

Sistem inteligent de

comutare a semafoarelor

Strugar Madalina -Alexandra

Grupa 30233

2023-2024

Cuprins

1. Introducere………………………………………………….2
   1. Context…………………………………………….........2
   2. Specificatii…………………………………………...….2
   3. Obiective…………………………………………...…....2
2. Studiu bibliografic…………………………………………...3
3. Design si analiza…………………………………………..…4
4. Implementare………………………………………………...5
5. Testare………………………………………………………..9
6. Concluzii……………………………………………………10
7. Bibliografie…………………………………………………10

1.Introducere

1.1 Context

Proiectul propus își propune să dezvolte și să implementeze o simulare a unui sistem inteligent de control al semnalelor rutiere, având ca scop principal optimizarea fluxului de trafic. Această simulare este concepută pentru a reprezenta în mod dinamic părți ale lumii reale, cu un accent deosebit pe dinamica traficului. Implementarea acestei simulări implică crearea unui model computerizat care oferă o imagine cuprinzătoare a interacțiunii dintre drumuri, vehicule și comenzile semnalului.

1.2 Specificații

Proiectul folosește limbajul de programare Python și constă dintr-o interfață grafică ușor de utilizat, care oferă vizualizare detaliată a modificărilor din rețeaua de drumuri, mișcările vehiculelor și schimbările semaforului. Datele necesare simulării sunt obținute din fișiere sau generate în cadrul proiectului, asigurând variabilitate și adaptabilitate la diferite scenarii de trafic.

1.3 Obiective

Scopul principal al proiectului este de a proiecta și implementa o simulare a unui sistem inteligent de control al semaforului. Sunt de asemenea luate în considerare aspecte precum siguranța rutieră și eficiența utilizării resurselor disponibile, contribuind la o gestionare mai inteligentă a fluxurilor de trafic urban.

2. Studiu bibliografic

Sistemele de semaforizare sunt dispozitive pozitionate la intersectii rutiere, treceri de pietoni si alte locatii pentru a controla fluxurile de trafic de semnalizare. Un semafor este un dispozitiv care este proiectat pentru a controla fluxul de trafic prin utilizarea a trei lumini colorate. Aceste lumini sunt in principal verde, galben si rosu.

Semaforizarea conventionala, care este cea mai des intalnita cand vine vorba de dirijarea traficului, presupune functionarea sistemului dupa un program si niste parametrii prestabiliti. Se doreste implementarea unui sistem care sa poata lua decizii in functie de anumiti parametri masurati in timp real.

Pentru realizarea acestei simulari vom folosi limbajul Python impreuna cu tehnologia Pygame in PyCharm IDE.

Pygame este o bibliotecă Python populară, utilizată pentru dezvoltarea de jocuri video. Este gratuită, cu sursă deschisă și multi-platformă, fiind un înveliș în jurul Simple DirectMedia Library (SDL). Abstractizarea funcțiilor SDL oferite de Pygame face foarte ușoară dezvoltarea de aplicații multimedia folosind Python.

Cu ajutorul acestei biblioteci vom reconstrui intr-un mod simplificat o situatie din trafic, o intersectie in cruce semaforizata. In timp real vor fi animate diferite masini pentru a simula functionalitatea intersectiei semaforizate.

Initial, un singur semafor va permite masinilor sa mearga mai departe, in timp ce restul asteapta. Asta va pastra ordinea de circulatie a masinilor.

3. Design si analiza

Python este un limbaj de programare de nivel înalt, de uz general. Filozofia sa de proiectare pune accentul pe lizibilitatea codului prin utilizarea unei indentări semnificative.

Python este dynamically typed(tipul variabilelor se pot schimba pe parcurs) si garbage-collected(sterge din memorie obiectele care nu mai sunt folosite). Suportă mai multe paradigme de programare, inclusiv programarea structurată (în special procedurală), orientată pe obiecte și funcțională. Este adesea descris ca un limbaj cu "baterii incluse" datorită bibliotecii sale standard cuprinzătoare.

Python este menit să fie un limbaj ușor de citit. Formatarea sa folosește adesea cuvinte-cheie în limba engleză, în timp ce alte limbi folosesc punctuația. Spre deosebire de multe alte limbaje, Python nu folosește paranteze ondulate pentru a delimita blocurile, iar punctele și virgulele după declarații sunt permise, dar rar folosite. Are mai puține excepții sintactice și cazuri speciale decât C sau Pascal.

Simularea sistemului consta in semaforizarea unei intersectii in cruce, unde traficul de pe fiecare ramura este controlat de catre un semafor, astfel ca avem 4 semafoare.

Acestea se schimba intermitent intre ele pentru a permite un trafic controlat, ca urmare am ales ca un singur semafor sa fie verde,in jur de 10 secunde, pentru a mentine ordinea in trafic.

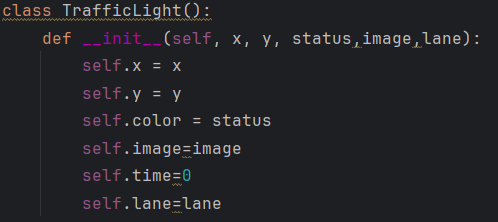
Drumul, masinile si semafoarele le-am reprodus cu ajutorul unor imagini png create in Procreate. Acestea le-am incarcat in proiect cu ajutorul librariei Pygame.

Masinile sunt generate periodic pentru a simula traficul din viata reala. Pentru fiecare directie(sus,jos,stanga,dreapta) verificam starea semafoarelor, respectiv a masinilor. Daca semaforul e verde masinile pot sa mearga in continuare. Insa, daca semaforul este rosu,acestea pot sa mearga in continuare pana la linia de stop(sau masina din fata), si doar dupa ce semaforul corespunzator masinilor devine verde, acestea pot sa isi continue drumul. Daca masinile au trecut de semafor, acestea pot merge in continuare indiferent de starea semaforului de care depindeau anterior.

Timpul va fi realizat cu ajutorul modului time care tine de Python.

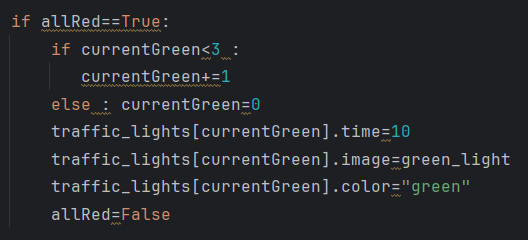
4. Implementare

Implementarea simularii a fost realizata in Python in Pycharm IDE. Principalele clase sunt Car si TrafficLight, care sunt principalele obiectele a sistemului.

 4.1 Implementarea semafoarelor:

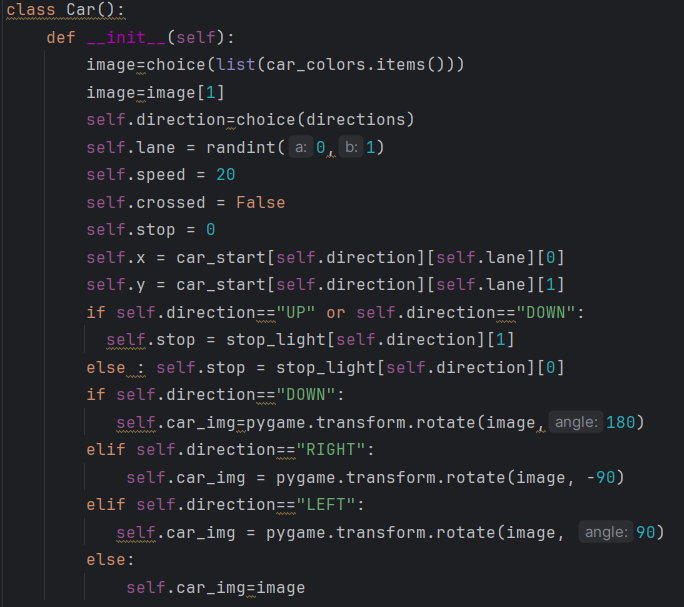
Acestea au 3 stari : rosu, galben si verde. Cele 4 semafoare se afla initial in starea rosu, iar pe rand fiecare va deveni verde pentru 10 secunde, respectiv galben pentru 3 secunde.

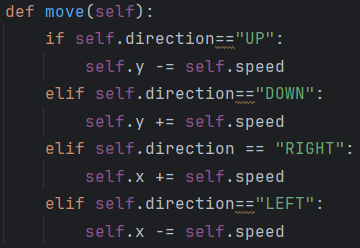
Cand timpul semaforului verde se scurge, alegem urmatorul semafor.



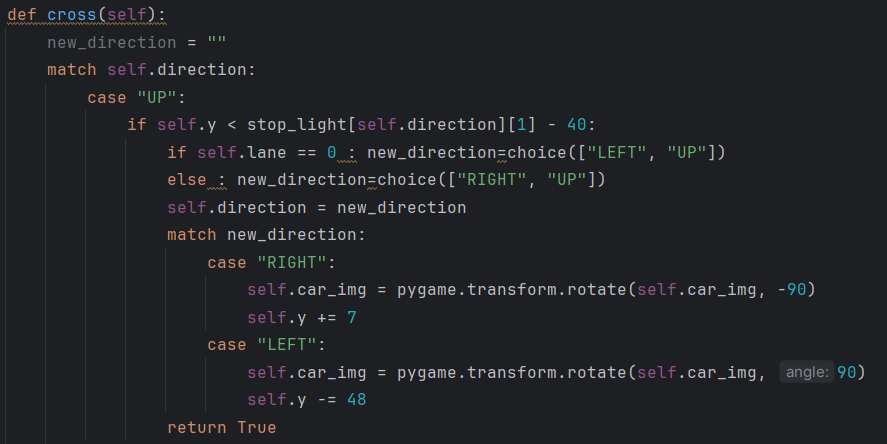
4.2 Implementarea masinilor:

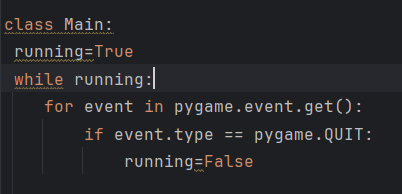
Masinile sunt generate random sa porneasca din una dintre cele 4 directii(sus, jos, stanga, dreapta) . Coordonatele x si y ne ajuta sa cunoastem si sa controlam pozitia masinilor. Variabila direction ne spune directia masinii, iar lane ne spune banda pe care se afla. Stop ne spune unde trebuie sa opreasca masina inainte de a trece de semafor(de exemplu daca are masini in fata sa pastreze distanta). Componenta speed ne spune viteza masinii, adica cu cati pixeli mutam imaginea pe ecran. Variabila crossed ne indica daca masina a trecut sau nu de semafor, pentru ca aceasta se comporta diferit in cele 2 scenarii.



Pentru ca masinile sa se deplaseze, ii modificam coordonatele in functie de directia ei (coordonata X pentru stanga si dreapta, respectiv Y pentru sus si jos).

Masinile vin pe 2 benzi de circulatie. Cele de prima banda pot sa mearga in dreapta sau sa isi continue drumul in fata, iar cele de pe a doua banda pot sa faca stanga sau sa isi continue drumul inainte. Dupa ce masina trece de semafor, este ales aleator o noua directie.



 Pentru ca simularea sa mearga, folosim o bucla while. Ca sa incheiem simularea, folsim un event (QUIT) din biblioteca Pygame care opreste bucla.

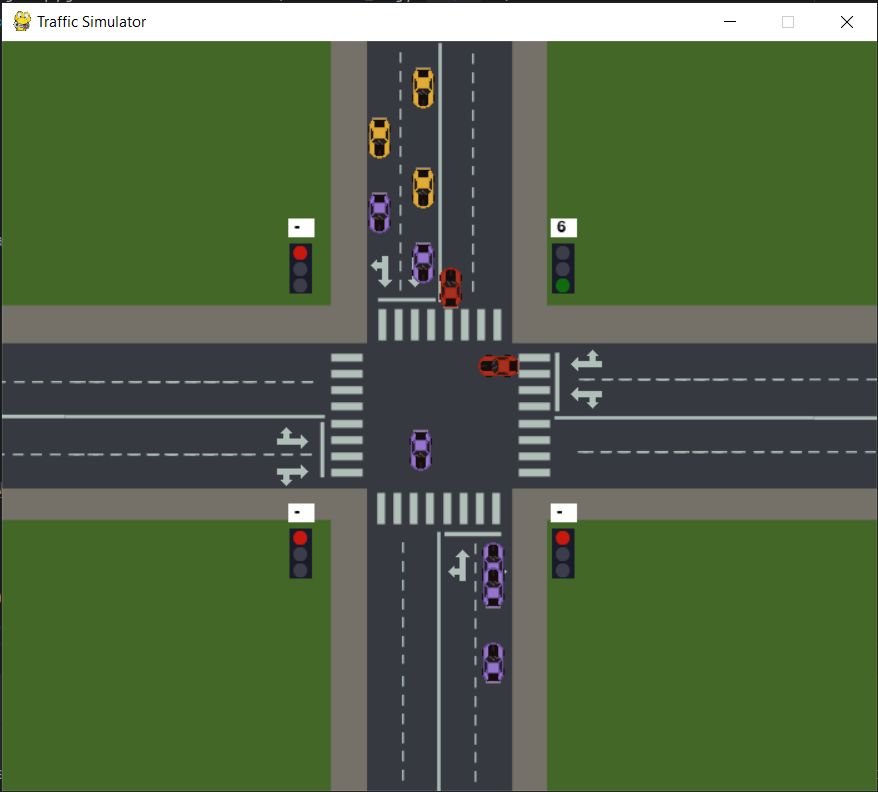
4.3 Interfata grafica(GUI):

Interfata grafica ofera o reprezentare a unui scenariu de simulare si arata o intersectie in forma de cruce. Reglarea circulatiei se realizeaza prin patru semafoare corespunzătoare fiecarui drum care intra in intersectie. Pentru a oferi informatii despre trafic, interfata grafica afiseaza timpul ramas pentru vehiculele care au dreptul sa mearga in directia specificata. Prin urmare, utilizatorii pot urmarii dinamica traficului în cadrul simulării și pot accesa vizual informații importante privind gestionarea intersecțiilor și secvențele de schimbare a semnalului.



5. Testare

In testarea aceseti aplicatii, am observat cum semafoarele comuta pe rand, avand starea de verde pentru 10 secunde, de galben pentru 3 secunde, iar apoi de rosu. De asemenea, se poate vedea cum masinile nu trec de linia de oprire cand au rosu la semafor. Cele care au verde isi continua drumul(merg in fata sau isi schimba directia in functie de banda pe care se afla).



6. Concluzii

Scopul acestui proiect a fost de a implementa o simulare a unui sistem inteligent de comutare a semafoarelor. Prin acest proiect, am explorat principiile de baza ale programarii in Python, lucrul cu interfete grafice si implementarea unui sistem de simulare a traficului.

Am reusit sa simulez in mod dinamic o intersecție rutiera, gestionand în acelasi timp conditiile de semnalizare si interactiunile vehiculelor in conformitate cu regulile de circulatie.

Aceasta implementare a demonstrat functionalitatea corecta a algoritmului inteligent de comutare a semaforelor si a simulat cu succes comportamentul realist al unei masini. In timpul acestui proiect, am dezvoltat abilitati practice in programarea Python, si manipularea unei interfete grafice.

7.Bibliografie

1. https://www.pygame.org/docs/
2. https://www.python.org/doc/