

UNIVERSITATEA POLITEHNICA BUCUREȘTI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE



PROIECT DE DIPLOMĂ

Racing For Real 2D

Boncu Dragoș-Andrei

Coordonator științific:

Prof. univ. Dr. ing. Florica Moldoveanu

BUCUREȘTI

2022

UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE DEPARTMENT



DIPLOMA PROJECT

Racing For Real 2D

Boncu Dragoş-Andrei

Thesis advisor:

Prof. univ. Dr. ing. Florica Moldoveanu

BUCHAREST

2022

CUPRINS

Sinopsis	3
Abstract.....	3
Mulțumiri	4
1. Introducere	5
1.1. Context	5
1.2. Problema	5
1.3. Obiective	6
1.4. Structura lucrării.....	6
2. Analiza și specificarea cerințelor	7
3. Studiu de piață.....	9
3.1. Istoric.....	9
3.2. Indicatori de evaluare a produselor și limitările clienților	11
3.3. Soluții asemănătoare	11
3.4. Tehnologii folosite.....	13
3.5. Tehnologii alternative	14
4. Soluția propusă.....	16
5. Detalii de implementare.....	19
5.1. Meniul	19
5.1.1. Meniul principal	20
5.1.2. Meniul opțiunilor	21
5.1.3. Garajul.....	22
5.1.4. Lista curselor	24
5.2. Cursa.....	24
5.2.1. Mașina.....	25
5.2.2. Elementele de UI.....	29
5.2.3. Meniul de pauză.....	30
5.2.4. Adversarii	31
5.2.5. Mediul	31
5.2.6. Coliziunea exterioară	32
6. Evaluarea rezultatelor	34

7. Concluzii.....	36
8. Bibliografie.....	37

SINOPSIS

Lucrarea prezintă realizarea unui joc video 2D de curse. Având în vedere performanțele curente care se pot atinge când vorbim despre construirea unui joc, jocurile 2D sunt considerate, de majoritatea populației, inferioare celor 3D, din numeroase aspecte, cel mai întâlnit fiind faptul că, un joc 2D este un joc la nivelul căruia s-a muncit mai puțin, și astfel, nivelul de satisfacție a jucătorului nu se va ridica la același standard. Totuși, beneficiile pe care jocurile 3D le aduc, împreună cu părerile fanilor acestor tipuri de jocuri, nu vor distruge niciodată viitorul apropiat al jocurilor 2D. Deși jocurile 2D au fost primele, și sunt intitulate mai vechi, dezvoltarea lor va rămâne la fel de interesantă, și efortul depus elaborării lor va crește pe aceeași lungime de undă cu a celor 3D. Aplicația prezentată în lucrare este un joc distractiv, ce încearcă să aducă cât mai multe caracteristici prezente și plăcute de public pe piața jocurilor de acest gen, și, indiferent că nu este unic în această categorie, dorește să participe în această cursă a îmbinării distracției și relaxării cu îndemânarea și competitivitatea.

ABSTRACT

The paper is about a 2D racing video game. Considering the current performances that can be achieved when we talk about building a game, 2D games are considered, by the majority of the population, inferior to 3D ones, in many aspects, the most common being the fact that a 2D game is a game that has had less work done on its level, and thus the level of player satisfaction will not rise to the same standard. However, the benefits that 3D games bring, along with the opinions of fans of these types of games, will never destroy the near future of 2D games. Although 2D games were first, and are titled older, their development will remain just as interesting and the effort put into their development will grow on the same wavelength as 3D ones. The application presented in the paper is a fun game, which tries to bring as many features present and liked by the public as possible to the market of games of this kind, and, regardless of the fact that it is not unique in this category, it wants to participate in this race of combining fun and relaxation with skill and competitiveness.

MULȚUMIRI

Având în vedere tehnologiile folosite și întemeierea bazelor acestei aplicații, primele mulțumiri se adresează motivului urmăririi acestei căi, și anume, participarea la școala de vară 3DUPB, împreună cu participarea la workshop-ul “Rapid Game Development” și a echipei din spatele acestuia, Alexandru Grădinaru și Maria-Anca Băluțoiu. Totodată, vreau să aduc multe mulțumiri doamnei profesoare dr. Ing. Florica Moldoveanu pentru acceptarea de a mă îndruma către acest rezultat și echipei valoroase din care face parte, ce mi-a oferit plăcerea de studia interactiv și creativ în cadrul ciclului de licență la materii ca Ingineria Programelor, Elemente de Grafică pe Calculator și Sisteme de Prelucrare Grafică. De asemenea, mulțumesc prietenilor ce m-au ajutat pe parcursul dezvoltării acestui proiect, cu idei sau cu feedback în urma testării produsului.

1. INTRODUCERE

1.1. Context

Proiectul constituie un joc video 2D de curse. Jocul este realizat folosind o tehnologie des întâlnită și utilizată, Unity. Jocul presupune întrecerea în opoziție cu timpul și cu alți adversari “artificiali”, de a gestiona corect banii în obținerea autovehiculelor și de a pune în aplicare aptitudinile pe care le va dobândi pe parcurs. Obiectivul final este de a debloca dorința utilizatorului de a obține fiecare mașină și de a le customiza la performanțele maxime și, totodată, de a deveni cât mai bun la nivel de competitivitate.

Proiectul reprezintă un prototip realizat în paralel cu studierea acestei tehnologii folosite în cadrul unui workshop de game development. Fiind încă la început, cea mai bună a fost alegerea de a porni dezvoltarea pe o ramură bidimensională, indiferent că opțiunile sunt mult mai limitate față de cea tridimensională. Ideea alegerii categoriei jocului a fost o preferință proprie pe care o consider interesantă și care îmi creează cea mai mare plăcere, indiferent că este destul de dificilă pentru un începător.

Jocurile din categoria racing se pot încadra în numeroase subcategorii, cea ideală potrivirii cu aplicația prezentă fiind categoria “Simulation Racing”. Jocurile de acest gen presupun dezvoltarea unei mecanici destul de avansate în jurul aspectului său. Acestea se străduiesc să reproducă în mod convingător manevrarea unui vehicul real. Pentru a îmbunătăți acest lucru, producătorii jocurilor de acest tip licențiază mașini reale sau ligi ce încadrează mașini de curse, dar, ca și în cazul de față, folosesc mașini fantastice, unele încercând să apropie înfățișarea de realitate, altele nu. Factorul cheie al acestor jocuri este fizica comportamentului mașinii. Tehnica corectă de viraj și manevrele de curse au prioritate în categoria de simulator de curse. Multe jocuri și aplicații de acest gen sunt folosite și la nivel înalt, și se pot stăpâni doar de persoane experimentate. Spre exemplu, sunt simulatoare din această categorie ce sunt prezente în antrenamentele off-track ale piloților de Formula 1. Acestea, spre deosebire de alte jocuri ce prezintă ajutoare adiționale pentru stăpânirea mașinii, ca ABS sau Traction Control, sunt contruite cât mai realist și îngreunează condusul monoposturilor cât mai mult pentru piloți, dirijând activitatea spre o experiență cât mai reală și în același timp sigură.

Vorbind, din nou, despre realistica oferită de această categorie, odată cu dezvoltarea din ce în ce mai mare a jocurilor, s-au dezvoltat atât componentele calculatoarelor sau consolelor cât și perifericele acestora. Realistica jocurilor de acest gen, și nu numai, sunt puternic influențate de prezența unui periferic corespunzător cu categoria respectivă. Spre exemplu, un joc simulator de curse, se simte diferit la un volan cu schimbator față de jucatul propriu-zis cu tastatura.

1.2. Problema

Problema principală o constituie interesul tot mai scăzut pentru jocurile 2D, indiferent de categorie. Jucătorii înrăiți sau persoanele ce frecventează jocurile, atât pe calculator, cât și pe

consolă, își îndreaptă privirea tot mai mult către jocurile 3D. Nu este un lucru rău faptul că acestea au mai mulți fani. Într-adevăr, jocurile 2D au o barieră în dezvoltarea lor pe care o atinge mult mai rapid decât celelalte, dar, acestea, nu pot fi numite doar vechi și plictisitoare. Chiar și în momentul acesta, există jocuri de acest tip în dezvoltare, la nivelul cărora se depune foarte mult efort și se încearcă ca experiența oferită pe parcursul jocului să fie una cât mai distractivă și unică.

1.3. Obiective

Obiectivul proiectului este de a captiva atenția jucătorului și de a demonstra faptul că un joc bidimensional prezintă și el greutatea lui în dezvoltare, și poate oferi distracție și realistă la fel de mult ca unul tridimensional. Vorbind strict despre categoria specificată mai sus, și anume, jocurile de curse, integrată, atât în 3D, cât și în 2D, este și va rămâne egal de plăcută și satisfăcătoare publicului prin simplitatea sa și senzația de adrenalină pe care o oferă [\[1\]](#).

Pe de altă parte, o subcategorie atinsă de această aplicație, simulatoare de curse, reprezintă un alt mod de a menține jucătorii activi. Indiferent că un joc 3D va arata mai bine la față și va parea mult mai realist, când este vorba de un simulator de curse, ce este în spate este mai important, cum se comportă un obiect, în cazul nostru mașina, pe parcursul jocului, în condiții reale, este mult mai important și va conta mult mai mult pentru pasionații de acest gen. Jocul prezentat încercă să aducă caracteristici prezente și plăcute de utilizatorii din această categorie și prezintă multe moduri prin care să se apropie de realitate.

1.4. Structura lucrării

Lucrarea parcurge mai mulți pași pentru a realiza o demonstrație și pentru a accentua viziunea despre principala problemă împreună cu posibila rezolvare a acesteia. Capitolul 2 descrie planul de abordare a soluției împreună cu cerințele clienților de a minimiza numărul de probleme. Capitolul 3 prezintă câteva idei asemănătoare, ce pot reprezenta surse de inspirație pentru viitoare proiecte, dar și tehnologiile folosite în cadrul proiectului împreună cu câteva căi alternative. Capitolele 4 și 5 au o mică legătură întrucât, capitolul 4 descrie soluția propusă și trunchiul de funcționare al aplicației prin scheme aferente, iar capitolul 5 descrie modul și logica de implementare a întregului proiect analizând fiecare mic pas din construcția sa. Într-un final, capitolul 6 prezintă câteva comparații între produsul final și câteva idei asemănătoare și ideale, prezentate mai pe larg și în capitolul 3.

2. ANALIZA ȘI SPECIFICAREA CERINȚELOR

Pe parcursul întemeierii planului dar și al jocului propriu-zis s-a realizat o analiză la preferințele utilizatorilor, și anume, ce își doresc să regăsească la nivelul aplicației.

Adăugându-mă și pe mine, am adunat utilizatori care se încadrează în categoria celor potriviți pentru acest tip de joc, simulator de curse. Am considerat potrivită și adăugarea mea în cadrul acestui interviu deoarece mă consider un candidat ideal pentru această arie de jocuri, din care am dezvoltat suficiente cunoștințe, fapt ce reprezintă și motivația personală de a duce la capăt acest proiect.

Pentru întocmirea planului și punerea bazelor aplicației s-au obținut, pentru început, următoarele cerințe:

- adăugarea mai multor mașini, de diferite categorii, altfel încât să acopere mai multe arii, inclusiv cea a motosportului(Formula);
- adăugarea mai multor curse, astfel încât utilizatorul să nu se plictisească parcurgând aceeași cursă de mai multe ori, și, totodată, diversificarea lor, astfel încât să existe curse mai ușoare, curse mai grele, curse mai scurte, curse mai lungi;
- un sistem de gestionat banii, o provocare pe care o va avea jucătorul, de a fi atent la riscul pe care și-l asumă când cumpără o mașină ori când alege o cursă de dificultate înaltă;
- un sistem de customizat vehiculul, astfel încât, cu banii obținuți, să îți poți ridica mașina la o performanță mai înaltă;
- o fizică a mașinii cât mai realistă, cu valori diferite ale componentelor de la mașină la mașină, mașini mai rapide, mașini cu o accelerație mai bună, mașini mai bune pe curbe.

Lansarea jocului s-a efectuat pe un executabil de Windows pe 64 biți, generat automat de către Unity ce a avut inițial probleme ale rezoluției de la utilizator la utilizator, în sensul în care multe obiecte ale jocului apăreau în locuri necorespunzătoare.

Ulterior, după testarea produsului de către jucători, s-au realizat alte propuneri, plecând de la prima variantă a aplicației, pentru a încadra jocul și mai mult în subcategoria de simulator:

- adăugarea unor boți ce vor încurca jucătorul în parcurgerea traseului prin coliziunea cu ei;
- îmbunătățirea fizicii mașinii cu un sistem de schimbare a treptelor de viteze în funcție de rotațiile pe minut ale motorului;
- adăugarea unui stil nou de cursă, "Drag Race", ce presupune schimbarea manuală a treptelor de viteză, avantajul fiind primit în funcție de timpul de reacție mai bun la start, etc.;
- adăugarea unui motor bazat pe combustibil nitro, ce oferă un avantaj temporar accelerației vehiculului.

Cele de mai sus sunt doar câteva funcționalități care se pot adăuga jocului, procesul de testare și feedback putând continua pe oricât posibil, având în vedere că există o gamă largă de posibilități de implementare asupra jocului. Este de subliniat faptul că jocul ar putea trece pe viitor cu ușurință la un sistem manual de configurare al controalelor mașinii și astfel produce o experiență de joc mult mai realistă cu un volan virtual și schimbator. Nu în ultimul rând, pe lista de “update-uri” aduse jocului pe parcursul construcției, dar și în varianta finală, se regăsește fixarea erorilor și a lucrurilor ce se întâmplau neobișnuit prin testarea, atât a mea, cât și a clienților.

3. STUDIU DE PIAȚĂ

3.1. Istoric

Datorită performanțelor scăzute din trecut, dezvoltarea, atât a unui joc, cât și a unei aplicații necesită un pachet enorm de îndemânare și răbdare. Astfel primele jocuri dezvoltate au fost, evident, cele 2D. Un joc 2D este un joc realizat folosind grafică computerizată bidimensională. Grafică computerizată bidimensională se ocupă cu generarea de imagini digitale bazată pe computer, în cea mai mare parte din modele bidimensionale și prin tehnici specifice acestora. Modelele grafice 2D pot combina modele geometrice, imagini digitale, text de afișat, funcții și ecuații matematice, etc. Aceste componente pot fi modificate și manipulate prin transformări geometrice bidimensionale precum translația, rotația, scalarea.

Vorbind despre categoria jocurilor de curse, fiind doar bidimensionale, primele jocuri se expandau pe cele 2 axe în 2 moduri:

- modul 1, unde obiectele se mișcă de sus în jos și din stânga în dreapta, asemănător cu figura 1;



Figura 1 Privire de ansamblu de sus

- modul 2 unde obiectele se mișcă din față în spate și din stânga în dreapta, exact ca în figura 2.



Figura 2 Privire de ansamblu dintr-o parte(spate)

Primul joc apărut, din această categorie, a fost dezvoltat în 1973 de Atari, o filială și o marcă franceză ce a fost un pionier în jocuri arcade, console de jocuri și computere personale. Numele jocului este "Space Race" și este al doilea joc al companiei după jocul "Pong" ce a marcat începutul erei dezvoltării jocurilor. Jocul este despre doi piloți care au preluat controlul unor navelor spațiale în încercarea de a fi primii care traversează ecranul. Pe parcurs sunt liniuțe reprezentând asteroizi care se deplasează pe ecran de la stânga la dreapta, pe care jucătorii trebuie să le evite. Comenzile de mișcare sunt limitate la mutarea navei spațiale în sus sau în jos; dacă nava jucătorului este lovită de un asteroid, aceasta dispăre pentru câteva secunde înainte de a reapărea din nou în partea de jos [\[2\]](#).

Un alt joc de curse, apărut printre primele și reprezentând un nou mod de așezare pe ecran(modul 2) este "Pole Position". Apărut în 1982 este jocul ce a influențat cel mai mult piața și dezvoltarea în continuare, influențând producerea a peste 800 de noi jocuri. "Pole Position" este un joc dezvoltat doar pentru o singură persoană ce imită o cursă de Formula 1 unde concurează alte 7 mașini controlate de calculator. Obiectivul acestuia este de a obține un scor cât mai mare, scorul jucătorului fiind calculat în funcție de distanța parcursă în timpul oferit la dispoziție [\[3\]](#).

Următoarele jocuri au continuat să se răspândească de la realistică până la arcade. Nintendo a intrat pe piață cu două titluri iconice, "F-Zero", prezentând vehicule estetice și futuriste inspirate de science-fiction și în 1992 cu "Super Mario Kart" care a preluat aspecte ale curselor și le-a combinat cu lupta capcanelor. Mai presus de toate, cea mai importantă parte a anilor 90 pentru jocurile de curse a fost nașterea unor francize de lungă durată precum "Need For Speed" și "Gran Turismo".

3.2. Indicatori de evaluare a produselor și limitările clienților

Preferințele clienților și iubitorilor de jocuri diferă de la cer la pământ, indiferent că vorbim doar despre categoria de jocuri de curse, fie unii prefera să fie complet realist, fie parțial, fie complet ficțional, lucru ce ar putea reprezenta un indicator de evaluare ale acestor produse.

Având în vedere faptul că multe jocuri din această categorie aparțin unei francize, un alt indicator de evaluare ar fi jocul precedent, aspectele negative, venite la pachet cu noul joc sau prezente în continuare, și îmbunătățirile ce au fost solicitate din vremea jocului anterior. Desigur, schimbările majore aduse pot avea și consecințe, indiferent că sunt plăcute pentru majoritatea publicului sau nu. Aceste schimbări ar putea reprezenta limitări pentru iubitorii francizei respective, ce refuză să proceseze spre următoarea etapă care este schimbată drastic. Totodată, limitări pot fi considerate și dificultăți în parcurgerea jocului. După cum am spus în capitolele precedente, sunt jocuri ce au ca scop antrenamentul piloților reali de curse, jocuri care prezintă o dificultate destul de ridicată și astfel mulți preferă să-l evite decât să își dea silința în a îl învăța.

3.3. Soluții asemănătoare

Pentru a oferi câteva exemple de astfel de francize și pentru a asocia câteva jocuri similare cu soluția prezentă, care le sunt pe plac și clienților curenți ai aplicației, vom extrage câteva jocuri preferate de majoritatea iubitorilor de simulatoare care au ajuns la un nivel incredibil de mare pe plan de dezvoltare.

Need For Speed, abreviat și NFS, este o serie de jocuri video de curse dezvoltată pe mai multe platforme, primul joc, "The Need for Speed", fiind lansat în 1994. Electronic Arts a anunțat că franciza a vândut peste 100 de milioane de unități, făcând din "Need For Speed" seria de jocuri de curse cu cea mai mare succes comercial la acea vreme, iar "Need For Speed: Most Wanted" fiind cel mai bine vândut joc din serie, cu 16 milioane de exemplare vândute. Fiind o serie de jocuri foarte bine vândută, multe jocuri vechi ai primit și o versiune remasterizată, cu grafici înnoite, mașini noi adăugate, detalii modificate și adaptate cu versiunile noi, etc. Popularitatea jocului a ajuns chiar și la nivel de cinema, filmul cu același nume este o adaptare cinematografică bazată pe franciză care a fost lansată pe 14 martie 2014 și la producerea lui au contribuit atât DreamWorks Studios, cât și de Electronic Arts, franciza jocului inspirație. Franciza a dezvoltat un joc de curse extrem de plăcut și iubit chiar și de iubitorii de simulatoare, cu toate că jocul nu este integral realist, cel puțin la nivelul fizicii [\[4\]](#).

The Crew, tot o serie asemănătoare cu NFS, a devenit popular cu lansarea jocului cu titlul al doilea, "The Crew 2", apărut în 2018 a fost un joc de succes care a fost comparat și care s-a batut cu "Forza Horizon 4" pentru cele mai bune jocuri din perioada respectivă. La fel ca și jocurile NFS, prezintă o gamă largă de mașini care se pot customiza într-un mod extrem de avansat iar băcăliile se desfășoară pe străzile întregii Americi în miniatură. Pe lângă faptul că jocul prezintă un mediu vast de desfășurare a acțiunii, dar, față de NFS, prezintă și alte moduri de întreceri din diferite sporturi: pe apă, în aer, cu motocicletele, cu monster truck-urile,

Formula 1, Turing, off-road, etc. Totuși, asemănător cu jocurile NFS, prezintă câteva aspecte mai puțin realiste care îi pun pe gânduri pe unii jucători sau clienți în alegerea acestuia [5].

Forza Horizon. În ultimii ani, este greu să vorbești despre un joc de curse bun și să nu menționezi titlul “Forza”, mai ales cu ultimele 2 jocuri “Forza Horizon 4” și “Forza Horizon 5”, jocuri ce au avut un succes global pentru franciză. “Forza Horizon 4” este absolut plin de opțiuni pentru jucătorii intermediari și experți. Majoritatea clienților doresc să se scufunde în meniul de setări, unde se găsește o cantitate amețitoare de dificultăți, asistență și ajustări de control care pot fi adaptate la anumite preferințe și stiluri de joc. Jocul reprezintă o demonstrație de realitate pe ecran, de la grafica imaculată și până la controalele perfecte ale mașinii. Atenția dezvoltatorului pentru detalii și măiestria de calitate este subliniată în special în cele 4 anotimpuri, oferind “mult mai mult decât zilele ploioase de primăvară, condițiile de gheață de iarnă, suprafața acoperită de soare de vară și frunzele căzute de toamnă”. Fiecare sezon introduce o serie de schimbări care nu numai că aduc frumusețe și efecte cosmetice unice, ci și modificări de mediu care afectează terenul și modul în care vehiculele reacționează la acesta. Din punctul meu de vedere, dar și al altor pasionați în acest domeniu, jocul este atât de atrăgător încât poate reprezenta o motivație pentru toți dezvoltatorii. Mai jos se regăsește o imagine(figura 3) ce pune una lângă alta, o poză reală și una din joc, pentru a demonstra cât de dificil este de a le interpreta și cât de ușor se apropie de realitate jocul [6].



Figura 3 Forza Horizon 4 vs Imagine reală

Pentru a alege și un joc 2D din aceeași categorie care a avut un succes destul de mare ne putem gândi la “Ultimate Racing 2D”. Fiind doar un joc 2D, într-adevăr, nu poate ajunge la aspectul fizic al jocurilor menționate mai sus, dar reprezintă o idee ideală pentru clienții

limitați de performanțele susținute de sistemul pe care are loc rularea jocului și pentru cei care sunt interesați mai mult de ce se ascunde în spatele jocului decât în față, conceptul de simulare al mecanicii și fizicii. “Ultimate Racing 2D” își face apariția cu stilul de joc de la Karting până la Formula. Utilizatorul se poate juca online alături de alți clienți pe una din pistele internaționale curent de Formula dar și istorice. Totodată, ca și în jocurile de mai sus, poate alege să concureze pe un traseu diferit: Dirt Racing, Touring Racing, Oval Racing, cu o mașină din categoria: Formula, motociclete, mașini nemodificate, Supercar-uri, tractoare, camioane, karturi, stivuitoare, mașini GT, etc. [7].

3.4. Tehnologii folosite

Cea mai importantă tehnologie folosită este instrumentul prin care a fost dezvoltat jocul, Unity. Unity reprezintă un motor de dezvoltare de jocuri, atât 3D, cât și 2D. Principalul obiectiv al motorului de joc este atât de a oferi cel mai robust set de instrumente posibil pentru industria de dezvoltare a jocurilor, cât și de a face cât mai ușor posibil pentru dezvoltatorii de jocuri de orice nivel de calificare să folosească motorul. S-a dezvoltat și în alte industrii, cu un accent mare pe dezvoltarea 3D în timp real, făcându-l unul dintre cele mai puternice motoare disponibile [8].

În cadrul acestuia, fiecare tip de grafică vine cu propriile instrumente de set specializate, în cazul graficii 2D fiind tăierea sprite sheet, și chiar are propriile API-uri de script apelate pentru diferite opțiuni de fizică care sunt potrivite pentru fiecare stil. API-urile oferă acces rapid la cele mai frecvente funcții necesare și include atât caracteristici generale ale jocului, cât și apeluri API specifice care permit accesarea și modificarea anumitor componente, astfel, ajustarea unor valori, spre exemplu, să fie permisă, atât din Unity Inspector, cât și din cod.

Cât despre aceste componente, atașarea lor către obiectele jocului este esențială, atât pentru partea vizuală cât și pentru partea de funcționalitate. Câteva dintre cele mai des întâlnite în cadrul proiectului sunt: Sprite Renderer, Rigidbody 2D, Box Collider 2D. Pe lângă acestea, Unity permite crearea unor componente personalizate cu ajutorul scripturilor. Unity acceptă, pentru scrierea scripturilor, limbajul de programare C#.

C# este un limbaj standard din industrie similar cu Java sau C++. Un script își face legătura cu funcționarea internă a Unity prin implementarea unei clase care derivă din clasa încorporată numită MonoBehaviour. Această clasă devine noua componentă a obiectului și va crea, la atașare, o nouă instanță a obiectului definit de proiect. Clasa prezintă 2 funcții inițiale de bază: Start și Update. Funcția Start este funcția apelată o singură dată pe parcursul jocului, de la instanțierea obiectului și până la distrugerea lui. Funcția are rolul de a seta parametrii de start ale obiectului înainte de începerea jocului propriu-zis. După apelarea funcției și începerea jocului, la fiecare cadru, va fi apelată funcția Update. Update este apelată încontinuu și are rolul de a actualiza constant parametrii, fie cei declarați inițial, fie alți parametrii ajutători [9].

Am ales să continui dezvoltarea proiectului în acest mod deoarece Unity prezintă o interfață de comunicare cu dezvoltatorul extrem de plăcută și satisfăcătoare. Acesta oferă o senzație

de confort și control absolut al capacităților de dezvoltare pentru orice utilizator, indiferent de nivelul de cunoștințe și, datorită numărului mare de tehnologii ajutoare care vin la pachet cu acesta, oferă încredere și impresia că nimic nu este imposibil de dezvoltat. Unity este extrem de folositor și cu materiale pentru proiect. Unity furnizează materiale dezvoltatorilor prin intermediul "Unity Asset Store" de diferite tipuri, puse la vânzare de creatorii de conținut. Acest lucru a reprezentat un avantaj pentru proiectul meu ce a necesitat o mulțime destul de mare de sprite-uri pentru obiectele folosite în joc.

3.5. Tehnologii alternative

Unreal Engine este un motor de joc la fel de popular ca și Unity utilizat pe scară largă, dezvoltat de Epic Games, una dintre cele mai mari platforme de vânzare jocuri. Este folosit în multe jocuri moderne, cum ar fi propriul shooter al platformei, Fortnite sau alte jocuri de succes precum „Rocket League”. Permite dezvoltarea pe mai multe platforme de la PC la console precum PS4, Xbox One și Nintendo Switch, motiv pentru care este atât de utilizat pe scară largă, fiind flexibil de a lucra între aceste platforme diferite. Jocul prezintă scripturi, la fel ca și Unity, în care, programatorii mai experimentați, pot folosi C++. De asemenea, are materiale puternice și instrumente de animație pentru artiști, care permit realizarea rapidă de scene complexe. Un mare avantaj al Unreal Engine este că este complet gratuit de utilizat. Indiferent dacă utilizatorul va lua rolul de dezvoltator pasionat sau un studio AAA, nu există nicio taxă în avans pentru a utiliza Unreal. Așadar, Unreal Engine reprezintă o metodă foarte asemănătoare cu Unity pentru dezvoltarea unui proiect asemănător cu cel prezentat însoțit de plusuri și minusuri, decizia rămânând la plăcerea dezvoltatorului [\[10\]](#).

GameMaker este un motor pentru începători, artiști, scriitori, non-programatori, oameni cu idei care nu au scris niciodată o singură linie de cod și nu ar ști de unde să înceapă. Este un motor 2D de ultimă generație, un motor de jocuri de acțiune de sus în jos, motor pixel art. După părerea multor dezvoltatori, GameMaker este cel mai bun motor când este vorba de construcție de jocuri 2D bazate pe sprite. La fel ca și Unity, folosește un sistem drag-and-drop pentru plasarea și modificarea obiectelor componente ale jocului, dar, totodată folosește propriul limbaj. Limbajul de cod folosit de motor se numește „GameMaker Language (GML)” Nu este la fel de robust sau orientat pe obiecte precum limbajele obișnuite precum C# sau C++, dar pentru începători este o modalitate mult mai plăcută. Astfel, GameMaker, este o variantă îndreptată pentru începătorii din domeniul de programare pentru crearea unui astfel de proiect [\[11\]](#).

PyGame este cea mai populară bibliotecă pentru dezvoltare de jocuri din Python. Python este un program extrem de evitat când vine vorba de această categorie, dezvoltare de jocuri. Este estimat că poate ajunge până la de 25 de ori mai încet ca C++, diferență de la cer la pământ. Totuși, Python nu este degeaba cel mai popular limbaj de programare și prezintă și avantaje. Datorită simplității, Python este o alegere ideală pentru prototipuri deoarece livrează efectele muncii imediat și, pentru proiecte asemănătoare cu acestea, reprezintă un avantaj. PyGame este foarte bun pentru începători, este un tool foarte ușor de învățat și poate

rulea pe orice sistem de operare. Desigur, PyGame reprezintă o simplă bibliotecă și, pe cât de excepțională este, este limitată doar la jocurile bidimensionale. Așadar, pentru o dezvoltare, mai departe, către 3D, se pot folosi alte biblioteci Python ca Panda3D sau PyKya [\[12\]](#).

OpenGL sau Open Graphics Library este interfața standard din industria computerelor pentru definirea imaginilor grafice 2D și 3D. Prin OpenGL se specifică un set de comenzi sau funcții care urmează să fie executate. Fiecare comandă direcționează o acțiune de desen sau provoacă efecte speciale. Astfel, se poate crea o listă cu aceste comenzi pentru producerea unui efect repetitiv. OpenGL vine cu un număr mare de capabilități încorporate solicitate prin API. Acestea includ eliminarea suprafețelor ascunse, producerea transparenței, antialiasing, maparea texturii, operațiile cu pixelii, transformările de vizualizare și modelare și efecte atmosferice ca și nitro-ul sau fumul. OpenGL reprezintă o metodă extrem de grea pentru a dezvolta acest tip de proiect care nu se bazează doar pe aspect, dar nu reprezintă un mijloc imposibil [\[13\]](#).

4. SOLUȚIA PROPUȘĂ

Aplicația, în ciuda faptului că reprezintă o variantă finală și completă a jocului, nu prezintă foarte multe interacțiuni ale utilizatorului cu interfața pentru a complica experiența folosirii acesteia. Figura 4, reprezentând diagrama de cazuri de utilizare, va ilustra fiecare posibil contact al unui jucător uzual cu aplicația.

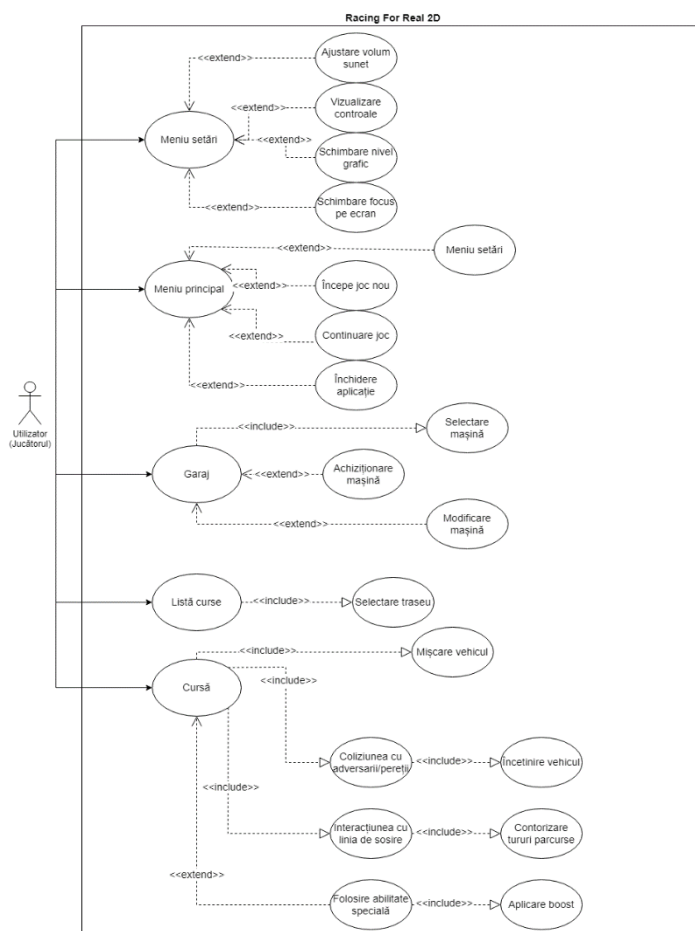


Figura 4 Diagrama cazurilor de utilizare

De asemenea, avem prezentă o diagramă de activitate, figura 5, pentru conturarea întregului traseu pe care îl are utilizatorul de la un capăt la altul, de la lansarea programului și până la finalizarea unui ciclu de joc(ciclu de joc însemnând participarea la o singură cursă, cursa fiind etapa "finală", chiar dacă jucătorul poate relua acest proces la infinit).

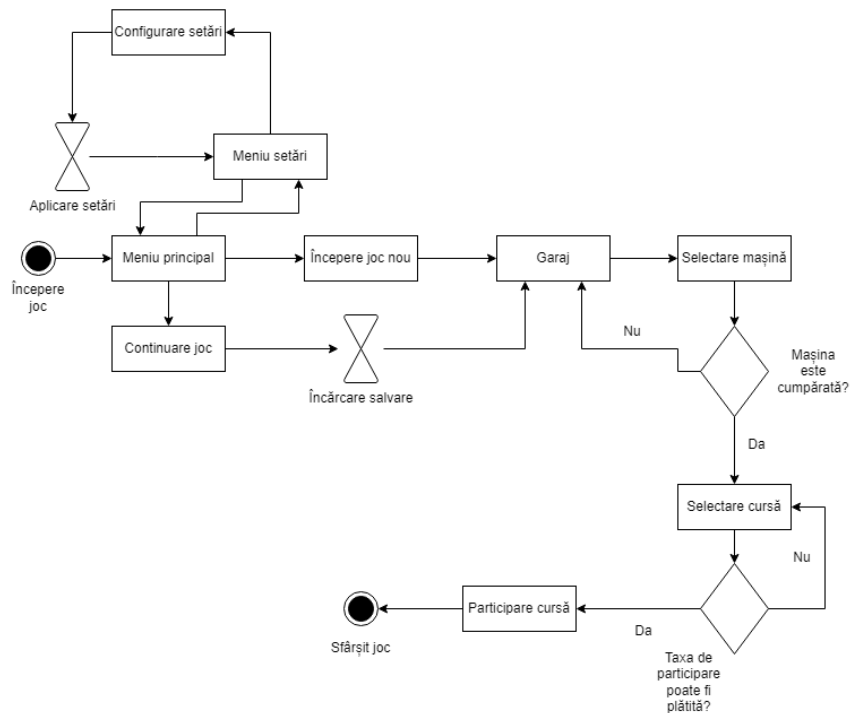


Figura 5 Diagrama de activitate

Numărul etapelor este relativ mic iar singurele obstacole de care se pot lovi clienții sunt cele 2 condiții impuse (cele afișate cu figura geometrică romb) legate de sistemul de gestionat banii ce interzice astfel ca suma deținută de jucător să fie negativă și participarea în cursă să se realizeze doar cu o mașină deblocată de acesta, pentru a nu oferi un avantaj considerabil în ciuda câștigurilor slabe pe parcursul carierei.

Vorbind despre mașină și revenind simultan la imaginea anterioară, figura 4, vehiculul controlat de jucător și implementarea acestuia reprezintă “personajul principal” din această aplicație, indiferent de aspectul său de la mașină la mașină. Scena cursei propriu-zise reprezintă scopul întregii aplicații, iar vehiculul reprezintă baza acesteia deoarece controlul său de către utilizator va aduce rezultate dorite pe parcurs. Astfel, avem cea mai importantă clasă, clasa Mașină (figura 6), integrată în scriptul “CarScript” atribuit ca și componentă obiectelor de acest tip, care se va ocupa de fixarea mecanicii mașinii.

Mașină	
Top Speed	float
Acceleration	float
Brake Force	float
Handling	float
Nitro	float
Speed	float

Figura 6 Clasa mașină împreună cu variabilele și tipurile lor

Mașina prezintă 4 variabile importante folosite pe întreaga definiție a mecanicii: viteză maximă, accelerație, forță de frânare și manevrabilitate. Aceste atribute vor diferenția obiectele de acest tip între ele, vor crea disproporțiile dintre performanțele mașinilor din garaj, astfel, cu cât valorile lor sunt mai mari, cu atât mașina este mai bună, iar singura metodă de a le modifica este prin intermediul procesului de upgrade, ce le vor aduce la o valoare maximă standard.

Pe lângă aceste 4 atribute fixe, mai putem adăuga 2 adiționale, valoarea rezervorului de nitro și viteza curentă, valori ce pot varia la fiecare frame, spre deosebire de cele 4 care sunt permanent constante.

Pe parcursul mișcării obiectului din clasa mașină, viteza este modificată prin intermediul unui calcul matematic al valorilor accelerației și al forței de frânare, depinzând ce tastă este, sau nu, apăsată în momentul respectiv, iar valoarea maximă și minimă, în cazul în care mașina merge cu spatele, va depinde de valoarea top speed-ului. Pe de altă parte, handling-ul va reprezenta un parametru folosit la rotirea mașinii și nu va avea nici o legătură cu viteza mașinii. Boost-ul oferit la activarea alimentării cu nitro va crește temporar accelerația și viteza maximă și va reveni la normal imediat cu dezactivarea lui.

5. DETALII DE IMPLEMENTARE

Jocul se bazează pe un sistem de funcționare destul de simplu. Jucătorul prezintă un Store unde își poate alege mașina cu care dorește să concureze, atât timp cât o deține, și unde o poate configura la o performanță superioară. În funcție de banii din cont, își poate alege un traseu pe care să concureze, analizând performanțele necesare pentru un succes și riscul pe care îl asuma în raport cu premiul/taxa de participare. La nivelul cursei propriu-zise, va trebui să parcurgă toate lap-urile necesare și să atingă linia de sosire cu un timp total mai mic decât cel indicat în dreptul informațiilor fiecărui traseu. Totodată, va întâlni un număr de oponenți ce îi va îngreuna parcurgerea traseului prin evitarea coliziunilor cu aceștia, dar, îi vor oferi și o idee cu privire la progresul cursei, mai precis, cât de rapid trebuie să fie.

5.1. Meniul

Meniul reprezintă interfața utilizatorului, interfața cu care utilizatorul va interacționa cel mai mult, și este alcătuit doar din elemente ce țin strict de UI, cum sunt: butoanele, slidererele, imaginile(de background), toggle-urile, dropdown-urile și textele.

Jocul deține un font specific tuturor textelor, atât simple, cât și de la nivelul altor elemente de UI creat cu ajutorul componentei TextMeshPro ce permite modificarea mai în detaliu a acestora. Fontul arată astfel(figura 7):



Figura 7 Textul fără highlight

În cazul elementelor de UI, când utilizatorul se află cu cursorul mouse-ului pe acestea, un element de highlight se va activa automat pentru a înștiința jucătorul ca poate interacționa cu elementul respectiv. Astfel, textul de mai sus va arăta așa(figura 8):



Figura 8 Textul cu highlight

Navigarea prin meniu se realizează conform schemei de mai jos(figura 9):

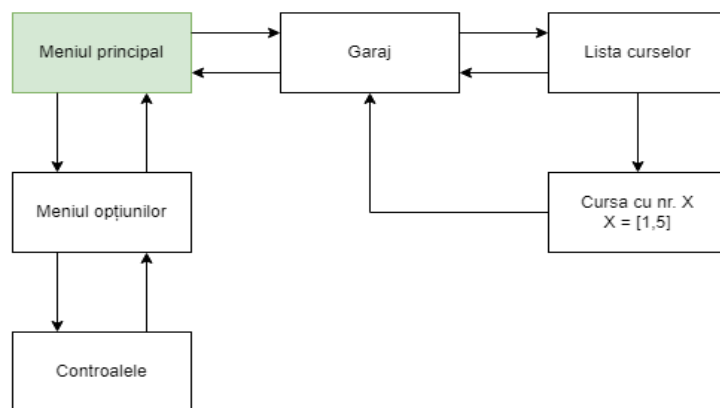


Figura 9 Schema de navigare în meniu

5.1.1. Meniul principal

Meniul principal(figura 10) prezintă 4 butoane:

- “New Game” ce va încarca o nouă scena pe fundal cu Garajul și cu setările inițiale ale jocului(numărul de bani inițial, nici o mașină deținută și nici o mașină modificată);
- “Continue” ce va realiza un lucru asemănător doar că va prelua informațiile salvate într-un fișier din directorul jocului și astfel va încarca scena cu informații diferite față de butonul precedent;
- “Options” ce va schimba perspectiva curentă cu meniul opțiunilor ca în figura 11;
- “Quit” ce va închide aplicația printr-o simplă comandă.

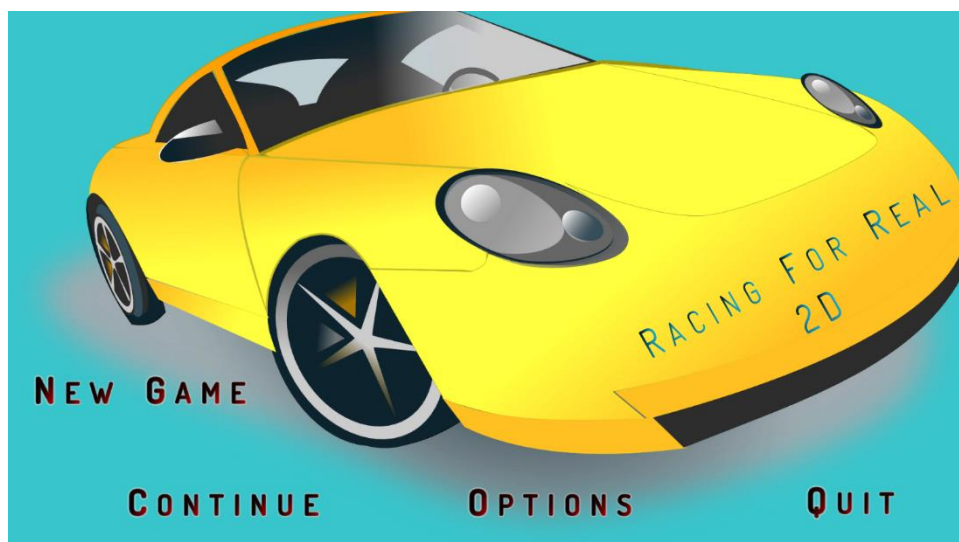


Figura 10 Meniul principal

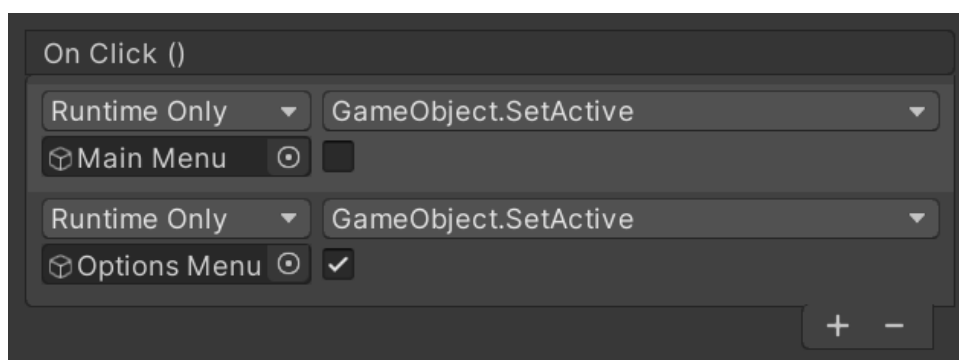


Figura 11 Componenta de schimbare a meniurilor

5.1.2. Meniul opțiunilor

Opțiunile disponibile la nivelul aplicației sunt:

- de a schimba nivelul grafic pe high, medium sau low, accesând dropdown-ul din stânga meniului;
- de a seta volumul sunetului venit de la elementele cu o componentă specifică producerii sunetului, modificând butonul slider-ului din mijlocul meniului;
- de a transforma jocul dintr-o aplicație ce nu apare pe întregul ecran(windowed) într-o aplicație ce apare pe întregul ecran(fullscreen) și invers;
- de a vizualiza informații despre controalele folosite pe parcursul jocului(figura 13).

De asemenea, butonul “Back”, după cum spune și numele, va schimba perspectiva pe cea anterioară, în cazul figurii 12, pe meniul principal.

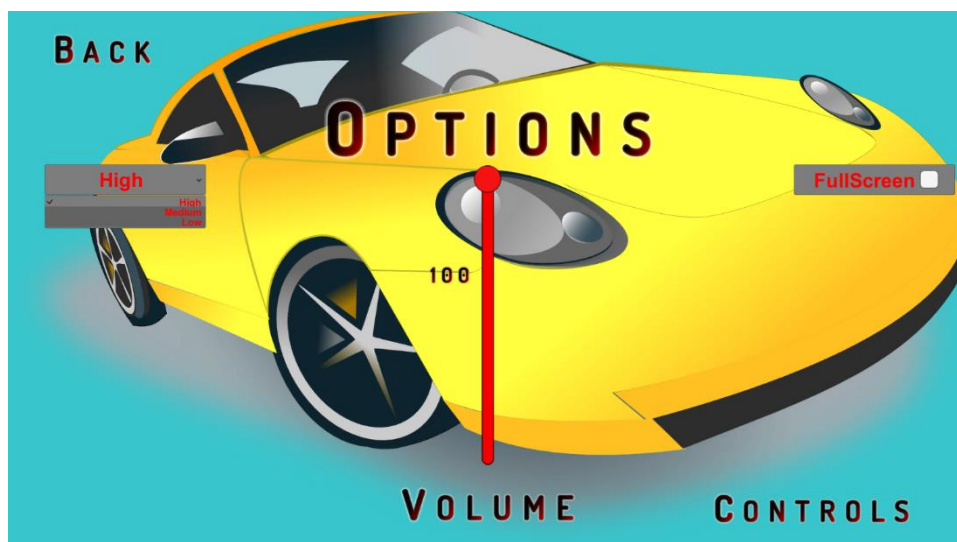


Figura 12 Meniu opțiunilor



Figura 13 Controalele jocului

5.1.3. Garajul

În această scenă, jucătorul va putea alege una dintre cele 10 mașini disponibile și va avea la îndemână 3 butoane importante specifice fiecărei mașini:

- “Buy” pentru a cumpăra mașina(figura 14);
- “Upgrade to max performance” pentru a ridica performanța mașinii la nivelul maxim(figura 15);
- “Select” pentru a selecta mașina curentă și a trece mai departe la selectarea cursei(figura 16).

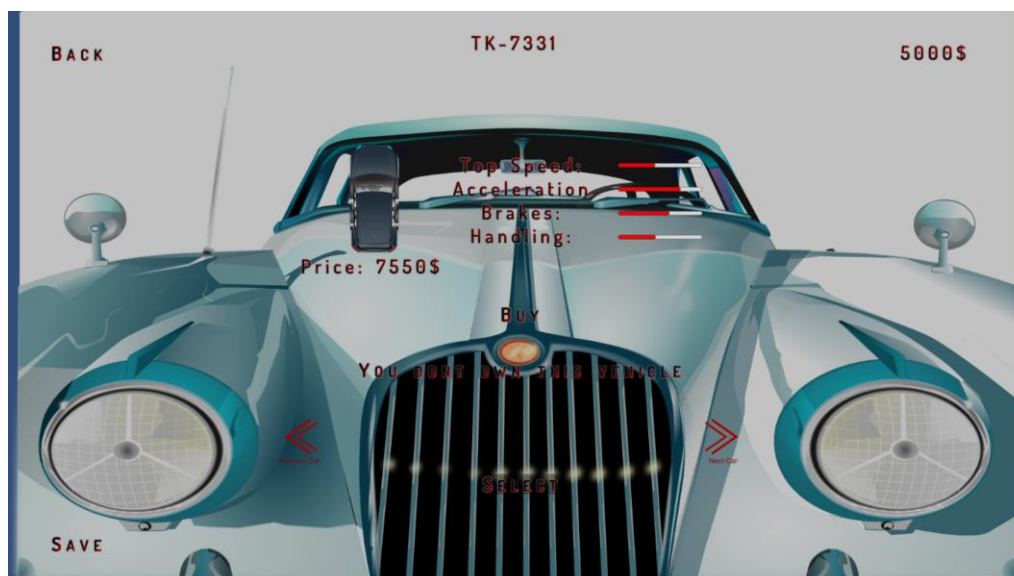


Figura 14 Prezentarea unei mașini neachiziționate

De menționat este faptul că, un jucător nu poate cumpăra/modifica o mașină dacă nu are banii necesari, nu poate modifica o mașină dacă nu o deține și nu poate alege o mașină pentru cursă dacă nu este cumpărată. Totodată, fiecare mașină prezintă și informații despre performanța ei pentru a oferi utilizatorului o privire de ansamblu în alegerea unei mașini corespunzătoare.

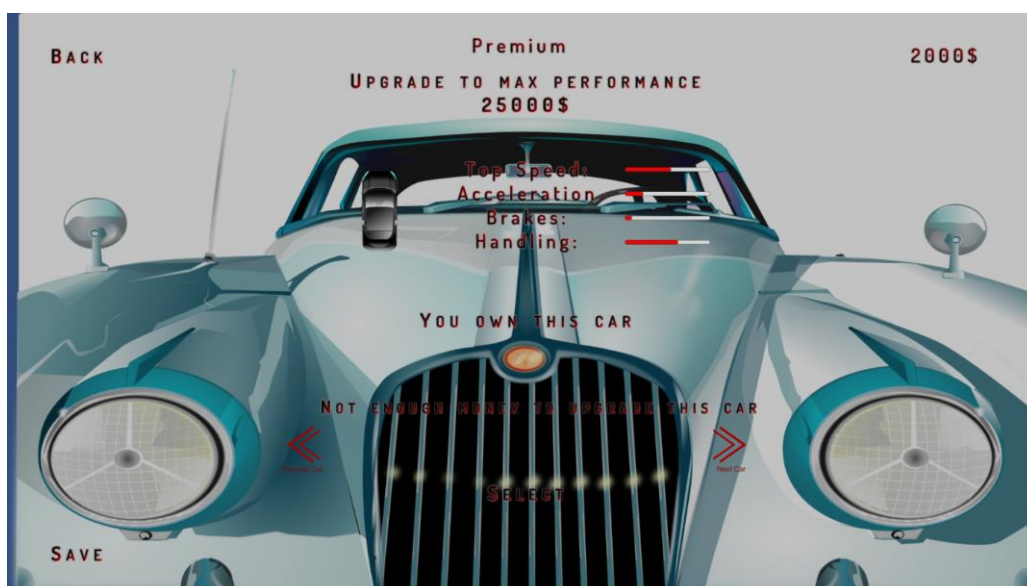


Figura 15 Prezentarea unei mașini achiziționate

Adițional, avem 2 butoane pentru a naviga printre mașini și un buton de Save ce va salva progresul curent într-un fișier separat din directorul aplicației. Salvarea se realizează cu ajutorul scrierii într-un fișier a unor date despre progresul curent din garaj, iar încărcarea ("Continue") cu ajutorul citirii acestor date și modificarea elementelor corespunzătoare.

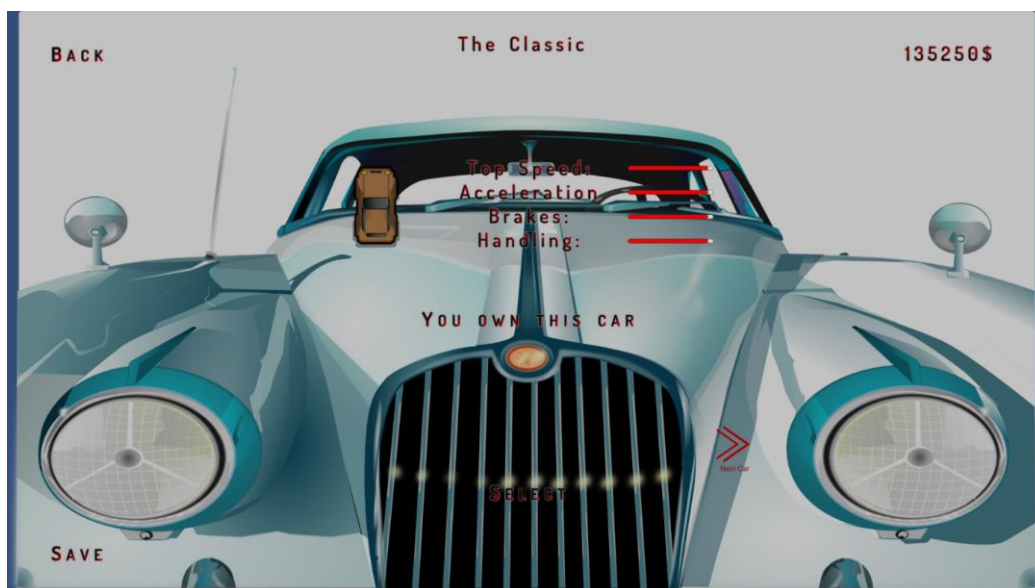


Figura 16 Prezentarea unei mașini achiziționate și modificate

5.1.4. Lista curselor

Asemănător cu scena de mai sus, jucătorul, în funcție de informațiile puse la dispoziție, își poate alege un traseu din cele 5 pe care să concureze (figura 17), iar, dacă deține banii pentru a plăti taxa de intrare, poate apăsa butonul “Play” pentru a porni o nouă scenă cu, cursa propriu-zisă.

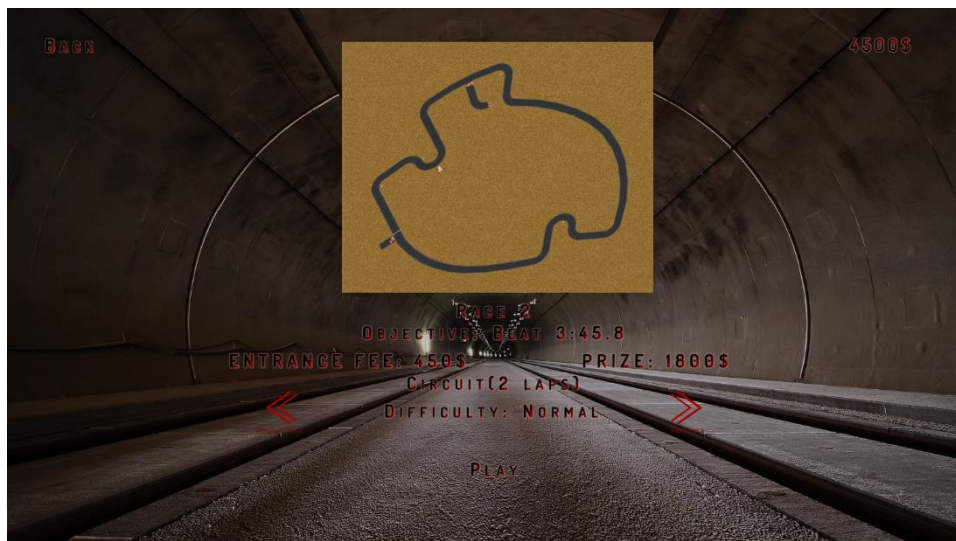


Figura 17 Meniul curselor

5.2. Cursa

Jocul prezintă 2 tipuri de curse:

- cu viraje (circuit sau sprint), cu unul sau mai multe tururi de acoperit, unde jucătorul necesită o anumită dexteritate în a lua corect virajele;

- fără viraje(drag), cursă desfășurată pe o linie dreaptă, unde jucătorul necesită să schimbe cât mai indicat posibil treptele de viteză pentru a obține o accelerație eficientă.

Arborele elementelor unei scene cursă arată astfel(figura 18):

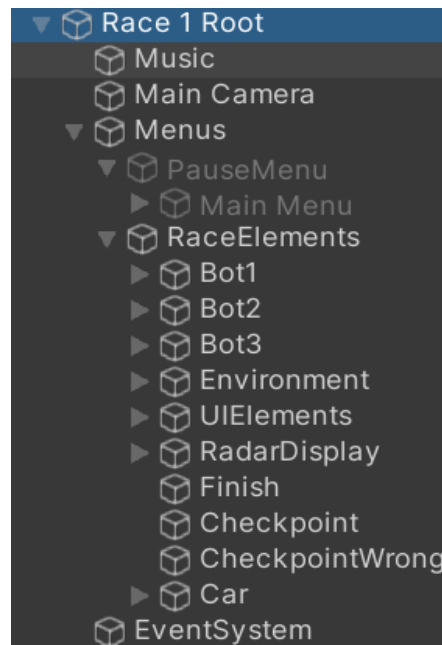


Figura 18 Obiectele dintr-o scenă cursă

5.2.1. Mașina

Mașina sau elementul “Car” reprezintă obiectul din joc la nivelul căruia utilizatorul are control absolut. Obiectul este preluat din scena “Garage”(ce rulează inactiv pe parcursul cursei) unde îi sunt efectuate modificările și este translatat la linia de start.

Obiectul prezintă 4 componente importante:

- Sprite Renderer ce reprezintă partea din față a obiectului, toate transformările aplicându-se “imaginii” vehiculului. Toate sprite-urile folosite în joc se pot vedea în figura 19;
- Rigidbody 2D ce reprezintă partea din spate a obiectului, componenta care oferă fizică mașinii astfel încât impune o masă mașinii, obținându-se frecare cu asfaltul și impuls de viteză pentru mișcare;
- Box Collider 2D ce conturează exteriorul sprite-ului și interzice trecerea prin zone restricționate(în afara circuitului) sau trecerea prin oponenți și verifica trecerea liniei de sosire;
- scriptul obiectului ce folosește cele 4 atribute importante ale mașinii și le pune în aplicare în funcție de preferința utilizatorului.



Figura 19 Sprite-urile folosite pentru mașini

Scriptul mașinii sau ScriptCar prezintă 3 funcții principale impuse de Unity, suprascrise pentru a transforma fizica mașinii într-una cât mai realistă:

- funcția `Start()` aduce toate atributele(exceptând cele 4 importante) folosite pe parcursul drumul la valorile inițiale, cum ar fi, viteza curentă a mașinii, valoarea rezervorului de nitro, dezactivarea mersului cu spatele, treapta de viteză pe neutru, rotațiile pe minut ale mașinii, etc.
- funcția `FixedUpdate()` ce verifică la fiecare frame daca utilizatorul a apăsât tastele corespunzătoare mașinii, dacă mașina alunecă pe asfalt astfel încât să producă efectul de fum daca nitro-ul a fost consumat. Totodată impune rotație și viteză mașinii la fiecare frame. Se poate observa în figura 20 o parte din această funcție;
- funcția `OnTriggerEnter2D(Collider2D collision)` ce verifică trecerea peste linia de sosire sau peste checkpoint pentru a verifica corectitudinea parcurgerii traseului. În figura 21 se poate observa această funcție.

```

if(Input.GetKey(KeyCode.W))
{
    if(carSpeed == 0)
    {
        reverse = false;
        if(reverse)
        {
            if(carSpeed < carBrakeForce)
                carSpeed = 0;
            else
            {
                carSpeed -= carBrakeForce;
                carBody.AddForce(transform.up * carBrakeForce);
            }
        }
        else if(carSpeed <= carTopSpeed)
        {
            carSpeed += carAcceleration;
            carBody.AddForce(transform.up * carAcceleration);
        }
    }
    else if(Input.GetKey(KeyCode.S))
    {
        if(carSpeed == 0)
        {
            reverse = true;
            if(!reverse)
            {
                if(carSpeed < carBrakeForce)
                    carSpeed = 0;
                else
                {
                    carSpeed -= carBrakeForce;
                    carBody.AddForce(transform.up * -carBrakeForce);
                }
            }
        }
        else if(carSpeed <= carTopSpeed * 0.25f)
        {
            carSpeed += carAcceleration;
            carBody.AddForce(transform.up * -carAcceleration);
        }
    }
}

if((Input.GetKey(KeyCode.A) || Input.GetKey(KeyCode.D)) && carSpeed != 0)
{
    if(carSpeed > 100)
    {
        smoke.SetActive(true);
        if(carSpeed < carBrakeForce)
            carSpeed = 0;
        else
        {
            carSpeed -= carBrakeForce;
            if(reverse)
            {
                carBody.AddForce(transform.up * carBrakeForce);
            }
            else
            {
                carBody.AddForce(transform.up * -carBrakeForce);
            }
        }
        if(reverse)
        {
            carRotation.z += Input.GetAxis("Horizontal") * Time.deltaTime * carHandling * (-1);
        }
        else
        {
            carRotation.z += Input.GetAxis("Horizontal") * Time.deltaTime * carHandling;
        }
        this.transform.eulerAngles = carRotation;
    }
    else if(carSpeed != 0)
    {
        smoke.SetActive(false);
        if(carSpeed <= carBrakeForce)
            carBody.velocity = Vector2.zero;
    }
}

```

Figura 20 Secțiuni de cod din funcția Update

```

void OnTriggerEnter2D(Collider2D collision)
{
    if(collision.gameObject.tag == "Finish")
    {
        if(RaceLogic.checkpointTaken && !RaceLogic.wrongDirection)
        {
            RaceLogic.lapsCounter++;
            RaceLogic.checkpointTaken = false;
            RaceLogic.wrongDirection = false;
        }
    }
    if(collision.gameObject.tag == "Checkpoint")
    {
        RaceLogic.checkpointTaken = true;
    }
    if(collision.gameObject.tag == "WrongDirection")
    {
        if(!RaceLogic.checkpointTaken)
        {
            RaceLogic.wrongDirection = true;
        }
    }
}

```

Figura 21 Secțiuni de cod din funcția OnTriggerEnter2D

Mișcarea mașinii nu se realizează prin translația mașinii la fiecare frame pe o poziție diferită ci prin funcția specifică Rigidbody-ului, AddForce, ce impune o forță mașinii, pozitivă sau negativă, pentru a se deplasa, forță oferită de accelerația mașinii sau de puterea de frânare. Ca și “abilitate specială”, mașina prezintă combustibil pe bază de nitro ce oferă o creștere semnificativă a accelerației vehiculului cât timp există combustibil în rezervor, rezervor ce se încarcă automat la câteva secunde de la ultima utilizare.

Pentru cursa diferită, de drag, mașina prezintă puncte forte și puncte slabe. Având în vedere că aceasta se bazează pe schimbarea treptelor de viteză, corectitudinea schimbării nu se aplică în orice mod. Scriptul calculează, în funcție de viteza mașinii, o treaptă de viteză optimă pentru ca aceasta să ruleze la parametrii maximi(figura 22). Astfel, impulsul impus mașinii este modificat de către un factor ce analizează această diferență dintre treptele de viteză și apar dezavantajele folosirii unei trepte neindicate. Cu cât treapta de viteză este mai mare decât cea optimă, cu atât mașina va accelera mai încet. Cu cât treapta de viteză este mai mică iar rotațiile motorului sunt mai mari, cu atât mașina va decelera mai rapid până la viteza maximă atinsă de treapta respectivă.

```

private List<float> minSpeedPerGear = new List<float>();
private List<float> maxSpeedPerGear = new List<float>();
gearNr = 6;
rpm = 0;
nitro = 100;
for(int i = 1; i <= gearNr; i++)
{
    minSpeedPerGear.Add(carTopSpeed * (i-1));
    maxSpeedPerGear.Add(carTopSpeed * i);
}
if(currentGear != 0)
{
    rpm = 100.0f - (carTopSpeed * currentGear - carSpeed);
    for(int i = 1; i <= gearNr; i++)
    {
        if(carSpeed <= maxSpeedPerGear[i-1] && carSpeed >= minSpeedPerGear[i-1])
            optimalGear = i;
    }
}
if(carSpeed < carTopSpeed * currentGear)
{
    int speedFactor = 1;
    if(optimalGear != currentGear)
        speedFactor = currentGear - optimalGear;
    carSpeed += carAcceleration / speedFactor;
    carBody.AddForce(transform.up * carAcceleration / speedFactor);
}
else if(carSpeed - carAcceleration > carTopSpeed * currentGear)
{
    int speedFactor = 1;
    if(optimalGear != currentGear && currentGear != 0)
        speedFactor = optimalGear - currentGear;
    carSpeed -= carAcceleration * 0.5f * speedFactor;
    carBody.AddForce(transform.up * -carAcceleration * 0.5f * speedFactor);
}

```

Figura 22 Secțiune de cod din funcția Update al mașinii de drag

Mașina prezintă 2 obiecte adiționale de tip "Effect", unul gri pentru fumul produs de roți în timpul alunecării pe asfalt(figura 23) și altul albastru pentru eliminarea gazelor de nitro din evacuare(când este nitro-ul activat, figura 24).

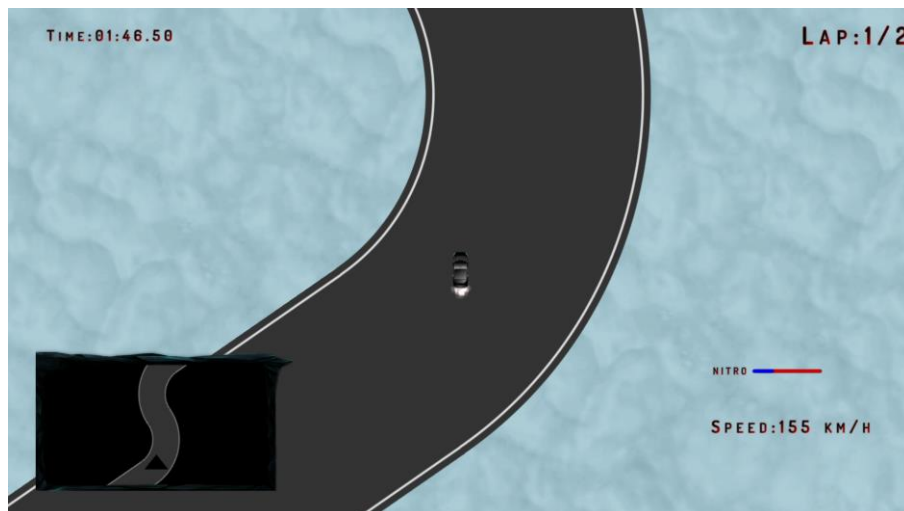


Figura 23 Efectul de fum al caucicurilor

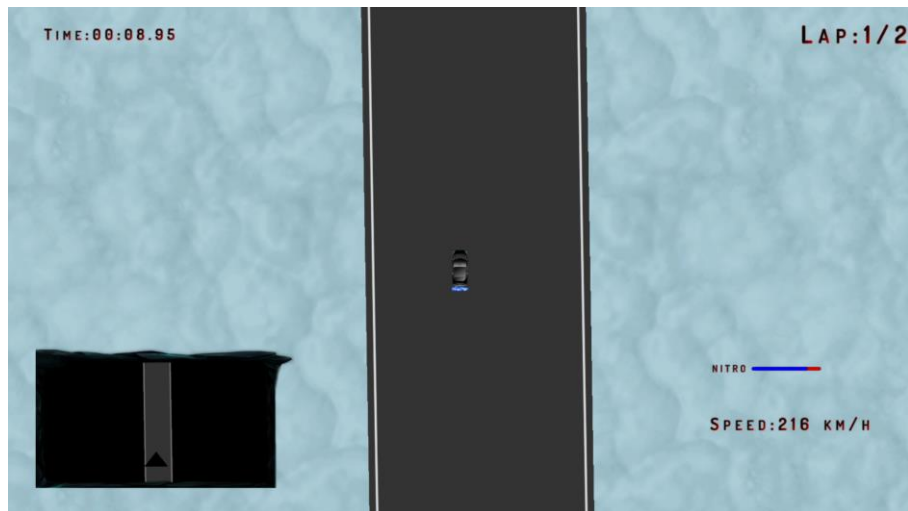


Figura 24 Efectul de nitro

5.2.2. Elementele de UI

Pe parcursul cursei, indiferent de poziția mașinii, pe ecran vor apărea anumite informații specifice interfeței utilizatorului atât despre progresul cursei, cum ar fi timpul total petrecut pe traseu, numărul de ture acoperite pe traseu, direcția de mers(dacă este greșită), cât și despre statusul mașinii, turațiile mașinii, nivelul curent de nitro, treapta de viteză și viteza(figura 25 si 26).

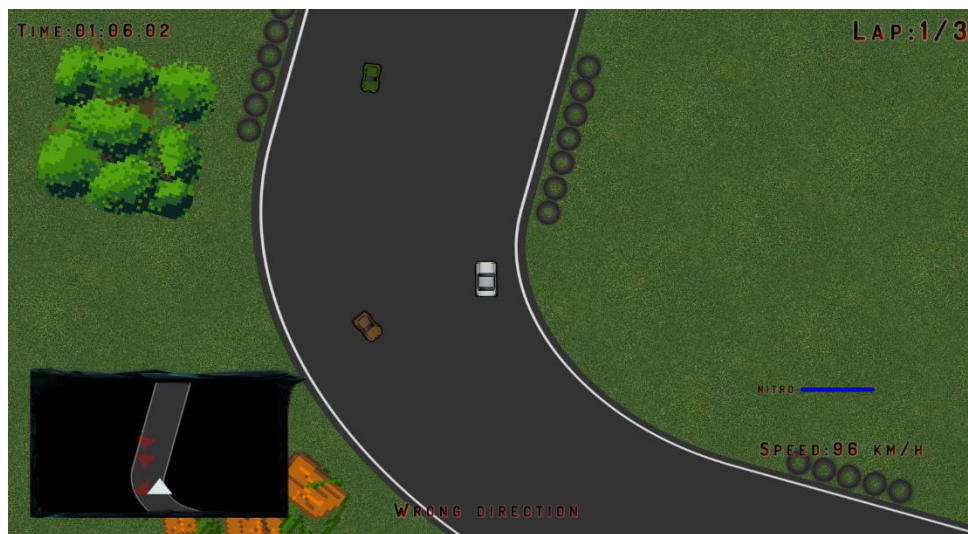


Figura 25 UI-ul din cadrul unei curse de circuit



Figura 26 UI-ul din cadrul unei curse de drag

De asemenea, pentru o vedere mai bună și de la distanță, o cameră secundară, ce contorizează doar mașinile și traseul ce trebuie parcurs, se va regăsi în partea stângă a interfeței, și va reprezenta jucătorul cu o culoare asemănătoare mașinii pe care o conduce, iar concurenții cu o culoare roșiatică.

5.2.3. Meniul de pauză

Un alt element ce aparține de partea de UI este meniul din cadrul cursei ce se activează la apăsarea tastei “Esc”. Cât timp meniul este activat, toate elementele trec într-o stare de pauză cu ajutorul unui script(figura 27). În meniul de pauză, utilizatorul are la dispoziție 3 butoane, “Resume” pentru a reveni la jocul propriu-zis unde a rămas înaintea apăsării tastei, “Abandon” pentru a părăsi cursa și a reveni în scena “Garage”(considerându-se că jucătorul a pierdut cursa, astfel încât pierde taxa de intrare și nu primește nici un premiu) și “Quit”, ca cel din meniul principal, care închide imediat aplicația(figura 28).

```
void Update()
{
    if(Input.GetKeyDown(KeyCode.Escape) && !buttonPressed)
    {
        if(isPaused)
            Resume();
        else
            Pause();
    }
}

public void Resume()
{
    pauseMenu.SetActive(false);
    raceLogic.SetActive(true);
    BotScript.play = true;
    Time.timeScale = 1f;
    isPaused = false;
}

public void Pause()
{
    pauseMenu.SetActive(true);
    raceLogic.SetActive(false);
    Time.timeScale = 0f;
    isPaused = true;
}
```

Figura 27 Secvență de cod ce ține de meniul de pauză



Figura 28 Opțiunile din meniul de pauză

5.2.4. Adversarii

Fiecare traseu prezintă un număr de boți pe post de adversari(figura 29, mașina crem și cele 2 mașini albastre) specifici ce vor parcurge cursa împreună cu jucătorul și îi vor îngreuna cursa având în vedere că obiectele prezintă componenta Box Collider 2D iar coliziunea cu acesta nu v-a face decât să încetinească mașina utilizatorului. Vehiculul bot este translatat pe o poziție anume citită dintr-un fișier pe care a fost înregistrat iar cel mai rapid dintre ei(dacă sunt mai mulți) va ajunge la timpul scris în informațiile fiecărei curse, astfel încât, pentru a câștiga cursa, jucătorul trebuie să termine într-un timp mai scurt, adică să ajungă înaintea tuturor adversarilor.



Figura 29 Adversarii din timpul cursei

5.2.5. Mediul

Toate cele 5 curse se desfășoară în locații aproximativ diferite. Pe lângă stradă, mediul este decorat cu obiecte Sprite Renderer specifice locului în care are loc întrecerea. Totuși, un lucru

ce îl au în comun sunt străzile și cele 3 obiecte de tip “checkpoint”(unde intră și linia de sosire). Rolul lor este de a verifica coliziunea cu mașina jucătorului și de a se asigura că traseul este parcurs în mod corect. Linia de sosire(figura 30) reprezintă primul “checkpoint” și contorizează turele parcurse de jucător(finalul cursei în cazul în care este vorba de ultimul tur). Al doilea verifică faptul că turul nu este parcurs în sens invers. Parcurgerea în sens invers nu reprezintă neapărat un avantaj pe care și-l poate procura dar este folosit pentru frumusețea jocului. Al treilea “checkpoint” este folosit în paralel cu linia de sosire, astfel încat, dacă jucătorul trece linia de sosire de multiple ori, turul nu va fi contorizat dacă nu exista o coliziune cu “checkpoint-ul” numărul 3, evitandu-se crearea avantajului de a termina cursa din câteva mișcări.

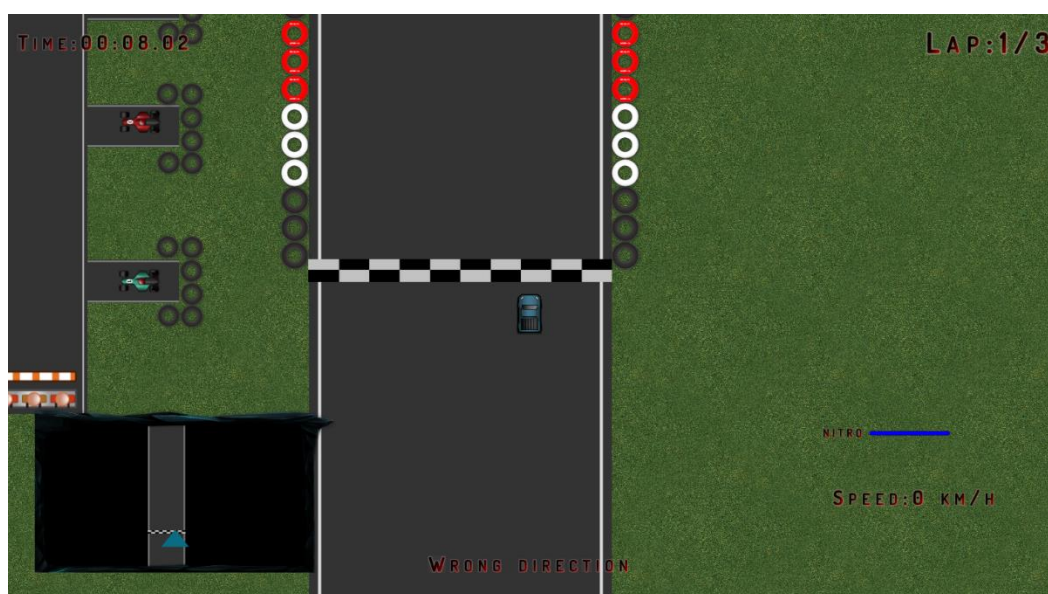


Figura 30 Linia de sosire

5.2.6. Coliziunea exterioară

Pe lângă străzi, un alt lucru comun al celor 5 trasee este coliziunea cu restul mediului înconjurător. Jucătorul este “blocat” pe străzi și nu poate folosi o altă rută în terminarea cursei, astfel se evită crearea avantajelor și se păstrează frumusețea jocului. După cum se poate observa în figura 31, avem o coliziune exterioară(culoarea verde deschis) ce se plimbă pe lângă stradă și impune interzicerea trecerii unui obiect cu RigidBody. Componenta folosită pentru determinarea acestei coliziuni diferă în funcție de circuit, încât să fie cât mai precisă și să nu slăbească performanțele, dar, în principiu, cea mai bună variantă este Polygon Collider 2D.

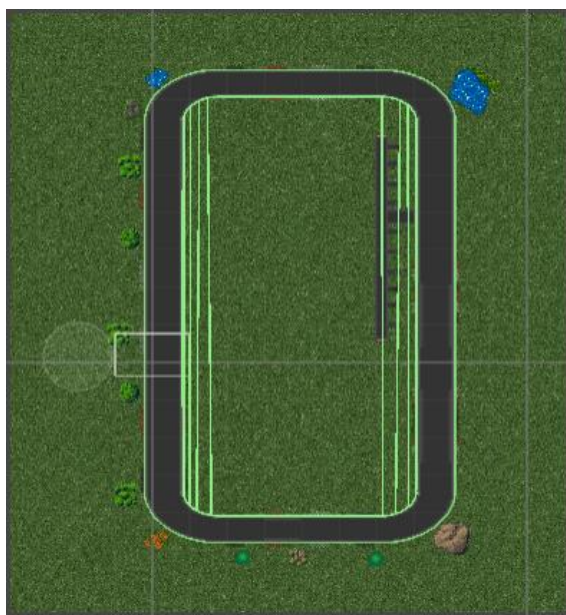


Figura 31 Reprezentarea restricțiilor traseului

6. EVALUAREA REZULTATELOR

Realizând o comparație dintre produsul prezent și majoritatea soluțiilor asemănătoare prezentate la nivelul capitolului 3, proiectăm ideea că jocurile tridimensionale vor utiliza mai multe resurse decât un joc bidimensional, evident, datorită dimensiunii spațiilor de asamblare.

În primul rând, obiectivul proiectului nu este neapărat de a folosi cât mai puține resurse și pe acest principiu să fie considerat mai bun. Ideea principală de urmărit a fost de a obține rezultate la fel de bune cu alte aplicații din aceeași categorie și de oferi clientului aceleași nevoi, în cazul de față fiind să punem la înaintare un joc realist 2D.

Desigur, se pot stabili și anumite criterii legate de performanță și cum se desfășoară jocul în paralel cu alte jocuri din categoria bidimensională, și nu numai, cu, curse de mașini, menționate la capitolul 3.

Testarea jocului a fost efectuată cu ajutorul a mai multor sisteme, diferite, aparținând clienților ce au ajutat dezvoltarea produsului cu feedback și propuneri, conform capitolului 2. Printre acestea, se numără și sisteme mai puțin performante care au reușit să ruleze jocul fără probleme, iar jucătorii au avut parte de un gameplay satisfăcător. Sistemele slabe ale acestora au contribuit la construirea unei recomandări de sistem pentru rularea aplicației, pe care am adăugat-o în tabelul de mai jos.

Astfel, avem tabelul 1 de mai jos cu câteva cerințe importante ale sistemului pentru o rulare decentă ale jocurilor, statistici obținute folosind aplicația Steam:

Tabel 1 Cerințele recomandate

Joc / Componentă	Procesor	Placă video	Memorie RAM	Memorie internă
Racing For Real 2D	1.2Ghz	512 MB video memory	1 GB RAM	250 MB
Ultimate Racing 2D	1.8Ghz	512 MB video memory	2 GB RAM	700 MB
NFS Carbon	Pentium 4	1 GB video memory	1 GB RAM	7 GB
Forza Horizon 4	Intel i7-3820 @ 3.6Ghz	NVidia GTX 970	12 GB RAM	100 GB

Tabelul este ordonat, oarecum, după performanțele componentelor sistemului.

Se observă, destul de evident, diferența dintre un joc 2D și unul 3D. Chiar dacă titlul “NFS Carbon” este cel mai vechi joc din această listă, cerințele lui sunt puțin peste jocurile 2D, mai ales în cazul memoriei interne folosite. Jocurile 2D, adică proiectul de față împreună cu “Ultimate Racing 2D”, au aproximativ aceleași cerințe, extrem de mici față de nivelul la care poate ajunge un calculator normal în prezent, un calculator normal fiind numit un sistem care se apropie de cerințele jocului “Forza Horizon 4”, adică cu performanțe destul de ridicate.

Pe de altă parte, vorbind strict de rularea jocului și de feedback-ul primit pe varianta finală a jocului, totul pare foarte promițător. Jocul merge și efectuează întocmai cerințele primite din partea dornicilor clienți și astfel a primit, în mod neprofesionist, calificative destul de ridicate.

În tabelul următor, tabelul 2, am adăugat câteva medii obținute din review-urile oferite de, atât utilizatorii proiectului de față, cât și de publicul larg, prin intermediul aceleiași platforme de jocuri, Steam.

Tabel 2 Calificativele oferite jocurilor

Joc	Notă
Racing For Real 2D	8,4
Ultimate Racing 2D	9
NFS Carbon	7,8
Forza Horizon 4	8,9

Chiar dacă review-urile nu sunt într-un număr la fel de mare ca celelalte 3 jocuri (aproximativ 15 review-uri), "Racing For Real 2D" a fost strict evaluat de clienți cu o notă destul de favorabilă având în vedere că toate celelalte jocuri au și aspecte neplăcute ce au micșorat nota maximă posibilă.

7. CONCLUZII

După analiza efectuată mai sus, am obținut un rezultat, după parerea mea, dar și a primilor utilizatori, mai mult decât pozitiv. Dezvoltarea aplicației a atins mai multe obiective. Pentru atingerea problemei discutate la nivelul capitolului 1, procesul bazat pe feedback și sondaj, analizat în capitolul 2, a dat roade asupra creării jocului. Toate tehnologiile folosite nu au făcut decât să simplifice tot procesul de dezvoltare al aplicației astfel încât absolut toate ideile propuse și primite s-au îndreptat către succes. În momentul de față, jocul reprezintă un progres enorm de la prototipul ce inițial a avut doar scopul de a fi un exemplu de implementare și testare, și are toate calitățile din lista de obiective propuse pentru a-l plasa într-o categorie interesantă de jocuri.

Cât despre o dezvoltare ulterioară, în ritmul excelent de evoluție al proiectului, se poate produce un efect de iterație, în care se cer propuneri de la clienți, și, cu ajutorul feedback-ului, se poate continua producerea a cât mai multor funcționalități la nivelul jocului. Totodată, proiectul poate inspira redirectionarea pe alte ramuri, în aceeași manieră de lucru(dezvoltarea altor jocuri, în Unity sau în alt motor, folosindu-se de această sursă de inspirație), sau poate chiar transpunerea proiectului curent într-un joc 3D pentru a destrăma limitele pe care spațiul bidimensional nu le poate atinge.

8. BIBLIOGRAFIE

[1] Diferențele dintre jocurile 2D și cele 3D. Valabil la adresa:

<https://www.makeuseof.com/2d-games-vs-3d-games-differences/>

[Accesat în 8 09 2022]

[2] Evoluția jocurilor de curse. Valabil la adresa:

<https://antidote.gg/evolution-of-racing-games/>

[Accesat în 9 09 2022]

[3] Despre jocul „Pole Position”. Valabil la adresa:

https://www.retrogamer.net/retro_games80/pole-position-2/

[Accesat în 9 09 2022]

[4] Despre filmul intitulat „Need For Speed”. Valabil la adresa:

<https://www.motortrend.com/features/on-the-set-of-need-for-speed-movie-for-a-day/>

[Accesat în 11 09 2022]

[5] Articol făcut asupra jocului „The Crew 2”. Valabil la adresa:

<https://www.pcgamer.com/the-crew-2-review/>

[Accesat în 11 09 2022]

[6] O părere despre jocul „Forza Horizon 4”. Valabil la adresa:

<https://www.stuff.tv/review/forza-horizon-4-review/>

[Accesat în 11 09 2022]

[7] Informații despre jocul „Ultimate Racing 2D”. Valabil la adresa:

<https://www.ultimateracing2d.com/>

[Accesat în 12 09 2022]

[8] Informații despre motorul de dezvoltare Unity. Valabil la adresa:

<https://gamedevacademy.org/what-is-unity/>

[Accesat în 21 09 2022]

[9] Despre componenta de scripting de la nivelul Unity. Valabil la adresa:

<https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>

[Accesat în 21 09 2022]

[10] Despre Unreal Engine. Valabil la adresa:

<https://conceptartempire.com/what-is-unreal-engine/>

[Accesat în 22 09 2022]

[11] De ce GameMaker este un motor bun pentru jocurile 2D. Valabil la adresa:

<https://www.gamesindustry.biz/what-is-the-best-game-engine-is-gamemaker-the-right-game-engine-for-you>

[Accesat în 22 09 2022]

[12] Articol despre Python și PyGame. Valabil la adresa:

<https://www.stxnext.com/blog/python-for-game-development/>

[Accesat în 29 09 2022]

[13] Informații despre OpenGL. Valabil la adresa:

<https://www.easytechjunkie.com/what-is-opengl.html>

[Accesat în 22 09 2022]

[14] Sursa imaginii de comparație între Forza Horizon 4 și realitatea. Valabilă la adresa:

<https://imgur.com/gallery/M5mDXfn/comment/1511859185>

[Accesat în 11 09 2022]