ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА

Факультет суспільних та прикладних наук

Кафедра інформаційних технологій

на правах рукопису

Шиньквар Максим Васильович

УДК 004.4

Розробка бета версії відеогри в жанрі RPG

Спеціальність 121 – «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття кваліфікації бакалавр

Нормоконтроль

_____Стисло О.В.
(підпис, дата, розшифрування підпису)

Студент

_____Шиньквар М.В.
(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. Ващишак С.П.
(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

Шиньквар Максим Васильович

(прізвище, ім’я, по батькові)

1. Тема кваліфікаційної роботи:

Розробка бета версії відеогри в жанрі RPG

керівник роботи:

Ващишак Сергій Петрович, к.т.н., доцент

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

2. Термін подання студентом роботи 05.06.2024

3. Вихідні дані роботи: мова програмування C# та ігровий рушій Unity

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

1. Опис об’єкту та аналіз аналогів
2. Дослідження ігрових рушіїв та опис доступних мов програмування
3. Розробка функціоналу та модель гри
4. Програмна реалізація гри

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.	Дослідження існуючих аналогів	20.03.2024	Виконано
2.	Проектування архітектури та створення дизайну додатку	27.03.2024	Виконано
3.	Програмна реалізація додатку	03.04.2024	Виконано
4.	Розгортання та тестування додатку	16.04.2024	Виконано
5.	Формування висновків	30.04.2024	Виконано
6.	Оформлення пояснювальної записки	06.05.2024	Виконано
7.	Оформлення графічного матеріалу та підготовка до захисту роботи	21.05.2024	Виконано

Студент _____ Шиньквар М.В.

(підпис) (прізвище та ініціали)

Керівник роботи _____ Ващишак С.П.

(підпис) (прізвище та ініціали)

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

Сторінка	Опис графічного матеріалу	Сторінка	Опис графічного матеріалу
17	Логотип гри "Gothic"	35	Вайрфрейм вікна з інвентарем
18	Скріншоти з гри “Gothic” на офіційній сторінці в Steam	36	Вайрфрейм вікна з навичками та магією
19	Логотип гри "Elex" що була на коробці	38	Початкове вікно програми Unity Hub
20	Скрішнот гри “Elex” з офіційної сторінки гри в стімі	39	вікно вибору типу гри
22	Обкладинка гри “Відьмак 2: Вбивці королів”	39	сцена ігрового середовища
23	Мирне рішення в кінці гри “Відьмак 2”	47	додавання анімацій
34	Вайрфрейм графічного інтерфейсу		

АНОТАЦІЯ

Кваліфікаційна робота присвячена вивченню, аналізу та розробці відеоігор на рушії “Unity”, а саме 3D гри в жанрі RPG застосовуючи мову програмування C# та безкоштовні асети з Unity Assets Store.

Перший розділ присвячений опису та історії відео ігрової індустрії, жанру RPG та його історії. Також було проаналізовано конкурентів та виявлено переваги та недоліки конкурентів.

В другому розділі описується рушій, його аналоги, мови програмування, візуальний дизайн та архітектура гри, технології та інструменти, використані під час розробки.

В третьому розділі розповідається про розробку відеогри та описані всі пройдені кроки за період роботи.

В результаті було створено бета версію відеогри, що в подальшому буде доопрацьована. В кінцевому результаті гра буде спроможна задовольнити потреби цільової аудиторії, і в подальшому буде розширюватись, тестуватись і зростати, до подальшої публікації.

КЛЮЧОВІ СЛОВА: UNITY, RPG, C#, UNITY ASSETS STORE, АСЕТ, РУШІЙ, БЕТА ВЕРСІЯ

SUMMARY

Qualification work is dedicated to the study, analysis, and development of video games using the "Unity" engine, specifically a 3D RPG game employing the C# programming language and free assets from the Unity Assets Store.

The first chapter is devoted to the description and history of the video game industry, the RPG genre, and its history. Competitors were also analyzed, and their advantages and disadvantages were identified.

The second chapter describes the engine, its analogs, programming languages, visual design and game architecture, technologies, and tools used during development.

The third chapter covers the development of the video game and describes all the steps taken during the work period.

As a result, a beta version of the video game was created, which will be further refined. Ultimately, the game will be able to meet the needs of the target audience and will be expanded, tested, and grown further until its eventual release.

KEYWORDS: UNITY, RPG, C#, UNITY ASSETS STORE, ASSET, ENGINE, BETA VERSION

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

RPG (Role Play Games) – це жанр відеоігор, де основна частина ігрового процесу полягає в управлінні персонажем чи групою персонажів, які досліджують ігровий світ, виконують різноманітні завдання та розвиваються, слідуючи сюжету;

Відеоігри – програмне забезпечення, створене в розважальних цілях;

DnD (Dungeon and Dragons) – настільна рольова гра в стилі фентезі, за часом видання перша рольова гра у світі;

Геймер (Від англійського слова gamer) – це людина, яка вільний час проводить за комп'ютерними або консольними іграми;

Квест (Від англійського слова quest) – це завдання, яке керований гравцем персонаж, чи група персонажів може виконати за винагороду. Винагорода може включати підвищення досвіду персонажа задля вивчення нових навичок і вмінь, здобич або скарб, ігрова валюта, доступ до нових місць або регіонів, чи ж будь-яка комбінація переліченого;

Ком’юніті (від англ. Community) – група людей, яких об’єднують спільні цілі, цінності та інтереси;

Баг – помилка, вада або дефект в комп'ютерній програмі або системі, що викликає в ній неправильний або неочікуваний результат чи неочікувану поведінку;

Сеттинг – це сукупність різнопланових елементів, які однозначно ідентифікують світ, де відбуваються події певного художнього твору, відеогри або настільної рольової гри;

Катсцена – відеоролик в якому гравець не приймає участі і просто спостерігає за всім зі сторони;

UE – Unreal Engine. ігровий рушій, що розробляється та підтримується компанією Epic Games;

PSP - PlayStation Portable портативна гральна консоль виробництва Sony Interactive Entertainment;

Поінти (Від англійського слова points) – це діюча кількісна цінність, яка дається за якусь одиничну дію або за комбінацію дій;

Хотбар (Від англійського слова hotbar) – накладка у вигляді стрічки на екрані в іграх, часто знизу - що містить кнопки доступу до речей з інвентаря (майна), або зброї, чи дій;

Крафт (Від англійського слова craft) – створення внутрішньо ігрових предметів.

Вайрфрейм (Від англійського слова wireframe) – це грубий малюнок структури товару. Він дає змогу швидко зафіксувати ідею того, як усе буде влаштовано, не потребуючи багато часу на замальовку.

Джетпак (Від англійського слова Jetpack) – Реактивний ранець.

ВСТУП

Актуальність теми: відеоігри завжди мали своїх шанувальників, а ринок ігор є одним з найпопулярніших у світі. Індустрія сильно зросла після початку пандемії Covid-19. Основними причинами такого зростання є те що люди повинні були сидіти вдома і в пошуку різних розваг люди почали грати в відеоігри. Станом на 2019 рік глобальний прибуток становив: 152.1 млрд $, а прибуток виключно з України 203 млн $. Станом на 2023 рік глобальний прибуток: 211.44 млрд $, а по Україні: 338 млн $.

Відеоігри утримують свою актуальність тому що вони відповідають різним забаганкам людей. Завдяки іграм люди мають можливість скоротати свій вільний час, деякі ігри дають можливість навчатись, спілкуватись, змагатись з людьми, а деякі ігри можуть замінити книгу або фільм.

Саме жанр RPG зародився після успіху настільної гри DnD. Перші представники цього жанру, такі як "Wizardry" та "Ultima", відобразили основні концепції DnD у віртуальному середовищі, дозволяючи гравцям вибирати персонажів, розвивати їх та відчувати себе частиною фантастичного світу.

З розвитком технологій та з'явленням більш потужних комп'ютерів і консолей, жанр RPG почав набирати обертів. "Final Fantasy" від Square Enix та "The Elder Scrolls" від Bethesda Softworks стали відомими іграми, які привнесли нові ідеї та механіки у світ RPG.

Мета і завдання дослідження: розробити комп’ютерну відеогру в жанрі RPG, використовуючи обраний рушій та обрану мову програмування.

Для досягнення поставленої мети, потрібно вирішити наступні завдання:

- проаналізувати аналоги гри;
- проаналізувати методи виконання завдання;
- проаналізувати обраний рушій та його аналоги;
- розробити дизайн;
- продумати механіки та сюжет;

– протестувати гру.

Об’єкт дослідження: процес створення відеогри в жанрі RPG з використанням ігрового рушія Unity.

Предмет дослідження: інструменти розробки відеогри, геймплей та механіки гри, графічний дизайн та візуальний стиль, рушій "Unity".

Методи дослідження: для досягнення бажаного результату та виконання завдань дослідження будуть використані наступні методи:

– аналітичне дослідження;

– моделювання;

– тестування.

Практичне значення одержаних результатів. Результатом даної кваліфікаційної роботи буде бета версія гри, яка в майбутньому може бути доопрацьована для випуску її на онлайн платформи в комерційних цілях.

Апробація результатів дослідження. Матеріали кваліфікаційної роботи були представлені на XI науковій конференції “Студентські наукові дискусії поза форматом”, яка відбулася 11 квітня 2024 року в Університеті Короля Данила.

Структура. Розділи – 3. Обсяг основної частини – 43 сторінок. Список використаних джерел – 20.

РОЗДІЛ 1. ОПИС АНАЛОГІВ ВІДЕОІГОР В ЖАНРІ RPG

1.1. Відеоігри. Історія та актуальність

Геймінг існує вже десятиліттями і змінювався разом з розвитком технологій. Як системи ставали складнішими, ігри ставали більш глибокими і чутливими до можливостей гравців

Вже з 1950-х років вчені-комп'ютерники почали розробляти прості ігри як симуляції (ще до появи Atari Pong). Деякі з найраніших ігор включали Nimrod (1951), Draughts Program Джека Стрейчі (1951), Крестики-нолики (1951) і Tennis for Two (1958).

Коли з'явилися комп'ютери з вищою продуктивністю, такі як IBM 1560, ігри, такі як Moon Landing, 3D Tic-Tac-Toe і Spacewar!, стали популярнішими. Однак ці ігри зазвичай були доступні лише в провідних навчальних закладах, наприклад, MIT. Геймінг не отримав значної популярності серед широкого загалу до 1970-х і 80-х років.

Історія геймінгу справді почалася, коли Ральф Баєр винайшов ідею розважального пристрою, який можна було підключити до телевізійного монітора. "Коричнева скринька" Ральфа Баєра була відеоігровою консоллю, яка могла грати у настільний теніс.

Баєр очолив команду в Sanders Associates, щоб побудувати першу комерційну домашню консоль, Magnavox Odyssey, у 1972 році. Гравці використовували пластикові насадки на своїх телевізорах, щоб показувати візуальні елементи, які підтримували правила гри. Вони використовували контролери, щоб рухати крапку і вести свої власні рахунки.

У 1972 році був випущений Pong від Atari і став однією з перших аркадних відеоігор. Галузь "м'яч і ракетка" застоювалася, але породжувала початок ігор та ранніх хітів, включаючи:

- Gran Trak 10 (1974);

- Tank (1974);
- Wheels (1975);
- Gun Fight (1975);
- Sea Wolf (1976).

Перше покоління домашніх консолей (1975-1977) і ігор на головних комп'ютерах (1971-1979) включало Atari VCS (1977) і Odyssey 2 (1978). Це був початок історії ігрових консолей, і вони працювали на мовах програмування BASIC і C [].

У кінці 70-х та на початку 80-х років сталося справжнє вибухове зростання розважальних ігор на монетах, що спричинило значний розвиток відеоігор. Діти вибрали аркади заради ігор, які вони не могли собі дозволити вдома. Під час Золотої ери геймінгу, яка акцентувала увагу на аркадних відеоіграх, галузь геймінгу в США зростала з $308 млн (1978 рік) до $968 млн (1979 рік), а потім до $2.8 млрд (1980 рік).

Нова хвиля аркадних ігор у цей час також акцентувалася на альтернативних персонажах та механіках, починаючи з Pac-Man (1980). Ігри з персонажами набирали популярності, з появою Ms. Pac-Man (1982), Donkey Kong (1981) і Q*bert (1982).

Зародження аркадного геймінгу викликало друге покоління домашніх консолей (1976-1982), які дозволяли зберігати дані гри на картах ROM. Популярними домашніми консолями стали Intellivision (1979) і популярні ігри для консолей, такі як River Raid (1982) і Pitfall! (1982). Початок розвитку комп'ютерних ігор в кінці 1970-х років дозволив їм стрімко розвиватися в 1980-х роках для великого вибуху в комп'ютерному геймінгу.

Домашні комп'ютери почали набирати обертів наприкінці 70-х і стали набагато більш популярними в 80-х роках. Друга хвиля ігрових комп'ютерів з'явилася після популярності Apple II та Commodore PET.

Комп'ютери, такі як Commodore VIC-20, Sinclair ZX 81 та NEC PC-800, допомогли каталізувати популярність особистих домашніх комп'ютерів та ринку

ігрових ПК. У початковому етапі комп'ютерні ігри охоплювали все від текстових пригод до азартних ігор, таких як блекджек.

У 1979 році був випущений Microvision як перша ручна система зі змінними картриджами, що призвело до випуску лінійки ручних електронних ігор Nintendo Game & Watch.

З 1982 по 1985 рік в США сталася велика рецесія в індустрії відеоігор. З великою кількістю ігор та консолей на ринку, він пересихав, і увага звернулася на особисті комп'ютери. Відеоігровий крах 1983 року призвів до популярності японської ігрової індустрії для компаній, таких як Nintendo та Sega.

1990-ті роки підготували ґрунт для подальших інновацій у відеоіграх, включаючи перехід графіки від растрової до 3D. Це призвело до появи популярних жанрів, таких як FPS, стратегії в реальному часі та MMO-ігри. Популярність аркадних ігор зменшилася через перенесення індустрії відеоігор у більш широкі домашні розважальні варіанти.

У цей час все більше наростала тривожність щодо насильственних відеоігор, таких як Mortal Kombat, Night Trap та Doom, що викликали стурбованість серед широкого загалу. Люди боялися, що насильствені ігри зроблять дітей байдужими та схильними до насильства в реальному житті після того, як вони часто бачили його на екрані.

PC-геймінг продовжував зростати в популярності завдяки цим покращенням у графіці 3D. Відомі жанри, такі як FPS, стратегії в реальному часі та MMO-ігри, включали в себе:

- Quake (що спричинило народження кіберспорту та FPS-ігор)
- Resident Evil (народження жахів виживання)
- SimCity (ігри-симулятори)
- Gun Fight (1975)
- Dune II (стратегії в реальному часі)

Додаткові інновації у 2000-х роках призвели до більш сильної індустрії відеоігор. Геймінг став високо конкурентним ринком для розробників, виробників апаратного забезпечення та створювачів портативних ігрових

систем. Стало популярним користувачам робити модифікації гри, або "моди", що породило відомі ігри, такі як Counter Strike (модифікація гри Half-Life), вже з 1999 року.

Сьоме покоління відкрилося ручними консолями, такими як PSP і Nintendo DS. PSP привертала досвідчених гравців, тоді як DS залучила випадкових гравців з випусками, такими як Nintendogs і Brain Age.

У консольному геймінгу Xbox 360 вийшла в листопаді 2005 року, а PS3 - у 2006 році. Обидві компанії додали онлайн-геймінг та платформи продажів разом із своїми консолями, головним чином Xbox Live і PlayStation Network. Це збільшило можливості додатків та онлайн-геймінгу нового покоління консолей для ще ширших ігрових можливостей.

Nintendo також здобула домінування на ринку домашніх відеоігор з випуском Nintendo Wii, яка пропонувала Wii Sports і Mario Kart Wii. У Wii були нижчі технічні характеристики, але його керування на основі рухів збільшило зацікавленість нетрадиційних гравців. Нижча цінова категорія зробила його популярною консоллю серед багатьох сімей для сімейного геймінгу.

Восьме покоління консолей розпочалося з випуску Nintendo 3DS і PS Vita в 2011 році. Wii U також був випущений у 2011 році і представлений як спадкоємець Wii.

Випуск PlayStation 4 і Xbox One у 2013 році призвів до того, що консолі зараз домінують на ринку домашніх відеоігор. Ексклюзивні ігри для консолей, такі як Uncharted 4, God of War, Spider-Man, Gears of War 5 та Forza Horizon 4, забезпечили успіх як для PS4, так і для Xbox One, підтримуючи консольну війну протягом всього покоління.

Останньою консоллю восьмого покоління була гібридна платформа під назвою Nintendo Switch, яка включала керування на основі рухів Wii і переносимість. Найпопулярніші ігри для Switch включають The Legend of Zelda: Breath of the Wild, Super Smash Bros, Animal Crossing: New Horizons та Mario Kart 8.

Галузь комп'ютерних ігор постійно розвивається завдяки неймовірній творчості розробників. Вони неустанно працюють над створенням новаторських ігрових механік, історій та світів, що сприяє зростанню популярності та актуальності ігор серед гравців.

Комп'ютерні ігри продовжують залишатися актуальною та важливою частиною сучасної культури, привертаючи увагу мільйонів гравців та відображаючи найсвіжіші технологічні та культурні тенденції. Їх вплив на наше суспільство та спосіб розваг неперебoрний, а їхній потенціал для творчості та інновацій невичерпний.

1.2. Жанр RPG. Опис та популярність в наш час

Багато RPG встановлені у фентезі чи науково-фантастичних середовищах. Серед найраніших і найпопулярніших RPG є Dungeons and Dragons, BattleTech та Star Wars Galaxies. Більшість RPG граються онлайн або цифрово, що робить їх частиною жанру відеоігор.

У RPG зазвичай є сюжет і центральне завдання, яке гравці, мають завершити, приймаючи роль вигаданого персонажа у вигаданому світі. Деякі RPG також включають додаткові побічні завдання, які можуть бути не обов'язковими. Ці побічні завдання дозволяють гравцям отримувати більше досвіду в грі та покращувати характеристики та навички свого персонажа.

Більшість сучасних RPG зазвичай містять один чи декілька з цих елементів:

– рівні, що дозволяють гравцеві покращувати персонажа та заробляти більше очок по мірі прогресу;

– статистика, яка відображає кількість очок, які гравець заробив, та його позицію у порівнянні з іншими гравцями;

– бойова система, що дозволяє гравцям обирати свої дії та навички стратегічно;

– екіпіровка, така як броня та зброя, яка робить персонажів сильнішими;

– елементи взаємодії, що дозволяють гравцям взаємодіяти з іншими гравцями та ігровим оточенням, такі як пастки, двері тошо;

– класи персонажів, які кожен гравець може обирати, такі як воїн, чарівник, розбійник тощо.

У багатьох RPG є учасник, відомий як "ведучий гри", який зазвичай діє як розповідач і тримач правил. У електронних RPG, на відміну від ручних RPG, роль ведучого гри автоматизована. Гра або комп'ютер вибирає дії неігрових персонажів та ворогів.

У комп'ютерних відеоіграх RPG, особливо в онлайн іграх, існує можливість взаємодії з іншими гравцями з усього світу через Інтернет. Це дає можливість гравцям спілкуватися, обмінюватися досвідом, а також відчувати соціальну взаємодію, подібну до тих, що можна знайти у реальному житті.

Багато комп'ютерних RPG також мають можливість взаємодії з ігровим середовищем, що реагує на дії гравців. Наприклад, гравець може взаємодіяти з різними об'єктами в грі, активувати пастки, розглядати різні розміщення та вирішувати різні головоломки.

Однією з переваг комп'ютерних RPG є можливість впровадження складних систем бойових механік та аналізу даних, що може зробити гру більш цікавою та різноманітною для гравців. Крім того, завдяки великим обсягам даних, комп'ютерні RPG можуть пропонувати більш розгалужені історії та більш складні світи, які гравці можуть досліджувати.

В RPG також часто використовуються різні механіки прогресування, такі як системи досягнень, винагород за виконання завдань або розвиток персонажа шляхом збирання досвіду. Ці елементи стимулюють гравців до подальшого вивчення світу гри і виконання різноманітних завдань.

Більшість комп'ютерних RPG також пропонують можливість налаштування персонажа, дозволяючи гравцям вибирати різні вміння, характеристики та спеціалізації в залежності від їх геймплейних уподобань.

Крім того, деякі комп'ютерні RPG включають різні режими гри, такі як режими мультиплеєра, кооперативні режими або навіть режими будівництва, які дозволяють гравцям спільно створювати або взаємодіяти з ігровим світом.

1.3. Аналіз аналогів відеоігор в жанрі RPG

Gothic (рис. 1.1) – популярна серія ігор розроблена компанією Piranha Bytes, перша частина якої була випущена в 2002 році, та незважаючи на незвичне керування вона набула своєї популярності. Керування в цій грі розроблялось під консолі тому в комп’ютерній версії воно получилось геть зовсім не зручним.

Рисунок 1.1 – логотип гри "Gothic"

Війна вибухнула по всьому королівству Міртана. Орки вторглися на територію людей, і король країни потребував великої кількості магічної руди, щоб викувати достатньо зброї, щоб його армія могла протистояти цій загрозі. Кожен, хто порушує закон, незалежно від його величини, засуджується до відбування покарання у гігантській виправній колонії Хорініс, де добувають таку необхідну руду.

Весь регіон, який отримав назву "Колонія", оточений магічним бар'єром, сферою діаметром два кілометри, що відділяє колонію від зовнішнього світу. Бар'єр можна перейти ззовні всередину, але як тільки потрапиш всередину,

ніхто не може втекти. Бар'єр був двостороннім мечем – незабаром ув'язнені скористались можливістю і розпочали повстання (рис. 1.2). Колонія розділилася на три фракції які бились між собою, і король був змушений вести переговори за руду, а не просто вимагати її.

Рисунок 1.2 – скріншоти з гри “Gothic” на офіційній сторінці в Steam

В Готиці мав бути дуже наповнений, нехай і маленький, відкритий світ, з більшою кількістю всяких захованих предметів і локацій наповнених різною фауною, але на жаль до фінальної версії гри дійшло не все з запланованого, було вирізано багато другорядних квестів та ігрових механік. Згодом ком’юніті розробили різні неофіційні оновлення які повертають вирізаний контент, та виправляють різні баги.

Недоліки полягають у наступному:

- незрозуміле керування;
- велика кількість вирізаного контенту;
- некритичні баги, які мають невеликий вплив на геймплей;
- в грі без неофіційних оновлень, пустий відкритий світ.

Переваги :

- приємний звуковий супровід;

- бойова система;
- сюжет;
- прописані неігрові персонажі;
- цікава система прокачки;
- живий ігровий світ.

ELEX (рис. 1.3) – рольова відеогра в жанрі Action/RPG, розроблена студією Piranha Bytes та видана THQ Nordic. Гра вийшла на платформах Microsoft Windows, Xbox One і PlayStation 4 17 жовтня 2017. Сеттинг гри еклектичний: розробники поєднали елементи середньовічного фентезі, постапокаліпсису та футуристичного кіберпанку.

У світі гри Piranha Bytes, гравець вперше зустрічається з головним героєм на ім'я Джакс, який має власну передісторію. Джакс відправляється досліджувати світ, зустрічаючи різних персонажів, які доручають йому квести, боротьбу з ворогами та виконання завдань. Гравець може використовувати як холодну, так і вогнепальну зброю, а також магію. В грі є два типи ближнього бою: швидкий, але слабкий, та повільний, але потужний. Джакс також може ухилятися від атак і блокувати їх, витрачаючи витривалість та здоров'я.

Рисунок 1.3 – логотип гри "Elex" що була на коробці

Світ гри розділений на великі регіони з різними умовами і наповнений різноманітними жителями. Гравець може користуватися джетпаком для коротких перельотів та телепортами, розташованими по всьому світу(рис. 1.4).

Джакс заробляє досвід за знищення ворогів та виконання квестів, який використовується для розвитку характеристик та здібностей персонажа. Він також може навчатися спеціальних умінь від "учителів" за певну плату. Предмети збираються в інвентарі, які можна використовувати або продавати.

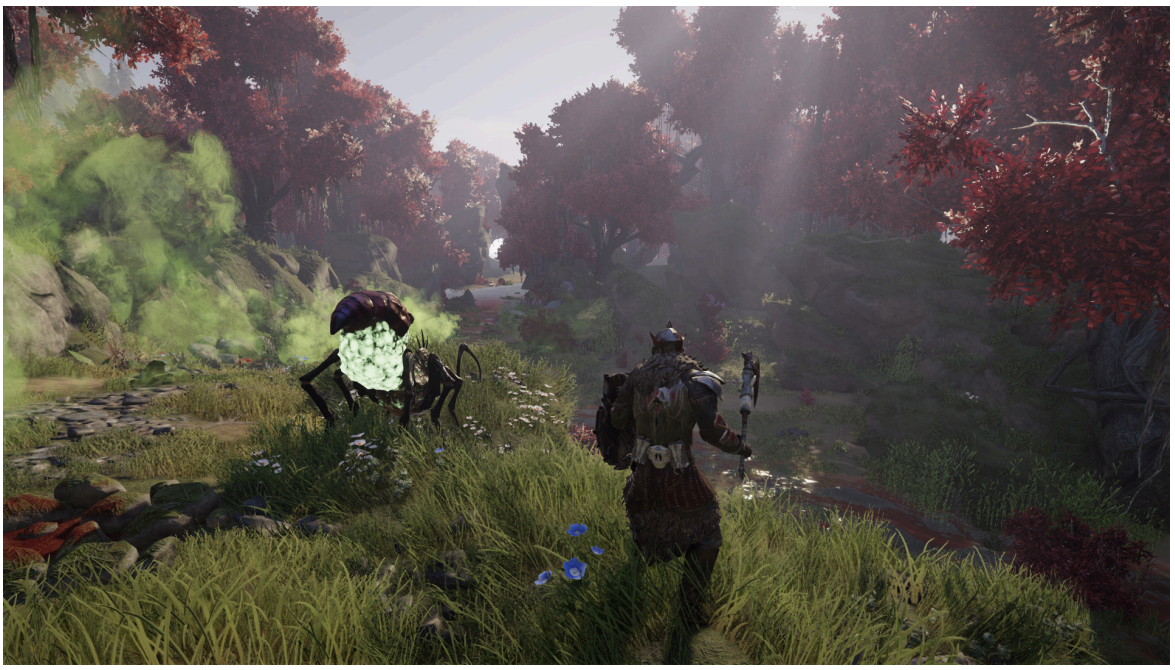


Рисунок 1.4 – скрішнот гри “Elex” з офіційної сторінки гри в стімі

Гравець може приєднатися до однієї з трьох фракцій: Берсеркери, Вигнанці та Клірики, кожна з яких має свої особливості. Відповіді гравця у діалогах впливають на стосунки з персонажами та подальший розвиток подій. Джакс може мати до шести компаньйонів, кожен з яких має свої квести, розпорядок дня та особистість.

Переваги:

1. Великий відкритий світ. Гра має величезний відкритий світ, наповнений різноманітними локаціями для дослідження. Відкритість світу дає гравцям велику свободу вибору шляху та дій.

2. Свобода вибору. Гравці можуть вибирати між різними фракціями, розвивати персонажа відповідно до своїх уподобань та вибирати різні шляхи вирішення завдань.

3. Глибокий сюжет. Гра має цікавий та загадковий сюжет, що зацікавлює гравців і примушує досліджувати світ гри, щоб розкрити його таємниці.

4. Різноманітність ворогів. У грі представлені різноманітні вороги, включаючи мутантів, роботів та інших створінь, які створюють різноманітні виклики для гравців.

Недоліки:

1. Складний геймплей. Деякі гравці відзначають, що гра може бути дещо важкою і незрозумілою на початку через складність бою та систему розвитку персонажа.

2. Технічні проблеми. У деяких версіях гри спостерігалися технічні проблеми, такі як погана оптимізація та відсутність полірування.

3. Застаріла графіка. В порівнянні з іншими сучасними іграми, гра може виглядати застарілою в плані графіки.

4. Неоднозначність сюжетних рішень. Деякі гравці відчувають, що сюжетні рішення в грі можуть бути неоднозначними або навіть оманливими, що може призвести до непередбачуваних наслідків.

The Witcher 2 Assassins of Kings - гра аналог (друга частина з трилогії). Відьмак – одна з найулюбленіших трилогій в світі, її обожнюють мільйони гравців по всьому світу. Перша частина цієї трилогії була розроблена в 2007 році, за мотивами однойменної серії романів "Відьмак" Анджея Сапковського, яка налічує 8 книг. Нехай гра і написана за мотивами романів, але розказує історію яка відбувається вже після їх завершення. В одному із своїх інтерв'ю розробники розказували що при розробці першої частини вони надихалися грою "Gothic" яка в той час була дуже популярною.

Відьмак 2: Вбивці королів (рис. 1.5) це відеогра з відкритим світом в жанрі RPG, яка розказує продовження історії Геральта з Ривії, відьмака який

довгі роки вважався мертвим та загубив свою пам’ять. Протягом гри Геральт переживає багато ситуацій, вбивство короля, у якого він був на службі, за день до того щоб піти у відставку, хибне звинувачення в вбивстві, тюремний срок, спроби відбілити свою репутацію, пошук інформації яка могла б нагадати йому про його минуле.

Рисунок 1.5 – обкладинка гри “Відьмак 2: Вбивці королів”

Протягом гри Геральт постійно робить вибори які впливають на наступну частину гри. Вибір між союзником який допоміг йому в тому щоб втекти з тюрми, в якій його катували за хибне звинувачення, та ельфом-розбійником Йорветом, який бореться за права ельфів яких “пригнічують”, заступитись за перехожого, якого грабують, чи пройти повз зробивши вигляд що нічого не помічаєш, вбити розумного тролля що перестав ремонтувати міст і на якого селяни зробили замовлення через це, при тому що люди вбили жінку троля і він впав в депресію, чи допомогти тролю розібравшись в ситуації і знайти винуватців цієї жахливої події, розібратись з людиною через яку його хибно звинуватили, чи випити з ним та розійтись з миром (рис. 1.6). І це лише

невелика частина всіх виборів що є в грі, все в світі відьмака відповідає середньовічному сеттінгу, це максимально жорстокий світ.

Рисунок 1.6 – мирне рішення в кінці гри “Відьмак 2”

Сюжет гри “Відьмак 2” відбувається після подій першої частини, де в фінальній катсцени показали замах на вбивство короля Фольтеста, а Геральт з легкістю врятував короля чим заслужив його милість та залишився в нього на службі. Гра починається із того що, головний герой відьмак Геральт із Ривії напівживий йде лісом. Після того він опиняється у замку де його допитує капітан «Блакитних смуг». Під час розмови герой згадує події від початку штурму замку, аж до вбивства короля Фольтеста. Після штурму король хоче побачити своїх дітей, які знаходились у монастирі, де причаївся найманець переодягнений монахом. Геральт іде із Фольтестом у монастир, та цього разу врятувати короля вже не вдається, вбивця тікає через вікно, стрибнувши у воду. Солдати, які прийшли на місце злочину, вирішили що вбивцею є Геральт та ув'язнили його. Історія повертається до розповіді капітана «Блакитних смуг». Капітан зрозумів, що Геральт невинний, і дозволив йому втекти із замку. Після втечі Геральт подався на пошуки вбивці короля, а з ним Трісс — його коханка,

та капітан королівського спецпідрозділу «Блакитні смуги» - Вернон Рош, який і влаштував втечу.

Переваги:

– глибокий сюжет. Гра має захоплюючий та інтригуючий сюжет, що базується на серії книг Анджея Сапковського. Вона веде гравця через багато гілок сюжету та виборів, що впливають на кінцевий результат;

– інноваційний бій. Відмінна бойова система, яка поєднує в собі стратегічне планування, швидкість та уміння. Бої в грі вимагають від гравця використання різноманітних прийомів, зброї та магії;

– графіка та атмосфера. Відмінний рівень деталізації середовищ та персонажів створює реалістичну та захоплюючу атмосферу світу Відьмака. Відтворення міст та локацій дуже вражаюче;

– відкритий світ. Гра пропонує великий відкритий світ з численними завданнями та можливістю вільно досліджувати його.

Недоліки:

– системні вимоги. Гра може вимагати відносно сильного комп'ютера для плавної роботи на максимальних налаштуваннях;

– складність боїв. Деякі гравці можуть відчувати, що бої в грі складні, особливо на вищих рівнях складності;

– технічні проблеми. На деяких системах можуть виникати проблеми з оптимізацією та відмови від праці гри;

– обмежені можливості дослідження. Хоча світ гри є великим, деякі гравці можуть відчувати обмеження у можливості досліджувати його, оскільки деякі області заблоковані до досягнення певного етапу сюжету.

Висновки до розділу 1

В даному розділі було досліджено історію виникнення відеоігор. Описано розвиток як відеоігор так і систем на яких можна в них грати та їх актуальність

в сучасності. Було описано сам жанр RPG та його популярність і актуальність в наш час.

Також було досліджено 3 аналоги та конкуренти мого продукту, описано їхні переваги та недоліки. Основним плюсом конкурентів був прописаний світ та сюжет. Також для себе я підмітив кілька плюсів які мені сподобались, наприклад наповнений та живий відкритий світ, прописані персонажі та свобода вибору.

Завдяки цьому дослідженню я отримав уявлення в яку сторону мені керуватися при написанні гри, де можна запозичити сильні сторони, а де не повторювати помилок яких допустились розробники аналогів.

РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА ПЛАН РОЗРОБКИ ВІДЕОГРИ

2.1 Опис рушія, порівняння з аналогами

Для реалізації проєкту мною було обрано рушій “Unity”, через його популярність, простоту та гнучкість. Завдяки цьому рушію можна розробляти як 2D ігри так і 3D, також можна розробляти анімаційні відео, додатки по типу різних симуляторів, навчальні програми, додатки віртуальної реальності та інтерактивні веб додатки.

Unity – це інтегроване середовище розробки (IDE) для створення ігор та інтерактивних додатків. Він вирізняється своєю простотою використання, гнучкістю та широким функціоналом.

Основні компоненти та функції Unity:

1. Графічний рушій: Unity використовує потужний графічний рушій, який дозволяє створювати вражаючі графічні ефекти та візуальні елементи. Він підтримує різні види текстур, освітлення, тіні, анімації та спеціальні ефекти.
2. Фізика: Unity має вбудовану систему фізики, яка дозволяє моделювати реалістичні фізичні ефекти, такі як рух та зіткнення об'єктів, сили та тиснення.
3. Штучний інтелект: рушій має набір інструментів для реалізації штучного інтелекту в іграх. Це включає в себе алгоритми поведінки, системи прийняття рішень та штучний інтелект для контрольованих об'єктів.
4. Звукові ефекти: Unity дозволяє додавати та керувати звуковими ефектами в іграх. Він підтримує різні формати аудіофайлів та дозволяє створювати реалістичні звукові ефекти та музику.
5. Анімація: Рушій має розширені інструменти для створення анімації об'єктів та персонажів. Він підтримує як ручну анімацію, так і анімацію на основі фізики.

6. Кросплатформеність: Unity дозволяє розробляти ігри та додатки для різних платформ, таких як Windows, macOS, Android, iOS, консолі та веб-браузери, що робить його ідеальним вибором для кросплатформної розробки.

7. Unity Asset Store: Це онлайн-магазин, де розробники можуть придбати або знайти безкоштовні ресурси, такі як моделі, текстури, анімації, скрипти та інше, щоб розширити функціональність своїх проектів.

Unity має як свої переваги так і недоліки, ось кілька з них:

Переваги:

– простота використання: Unity має інтуїтивний інтерфейс, що дозволяє швидко створювати ігри навіть новачкам у галузі розробки.

– кросплатформеність: розроблені за допомогою Unity ігри можуть бути запущені на різних платформах, таких як Windows, macOS, Android, iOS, консолі та веб-браузери.

– велика спільнота: Unity має велику спільноту розробників, яка активно допомагає один одному, ділиться ресурсами та рішеннями проблем.

– магазин активів: Unity Asset Store містить безліч безкоштовних та платних активів, які можна використовувати для розширення функціоналу вашого проекту.

Недоліки:

– продуктивність: деякі розробники відзначають, що Unity може мати проблеми з продуктивністю для деяких видів ігор або проектів з великою кількістю об'єктів на сцені.

– висока вартість ліцензії: Ххоча Unity пропонує безкоштовний план для початківців та маленьких команд, використання рушія на великих проектах може вимагати значних витрат на ліцензії та додаткові сервіси.

– обмеження в плануванні: деякі функції та можливості можуть бути недоступні для безкоштовного плану або вимагати покупки додаткових плагінів або активів.

В “Unity” є декілька аналогів, таких як Unreal Engine, Gogot Engine, LibGDX. Проте найпопулярнішим з них є UE, зараз на ньому розробляється набагато більше ігор ніж на Unity і він зміг змістити його з лідуючої позиції.

Unreal Engine - це ігровий рушій, розроблений компанією Epic Games, який використовується для створення ігор різних жанрів та для різних платформ.

Основні компоненти UE

1. Графічний рушій: UE має потужний графічний рушій, який дозволяє створювати вражаючі візуальні ефекти, реалістичні текстури та динамічне освітлення. Це дозволяє розробникам створювати візуально захоплюючі світи для своїх ігор.

2. Фізичний рух: UE має розширену систему фізики, яка дозволяє створювати реалістичні фізичні ефекти, такі як рух тіл, колізії та зіткнення.

3. Інтегрований редактор: UE постачається з інтегрованим редактором, що дозволяє розробникам створювати, редагувати та відлагоджувати групи, а також керувати ресурсами гри.

4. Мови програмування: Unreal Engine підтримує мови програмування, такі як C++ та Blueprints. C++ використовується для програмування високопродуктивного коду, тоді як Blueprints - це візуальний інтерфейс, що дозволяє розробникам створювати логіку гри без програмування.

5. Крос-платформеність: UE підтримує різні платформи, включаючи ПК, консолі, мобільні пристрої та віртуальну реальність, що робить його універсальним інструментом для розробки ігор для різних цільових аудиторій.

6. Інструменти розробки: UE має велику кількість вбудованих інструментів, таких як редактор матеріалів, система анімації та інструменти для роботи зі штучним інтелектом, що полегшує процес розробки гри.

UE так само як і Unity має свої переваги та недоліки:

Переваги:

– графічна якість: UE відомий своїми передовими графічними можливостями. Він має потужність для створення реалістичних світів з високоякісними текстурами, освітленням, тінями та ефектами частинок.

– гнучкість: рушій підтримує широкий спектр платформ, включаючи ПК, консолі, мобільні пристрої та віртуальну реальність, що робить його відмінним вибором для розробки ігор різних жанрів;

– широка спільнота розробників: UE має велику та активну спільноту розробників, яка надає безкоштовні ресурси, підтримку та поради для новачків та досвідчених розробників;

– інструменти для розробки: UE постачається з багатьма потужними інструментами для розробки ігор, такими як візуальний редактор матеріалів, система анімації та редактор поведінки, що полегшує процес створення ігор;

– підтримка мов програмування: UE підтримує мови програмування, такі як C++ та Blueprints, що дозволяє розробникам використовувати різні підходи до розробки ігор.

Недоліки:

– вимоги до ресурсів: через високу якість графіки та функціонал, UE може вимагати потужного комп'ютера для роботи, що може бути обмеженням для деяких користувачів;

– навчання: навчання UE може вимагати часу та зусиль, особливо для тих, хто тільки починає займатися розробкою ігор;

– вартість: хоча сам рушій є безкоштовним для некомерційного використання, прибуткові проекти підпадають під ліцензійні умови, що передбачають відрахування певної частки від прибутку до Epic Games;

– системні вимоги: UE може бути вимогливим до системних ресурсів, що може призводити до необхідності оновлення обладнання для ефективної роботи.

Що до переваг Unity над UE:

1. Простота в освоєнні: Unity вважається більш доступним для початківців, оскільки він має меншу крутизну криву навчання порівняно з

Unreal Engine. Інтерфейс Unity і концепції роботи можуть бути легше зрозуміти для тих, хто тільки починає займатися розробкою ігор.

2. Ширша підтримка мов програмування: Unity підтримує різні мови програмування, включаючи C#, JavaScript та Boo, що дозволяє розробникам обирати мову, з якою вони найбільш комфортно працювати.

3. Крос-платформеність: Unity має добре розвинену крос-платформенну підтримку, що дозволяє легко створювати ігри для різних платформ, включаючи ПК, консолі, мобільні пристрої та веб.

4. Екосистема активних користувачів та ресурсів: Unity має велику та активну спільноту розробників, яка надає безкоштовні ресурси, допомогу та підтримку через форуми, блоги та соціальні мережі.

5. Наявність безкоштовного варіанту: Unity надає можливість безкоштовного використання для особистих та комерційних проектів з деякими обмеженнями, що полегшує доступ до рушія для новачків та незалежних розробників.

Дивлячись на всі ці переваги та власне бажання, вирішив розробляти свій проєкт на рушії Unity. Після того як визначився з рушієм залишилось обрати мову програмування на якій буде розроблятись продукт.

2.2 Вибір мови програмування для розробки гри

Unity налічує багато мов програмування для реалізації та розробки ігор. Найпопулярнішою мовою для розробки на рушії Unity є мова програмування C#, вона зараз єдина мова програмування яку досі підтримують.

C# – це найпопулярніша мова програмування, яка використовується в розробці ігор на Unity. Це сучасна мова програмування, розроблена Microsoft, і використовується для створення різноманітних програм, включаючи ігри. C# легко вивчити, має чистий синтаксис і підтримує різноманітні функції, такі як класи, об'єкти, успадкування та поліморфізм. C# використовується для написання скриптів в Unity та надає доступ до API Unity для розробки ігор.

C/C++ – це мова програмування низького рівня, яка використовується для розробки високопродуктивних програм, таких як ігри. Вона використовується в розробці Unity для написання власних плагінів для завдань, де важлива продуктивність, таких як фізика, рендеринг та штучний інтелект. C/C++ надає доступ до апаратного забезпечення та пам'яті, що робить його ідеальним вибором для розробки ігор, які вимагають високої продуктивності.

Rust – це сучасна мова програмування, призначена для системного програмування, такого як розробка ігор. Це безпечна та швидка мова, яка забезпечує безпеку пам'яті без втрати продуктивності. Rust використовується в розробці ігор на Unity для написання власних плагінів для завдань, де важлива продуктивність, таких як фізика, рендеринг та штучний інтелект.

IronPython – це мова програмування, яка поєднує Python та .NET. Вона використовується в розробці ігор на Unity для написання скриптів для логіки гри та штучного інтелекту. IronPython має легкий для вивчення синтаксис та дозволяє розробникам використовувати величезний екосистему бібліотек Python в розробці ігор на Unity.

Lua – це легка скриптова мова програмування, яка використовується в розробці ігор на Unity для написання логіки гри та штучного інтелекту. Lua легка вивчити, має простий синтаксис та надає високу продуктивність. Lua використовується в багатьох ігрових рушіях і надає зручний інтерфейс для розробки ігор.

Java – це популярна мова програмування, яка використовується в різних програмах, включаючи розробку ігрових додатків. Вона використовується в розробці ігор на Unity для написання плагінів та розширень. Java має чистий синтаксис, величезний екосистему бібліотек та підтримує крос-платформеність, що робить її ідеальним вибором для розробки ігор для різних платформ.

JavaScript – це популярна мова скриптів, яка використовується в веб-розробці та розробці ігор. Вона використовується в розробці ігор на Unity для написання скриптів для логіки гри та користувацького інтерфейсу. JavaScript має легкий для вивчення синтаксис, високу продуктивність та

підтримує крос-платформеність, що робить її ідеальним вибором для розробки ігор для різних платформ.

SQL – це мова програмування, яка використовується для управління реляційними базами даних. Вона використовується в розробці ігор на Unity для управління гральними даними, такими як рахунки, досягнення та інформація користувача. SQL має легкий для вивчення синтаксис та дозволяє розробникам ефективно управляти гральними даними.

HTML 5 – це мова розмітки, яка використовується в веб-розробці та розробці ігор. Вона використовується в розробці ігор на Unity для створення веб-заснованих ігор та користувацького інтерфейсу. HTML 5 має легкий для вивчення синтаксис, підтримує крос-платформеність та мультимедійні можливості, що робить її ідеальним вибором для розробки веб-заснованих ігор.

CSS – це мова стилів, яка використовується в веб-розробці та розробці ігор. Вона використовується в розробці ігор на Unity для стилізації веб-заснованих ігор та користувацького інтерфейсу. CSS має легкий для вивчення синтаксис та дозволяє розробникам ефективно стилізувати інтерфейс.

Після огляду всіх доступних мов, мною було обрано мову програмування C#, через її зручність, лаконічність, синтаксис та її орієнтованість на рушій.

2.3. Система прокачки та світ гри

Система прокачки персонажа передбачає один основний шлях розвитку, який включає набуття досвіду через виконання різноманітних завдань, перемогу у боях та виконання специфічних дій у світі гри. За кожну успішну дію гравець отримує досвід, який додається до загального показника. При досягненні певної кількості досвіду рівень персонажа піднімається, а гравець отримує додаткові поінти рівня для подальшої прокачки.

Додатково гравець може відвідувати вчителів, які надають можливість вивчати нові навички та покращувати вміння у різних сферах. Гравець може витрачати свої зароблені поінти рівня, а також поінти героя, які він отримує

після перемоги над ексклюзивними босами, для отримання цих навичок. Таким чином, система прокачки персонажа включає в себе як набуття досвіду, так і вивчення нових навичок у вчителів за допомогою зароблених поінтів.

Поінти рівня є основною валютою прокачки, яку гравець може заробляти, підвищуючи свій рівень, виконуючи складні квести та перемагаючи могутніх босів. Ці поінти можна витрачати на покращення основних характеристик персонажа або на вивчення нових навичок у вчителів.

Поінти героя є важливою додатковою валютою у грі. Вони набуваються виключно в результаті перемоги над ексклюзивними босами, що робить їх досить цінними. Ці поінти відкривають доступ до елітних навичок та особливих можливостей, які можуть змінити геймплей гравця в значний спосіб.

Здобуття досвіду є ключем до розвитку персонажа. Гравець може отримувати досвід, перемагаючи ворогів, виконуючи складні завдання або навіть виконуючи певні складні дії у грі. За кожну успішну дію гравець отримує відповідну кількість досвіду, що додається до загального показника. При досягненні певної кількості досвіду рівень персонажа піднімається, а гравець отримує додаткові поінти рівня для подальшої прокачки.

Враховуючи аспекти відкритого світу гри, слід відзначити, що світ буде докладно прорисованим та наповненим різноманітними локаціями, зброєю та предметами. Певні зони можуть бути більш складними, з ворогами різного рівня складності, що вимагає від гравця вибіркового підходу до розвитку персонажа. Також важливо зазначити, що в грі будуть присутні різноманітні види тварин та ворогів, які населяють світ гри і можуть впливати на геймплей гравця.

2.4. Реалізація інтерфейсу користувача

2.4.1. Інтерфейс в грі

Інтерфейс користувача був створений за допомогою програми “Figma”, яка відкриває широкі можливості для створення та налаштування дизайну.

Використання різноманітних компонентів, стилів та шаблонів дозволило нам ефективно втілити в життя інтерфейс гри.

Сам інтерфейс користувача складається з трьох основних елементів: двох полосок та півкола. Червона полоска відображає рівень здоров'я гравця, синя - рівень магічної енергії, а півколо зеленого кольору представляє витривалість. Використання цих кольорів є стандартним у більшості відеоігор, що дозволяє гравцям інтуїтивно розуміти їхнє значення. Ці елементи зазвичай розташовуються в лівому верхньому куті, що є типовим для ігор подібного жанру, хоча існують винятки, коли їх розміщують внизу екрану (рис. 2.1).

У нижньому лівому куті розташована мінікарта, яка є необов'язковою та може бути вимкнена у налаштуваннях. Вона додається для зручності орієнтування гравця у світі гри (рис. 2.1).

У центрі знизу розташовано меню хотбару, куди гравець може розмістити різноманітні предмети, такі як їжа, зброя, сувої та магічні здібності, для швидкого доступу під час гри(рис. 2.1).

В верхньому правому куті знаходиться панель з активними ефектами, яка відображає як позитивні, так і негативні ефекти. Ці ефекти можуть бути отримані гравцем під час бою або внаслідок використання певних предметів чи навичок(рис. 2.1).

Рисунок 2.1 – вайрфрейм графічного інтерфейсу

2.4.2. Інтерфейс інвентарю, магії та навичок

У розділі інвентарю, користувач зустріне аватар свого персонажа, де може роздивитися його зовнішній вигляд, а також знайде полоски здоров’я, магії та витривалості, які супроводжуються цифровими значеннями. Поруч із аватаром розташоване поле, де вказані всі характеристики персонажа, такі як сила, витривалість, інтелект, воля, харизма та живучість (рис. 2.2).

Справа від поля характеристик розташовано інвентар, який поділений на кілька розділів: всі предмети, зброя, обладунки, їжа та інші предмети, які не використовуються безпосередньо (рис. 2.2).

У розділі магії та навичок планується аналогічний інтерфейс, де залишаються аватар та поле з атрибутами персонажа. Однак буде додано поле з переліком доступної магії та всіх навичок, як відкритих, так і закритих. На недоступних навичках будуть вказані вимоги для їх отримання.

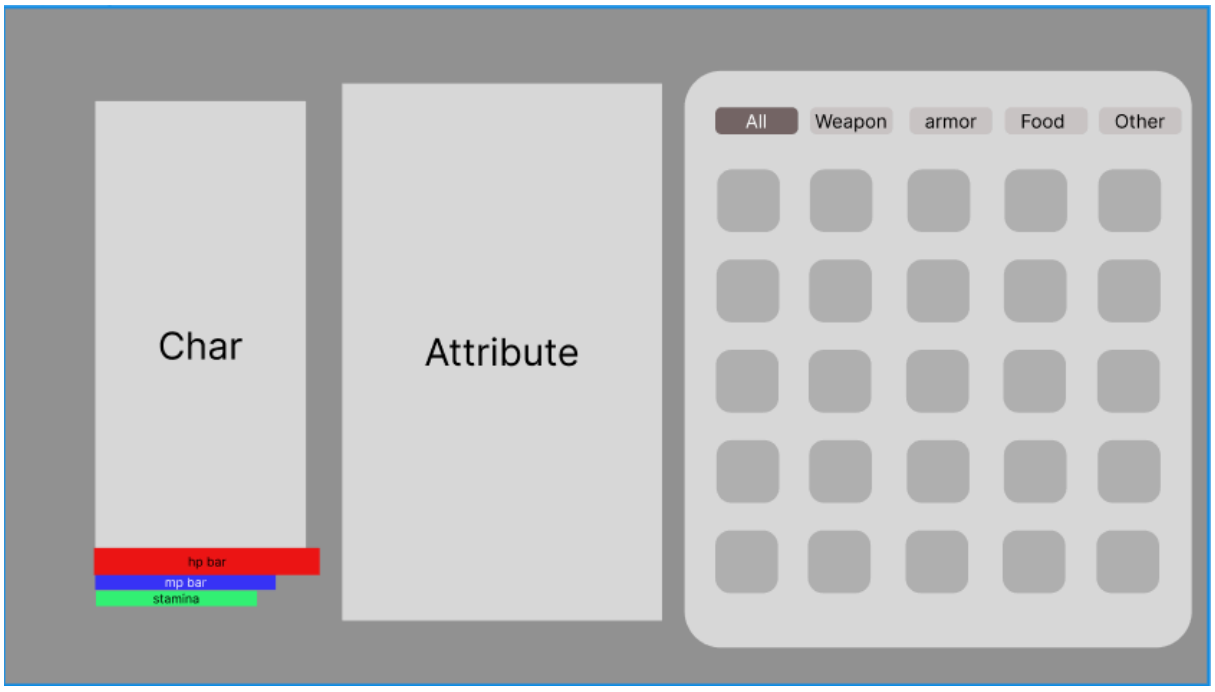


Рисунок 2.2 – вайрфрейм вікна з інвентарем

В вікні магії вся доступна магія буде розділена по школам, наприклад, вогняна куля буде в школі магії вогню, крижаний спис - в школі криги, земляна

куля - в школі землі, а священний супровід та святе світло будуть віднесені до школи магії підтримки. На відміну від вікна з навичками, як отримати ту чи іншу магію не буде вказано.

Рисунок 2.3 – вайрфрейм вікна з навичками та магією

2.5. Система крафту

У гру планується інтеграція системи крафту для створення різноманітних предметів, таких як зброя ближнього та дальнього бою, ювелірні вироби, зілля, їжа та квестові предмети. Всі власноруч створені предмети будуть кращими за свої аналоги, які можна знайти або придбати, і гравці також матимуть можливість покращувати будь-яку зброю, маючи навички її створення.

Наприклад, для створення меча гравець повинен буде знайти креслення та матеріали. Матеріали, необхідні для ковки, будуть описані в кресленні, наприклад, 1 сталева болванка, 1 оброблена шкіра та 3 металевих злитки.

Після того, як гравець знайде матеріали та отримає навичку коваля, йому доведеться пройти увесь процес ковки, від розігрівання болванки до заточування отриманої зброї.

У разі бажання гравця зварити зілля, йому також доведеться знайти всі необхідні інгредієнти, вивчити або знайти рецепт та використати хімічну лабораторію.

Це дозволить гравцям не лише створювати унікальні предмети, але й розвивати свої навички та досліджувати глибини гри через взаємодію з різноманітними механіками крафту.

Висновки до розділу 2

В даному розділі було розглянуто рушій Unity та його аналоги, мови програмування які підходять для нього та розглянуто переваги та недоліки рушія. Було обрано мову програмування та описано її переваги та недоліки.

Також розроблені вайфрейми з прикладами інтерфейсів користувача та інвентарю.

А також описані механіки які будуть додані в гру в кінцевому результаті, такі як механіки крафту а також підняття рівня та прокачки навичок.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ВІДЕОГРИ

3.1. Створення базового середовища відеогри

Щоб почати розробку відеогри потрібно встановити сам рушій Unity, а також середовище розробки Microsoft Visual Studio. Зробити це можна з офіційних сайті рушія[17] та середовища розробки[18].

Після того як я встановив рушій потрібно створити новий проєкт, для цього я відкрив Unity Hub та клацнув на кнопку “New project” (рис. 3.1).

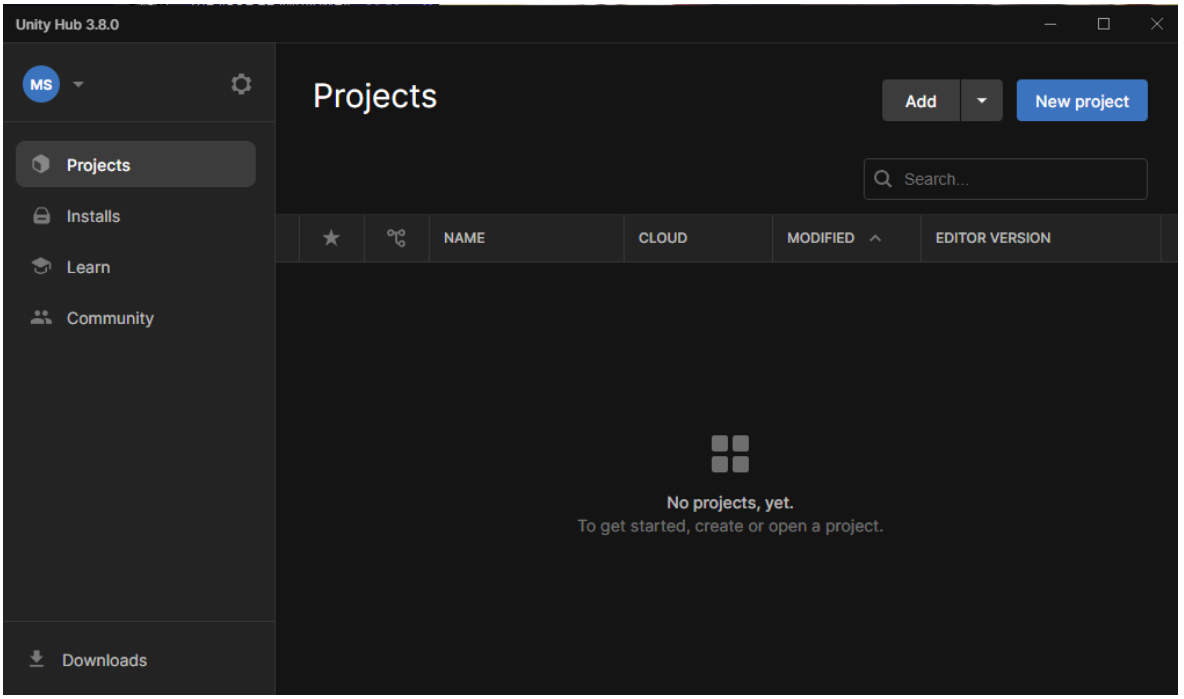


Рисунок 3.1 – початкове вікно програми Unity Hub

В вікні що відкрилось було вибрано пункт 3D (Built-In Render Pipeline), що надає можливість створити пустий проєкт в 3D, після цього було клацнуто кнопку “Create project”, що починає створення проєкта (рис. 3.2).

Після того як проєкт створено відкривається сама програма в якій знаходиться лише пустий простір, далі було створено платфору та персонажа для тестування механік(рис. 3.3).

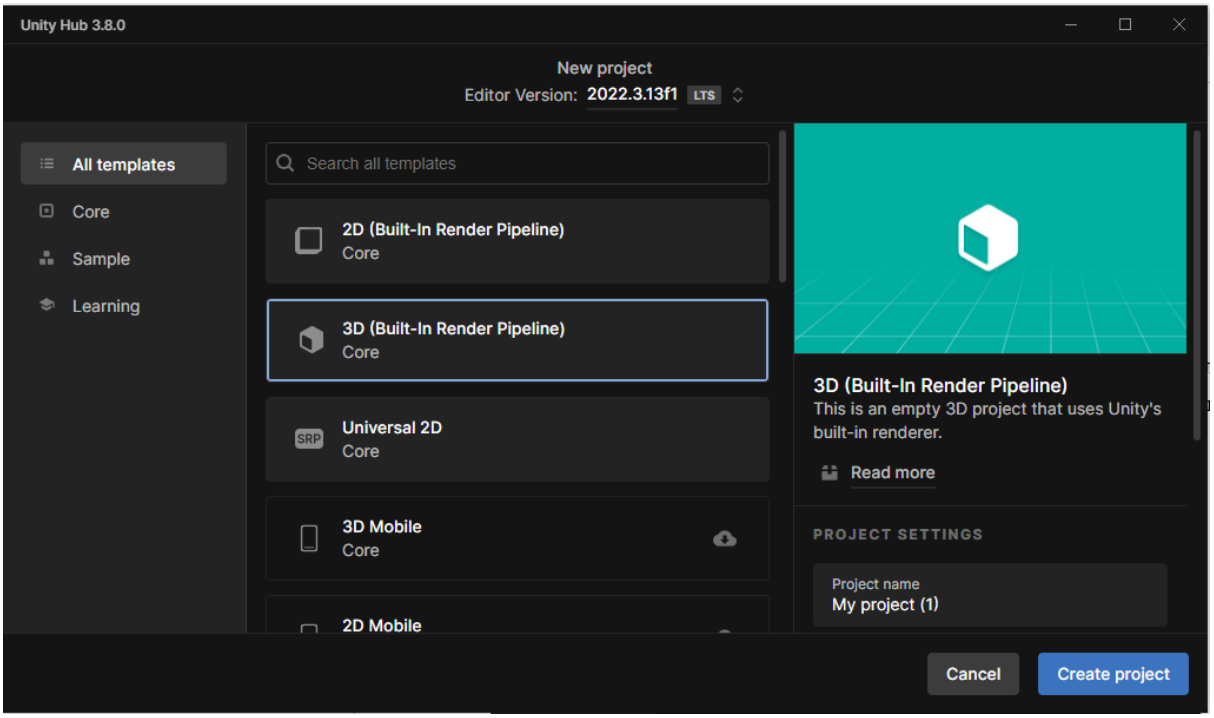


Рисунок 3.2 – вікно вибору типу гри

Для початку, я буду використовувати безкоштовні ассети для юніті, для того щоб мати манекен на якому буде перевірятись вся робота скриптів. Мною був обраний ассет під назвою “Starter Assets - ThirdPerson”[19], з нього мною було запозичено персонажа та декілька текстур.

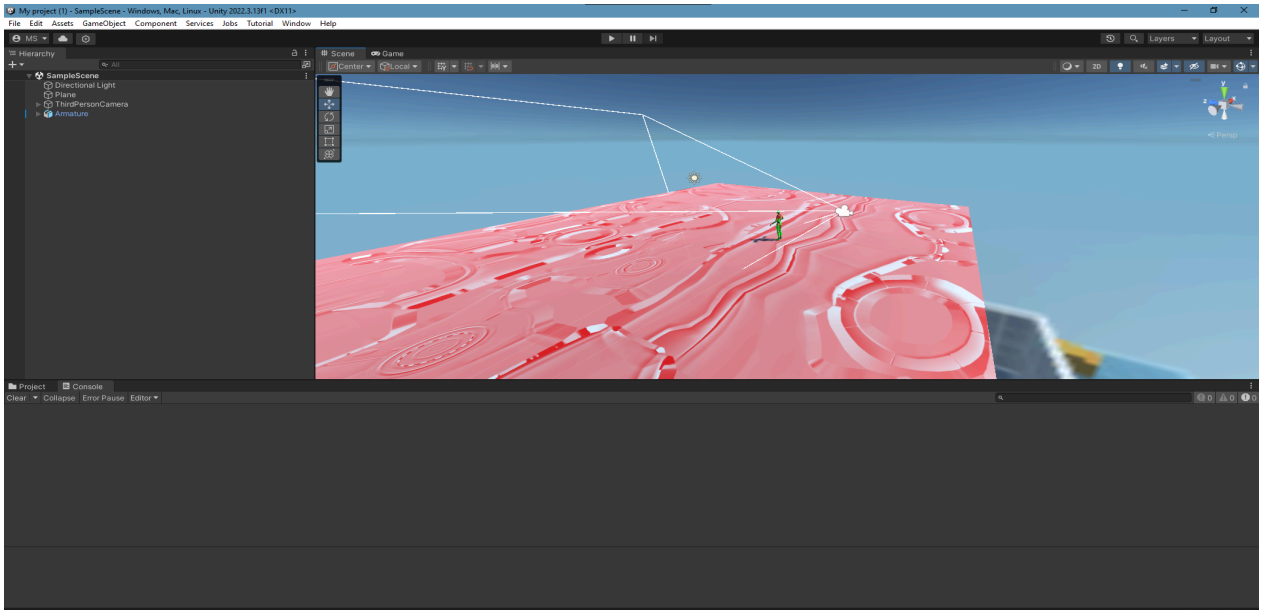


Рисунок 3.3 – сцена ігрового середовища

Зайшовши в сцену я створив платформу, на якій будуть проводитись перевірки, персонажа та камеру, після чого перейшов до написання перших скриптів (рис. 3.3).

3.2. Написання базових скриптів та створення анімацій

Для написання перших скриптів для початку потрібно їх створити, для цього я створив папку Scripts а в ній 3 скрипта під назвами “CameraFollow”, для того аби камера слідувала за персонажем, “MovemantCharacteristic”, для опису характеристик переміщення, “ThirdPersonMovemant” для розписання самого переміщення.

Код скрипта “CameraFollow”:

```
public class CameraFollow : MonoBehaviour
{
    [SerializeField]
    [Range(80f, 100f)] private float _angularSpeed = 100f;

    [SerializeField] private Transform _target;

    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
    }
```

public class CameraFollow : MonoBehaviour: це оголошення класу, який успадковує клас MonoBehaviour. Класи у Unity, які успадковують MonoBehaviour, можуть містити методи, які автоматично викликаються під час різних подій в життєвому циклі об'єкта;

[SerializeField] [Range(80f, 100f)] private float _angularSpeed = 100f;: це оголошення приватної змінної, яка визначає швидкість обертання камери. Атрибут [SerializeField] дозволяє відображати цю змінну в Інспекторі Unity для

зручності редагування під час розробки. Атрибут [Range(80f, 100f)] встановлює межі значень для цієї змінної в Інспекторі;

[SerializeField] private Transform _target;: це оголошення приватної змінної, яка вказує на об'єкт, за яким буде слідувати камера. Вона також відображається в Інспекторі Unity;

void Start(): це метод, який викликається один раз при запуску програми. У цьому методі встановлюється поведінка курсора.

```
       void Update()
          {
               float deltaX = Input.GetAxis("Mouse X") * _angularSpeed * 
Time.deltaTime;
               float deltaY = Input.GetAxis("Mouse Y") * _angularSpeed * 
Time.deltaTime;

                 transform.RotateAround(_target.position,     Vector3.up, 
deltaX);
              transform.Rotate(Vector3.left * deltaY);

              transform.position = _target.transform.position;
          }
```

void Update(): це метод, який викликається один раз на кожний кадр. У цьому методі оброблюється вхід користувача і здійснюється слідування камери за об'єктом;

float deltaX = Input.GetAxis("Mouse X") * _angularSpeed * Time.deltaTime; та float deltaY = Input.GetAxis("Mouse Y") * _angularSpeed * Time.deltaTime;: ці рядки визначають змінні, які представляють зміну позиції курсора миші по горизонталі та вертикалі, помножену на швидкість обертання камери та час, який пройшов з останнього кадру;

transform.RotateAround(_target.position, Vector3.up, deltaX);: цей рядок обертає камеру навколо позиції _target по горизонталі залежно від зміни позиції курсора миші по горизонталі;

transform.Rotate(Vector3.left * deltaY);: цей рядок обертає камеру по вертикалі відповідно до зміни позиції курсора миші по вертикалі;

transform.position = _target.transform.position;: цей рядок встановлює позицію камери рівною позиції _target, що дозволяє камері слідувати за об'єктом.

```
private void OnApplicationFocus(bool hasFocus)
    {
        if (hasFocus)
        {
            Cursor.lockState = CursorLockMode.Locked;
            Cursor.visible = false;
        }
        else
        {
            Cursor.lockState = CursorLockMode.None;
            Cursor.visible = true;
        }
    }
```

private void OnApplicationFocus(bool hasFocus): цей метод викликається, коли додаток отримує або втрачає фокус. У разі втрати фокусу курсор робиться видимим, а у разі повернення фокусу курсор приховується, щоб не відволікати користувача.

Код скрипта “MovemantCharacteristic”:

```
[CreateAssetMenu(fileName ="Characteristics", menuName = "Movement/MovementCharacteristics", order = 1)]
public class MovementCharacteristics : ScriptableObject
{
    [SerializeField] private bool _visibleCursor = false;
    [SerializeField] private float _movementSpeed = 1f;
    [SerializeField] private float _angularSpeed=150f;
    [SerializeField] private float _gravity = 15f;
    public bool VisibleCursor => _visibleCursor;
    public float MovementSpeed => _movementSpeed;
```

```
    public float AngularSpeed => _angularSpeed;
    public float Gravity => _gravity;
}
```

[CreateAssetMenu(fileName ="Characteristics", menuName = "Movement/MovementCharacteristics", order = 1)]: ця атрибутна мітка додає можливість створення нових об'єктів MovementCharacteristics через редактор Unity. Вона розташована у верхній частині класу. fileName вказує на ім'я файлу за замовчуванням, menuName - на назву пункту меню у редакторі Unity, а order - на порядок в меню;

[SerializeField] private bool _visibleCursor = false;: ця змінна вказує, чи має курсор бути видимим. Позначка [SerializeField] дозволяє відображати цю змінну в Інспекторі Unity;

[SerializeField] private float _movementSpeed = 1f;: ця змінна визначає швидкість руху;

[SerializeField] private float _angularSpeed=150f;: ця змінна визначає швидкість обертання;

[SerializeField] private float _gravity = 15f;: ця змінна визначає значення гравітації;

public bool VisibleCursor => _visibleCursor;: цей рядок визначає публічну властивість VisibleCursor, яка повертає значення _visibleCursor. це дає змогу іншим об'єктам звертатися до цієї властивості;

public float MovementSpeed => _movementSpeed;: подібно до попереднього рядка, цей рядок визначає публічну властивість MovementSpeed, яка повертає значення _movementSpeed.

Код скрипта “ThirdPersonMovemant”:

```
[RequireComponent(typeof(CharacterController))]
public class ThirdPersonMovement : MonoBehaviour
{
    [SerializeField] private Transform _camera;
```

```
                [SerializeField]   private   MovementCharacteristics
_characteristics;

        private float _vertical, _horizontal;

        private readonly string STR_VERTICAL = "Vertical";
        private readonly string STR_HORIZONTAL = "Horizontal";

        private const float DISTANCE_OFFSET_CAMERA = 5f;

        private CharacterController _controller;

        private Vector3 _direction;
        private Quaternion _look;
```

[RequireComponent(typeof(CharacterController))]: ця атрибутна мітка вказує, що компонент CharacterController повинен бути доданий до цього об'єкта або буде доданий автоматично. Це потрібно для забезпечення наявності контролера персонажа;

public class ThirdPersonMovement : MonoBehaviour: це оголошення класу ThirdPersonMovement, який успадковує клас MonoBehaviour;

[SerializeField] private Transform _camera;: ця змінна визначає трансформ компонент камери, яка буде використовуватися для орієнтації руху персонажа;

[SerializeField] private MovementCharacteristics _characteristics;: ця змінна визначає об'єкт MovementCharacteristics, який містить характеристики руху персонажа;

private float _vertical, _horizontal;: ці змінні визначають вертикальну та горизонтальну компоненти вхідних даних руху;

private readonly string STR_VERTICAL = "Vertical"; private readonly string STR_HORIZONTAL = "Horizontal";: ці рядки визначають назви вхідних осей руху.

private const float DISTANCE_OFFSET_CAMERA = 5f;: ця константа визначає відстань між камерою та персонажем;

private CharacterController _controller;: ця змінна зберігає посилання на компонент CharacterController, який управляє рухом персонажа;

```
    private Vector3 TargetRotate => _camera.forward * DISTANCE_OFFSET_CAMERA;
        private bool Idle => _horizontal == 0.0f && _vertical == 0.0f;
```

private Vector3 _direction; private Quaternion _look;: ці змінні визначають напрямок руху та кут огляду персонажа;

```
    private void Movement()
        {
            if (_controller.isGrounded)
            {
                _horizontal = Input.GetAxis(STR_HORIZONTAL);
                _vertical = Input.GetAxis(STR_VERTICAL);
                _direction = transform.TransformDirection(_horizontal, 0, _vertical).normalized;
            }
            _direction.y -= _characteristics.Gravity * Time.deltaTime;
             Vector3 dir = _direction * _characteristics.MovementSpeed * Time.deltaTime;
            _controller.Move(dir);
        }
```

private void Movement(): цей метод визначає рух персонажа. Він отримує вхідні дані від осей клавіатури або контролера та переміщує персонажа відповідно до цих даних.

```
    private void Rotate()
        {
            if (Idle) return;
            Vector3 target = TargetRotate;
            target.y = 0;
            _look = Quaternion.LookRotation(target);
```

```
            float speed = _characteristics.AngularSpeed
*Time.deltaTime;
            transform.rotation =
Quaternion.RotateTowards(transform.rotation, _look, speed);
        }
    }
```

private void Rotate(): цей метод визначає орієнтацію персонажа. Він повертає персонажа в напрямку, в якому дивиться камера.

```
    private void Start()
        {
            _controller = GetComponent<CharacterController>();
            Cursor.visible = _characteristics.VisibleCursor;
        }
     private void Update()
        {
            Movement();
            Rotate();
        }
```

private void Start(): Цей метод викликається один раз при запуску програми. У ньому відбувається ініціалізація контролера персонажа та налаштування видимості курсора відповідно до параметрів MovementCharacteristics;

private void Update(): Цей метод викликається один раз на кожний кадр. У ньому відбувається оновлення руху та орієнтації персонажа.

Весь цей код дає моєму персонажу змогу переміщатись, та блокує камеру в нього за спиною

Далі мені потрібно було створити анімації для переміщення, для цього було створено новий контроллер, і в нього додано два поля типу float, їх було названо Horizontal і Vertical, після цього створене нове дерево змішування. В самому дереві вибирано тип змішування з назвою “2D Freeform Direction” куди додаю 2 параметри Horizontal і Vertical, додаю 3 поля для руху, надаю їм параметри “-1,0,1” та вставляю анімації з безкоштовного асету (рис 3.4).

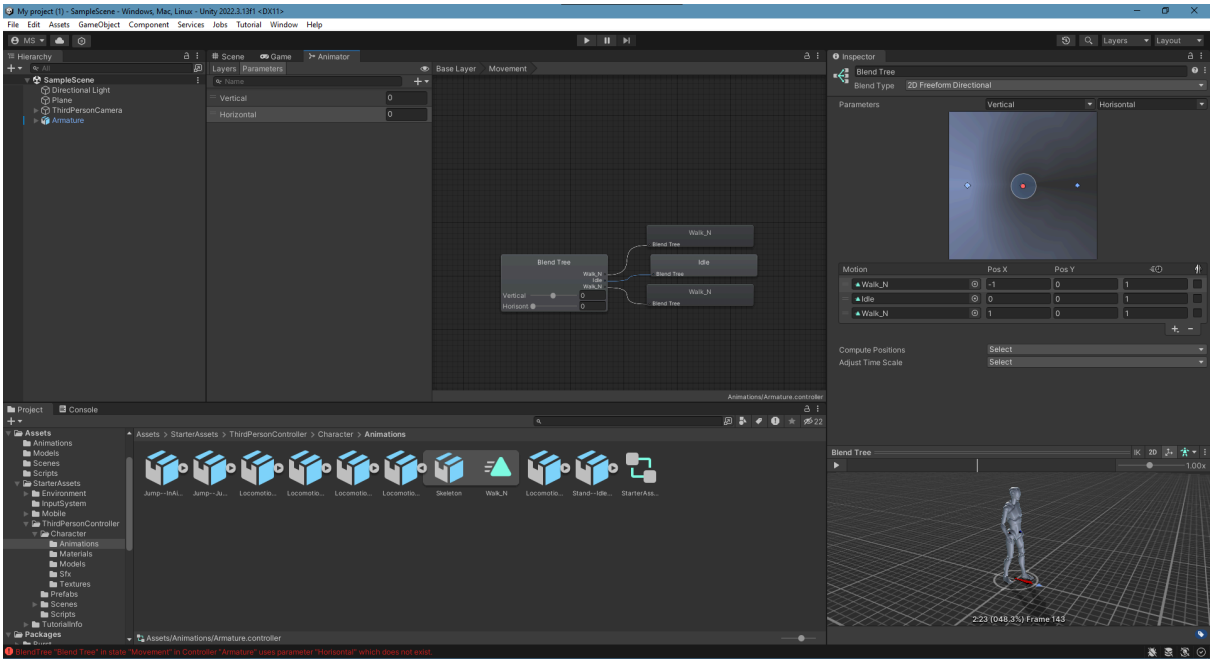


Рисунок 3.4 – додавання анімацій

Висновки до розділу 3

В даному розділі було детально описано кроки розробки в рушії Unity, налаштовано середовище, написано базові скрипти та створено анімації.

Було створено скрипт для переміщення гравця, який використовує величини vertical та horizontal що здійснюють переміщення вперед, назад а також по бокам.

Було написано скрипт що заставляє камеру слідувати за персонажем, скрипт відслідковує рухи мишкою для поворота камерою, при повороті в праву та ліву сторони повороти здійснюються по горизонталі залежно від зміни позиції курсора миші по горизонталі та вертикалі.

Було написано код, в якому зберігаються характеристики переміщення такі як швидкість руху, гравітація та видимість курсору, що використовуються в інших скриптах.

А також було зроблено базові анімації переміщення використовуючи аніматор Unity.

ВИСНОВОК

У рамках кваліфікаційної роботи було досліджено предметну область проекту, що передбачає створення бета-версії відеоігри в жанрі 3D RPG, використовуючи мову програмування C# в середовищі Unity. Аналіз існуючих аналогів допоміг визначити переваги та недоліки конкурентів, що дозволило покращити власну відеогру.

При виборі інструментів для розробки додатку враховувалися їхня доступність, надійність та популярність, що сприяло створенню стабільно працюючого ігрового додатку. Рушій Unity забезпечив швидку та ефективну розробку, тоді як поєднання Unity Assets з Figma задовільнило потреби в графічних елементах та дало уявлення про вигляд інтерфейсу.

Було розроблено систему, яка відповідає за переміщення персонажа та автоматичне переслідування його камерою, забезпечуючи плавний і природний ігровий досвід. До гри також були додані різноманітні моделі та анімації, що значно покращило візуальну складову та наділило персонажів життя і реалістичністю. Крім того, була створена комплексна система, що відповідає за накопичення досвіду та підвищення рівня персонажів, яка додає елемент прогресії та мотивації для гравців, дозволяючи їм розвивати своїх героїв та досягати нових висот у грі.

Результатом роботи стала створена бета-версія гри, яку можна запускати на інших персональних комп'ютерах, що дозволяє гравцям тестувати її та надавати цінний зворотний зв'язок. У майбутньому планується значно доопрацювати гру: перемалювати всі моделі та текстури для досягнення більш високої якості графіки, додати нові ігрові механіки для розширення можливостей геймплею, а також розробити систему завдань і детальний сюжет, що зробить гру більш захопливою та інтерактивною для користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Безкоштовний звіт: Звіт про світовий ринок ігор Newzoo 2019: веб-сайт.URL:https://newzoo.com/resources/trend-reports/newzoo-global-games-market-report-2019-light-version (дата звернення 12.03.2024).

2. Ігрова індустрія в цифрах: скільки українці витрачають на відеоігри Mind.ua: веб-сайт. URL: https://mind.ua/publications/20222353-igrova-industriya-v-cifrah-skilki-ukrayinci-vitrachayut-na-videoigri (дата звернення 12.03.2024).

3. Game Industry Usage and Revenue Statistics 2023 - Helplama.com: веб-сайт. URL: https://helplama.com/game-industry-usage-revenue-statistics/ (дата звернення 17.03.2024).

4. Теоретичні методи. Аналіз та синтез: веб-сайт. URL: https://naurok.com.ua/teoretichni-metodi-analiz-ta-sintez-376029.html (дата звернення 17.03.2024).

5. Порівняльний метод - Sapiens Methodology: веб-сайт. URL: https://metodologiasapiens.com/uk/metodos/metodo-comparativo/ (дата звернення 17.03.2024).

6. Тестування програмного забезпечення: етапи та методи: веб-сайт. URL:https://foxminded.ua/testuvannia-prohramnoho-zabezpechennia/ (дата звернення 18.03.2024).

7. A Brief History of Gaming & The Gaming Industry: веб-сайт. URL: https://www.lg.com/ca_en/gaming/a-brief-history-of-gaming/ (дата звернення 18.03.2024).

8. What is a role-playing game? | Definition from TechTarget: веб-сайт. URL: https://www.techtarget.com/whatis/definition/role-playing-game-RPG (дата звернення 19.03.2024).

9. Gothic 1 on Steam: веб-сайт. URL:https://store.steampowered.com/app/65540/Gothic_1/ (дата звернення 21.03.2024).

10. ELEX — Вікіпедія: веб-сайт. URL:https://uk.wikipedia.org/wiki/ELEX (дата звернення 21.03.2024).

11. ELEX on Steam: веб-сайт. URL:https://store.steampowered.com/app/411300/ELEX/ (дата звернення 21.03.2024).

12. Відьмак 2: Убивці королів — Вікіпедія: веб-сайт. URL: https://uk.wikipedia.org/wiki/Відьмак_2:_Убивці_королів (дата звернення 21.03.2024).

13. Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine: веб-сайт. URL:https://unity.com (дата звернення 28.03.2024).

14. The most powerful real-time 3D creation tool -Unreal Engine: веб-сайт. URL: https://www.unrealengine.com/en-US (дата звернення 28.03.2024).

15. Best Unity alternatives for game development: веб-сайт. URL: https://www.androidpolice.com/best-unity-game-engine-alternative/ (дата звернення 28.03.2024).

16. Learn these 10 Unity Game Development Languages By 2023: веб-сайт. URL:https://medium.com/codex/learn-these-10-unity-game-development-languages-by-2023-31c3d66c66e8 (дата звернення 28.03.2024).

17. Розпочніть свої творчі проекти та завантажуйте Unity | Unity: веб-сайт. URL: https://unity.com/download (дата звернення 21.04.2024).

18. Free Developer Software & Services - Visual Studio: веб-сайт. URL: https://visualstudio.microsoft.com/free-developer-offers/ (дата звернення 21.04.2024)

19. Starter Assets - ThirdPerson | Updates in new CharacterController package Essentials Unity Asset Store: веб-сайт: URL: https://assetstore.unity.com/packages/essentials/starter-assets-thirdperson-updates-in-new-charactercontroller-pa-196526(дата звернення 25.04.2024)

20. FREE Low Poly Human - RPG Character | Characters | Unity Asset Store:веб-сайт:URL:https://assetstore.unity.com/packages/3d/characters/humanoids/fantasy/free-low-poly-human-rpg-character-219979 (дата звернення 25.04.2024)

метадані

Заголовок

Розробка бета версії відеогри в жанрі RPG

Автор
Шиньквар М.

Науковий керівник / Експерт
кандидат технічних наук Сергій Ващишак

підрозділ
King Danylo University

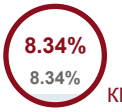
Тривога

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.

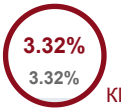
Заміна букв	ß	2
Інтервали	A→	0
Мікропробіли	·	0
Білі знаки	ß	0
Парафрази (SmartMarks)	a	23

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

7910
Кількість слів

58852
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	https://www.wikidata.uk-ua.nina.az/The_Witcher_2:_Assassins_of_Kings.html	125	1.58 %
2	https://essuir.sumdu.edu.ua/bitstream/123456789/85403/1/Fomenko_bac_rob.pdf	46	0.58 %
3	https://essuir.sumdu.edu.ua/bitstream/123456789/85403/1/Fomenko_bac_rob.pdf	34	0.43 %
4	https://essuir.sumdu.edu.ua/bitstream/123456789/85403/1/Fomenko_bac_rob.pdf	30	0.38 %
5	https://essuir.sumdu.edu.ua/bitstream/123456789/85403/1/Fomenko_bac_rob.pdf	28	0.35 %