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

**FACULTATEA DE INFORMATICĂ**

****

**LUCRARE DE LICENȚĂ**

**My driving school**

propusa de

Bălan 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

Bălan Sabin-Marian

**Sesiunea:** februarie, 2019

Coordonator științific

Conf. dr. Vitcu Anca

**Avizat,**

**Îndrumător Lucrare de Licenţă**

Titlul, Numele şi prenumele Conf. Dr. Vitcu Anca

Data 13-02-2019 Semnătura \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**DECLARAŢIE privind originalitatea conţinutului lucrării de**

**licenţă**

Subsemnatul(a) Bălan Sabin-Marian

domiciliul în sat Laza, Comuna Laza, judetul Vaslui

născut(ă) la data de 10-11-1996 , identificat prin CNP 1961110374529

absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iaşi, Facultatea de Informatica

specializarea Informatica , promoția 2018 ,

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 si 5 referitoare la plagiat, că lucrarea de licență cu titlul: My Driving School elaborată sub îndrumarea d-na Prof. Dr. Vitcu Anca , pe care urmează să o susțină în fata comisiei este originală, îmi aparține şi 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 in vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație şi în acest sens, declar pe proprie răspundere că lucrarea de fată nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, 13-02-2019 Semnătură student

DECLARAŢIE DE CONSIMŢĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „My driving school”, 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, 13.02.2019 Absolvent Bălan Sabin-Marian,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Cuprins**

**Capitolul I.** Introducere ………………………………………………….... 6

**I.1.** Motivație …………………………..………………….………… 6

**I.2.** Obiectiv …………………………………..…………….……….. 7

**I.3.** Scurt istoric ................................................................................... 7

**I.4.** Aplicații similare …...……………………….…………………... 9

**I.5.** Contribuții …………...…………………………..………….….. 11

**Capitolul II.** Tehnologii utilizate …………………………………….…… 12

**Capitolul III.** Descrierea aplicației ……………………………….…….… 13

**III.1.** Interfața ……………………………………………...…….…. 16

**III.2.** Întrebările ………………………………………………….… 18

**III.3.** Arhitectura situațiilor practice ……………………………...... 20

**III.4.** Sistemul de path-uri ………………………………………….. 21

**III.5.** Sistemul de lumini ……………………………………………. 22

**III.6.** Mașinile din trafic ….………………………………………… 24

**III.7.** Controlul jucătorului …………………………………………. 29

**Capitolul IV.** Concluzii finale si direcții viitoare ……………….…….….. 31

**Capitolul V.** Bibliografie ……………………………..……………….….. 32

**Capitolul I. Introducere**

**I.1. Motivație**

În România, numărul persoanelor care încep școala de șoferi în vederea obținerii permisului de conducere este într-o continuă creștere. Totuși, există o problema arzătoare cu care se confruntă mai toate județele din țară: scăderea ratei mediei de promovabilitate până la procente care sunt cu mult sub așteptări.

Privind harta cu rata de promovabilitiate pe județe realizată în baza informațiilor puse la dispoziție de 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 întreagă țară.

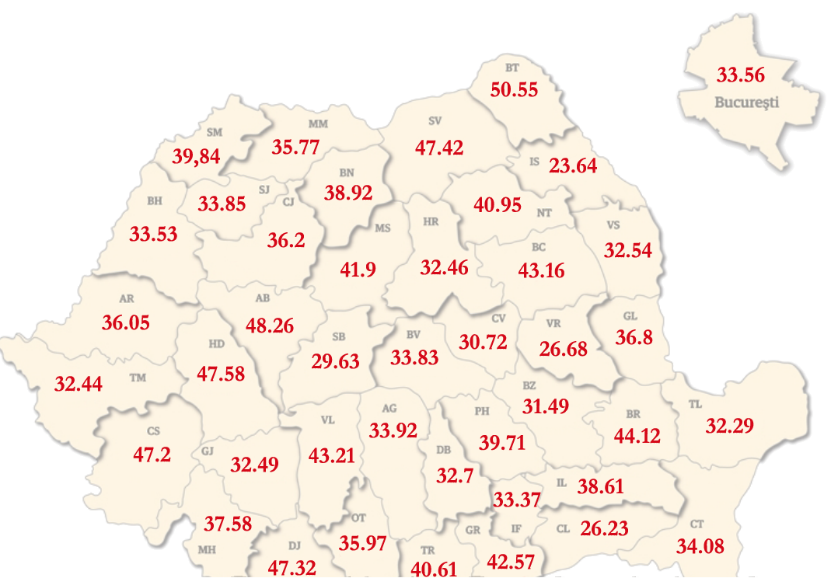


Fig. 1 Rata de promovabilitate a școlilor de șoferi pe județe

Sursa: [scoaladesoferi.odat.ro/rata-promovabilitate-scoli-auto](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. Așa cum arată și statistica prezentată anterior în Fig. 1, metodele actuale au o rată de succes scăzută.

O soluție pentru această problema ar putea consta în crearea unui mediu de învățare care să îmbine atât întrebările de ordin teoretic, cât și situațiile de ordin practic prin care utilizatorul să învețe regulile de circulație într-un mod distractiv și ușor. Jucătorul va lua contact cu numeroase situații din trafic iar deciziile sale vor influența direct soarta jocului, traseul creat punându-l într-o ipostază aseamanatoare cu cea din timpul unui examen auto practic. Astfel, jucătorul este constrâns să conducă mașina cât mai corect cu putință și să răspundă cu atenție la întrebările de legislație, miza fiind promovarea sau picarea examenului auto simulat. De asemenea, adăugarea unui scor și a unor pedepse mai severe (ex.: încheierea nivelului) stârnesc spiritul competitiv al utilizatorului.

**I.2. Obiectiv**

Scopul general al aplicației “My Driving School” este reprezentat de aducerea pe piață a unei noi metode de invățare a legislației auto care să conducă la imbunătățirea procentajului de promovabilitate al examenului auto.

“My Driving School” urmărește să îmbine într-un mod cât mai plăcut întrebările de ordin teoretic cu situațiile practice din trafic realizând astfel un mediu de învățare interactiv si antrenant.

**I.3. Scurt istoric**

Oamenii au avut dintotdeauna o puternică legătură cu orice concept legat de jocuri, acestea fiind percepute ca modalități de divertisment dar și ca fiind educative, aducând o contribuție semnificativă la dezvoltarea abilităților fizice și de gândire ale individului.

Jocurile de tip simulator și-au făcut apariția la mijlocul anilor 1980 atunci când dezvoltatori precum Code Masters și Oliver Twins au lansat pe piață primele jocuri care să conțină “Simulator” in titlu: BMX Simulator (1986), Grand Prix Simulator (1986) și Pro Boxing Simulator (1988). Aceștia s-au inspirit din titluri best-seller, care la rândul lor, se bazau pe sporturi reale cum ar fi fotbalul sau cursele de biciclete BMX.

Scopul acestui tip de jocuri, după cum o spune și numele, este de a simula o experiență pe cât de real posibil, iar jocurile ulterior create, SimLife și SimEarth, au fost capabile să învețe jucătorii bazele ecosistemelor globale. În timp ce mulți consideră că jocurile simulator au început cu Will Wright și al său SimCity în 1989, adevăratul strămoș al acestui gen a fost Fortune Builder lansat în 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.

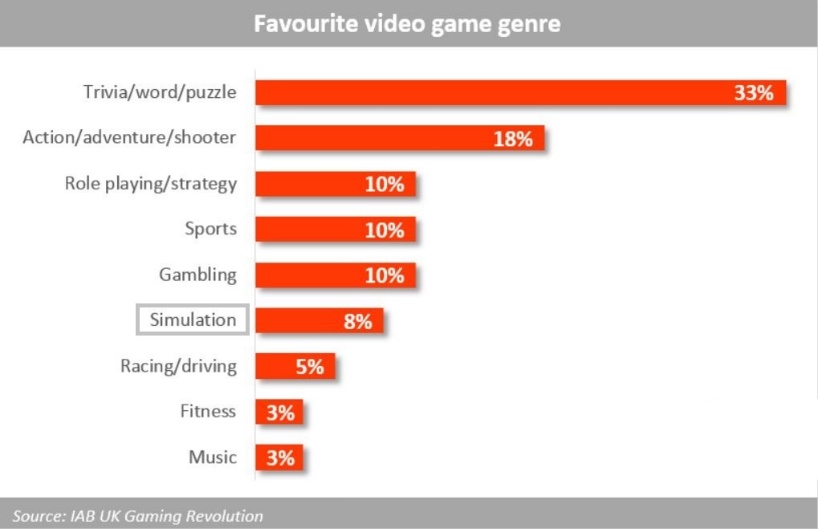


Fig. 2 Topul categoriilor de jocuri video

De ce sunt jocurile simulator atât de apreciate în prezent? Deoarece acestea transpun jucătorul într-un mediu ales de el, foarte aseamanator cu realitatea, în care poate experimenta o varietate mare de activități și trăiri ce pot fi împărțite 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 adunând vânzări de peste 200 de milioane de copii la nivel global;
* City-Building: SimCity (1989 – prezent; ultimul titlu lansat în 2014 “SimCiry 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 anul 1993;
* Farm simulator: “Farming simulator” unde încă din 2008 până in prezent jucătorii experimentează ce înseamnă cultivarea pământului;
* Vehicle simulation:
  + Racing simulation: “Forza horizon 4” castigător al premiului de “Cel mai bun joc de racing” din 2018;
  + Ship simulator: “World of Warships” lansat în septembrie 2015;
  + Flight simulator: “Ace combat” un alt titlu longeviv (1995-prezent);
  + Tank simulation: “World of Tanks” cu peste 75 milioane de jucători înregistrați;
  + Truck simulator: “Euro truck simulator” cu 300.000 de copii vândute în Europa.

**I.4. Aplicații similare**

În prezent, există puține simulatoare auto care fac un scop principal din prezentarea elementelor teoretice ale legislației și respectarea unor restricții pe parcursul traseului. Majoritatea oferă o libertate mult prea mare jucătorului care nu învață foarte multe lucruri doar plimbându-se cu mașină pe o harta de tipul open-world. Cele mai notabile titluri sunt City Car Driving (PC) și Driving School 2017 (Android & Ios). Ambele adună un număr consistent de jucători și au recenzii suficient de bune însă nu sunt orientate spre a învața utilizatorul noi reguli de circulație.



Fig. 3 Driving School 2017 Fig. 4 City Car Driving

Sursa fig. 3: play.google.com/store/apps/details?id=com.ovilex.drivingschool2017&hl=en

Sursa fig. 4: city-car-driving.de.softonic.com

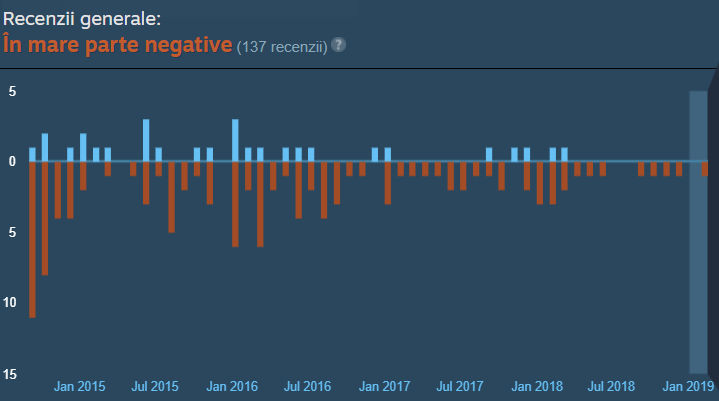
Jocurile precum “Driving School Simulator” care oferă o experiență asemănătoare cu cea a școlii de șoferi au în mare parte recenzii negative.

Fig. 5 Driving School Simulator Fig. 6 Recenzii Driving School Simulator

Sursa: store.steampowered.com/app/273730/Driving\_School\_Simulator

O altă soluție este reprezentată de simulatoarele auto pentru antrenament și cercetare dar acestea nu sunt accesibile oricui. Aceste simulatoare performante sunt prezente în puține dintre școlile de șoferi din România, ele fiind folosite în principal pentru antrenamentul șoferilor de pe ambulanță, camioane, autobuze etc.

Principalele utilizări ale simultoarelor profesionale:

* Antrenarea și testarea șoferilor începători;
* Antrenarea și testarea șoferilor profesioniști;
* Antrenarea șoferilor pentru situații critice;
* Analiza comportamentului șoferului;
* Analiza reflexelor șoferului;
* Evaluarea performanțelor șoferului în diferite situații (explodarea unui cauciuc, gheață pe suprafața carosabilă etc.);
* Evaluarea performanțelor șoferului în diferite condiții (consum de alcool, consum de droguri, lipsă de somn etc.).

Fig. 7 Simulatoare auto profesional 1 Fig. 8 Simulatoare auto profesional 2

[Sursa: irishtimes.com](Sursa:%20irishtimes.com) [Sursa: en.wikipedia.org/wiki/Driving\_simulator](https://en.wikipedia.org/wiki/Driving_simulator)

**I.5. Contribuții**

Spre deosebire de restul aplicațiilor similare de pe piață, “My driving school” introduce o viziune nouă, focusându-se cel mai mult pe cunostințele jucătorului. Jocul este mult mai restrictiv, oferind prin text si imagini indicații clare referitoare la traseu, experiența fiind asemănătoare cu cea de la un examen auto practic.

Principala inovație constă în faptul că jocul alternează situațiile de ordin practic cu întrebările de ordin teoretic. Pe măsură ce jucătorul avansează pe traseu acesta este pus în diferite imprejurări unde i se testează cunoștințele teoretice, abilitățile practice si capacitatea de a fi atent la ceea ce se întâmplă în jurul său. Jocul îmbină constant teoria cu practica și oferă feedback imediat referitor la corectitudinea cu care s-a îndeplinit o manevră sau răspunsul corect la o intrebare.

O altă noutate o reprezintă modul în care utilizatorul interacționează cu mașina sa. Controlul mașinii alternează între manual si automat, scopul principal al acestei funcționalități fiind acela de a prelua atenția jucătorului de pe condusul mașinii și a o focusa strict pe situațiile prezentate. Astfel, jucătorul are mai mari șanse să nu omită detalii importante din trafic și să răspundă corect la întrebări.

**Capitolul II. Tehnologii utilizate**

****

Unityeste un program dezvoltat de către cei de la Unity Technologies și lansat pentru prima oară în iunie 2015 la “Apple Inc.’s Worldwide Developers Conference” ca și un editor de jocuri exclusiv pentru Mac OS X. În prezent, engine-ul a fost extins pentru a suporta crearea de jocuri pentru un număr de până la 27 de platforme, printre cele mai importante numărându-se Windows, macOS, Android, iOS, Playstation4, Xbox One, Android TV etc.

IDE-ul (Integrated Development Environment) celor de la Unity Technologies oferă utilizatorilor săi posibilitatea de a crea atât jocuri 2D cât și 3D, utilizând limbajul de programare C# drept principal API pentru scripting atât pentru plugin-uri cât și pentru jocul creat. De fapt, Unity este reuniunea urmatoarelor trei lucruri:

* un game engine – permite ca jocurile create să fie rulate în diferite environment-uri;
* un editor în care toate componentele vizibile ale unui joc pot fi manipulate și care îți oferă la dispoziție funcția de ‘play’ în orice moment;
* un editor de cod – cel standard este MonoDevelop însă dezvoltatorii pot alege altele (eu am utilizat Visual Studio 2017).

Aplicațiile făcute în Unity au ajuns pe nu mai puțin de 2.7 miliarde de dispozitive la nivel global și au fost instalate de mai mult de 24 miliarde de ori în ultimele 12 luni. Astfel, Unity este unul dintre cei mai importanți jucători de pe piață revoluționând în mod constant industria jocurilor.

Sursa: unity3d.com

**Capitolul III. Descrierea aplicației**

Această lucrare are ca obiectiv implementarea unui joc 3D de tip simulator pentru Windows, accesibil unui număr cât mai mare de utilizatori. Jocul este creat în totalitate în engine-ul Unity prin intermediul căruia s-a construit întreaga hartă și s-a dezvoltat comportamentul obiectelor. De asemenea, jocul a fost gândit pentru a fi cât mai simplu și intuitiv.

Accentul lucrării a picat mai mult pe partea de programare a situațiilor din trafic și a întrebărilor. Prin urmare,  în vederea construirii mediului de joc s-au importat și adaptat următoarele pachete de modele gratis:

1. Low Poly Street Pack – modelat de către cei de la Dynamic Art 3D

Acest pachet a fost utilizat în vederea modelarii a tot ceea ce ține de infrastructura stradală (șosele, poduri, trotuare, semafoare, intersecții, etc.) fără a fi aduse modificări elementelor.

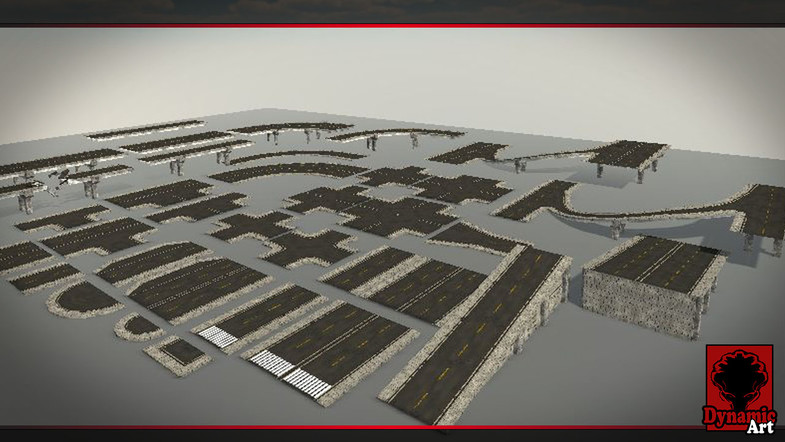


Fig. 9 Low poly street pack

[Sursa: assetstore.unity.com/packages/3d/environments/urban/low-poly-street-pack-67475](https://assetstore.unity.com/packages/3d/environments/urban/low-poly-street-pack-67475)

1. Simple City Pack Plain – modelat de către cei de la Polygon Land

Acest pachet a fost utilizat pentru a construi decorul urban al jocului. Modificări s-au adus doar în privința materialelor de pe obiecte și a culorilor utilizate pentru prefabricate.



Fig. 10 Simple City Pack Plain

[Sursa: assetstore.unity.com/packages/3d/environments/urban/simple-city-pack-plain-100348](https://assetstore.unity.com/packages/3d/environments/urban/simple-city-pack-plain-100348)

1. Simple Cars Pack – modelat de către cei de la Myxerman

Acest pachet conține modelele mașinilor utilizate în joc care au fost adaptate ulterior pentru a putea primi un anumit comportament dorit în urma implementării scripturilor necesare. Pe mașini s-au adăugat lumini de semnalizare și de frână și au fost modificate anumite texturi.



Fig. 11 Simple Cars Pack

[Sursa: assetstore.unity.com/packages/3d/vehicles/land/simple-cars-pack-97669](https://assetstore.unity.com/packages/3d/vehicles/land/simple-cars-pack-97669)

1. Traffic signs – modelat de către Lapi Games

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



Fig. 12 Traffic Signs

[Sursa: lapigames.itch.io/unity-city-traffic-signs](https://lapigames.itch.io/unity-city-traffic-signs)

1. BodyGuards – modelat de către cei de la Batewar

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



Fig. 13 BodyGuards

[Sursa: assetstore.unity.com/packages/3d/characters/humanoids/bodyguards-31711](https://assetstore.unity.com/packages/3d/characters/humanoids/bodyguards-31711)

De menționat este faptul că pachetele importate conțin doar modelele 3D, fără niciun fel de animație sau algoritm.

**III.1. Interfața**

Prin interfață, înțelegem toate informațiile care apar pe ecran înaintea și în timpul execuției jocului. O interfață cât mai bună și mai prietenoasă îmbunătățește cu mult experiența de joc. Din acest motiv, în realizarea interfeței am urmărit următoarele lucruri:

1. Interfața îmi spune doar ce trebuie să știu când trebuie să știu?
2. Informațiile de care am nevoie sunt ușor de găsit, fără a petrece prea mult timp în a le căuta?
3. Se poate utiliza interfața fără a avea nevoie de niciun fel de instrucțiuni regăsite în altă parte?
4. Cât timp trebuie să se păstreze informațiile pe ecran?
5. Există elemente inutile sau modalități mai simple de a afișa anumite informații?

**Meniul**

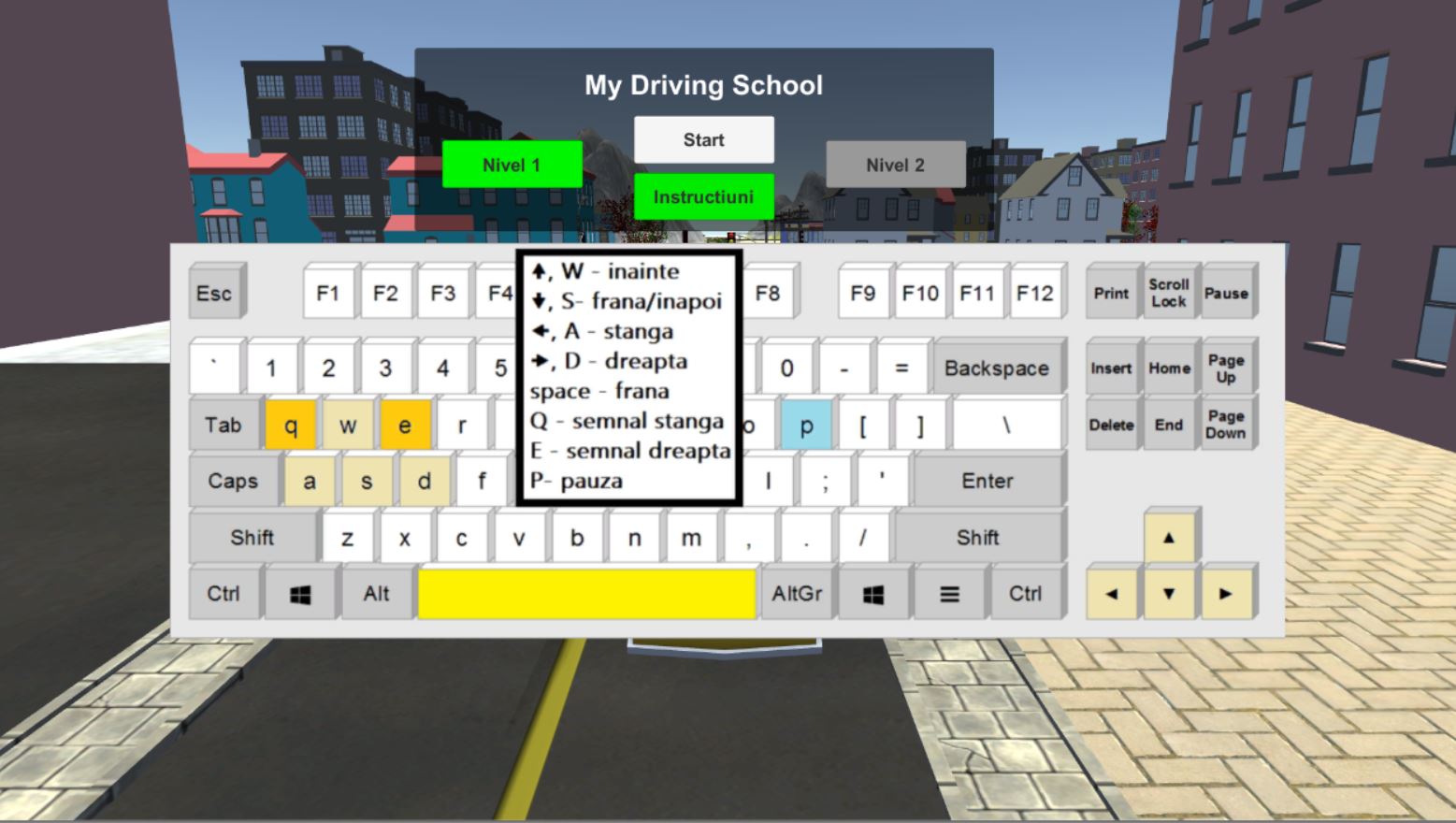
Meniul jocului reprezintă o listă de comenzi care pot fi selectate cu ajutorul mouse-ului. Am optat pentru un meniu cât mai simplu, fără informații inutile și fără alte submeniuuri. Acesta conține butonul de start al nivelului selectat, butoane de selecție pentru nivelele jocului și un buton de instrucțiuni. La apăsarea butonului de instrucțiuni, pe ecran apare o imagine cu toate informațiile necesare referitoare la control. Aceasta poate fi închisă prin reapăsarea butonului de instrucțiuni sau prin pornirea nivelului din butonul de start.

Fig. 14 Meniul jocului

**Interfața din timpul jocului**

Pe întreg parcursul jocului este necesară afișarea unei serii de informații indispensabile jucătorului. Acestea apar pe ecran doar când sunt necesare și sunt înlăturate imediat ce nu mai conțin indicații relevante. Stilul interfeței urmărește aceleași linii ca meniul, cu informații cât mai simple și cât mai relevante, afișate pe un fundal negru transparent. Scopul lor este de a ține jucătorul la curent cu rezultatul deciziilor sale și de a-i oferi indicații despre cum trebuie să procedeze in continuare.

Fig. 15 Interfața în timpul jocului

1. Scorul – determinat de numărul întrebărilor la care jucătorul răspunde corect și numărul manevrelor executate corespunzător. Apare pe ecran pe tot parcursul jocului;
2. Timpul – un temporizator descrescător exprimat în secunde care afișează timpul rămas jucătorului pentru executarea unei manevre. Acesta apare pe ecran doar în situațiile în care este necesar și dispare după ce manevra a fost completă. Expirarea timpului conduce la picarea examenului. Toată lumea din trafic este deranjează de un șofer mult prea lent care petrece prea mult timp pentru a lua decizii;
3. Zona principală de mesaje – apare pe ecran doar când este necesară și afișează informații precum “Felicitari! Ai executat manevra corect!”, “Din păcate nu ai semnalizat corespunzător!” și altele;
4. Zona secundară de mesaje – apare pe ecran doar când sunt necesare indicații suplimentare. Ex.: la schimbarea benzii apare mesajul “Nu uita să semnalizezi pe tot parcursul manevrei!”;
5. Anunțul cu privire la controlul mașinii – de fiecare dată când mașina trece din modul manual pe modul automat și invers, jucătorul este anunțat prin acest mesaj;
6. Butonul de meniu principal – apare doar în cazul în care jucătorul încalcă grav o reguală de circulație, fapt care conduce la încheierea nivelului. Apăsarea butonului deschide meniul jocului;
7. Zona de indicații – apare pe ecran atunci când jucătorul ajunge într-o intersecție și dispare când nu este necesară. Aceste indicații țin locul celor oferite de un polițist într-un examen auto din viața reală.

Mesajele temporare dispar după 3 secunde de la afișare.

La finalul jocului, zona de informații se mărește afișându-se rezultatele jucătorului la cele două capitole: manevre în trafic și întrebari.

**III.2. Întrebările**

Pe măsură ce jucătorul avansează pe traseu, acestuia i se vor adresa numeroase întrebări teoretice. La fel ca în chestionarele de la examenul teoretic concepute și oferite de DRPCIV (Direcția Regim Permise de Conducere și Înmatriculare a Vehiculelor) întrebările au 3 variante de răspuns dintre care una, două sau toate trei pot fi corecte. Jucătorul selectează răspunsurile sale prin bifarea unor căsuțe, după care apasă butonul “Răspunde”. Verificarea se realizează pe loc, un mesaj de informare apare pe ecran cu scopul de a anunța dacă variantele alese sunt sau nu cele corecte în timp ce textul celor 3 răspunsuri își va schimba culoarea după cum urmează:

1. text verde – răspuns corect și selectat de către jucător;
2. text roșu – răspuns greșit selectat de către jucător;
3. text galben – răspuns corect dar neselectat de către jucător.

În urma verificării răspunsurilor, scorul jucătorului se actualizează, butonul “Răspunde” se transformă în “Continuare” iar acționarea acestuia duce la dispariția întrebării de pe ecran și continuarea jocului.

Întrebările sunt salvate ca fișiere și au următorul format: întrebare, răspuns 1, răspuns 2, răspuns 3, valoare răspuns 1, valoare răspuns 2, valoare răspuns 3 (de tip bool: 0 – răspuns incorect, 1 – răspuns corect).

Fig. 16 Exemplu întrebare

if (GlobalVariables.ans1\_value)

{

if (GlobalVariables.answer1)

ans1\_text.GetComponent<Text>().color = Color.green;

else

{

ans1\_text.GetComponent<Text>().color = Color.yellow;

trueFlag = false;

}

}

else

{

if (GlobalVariables.answer1)

{

ans1\_text.GetComponent<Text>().color = Color.red;

trueFlag = false;

}

}

\*Verificarea raspunsurilor si setarea culorilor

**III.3. Arhitectura situațiilor practice**

Fiecare situație practică în care jucătorul va fi transpus în acest joc de simulare este un game object de sine stătător alcătuit la rândul sau din numeroase alte obiecte (mașini, semafoare) cu roluri bine definite. Situațiile sunt controlate de o serie de triggere răspândite pe toată harta care au rolul de a activa sau dezactiva anumite obiecte în funcție de anumite criterii. Există situații în care se controlează luminile unui semafor, situații în care pe șosea trebuie să apară indicații de orientare, situații în care mașinile din trafic trebuie să realizeze anumite acțiuni și multe altele. Aceste lucruri nu ar putea fi posibile fără obiectul denumit “case controller” care manageriază toate aceste situații în funcție de unde se află jucătorul la un moment dat.

Fig. 17 Exemplu de situație

În figura 17 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 urmatoarele:

1. Ceilalți participanți la trafic;
2. Traseul celorlalți participanți la trafic;
3. Serie de triggere utilizate pentru a verifica dacă jucătorul a semnalizat în mod corespunzător manevra;
4. Serie de triggere utilizate pentru a verifica dacă jucătorul a respectat sau nu indicațiile. De menționat că nerespectarea indicațiilor conduce la picarea examenului și încheierea nivelului.

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

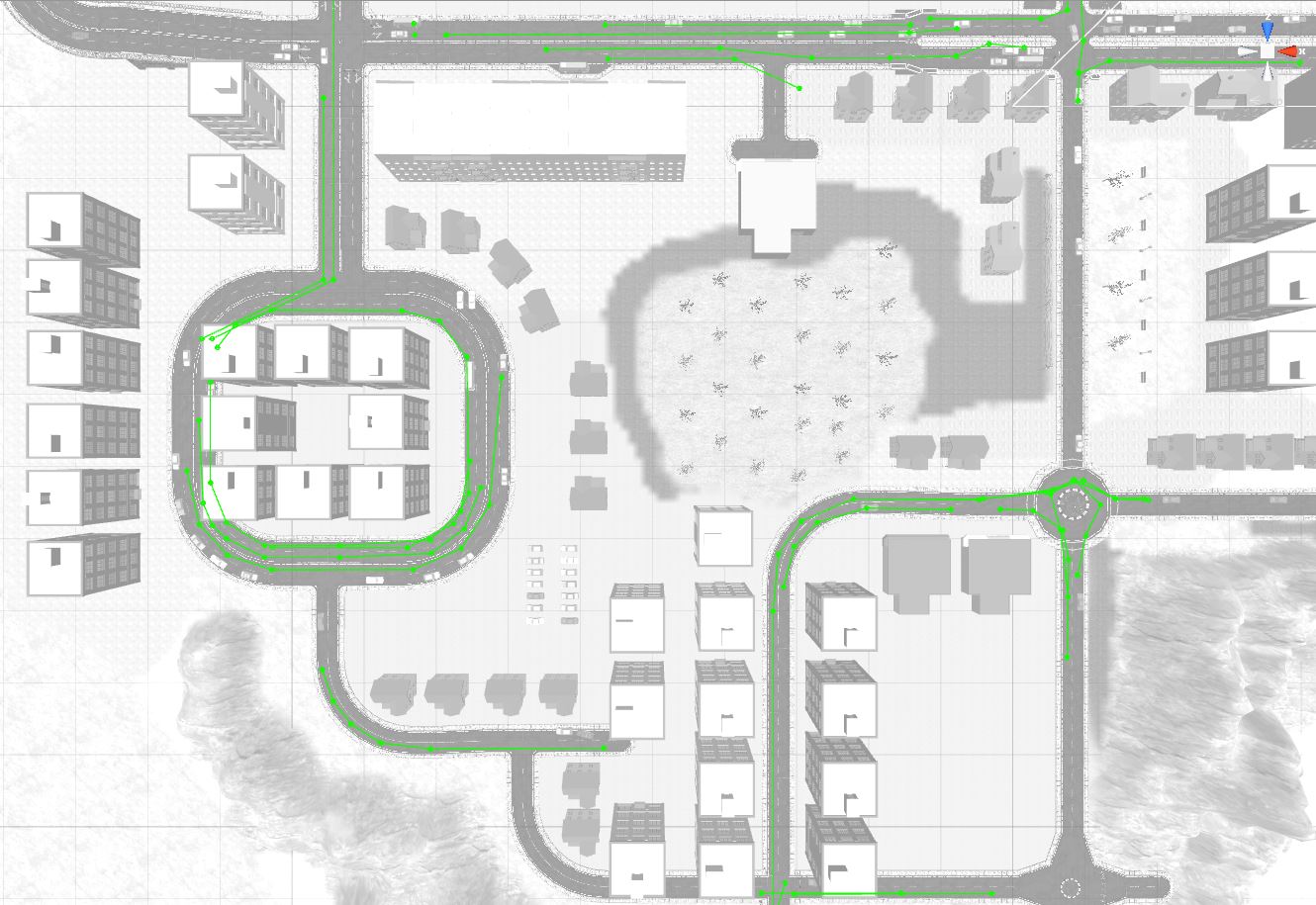
****

Fig. 18 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 rulării prin execuția scriptului ‘Path.cs’. Acesta are rolul de a lua nodurile și a le stoca într-un vector în ordinea in care au fost citite și de a crea o reprezentare vizuală a path-ului respectiv. 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 încurcătură și aglomerarea inutilă a străzilor, am conceput ca path-ul să devină vizibil doar atunci când acesta este selectat. Pentru reprezentarea nodurilor am utilizat un wiresphere iar pentru liniile ce reprezintă calea propriuzisă am creat segmente între noduri. Culoarea aleasă este verde pentru ca path-ul să iasă ușor în evidență pe hartă atunci când este selectat în vederea realizării unei modificari.

Prin sistem de path-uri înțelegem un ansamblu de elemente care formează un întreg organizat, coordonat și influențat de aceleași entități superioare lor în vederea indelpinirii unei funcții comune (ghidarea mașinilor din trafic). Obiectele care controlează dacă un path este utilizat sunt cele de tip ‘Case’ prezentate anterior care definesc, în funcție de situația în care se află jucătorul, ce obiecte trebuiesc folosite.

**III.5. Sistemul de lumini**

Sistemul de lumini al unui vehicul constă în dispositive de luminare și semnalizare montate sau integrate în fața, spatele, lateralele mașinii și în unele cazuri speciale chiar deasupra. Deși există o diversitate mare a variantelor constructive, toate vehiculele trebuie să conțină următoarele mijloace de iluminare și semnalizare:

* Partea din față:
  + Faruri cu lumină de întalnire (faza scurtă), faruri cu lumină de drum (faza lungă), lămpi de poziție-staționare, lămpi de semnalizare a direcției, lămpi de avarii (identice cu cele de semnalizare a direcției) – obligatorii;
  + Faruri de ceață (proiectoare) – nu sunt obligatorii.
* Partea din spate:
  + Lămpi de poziție/staționare, lămpi de semnalizare a direcției, lămpi de avarii, lămpi de frânare, lămpi de mers înapoi, lămpi de iluminare a numarului de inmatriculare – obligatorii;
  + Lămpi de ceață – nu sunt obligatorii.

Sursa: [www.scoalarutiera.ro/curs-legislatie/codul-rutier/sistemul-de-iluminare-si-semnalizare-luminoasa](http://www.scoalarutiera.ro/curs-legislatie/codul-rutier/sistemul-de-iluminare-si-semnalizare-luminoasa)

În joc, accentual pică mai mult pe luminile de frână, luminile de semnalizare a direcției de mers și luminile speciale precum cele ale mașinii de poliție. Pentru a putea fi creat sistemul de lumini, a fost necesară modificarea modelelor importate prin atașarea de noi obiecte (cuburi și sfere) care să îndeplinească rolul blocurilor de lumini. De asemenea, s-au creat și noi materiale utilizate în următoarele două scopuri:

* Imitarea luminilor stinse – materiale normale;
* Imitarea luminilor aprinse – materiale emisive.

|  |  |
| --- | --- |
|  |  |
|  |  |

Fig. 19 Sistemul de lumini

În cazul jucătorului, luminile sunt acționate după cum urmează: luminile de frână se aprind cât timp mașina frânează (apăsarea tastelor ‘s’, ‘spațiu, ‘săgeată jos’); luminile de semnalizare se aprind și se sting printr-o singură apăsare a tastei (‘e’ – semnalizare dreapta, ‘q’ – semnalizare stânga). În cazul celorlalți participanți la trafic, luminile sunt acționate fie automat (ex.: luminile de semnalizare in sensul giratoriu), fie prestabilit (ex.: semnalele luminoase ale mașinii de poliție).

**III.6. Mașinile din trafic**

Mașinile din trafic sunt concepute pentru a simula o experiență cât mai apropiată cu cea din realitate. Acestea sunt capabile să urmeze un traseu prestabilit, să semnalizeze intenția schimbării direcției de mers, să frâneze, să evite obstacole și să evite eventualele coliziuni cu jucătorul sau cu celelalte mașini. Pentru a putea fi integrate în mediul virtual, mașinile sunt tratate ca niște corpuri din viața reală având proprietăți fizice precum: masă (900-1200kg), centru de greutate, rezistență aerodinamică, forță centripetă și gravitație. Astfel, deși din punct de vedere al graficii mașinile sunt simple, din punct de vedere fizic ele respectă cu fidelitate comportamentul unei mașini adevărate cu proprietăți aseamanatoare.

“Creierul” tuturor mașinilor se regăsește în scriptul ‘CarEngine’ atașat fiecărui vehicul din joc. Deoarece Unity oferă posibilitatea de editare independentă a variabilelor publice din orice script atașat unui obiect, și scriptul ‘CarEngine’ conține o serie de parametri publici ce facilitează personalizarea fiecărei mașini în funcție de necesități. Dintre acestea cele mai notabile sunt: cuplul, viteză maximă, unghiul maxim de bracare al roților și puterea maximă de frânare.

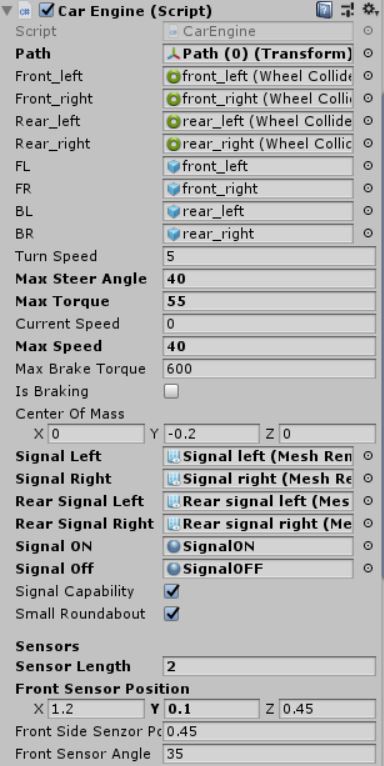


Fig. 20 Managementul variabilelor publice din scriptul “Car Engine” atașat unei mașini

‘CarEngine’ controlează toate funcționalitățile mașinii după cum urmează:

* Deplasarea pe un traseu prestabilit:

Mișcarea vehiculelor nu este posibilă fără un path dat ca parametru scriptului. În funcția ‘Start’ (funcția care se ocupă cu inițializările necesare) path-ul dat ca parametru este preluat și salvat într-un vector local care va servi la calcularea următorului punct către care trebuie să se îndrepte mașina.

Mișcarea vehiculelor pe stradă are loc strict prin aplicarea de forțe asupra roților din spate și prin bracarea roților din față. Pentru realizarea acestui lucru, am aplicat componenta Wheel Collider (contur verde în figura) pe fiecare roată a modelului (contur portocaliu în figura).

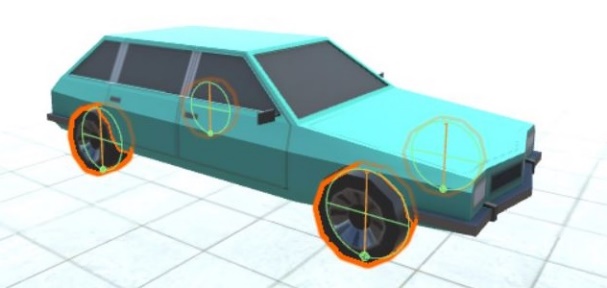


Fig. 21 Componente roți

Deoarece s-a observat un demaraj mai bun, mașinile au tracțiune spate. Singura condiție necesară pentru a aplica forță asupra roților din spate este ca viteza curentă a mașinii (calculată prin formulă currentSpeed = 2 \* 22 / 7 \* rear\_left.radius \* rear\_left.rpm \* 60 / 1000) să nu depășească viteză maximă setată.

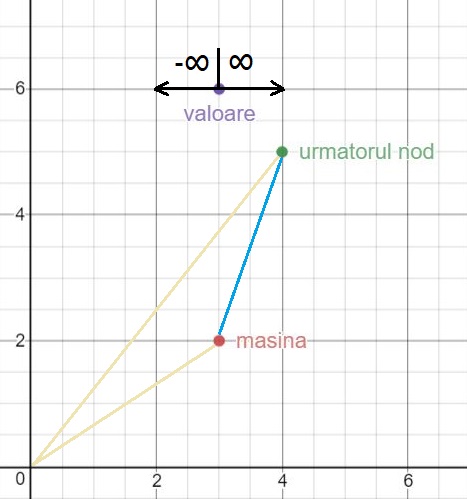


Fig. 22 Determinare unghi bracare

Pentru determinarea unghiului de bracare al roților au fost necesare următoarele calcule:

Deoarece în Unity coordonatele oricărui obiect sunt exprimate în funcție de origine (0,0), primul lucru pe care trebuie să îl aflăm este poziția următorului nod al path-ului în funcție de poziția actuală a mașinii. Din fericire, Unity oferă o funcție (InverseTransformPoint) care calculează această valoare și returnează un vector (vectorul albastru din figura 22). După calcularea vectorului relativ putem ști deja dacă următorul punct se află fie în partea dreapta (x-ul este pozitiv) a mașinii fie în partea stânga (x-ul este negativ). În exemplul nostru (figura 22), vectorul rezultat este (1,3) deci următorul punct din path se află în dreapta mașinii.

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

În continuare, avem nevoie de o valoare din intervalul (-1,1) pe care să o înmulțim cu variabilă publică maxSteerAngle (unghiul maxim de bracare). Această valoare o vom obține luând vectorul relativ definit anterior și împărțindu-l la propria sa lungime.

relativeVector /= relativeVector.magnitude;

Următorul pas este să înmulțim valoarea rezultată cu maxSteerAngle pentru a obține unghiul de bracare necesar și aplicarea acestuia asupra wheel colliderelor roților din față.

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

Acest lucru se întâmplă la fiecare frame în parte iar în momentul în care mașina ajunge la o distanță mai mică de 0.5m față de nodul curent, funcția denumită WaypointDistance realizează tranziția la nodul următor din path.

Efectul vizual de invârtire al roților și cel de virare au fost obținute prin transformarea poziției și rotației fiecărui model de roată în parte în funcție de wheel collider-ul său.

* Semnalizarea intenției schimbării direcției de mers:

Semnalizarea automată a intenției schimbării direcției de mers se realizează în funcție de unghiul de bracare al roților din față. Deși în cele mai multe situații această modalitate de a stabili când și în ce direcție trebuie să semnalizeze mașina funcționează corect, am constatat că există și o excepție. Este vorba despre semnalizarea în sensul giratoriu atunci când mașina execută o serie de viraje dar trebuie să semnalizeze doar stânga în timpul parcurgerii sensului și dreapta la ieșire. Dacă s-ar păstra regula generală, mașina ar semnaliza haotic deoarece roțile ar executa o serie de viraje in direcții diferite. Ca rezolvare pentru această situație am schimbat unghiul definitoriu pentru tipul de semnalizare in funcție de dimensiunea sensului giratoriu și am adăugat la fiecare ieșire din sens un semnal care să oprească semnalizarea.

if (smallRoundabout)

{

if (turnAngle > 19) rightSignalFlag = true;

if (rightSignalFlag)

{

signalRight.material = signalON;

signalLeft.material = signalOff;

rearSignalRight.material = signalON;

rearSignalLeft.material = signalOff;

float floor = 0f;

float ceiling = 1f;

float emission = floor + Mathf.PingPong(Time.time \* 2f, ceiling - floor);

signalRight.material.SetColor("\_EmissionColor", new Color(1f, 1f, 1f) \* emission);

rearSignalRight.material.SetColor("\_EmissionColor", new Color(1f, 1f, 1f) \* emission);

}

else

{

signalLeft.material = signalON;

signalRight.material = signalOff;

rearSignalLeft.material = signalON;

rearSignalRight.material = signalOff;

float floor = 0f;

float ceiling = 1f;

float emission = floor + Mathf.PingPong(Time.time \* 2f, ceiling - floor);

signalLeft.material.SetColor("\_EmissionColor", new Color(1f, 1f, 1f) \* emission);

rearSignalLeft.material.SetColor("\_EmissionColor", new Color(1f, 1f, 1f) \* emission);

}

}

\*semnalizarea într-un sens giratoriu mic

* Evitarea obstacolelor:

Pentru detectarea obiectelor au fost construiți și plasați frontal 5 senzori cu lungimea și unghiul de orientare editabile în funcție de scopul și nevoile fiecărei mașini. De asemenea, am împărțit obiectele din joc în anumite categorii și le-am acordat un tag sugestiv după care urmează să se realizeze recunoașterea. În momentul în care un senzor lovește un obiect cu un tag corespunzător, mașina va intra într-o fază de evitare ignorând pe moment path-ul prestabilit și ajustând unghiul de bracare al roților astfel încât evitarea obstacolului să fie posibilă.

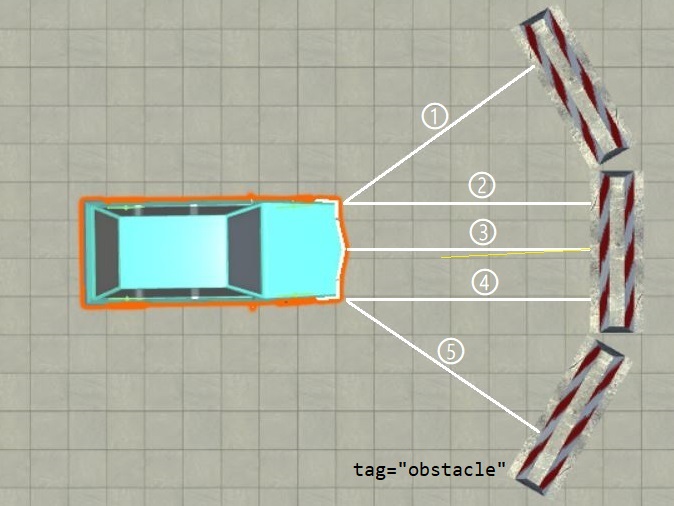


Fig. 23 Senzori mașină

* 1. Obstacol în stânga mașinii – deoarece riscul de impact este scăzut, se va aplica o corecție de +0.5 înmulțită cu unghiul maxSteerAngle.
  2. Obstacol in fața mașinii – deoarece impactul este iminent, se va aplica o corecție de +1 înmulțită cu unghiul maxSteerAngle. Rezultatul va fi un viraj maxim la dreapta pentru a evita obstacolul detectat pe partea stangă.

1. Obstacol in fața masinii – deoarece impactul este iminent, se va aplica o corecție de -1 înmulțită cu unghiul maxSteerAngle. Rezultatul va fi un viraj maxim la stânga pentru a evita obstacolul detectat pe partea dreaptă.
2. Obstacol in dreapta mașinii – deoarece riscul de impact este scăzut, se va aplica o corecție de -0.5 înmulțită cu unghiul maxSteerAngle.
   1. Obstacol in fața mașinii – acest senzor intervine pentru a se lua o decizie în cazul în care atât senzorul cu numărul 2 cât și senzorul cu numărul 4 detectează un obstacol. În acest caz, corecția ar fi de +1-1=0 deci mașina ar merge înainte. Prin urmare, trebuie să construim normala obiectului detectat și să luăm o decizie în funcție de aceasta.

(normala < 0) ? corecție = 1 : corecție = -1

**III.7. Controlul jucătorului**

Pe parcursul jocului controlul mașinii alternează între automat și manual. Prin automat înțelegem că mașina urmărește de una singură un anumit traseu și realizează anumite manevre cu scopul de a prelua atenția jucătorului de pe efectuarea mișcărilor și de a o canaliza pe ceea ce se întâmplă în trafic. Astfel, când o nouă întrebare apare pe ecran, se testează capacitatea jucătorului de a aprecia diferite situații. Când mașina este pe modul manual, jucătorul are responsabilitate totală asupra a ceea ce se întâmplă, deciziile sale din trafic conducând la următoarele consecințe:

* Întreruperea examenului auto – atunci când utilizatorul a întocmit o greseală majoră precum nerespectarea indicațiilor, nerespectarea timpului maxim de efectuare al manevrei, neacordare de prioritate, trecere pe culoare roșie a semaforului etc.
* Continuarea examenului auto dar neacordarea de punctaje – situație prezentă atunci când utilizatorul nu comite o greșeală majoră. Acesta își poate continua traseul însă pierde puncte pentru execuția incorectă a manevrelor (ex.: semnalizarea incorectă într-un sens giratoriu).
* Continuarea examenului auto și primirea de punctaje corespunzătoare – atunci când jucătorul efectuează manevra fără greșeli.

În toate cele 3 cazuri, jucătorul este anunțat la încheirea manevrei dacă a efectuat mișcările în mod corect sau ce a greșit în cazul în care manevra nu a fost efectuată corespunzător.

Atât controlul manual al mașinii cât și cel automat sunt implementate în același script. Tipul controlului este definit de variabila globală “self\_driving” care la rândul ei își schimbă valoarea în funcție de anumite checkpoint-uri de-a lungul jocului. Pe ramura ‘if’ a funcției ‘FixedUpdate’ (care se apelează la un număr regulat de frame-uri) se apelează funcțiile de conducere autonomă iar pe ramură ‘else’ este implementat controlul manual al mașinii. Funcția ‘WaypointDistane’ are rolul de a verifică checkpointurile prestabilite pentru jucător indiferent de modul de control al mașinii.

void FixedUpdate () {

if (GlobalVariables.self\_driving)

{

ApplySteer();

Drive();

Braking();

SmoothSteer();

}

else

{

Forward = Input.GetAxis("Vertical");

Turn = Input.GetAxis("Horizontal");

Brake = Input.GetAxis("Jump");

[...]

}

WaypointDistance();

}

\*controlul mașinii

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

* W, săgeata în sus – mergi înainte;
* S, săgeata în jos – frână ușoară, mergi înapoi;
* Spațiu – frână puternică;
* A, săgeata stanga – viraj stânga;
* D, săgeata dreapta – viraj dreapta;
* Q – semanl stânga pornit/oprit;
* E – semnal dreapta pornit/oprit;
* P – pauză.

**Capitolul IV. Concluzii finale 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. Deși fiecare om își dezvoltă un stil propriu de învățare, una dintre cele mai bune metode didactice rămâne învățarea prin joc. Aplicația are rolul de a introduce utilizatorul într-un mediu cât mai apropiat de realitatea de la volan încercându-se pregătirea atât din punct de vedere teoretic cât și practic.

În opinia mea, aplicațiile de acest gen au un impact destul de mare asupra utilizatorilor deoarece sunt puține aplicații pe piață care să lucreze în același scop ca “My driving school”, principala lor problema fiind lipsa restricțiilor și o libertate prea mare acordată jucătorului. În plus, întrebările teoretice sunt foarte aseamanatoare cu cele din chestionarele auto standard, diferența sesizabilă fiind modul mai interactiv de prezentare al situațiilor. Acest joc utilizează destul de intens atenția jucătorului, exact cum ar trebui să fie și în trafic.

Îmbunătățiri și direcții viitoare:

* Trecerea la un alt tip de grafică pentru a permite scalarea aplicației în funcție de rezoluția ecranului fără a pierde din calitatea imaginii;
* Introducerea unor noi situații în trafic precum depășiri, mașini speciale etc. Făcând o paralelă cu traficul din viața reală, pe aceasta îmbunătățire se poate lucra foarte mult;
* Crearea unui mod în care utilizatorul învață să parcheze;
* Implementarea unui mediu de antrenamen doar cu întrebări unde acțiunile mașinilor să fie dictate de răspunsurile alese de jucător.
* Adăugarea de noi modele 3D mai plăcute din punct de vedere vizual.

**Capitolul V. Bibliografie**

[1] Informații Unity

unity3d.com/

[2] Statistici cu privire la promovabilitatea examenului auto in România

scoaladesoferi.odat.ro/rata-promovabilitate-scoli-auto/

[3] Informații simulatoare profesionale

en.wikipedia.org/wiki/Driving\_simulator

[4] City Car Driving

[citycardriving.com/](https://citycardriving.com/)

[5] Driving school 2017

[play.google.com/store/apps/details?id=com.ovilex.drivingschool2017&hl=en](https://play.google.com/store/apps/details?id=com.ovilex.drivingschool2017&hl=en)

[6] Unity asset store

[assetstore.unity.com/](https://assetstore.unity.com/)

[7] Top genuri ale jocurilor video

sites.google.com/site/reviewgames11/top-genres-video-games