

**UNIVERSITATEA „ALEXANDRU IOAN CUZA” DIN IAȘI**

**FACULTATEA DE INFORMATICĂ**



**LUCRARE DE LICENȚĂ**

**My driving school**

propusa de  
Balan Sabin-Marian

**Sesiunea:** februarie, 2019

Coordonator științific  
Conf. dr. Vitcu Anca

**UNIVERSITATEA „ALEXANDRU IOAN CUZA” DIN IAȘI**

**FACULTATEA DE INFORMATICĂ**

**LUCRARE DE LICENȚĂ**

**My driving school**

propusa de  
Balan Sabin-Marian

**Sesiunea:** februarie, 2019

Coordonator științific  
Conf. dr. Vitcu Anca

**Avizat,**

**Îndrumător Lucrare de Licență**

Titlul, Numele și prenumele \_\_\_\_\_

Data \_\_\_\_\_ Semnătura \_\_\_\_\_

**DECLARAȚIE privind originalitatea conținutului lucrării de  
licență/diplomă/disertație/absolvire**

Subsemnatul(a) .....

domiciliul în .....

născut(ă) la data de ....., identificat prin CNP .....

absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de .....

specializarea ....., promoția .....

declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul: \_\_\_\_\_

elaborată sub îndrumarea dl. / d-na \_\_\_\_\_, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență/diplomă/disertație/absolvire să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi, .....

Semnătură student .....

## DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „Alegerea profului de licență”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza Iași să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, --.02.2019

Absolvent Balan Sabin-Marian,

---

## Cuprins

MOTIVATIE.....	6
INTRODUCERE.....	7
Capitolul I. Aplicatii similare.....	9
Capitolul II. Tehnologii utilizate.....	11
Capitolul III. Descrierea aplicatiei.....	12
III.1. Interfata.....	15
III.2. Intrebarile.....	18
III.3. Arhitectura situatiilor practice.....	19
III.4. Sistemul de path-uri.....	21
III.5. Sistemul de lumini.....	22
III.6. Masinile din trafic.....	24
III.7. Controlul jucatorului.....	29
Concluzii finale si directii viitoare.....	30
BIBLIOGRAFIE.....	31

## Motivatie

În România, numărul persoanelor care încep școala de șoferi în vederea obținerii permisului de conducere este într-o continuă creștere dar există o problemă destul de arzătoare cu care se confruntă mai toate județele din țară, și anume scăderea ratei medii de promovabilitate până la procente care sunt cu mult sub cum ar trebui să arate.

Privind harta cu rata de promovabilitate pe județe realizată în baza informațiilor puse la dispoziție de [drpciv.ro](http://drpciv.ro) și care ia în calcul toate școlile de șoferi, PFA-urile și toți instructorii autorizați putem trage concluzia că în județul Iași se înregistrează cea mai mică rată de promovabilitate din întreaga țară.

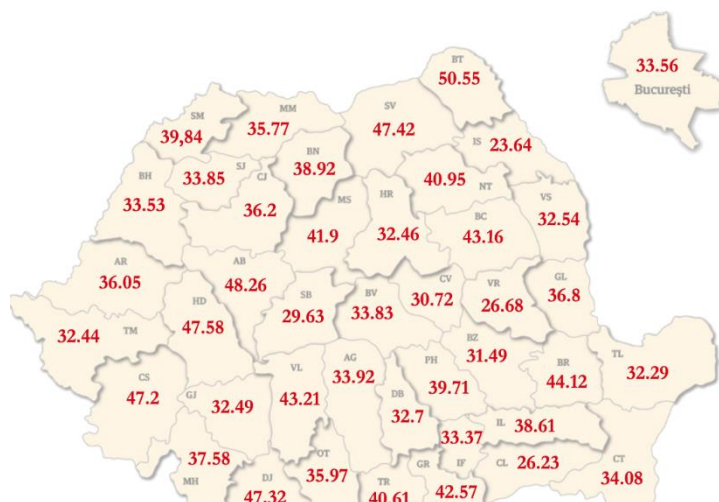


Fig. 1 Rata de promovabilitate a școlilor de șoferi pe județe  
<https://scoaladesoferi.odat.ro/rata-promovabilitate-scoli-auto/>

Pentru multe persoane legislația rutieră este un domeniu vast care ridică numeroase dificultăți de înțelegere și învățare, metodele actuale având o rată de succes destul de scăzută așa cum a fost prezentat și în statistica e mai sus.

O solutie pentru aceasta problema ar putea consta in crearea unui mediu de invatare care sa imbine atat intrebarile de ordin teoretic cat si situatii de ordin practic prin care utilizatorul sa invete regulile de circulatie intr-un mod distractiv si usor. Jucatorul va lua contact cu numeroase situatii din trafic iar deciziile sale vor influenta direct soarta jocului, traseul creat punandu-l intr-o ipostaza asemanatoare cu cea din timpul unui examen auto practic. Astfel, jucatorul este constrans sa conduca masina cat mai corect cu putinta si sa raspunda cu atentie la intrebarile de legislatie, miza fiind promovarea sau picarea examenului auto simulat. De asemenea, adaugarea unui scor si a unor pedepse mai severe precum resetarea nivelului starnesc spiritul competitiv al utilizatorului.

## **Introducere**

Oamenii au avut dintotdeauna o puternica legatura cu orice concept legat de jocuri, acestea fiind privite drept modalitati de divertisment dar si ca fiind educative, contribuind la dezvoltarea abilitatilor fizice si a gandirii.

Jocurile de tip simulator si-au facut aparitia la mijlocul anilor 1980 atunci cand dezvoltatori precum Code masters si Oliver Twins au lansat pe piata primele jocuri care sa contina "Simulator" in titlu: BMX Simulator (1986), Grand Prix Simulator (1986) si Pro Boxing Simulator (1988). Acestia s-au inspirit din titluri best-seller care la randul lor se bazau pe sporturi reale cum ar fi fotbalul sau cursele de biciclete BMX.

Scopul acestui tip de jocuri, dupa cum o spune si numele, este de a simula o experienta pe cat de real posibil iar jocurile ulterior create SimLife si SimEarth au fost capabile sa invete jucatorii bazele geneticii si ecosistemelor globale.

In timp ce multi considera ca jocurile simulator au inceput cu Will Wright si al sau SimCity in 1989, adevaratul stramos al acestui gen a fost Fortune Builder, lansat in 1984 de ColecoVision.

În anul 2016, jocurile de tip simulator ocupau locul 6 în topul celor mai apreciate genuri iar tendința este de o uriașă creștere totodată cu apariția tehnologiilor precum cele de realitate virtuală și realitate augmentată.

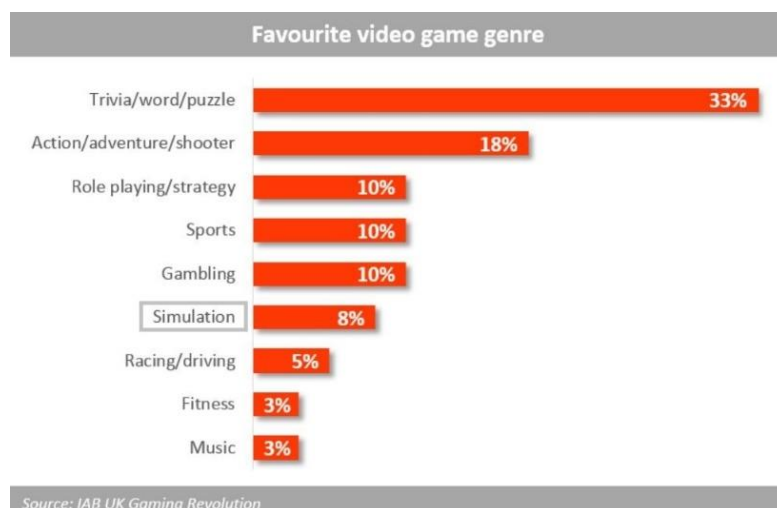


Fig. 2 Topul categoriilor de jocuri video

De ce sunt jocurile simulator atât de apreciate în prezent? Deoarece acestea transpun jucătorul într-un mediu ales de el, foarte asemănător cu realitatea, în care poate experimenta o varietate mare de activități și trăiri care pot fi împartite în următoarele categorii:

- Biological simulation : cu unul din cele mai cunoscute titluri “Goat Simulator” care are peste 5.5 milioane de descărcări doar pe platforma Android;
- Social simulation: cu faimoasa franciză The Sims care s-a bucurat de un succes enorm cu vânzări de peste 200 de milioane de copii la nivel global;
- City-Building: SimCity (1989-prezent cu ultimul titlu lansat în 2014 “SimCity BuildIt”)
- Sports management: “Top Eleven” care are peste 50 milioane de descărcări doar pe platforma Android;
- Sports simulation: de unde nu putem exclude FIFA care este lansat anual de către cei de la Electronic Arts încă din 1993;



- Farm simulator: “Farming simulator” unde din 2008 pana in prezent jucatorii experimenteaza pe toate platformele cele mai importante ce inseamna cultivarea pamantului;
- Medical simulation;
- Vehicle simulation:
  - Racing simulation: “Forza horizon 4” castigator al premiului de “Cel mai bun joc de racing” din 2018;
  - Ship simulator: “World of Warships” lansat in septembrie 2015;
  - Flight simulator: “Ace combat” un alt titlu longeviv (1995-prezent);
  - Tank simulation: “World of Tanks” cu 75 milioane de jucatori inregistrati;
  - Truck simulator: “Euro truck simulator” cu 300.000 de copii vandute in Europa.

## **I.     Aplicatii similare**

In prezent, exista putine simulatoare auto care fac un scop principal din prezentarea elementelor teoretice ale legislatiei si respectarea unor restrictii pe parcursul traseului. Majoritatea ofera o libertate mult prea mare jucatorului care nu invata foarte multe lucruri doar plimbandu-se cu masina pe o harta de tipul open-world.

Cele mai notabile titluri sunt City Car Driving (PC) si Driving School 2017 (Android & Ios). Ambele aduna un numar consistent de jucatori si au recenzii suficient de bune insa nu sunt orientate spre a invata utilizatorul noi reguli de circulatie.



Fig. 3 Driving School 2017

<https://play.google.com/store/apps/details?id=com.ovilex.drivingschool2017&hl=en>

<https://city-car-driving.de.softonic.com/>



Fig. 4 City Car Driving

Jocurile precum Driving School Simulator care ofera o experienta aseamantoare cu cea a scolii de soferi au in mare parte recenzii negative.



Fig. 5 Driving School Simulator

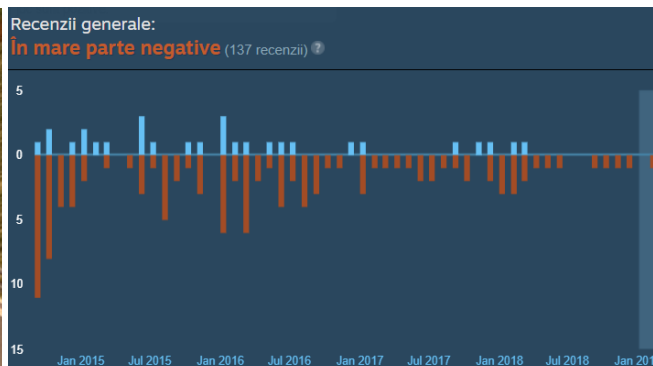


Fig. 6 Recenzii Driving School Simulator

[https://store.steampowered.com/app/273730/Driving\\_School\\_Simulator/](https://store.steampowered.com/app/273730/Driving_School_Simulator/)

O alta solutie este reprezentata de simulatoarele auto pentru antrenament si cercetare dar acestea nu sunt accesibile oricui. Aceste simulatoare performante sunt prezente in putine dintre scolile de soferi din Romania, ele fiind folosite in principal pentru antrenamentul soferilor de pe ambulanta, camioane, autobuze etc.

Principalele utilizari:

- Antrenarea si testarea soferilor incepatori;

- Antrenarea si testarea soferilor profesioniști;
- Antrenarea soferilor pentru situatii critice;
- Analiza comportamentului soferului;
- Analiza reflexelor soferului;
- Evaluarea performantelor soferului in diferite situatii (explozarea unui cauciuc, gheata pe suprafata carosabila etc.);
- Evaluarea performantelor soferului in diferite conditii (consum de alcool, consum de droguri, lipsa de somn etc.).



Fig. 7 Simulatoare auto profesioniști

<https://irishtimes.com>

[https://en.wikipedia.org/wiki/Driving\\_simulator](https://en.wikipedia.org/wiki/Driving_simulator)

## II. Tehnologii utilizate



Unity este un game engine dezvoltat de catre cei de la Unity Technologies si lansat pentru prima oara in iunie 2015 la “Apple Inc.’s Worldwide Developers Conference ca si un editor de jocuri exclusiv pentru Mac OS X. In prezent, engine-ul a fost extins pentru a suporta crearea de jocuri pentru un numar de pana la 27 de platforme, printre cele mai importante numarandu-se Windows, macOS, Android, iOS, Playstation4, Xbox One, Android TV etc.

Editorul celor de la Unity Technologies ofera utilizatorilor sai posibilitatea de a crea atat jocuri 2D cat si 3D, utilizand limbajul de programare C# drept principal API pentru scripting atat pentru plugin-uri cat si pentru jocuri in sine. De asemenea, editorul este dotat cu multiple tooluri care faciliteaza editarea rapida a jocurilor prin metode de tipul drag and drop si prin functionalitati precum “Play mode” folosit pentru previzualizarea muncii tale in timp real.

Aplicatiile facute in Unity au ajuns pe nu mai putin de 2.7 miliarde de dispozitive la nivel global si au fost instalate de mai mult de 24 miliarde de ori in ultimele 12 luni. Astfel, Unity este unul dintre cei mai importanti jucatori de pe piata revolutionand in mod constant industria jocurilor.

### **III. Descrierea aplicatiei**

Aceasta lucrare are ca obiectiv crearea unui joc 3D de tip simulator, accesibil unui numar cat mai mare de utilizatori, cu numele “My Driving School” pentru sistemul de operare Windows. Jocul este creat in totalitate in engine-ul Unity prin intermediul caruia s-a construit intreaga harta si s-a dezvoltat comportamentul obiectelor. Jocul a fost construit pentru a fi cat mai simplu si intuitiv, aplicatia verificand in mod special pregatirea teoretica a utilizatorului.

Accentul lucrarii a picat mai mult pe partea de programare a situatiilor din trafic. Prin urmare, in vederea construirii mediului de joc s-au importat si adaptat urmatoarele pachete de modele gratis:

- Low Poly Street Pack – modelat de catre cei de la Dynamic Art 3D

Acest pachet a fost utilizat in vederea modelarii a tot ceea ce tine de infrastructura stradala (strazi, poduri, trotuare, semafoare, intersectii, etc.) fara a fi aduse modificari elementelor.

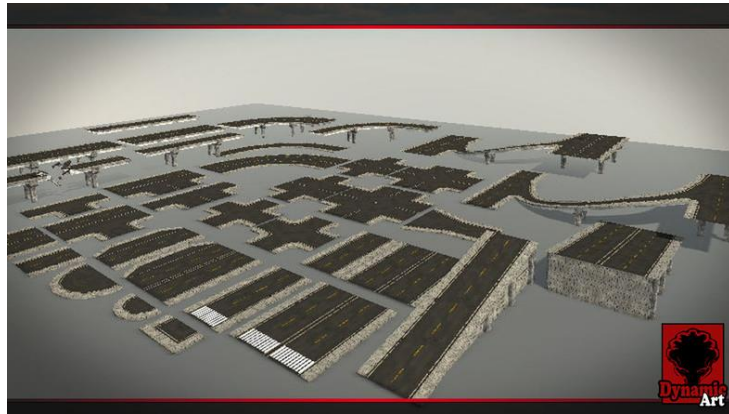


Fig. 8 Low poly street pack

<https://assetstore.unity.com/packages/3d/environments/urban/low-poly-street-pack-67475>

- Simple City Pack Plain – modelat de catre cei de la Polygon Land

Acest pachet a fost utilizat pentru a construi decorul urban al jocului. Modificari s-au adus doar in privinta materialelor de pe obiecte si a culorilor utilizate pentru prefabricate.



Fig. 9 Simple City Pack Plain

<https://assetstore.unity.com/packages/3d/environments/urban/simple-city-pack-plain-100348>

- Simple Cars Pack – modelat de catre cei de la Myxerman

Acest pachet contine modelele masinilor utilizate in joc care au fost adaptate ulterior pentru putea primi un anumit comportament dorit in urma programarii scripturilor necesare. S-au adaugat lumini de semnalizare si de frana pe masini si au fost modificate anumite texturi.



Fig. 10 Simple Cars Pack

<https://assetstore.unity.com/packages/3d/vehicles/land/simple-cars-pack-97669>

- Traffic signs – modelat de catre Lapi Games

Din acest pachet s-au utilizat un numar foarte mic de indicatoare rutiere, unora fiindu-le adaptata textura dupa nevoie.



Fig. 11 Traffic Signs



<https://lapigames.itch.io/unity-city-traffic-signs>

- BodyGuards – modelat de catre cei de la Batewar

Din acest pachet s-a folosit un singur model uman care in joc a are rolul de politist rutier.



Fig. 12 BodyGuards

<https://assetstore.unity.com/packages/3d/characters/humanoids/bodyguards-31711>

De mentionat este faptul ca pachetele importante contin doar modelele 3D, fara niciun fel de animatie sau algoritm, implementarea functionalitatilor din joc fiind realizata in totalitate de catre mine.

### III.1. Interfata

Prin interfata, intelegem toate informatiile care apar pe ecran inaintea si in timpul exectuiei jocului. O interfata cat mai buna si mai prietenoasa imbunatateste cu mult experienta de joc. Din acest motiv, in realizarea interfetei am urmarit urmatoarele lucruri:

1. Interfata imi spune doar ce trebuie sa stiu cand trebuie sa stiu?
2. Informatiile de care am nevoie sunt usor de gasit, fara a petrece prea mult timp in a le cauta?
3. Se poate utiliza interfata fara a avea nevoie de niciun fel de instructiuni regasite in alta parte?

4. Cat timp trebuie sa se pastreze informatiile pe ecran?
5. Exista elemente inutile sau modalitati mai simple de a afisa anumite informatii?

## Meniul

Meniul jocului reprezinta o lista de comenzi care pot fi selectate cu ajutorul mouseului. Am optat pentru un meniu cat mai simplu, fara informatii inutile si fara alte submeniuuri. Acesta contine butonul de start al nivelului selectat, butoane de selectie pentru nivelele jocului si un buton de instructiuni. La apasarea butonului de instructiuni, pe ecran apare o imagine cu toate informatiile necesare cu privire la control. Aceasta poate fi inchisa prin reapasarea butonului de instructiuni sau prin pornirea nivelului din butonul de start.

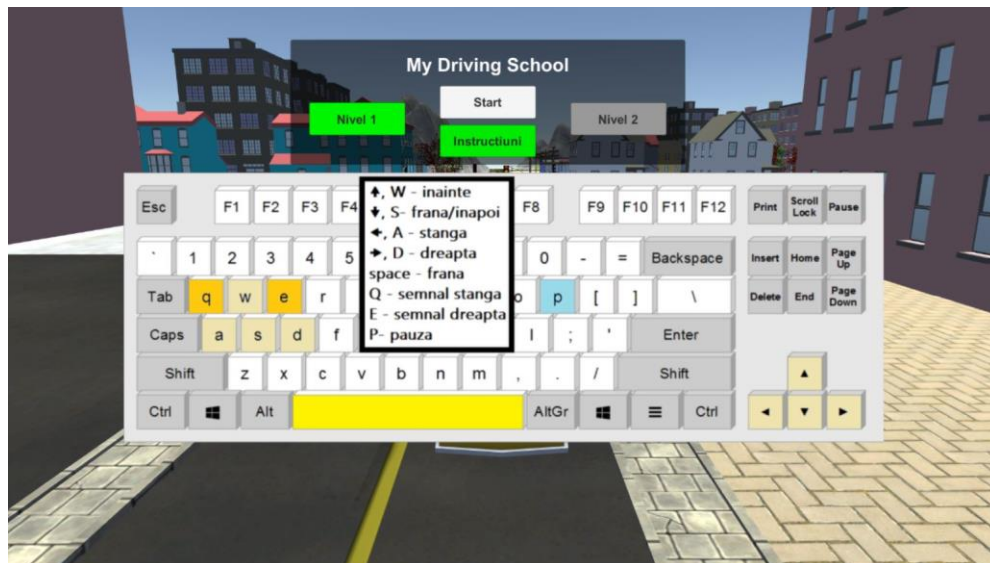


Fig. 13 Meniul jocului

## Interfata in timpul jocului

Pe intreg parcursul jocului este necesara afisarea unei serii de informatii indispensabile jucatorului. Acestea apar pe ecran doar cand sunt necesare si sunt inlaturate imediat ce nu mai contin indicatii relevante. Stilul interfetei urmareste aceleasi linii ca si meniul, cu informatii cat mai simple si cat mai relevante, afisate pe un fundal negru transparent. Scopul lor este de a tine



jucatorul la curent cu rezultatul deciziilor sale si de a-i oferi indicatii despre cum trebuie sa procedeze in continuare.

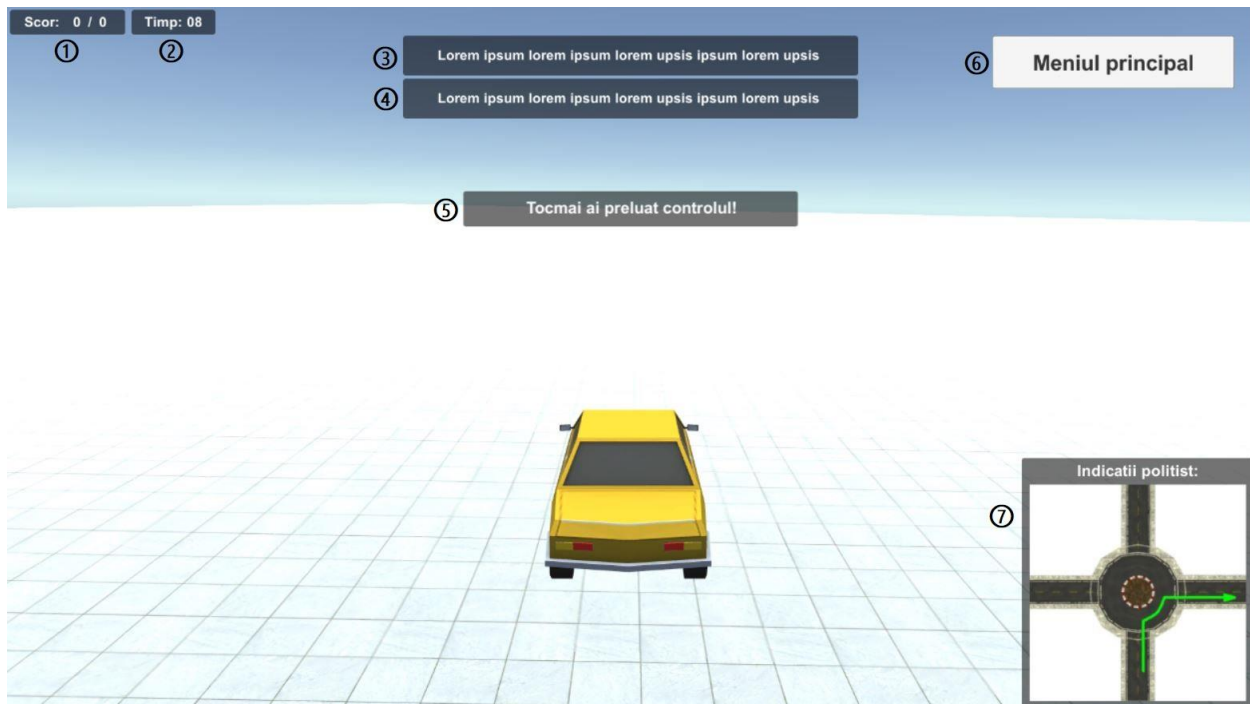


Fig. 14 Interfata in timpul jocului

1. Scorul – determinat de numarul intrebarilor la care jucatorul raspunde corect si numarul manevrelor executate corespunzator. Apare pe ecran pe tot parcursul jocului;
2. Timpul – un temporizator descrescator exprimat in secunde care exprima timpul ramas jucatorului pentru executia unei manevre. Acesta apare pe ecran doar in situatiile in care este necesar si dispare dupa ce manevra a fost completa. Expirarea timpului duce la picarea examenului. Pe toata lumea din trafic deranjeaza un sofer mult prea lent care petrece prea mult timp in luarea anumitor decizii;
3. Zona principala de mesaje – apare pe ecran doar cand este necesare afisand informatii precum “Felicitari! Ai executat manevra corect!”, “Din pacate nu ai semnalizat corespunzator!” si altele;
4. Zona secundara de mesaje – apare pe ecran doar cand sunt necesare indicatii suplimentare. Ex.: la schimbarea benzii apare mesajul “Nu uita sa semnalizezi pe tot parcursul manevrei!”;
5. Anuntul cu privire la controlul masinii – de fiecare data cand masina trece din modul manual pe modul automat jucatorul este anuntat prin acest mesaj;

6. Butonul de meniu principal – apare doar in cazul in care jucatorul incalca grav o regula de circulatie care conduce la incheierea nivelului. Apasarea deschide meniul jocului;
7. Zona de indicatii – apare pe ecran cand jucatorul ajunge intr-o intersectie si dispare cand nu este necesara. Aceste indicatii tin locul indicatiilor oferite de un politist intr-un examen auto din viata reala.

Mesajele temporare dispar dupa 3 secunde de la afisare.

La finalul jocului, zona de informatii se mareste afisandu-se rezultatele jucatorului la cele doua capitole: manevre si intrebari.

### **III.2. Intrebarile**

Pe masura ce jucatorul avanseaza pe traseu, acestuia i se vor adresa numeroase intrebari teoretice. La fel ca in chestionarele de la examenul teoretic concepute si oferite de DRPCIV (Directia Regim Permise de Conducere si Inmatriculare a Vehiculelor) intrebarile au 3 variante de raspuns dintre care unu, doua sau toate trei pot fi corecte. Jucatorul selecteaza raspunsurile sale prin bifarea unui checkbox aferent dupa care apasa butonul “Raspunde”. Verificarea se realizeaza pe loc, un mesaj de informare apare pe ecran cu scopul de a anunta daca variantele alese sunt sau nu cele corecte in timp ce textul celor 3 raspunsuri isi va schimba culoarea dupa cum urmeaza:

- text verde – raspuns corect si selectat de catre jucator;
- text rosu – raspuns gresit selectat de catre jucator;
- text galben – raspuns corect dar neselectat de catre jucator.

In urma verificarii raspunsurilor, scorul jucatorului se actualizeaza, butonul “Raspunde” se transforma in “Continuare” iar actionarea acestuia duce la disparitia intrebarii de pe ecran si continuarea gameplay-ului.

Intrebarile sunt salvate ca fisiere in folderul “questions” avand urmatorul format : intrebarea, raspuns 1, raspuns 2, raspuns 3, valoare raspuns 1, valoare raspuns 2, valoare raspuns 3 (de tip bool : 0 – raspuns incorect, 1 – raspuns corect).



Fig. 15 Exemplu intrebare

### III.3. Arhitectura situatiilor practice

Fiecare situatie practica in care jucatorul va fi transpus in acest joc de simulare este un game object de sine statator alcatuit la randul sau din numeroase alte obiecte (masini, semafoare) cu roluri bine definite. Situatiile sunt controlate de o serie de triggere reaspondite pe toata harta care au rolul de a activa sau dezactiva obiectele corespunzatoare in functie de anumite criterii. Exista situatii in care se controleaza luminile unui semafor, situatii in care pe sosea trebuie sa apara indicatii de orientare, situatii in care masinile din trafic trebuie sa realizeze anumite actiuni si multe altele. Aceste lucruri nu ar putea fi posibile fara obiectul denumit “case controller” care manageriaza toate aceste situatii in functie de unde se afla playerul la un moment dat.

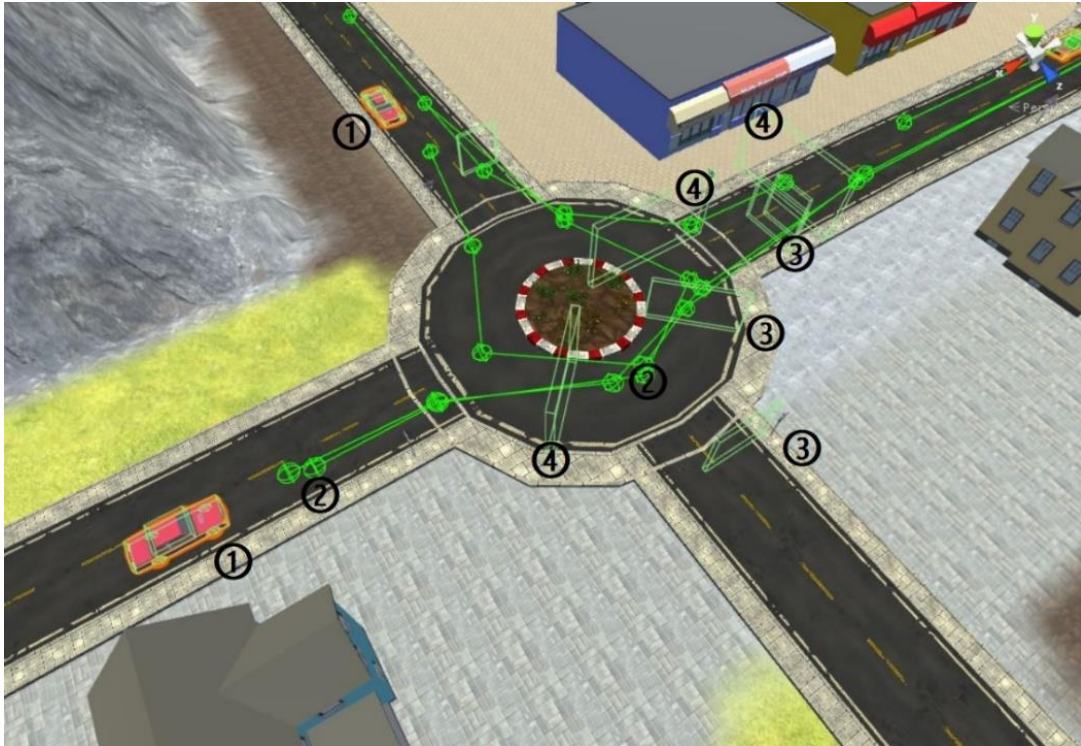


Fig. 16 Exemplu situatie

În imaginea de mai sus avem prezentată o situație din joc în care jucătorul deține controlul asupra mașinii fiind nevoit să intre într-un sens giratoriu și să îl părăsească la prima ieșire. În această situație putem observa următoarele:

- ceilalți participanți la trafic (1);
  - Traseul celorlalți participanți la trafic (2);
  - Serie de trigger-uri utilizate pentru a verifica dacă jucătorul a semnalizat în mod corespunzător manevra (3);
  - Serie de trigger-uri utilizate pentru a verifica dacă jucătorul a respectat sau nu indicațiile (4).
- De menționat că nerespectarea indicațiilor conduce la pierderea examenului și încheierea nivelului.



### III.4. Sistemul de path-uri

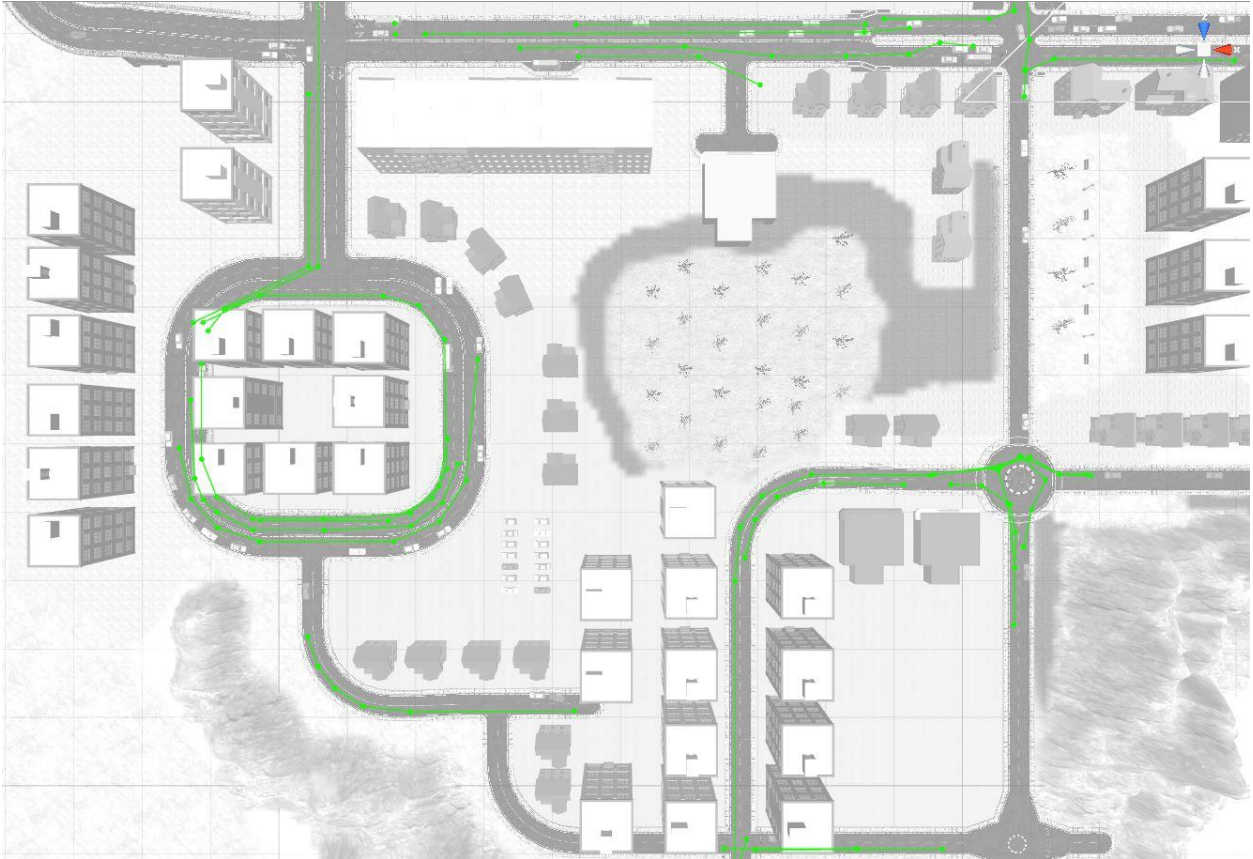


Fig. 17 Sistemul de path-uri

Un path reprezintă traseul standard pe care o anumită mașină trebuie să îl urmeze. Un obiect de tipul path este un game object gol cu un anumit număr de copii (noduri) care la rândul lor sunt tot obiecte goale. Crearea path-urilor și salvarea lor are loc la momentul rularii prin executia scriptului 'Path.cs'. Acesta are rolul de a lua nodurile și a le stoca într-un vector în ordinea în care au fost citite și de a crea o reprezentare vizuală a path-ului creat. Reprezentarea vizuală a fost indispensabilă pe parcursul modelării, ușurând cu mult procesul de design al sistemului de path-uri. Pentru a evita orice tip de încurcatură și aglomerarea inutilă a strazilor, am conceput ca path-ul să devină vizibil doar atunci când acesta este selectat. Pentru reprezentarea nodurilor am folosit un wireframe iar pentru liniile ce reprezintă calea propriu-zisă am creat segmente între noduri.

Culoarea aleasa este verde pentru ca path-ul sa iasa usor in evidenta pe harta atunci cand este selectat in vederea unei modificari.

Prin sistem de path-uri intelegem un ansamblu de elemente care formeaza un intreg organizat, coordonat si influentat de aceleasi entitati superioare lor in vederea indelpinirii unei functii comune (ghidarea masinilor din trafic). Obiectele care controleaza daca un path este utilizat sunt cele de tip ‘case’ prezentate anterior care definesc, in functie de situatia in care se afla jucatorul, ce obiecte trebuiesc folosite.

### **III.5. Sistemul de lumini**

Sistemul de lumini al unui vehicul consta in dispozitive de luminare si semnalizare montate sau integrate in fata, spatele, lateralele masinii si in unele cazuri speciale chiar deasupra. Desi exista o diversitate mare a variantelor constructive, toate vehiculele trebuie sa contina urmatoarele mijloace de iluminare si semnalizare:

- Partea din fata:
  - Faruri cu lumina de intalnire (faza scurta), faruri cu lumina de drum (faza lunga), lampi de pozitie-stationare, lampi de semnalizare a directiei, lampi de avarii (identice cu cele de semnalizare a directiei), lampi de mers inapoi, lampi de iluminare a numarului de inmatriculare – obligatorii;
  - Faruri de ceata (proiectoare) – nu sunt obligatorii.
- Partea din spate:
  - Lampi de pozitie/stationare, lampi de semnalizare a directiei, lampi de avarii, lampi de franare – obligatorii;
  - Lampi de ceata – nu sunt obligatorii.

In joc, accentual pica mai mult pe luminile de frana, luminile de semnalizare a directiei de mers si luminile speciale precum cele ale masinii de politie. Pentru a putea fi creat sistemul de

lumini, a fost necesara modificarea modelelor importate prin atasarea de noi obiecte (cuburi si sfere) care sa indeplineasca rolul blocurilor de lumini. De asemenea, s-au creat si noi materiale utilizate in urmatoarele doua scopuri:

- Imitarea luminilor stinse – materiale normale;
- Imitarea luminilor aprinse – materiale emiseive.

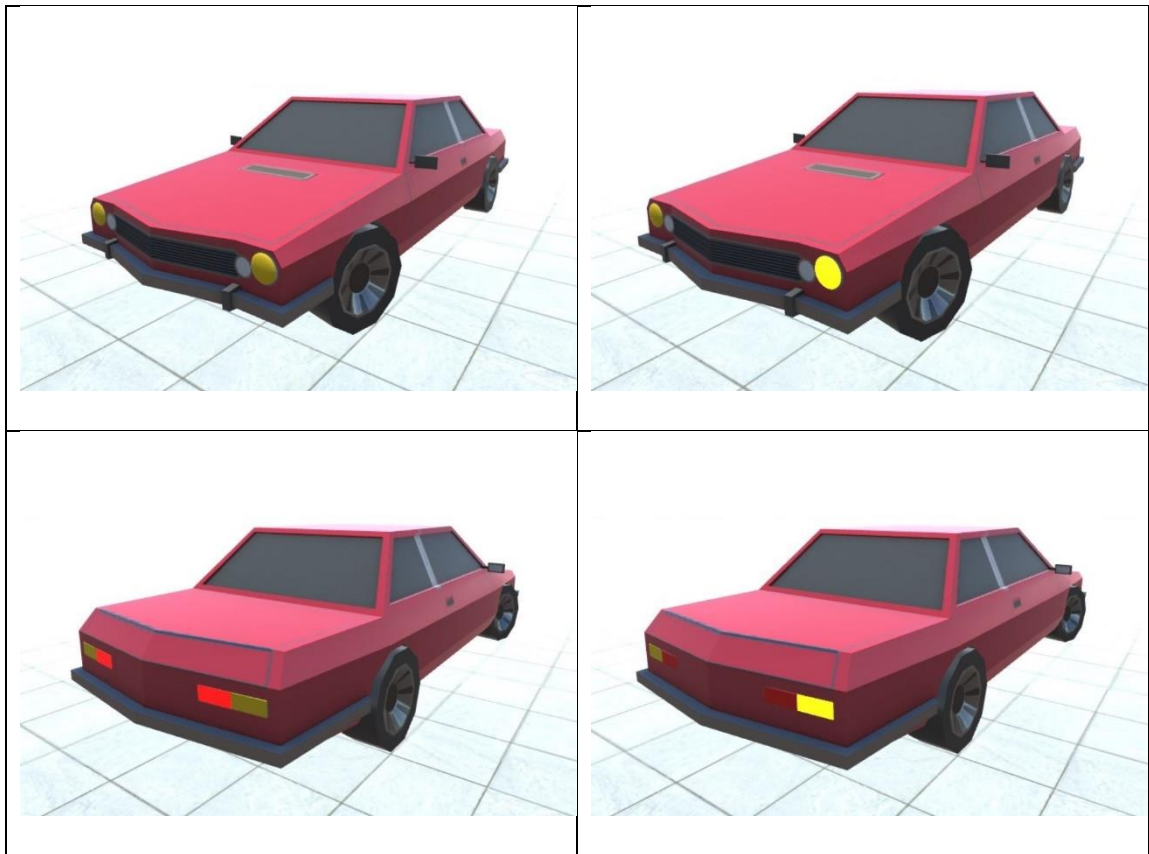


Fig. 18 Sistemul de lumini

In cazul jucatorului, luminile sunt actionate dupa cum urmeaza: luminile de frana se aprind cat timp masina franeaza (apsarea tastelor ‘s’, ‘space’, ‘down arrow’); luminile de semnalizare se aprind si se sting printr-o singura apasare a tastei (‘e’ – semnalizare dreapta, ‘q’ – semnalizare stanga). In cazul celorlalti participanti la trafic, luminile sunt actionate fie automat (luminile de frana si semnalizare in sensul giratoriu), fie prestabilit (semnalele luminoase ale masinii de politie).

### III.6. Masinile din trafic

Masinile din trafic sunt concepute pentru a simula o experienta cat mai apropiata cu cea din trafic. Acestea sunt capabile sa urmeze un traseu prestabilit, sa semnalizeze intentionarea schimbarii directiei de mers, sa franeze, sa evite obstacole si sa evite eventualele coliziuni cu jucatorul sau cu celelalte masini. Pentru a putea fi integrate in mediul virtual, masinile sunt tratate ca niste corpuri din viata reala avand proprietati fizice precum: masa (900-1200kg), centru de greutate, rezistenta aerodinamica, forta centripeta si gravitatie. Astfel, desi din punct de vedere grafic masinile sunt mediocre, din punct de vedere fizic ele respecta cu fidelitate comportamentul unei masini adevarate cu proprietati asemanatoare.

“Creierul” tuturor masinilor se regaseste in scriptul ‘CarEngine’ atasat fiecarui vehicul din joc. Deoarece editorul Unity ofera posibilitatea de editare independenta a variabilelor publice din orice script atasat unui obiect, si scriptul ‘CarEngine’ contine o serie de parametri publici ce faciliteaza personalizarea fiecarei masini in functie de necesitati. Dintre acestea cele mai notabile sunt: cuplul, viteza maxima, unghiul maxim de bracara al rotilor, puterea maxima de franare.

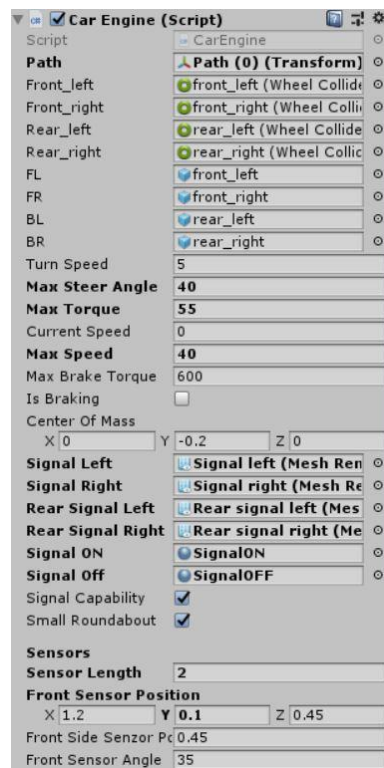


Fig. 19 Managementul variabilelor publice din scriptul “Car Engine” atasat unei masini



‘CarEngine’ controleaza toate functionalitatile masinii dupa cum urmeaza:

- Deplasarea pe un traseu prestabilit:

Miscarea vehiculelor nu este posibila fara un path dat ca parametru scriptului. In functia ‘Start’ (functia care se ocupa cu initializarile necesare) path-ul dat ca parametru este preluat si salvat intr-un vector local care va servi la calcularea urmatorului punct catre care trebuie sa se indrepte masina.

Miscarea vehiculelor pe strada are loc strict prin aplicarea de forte asupra rotilor din spate si prin bracarea rotilor din fata. Pentru realizarea acestui lucru, am aplicat componenta Wheel Collider (contur verde in figura) pe fiecare roata a modelului (contur portocaliu in figura).

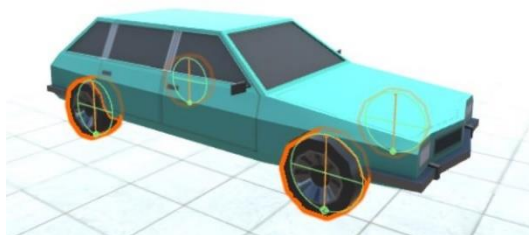


Fig. 20 Componente roti

Deoarece s-a observat un demaraj mai bun, masinile au tractiune spate. Singura conditie necesara pentru a aplica forta asupra rotilor din spate este ca viteza curenta a masinii (calculata prin formula  $currentSpeed = 2 * 22 / 7 * rear\_left.radius * rear\_left.rpm * 60 / 1000$ ) sa nu depaseasca viteza maxima setata.

Pentru determinarea unghiului de bracare al rotilor au fost necesare urmatoarele calcule:

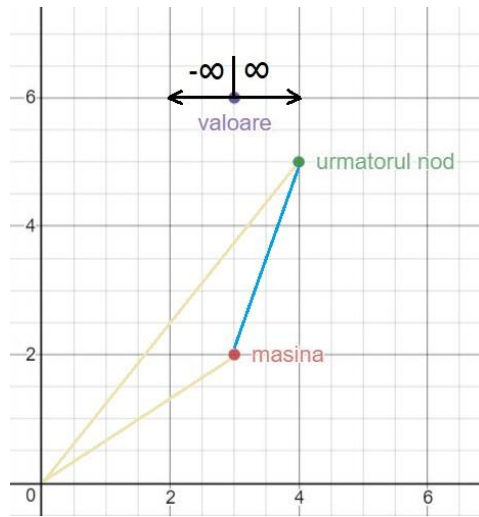


Fig. 21 Determinare unghi bracare

Deoarece in Unity coordonatele oricarui obiect sunt exprimate in functie de origine (0,0), primul lucru pe care trebuie sa il aflam este pozitia urmatorului nod al path-ului in functie de pozitia actuala a masinii. Din fericire, Unity ofera o functie (`InverseTransformPoint`) care calculeaza aceasta valoare si returneaza un vector (vectorul albastru din figura). Dupa calcularea vectorului relativ putem sti deja daca urmatorul punct se afla fie in partea dreapta (x-ul este pozitiv) a masinii fie in partea stanga (x-ul este negativ). In exemplul nostru, vectorul rezultat este (1,3) deci urmatorul punct din path se afla in dreapta masinii.

```
Vector3 relativeVector=transform.InverseTransformPoint(nodes[currentNode].position);
```

In continuare, avem nevoie de o valoare din intervalul (-1,1) pe care sa o inmultim cu variabila publica `maxSteerAngle`<sup>1</sup>. Aceasta valoare o vom obtine luand vectorul relativ definit anterior si impartindu-l la propria sa lungime.

```
relativeVector /= relativeVector.magnitude;
```

Urmatorul pas este sa inmultim valoarea rezultata cu `maxSteerAngle` pentru a obtine unghiul de bracare si aplicarea acestuia asupra colliderelor rotilor din fata.

```
float newSteer = (relativeVector.x / relativeVector.magnitude) * maxSteerAngle;
```

Acest lucru se intampla la fiecare frame in parte iar in momentul in care masina ajunge la o distanta mai mica de 0.5m fata de nodul curent, functia denumita WaypointDistance realizeaza tranzitia la nodul urmator din path.

Efectul vizual de invartire al rotilor si cel de virare au fost obtinute prin transformarea pozitiei si rotatiei fiecarui model de roata in parte in functie de wheel collider-ul sau (functia WheelsVisuals).

- Semnalizarea intentionarii schimbarii directiei de mers:

Semnalizarea automata a intentionarii schimbarii directiei de mers se realizeaza in functie de unghiul de bracara al rotilor din fata. Desi in cele mai multe situatii aceasta modalitate de a stabili cand si in ce directie trebuie sa semnalizeze masina functioneaza corect, am constatat ca exista si o exceptie. Este vorba despre semnalizarea in sensul giratoriu atunci cand masina executa o serie de viraje dar trebuie sa semnalizeze doar stanga in timpul parcurgerii sensului si dreapta la iesire. Daca s-ar pastra regula generala, masina ar semnaliza haotic deoarece rotile ar executa o serie de viraje in directii diferite. Ca rezolvare pentru aceasta situatie am schimbat unghiul definitoriu pentru tipul de semnalizare in functie de dimensiunea sensului giratoriu si am adaugat la fiecare iesire din sens un semnal care sa opreasca semnalizarea.

- Evitarea obstacolelor:

Pentru detectarea obiectelor au fost construiti 5 senzori plasati frontal cu lungimea si unghiul de orientare editabila in functie de scopul si nevoile fiecarei masini. De asemenea, am impartit obiectele din joc in anumite categorii si le-am acordat un tag sugestiv dupa care urmeaza sa se realizeze recunoasterea. In momentul in care un senzor loveste un

obiect cu un tag corespunzator, acesta va intra intr-o faza de evitare ignorand pe moment path-ul prestabilit si ajustand unghiul de bracare al rotilor astfel incat evitarea obstacolului sa fie posibila.

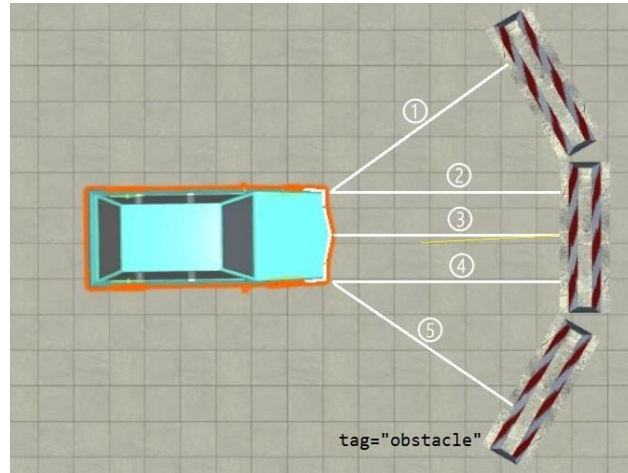


Fig. 21 Senzori masina

1. Obstacol in stanga masinii – deoarece riscul de impact este scazut, se va aplica o corectie de +0.5 inmultita cu unghiul maxSteerAngle.
2. Obstacol in fata masinii – deoarece impactul este iminent, se va aplica o corectie de +1 inmultita cu unghiul maxSteerAngle. Rezultatul va fi un viraj maxim la dreapta pentru a evita obstacolul detectat pe partea stanga.
4. Obstacol in fata masinii - deoarece impactul este iminent, se va aplica o corectie de -1 inmultita cu unghiul maxSteerAngle. Rezultatul va fi un viraj maxim la stanga pentru a evita obstacolul detectat pe partea dreapta.
5. Obstacol in dreapta masinii - – deoarece riscul de impact este scazut, se va aplica o corectie de -0.5 inmultita cu unghiul maxSteerAngle.
3. Obstacol in fata masinii – acest senzor intervine pentru a se lua o decizie in cazul in care atat senzorul cu numarul 2 cat si senzorul cu numarul 4 detecteaza un obstacol. In acest caz, corectia ar fi de  $+1-1=0$  deci masina ar merge inainte. Prin urmare, trebuie sa construim normala obiectului detectat si sa luam o decizie in functie de aceasta.

$$(\text{normala} < 0) ? \text{corectie} = 1 : \text{corectie} = -1$$

### III.7. Controlul jucatorului

Pe parcursul jocului controlul masinii alterneaza intre automat si manual. Prin automat intelegem ca masina urmareste de una singura un anumit traseu si realizeaza anumite manevre cu scopul a prelua atentia jucatorului de pe efectuarea miscarilor si de a o canaliza pe ceea ce se intampla in trafic. Astfel, cand o noua intrebare apare pe ecran, se testeaza capacitatea jucatorului de a aprecia diferite situatii. Cand masina este pe modul manual, jucatorul are responsabilitate totala asupra a ceea ce se intampla, deciziile sale din trafic conducand la urmatoarele consecinte:

- Intreruperea examenului auto – atunci cand utilizatorul a intocmit o greseala majora precum nerespectarea indicatiilor, nerespectarea timpului maxim de efectuare al manevrei, neacordare de prioritate, trecere pe culoare rosie a semaforului etc.
- Continuarea examenului auto dar neacordarea de punctaje – situatie generala prezenta atunci cand utilizatorul nu comite o greseala majora. Acesta isi poate continua traseul insa pierde puncte pentru executia incorecta a manevrelor (ex. Semnalizarea incorecta intr-un sens giratoriu).
- Continuarea examenului auto si primirea de punctaje corespunzatoare – atunci cand jucatorul efectueaza manevra fara greseli.

In toate cele 3 cazuri, jucatorul este anuntat la incheirea manevrei daca a efectuat miscarile in mod corect sau ce a gresit in cazul in care manevra nu a fost efectuata corespunzator.

Atat controlul manual al masinii cat si cel automat sunt implementate in acelasi script. Tipul controlului este definit de variabila globala “self\_driving” care la randul ei isi schimba valoarea in functie de anumite checkpoint-uri de-a lungul gameplay-ului. Pe ramura ‘if’ a functiei ‘FixedUpdate’ (care se apeleaza la un numar regulat de frame-uri) se apeleaza functiile de conducere autonoma iar pe ramura ‘else’ este implementat controlul manual al masinii. Functia ‘WaypointDistane’ are rolul de a verifica checkpointurile prestabilite pentru jucator indiferent de modul de control al masinii.

```

void FixedUpdate () {
    if (GlobalVariables.self_driving)
    {
        ApplySteer();
        Drive();
        Braking();
        SmoothSteer();
    }
    else
    {
        Forward = Input.GetAxis("Vertical");
        Turn = Input.GetAxis("Horizontal");
        Brake = Input.GetAxis("Jump");
        [...]
    }
    WaypointDistance();
}

```

În joc, controlul manual al mașinii este preferențial (unii jucători preferă să folosească mâna stângă, alții mâna dreaptă) și foarte intuitiv:

- W, săgeată în sus – mergi înainte;
- S, săgeată în jos – frână ușoară, mergi înapoi;
- Spațiu – frână puternică;
- A, săgeată stângă – viraj stâng;
- D, săgeată dreaptă – viraj drept;
- Q – semnal stângă pornit/oprit;
- E – semnal dreptă pornit/oprit;
- P – pauză.

## Concluzii generale și direcții viitoare

Dorința oamenilor de a obține permisul de conducere a devenit în ultimii ani tot mai comună, aceștia apelând la diferite metode pentru a se familiariza mai ușor cu acest domeniu nou pentru ei la momentul respectiv. Deși fiecare om își dezvoltă un stil propriu de învățare, una dintre

cele mai bune metode didactice ramane invatarea prin joc. Aplicatia dezvoltata are rolul de a introduce utilizatorul intr-un mediu cat mai apropiat de realitatea din trafic incercandu-se pregatirea atat din punct de vedere teoretic cat si practic.

In opinia mea, aplicatiile de acest gen au un impact destul de mare asupra utilizatorilor deoarece sunt putine aplicatii pe piata care sa lucreze in acelasi scop ca “My driving school”, principala lor problema fiind lipsa restrictiilor si o libertate prea mare acordata jucatorului. In plus, intrebarile teoretice sunt foarte asemanatoare cu cele din chestionarele auto standard, diferenta sesizabila fiind modul de prezentare mai interactiv al situatiilor. Acest joc utilizeaza destul de intens atentia jucatorului, exact cum ar trebui sa fie si in trafic mai ales la inceput.

Imbunatatiri si directii viitoare:

- Trecerea la un alt tip de grafica pentru a permite scalarea aplicatiei in functie de rezolutia ecranului fara a pierde calitatea imaginii;
- Introducerea unor noi situatii in trafic precum depasiri, masini speciale etc. Facand o paralela cu traficul din viata reala, pe aceasta imbunatatire se poate lucra foarte mult;
- Crearea unui mod in care utilizatorul invata sa parcheze;
- Adaugarea de noi modele 3D mai placute din punct de vedere vizual.

## Bibliografie

- [1] Informatii Unity  
<https://unity3d.com/>
- [2] Statistici privind promovabilitatea examenului auto in Romania  
<https://scoaladesoferi.odat.ro/rata-promovabilitate-scoli-auto/>
- [3] Informatii simulatoare profesionale  
[https://en.wikipedia.org/wiki/Driving\\_simulator](https://en.wikipedia.org/wiki/Driving_simulator)

- [4] City Car Driving  
<https://citycardriving.com/>
- [5] Driving school 2017  
<https://play.google.com/store/apps/details?id=com.ovilex.drivingschool2017&hl=en>
- [6] Unity asset store  
<https://assetstore.unity.com/>
- [7] Top genuri ale jocurilor video  
<https://sites.google.com/site/reviewgames11/top-genres-video-games>