerem je vsako izmed njih imelo določeno začetno in končno pozicijo.

Kljub temu da sem s pomočjo simulacije lahko izničil človeški faktor, ki močno vpliva na potek prometa v realnem življenju, sem še vedno naletel na zamašitev krožišča zaradi dominacije določene relacije, na konkretnem primeru, relacija Foitova cesta – Celjska cesta. V moji poenostavljeni simulaciji Velenjskega krožnega križišča je omenjena relacija povzročila otežen vstop vozil na uvozu Žarove ceste, saj je bilo razmerje preobremenjenosti krakov premajhno. S tem je bil prometni tok skorajda neprekinjen iz vseh ostalih treh uvozov, pri čemer so bile čakalne dobe za vključevanje v krožišče minimalne.

Rezultati te simulacije so privedli do ugotovitve, da je potrebno za uravnoteženo pretočnost na vseh uvozih krožnega križišča, začasno zaustaviti promet iz tistih krakov, ki imajo preveliko gostoto in posledično omejujejo morebitno vključitev ostalih uvozov.

Uvedba semaforizirane signalizacije na vseh uvozih, ki je po potrebi zaustavljala vključitev v krožišče, če so bile čakalne dobe na ostalih uvozih predolge. S sprotnim računanjem čakalnih dob vozil na posameznih uvozih, sem lahko optimiziral prometno omrežje tako, da so se čakalne dobe, med vozili na različnih uvozih porazdelile, pri čemer je povprečna čakalna doba vozila, od trenutka, ko se je krožnemu križišču približal, do vključitve v krožni tok, znašala 6,9 sekund v celotnem prometnem omrežju. Največje odstopanje čakalne dobe za vključitev v krožni tok, med prometno bolj gostim ter manj gostim uvozom pa je znašalo približno 6,7 sekund, kjer je bila povprečna čakalna doba vozila za vključitev na uvozu Žarove ceste v simulaciji približno 11,2 sekunde.



**Graf 11: Povprečna čakalna doba vozil za vključitev v krožišče**



## 8 RAZPRAVA

Glede na dane rezultate lahko sedaj potrdim, oziroma ovržem izbrane hipoteze.

Prvo hipotezo, da širjenje prometnih povezav poveča pretočnost prometa in posledično zmanjša potovalni čas od točke A do B, **OVRŽEM**.

Primer simulacije na prometnem omrežju, kjer dodamo dodatno relacijo, potrdi teorijo Braessovega paradoksa, kjer dodajanje novih povezav pretočnost zmanjša in ne poveča. Kljub temu da so primeri, kjer bi nove relacije zmanjšale končni potovalni čas, to ne velja za večino primerov. Eden izmed teh primerov je tudi dvopasovno krožno križišče v Velenju, kjer sem s pomočjo analize in rezultatov ankete ugotovil, da zaradi nepravilne uporabe dvopasovnega krožišča, dodatni prometni pas prometni tok zmanjša in ne poveča.

Drugo hipotezo, da je optimizacija pretočnosti lokalnega prometnega krožnega križišča možna brez rekonstrukcije, **POTRDIM**.

Čeprav je načinov za optimizacijo lokalnega krožnega križišča več, nekateri izmed teh so tudi poseg v konstrukcijo samega krožnega križišča, sem s pomočjo analize in simulacije ugotovil, da to za povečanje pretočnosti ni nujno potrebno. Dva izmed načinov, na tem specifičnem primeru, sta denimo semaforizirana signalizacija na uvozih v krožišče in povečanje oddaljenosti prehodov za pešce, na uvozih in izvozih, od krožnega toka.

Tretjo hipotezo, da lahko ustvarimo računalniško simulacijo poljubnega prometnega omrežja, ki omogoči prikaz optimiziranega prometnega toka brez človeškega vpliva, **POTRDIM**.

S pomočjo izdelane računalniške simulacije, sem omogočil prikaz prometnega toka na poljubnem prometnem omrežju, kjer sam človek nima neposrednega vpliva na prometni tok. S tem sem izničil možnost morebitnega trčenja vozil zaradi človeške malomarnosti, zavajanja ostalih prometnih udeležencev z uporabo nepravilne signalizacije, povečal pretočnost prometa zaradi odstranitve reakcijskega časa pri odzivnosti v prometu. Ob vsem tem pa sem lahko prikazal, kako bi lahko v teoriji izgledalo prometno omrežje, če

bi bili edini udeleženci v prometu, samovozeča vozila, katerih namen bi bil čim večja pretočnost celotnega omrežja in ne le sebična izbira najhitrejše poti posameznika.

## 9 ZAKLJUČEK

Promet je ena izmed ključnih sestavin sodobnega urbanega življenja, vendar se soočamo z izzivi, kot so zastoji, podaljšan čas potovanja in negativni vplivi na okolje. V zadnjih letih so se razvile številne tehnološke rešitve, ki ponujajo možnosti za izboljšanje prometnega sistema, vendar se s tem v družbi odpirajo tudi vprašanja glede učinkovitosti in sprejemljivosti teh rešitev.

Avtomatizacija prometa se izkaže za optimizacijo prometnega toka kot obetavna rešitev. S prenosom nadzora na računalniške algoritme bi lahko zmanjšali vpliv človeških napak in posledično pretočnost prometa izboljšali. To bi lahko pripomoglo k zmanjšanju zastojev, krajšemu času potovanja in bi imelo tudi pozitiven vpliv na okolje.

Spremembe v prometnem sistemu zahtevajo premišljen pristop, saj lahko le tako dosežemo bolj učinkovit, varnejši in trajnostno naravnani prometni sistem, ki bo ustrezal potrebam sodobne družbe in hkrati ohranjal ravnovesje med tehničnim napredkom in človeškimi vrednotami.

Skozi raziskovanje sem ugotovil, da smo ravno mi, ljudje, glavni krivec za nizko pretočnost prometnih omrežij in nastanek prometnih zastojev, saj s svojo nepremišljenostjo in malomarnostjo vplivamo na ostale udeležence v prometu, saj smo ljudje in nismo popolni in delamo napake. Poleg tega, pa je ogromno odvisno tudi od same psihofizične pripravljenosti vsakega izmed voznikov, med katerimi se ta konkretno razlikuje. Ravno to je razlog zakaj ni rešitve, ki bi v celoti izničila morebitne zastoje ali dolge potovalne čase. Pri tem pa je potrebno upoštevati, da za majhno pretočnost ni vedno krivo le eno vozlišče, temveč prometno omrežje kot celota.

## **10 ZAHVALA**

Iskreno bi se rad zahvalil svojemu mentorju Alešu Spitalu za pomoč in podporo pri sami raziskavi ter Nataši Meh Peer in Sonji Lubej za lektoriranje raziskovalne naloge.

Zahvalil bi se rad vsem tistim, ki odgovorili na mojo anketo. Prav tako pa bi rad svojo hvaležnost izkazal vsem, ki so me pri samem raziskovanju na kakršen koli način podpirali.

## 11 VIRI IN LITERATURA

- [1] Downs, Anthony (2004). "Traffic: Why It's Getting Worse, What Government Can Do". Brookings. Dostopno na: <https://www.brookings.edu/articles/traffic-why-its-getting-worse-what-government-can-do/>
- [2] Kolata, Gina (1990). "What if They Closed 42d Street and Nobody Noticed?". New York Times. Retrieved 16 November 2008. Dostopno na: <https://www.nytimes.com/1990/12/25/health/what-if-they-closed-42d-street-and-nobody-noticed.html>
- [3] Boyd, Andrew. "Braess' Paradox". Engines of Our Ingenuity. Episode 2814. Dostopno na: <https://engines.egr.uh.edu/episode/2814>
- [4] Milchtaich, Igal (2006). "Network topology and the efficiency of equilibrium". Games and Economic Behavior. 57 (2): 321–346. Dostopno na: <https://www.sciencedirect.com/science/article/abs/pii/S0899825605001284?via%3Dihub>
- [5] Myerson, Roger B. (1991). Game Theory: Analysis of Conflict. Harvard University Press.
- [6] Dal Bó, Pedro; Fréchette, Guillaume R. (2019). "Strategy Choice in the Infinitely Repeated Prisoner's Dilemma". American Economic Review. 109 (11): 3929–3952. Dostopno na: <https://www.aeaweb.org/articles?id=10.1257/aer.20181480>

[7] Axelrod, Robert (1980). "Effective Choice in the Prisoner's Dilemma". *The Journal of Conflict Resolution*. 24 (1): 3–25. Dostopno na:

<https://journals.sagepub.com/doi/10.1177/002200278002400101>

[8] Cornell University (2017). »Braess' Paradox – The Cheonggyecheon Restoration Project«. Dostopno na:

<https://blogs.cornell.edu/info2040/2017/09/14/braess-paradox-the-cheonggyecheon-restoration-project/>

[9] Cornell University (2017). »Braess's Paradox in South Korea«. Dostopno na:

<https://blogs.cornell.edu/info2040/2017/10/17/braess-paradox-in-south-korea/>

[10] Washington Faculty (2017). »Traffic flow: the Lighthill-Whitham-Richards model«. Dostopno na:

[https://faculty.washington.edu/rjl/riemann\\_book/Traffic\\_flow.html](https://faculty.washington.edu/rjl/riemann_book/Traffic_flow.html)

[11] Giovanni De Nunzio, Per-Olof Gutman (2017). »An application of shock wave theory to urban traffic control via dynamic speed advisory«. Dostopno na:

[https://transp-or.epfl.ch/heart/2017/abstracts/hEART2017\\_paper\\_184.pdf](https://transp-or.epfl.ch/heart/2017/abstracts/hEART2017_paper_184.pdf)

[12] Carmona, Guilherme; Podczeck, Konrad (2009). »On the Existence of Pure Strategy Nash Equilibria in Large Games«. *Journal of Economic Theory*. 144 (3): 1300–1319. Dostopno na:

<https://www.sciencedirect.com/science/article/abs/pii/S0022053108001695?via%3Dihub>

b

[13] Schelling, Thomas (1980). »The Strategy of Conflict«, Harvard University Press.

[14] Wikipedia. »An example prisoner's dilemma payoff matrix«. Dostopno na:  
[https://en.wikipedia.org/wiki/Prisoner%27s\\_dilemma#/media/File:Prisoners\\_dilemma.s  
vg](https://en.wikipedia.org/wiki/Prisoner%27s_dilemma#/media/File:Prisoners_dilemma.svg)

[15] Bily, Barbara (2016). »The method of finding points of intersection of two cubic bezier curves using the sylvester matrix«. Silesian University of Technology. 6 (1): 155-176. Dostopno na:  
[https://mat.polsl.pl/sjpam/zeszyty/z6/Silesian\\_J\\_Pure\\_Appl\\_Math\\_v6\\_i1\\_str\\_155-  
176.pdf](https://mat.polsl.pl/sjpam/zeszyty/z6/Silesian_J_Pure_Appl_Math_v6_i1_str_155-176.pdf)

[16] Melo, Mateus (2021). »Understanding Bézier Curves«. Dostopno na:  
<https://mmrndev.medium.com/understanding-b%C3%A9zier-curves-f6eaa0fa6c7d>

[17] Google Zemljevidi (2024). Dostopno na: <https://www.google.com/maps>

[18] Geopedia (2021). Dostopno na: <http://www.geopedia.si/>

[19] Lešnik, Anže (2016). »Rekonstrukcija dvopasovnega krožnega križišča v Velenju«. Dostopno na: <https://dk.um.si/Dokument.php?id=85913>

[20] Spletni portal Vozim se (2019). »Vse, kar morate vedeti, da vas noben rondo ne bo zmedel«. Dostopno na: [https://vozimse.si/cpp/vse-kar-morate-vedeti-da-vas-noben-  
rondo-ne-bo-zmedel/](https://vozimse.si/cpp/vse-kar-morate-vedeti-da-vas-noben-rondo-ne-bo-zmedel/)

[21] Javna agencija Republike Slovenije za varnost prometa (2020). »Pravila vožnje na krožnem križišču«. Dostopno na: <https://www.avp-rs.si/pravila-voznje-na-kroznem-kriziscu>

[22] Wikipedia. »International Road Network«. Dostopno na: [https://upload.wikimedia.org/wikipedia/commons/6/60/International\\_E\\_Road\\_Network\\_green.png](https://upload.wikimedia.org/wikipedia/commons/6/60/International_E_Road_Network_green.png)

[23] City Of Ottawa (2019). »How to use a roundabout«. Dostopno na: <https://www.youtube.com/watch?v=46mOPz3rhHs>



## PRILOGE

### Priloga 1: Anketa

## Analiza in optimizacija prometa v Velenju (raziskovalna naloga)

Anketa je namenjena zbiranju podatkov o pretočnosti prometa v lokalnem okolju, z osredotočenostjo na dvopasovno krožišče. Cilj je pridobiti vpogled v izkušnje voznikov in udeležencev v prometu ter oceniti morebitne težave ali izzive pri prehodu skozi krožišče.

Ob tem, pa je namen raziskave ta, da bi pretočnost prometa čimbolj povečali.

Raziskava poteka v sklopu srednješolske raziskovalne naloge, zato so vaši odgovori zelo zaželeni.

*\* Nakazuje obvezno vprašanje*

### Dvopasovno krožišče v Velenju



1. Ali imaš veljaven vozniški izpit? \*

*Označite samo en oval.*

Da

Ne

2. V katero starostno skupino spadaš? \*

*Označite samo en oval.*

- 18 - 25 let  
 26 - 35 let  
 36 - 45 let  
 46 - 55 let  
 56 - 65 let  
 Več kot 65 let

3. Koliko časa že imaš vozniški izpit? \*

*Označite samo en oval.*

- Sem voznik začetnik  
 do 5 let  
 do 10 let  
 do 15 let  
 do 20 let  
 do 30 let  
 Več kot 30 let

Neimenovan razdelek

4. Kako bi ocenil svoje poznavanje cestno prometnih predpisov? \*

1 - Zelo slabo, 5 - Odlično

*Označite samo en oval.*

- 1  
 2  
 3  
 4  
 5

5. Kako pogosto bi ocenil, da se udeležuješ prometa? \*

1 - Skoraj nikoli, 5 - Vsak dan

*Označite samo en oval.*

1

2

3

4

5

6. Kako bi ocenil svojo zbranost med vožnjo? \*

1 - Zelo slabo, 5 - Odlično

*Označite samo en oval.*

1

2

3

4

5

7. Kako pogosto se med vožnjo pogovarjaš s sopotniki? \*

1 - Nikoli, 5 - Vedno

*Označite samo en oval.*

1

2

3

4

5

8. Ali si pri rdeči luči na semaforiziranem križišču v pripravljenosti na speljevanje? \*

1 - Ne posvečam preveč pozornosti  
5 - Ves čas v nizkem startu

Označite samo en oval.

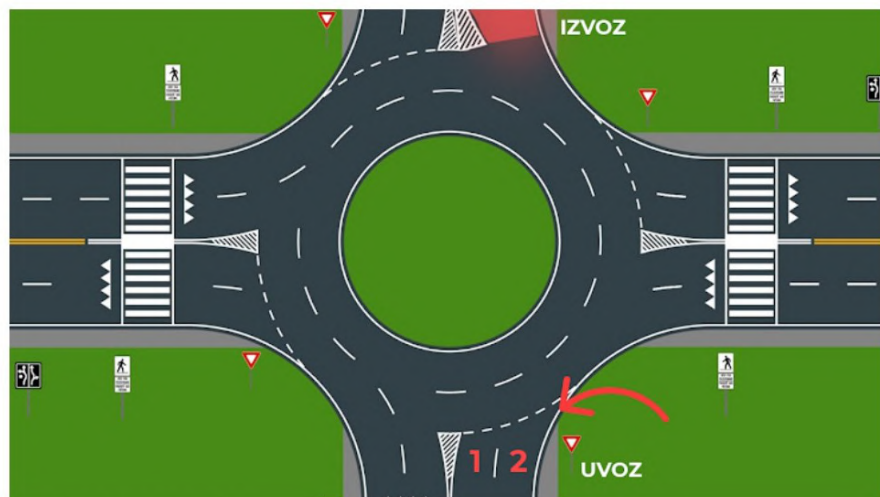
- 1  
 2  
 3  
 4  
 5

9. Ali si že imel prometno nesrečo, v kateri si bil krivec ti? \*

Označite samo en oval.

- Da  
 Ne

10. Na kateri vozni pas bi se razvrstil pred vključitvijo v dvopasovno krožišče, če bi krožišče zapustil pri drugem izvozu? \*

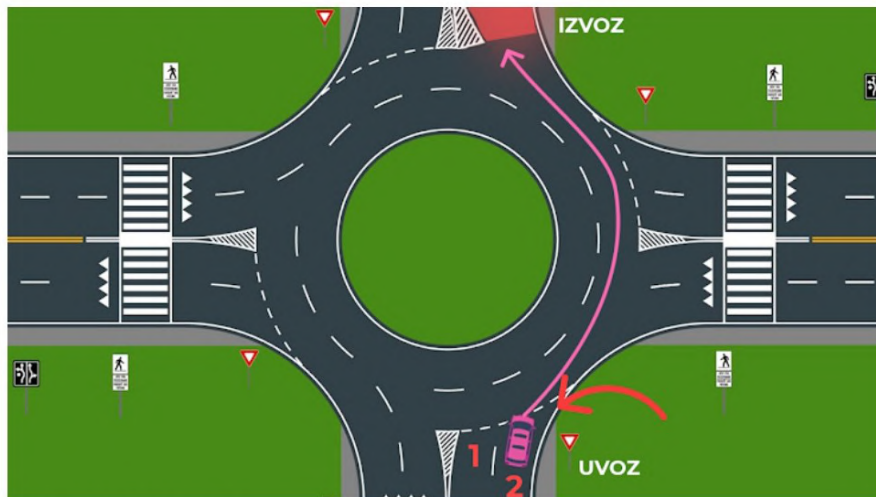


Označite samo en oval.

- Pas št 1  
 Pas št 2

11. Na kateri vozni pas bi se pa razvrstil v tem primeru, če bi rad krožišče zapustil v drugem izvozu, če se pred tabo že vključuje voznik roza vozila, ki bo prav tako zapustil krožišče v drugem izvozu? \*

Tukaj moramo predpostavljati, da pred vključitvijo v krožišče vemo, kam je avtomobil pred nami namenjen. V primeru, da izberemo prvi vozni pas, nam bo izvoz onemogočen, saj se ne bomo mogli pravilno razporediti in bomo posledično morali narediti dodaten krog.



Označite samo en oval.

- Pas št 1, vzporedno z roza avtomobilom
- Pas št 2, za roza avtomobilom

12. Kako bi ocenil zadovoljstvo z največjim dvopasovnim krožiščem v Velenju? \*

1 - Sploh mi ni všeč  
5 - Zelo mi je všeč

Označite samo en oval.

- 1
- 2
- 3
- 4
- 5

13. Kako bi ocenil pretočnost dvopasovnega krožišča v Velenju? \*

1 - Sploh ni pretočno

5 - Zelo pretočno

*Označite samo en oval.*

1

2

3

4

5