SVEUČILIŠTE U MOSTARU

FAKULTET STROJARSTVA, RAČUNARSTVA I ELEKTROTEHNIKE

SEMINARSKI RAD

SIMULACIJA RADA SERVISA „BRZI TOČAK“ ZA POPRAVAK BICIKALA U NETLOGU

Hrvoje Sesar  
Antonio Šego

Mostar, lipanj 2024.

# Sadržaj

[1 Uvod 1](#_Toc168413306)

[2 Opis problema 2](#_Toc168413307)

[3 Metodologija 3](#_Toc168413308)

[3.1 Korišteni alati i tehnologije 3](#_Toc168413309)

[3.2 Definiranje agenata 3](#_Toc168413310)

[3.3 Parametri simulacije 3](#_Toc168413311)

[3.4 Raspored dolazaka uređaja 4](#_Toc168413312)

[3.5 Model simulacije 4](#_Toc168413313)

[3.6 Validacija modela 5](#_Toc168413314)

[4 Prikaz koda i funkcionalnosti simulacije 7](#_Toc168413315)

[5 Zaključak 21](#_Toc168413316)

# Popis slika

Slika 4.1. Deklaracija varijabli okruženja simulacijskog modela ...................................................7

Slika 4.2. Postavljanje modela simulacije .....................................................................................8

Slika 4.3. Postavljanje simulacije servisa .....................................................................................9

Slika 4.4. Kreiranje novog klijenta unutar simulacije ..................................................................10

Slika 4.5. Pokretanje simulacije ..................................................................................................11

Slika 4.6. Funkcija za pronalaženje najbližeg slobodnog servisera ............................................13

Slika 4.7. Logika ponašanja klijenta unutar simulacije (1.dio) ....................................................14

Slika 4.8. Logika ponašanja klijenta unutar simulacije (2.dio) ....................................................14

Slika 4.9. Prikaz proceudra za upravljanje brojem aktivnih servisera .........................................16

Slika 4.10. Pirkaz stvaranja novog servisera unutar simulacije ..................................................17

Slika 4.11. Prikaz generiranja dolaska klijenta u servisni centar ................................................18

Slika 4.12. Prikaz funkcije za kraj radog vremena i za spremanje CSV datoteke ......................19

Slika 4.13. Prikaz simulacijskog riješenja ...................................................................................20

# Uvod

U današnjem užurbanom životu, bicikli su postali važan dio svakodnevnice za mnoge ljude, bilo da ih koriste za prijevoz, rekreaciju ili sport. Međutim, kao i svako drugo sredstvo za kretanje, bicikli su skloni kvarovima i oštećenjima koji zahtjevaju stručne popravke. Servisi za popravak bicikala igraju ključnu ulogu u održavanju bicikala u ispravnom stanju omogućujući njihovim vlasnicima da se nesmetano koriste svojim dvokotačima.

Upravljanje servisom za popravak bicikala može biti izazovno, posebno kada je potrebno optimizirati radnu snagu i resurse kako bi se osigurala učinkovita usluga i maksimizirala zarada. Servis „Brzi Točak“ suočava se sa tim izazovima svakodnevno, nastojeći pronaći ravnotežu između broja servisera na raspolaganju i broja klijenata koji dolaze sa različitim vrstama bicikala.

U ovom radu opisujemo simuaciju rada servisa za popravak bicikala pomoću NetLogo programskog jezika. Cilj je razviti model koji će nam omogućiti testiranje različitih strategija zapošljavanja servisera kako bismo maksimizirali zaradu servisa uz istovremeno održavanje visoke kvalitete usluge i smanjenje vremena čekanja za klijente.

U nastavku ćemo detaljnije razmotriti problematiku koja se tiče servisa za popravak bicikala, objasniti metodologiju korištenu za simulaciju, modelirati rad servisa koristeći NetLogo, te analizirati rezultate simulacije kako bismo dobili uvid u optimalne strategije upravljanja servisom „Brzi Točak“.

# Opis problema

Servis za popravak bicikala „Brzi Točak“ suočava se s nizom izazova u svakodnevnom poslovanju. Primarni cilj ovog servisa je osigurati brz i kvalitetan popravak bicikala uz istovremeno maksimiziranje zarade. Ključni izazovi sa kojima se suočava servis uključuju:

* **Optimizacija radne snage** – Važno je odrediti optimalan broj servisera potrebnih za obradu dolazaka bicikala u servis, uzimajući u obzir razlicite vrste bicikala i specijalizacije servisera.
* **Upravljanje vremenom** – Potrebno je osigurati da se bicikli popravljaju u razumnom vremenskom roku kako bi se smanjio broj klijenata koji odustaju zbog dugog čekanja
* **Maksimiziranje zarade** – Neophodno je pronaći ravnotežu između troškova rada servisera i prihoda od popravaka kako bi se osigurala profitabilnost poslovanja

U ovom radu fokusiramo se na simulaciju rada servisa za popravak bicikala pomoću NetLogo programskog jezika. Cilj je istražiti različite strategije zapošljavanja servisera i njihovu specijalizaciju za određene vrste bicikala, kako bi smo optimizirali poslovanje servisa. Modeliranje agenata, koji predstavljaju servisere i klijente, omogućava nam da simuliramo različite scenarije i analiziramo njihov utjecaj na ukupnu unčikovitost i zaradu servisa.

# Metodologija

## Korišteni alati i tehnologije

Za simulaciju rada servisa za popravak bicikala koristili smo NetLogo, alat za modeliranje simulaciju složenih sustava pomoću agenata. NetLogo pruža jednostavno sučelje za modeliranje agenata, omogućuje simulaciju složenih scenarija i interakciju izmedju agenata u servisu.

## Definiranje agenata

U našoj simulaciji, definirali smo dvije glavne vrste agenata:

* Serviseri – Serviseri su odgovorni za popravak bicikala. Svaki serviser ima određene vještine i može popravljati samo dvije od tri vrste bicikala (dječi,brdski i trkaći). Njihova efikasnost i dostupnost ključni su za optimizaciju rada servisa. Serviseri mogu biti stalni (aktivni tijekom cijelog radnog dana) ili privremeni (aktivni po potrebi)

Broj aktivnih tahničara može se kontrolirati klizačem (od 0 do 9 serviisera)

* Klijenti – Klijenti predstavljaju bicikla koji dolaze na popravak. Bicikli se razlikuju po vrsti (dječiji,brdski,trkaći), a vrijeme potrebno za popravak ovisi o vrsti bicikala i složenosti kvara. Prosječno vrijeme popravka je 20 minuta.

## Parametri simulacije

Postavili smo nekoliko ključnih parametara za našu simulaciju:

* Plaća servisera: 25 KM po satu
* Prosječno vrijeme popravka uređaja: 20 minuta
* Svaki popravak bicikla naplaćuje se 15 KM
* Odustajanje klijenata: Klijenti koji čekaju duže od 15 minuta odustaju od usluge, što negativno utječe na ukupnu zaradu i reputaciju servisa

## Raspored dolazaka klijenata

Simulirali smo dolazak klijenata u servis tijekom radnog dana u slučajnim vremenima unutar jednog sata:

* Od 08:00 do 11:00 🡪 6 klijenata po satu
* Od 11:00 do 15:00 🡪 12 klijenata po satu
* Od 15:00 do 18:00 🡪 8 klijenata po satu

## Model simulacije

Naš model simulacije obuhvaća nekoliko koraka specifičnih za popravak bicikala u servisu:

1. Inicijalizacija

* Postavljanje početnih uvjeta za simulaciju, uključujući stvaranje servisera i definiranje ulaza i izlaza za klijente.
* Definiranje tipova biicikala koje serviseri mogu popraviti (dječiji,brdski,trkaći).

1. Dodjela bicikala serviserima

* Klijenti se dodjeljuju slobodnim serviserima na temelju vrste bicikla i dostupnosti servisera.
* Ako nema slobodnih servisera koji mogu popraviti određeni bicikl, stvara se novi privremeni serviser.

1. Popravak bicikla

* Serviseri popravljaju bicikla prema vrsti kvara i vremenu potrebnom za popravak, koje varira od 1 do 30 minuta.
* Klijenti koji čekaju duže od 15 minuta odlaze iz servisa bez popravka, što se bilježi kao neuspješan popravak.

1. Praćenje zarade i troškova

* Evidentiranje praćenja zarade od popravaka i troškova rada serviseri.
* Praćenje broja uspješnih i neuspješnih popravaka.
* Ukupna zarada, troškovi i profit prate se pomoću globalnih varijabli, a rezultati se pohranjuju u Excel datoteku na kraju radnog dana.

1. Optimizacija broja servisera

* Siulacija automatski dodaje nove servisere kada je potražnja visoka i uklanja ih kada više nisu potrebni.
* Dinamička prilagodba broja servisera osigurava optimalnu efikasnost rada servisa, smanjujući troškove rada i vrijeme čekanja klijenata.

## Validacija modela

Radi osiguranja preciznosti i pouzdanosti našeg modela, proveli smo niz simulacijskih testova. Provjerili smo rezultate simulacije uspoređujući ih sa očekivanim rezultatima na temelju stvarnih podataka iz sličnih servisa. Na kraju simulacije, rezultati su pohranjeni u CSV datoteku.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Broj servisera | Max broj servisera u isto vrijeme | Ukupna dnevna zarada | Ukupna dnevna cijena rada | Ukupan broj klijenata | Broj uspješnih popravaka | Broj neuspješnih popravaka |
| 1 | 7 | 525 | 225 | 90 | 35 | 55 |
| 2 | 6 | 645 | 450 | 90 | 43 | 47 |
| 3 | 8 | 735 | 675 | 90 | 49 | 41 |
| 4 | 7 | 615 | 900 | 90 | 41 | 49 |
| 5 | 6 | 630 | 1125 | 90 | 42 | 48 |
| 6 | 7 | 540 | 1350 | 90 | 36 | 54 |
| 7 | 7 | 765 | 1575 | 90 | 51 | 39 |
| 8 | 8 | 540 | 1800 | 90 | 36 | 54 |
| 9 | 9 | 675 | 2025 | 90 | 45 | 45 |

Tablica prikazuje rezultate simulacije za različite konfiguracije broja servisera. Prvi broj u svakom retku označava broj servisera koji su trebali raditi, dok drugi broj označava broj servisera koji su dosli kao ispomoć. Ostali stupci sadrže informacije o ukupnoj dnevnoj zaradi, ukupnoj dnevnoj cijeni rada, ukupnom broju klijenata, te broju uspješnih i neuspješnih popravaka.

Na temelju rezultata prikazanih u tablici, primjetan je zanimljiv trend u ukupnoj dnevnoj zaradi i broju neuspješnih popravaka kako se mijenja broj servisera i dolazak dodatnih servisera u servis. Iako se može očekivati da bi veći broj servisera rezultirao većom zaradom, čini se da to nije slučaj u ovom modelu. Povećanje broja servisera ne pokazuje značajno povećanje ukupne dnevne zarade, što sugerira da postoji složenija dinamika koja utječe na profitabilnost poslovanja. Potrebno je detaljnije istražiti razloge za ovako ponašanje, uključujući i moguće ograničenje u kapacitetu servisera ili druge faktore koji mogu utjecati na učinkovitost poslovanja.

# Prikaz kOda i funkcionalnosti simulacije

Slika 4.1. prikazuje dio koda koji služi kao osnovna postavka za simulaciju rada servisnog centra za bicikle. U kodu se definiraju dvije vrste entiteta (serviseri i klijenti) i uspostavljaju se globalne varijable koje prate važne metrike poput zarade, troškova i broja popravaka. Serviseri i klijenti imaju specifične osobine koje im omogućuju međusobnu interakciju i izvršavanje zadataka unutar simulacije, kao što su popravak bicikala i praćenje vremena čekanja. Varijabla „privremeni“ označava je li serviser stalno zaposlen ili je priveremeni, što omogućuje precizno praćenje i alokaciju resursa, također su uključene varijable za upravljanje rasporedom dolazaka klijenata i alokacijom servisera,što omogućuje detaljno plaćanje i optimizaciju operativnih procesa u servisu.

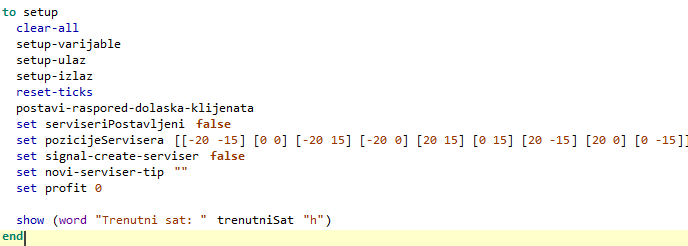


Slika 4.1. Deklaracija varijabli okruženja simulacijskog modela

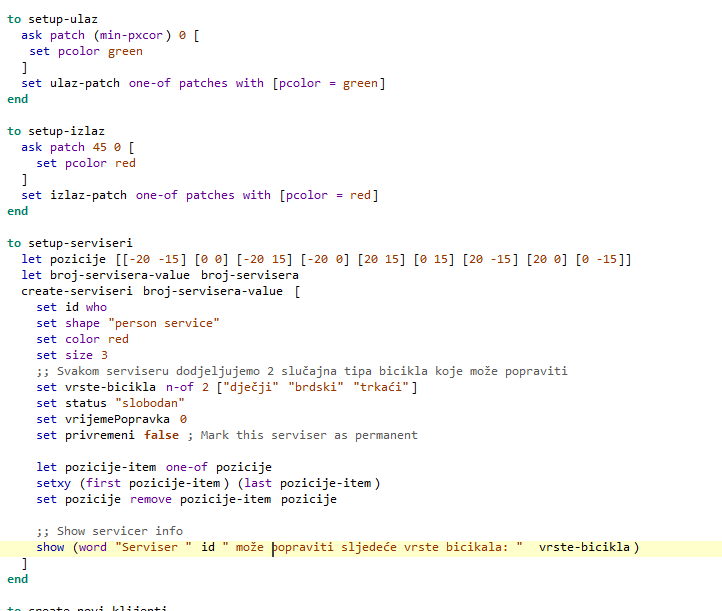
Slika 4.2. prikazuje kod kojim se inicijalizira simulacija servisa. Ovaj dio koda osigurava da su svi početni uvjeti postavljeni kako bi simulacija mogla pravilno funkcionirati. Proces započinje čišćenjem svih prethodnih postavki i objekata kako bi se osigurala čista ploča za novu simulaciju. Nakon toga postavljaju se sve potrebne varijable koje će pratiti ključne metrike kao što su zarada, troškovi, broj popravaka i ostale statistike.

Sljedeći korak u inicijalizaciji uključuje definiranje točaka ulaska i izlaska za klijente unutar simulacije, omogućujući kretanje klijenata kroz servis. Resetira se brojač simulacije, postavljajući vrijeme na nulu, čime se omogućuje praćenje svih događaja od samog početka. Nadalje, postavlja se raspored dolaska klijenata tokom radnog dana, osiguravajući pravilnu distribuciju posla za servisere.

Varijabla „serviseriPostavljeni“ postavljena je na false, osiguravajući da serviseri još nisu postavljeni. Definirane su koordinatne pozicije, na kojima će serviseri biti smješteni, čime se osigurava organizirani raspored unutar servisa. Signal za kreiranje novih servisera postavljen je na false spriječavajući trenutno dodavanje novih servisera, dok varijabla za tip novog servisera ostaje prazna. Početni profit je također postavljen na nulu, omogućujući praćenje profita od samog početka simulacije. Na kraju, prikazuje se trenutni sat simulacije.



Slika 4.2. Postavljanje modela simulacije

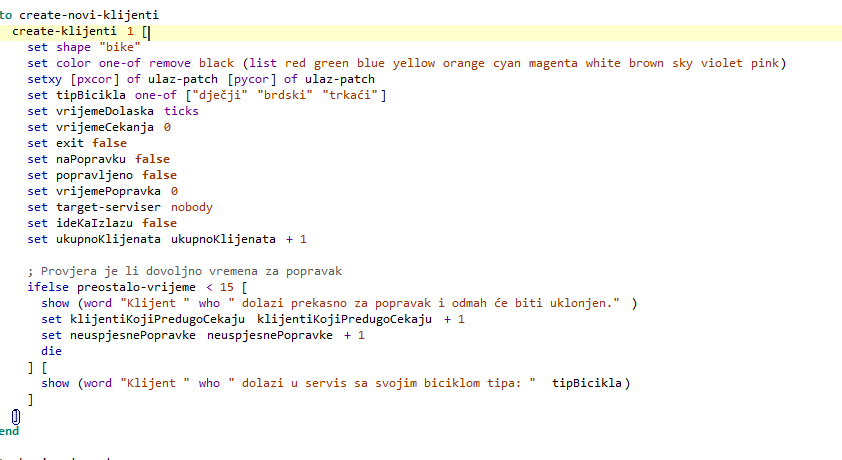


Slika 4.3. Postavljanje simulacije servisa

Na slici 4.3 imamo prikazane: 'setup-ulaz' postavlja ulaznu lokaciju, koja je označena zelenom bojom na poziciji s minimalnom x-koordinatom i y-koordinatom 0. Zatim se odabire jedan od patcheva s postavljenom zelenom bojom i označava se kao ulazna lokacija za klijente.

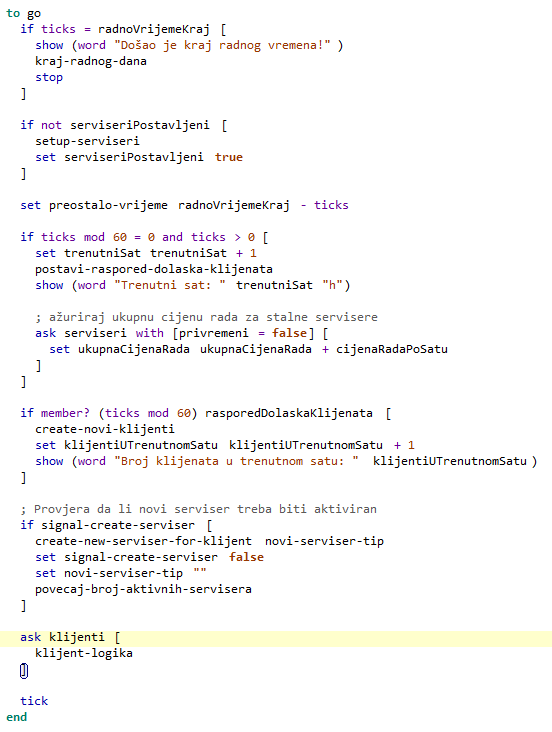
'setup-izlaz' postavlja izlaznu lokaciju, koja je označena crvenom bojom na poziciji s x-koordinatom 45 i y-koordinatom 0. Također se odabire jedan od patcheva s postavljenom crvenom bojom i označava se kao izlazna lokacija za klijente koji su završili s popravkom.

'setup-serviseri' stvara servisere unutar simulacije. Svaki serviser dobiva određene karakteristike kao što su oblik, boja, tipovi bicikala koje može popraviti, status popravka i pozicija. Serviseri se postavljaju na predefinirane pozicije unutar simuliranog prostora, a svaki od njih je sposoban popraviti dva slučajno odabrana tipa bicikala. Nakon postavljanja, informacije o svakom serviseru prikazuju se u konzoli kako bi korisnik imao pregled o njihovim karakteristikama.



Slika 4.4. Kreiranje novog klijenta unutar simulacije

Funkcija create-novi- prikazana na slici 4.4 je odgovorna za stvaranje novih klijenata unutar simulacije. Kada se pozove, stvara se jedan klijent koristeći create-klijenti 1. Svaki klijent dobiva oblik "bike" i nasumičnu boju koja nije crna. Pozicija klijenta postavlja se na ulaznu točku simulacije. Također, klijentu se dodjeljuje nasumičan tip bicikla između "dječji", "brdski" i "trkaći". Vrijeme dolaska postavlja se na trenutni trenutak simulacije, a vrijeme čekanja na nulu jer tek dolazi. Statusi kao što su "exit" (izlazak iz servisa), "naPopravku" (je li na popravku), "popravljeno" (je li popravak završen) postavljaju se na "false" jer je novi klijent tek stvoren. Također, ciljani serviser postavlja se na "nobody", a zastavica "ideKaIzlazu" na "false". Konačno, ukupan broj klijenata se povećava za jedan. Nakon toga, provjerava se ima li preostalog vremena za popravak. Ako je manje od 15 taktova, klijent je došao prekasno te se uklanja iz simulacije. U suprotnom, prikazuje se poruka o dolasku klijenta i njegovom tipu bicikla.



Slika 4.5. Pokretanje simulacije

Na slici 4.5 prikazan je kod koji opisuje glavnu petlju simulacije servisa, upravljajući događajima koji se odvijaju cijelog dana.

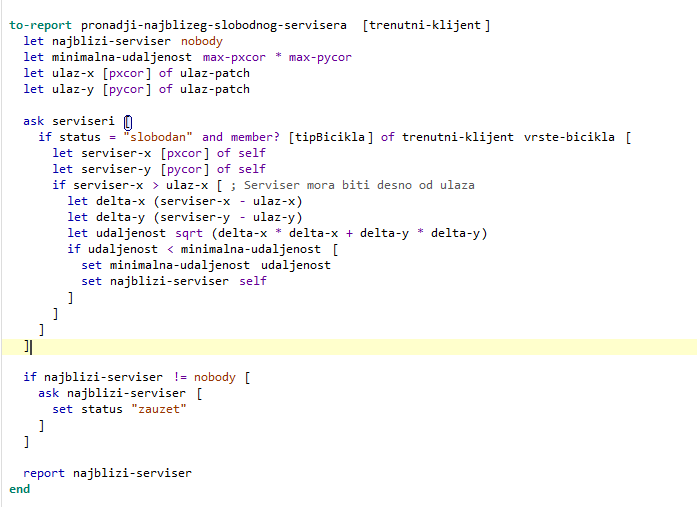
Na početku svake iteracije, provjerava se je li radno vrijeme završeno. Ako je, simulacija se zaustavlja, a korisniku se pokazuje poruka „Došao je kraj radnog vremena“.

Zatim ako serviseri još nisu postavljeni, poziva se funkcija 'setup-serviseri' koja postavlja servisere na njohove početne pozicije, a varijabla 'serviseriPostavljeni' postavlja se na *true* što osigurava da se postavljanje servisera izvrši samo jednom.

Varijabla 'preostalo-vrijeme' ažurira se kako bi pratila preostalo vrijeme do kraja radnog dana. Svakih 60 ticks-a, što prestavlja jedan sat simulacije, povećava se varijabla 'trenutniSat' i poziva se funkcija za postavljanje rasporeda dolaska klijenta i prikazuje se sat korisniku. Također se ažurira ukupna cijena rada za stalne servisere, povećavajući 'ukuonaCijenaRada' za svakog stalnog servisera.

Ako trenutni tick odgovara rasporedu dolaska klijenata, kreiraju se novi klijenti i povećava se brojač klijenata u trenutnom satu, a korisniku se prikazuje broj klijenata koji su došli u tom satu.

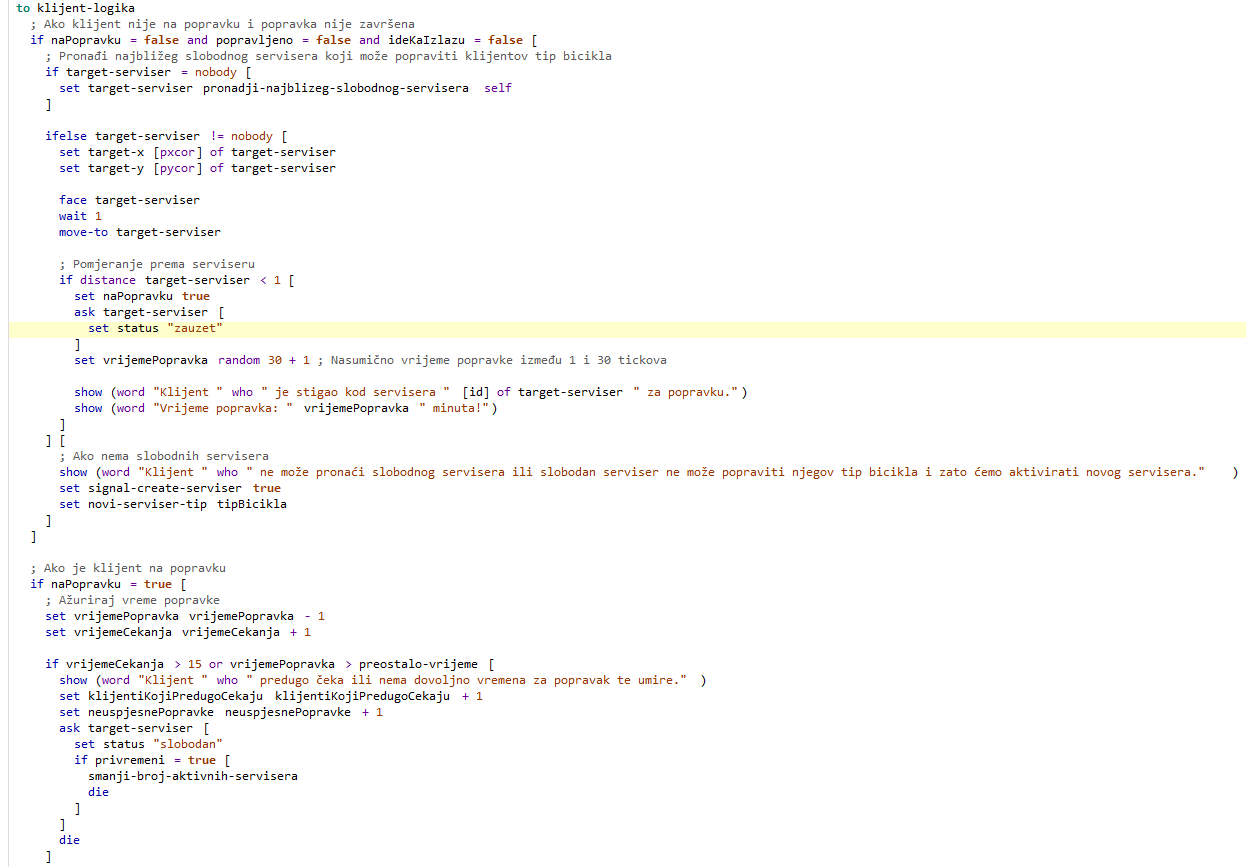
Dodatno, provjerava se signal za kreiranje novog servisera. Ako je signal aktiviran, kreira se novi serviser specifičnog tipa za klijenta, signal se resetira, a broj aktivnih servisera povećava se za jedan. Svaki klijent u simulaciji izvodi svoju logiku putem funkcije klijent-logika, koja upravlja njihovim ponašanjem unutar simulacije. Na kraju svake iteracije, poziva se tick funkcija koja ažurira stanje simulacije za sljedeći korak. Ovaj kod osigurava dinamičko upravljanje događajima i procesima unutar simulacije, omogućujući realističnu simulaciju radnog dana u servisu za popravak bicikala.



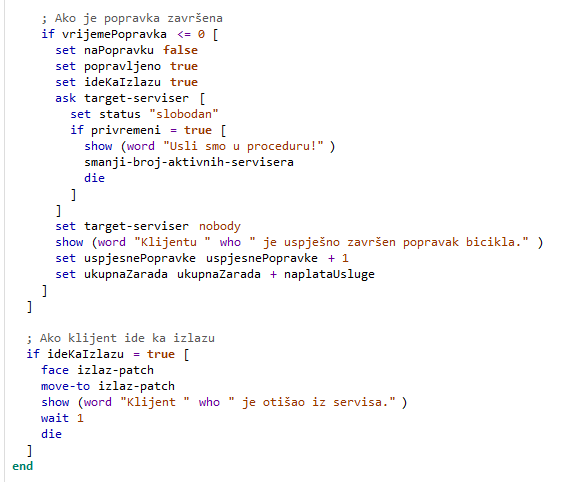
Slika 4.6. Funkcija za pronalaženje najbližeg slobodnog servisera

Na slici 4.6 prikazan je dio koda koji definira funkciju koja pronalazi najbližeg slobodnog servisera za trenutnog klijenta. Prvo se inicijaliziraju varijable za praćenje najbližeg servisera i minimalne udaljenosti. Koordinate ulaznog patcha također se postavljaju kako bi se koristile u izračunima udaljenosti. Zatim funkcija prolazi kroz sve servisere i provjerava jesu li slobodni te mogu li popraviti tip bicikla koji klijent ima.

Ako su ovi uvjeti zadovoljeni, izračunava se udaljenost između servisera i ulaza, ali samo za one servisere koji su pozicionirani desno od ulaza. Ako je udaljenost manja od trenutne minimalne udaljenosti, funkcija ažurira najbližeg servisera i minimalnu udaljenost. Kada se pronađe najbliži serviser, njegov status se postavlja na "zauzet", što znači da je sada dodijeljen klijentu. Na kraju, funkcija vraća najbližeg servisera. Ova funkcija omogućuje učinkovitu alokaciju servisera klijentima temeljem njihove blizine i dostupnosti, čime se optimizira proces popravka i smanjuje vrijeme čekanja klijenata.



Slika 4.7. Logika ponašanja klijenta unutar simulacije (1.dio)

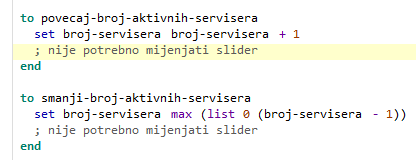


Slika 4.8. Logika ponašanja klijenta unutar simulacije (2.dio)

Na slici 4.7 i 4.8 prikazan je dio koda koji definira logiku ponašanja klijenta unutar simulacije servisa popravka bicikala. Prvo se provjerava je li klijent trenutno nije na popravku, popravak još nije završen i klijent nije na putu prema izlazu. Ako su ti uvjeti ispunjeni, klijent pokušava pronaći najbližeg slobodnog servisera koji može popraviti njegov tip bicikla. Ako serviser postoji, klijent se kreće prema njemu, čime započinje proces popravka. Ukoliko nema slobodnih servisera ili nijedan serviser ne može popraviti klijentov bicikl, generira se signal za stvaranje novog servisera.

Ako je klijent već na popravku, provjerava se koliko je vremena prošlo od početka popravka te se ažurira i vrijeme čekanja i vrijeme popravka. Ako klijent čeka predugo ili nema dovoljno vremena za popravak, proces se prekida i označava se kao neuspješan.

Kada je popravak završen, klijentu se označava da je bicikl uspješno popravljen, a on kreće prema izlazu. Konačno, kada klijent dođe do izlaza, proces se završava, a klijent se uklanja iz simulacije.

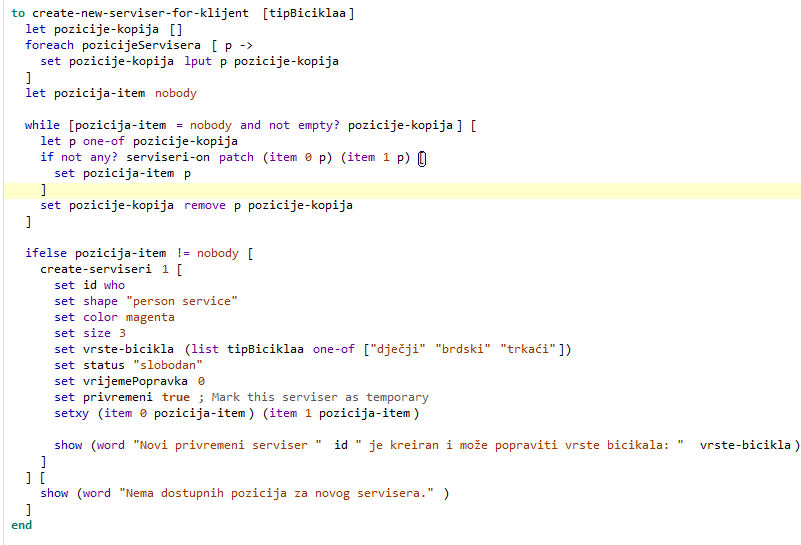


Slika 4.9.Prikaz procedura za upravljanje brojem aktivnih servisera

Na slici 4.9 su preikazane procedure koje upravljaju promjenom broja aktivnih servisera unutar simulacije.

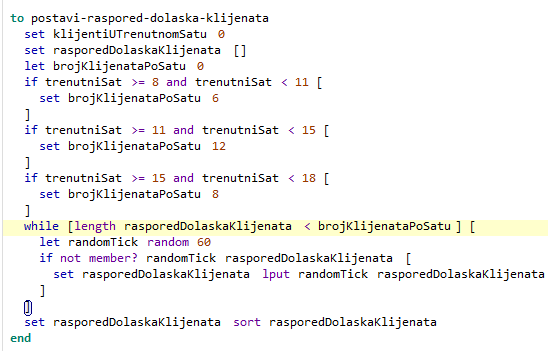
Procedura povecaj-broj-aktivnih-servisera povećava broj aktivnih servisera za jedan. Ovo se obavlja dodavanjem jedinice trenutnom broju servisera u simulaciji. Ovo je korisno kada je potrebno dodati više radnika u sustav kako bi se nosili s većim opterećenjem ili povećao kapacitet popravaka.

Suprotno tome, procedura smanji-broj-aktivnih-servisera smanjuje broj aktivnih servisera za jedan. Ovo se postiže smanjenjem trenutnog broja servisera za jedan, ali se istovremeno osigurava da broj servisera nikada ne bude manji od nule, koristeći funkciju max koja osigurava minimalnu granicu. Ovo je korisno kada se potreba za popravcima smanjuje ili kada je potrebno optimizirati troškove rada smanjenjem broja zaposlenika.



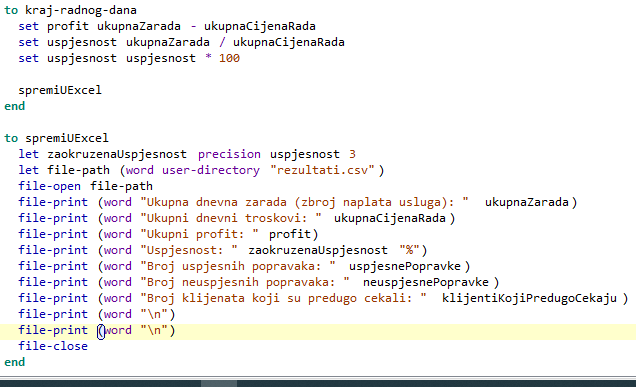
Slika 4.10. Prikaz stvaranja novog servisera unutar simulacije

Funkcija create-new-serviser-for-klijent prikazana na slici 4.10 služi za stvaranje novog servisera koji će popravljati bicikle za određenog klijenta. Prvo se kreira kopija liste pozicija servisera kako bi se izbjegla dvostruka dodjela iste pozicije serviseru. Zatim se provjerava dostupnost svake pozicije servisera. Ako je pozicija slobodna, serviser se kreira na toj lokaciji, a zatim mu se dodjeljuju odgovarajuće osobine poput ID-a, oblika, boje, vrsta bicikla koje može popravljati i statusa. Ako nema dostupnih pozicija za novog servisera, prikazuje se odgovarajuća poruka. Ova funkcija osigurava dinamično dodavanje novih servisera u simulaciju kad god je to potrebno.



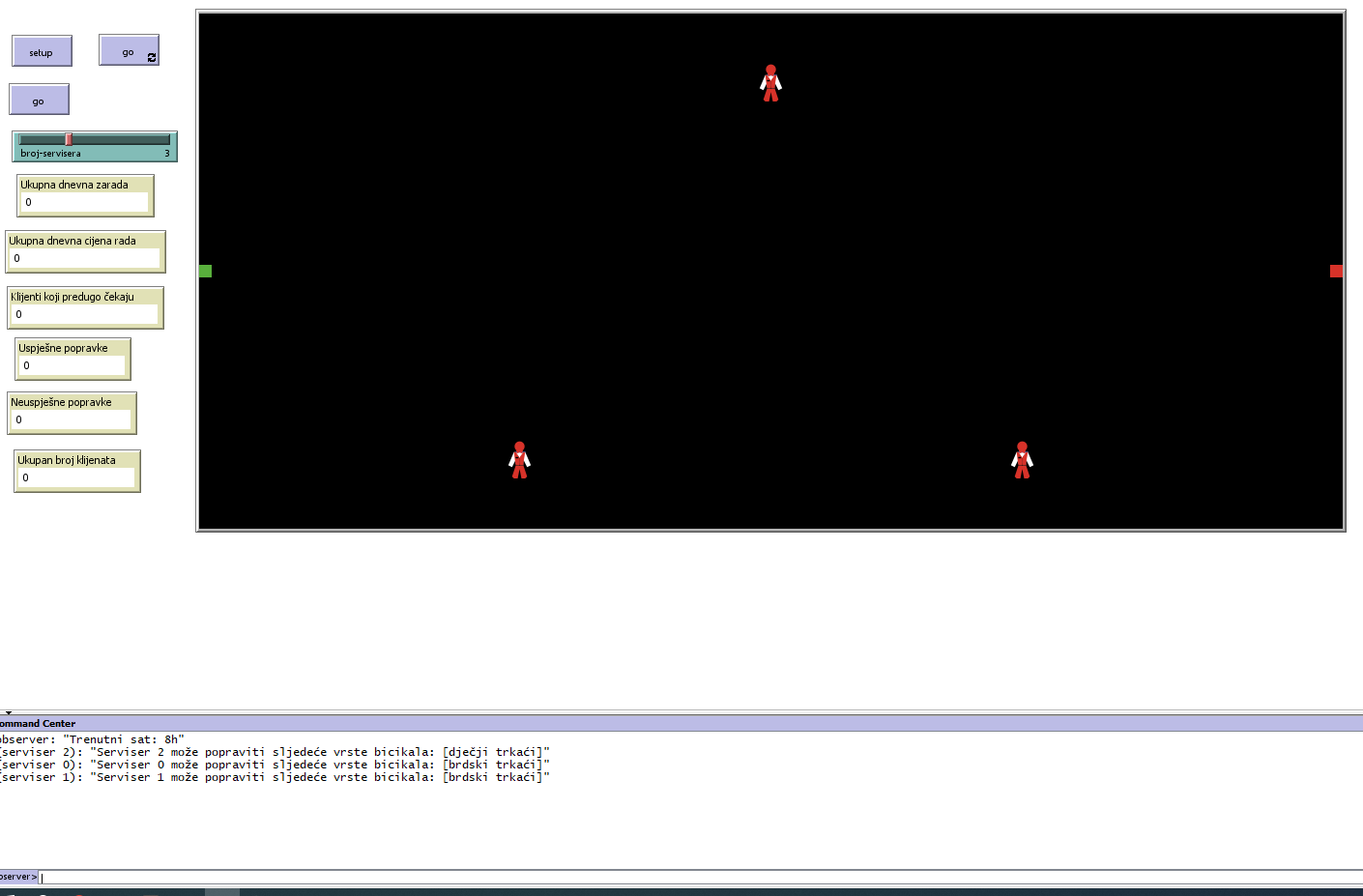
Slika 4.11. Prikaz generiranja dolaska klijenata u servisni centar

pFunkcija 'postavi-raspored-dolaska-klijenata' prikazana na slici 4.11 dinamično generira raspored dolaska klijenata u servisni centar tijekom radnog dana simulacije. Na temelju trenutnog vremena u simulaciji, određuje se koliko će klijenata doći u svakom satu, prilagođavajući se različitim dijelovima dana. Nakon određivanja broja klijenata, generiraju se nasumični trenuci dolaska unutar svakog sata. Petlja osigurava jedinstvenost svakog trenutka dolaska, izbjegavajući duple dolaske. Konačno, raspored se sortira po vremenskom redoslijedu. Ova funkcija omogućuje simulaciji da dinamički upravlja dolaskom klijenata, pružajući realističan prikaz aktivnosti u servisnom centru tijekom radnog dana.



Slika 4.12. Prikaz funkcije za kraj radnog vremena te spremanje CSV datoteke

Na slici 4.12. prikazana je funkcija kraj-radnog-dana završava simulaciju za taj radni dan, izračunava profit i uspješnost servisa na temelju prikupljenih podataka te sprema te podatke u Excel datoteku. Funkcija spremiUExcel koristi putanju korisničkog direktorija kako bi otvorila CSV datoteku, ispisuje relevantne podatke o zaradi, troškovima, profitu, uspješnosti i broju popravaka te zatvara datoteku.



Slika 4.13. Prikaz simulacijskog okruženja

Simulacijsko okruženje, prikazano na slici 4.13., sadrži vizualne elemente koji simboliziraju servisere i bicikle koji predstavljaju klijente. Također, uključuje gumbe koji omogućuju postavljanje simulacijskog modela, ručno pokretanje te pokretanje simulacije od početka do kraja radnog vremena. Klizač se koristi za odabir broja aktivnih servisera, dok se monitori koriste za praćenje analitičkih podataka kao što su zarada, troškovi te broj uspješnih i neuspješnih popravaka. Ovaj kompletan okvir omogućuje korisnicima uvid u rad servisa kroz simulaciju te omogućuje praćenje i analizu ključnih performansi.

# Zaključak

Analizom simulacijskog okruženja, duboko se uranjamo u složeni proces rada servisa za popravak bicikala. Kroz praćenje ključnih metrika poput zarade, troškova i uspješnosti popravaka, dobivamo uvid u performanse servisa tijekom radnog dana. Korištenje simulacijskog modela omogućuje nam ne samo vizualizaciju, već i interaktivno ispitivanje različitih scenarija i postavki, pružajući nam priliku za optimizaciju procesa. Na temelju analitičkih podataka i dinamičnih promjena u simulaciji, možemo identificirati ključne čimbenike koji utječu na efikasnost servisa, kao i potencijalne mjere za poboljšanje, poput prilagodbe broja servisera ili promjene cijena usluga. Sveukupno, simulacijsko okruženje pruža dragocjen alat za informirano odlučivanje i unaprjeđenje operativnih procesa, čime se postiže veća profitabilnost i konkurentnost servisa za popravak bicikala.