

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач випускової кафедри  
\_\_\_\_\_ Аліна САВЧЕНКО  
«\_\_\_» \_\_\_\_\_ 2024 р.

# **КВАЛІФІКАЦІЙНА РОБОТА**

**(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

**ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”**

**Тема:** «Відеогра у жанрі перегонів в середовищі Unity з елементами мультіплеєру»

**Виконавець:** Денис МИРОШНИКОВ

**Керівник:** к.т.н., доцент Олена ТОЛСТИКОВА

**Нормоконтролер:** к.т.н., доцент Вікторія СИДОРЕНКО

Київ 2024

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра Комп'ютерних інформаційних технологій

Спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ:

завідувач кафедри КІТ

\_\_\_\_\_Аліна САВЧЕНКО  
«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

## ЗАВДАННЯ

### на виконання кваліфікаційної роботи

*Мирошникова Дениса Олексійовича*

(прізвище, ім'я, по батькові здобувача вищої освіти в родовому відмінку)

1. Тема кваліфікаційної роботи: «Відеогра у жанрі перегонів в середовищі Unity з елементами мультиплеєру» затверджена наказом ректора від «05» квітня 2024р. № 517/ст
2. Термін виконання роботи: з 6 травня 2024 року по 16 червня 2024 року.
3. Вихідні дані до роботи: Проект на основі ігрового движка Unity, який представляє собою комп'ютерну гоночну гру на двох реалізовану мовою C#
4. Зміст пояснювальної записки: 1) Середовище та механіка розробки відеоігор. 2) Технології та реалізація відеогри. 3) Проектування та тестування відеогри.
5. Перелік обов'язкового ілюстративного матеріалу: 1) Перша комп'ютерна гра «Теніс на Двох». 2) Види мультиплеєрних моделей. 3) Аналіз ринку комп'ютерної індустрії. 4) Онлайн мульти-конференція одного з університетів у SecondLife. 5) Приклади ігрових об'єктів в роботі.

## 6. Календарний план-графік

№ з/п	Завдання	Термін виконання	Підпис керівника
1.	Складання плану, планування процесу написання дипломної, представлення керівнику.	06.05.2024- 08.05.2024	
2.	Закінчення плану, початок реалізації проекту, представлення керівнику.	09.05.2024- 19.05.2024	
3.	Написання першого та другого розділів, завершення розробки проекту, представлення керівнику.	20.05.2024- 22.05.2024	
4.	Завершення написання всіх розділів, аналіз проекту на недоліки, представлення керівнику.	23.05.2024- 25.05.2024	
5.	Проходження нормоконтролю, перепліт пояснювальної записки.	26.05.2024- 31.06.2024	
6.	Оформлення презентації, підготовка тексту, необхідної інформації до доповіді	01.06.2024- 07.06.2024	

7. Дата видачі завдання: «06» травня 2024р.

Керівник кваліфікаційної роботи

Олена ТОЛСТІКОВА

\_\_\_\_\_  
(підпис керівника)

Завдання прийняв до виконання

Денис МИРОШНИКОВ

\_\_\_\_\_  
(підпис випускника)

## РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи на тему: «Відеогра у жанрі перегонів в середовищі Unity з елементами мультиплеєру» містить: 55 с., 43 рис., 20 літературних джерел.

**Об'єкт дослідження** – розробка відеогри.

**Предмети дослідження** – фізика руху, дизайн навколишньої середовища, інтерфейс користувача та додавання об'єктів.

**Мета кваліфікаційної роботи** – вдосконалення та демонстрація здобутих навичок у сфері розробки комп'ютерних ігор та застосування сучасних технологій та інструментів розробки платформи Unity.

**Методи дослідження** – розгляд оптимальних рішень для розробки цілісної ігрової моделі, освоєння платформи Unity для: розробки сцени, програмування об'єктів та реалізації інтерфейсу користувача.

**Результати** кваліфікаційної роботи можна буде використовувати для покращення концентрації, реакції та розважання з друзями. Для розробки відеогри було використано мову програмування C# для написання різних скриптів для камер, сцен та меню. Платформу Unity для розміщення та додавання фізики і границь об'єктам, розробки основної механіки поведінки гравців, середовища та траси.

**Наукова новизна** була досягнута шляхом об'єднання декілька основних концепцій відеоігри в одну: застосування SplitScreen для можливості грати двом гравцям з одного пристрою, механіки заносу та дрифту в поворотах і захоплюючих перешкод на трасі.

**Ключові слова:** МУЛЬТИПЛЕЄР, VR, AR, UNITY, P2P, ММО, АРХІТЕКТУРА, МЕХАНІКА, КОЛАЙДЕР, МЕРЕЖА, ОБ'ЄКТ, GAME, РИНОК, МЕТОД, ПОТЕНЦІАЛ

# ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ, ПОЗНАЧЕНЬ, ТЕРМІНІВ	6
ВСТУП	8
РОЗДІЛ 1. СЕРЕДОВИЩЕ ТА МЕХАНІКА РОЗРОБКИ ВІДЕОІГОР	10
1.1. Історія виникнення відеоігор та мультиплеєра	10
1.2. Значення ігор у сучасному світі	14
1.3. Середовище розробки Unity	16
1.3.1. Про середовище розробки	16
1.3.2. Архітектура Unity	18
1.4. Механіки розробки ігор	20
ВИСНОВКИ ДО РОЗДІЛУ 1	22
РОЗДІЛ 2. ТЕХНОЛОГІЇ ТА РЕАЛІЗАЦІЯ ВІДЕОГРИ	23
2.1. Компоненти розробки відеогри на Unity	23
2.2. Етапи створення відеогри	26
2.2.1. Створення сцен	26
2.2.2. Створення гравців	31
2.2.3. Створення чекпоінтів для старту та фінішу	34
ВИСНОВКИ ДО РОЗДІЛУ 2	36
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ ВІДЕОГРИ	37
3.1. Розробка концепції	37
3.2. Створення сцен	37
3.3. Створення та обробка об'єктів	43
3.4. Геймплей	46
3.5. Тестування гри	49
ВИСНОВКИ ДО РОЗДІЛУ 3	51
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ, ПОЗНАЧЕНЬ, ТЕРМІНІВ

- PDP-1 – Programmed Data Processor-1— перший комп'ютер із серії PDP, випущений у 1960му році
- MIT – Massachusetts Institute of Technology
- EDSAC – Electronic Delay Storage Automatic Calculator це електронна обчислювальна машина, створена 1949 в Кембридзькому університеті
- VR – Virtual Reality -Віртуальна реальність
- AR – Augmented Reality – Доповнена реальність
- LAN – Local Area Network - Локальна мережа клієнт/сервер
- PBR – Physically Based Rendering - це підхід до рендерингу комп'ютерної графіки
- ICMP – Internet Control Message Protocol - мережевий протокол, що входить у стек протоколів TCP/IP
- DSL – Digital Subscriber Line - це технологія, яка використовується для передачі цифрових даних через телефонні лінії
- QA – Quality Assurance - це процес забезпечення якості продукту чи послуги,
- NPC – Non-Player Character - це персонаж у відеоіграх, який контролюється комп'ютером, а не гравцем.
- P2E – P2E Play-to-Earn - це модель у відеоіграх, яка дозволяє гравцям заробляти реальні гроші

- P2P – Peer-to-Peer - це децентралізована мережна архітектура, де кожен учасник одночасно виступає і клієнтом, і сервером
- GPU – Graphics Processing Unit - це спеціалізований процесор
- TCP – Transmission Control Protocol - це один із основних протоколів Інтернету,
- IP – Internet Protocol - це основний мережевий протокол
- UDP – User Datagram Protocol - це один із основних протоколів Інтернету
- RTS – Real-Time Strategy - це жанр відеоігор, у яких гравці керують ресурсами у реальному часі
- DLC – Downloadable Content - це додатковий контент для відеоігор
- MMO – Massively Multiplayer Online - це жанр відеоігор, в яких велика кількість гравців одночасно взаємодіють у спільному віртуальному світі
- VAC – Valve Anti-Cheat - це система боротьби з шахрайством

## ВСТУП

В сучасному світі ігри стають не лише розважальним засобом, але й важливим елементом різних сфер життя, включаючи освіту, бізнес, медицину та соціальні відносини.

За останні десятиліття геймінг став не лише розвагою для молоді, але й потужним інструментом для розвитку критичного мислення, комунікаційних навичок та творчого підходу. У цьому контексті, практика з використання ігор набуває великого значення. Вона дозволяє вивчати та розуміти потенціал ігор для різних цілей, таких як навчання, тренінг, командобудування та інші.

Перша мультиплеєрна гра, що набула популярності, - це "Spacewar!". Вона була розроблена в 1962 році для PDP-1 (Programmed Data Processor-1) – одного з перших комп'ютерів цього класу. Гра стала відомою своєю інноваційною геймплейною концепцією – два кораблі заборонено керувати в атмосфері зір, а їх завданням було взаємодіяти один з одним та уникати зір та тяжіння планети.

З того часу мультиплеєрні ігри пройшли великий шлях, розвиваючи нові технології та відкриваючи нові можливості для геймерів у всьому світі. З'явилися різноманітні жанри та підходи до мультиплеєрного геймплею, включаючи онлайн-гри, кооперативні режими та багатокористувацькі онлайн-гри з масовою участю.

**Актуальність** теми кваліфікаційної роботи «Відеогра у жанрі перегонів в середовищі Unity з елементами мультиплеєру» ґрунтується на тому, що відеоігри у сучасності стали найпопулярнішим видом розваг, ігрові індустрії постійно розширюються, кількість гравців постійно зростає, разом з цим ростуть і прибутки з продажу та розробки ігор. Особливо популярними стали ігри з мультиплеєрним елементом, який дозволяє гравцям взаємодіяти один з одним і надають новий геймплей. Дана кваліфікаційна робота відповідає актуальним



тенденціям розвитку ігрової індустрії та відкриває можливості для подальшого дослідження та розвитку в області розробки комп'ютерних ігор.

**Об'єкт дослідження** – розробка відеогри.

**Предмети дослідження** – фізика руху, дизайн навколишньої середовища, інтерфейс користувача та додавання об'єктів.

**Мета кваліфікаційної роботи** – вдосконалення та демонстрація здобутих навичок у сфері розробки комп'ютерних ігор та застосування сучасних технологій та інструментів розробки платформи Unity.

**Методи дослідження** – розгляд оптимальних рішень для розробки цілісної ігрової моделі, освоєння платформи Unity для: розробки сцени, програмування об'єктів та реалізації інтерфейсу користувача.

**Результати** кваліфікаційної роботи можна буде використовувати для покращення концентрації, реакції та розважання з друзями. Для розробки відеогри було використано мову програмування C# для написання різних скриптів для камер, сцен та меню. Платформу Unity для розміщення та додавання фізики і границь об'єктам, розробки основної механіки поведінки гравців, середовища та траси.

**Наукова новизна** була досягнута шляхом об'єднання декілька основних концепцій відеоігри в одну: застосування SplitScreen для можливості грати двом гравцям з одного пристрою, механіки заносу та дрейфу в поворотах і захоплюючих перешкод на трасі.

**Практичне значення отриманих результатів**

Результати кваліфікаційної роботи можуть бути корисні як для освітніх завдань, так і для практичного втілення нових ігрових проектів у майбутньому, сприяючи подальшому розвитку навичок розробки.

## РОЗДІЛ 1

### СЕРЕДОВИЩЕ ТА МЕХАНІКА РОЗРОБКИ ВІДЕОГРИ

#### 1.1. Історія виникнення ігор та мультиплеєра

Перші відеоігри були розроблені в кінці 1940-х і на початку 1950-х років. Вони були системно-специфічними, це означало, що їх можна було відтворювати лише на комп'ютері, на якому вони були створені, і вони були дуже простими симуляціями, такими як хрестики-нулики та подібні проекти. Це зайняло близько десяти років для відеоігор, які можна було грати на інших комп'ютерах.

Однією з таких ігор була Spacwar. Це космічна бойова відеогра, розроблена для мейнфреймів PDP-1 в MIT. У грі були два космічні кораблі, кожен з яких контролювався гравцем, який брав участь у повітряному бою. Коли студенти MIT покинули школу, вони взяли з собою вихідний код для гри. Це стала однією з перших ігор, яка отримала широке поширення.

«Теніс на двох» був першим?

Тенісу для двох Хігінботама насправді передувало кілька інших винаходів — один наприкінці 1940-х і два на початку 1950-х. Але було б несправедливо чи правильно присуджувати титул першої відеоігри якомусь із цих конкретних винаходів.

У 1948 році, за десять років до «Тенісу для двох» Хігінботама, Томас Т. Голдсміт-молодший і Естл Р. Манн запатентували так званий розважальний пристрій з електронно-променевою трубкою, зробивши цю наразі найдавнішу задокументовану попередницю відеоігри.

Кафедра КІТ				НАУ 24 27 30 000 ПЗ			
	ПІБ			СЕРЕДОВИЩЕ ТА МЕХАНІКА РОЗРОБКИ ВІДЕОІГОР	Літ.	Аркуш	Аркушів
Розроб.	Мирошников Д.О.					10	55
Керівник	Толстікова О.В.				ТП-415Б - 122		
Н-контроль	Сидоренко В.М.						

Однак розважальний пристрій вимагав від гравців накладати зображення або ілюстрації цілей, таких як літаки, перед екраном, узгоджуючи дію гри. Це було не так, як «Теніс на двох» Хігінботама (Рис. 1.1.), який повністю відображав візуальні ефекти гри на екрані [1].



Рис. 1.1. «Теніс на двох»

Що до комерційних ігор, то ці ігри дають уявлення про те, як наукові дослідження та дослідження можуть інформувати майбутнє відеоігор, як для освіти, так і для розваг, а також про те, як історичні методології досліджень можуть інформувати процес ітераційного проектування. Вони пропонують приклади того, як можна побудувати наукову гру, поєднуючи традиційні історичні методології з ігровим дизайном.

Ці ігри показують можливість побудови віртуального середовища, механіки та розповіді з історичного матеріалу. Крім того, існує схожість між ітеративним процесом проектування, що використовується для розробки комп'ютерних та відеоігор, та процесом дослідження історичної науки. Незважаючи на абсолютно різні результати, процес дослідження та написання історичної науки є ітеративним за своєю природою.

Після різького напливу користувачів, ринок комп'ютерних ігор зазнав неймовірної популярності. У той час як домашні консолі отримали популярність, персональні комп'ютери також зіграли значну роль у зростанні ігрової індустрії. Комп'ютерні ігри пропонують інший досвід, з більш складними і стратегічними іграми, які сподобалися іншій аудиторії.

Історія та еволюція відеоігор відображають людську винахідливість, творчість та прагнення до захоплюючих розваг. Оскільки технології продовжують розвиватися, ми можемо лише передбачити ще більш захоплюючий та інноваційний ігровий досвід у майбутньому.

Що до появи мультиплеєру та багатокористувацьких онлайн ігор, то 1990-ті роки ознаменували початок багатокористувацьких онлайн-ігор. Оскільки Інтернет став доступнішим, такі ігри, як «Doom» (1993), запровадили мережеву гру, що дозволило гравцям змагатися через з'єднання LAN.

Лише коли Інтернет став швидким і дешевим, мультиплеєр онлайн став «звичайним». Ми пройшли довгий шлях від необхідності підключатися до інтернету, і все продовжуватиме змінюватися, як і ми. Будемо сподіватися на краще!

Мультиплеєрні ігри, або ігри зі спільним режимом гри для кількох гравців, мають багато користі для гравців. Вони сприяють реакції, координації рухів, стратегічному мисленню, швидкому прийняттю рішень та спритності. Також вони дозволяють покращити свої комунікаційні навички, так як спілкування з іншими гравцями є важливою частиною гри. Мультиплеєрні ігри можуть бути реалізовані у різних форматах, а саме:

- **Кооперативні ігри** – ігри у яких гравці співпрацюють, об'єднують свої зусилля та ресурси для досягнення спільних цілей. Вони можуть виконувати завдання разом, взаємодіяти у боротьбі зі спільними ворогами або розв'язувати головоломки, використовуючи командну співпрацю.

- **Змагальні ігри** – у яких гравці змагаються один з одним за перемогу. Це можуть бути командні або індивідуальні змагання в різних форматах, наприклад, шутери, файтинги, спортивні симулятори тощо.
- **ММО** - цей піджанр мультиплеєрних ігор передбачає велику кількість гравців, які можуть одночасно грати в одній віртуальній грі. ММО-ігри часто включають великі віртуальні світи з розвинутою соціальною інтеракцією, економікою, завданнями та PvP (гравець проти гравця) битвами..
- **Головоломки** - в цих іграх головною метою є розв'язування різноманітних головоломок, логічних завдань та загадок. Головоломки можуть бути різних типів і складності, вимагати розумового напруження, кмітливості та творчого мислення гравця.
- **Інді-ігри** - це незалежно розроблені відеоігри, створені невеликими студіями або навіть окремими розробниками без фінансової підтримки великих видавництв. Цей жанр відомий своєю творчістю, нестандартністю та оригінальністю.
- **VR-ігри** - жанр віртуальної реальності відкриває нові можливості для ігрової індустрії та дозволяє гравцям зануритися в цілком інший віртуальний світ. VR-ігри створюються з використанням спеціального обладнання, такого як віртуальні навушники, контролери руху та інші пристрої, які забезпечують іммерсивний геймплей.

Проблеми включають втрату пакетів і придушення, які можуть перешкодити гравцеві зареєструвати свої дії на сервері. У шутерах від першої особи ця проблема з'являється, коли кулі потрапляють у ворога без пошкоджень. Зв'язок гравця - не єдиний фактор; деякі сервери повільніші за інші [2].

## 1.2. Значення ігор у сучасному світі

Відеоігри — доволі популярний вид дозвілля серед дітей та дорослих.

У чому же користь комп'ютерних ігор?

- Комп'ютерні та консольні ігри розвивають дрібну моторику.
- Покращують реакцію за допомогою спецефектів та скрімерів, екшн-ігор.
- Тренується уважність, яка потрібна для кращого запам'ятовування сюжетних деталей, а також суперницьких ігор.
- Критичне мислення та аналіз. Ці якості розвиваються внаслідок проходження рольових, стратегічних та пригодницьких жанрів.
- Вивчення нових мов. У процесі спілкування між собою геймери можуть навчитись іноземних мов просто під час гри.
- Фізичні якості. Цьому сприяють ігри, які потребують використання контролерів руху для танців, віртуального гольфу тощо.
- Можливість заробляти шляхом летсплеїв та стрімів.
- Віртуальні ігри – як найбільш потенціальний та інноваційний вид ігор.

Віртуальні світи , так звані Virtual Worlds – це ігри більше не моделюють життя, вони самі стають життям .Такий вислів чудово підходить для у ActiveWorlds, SecondLife, OpenSim, Blue Mars і більшість інших віртуальних світів.

Так, наприклад, обсяг економіки становить \$567 млн., або 25 % усього ринку віртуальних товарів у США [3].

За офіційними даними, свої острови (3д локації) у SecondLife мають 53 університети, включаючи Standford, MIT, Harvard, Cambridge, Illinois, Cornell university, Princeton, California Institute of Technology, Drexel University і ряд інших провідних університетів світу.

[15] На думку нейропсихолога Дмитра Вакуленка щодо впливу відеоігор на психіку та розвиток дітей та дорослих він каже: “На моторику, критичне

мислення та аналіз комп'ютерні ігри впливають позитивно. Все залежить від самої гри. Якщо ми говоримо про ігри, в яких немає хоррор-контенту, вони розвивають координацію, моторику рук та стратегічне мислення.”

Але також є і негативні фактори такі як:

- Здоров'я як фізичне, так і психологічне
- Витрати грошей такі як мікротранзакції та додатковий контент
- Насилля через ігри з жорстоким концептом
- Соціальний вплив як на навчання, так й ізоляцію

Важливо розуміти, що багато з цих ризиків можуть бути зменшені шляхом розсудливого використання ігор, обмежень годин гри, а також свідомого вибору ігор з урахуванням їх вікового рейтингу та вмісту.

Ще один серйозний вплив ігор на гравців - ігрова залежність. Досить легко зрозуміти причину цієї залежності, оскільки ігри забезпечують відчуття комфорту, свободи, соціального контакту та досягнення для гравців, яких вони можуть не мати в реальному світі. Ця залежність настільки проблематична, що впливає на повсякденну діяльність геймера і викликає серйозну непродуктивність і неухважність в їх житті. Надмірний вплив цих онлайн-ігор призвів до того, що деякі зрілі дорослі геймери розлучаються, звільняються з роботи, і в рідкісному випадку хтось помирає внаслідок відсутності сну або правильного харчування. [4].

Відеоігри перетворилися на культурний феномен, який зачіпає різні аспекти суспільства. Вони пропонують розваги, освіту, соціальну взаємодію та технологічні інновації. Оскільки промисловість продовжує зростати і розвиватися, важливо визнати як їх позитивний вплив, так і проблеми, які вони представляють. Відеоігри - це більше, ніж просто розваги, вони є відображенням нашого мінливого світу і рушійною силою.

## 1.3. Середовище розробки Unity

### 1.3.1. Про середовище розробки

Unity 3D (Рис. 1.2.) являє собою потужну платформу для розробки, яка дозволяє творцям створювати широкий спектр додатків, включаючи 2D та 3D ігри, інтерактивні симуляції та віртуальну реальність VR та досвід доповненої реальності AR. Він широко популярний серед розробників завдяки своїй гнучкості, комплексному набору інструментів та кросплатформним можливостям.



Рис. 1.2. Unity 3D

В основі Unity 3D лежить Unity Editor, основний інтерфейс, де ви можете створювати і керувати своїми проектами. У редакторі ви можете проектувати сцени, впорядковувати об'єкти, застосовувати матеріали та текстури, налаштовувати властивості фізики та створювати анімації. Редактор інтуїтивно



зрозумілий, забезпечуючи як новачкам, так і досвідченим розробникам зручне середовище для втілення своїх ідей в життя.

Основні будівельні блоки в Unity 3D називаються GameObjects. Це може бути що завгодно, від простих форм, таких як куби і сфери, до складних моделей, таких як символи або транспортні засоби. Ви можете додати різні компоненти до GameObjects, щоб надати їм специфічні властивості або поведінку, такі як взаємодія з фізикою, можливості рендерингу або звукові ефекти. Цей модульний підхід дозволяє легко налаштувати і розширити функціональність ваших об'єктів.

Скриптинг є важливим аспектом Unity 3D, що дозволяє реалізувати ігрову логіку та інтерактивну поведінку. Скрипти зазвичай написані на C# і прикріплені до GameObjects як компоненти.

Фізичний двигун в Unity 3D імітує фізику реального світу, змушуючи об'єкти рухатися і взаємодіяти реалістично. Це включає виявлення зіткнень, динаміку ригідного тіла та складні суглоби та обмеження. Якщо ви створюєте просту гру-головоломку або складне моделювання, фізичний рушій Unity допомагає зробити взаємодію природною та захоплюючою. Анімація в Unity 3D - ще одна потужна функція.

Unity3D - найкраща платформа для створення 3D-додатків. При використанні технології для створення не ігрового додатка можна розглянути наступні напрямки такі як:

- Додатки для навчання
- Головна Дизайн програми
- Маркетингові додатки
- Програми для 3D моделювання
- Локальні додатки як особисті помічники у великих будівлях, таких як аеропорти
- Бізнес-додатки

- Архітектура додатків
- Заявки на виробниче навчання [16]

Система анімації дозволяє створювати, керувати та керувати анімацією для ваших об'єктів, будь то ходьба персонажа, відкриття дверей або складний cutscene. Компонент аніматор і анімація надають інструменти для створення гладких і складних анімацій. Unity 3D також пропонує магазин активів, ринок, де ви можете купити або продати активи, такі як 3D-моделі, текстури, звуки, скрипти та повні шаблони проектів. Це може значно прискорити розробку, надаючи готові ресурси, які можна інтегрувати в свої проекти.

Спільноти і форуми, такі як Unity Community, Stack Overflow і Reddit, є чудовими місцями для отримання підтримки та обміну досвідом з іншими розробниками. YouTube канали, такі як Brackeys, офіційний канал Unity та Code Monkey, пропонують багато відеоуроків та корисних порад [5].

### **1.3.2. Архітектура Unity**

Unity - це комплексна платформа для розробки ігор, яка дозволяє створювати інтерактивний 2D і 3D контент. Його архітектура розроблена таким чином, щоб бути модульною та гнучкою, підтримуючи широкий спектр функцій та інструментів для задоволення потреб розробників. Ця модульна архітектура дозволяє розробникам налаштовувати та розширювати двигун відповідно до вимог проекту.

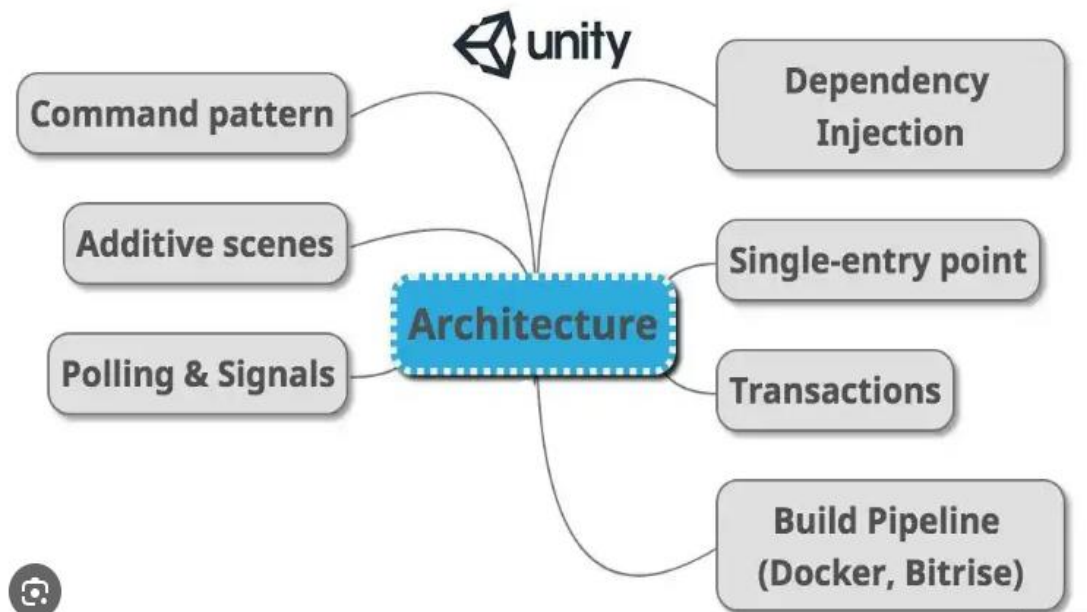


Рис. 1.3. Архітектура Unity

### Основні компоненти архітектури єдності:

Редактор єдності - центральний центр розробки, що забезпечує візуальний інтерфейс для проектування сцен, управління ресурсами та налаштування ігрових об'єктів. Він включає різні вікна, такі як перегляд сцени, перегляд гри, ієрархія.

GameObjects та Components - основні будівельні блоки в Unity. GameObjects є основними об'єктами в сцені Unity, в той час як Компоненти приєднуються до GameObjects, щоб визначити їх поведінку і зовнішній вигляд. Наприклад, GameObject може мати такі компоненти, як Transform, MeshRender, Collider та власні скрипти.

Сценарії - сценарії, в основному написані на C #, використовуються для визначення поведінки GameObjects. API сценаріїв забезпечує доступ до функцій рушія та дозволяє розробникам реалізовувати ігрову логіку, обробляти події та взаємодіяти з іншими компонентами [6].

Physics Engine - Unity включає інтегровані двигуни фізики для 2D такі як Box2D і PhysX для 3D. Ці двигуни моделюють поведінку фізики в реальному

світі, включаючи зіткнення, динаміку ригідного тіла та суглоби, забезпечуючи реалістичні взаємодії між об'єктами.

Анімаційна система - анімаційна система Unity, відома як Mecanim, дозволяє створювати та контролювати складні анімації. Він підтримує скелетну анімацію, змішування дерев, станів машин та змішування анімації, що робить його придатним для анімації персонажів та інших динамічних елементів.

Networkin - Unity забезпечує вбудовану підтримку багатокористувацьких мереж. Це включає в себе API високого рівня для створення мережевих ігор, а також API нижчого рівня для тонкого контролю над мережевою комунікацією.

Asset Store або Unity's Asset Store - це інтернет-ринок, де розробники можуть купувати та продавати активи, включаючи моделі, текстури, скрипти та повні шаблони проектів. Цей ресурс може значно прискорити розробку, надаючи готові рішення.

#### **1.4. Механіки розробки ігор**

Розробка ігор включає в себе різні механіки і процеси, які беруть гру від концепції до завершення. Ці механіки можуть бути в цілому розділені на кілька ключових етапів, кожен з яких вимагає конкретних навичок і підходів. Перейдемо до таких аспектів як:

1. Ідея та концепція, які є початковим етапом, на якому створюється основна концепція гри. Він передбачає визначення жанрової теми основних цілей і фундаментальної механіки гри. Також досліджуються ідеї про сюжетну лінію персонажів і ігровий світ.
2. Після створення ідеї створюється детальний документ, який описує всі аспекти гри включає в себе геймплей механіки сюжетних персонажів рівнів правил візуального стилю і звукового оформлення

3. Створення прототипу. Базова версія гри розроблена, що містить основну механіку геймплея. Це дозволяє розробникам перевірити, чи працюють основні ідеї на практиці та виявити потенційні проблеми на ранній стадії.
4. Програмування та розробка. Це основний етап розробки, де написаний код гри. Різні мови програмування та середовища розробки використовуються в залежності від платформи та складності гри. Ключові аспекти включають програмування геймплея. Реалізація основної механіки геймплея та забезпечення їх функціонування за призначенням AI. Програмування створення поведінки для неігрових персонажів, щоб зробити їх взаємодію з гравцем і навколишнім середовищем реалістичним способом.
5. Тестування. Проведення різних тестів для забезпечення безперебійної роботи гри та відсутності помилок. Це включає в себе Alpha Testing. Внутрішнє тестування для виявлення та усунення основних проблем на початку розробки. Бета-тестування. Зовнішнє тестування з обмеженою аудиторією для збору відгуків та виявлення інших проблем.
6. Після випуску гри розробники надають оновлення та виправлення, щоб вирішити будь-які проблеми, з якими стикаються гравці. Це також може включати випуск додаткового вмісту, такого як розширення або оновлення завантажуваного вмісту (DLC), щоб гра залишалася свіжою та привабливою [7].

## ВИСНОВКИ ДО РОЗДІЛУ 1

У першому розділі було розглянуто такі аспекти як: історія виникнення відеоігор, їх позитивні та негативні сторони, розглянуто основні аспекти технологій розробки відеоігор на движку Unity3D, та основні механіки, які треба включати у створення проекту.

Ера зародження відеоігор розпочалася з «Тенісу для двох», та упродовж останніх 75 років пройшла етап комп'ютерних відеоігор, консольних відеоігор, мобільних відеоігор, та продовжує розвиватися на етапі мультиплеєра та VR.

Кожна людина зможе знайти для себе її улюблений жанр з великого переліку інді-ігор, головоломок, аркадних ігор, шутерів, VR-ігор, кооперативні ігри, ММО ігри.

Зараз геймдев переживає нову еру завдяки хмарним технологіям та стрімінговим платформам, що роблять ігри більш доступними та мобільними. Кросплатформеність, соціальні функції та новаторські геймплейні рішення стають стандартом. Мультиплеєр визначає не лише майбутнє геймдеву, а й спосіб, яким гравці взаємодіють із віртуальним світом, роблячи ігри не просто розвагою, а ареною для творчості, спілкування та справжнього віртуального пригодництва

Платформа Unity3D використовується у багатьох сферах таких як: дизайну, бізнес-додатків, програм, 3D моделювання. Також ця платформа має великий спектр архітектурних можливостей: Asset Store або Unity's Asset Store, анімаційні системи, GameObjects та Components. Що надає перевагу над багатьма іншими платформами.

## РОЗДІЛ 2

### ТЕХНОЛОГІЇ ТА РЕАЛІЗАЦІЯ ВІДЕОГРИ

#### 2.1. Компоненти розробки відеогри на Unity

Створення гри в Unity є дуже різноманітними, дозволяючи розробникам створювати як прості, так і складні ігри. Ось деякі з основних компонентів та технологій Unity, які часто використовуються в розробці ігор: **Колайдери (Colliders)** у Unity є важливими компонентами для обробки колізій між об'єктами у грі. Вони визначають форму об'єктів, з якою можуть взаємодіяти інші об'єкти [18]. Ось основні типи колайдерів:

##### Box Collider

- Box Collider визначає колізію у формі коробки.
- Використовується для об'єктів з прямокутною формою, наприклад, ящики, будівлі.

##### Sphere Collider

- Sphere Collider визначає колізію у формі сфери.
- Ідеально підходить для сферичних об'єктів, таких як м'ячі.

##### Capsule Collider

- Capsule Collider використовується для циліндричних об'єктів з півсферичними кінцями.
- Часто використовується для моделювання персонажів.

Кафедра КІТ				НАУ 24 27 30 000 ПЗ			
	ПІБ			ТЕХНОЛОГІЇ ТА РЕАЛІЗАЦІЯ ВІДЕОГРИ	Літ.	Аркуш	Аркушів
Розроб.	Мирошников Д.О.					23	55
Керівник	Толстікова О.В.				ТП-415Б - 122		
Н-контроль	Сидоренко В.М.						

## Mesh Collider

- Mesh Collider дозволяє використовувати складні форми об'єктів для колізії.
- Зазвичай використовується для об'єктів з деталізованими формами, та як скелі [8].

## Ріджітбоді (Rigidbody)

Компонент Rigidbody додає об'єктам фізичні властивості, такі як маса, гравітація та сили, що діють на них. З цим компонентом об'єкти стають частиною фізичної симуляції Unity. Основні властивості:

- Mass: маса об'єкта.
- Drag: опір повітря, який впливає на швидкість об'єкта.
- Angular Drag: опір повітря, який впливає на кутову швидкість об'єкта.
- Use Gravity: чи підпорядковується об'єкт гравітації.

Rigidbody також дозволяє контролювати рух об'єктів за допомогою скриптів, використовуючи методи, такі як AddForce для додавання сил або AddTorque для додавання крутних моментів. Параметр Is Kinematic дозволяє виключити об'єкт з фізичної симуляції, але при цьому дозволяє керувати його рухом вручну. Interpolate і Extrapolate використовуються для згладжування руху об'єктів, що може бути корисним для створення більш плавної анімації. Constraints дозволяють обмежити рух або обертання об'єкта по певних осях. Таким чином, Rigidbody є ключовим компонентом для створення реалістичної фізики у грі.

## Скрипти

Скрипти у Unity пишуться на мовах програмування C# або UnityScript (аналог JavaScript). Вони додають логіку і поведінку об'єктам у грі. Основні елементи скриптів:

- MonoBehaviour: базовий клас для всіх скриптів в Unity.
- Start(): метод, який виконується при запуску сцени.
- Update(): метод, який виконується кожен кадр [19].



## **Магазин асетів (Assets Store)**

Unity Asset Store - це онлайн-платформа, де розробники можуть купувати або завантажувати безкоштовні активи для своїх ігор. Це можуть бути моделі, текстури, звукові ефекти, анімації, скрипти, та інші ресурси, що прискорюють процес розробки [20].

## **Камери**

Камери в Unity визначають те, що бачить гравець. Вони можуть мати різні властивості та методи управління:

- **Field of View (FOV):** кут огляду камери, який визначає ширину зони видимості. Цей параметр може бути змінений для створення різних ефектів, таких як "риб'яче око" або вузький фокус.
- **Clipping Planes:** визначає мінімальну та максимальну відстань рендеринга. Об'єкти поза цими межами не будуть видимі, що допомагає оптимізувати продуктивність.
- **Depth:** порядок рендеринга, коли є кілька камер. Камера з меншою глибиною рендеритиме об'єкти перед камерами з більшою глибиною [17].

## **Сцени**

Сцени - це окремі рівні або етапи в грі. Кожна сцена може містити різні об'єкти, світла, камери та інші компоненти. Розробники можуть створювати та управляти різними сценами для організації ігрового процесу.

## **Анімації**

Анімаційна система Unity дозволяє створювати та контролювати анімації для персонажів та об'єктів. Основні компоненти:

- **Animator:** контролер анімацій, який керує зміною станів.
- **Animation Clips:** окремі анімації для конкретних дій [9].

## Шейдери та матеріали

Шейдери визначають, як об'єкти відображаються на екрані. Вони відповідають за обробку візуальних ефектів, таких як освітлення та тіні. Матеріали використовують шейдери для визначення візуальних властивостей об'єктів.

Використання цих технологій дозволяє розробникам створювати складні та реалістичні ігри з багатою графікою та інтерактивністю [10].

## 2.2. Стапи створення відеогри

### 2.2.1. Створення сцен

Спочатку створимо першу пусту сцену, яку будемо використовувати як меню - MainMenu.

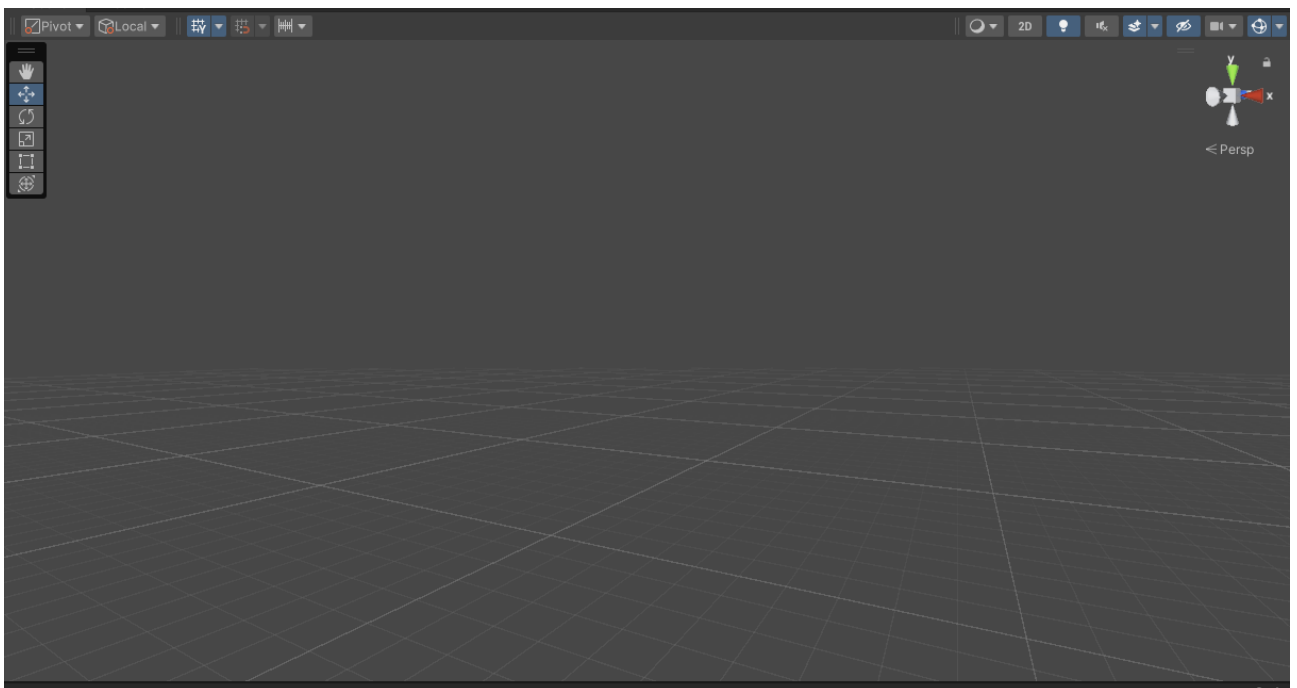


Рис. 2.1. Пуста сцена

Додаємо Canvas до сцени, який є контейнером для всіх UI елементів в Unity. У Canvas додаємо три основні кнопки: "Play", "Settings" та "Exit". Створимо дві панелі, першу панель назвемо PlayPanel, а другу - SettingsPanel. Налаштовуємо кнопки так, щоб при натисканні вони відкривали відповідні панелі. Для кнопки "Play" додамо компонент Button (Script) і в полі OnClick() встановимо PlayPanel активною при натисканні. Для кнопки "Settings" додамо компонент Button (Script) і в полі OnClick() встановимо SettingsPanel активною при натисканні.

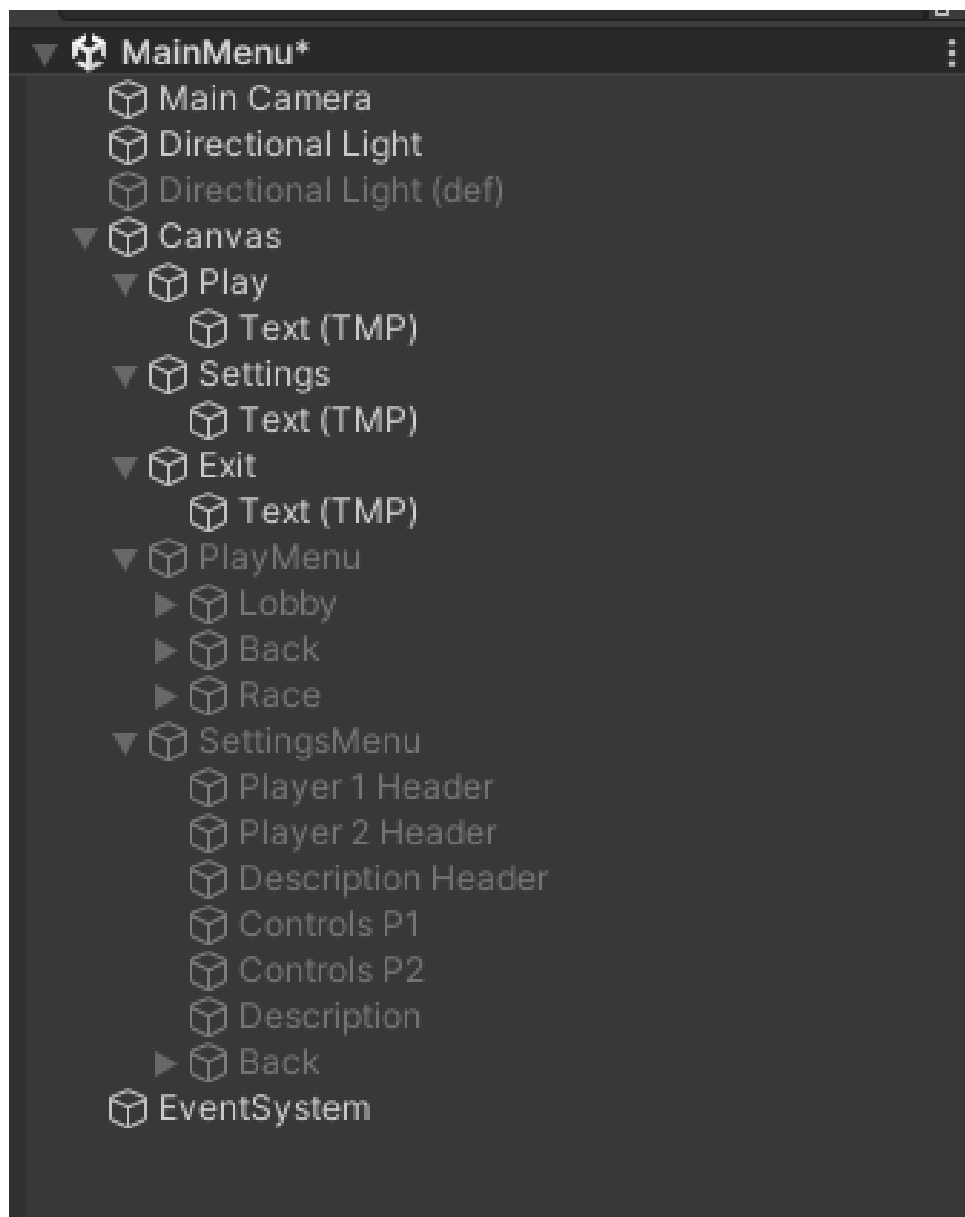


Рис. 2.2. Елементи ієрархії на сцені MainMenu

Також добавимо стилі для кнопок, щоб надати їм привабливий вигляд та встановлюємо skybox для сцени, щоб забезпечити гарне фонове зображення. Призначаємо відповідні дії на всі кнопки: кнопка "Play" відкриватиме панель з ігровими налаштуваннями, кнопка "Settings" відкриватиме панель з налаштуваннями гри, а кнопка "Exit" закриватиме гру.

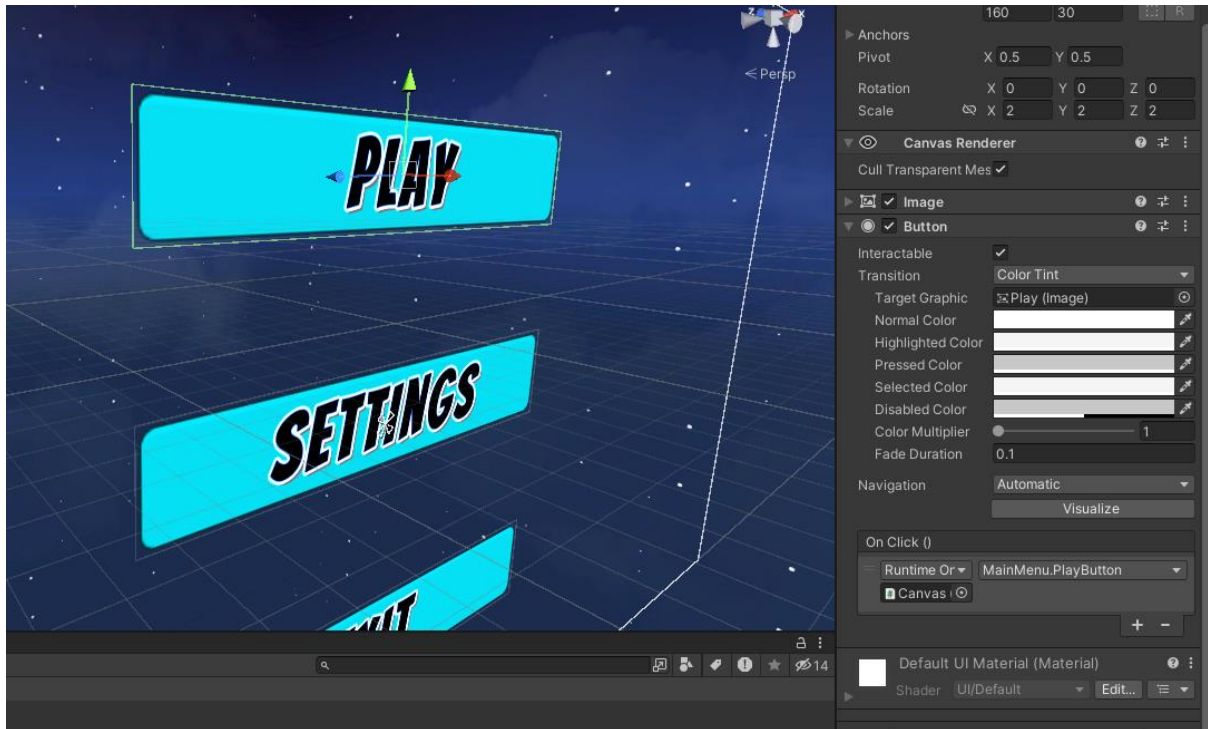


Рис. 2.3. Налаштування кнопок в MainMenu

Створюємо другу сцену для тренувань – Lobby, і одразу призначаємо skybox для цієї сцени, щоб надати їй відповідну атмосферу. Додаємо декоративні об'єкти до сцени Lobby: дерева, будинки, дорогу, розмітку, конуси та інше, щоб створити реалістичне та привабливе середовище для тренувань.

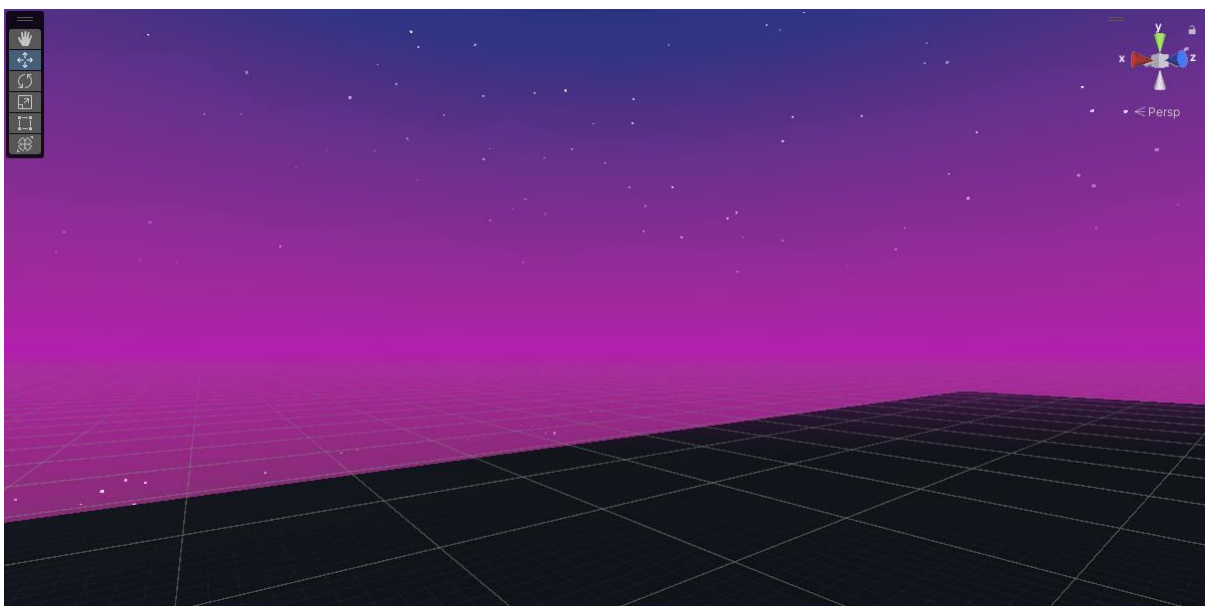


Рис. 2.4. Пуста сцена Lobby з Skybox



Рис. 2.5. Декоративні об'єкти сцени Lobby

Імпортуємо з Unity Asset Store декілька моделей автомобілів та призначаємо їм компоненти rigidbody для фізичних властивостей і box collider для границь.



Рис. 2.6. Рухливі об'єкти сцени Lobby

Те ж саме робимо і для третьої сцени – RaceScene. Додаємо skybox, налаштовуємо освітлення, розташовуємо та плануємо маршрут траси. Розміщуємо декоративні елементи, такі як глядацькі трибуни, рекламні щити, дерева, і встановлюємо необхідні компоненти для всіх об'єктів. Це включає призначення компонентів rigidbody для фізичних властивостей та collider для обробки зіткнень. Таким чином, створюємо реалістичну та детально опрацьовану трасу для гонок.

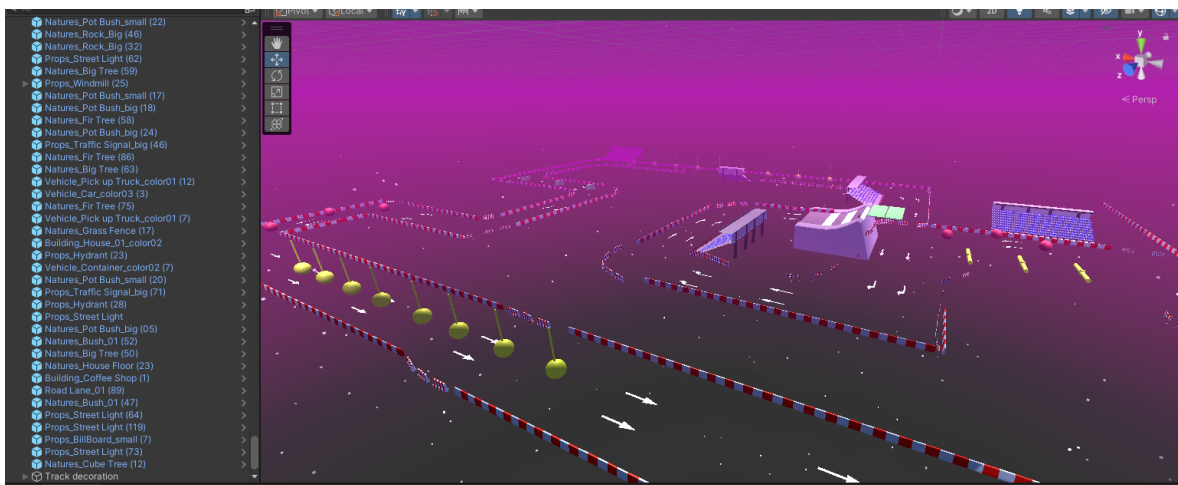


Рис. 2.7 Маршрут траси на сцені RaceScene

### 2.2.2. Створення гравців

Спочатку створюємо пустий об'єкт і призначаємо на нього Rigidbody. Всередині нього буде декілька елементів:

- Тіло об'єкту: це буде модель машини з призначеним колайдером та шейдерами.
- Колеса: додаємо колеса з колайдерами та шейдерами для реалістичного відображення і взаємодії з поверхнею.
- Ефекти диму: додаємо дим, який розташований біля задніх коліс, і налаштовуємо його для відтворення під час крутих поворотів [11].

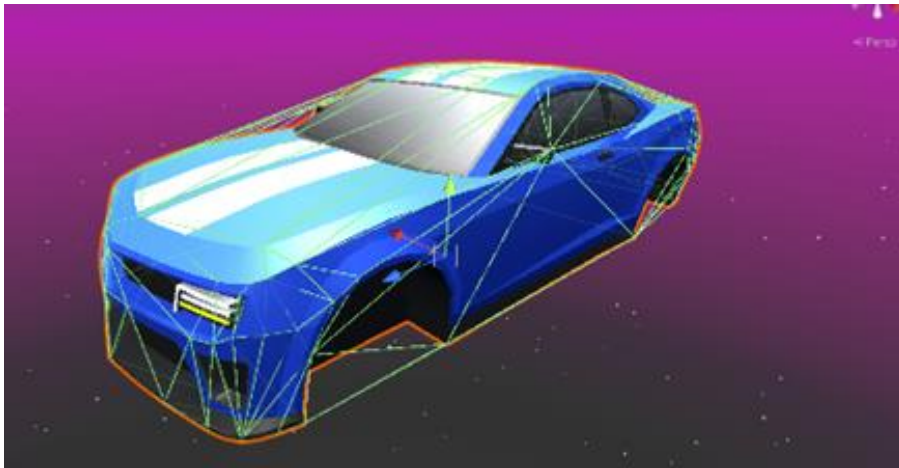


Рис. 2.8. Модель машини гравця

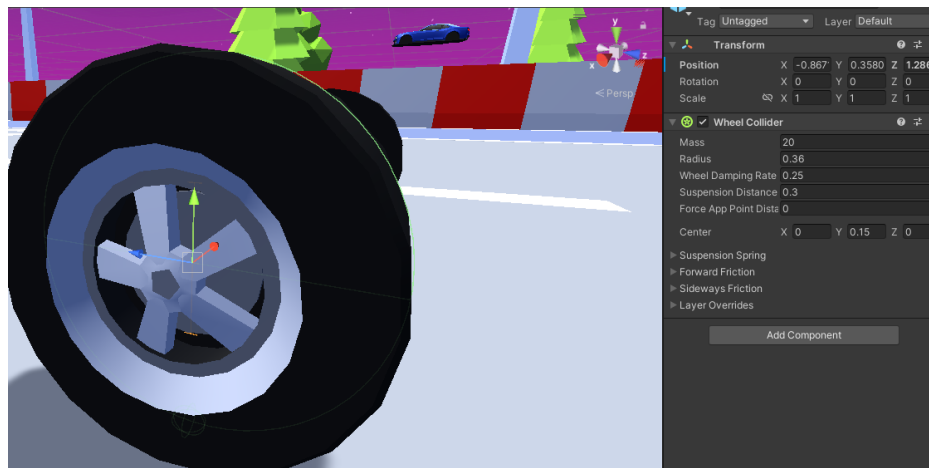


Рис. 2.9. Колеса машини гравця

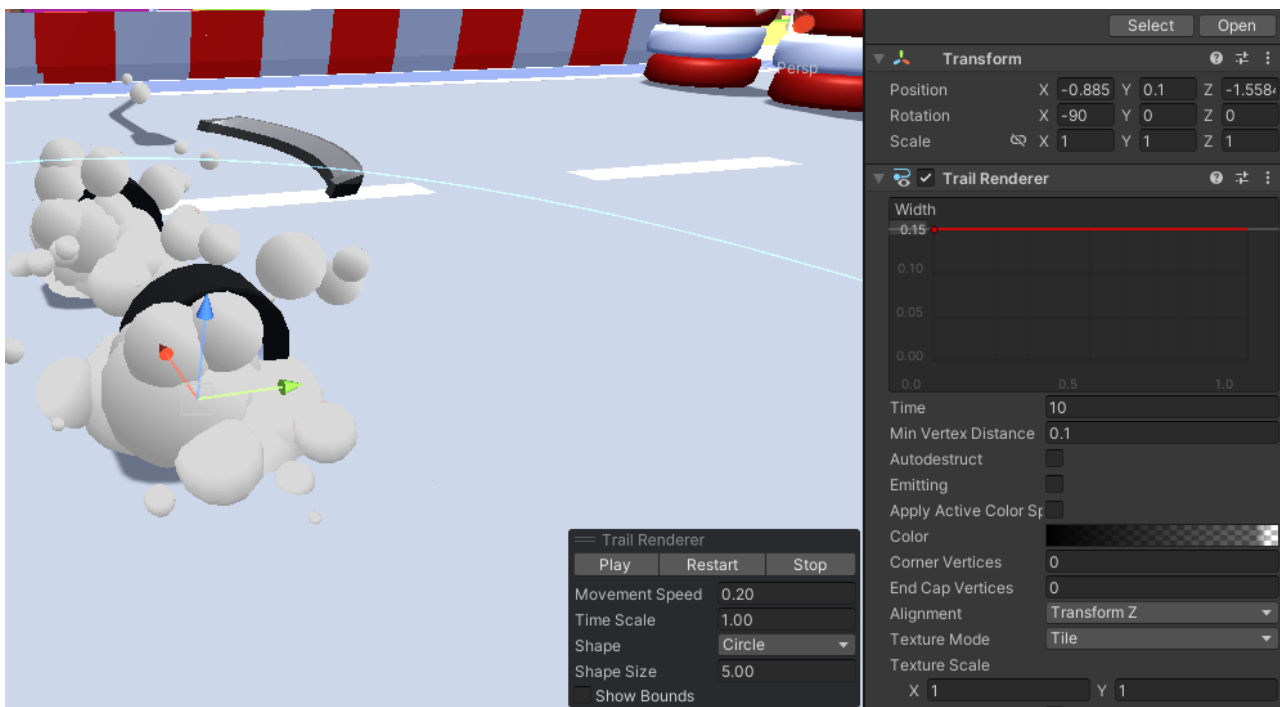


Рис. 2.9. Ефект диму з під колес машини гравця

Далі на наш об'єкт напишемо скрипт, у якому буде реалізовано керування автомобілем. Додамо глобальні змінні, щоб легко змінювати властивості гравця без необхідності відкривати сам файл зі скриптом.

Основні елементи скрипта:

- Глобальні змінні: для налаштування швидкості, керуваності, сили гальмування тощо.
- Метод для поворотів: керуватиме обертанням автомобіля залежно від введення користувача.
- Метод для заднього ходу: дозволить автомобілю рухатися назад.
- Метод для ручного гальма: забезпечить можливість входу в занос.
- Ефекти диму: метод, який активуватиме ефекти диму під час крутих поворотів.
- Звукові ефекти: методи для відтворення звуків двигуна, гальмування та інших звукових ефектів [12].



Ці елементи забезпечать комплексне та реалістичне керування автомобілем у грі.

```
public void TurnLeft(){
    steeringAxis = steeringAxis - (Time.deltaTime * 10f * steeringSpeed);
    if(steeringAxis < -1f){
        steeringAxis = -1f;
    }
    var steeringAngle = steeringAxis * maxSteeringAngle;
    frontLeftCollider.steerAngle = Mathf.Lerp(frontLeftCollider.steerAngle, steeringAngle, steeringSpeed);
    frontRightCollider.steerAngle = Mathf.Lerp(frontRightCollider.steerAngle, steeringAngle, steeringSpeed);
}

public void TurnRight(){
    steeringAxis = steeringAxis + (Time.deltaTime * 10f * steeringSpeed);
    if(steeringAxis > 1f){
        steeringAxis = 1f;
    }
    var steeringAngle = steeringAxis * maxSteeringAngle;
    frontLeftCollider.steerAngle = Mathf.Lerp(frontLeftCollider.steerAngle, steeringAngle, steeringSpeed);
    frontRightCollider.steerAngle = Mathf.Lerp(frontRightCollider.steerAngle, steeringAngle, steeringSpeed);
}
```

Рис. 2.10. Методи поворотів

```
// This method matches both the position and rotation of the WheelColliders with the WheelMeshes.
void AnimateWheelMeshes(){
    try{
        Quaternion FLWRotation;
        Vector3 FLWPosition;
        frontLeftCollider.GetWorldPose(out FLWPosition, out FLWRotation);
        frontLeftMesh.transform.position = FLWPosition;
        frontLeftMesh.transform.rotation = FLWRotation;

        Quaternion FRWRotation;
        Vector3 FRWPosition;
        frontRightCollider.GetWorldPose(out FRWPosition, out FRWRotation);
        frontRightMesh.transform.position = FRWPosition;
        frontRightMesh.transform.rotation = FRWRotation;

        Quaternion RLWRotation;
        Vector3 RLWPosition;
        rearLeftCollider.GetWorldPose(out RLWPosition, out RLWRotation);
        rearLeftMesh.transform.position = RLWPosition;
        rearLeftMesh.transform.rotation = RLWRotation;

        Quaternion RRWRotation;
        Vector3 RRWPosition;
        rearRightCollider.GetWorldPose(out RRWPosition, out RRWRotation);
        rearRightMesh.transform.position = RRWPosition;
        rearRightMesh.transform.rotation = RRWRotation;
    }catch(Exception ex){
        Debug.LogWarning(ex);
    }
}
```

Рис. 2.11. Метод анімації диму

```
public void DriftCarPS(){  
  
    if(useEffects){  
        try{  
            if(isDrifting){  
                RLWParticleSystem.Play();  
                RRWParticleSystem.Play();  
            }else if(!isDrifting){  
                RLWParticleSystem.Stop();  
                RRWParticleSystem.Stop();  
            }  
        }catch(Exception ex){  
            Debug.LogWarning(ex);  
        }  
    }  
  
    try{  
        if((isTractionLocked || Mathf.Abs(localVelocityX) > 5f) && Mathf.Abs(carSpeed) > 12f){  
            RLWTireSkid.emitting = true;  
            RRWTireSkid.emitting = true;  
        }else {  
            RLWTireSkid.emitting = false;  
            RRWTireSkid.emitting = false;  
        }  
    }catch(Exception ex){  
        Debug.LogWarning(ex);  
    }  
}
```

Рис. 2.12. Метод для активації заносу

### 2.2.3. Сторення чекпоінтів для старту та фінішу

В Hierarchy додаємо два пусті об'єкти – StartCheckpoint та FinishCheckpoint. Розміщуємо їх у потрібних місцях на сцені, додаємо box collider і встановлюємо прапорець “Is Trigger”.

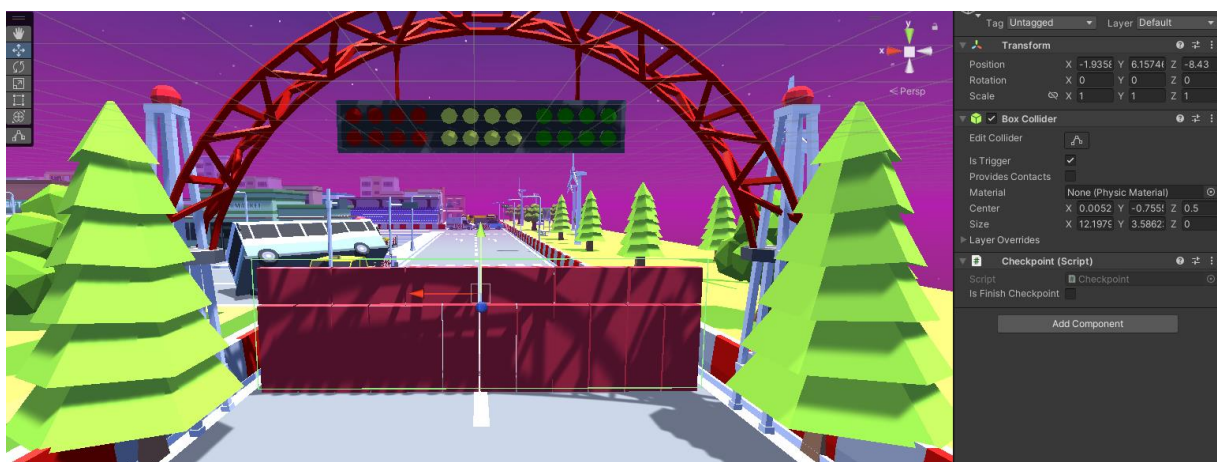


Рис. 2.13. Вигляд старту з StartCheckpoint

Далі створюємо скрипт для перевірки чи перетнув гравець наш чекпоінт та додаємо тег “Player” гравцям.

```
public class Checkpoint : MonoBehaviour
{
    public bool isFinishCheckpoint;

    private void OnTriggerEnter(Collider other)
    {
        Debug.Log("OnTriggerEnter called on " + gameObject.name + " by " + other.name);

        if (other.CompareTag("Player"))
        {
            Debug.Log("Checkpoint triggered by " + other.name + " | Finish: " + isFinishCheckpoint);
            RaceManager.Instance.PlayerPassedCheckpoint(other.gameObject, isFinishCheckpoint);
        }
        else
        {
            Debug.Log(other.name + " does not have the Player tag.");
        }
    }
}
```

Рис. 2.14. Метод для перевірки чекпоінтів

## ВИСНОВКИ ПО РОЗДІЛУ 2

У другому розділі було розглянуто основні компоненти та етапи створення відеоігор на платформі Unity, що включають:

### 1. Компоненти розробки відеоігор на Unity:

Колайдери та компонент Rigidbody забезпечують реалістичну фізичну взаємодію об'єктів у грі. Колайдери визначають форму об'єктів та обробляють зіткнення, а Rigidbody додає фізичні властивості, такі як маса та гравітація.

### 2. Створення сцен:

Процес створення сцен включає головне меню, лоббі та гоночний трек. Додаються декоративні об'єкти, налаштовуються камери та інтегруються активи з Unity Asset Store для створення реалістичного середовища гри.

### 3. Створення гравців:

Розробка ігрових персонажів, таких як автомобілі, включає додавання коліс, ефектів диму та скриптів для керування. Це забезпечує комплексне та реалістичне керування автомобілем у грі.

### 4. Створення чекпоінтів:

Створення стартових та фінішних чекпоінтів включає додавання колайдерів та скриптів для перевірки перетину чекпоінтів гравцем, що дозволяє точно реєструвати результати гонок [13].

Даний розділ демонструє, що використання технологій Unity дозволяє створювати складні та реалістичні ігри. Використання ключових компонентів, таких як колайдери, Rigidbody та скрипти, а також організація сцен і персонажів, забезпечує високу якість та інтерактивність ігрового процесу.

## РОЗДІЛ 3

### ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ ВІДЕОГРИ

#### 3.1. Розробка концепції

Основна ідея гоночної відеогри ‘Drift Duel’, де два гравці керують автомобілем – це прийти першим до фінішу оминаючи прешекоди та проходячи трасу з різкими поворотами, де головною ідеєю є елемент дрифту.

Відеогра орієнтована для любителів аркадних гоночних ігор. Вона підійде як для дорослих так і дітей. У цю відеогру можна грати тільки на комп’ютері, а керувати за допомогою клавіатури для обох гравців.

У цій відеогрі реалізовані такі елементи як: інтерфейс користувача, головне меню, вибір автомобілів, вибір кнопок для керування, присутні звукові ефекти, а саме звук двигуна та колес. Для всіх необхідних об’єктів було додано колайдери, які відповідають всіма необхідними формами та розмірами.

#### 3.2. Сцени відеогри

Відеогра ‘Drift Duel’ налічує загалом 3 сцени. Перша сцена – сцена інтерфейсу, куди входить головне меню (Рис. 3.1., Рис. 3.2., Рис. 3.3.)

Кафедра КІТ				НАУ 24 27 30 000 ПЗ				
	ПІБ			ПРОЕКТУВАННЯ ТА ТЕСТУВАННЯ ВІДЕОГРИ	Літ.	Аркуш	Аркушів	
<i>Розроб.</i>	Мирошников Д.О.					37	55	
<i>Керівник</i>	Толстікова О.В.				ТП-415Б - 122			
<i>Н-контроль</i>	Сидоренко В.М.							

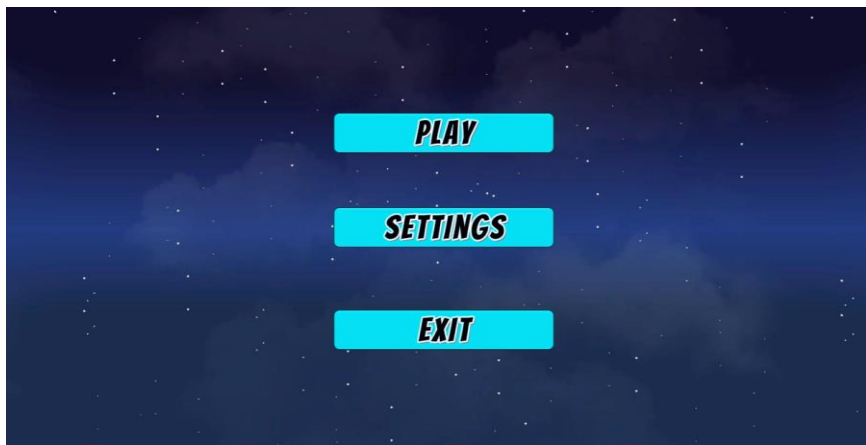


Рис. 3.1. Головне меню

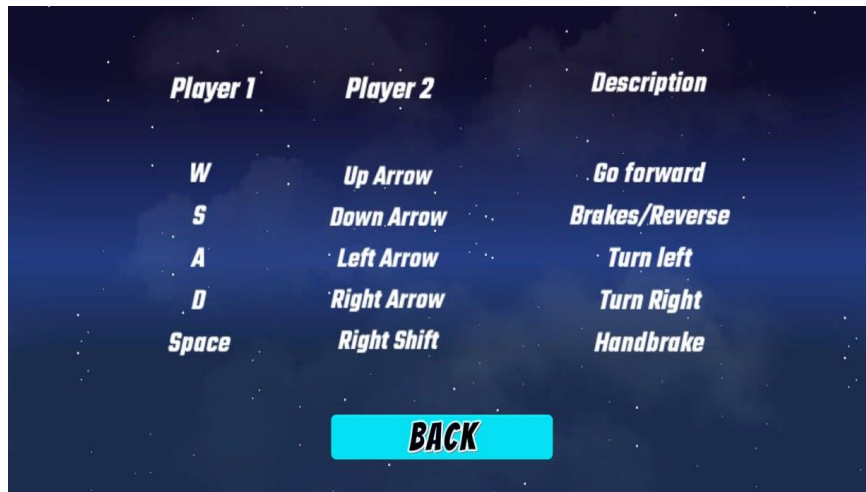


Рис. 3.2. Керування автомобілем

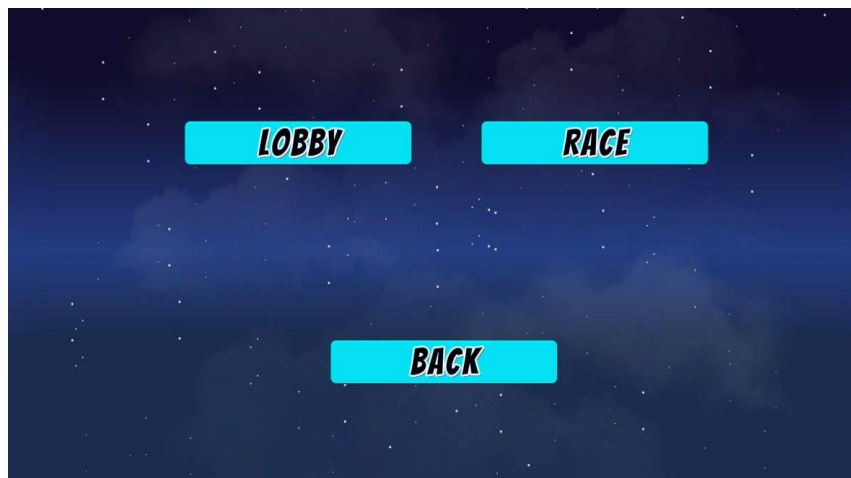


Рис. 3.3. Вибір локації

Друга сцена – сцена самого треку, де проходить основний геймплей гравців. (Рис. 3.4., Рис. 3.5.)



Рис. 3.4. Головна мапа гри

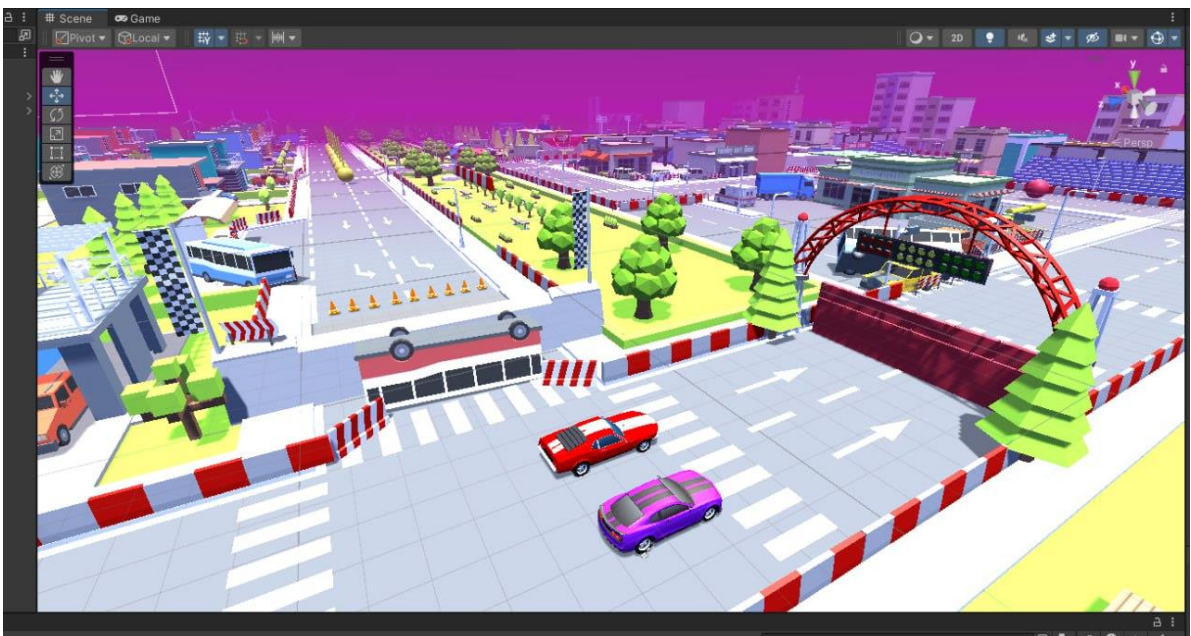


Рис. 3.5. Початкова зона гравців

Третя сцена – сцена лоббі, для ознайомлення геймплею. У цьому лоббі гравці можуть відточити навички їзди, для кращої гри на трасі (Рис. 3.6.).

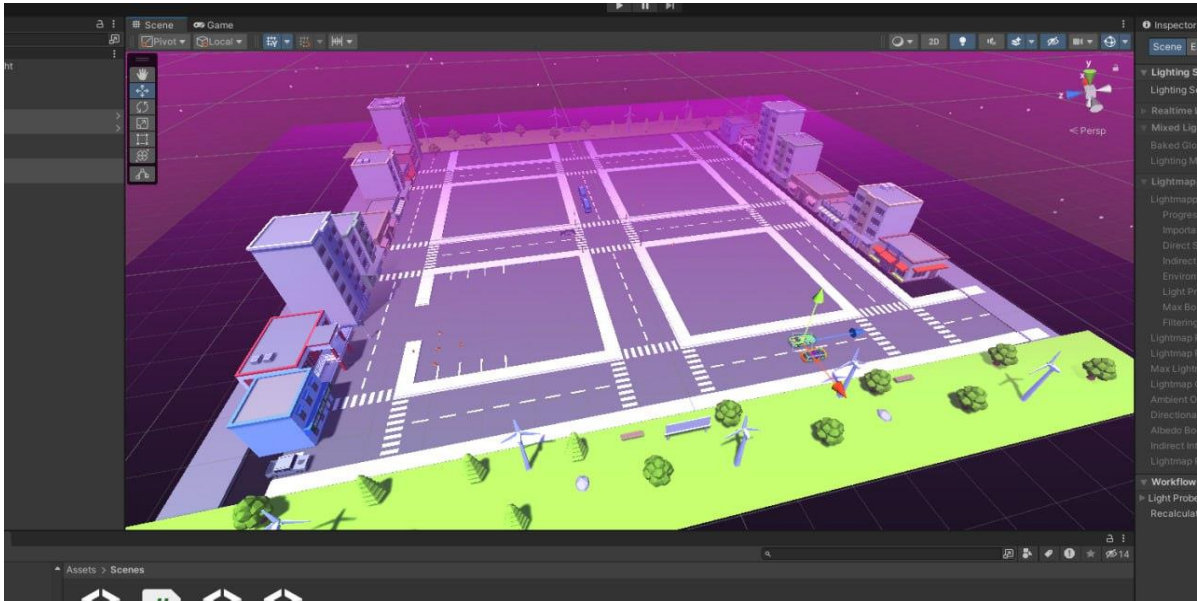


Рис. 3.6. Лоббі

**Також до елементів сцени відносяться:**

1. Камери – це об'єкти, які визначають точку зору для візуалізації сцени. Камера захоплює все, що знаходиться в її полі зору, і проектується на екрані, забезпечують взаємодію зі сценами (Рис. 3.7., Рис. 3.8.).

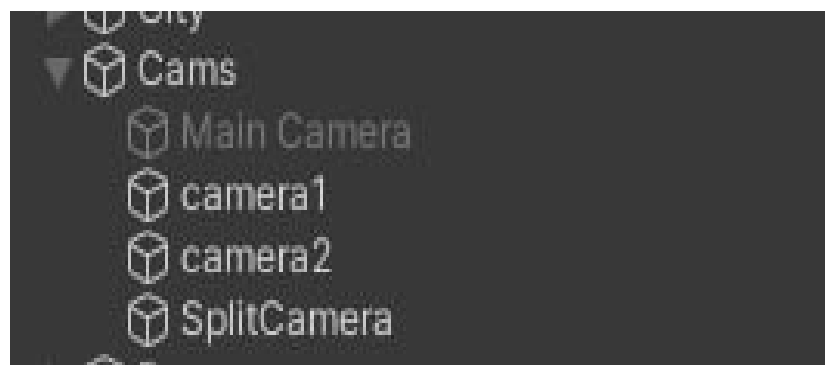


Рис. 3.7. Камери на другій сцені



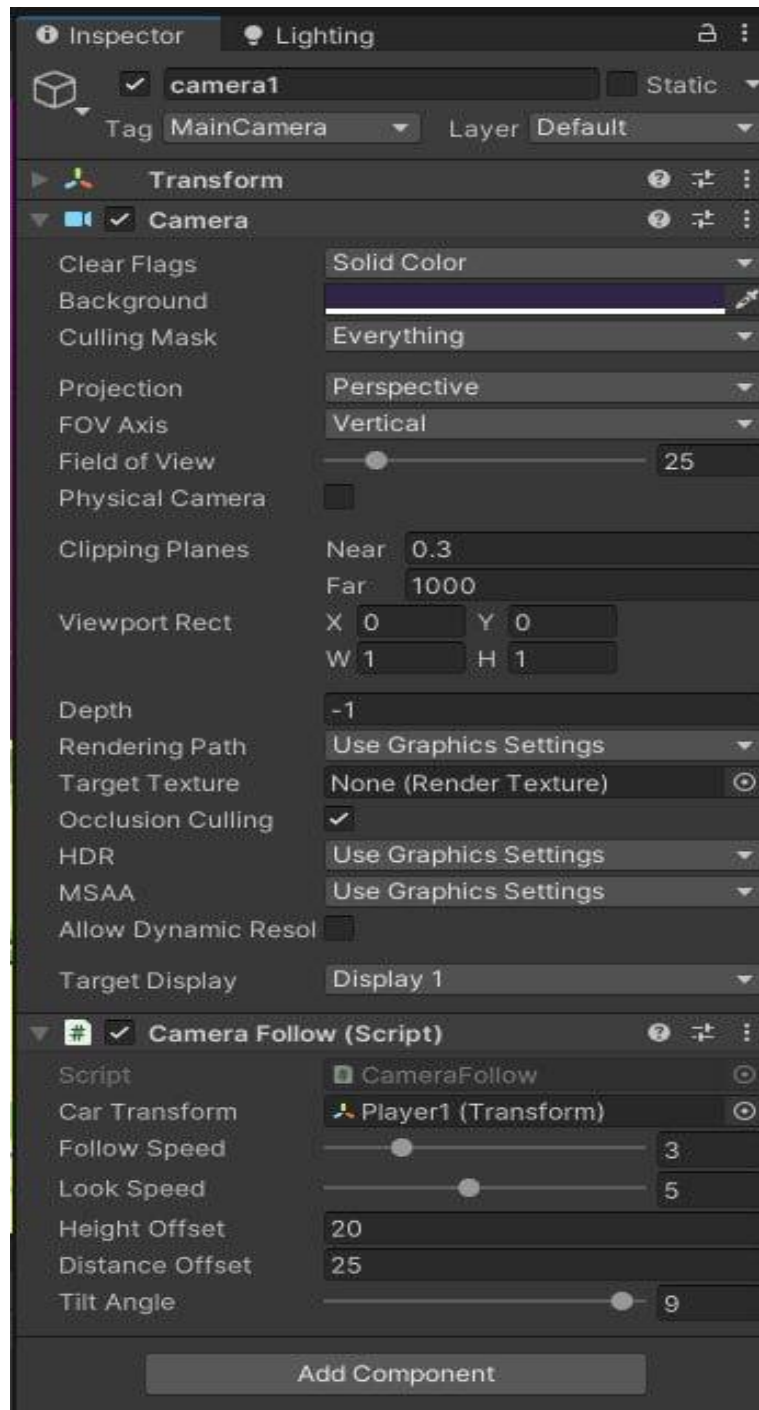


Рис. 3.8. Налаштування камери

2. SplitCamera – поділ екрану на декілька частин, у грі це реалізовано, для того, щоб кожен гравець під час заїзду бачив на екрані як свою машину, так і суперника (Рис. 3.9.).

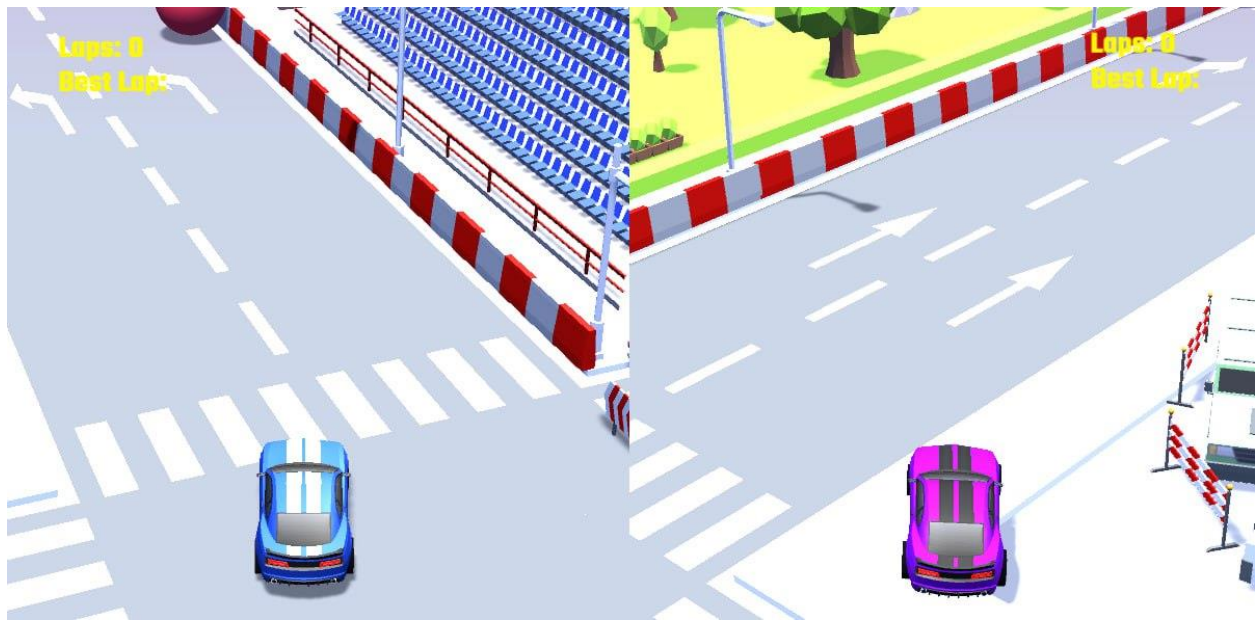


Рис. 3.9. SplitCamera для обох гравців

3. Освітлення – аспект, який визначає чи буде сцена освітлена чи ні, як об'єкти будуть виглядати в різних умовах.

4. SkyBox - це вид матеріалу, який використовується для створення фону в сцені. Він імітує небо, космос або будь-яке інше середовище. Skybox оточує сцену і створює ілюзію величезного простору навколо неї (Рис. 3.10.).

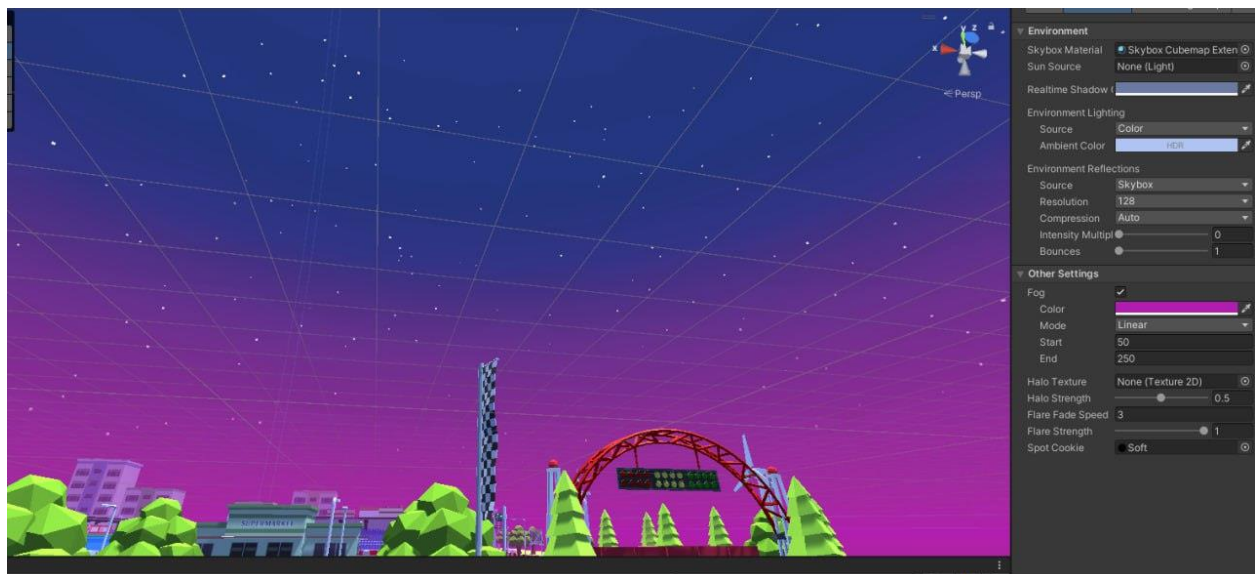


Рисунок 3.10. Вид SkyBox у відеогрі

### 3.3. Створення та обробка об'єктів

Створення та додавання об'єктів є базою для створення любого проекту та гри. До основних елементів розробки гри я віднесу:

#### 1. Додавання колайдерів

Колайдери - це компоненти, які додаються до ігрових об'єктів щоб визначити їх фізичні меж, щоб вони могли взаємодіяти з навколишніми елементами. Вони використовуються разом з компонентами RigidBody для керування фізикою об'єктів, які і були використані в роботі (Рис. 3.11., Рис. 3.12.).



Рис. 3.11. Mesh Collider для другого гравця



Рис. 3.12. Mesh Collider для елементів декорації

## 2. Об'єкти декорації

Ось приклад елементів декорації, які було додано до роботи (Рис. 3.13.)

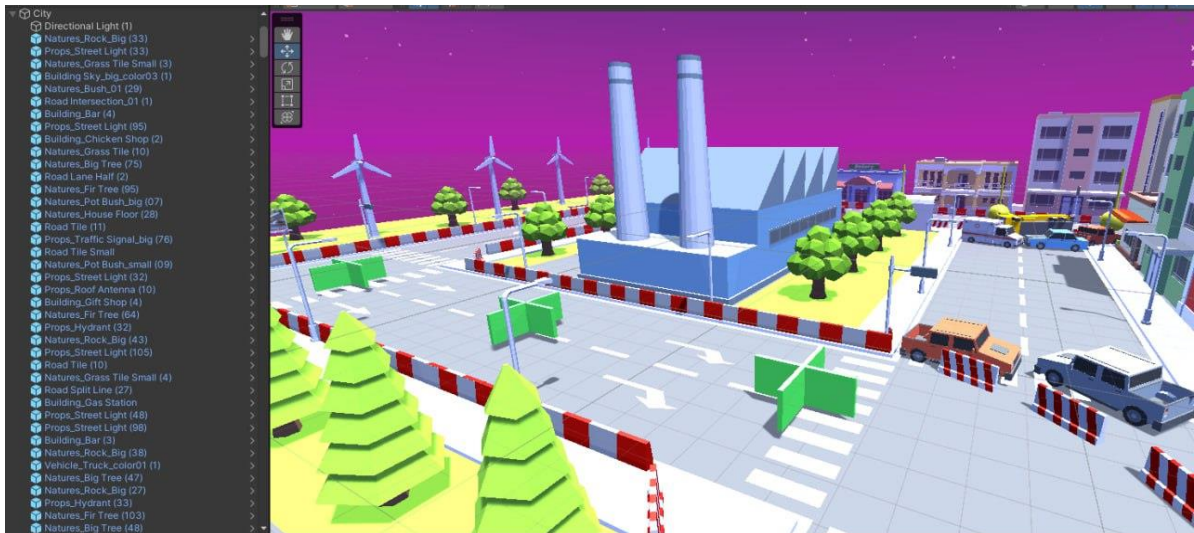


Рис. 3.13. Елементи декорації

## 3. Траса

Траса – елемент гри, на якій проходить основний геймплей (Рис. 3.14.).

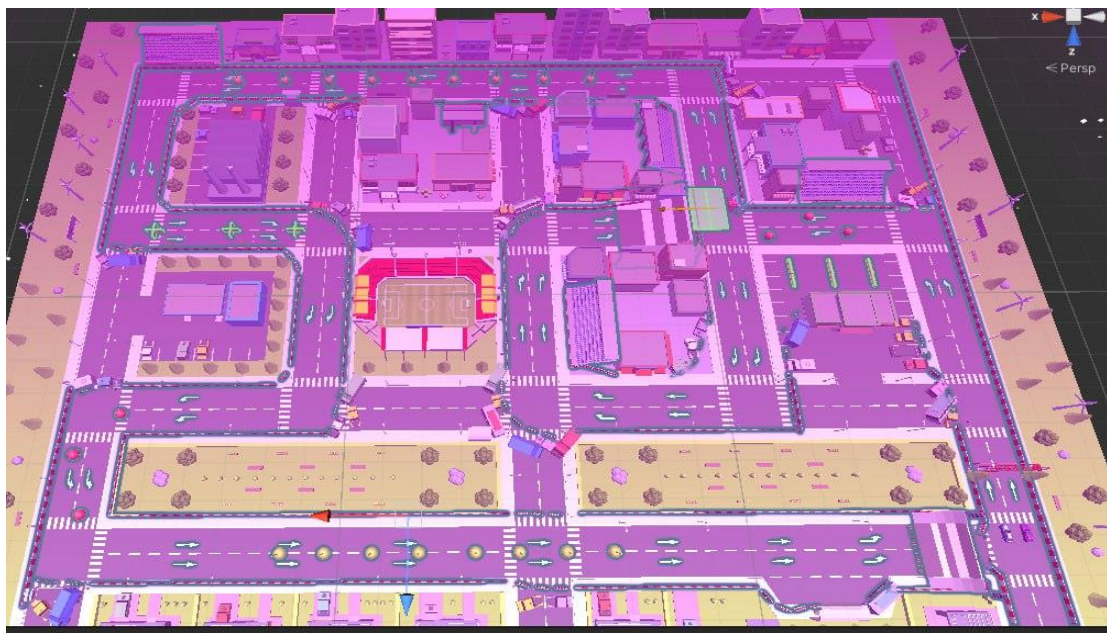


Рис. 3.14. Траса

#### 4. Перешкоди

Перешкоди у грі – елемент, який має сповільнити та ускладнити проходження траси для обох гравців ( Рис. 3.15.).

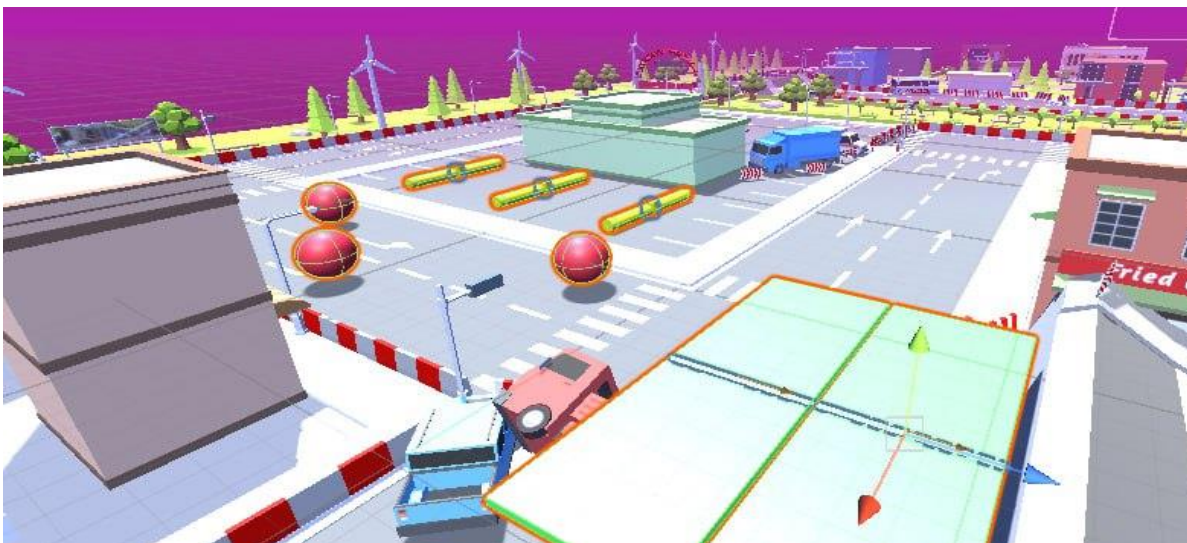


Рис. 3.15. Виділення об'єктів перешкоди

#### 5. Змінні

У роботі змінні такі як ( елементи швидкості, маса, дрифт, кількість кругів для перемоги (Рис. 3.16.), (Рис. 3.17.), було використано для геймплею.

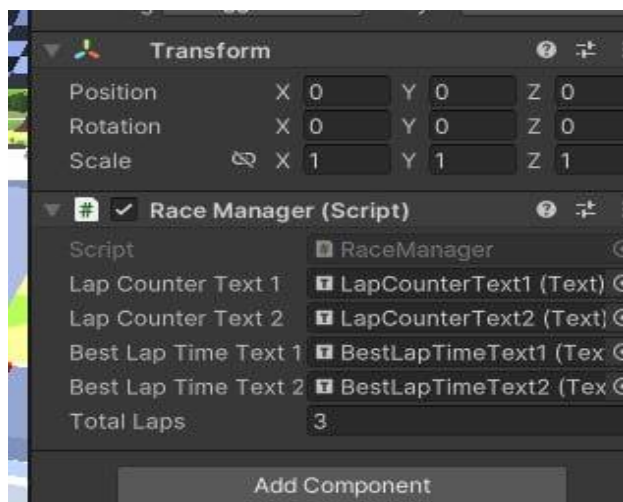


Рис. 3.16. Налаштування кількості кругів для перемоги

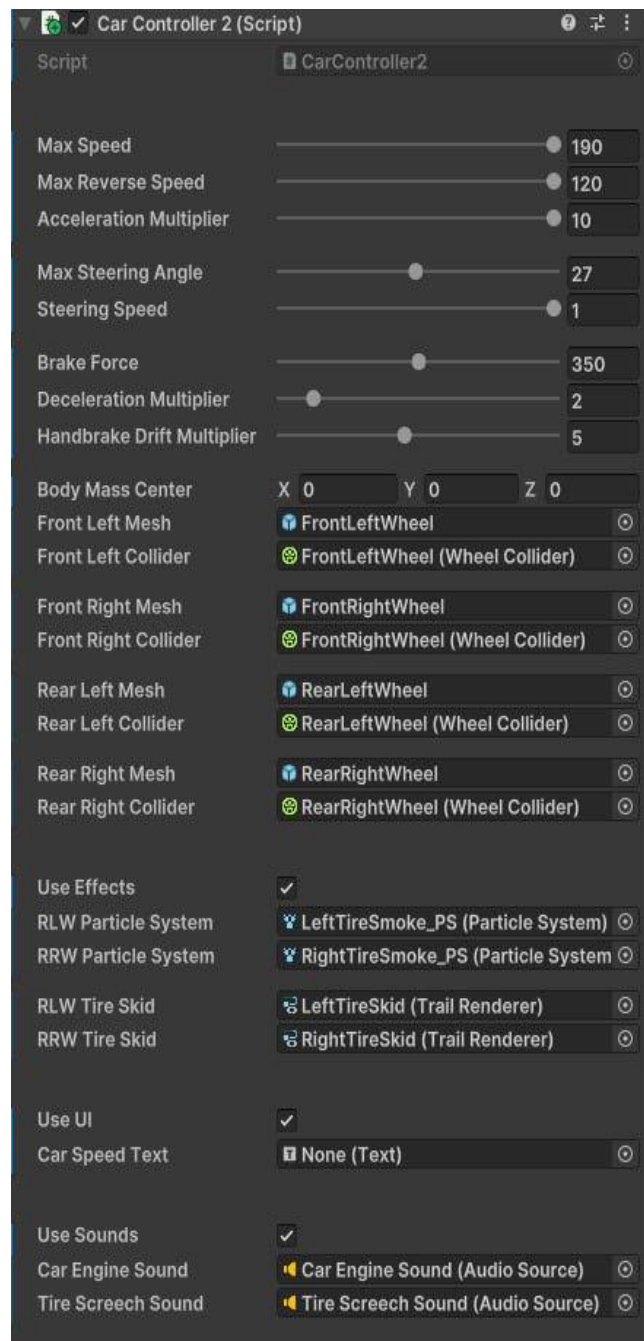


Рис. 3.17. Налаштування функціоналу автомобіля

### 3.4. Геймплей

Основний геймплей полягає в тому, щоб 2 гравці якомога швидше пройшли трасу з перешкодами, які під час усього заїзду будуть докучати гравцю. Зверху екрана у кожного гравця буде вказана кількість кругів, яку вони пройшли. Ось скріни геймплею з самої гри.



Рис. 3.18. Геймплей

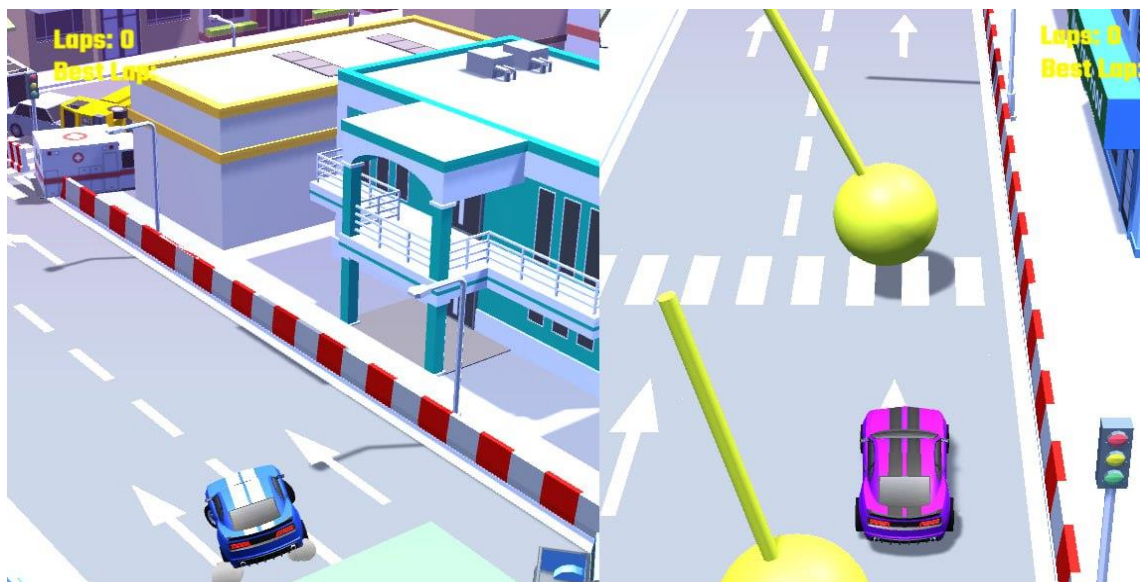


Рис. 3.19. Геймплей



Рис. 3.20. Геймплей

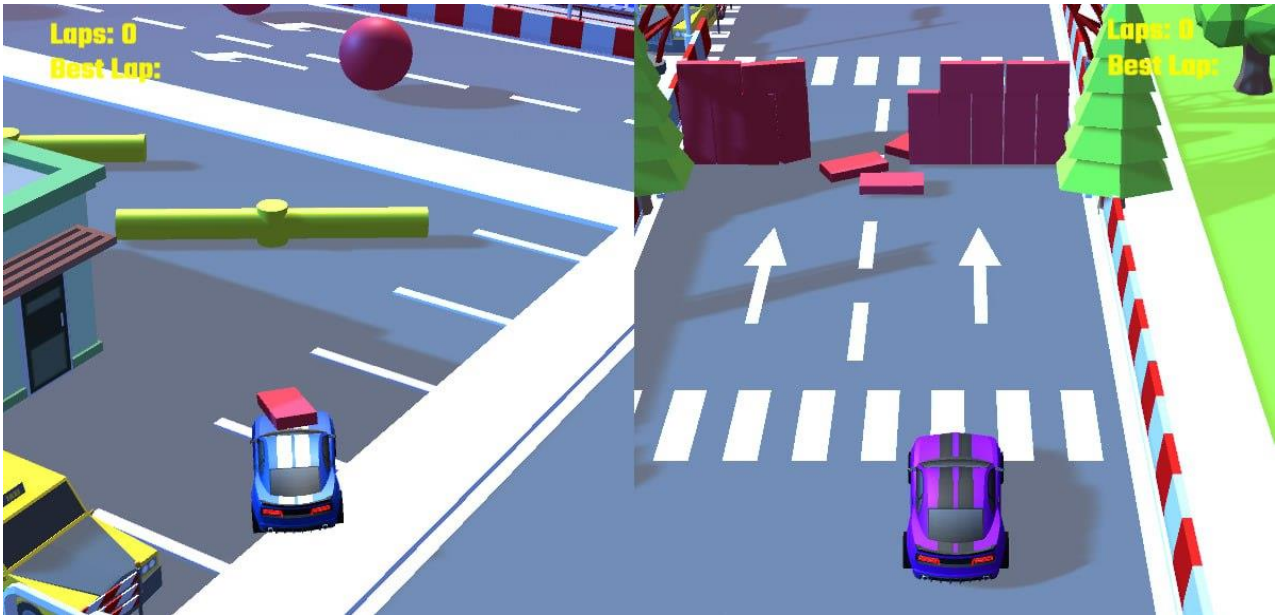


Рис. 3.21. Геймплей



### 3.5. Тестування гри

Під час тестування гри, основними аспектами було виявлення помилок, а саме таких помилок як:

- Прозорі колайдери: виявлення областей, де колайдери невидимі або неправильно працюють, що може призводити до проблем з фізикою гри.

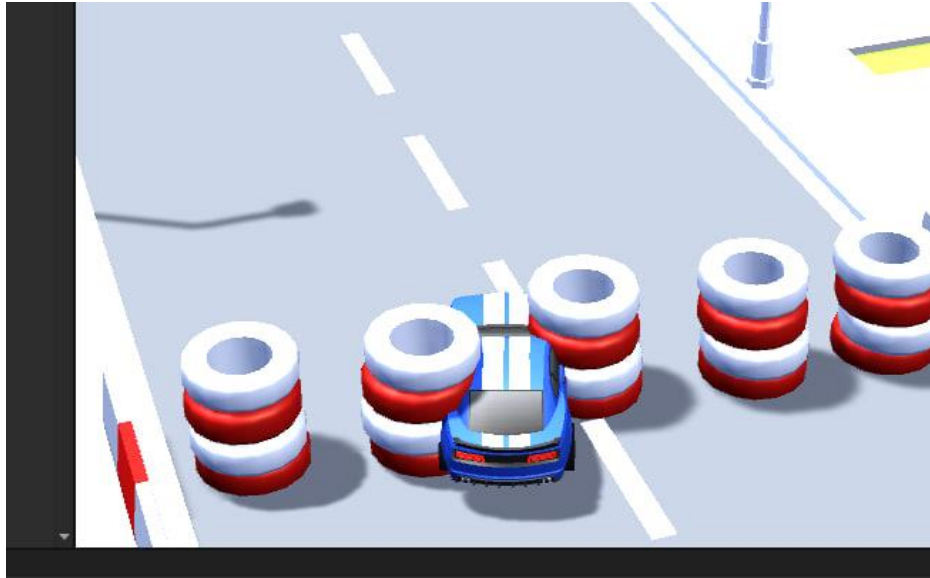


Рис. 3.22. Проблема з колайдером

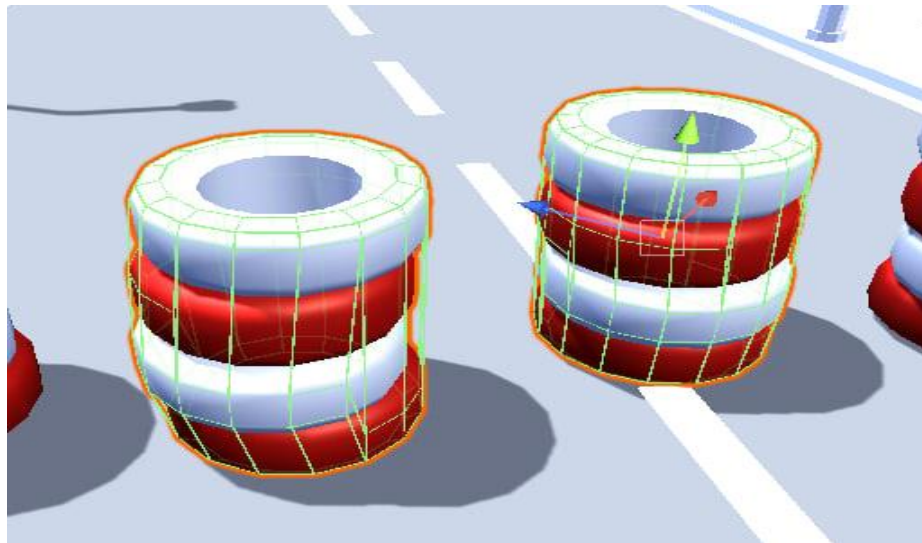


Рис. 3.21. Виправленні колайдери та межі між трасою, та декораціями

- Зміна розположення камери, що вона не заважала обзору гравцям;

```
public class CameraFollow : MonoBehaviour
{
    public Transform carTransform;
    [Range(1, 10)]
    public float followSpeed = 2;
    [Range(1, 10)]
    public float lookSpeed = 5;
    public float heightOffset = 5.0f; // Додаткове налаштування по висоті
    public float distanceOffset = 10.0f; // Додаткове налаштування зміщення по дистанції
    [Range(-10, 10)]
    public float tiltAngle = 2.0f; // Кут нахилу вперед

    Vector3 initialCameraPosition;
    Vector3 initialCarPosition;
    Vector3 absoluteInitCameraPosition;
}
```

Рис. 3.23. Змінні для зручного налаштування камер

- Оптимізація: поліпшення продуктивності гри, зменшення лагів та збоїв.
- Зміна інтерфейсу: редизайн користувацького інтерфейсу для покращення взаємодії гравців з грою, забезпечення інтуїтивного розташування елементів.
- Налаштування візуальних ефектів.

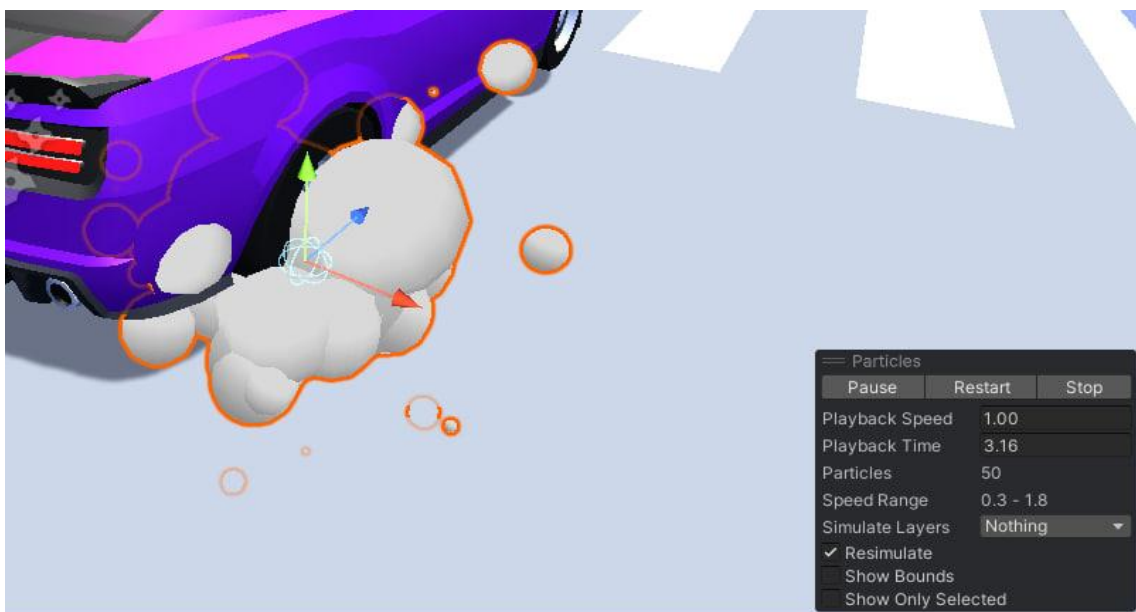


Рис. 3.24. Анімація диму

## ВИСНОВКИ ПО РОЗДІЛУ 3

У третьому розділі було здійснено проектування та тестування гоночної гри, в якій було реалізовано основні аспекти гри: Сюди відноситься:

1. Розробку концепції, яка охоплювала визначення головної ідеї гри гоночна аркадна гра з елементами дрифту для двох гравців з основними та необхідними елементами.
2. Створення сцен, це етап який включав розробку трьох основних сцен: головного меню, основного треку для гонок та сцени лоббі для ознайомлення з геймплеєм.
3. Обробка об'єктів передбачала додавання необхідних колайдерів для всіх ігрових об'єктів, включаючи автомобілі, перешкоди, декорації. Завдяки Mesh Collider було легко визначити границі тіл та реалізувати їх взаємодію.
4. Етап тестування гри який включав виявлення та виправлення помилок, таких як прозорі колайдери, межі між трасою та декораціями, налаштування розташування камери та оптимізація гри для забезпечення плавного ігрового процесу.

Загалом розробка гри “Drift Duel” завершилася успішною реалізацією всіх необхідних аспектів, які біло вказано в плані роботи. Гра була протестована та оптимізована, що забезпечило її високу якість та готовність до використання гравцями.

## ВИСНОВКИ

Під час роботи було обговорені такі теми як: аналіз ринку, технологій та механіки Unity, мережева взаємодія, а також проектування і тестування гри.

Ринок комп'ютерних ігор є одним з найбільш динамічних і швидкозростаючих. Вартість глобального ринку відеоігор продовжує зростати завдяки розвитку хмарних технологій. Сегмент комп'ютерних ігор є найбільш прибутковим, а консольний ринок демонструє стабільне зростання.

У роботі було розглянуто питання такі як:

- Вплив відеоігор на людину, їх значення у сучасності,
- Було розглянуто напрямки для 3D-додатків в Unity, а саме маркетингу, бізнесу, додатки для навчання, відеоігри
- Розглянуто технології для мультиплеєра: мережеві протоколи, синхронізація станів, архітектура серверів, інфраструктура сервера.

Що до розробки відеогри, вона відбувалася на платформі Unity, мовою C#, то основними етапами проекту були:

- Розробка головного меню, де гравці можуть легко переходити між сценами, та за потреби подивитися схему управління.
- Створення основної сцени, де проходив геймплей.
- Розміщення моделей на основній сцені.
- Написання коду для керування автомобілями.
- Написання скрипта для колес машини, щоб вони залишали полоси на трасі.
- Деталізована модель машини, компонентів колес для появи полос на трасі.
- Виставлення перешкод на трасі та надання їм фізичної форми через колайдери.
- Додавання чекпоінтів старт-фініш та скрипта для них, для того, щоб відслідковувати кількість пройдених кругів у кожного з гравців.
- Створення лоббі – щоб удосконалити навички дрифту без перешкод.

- Тестування гри, де було виправлено настройки камер та колайдерів. Важливо враховувати всі аспекти, від програмування до дизайну та звукового оформлення.

В результаті роботи було отримано важливі наукові та практичні результати. Наукові результати включають детальний аналіз ринку відеоігор, що підтверджується числовими показниками зростання ринку та його сегментів. Вивчено вплив відеоігор на когнітивні здібності людини, що сприяє розвитку критичного мислення та комунікативних навичок. Технологічно було досліджено та впроваджено сучасні мережеві протоколи та архітектуру серверів для забезпечення стабільної та ефективної мультиплеєрної взаємодії.

Практичні результати включають успішну розробку відеогри на платформі Unity з використанням мови програмування C#. В процесі розробки були застосовані інноваційні підходи до створення ігрових механік, фізичної взаємодії та користувацького інтерфейсу. Ці результати можуть бути використані для подальшого розвитку відеоігор, впровадження нових методик навчання через гейміфікацію та розширення можливостей соціальної взаємодії у віртуальних середовищах. Отримані результати також можуть бути основою для розробки нових інструментів та технологій у сфері геймдева, що дозволить ще більше розширити межі можливого в цій індустрії.

Таким чином, індустрія відеоігор переживає етап бурхливого розвитку, стаючи важливим елементом сучасного суспільства. Ігри не тільки розважають, але й сприяють розвитку критичного мислення, комунікативних навичок та креативності. Мультиплеєрні ігри відкривають нові можливості для соціальної взаємодії та створення віртуальних спільнот. Розробка ігор на платформах, таких як Unity 3D, дозволяє створювати високоякісні продукти, які відповідають вимогам сучасного ринку. З кожним новим проектом геймдев розширює свої межі, стаючи не лише розважальною індустрією, але й важливим культурним явищем.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chen C, Leung L (2019) An exploratory study linking psychological factors to mobile social game addiction. *Telemat Inf* 33(4 1114-1116).
2. Kamadi VV, Allam AR, Thummala SM (2020) A computational intelligence technique for the effective diagnosis of diabetic patients using principal component analysis (PCA) and modified fuzzy SLIQ decision tree approach. *Appl Soft Comput* 49:137–140.
3. Laureanti, J.; Brandi, J.; Offor, E.; Engel, D.; Rallo, R.; Ginovska, B.; Martinez, X.; Baaden, M.; Baker, N. A. Visualizing Biomolecular Electrostatics in Virtual Reality with UnityMol-APBS. *Protein Sci.* 2020, 29 (1), 237– 238.
4. Doak, D. G.; Denyer, G. S.; Gerrard, J. A.; Mackay, J. P.; Allison, J. R. Peppy: A Virtual Reality Environment for Exploring the Principles of Polypeptide Structure. *Protein Sci.* 2020, p. 29.
5. "Mastering Unity 2019 Game Development with C#" by Alan Thorn: This book explores advanced topics in Unity, including optimization, advanced scripting, and the use of Unity's more sophisticated features.
6. Perry, R., Drachen, A., Kearney, A., Kriglstein, S., Nacke, L. E., Sifa, R., et al. (2018). 79, 202–203.
7. Kircaburun, K., Pontes, H. M., Stavropoulos, V., and Griffiths, M. D. (2020). *Curr. Opin. Psychol.* 36, 38–43.
8. "Impact of Video Games on Society." *UKEssays.*, 2018, p. 22-25.
9. Integrating augmented reality into problem based learning: The effects on learning achievement and attitude in physics education. *Computers & Education* (2019) 69-70.
10. Bibic, L.; Druskis, J.; Walpole, S.; Angulo, J.; Stokes, L. Bug Off Pain: An Educational Virtual Reality Game on Spider Venoms and Chronic Pain for Public Engagement. *J. Chem. Educ.* 2019, 96-97, 1486-1488.

11. Hanson, J.; Andersen, P.; Dunn, P. K. Effectiveness of Three-Dimensional Visualisation on Undergraduate Nursing and Midwifery Students' Knowledge and Achievement in Pharmacology: A Mixed Methods Study. *Nurse Educ. Today* 2019, 81, 19– 25.
12. Мультиплеєрна гра віртуальній реальності [Електронний ресурс] - <https://pubs.acs.org/doi/full/10.1021/acs.jchemed.0c00866> (дата звернення 29.05.24)
13. Інструменти критичної оцінки досліджень [Електронний ресурс] - <https://joannabriggs.org/critical-appraisal-tools> (дата звернення 29.05.24)
14. Інструменти критичної оцінки досліджень [Електронний ресурс] - <https://www.redbull.com/us-en/history-of-video-gamesdesign#:~:text=01,The%20Birth%20of%20Video%20Games,%2Dtac%2Dtoe%20or%20Nim.> (дата звернення 29.05.24)
15. Народження та еволюція відеоігор [Електронний ресурс] - <https://www.history.com/topics/inventions/history-of-video-games> (дата звернення 29.05.24)
16. Історія відеоігор [Електронний ресурс] - <https://www.bnl.gov/about/history/firstvideo.php> (дата звернення 29.05.24)
17. Перша відеогра та її історичне значення [Електронний ресурс] - <https://www.jstor.org/stable/1354423> (дата звернення 29.05.24)
18. Unity та наука про розробку ігор [Електронний ресурс] - <https://www.taylorfrancis.com/chapters/edit/10.4324/9781315650647-21/unity-science-jordi-cat> (дата звернення 29.05.24)
19. Найкращі ігри, розроблені на Unity [Електронний ресурс] - <https://itvdn.com/ua/blog/article/7best-unity-games> (дата звернення 29.05.24)
20. Звіт про Unity Multiplayer 2022 [Електронний ресурс] – <https://gamedev.dou.ua/news/multiplayer-2022-unity-report/> (дата звернення 29.05.24)