КВАЛІФІКАЦІЙНА РОБОТА

Група ІПЗс-20

Вербіцький В.А.

2024

ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА

Факультет суспільних та прикладних наук

Кафедра інформаційних технологій

на правах рукопису

Вербіцький Віталій Анатолійович

УДК 004.4

Розробка гри 3D ранер на движку Unity

Спеціальність 121 – «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття кваліфікації бакалавр

Нормоконтроль

_____Стисло О.В.

(підпис, дата, розшифрування підпису)

Студент

_____Вербіцький В.А.

(підпис, дата, розшифрування підпису)

Допускається до захисту

Завідувач кафедри

_____к.т.н., доц. Ващишак С.П.

(підпис, дата, розшифрування підпису)

Керівник роботи

_____ к.т.н., доц. Слабінога М.О.

(підпис, дата, розшифрування підпису)

Івано-Франківськ – 2024

ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА

Факультет суспільних та прикладних наук

Кафедра інформаційних технологій

Освітній ступінь: «бакалавр»

Спеціальність: 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____

«____» _____ 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Вербіцькому Віталію Анатолійовичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:

Розробка гри 3D ранер на движку Unity

керівник роботи:

Слабінога Мар'ян Остапович, к.т.н., проф. каф. ІТ

затверджена наказом вищого навчального закладу від « 12 » березня 2024 року № 19/1

2. Термін подання студентом роботи 05.06.2024

3. Вихідні дані роботи: двигун розробки Unity, редактор коду Visual Studio

мова програмування C#.

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)

1. Огляд предметної області
2. Розробка структури
3. Структура проекту

5. Дата видачі завдання 14.03.2024

КОНСУЛЬТАНТИ РОЗДІЛІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Розділ	Консультант (прізвище, ініціали та посада)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Огляд предметної області	20.03.2024	Виконано
2.	Огляд існуючих рішень. Вибір засобів для реалізації	28.03.2024	Виконано
3.	Розробка загальної структури	13.04.2024	Виконано
4.	Розробка гри та тестування роботи системи	26.04.2024	Виконано
5.	Формування висновків	28.05.2024	Виконано
6.	Оформлення пояснювальної записки	03.05.2024	Виконано
7.	Оформлення графічного матеріалу та підготовка до захисту роботи	08.05.2024	Виконано

Студент _____ Вербіцький В.А.

(підпис) (прізвище та ініціали)

Керівник роботи _____ Слабінога М.О.

(підпис) (прізвище та ініціали)

Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Сторінка	Опис графічного матеріалу	Сторінка	Опис графічного матеріалу
15	Гемплей гри Temple Run	36	Коренева папка Assets
17	Гемплей гри Subway Surfers	37	Вміст каталогу Scripts
19	Гемплей гри Sonic Dash	46	Створення рівня-платформи з перешкодами
22	Гемплей гри Lara Croft: Relic Run	48	Помилка через яку персонаж не може рухатись далі
35	Вигляд структури вікна Project		

АНОТАЦІЯ

Метою даної кваліфікаційної роботи є розробка повнофункціональної гри у жанрі 3D ранер з використанням сучасних технологій та підходів до ігрового дизайну та програмування. Створення такої гри вимагає ретельного аналізу предметної області, вивчення існуючих рішень та застосування відповідних методів і засобів для реалізації поставленої задачі.

У першому розділі роботи проведено огляд предметної області відеоігор, зокрема жанру ранер. Тут розглянуті основні концепції, механіки та особливості ігор цього жанру, а також проаналізовано популярні існуючі рішення на ринку. Крім того, буде сформульована чітка постановка задачі для розробки гри 3D ранер з урахуванням вимог та обмежень.

Другий розділ буде присвячено розробці структури майбутньої системи. У ньому будуть обґрунтовані вибір засобів для реалізації, таких як ігровий рушій, бібліотеки та фреймворки. Також буде описана загальна структура системи та організація проекту.

У третьому розділі буде детально викладено процес реалізації та тестування розробленої системи. Будуть надані пояснення щодо реалізації ключових механік гри, таких як система переміщення персонажа, генерація перешкод, система бонусів, керування та інтерфейс користувача, а також аудіовізуальні ефекти. Крім того, буде висвітлено процес тестування роботи системи та забезпечення її стабільності та коректної поведінки.

Наприкінці роботи будуть підведені підсумки та сформульовані висновки, що стосуються досягнення поставленої мети, а також можливих напрямків подальшого вдосконалення та розвитку розробленої гри.

КЛЮЧОВІ СЛОВА: UNITY, C#, РАНЕР, КОЛАЙДЕР, ECS, VISUAL STUDIO, UNITY ASSET STORE.

SUMMARY

The purpose of this qualification work is to develop a fully functional game in the 3D runner genre using modern technologies and approaches to game design and programming. Creating such a game requires a thorough analysis of the subject area, studying existing solutions and applying appropriate methods and tools to accomplish the task.

The first section of the paper provides an overview of the subject area of video games, in particular the runner genre. It discusses the basic concepts, mechanics and features of games of this genre, as well as analyzes popular existing solutions on the market. In addition, a clear task statement will be formulated for the development of a 3D runner game, taking into account the requirements and limitations.

The second section will be devoted to the development of the structure of the future system. It will justify the choice of tools for implementation, such as a game engine, libraries, and frameworks. It will also describe the overall structure of the system and the organization of the project.

The third chapter will describe in detail the process of implementation and testing of the developed system. It will explain the implementation of key game mechanics, such as the character movement system, obstacle generation, bonus system, control and user interface, and audiovisual effects. In addition, the process of testing the system and ensuring its stability and correct behavior will be covered.

At the end of the paper, the results will be summarized and conclusions will be drawn regarding the achievement of the goal, as well as possible directions for further improvement and development of the developed game.

KEYWORDS: UNITY, C#, RUNNER, COLLIDER, ECS, VISUAL STUDIO, UNITY ASSET STORE.

ЗМІСТ

ВСТУП

Актуальність теми. Розробка відеоігор є однією з найдинамічніше розвиваючихся галузей інформаційних технологій. За останні десятиліття ця індустрія пережила стрімке зростання, ставши одним з найпопулярніших видів розваг у світі. 3D-ігри, зокрема, відіграють все більш значущу роль у цьому сегменті, завдяки своїй реалістичній графіці, захоплюючому геймплею.

Жанр 3D-ранер є одним з найпоширеніших та улюблених жанрів 3D-ігор. Він характеризується динамічним геймплеєм, простим управлінням та яскравою графікою. 3D-ранер приваблює гравців завдяки своїй простоті та доступності.

Вивчення теоретичних основ та практичних аспектів розробки 3D-ранер на движку Unity є актуальною темою, яка має значний науковий та практичний інтерес.

Мета дослідження. Розробка 3D-гри жанру ранер на ігровому двигуні Unity з використанням.

Завдання дослідження. Для досягнення завдання були визначені наступні завдання:

- провести огляд предметної області та проаналізувати існуючі рішення в жанрі 3D-ранер;
- вибрати засоби реалізації та реалізувати основи гри;
- провести тестування та налагодження створеної гри.

Практичне значення. Результатом даного дослідження полягає створенні 3D-ранер на движку Unity. Розроблена гра може слугувати навчальним прикладом для розробників 3D-ігор, які хочуть вивчити процес створення 3D-ранер на движку Unity. Також може бути використана у розважальних ціялх.

Апробація результатів дослідження. Матеріали кваліфікаційної роботи були представлені на XI Міжнародної наукової конференції «Студентські

наукові дискусії поза форматом», яка відбулася 11 квітня 2023 року в Університеті Короля Данила.

Структура. Загальний обсяг основної частини – 43 сторінки. Список використаних джерел – 20. Всього розділів – 3.

РОЗДІЛ 1. ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Огляд предмету області

Розробка відеоігор є хорошою темою для написання кваліфікаційної роботи тай взагалі для розвитку індустрії комп’ютерних ведеоігор в Україні, яка розвивається чи не найшвидше в сфері розваг. Світ геймінгу переживає справжній бум. З'являються нові ігри, які за лічені місяці стають культовими, а ремастери улюблених класиків дарують нове життя легендарним сюжетам.

Люди різного віку грають у відеоігри з різних причин [1]. За останні роки інтерес до цієї індустрії збільшується. Геймінг з кожним роком набирає обертів.

Відеоігри як розваги та задоволення: на мою думку це найпоширеніша причина. Відеоігри можуть бути захоплюючим способом розслабитися, особливо після робочого часу, зняти стрес і просто весело провести час. Вони пропонують широкий спектр жанрів і стилів, тому кожен може знайти щось собі до вподоби, чи то стрілялки, гонки, рольові ігри, ранер, головоломки чи щось інше.

Втеча від реальності: відеоігри можуть запропонувати втечу від повсякденного життя і перенести вас у інші світи та часи. Це може бути особливо привабливим для людей, які переживають стрес або труднощі у своєму реальному житті.

Соціальна взаємодія: відеоігри можуть бути чудовим способом спілкуватися з друзями та родиною. Багато людей грають онлайн разом, а деякі ігри навіть спеціально розроблені для командної гри. Багато хто так знаходить собі друзів та однодумців.

Навчання та розвиток: відеоігри можуть навчити нас новим навичкам і знанням. Наприклад, стратегічні ігри можуть навчити критичному мисленню та прийняттю рішень, а головоломки - просторовому міркуванню.

Використання комп'ютерних ігор в навчанні має багато переваг, ось деякі знайважливіших:

– покращення навичок вирішення проблем та прийняття рішень: Багато комп'ютерних ігор вимагають від гравців вирішувати проблеми, стратегічно мислити та приймати рішення. Це може допомогти учням розвинути ці навички, які будуть їм корисні як у школі, так і в реальному житті;

– підвищення мотивації та залучення: комп'ютерні ігри можуть бути дуже захоплюючими та мотивувальними, що може зробити навчання більш приємним та цікавим для учнів;

– розвиток когнітивних навичок: деякі комп'ютерні ігри можуть допомогти учням розвинути когнітивні навички, такі як пам'ять, увага та швидкість обробки інформації.

Досягнення та визнання. Також можуть давати нам відчуття досягнення та визнання. Коли ми проходимо складний рівень або перемагаємо гідного супротивника, це може бути дуже корисним для нашої самооцінки.

Творчість та самовираження. Відеоігри можуть бути способом виразити себе та свою творчість. Багато ігор дозволяють гравцям робити вибір і налаштовувати своїх персонажів, що може бути способом дослідити свою особистість і цінності.

Геймінг стає все більш масовим явищем. Кіберспорт вже здобув широке визнання, а його включення до переліку олімпійських дисциплін вже не здається фантастикою. За даними NewZoo, на кінець 2020 року у світі налічувалося близько 2,7 млрд гравців – це майже половина дорослого населення планети!

Кіберспорт також стимулював стрімкий розвиток стрімінгу – онлайн-трансляцій проходження ігор, які збирають тисячі глядачів. Це не лише розвага, але й можливість для геймерів поділитися своїм досвідом, отримати підтримку та заробити на своїй майстерності [19].

Відеоігри давно перестали бути просто розвагою. Сьогодні вони перетворилися на величезну індустрію, яка генерує мільярди доларів щороку.

Існує безліч способів заробляти на відеоіграх, як для професіоналів, так і для аматорів.

Професіональний геймінг. Професійний геймінг - це вид спорту, в якому гравці змагаються один з одним у відеоіграх на змагальному рівні. Професійні геймери можуть заробляти великі гроші на призових, зарплатах та спонсорських угодах.

Переваги:

- можливість заробити великі гроші;
- подорожі світом та участь у змаганнях;
- визнання та слава.

Недоліки:

- дуже висока конкуренція;
- потрібні багато годин тренувань;
- може бути дуже стресовим.

Стримінг. Стримінг - це трансляція ігрового процесу в прямому ефірі на платформі, такій як Twitch або YouTube. Стримери можуть заробляти гроші за рахунок пожертв, реклами або спонсорських угод.

Переваги:

- можливість заробити гроші, роблячи те, що вам подобається;
- потенційно велика аудиторія;
- можливість спілкуватися з глядачами.

Недоліки:

- може бути складно набрати аудиторію;
- потрібне якісне обладнання та підключення до Інтернету;
- може бути дуже конкурентним.

Створення контенту.Створення контенту - це створення відео, статей або інших матеріалів, пов'язаних з відеоіграми. Створення контенту може включати в себе огляди ігор, проходження, посібники та електронні книги. Створці контенту можуть заробляти гроші за рахунок реклами, спонсорських угод та партнерських програм.

Переваги:

- можливість заробити гроші, роблячи те, що вам подобається;
- потенційно велика аудиторія;
- можливість поділитися своєю любов'ю до відеоігор з іншими.

Недоліки:

- може бути складно набрати аудиторію;
- потрібні навички письма або відеозйомки;
- може бути дуже конкурентним.

Кіберспорт. Кіберспорт - це змагальний вид спорту, в якому команди або окремі гравці змагаються один з одним у відеоіграх. Кіберспортсмени можуть заробляти гроші на призових, зарплатах та спонсорських угодах.

Переваги:

- можливість заробити великі гроші;
- подорожі світом та участь у змаганнях;
- визнання та слава.

Недоліки:

- дуже висока конкуренція;
- потрібні багато годин тренувань;
- може бути дуже стресовим.

Розробка ігор. Розробка ігор - це процес створення відеоігор. Розробники ігор можуть працювати в великих компаніях або створювати власні ігри. Розробники ігор можуть заробляти гроші за рахунок продажу ігор, реклами.

Переваги:

- можливість створювати власні ігри;
- потенційно велика аудиторія;
- можливість поділитися своїми творчими ідеями з іншими;

Недоліки:

- може бути складно створити успішну гру;
- потрібні навички програмування та дизайн.

Потреби користувачів. Ігри жанру раннер - це популярний жанр мобільних ігор, який пропонує простий, але захоплюючий ігровий процес. Ці ігри зазвичай мають простий дизайн, де гравець керує персонажем, який біжить по нескінченному шляху, уникаючи перешкоди і збирає різні предмети.

Щоб створити таку гру, потрібно розуміти потреби та очікування користувачів.

Простий і захоплюючий ігровий процес. Користувачі хочуть, щоб ігри-раннери були простими у вивченні, але складними у освоєнні. Це означає, що механіка керування має бути інтуїтивно зрозумілою для користувача, але гра, має пропонувати виклик, який змусить гравців повертатися знову і знову.

Різноманітність контенту. Користувачі хочуть, щоб ігри-раннери пропонували різноманітний контент, щоб гра залишалася свіжою та цікавою. Це може включати різні середовища, перешкоди, бонуси та персонажів.

Красива графіка та звукові ефекти. Користувачі хочуть, щоб ігри-раннери мали приємну для очей графіку та якісні звукові ефекти. Це допоможе створити захоплюючий ігровий досвід.

Чутливе керування. Користувачі хочуть, щоб ігри-раннери мали чуйне керування, щоб гравці могли точно керувати своїми персонажами. Це особливо важливо в іграх-раннерах, де уникання перешкод є ключем до успіху.

Можливість змагатися з друзями. Користувачі хочуть, щоб ігри-раннери пропонували можливість змагатися з друзями, та й взагалі мати світовий рейтинг, де гравці можуть відслідковувати досягнення один одного. Це може бути зроблено за допомогою таблиць лідерів, соціальних функцій або режиму мультиплеєра.

Безкоштовна або доступна ціна. Користувачі хочуть, щоб ігри-раннери були безкоштовними або мали доступну ціну. Це робить гру більш доступною для широкої аудиторії.

Відсутність нав'язливої реклами. Користувачі не хочуть, щоб ігри-раннери були переповнені нав'язливою рекламою. Це може зробити ігровий досвід неприємним і змусити гравців уникати гри.

Регулярні оновлення. Користувачі хочуть, щоб ігри-раннери регулярно оновлювалися новим контентом та функціями. Це допоможе зберегти інтерес гравців до гри.

Розуміння потреб користувачів є головним моментом для створення успішної гри. Розробники, які зосередяться на створенні простих, але захоплюючих ігор з різноманітним контентом, красивою графікою, чуйним керуванням та збалансованою монетизацією, ймовірно, створять ігри, які сподобаються користувачам і залишаться популярними протягом багатьох років.

1.2. Огляд існуючих рішень

3D-раннери - це популярний жанр мобільних ігор, який пропонує динамічний і захоплюючий ігровий процес. Ці ігри зазвичай ставлять гравця на місце персонажа, який біжить по 3D-середовищу, уникаючи перешкод, збираючи предмети та виконуючи різні завдання.

3D-раннери відрізняються від своїх 2D-попередників більш продуманою графікою, більш складними рівнями та більшим розмаїттям ігрових механік. Це робить їх більш захоплюючими та динамічними, що робить їх популярними серед гравців усіх вікових категорій.

У цьому розділі ми розглянемо деякі з найпопулярніших існуючих 3D-раннерів. Ми проаналізуємо механіку ігрового процесу, візуальний стиль, наратив та інші ключові аспекти, які формують унікальний ігровий досвід кожної гри (рис. 1.1).

Temple Run - це мобільна гра-раннер, розроблена компанією Imangi Studios, яка захопила світ мобільних ігор з моменту свого випуску в 2011 році [16]. Ця гра пропонує простий, але захоплюючий ігровий процес, який занурює гравців у світ небезпечних храмів та скарбів. Що ж, давайте детально розглянемо Temple Run, проаналізувавши її успіх, ігрові механіки, візуальний стиль, контент та інші ключові аспекти, що сформували ігровий досвід.

Рисунок 1.1 – Гемплей гри Temple Run

Ігровий процес. У Temple Run гравці беруть на себе роль археолога, який вкрав артефакт з загубленого храму. Розгнівані боги переслідують гравця, який вимушений бігти по нескінченному шляху, уникаючи перешкод, збираючи монети та виконуючи різні завдання.

Гра пропонує просте та інтуїтивно зрозуміле управління, яке легко освоїти, але складно опанувати. Гравці можуть ковзати вліво, вправо, вгору та вниз, щоб переміщатися по шляху, перестрибувати через перешкоди, збирати бонуси та уникати падіння з обривів. З часом швидкість гри зростає, що робить ігровий процес все більш динамічним і складним.

Візуальний стиль. Temple Run має мальовничий візуальний стиль, який поєднує в собі елементи пригодницьких ігор та індійської міфології. Гра використовує 3D графіку з мультяшним дизайном, який робить її візуально приємною та легкою для сприйняття. Середовища гри різноманітні та постійно змінюються, пропонуючи гравцям нові пейзажі та виклики.

Контент. Temple Run пропонує широкий спектр контенту, який підтримує зацікавленість гравців протягом тривалого часу. До ключових елементів контенту належать:

– персонажі: гравці можуть вибирати з десятків різних персонажів, кожен з яких має свій унікальний зовнішній вигляд та здібності;

– артефакти: гравці можуть збирати різні артефакти, які надають їм бонуси та покращують характеристики;

– режим "Без кінця": цей режим кидає виклик гравцям пробігти якнайдовше, уникаючи перешкод та збираючи монети.

– режим "Втеча": цей режим пропонує гравцям бігти через різні рівні з чітко визначеними цілями;

– досягнення та лідерські таблиці: гравці можуть розблоковувати досягнення, змагатися з друзями та іншими гравцями в лідерських таблицях.

Монетизація. Temple Run використовує безкоштовну модель з внутрішніми покупками. Гравці можуть грати в гру безкоштовно, але вони можуть купувати додаткові життя, монети та інші предмети, щоб покращити свій прогрес та отримати доступ до ексклюзивного контенту. Ця модель монетизації виявилася дуже успішною, оскільки вона дозволяє гравцям отримати доступ до гри безкоштовно, але також пропонує їм можливість підтримати розробників та отримати додаткові переваги.

Subway Surfers - це безкоштовна мобільна гра-раннер, розроблена компанією Kiloo та SYBO Games, яка завоювала серця мільйонів гравців по всьому світу [15]. Випущена в 2012 році, гра швидко стала вірусною, очоливши рейтинги завантажень у багатьох країнах і здобувши численні нагороди.

Subway Surfers пропонує простий, але захоплюючий ігровий процес, який завдяки своєї простоті та динамічності став феноменом мобільних ігор.

Ігровий процес. У Subway Surfers гравці беруть на себе роль юного графітера, який тікає від інспектора метро та його собак по жвавих залізничних коліях. Гравці біжать по нескінченному треку, уникаючи поїздів, вагонів метро та інших перешкод, збираючи монети та спеціальні бонуси (рис. 1.2).

Рисунок 1.2 – Гемплей гри Subway Surfers

Гра пропонує просте та інтуїтивно зрозуміле управління, яке легко освоїти, але складно опанувати. Гравці можуть ковзати вліво, вправо, вгору та вниз, щоб переміщатися по треку, перестрибувати через перешкоди та збирати бонуси. З часом швидкість гри зростає, що робить ігровий процес все більш динамічним і складним.

Візуальний стиль. Subway Surfers має барвистий та мальовничий візуальний стиль, який привертає увагу гравців усіх віків. Гра використовує 3D-графіку з мультяшним дизайном, який робить її візуально приємною та легкою для сприйняття. Середовища гри різноманітні та постійно змінюються, пропонуючи гравцям нові пейзажі та виклики.

Контент. Subway Surfers пропонує широкий спектр контенту, який підтримує зацікавленість гравців протягом тривалого часу. До ключових елементів контенту належать:

– персонажі: гравці можуть вибирати з десятків різних персонажів, кожен з яких має свій унікальний зовнішній вигляд та здібності;

– скейтборди: гравці можуть збирати та використовувати різні скейтборди, які надають їм бонуси та покращують характеристики;

– предмети: гравці можуть збирати різні предмети, такі як ключі, джети та магніти, які допомагають їм долати перешкоди та збільшувати результат;

– місії: гра пропонує щоденні та щотижневі місії, які дають гравцям додаткові нагороди та стимулюють повторне проходження;

– оновлення: гра регулярно оновлюється новим контентом, персонажами, місіями та іншими функціями, що робить її свіжою та цікавою.

Монетизація. Subway Surfers використовує безкоштовну модель з внутрішніми покупками. Гравці можуть грати в гру безкоштовно, але вони можуть купувати монети, дорогоцінні камені та інші предмети, щоб покращити свій прогрес та отримати доступ до ексклюзивного контенту. Ця модель монетизації виявилася дуже успішною, оскільки вона дозволяє гравцям отримати доступ до гри безкоштовно, але також пропонує їм можливість

підтримати розробників та отримати додаткові переваги.

Успіх. Subway Surfers стала однією з найуспішніших мобільних ігор усіх часів. Станом на 2023 рік гру було завантажено понад 3 мільярди разів, і вона приносить своєму розробнику мільйони доларів щороку. Її успіх можна приписати ряду факторів, включаючи:

– простий та захоплюючий ігровий процес: Subway Surfers пропонує простий і зрозумілий ігровий процес, який легко освоїти, але складно опанувати. Це робить гру доступною для гравців усіх віків і рівнів навичок;

– барвистий та мальовничий візуальний стиль: гра має привабливий візуальний стиль, який подобається гравцям усіх віків;

– регулярні оновлення з новим контентом: гра постійно оновлюється новим контентом, що робить її свіжою та цікавою;

– ефективна стратегія монетизації: безкоштовна модель з внутрішніми покупками дозволяє гравцям отримати доступ до гри безкоштовно,

але також пропонує їм можливість підтримати розробників та отримати додаткові переваги.

Вплив. Subway Surfers справила значний вплив на мобільну ігрову індустрію. Гра популяризувала жанр 3D-раннерів та надихнула на створення численних інших ігор у цьому жанрі. Subway Surfers також показала, що безкоштовні мобільні ігри з внутрішніми покупками можуть бути дуже успішними, якщо вони пропонують якісний ігровий досвід.

Sonic Dash - це мобільна гра-раннер, розроблена компанією SEGA, яка занурює гравців у світ легендарного їжачка Соніка. Випущена в 2013 році, гра швидко стала популярною, завоювавши серця мільйонів фанатів Соніка по всьому світу [14].

Sonic Dash пропонує динамічний і захоплюючий ігровий процес, який поєднує в собі знайомі елементи з новим та свіжим досвідом. Проаналізуємо її успіх, ігрові механіки, візуальний стиль, контент та інші ключові аспекти, що сформували унікальний ігровий досвід (рис. 1.3).

Рисунок 1.3 – Гемплей гри Sonic Dash

Ігровий процес. У Sonic Dash гравці беруть на себе роль Соніка або одного з його друзів, які біжать по нескінченному шляху, збираючи кільця, уникаючи перешкод та борючись з доктором Еґманом та його роботами.

Гра пропонує швидкий та динамічний ігровий процес, де гравцям необхідно реагувати швидко та точно, щоб уникнути зіткнення з перешкодами. Гравці можуть використовувати різні здібності Соніка та його друзів, такі як прискорення, стрибки та атаки, щоб долати перешкоди та збирати більше кілець. З часом швидкість гри зростає, що робить ігровий процес все більш складним та захоплюючим.

Візуальний стиль. Sonic Dash має яскравий та барвистий візуальний стиль, який відповідає характерному стилю франшизи Sonic. Гра використовує 3D-графіку з мультяшним дизайном, який робить її візуально приємною та легкою для сприйняття. Середовища гри різноманітні та постійно змінюються, пропонуючи гравцям нові пейзажі та виклики.

Контент. Sonic Dash пропонує широкий спектр контенту, який підтримує зацікавленість гравців протягом тривалого часу. До ключових елементів

контенту належать:

– персонажі: гравці можуть вибирати з Соніка, Тейлза, Емі Роуз, Наклза та інших знайомих персонажів з франшизи Sonic;

– здібності: кожен персонаж має унікальні здібності, які гравці можуть використовувати для покращення своїх результатів;

– режим "Без кінця": цей режим кидає виклик гравцям пробігти якнайдовше, збираючи кільця та уникаючи перешкод;

– режим "Змагання": цей режим пропонує гравцям змагатися з друзями та іншими гравцями за найкращий результат;

– досягнення та лідерські таблиці: гравці можуть розблоковувати досягнення, змагатися з друзями та іншими гравцями.

Монетизація. Sonic Dash використовує безкоштовну модель з внутрішніми покупками. Гравці можуть грати в гру безкоштовно, але вони можуть купувати додаткові життя, червоні кільця та інші предмети, щоб покращити свій прогрес

та отримати доступ до ексклюзивного контенту. Ця модель монетизації виявилася дуже успішною, оскільки вона дозволяє гравцям отримати доступ до гри безкоштовно, але також пропонує їм можливість підтримати розробників та отримати додаткові переваги.

Успіх. Станом на 2023 рік гру було завантажено понад 500 мільйонів разів, і вона принесла компанії SEGA мільйони доларів прибутку. Її успіх можна приписати ряду факторів, включаючи:

– швидкий та динамічний ігровий процес: Sonic Dash пропонує захоплюючий ігровий процес, який тримає гравців у напрузі;

– яскравий та барвистий візуальний стиль: гра має привабливий візуальний стиль, який подобається фанатам Sonic та новим гравцям;

– знайомі та улюблені персонажі: Sonic Dash дозволяє гравцям керувати Соником та його друзями, що робить гру ще більш привабливою для фанатів франшизи;

– регулярні оновлення з новим контентом: гра постійно оновлюється новими персонажами, режимами та іншим контентом, що робить її свіжою та цікавою;

– ефективна стратегія монетизації: безкоштовна модель з внутрішніми покупками дозволяє гравцям отримати доступ до гри безкоштовно, але також пропонує їм можливість підтримати розробників та отримати додаткові переваги.

Вплив. Sonic Dash справила значний вплив на мобільну ігрову індустрію. Гра популяризувала жанр 3D-раннерів та надихнула на створення численних інших ігор у цьому жанрі. Sonic Dash також показала, що безкоштовні мобільні ігри з внутрішніми покупками можуть бути дуже успішними, якщо вони пропонують якісний ігровий досвід.

Це блискавична та захоплююча мобільна гра, яка завоювала серця мільйонів гравців по всьому світу. Її швидкий ігровий процес, яскравий візуальний стиль, знайомі персонажі та ефективна стратегія монетизації зробили її однією з найуспішніших мобільних ігор всех часів. Sonic Dash не

лише розважає гравців, але й справила значний вплив на мобільну ігрову індустрію, популяризувавши жанр 3D-раннерів та продемонструвавши потенціал безкоштовних мобільних ігор з внутрішніми покупками.

Lara Croft: Relic Run - це мобільна гра-раннер, розроблена компанією Square Enix, яка вийшла в 2016 році [13]. Гра пропонує захоплюючий ігровий процес, поєднуючи елементи безперервного бігу з розгадуванням загадок та битвами з ворогами. Беручи на себе роль легендарної розкрадачки гробниць Лари Крофт, гравці вирушають у захоплюючу пригоду по всьому світу в пошуках стародавніх реліквій (рис. 1.4).

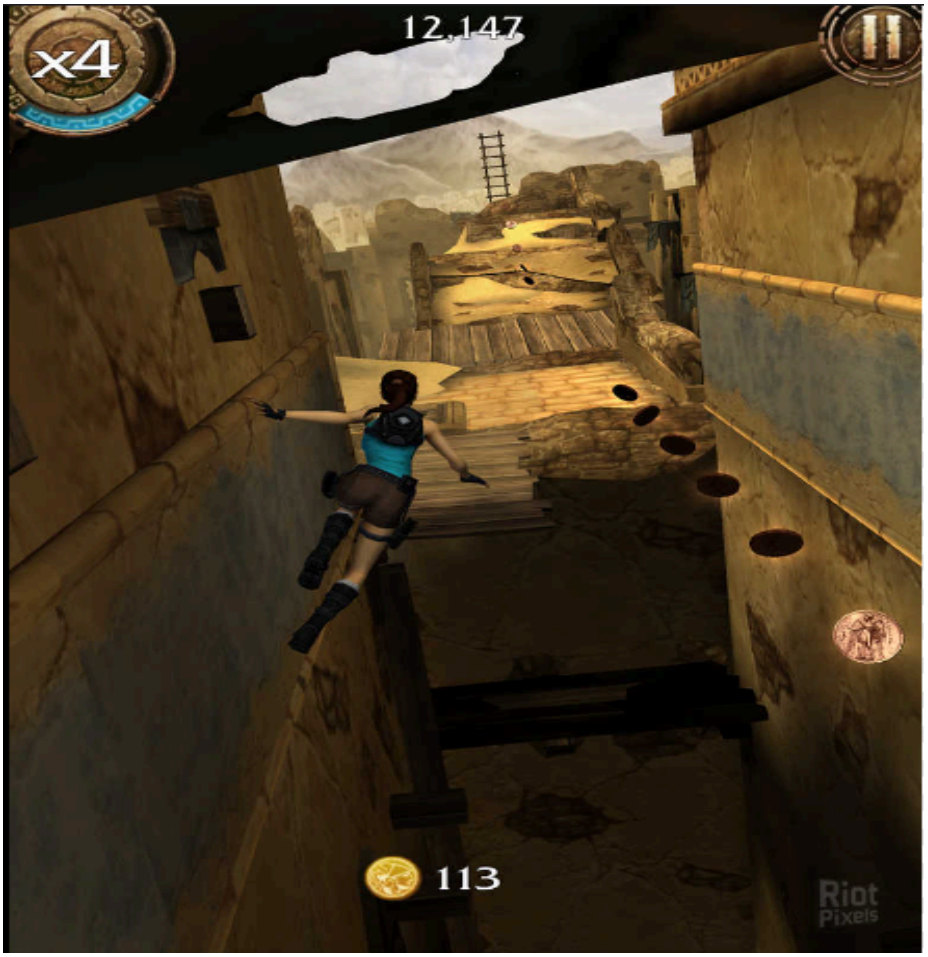


Рисунок 1.4 – Гемплей гри Lara Croft: Relic Run

Ігровий процес. Lara Croft: Relic Run пропонує динамічний і насичений подіями ігровий процес. Гравці керують Ларою, яка біжить по нескінченних рівнях, уникаючи перешкод, стрибаючи по платформах, збираючи монети та

коштовності. На своєму шляху вона також стикається з ворогами, яких можна знешкодити за допомогою пістолетів, дробовиків та інших видів зброї.

Гра пропонує різноманітні механіки, які урізноманітнюють ігровий процес. Гравці можуть використовувати мотоцикли та снігоходи для більш швидкого пересування, а також акробатичні трюки, щоб красиво перестрибувати перешкоди. Крім того, на рівнях розкидані стародавні механізми, які необхідно активувати, щоб відкрити нові шляхи та дістатися до реліквій.

Візуальний стиль. Lara Croft: Relic Run має приємний візуальний стиль з мультяшною графікою. Гра використовує яскраві кольори та деталізовані об'єкти, які створюють атмосферу захоплюючих пригод. Середовища гри різноманітні й включають джунглі, стародавні храми, засніжені гори та підземні печери.

Контент. Lara Croft: Relic Run пропонує широкий спектр контенту, який підтримує зацікавленість гравців протягом тривалого часу. До ключових елементів контенту належать:

– різноманітні рівні: гра пропонує безліч рівнів з унікальним дизайном та завданнями;

– зброя та спорядження: гравці можуть збирати та покращувати різні види зброї, щоб ефективніше боротися з ворогами;

– костюми Лари: доступно безліч різних костюмів для Лари, які відкриваються за проходження рівнів або за допомогою внутрішніх покупок;

– щоденні завдання та місії: гра пропонує щоденні завдання та місії, які дають гравцям додаткові нагороди та стимулюють повторне проходження;

– колекціонування реліквій: основна мета гри - зібрати різноманітні стародавні реліквії, розкидані по рівнях.

Монетизація. Lara Croft: Relic Run використовує безкоштовну модель внутрішніми покупками. Гравці можуть грати в гру безкоштовно, але вони можуть купувати додаткові життя, монети, коштовності та інші предмети, щоб покращити свій прогрес та отримати доступ до ексклюзивного контенту. Ця

модель монетизації дозволяє гравцям отримати доступ до гри безкоштовно, але також пропонує їм можливість підтримати розробників та отримати додаткові переваги.

Успіх. Lara Croft: Relic Run стала хорошою грою в своєму жанрі. Станом на 2023 рік гру було завантажено понад 100 мільйонів разів, і вона отримала багато позитивних відгуків від гравців та критиків. Її успіх можна приписати ряду факторів, включаючи:

– захоплюючий і динамічний ігровий процес: Lara Croft: Relic Run пропонує безперервний біг з елементами розгадування загадок та битв, що робить її цікавою та захоплюючою;

– впізнаваний персонаж: гра заснована на популярній франшизі Tomb Raider, що привернуло до неї багато фанатів Лари Крофт;

– красивий візуальний стиль: Lara Croft: Relic Run має приємний візуальний стиль з мультяшною графікою, який підкреслює атмосферу пригод;

– різноманітний контент: гра пропонує безліч рівнів, зброї, костюмів та інших елементів, які підтримують зацікавленість гравців протягом тривалого часу.

Вплив. Lara Croft: Relic Run справила значний вплив на мобільну ігрову індустрію. Гра популяризувала жанр безперервного ранера з елементами розгадування загадок та битв, відрізняючись цим від інших ранерів, а також надихнула на створення численних інших ігор у цьому жанрі. Lara Croft: Relic Run також продемонструвала, що популярні франшизи можуть успішно адаптуватися до мобільних платформ.

1.3. Постановка задачі

Метою даної кваліфікаційної роботи є розробка повнофункціональної відеогри у популярному жанрі 3D ранер на ігровому рушії Unity. Гра має забезпечувати захопливий та динамічний ігровий процес, відповідаючи сучасним вимогам та тенденціям ігрової індустрії.

Основними механіками, які повинні бути реалізовані у кінцевому програмному продукті, є:

1. Система переміщення ігрового персонажа. Персонаж має автоматично рухатися вздовж заздалегідь визначеного маршруту. Необхідно передбачити можливість керування стрибками та зміною напрямку руху з метою подолання перешкод.

2. Генерація рівня. На шляху персонажа мають генеруватися рівні з об'єктами (перешкодами), зіткнення з якими призводить до програшу.

3. Система бонусів. Розподіл уздовж маршруту спеціальних об’єктів, які персонаж може збирати для отримання переваги.

4. Керування та інтерфейс користувача. Потрібно реалізувати зручну систему керування рухом персонажа та інтуїтивний графічний інтерфейс для відображення поточного стану гри.

Реалізація зазначених механік дозволить створити захопливу та сучасну гру у жанрі 3D ранер, яка відповідатиме очікуванням користувачів та зможе конкурувати з популярними представниками цього жанру на ринку відеоігор.

Висновок до розділу 1

У рамках даного дослідження було проведено всебічне вивчення предметної області, яке охоплювало специфіку жанру 3D-ранер.

Здійснено аналіз існуючих рішень у жанрі 3D-ранер, що дозволило виокремити ключові механіки геймплею, візуальні стилі та особливості дизайну цих ігор.

На ґрунті проведеного аналізу чітко сформульовано задачу дослідження, яка полягає у розробці 3D-ранер на движку Unity, атмосферним візуальним стилем.

Очікується, що реалізація цього проекту дасть змогу продемонструвати ефективність використання движка Unity для розробки 3D-ігор та створити цікаву та захоплюючу гру, яка зацікавить широку аудиторію гравців.

РОЗДІЛ 2. РОЗРОБКА СТРУКТУРИ

2.1. Вибір засобів для реалізації

Для розробки мобільної гри-раннера на C# вам знадобляться відповідні інструменти та програмне забезпечення. У цьому розділі будуть розглянуті два популярні вибори: Unity та Visual Studio.

Unity - це кроссплатформний ігровий двигун, який дозволяє розробникам створювати 2D та 3D ігри для різних платформ, включаючи ПК, мобільні пристрої, консолі та віртуальну реальність. Unity використовує мову C# для написання сценаріїв та логіки гри [1].

Unity має візуально орієнтований інтерфейс користувача, який полегшує розробку ігор, навіть для початківців. Інтерфейс складається з кількох вікон, включаючи:

1. Вікно ієрархії: показує всі об'єкти у вашій сцені, включаючи персонажів, декорації, елементи інтерфейсу та інші.
2. Вікно сцени: відображає 3D-вид вашої сцени, де ви можете розташовувати об'єкти та налаштовувати їхні позиції.
3. Інспектор: відображає властивості та параметри вибраного об'єкта, які можна змінювати для налаштування його поведінки.
4. Вікно проекту: показує всі ваші ресурси, такі як сценарії, моделі, текстури та аудіофайли.

Переваги:

Легкий у вивченні та використанні. Має інтуїтивно зрозумілий інтерфейс та велику кількість навчальних матеріалів, що робить його доступним для початківців.

Великий вибір ресурсів Unity Asset Store пропонує безліч безкоштовних та платних ресурсів, таких як моделі, текстури, аудіо та інші, які допоможуть вам швидко розпочати роботу над вашою грою.

Спільнота та підтримка в Unity має велику та активну спільноту розробників, де ви можете знайти допомогу та поради. Офіційна документація Unity також дуже детальна та корисна.

Кросплатформна розробка програми дозволяє створювати ігри для різних платформ з однієї кодової бази, що економить час і зусилля.

Підтримка C#: Unity використовує C# як мову програмування, яка є популярною та легкою у вивченні мовою з великою спільнотою розробників [2].

З недоліків можна зазначити що Unity може бути складним для складних проектів: Unity може бути складною для використання при розробці складних 3D-ігор з великою кількістю контенту та складних систем.

Безкоштовна версія має деякі обмеження, такі як водяний знак на ваших іграх та обмеження на розмір публікації.

Visual Studio - це інтегроване середовище розробки (IDE) від компанії Microsoft, яке використовується для створення програмного забезпечення, веб-сайтів та ігор [18]. VS підтримує широкий спектр мов програмування, включаючи мову C# яка і використовується для реалізації гри.

Visual Studio має складний та потужний інтерфейс користувача. Він пропонує безліч вікон, панелей інструментів та меню, які дозволяють розробнику писати код, тестувати, налагоджувати та розгортати програмне забезпечення. Деякі з ключових компонентів інтерфейсу програми включають:

1. Редактор коду. Це основне вікно, де розробники пишуть та редагують код. Програма пропонує синтаксичне підсвічування та навіть автозаповнення коду що дуже спрощує роботу у програмі. Також є і інші функції, які полегшують написання коду.

2. Інспектор об'єктів. Ця панель відображає властивості та параметри вибраного об'єкта в коді. Розробники можуть змінювати ці значення, щоб налаштувати поведінку програми.

3. Вікно виводу це панель яка відображає повідомлення про помилки, та вказує де саме допущено помилку, для комфортної роботи це саме те що

потрібно для розробника гри. Також показує і іншу інформацію, яка може бути корисною під час розробки.

4. Засоби налагодження. Visual Studio має широкий спектр інструментів для налагодження. Вони дозволяють розробникам покроково виконувати код, перевіряти значення змінних та знаходити помилки

Переваги:

1. Потужний IDE: Visual Studio - це один з найпотужніших IDE на ринку. Він пропонує безліч функцій, які допомагають розробникам писати код, тестувати, налагоджувати та розгортати програмне забезпечення.

2. Підтримка широкого спектру мов програмування: VS підтримує широкий спектр мов програмування, що робить його універсальним інструментом для розробників.

3. Інтеграція з іншими інструментами Microsoft: VS інтегрується з іншими інструментами Microsoft, такими як SQL Server та Azure, що робить його зручним для розробки підприємних застосунків.

4. Велика спільнота та підтримка: Visual Studio має велику та активну спільноту розробників, де можна знайти допомогу та поради. Офіційна документація VS також дуже детальна та корисна.

Недоліки:

1. Складний інтерфейс. Visual Studio може бути складним для початківців через його складний інтерфейс та безліч функцій.

2. Ліцензія. Visual Studio платний, і вам знадобиться купити ліцензію для його використання.

3. Не такий простий у використанні, як Unity: Unity - це ігровий двигун, який може бути простішим у використанні для розробки ігор, особливо для початківців.

Я ретельно обирав інструменти для розробки моєї мобільної гри-раннера на C# і зупинився на комбінації Unity та Visual Studio.

Unity - це чудовий вибір для початківців, завдяки його інтуїтивно зрозумілому інтерфейсу, великій кількості навчальних матеріалів та доступності

безкоштовної версії. Цей ігровий двигун дозволить швидко розпочати роботу над візуальною частиною гри, використовуючи готові ресурси та інструменти.

Visual Studio, з іншого боку, пропонує потужну інтегроване середовище розробки, яка стане незамінною при написанні коду на C#. Цей IDE надає безліч інструментів для налагодження та тестування, що гарантує високу якість мого програмного забезпечення.

Об'єднання Unity та Visual Studio дає мені все необхідне для створення захоплюючої та стійкої мобільної гри-раннера. Unity полегшить мені візуальну розробку, а Visual Studio забезпечує ефективне написання та тестування коду.

Впевнений, що ця комбінація інструментів дозволить мені реалізувати всі свої ідеї та створити гру, яка буде цікавою для гравців.

2.2. Загальна структура системи

У розділі "Загальна структура системи" описано основні компоненти системи та їх взаємозв'язки, які реалізовувались у проекті. Він дає загальне уявлення про те, як система організована та як її частини працюють разом для досягнення загальної мети.

Структура системи гри 3D ранера записана на архітектурному шаблоні “Entity Component System” (ECS) [9].

ECS (Entity Component System) - це архітектурний шаблон програмного забезпечення, який часто використовується в розробці відеоігор, але також може застосовуватися в інших областях, таких як симуляції та робототехніка.

Основні принципи ECS:

1. Сутності – це як окремі блоки, кожен з яких має унікальну мету. Наприклад, сутність може бути персонажем гравця, противником, предметом або перешкодою.

2. Компоненти – це як деталі, які надають сутності її властивості. Наприклад, компонент може зберігати інформацію про місцезнаходження сутності, швидкість, здоров'я, тип тощо.

3. Системи – це інструкції, які кажуть, що робити з різними частинами. Наприклад, система може керувати рухом сутностей, їх взаємодією, поведінкою противників тощо.

Переваги ECS:

1. Гнучкість. Можна легко додавати, видаляти та змінювати сутності, компоненти та системи, щоб налаштувати свою гру.

2. Масштабованість. ECS добре підходить для створення складних і динамічних ігор з багатьма сутностями.

3. Продуктивність. ECS може бути дуже продуктивним, що робить його ідеальним для створення ігор, які потребують високої швидкості.

В грі персонаж, предмет чи перешкода представлено сутністю. Компоненти використовуються для зберігання інформації про положення об’єкта, швидкість та інші характеристики сутностей. Системи використовуються для керування рухом сутностей, їх зіткненнями, поведінкою противників та генерацією світу.

У структурі системи гри можна виділити кілька ключових систем, наприклад як система фізики.

Система фізики (Physics System) – це потужний інструмент, який використовується для симуляції фізичних законів у 3D-світі [8]. Вона дозволяє об'єктам рухатися, взаємодіяти один з одним та реагувати на сили, такі як гравітація та зіткнення. Це робить гру більш динамічною, захоплюючою та реалістичною. Компонентами системи є:

– компонент колізій (Collision Component) - це компонент Unity, який використовується для визначення того, як об'єкт у вашому 3D-світі взаємодіє з іншими об'єктами [10]. Він визначає форму та розмір об'єкта, а також те, як він реагує на зіткнення. Як приклади використання компонентів у грі відноситься персонаж гравця. Для нього використовується компонент колізії, щоб визначити форму та розмір персонажа гравця, а також те, як він реагує на зіткнення зі стінами, рух по платформі. Предмети також використовуються як компоненти

колізій, щоб визначити форму та розмір предметів, а також те, як вони реагують на зіткнення з гравцем та іншими об'єктами.

– компонент фізичного руху (Physics Engine Component). В Unity компонент Physics Engine не є спеціальним вбудованим компонентом, який безпосередньо приєднується до об’єктів гри. Функціональність фізичного механізму обробляється через загальну фізичну систему. Однак є кілька компонентів, які працюють разом із системою фізики, щоб визначити, як об’єкти взаємодіють із симульованою фізикою у грі. Ключові аспекти, пов'язані з системою:

1. Компонент колайдер (Collider Components) – це компоненти визначають форму та розмір об'єкта для виявлення зіткнень. Вони приєднуються до ігрових об'єктів і визначають, як об'єкт взаємодіє з іншими об'єктами в імітації фізики.

2. Компонент жорсткого тіла (Rigidbody Component). Цей компонент визначає, як об'єкт реагує на сили та зіткнення в імітації фізики. Ви можете приєднувати його до ігрових об'єктів, щоб зробити їх фізично імітованими об'єктами. Компонент тригерів додається до об'єкта. У грі він додається до об’єкту перешкоди у сцені, коли персонаж взаємодіє з тригером гра припиняється. Тригери також можна використовувати для активації подій, виявлення об'єктів та створення цікавих ігрових механізмів.

Система відображення Unity - це потужний комплекс, який перетворює 3D-світ на захоплюючі візуальні образи на екрані. Вона тісно взаємодіє з багатьма іншими системами Unity, щоб створити цілісний ігровий досвід. Її робота включає:

1. Проектування 3D-світу: перетворення 3D-координат в 2D-координати на екрані, враховуючи перспективу, камеру та положення об'єктів;

2. Освітлення: застосування різних моделей освітлення (локальне, глобальне, запечені карти) до об'єктів для створення реалістичних ефектів.

До прикладу взаємодіє з системою введення, система отримує дані про дії користувача (натискання кнопок чи руху мишки) і передає їх системі відображення. Також система відображення використовує ці дані, щоб визначити, як реагувати на дії користувача візуально. До прикладу, коли гравець натискає клавішу що відповідає за “стрибок”, система відображає анімує стрибок персонажа.

Взаємодія з системою фізичного руху відбувається тоді, коли система розраховує рух об’єктів, грунтується вона на законах фізики.

Загалом, система відображення Unity - це хороший інструмент, він тісно взаємодіє з багатьма іншими системами, щоб створити захоплюючі та інтерактивні ігри. Розуміння цих взаємодій допомагає розробникам створювати більш динамічні та цікаві ігри.

Система введення (Input System) - цей інструмент дозволяє реагувати на дії користувача у грі. Вона замінює стару систему Input Manager та пропонує ряд переваг:

4. Гнучкість. Підтримує широкий спектр пристроїв введення, включаючи клавіатури, миші, геймпади, сенсорні екрани, віртуальні-контролери та багато іншого;

5. Простота використання. Інтуїтивно зрозумілий інтерфейс для створення та налаштування дій;

6. Потужність яка підтримує складні сценарії введення, такі як одночасне натискання кнопок, комбінації клавіш та контекстно-залежні дії.

Система введення Unity - це незамінний інструмент для створення інтерактивних та захоплюючих ігор. Її гнучкість, простота використання та потужність роблять її ідеальним вибором для розробників будь-якого рівня досвіду.

ECS (Entity Component System) та його системи забезпечують потужну та гнучку основу для розробки відеоігор. Архітектура пропонує ряд переваг, які роблять її ідеальним вибором для створення складних і захоплюючих ігор.

Гнучкість ECS дозволяє легко додавати, видаляти та змінювати компоненти ентитетів, що робить його дуже гнучким для реалізації різних ігрових механік. Масштабованість добре підходить для створення великих і складних ігор, оскільки він може ефективно обробляти велику кількість ентитетів та компонентів. Простота використання: ECS відносно простий у використанні та розумінні, що робить його доступним для розробників будь-якого рівня досвіду.

Системи ECS, такі як системи відображення, введення, фізичного руху забезпечують чітку структуру та організованість для розробки ігор. Кожна система відповідає за певний аспект ігрової логіки, що робить код більш модульним та легким для обслуговування. Завдяки своїм перевагам ECS стає все більше популярнішою у сфері розробки відеоігор, та відіграє важливу роль у цій галузі.

2.3. Структура проекту

Розділ "Структура проекту" в Unity присвячений важливості належної організації та структурування проекту. Це має ключове значення для ефективного управління проектом, адже добре організований проект дозволяє легко знаходити та керувати ресурсами. Також підтримання чіткості та логічності проекту полегшує розуміння його загальної організації та взаємозв'язків між його компонентами. Завдяки правильній організації структури розробники досягають спрощенню співпраці. Добре структурований проект полегшує співпрацю з іншими розробниками, оскільки всі мають чітке уявлення про те, де знаходяться ресурси та як вони пов'язані між собою.

Підвищення масштабованості, де добре організований проект легше масштабувати, додаючи нові функції та вміст, оскільки структура проекту вже готова до цього.

Unity пропонує ряд інструментів, які допомагають організувати проект.

Починаючи з папки проекту. Це головна папка вашого проекту, де зберігаються всі ваші ресурси. Переходячи до створення підпапок, щоб організувати свої ресурси за категоріями, наприклад, "Моделі", "Текстури", "Сценарії".

Ім’я файлів також потрібно правильно заповнювати. Використовуючи чіткі та описові імена файлів, щоб легко знаходити потрібні ресурси (рис. 1.5.).

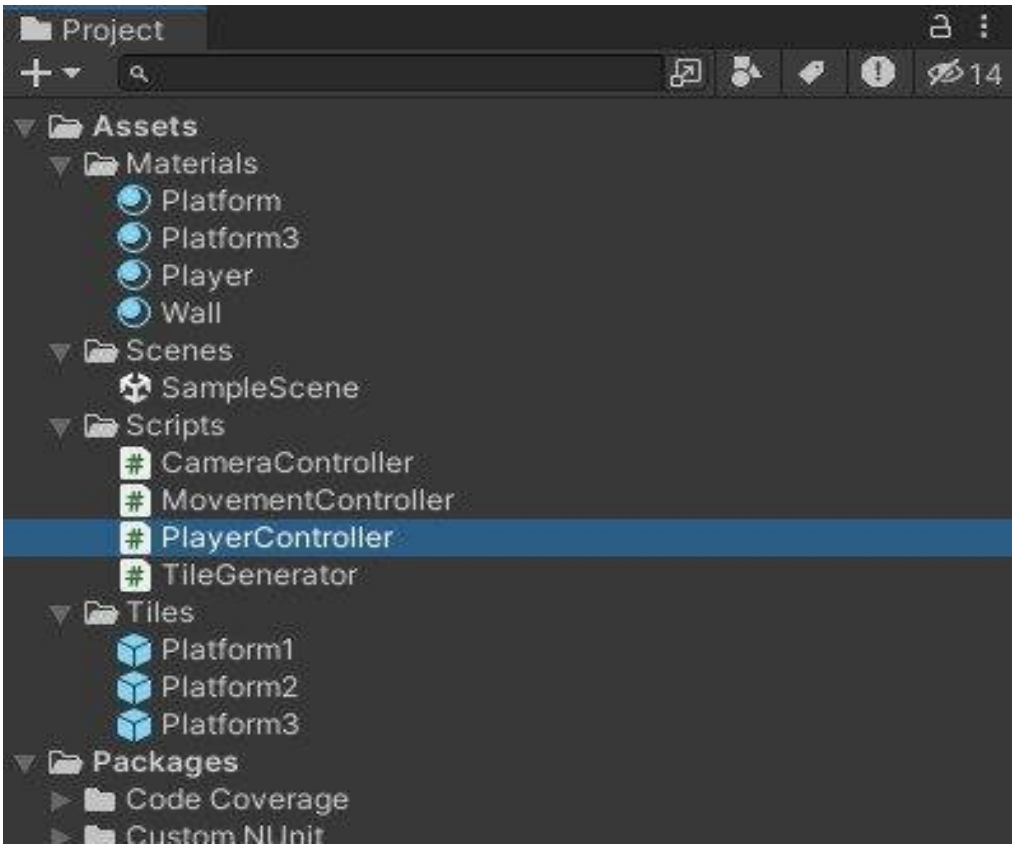


Рисунок 1.5 – Вигляд структури вікна Project

На рисунку зображено структуру проекту де використовується наступна структура папок:

– Assets. Це головна папка проекту, де зберігаються всі ресурси, такі як моделі, текстури, сценарії, аудіо та інші файли. У головній папці створюються підпапки, для того щоб організувати свої ресурси за категоріями. Наприклад, в проекті створено підпапки для моделей, текстур, сценаріїв, та інших типів ресурсів (рис. 1.6.).

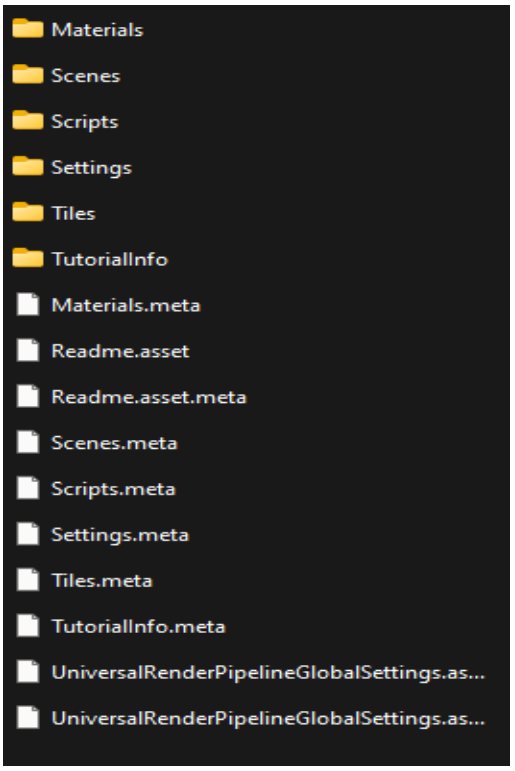


Рисунок 1.6 – Коренева папка Assets

– Materials: це папка, яка містить матеріали проекту. Матеріали визначають зовнішній вигляд моделей, використовуючи текстури, кольори та інші візуальні властивості;

– Tiles: це папка в якій містить платформи з об’єктами для гри по яким персонаж буде рухатися;

– Scenes: ця папка містить сцени проекту. Сцени - це окремі рівні або етапи гри;

– Scripts: папка в якій міститься сценарії проекту. Сценарії - це фрагменти коду, які керують поведінкою вашої гри (рис. 1.7).

– Code Coverage - це папка яка містить інформацію про покриття кодом проекту. Покриття кодом показує, яка частина вашого коду тестується;

– Packages - це папка яка містить пакети, які ви використовуєте у своєму проекті. Пакети - це надбудови, які додають нові функції та можливості до вашого проекту;

– Custom Nunit - це папка що містить власні тести Nunit, які ви написали. Nunit - це фреймворк для тестування одиниць, який використовується для перевірки вашого коду.

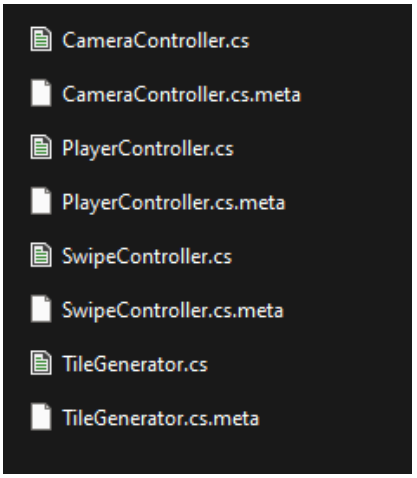


Рисунок 1.7 – Вміст каталогу Scripts

Структура проекту є ключовим фактором, який визначає успіх розробки в Unity. Вона впливає на те, як легко та ефективно ви будете працювати над своїм проектом, а також на те, як інші розробники зможуть зрозуміти та співпрацювати з вами. Дотримуючись рекомендацій, описаних вище, ви зможете створити добре організований проект, який буде:

– легким у навігації: ви зможете швидко знаходити потрібні ресурси та редагувати свій проект;

– зрозумілим: чітка структура допоможе вам краще зрозуміти свій проект та те, як всі його частини взаємодіють між собою;

– стійким до помилок: зменшується ризик помилок та дублювання роботи завдяки чіткій структурі;

– спів працездатним: іншим розробникам буде легше розуміти та працювати з вашим проектом;

– масштабованим: легше додавати нові функції та розширювати проект в майбутньому.

Поради щодо створення та підтримки хорошої структури проекту:

– плануйте заздалегідь: перш ніж розпочати розробку, подумайте про те, як ви будете організовувати свій проект. Це допоможе вам уникнути проблем згодом;

– будьте послідовні: використовуйте однакові принципи організації для всіх папок та елементів у вашому проекті;

– регулярно чистіть: видаляйте непотрібні ресурси та папки, щоб ваш проект не засмічувався;

– Використовуйте інструменти: unity пропонує різні інструменти, які можуть допомогти вам організувати ваш проект, такі як вікно Project та Hierarchy;

– не бійтеся змінювати: ваша структура проекту повинна розвиватися разом з вашим проектом. Не бійтеся змінювати її, якщо це необхідно.

Пам'ятайте, що час, витрачений на створення хорошої структури проекту, окупиться з лишком у довгостроковій перспективі. З добре організованим проектом ви будете працювати ефективніше, зрозуміліше та з меншою кількістю помилок.

Висновок до розділу 2

У ході дослідження було здійснено обґрунтований вибір засобів для реалізації проекту з розробки 3D-ранер на движку Unity.

Оптимальним вибором визначено двигун Unity, який володіє необхідними інструментами та можливостями для створення 3D-ігор, а також має широку спільноту розробників та багату документацію.

Розроблено загальну структуру системи, яка включає основні модулі та їх взаємозв'язки. Ця структура забезпечує чітку організацію проекту та сприяє його ефективній реалізації.

Створено детальну структуру проекту, яка описує всі його компоненти, підсистеми та елементи. Ця структура слугує дорожньою картою для розробки проекту.

РОЗДІЛ 3. СТРУКТУРА ПРОЕКТУ

3.1. Реалізація системи

Створення повнофункціональної відеогри у жанрі 3D ранер вимагає ретельної реалізації численних ігрових механік та систем, які забезпечують захопливий та динамічний ігровий процес. У цьому розділі детально описано технічні аспекти втілення ключових компонентів розробленої гри відповідно до поставлених вимог на етапі постановки задачі.

Насамперед, розглядається система переміщення ігрового персонажа, яка є однією з основних складових жанру ранер. Пояснюються методи автоматичного пересування персонажа вздовж заданої траєкторії, а також способи реалізації керування стрибками, зміною напрямку та взаємодії з перешкодами.

Наступним кроком є опис генерації рівнів з об’єктами-перешкодами на шляху персонажа. Висвітлюються алгоритми розташування платформ, керування їх поведінкою та система виявлення зіткнень з персонажем.

У кожному підрозділі наводяться детальні пояснення, коментарі до коду, що демонструє технічну реалізацію відповідних компонентів системи [3].

```
public class PlayerController : MonoBehaviour
{

    private CharacterController controller;
    private Vector3 dir;
    [SerializeField] private int speed;
    [SerializeField] private float jumpForce;
    [SerializeField] private float gravity;
    private int lineToMove = 1;
    public float lineDistance = 4;
    void Start()
    {
```

```
        controller = GetComponent <CharacterController>();
    }
    private void Update()
    {
        if (SwipeController.swipeRight)
        {
            if (lineToMove < 2)
                lineToMove++;
        }
        if (SwipeController.swipeLeft)
        {
            if(lineToMove > 0)
                lineToMove--;
        }
        if (SwipeController.swipeUp)
        {
            Jump();
        }
            }
    private void Jump()
    {
        dir.y = jumpForce;
    }
```

Цей код є реалізацією системи керування персонажем у грі [4], написаний на C# з використанням Unity:

Using директиви імпортують необхідні простори імен для роботи з основними класами Unity [11]. Клас PlayerController наслідується від MonoBehaviour і являє собою компонент, який можна прикріпити до ігрового об'єкта в Unity. Змінна controller зберігає посилання на компонент CharacterController, який відповідає за переміщення персонажа. dir - вектор, що визначає напрямок руху персонажа. Поля speed, jumpForce та gravity є серіалізованими полями, значення яких можна налаштувати в інспекторі Unity. Вони відповідають за швидкість руху, силу стрибка та прискорення гравітації. lineToMove - змінна, що визначає, на якій "лінії" рухається персонаж (0 - ліва, 1

- центр, 2 - права), а lineDistance задає відстань між цими лініями. У методі Start() отримується посилання на компонент CharacterController.

Метод Update() викликається кожен кадр і обробляє вхідні дані від гравця (жести свайпу) для керування персонажем. Залежно від свайпу, персонаж змінює лінію руху або виконує стрибок. Цикл Update() обчислює цільову позицію персонажа відповідно до обраної лінії руху та переміщує персонажа до цієї позиції з певною швидкістю. Метод Jump() змінює вертикальну складову вектора напрямку руху dir для імітації стрибка.

```
// Update is called once per frame
    void FixedUpdate()
    {
        dir.y += gravity * Time.fixedDeltaTime;
        dir.z = speed;
        controller.Move(dir * Time.fixedDeltaTime);
    }
```

Метод FixedUpdate() викликається з фіксованою частотою (за замовчуванням, 50 разів на секунду) і відповідає за оновлення положення персонажа відповідно до вектора напрямку руху dir та застосування гравітації.

Цей код забезпечує основну логіку керування персонажем у грі, включаючи рух вперед, зміну ліній руху та стрибки. Він використовує компонент CharacterController для переміщення персонажа та обробляє вхідні дані від гравця жестами свайпу за допомогою класу SwipeController.

Vector3 являє собою структуру даних у Unity, яка представляє собою вектор із трьома координатами (x, y, z) які використовується для представлення позицій та напрямків [5].

```
    Vector3 targetPosition = transform.position.z * transform.forward
+ transform.position.y * transform.up;
            if (lineToMove == 0)
                targetPosition += Vector3.left * lineDistance;
            else if (lineToMove == 2)
```

```
            targetPosition += Vector3.right * lineDistance;
        if (transform.position == targetPosition)
            return;
        Vector3 diff = targetPosition - transform.position;
        Vector3 moveDir = diff.normalized * 25 * Time.deltaTime;
        if (moveDir.sqrMagnitude < diff.sqrMagnitude)
            controller.Move(moveDir);
        else
            controller.Move(diff);
    }
    private void Jump()
    {
        dir.y = jumpForce;
    }
```

Vector3 dir; - змінна dir типу Vector3 зберігає напрямок руху персонажа, x та z визначають горизонтальний напрямок, а y - вертикальний для стрибків та гравітації.

Vector3 targetPosition = transform.position.z * transform.forward + transform.position.y * transform.up; - ця лінія обчислює цільову позицію персонажа, базуючись на його поточній позиції transform.position та напрямку transform.forward (вектор, що вказує напрямок, у який дивиться персонаж).

targetPosition += Vector3.left * lineDistance; та targetPosition += Vector3.right * lineDistance; - ці рядки змінюють цільову позицію персонажа вліво або вправо відповідно до обраної "лінії" руху (lineToMove).

Vector3 diff = targetPosition - transform.position; - обчислює вектор різниці між цільовою позицією та поточною позицією персонажа.

Vector3 moveDir = diff.normalized * 25 * Time.deltaTime; - створює вектор руху для переміщення персонажа в напрямку до цільової позиції з певною швидкістю (25 у цьому випадку). Controller.Move(moveDir); та controller.Move(diff); - ці виклики методу Move компонента CharacterController переміщують персонажа на вектор moveDir або diff відповідно.

Таким чином, Vector3 використовується для представлення позицій, напрямків та переміщень персонажа у тривимірному просторі. Ця структура даних є невід'ємною частиною роботи з графікою та геометрією у Unity та інших ігрових рушіях [6].

```
using UnityEngine;

public class TileGenerator : MonoBehaviour
{
    public GameObject[] tilePrefabs;
    private List<GameObject> activeTiles = new List<GameObject>();
    private float spawnPos = 0;
    private float tileLength = 100;

    [SerializeField] private Transform player;
    private int startTiles = 6;
    // Start is called before the first frame update
    void Start()
    {
        for (int i = 0; i < startTiles; i++)
        {
            SpawnTile(Random.Range(0, tilePrefabs.Length));
        }
    }
```

Цей код реалізує систему генерації рівня на льоту для гри. Він відповідає за створення та видалення окремих секцій рівня (Tile) під час гри.

Розглянемо його детально.

tilePrefabs - це масив ігрових об’єктів, які представляють різні варіанти секцій рівня (Tile). Вони можуть містять різні перешкоди activeTiles - список поточно активних секцій рівня в грі. SpawnPos - змінна, що зберігає позицію, де буде створена наступна секція рівня. tileLength - довжина кожної секції рівня. player - посилання на ігровий об’єкт персонажа, щоб відстежувати його позицію. startTiles - кількість секцій рівня, що генеруються на старті гри. У

методі Start() викликається SpawnTile певну кількість разів (startTiles), щоб згенерувати початкові секції рівня випадковим чином.

```
       void Update()
         {
                if (player.position.z - 60 > spawnPos - (startTiles *
tileLength))
             {
                 SpawnTile(Random.Range(0, tilePrefabs.Length));
                 DeleteTile();
         private void SpawnTile(int tileIndex)
         {
              GameObject nextTile = Instantiate(tilePrefabs[tileIndex],
transform.forward * spawnPos, transform.rotation);
             activeTiles.Add(nextTile);
             spawnPos += tileLength;
         }
         private void DeleteTile()
         {
             Destroy(activeTiles[0]);
             activeTiles.RemoveAt(0);
         }
```

У методі Update() відбувається перевірка позиції персонажа відносно вже згенерованих секцій рівня. Якщо персонаж наблизився до кінця поточних секцій, викликаються методи SpawnTile та DeleteTile для створення нової секції та видалення найстарішої [20].

Метод SpawnTile(int tileIndex) створює нову секцію рівня, використовуючи один із префабів з масиву tilePrefabs за вказаним індексом. Нова секція додається до списку activeTiles, а позиція спавну оновлюється. Метод DeleteTile() видаляє найстарішу секцію рівня зі списку activeTiles та знищує відповідний ігровий об’єкт у сцені.

Ця система генерації рівня на льоту забезпечує безперервний ігровий процес у 3D ранері, створюючи нові секції рівня з різними конфігураціями

перешкод та видаляючи старі секції позаду персонажа. Це дозволяє створювати нескінченні рівні з випадковими конфігураціями, не використовуючи попередньо створені рівні (рис. 1.8).

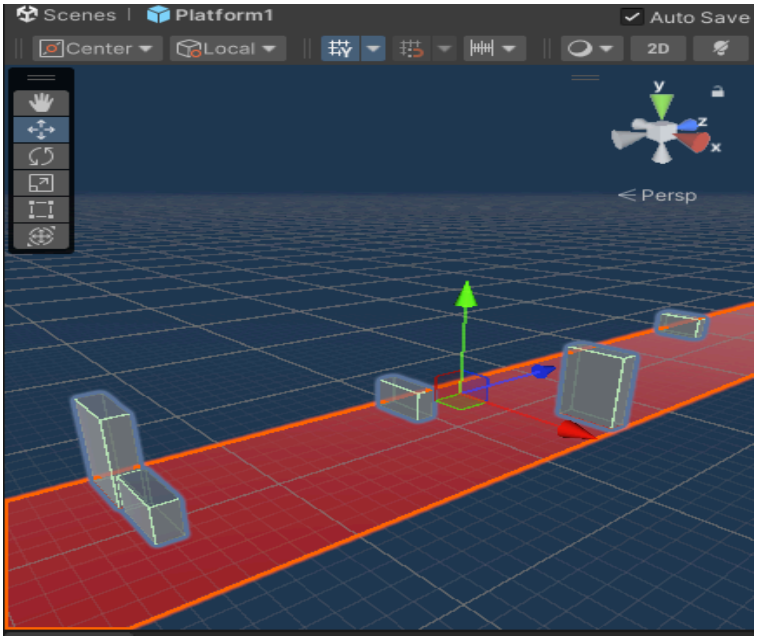


Рисунок 1.8 – Створення рівня-платформи з перешкодами

```
if (player.position.z - 60 > spawnPos - (startTiles * tileLength))
{
    SpawnTile(Random.Range(0, tilePrefabs.Length));
    DeleteTile();
}
```

Розглянемо цю частину коду детальніше: ця умовна конструкція перевіряє, чи досягла позиція персонажа певної точки, щоб згенерувати нову секцію рівня та видалити стару. Розберемо її по частинах:

Елемент player.position.z. Ця частина отримує координату Z (вглиб сцени) поточної позиції персонажа. У 3D ранерах зазвичай персонаж рухається вздовж осі Z. Число - 60 від координати Z персонажа віднімається 60. Це відстань, на якій нова секція рівня буде згенерована перед тим, як персонаж досягне кінця поточних секцій. Це дозволяє плавно переходити між секціями, не створюючи розривів. Елемент spawnPos являє собою змінну яка містить координату Z, де

була створена остання секція рівня, (startTiles * tileLength): Від змінної spawnPos віднімається добуток startTiles * tileLength. startTiles - це кількість початкових секцій рівня, які були згенеровані під час запуску гри. tileLength - це довжина кожної секції рівня. Таким чином, ця частина обчислює координату Z, де була створена перша секція рівня на старті.

Отже, умова порівнює координату Z персонажа (зміщену на 60 одиниць вперед) з координатою Z, де була створена перша секція рівня. Коли персонаж досягає цієї точки, це означає, що він наблизився до кінця поточних згенерованих секцій рівня, і потрібно створити нову секцію.

3.2. Тестування роботи системи

Після завершення етапу реалізації системи необхідно провести ретельне тестування для забезпечення коректної роботи всіх компонентів гри та виявлення потенційних помилок чи недоліків. Тестування є невід'ємною частиною процесу розробки програмного забезпечення, особливо в контексті ігрової індустрії, де високі вимоги до якості та зручності користувача є критичними.

У процесі тестування розробленої гри 3D ранер на двигуні Unity було застосовано комплексний підхід, що охоплював різні рівні та аспекти системи. Тестування проводилося як в автоматизованому режимі з використанням вбудованих інструментів Unity, так і шляхом ручної перевірки.

Завдяки ретельному та всебічному процесу тестування вдалося забезпечити високу якість розробленої гри 3D ранер, її стабільність та відповідність поставленим вимогам. Це є важливою передумовою для успішного випуску продукту та задоволення очікувань користувачів.

Відбулось функціональне тестування. За його результатами було виявлено баг де персонаж не може рухатись коли об’єкт (перешкода) розташований на одній з трьох ліній, а саме середній. Спроби виправити помилку через Колайдер платформи були марні (рис. 1.9) [3].

Помилку було виправленно за допомогою такої конструкції:

```
Vector3 diff = targetPosition - transform.position;
        Vector3 moveDir = diff.normalized * 25 * Time.deltaTime;
        if (moveDir.sqrMagnitude < diff.sqrMagnitude)
            controller.Move(moveDir);
        else
            controller.Move(diff);
```

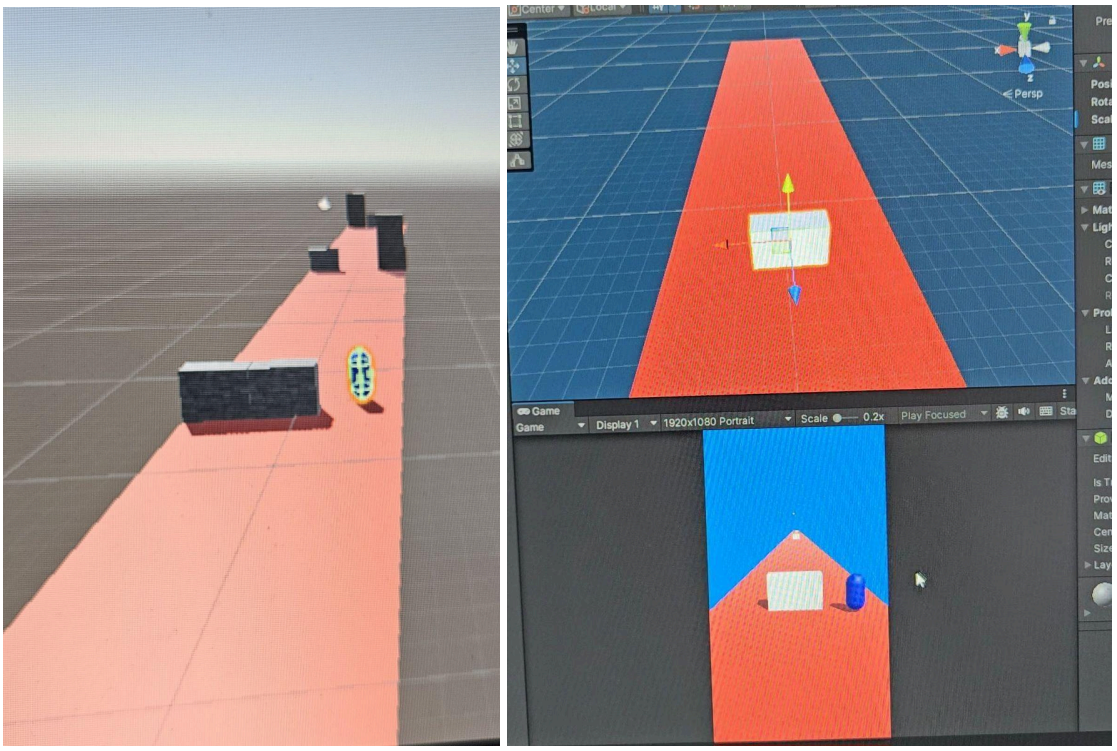


Рисунок 1.9 – Помилка через яку персонаж не може рухатись далі

Ця конструкція перевіряє, чи достатньо великий вектор moveDir, щоб досягти цільової позиції за один кадр. Якщо квадрат довжини moveDir (moveDir.sqrMagnitude) менший за квадрат довжини diff (diff.sqrMagnitude), це означає, що персонаж може досягти цільової позиції за один кадр. У такому випадку ми викликаємо метод controller.Move(moveDir), щоб переміститися на вектор moveDir.

Якщо умова не виконується (тобто moveDir більший або дорівнює diff), це означає, що персонаж не може досягти цільової позиції за один кадр. У цьому

випадку ми викликаємо controller.Move(diff), щоб переміститися на весь вектор diff відразу, що дозволить досягти цільової позиції за один крок.

Також плюсом додається те що цей підхід забезпечує плавний рух персонажа до позиції , якщо відстань до позиції достатньо велика. Якщо ж відстань мала, персонаж просто переміститься на цю відстань за один крок.

Також було проведене нефункціональне тестування. Гра була запущена на декількох різних пристроях для перевірки FPS. Тест гри було проведено як на комп’ютері так і на телефоні, адже гра підтримує кросплатформеність.

Характеристики комп’ютера на якому тестувалась гра:

- відеокарта: Nvidia Geforce RTX 4060Ti;
- процесор: AMD Ryzen 5 5500;
- оперативна пам’ять – 16 GB DDR4.

Характеристики смартфона на якому тестувалась гра:

- процесор: snapdragon 8 Gen2;
- оперативна пам’ять: 8GB;
- внутрішня пам’ять: 256GB.

Результат тестування пройшло успішно. Гра працює стабільно та без просадок у частоті кадрів. Частота кадрів на комп’ютері складала у 258, телефон показав відмітку у 156 кадрів.

Висновок до розділу 3

Завдяки ретельному та всебічному підходу до тестування вдалося забезпечити високу якість розробленого програмного продукту, його стабільність, продуктивність та відповідність поставленим вимогам, що є важливою передумовою для успішного випуску гри та позитивного досвіду користувачів. Результати проведеного тестування свідчать про готовність системи до випуску та впровадження.

Загалом, дане дослідження закладає фундамент для успішної реалізації проекту з розробки 3D-ранер на движку Unity

ВИСНОВОК

У рамках даної кваліфікаційної роботи було успішно досягнуто поставленої мети - розроблено повно-функціональну гру у жанрі 3D ранер на ігровому рушії Unity.

На початковому етапі було проведено ґрунтовний огляд предметної області відеоігор жанру "ранер", визначено його основні концепції та механіки, а також проаналізовано популярні існуючі рішення на ринку. Це дозволило сформулювати чітку постановку задачі з урахуванням сучасних тенденцій ігрового дизайну та очікувань користувачів.

Під час розробки структури системи було обґрунтовано вибір ігрового двигуна Unity як основного засобу реалізації, а також визначено загальну архітектуру та організацію проекту. Це забезпечило ефективну інтеграцію та взаємодію різних компонентів системи.

Процес реалізації охоплював втілення ключових механік гри, таких як система переміщення персонажа, генерація перешкод, система бонусів. Для забезпечення якості та стабільності системи було проведено тестування.

Результатом роботи став готовий програмний продукт, що задовольняє поставлені вимоги та забезпечує захопливий ігровий досвід у жанрі 3D ранер.

Проведена робота продемонструвала ефективність застосування сучасних технологій та підходів до розробки відеоігор, а також необхідність ретельного проектування, реалізації та тестування системи. Одержані результати можуть бути використані як основа для подальшого вдосконалення та розширення функціоналу розробленої гри або створення нових ігрових проектів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація бібліотеки MonoBehaviour. Веб сайт. URL: https://docs.unity3d.com/Manual/class-MonoBehaviour.html (дата звернення: 15.04.2024)

2. Документація бібліотеки MonoBehaviour. Веб сайт. URL: https://docs.unity3d.com/Manual/class-MonoBehaviour.html (дата звернення: 18.04.2024)

3. Документація ігрового двигуна Unity. Веб сайт. URL: https://docs.unity3d.com/Manual/UnityOverview.html (дата звернення: 26.03.2024)

4. Документація мовою програмування C#. Веб сайт. URL: https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/overview (дата звернення: 20.03.2024)

5. Документація по властивості Introduction to collision. Веб сайт. URL: https://docs.unity3d.com/Manual/CollidersOverview.html (дата звернення: 01.04.2024)

6. Документація по налагодженню коду C# в Unity. Веб сайт. URL: https://docs.unity3d.com/Manual/ManagedCodeDebugging.html (дата звернення: 25.03.2024)

7. Документація по Visual Studio. Веб сайт. URL: https://learn.microsoft.com/ru-ru/visualstudio/windows/?view=vs-2022 (дата звернення: 01.04.2024)

8. Форум ігрового двигуна Unity. Веб сайт. URL: https://forum.unity.com/forums/physics.78/ (дата звернення: 05.05.2024)

9. Taylor & Frankis, Alex Okita, Learning C# Programming with Unity 3D, second edition, 2019, 690 p.

10. Unity 3D Game Development by Example: Learn to Build High-Performance 3D Games with C#, 2017

11. Unity in Action: Second Edition by Joseph Labonte, 2020, 400p.

12. Unity - Scripting API: Collision. Unity - Manual: Unity User Manual 2022.3 (LTS). Веб сайт. URL: https://docs.unity3d.com/ScriptReference/Collision.html (дата звернення: 05.05.2024)

13. Вікіпедія по грі Lara Croft: Relic Run. Веб сайт. URL: https://en.wikipedia.org/wiki/Lara_Croft:_Relic_Run (дата звернення: 30.04.2024)

14. Вікіпедія по грі Sonic Dash. Веб сайт. URL: https://en.wikipedia.org/wiki/Sonic_Dash (дата звернення: 30.04.2024)

15. Вікіпедія по грі Subway Surfers. Веб сайт. URL: https://uk.wikipedia.org/wiki/Subway_Surfers (дата звернення: 30.04.2024)

16. Вікіпедія по грі Temple Run. Веб сайт. URL: https://en.wikipedia.org/wiki/Temple_Run (дата звернення: 30.04.2024)

17. Ігровий двигун Unity. Веб сайт. URL: https://unity.com/download (дата звернення: 25.03.2024)

18. Редактор коду Visual Studio. Веб сайт. URL: https://visualstudio.microsoft.com/ (дата звернення: 25.03.2024)

19. Статті по відеоіграм. Веб сайт. URL: https://coworkingplatforma.com/ua/blog/videoigry-eto-lucsaa-socialnaa-set/ (дата звернення: 29.04.2024)

20. Статті по відеоіграм. Веб сайт. URL: https://nachasi.com/videogames/2021/10/01/chomu-nam-podobayetsya-graty-u-videoigry/ (дата звернення: 29.04.2024)

метадані

Заголовок

Розробка гри 3D ранер на движку Unity

Автор

Вербіцький Віталій Науковий керівник / Експерт

підрозділ

King Danylo University

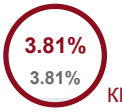
Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

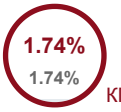
Заміна букв	ß	3
Інтервали	A→	0
Мікропробіли	·	0
Білі знаки	ß	0
Парафрази (SmartMarks)	a	23

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1



КП 2

25
Довжина фрази для коефіцієнта подібності 2

9075
Кількість слів

68519
Кількість символів

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)	
1	http://repository.ukd.edu.ua/bitstream/handle/123456789/395/%D0%94%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC%D0%BD%D0%B0%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B0%20%D0%A1%D1%82%D0%B5%D0%BF%D0%B0%D0%BD%D1%8E%D0%BA.pdf?sequence=1	81	0.89 %
2	http://repository.ukd.edu.ua/bitstream/handle/123456789/395/%D0%94%D0%B8%D0%BF%D0%BB%D0%BE%D0%BC%D0%BD%D0%B0%20%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B0%20%D0%A1%D1%82%D0%B5%D0%BF%D0%B0%D0%BD%D1%8E%D0%BA.pdf?sequence=1	41	0.45 %