

Traffic Simulator

Răzvan-Andrei Zavalichi
Departamentul de Calculatoare
Universitatea "Gheorghe Asachi"
Iași, România
Email: razvan.zavalichi@gmail.com

Adrian Alexandrescu
Departamentul de Calculatoare
Universitatea "Gheorghe Asachi"
Iași, România
Email: aalexandrescu@tuiasi.com

Abstract— Congestionarea sau aglomerarea traficului rutier reprezintă starea în care se află rețeaua de transport, caracteristică de viteze de deplasare mici, timpi și cozi de așteptare lungi. Se consideră că traficul rutier devine congestionat atunci când cererea de transport, atât autovehicule, cât și pietoni, depășește capacitatea rețelei de transport. Principalele metode ce se pot adopta în scopul decongestionării traficului rutier sunt următoarele: implementarea soluțiilor de conducere dinamică a semafoarelor, în funcție de condițiile reale de trafic și dezvoltarea infrastructurii actuale prin construirea de noi drumuri în jurul orașelor sau chiar supraetajate.

Soluția propusă pentru rezolvarea problemei traficului rutier este un mediu virtual în care se pot face diverse simulări. Acest mediu permite interacțiunea cu utilizatorul, acesta putând configura după bunul plac atât semafoarele cât și participanții la trafic.

Keywords— Algoritmă și programare, Aplicații software

I. INTRODUCERE

Studiile de trafic prezentate de literatură de specialitate consideră că principalele cauze ale congestionării traficului rutier sunt reprezentate de: condiții meteorologice nefavorabile; evenimente de trafic neprevăzute; efectuarea operațiilor de reabilitare a infrastructurii rutiere; timpi de semaforizare ce nu sunt adaptați condițiilor reale din trafic, în funcție de zi și oră sau de condițiile atmosferice. Formele de dirijare a autovehiculelor și pietonilor într-o anumită zonă rutieră reprezintă de fapt conducerea traficului rutier, ce are ca obiectiv principal optimizarea în vederea fluidității traficului.

Metodele de conducere a traficului rutier implică pe lângă soluțiile adoptate în vederea fluidizării traficului și metodele de gestionare a evenimentelor special generate de ambulanță, poliție, pompieri sau a blocajelor pe anumite artere rezultate în urmă accidentelor rutiere sau a lucrărilor de mentenanță.

Metodele de dirijare și conducere a intersecțiilor semaforizate sunt clasificate în patru categorii:

- 1) Conducere statică – se bazează pe timpi prestabiliți pentru fiecare fază a ciclului de semaforizare;
- 2) Conducere dinamică – determinate de timpi de semaforizare calculați în funcție de condițiile reale de trafic;
- 3) Conducere prin coordonare – se realizează prin timpi de semaforizare stabiliți de către un anumit punct de control al traficului pe baza informațiilor generate de trafic;
- 4) Conducere la cerere – activată prin apăsarea unui buton de către pietoni;

În fiecare zi ne întâlnim cu problema traficului care de multe ori duce la diferite tipuri de accidente. Această problemă este cauzată de neatenția participanților la trafic sau de metodele de conducere și dirijare a intersecțiilor semaforizate.

În continuare, voi descrie câteva exemple de aplicații de simulare a traficului:

- 1) **Cities: Skylines** este o aplicație dezvoltată de Colossal Order și publicată de către Paradox Interactive. Aceasta este un joc cu un singur jucător care îi permite să construiască orașe și să stabilească propria strategie pentru simularea traficului.
- 2) **PTV Vissim** este un grup complex care se ocupă de simularea traficului rutier prezentând situații complexe din trafic.
- 3) **Wbots: The autonomous vehicle simulator** este o aplicație de simulare a traficului rutier care folosește vehicule autonome care se bazează pe inteligență artificială și algoritmi de prelucrare a imaginilor.

În scopul realizării acestei lucrări am dezvoltat o aplicație 3D în Unity, în cadrul căreia utilizatorul poate alege diferite zone pentru simulare. Acesta poate genera mașini în diferite puncte de pe hartă. Pentru a realiza o simulare dorită, utilizatorul are posibilitatea de a schimba configurațiile fiecărui semafor precum și a mașinilor. Pentru a interacționa cât mai mult cu simularea creată există posibilitatea generării unei mașini care poate fi condusă de către utilizator.

Modul prin care am atins obiectivul propus a fost prin scrierea unor scripturi în limbajul de programare C# pe care le-am aplicat unor obiecte în Unity și crearea atât a zonelor de simulare și a obiectelor 3D în Blender pe care le-am introdus în Unity.

II. SOLUȚIA PROPUȘĂ

A. Prezentare Generală

Soluția propusă presupune realizarea unui mediu virtual în care se poate simula traficul rutier, putându-se observa comportamentul participanților la trafic pentru diferite configurații ale semafoarelor.

Fiecare intersecție va conține semafoare care pot fi configurate de către utilizator. Acesta poate schimba timpii de staționare pentru fiecare semnal luminos, precum și activarea semnalului luminos dorit, ulterior aceste configurații putând fi salvate într-un fișier xml. Fișierul xml va avea următoarea structură:

- *Numele intersecției*
 - *Numele semaforului*
 - *Timpul de staționare și starea curentă pentru culoarea roșie*
 - *Timpul de staționare și starea curentă pentru culoarea galbenă*
 - *Timpul de staționare și starea curentă pentru culoarea verde*

Participanții la trafic vor fi de trei tipuri: mașini reale (Realistic car), mașini simple (Simple car) și mașina

controlată (Car Controller). Caracteristicile mașinilor precum viteza pot fi schimbate de către utilizator.

- *Simple car*: este un tip de mașină care nu ține cont de coliziuni și de regulile fizicii. Mișcarea acesteia se realizează prin mutarea obiectului după o direcție, mai precis prin schimbarea poziției (coordonate x, y, z).

Fiecare mașină este prevăzută cu o logică simplă de circulație, aceasta având 3 senzori activi, poziționați în fața mașinii (stânga, centru, dreapta).

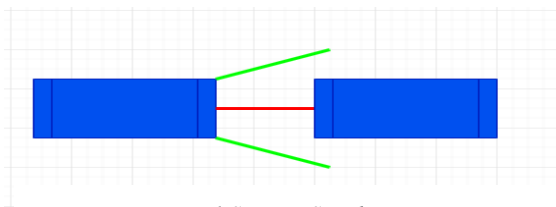


Fig. 1 Senzori-Simple car

Dacă unul dintre senzori detectează un obiect mașina se oprește. În intersecții mașinile care merg înainte au prioritate.

- *Realistic car*: acest tip de mașină va avea o masă aplicată unui corp rigid, precum și roți. Mișcarea se va realiza prin aplicarea unei forțe de rotații asupra roților. Pentru direcția mașinii se va calcula unghiul dintre roțile din față și punctul în care trebuie să ajungă mașina.

Pentru participarea la trafic, acest tip de mașină este prevăzută cu un număr mai mare de senzori utilizați pentru o precizie mai bună de determinare a poziției acesteia față de ceilalți participanți la trafic. Spre deosebire de mașinile de tip Simple Car acești senzori au scopul principal de a evita coliziunile cu ceilalți participanți la trafic urmând logica de acordare a priorității. Senzorii sunt poziționați astfel:

- Trei senzori sunt în față (stânga, centru, dreapta)
- Doi senzori sunt în față (stânga, dreapta) care au un unghi de 15° , respectiv -15°
- Doi senzori sunt în față pe lateral (stânga, dreapta) care au un unghi de 90° , respectiv -90°
- Doi senzori sunt în față (stânga, dreapta) care au un unghi de 15° , respectiv -15°

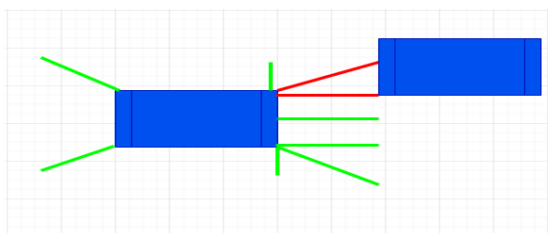


Fig. 2 Senzori-Realistic car

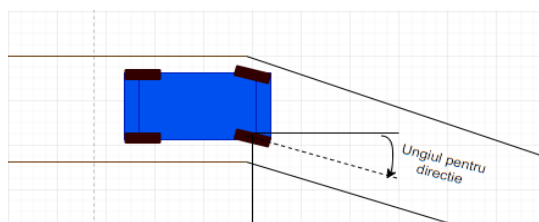


Fig. 4 Unghiul pentru direcție-Realistic car

- Car controller: acest tip de mașină poate fi controlat de către utilizator, acesta având posibilitatea să determine ruta minimă de fiecare dată când ajunge într-o intersecție.

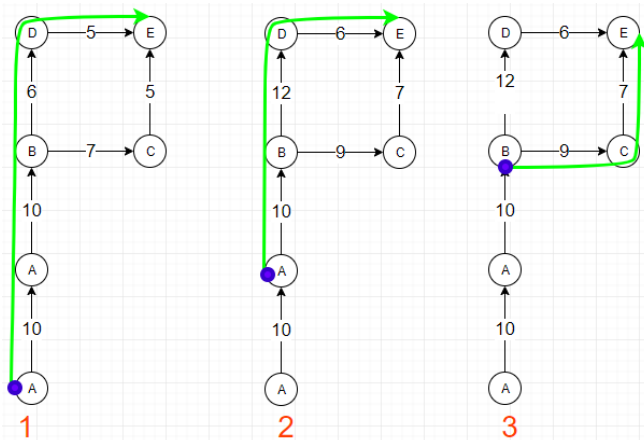


Fig. 5 Car Controller – Actualizarea rutei optime

o Algoritm de determinare a rutei optime

Întrucât totalitatea drumurilor reprezintă un graf orientat ponderat, pentru a rezolva problema rutei optime se va folosi Dijkstra.

Algoritmul Dijkstra este un algoritm pe care îl putem folosi pentru a găsi distanțele cele mai scurte sau costurile minime în funcție de ceea ce este reprezentat într-un grafic. Acest algoritm poate găsi calea cea mai scurtă de la un nod într-un grafic la fiecare alt nod din aceeași structură de date grafice, cu condiția ca nodurile să fie accesibile din nodul de pornire. Algoritmul va continua să ruleze până când vor fi vizitate toate vârfurile din grafic.

În cazul soluției propuse valoarea numerică a muchiilor este dată de timpul în care o mașină a trecut dintr-un punct în altul. Inițial toate muchiile reprezintă distanța dintre vârfurile corespundente, iar în momentul în care o mașină ajunge într-un punct, valoarea numerică va reprezenta timpul de parcurgere a distanței dintre punctul respectiv și cel precedent, astfel se va genera un drum de cost minim care corespunde cu traficul din acel moment.

B. Specificații funcționale

Traffic Simulator este o aplicație 3D care are scopul de a realiza simulări ale traficului rutier pentru diferite metode de dirijare și conducere a intersecțiilor semaforizate. Astfel, utilizatorul poate alege una dintre zonele existente pentru a face simularea. După ce acesta a ales o zonă poate începe simularea propriu-zisă, generând mașinile.

Fiecare zonă creată conține un set de puncte, care alcătuiesc un graf orientat ponderat. Fiecare punct conține poziția sa în coordonate x, y, z , succesorii săi (punctele către

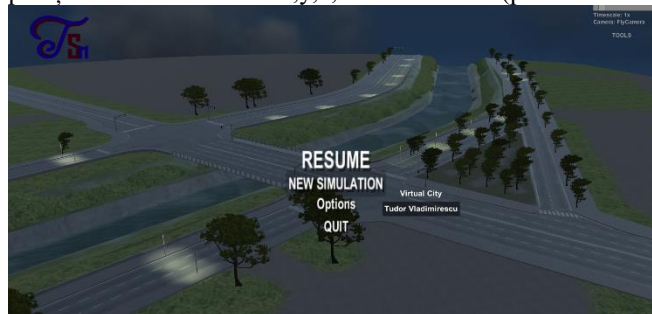


Fig. 6 Zonă simulare-Tudor Vladimirescu

care se pot deplasa mașinile), predecesorii săi (punctele din

care vin mașinile) și distanțele parcurse de la fiecare predecesor până la punctul curent. Acest graf este cunoscut de către fiecare mașină și reprezintă totalitatea rutelor (drumurilor) din zona respectivă.

Dacă o mașină ajunge într-un punct terminal (punct cu gradul 1, care nu are succesori), aceasta este "distrusă" și dispare din simulare.

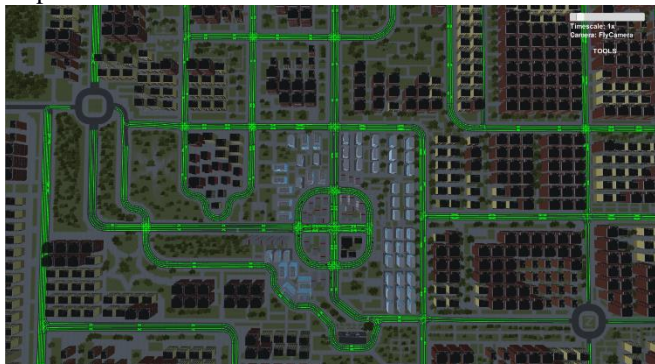


Fig. 7 Graf complet orientat



Fig. 8 Graf complet orientat

A. Obiectele din simulare

- 1) Mașina
- 2) Mașina pentru control
- 3) Intersecție
- 4) Indicator rutier
- 5) Semafor
- 6) Camera

1. **Mașina:** reprezintă participantul la trafic. Aceasta primește un punct inițial, urmând ca de fiecare dată să aleagă aleator următorul punct din graf. Selectarea următorului punct se face aleator pe baza probabilității.

```
public Waypoint GetNextWaypoint()
{
    int prob = Random.Range(0, 100);
    int sum = 0;
    for (int i = 0; i < Outs.Count; ++i)
    {
        sum += Outs[i].Probability;
        if (prob <= sum)
            return Outs[i].Waypoint;
    }
    SetSame();
    Debug.Log(this.name + " Probabilities has been corrected");
    return Outs[Outs.Count - 1].Waypoint;
}
```

Fig. 9 Funcție-Preia următorul punct

2. **Intersecție:** este un obiect invizibil care are dimensiunea intersecției. Acest obiect este folosit pentru a detecta mașinile care intră, respectiv ies din intersecție.

3. **Mașina pentru control:** această mașină este un obiect care poate fi controlat de către utilizator. Dacă este selectată opțiunea "Show shortest path", mașina va genera o nouă rută optima de fiecare data când se află într-un punct cu mai mulți succesori



Fig. 10 Car controller cu "Show shortest path" activat

4. **Sem rutier:** este un obiect care determină când o mașină este în fața lui și îi comunică ce indicații rutiere trebuie să respecte.



Fig. 11 Semn rutier - Stop

5. **Semafor:** acest obiect este controlat de către un "TrafficLights Manager" atribuit fiecărei intersecții. Fiecare semafor detectează ce mașină se află la intrarea în intersecție și în funcție de culoarea activă (roșu, galben, verde) îi spune ce indicații rutiere trebuie să respecte.
6. **Camera:** pentru a putea vizualiza cât mai bine simularea dorită au fost create două tipuri de camere:



Fig. 11 Semafor

- **Follow Camera** urmărește obiectul atașat, aceasta aflându-se în spate.
- **Fly Camera** permite utilizatorului să observe fiecare zonă din scenă, acesta putându-i schimba poziția, înălțimea precum și rotația.

B. Interacțiunea cu utilizatorul

Aplicația prezintă mai multe posibilități prin care poate interacționa cu utilizatorul. Acesta are la dispoziție un meniu principal în care poate începe o nouă simulare, oprește temporar simularea, schimbă setările aplicației sau închide

aplicația, precum și un meniu “Tools” din care poate controla simularea.

1. Setările aplicației



Fig. 12 Meniul „Options”

- Fereastra FPS: această fereastră arată numărul de cadre pe secundă. Este utilă pentru a vedea dacă sunt generate prea multe mașini sau dacă aplicația poate fi rulată în parametri optimi pe calculatorul respectiv.
- Fereastra Minimap: această fereastră este utilă pentru a vedea o anumită zonă din simulare cu camera principală și pentru a avea o viziune asupra întregii simulări folosind fereastra Minimap.

2. Meniul “Tools” conține următoarele funcționalități:

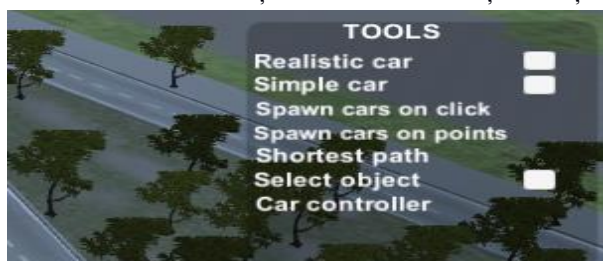


Fig. 13 Meniul „Tools”

- Realistic Car*
 - Simple Car*
 - Spawn cars on click*
 - Spawn cars on points*
 - Shortest path*
 - Select object*
 - Car controller*
- Realistic Car*: este un buton de comutare. Atunci când acest buton este apăsat mașinile generate vor fi de tipul Realistic car
 - Simple Car*: este un buton de comutare. Atunci când acest buton este apăsat mașinile generate vor fi de tipul Simple car
 - Spawn cars on click*: în urma apăsării acestui buton utilizatorul va trebui să introducă numărul de mașini pe care vrea să le genereze într-un anumit punct de pe drum. Punctul va fi selectat prin apăsarea butonului stâng al mouse-ului. Mașinile se vor genera în cel mai apropiat punct al grafului față de punctul selectat.
 - Spawn cars on points*: în urma apăsării acestui buton utilizatorul va trebui să introducă numărul de mașini pe care vrea să le genereze în punctele deja plasate la capetele de intrare în scena activă.

- Shortest path*: permite utilizatorului să genereze ruta de cost minim. În urma apăsării acestui buton utilizatorul trebuie să selecteze punctul de start și punctul de final.

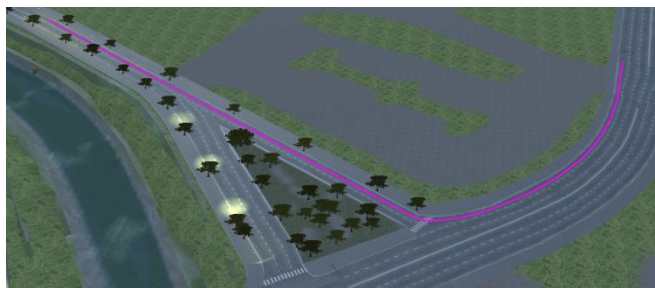


Fig. 12 Ruta optimă

- Select object*: este un buton de comutare care permite atât vizualizarea cât și editarea informațiilor obiectelor selectate. În urma selectării unui obiect va apărea o fereastră cu informațiile acestuia în stânga ecranului. Obiectele selectate pot fi de două tipuri:
 - Mașină*: atunci când o mașină este selectată apar următoarele informații: senzorul care este activ, mașina care se află în față și viteza mașinii care poate fi schimbată. Pentru fiecare mașină în parte există posibilitatea ștergerii acesteia de către utilizator, adăugării camerei Follow Camera sau setării ca centru pentru Minimap.



Fig. 13 Meniu pentru mașina selectată

- Semafor*: atunci când un semafor este selectat va apărea o fereastră cu informațiile despre toate semafoarele din intersecția care conține semaforul respectiv. Utilizatorul poate schimba configurațiile fiecărui semafor și anume: schimbare semnalului luminos activ, schimbarea timpului de staționare pentru fiecare semnal luminos în parte. De asemenea configurațiile făcute pot fi salvate pentru fiecare intersecție în parte, acestea putând fi încărcate în orice altă simulare care conține o intersecție cu aceeași denumire.



Fig. 13 Meniu pentru mașina selectată

Fiecare intersecție conține butonul de comutare către o intersecție de tip “Smart intersection”. Acest tip de intersecție obligă semafoarele care au activ semnalul luminos roșu să nu-l schimbe atâta timp cât în intersecție încă sunt mașini.

- g. *Car controller*: atunci când acest buton este apăsat utilizatorul trebuie să aleacă în ce punct să fie generată o mașină pe care o poate controla. De asemenea și aceasta este prevăzută cu o fereastră informativă care îi permite ștergerea mașinii, setarea acesteia ca centru pentru minimap, atașarea unei camere de tip Follow Camera și să genereze ruta de cost minim până la un anumit punct.



Fig. 13 Meniu pentru mașina controlată

Ruta generată va fi actualizată de fiecare dată când mașina se va afla într-un punct care are mai mulți succesori, astfel aceasta îi va calculată mereu în funcție de traficului curent.

III. CONCLUZII

Soluția propusă poate duce la determinarea unor soluții pentru rezolvarea problemei traficului. Scopul de bază al acesteia este de a observa rezultatele care pot apărea în trafic în urma modificărilor structurii unor intersecții sau a semafoarelor precum și generarea rutei de cost minim în funcție de traficul prezent.

De asemenea, această aplicația poate fi folosită și în alte scopuri precum: interacțiunea utilizatorului cu aplicația pentru divertisment sau poate fi folosită ca suport pentru alte aplicații care necesită un simulator de trafic.

IV. ÎMBUNĂTĂȚIRI VIITOARE

Aplicația prezentată reprezintă doar o parte dintr-un proiect mai mare ce urmează a fi implementat. Proiectul final va avea la bază Traffic Simulator la care se vor adăuga următoarele:

- Metode de optimizare a traficului rutier prin semafoare inteligente;
- Extinderea simulării pe orașe sau zone mai mari dintr-un oraș;
- Sincronizarea intersecțiilor;
- Adăugarea unor noi tipuri de vehicule precum: autobuze, poliție, salvare, pompieri și tramvaie;
- Adăugarea altor participanți la trafic precum: pietoni și bicicliști;
- Adăugarea posibilității de a schimba condițiile meteorologice;

- Implementarea unor algoritmi de detectare a indicatoarelor rutiere și a participanților la trafic pe baza imaginilor;
- Implementarea unei mașini inteligente care învață să participe la trafic și să respecte regulile de circulație;
- Generarea orașelor folosind openstreetmap

O altă metodă ar fi adăugarea unor noi tipuri de autovehicule precum: autobuze, poliție, salvare, pompieri și tramvaie.

REFERINȚE

- [1] Documentația oficială, Unity3D, <http://docs.unity3d.com/>
- [2] Documentația oficială, Blender, <https://docs.blender.org/manual/en/latest/index.html>
- [3] Florin Leon, http://florinleon.byethost24.com/lab_ip.htm, Ingineria programării, laboratoare 1, 2, 11
- [4] Joe Mayo, “C# Succinctly”, 2015