

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО
ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ
Навчально-науковий інститут кібернетики, інформаційних технологій та
інженерії
Кафедра комп'ютерних наук та прикладної математики

“До захисту допущений”
Зав. Кафедри комп'ютерних наук та
прикладної математики

«___» _____ 20__ р.

КВАЛІФІКАЦІЙНА РОБОТА

Реалізація ігрової логіки
карткової гри з
використанням NFC-карток

Виконав: Лукашенко Максим Петрович
(прізвище, ім'я, по батькові)

(підпис)

Група КН-41

Керівник: асистент Белозорова О. Д.
(науковий ступінь, вчене звання, посада, прізвище, ініціали)

(підпис)

ЗМІСТ

Завдання на кваліфікаційну роботу	
Реферат	3
Вступ.....	4
Розділ 1. Програмні інструменти для розробки ігор	6
1.1. Дослідження існуючих можливостей створення ігор	6
1.1.1. Ігровий рушій Unity	6
1.1.2. Ігровий рушій Unreal Engine	7
1.2. Аналіз існуючих рішень карткових ігор	8
1.2.1. Фізичні карткові ігри	8
1.2.2. Віртуальні карткові ігри	11
1.2.3. Карткові ігри змішаного типу	14
Розділ 2. Розробка гри.....	16
2.1. Опис поставленої задачі	16
2.2. Алгоритм реалізації проекту	17
2.2.1. Arduino для зчитування NFC та передачі через USB та Bluetooth	17
2.2.2. Unity для обробки та подання даних	19
2.3. Алгоритми розв'язання поставлених задач	20
2.3.1. Задача зчитування картки	20
2.3.2. Задача опрацювання даних картки	23
2.3.3. Реалізація формули для обрахування	25
2.3.4. Задача віртуалізації картки	26
Розділ 3. Детальний опис структури програми та реалізованих функцій	27
3.1. Розробка внутрішньої структури гри	27
3.1.1. Скрипт InfoArray	27
3.1.2. Скрипт CardCreator	28
3.1.3. Скрипт CardStateManager	28
3.1.4. Скрипт CardFight	28
3.2. Процес гри	29
Висновок	34
Список використаних джерел	36

РЕФЕРАТ

Кваліфікаційна робота: 36 с., 18 малюнків, 7 джерел.

Метою кваліфікаційної роботи є розробка ігрової логіки карткової гри з використанням NFC-карток, що поєднує традиційні колекційні картки з сучасними технологіями для підвищення інтерактивності та залученості гравців.

Об'єкт дослідження – процес інтеграції фізичних карток у віртуальну гру з використанням NFC-технології.

Предмет дослідження – методи і засоби взаємодії фізичних карток з віртуальною грою за допомогою технологій NFC, Arduino та Unity.

Методи дослідження – використання платформи Arduino для зчитування NFC-карток та ігрового рушія Unity для обробки та візуалізації даних, створення цільових скриптів для забезпечення функціонування гри.

Основні досягнення роботи включають розробку алгоритмів для інтеграції фізичних карток у віртуальну гру, впровадження поняття "параметрів заміщення" для моделювання ігрових механік та збалансування характеристик карток.

Практична цінність полягає у створенні гри, яка дозволяє користувачам взаємодіяти з фізичними картками через NFC, що підвищує інтерактивність та робить гру доступною на різних платформах. Економічна та соціальна цінність роботи полягає у можливості створення високоякісних продуктів з обмеженими ресурсами, що може привабити широку аудиторію.

Прогнозується, що в майбутньому технології, такі як доповнена реальність (AR) та штучний інтелект (AI), дозволять ще глибше інтегрувати фізичні та віртуальні елементи у карткові ігри, створюючи унікальні ігрові досвіди та сприяючи розвитку ігрової індустрії.

Ключові слова: NFC-КАРТКИ, UNITY, ARDUINO, ІГРОВА ЛОГІКА, КАРТКОВІ ІГРИ, ВІЗУАЛІЗАЦІЯ, ІНТЕРАКТИВНІСТЬ, ПАРАМЕТРИ ЗАМІЩЕННЯ

ВСТУП

Сфера розваг завжди відігравала важливу роль у житті людей, забезпечуючи відпочинок і винагороду за працю. Настільні ігри, такі як індійські шахи, перські нарди та єгипетський сенет, залишаються популярними й досі. До недавнього часу велику популярність мали карткові ігри з різноманітними тематиками, де гравці могли обирати картки з кольоровими цифрами, машинами світових брендів або улюбленими персонажами мультфільмів.

З початком ери цифрової трансформації з'явилися динамічні та реалістичні комп'ютерні ігри. Сучасні ігри створюються за допомогою ігрових рушіїв, деякі з яких спеціально розробляються для конкретних проектів, а інші є універсальними інструментами для створення ігор різних жанрів. Прикладами таких універсальних рушіїв є Unreal Engine та Unity.

Unity, розроблений у 2005 році компанією Unity Technologies, відомий своєю універсальністю та доступністю. Цей рушій підтримує розробку для різних платформ, включаючи ПК, мобільні пристрої, веб-браузери, а також віртуальну та доповнену реальність. Unity має інтуїтивно зрозумілий інтерфейс і високу продуктивність на мобільних пристроях, що робить його популярним серед інди-розробників та новачків.

Unreal Engine, створений компанією Epic Games і вперше представлений у 1998 році, славиться здатністю створювати високоякісну фотореалістичну графіку. Використання технологій, таких як Lumen для глобального освітлення та Nanite для віртуалізації геометрії, дозволяє Unreal Engine створювати деталізовані ігрові світи, що робить його вибором для великих AAA-проектів та кіноіндустрії. Unreal Engine має складніший інтерфейс і високу криву навчання, але його система Blueprints дозволяє створювати ігрову логіку без написання коду, що може бути корисним для дизайнерів.

Графічні можливості Unreal Engine перевершують Unity, особливо у випадку фотореалістичної графіки. Unity краще підходить для менш вимогливих до ресурсів проєктів, таких як мобільні ігри та 2D-ігри. Щодо продуктивності, Unreal Engine може вимагати більше ресурсів, але забезпечує високу якість зображення і підходить для великих проєктів. Unity оптимізований для роботи на широкому спектрі пристроїв і часто використовується для створення мобільних додатків та ігор.

Вартість використання також відрізняється: Unity пропонує безкоштовну версію для індивідуальних розробників та невеликих команд, тоді як платні версії пропонують розширені можливості без роялті. Unreal Engine можна використовувати безкоштовно з умовою роялті у розмірі 5% від доходів після досягнення \$1 мільйона. Спільнота розробників обох рушіїв є великою та активною, пропонуючи багато ресурсів для навчання та підтримки.

Моя мета полягає у створенні гри, яка поєднує автентичність класичних карткових ігор з підвищеною інтерактивністю та сучасністю. Використовуючи можливості Unity, я зможу створити яскравий та динамічний інтерфейс, адаптований для різних платформ. Основний пріоритет – забезпечення плавного та приємного ігрового процесу, зручного управління та красивої візуалізації. Гра збереже дух класичних карткових ігор, залучаючи нових гравців і шанувальників традиційних розваг. Особливістю гри буде використання характеристик персонажів, близьких до їхніх реальних аналогів.

РОЗДІЛ 1

ПРОГРАМНІ ІНСТРУМЕНТИ ДЛЯ РОЗРОБКИ ІГОР

1.1. Дослідження існуючих можливостей створення ігор

Ігрові рушії є основними інструментами для розробки відеоігор, надаючи розробникам необхідні засоби для створення інтерактивних світів, управління графікою, фізикою, звуком та іншими аспектами гри. Сучасні рушії, такі як Unity та Unreal Engine, пропонують широкий спектр можливостей і функцій, які дозволяють створювати ігри будь-якого рівня складності — від простих 2D-ігор до масштабних 3D-проектів з високою деталізацією.

Окрім цих відомих рушіїв, існують також інші платформи, такі як Godot, CryEngine та GameMaker Studio, кожна з яких має свої унікальні особливості та переваги. Godot, наприклад, є безкоштовним та відкритим рушієм, що дозволяє розробникам повністю контролювати процес створення гри та вносити будь-які зміни у вихідний код.

У випадках, коли існуючі інструменти не можуть задовольнити специфічні вимоги проекту або коли розробники прагнуть отримати повний контроль над усіма аспектами гри створюється власний рушій. Створення власного ігрового рушія — це завдання, яке вимагає глибоких знань програмування, математики та комп'ютерної графіки. Власний рушій дозволяє реалізувати унікальні функції та оптимізувати продуктивність під конкретні потреби гри.

Підсумовуючи, ігрові рушії є ключовим компонентом у сучасній розробці ігор, надаючи розробникам потужні інструменти для реалізації їхніх творчих ідей. Вибір між використанням існуючого рушія та створенням власного залежить від багатьох факторів, включаючи складність проекту, технічні вимоги та доступні ресурси.

1.1.1. Ігровий рушій Unity

Unity є одним з найпопулярніших ігрових рушіїв завдяки своїй універсальності та зручності у використанні. Він підтримує розробку для різних платформ, включаючи мобільні пристрої, консолі, ПК та веб-браузери. Unity також пропонує велику кількість готових до використання активів через Unity Asset Store, що значно прискорює процес розробки. Інтуїтивно

зрозумілий інтерфейс та доступність роблять Unity ідеальним вибором для інди-розробників та початківців. Unity використовує мову програмування C# для написання скриптів. Раніше Unity також підтримував JavaScript (відомий у середовищі Unity як UnityScript), але з 2017 року підтримка цієї мови була припинена, і C# став основною мовою для створення ігрової логіки та поведінки в Unity. C# є потужною та зручною мовою, яка забезпечує високу продуктивність і гнучкість, що робить її ідеальним вибором для розробки як простих, так і складних ігрових проектів.

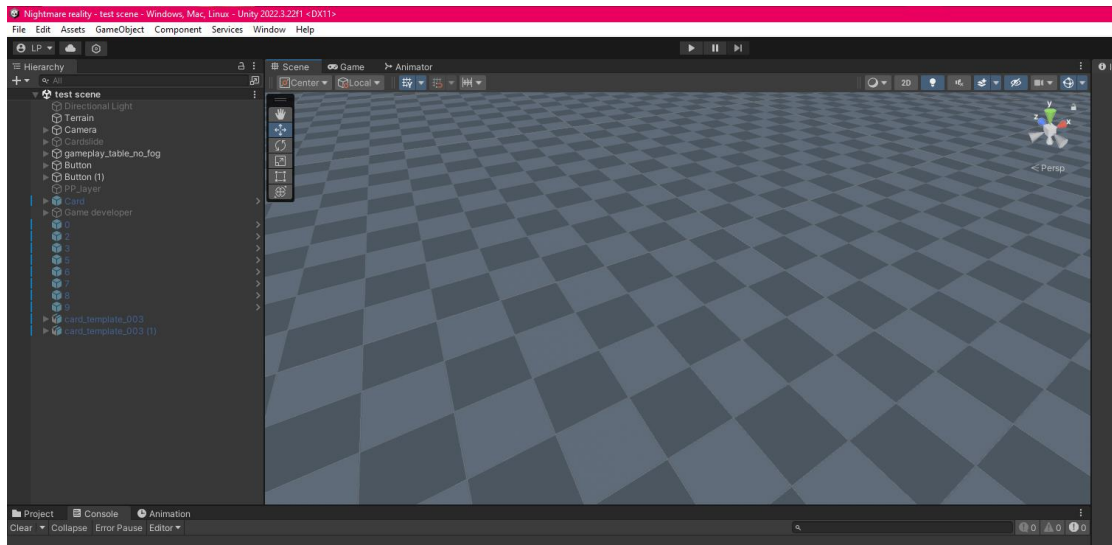


Рис 1. Інтерфейс Unity

1.1.2. Ігровий рушій Unreal Engine

Unreal Engine, розроблений компанією Epic Games, відомий своїми передовими графічними можливостями. Він використовується для створення високоякісних AAA-ігор, а також у кіноіндустрії для створення спецефектів та віртуальних зйомок. Unreal Engine надає розробникам потужні інструменти для роботи з графікою, анімацією та фізикою, дозволяючи створювати реалістичні та детальні ігрові світи. Хоча цей рушій має вищу криву навчання, його система Blueprints дозволяє створювати складну ігрову логіку без написання коду. Unreal Engine використовує мову програмування C++ для розробки ігрової логіки та систем низького рівня. C++ надає високу продуктивність і гнучкість.



Рис 2. Вигляд редактора рушія Unreal Engine

1.2. Аналіз існуючих рішень карткових ігор

1.2.1. Фізичні карткові ігри

Однією з найвідоміших карткових ігор є «Бридж». Це командна гра для чотирьох гравців, які грають в парах. Гравці використовують стандартну колоду з 52 карт, а метою гри є отримання максимальної кількості очок шляхом виконання торгів. Гра складається з фаз торгів і гри картами, де торги визначають контракт, який повинна виконати пара.

Uno — сімейна гра з використанням унікальної колоди карт, є дуже популярною завдяки своїй простоті та захоплюючому геймплею. Метою гри є позбутися всіх своїх карт, викладаючи їх на стіл за кольором або номером. Гравці по черзі викладають карти з рук на центральну стопку, але лише ті, що відповідають кольору або номеру верхньої карти на стопці.

Основна колода Uno складається з карт чотирьох кольорів: червоного, жовтого, зеленого та синього. Кожен колір включає карти з номерами від 0 до 9, а також різноманітні спеціальні карти, які додають до гри елемент стратегії і непередбачуваності. До спеціальних карт належать "Пропуск ходу", "Зміна напрямку", "Візьми дві", "Дикі карти" та "Дикі карти з вимогою взяти чотири".

Карта "Пропуск ходу" змушує наступного гравця пропустити свій хід, що може змінити хід гри і надати стратегічну перевагу тому, хто зіграв цю

карту. "Зміна напрямку" змінює напрямок гри, що особливо корисно в ситуаціях, коли гравець хоче зупинити опонента, який наближається до перемоги. Карта "Візьми дві" змушує наступного гравця взяти дві додаткові карти з колоди і пропустити свій хід.

"Дикі карти" дозволяють гравцю змінити колір гри на будь-який інший, що особливо корисно, коли у гравця немає відповідних карт на руках. "Дикі карти з вимогою взяти чотири" не лише змінюють колір гри, але й змушують наступного гравця взяти чотири додаткові карти і пропустити свій хід. Ці спеціальні карти додають до гри тактичної глибини і роблять її більш динамічною.

Гра триває до тих пір, поки один з гравців не позбудеться всіх своїх карт. У цей момент він вигукує "Uno!", сигналізуючи іншим гравцям про те, що у нього залишилася лише одна карта. Якщо інші гравці виявляють, що він не вигукнув "Uno!" перед тим, як викласти свою передостанню карту, вони можуть змусити його взяти додаткові карти як штраф. Гра закінчується, коли один з гравців повністю позбувається всіх карт, і підраховуються очки. Карти, що залишилися у руках інших гравців, приносять їм штрафні очки, які враховуються при підрахунку загального результату.



Рис 3. Картки гри Uno

Yu-Gi-Oh! є популярною колекційною картковою грою на основі відомого аніме та манги. Метою гри є зменшення очок життя суперника до нуля, використовуючи карти монстрів, заклинань і пасток. Гравці використовують персоналізовані колоди для виклику монстрів і розігрування заклинань.

Pokemon TCG – це відома колекційна карткова гра, заснована на популярній франшизі "Покемон". Створена у 1996 році, вона швидко здобула популярність серед фанатів покемонів і стала однією з найуспішніших

колекційних карткових ігор у світі. Метою гри є перемогти всіх покемонів суперника або зібрати шість призових карт, використовуючи різноманітні стратегії та тактики.

Гра починається з того, що обидва гравці витягують по сім карт і розміщують одного базового покемона на своєму полі як активного покемона, а решту базових покемонів можна розмістити на лаві запасних покемонів. Гравці по черзі розігрують карти, додають енергію до своїх покемонів, використовують карти тренерів і еволюціонують своїх покемонів. Атака відбувається шляхом витрачання енергії, прикріпленої до атакуючого покемона, і зменшення НР (рівня здоров'я) покемона суперника. Якщо НР покемона суперника знижується до нуля, він вважається нокаутованим, і гравець, який здійснив атаку, витягує одну призову карту.

Основною метою гри є зібрати шість призових карт або нокаутувати всіх активних покемонів суперника. Карти покемонів включають базових покемонів, а також їхні еволюційні форми – першої та другої стадії, які володіють більш потужними атаками та здібностями. Карти енергії використовуються для активації атак покемонів, а карти тренерів – для надання певних переваг, таких як витягування додаткових карт, лікування покемонів або пошук необхідних карт у колоді.

Стратегія гри полягає у створенні збалансованої колоди, плануванні атак, використанні карт тренерів у потрібний момент і еволюції покемонів для досягнення максимальної ефективності в битві. Важлива також взаємодія з суперником, аналіз його стратегії і коригування власних дій для досягнення переваги в грі.

Exploding Kittens є грою на виживання з унікальною ілюстрацією і гумором. Метою гри є уникнути вибухових кошенят за допомогою стратегій і спеціальних карт. Гравці по черзі витягують карти з колоди, намагаючись не витягнути вибухового кошеня, що викликає програш. Спеціальні карти допомагають уникнути вибухів або змінюють хід гри.

Ці карткові ігри є відомими і популярними завдяки своїй різноманітності, захоплюючим механікам та можливостям для стратегічного мислення. Вони приваблюють гравців різного віку і рівня підготовки, забезпечуючи цікаве та інтерактивне проведення часу.



Рис 4. Картки гри «Вибухові кошенята»

1.2.2. Віртуальні карткові ігри

Віртуальні карткові ігри представляють собою як адаптацію аналогової класики, так і нові твори, що народилися в цифровій сфері. Вони сумісні з різним апаратним забезпеченням, включаючи, але не обмежуючись ПК, консолями, смартфонами та планшетами. У ігри можна грати як поодиночки, так і з іншими — друзями чи незнайомцями — дозволяючи змагатися наживо чи по черзі.

Різноманітність жанрів і стилів робить віртуальні карткові ігри популярними серед мільйонів гравців у всьому світі. Нижче наведено кілька прикладів:

Magic: The Gathering Arena — віртуальна адаптація знаменитої карткової гри Magic: The Gathering, яка є основоположником жанру колекційних карткових ігор. Гра пропонує гравцям захоплюючий ігровий досвід завдяки візуально привабливій графіці та анімації. Одна з головних особливостей Magic: The Gathering Arena — це можливість створювати власні колоди з безлічі карт, які гравці можуть колекціонувати та обмінюватися. Часті оновлення гри представляють нові карти та набори, що додають різноманітність та збагачують ігровий процес.

Magic: The Gathering Arena зберігає основні принципи фізичної гри, включаючи складні механіки і стратегії. Кожна гра починається з певної кількості карт на руках гравців. Гравці по черзі розігрують карти, сплачуючи за них ману. Істоти можуть атакувати опонента або захищатися від атак, заклинання створюють різні ефекти, а артефакти надають постійні або тимчасові бонуси. Гра розбита на кілька фаз, серед яких початкова фаза, основна фаза, бойова фаза, друга основна фаза та кінцева фаза. Під час бойової фази гравці можуть оголошувати атаки своїми істотами, на які опонент може реагувати, захищаючись своїми істотами. Якщо атака пройшла успішно, опонент втрачає очки життя. Однією з ключових особливостей Magic: The Gathering Arena є її глибокий та стратегічний ігровий процес. Крім того, онлайн-формат дозволяє гравцям брати участь у змаганнях та турнірах, що робить гру ще більш динамічною та конкурентною. Magic: The Gathering Arena є чудовим прикладом того, як класична карткова гра може бути успішно перенесена у цифровий формат, зберігаючи свою глибину та складність.

Hearthstone — гра створена Blizzard Entertainment, є однією з найпопулярніших віртуальних карткових ігор, яка відрізняється своїм простим і доступним геймплеєм. Гра розгортається у всесвіті Warcraft, що надає їй впізнаваний і привабливий контекст. У Hearthstone гравці створюють колоди з карт, які представляють різних героїв, заклинання та істот. Кожна колода належить до певного класу, що визначає доступні стратегії та стилі гри.

Гра Hearthstone відрізняється своєю простотою в освоєнні та швидкими матчами, що робить її привабливою для широкої аудиторії. Гравці починають з певної кількості карт на руках і по черзі розігрують карти. У грі доступні різні типи карт, включаючи істот, заклинання та зброю, які можуть використовуватися для атаки опонента або захисту. Кожна карта має певну вартість у мані та унікальні властивості. Гра пропонує різноманітні режими, включаючи соло-кампанії, дуелі з іншими гравцями та арени, де гравці змагаються за найкращі результати. Hearthstone успішно поєднує доступність з глибиною стратегії, що робить її однією з найпопулярніших карткових ігор у світі.

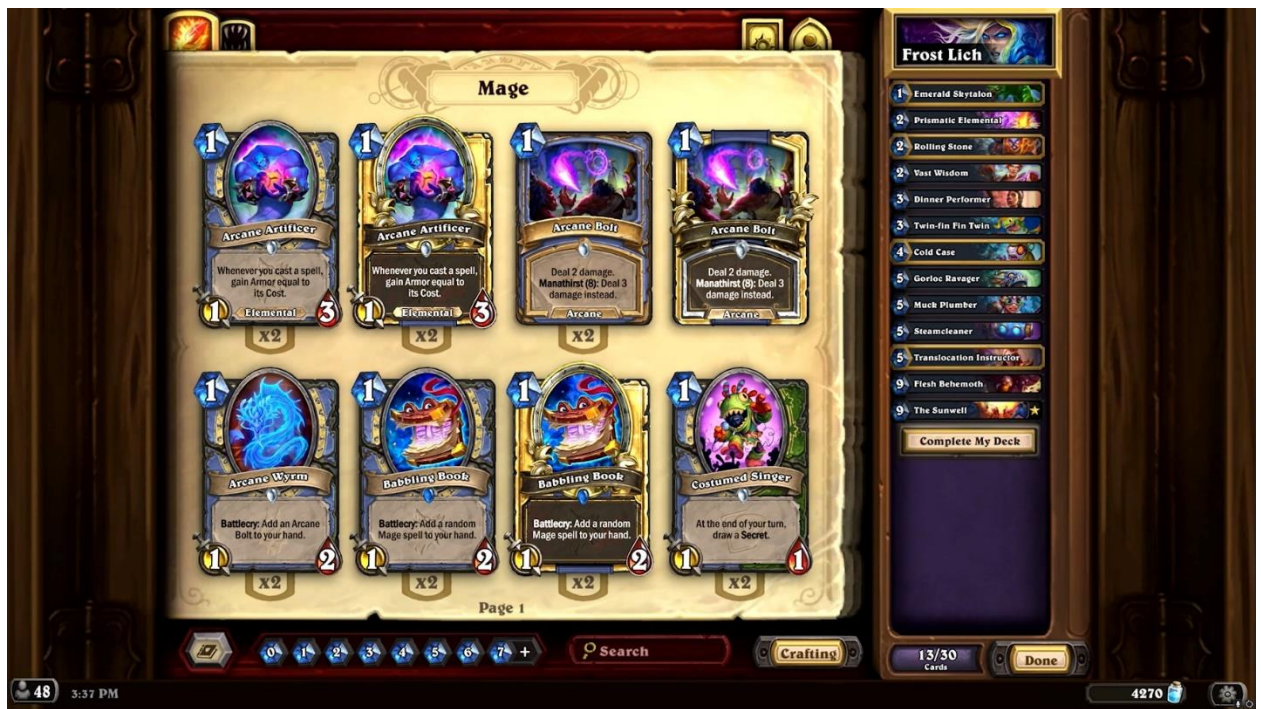


Рис 5. Онлайн гра Hearthstone

Gwent: The Witcher Card Game – гра розроблена CD Projekt Red, базується на всесвіті "Відьмака" і пропонує унікальний підхід до карткових баталій. Гра відрізняється своєю механікою, яка фокусується на стратегічному використанні карт у трьох раундах. Гравці змагаються, намагаючись набрати більше очок, розігруючи карти на різних лініях бою.

Gwent відрізняється своєю глибиною тактичних можливостей. Кожна карта може мати різні ефекти в залежності від її позиції на полі бою. Гра наголошує на плануванні та передбаченні дій супротивника, що робить кожен матч захоплюючим і непередбачуваним. Гравці можуть використовувати різні тактики для перемоги в кожному з трьох раундів, що додає глибини ігровому процесу. Однією з ключових особливостей Gwent є її унікальна механіка побудови колоди та використання карт.

Yu-Gi-Oh! Duel Links – це віртуальна версія відомої колекційної карткової гри Yu-Gi-Oh!, розроблена Konami. Гра пропонує гравцям змагатися в дуелях, використовуючи карти монстрів, заклинань і пасток. Duel Links включає в себе елементи сюжетної лінії з аніме-серіалу, а також регулярні оновлення та події, що додають нові карти та механіки.

Faeria – це унікальна карткова гра, яка поєднує в собі елементи стратегічних настільних ігор і класичних колекційних карткових ігор. Гравці створюють власні колоди і змагаються один з одним на динамічному ігровому

полі, яке формується в ході гри. Faeria відома своїм інноваційним підходом до карткових ігор та гарною графікою.

1.2.3. Карткові ігри змішаного типу

Окремий вид ігор є змішані карткові ігри, що поєднують в собі елементи як фізичні, так і віртуальні. Вони можуть включати в себе віртуалізацію фізичних карток або використовувати технології доповненої реальності (AR).

Віртуалізація фізичних карток

Цей підхід полягає у переведенні куплених фізичних карток у цифровий формат, подібно до того, як колись могли активуватися ігри за допомогою ключів. Гравець надає код, який додає цифрову версію картки до його особистої колекції.

Переваги віртуалізації:

1. **Безпека від підробки:** Цифрові картки важко підробити, оскільки необхідна інформація знаходиться на сервері, що захищає гравців від обману.
2. **Матеріальність:** Фізична картка додає відчуття вагомості покупки, що підвищує цінність колекціонування.

Недоліки віртуалізації:

3. **Втрата цінності після використання:** Після активації фізичні картки втрачають свою цінність і стають непотрібними у фізичному плані.

Ігри доповненої реальності (AR)

Ігри доповненої реальності використовують фізичні картки, які оживають на екрані пристрою за допомогою камери. Прикладом такої гри є Arena TCG, де гравці бачать свої картки у їх 3d версії.



Рис 6. Arena TCG

Переваги ігор доповненої реальності:

1. **Фокус на фізичну картку:** Гравець продовжує використовувати і зберігати фізичні картки, що зберігає їх цінність.
2. **Інтерактивний досвід:** Доповнена реальність додає новий рівень взаємодії з картками, роблячи гру ще більш цікавою для кінцевого користувача.

Недоліки ігор доповненої реальності:

1. **Високі вимоги до пристроїв:** Для створення якісних AR-ефектів потрібні потужні пристрої, що можуть бути недоступні всім гравцям.
2. **Складність реалізації:** Створення реалістичних і захоплюючих AR-ефектів є технічно складним і вимагає значних ресурсів.
3. **Захист від підробки:** Необхідно забезпечити захист фізичних карток від підробки, щоб уникнути шахрайства.

Змішані карткові ігри об'єднують найкращі аспекти фізичних і віртуальних карткових ігор, але також стикаються з унікальними проблемами, пов'язаними з їх реалізацією і подальшою підтримкою.

Ціллю мого проекту є спроба об'єднати віртуальну та фізичну складові в одне ціле, щоб забезпечити якомога комфортніший геймплей.

РОЗДІЛ 2.

РОЗРОБКА ГРИ

2.1. Опис поставленої задачі

Ціллю роботи було створити гру, що брала б за основу колекційні картки дитинства («Супергонки», «людина павук» та інші) та виправила б їх недоліки, такі як важкість гри по кільком характеристикам та відсутність інтерактивності.



Рис 7. Картки «Супергонки»

За діючих персонажів вирішено було взяти професії, а у якості характеристик – життєві показники цієї професії. Ціллю показників було якомога точніше відтворити стереотипного представника свого роду занять, подібно до того, як машину характеризують її характеристиками двигуна та ваги тощо.

Особливістю цієї гри стане не тільки інтерактивність на відміну від звичайних карток, а ще й обрахунок «параметрів заміщення». Це означає, що кожен параметр може з певною ефективністю компенсувати недочаху іншого, проте для реалізації спочатку необхідно зробити представлення цієї карти у своїй програмі.

2.2. Алгоритм реалізації проекту

Архітектура гри включає в себе два основні елементи, які будуть забезпечувати її функціонал та здатність сприймати картки.

Першою складовою є використання платформи Arduino та їх плати Arduino Uno. Вона забезпечує правильне зчитування NFC картки та вивід її ідентифікатора за допомогою USB для подальшого опрацювання програмою

Другим елементом є ігровий рушій Unity, скрипти в якому будуть приймати відповідні значення та опрацьовувати їх для подальших обрахунків.

2.2.1. Arduino для зчитування NFC та передачі через USB та Bluetooth

Arduino Uno є ключовим елементом у зчитуванні NFC-карток для моїх цілей. Використання платформи Arduino дозволяє легко інтегрувати функціонал зчитування карток завдяки своїй доступності та простоті у використанні.

Підключення NFC-модуля: Для зчитування NFC-карток до плати Arduino Uno має бути підключено NFC-модуль, наприклад, PN532 або MFRC522. Ці модулі призначені для роботи з RFID та NFC-картками і здатні зчитувати унікальні ідентифікатори (UID) карток, які будуть використовуватись як ідентифікатор у грі. Для проекту я вибрав RFID-RC522 від keyestudio. Його основною перевагою є відповідність вимогам та відносно велика дальність зчитування. Тег NFC включає три основні компоненти: чіп, антену та щось, що утримує це все разом. [1]



Рис 8. NFC модуль RC-522



Рис 9. NFC картка Mifare-One 13.56МГц

Процес зчитування: . Коли NFC-картка прикладається до модуля, RC-522 зчитує її UID. Плата Arduino Uno відповідає за обробку цього сигналу і передачу зчитаних даних до центрального пристрою [2]. Цей процес включає контроль NFC-модуля, виявлення картки в зоні дії модуля, перевірку даних на повторення та формат, зчитування UID картки і передачу цього UID через послідовний порт (USB).

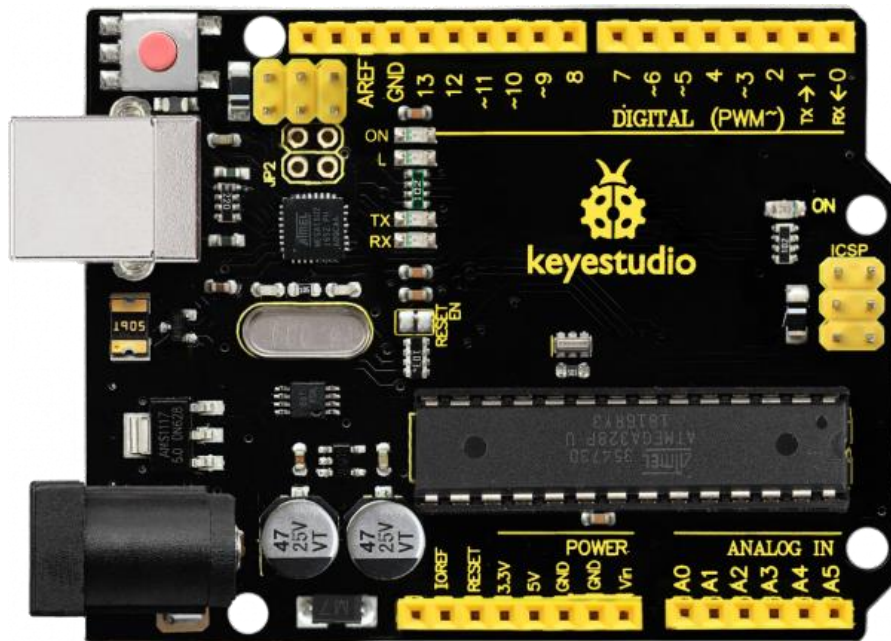


Рис 10. Плата Arduino Uno від keyestudio

Передача даних через USB: Після зчитування UID картки, Arduino Uno передає ці дані на пристрій для подальшого опрацювання скриптом. Це здійснюється за допомогою USB-з'єднання, яке забезпечує швидку та надійну передачу даних. Ця взаємодія дозволяє скриптам в грі використовувати

зчитані ідентифікатори для ініціалізації картки, відображення інформації про картку або виконання певних ігрових механік.

Підключення Bluetooth-модуля

Для альтернативної відправки даних на телефон або бездротовим способом на комп'ютер використовується модуль HM-10. Цей модуль дозволяє передавати дані разом з кабелем через Bluetooth, забезпечуючи додаткові можливості у передачі інформації.



Рис 11. Bluetooth модуль HM-10

Програмування та налаштування: Для забезпечення правильного функціонування системи необхідно написати відповідний код для плати Arduino Uno. Цей код включатиме інструкції для ініціалізації NFC-модуля, зчитування UID карток і передачі цих даних на пристрій. Завдяки великій кількості бібліотек, доступних для платформи Arduino, можна реалізувати зчитування та простий захист від повтору в карток.

Таким чином, використання платформи Arduino та плати Arduino Uno забезпечує надійне та ефективне зчитування NFC-карток, що є важливою складовою частиною ігрової архітектури.

2.2.2. Unity для обробки та подання даних

Unity є другим ключовим елементом у нашій архітектурі гри, забезпечуючи обробку та візуалізацію всіх даних, отриманих від плати Arduino. Використання Unity дозволяє створювати інтерактивні та візуально привабливі ігрові елементи, які відобразять зчитані дані NFC-карток.

Обробка даних: Unity отримує ідентифікатори (UID) карток через USB-з'єднання з Arduino. Далі, Unity використовує ці дані для обробки та візуалізації у грі. Наприклад, на основі UID можна визначити, яку картку було

зчитано, і викликати відповідні ігрові події або анімації. Це може включати відображення інформації про картку або зміну стану гри.

Інтеграція з Arduino: Для інтеграції з Arduino в Unity використовуються спеціальні бібліотеки, які дозволяють зчитувати дані з серійного порту.

Подання даних: Unity пропонує широкі можливості для візуалізації та взаємодії з даними. Отримані від Arduino UID карток можуть бути використані для динамічної зміни інтерфейсу користувача, відображення графічних елементів або запуску анімацій.

Розширюваність: Використання Unity дозволяє легко розширювати функціонал гри. Наприклад, можна додати нові типи карток або інтерактивні елементи без значних змін в архітектурі гри. Unity також підтримує різні платформи, що дозволяє розробнику розширити аудиторію гри на різні пристрої, включаючи мобільні телефони, планшети та ПК.

Таким чином, Unity виступає потужним інструментом для обробки та подання даних, що надає змогу створити захоплюючу гру на основі зчитаних NFC-карток. Завдяки своїм можливостям, Unity забезпечує гнучкість, необхідну для реалізації будь-яких ігрових ідей.

2.3. Алгоритми розв'язання поставлених задач

2.3.1. Задача зчитування картки

Проблематика даної задачі полягала у виборі правильних компонентів для реалізації проекту та врахуванні його специфічних особливостей. Основними вимогами до компонентів були достатня дальність зчитування NFC-карток, зручність підключення, а також мінімізація хибних спрацювань.

Перш за все, для забезпечення коректної роботи системи було важливо обрати NFC-модуль, здатний зчитувати картки на достатній відстані, що дозволяє уникнути проблем зі зчитуванням даних під час використання, оскільки модуль використовуватиметься за декоративною перешкодою. Це критично важливо для забезпечення безперебійного та зручного користування системою, особливо у випадках, коли картки мають читатися під час динамічного руху.

Крім того, зручне підключення компонентів було важливим аспектом проекту. Підключення NFC-модуля до плати Arduino Uno повинно бути

простим і надійним, щоб забезпечити стабільну роботу всієї системи. Вибір компонентів також враховував можливість комунікації з іншими частинами проекту, такими як скрипти об'єктів.

Уникнення хибних спрацювань було ще однією важливою вимогою до цієї частини проекту. Система повинна виключати повторне зчитування тієї самої картки та запобігати хибним сигналам, що можуть призвести до неправильної обробки даних.

Таким чином, вибір правильних компонентів для реалізації проекту було задачею, що вимагала ретельного аналізу та врахування специфічних технічних вимог. Важливо було забезпечити, щоб обрані компоненти не тільки відповідали основним вимогам проекту, але й працювали разом як єдина система, забезпечуючи високу надійність та ефективність роботи.



Рис 12. Тестування дальності зчитування картки

Після тестування датчик показав результати:

Тест на відкритому просторі – до 5,5 см.

Тест з перешкодою (дерев`яна дошка товщиною два сантиметри) – до чотирьох сантиметрів.

Тест з перешкодою у вигляді ще однієї карти – до двох сантиметрів, спрацювання на другу карту, поки перша не змінить положення у просторі.

Тест з перешкодою двох карт – до 0,3 см, спрацювання в окремих випадках, карти лежать нерухомо.

2.3.2. Задача опрацювання даних картки

Для опрацювання карток у грі було створено три різні сценарії, кожен з яких має свої переваги та недоліки. Кожен сценарій призначений для конкретних умов використання та враховує різні аспекти безпеки, зручності та масштабованості.

Перший сценарій передбачає використання бази даних MySQL для збереження інформації про картки. Цей підхід включає розміщення даних про картки в централізованій базі даних, до якої надсилаються запити у вигляді PHP скриптів.

Переваги цього сценарію:

- **Централізоване зберігання даних:** База даних дозволяє зберігати всі дані в одному місці, що полегшує управління інформацією та її оновлення.
- **Безпека та контроль доступу:** MySQL забезпечує різні рівні безпеки та контроль доступу, що ускладнює несанкціонований доступ до даних.
- **Масштабованість:** Така система може бути легко масштабована для підтримки великої кількості користувачів і карток.
- **Легкість інтеграції:** Використання PHP скриптів дозволяє легко інтегрувати базу даних з іншими веб-додатками та сервісами.

Недоліки цього сценарію:

- **Залежність від серверної інфраструктури:** Для роботи необхідний сервер, що може збільшити витрати та складність налаштування системи.
- **Мережеві затримки:** Доступ до бази даних може бути обмежений швидкістю мережі, що може вплинути на продуктивність гри в режимі реального часу.

Другий сценарій передбачає збереження даних про картку безпосередньо на самій NFC картці в користувацькій області пам'яті. Це дозволяє зберігати всю необхідну інформацію на фізичній картці, що робить процес зчитування та оновлення даних дуже простим.

Переваги цього сценарію:

- **Простота налаштування та використання:** Немає потреби в налаштуванні серверної інфраструктури або складних систем збереження даних.
- **Можливість легко оновлювати дані:** Дані на картці можна легко оновлювати або доповнювати без необхідності доступу до централізованої бази даних.
- **Автономність:** Картки можуть використовуватись автономно, без постійного підключення до інтернету чи бази даних.

Недоліки цього сценарію:

- **Безпека:** Дані на NFC картках можуть бути змінені або підроблені іншими користувачами, що може створити проблеми з безпекою.
- **Обмежений обсяг пам'яті:** Користувацька область пам'яті на NFC картках зазвичай обмежена, що може стати проблемою для зберігання великої кількості інформації.

Третій сценарій передбачає створення внутрішнього списку, який містить інформацію про картки без використання сторонніх засобів обробки даних. Такий підхід є особливо зручним для локальних ігор з друзями, де не потрібна складна інфраструктура.

Переваги цього сценарію:

- **Зручність для локального використання:** Внутрішній список може бути легко налаштований та використовуваний для невеликих ігор з обмеженою кількістю учасників.
- **Незалежність від зовнішніх систем:** Не потрібна наявність бази даних або серверів, що знижує витрати та спрощує налаштування.
- **Швидкодія:** Внутрішні списки працюють дуже швидко, оскільки не потребують звернення до зовнішніх джерел даних.

Недоліки цього сценарію:

- **Обмежена масштабованість:** Цей підхід не підходить для великих проектів або онлайн-ігор з багатьма користувачами.

- **Обмежена безпека:** Внутрішні списки не забезпечують високого рівня безпеки та можуть бути легко змінені.

Таким чином, кожен з трьох сценаріїв обробки даних має свої переваги та недоліки.

2.3.3. Реалізація формули для обрахування

Задля збільшення реалістичності гри окрім обрахування відповідних параметрів карток, можуть обчислюватись інші параметри у ролі альтернативних (з власним коефіцієнтом, який відображає наскільки ефективно той чи інший параметр може замінити собою). Формула виглядає наступним чином:

$$param1 = param1 - ((param2 * kf) * (1 - socialitydiff * 0.005))$$

де $param1$ – це характеристика картки атаки, $param2$ – компенсуюча характеристика картки захисту, kf – коефіцієнт із таблиці (стовпець $param2$, рядок $param1$), $socialitydiff$ – різниця показників соціальності поточної та попередньої карток захисту, враховується у випадку, коли ця різниця перевищує 10.

	Strength	Perseverance	Intelligence	Salary
Strength	1	1,05	1,08	0,95
Perseverance	0,95	1	0,9	0,95
Intelligence	0,7	0,9	1	0,85
Salary	0,85	0,9	0,95	1

Рис 13. Таблиця коефіцієнтів

Загалом картка має чотири активних параметри та 1, що не бере участі у безпосередньому поєдинку.

2.3.4. Задача віртуалізації картки

Відеоігри не були б дуже відео без графічних компонентів [3]. Створення графічного представлення на сцені перед користувачем є надзвичайно важливою задачею. Візуалізація графічного зображення на екрані перед користувачем є надзвичайно важливим завданням, яке має вирішальне значення для забезпечення якісного користувацького досвіду.

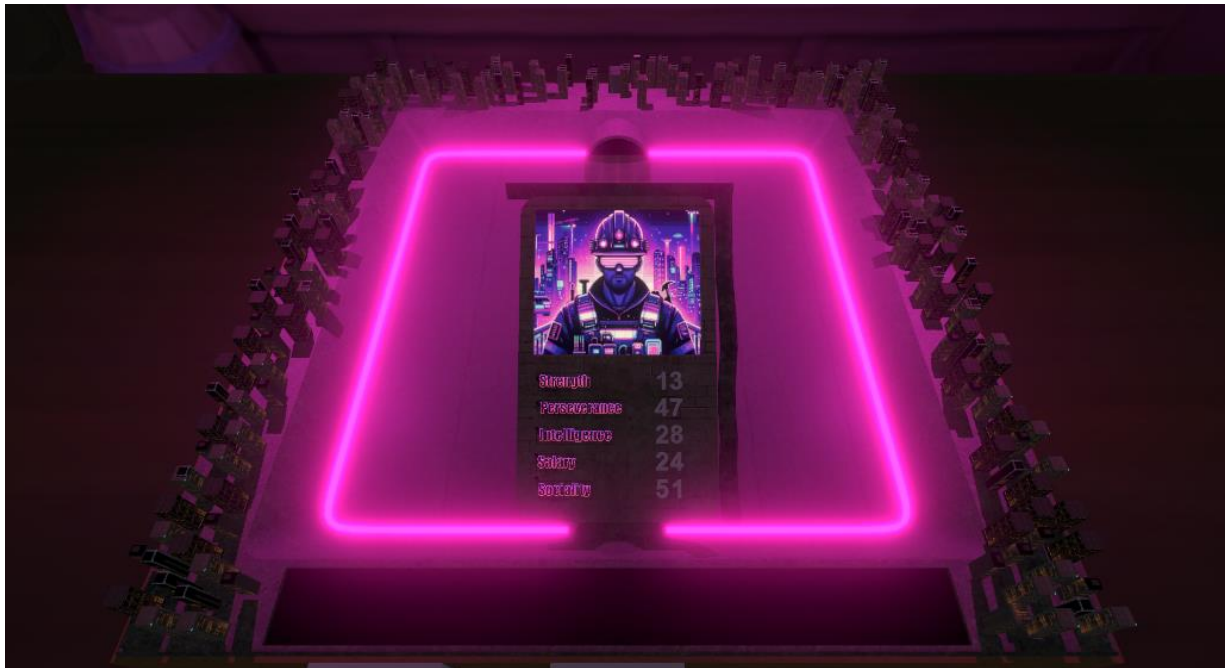


Рис 14. Приклад новоствореної картки

Після опрацювання інформації про NFC картку виникає необхідність візуалізувати її на сцені, а також зберегти її представлення всередині коду. Для зберігання в коді використовується створений клас “Card”. Для візуалізації використовується префаб карти, та набір функцій, які отримуючи необхідні параметри генерують цифрові значення на карті.

РОЗДІЛ 3

ДЕТАЛЬНИЙ ОПИС СТРУКТУРИ ПРОГРАМИ ТА РЕАЛІЗОВАНИХ ФУНКЦІЙ

3.1. Розробка внутрішньої структури гри

Для кращої структуризації коду було вирішено поділити скрипти на окремі одиниці. Після розбиття я отримав кілька цільових скриптів, таких як “CardCreator”, “CardFight”, “CardStateManager” та “InfoArray”. Розробка цих цільових скриптів забезпечує чітке розділення функцій у програмному забезпеченні, що значно спрощує процеси розробки, тестування та підтримки.

Також було створено клас “Card”. У C# все відбувається всередині класу. Жоден метод не може виконуватися поза класом, навіть Main(). Метод Main() є точкою входу для вашої програми; він викликається операційною системою і є місцем, де починається виконання вашої програми. Зазвичай ви створюєте невеликий клас для розміщення Main(), тому що, як і будь-який метод, Main() повинен знаходитися всередині класу [4].

Такий підхід до організації коду дозволяє підтримувати його гнучкість і масштабованість. Завдяки чітко визначеним завданням кожного скрипту, можна легко вносити зміни або додавати нові функціональні можливості без ризику порушення роботи інших компонентів. Це забезпечує стабільність системи та полегшує її розширення і підтримку у майбутньому.

3.1.1. Скрипт InfoArray

Скрипт InfoArray відповідає за забезпечення правильності зчитування та розпізнавання картки. Його основна функція полягає у пошуку картки за її унікальним ідентифікатором (UID) та запуску подальших дій для створення її образу.

Основні функції скрипту InfoArray:

- **Зчитування UID картки:** Визначає унікальний ідентифікатор картки при її скануванні.
- **Пошук відповідної картки:** Використовує UID для пошуку відповідної інформації про картку у базі даних.

- **Ініціація створення образу:** Після успішного пошуку та перевірки на помилки запускає процес створення візуального образу картки.

3.1.2. Скрипт CardCreator

Скрипт CardCreator призначений для створення візуального образу картки, а також трансформації її цифрових характеристик у наглядні 3D-цифри на площині.

Основні функції скрипту CardCreator:

- **Створення образу картки:** Відповідає за генерацію візуального представлення картки на основі її характеристик.
- **Візуалізація характеристик:** Перетворює цифрові параметри картки (атака, витривалість, інтелект тощо) у візуальні елементи на 3D-моделі картки.
- **Розміщення на сцені:** Відповідає за правильне розміщення створеного образу картки на ігровій сцені.

3.1.3. Скрипт CardStateManager

Скрипт CardStateManager відповідає за управління станами карток. Він записує та відслідковує поточний стан картки, наприклад, чи знаходиться картка в режимі атаки або захисту.

Основні функції скрипту CardStateManager:

- **Запис станів карток:** Відслідковує та зберігає поточний стан кожної картки на полі.
- **Управління типом картки:** Визначає та записує, чи картка використовується для атаки або захисту.
- **Створення об'єктів класу:** Ініціалізує об'єкти класу Card та управляє їх станами протягом гри.

3.1.4. Скрипт CardFight

Скрипт CardFight реалізує механіку бою між картками, змінює цифрові значення характеристик карток та запускає функції для візуалізації цих змін.

Основні функції скрипту CardFight:

- **Реалізація механіки бою:** Визначає логіку та правила бою між картками на основі їх характеристик.
- **Зміна характеристик карток:** Оновлює цифрові значення параметрів карток (атака, захист тощо) відповідно до результатів бою.
- **Візуалізація змін:** Запускає функції, що оновлюють візуальні елементи карток для відображення змін у їх характеристиках.

3.2. Процес гри

Після натиснення кнопки "Play" на екрані відбувається декілька важливих дій, які запускають гру. Ці дії включають підняття камери над ареною та перехід у стан очікування на картки. Детальний опис цього процесу наведено нижче.

Після натиснення кнопки "Play"

1. Анімація камери

- **Початкове положення:** Коли гравець натискає кнопку "Play", камера починає рухатися з поточного положення.
- **Підняття камери:** Камера плавно піднімається над ареною, забезпечуючи гравцю повний огляд ігрового поля. Це створює ефект переходу від меню до ігрового режиму.
- **Завершення руху:** Після досягнення необхідної висоти і кута огляду, камера зупиняється у визначеному положенні, забезпечуючи оптимальний огляд арени.

2. Очікування карток

- **Активування режиму очікування:** Після завершення анімації камери гра переходить до режиму очікування карток.
- **Зчитування NFC-карток:** Гравці можуть підносити свої NFC-картки до датчику. Кожна картка розпізнається і додається до гри в залежності від порядку зчитування.

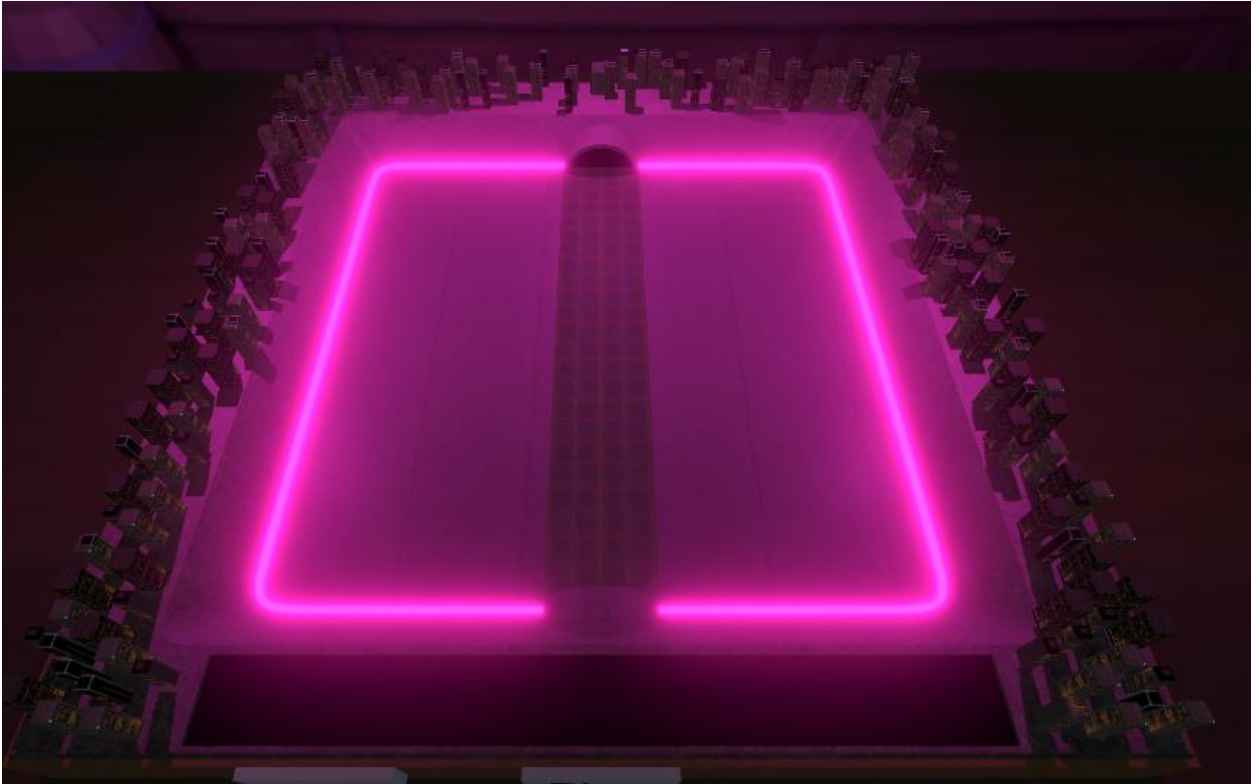


Рис 15. Вид сцени зверху

Після отримання інформації про картку від Arduino, відбувається кілька важливих етапів, що включають візуалізацію картки, демонстрацію її характеристик гравцям та розміщення на арені. Цей процес забезпечує зручність та інформативність для гравців, допомагаючи їм краще зрозуміти властивості кожної картки перед її використанням у грі.

Процес візуалізації та розміщення картки на арені

1. Візуалізація картки

- **Створення візуального образу:** Скрипт "CardCreator" відповідає за створення візуального образу картки. Це включає генерацію 3D-моделі картки з усіма її візуальними елементами.
- **Перетворення характеристик:** Цифрові характеристики картки трансформуються у видимі елементи на моделі картки, такі як 3d цифри.

2. Демонстрація характеристик гравцям

- **Показ характеристик:** Перед розміщенням на арені, картка демонструє свої характеристики гравцям. Це реалізується через спеціальну анімацію, яка підкреслює важливі параметри картки.

3. Розміщення картки на арені

- **Розміщення:** Після демонстрації характеристик, картка розміщується на арені у відповідному місці. Це забезпечує готовність до подальших дій, таких як атака або захист.
- **Ініціалізація стану:** Скрипт "CardStateManager" оновлює стан картки на полі, визначаючи, чи картка буде використовуватися для атаки або захисту.

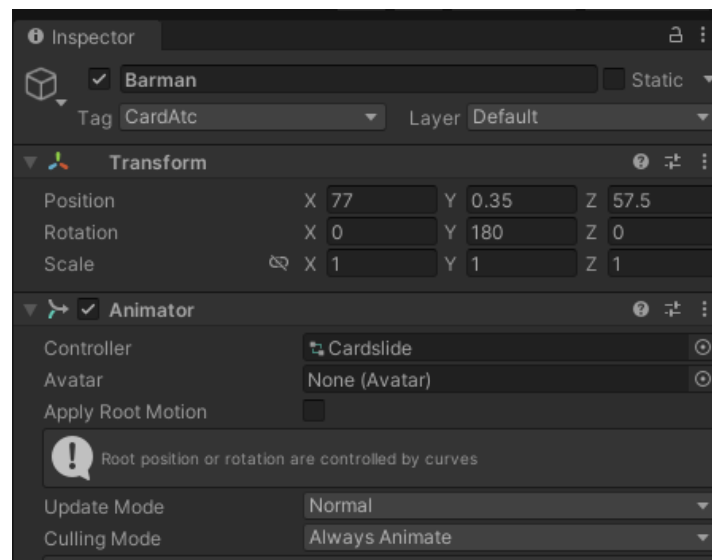


Рис 16. Вигляд новоствореної картки в інспекторі

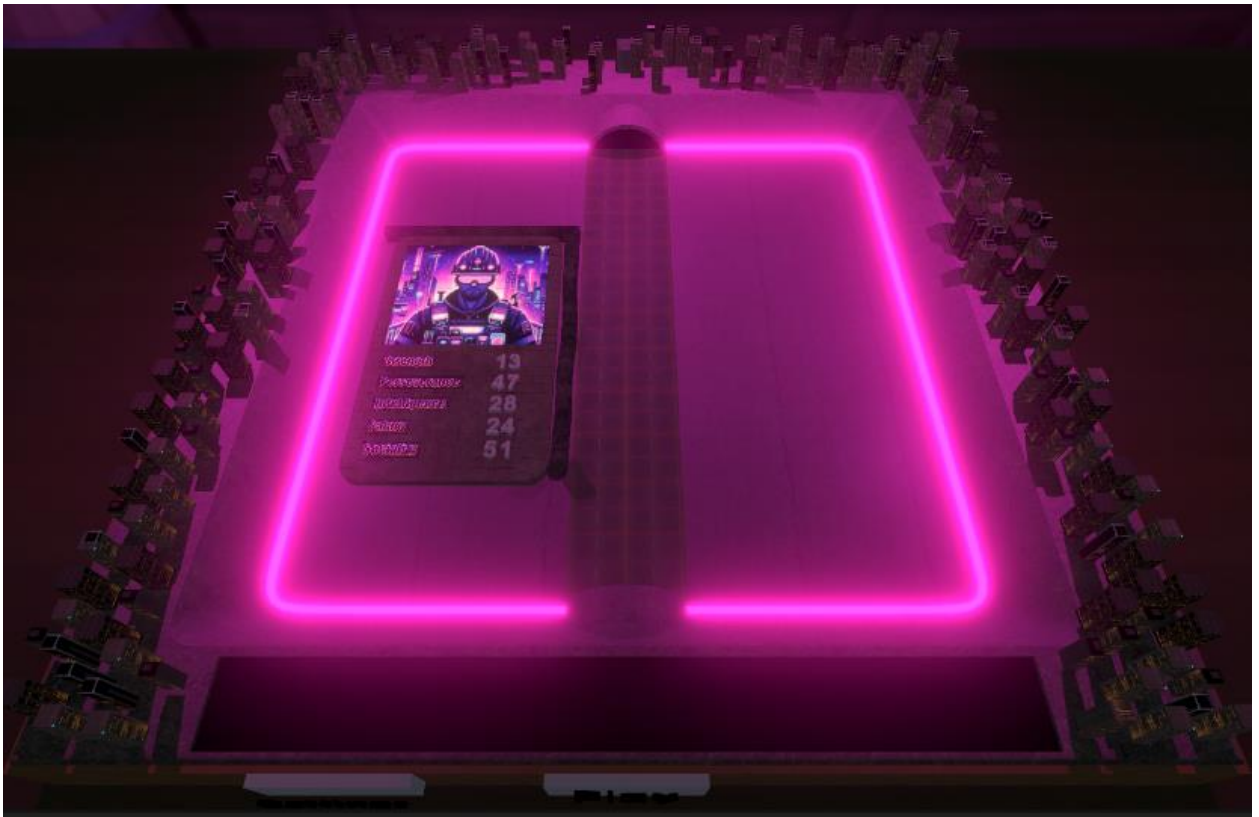


Рис 17. Вигляд новоствореної картки на сцені

Після генерації першої картки та отримання другої, вона візуалізується аналогічним способом та виконуються наступні дії:

1. Запуск скрипту бою

- Після візуалізації другої картки запускається скрипт "CardFight", який відповідає за механіку бою між двома картками.

2. Перевірка характеристик

- Скрипт "CardFight" порівнює відповідні характеристики двох карток, якщо картка захисту має більші відповідні параметри за атакуючу, може бути визначена дострокова перемога.

3. Дострокова перемога

- У випадку дострокової перемоги гра генерує повідомлення для гравця про результат бою, і бій завершується.

4. Віднімання параметрів

- Якщо дострокова перемога не була досягнута, скрипт "CardFight" віднімає відповідні параметри однієї картки від параметрів іншої.

- Цифрові значення характеристик карток оновлюються відповідно до результатів бою.

5. Компенсація надлишкових параметрів

- Запускається функція, яка компенсує недостатні параметри за допомогою надлишкових з використанням коефіцієнтів.
- Для кожного параметра визначається власний коефіцієнт, який показує, наскільки ефективно надлишковий параметр має змогу замінити інший.
- Параметри оновлюються з урахуванням коефіцієнтів, що дозволяє збалансувати характеристики карток та отримати максимум з їх характеристик.

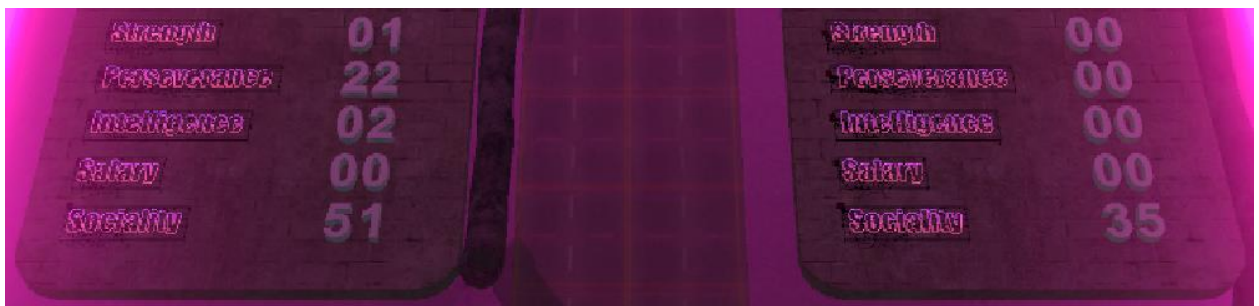


Рис 18. Результати обрахунку слабшої карти захисту

Гра продовжується доки у когось з гравців не закінчатся карти, через певний час очікування гра подасть світловий сигнал що означатиме капітуляцію гравця.

ВИСНОВОК

У даній кваліфікаційній роботі було розглянуто розробку ігрової логіки карткової гри з використанням NFC-карток з метою поєднання традиційних колекційних карток з сучасними технологіями. Основні досягнення, отримані в рамках цього дослідження, полягають у розробці структури гри з використанням Arduino для зчитування NFC-карток і Unity для обробки та візуалізації даних. Було запропоновано та реалізовано декілька цільових скриптів, таких як "CardCreator", "CardFight", "CardStateManager" і "InfoArray".

Практична цінність розробленої гри полягає в тому, що вона дозволяє користувачам взаємодіяти з фізичними картками через NFC, підвищуючи інтерактивність та залученість гравців. Використання Unity забезпечує можливість легкого масштабування та адаптації гри для різних платформ, включаючи мобільні пристрої. Економічна та соціальна цінність проекту полягає у поєднанні елементів колекціонування та сучасних технологій, що може привабити широку аудиторію та підвищити комерційну привабливість продукту. Використання відкритих платформ та доступних компонентів, таких як Arduino і Unity, знижує витрати на розробку та дозволяє створювати високоякісні продукти з обмеженими ресурсами. У порівнянні з традиційними картковими іграми, розроблена методика надає новий рівень інтерактивності та можливостей для гравців. Використання NFC-технології дозволяє автоматизувати процес зчитування та обробки даних, що значно спрощує гру та робить її більш зручною. Крім того, застосування віртуальних елементів відкриває можливості для додавання нових функцій та контенту без необхідності фізичних змін у картках. Методика підходу полягала у використанні двох основних компонентів: Arduino для зчитування та передачі даних NFC-карток і Unity для обробки цих даних і візуалізації у грі. Це дозволило забезпечити високий рівень інтеграції між фізичними та віртуальними елементами гри. Основний внесок полягає в розробці алгоритмів для обробки даних карток, реалізації механіки бою та компенсації параметрів за допомогою коефіцієнтів, що значно підвищило гнучкість та реалістичність гри.

Пропонується розширити функціонал, додавши нові типи карток та ігрових механік для підвищення різноманітності гри, а також покращити візуалізацію за рахунок використання складніших анімацій та графічних ефектів. Важливо також впровадити додаткові методи захисту даних на NFC-картках для запобігання шахрайству та несанкціонованому доступу. У майбутньому можна очікувати подальшого розвитку технологій, що дозволить

ще глибше інтегрувати фізичні та віртуальні елементи у карткові ігри. Використання доповненої реальності (AR) та штучного інтелекту (AI) може відкрити нові можливості для створення унікальних ігрових досвідів. Такий підхід не тільки підвищить якість гри, але й залучить нові аудиторії, що буде сприяти розвитку ігрової індустрії в цілому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Tom Igoe, Don Coleman, Brian Jepson Beginning NFC: O'Reilly Media, Inc., 2014p, 11 с.
2. What Are NFC Tags? — Complete Guide, url:
<https://stfalconcom.medium.com/what-are-nfc-tags-complete-guide-7e7095ca6c44>
3. Mike Geig Unity Game Development in 24 Hours: Sams Publishing, 2013 p., 37 с.
4. Jesse Liberty Learning C#: O'Reilly Media, Inc., 2002 p., 88 с.
5. Unity Documentation, url: <https://docs.unity.com/>
6. Unity Scripting API, url:
<https://docs.unity3d.com/Manual/ScriptingSection.html>
7. Best Card Games - NY Times Wirecutter, url:
<https://www.nytimes.com/wirecutter/reviews/best-card-games/>