

Міністерство освіти і науки України  
Національний технічний університет  
«Дніпровська політехніка»  
Навчально-науковий  
інститут  
електроенергетики  
(інститут)  
Факультет інформаційних технологій  
(факультет)  
Кафедра інформаційних технологій та комп'ютерної інженерії  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**кваліфікаційної роботи ступеня бакалавра**

студента Малова Максима Юрійовича  
(ПІБ)

академічної групи 123-20-2  
(шифр)

спеціальності 123 Комп'ютерна інженерія  
(код і назва спеціальності)

за освітньо-професійною програмою 123 Комп'ютерна інженерія  
(офіційна назва)

на тему “Розробка кіберфізичної системи школи екстремального водіння”  
(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц.Соколова Н.О.			
спеціальної частини	доц.Соколова Н.О.			
розділів:				
розробка апаратної частини	доц. Ткаченко С.М.			
розробка корпоративної мережі	ас. Бешта Л..В.			

Рецензент	доц.Клименко А.В.			
Нормоконтролер	проф. Цвіркун Л.І.			

Дніпро  
2024

**ЗАТВЕРДЖЕНО:**

завідувач кафедри  
інформаційних технологій  
та комп'ютерної інженерії  
(повна назва)

\_\_\_\_\_ Гнатушенко В.В.  
(підпис) (прізвище, ініціали)  
"25" січня 2024 року

**ЗАВДАННЯ**

**на кваліфікаційну роботу  
ступеня бакалавр**

студента Малова М.Ю. академічної групи 123-20-1  
(прізвище та ініціали) (шифр)

спеціальності 123 «Комп'ютерна інженерія»

за освітньо-професійною програмою 123 «Комп'ютерна інженерія»

(офіційна назва)

на тему “Розробка кіберфізичної системи школи екстремального водіння”

затверджену наказом ректора НТУ «Дніпровська політехніка» від 23.05.2024 №  
469-с

Розділ	Зміст	Термін виконання
Стан питання та постановка завдання	На основі матеріалів виробничих практик, інших науково-технічних джерел конкретизується предмет та мету роботи та виконується постановка завдання	20.05.2024
Розробка апаратної частини	На основі аналізу підприємства формулюються технічні вимоги до комп'ютерної системи та розробляється апаратна частина системи	31.05.2024
Розробка корпоративної мережі	Виконується розрахунок налаштувань корпоративної мережі та перевірка роботи системи, розробляються методи та налаштування обладнання для захисту інформації в системі	05.06.2024
Розробка компонента системи	Виконується детальна розробка компонента системи	10.06.2024

**Завдання видано**

\_\_\_\_\_ (підпис керівника)

доц.Соколова Н.О.

(прізвище, ініціали)

**Дата видачі** 25.01.2024

**Дата подання до екзаменаційної комісії** 24.06.2024

**Прийнято до виконання**

\_\_\_\_\_

Малов М.Ю.

## РЕФЕРАТ

Пояснювальна записка: с.70, рис.39, табл.7, дод.1, джерел 7

КОМП'ЮТЕРНА МЕРЕЖА, ШКОЛА ЕКСТРЕМАЛЬНОГО ВОДІННЯ,  
КОМП'ЮТЕРНА СИСТЕМА, CISCO, СИМУЛЯТОР, МОБІЛЬНИЙ  
ЗАСТОСУНОК.

**Об'єкт розробки** – комп'ютерна модель кіберфізичної системи школи екстремального водіння із опрацюванням розробки мобільного симулятора.

**Мета роботи** – створення кіберфізичної системи для навчання водіїв елементам контраварійного керування автомобілем із ігровими елементами.

Було здійснено розробку комп'ютерної системи для школи екстремального водіння «ProDriver». Для цього було використано сучасне обладнання, яке відповідає всесвітнім стандартам якості та продуктивності. Комп'ютерна мережа має налаштовані протоколи безпеки та має великий потенціал до розширення. Мобільний симулятор має цікавий геймплей та приємний візуальний вигляд. Також була створена реалістична фізика поведінки автомобіля.

Комп'ютерна система має потенціал для модернізації окремих апаратних компонентів мережі та встановленого програмного забезпечення з метою досягнення наступних результатів:

- підвищення швидкості передачі даних;
- збільшення продуктивності робочих станцій і системи загалом;
- розширення філіалів та збільшення прибутку від компанії;
- покращення безпеки комп'ютерної мережі.

Створена комп'ютерна мережа відповідає завданню кваліфікаційної роботи бакалавра. Програмне забезпечення Cisco Packet Tracer було використано для тестування роботи комп'ютерної системи з використанням моделі топології корпоративної мережі.

## ЗМІСТ

Перелік скорочень, умовних познач, одиниць і термінів .....	7
Вступ.....	8
1 Стан питання та постановка завдання .....	9
1.1 Стисла характеристика галузі та умов застосування кіберфізичної системи школи екстремального водіння.....	9
1.2 Характеристика і структура школи екстремального водіння .....	10
1.3 Стислі відомості про технології збору й передачі даних та топологічна схема розміщення структурних підрозділів школи «ProDriver»	13
1.4 Принципи та технічні способи інформаційного забезпечення школи «ProDriver».....	16
1.5 Аналітичний огляд існуючих способів обробки та передачі інформації, принципів побудови об'єкта проєктування, відомих рішень у галузі проєктування шкіл водіння .....	18
1.6 Завдання і мета роботи .....	19
1.7 Визначення можливих напрямків рішення поставлених завдань .	20
2 Розробка апаратної частини кіберфізичної .....	21
2.1 Технічні вимоги до кіберфізичної системи школи екстремального водіння «ProDriver».....	21
2.1.1 Вимоги до системи в цілому.....	21
2.1.1.1 Вимоги до структури і функціонування системи школи екстремального водіння «ProDriver».....	21
2.1.1.1.1 Перелік підсистем, їх призначення й основні характеристики.....	21
2.1.1.1.2 Вимоги до способів і засобів зв'язку для інформаційного обміну між компонентами системи.....	22
2.1.1.1.3 Вимоги до характеристик взаємозв'язків створюваної системи із суміжними системами.....	22
2.1.1.1.4 Вимоги до режимів функціонування системи .....	23
2.1.1.1.5 Вимоги до діагностування системи .....	24
2.1.1.1.6 Перспективи розвитку, модернізації системи.....	24

2.1.1.2	Показники призначення .....	25
2.1.1.3	Вимоги до експлуатації, технічного обслуговування, ремонту і збереження компонентів системи школи екстремального водіння «ProDriver».....	26
2.1.1.3.1	Умови і регламент експлуатації, що повинні забезпечувати використання технічних засобів системи.....	26
2.1.1.3.2	Вимоги до кількості, кваліфікації обслуговуючого персоналу і режимам його роботи.....	27
2.1.1.3.3	Вимоги до регламенту обслуговування.....	27
2.1.1.4	Вимоги до патентної чистоти .....	28
2.1.1.5	Додаткові вимоги .....	28
2.1.1.5.1	Вимоги до розробки компонента системи .....	28
2.1.2	Вимоги до задач (налаштувань), які виконуються у комп'ютерній системі .....	28
2.2	Розробка інженерних рішень і комплексу технічних засобів .....	29
2.2.1	Вибір і обґрунтування структурної схеми комплексу технічних засобів комп'ютерної системи шляхом узгодження структури з топологічними особливостями об'єкту розробки.....	29
2.2.2	Розробка специфікацій апаратних засобів комп'ютерної системи .....	30
3	Розробка корпоративної мережі підприємства .....	36
3.1	Розрахунок схеми адресації корпоративної мереж .....	36
3.2	Розробка топологічної схеми корпоративної мережі.....	38
3.3	Налаштування мережі.....	40
3.3.1	Базове налаштування.....	40
3.3.2	Налаштування маршрутизації корпоративної мережі .....	42
3.3.3	Налаштування NAT .....	44
3.3.4	Налаштування VPN .....	46
3.3.5	Налаштування VLAN .....	47
3.4	Перевірка роботи комп'ютерної системи .....	48
4	Розробка компонента системи .....	52
4.1	Опис застосунку .....	52
4.1.1	Загальні відомості про застосунок .....	52

4.1.2 Платформа Unity .....	53
4.2 Розробка проекту.....	54
4.2.1 Архітектура гри.....	54
4.2.2 Автомобіль.....	57
4.2.2.1 Скрипт Car Controller.....	58
4.2.3 Система оцінки дій користувача .....	63
4.2.4 Вибір та наповнення мап .....	64
4.2.5 Інтерфейс користувача .....	66
Висновки .....	69
Перелік джерел посилань.....	70
Додаток А.....	71

**ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАК, ОДИНИЦЬ І ТЕРМІНІВ**

ДТП – дорожно-транспортна пригода, аварія;

ІТ – інформаційні технології;

ПК – персональний комп'ютер;

ISP – компанія-постачальник Інтернет-послуг;

LAN – локальна мережа;

LMS – система управління навчальним процесом;

DHCP – протокол динамічного розподілу адрес вузлам;

HTTP – протокол передачі гіпертексту;

VLAN – віртуальна локальна мережа;

OSPF – протокол динамічної маршрутизації;

VPN – віртуальна приватна мережа;

КМ – корпоративна мережа;

КС – комп'ютерна система;

JSON – текстовий формат обміну даними між комп'ютерами;

MAC-адреса – унікальний ідентифікатор пристрою.

## ВСТУП

У сучасному світі безпека дорожнього руху є однією з найважливіших проблем, з якими стикаються суспільство та держави. Підвищення рівня підготовки водіїв, особливо у екстремальних умовах, є ключовим фактором для зменшення аварійності на дорогах. У цьому контексті значну роль відіграють школи екстремального водіння, які надають водіям можливість отримати практичні навички керування автомобілем у складних ситуаціях.

Метою даної дипломної роботи є розробка кіберфізичної системи для школи екстремального водіння. Кіберфізичні системи об'єднують фізичні процеси з комп'ютерними алгоритмами і мережевими технологіями, що дозволяє створювати інтерактивні, високоефективні та безпечні навчальні середовища. Така система не лише покращить якість навчання, але й сприятиме більш глибокому розумінню водіями поведінки автомобіля в екстремальних умовах.

Для створення та тестування системи були використані передові технології в галузі кіберфізичних систем, зокрема, платформи для розробки симуляційного програмного забезпечення та мережеві рішення для інтеграції різних компонентів системи. Це дозволило створити комплексну систему, яка відповідає всім сучасним вимогам до навчання екстремальному водінню.

У пояснювальній записці та додатках до неї детально описані етапи розробки та впровадження кіберфізичної системи, а також результати її тестування. Це дозволяє чітко зрозуміти процес створення системи та оцінити її ефективність у навчанні водіїв.

Таким чином, дана кваліфікаційна робота є важливим кроком у підвищенні рівня безпеки дорожнього руху шляхом удосконалення методів навчання екстремальному водінню, що сприятиме зменшенню аварійності та підвищенню культури керування автомобілем.



## 1 СТАН ПИТАННЯ ТА ПОСТАНОВКА ЗАВДАННЯ

### 1.1 Стисла характеристика галузі та умов застосування кіберфізичної системи школи екстремального водіння

Об'єктом проектування даної кваліфікаційної роботи є розробка кіберфізичної системи школи екстремального водіння «ProDriver», що відноситься до галузі інформаційних технологій.

ІТ сфера – це галузь, яка відповідає за збір, зберігання і передачу інформації за допомогою технічних пристроїв і, в цілому, за спілкування людей на відстані. Це дуже велика ніша, яка охоплює дуже багато складових життя, має дуже багато різних професій та відгалуджень. У даній роботі розглядається розробка кіберфізичної системи школи екстремального водіння, а саме побудова комп'ютерної мережі та розробка мобільного застосунку для навчання.

Існує велика кількість відомих та якісних автошкіл, які навчають як теоретичних навичок, так і навичок практичного керування автомобілів. Станом на 5 травня 2023 року, в Україні налічувались 1753 акредитовані заклади з підготовки водіїв[1]. Серед найвідоміших у Дніпропетровській області можна виділити «Проспект», «Антарес», «Автошкола Коваленко». Ці автошколи підготовлюють водіїв категорій А, А1, В, С1, С, С1Е, СЕ, В1, ВЕ, D1, D.

Але переважна більшість автошкіл навчають саме базовим знанням водіння автомобіля, тобто випускники цих шкіл мають мінімальний набір навичок, необхідних для керування. Далі, із плином часу, водії самі набираються досвіду, вже управляючи транспортним засобом безпосередньо в містах. Але при потраплянні у ситуації, де потрібна швидка реакція та чіткість дій, вони можуть губитись та панікувати, через що є вірогідність потрапити у ДТП.

Окрім того, ті люди, які прагнуть розвиватися в автоспорті, але не мають знань і принципів керування автомобілем на треку, стикаються з численними

викликами. Без розуміння траєкторії, правильного гальмування та набору швидкості, а також без знання правил поведінки на треку, навіть при наявності коштів на добре налаштований і збалансований спортивний автомобіль, вони можуть зіткнутися з серйозними проблемами. При виїзді на трек такі водії у кращому випадку почуватимуться безпорадними та рухатимуться дуже повільно. У гіршому випадку вони можуть спричинити дорожньо-транспортну пригоду, що призведе до значних фінансових витрат на відновлення автомобіля та можливих травм. Таким чином, знання та практика є надзвичайно важливими для успішного та безпечного розвитку в автоспорті. Без відповідної підготовки, навіть найдорожчий болід не гарантує безпеки та успіху на треку. Це підкреслює важливість спеціалізованих шкіл екстремального водіння, які можуть надати необхідні знання та навички.

На даний момент це доволі вузька ніша, але при цьому має попит, цікава різній аудиторії та має великий потенціал для розвитку. Автошкола «ProDriver» і буде спеціалізуватись на навчанні контраварійного та екстремального керування: керування під час ожеледиці та під час темної пори доби, даватиме змогу кожній людині спробувати свої навички в автоспорті.

## **1.2 Характеристика і структура школи екстремального водіння**

Школа «ProDriver» є передовою у своїх технологіях та рішеннях і має великий асортимент послуг.

Основні види послуг автошколи:

- навчання теоретичної та практичної частини контраварійного керування;
- навчання основам спортивних дисциплін, таких як кільцеві гонки та дрифт;
- навчання на автосимуляторі.

Заняття із теорії будуть проводитись у класі в навчальному центрі, автосимулятори будуть знаходитись в віртуальному класі. Для практичної частини та відпрацювань теоретичних знань у реальному житті будуть заїзди на полігоні, під наглядом інструктора та відеокамер.

Навчання контраварійного керування буде відбуватись таким чином, спочатку буде набрана група із учнів, яким буде йти начитка теоретичного курсу, пояснення основ керування при поганій видимості, на слизькому покритті, і тп. Також завдяки проєктору буде розбір помилок та дії, щоб їх уникнути, на існуючих відео учнів на полігоні, розбір найпопулярніших ДТП у містах. Після цього, учні будуть допущені до керування транспортним засобом на полігоні, щоб, разом із інструктором, відпрацювати свої навички у заїзді на естакаду під час ожеледі, виходу та реакції під час недостатньої та надмірної повертаємості автомобіля(заносу), реакції на неочікувану перешкоду, та інше.

Навчання спортивних дисциплін буде проходити за схожою методикою, але буде викладатись саме принципи гальмування, методи проходження повороту, правильність траєкторії, особливості керування автомобілів із різним типом приводу, правила змагань у потрібних дисциплінах, теоретичні особливості керування потужними автомобілями, тощо. Далі учні матимуть змогу відпрацювати окремі частини проїзду треку на полігоні.

Також клієнти зможуть попрактикуватись на автосимуляторах, що дозволить зберегти час, симулювати конкретні аварійні ситуації, спробувати різницю керування авто із різним типом приводу та потужності, та дозволить учням позмагатись між собою на окремих треках за швидкість та правильність проходження траси.

На рисунку 1.1 зображена організаційна структура автошколи.

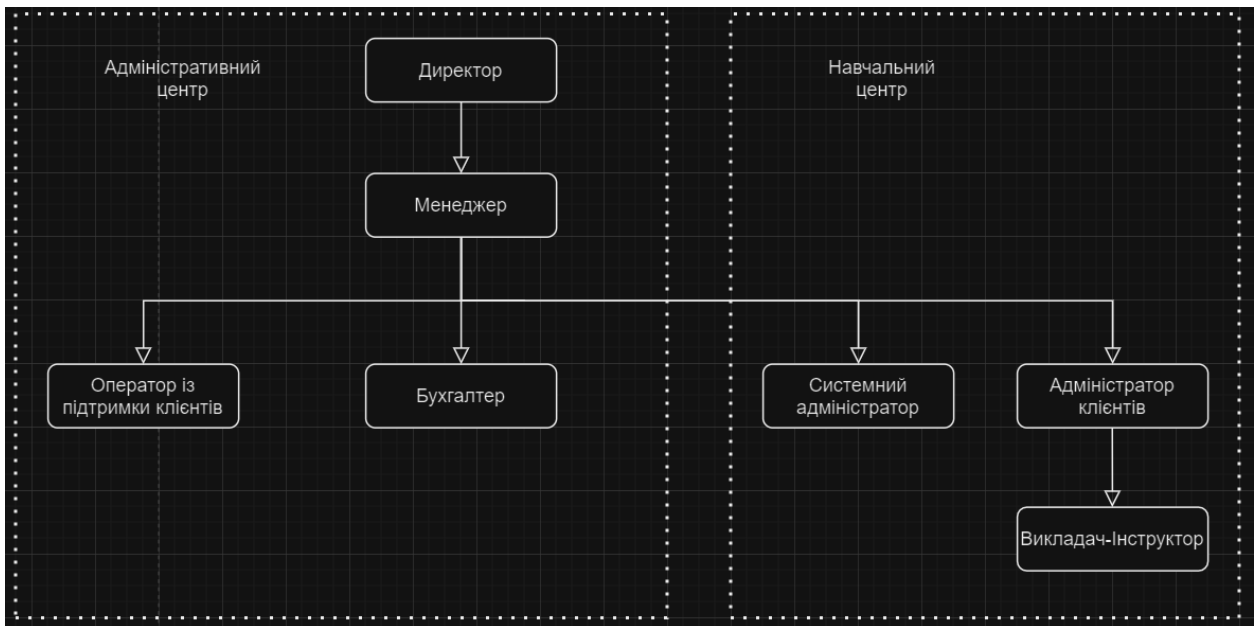


Рисунок 1.1 – Структура компанії «ProDriver»

Штат компанії (табл.1.1) включає в себе директора, менеджера, оператора із підтримки клієнтів, системного адміністратора, бухгалтерії, адміністраторів та викладачів.

Таблиця 1.1 – Штат автошколи

Посада	Кількість штатних одиниць
Директор	1
Менеджер	1
Оператор із підтримки клієнтів	1-3
Системний адміністратор	1
Бухгалтер	1-3
Адміністратор по роботі з клієнтами	2
Викладач	6

Директор компанії керує всією діяльністю автошколи, визначає стратегічні напрямки розвитку та приймає ключові рішення щодо управління компанією.

Менеджер організовує і координує роботу команди, забезпечує реалізацію навчальних програм та відповідає за взаємодію з партнерами і постачальниками.

Відповідає за обробку запитів клієнтів оператор із підтримки клієнтів, він надає інформацію про послуги автошколи та допомагає вирішувати проблеми клієнтів.

Системний адміністратор забезпечує безперебійну роботу комп'ютерних систем і мереж автошколи, проводить технічне обслуговування та вирішує технічні проблеми.

Бухгалтер веде фінансову звітність, обліковує доходи та витрати компанії, займається податковими питаннями та контролює фінансові операції.

Адміністратор по роботі з клієнтами: координує запис і розклад занять, веде базу даних клієнтів, здійснює комунікацію з учнями щодо організаційних питань.

Викладачі проводять теоретичні та практичні заняття з водіння, навчають учнів навичкам екстремального водіння та надають індивідуальні консультації.

### **1.3 Стислі відомості про технології збору й передачі даних та топологічна схема розміщення структурних підрозділів школи «ProDriver»**

Автошкола екстремального водіння «ProDriver» використовує сучасні технології збору і передачі даних для забезпечення ефективної роботи та взаємодії між підрозділами. Для цього розгорнута корпоративна мережа із використанням власних серверів, а також налаштований вихід до мережі

інтернет. Для передачі даних між автомобілями, викладачами та центральною базою даних використовуються Wi-Fi та мобільний інтернет, що забезпечує постійний зв'язок та доступ до необхідної інформації.

Задля збору даних використовується обладнання на тестовому полігоні, а саме відеокамери, встановлені як на периметрі площадки, так і в самому автомобілі. Також, як варіант подальшого розвитку автошколи, машини можуть обладнуватись датчиками та телеметричними системами, які вимірюють та передають дані про прискорення, швидкість, траєкторію авто, натискання педалей та положення важеля коробки передач.

Офіс знаходиться у центрі міста Дніпро, і займає два поверхи, і розділений на навчальний центр, що знаходиться на першому поверсі, та адміністративний офіс, що займає другий поверх. Навчальний полігон побудований на місці поля, за межею міста. Адреса розташування зображена на рисунку 1.2.

- полігон: 93Q5+F5R Дніпро, Дніпропетровська область;
- офіс: просп. Лесі Українки, 77, Дніпро, Дніпропетровська область, 49000.

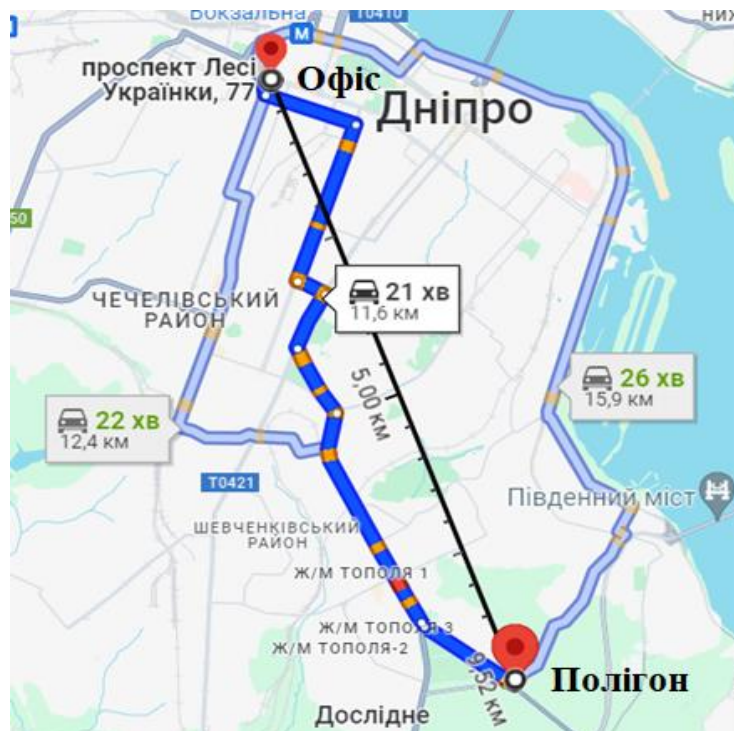


Рисунок 1.2 – Адреса розташування філіалів

Офіс розташований у центрі міста, що дозволяє легко до нього добиратись, але при цьому не в центральному районі, що дає вигоду у ціні аренди приміщень. Полігон же побудований у віддаленому місці, що дозволяє не заважати людям гучними звуками автівок, та забезпечує водіїв від відволікаючих факторів. На рисунках 1.3 – 1.4 зображено плани адміністративних та навчальних приміщень.



Рисунок 1.2 – План навчальної частини офісу на 1 поверсі

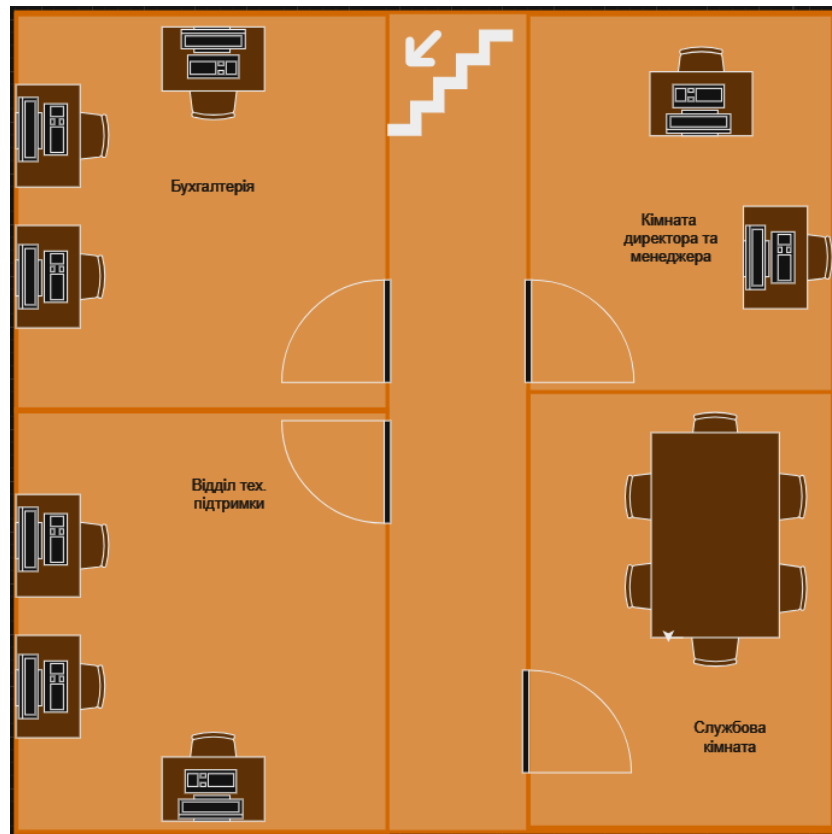


Рисунок 1.3 – План адміністративного офісу на 2 поверсі

#### 1.4 Принципи та технічні способи інформаційного забезпечення школи «ProDriver»

Інформаційне забезпечення «ProDriver» для автошколи екстремального водіння базується на ідеї постійного доступу до інформації, зручності та безпеки. Це досягається за допомогою використання сучасних інформаційних технологій і програмного забезпечення.

Комп'ютерна мережа автошколи, яка складається з робочих станцій і серверів, повинна забезпечувати швидкий і надійний обмін даними між усіма підрозділами та дозволяти керівництву, викладачам, студентам і технічному персоналу отримати доступ до необхідної інформації. При розробці комп'ютерної мережі необхідно врахувати, що буде 2 офіси в одній будівлі та полігон. Потрібно встановити з'єднання та під'єднати їх до мережі Інтернет. Також необхідно забезпечити достатній захист, так як це важливий елемент мережі.



Комп'ютери у адміністративному центрі повинні мати такі характеристики, щоб на них можна було запуснути базові програми без особливих навантажень. ПК у навчальному центрі ж, повинні запускати симулятори, тому необхідно, щоб вони мали набагато вищі характеристики. Усі комп'ютери мають бути оснащені операційною системою Windows 10.

У забезпеченні безперебійної роботи комп'ютерних систем і мереж системні адміністратори виконують важливу функцію. Вони обслуговують систему, вирішують технічні проблеми та захищають дані від несанкціонованого доступу та збоїв.

У процесі навчання студентів використовується комплексний підхід, який охоплює теоретичні аспекти та практичні заняття. Теоретичні заняття проводяться за допомогою мультимедійних інструментів, які роблять навчання більш наочним і ефективним. Студенти можуть навчатися дистанційно контраварійному керуванню автомобілем за допомогою мобільного додатку.

Комплексний підхід використовується для навчання студентів, який включає теоретичні знання та практичні заняття. Теоретичні заняття проводяться за допомогою мультимедійних інструментів, які роблять навчання більш наочним і ефективним. Мобільний додаток для дистанційного навчання контраварійному керуванню автомобілем дозволяє учням вивчати основні правила, моделювати аварійні ситуації та отримувати оцінку своїх дій.

Учні можуть практикувати свої навички керування в різних умовах за допомогою симуляторів. Автосимулятори дозволяють учням змагатися між собою на окремих треках за правильність проходження траси та швидкість, моделюючи різні аварійні ситуації.

## **1.5 Аналітичний огляд існуючих способів обробки та передачі інформації, принципів побудови об'єкта проєктування, відомих рішень у галузі проєктування шкіл водіння**

Системи управління навчальним процесом (LMS) є сучасними методами обробки інформації в автошколах. LMS дозволяють зберігати, обробляти та аналізувати дані про студентів, викладачів і навчальні матеріали в одному місці. LMS автоматизують багато адміністративних завдань, таких як облік успішності, планування занять, реєстрація студентів і зворотний зв'язок [2].

Для передачі даних використовуються різні канали, такі як електронна пошта, мобільні додатки, вебпортали та соціальні мережі. Це дозволяє швидко та легко отримати доступ до навчальних матеріалів, оновлень розкладу та інших важливих повідомлень для викладачів і студентів.

У концепції об'єкта проєктування автошколи є кілька основних елементів. По-перше, гнучкість системи, яка полегшує швидке розширення та оновлення функціональності. По-друге, безпека даних, яка гарантує, що особисті дані студентів будуть захищені.

Використання спеціалізованих програмних комплексів, які поєднують теоретичне навчання та практичні заняття, часто є відомим способом проєктування шкіл водіння. Наприклад, автосимулятори є важливою частиною сучасних автошкіл, оскільки вони дозволяють учням навчатися керувати машиною в безпечному середовищі. Для прикладу, завдяки грі «Gran Turismo», та програмі «GT Academy», у кожного із 25 000 віртуальних гонщиків, було відібрано 8 найшвидших, які мали змогу змагатись у фіналі на реальних автомобілях. Переможець, іспанець Лукас Ордоньез отримав професійний контракт із Ніссан та представляв команду протягом 10 років[3].

Оскільки вони дозволяють учням вивчати теорію в зручній для них час і місце, мобільні додатки та онлайн-платформи для дистанційного навчання стають все більш популярними [4]. Це особливо важливо для людей, які не можуть регулярно відвідувати заняття в класі.

Використання віртуальної та доповненої реальності для моделювання складних дорожніх ситуацій, що покращує розуміння та відпрацювання необхідних навичок, є ще одним інноваційним рішенням. Автошколи можуть оптимізувати навчальні програми, покращувати їх ефективність і адаптувати їх до потреб студентів за допомогою великих даних і аналітичних інструментів.

### **1.6 Завдання і мета роботи**

Мета даної роботи створення корпоративної комп'ютерної мережі школи екстремального водіння «ProDriver» та розробка мобільного застосунку для навчання контраварійного керування.

Для досягнення мети кваліфікаційної роботи необхідно виконати наступні завдання:

- Розробити архітектуру та топологію корпоративної мережі;
- Розробити схему комплексу технічних засобів;
- Вибрати апаратне забезпечення для корпоративної мережі
- Оптимізувати передачу даних за допомогою мережевого обладнання;
- Змінити адресацію кінцевих пристроїв;
- Створити програму-симулятор для навчання водіння.

Для успішного виконання цих завдань необхідно ретельно вивчити потреби та бажання клієнта. Після цього можна приступати до проєктування та вибору частин мережі. При побудові та налаштуванні мережі також слід забезпечити можливість легкого масштабування, якщо це потрібно.

## 1.7 Визначення можливих напрямків рішення поставлених завдань

Першим етапом є вибір мережевої архітектури, що є критично важливим у процесі проектування корпоративної мережі. Від правильного вибору залежить ефективність, надійність та масштабованість мережі. У даній роботі була вибрана модель Ethernet через її популярність та відповідність необхідним вимогам. Топологія мережі була задана замовником і модифікована згідно з поставленими задачами.

Для створення комплексу технічних засобів необхідно обрати компанію, яка їх надаватиме. У цій роботі буде використовуватися обладнання компанії Cisco через її популярність, великий об'єм документації та курси, які вивчалися у складі спеціальності «Комп'ютерна інженерія». Серед пристроїв, необхідних для проектування, обов'язковими є сервери, маршрутизатори, комутатори (свічі), камери та комп'ютери.

Для оптимізації передачі та підвищення рівня безпеки даних мережу було розділено на підмережі. Це дозволило зменшити внутрішній трафік та підвищити ефективність роботи мережі. Також було налаштовано VPN (Virtual Private Network – віртуальна приватна мережа) для забезпечення захищеного з'єднання між різними сегментами мережі та віддаленими користувачами. Це дозволяє зберігати конфіденційність даних і захищати їх від несанкціонованого доступу.

Для розробки застосунку був обран ігровий рушій Unity, який є одним із найпопулярніших рушіїв для ігр. Він має простий інтерфейс, велику кількість як власної документації, навчальних матеріалів, так і велику кількість сторонніх курсів. Розробка відбувалась для мобільної платформи, уможливорює використання гри незалежно від наявності ПК чи симуляційної установки, що дозволяє учням навчатись керуванню із будь-якої точки світу.

## **2 РОЗРОБКА АПАРАТНОЇ ЧАСТИНИ КІБЕРФІЗИЧНОЇ СИСТЕМИ КОМПАНІЇ**

### **2.1 Технічні вимоги до кіберфізичної системи школи екстремального водіння «ProDriver»**

#### **2.1.1 Вимоги до системи в цілому**

##### **2.1.1.1 Вимоги до структури і функціонування системи школи екстремального водіння «ProDriver»**

###### **2.1.1.1.1 Перелік підсистем, їх призначення й основні характеристики**

Кіберфізична система школи екстремального водіння «ProDriver» призначена для забезпечення задач комутації кінцевих пристроїв на полігоні та в навчальних аудиторіях та маршрутизації даних, отриманих під час навчання та повинна складатися з наступних підсистем:

- підсистема «Адміністративний офіс», яка містить мережу для робочого простіру з 8 комп'ютерів;
- підсистема «Навчальний офіс», яка в свою чергу містить мережу, що виділена на 8 комп'ютерів та серверну кімнату, де розташовано 2 сервери;
- підсистема «Автодром», що містить в собі робочий простір з 1 комп'ютера, 1 сервер та Wi-Fi роутер, до якого під'єднані 5 камер відеоспостереження, 4 на території площадки та 1 в автомобілі.

Кожна підсистема повинна захищати локальні підмережі від несанкціонованого доступу та забезпечувати надійний обмін і обробку даних. Стек протоколів, який забезпечує обмін даними в мережі, включає віртуалізацію локальних мереж, створення приватних тунельних з'єднань і використання засобів керування базами даних MS SQL.

Підсистема «Адміністративний офіс» міститиме облікові данні клієнтів, тому безпека у данному філіалі надважлива.

Підсистема «Навчальний офіс» має мати спеціальне програмне забезпечення для автосимуляторів та можливість у теоретичному класі встановлення проектору, задля демонстрації презентацій.

У підсистемі «Автодром» необхідно встановити та налаштувати програмне забезпечення задля записування та обробки відеоспостереження території та автомобіля.

#### **2.1.1.1.2 Вимоги до способів і засобів зв'язку для інформаційного обміну між компонентами системи**

Між компонентами системи повинен забезпечуватися зв'язок за допомогою кабельного LAN-підключення між собою, кабелем типу «вита пара». Роутери різних структурних підрозділів об'єднуються оптоволоконним кабелем. Протокол Ethernet дозволяє передавати дані зі швидкостями до 100 Мбіт/сек на рівні доступу, а також до 1 Гбіт/сек на рівні розподілу та ядра. На філіалі полігону необхідно забезпечити стабільне з'єднання камер із маршрутизатором, завдяки бездротовій технології Wi-Fi, на частотах 2.4 ГГц задля збільшення радіусу дії.

#### **2.1.1.1.3 Вимоги до характеристик взаємозв'язків створюваної системи із суміжними системами**

Доступ до внутрішніх корпоративних, зовнішніх ресурсів чи суміжних систем має здійснюватися через мережу Інтернет. Для роботи відділу бухгалтерії необхідно забезпечити доступ для роботи у платформі Microsoft 365. Системним адміністраторам необхідно керувати серверами, для цього необхідно використовувати програму Cisco Network Assistant. Задля роботи та редагування камер необхідно встановити таке програмне забезпечення, яке буде дозволяти записувати, транслювати, та редагувати зображення із різних камер.

#### 2.1.1.1.4 Вимоги до режимів функціонування системи

Система повинна забезпечити два режими функціонування, а саме:

- Повний режим активності;
- Економний режим електроенергії.

Повний режим активності передбачає ввімкнення мережевого обладнання, серверів та комп'ютерів у навчальних класах одночасно (від 65% до 100%). Такий режим функціонування буде використовуватися кожен робочий день, тому необхідно забезпечити високий рівень якості компонентів мережі та виконаних консольних налаштувань. У цьому режимі умикається освітлення класів, системи вентиляції, кондиціонування або обігрівачів за необхідності.

Режим економії електроенергії функціонує у вихідні дні та в умовах обмеження постачання електрики в рамках стабілізаційних відключень, коли крім роутерів, комутаторів та серверів ввімкнено менше 10% робочих комп'ютерів.

Режим роботи адміністративної та навчальної частини в певні дні, навіть в різні години може відрізнятися, що зумовлює більший рівень децентралізації системи, що знижує ризики у разі поламки або хакерських атак, а також при застосуванні графіків стабілізаційних відключень. Співробітники зобов'язані вимикати світло при завершенні робочого дня, а також налаштовувати системи обігріву/охолодження повітря до оптимальних, але не до занадто затратних режимів.

Завдяки цим заходам забезпечується ефективне використання ресурсів, підтримується надійність системи та знижуються витрати на енергоспоживання, що є критично важливим для безперервної роботи автошколи.

#### **2.1.1.1.5 Вимоги до діагностування системи**

Для діагностування стану системи передбачено інтегрування мережевого контролера від компанії Cisco в одну з підмереж компанії. Cisco Network Controller API надає можливість отримувати дані у зручному форматі JSON, що полегшує збирання та обробку інформації про доступність мережевого обладнання.

Як компонент системи можна використовувати програму, написану на мові Python, яка є у вільному доступі. Ця програма, спираючись на зібрану контролером інформацію, може автоматично відправляти електронні листи на корпоративну пошту адміністратора у разі виявлення несправностей.

Сформований електронний лист буде містити ім'я приладу, його MAC-адресу та причину недоступності. Всі ці дані будуть зібрані з JSON файлу та оброблені програмним кодом, що забезпечить швидке реагування на проблеми в мережі та підвищить надійність системи. Завдяки цьому підходу, адміністрація буде завжди в курсі стану мережі та зможе оперативно вживати необхідних заходів для усунення поломок.

#### **2.1.1.1.6 Перспективи розвитку, модернізації системи**

Дана система повинна мати можливість масштабування, збільшення існуючої кількості хостів та філіалів. Також можливе розширення кількості робочих місць для збільшення продуктивності та зменшення навантаження на кожного співробітника. Це обов'язковий сценарій у випадку розширення філіалів. Необхідно буде збільшити штат працівників служби підтримки, бухгалтерії, системних адміністраторів, менеджерів та викладачів. Для майбутнього масштабування мережі необхідно збільшити кількість додаткових портів на комутаторах та продумати запасні розетки. Також необхідно враховувати подальший розвиток програмного забезпечення та збільшення



вимог до нього, тому необхідно мати запас по продуктивності комп'ютерів та симуляційних систем як мінімум вдвічі від рекомендованих вимог.

Необхідно врахувати, що при збільшенні масштабів системи необхідно модифікувати рівні безпеки, можливо поставити охоронні сигналізації та системи відеонагляду у навчальних класах, так як цей філіал містить велику кількість дорогого обладнання.

Окрім цього, необхідно врахувати подальшій можливий розвиток практичної частини занятть, а саме оснащення автомобілів камерами, які можуть знімати педалі, коробку передач, керування автомобіля водієм, камери які встановленні зовні авто та знімають кут повороту колес, і тд. Також можливе обладнання камер елементами штучного інтелекту, задля відстеженні моментів наїзду на конус автомобіля та інших помилок у керуванні.

Мобільний застосунок можна модифікувати додаванням автомобілів із різним типом приводу, вагою та потужністю. Окрім цього є потенціал розвитку системи оцінювань, додаванні різних завдань, цікавих та корисних треків для навчання.

### **2.1.1.2 Показники призначення**

Необхідно, щоб комп'ютерна мережа школи екстремального водіння «ProDriver» працювала безперебійно та виконувала всі необхідні завдання, такі як обмін даними між робочими комп'ютерами, доступ до Інтернет-ресурсів і системи корпоративного зв'язку.

- кількість робочих комп'ютерів, необхідних для роботи програмним забезпеченням, повинна дорівнювати 17;
- швидкість Інтернет-з'єднання не повинна бути нижчою за 100 Мбіт/сек;
- кількість облікових записів на поштовому сервері повинна дорівнювати кількості співробітників.

### **2.1.1.3 Вимоги до експлуатації, технічного обслуговування, ремонту і збереження компонентів системи школи екстремального водіння «ProDriver»**

#### **2.1.1.3.1 Умови і регламент експлуатації, що повинні забезпечувати використання технічних засобів системи**

Таблиця 1.1 містить інформацію про кількість персоналії. Співробітники адміністративної служби компанії «ProDriver» працюють за графіком п'яти робочих днів на тиждень і двох вихідних днів на тиждень. Робочий день складається з восьми годин, починаючи о 9:00 і закінчуючи о 18:00.

Співробітники навчальної частини працюють за індивідуальним графіком згідно розкладу занять з урахуванням побажань клієнтів, але з дотриманням норм чинного законодавства України.

Проте після закінчення робочого дня вимикаються лише робочі комп'ютери, які не беруть участь у навчальному процесі, серверне й комутуюче обладнання та робочі станції в навчальних аудиторіях та обладнання на полігоні залишається ввімкненим під час проведення занять. Такий підхід може враховувати графіки стабілізаційних відключень, забезпечувати ефективну роботу школи та запобігати значним витратам електроенергії.

Система повинна бути експлуатована без додаткового обслуговування за умови справності ключових елементів мережі. Планове обслуговування комп'ютерної мережі має проводитися раз на 6 місяців, а термінове – за потреби.

### **2.1.1.3.2 Вимоги до кількості, кваліфікації обслуговуючого персоналу і режимам його роботи**

Для підтримки роботи мережі достатньо 1 системного адміністратора. Він повинен мати змогу їздити на полігон, у разі несправностей там. Для цього компанією може надаватись службове авто. Також він повинен мати як мінімум 1 рік досвіду роботи системним адміністратором.

### **2.1.1.3.3 Вимоги до регламенту обслуговування**

Вимоги до регламенту обслуговування комп'ютерної мережі у автошколі включають кілька критично важливих заходів, які повинні виконуватися за чітко визначеним графіком.

По-перше, необхідно проводити регулярне обслуговування мережевого обладнання кожного першого понеділка місяця. Це включає перевірку стану кабелів, підключень, а також оновлення прошивок та налаштувань роутерів, комутаторів і серверів.

По-друге, постійний моніторинг мережевого трафіку та стану обладнання має здійснюватися щоденно. Для цього використовується спеціалізоване програмне забезпечення, таке як Cisco Network Controller. Дані моніторингу повинні аналізуватися щопонеділка та щочетверга о 9:00 ранку для виявлення та негайного реагування на будь-які проблеми.

Оновлення операційних систем та антивірусного програмного забезпечення повинно проводитися щомісяця, кожного другого вівторка. Це забезпечить захист мережі від нових загроз та підвищить загальну ефективність її роботи.

Додатково, кожен третій понеділок місяця необхідно проводити резервне копіювання критичних даних мережі, щоб забезпечити їх збереження у випадку непередбачуваних збоїв.

#### **2.1.1.4 Вимоги до патентної чистоти**

Програмні та технічні забезпечення, які використовуються, повинні зберігати патентну чистоту. Авторські права виробників мережевого обладнання та програмного забезпечення охороняються як українським, так і міжнародним законодавством. Отже, дозволяється лише цільове використання обладнання або програмного забезпечення.

#### **2.1.1.5 Додаткові вимоги**

##### **2.1.1.5.1 Вимоги до розробки компонента системи**

Компонент системи має представляти із себе застосунок на мобільний телефон, який повинен мати приємну візуальну складову, пропрацьовану та реалістичну фізику. Також у застосунку повинна бути система оцінки користувача та об'єкти, із яких можна будувати необхідні для навчання ситуації. Користувацький інтерфейс повинен бути простим у розумінні, але при цьому інтуїтивно зрозумілим.

#### **2.1.2 Вимоги до задач (налаштувань), які виконуються у комп'ютерній системі**

Мережа повинна складатись із трьох підмереж, LAN1-LAN3.

Кількість вузлів повинна становити відповідно: 51, 84, 204.

Блок адрес повинен бути 10.25.80.0/22.

У локальній мережі між роутерами використовуватиметься блок адрес 10.1.10.0 /24.

Необхідно виконати базове налаштування пристроїв, задати DHCP та динамічну маршрутизацію, налаштувати статичний та динамічний NAT, встановити з'єднання типу VPN та виконати налаштування VLAN.

## 2.2 Розробка інженерних рішень і комплексу технічних засобів

### 2.2.1 Вибір і обґрунтування структурної схеми комплексу технічних засобів комп'ютерної системи шляхом узгодження структури з топологічними особливостями об'єкту розробки

Комп'ютерна мережа компанії є комплексною і взаємопов'язаною системою, що складається з ПК, серверів, комутаторів та маршрутизаторів. Основними завданнями цієї мережі є забезпечення інтернет-з'єднання та зв'язок між різними підмережами. З огляду на раніше зазначені вимоги, була розроблена структурна схема мережі підприємства, яка показана на рисунку 2.1.

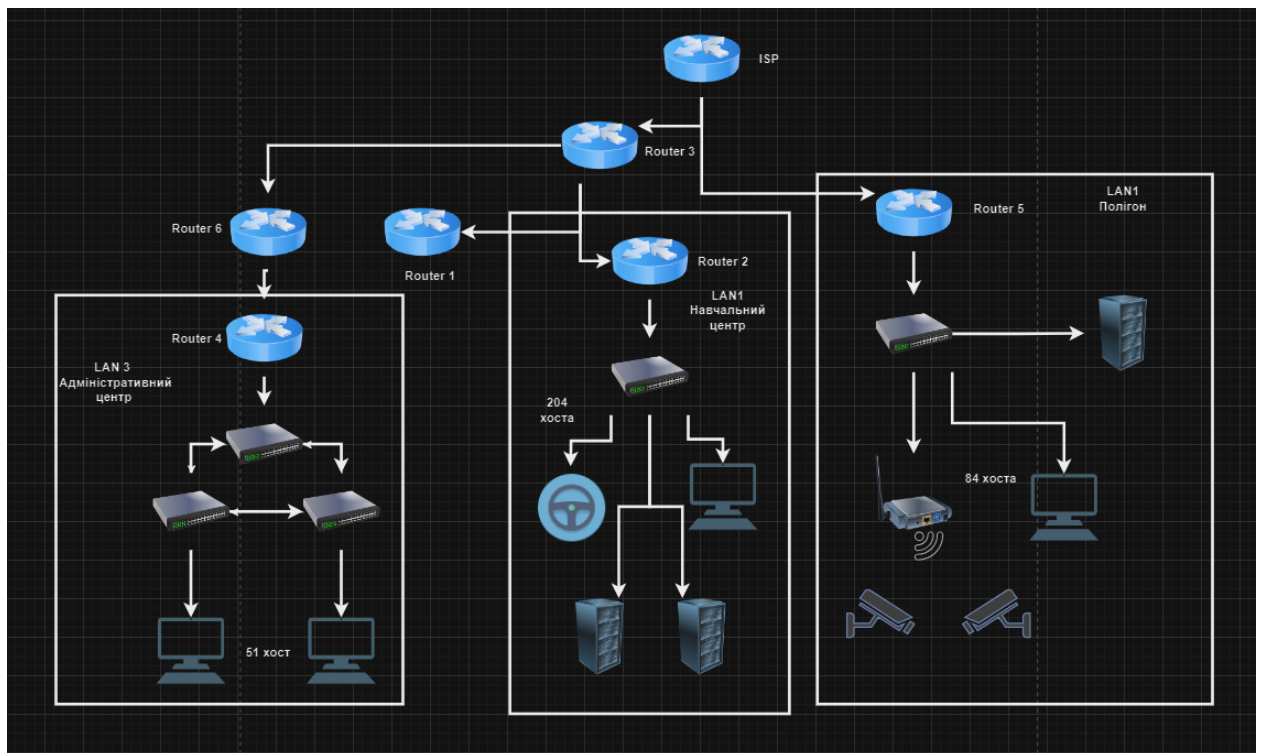


Рисунок 2.1 – Структурна схема комплексу технічних засобів

### **2.2.2 Розробка специфікацій апаратних засобів комп'ютерної системи**

Розробка специфікацій апаратних засобів комп'ютерної системи передбачає детальне визначення вимог до обладнання, яке буде використовуватися для забезпечення функціонування мережі. Це включає вибір відповідних серверів, ПК, комутаторів, маршрутизаторів та іншого мережевого обладнання.

Необхідно враховувати продуктивність серверів, оскільки вони будуть обробляти велику кількість запитів і забезпечувати надійне зберігання даних. Сервери мають мати високошвидкісні процесори, достатній обсяг оперативної пам'яті та місткі жорсткі диски або SSD.

ПК, які будуть використовуватися співробітниками, мають відповідати вимогам продуктивності для виконання завдань, пов'язаних з їхніми обов'язками. Комп'ютери для симуляційного класу повинні мати такі системні характеристики, які будуть переважати рекомендовані характеристики для програм – симуляторів як мінімум в два рази.

Комутатори та маршрутизатори повинні забезпечувати надійний і швидкий зв'язок між різними сегментами мережі. Вони мають підтримувати сучасні стандарти мережевих протоколів та забезпечувати високу пропускну здатність для зменшення затримок у передачі даних.

Також необхідно врахувати надійність і безпеку обладнання. Використання відомих виробників, таких як Cisco, може гарантувати якість і сумісність компонентів, а також забезпечити належну технічну підтримку та оновлення програмного забезпечення.

Для поставки апаратних засобів було обрано компанію Cisco. Ця компанія має кілька важливих переваг:

1. Надійність та якість: Обладнання Cisco відоме своєю високою якістю і надійністю. Використання таких компонентів знижує ризик відмов системи, забезпечуючи стабільну роботу мережі;

2. Підтримка та обслуговування: Cisco пропонує комплексну технічну підтримку, включаючи регулярні оновлення програмного забезпечення, консультації та послуги з обслуговування. Це гарантує, що обладнання завжди буде працювати ефективно та безпечно;

3. Масштабованість: Продукти Cisco легко інтегруються в існуючу мережу і можуть бути розширені за потреби. Це особливо важливо для підприємств, які планують рости і розширювати свою інфраструктуру;

4. Безпека: Обладнання Cisco має вбудовані функції безпеки, які допомагають захищати мережу від кіберзагроз. Це включає міжмережеві екрани (firewalls), VPN, системи виявлення вторгнень та інші засоби захисту;

5. Широкий вибір рішень: Cisco пропонує широкий асортимент продуктів і рішень, які можуть задовольнити різні потреби підприємства, від малих бізнесів до великих корпорацій. Це дозволяє підібрати оптимальне обладнання для конкретних завдань.

Cisco також забезпечує високий рівень безпеки завдяки вбудованим функціям захисту, таким як міжмережеві екрани, VPN і системи виявлення вторгнень. Ці переваги роблять Cisco оптимальним вибором для побудови надійної і захищеної мережі.

У всіх підмережах було обрано маршрутизатори на 24 порти, до яких під'єднується різна кількість ПК. Це дає можливість у майбутньому масштабувати мережу та розширювати компанію. Данні із специфікацією ПК наведені у таблиці 2.1.

Таблиця 2.1 – Специфікація обладнання

Позиція	Найменування і Технічна характеристика	Тип, марка, позначення документа, опитувального листа	Кількість	Примітки
1.	Коммутатор  Деталі порту/слота розширення:  24 x Fast Ethernet Network  2 x Gigabit Ethernet Uplink  2 x Gigabit Ethernet Expansion	Cisco WS-C2960-24LC-S	5	Головний офіс - 4 Віддалений - 1
2.	Маршрутизатор  RJ-45 Ports: 3  Тип слота розширення:  HWIC, PVDM  Технологія Ethernet:  Гігабітний Ethernet	Cisco 2911/K9	7	Головний офіс - 5 Віддалений - 2
3.	Сервер  Кількість портів Ethernet 2x RJ45  Швидкість з'єднання 1  Гбіт/с  Тип процесора Intel Xeon E5  v3/v4	Cisco UCS C240 M4 12 LFF 2U	3	Головний офіс - 2 Віддалений - 1



Продовження таблиці 2.1

4.	Комп'ютер Процесор: Intel Core i7-12700KF GeForce RTX3080 Обсяг оперативної пам'яті: 16 ГБ Обсяг SSD: 500 ГБ	-	5	Головний офіс - 5
5.	Комп'ютер Процесор: Intel Core i5 GeForce GTX1050Ti Обсяг оперативної пам'яті: 16 ГБ Обсяг SSD: 500 ГБ	-	11	Головний офіс - 10 Віддалений - 1
6.	Клавіатура + миша бездротова Logitech	MK235 Dark Gray	16	Головний офіс - 15 Віддалений - 1
7.	Монітор Samsung	S24C330	16	Головний офіс - 15 Віддалений - 1
8.	Кермо із педалями для ПК Logitech	G923 Racing	5	Головний офіс - 5
9.	Камера TP-LINK	Таро C320WS	5	Віддалений - 5

Для вибору системи кабельних каналів необхідно врахувати розміщення комп'ютерної системи. Спираючись на кількість пристроїв та на план будівлі було розроблено схему розташування кабельних каналів, приклад підмережі рисунок 2.2. Синім кольором зображено lan-кабель, фіолетовим зображено кабель живлення.

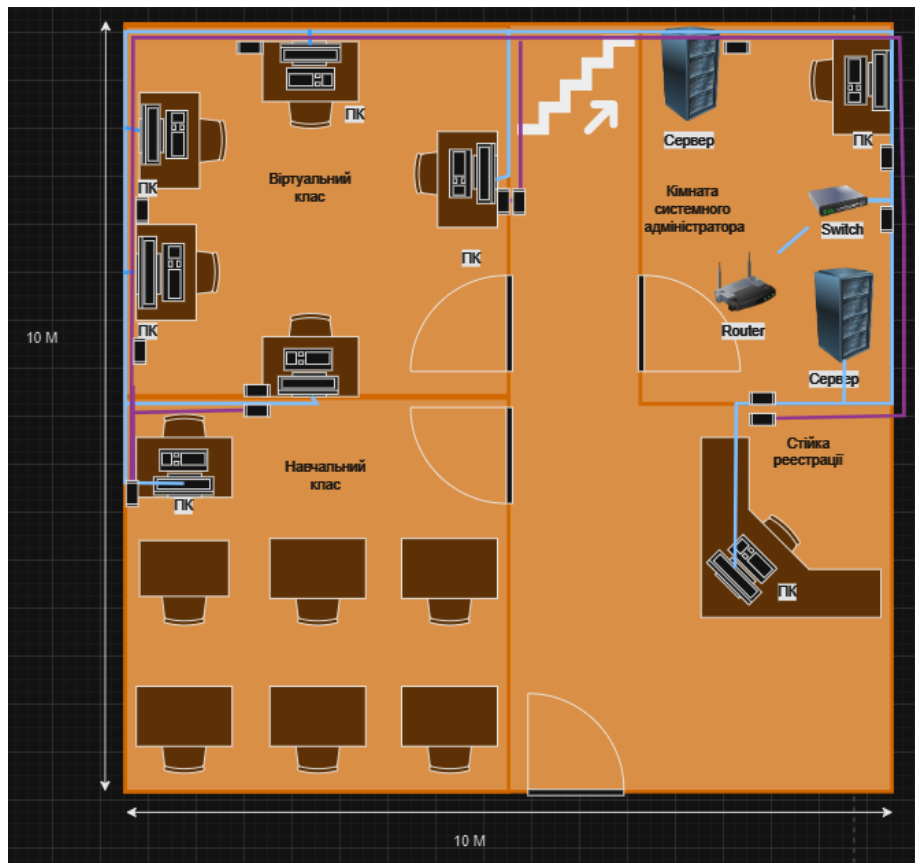


Рисунок 2.2 – Схема розташування кабельних каналів КС

Після проектування планів розміщення, було створено специфікацію кабельних каналів, вона зображена на таблиці 2.2.

Таблиця 2.2 – Специфікація кабельного обладнання

Позиція	Найменування і Технічна характеристика	Тип, марка, позначення документа, опитувального листа	Кількість	Одиниці виміру	Примітки
1.	Розетка RJ45 Клас захисту: IP20 Кількість портів: 1	NCAC-1F5E-01	11	од	Для LAN2
2.	Розетка живлення Asfora із заземленням Клас захисту: IP20	Schneider Electric	13	од	Для LAN2
3.	Лан кабель	Патч-корд RZTK CAT5e RJ45 UTP	250	м	Для LAN2
4.	Кабель живлення	Мідний провід ПВС 3x1	80	м	Для LAN2

### 3 РОЗРОБКА КОРПОРАТИВНОЇ МЕРЕЖІ ПІДПРИЄМСТВА

#### 3.1 Розрахунок схеми адресації корпоративної мереж

Згідно з завданням, у мережах LAN1-LAN3 для зв'язку між маршрутизаторами було використано блок адрес 10.25.80.0/22, що зображено на таблиці 3.1.

Таблиця 3.1 – Схема адресації мережі

Блок адрес	LAN 1 Навчальний офіс	LAN 2 Полігон	LAN 3 Адміністративний офіс
10.25.80.0/22	51	84	204

Для розрахунку використовується метод VLSM (Variable Length Subnet Masks). Метод VLSM розділяє мережу на підмережі, які також можуть розділитися[5]. Для мережі LAN1, в яку входить 51 вузол, необхідна маска підмережі /26, доступні адреси 10.25.81.129 - 10.25.81.190, ширококомовлення за адресою 10.25.81.128. Для мережі LAN2, в яку входить 84 вузла, необхідна маска підмережі /25, доступні адреси 10.25.81.1 - 10.25.81.126, ширококомовлення за адресою 10.25.81.0. Для мережі LAN3, в яку входить 204 вузла, необхідна маска підмережі /24, доступні адреси 10.25.80.1 - 10.25.81.254, ширококомовлення за адресою 10.25.80.0. Ці дані зображені на таблиці 3.2

Таблиця 3.2 – Схема адресації підмереж

Довжина	Ім'я	К-сть	IP мережі	Маска	Перша адреса	Остання адреса
Найбільша	LAN3	254/204	10.25.80.0	/24	10.25.80.1	10.25.80.254
	LAN2	126/84	10.25.81.0	/25	10.25.81.1	10.25.81.126
Найменша	LAN1	62/51	10.25.81.128	/26	10.25.81.129	10.25.81.190

Згідно з завданням, для зв'язку між маршрутизаторами нами було використано блок адрес 10.1.10.0 /24. На таблиці 3.3 зображена повна схема адресації мережі.

Таблиця 3.3 – Схема адресації мережі

Назва мережі	К-сть вузлів	Номер мережі	Маска мережі	Початкове значення діапазону можливих адрес вузлів у підмережі	Кінцеве значення діапазону можливих адрес вузлів у підмережі
LAN3	254/204	10.25.80.0	/24	10.25.80.1	10.25.80.254
LAN2	126/84	10.25.81.0	/25	10.25.81.1	10.25.81.126
LAN1	62/51	10.25.81.128	/26	10.25.81.129	10.25.81.190
WAN 1	2/2	10.1.10.0	/30	10.1.10.1	10.1.10.2
WAN 2	2/2	10.1.10.4	/30	10.1.10.5	10.1.10.6
WAN 3	2/2	10.1.10.8	/30	10.1.10.9	10.1.10.10
WAN 4	2/2	10.1.10.12	/30	10.1.9.13	10.1.10.14
WAN 5	2/2	10.1.10.16	/30	10.1.9.17	10.1.10.18
WAN 6	2/2	10.1.10.20	/30	10.1.9.21	10.1.10.22
ISP 1	2/2	209.165.202.0	/30	209.165.202.1	209.165.202.2
ISP 2	2/2	64.100.13.0	/30	64.100.13.1	64.100.13.2

Серверам було привласнено IP-адреса за правилом: IP-адрес дорівнює першому можливому адресу у мережі+9+№, де № – номер варіанту студента за списком у групі. Адресація серверів зображена на таблиці 3.4

Таблиця 3.4 – Адресація інтерфейсів серверів

Назва серверу	Назва інтерфейсу	IP-адреса	Маска	Шлюз
Server HTTP	Fa0	10.25.81.149	26	10.25.80.129
Server DNS	Fa0	10.25.81.150	26	10.25.80.129
Server TFTP	Fa0	10.25.81.20	25	10.25.81.1

### 3.2 Розробка топологічної схеми корпоративної мережі

У програмі Cisco Packet Tracer була розроблена топологічна схема, що складається із 3 підмереж, а саме з адміністративного і навчального офісу та полігону. Загальну архітектуру мережі зображено на рисунку 3.1, схеми підмереж на рисунку 3.2 - 3.3.

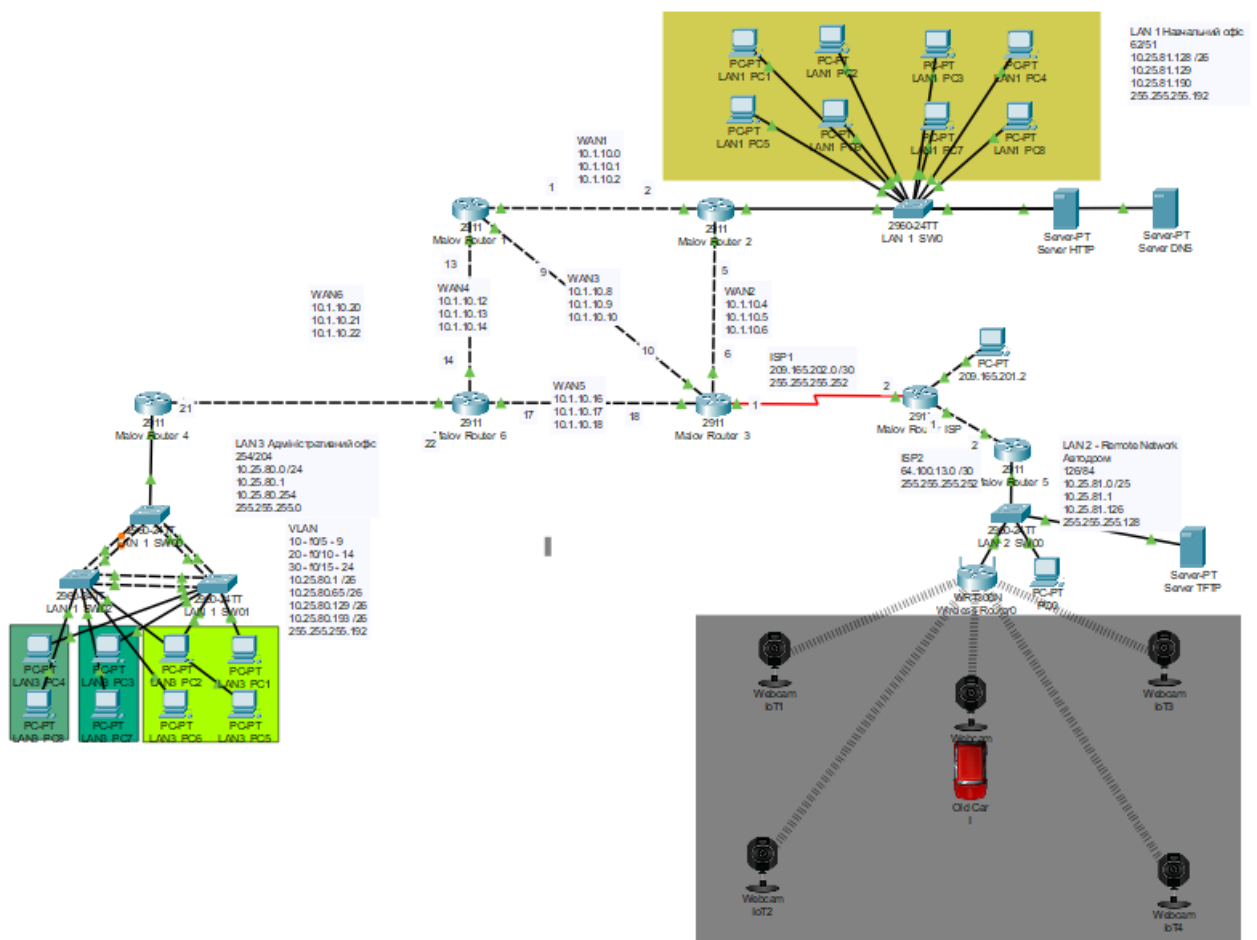


Рисунок 3.1 – Топологія мережі

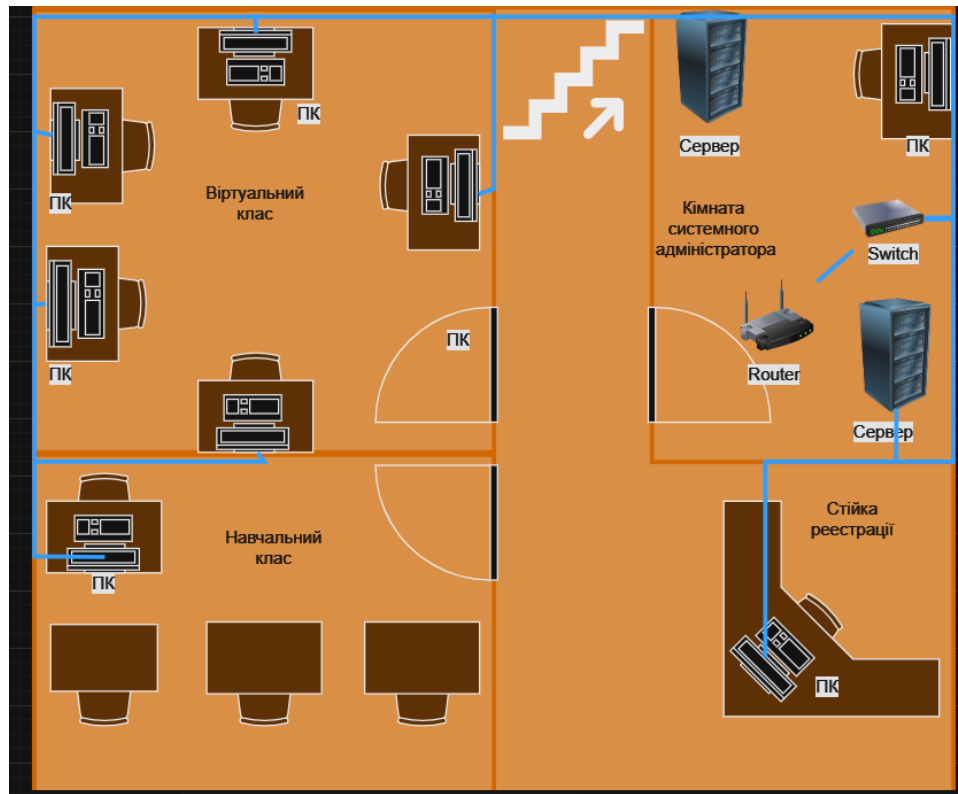


Рисунок 3.2 – Схема підключення пристроїв у навчальному центрі

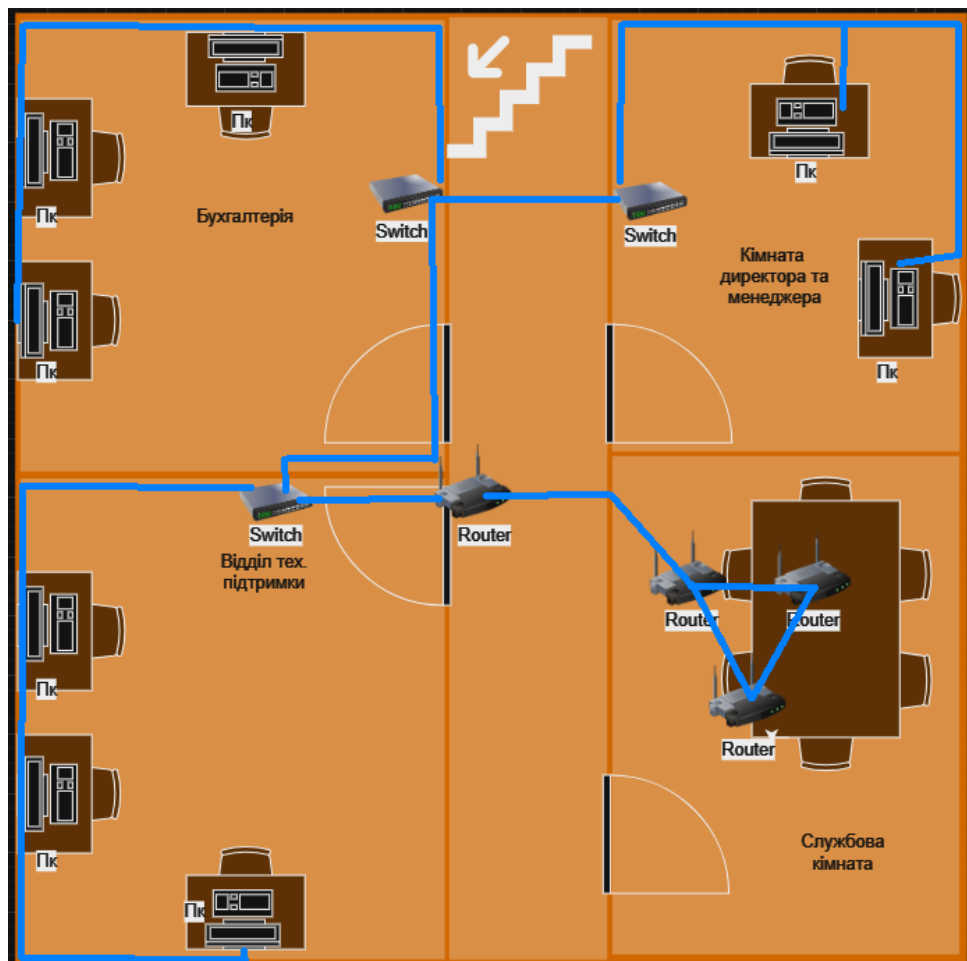


Рисунок 3.3 – Схема підключення пристроїв у адміністративному центрі

### 3.3 Налаштування мережі

#### 3.3.1 Базове налаштування

Для початку необхідно виконати базове налаштування пристроїв. Для цього, згідно із рекомендаціями, необхідно:

- назначити назви пристроям за наступним правилом:

Прізвище студента\_тип пристрою\_номер пристрою, наприклад, Malov\_Router\_1;

- для всіх пристроїв встановити пароль cisco до консолі та vty;

- на всіх пристроях назначити пароль class до привілейованого режиму;

- усі паролі, що зберігаються у відкритому вигляді, пропонується під час налаштування моделі комп'ютерної системи зашифрувати;

- створити банер MOTD;

- для всіх ліній vty налаштувати протокол ssh;

- призначити на всіх пристроях користувача за правилом:

група\_прізвище,

наприклад 123202\_Malov, з паролем admincisco;

- в якості імені домена використати ім'я пристрою для шифрування даних;

- створювати ключ RSA завдовжки 1024 біт;

- на DCE-інтерфейсах маршрутизаторів призначити встановлення значення тактової частоти – 128000.

Для прикладу наведено налаштування на роутері Malov\_Router\_2.

Призначимо назву за правилом Malov\_Пристрій\_Номер.

```
Malov_Router_2(config)#hostname Malov_Router_2
```

Призначимо пароль cisco до консолі і vty.

```
Malov_Router_2(config)#line console 0
```

```
Malov_Router_2(config-line)#password cisco
```

```
Malov_Router_2(config-line)#login
```



```
Malov_Router_2(config-line)#exit
```

```
Malov_Router_2(config)#line vty 0 15
```

```
Malov_Router_2(config-line)#password cisco
```

```
Malov_Router_2(config-line)#login
```

```
Malov_Router_2(config-line)#
```

Призначимо пароль class до привілейованого режиму

```
Malov_Router_2(config)#enable secret class
```

Паролі, що зберігаються у відкритому вигляді зашифруємо.

```
Malov_Router_2(config)#service password-encryption
```

Розробимо банер MOTD.

```
Malov_Router_2(config)#banner motd "Welcome to Malov_Router_2"
```

Призначимо на усіх лініях vty використання протоколу ssh.

```
Malov_Router_2(config)#line vty 0 15
```

```
Malov_Router_2(config-line)#transport input ssh
```

Призначимо на всіх пристроях користувача за правилом: 123202\_ Malov з паролем admincisco.

```
Malov_Router_2(config)#username 123202_ Malov privilege 15 secret admincisco
```

В якості імені домена використаємо ім'я пристрою.

```
Malov_Router_2(config)#ip domain name Malov_Router_2
```

Для шифрування даних створимо ключ RSA завдовжки 1024 біт.

```
Malov_Router_2(config)#crypto key generate rsa
```

The name for the keys will be: Malov\_Router\_2. Malov\_Router\_2

Choose the size of the key modulus in the range of 360 to 4096 for your General Purpose Keys.

How many bits in the modulus [512]: 1024

На DCE-інтерфейсах маршрутизаторів призначимо встановлення значення тактової частоти – 128000.

```
Malov_Router_2(config-if)#clock rate 128000
```

Після налаштування збережемо конфігурацію пристрою.

Malov\_Router\_2#copy running-config startup-config

### 3.3.2 Налаштування маршрутизації корпоративної мережі

Існує 2 види маршрутизації, статична та динамічна. Статична маршрутизація передбачає ручне налаштування маршрутів адміністратором мережі, що забезпечує простоту налаштування та повний контроль, але погано масштабується і не адаптується до змін у мережі. Натомість, динамічна маршрутизація використовує протоколи, такі як RIP, OSPF і BGP, для автоматичного визначення найкращих маршрутів. Це забезпечує адаптивність і масштабованість, дозволяючи мережі автоматично реагувати на зміни. Динамічна маршрутизація підходить для великих і складних мереж, але вимагає більше знань для налаштування та споживає більше ресурсів для обміну маршрутизуючою інформацією. У данному проєкті, задля маштабованості була обрана саме динамічна маршрутизація і протокол OSPF.

Серед переваг протоколу OSPF є:

- Висока швидкість збіжності;
- Підтримка мережних масок змінної довжини VLSM;
- Відсутність обмежень досяжності;
- Оптимальне використання пропускної здатності мережі;
- Оптимальний вибір шляху маршрутизації.

Перед налаштуванням маршрутизації кожному мережевому пристрою потрібно надати IP-адресу та маску мережі відповідно до таблиць маршрутизації. Для цього обирається необхідний інтерфейс за допомогою команди:

```
Malov_Router_2(config)#interface назва_інтерфейсу
```

І потім вводиться IP адреса і маска підмережі, за допомогою команди:

```
Malov_Router_2(config-if)#ip address IP-адреса маска_підмережі.
```

Після цього можна приступати до налаштування інтерфейсу OSPF.

Приклад налаштування OSPF на роутері Malov\_Router\_5 зображено на рисунку 3.4.

```

Malov_Router_5
Physical Config CLI Attributes
IOS Command Line Interface
Malov_Router_5(config)#router ospf 10
Malov_Router_5(config-router)#network 64.100.13.0 0.0.0.3 area 0
Malov_Router_5(config-router)#
01:28:16: %OSPF-5-ADJCHG: Process 10, Nbr 209.165.202.2 on GigabitEthernet0/0 from LOADING
to FULL, Loading Done

Malov_Router_5(config-router)#network 10.25.81.0 0.0.0.127 area 0
Malov_Router_5(config-router)#passive-interface default
Malov_Router_5(config-router)#no passive-interface GigabitEthernet0/0
Malov_Router_5(config-router)#auto-cost reference-bandwidth 1000
01:29:00: %OSPF-5-ADJCHG: Process 10, Nbr 209.165.202.2 on GigabitEthernet0/0 from FULL to
DOWN, Neighbor Down: Interface down or detached

% OSPF: Reference bandwidth is changed.
Please ensure reference bandwidth is consistent across all routers.
Malov_Router_5(config-router)#
Malov_Router_5(config-router)#
01:29:40: %OSPF-5-ADJCHG: Process 10, Nbr 209.165.202.2 on GigabitEthernet0/0 from LOADING
to FULL, Loading Done

```

Рисунок 3.4 – Налаштування OSPF

Окрім цього, було виконано налаштування DHCP. Також необхідно виконати налаштування всіх маршрутизаторів на підтримку служби AAA таким чином:

- задати пропускну спроможність на serial-інтерфейсах = 128 Кб/с, вартість метрики = 7500;
- для перевірки підключень до VTU ліній на маршрутизаторі використовувати локальну базу даних користувачів;
- для доступу до консолі використовувати аутентифікацію на основі протоколу RADIUS і якщо немає – локальну базу даних;
- radius-сервер налаштувати наступним чином: ключове слово – radius123;
- в якості облікового запису користувачів використовувати ім'я пристрою з паролем admin123.

В якості Radius сервера був використаний сервер NTTP, що дозволяє зменшити витрати на технічне обладнання та спростити роботу системного

адміністратора. Виконане налаштування Radius сервера зображено на рисунку 3.5.

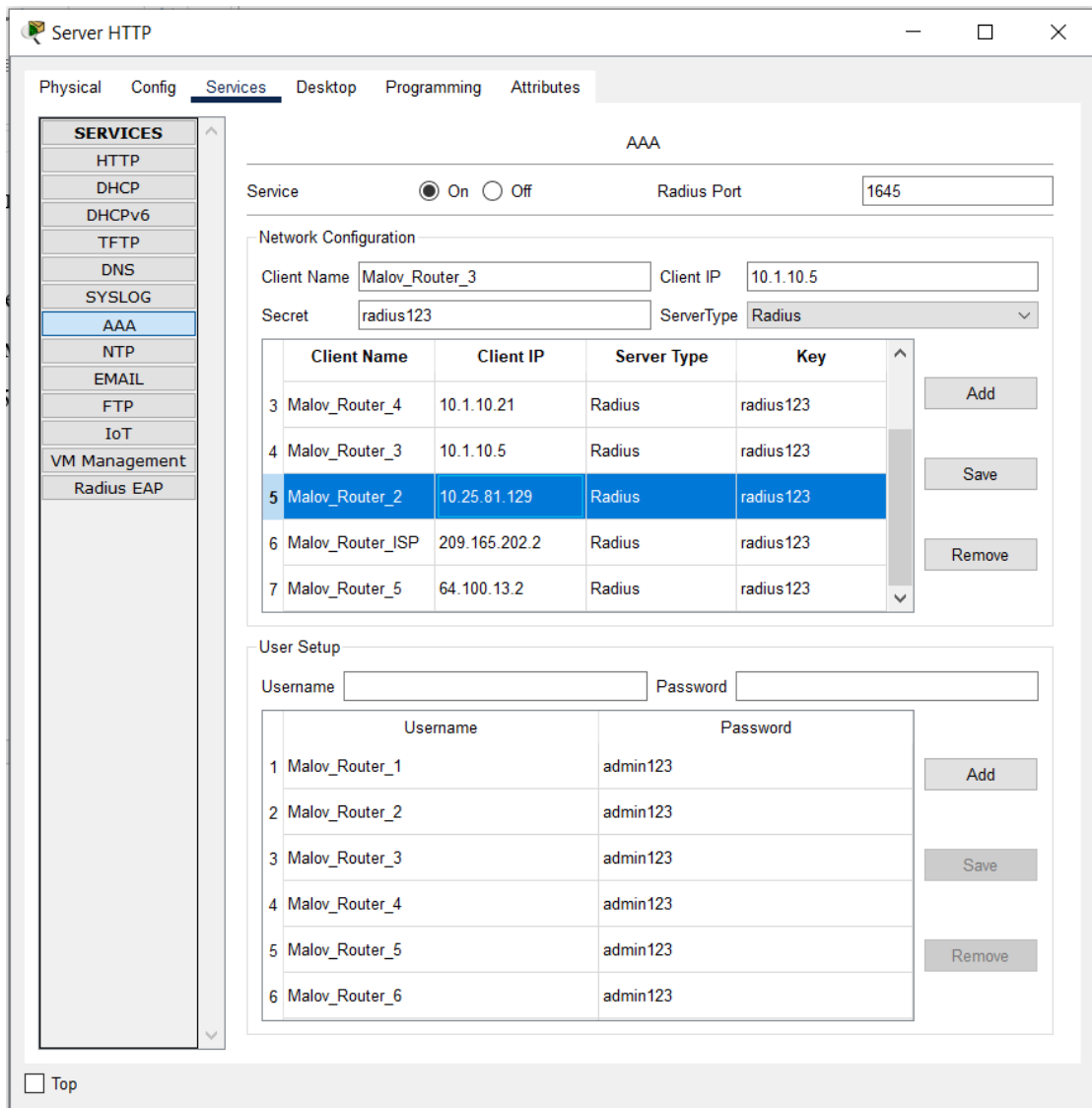


Рисунок 3.5 – Виконані налаштування Radius сервера

### 3.3.3 Налаштування NAT

Наступним кроком у налаштуванні мережі буде налаштування мережевого стандарту NAT (Network Address Translation, з англ. трансляція мережевих адрес). Ідея і головне призначення NAT полягає в тому, щоб дозволити кільком окремим пристроям виходити і отримувати доступ в Інтернет через одну публічну адресу. Щоб це здійснити, потрібно здійснити перетворення приватної IP-адреси в загальнодоступну IP-адресу[6].

Згідно із завдання, необхідно:

– для виходу робочих станцій в Інтернет необхідно налаштувати пограничний маршрутизатор з динамічним NAT за такими даними: ім'я пула Internet, пул адресів з 209.165.200.5 по 209.165.200.30, номер списку доступу 10;

– налаштувати сервер HTTP, щоб на вузлах при вводі в рядку браузера `http://123.dnipro.ua` (`http://209.165.200.4`) відкривався веб-сайт.

Послідовність дій, яка була виконана:

1. Створено пул Internet.
2. Налаштовано списки доступу.
3. Налаштовано NAT
4. Вказано, які інтерфейси вихідні, а які вхідні.
5. Налаштовано статичний нат для серверу HTTP та DNS.
6. Налаштовано сервер DNS щоб при вводі `http://123.dnipro.ua` (`http://209.165.200.4`) відкривався веб-сайт.

Результатом цих дій було налаштовано NAT, що зображено на рисунку 3.6.

Malov\_Router\_5

```

Physical  Config  CLI  Attributes
IOS Command Line Interface

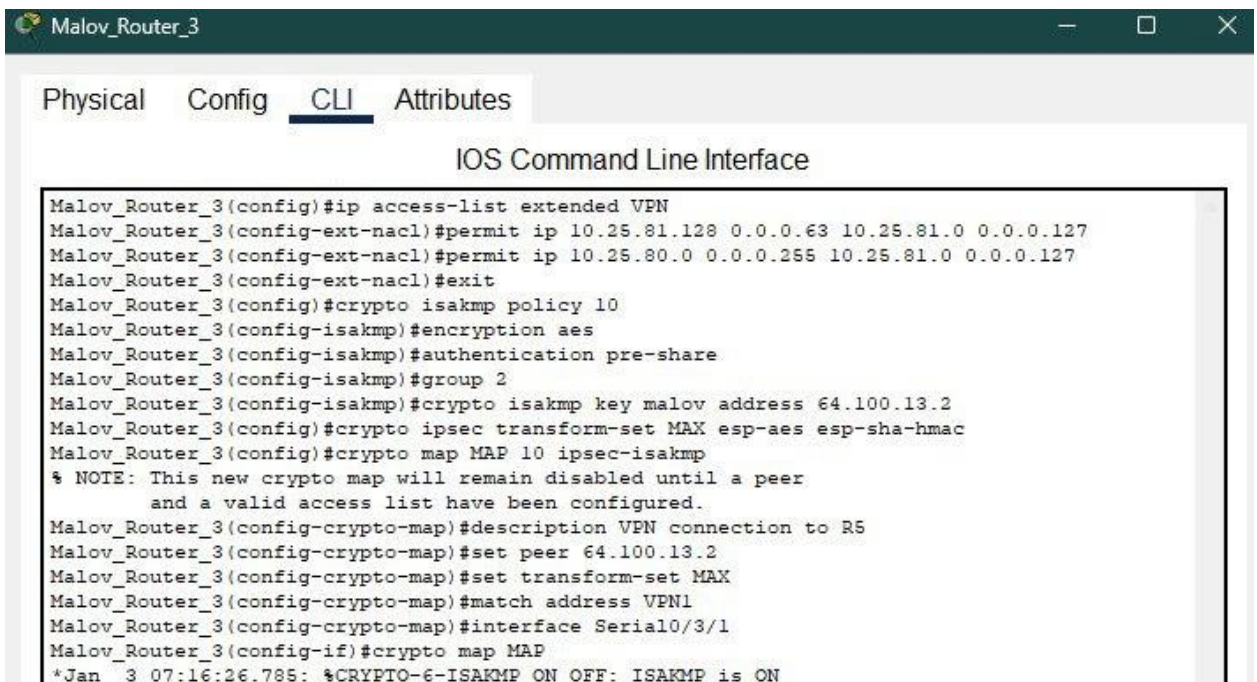
Malov_Router_5#show ip nat statistics
Total translations: 0 (0 static, 0 dynamic, 0 extended)
Outside Interfaces: GigabitEthernet0/0
Inside Interfaces: GigabitEthernet0/1
Hits: 0 Misses: 274
Expired translations: 0
Dynamic mappings:
-- Inside Source
access-list NAT pool Internet refCount 0
 pool Internet: netmask 255.255.255.224
   start 209.165.200.37 end 209.165.200.62
   type generic, total addresses 26 , allocated 0 (0%), misses 0
  
```

Рисунок 3.6 – Результат перевірки роботи NAT

### 3.3.4 Налаштування VPN

VPN (Virtual Private Network) – це віртуальна приватна мережа, яка шифрує трафік між клієнтом і VPN-сервером, а також змінює IP-адресу. При підключенні до VPN між комп'ютером користувача та VPN-сервером створюється захищений канал. У даній топології, згідно завдання, було налаштовано и site-to-site VPN з використанням IPsec для трафіку між мережами LAN3 та LAN.

Першим кроком налаштування є створення списку. Це дозволить передачі даних з підмережі організації. Після цього потрібно створити нову криптографічну політику, яка визначить спосіб автентифікації та тип шифрування даних. Варто зауважити, що тип шифрування та створена криптографічна група мають збігатися в маршрутизаторах, між якими створюється приватна мережа. Наступним кроком є створення VPN-мапи та зазначення списку доступу, який було створено на першому етапі налаштування. Після цього криптографічна мапа та Peer роутер встановлюються на інтерфейс Інтернету. Приклад налаштування 3 роутері зображено на рисунку 3.7



```

Malov_Router_3
Physical  Config  CLI  Attributes
IOS Command Line Interface
Malov_Router_3(config)#ip access-list extended VPN
Malov_Router_3(config-ext-nacl)#permit ip 10.25.81.128 0.0.0.63 10.25.81.0 0.0.0.127
Malov_Router_3(config-ext-nacl)#permit ip 10.25.80.0 0.0.0.255 10.25.81.0 0.0.0.127
Malov_Router_3(config-ext-nacl)#exit
Malov_Router_3(config)#crypto isakmp policy 10
Malov_Router_3(config-isakmp)#encryption aes
Malov_Router_3(config-isakmp)#authentication pre-share
Malov_Router_3(config-isakmp)#group 2
Malov_Router_3(config-isakmp)#crypto isakmp key malov address 64.100.13.2
Malov_Router_3(config)#crypto ipsec transform-set MAX esp-aes esp-sha-hmac
Malov_Router_3(config)#crypto map MAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
and a valid access list have been configured.
Malov_Router_3(config-crypto-map)#description VPN connection to R5
Malov_Router_3(config-crypto-map)#set peer 64.100.13.2
Malov_Router_3(config-crypto-map)#set transform-set MAX
Malov_Router_3(config-crypto-map)#match address VPN1
Malov_Router_3(config-crypto-map)#interface Serial0/3/1
Malov_Router_3(config-if)#crypto map MAP
*Jan  3 07:16:26.785: %CRYPTO-6-ISAKMP ON OFF: ISAKMP is ON

```

Рисунок 3.7 – Налаштування VPN

### 3.3.5 Налаштування VLAN

VLAN (Virtual Local Area Network) – це технологія, що дозволяє створювати віртуальні локальні мережі всередині однієї фізичної мережі. Це досягається шляхом сегментації мережевих пристроїв на логічні групи, незалежно від їхнього фізичного розташування. Основна мета VLAN – покращити продуктивність мережі, підвищити безпеку та спростити управління мережею.

Основні аспекти VLAN:

- сегментація мережі: VLAN дозволяють розділити одну фізичну мережу на кілька логічних сегментів. Це зменшує обсяг ширококомовного трафіку в кожному сегменті та підвищує загальну ефективність мережі;
- підвищення безпеки: З VLAN можна ізолювати різні групи користувачів. Наприклад, співробітники фінансового відділу можуть бути відокремлені від загальної мережі, що зменшує ризик несанкціонованого доступу до конфіденційної інформації;
- гнучкість та масштабованість: VLAN спрощують управління мережею, дозволяючи легко змінювати мережеву конфігурацію без необхідності фізично перепідключати пристрої. Це особливо корисно в середовищах, де потрібно швидко адаптуватися до змін;
- ефективність управління трафіком: VLAN дозволяють контролювати та пріоритизувати трафік, забезпечуючи оптимальну якість обслуговування для важливих додатків.

Для налаштування мереж VLAN і маршрутизації між ними необхідно:

- створити 28, 38, 48, 99, 100 Vlan;
- вимкнути усі невикористовувані фізичні порти комутаторів;
- налаштувати Vlan;
- налаштувати транкові порти і порти доступу;
- налаштувати SVI-інтерфейси на комутаторах;
- налаштувати маршрутизацію між мережами VLAN;

- налаштувати маршрутизатор, що здійснює маршрутизацію між VLAN, в якості сервера DHCP для мереж VLAN;
- створити пули DHCP під назвою pollvlan№, де № – номер VLAN;
- виключити з пулу адреса мережевого обладнання;
- для кожного пулу вказати адресу DNS-сервера і шлюз за умовчанням.

Після виконання цих дій, було виконано перевірку динамічного розподілу адрес, що зображено на рисунках 3.8 та 3.9.

The screenshot shows a configuration window titled "IP Configuration". Under "IP Configuration", the "DHCP" radio button is selected, and the "Static" radio button is unselected. Below this, the "IPv4 Address" field contains the value "10.25.80.141" and the "Subnet Mask" field contains the value "255.255.255.192".

Рисунок 3.8 – Перевірка отримання вузлом адреси через DHCP

The screenshot shows a configuration window titled "Gateway/DNS IPv4". Under "Gateway/DNS IPv4", the "DHCP" radio button is selected, and the "Static" radio button is unselected. Below this, the "Default Gateway" field contains the value "10.25.80.129" and the "DNS Server" field contains the value "10.25.81.150".

Рисунок 3.9 – Перевірка отримання вузлом Gateway/DNS через DHCP

### 3.4 Перевірка роботи комп'ютерної системи

Проведемо відправку пакетів між маршрутизаторами, щоб перевірити роботу OSPF та між ПК у різних підмережах, щоб перевірити статичний NAT та налаштовану маршрутизацію. Результат зображено на рисунку 3.10.



PDU List Window

Fire	Time(sec)	Last Status	Source	Destination	Type	Color	Num	Periodic	Edit	Delete
	0.000	Successful	Malov_Router_6	Malov_Router_1	ICMP		0	N	(edit)	
	0.000	Successful	Malov_Router_1	Malov_Router_3	ICMP		1	N	(edit)	
	0.000	Successful	Malov_Router_4	Malov_Router_3	ICMP		2	N	(edit)	
	0.000	Successful	Malov_Router_4	Malov_Router_ISP	ICMP		3	N	(edit)	
	0.000	Successful	LAN1_PC6	LAN3_PC1	ICMP		4	N	(edit)	
	0.000	Successful	LAN1_PC6	LAN2_PC1	ICMP		5	N	(edit)	
	0.000	Successful	LAN1_PC6	LAN3_PC1	ICMP		6	N	(edit)	

Рисунок 3.10 – Результат перевірки OSPF, NAT та маршрутизації

Для перевірки DHCP необхідно перевірити IP конфігурації на ПК у різних підмережах, це зображено на рисунках 3.11-3.13.

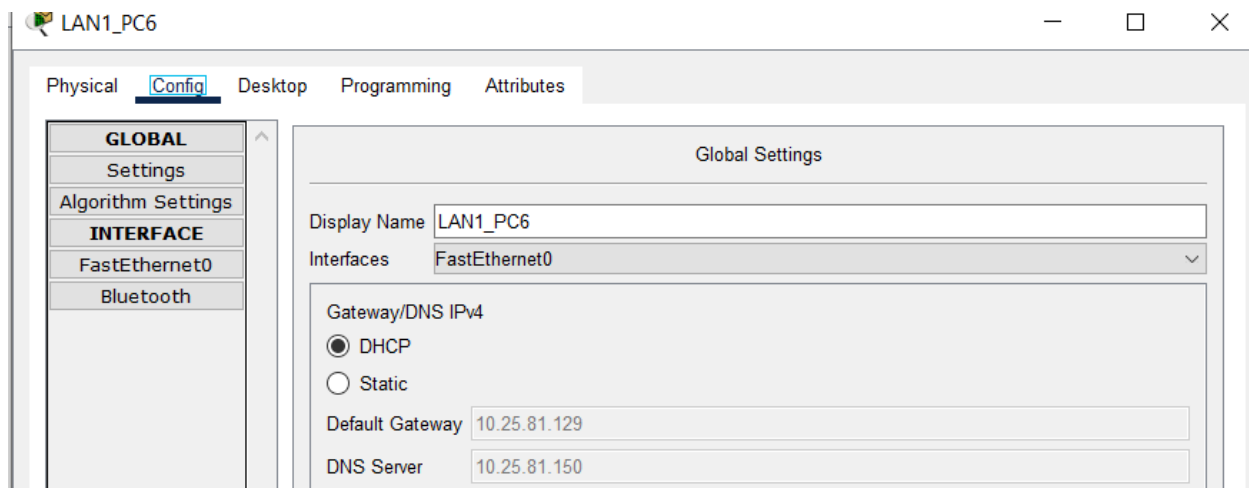


Рисунок 3.11 – Перевірка DHCP на LAN1

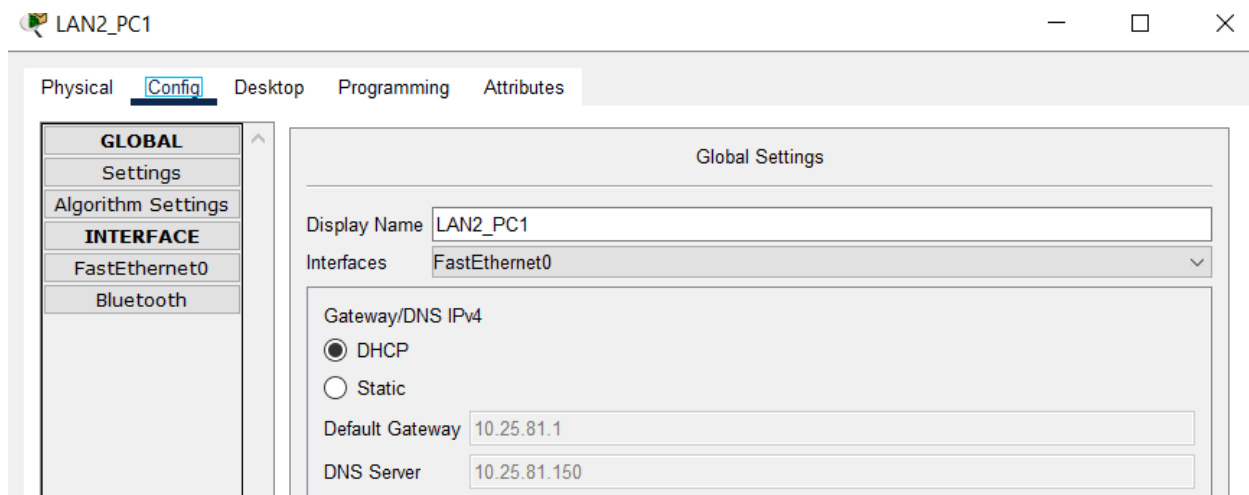


Рисунок 3.12 – Перевірка DHCP на LAN2

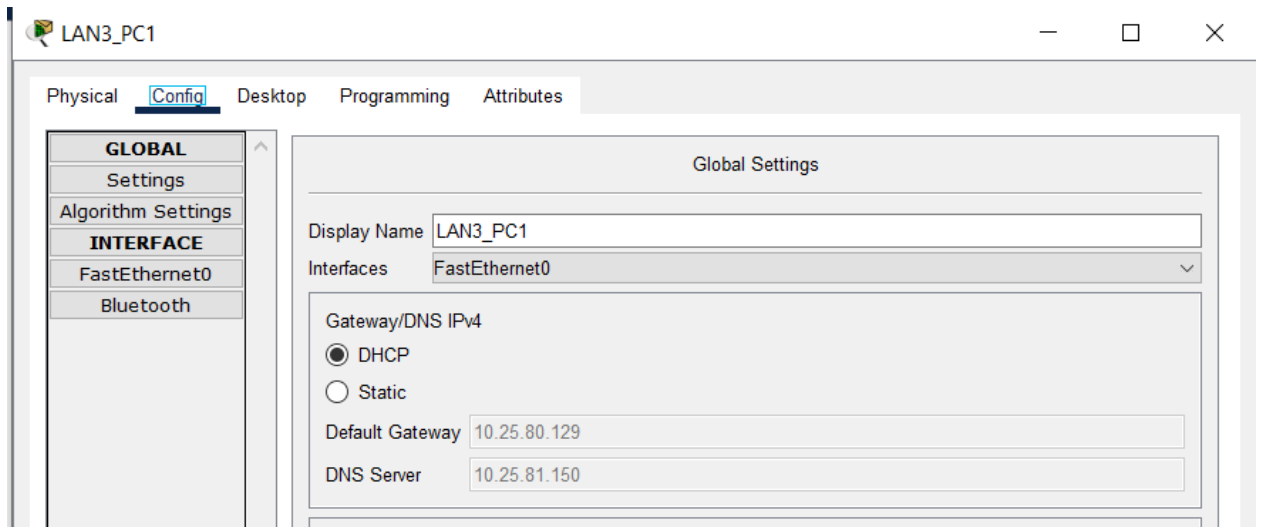


Рисунок 3.13 – Перевірка DHCP на LAN3

Для перевірки статичного NAT для серверів DNS та HTTP необхідно перейти за адресою 123.dnipro.ua, результат зображений на рисунку 3.14.

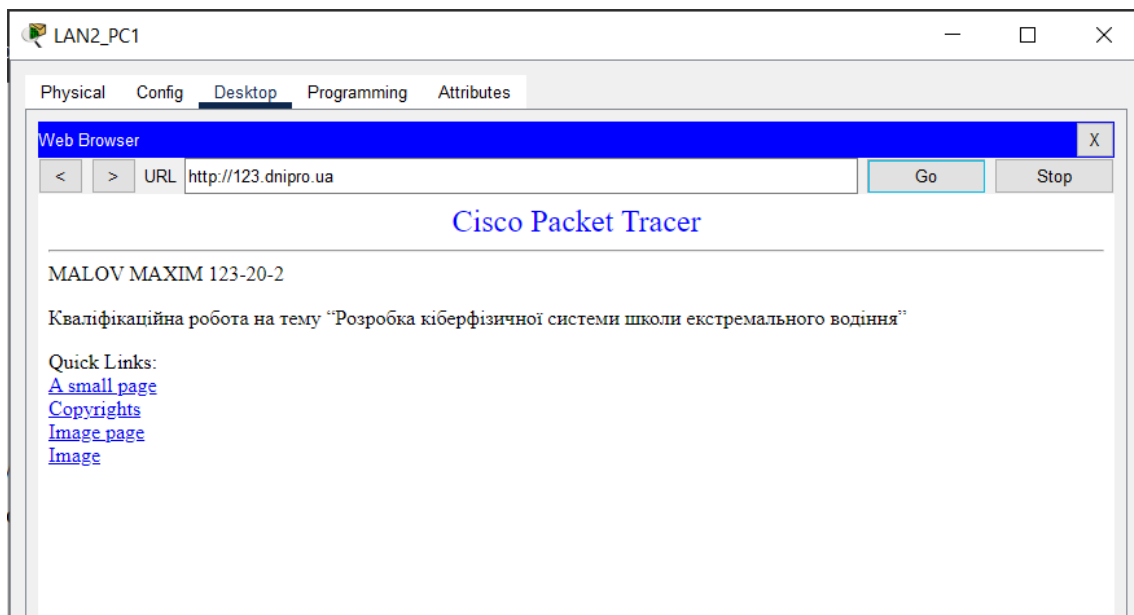


Рисунок 3.14 – Перевірка статичного NAT

Для перевірки роботи VPN була використана команда:  
`show crypto isakmp sa,`  
 що відображає параметри обміну ключами Інтернету (IKE), результат цієї команди зображено на рисунку 3.15.

Malov\_Router\_5 (config)#do show crypto isakmp sa

```

IPv4 Crypto ISAKMP SA
dst          src          state          conn-id slot status
209.165.202.1 64.100.13.2  QM_IDLE       1058    0  ACTIVE

IPv6 Crypto ISAKMP SA

```

Рисунок 3.15 – Перевірка роботи VPN

Також було перевірено роботу VLAN, за допомогою команди `show vlan brief`, що зображено на рисунку 3.16.

Malov\_Switch#

```

Malov_Switch#show vlan brief

VLAN Name                Status    Ports
-----
1    default                 active    Gig0/1
10   Accounting              active    Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                           Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                           Fa0/23, Fa0/24
20   Resources_Department   active    Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                           Fa0/14
30   Guest                  active    Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                           Fa0/9
99   Management             active
100  Native                 active
1002 fddi-default           active
1003 token-ring-default   active
1004 fddinet-default       active
1005 trnet-default        active
Malov_Switch#

```

Рисунок 3.16 – Перевірка VLAN

## **4 РОЗРОБКА КОМПОНЕНТА СИСТЕМИ**

### **4.1 Опис застосунку**

#### **4.1.1 Загальні відомості про застосунок**

Для дистанційного навчання контраварійному керуванню автомобіля та для покращення тренованості учнів було розроблено застосунок на телефон, який дозволяє вивчити принципи контраварійного керування автомобілем. За рахунок цього, кожен учень, не зважаючи на рівень підготовки, може набути навичок та вивчити фізику поведження автомобіля у різних ситуаціях та відпрацювати принципи кермування, щоб уникнути ДТП. Також у застосунок була додана система оцінювання дій гравця, що допомагає створювати рівні та моделювати різні ситуації, справедливо оцінюючи дії учня.

Цей застосунок не є повноцінним симулятором дорожніх обставин та аварійних ситуацій, але його функціонал та фізика дозволяє імітувати велику кількість аварійних ситуацій та додати спортивну складову у навчання учнів. Серед навичок, які учні можуть опанувати є: входження в контрольований занос та керування автомобілем у ньому, оцінку правильності траєкторії, візуальний розрахунок швидкості входження у поворот та багато чого іншого.

У основі цього застосунку стоїть ігровий рушій Unity, який дозволяє створювати застосунки на різні платформи із різними цілями. Платформою ж була обрана Android, через свою розповсюдженість та можливість працювати, у більшості випадків, навіть під час блекаутів, що робить її більш зручною та багатофункціональною, ніж ПК.

### 4.1.2 Платформа Unity

Unity Engine – це безкоштовний інструмент для розробки комп'ютерних ігор і додатків. Наразі він є одним із найпопулярніших у геймдеві для інді-розробників і невеликих компаній. Це пов'язано з його простотою використання, безкоштовною моделю розповсюдження та можливістю легко монетизувати продукт. Він має функціонал для створення програм на різних платформах, таких як Windows, PlayStation, Xbox, Android і iOS. Він може створювати програми як у двомірному, так і в трьохмірному просторі.

Головні переваги цієї платформи – низькій рівень входження у розробку ігор та велике ком'юніті, у якому дуже часто можна знайти вирішення проблем. Також, великою перевагою є безліч так званих «асетів» – наборів готових текстур, скриптів, плагінів. Це допомагає розробникам ділитись своїми напрацюваннями і, так само, користуватись створеними ресурсами інших людей, економлячи час та сили.

Unity має простий, інтуїтивно зрозумілий інтерфейс, який складається із різних вікон, що дозволяє зручно редагувати та налагоджувати гру. Для написання скриптів використовується мова C#, хоча раніше використовувався фреймворк JavaScript, відомий як UnityScript. Для розрахунку 3D фізики використовується рушій PhysX от NVIDIA.

PhysX – фізичний рушій із відкритим кодом, який слугує для симуляції деяких фізичних явищ. [7] Серед головних компонентів обробки фізики можна виділити:

- обробка твердих тіл (англ. *rigid body*);
- обробка тканин (англ. *cloth*);
- обробка рідин (англ. *fluid*).

У даному проекті була проведена робота по опрацюванню документації як рушія Unity, так і рушія PhysX, для розуміння роботи симуляції фізичних моделей.

## 4.2 Розробка проекту

### 4.2.1 Архітектура гри

Для початку треба розібрати, як виглядає робоча зона у рушії Unity. Ліворуч знаходиться вкладка «Hierarchy», у якій знаходяться всі об'єкти, що відносяться до певної сцени. Сцени це об'єкти які містять різні елементи, вони можуть використовуватись як для створення головного меню, так і для створення інших рівнів. Знизу, у вкладці «Project» знаходяться всі файли, що відносяться до цього застосунку, а також завантажені пакети. Праворуч знаходиться «Inspector» у якому відображаються компоненти вибраного об'єкту на сцені. Вкладка «Scene» являє собою сцену, де візуально можна побачити всі об'єкти, перемістити їх, зменшити чи збільшити розмір та усіляко із ними взаємодіяти. На вкладці «Game» відображається те, що бачить гравець у різний момент часу. Одразу може здатись, що це доволі важка в розумінні програма, але насправді це не так.

На рисунку 4.1 зображен вигляд робочої зони проекту в Unity.

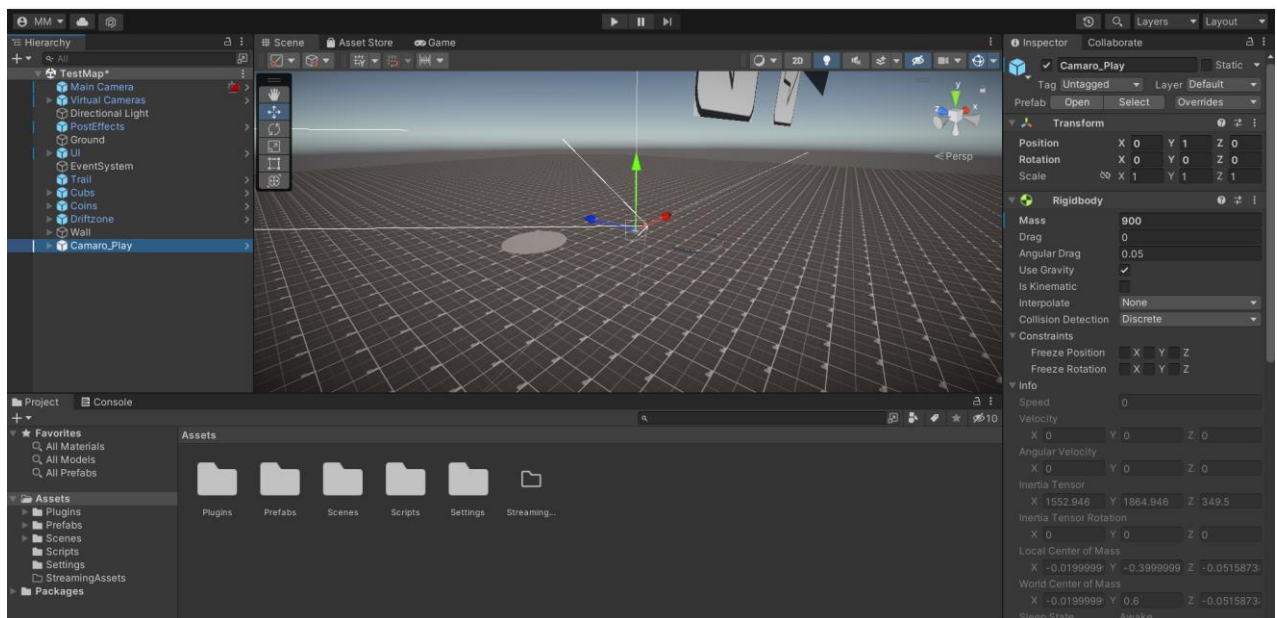


Рисунок 4.1 – Робоча зона Unity

Роздивимось ближче вкладку «Hierarchy». На рисунку 4.2 для прикладу зображен вміст сцени «TestMap». Це головна сцена, по якій гравець рухається і виконує певні завдання, умовний тестовий полігон.

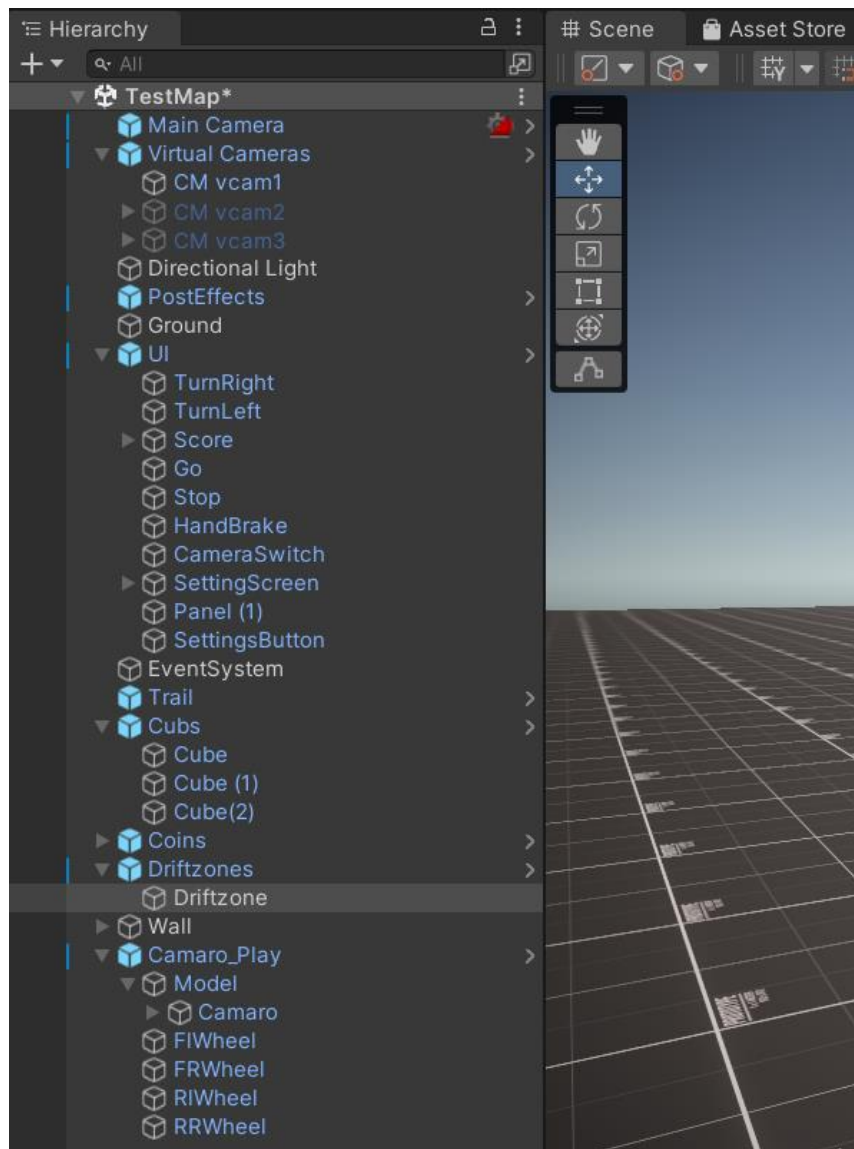


Рисунок 4.2 – Вміст сцени «TestMap»

Об'єкти, які одразу створюються разом із новим проектом – це об'єкти Main Camera та Directional Light. Перший об'єкт це камера, поле зору якої бачить гравець. Другий об'єкт це елемент освітлення, який можна налаштовувати, обирати яскравість, спектр, тип. Задля більш зручного налаштування камери був доданий модуль від Unity під назвою Cinemachine. Він дозволяє створювати та зручно налаштовувати віртуальні камери, стежити за об'єктом, має налаштування демфування, що дуже сильно спрощує роботу в порівнянні із стандартною камерою. У даному проекті було створено 3 віртуальні камери, між якими гравець має змогу перемикатись за допомогою

кнопки перемикання камери. Задля більш гарної картинки було додано та налаштовано елемент PostEffects, який відповідальний за постобробку зображення та додає гарний візуальний стиль. EventSystem відповідальний за доставку натискань на клавіатурі або на екрані телефону до об'єктів застосунку, які це оброблюють. UI це інтерфейс користувача. Він відповідальний за відображення кнопок і інших елементів, які користувач може бачити та із якими він взаємодіє. Можемо побачити, що він містить кнопки керування авто, ручне гальмо, кнопки перемикання камери та налаштування, а також панель меню та відображення очок гравця. Елемент trail це слід від покришок авто, він вмикається та вимикається за необхідністю. Елементи, із якими гравець фізично може взаємодіяти включають:

- ground – платформу, на якій розставлені об'єкти, землю;
- coins – монетки, які гравцю необхідно збивати, для заробляння очок;
- subs – елементи кубів, завдяки яким можна як будувати трасу, так і просто розставляти їх задля більш цікавого наповнення траси;
- driftzones – спеціальні зони для дрифту, у яких гравець набирає очки;
- wall – елементи стіни, які слугують межами траси або відбійниками;
- samago\_play – авто гравця, на якому він повинен взаємодіяти із усіма іншими об'єктами.

Елементи які позначені синім кольором, це глобальні елементи, які зберігаються у папках. Вони мають глобальні характеристики та налаштування, що дозволяє використовувати їх у різних сценах, не створюючи і не налаштовуючи їх із нічого, але при цьому, є можливість змінювати їх під конкретну сцену.



## 4.2.2 Автомобіль

Найголовніша частина застосунку про автошколу це звичайно автомобіль. Для створення авто на робочій сцені розміщується 3D об'єкт куб, який розтягується до потрібної довжини та ширини машини, та на цей об'єкт додаються 4 елементи “Wheel Collider”, які представляють колеса, та розставляються на потрібні місця. Це найпростіша модель автомобіля, який вже може стояти на землі, але не має керування та жодних налаштувань. Його вигляд зображено на рисунку 4.3.

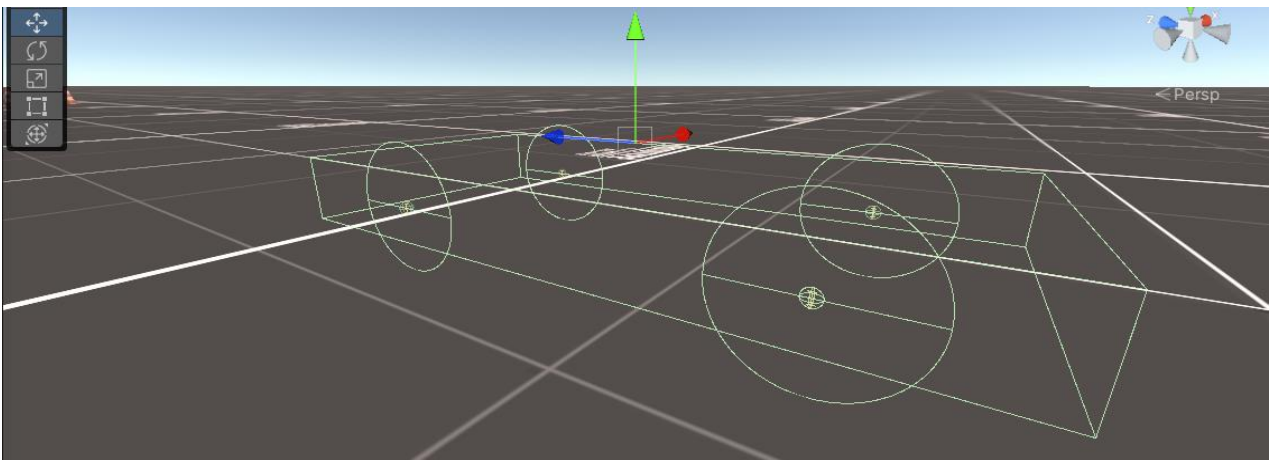


Рисунок 4.3 – Найпростіша модель авто

Після створення «каркасу» авто, на нього була додана візуальна модель і скрипти, які відповідають за керування та ефекти, що зображено на рисунку 4.4.

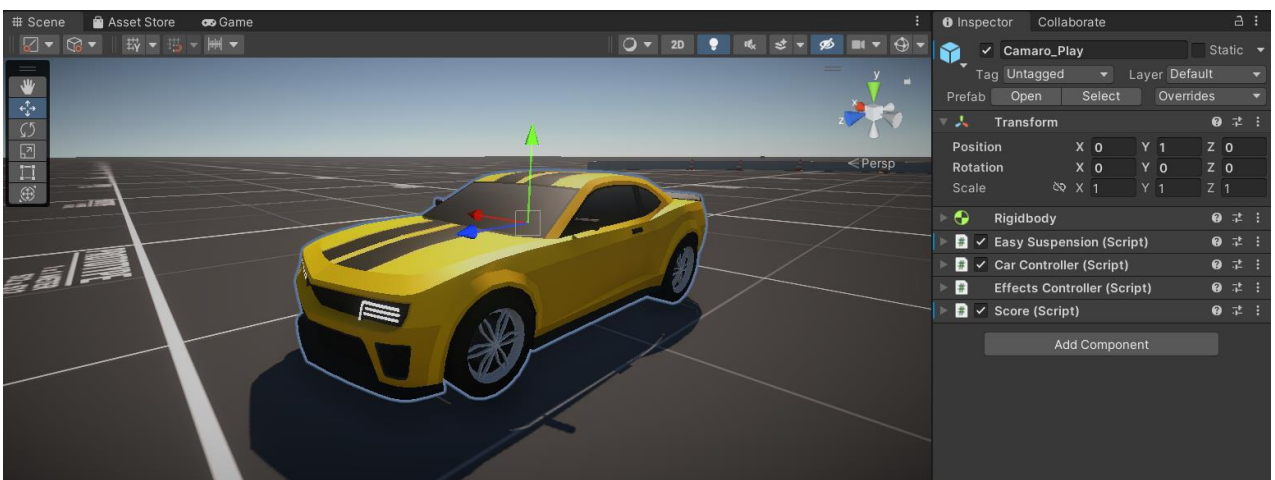


Рисунок 4.4 – Вигляд моделі авто зі скриптами

На об'єкті автомобіль «Сагао\_Play» було додано 2 скрипти, які полегшують налаштування автомобіля та відповідальні за керування і фізику, окрім вбудованого модулю фізики «Rigidbody». Скрипт із назвою Easy Suspension відповідальний за легке налаштування жорсткості пружин та амортизаторів та виставлення фізичного центру колеса. Наступний же скрипт, із назвою Car Controller, повністю дозволяє керувати автомобілем користувачу, також змінює фізичні характеристики під час гри, і відповідає за увімкнення та зміну візуальних ефектів. Його ми розберемо більш детально.

#### 4.2.2.1 Скрипт Car Controller

Для початку у скрипті було об'явлено глобальні змінні, які можна редагувати у режимі реального часу у «Інспекторі», що зображено на рисунках 4.5 та 4.6.

```
public class CarController : MonoBehaviour
{
    [Header("Physics")]

    public float maxAngle = 30; // Maximum steering angle
    public float maxTorque = 300; // Maximum torque for the wheels
    public float maxSpeed = 35; // Maximum speed of the car
    public float brakeforce = 1000; // Maximum brake force
    public float minDriftAngle = 5; // Minimum angle for drift to occur
    [Range(0f, 1f)] public float distributionBrakeForce = 0.3f; // Distribution of brake force between front and rear wheels
    [Range(0, 1f)] public float slipLimit; // Limit of slip before car starts drifting

    public bool InputFromPC; // Toggle between PC and mobile inputs

    [Header("Turning settings")]
    public float turnSpeed = 5f; // Speed of turning
    [Range(0f, 1f)] public float horizontalInputMax = 1f; // Maximum horizontal input value

    [Header("Prefabs")]
    [SerializeField] private Transform[] wheelsPrefab; // Array of wheel transforms for animation
    public GameObject smokePrefab; // Smoke effect prefab
    [SerializeField] public GameObject trailPrefab; // Trail effect prefab

    [HideInInspector] public WheelCollider[] wheels; // Array of wheel colliders
    private Rigidbody _rb; // Rigidbody component of the car
}
```

Рисунок 4.5 – об'явленні глобальних змінних

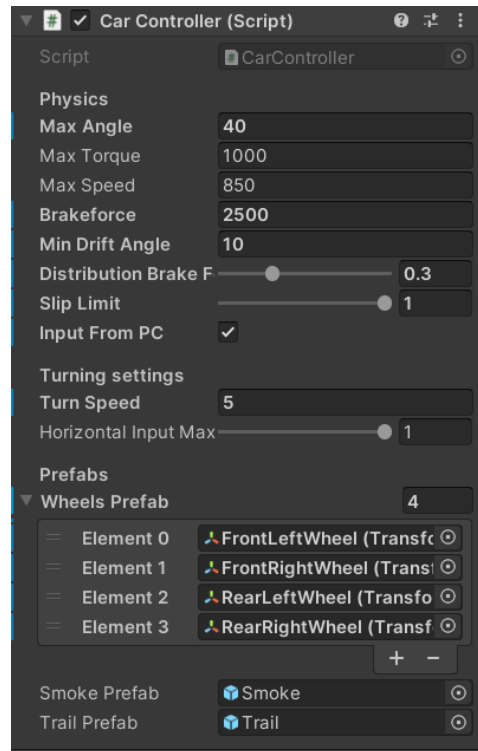


Рисунок 4.6 – Вигляд скрипта у вікні «Інспектора»

Далі об'являємо локальні змінні, та у Update викликаємо необхідні функції, рисунок 4.7

```
private void Update()
{
    // Call various functions to handle car behavior
    CarControl();
    TurnWheels();
    ApplyForceToWheels();
    AnimateWheels();
    ControlParticles();
    StopCar();
    CheckDrift();
    WheelSlip();
    HandBrake();
}
```

Рисунок 4.7 – Функція Update

Тепер розглянемо деякі із цих функцій більш детально.

Функція CarControl відповідальна за прийом та обробку даних від користувача, та передачу цих даних у інші функції, які відповідають за рух

авто. Також ця функція перемикає джерело вхідних даних із ПК на мобільний пристрій, або емулятор мобільного пристрою, як це відбувається у процесі розробки. За допомогою цього, можна зручно налаштовувати керування, фізичні властивості та поведінку автомобіля, не потребуючи кожного разу білдити застосунок та завантажувати його на мобільний пристрій. У цій функції реалізована симуляція обертання керма, тобто автомобіль кермуватись м'яко, що дозволяє плавно входити у повороти на великій швидкості, уникаючи недостатньо повертаємості. Також реалізоване налаштовуєме гальмування по осям, що дозволяє досягати більш ефективного гальмування. Через гальмування автомобіля передня вісь завантажується і має більше зчеплення із поверхнею, а задня вісь навпаки розвантажується, і має менше зчеплення, тому, у даних налаштуваннях підвіски, звичайне відношення гальмівного зусилля передньої до задньої осі як 50/50, не буде настільки ефективним, як наприклад 65/35, яке реалізовано у цьому скрипті.

Нижче, на рисунку 4.8 зображений вигляд данної функції.

```
private void CarControl()
{
    // Variables for car control inputs
    float clampedHorizontalInput = 0;
    float targetHorizontalInput = 0;
    float verticalInput = 0;
    float _handbrake = 0;
    float _brake = 0;
    bool goBackward = false;
    // Mobile input handling
    if (!InputFromPC || Application.platform == RuntimePlatform.Android || Application.platform == RuntimePlatform.IPhonePlayer)
    // PC input handling
    else
    {
        // Adjust torque and brake force for reversing
        if ((_rb.velocity.magnitude <= 10) && goBackward)
        {
            if (_torque > 0)
                _torque = 0;

            _brake = 0;
            verticalInput = -1;
        }

        // Smoothly interpolate horizontal input and clamp it
        _currentHorizontalInput = Mathf.Lerp(_currentHorizontalInput, targetHorizontalInput, Time.deltaTime * turnSpeed);
        clampedHorizontalInput = Mathf.Clamp(_currentHorizontalInput, -horizontalInputMax, horizontalInputMax);

        // Set car's steering angle, torque, and brake forces
        _angle = maxAngle * clampedHorizontalInput;
        _torque = maxTorque * verticalInput;
        _handbrakeforce = 10000 * _handbrake;
        fbrake = _brake * brakeforce * (1 - distributionBrakeForce);
        rbrake = _brake * brakeforce * distributionBrakeForce;
        _goBackward = goBackward;
    }
}
```

Рисунок 4.8 – Функція CarControl

Функція TurnWheels відповідає за повертання передніх коліс та контрруління під час заносу задньої осі. Для передачі обертального моменту на ведучі (задні) колеса та обмеження максимальної швидкості, використовується функція ApplyForceToWheels, код цих двох функцій зображено на рисунку 4.9

```
private void TurnWheels()
{
    // Adjust wheel angles based on car's speed and turning state
    if (wheels[1].rpm > 1 && !_goBackward)
    {
        _angle += (Vector3.SignedAngle(transform.forward, _rb.velocity + transform.forward, Vector3.up)) * 0.65f;
        _angle = Mathf.Clamp(_angle, -50f, 50f);
    }

    foreach (WheelCollider wheel in wheels)
    {
        if (wheel.transform.localPosition.z > 0)
            wheel.steerAngle = _angle;
    }
}

private void ApplyForceToWheels()
{
    // Apply torque to the wheels to control car's speed
    float currentSpeed = _rb.velocity.magnitude;
    if (currentSpeed > maxSpeed)
        _torque = 0;

    foreach (WheelCollider wheel in wheels)
    {
        if (wheel.transform.localPosition.z < 0)
            wheel.motorTorque = _torque;
    }
}
```

Рисунок 4.9 – Функції TurnWheels та ApplyForceToWheels

Для того, щоб ефективно зупинити автомобіль, та, за необхідності, пустити його в контрольований занос було реалізовані функції гальмування та ручного тормозу. Гальмування колес реалізоване шляхом використання змінної brakeTorque класу «WheelCollider». Також при використанні ручного гальма, задля імітації реальних шин автомобіля, було реалізовано зміну властивостей елемента «WheelCollider», а саме зменшення бокового тертя шин із поверхнею для кращого сковзання. Це реалізується завдяки програмній зміні extremumSlip та asymptoteSlip у sidewaysFriction. Функції StopCar та HandBrake наведені на рисунку 4.10

```

private void StopCar()
{
    foreach (WheelCollider wheel in wheels) // Apply brake force to the wheels to stop the car
    {
        if (wheel.transform.localPosition.z > 0) // Front wheel brake
            wheel.brakeTorque = fbrake;

        if (wheel.transform.localPosition.z < 0) // Rear wheel brake
            wheel.brakeTorque = rbrake;
    }
}

private void HandBrake()
{
    foreach (WheelCollider wheel in wheels) // Apply handbrake force to the rear wheels
    {
        if (wheel.transform.localPosition.z < 0)
        {
            if (rbrake <= 0 && _handbrakeforce > 0)
            {
                wheel.brakeTorque = _handbrakeforce;
                WheelFrictionCurve R_Sd_friction = wheel.sidewaysFriction;
                R_Sd_friction.extremumSlip = 1.4f;
                R_Sd_friction.asymptoteSlip = 1f;
                wheel.sidewaysFriction = R_Sd_friction;
            }
        }
    }
}

```

Рисунок 4.10 – Функції StopCar та HandBrake

Було написано функції по зміні зчеплення коліс із поверхнею, дивлячись на те, знаходиться машина в стані дрифту чи ні. Також була написана сама функція, яка перевіряє та визначає, чи машина у дрифті у даний момент. Реалізовано це шляхом перевірки відхилення кута машини відносно вектора руху. Далі, у функції WheelSlip відбувається зміна extremumSlip, extremumValue, asymptoteSlip та asymptoteValue для кожної вісі автомобіля. За допомогою даної функції реалізоване зручне кермування на низьких швидкостях, та контрольована та прогнозована поведінка автомобіля під час заносу.

Задля наглядності, було додано сліди шин, за допомогою створення об'єкта «Trail», який містить налаштований «Trail Render», та диму, який був створений за допомогою компонента «Particle System». У скрипті була написана функція ControlParticles, яка вмикає сліди від шин та дим відповідно до втрати зчеплення коліс із поверхнею. Ця функція зображена на рисунку 4.11

```

private void ControlParticles()
{
    // Control visual effects based on wheel slip
    for (int i = 0; i < wheels.Length; ++i)
    {
        if (wheels[i].GetGroundHit(out var hit))
        {
            var isEnabled = Mathf.Abs(hit.forwardSlip) + Mathf.Abs(hit.sidewaysSlip) >= slipLimit;

            _effectsController.ControlSmoke(i, isEnabled);
            _effectsController.ControlTrails(i, isEnabled);
        }
    }
}

```

Рисунок 4.11 – Функція ControlParticles

Також реалізована анімація обертання та кермування коліс шляхом присвоєння моделі колеса позиції та обертання об'єкту колеса у функції `animateWheels`.

### 4.2.3 Система оцінки дій користувача

Задля об'єктивної та чесної оцінки дій користувача було введено просту, але, на даний момент, доволі ефективну систему оцінювання. Вона являє собою набір елементів, із якими користувач взаємодіє деяким чином, за що відповідно нараховуються бали, які відображаються у лівому верхньому куті. Ці елементи можливо розставляти по мапі, формуючи різні завдання та ситуації, що дозволяє освоїти багато аспектів кермування авто. Розберемо більш детально кожен із компонентів:

- дорожні конуси – це моделька конусу, яка при наїзду на неї машини зникає та нараховує(або віднімає) певну кількість очок, за замовчуванням 100. За необхідності можна розставити конуси таким чином, щоб водій міг практикувати різні види парковки, і в залежності від наїзду на конус у нього віднімалися бали. Або навпаки, викласти певну траєкторію із конусів, і при проїзду автомобіля по заданій траєкторії та наїзду на конуси, зараховується певна кількість очок;
- дрифт-зона. Являє собою коло, при заїзді в яке машиною та виконанні деяких умов нараховуються певні бали. Якщо авто входить у коло, та знаходиться у стані дрифту, але при цьому має певну швидкість, за одиницю часу поступово нараховуються бали. Швидкість авто, кут дрифту, та швидкість нараховування очок можна легко налаштувати в редакторі.

За нарахування балів відповідальний скрипт `Score`. У основі його стоїть можливість рушія зчитувати, коли колайдери різних об'єктів торкаються один одного. За допомогою цього було написано, що коли об'єкт автомобіля торкається колайдеру конусу, та зникає, а гравцю нараховується задана

статична кількість балів. У дрифт зоні, завдяки формулі, що зображена на рисунку 4.12 вираховується кут авто відносно модуля його руху, через що можна зрозуміти, чи йде машина в заносі, чи ні.

```
// Calculate the angle between the player's velocity and the forward direction
float driftValue = Vector3.Dot(rb.velocity.normalized, transform.forward.normalized);
float driftAngle = Mathf.Acos(driftValue) * Mathf.Rad2Deg;
driftAngle = Mathf.Clamp(driftAngle, -80f, 80f); // Clamp the drift angle between -80 and 80 degrees
```

Рисунок 4.12 – Вирахування кута зносу задньої осі

Після цього, вираховується швидкість авто, і якщо вона більше допустимого значення, і, при цьому колайдер автомобіля торкається колайдеру дрифт зони, то гравцю рахується нарахування очок, за формулою

$$\text{score} += \text{driftScorePerSecond} * \text{Time.deltaTime};$$

І тільки потім гравцю виводяться набрані очки.

#### 4.2.4 Вибір та наповнення мап

Мапа TestMap представляє із себе просту рівну поверхню, на якій є такі елементи оточення:

- дорожні конуси;
- стіни;
- дріфт зони;
- куби.

Усі ці елементи можуть взаємодіяти з гравцем, що додає динаміки та інтерактивності ігровому процесу. Куби мають фізичні властивості, що дозволяють їм рухатися при зіткненні з автомобілем гравця, створюючи ефект реалістичності. Стіни є статичними об'єктами, які обмежують трасу та служать перешкодами, з якими гравець може зіткнутися. Дріфт зони та дорожні конуси відіграють важливу роль у системі підрахунку балів, стимулюючи гравця до точного та акуратного водіння.



Всі ці елементи разом створюють унікальне ігрове середовище, що сприяє різноманітним випробуванням для гравця. Вигляд усіх цих елементів на мапі TestMap представлений на рисунку 4.13, де можна побачити їхнє розташування та взаємодію в контексті загальної структури мапи. Це надає гравцям повне уявлення про те, з чим вони зіткнуться під час гри, і як ці елементи впливатимуть на їхній досвід.

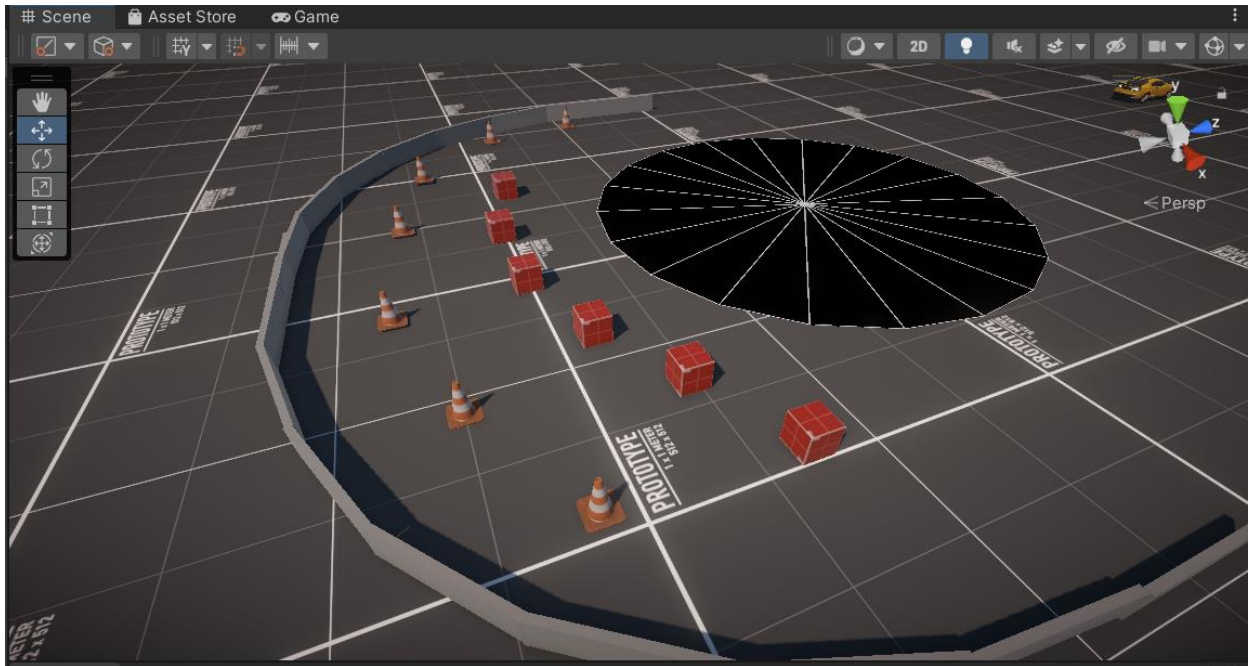


Рисунок 4.13 – Зображення елементів гри

Мапа TerrainMap представляє із себе кільцеву трасу по засніженому горному серпантину, яка має сильні перепади висот, 4 полоси для руху та багато поворотів різних кутів. Тренуючись на цій мапі, учень матиме змогу відпрацювати проходження поворотів як у великому куті заносу, так і у малому, але довгому. Також, цей трек дає можливість зрозуміти, як перепади висот впливають на керованість авто. Вигляд цього треку зображено на рисунку 4.14.

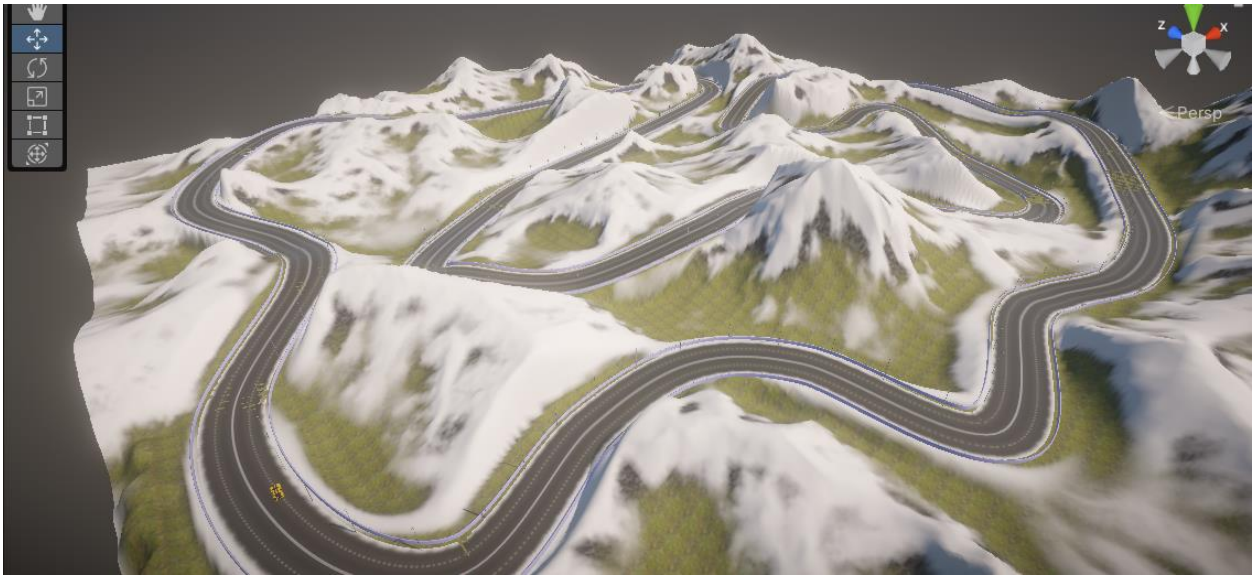


Рисунок 4.14 – Зображення треку TerrainMap

#### 4.2.5 Інтерфейс користувача

Інтерфейс користувача(UI) це візуальні елементи із якими користувач взаємодіє, призначені для виводу або вводу інформації користувачем. До них відносяться кнопки, виводи тексту, анімації, слайдери, тощо. У головному меню є кнопка вибору мап із назвою Start, рисунок 4.15. При натисканні на цю кнопку з'являється меню вибору мап, на даний момент доступно 2 мапи, Test Map та TerrainMap рисунок 4.16. При натисканні на будь-яку із цих мап завантажується відповідний рівень. Також є кнопка exit, яка закриває меню мап. При запуску гри та при переході на іншу мапу з'являється анімація.



Рисунок 4.15 – Головне меню

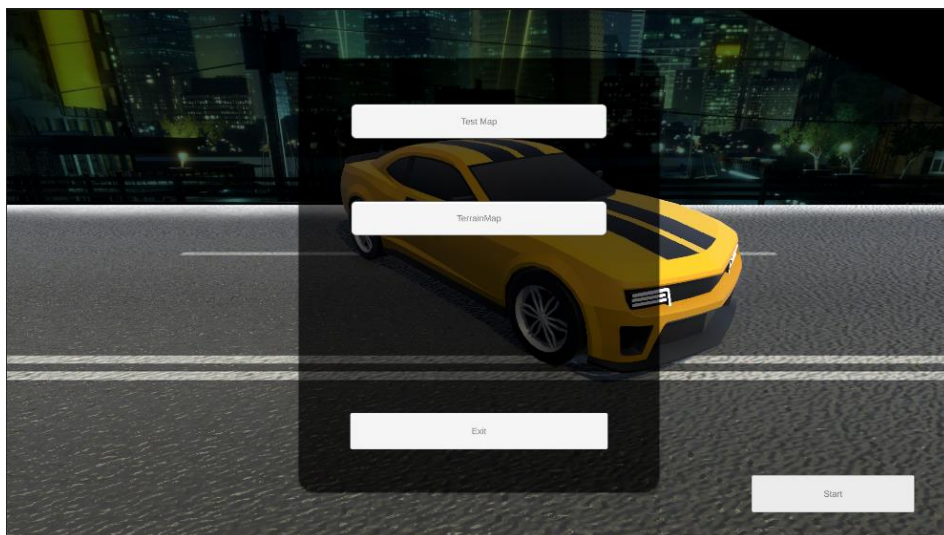


Рисунок 4.16 – Меню вибору мап

На сценах різних мап однакова кількість графічного інтерфейсу, відрізняється лиш фізичне наповнення цих рівнів. При завантаженні також програватиметься анімація, а саме плавне відкриття чорного квадрату, яке зкатується ліворуч. За анімації відповідає окремий скрипт під назвою Menu Animation Controller. Вигляд інтерфейсу користувача на мапі представлено на рисунку 4.17.

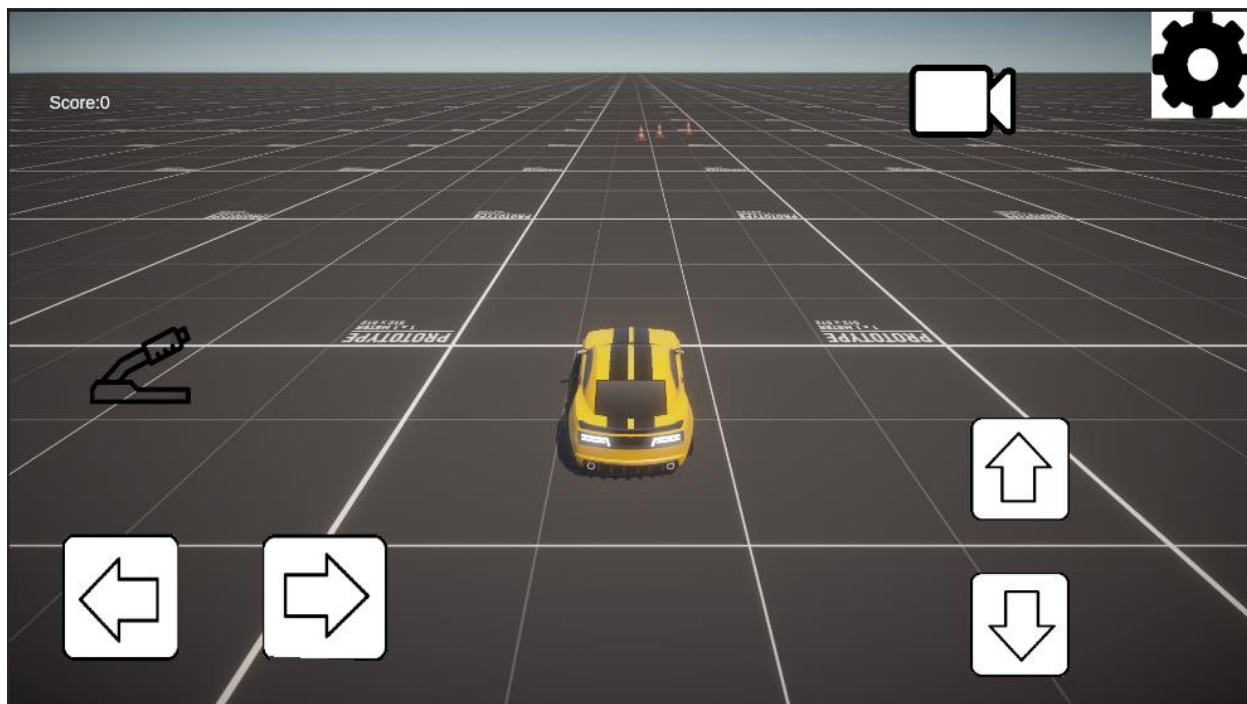


Рисунок 4.17 – Графічний інтерфейс на мапі TestMap

У якості об'єктів графічного інтерфейсу були додані кнопки керування автомобілем, а саме вперед, назад, ліворуч, праворуч, та кнопка ручного гальма. Також було додано кнопку зміни камери та кнопку входу у меню. Ці кнопки опрацьовуються скриптом UI Controller, який передає дані про натискання кнопок керування у скрипт CarController, опрацьовує дані про натискання кнопки перемикавання камери та перемикає віртуальні камери, створені за рахунок плагіну Cinemachine. Окрім цього, цей скрипт відповідальний за відкриття меню. У меню була створена кнопка повернення у головне меню та кнопка закриття, якщо користувач передумав. Також гра призупиняється, при відкритті даного вікна, що дозволяє ставити таким чином ставити її на паузу. Вигляд меню зображений на рисунку 4.18.

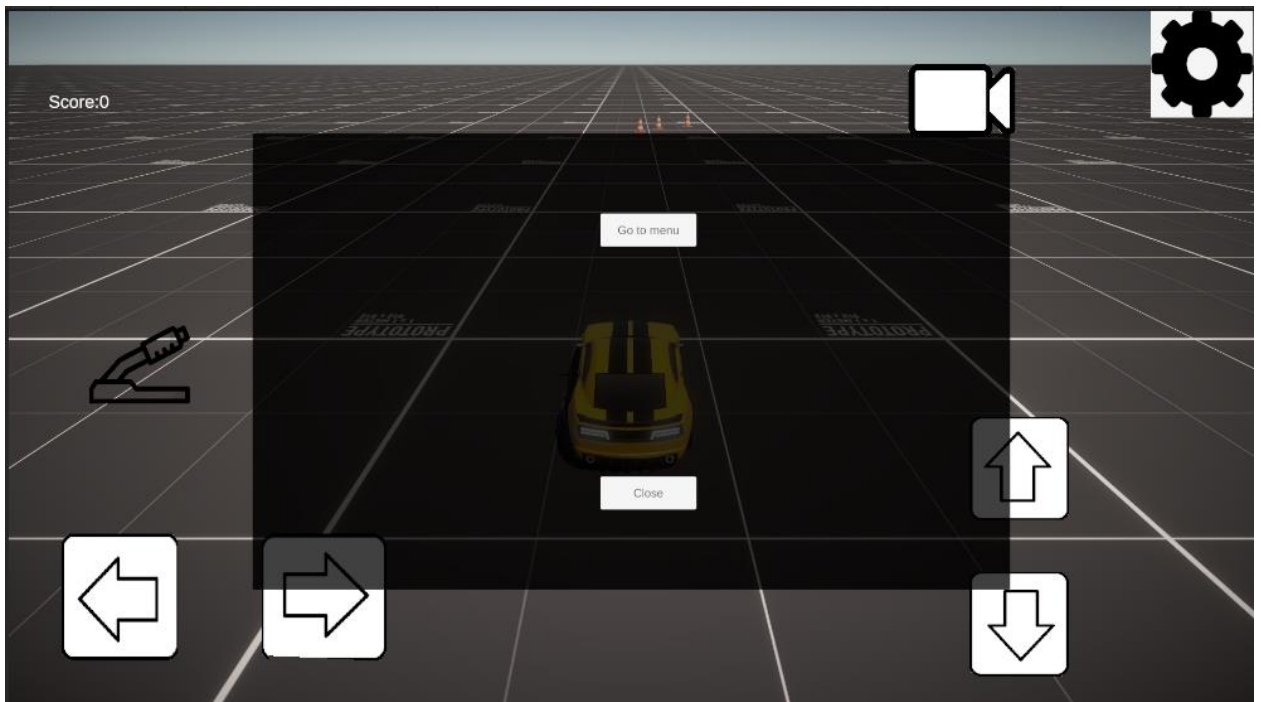


Рисунок 4.18 – Вигляд меню

Окрім цього, у лівому верхньому куті можна помітити текст Score, що означає кількість очок. Цей текст прив'язаний до скрипта Score, який відповідальний за підрахунок очок гравця. Це дозволяє гравцю у реальному часі бачити та розуміти які дії призводять до збільшення очок, а які ні.

## ВИСНОВКИ

У процесі виконання дипломної роботи було досягнуто головної мети – розробка кіберфізичної системи для школи екстремального водіння. Реалізація цього завдання включала декілька ключових етапів, результатом яких стали розробка комп'ютерної мережі для забезпечення інтерактивності навчального процесу та розробка мобільного застосунку-симулятора.

Було розроблено комп'ютерну мережу для школи екстремального водіння, яка забезпечує ефективну взаємодію між різними компонентами кіберфізичної системи. Мережа включає сервери, комутатори, маршрутизатори та персональні комп'ютери, що дозволяють викладачам та адміністраторам школи ефективно контролювати та керувати навчальним процесом. Висока пропускна здатність та надійність мережевого обладнання гарантують стабільну роботу системи та швидку передачу даних між усіма її частинами.

Завдяки використанню рішень від компанії Cisco, вдалося забезпечити високу якість та безпеку мережі, що є критично важливим для безперервного та безпечного функціонування школи. Це включає інтеграцію систем захисту від кіберзагроз та можливість масштабування мережі у майбутньому, що дозволяє адаптуватися до зростання кількості користувачів та розширення інфраструктури школи.

Також був розроблений застосунок для симуляції екстремального водіння на мобільну платформу, який надає користувачам можливість відпрацьовувати та вдосконалювати свої навички водіння. Це дозволяє користувачам безпечно тренуватися та покращувати свої навички водіння у віртуальному середовищі, що суттєво знижує ризик аварій на реальних дорогах.

Таким чином, дипломна робота досягла своєї мети і продемонструвала ефективність розроблених рішень.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ**

1. «Скільки коштує отримати права» [Електронний ресурс] – Режим доступу до ресурсу:  
<https://www.epravda.com.ua/publications/2023/05/15/700024/>
2. «Learning management systems» [Електронний ресурс] – Режим доступу до ресурсу:  
[https://web.archive.org/web/20100331234949/http://www.astd.org/NR/rdonlyres/12ECDB99-3B91-403E-9B15-7E597444645D/23395/LMS\\_fieldguide\\_20091.pdf](https://web.archive.org/web/20100331234949/http://www.astd.org/NR/rdonlyres/12ECDB99-3B91-403E-9B15-7E597444645D/23395/LMS_fieldguide_20091.pdf)
3. «Призер 24 годин Ле-Ману став пілотом через автосимулятор» [Електронний ресурс] – Режим доступу до ресурсу:  
<https://ua.tribuna.com/uk/blogs/gamesnp/3007531/>
4. «Освіта в смартфоні - топ-10 ідей застосунків для студентів» [Електронний ресурс] – Режим доступу до ресурсу:  
<https://kitapp.pro/uk/osvita-v-smartfoni-top-10-idej-zastosunkiv-dlya-studentiv-ta-shkolyariv/>
5. «VLSM Calculator» [Електронний ресурс] – Режим доступу до ресурсу: <https://subnettingpractice.com/vlsm.html>
6. «Що таке NAT» [Електронний ресурс] – Режим доступу до ресурсу: <https://hyperhost.ua/info/uk/shcho-take-nat-dlya-chogo-vikoristovuyu-daniy-standart>
7. «NVIDIA PhysX» [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.nvidia.com/physx-sdk?nvid=nv-int-solr-911757-vt37>

## ДОДАТОК А

Текст програми мобільного застосунку для школи екстремального водіння.

**Міністерство освіти і науки України**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**  
**“ДНІПРОВСЬКА ПОЛІТЕХНІКА”**

**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ**  
**МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ШКОЛИ ЕКСТРЕМАЛЬНОГО**  
**ВОДІННЯ**

Текст програми

804.02070743.24010-01 12 01

Листів 20



## АНОТАЦІЯ

Дана програма містить в собі програмний код мобільного застосунку для школи екстремального водіння.

Програма призначена для навчання користувачів основам екстремального водіння та симуляції різних аварійних ситуацій.

Текст програми написаний мовою C# відлагоджена із застосуванням середовища Visual Studio.

**Зміст**

<b>Скрипт CarController .....</b>	<b>4</b>
<b>Скрипт UIController .....</b>	<b>17</b>

## Скрипт CarController

```

using UnityEngine;

namespace Car
{
    public class CarController : MonoBehaviour
    {
        [Header("Physics")]

        public float maxAngle = 30; // Maximum steering angle
        public float maxTorque = 300; // Maximum torque for the wheels
        public float maxSpeed = 35; // Maximum speed of the car
        public float brakeforce = 1000; // Maximum brake force
        public float minDriftAngle = 5; // Minimum angle for drift to occur
        [Range(0f, 1f)] public float distributionBrakeForce = 0.3f; // Distribution of
brake force between front and rear wheels
        [Range(0, 1f)] public float slipLimit; // Limit of slip before car starts drifting

        public bool InputFromPC; // Toggle between PC and mobile inputs

        [Header("Turning settings")]
        public float turnSpeed = 5f; // Speed of turning
        [Range(0f, 1f)] public float horizontalInputMax = 1f; // Maximum horizontal
input value

        [Header("Prefabs")]

```

```

[SerializeField] private Transform[] wheelsPrefab; // Array of wheel
transforms for animation

public GameObject smokePrefab; // Smoke effect prefab

[SerializeField] public GameObject trailPrefab; // Trail effect prefab

[HideInInspector] public WheelCollider[] wheels; // Array of wheel colliders
private Rigidbody _rb; // Rigidbody component of the car

[HideInInspector] public bool go, brake, isTurningLeft, isTurningRight,
handbrake; // Control flags

private float _currentHorizontalInput; // Current horizontal input value

private float _angle; // Current steering angle
private float _torque; // Current torque
private float _handbrakeforce; // Current handbrake force
private float fbrake; // Front brake force
private float rbrake; // Rear brake force
private bool _goBackward; // Flag indicating if car is going backward
private float staticExtremumSlip; // Static friction curve value
private float staticAsymptoteSlip; // Static friction curve value
private bool inDrift; // Flag indicating if car is drifting
private EffectsController _effectsController; // Reference to effects controller
for managing visual effects

private void Start()
{
    _effectsController = GetComponent<EffectsController>();
    _rb = GetComponent<Rigidbody>();
    wheels = GetComponentsInChildren<WheelCollider>();

```

```

    _effectsController.SpawnSmokes(smokePrefab, wheels);
    _effectsController.SpawnTrails(trailPrefab, wheels);
}

```

```

private void Update()
{
    // Call various functions to handle car behavior
    CarControl();
    TurnWheels();
    ApplyForceToWheels();
    AnimateWheels();
    ControlParticles();
    StopCar();
    CheckDrift();
    WheelSlip();
    HandBrake();
}

```

```

public void CheckDrift()
{
    // Check if car is drifting based on the angle between its forward direction
and its velocity
    if (wheels[2].rpm > 1)
    {
        float driftAngle = (Vector3.SignedAngle(transform.forward, _rb.velocity
+ transform.forward, Vector3.up)) * 0.65f;
        driftAngle = Mathf.Abs(Mathf.Clamp(driftAngle, -80f, 80f));

        inDrift = driftAngle > minDriftAngle;
    }
}

```

```

}

private void WheelSlip()
{
    // Adjust wheel friction based on whether the car is drifting
    if (inDrift)
    {
        foreach (WheelCollider wheel in wheels)
        {
            if (wheel.GetGroundHit(out var hit))
            {
                if (Mathf.Abs(hit.forwardSlip) + Mathf.Abs(hit.sidewaysSlip) >=
slipLimit)
                {
                    if (wheel.transform.localPosition.z > 0)
                    {
                        WheelFrictionCurve F_Fr_friction = wheel.forwardFriction;
                        F_Fr_friction.extremumSlip = 0.8f;
                        F_Fr_friction.extremumValue = 1f;
                        F_Fr_friction.asymptoteSlip = 0.65f;
                        F_Fr_friction.asymptoteValue = 0.5f;
                        wheel.forwardFriction = F_Fr_friction;

                        WheelFrictionCurve F_Sd_friction = wheel.sidewaysFriction;
                        F_Sd_friction.extremumSlip = 0.7f;
                        F_Sd_friction.extremumValue = 1.2f;
                        F_Sd_friction.asymptoteSlip = 0.5f;
                        F_Sd_friction.asymptoteValue = 1f;
                        wheel.sidewaysFriction = F_Sd_friction;
                    }
                }
            }
        }
    }
}

```

```

if (wheel.transform.localPosition.z < 0)
{
    WheelFrictionCurve R_Fr_friction = wheel.forwardFriction;
    R_Fr_friction.extremumSlip = 0.8f;
    R_Fr_friction.extremumValue = 2f;
    R_Fr_friction.asymptoteSlip = 0.75f;
    R_Fr_friction.asymptoteValue = 2f;
    wheel.forwardFriction = R_Fr_friction;

    WheelFrictionCurve R_Sd_friction = wheel.sidewaysFriction;
    R_Sd_friction.extremumSlip = 1f;
    R_Sd_friction.extremumValue = 0.75f;
    R_Sd_friction.asymptoteSlip = 0.8f;
    R_Sd_friction.asymptoteValue = 1f;
    wheel.sidewaysFriction = R_Sd_friction;
}
}
}
}
else
{
    // Reset wheel friction to default values when not drifting
    foreach (WheelCollider wheel in wheels)
    {
        if (wheel.transform.localPosition.z > 0)
        {
            WheelFrictionCurve F_Fr_friction = wheel.forwardFriction;
            F_Fr_friction.extremumSlip = 0.6f;

```

```

F_Fr_friction.extremumValue = 1f;
F_Fr_friction.asymptoteSlip = 0.45f;
F_Fr_friction.asymptoteValue = 1f;
wheel.forwardFriction = F_Fr_friction;

```

```

WheelFrictionCurve F_Sd_friction = wheel.sidewaysFriction;
F_Sd_friction.extremumSlip = 0.5f;
F_Sd_friction.extremumValue = 1f;
F_Sd_friction.asymptoteSlip = 0.5f;
F_Sd_friction.asymptoteValue = 1f;
wheel.sidewaysFriction = F_Sd_friction;

```

```

}

```

```

if (wheel.transform.localPosition.z < 0)

```

```

{

```

```

WheelFrictionCurve R_Fr_friction = wheel.forwardFriction;
R_Fr_friction.extremumSlip = 0.4f;
R_Fr_friction.extremumValue = 2.5f;
R_Fr_friction.asymptoteSlip = 0.45f;
R_Fr_friction.asymptoteValue = 2f;
wheel.forwardFriction = R_Fr_friction;

```

```

WheelFrictionCurve R_Sd_friction = wheel.sidewaysFriction;
R_Sd_friction.extremumSlip = 0.5f;
R_Sd_friction.extremumValue = 2.5f;
R_Sd_friction.asymptoteSlip = 0.4f;
R_Sd_friction.asymptoteValue = 2f;
wheel.sidewaysFriction = R_Sd_friction;

```

```

}

```

```

}

```



```
    }  
}  
  
private void CarControl()  
{  
    // Variables for car control inputs  
    float clampedHorizontalInput = 0;  
    float targetHorizontalInput = 0;  
    float verticalInput = 0;  
    float _handbrake = 0;  
    float _brake = 0;  
    bool goBackward = false;  
    // Mobile input handling  
    if (!InputFromPC || Application.platform == RuntimePlatform.Android ||  
Application.platform == RuntimePlatform.IPhonePlayer)  
    {  
  
        if (go)  
        {  
            verticalInput = 1;  
        }  
  
        if (brake)  
        {  
            _brake = 1;  
            goBackward = true;  
        }  
  
        if (handbrake)  
        {
```

```
        verticalInput = 0;
        _handbrake = 1;
    }

    targetHorizontalInput = isTurningLeft ? -1f : (isTurningRight ? 1f : 0f);
}
// PC input handling
else
{

    targetHorizontalInput = Input.GetAxis("Horizontal");

    if (Input.GetKey("w"))
    {
        verticalInput = 1;
    }

    if (Input.GetKey("s"))
    {
        _brake = 1;
        goBackward = true;
    }

    if (Input.GetKey("space"))
    {
        _handbrake = 1;
    }
}

// Adjust torque and brake force for reversing
```

```

if ((_rb.velocity.magnitude <= 10) && goBackward)
{
    if (_torque > 0)
        _torque = 0;

    _brake = 0;
    verticalInput = -1;

}

// Smoothly interpolate horizontal input and clamp it
_currentHorizontalInput = Mathf.Lerp(_currentHorizontalInput,
targetHorizontalInput, Time.deltaTime * turnSpeed);
clampedHorizontalInput = Mathf.Clamp(_currentHorizontalInput, -
horizontalInputMax, horizontalInputMax);

// Set car's steering angle, torque, and brake forces
_angle = maxAngle * clampedHorizontalInput;
_torque = maxTorque * verticalInput;
_handbrakeforce = 10000 * _handbrake;
fbrake = _brake * brakeforce * (1 - distributionBrakeForce);
rbrake = _brake * brakeforce * distributionBrakeForce;
_goBackward = goBackward;
Debug.Log(Time.deltaTime);
}

private void ControlParticles()
{
    // Control visual effects based on wheel slip
    for (int i = 0; i < wheels.Length; ++i)

```

```

    {
        if (wheels[i].GetGroundHit(out var hit))
        {
            var isEnabled = Mathf.Abs(hit.forwardSlip) +
Mathf.Abs(hit.sidewaysSlip) >= slipLimit;

            _effectsController.ControlSmoke(i, isEnabled);
            _effectsController.ControlTrails(i, isEnabled);
        }
    }
}

private void TurnWheels()
{
    // Adjust wheel angles based on car's speed and turning state
    if (wheels[1].rpm > 1 && !_goBackward)
    {
        _angle += (Vector3.SignedAngle(transform.forward, _rb.velocity +
transform.forward, Vector3.up)) * 0.65f;
        _angle = Mathf.Clamp(_angle, -50f, 50f);
    }

    foreach (WheelCollider wheel in wheels)
    {
        if (wheel.transform.localPosition.z > 0)
            wheel.steerAngle = _angle;
    }
}

private void ApplyForceToWheels()

```

```

{
    // Apply torque to the wheels to control car's speed
    float currentSpeed = _rb.velocity.magnitude;
    if (currentSpeed > maxSpeed)
        _torque = 0;

    foreach (WheelCollider wheel in wheels)
    {
        if (wheel.transform.localPosition.z < 0)
            wheel.motorTorque = _torque;
    }
}

private void StopCar()
{
    foreach (WheelCollider wheel in wheels) // Apply brake force to the wheels
to stop the car
    {
        if (wheel.transform.localPosition.z > 0) // Front wheel brake
        {
            wheel.brakeTorque = fbrake;
        }

        if (wheel.transform.localPosition.z < 0) // Rear wheel brake
        {
            wheel.brakeTorque = rbrake;
        }
    }
}

```

```

private void HandBrake()
{

    foreach (WheelCollider wheel in wheels) // Apply handbrake force to the
rear wheels
    {
        if (wheel.transform.localPosition.z < 0)
        {
            if (rbrake <= 0 && _handbrakeforce > 0)
            {
                wheel.brakeTorque = _handbrakeforce;
                WheelFrictionCurve R_Sd_friction = wheel.sidewaysFriction;
                R_Sd_friction.extremumSlip = 1.4f;
                R_Sd_friction.asymptoteSlip = 1f;
                wheel.sidewaysFriction = R_Sd_friction;
            }
        }
    }
}

```

```

private void AnimateWheels()
{
    // Animate the wheels based on their positions and rotations
    for (int i = 0; i < 4; i++)
    {
        var wheel = wheels[i];
        var wheelTransform = wheelsPrefab[i];

        wheel.GetWorldPose(out var p, out var q);
    }
}

```

```
wheelTransform.position = p;
wheelTransform.rotation = q;
}
}

// Functions to handle control input flags
public void carGo() => go = true;
public void stopGo() => go = false;

public void carBrake() => brake = true;
public void stopBrake() => brake = false;

public void handBrake() => handbrake = true;
public void stophandbrake() => handbrake = false;

public void TurningLeft() => isTurningLeft = true;
public void StopsTurningLeft() => isTurningLeft = false;

public void TurningRight() => isTurningRight = true;
public void StopsTurningRight() => isTurningRight = false;
}
}
```

### Скрипт UIController

```
using System.Collections;
using System.Collections.Generic;
using Car;
using UnityEngine;
using Cinemachine;
using UnityEngine.EventSystems;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class UIController : MonoBehaviour
{
    public Button settingsButton;
    public Button Close;
    public Button GoToMainMenu;
    public Button SwitchCameraButton;

    public GameObject settingScreen;
    [Header("Controls Input")] public GameObject goButton;
    public GameObject brakeButton;
    public GameObject turnLeftButton;
    public GameObject turnRightButton;
    public GameObject HandbrakeButton;
    public GameObject car;
    [SerializeField]
    private CinemachineVirtualCamera[] virtualCameras;
    private int currentCameraIndex;
```



```
public Animator animator;

private CarController _carController;
private bool ifSettingScreenOpen = false;

void Start()
{
    settingScreen.SetActive(false);
    SceneLoaderHelper.SceneLoadedWithAnimation(this, animator);

    _carController = car.GetComponent<CarController>();

    SwitchCameraButton.onClick.AddListener(Switchcamera);
    settingsButton.onClick.AddListener(OpenSettings);
    Close.onClick.AddListener(CloseSetting);
    GoToMainMenu.onClick.AddListener(LoadMainMenu);

    OnEventHelper.HoldingDown(turnLeftButton, _carController.TurningLeft);
    OnEventHelper.HoldingUp(turnLeftButton,
_carController.StopsTurningLeft);
    OnEventHelper.HoldingDown(turnRightButton,
_carController.TurningRight);
    OnEventHelper.HoldingUp(turnRightButton,
_carController.StopsTurningRight);
    OnEventHelper.HoldingDown(goButton, _carController.carGo);
    OnEventHelper.HoldingUp(goButton, _carController.stopGo);
    OnEventHelper.HoldingDown(brakeButton, _carController.carBrake);
    OnEventHelper.HoldingUp(brakeButton, _carController.stopBrake);
    OnEventHelper.HoldingDown(HandbrakeButton, _carController.handBrake);
```

```
        OnEventHelper.HoldingUp(HandbrakeButton,
        _carController.stophandbrake);

    }

    public void Switchcamera()
    {
        virtualCameras[currentCameraIndex].gameObject.SetActive(false);
        currentCameraIndex++;
        if (currentCameraIndex >= virtualCameras.Length)
            currentCameraIndex = 0;
        virtualCameras[currentCameraIndex].gameObject.SetActive(true);
    }

    private void OpenSettings()
    {
        if (!ifSettingScreenOpen)
        {
            settingScreen.SetActive(true);
            Time.timeScale = 0f;
        }

        ifSettingScreenOpen = true;
    }

    private void CloseSetting()
    {
        if (ifSettingScreenOpen)
        {
            settingScreen.SetActive(false);
            Time.timeScale = 1f;
        }
    }
}
```

```
    }  
  
    ifSettingScreenOpen = false;  
}  
  
private void LoadMainMenu()  
{  
    Time.timeScale = 1f;  
    SceneLoaderHelper.LoadSceneWithAnimation(this, animator,  
Scenes.mainMenu);  
}  
}
```