

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних
технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: 2D піксельний платформер в середовищі розробки UNITY

Виконав студент 2 курсу групи МІС-22
спеціальності 122 Комп'ютерні науки
Митрофаненко Станіслав
Вікторович

Керівник к.ф.-м.н.,
Ткач Тетяна Борисівна

Рецензент к.т.н., доцент
Гнатовська Г.А.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	7
ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Аналіз стану ігрової індустрії	9
1.2 Порівняльний аналіз популярних ігрових пристроїв	18
1.3 Порівняльна характеристика ігрових движків	24
1.4 Аналіз операційних систем для ігор	36
2 ТЕХНОЛОГІЇ РОЗРОБКИ ІГРОВИХ ПРОДУКТІВ	41
3 ПРОЕКТУВАННЯ ГРИ	46
3.1 Визначення вимог до програмного забезпечення та розробка технічного завдання	46
3.2 Проектування архітектури програмного забезпечення	48
3.3 Підготовка графічних елементів	49
3.4 Внутрішня ігрова логіка	56
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ	59
4.1 Вибір засобів розробки	59
4.2 Структура та функціональне призначення модулів програми, їх взаємозв'язок	63
4.3 Розробка програмних модулів	72
4.4 Керівництво користувача	79
ВИСНОВКИ	82
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	84
ДОДАТОК А Вихідний код програми	87

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

2D – 2-Dimensional, 2-Вимірний

ОС – операційна система

ПЗ – програмне забезпечення

ПК – персональний комп'ютер

DLC – Downloadable Content, Завантажуваний контент

DOTS – Data-Oriented Technology Stack, Стек технологій, орієнтованих на дані

FPP – First Playable Prototype, Перший готовий прототип

IDE – Integrated Development Environment, Інтегроване середовище розробки

ІК – Inverse Kinematics, Обернена кінематика

PR – Public Relations, Зв'язки з громадськістю

QA – Quality Control, Контроль якості

TMP – TextMeshPro, Розширення для Unity, яке надає розширені можливості для роботи з текстом.

UI – User Interface, Користувацький інтерфейс

VS – Visual Studio, Інтегроване середовище розробки для програмістів, що підтримує різні мови програмування

ВСТУП

Індустрія відеоігор є галуззю економіки, що займається розробкою, маркетингом та випуском відеоігор. На сьогодні цей сектор вважається одним з найбільш перспективних, і його розвиток відбувається досить інтенсивно. З появою та еволюцією сучасних персональних комп'ютерів, консолей та мобільних платформ відбулася трансформація продукції геймдев-студій. Крім того, з'явилися нові методи поширення ігор, зокрема через онлайн платформи.

Ігрова індустрія є різноманітною та відкритою для новаторських ідей, передових технологій та досягнень. Сучасні ігри втілюють в собі не лише можливість цікаво провести час, але й дарують яскраві емоції та незабутні враження. Мотивація для гри полягає не лише у самій грі, але й у відчуттях та досвіді, які вона приносить. Гравці занурюються в ігри, щоб створити неперевершені враження та переживання, подолати складні виклики гри та знайти відпочинок від щоденних турбот.

Україна має активну геймдев-спільноту, в якій зосереджено багато талановитих програмістів, дизайнерів та художників, які займаються створенням ігор в різних жанрах, в тому числі й у 2D платформерах. Багато студій та команд з України вже здобули успіхи в галузі геймдеву, зокрема, були визнані на міжнародних конкурсах та отримали нагороди за свої проекти.

Метою кваліфікаційної роботи є проектування ігрового процесу, створення програмних модулів, здійснення етапів розробки комп'ютерної гри. Для реалізації мети дипломної роботи визначено наступні завдання:

- Провести дослідження ринку ігрової індустрії.
- Надати характеристику основним програмним інструментам для розробки ігор.
- Проаналізувати ключові етапи створення комп'ютерних ігор.
- На основі отриманих даних розробити комп'ютерну гру.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз стану ігрової індустрії

У другій половині двадцятого століття почала розвиватись індустрія відеоігор. Відеогра – комп'ютерна програма, що служить для організації ігрового процесу (геймплея), зв'язку з партнерами по грі, або сама виступає як партнер, в ігровому процесі гравець використовує інтерфейс користувача, щоб отримати зворотну інформацію з відеопристрою [1]. Індустрія відеоігор – це сектор економіки, який пов'язаний із розробкою, виробництвом, реалізацією, просуванням і продажем комп'ютерних ігор. Вона складається з великої кількості спеціальностей, на яких працюють люди в усьому світі.

Індустрія відеоігор зародилась на території США та Японії у 1970-их роках. За наступні п'ятдесят років ця сфера активно розвивалась та поширювалась по світу. На цей момент ігровий ринок є одним із найбільш перспективних, і розвивається він досить активно. Поява та розвиток сучасних персональних комп'ютерів, консолей та мобільних платформ призвели до еволюції продукції ігрових студій. Також з'явилися нові способи розповсюдження ігор, а саме через онлайн платформи такі, як: Steam, Origin, GOG, Xbox Store, Uplay, Epic Games Store, Battle net, Humble Bundle тощо. Через це ринок почав зростати швидше. За даними на жовтень 2020 року об'єм ринку зріс до 175 мільярдів доларів США, що на 19,6% більше, ніж у попередньому періоді [2]. Пандемія Covid-19 спричинила збільшення частки цифрових покупок.

З рисунку 1 бачимо, що з 2017 по 2022 роки загальний дохід ігрової індустрії збільшився більш, ніж у 2 рази [3]. Це вказує на постійний розвиток та розширення ринку, тобто збільшується кількість виробників якісної продукції, і розширюється аудиторія, яка її споживає. Якщо ж розглянути прогнозовані показники на наступні роки, то робимо висновок про те, що тенденція попередніх років не є випадковою, а індустрія відеоігор

продовжить зростати. Отримані дослідження підтверджують висунуту думку про зростаючу значимість даного сектору.

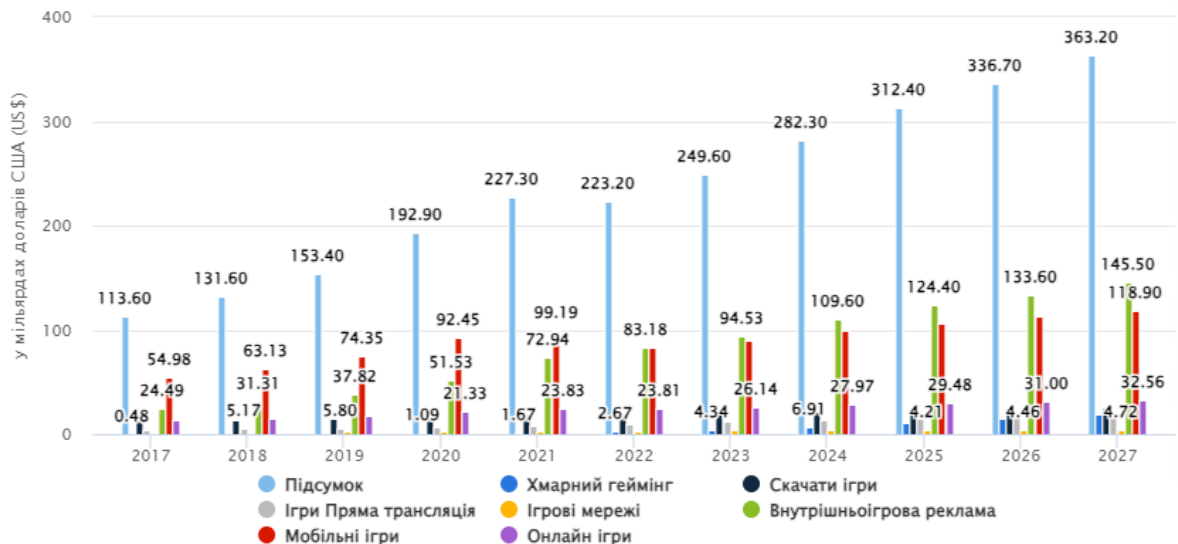


Рисунок 1.1 – Загальний дохід ігрової індустрії у світі за минулий час та прогноз на майбутні роки

Продукт діяльності великих ігрових компаній – відеоігри, які завоювали популярність не лише серед молоді, а й зацікавили також дорослих [4]. Про це свідчить щорічне дорослішання аудиторії. Що й не дивно, адже відеоігри можуть зацікавити своєю різноманітністю (Рисунок1.2).

Існує безліч жанрів ігор, наприклад доросла аудиторія більше зацікавлена в різних іграх таких жанрів, як стратегії та симулятори, тоді як молодь надає перевагу пригодницьким та рольовим іграм. Наведені жанри є одними з основних у сфері виробництва відеоігор. Проте ігри класифікують не лише за жанрами, а й за умовами доступу.

Buy-to-play (придбати для гри) – споживач купує гру один раз, щоб пограти, але тут гра, як правило, продовжує підтримуватися розробником або видавцем, підтримуючи онлайн-сервери або регулярно створюючи новий

контент для гри. Іноді гравцеві може знадобитися придбати доступ до цього нового вмісту, наприклад, через сезонний абонемент [5]. Слід зазначити, що нове наповнення гри розробляється після початку продажів для підтримання інтересу до продукту та залучення додаткових коштів. Buy-to-play модель застосовується для Alan Wake 2, Spider-Man 2, Destiny 2, The Witcher 3: Wild Hunt, The Elder Scrolls Online та інші.

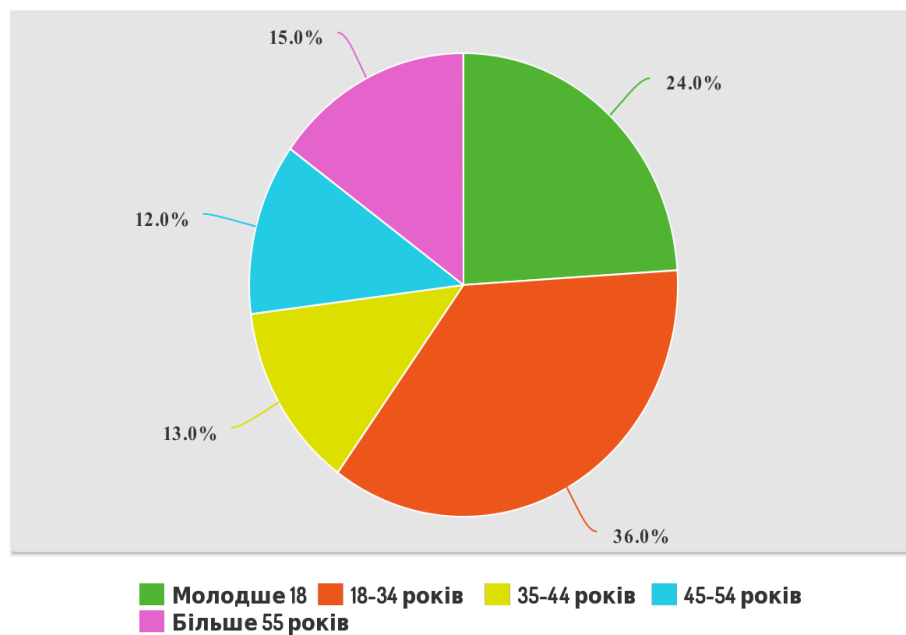


Рисунок 1.2 – Розподіл гравців за віковою категорією на 2022 рік

Free-to-play (безкоштовна гра) – у безкоштовних іграх гравцеві не потрібно купувати гру для того щоб пограти, хоча доступ до деяких функцій і контенту може вимагати придбання підписки або мікротранзакцій. Цю модель застосовують такі проекти, як Fortnite, Apex Legends, Hearthstone та інші.

Pay-to-play (модель підписки) – ігри, які вимагають від гравця регулярної плати за підписку для підтримки доступу до всіх частин гри. Гра, як правило, безкоштовна, щоб дозволити новим користувачам спробувати

гру, але повний доступ вимагає, щоб підписка була сплачена. Прикладам може бути MMORPG Eve Online, World of Warcraft та інші.

Freemium – це такий підтип за якого споживач користується продуктом безкоштовно до певного моменту, тобто гравцю надається безкоштовно тільки демо-версія (free). Щоб продовжити гру він має купити повну версію (premium). Така модель зараз застосовується для деяких мобільних проєктів. Прикладам може бути Plague Inc, Vampire survivor та інші.

Основані на рекламі – ігри, у які зазвичай можна грати безкоштовно, але гравцеві потрібно періодично переглядати рекламу, перш ніж вони зможуть продовжити, а розробник або видавець буде отримувати дохід від реклами. Така модель широко використовується для мобільних проєктів. Прикладом може бути Angry Birds, Cut the Rope та інші.

Ігрова індустрія постійно розвивається, кожного року виникають нові проєкти що захоплюють людей своїми новинками. Розробники відеоігор вкладають у свої проєкти мільйони доларів. Виробники відеоігор урізноманітнюють свою продукцію, представляють новинки на щорічних ігрових трансляціях, заходах (наприклад, Gamescon, MineCon, PAX West-East, Tokyo Game Show, The Game Awards, Game Developers Conference, Summer game Fest, BlizzCon, Starcon та інші). Дохід на ринку відеоігор також зростає за рахунок кіберспорту, що кожного року набуває все більшої популярності.

У світі ігор деякі компанії вражають своїм фінансовим успіхом [6]. Це не просто розробники веселощів для геймерів, а справжні ігрові магнати, які генерують значні доходи та впливають на всю індустрію (Рисунок 1.3).

Компанії, що випускають відеоігри працюють на комерційних засадах, а отже прагнуть максимізації прибутку. Для цього після закінчення розробки кожна компанія-розробник обирає найбільш ефективну модель монетизації свого продукту, його вибір залежить від специфіки розробленої гри.

Роздрібна торгівля – придбання фізичної копії гри у магазині або інтернет-магазині. До цієї моделі можна віднести багато сучасних ігор.

у мільярдах доларів США (US\$)

	2018	2019	2020	2021	2022
Activision Blizzard, Inc.	7.50	6.49	8.09	8.80	8.09
Bandai Namco Holdings Inc.	6.14	6.80	8.33		
Electronic Arts, Inc.	5.54	5.15	6.99		
Microsoft Corp.	110.40	168.10	198.30	110.40	
Nintendo Co., Ltd.	10.87	12.00	16.47	15.45	
Sega Sammy Holdings, Inc.	3.00	3.36	2.60	2.92	
Sony	78.48	75.77	84.28	90.40	
Take-Two Interactive Software, Inc.	2.67	3.09	3.37	3.51	
Tencent Holdings Ltd.	47.26	54.61	69.86	86.85	90.81

Останнє оновлення: червень 2023 р.

Джерело: Statista Company Insights

Рисунок 1.3 – Публічні компанії з найбільшим доходом від ігор

Підписка – підписка на послуги, які надають можливість грати онлайн, надають безкоштовно старі ігри або тимчасовий доступ до бібліотеки з різними проектами. До підписки можна віднести Game Pass від Microsoft, Origin Access, PlayStation Plus, EA Play і EA Play Pro, Ubisoft+, Apple Arcade, Google Play Pass.

Мікротранзакції – клієнт може придбати додаткові елементи гри: ексклюзивні предмети, текстури на зброя, заблокованих героїв тощо. Прикладом застосування виступають Deep Rock Galactic, Don't Starve Together, Overwatch, The Witcher 3: Wild Hunt.

Додаткові матеріали (DLC) – додатковий матеріал у вигляді сюжетних доповнень, тобто гравець окремо докупає частини гри. До ігор з DLC можна віднести Middle-earth: Shadow of War, Half-Life, Warhammer: Vermintide, Bioshock Infinite, DOOM Eternal, The Witcher 3: Wild Hunt.

Коробка зі здобиччю (Loot box) – додаткові ігрові елементи надаються випадково при відкритті коробки-рулетки. До ігор з коробками зі здобиччю

можна віднести Counter-Strike: Global Offensive, Overwatch, Star Wars Battlefront II.

Сезонний абонемент – придбання гри та повного комплекту доповнень. Ця купівля здійснюється зі знижкою у порівнянні із купівлею кожної частини окремо. До ігор з сезонним абонементом відносять Rainbow Six Siege, DOOM Eternal, The Witcher 3: Wild Hunt, L.A. Noire, Mortal Kombat, Anno 1800, Middle-earth: Shadow of Mordor.

Торгівля між гравцями – ігрові предмети та валюта можуть вільно обмінюватись між гравцями на внутрішньо ігровому ринку, це у свою чергу дозволяє видавцю гри отримувати частку від транзакцій, які здійснюють гравці. Найпопулярнішим представником даної моделі є платформа Steam.

Реклама – розміщення реклами всередині гри, яка може приймати різні форми (банерна реклама, реклама за додаткові бонуси, рекламні паузи та розміщення продукту, який рекламують, у грі). До цієї моделі можна віднести більшість сучасних мобільних ігор.

Для кращої візуалізації масштабів розвитку індустрії відеоігор у світі доречно провести порівняльний аналіз доходів, які генерує ця індустрія, з ВВП України та Чеської Республіки (рис.1.4). Виходячи з побудованого графіка, можна побачити, що з 2015 по 2018 рік ВВП України був меншим, ніж дохід від продажу продукції індустрії відеоігор у світі. Однак, якщо порівняти цей показник з ВВП Чехії, то дохід від ігрової індустрії у 2012 році становив трохи більше 25%, а в період з 2016 по 2019 рік - вже більше половини ВВП. Це порівняння підтверджує розширення індустрії відеоігор та її значущість і важливість у світовій економіці.

Необхідно зазначити, що кожній компанії потрібен персонал, який безпосередньо буде займатися розробкою та просуванням продукту [7]. Розвиток індустрії відеоігор зумовлює специфікацію посад та висуває нові вимоги для фахівців на ринку праці, що мають намір працевлаштуватися в компаніях даної сфери.

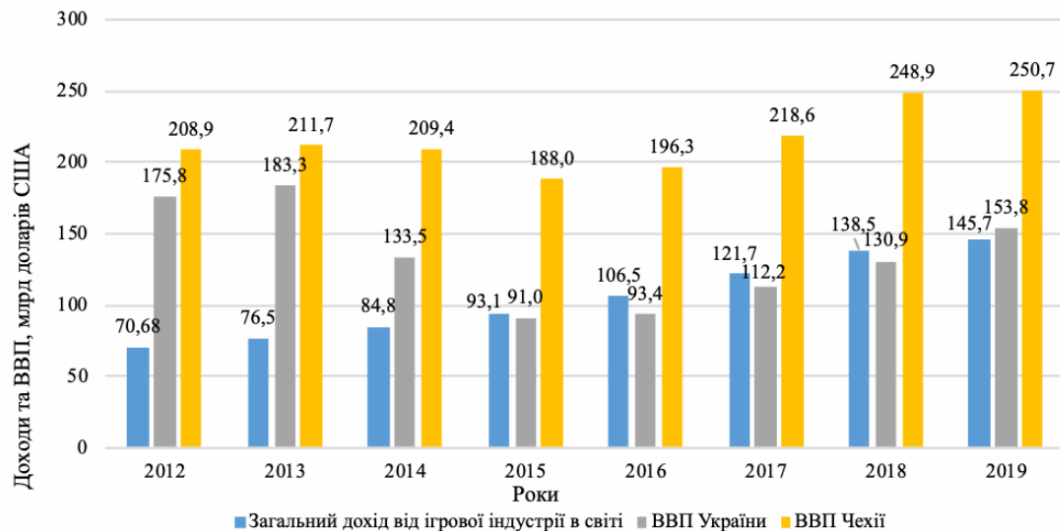


Рисунок 1.4 –ВВП Чехії та України у порівнянні з доходом від ігрової індустрії у світі

Програмісти – особи, які пишуть код, який робить гру придатною для гри на різних ігрових консолях і персональних комп'ютерах. До їхніх обов'язків входить розробка ігрового програмного забезпечення, включення ілюстрацій до гри та налаштування того, як користувачі отримуватимуть доступ до гри та взаємодіятимуть з нею.

Інженери-програмісти – особи, що не тільки займаються розробкою, а й займаються підтримкою та тестуванням програмного обладнання

Художники – для того, щоб повністю реалізувати зовнішній вигляд гри, може знадобитися ціла команда художників відеоігор, які застосовують свої знання та досвід, починаючи від створення кольорової палітри та ігрового середовища і закінчуючи спецефектами, такими як спалахи, дим, шторми, вода чи вогонь. Деякі ігрові художники працюють над перетворенням ескізів на 3D-графіку, інші - над налаштуванням освітлення, щоб зробити віртуальний світ більш реалістичним.

Аніматор – фахівець, який працює у сфері комп'ютерної графіки і візуальних ефектів. Ігрові аніматори поєднують мистецтво та технології для створення інтерактивних анімаційних зображень і середовищ для відеоігор,

тривимірних моделей, анімації персонажів, ефектів та інших візуальних компонентів. Аніматори відповідають за зображення руху та поведінки в грі.

Моделери – фахівці, які займаються моделюванням, створюють моделі для всіх арт-ресурсів у грі – персонажів, зброї, транспортних засобів, меблів, дерев, скель тощо. Вони часто починають з брифу або креслення від концепт-художника і будують на його основі свої моделі.

Геймдизайнери – фахівці, які займаються творчими елементами гри. Це може включати створення сюжетних ліній, персонажів, правил гри, цілей і завдань, а може бути спільним процесом, де людина, яка виконує цю роль, також виконує іншу роль, наприклад, програмування. Ігровий дизайн передбачає розробку ігрового процесу та представлення концепції видавцям і керівництву. Початковий дизайн гри може включати різних дизайнерів, а також сценаристів, які розробляють сюжет, діалоги та текст гри.

Продюсер – фахівець, який керує працівниками, які створюють відеоігри, відіграє керівну роль на кожному етапі розробки та забезпечує якість і своєчасність роботи команд, які працюють над проектом. Роль продюсера буде більше бізнес-орієнтованою, ніж творчою. Продюсер керує бюджетом та забезпечує запуск виробництва за графіком. Продюсери відеоігор підтримують виробничий графік і керують командою розробників. Вони контролюють такі речі, як бюджет, контракти, маркетинг і дотримання термінів виробництва. Фахівець, який забезпечує реалізацію ідеї всіма необхідними ресурсами для релізу готової гри з підтримкою проекту і подальшим постійним зростанням всіх ключових показників

Аналітик ігрових даних вивчає майбутні випуски ігор і аналізує, який вплив вони матимуть, наскільки добре їх приймуть і наскільки добре вони продаватимуться. Аналітики дуже добре вміють аналізувати дані та роблять обґрунтовані прогнози. Ці дані можуть виглядати як попередні замовлення, успіхи подібної гри в минулому, успіхи конкурентів на ринку або навіть популярність жанру гри на момент її випуску. Робота аналітика ігрових даних важлива, оскільки інвестори часто використовують зібрану

інформацію для прийняття рішень про те, які ігрові студії чи видавництва вони можуть надалі фінансувати та підтримувати.

Ігрові продукт-менеджери (GPM) контролюють весь життєвий цикл відеогри, від її задуму до релізу і далі. Вони тісно співпрацюють з різними командами, включаючи геймдизайнерів, розробників, художників і маркетологів, щоб гарантувати, що гра відповідає потребам цільової аудиторії та відповідає баченню компанії.

Звукоінженер – фахівець, який відповідає за додавання реалістичності ігровому світу за допомогою звуків. Кожна гра має власну реальність та індивідуальність, і звукорежисер повинен вміти розуміти та інкапсулювати суть гри у звуковому супроводі, який він створює. Займаються звуковими ефектами, музикою і озвучуванням персонажів

Тестувальники, QA – фахівець, який переконується, що відеогра є зручною для користувача та не містить багів перед її випуском. QA-тестер виявляє будь-які технічні проблеми, тоді як плейтестери більше зосереджуються на наданні зворотного зв'язку щодо користувацького досвіду та зручності гри. Тестувальникам відеоігор часто доручають тестувати певні аспекти гри, і вони діляться своїми висновками за допомогою детальних звітів, у яких зазначають будь-які помилки чи проблеми.

Маркетолог відеоігор відповідає за створення кампаній до, під час і після виходу гри, які будуть привабливими для ігрової аудиторії та охоплять якомога більше релевантних людей.

PR-менеджер – представники компанії, що працюють з журналістами і розміщує рекламу компанії в ЗМІ. Роль менеджера полягає в побудові значущих відносин і підтримці репутації ігрової студії у зовнішньому світі

Ком'юніті-менеджер (КМ) – фахівці, які піклуються про найважливішу частину будь-якої гри - про самих геймерів. У кожній великій гри є фанати - шанувальники, які грають один з одним, спілкуються в соціальних мережах, ходять на турніри і носять футболки. Ком'юніті-менеджери відповідають за

спільноту, яка зростає навколо гри. Ком'юніті-менеджери займаються спілкуванням з аудиторією і допомогою користувачам.

1.2 Порівняльний аналіз популярних ігрових пристроїв

У 2023 році геймери мають обширний вибір ігрових платформ, що відкривають перед ними неймовірні можливості для гри. Ігрова спільнота може вибрати серед трьох основних категорій платформ: мобільних пристроїв, особистих комп'ютерів і ігрових консолей [8].

Геймери мають доступ до широкого асортименту ігор різних жанрів, від епічних відеоігор до інноваційних інтерактивних історій та ігор віртуальної реальності. Завдяки постійному розвитку індустрії, асортимент ігор постійно поповнюється новими виданнями та оновленнями. Крім того, кількість гравців і надалі зростає, і ця тенденція підтримується не лише пандемією COVID-19, але і загальним інтересом до геймінгу (рис.1.5). Геймери продовжують віддавати свій час іграм, розвиваючи індустрію відеоігор та забезпечуючи її стале зростання у 2023 році.

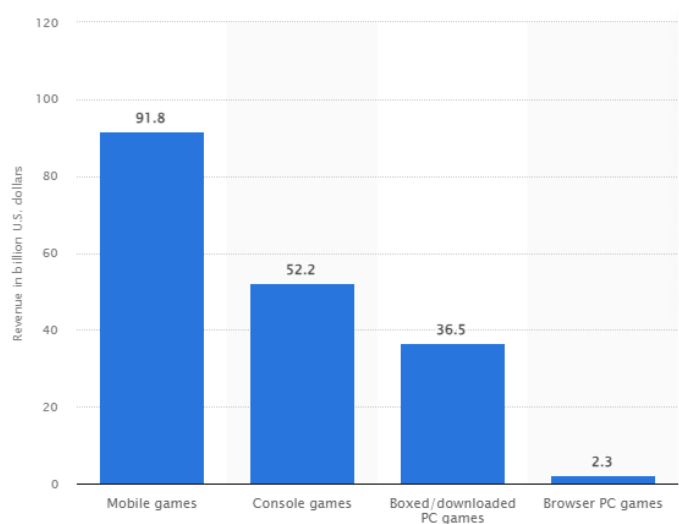


Рисунок 1.5 – Дохід ринку відеоігор у всьому світі у 2022 році за сегментами

У 2023 році кількість користувачів смартфонів по всьому світу продовжує швидко зростати, перевищуючи позначку у 4 мільярди осіб (рис.1.6). Ця надзвичайна кількість людей має необмежений доступ до захоплюючого світу ігор, які завжди на відстані руки і готові до запуску в будь-який час.

Розмови вже давно не йдуть лише про прості аркадні ігри, і навіть класична гра Snake, хоч і залишається в пам'яті багатьох, має статику перед сучасними іграми. Завдяки вражаючій потужності та передовим технологіям сучасних смартфонів, ігри на мобільних пристроях перетворилися на витвори мистецтва, що надають гравцям неймовірно захоплюючий геймплей та графічну якість.

Відомі компанії, такі як Sony, Samsung, Xiaomi, Asus, навіть випускають спеціальні ігрові смартфони з вражаючими технічними характеристиками та додатковими кнопками, розробленими спеціально для геймерів.

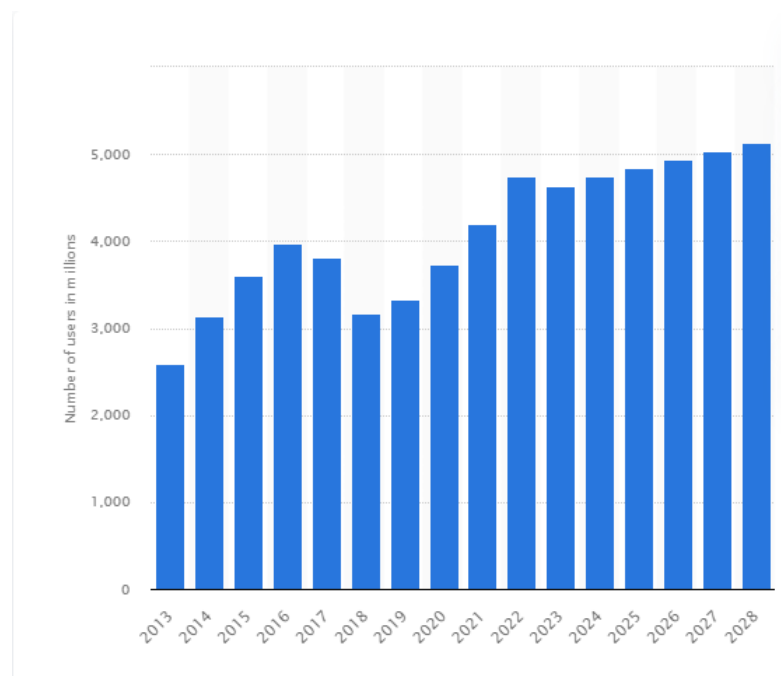


Рисунок 1.6 – Кількість користувачів смартфонів у всьому світі з 2013 по 2028 рік за прогнозами сайту Statista

Більшість ігор, призначених для мобільних пристроїв, можуть бути меншими за обсягом і графічно менш насиченими порівняно з версіями для ПК або ігрових консолей, але це ніяк не впливає на їхню захоплюючість та геймплей. Мобільні пристрої також мають величезний вибір ігор, ідеальних для коротких гейм-сесій або подовжених годин гри (рис.1.7). Ігри в стилі Candy Crush і подібні до них завжди популярні та ідеально підходять для мобільних платформ.



Рисунок 1.7 – Найбільш завантажені мобільні ігри на кінець літа 2023 року

В цілому, мобільні ігри відрізняються від інших платформ більшою портативністю, доступністю та зазвичай меншою вартістю ігор [9]. Незважаючи на те, що вони можуть бути меншими за обсягом та якістю порівняно з іграми для ПК або консолей, їх переваги включають:

- 1) Портативність, яка дозволяє грати в будь-якому місці та в будь-який час.

- 2) Різноманітність жанрів та типів ігор, що задовольняють потреби різних гравців.
- 3) Сприятлива вартість ігор.
- 4) Легкість доступу до ігор через магазини додатків.

У 2023 році багато геймерів залишаються переконаними, що особистий комп'ютер (ПК) – це найкраща ігрова платформа в цілому. Ця платформа зберігає свою привабливість завдяки численним перевагам, які її відрізняють від інших.

Сучасні ПК дозволяють геймерам насолоджуватися неймовірною якістю графіки та обчислювальною потужністю. Граючи на ПК, геймери можуть глибоко налаштовувати ігровий процес під свої уподобання та потреби, що робить цю платформу надзвичайно привабливою.

Можливість створити ПК, який відповідає різним бюджетам і ігровим вимогам, надає гравцям величезну свободу вибору. При цьому Геймери на ПК також мають можливість постійно модернізувати свої системи, щоб відповідати останнім технологічним досягненням або покращувати їх з часом. Це робить ПК гнучкою платформою, яка завжди може підтримувати нові вимоги геймерів.

Геймери можуть піднімати свій ігровий досвід на ПК на будь-який рівень. Це може бути проста гра за стандартними налаштуваннями або захоплюючий процес створення та налаштування складних геймінгових систем.

Окрім ігор, ПК також використовується для великої кількості завдань та додатків, що розширює його корисність. Геймери можуть використовувати ПК для роботи, навчання, відпочинку та багатьох інших цілей. Важливо пам'ятати, що для створення ігрового ПК може знадобитися більше технічних знань та досвіду, порівняно з ігровими консолями.

Загалом, ігри на ПК зазвичай коштують менше, ніж ігри для консолей, і вони мають вищий рівень налаштування та вільності. Проте важливо

враховувати, що деякі ігри все ще є ексклюзивами для консолей, тож вони можуть бути недоступні на ПК.

Отже, переваги ігор на ПК включають:

- Найвищий рівень продуктивності та графічної якості.
- Незмінну можливість налаштування та оновлення.
- Універсальність в застосуванні для різних завдань та потреб геймерів.
- Більш доступні ціни на ігри ніж у консолей.

У 2023 році ринок ігрових консолей залишається однією з найзахоплюючих і популярних галузей розваг. Ігрові консолі — це високо спеціалізовані пристрої, призначені виключно для гри, і вони здатні забезпечити геймерам унікальний і якісний ігровий досвід.

На сьогоднішній день двома основними лідерами на ринку консолей є PlayStation, випущена компанією Sony, та Xbox, розроблена Microsoft. Кожна з цих платформ пропонує свої унікальні можливості і бібліотеку ігор, які притягують геймерів своєю різноманітністю.

Ігри для консолей можна отримати у різних форматах, від фізичних дисків до завантажень з Інтернету. Розробники ігор вкладають багато зусиль у створення якісного відеоконтенту, який гідно відображається на сучасних телевізорах і моніторах.

Системи консолей відрізняються від звичайних ПК. Вони розробляються відповідними виробниками, що гарантує стабільну продуктивність та взаємодію ігор з обладнанням. Це робить використання консолей надзвичайно простим та зручним для користувачів, які не бажають грати на комп'ютерах і налаштовувати апаратну частину.

Ігрові консолі також славляться своїми ексклюзивами - це ігри, які доступні лише на певній платформі. Це можуть бути дуже популярні ігри, які стають основною причиною для геймерів вибирати той чи інший консольний бренд.

Загалом, переваги ігор на консолях включають:

- Стабільну продуктивність та низький рівень технічних проблем.
- Відсутність необхідності у наданні апаратного забезпечення.
- Простоту та зручність в експлуатації.
- Унікальність ігрового контенту завдяки ексклюзивам на певний час.

У 2023 році портативні ігрові пристрої переживають новий розцвіт і стають доволі популярним форматом розваг серед геймерів. Ці пристрої – мініатюрні ігрові консолі, які можна брати з собою куди завгодно.

Портативні ігрові пристрої вражають своєю мобільністю та автономністю. Вони оснащені власними невеликими екранами та акумуляторами, що дозволяє грати в ігри без прив'язки до стаціонарного монітора чи розетки.

Історія портативних ігрових пристроїв налічує десятиліття і включає в себе такі легендарні пристрої, як Game Boy від Nintendo, PlayStation Portable від Sony, а також багато інших. Але поява Nintendo Switch в 2017 році відзначило початок нового етапу в розвитку портативних консолей [10].

Завдяки своєму гібридному дизайну, Nintendo Switch дозволяє грати як вдома на телевізорі, так і в дорозі в режимі переносного пристрою. Його успіх відкрив нові горизонти для портативних ігрових консолей, і багато інших виробників приєдналися до гонитви за цією аудиторією.

Портативні ігри також стають більш доступними завдяки розвитку хмарних ігор та потокової передачі. Гравці можуть насолоджуватися великою кількістю ігор без необхідності завантажувати їх на пристрій, що робить це ще зручніше.

Загалом, портативні ігрові пристрої надають геймерам можливість грати в улюблені ігри в будь-якому місці і в будь-який час, роблячи їх надзвичайно популярними та зручними в сучасному ігровому світі.

1.3 Порівняльна характеристика ігрових движків

Сучасні ігрові движки – це програмні платформи, які надають розробникам ігор необхідний інструментарій для створення вражаючих ігор. Вони об'єднують в собі різноманітні технології та можливості, спрощуючи процес розробки і дозволяючи створювати ігри для різних платформ. Ось кілька ключових аспектів, що стосуються сучасних ігрових движків:

- 1) Графіка і візуалізація – сучасні ігрові движки дозволяють створювати реалістичні графічні образи з використанням різних технік, таких як рейтрейсинг, фізично-підданий рендеринг та високоякісні текстури. Вони підтримують створення вражаючих ігрових світів та персонажів.
- 2) Фізика та анімація – сучасні ігрові движки надають інструменти для реалістичної фізики та анімації, дозволяючи об'єктам та персонажам рухатися та взаємодіяти з оточуючим світом.
- 3) Інтерфейс та користувацький досвід – ігрові движки надають інструменти для створення інтуїтивного ігрового інтерфейсу та динамічного геймплею, що підвищує залученість гравців.
- 4) Спрощена розробка мультиплатформених ігор – сучасні ігрові движки дозволяють розробникам створювати ігри для різних платформ, таких як ПК, консолі, мобільні пристрої та віртуальна реальність, з меншими зусиллями та витратами.
- 5) Спільнота та підтримка – багато сучасних ігрових движків мають активні спільноти розробників та надають багато онлайн-ресурсів, у тому числі документацію, навчальні відео та форуми для обговорення проблем.
- 6) Розвиток ігор в реальному часі – деякі ігрові движки дозволяють розробникам переглядати та тестувати гру в реальному часі, що спрощує ітеративний процес розробки.

- 7) Підтримка мов програмування – сучасні ігрові движки підтримують різні мови програмування, що дозволяє розробникам використовувати ту, з якою вони найкраще володіють.

Сучасні ігрові движки відкривають безмежні можливості для створення різноманітних ігор, від інді-проектів до великих блокбастерів. Вони є суттєвим інструментом для розробників та сприяють постійному розвитку індустрії відеоігор.

Unreal Engine є одним із найвражаючіших ігрових движків у світі. Він розроблений компанією Epic Games і відомий своєю потужною графікою та фізикою, що дозволяє розробникам створювати вражаючі ігрові світи [11]. Unreal Engine підтримує створення ігор як у жанрі 2D, так і 3D, роблячи його відмінним інструментом для різних видів проектів.

Цей движок є безкоштовним для основного використання, але передбачає відрахування від доходів гри після досягнення певного рівня прибутку. Unreal Engine підтримує різні платформи, включаючи ПК, консолі, мобільні пристрої та віртуальну реальність.

Ще однією перевагою Unreal Engine є велика спільнота розробників і багато ресурсів для навчання. Це дозволяє розробникам швидше освоювати цей інструмент та знаходити відповіді на свої питання.

Epic Games вклала значні зусилля в подальше вдосконалення Unreal Engine (рис.1.8). Важливим кроком стала їхня придбана компанія Quixel, яка має величезну бібліотеку реальних зображень, придатних для створення вражаючих анімацій і відеоігор. Користувачі Unreal Engine отримали можливість використовувати пакетні інструменти Quixel, такі як Bridge і Mixer, а також з доступом до усіх ресурсів з бібліотеки Quixel Megascans. Це робить Unreal Engine ще більш привабливим для розробників і дозволяє створювати вражаючі візуальні образи та анімації.

У 2022 році відбувся офіційний запуск Unreal Engine 5, який вже має значний вплив на галузь розробки ігор.

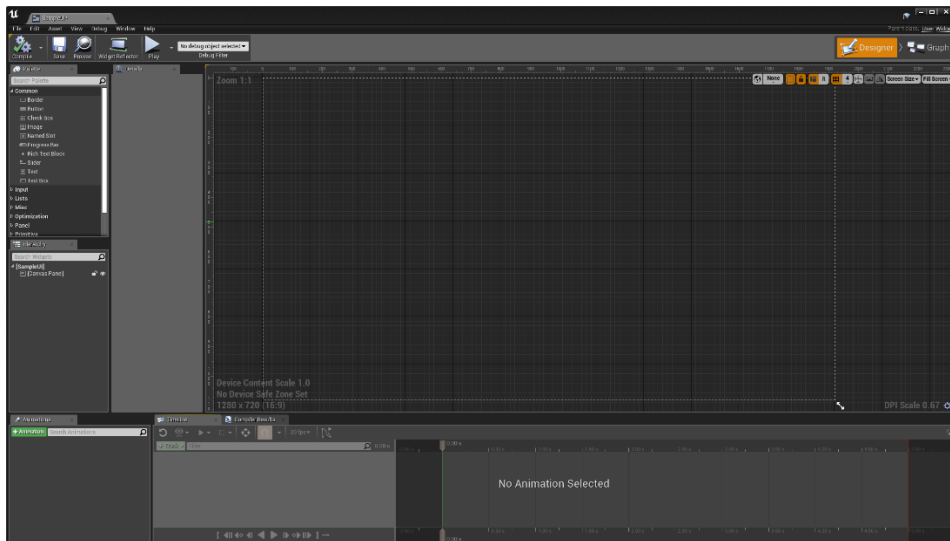


Рисунок 1.8 – Інтерфейс ігрового двигуна Unreal Engine

Більшість ігрових консолей наступного покоління, таких як PlayStation 5 і Xbox Series X, використовують Unreal Engine 5 для підтримки своїх ігор. Цей движок оснащений потужними інструментами, такими як Lumen і Nanite, які забезпечують вражаючу якість графіки та обробку деталей в реальному часі. До того ж, завдяки придбанню Quixel, Unreal Engine може легко імпортувати високо деталізовані матеріали для фотографій та інших ресурсів, що робить його відмінним інструментом для створення захоплюючих ігрових світів.

Плюси:

- Вражаюча графіка та фізика, що дозволяє створювати реалістичні ігри.
- Велика спільнота розробників та багато ресурсів для навчання.
- Вбудована система штучного інтелекту та фізичної симуляції.

Мінуси:

- Вимагає потужного обладнання для ефективної роботи та рендерингу високоякісної графіки.
- Деяка складність у навчанні та використанні.

Unity – це ігровий движок, який славиться своєю доступністю та розширюваністю. Він підходить як для початківців, так і для досвідчених розробників. Unity надає можливість створювати ігри для різних платформ, включаючи ПК, консолі, мобільні пристрої та віртуальну реальність [12].

Універсальний редактор Unity продовжує підтверджувати свою репутацію в ігровій індустрії. Створений для спільної роботи з різними операційними системами, такими як Windows, Mac, Linux, а також платформами, як iOS, Android, Nintendo Switch, Xbox, PlayStation, та інші, Unity спрощує розробку, забезпечуючи доступність для розробників на всіх основних пристроях та операційних системах. Його інтуїтивний інтерфейс робить процес створення ігор більш доступним, зменшуючи необхідність у складних навчаннях та дозволяючи розробникам швидше зануритися в творчість (рис.1.9).

Великий вибір готових ресурсів та інструментів, які регулярно додаються до Unity Asset Store, допомагає розробникам зекономити час і зусилля, прискорюючи розробку та роблячи її більш ефективною. Цей движок має велику кількість готових активів і розширень, що полегшує розробку ігор. Розробники можуть використовувати ці ресурси, щоб швидше створювати ігри і не витратити багато часу на розробку окремих компонентів.

Unity також надає можливість розробки як 2D, так і 3D ігор, що робить його універсальним інструментом для будь-якого типу проекту.

Unity залишається однією з переважних платформ для створення ігор у 2023 році, надаючи розробникам можливість перевести свої творчі задуми в реальність на різних пристроях та платформах.

Плюси:

- Універсальний ігровий движок для різних платформ, включаючи мобільні та VR.
- Багато готових активів і розширень для прискорення розробки.
- Можливість програмування на різних мовах, включаючи C#.

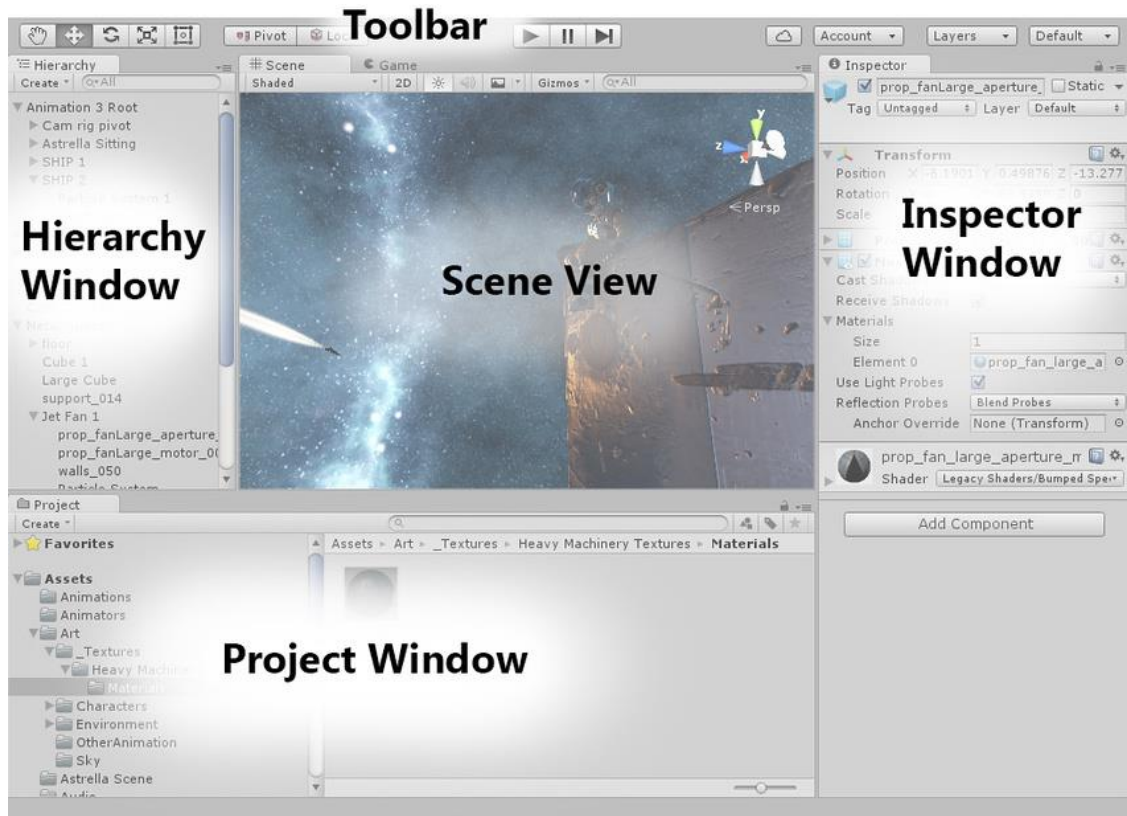


Рисунок 1.9 – Інтерфейс ігрового двигуна Unity

Мінуси:

- Велика кількість розширень може спричинити збільшення обсягу проєкту та завантаження процесора.
- Були відсутні візуальні сценарії. Деякі платні візуальні скрипти можна купити в магазині асетів.

Godot Engine – відкритий кросплатформовий 2D і 3D ігровий двигун під ліцензією MIT, який розробляється спільнотою Godot Engine Community (рис.1.10). До громадського релізу як відкритого ПО двигун використовувався всередині деяких компаній Латинської Америки.

Однією з ключових переваг Godot є його мультиплатформовість. Движок підтримує різні операційні системи, включаючи Windows, Mac, Linux, Android та iOS, а також інші платформи [13]. Це означає, що ви можете розробляти ігри, які працюють на різних пристроях і ОС, не переписуючи код з нуля для кожної платформи. Це зберігає час і ресурси

розробки і дозволяє досягти більшої аудиторії. Він надає інструменти для створення графіки, анімації, фізики та інших компонентів, необхідних для ігор обох типів. Це дає розробникам можливість створювати ігри різних жанрів та стилів, використовуючи один і той же движок.

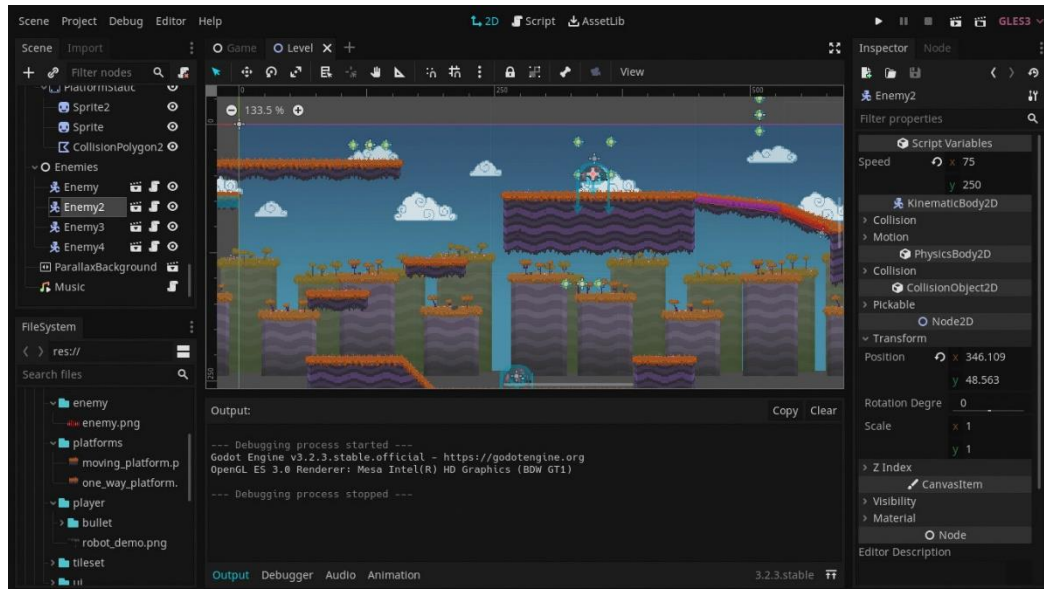


Рисунок 1.10 – Інтерфейс ігрового движуна Godot

Деякі з недоліків Godot включають обмежену кількість готових активів та ресурсів порівняно з іншими популярними ігровими движками. Це може означати, що вам доведеться створювати багато складних компонентів та активів власноруч або шукати сторонні ресурси. Також варто враховувати, що Godot, хоч і швидко розвивається, може бути менш потужним у сфері 3D розробки порівняно з іншими спеціалізованими 3D ігровими движками.

Плюси Godot:

- Godot є абсолютно безкоштовним для використання, що робить його доступним для широкого кола розробників.
- З інтуїтивним інтерфейсом та мовою скрипту GDScript, Godot легко вивчати, особливо для початківців.

- Godot дозволяє створювати ігри для багатьох платформ, що робить його дуже універсальним.

Мінуси Godot:

- Хоча Godot підтримує 3D розробку, він може бути менш потужним, ніж інші спеціалізовані 3D движки.
- Не така велика спільнота, як у Unity чи Unreal Engine. Може бути складно знайти відповіді на свої питання або підтримку на такому рівні, як у великих движках.
- Менше готових ресурсів. В порівнянні з Unity чи Unreal Engine, кількість готових активів та інструментів для Godot може бути обмеженою.

CryEngine – інноваційний ігровий двигун, що вирізняється не лише своєю вражаючою графікою, але і високотехнологічними рішеннями у сфері фізики. Цей двигун дозволяє розробникам створювати вражаючі геймплейні ефекти та реалістичні візуальні враження [14]. Підтримка передових технологій, таких як Ray Tracing, дозволяє досягти рівня реалізму, який зачаровує гравців.

Важливою перевагою CryEngine є велика кількість готових ресурсів, що спрощує процес розробки. Розгалужена бібліотека включає в себе графічні ефекти, моделі персонажів та інші активи, що дозволяє швидко реалізувати творчі ідеї. Можливість використання мови програмування C++ надає розробникам повний контроль над грою, розширюючи їхні можливості (рис.1.11).

Незважаючи на всі плюси, важкість у вивченні та використанні CryEngine може бути викликом для менш досвідчених користувачів. Розробники новачки можуть зіткнутися зі складнощами у розумінні роботи окремих функцій та налаштувань, що впливає на загальний процес розробки.

Однак з розвитком спільноти та активними форумами користувачів, усунення проблем стає більш доступним. Постійні оновлення та підтримка від розробників дозволяють вирішувати проблеми та долучати нові функції.

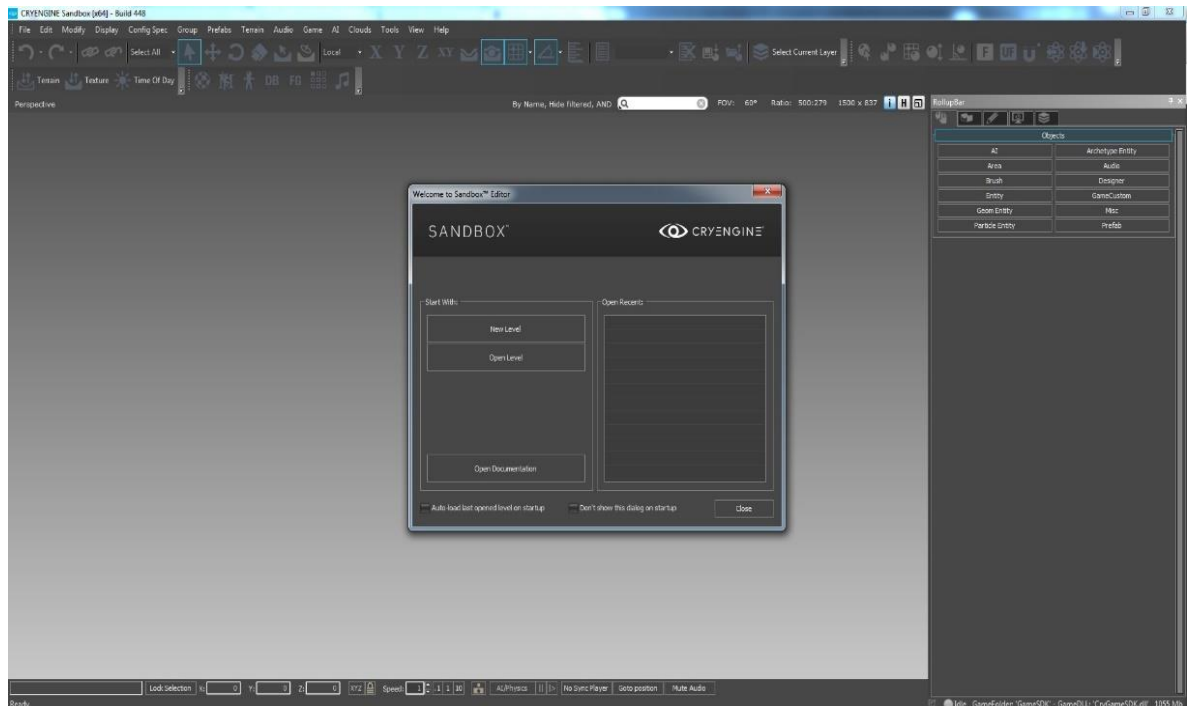


Рисунок 1.11 – Інтерфейс ігрового двигуна CryEngine

Варто відзначити, що для досягнення високої якості графіки, розробники повинні мати великий обсяг ресурсів та відповідне обладнання. Завдяки технічному потенціалу, CryEngine стає ідеальним інструментом для тих, хто прагне створювати геймплейно та візуально захоплюючі ігри високого рівня.

Переваги CryEngine:

- Вражаюча графіка: Двигун славиться своєю високоякісною графікою, створюючи реалістичні ігрові візуальні світи, що вражають користувачів.

- Підтримка передових технологій: CryEngine впроваджує передові технології у галузі графіки та фізики, що відкриває широкі можливості для розробників.
- Багато готових ресурсів: Наявність великої кількості готових активів та ресурсів спрощує розробку та прискорює процес створення ігор.

Недоліки CryEngine:

- Висока складність для новачків: Робота з обширним функціоналом може бути викликом для новачків, що може призвести до тривалого періоду освоєння для ефективного використання.
- Високі вимоги до обладнання: Для досягнення вражаючої якості графіки, CryEngine вимагає потужного обладнання. Це може бути обтяжливим для розробників із обмеженими ресурсами.
- Обмежена спільнота: Спільнота CryEngine, хоча високоякісна, менша порівняно з іншими двигунами, що може ускладнити отримання підтримки та відповідей на питання.

Source Engine, розроблений компанією Valve, є визнаним ігровим двигуном з довговічною історією. Його висока продуктивність і оптимізація дозволяють створювати графічно вражаючі ігри, зберігаючи ефективність. Завдяки широким можливостям геймдизайну розробники можуть імплементувати різноманітні ігрові концепції та механіки.

Інструментарій розробки Source Engine є зручним і ефективним, забезпечуючи розробникам зручність у роботі та інтуїтивність інтерфейсу [15]. Модульність і розширюваність дозволяють легко впроваджувати нові функції та розширювати можливості двигуна.

Багата спільнота та підтримка допомагають розробникам вирішувати проблеми, обмінюватися досвідом і створювати унікальні проекти. Однак Source Engine не є безкоштовним, що може бути обмежуючим для невеликих студій.

Хоча двигун підтримує декілька платформ, його універсальність обмежена порівняно з іншими ігровими двигунами. Також, заснований у 2004 році, двигун може виглядати трохи застарілим у порівнянні з останніми технологічними досягненнями (рис.1.12).

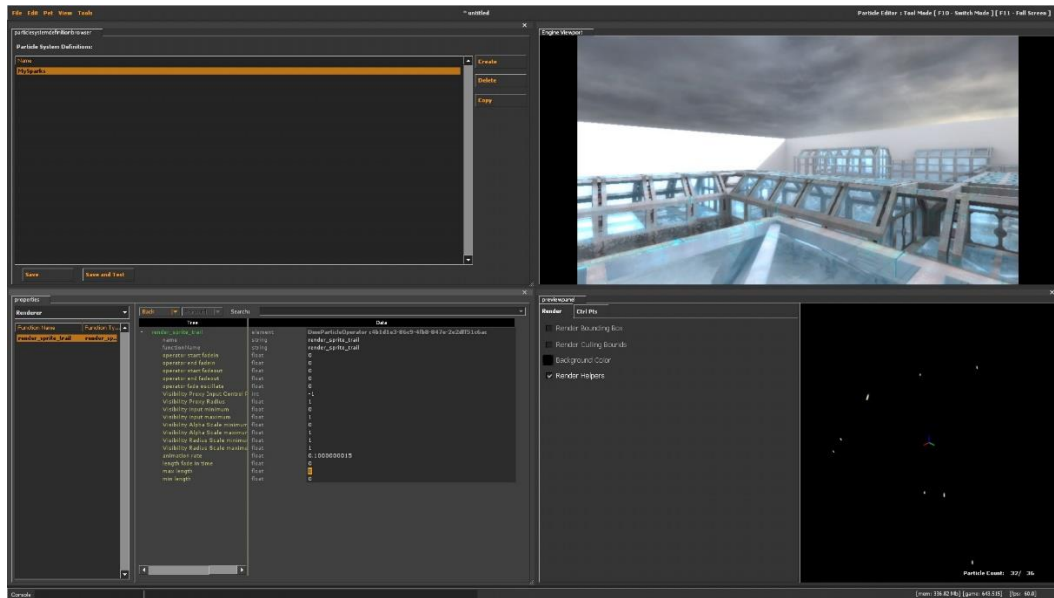


Рисунок 1.12 – Інтерфейс ігрового двигуна Source Engine

Незважаючи на це, Source Engine залишається сильним гравцем у світі розробки ігор, зокрема в жанрі шутерів від першої особи, і продовжує зберігати свою популярність завдяки своїм потужним можливостям та участі в таких культових проектах, як Half-Life і Portal.

Завдяки найновішим оновленням та патчам, Source Engine продовжує адаптуватися до змін у галузі геймдеву, зберігаючи свою конкурентоспроможність. Велика кількість навчальних матеріалів та документації полегшують процес вивчення для новачків, що важливо для залучення молодих розробників.

Плюси Source Engine:

- Source Engine відомий своєю вражаючою графікою і деталізацією, що дозволяє створювати візуально захоплюючі ігри.

- Двигун демонструє високу продуктивність і гарну оптимізацію, що робить його ідеальним для широкого кола пристроїв і користувачів.
- Інструментарій Розробки: Source Engine має потужний та зручний інструментарій розробки, який полегшує процес творення ігор і надає розробникам гнучкість у геймдизайні.

Мінуси Source Engine:

- Використання Source Engine не є безкоштовним, що може становити високі витрати для менших студій та незалежних розробників.
- Хоча двигун підтримує декілька платформ, він може бути менш універсальним порівняно з іншими конкурентами.
- Заснований у 2004 році, Source Engine може виглядати застарілим у порівнянні з новітніми технологіями, що може обмежувати його можливості в ряді аспектів.

Amazon Lumberyard є потужним ігровим двигуном, розробленим Amazon для створення якісних ігор. Заснований на CryEngine, Lumberyard привносить високий рівень графіки та фізики (рис.1.13). Його інтеграція з AWS дозволяє розробникам використовувати високопродуктивні та масштабовані хмарні рішення для гри.

Lumberyard підтримує як 3D, так і VR-розробку, що робить його універсальним для різних типів ігор. Інтуїтивний інтерфейс спрощує роботу, а функціональність мови Lua дозволяє створювати скрипти для геймплею [16]. Ще однією перевагою є широкий набір готових активів і ресурсів, що прискорює процес розробки. Отримавши безкоштовний доступ до коду, розробники можуть налаштовувати двигун під свої потреби, зберігаючи повний контроль над ситуацією.

Недоліками можуть бути високі вимоги до обладнання та обмежена документація порівняно з іншими популярними двигунами. Хоча Lumberyard продовжує розвиватися, менша кількість користувачів порівняно з іншими гравцями галузі може впливати на швидкість розвитку

спільноти та підтримки. Загалом, Lumberyard є потужним інструментом для серйозних розробників ігор, шукаючих високу якість і гнучкість у розробці.

Ще однією значущою особливістю Lumberyard є його інтеграція з різними сервісами Amazon, що дозволяє використовувати широкий спектр інструментів для монетизації та аналізу ігор. Це включає в себе можливість використовувати Amazon GameLift для створення масштабованих серверних архітектур для мультиплеєрних ігор та Amazon S3 для зберігання та управління контентом гри.

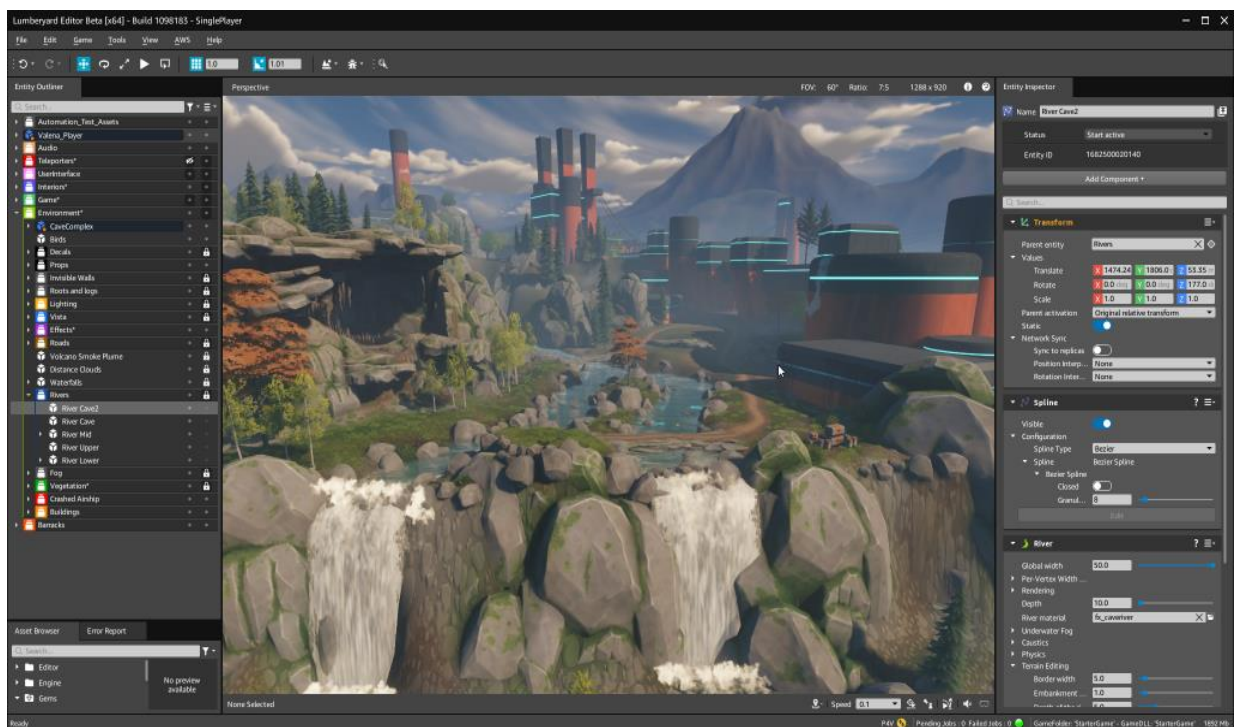


Рисунок 1.13 – Інтерфейс ігрового двигуна Lumberyard

Багатофункціональність ігрового двигуна також дозволяє впроваджувати складні механіки гри, такі як штучний інтелект, системи анімації та фізики, що робить Lumberyard конкурентоспроможним для розробки різноманітних ігрових жанрів.

Однак важливо відзначити, що високі технічні вимоги можуть виявитися викликом для менших студій чи незалежних розробників.

Повсякденна актуальність інструменту передбачає постійне вдосконалення в різних аспектах, оскільки галузь ігрової розробки швидко розвивається. Незважаючи на це, Amazon Lumberyard лишається перспективним вибором для комерційних та амбіційних проектів в ігровій індустрії.

Плюси Amazon Lumberyard:

- Заснований на CryEngine, Lumberyard привносить високий рівень графіки та фізичної взаємодії, надаючи розробникам інструменти для створення вражаючих ігор.
- Інтеграція з Amazon Web Services (AWS) дозволяє використовувати високопродуктивні та масштабовані хмарні рішення, що сприяє ефективній розробці та вдосконаленню гри.
- Lumberyard є універсальним інструментом, оскільки підтримує як 3D, так і VR-розробку, розширюючи можливості для різних типів ігор.

Мінуси Amazon Lumberyard:

- Використання Lumberyard може виявитися викликом для менших студій чи незалежних розробників через високі технічні вимоги.
- Для того щоб залишатися в курсі та використовувати всі переваги Amazon Lumberyard, розробникам доводиться постійно адаптуватися до нових можливостей та вивчати оновлення інструменту, що може стати додатковим завданням у робочому процесі.
- Порівняно з іншими популярними двигунами, Lumberyard має обмежену документацію та меншу спільноту, що може ускладнити отримання підтримки та вирішення проблем.

1.4 Аналіз операційних систем для ігор

Протягом останніх років ігрова індустрія пройшла значний шлях, а вибір операційної системи може значно вплинути на ваш ігровий досвід.

Варіанти, такі як Windows, MacOS та Linux, залишаються популярними серед геймерів, і кожен з них має свої особливості і можливості (рис.1.14).



Рисунок 1.14 – Три найпопулярніші ОС на ПК

Windows є найпоширенішою ОС і пропонує найбільшу бібліотеку ігор, включаючи як ексклюзивні для ПК, так і мультиплатформенні ігри. Це робить її ідеальним вибором як для казуальних, так і для серйозних геймерів, які хочуть мати доступ до найновіших ігор і великий вибір ігор на вибір. Крім того, Windows сумісна з більшістю ігрового обладнання та підтримує широкий спектр програмного забезпечення, що робить її універсальним варіантом для ігор.

З точки зору продуктивності, Windows 10 і 11 пропонують надійну продуктивність навіть у найвимогливіших іграх. З випуском Windows 10 була представлена підтримка DirectX 12, яка дозволяє оптимізувати роботу графічного та центрального процесорів для підвищення продуктивності в іграх. Відмінна підтримка нових ігор та програм для Windows забезпечує плавний ігровий досвід.

Поважну роль в виборі робить велика кількість ігрових платформ і магазинів, які належать до Windows. Будь-який геймер може знайти ігри на свій смак в Steam, Epic Games Store, Origin, або інших платформах, і

отримувати доступ до нового ігрового контенту, будь-якій грі. Зручність Windows, що відзначається простим процесом встановлення та користування іграми, зробила її популярною серед геймерів.

Windows залишається вразливою до вірусів і шкідливого програмного забезпечення, і безпека стає найважливішим питанням для гравців, які ставлять свою систему під загрозу. Оновлення і використання антивірусного програмного забезпечення є обов'язковими кроками для гарантування безпеки.

Загалом, Windows залишається найпопулярнішою операційною системою для геймерів завдяки величезній кількості ігор, високій продуктивності та зручності користування. Однак безпека залишається питанням уваги, і геймерам слід бути обережними, щоб захистити свою систему від потенційних загроз, а також вона все ще може мати проблеми з вимогливими іграми, і для роботи на максимальних налаштуваннях може знадобитися система високого класу.

MacOS, завдяки своєму елегантному дизайну та інтуїтивному інтерфейсу, залишається привабливою опцією для користувачів, які прагнуть естетичного задоволення та зручності в щоденному користуванні. Популярність цієї операційної системи серед казуальних гравців та тих, хто цінує дизайн, не поступається Windows.

Хоча бібліотека ігор MacOS досі менша порівняно з Windows, вона постійно поповнюється популярними іграми, включаючи інді-хіти, які здобули визнання критиків. Для казуальних гравців, які не вимагають безлічі ігор на вибір, macOS може бути гарним вибором.

MacOS також відома своєю високою продуктивністю та стабільністю, але її вимоги до апаратного забезпечення можуть бути більш вимогливими, і вона може бути дорожчою порівняно з Windows. MacOS оптимізовано для обладнання Apple, що робить її найкращим вибором для тих, хто віддає перевагу безперебійному та інтегрованому ігровому процесу.

Безумовно, безпека є однією з сильних сторін macOS. Ця операційна система славиться своєю безпекою та захистом від загроз, і це робить її привабливою для гравців, які ставлять на перший план безпеку своїх пристроїв. На відміну від Windows, де користувачам доводиться більше стежити за безпекою, macOS забезпечує вищий рівень захисту, зменшуючи ризики впливу вірусів і шкідливого програмного забезпечення.

Однак macOS має обмежені можливості для кастомізації, а її сумісність з ігровим обладнанням і програмним забезпеченням може бути більш обмеженою порівняно з Windows. Користувачі macOS також мають доступ до популярних ігрових платформ, таких як Steam, але вибір доступних ігор може бути більш обмеженим порівняно з Windows. З точки зору продуктивності, macOS забезпечує плавний ігровий досвід для більшості ігор, але вона може мати проблеми з вимогливими іграми і може вимагати високопродуктивної системи для роботи на максимальних налаштуваннях.

Linux – це операційна система з відкритим вихідним кодом, яка рідше використовується для ігор, але вона має зростаючу бібліотеку ігор, включаючи кілька популярних назв. Продуктивність геймінгу на Linux продовжує поліпшуватися завдяки розвитку графічних бібліотек та оптимізації драйверів. Хоча деякі геймери все ще можуть спостерігати різницю в продуктивності порівняно з Windows, особливо в іграх із високими графічними вимогами, Linux надає значно кращий ігровий досвід, ніж колись.

Linux також легко налаштовується і пропонує кращу продуктивність і стабільність порівняно з іншими ОС, що робить його привабливим варіантом для технічно підкованих геймерів. Linux відома як більш безпечна та ефективна порівняно з Windows, але вона має більш круту криву навчання і може вимагати більше технічних знань для налаштування та використання.

З точки зору ігор, Linux має меншу бібліотеку ігор порівняно з Windows і macOS, але вона пропонує доступ до популярних платформ, таких як Steam і GOG, і її сумісність з ігровим обладнанням і програмним

забезпеченням покращується. Linux також пропонує більш стабільний та ефективний ігровий досвід, але його продуктивність все ще може змінюватися залежно від системи, на якій він встановлений. Користувачам Linux також може знадобитися інвестувати в шари сумісності, такі як Wine або Proton, щоб грати в деякі ігри, призначені лише для Windows. З погляду ціни, Linux залишається найбільш доступною опцією. Операційна система безкоштовна, і ви можете встановити її на різні пристрої без додаткових витрат.

Отже, не існує єдиного переможця, який би підходив усім користувачам, найкраща ОС для ігор в кінцевому підсумку зводиться до особистих уподобань та типу ігор, в які людина хоче грати [17]. Windows пропонує найбільшу бібліотеку ігор і сумісність з широким спектром обладнання, що робить її найкращим вибором як для казуальних, так і для серйозних геймерів. MacOS – хороший вибір для тих, хто надає перевагу естетиці та зручному інтерфейсу, тоді як Linux ідеально підходить для технічно підкованих геймерів, які цінують продуктивність та стабільність.

2 ТЕХНОЛОГІЇ РОЗРОБКИ ІГРОВИХ ПРОДУКТІВ

У розробці ігрових проектів у 2023 році стає очевидним, що успішний шлях до реалізації продукту полягає не лише в урахуванні концепційних аспектів, але і в гнучкому адаптивному підході до динамічної індустрії. Розробники активно використовують концепцію "пайплайну", створюючи віртуальний конвеєр ідей для швидкого перетворення концепцій в реальні продукти.

Пайплайн (від англійського pipeline – “трубопровід”) – це документ, що візуалізує процес розробки продукту. Він являє собою послідовність етапів, розташованих так, що кінець попереднього є початком наступного. Хоча пайплайни різняться між проектами та студіями, процес досить схожий, незалежно від того, чи працює розробник над AAA, інді-грою, чи мобільною грою.

Процес розробки гри стає все більш гнучким і може еволюціонувати під час власного розвитку. Важливо враховувати, що нові ідеї можуть з'являтися в будь-який момент, перевертаючи попередньо визначену концепцію гри. Такий динамічний підхід вимагає визначення чітких етапів розробки для забезпечення стабільності та ефективності (рис.2.1).

Сучасні розробники також використовують поняття пайплайну для організації робочих процесів. Цей віртуальний конвеєр ідей стає фундаментом для автоматизації та використання інноваційних інструментів, що дозволяють прискорити розробку та гарантувати високу якість гри [18].

У контексті розробки ігор стає дедалі важливіше використання метрик для контролю кожного етапу. Від часу процесу та часу виконання до часу затримки та відсотку виконаних робіт без переробки - ці параметри стають ключовими для ефективного керування та уникнення можливих затримок.

Сучасний підхід до розробки ігор визначається не лише технічними аспектами, але й глибоким розумінням потреб гравців та динаміки ринку.

Гнучкий підхід дозволяє ефективно взаємодіяти з постійно мінливими умовами та негайно реагувати на вимоги споживачів.

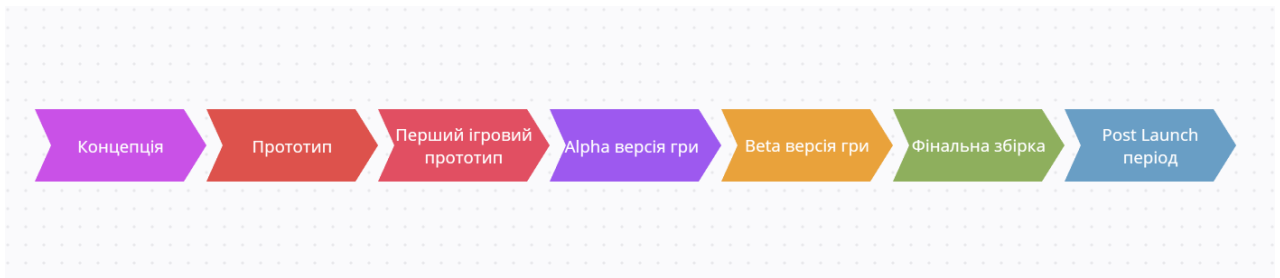


Рисунок 2.1 – Пайплайн розробки ігри

Етап концепції включає в себе глибокий аналіз ринку ігор, де розробники стежать за трендами та виявляють популярні жанри серед гравців. Додатково, вони проводять дослідження цільової аудиторії, враховуючи їхні вподобання та очікування. Однією з ключових задач є сформулювати унікальну концепцію гри, яка відрізняється від інших на ринку, привертає увагу та відповідає потребам гравців.

Після визначення загальної концепції, розробники розгортають стратегію використання унікальних механік гри, що може включати в себе новаторські геймплейні рішення чи інтерактивний світ. Важливо також визначити головних персонажів, їхні характери та розвиток, створюючи зміст, який гравці будуть знаходити захоплюючим.

У процесі розробки концепції, команда активно спілкується між собою, обговорюючи ідеї, вносячи поправки та забезпечуючи загальне бачення гри. Крім того, вони можуть залучати фахівців з інших галузей, таких як художники чи письменники, для забезпечення комплексного та якісного підходу до концепції.

Остаточним етапом концепції є створення детального опису гри, який включає в себе всі аспекти, починаючи від загальної сюжетної лінії та закінчуючи конкретними елементами геймплею. Цей документ служить

основним посібником для подальших етапів розробки та допомагає всій команді спільно працювати над спільною метою.

Створення прототипу – це ключовий крок, де абстрактні ідеї концепції набувають конкретної форми. Розробники використовують спеціальні інструменти та програми для візуалізації інтерфейсу, геймплейних механік та основних елементів гри. Це може включати в себе створення макетів рівнів, дизайн персонажів, анімаційні прототипи та інші важливі аспекти гри.

Прототип дозволяє команді розробників експериментувати з різними концепціями, тестувати їх ефективність та забезпечити спільний розуміння того, як повинна виглядати гра в майбутньому. Важливо враховувати фідбек від всієї команди та, можливо, від зовнішніх консультантів чи тестерів, для максимальної об'єктивності та вдосконалення ідеї гри.

Додатково, прототип може слугувати як інструмент для презентації проекту перед інвесторами чи партнерами. Чітке відображення можливостей гри може значно полегшити залучення фінансування та партнерських угод.

Перший ігровий прототип (FPP – first playable prototype) виходить на наступний рівень, представляючи собою вже відносно повноцінний варіант гри. Розробники створюють базові рівні, систему управління та додаткові функції, які можна перевірити в контексті гри.

На цьому етапі розробники можуть виявити проблеми, які не були очевидними на етапі прототипу, і вносять необхідні зміни. Також вони мають можливість долучити перших тестерів, щоб отримати повноцінний зворотній зв'язок щодо геймплею, багів та інших аспектів гри.

Перший ігровий прототип дозволяє розробникам відобразити ключові аспекти гри, її можливості та шляхи для майбутнього розвитку. Він також може служити інструментом для просування гри, привернення уваги гравців та створення очікування щодо її релізу.

Alpha версія гри є першою повноцінною ігровою версією, яка включає всі заплановані функції та елементи. Розробники фокусуються на тому, щоб гра була готовою для внутрішнього тестування та перевірки. Основна мета -

виявити та виправити всі технічні та геймплейні проблеми, які можуть вплинути на якість гри перед тим, як вона потрапить в руки зовнішніх тестерів чи гравців.

Окрім технічних аспектів, розробники можуть вносити корективи в естетичний бік гри, вдосконалюючи графічний дизайн та звукове супроводження. Це також час для вдосконалення існуючих геймплейних елементів на основі фідбеку тестерів, з метою досягнення оптимального балансу та задоволення гравців.

Alpha версія є важливим етапом, оскільки вона визначає, наскільки готова гра до публічного випуску. Зазвичай вона випускається для внутрішнього кола тестування, такого як команда розробників чи обраних тестерів, які надають зворотний зв'язок та допомагають виправити помилки та недоліки.

Beta версія гри – це етап, коли гра вперше стикається з реальною аудиторією. Розробники випускають її для обраних тестерів або навіть відкрито для гравців, які можуть випробувати гру перед остаточним випуском. Головна мета – отримати відгуки та виявити останні проблеми перед тим, як гра буде доступна для широкого загалу.

На цьому етапі розробники активно взаємодіють із спільнотою гравців, аналізуючи їхні відгуки, відстежуючи поведінку в грі та вносячи необхідні зміни. Beta тестування допомагає зрозуміти, як гравці реагують на гру в реальних умовах, які аспекти їм подобаються, а які можуть бути вдосконалені.

Після закінчення бета тестування розробники роблять останні корекції, зокрема виправлення багів, збалансування геймплею та оптимізацію продукту. Важливо враховувати фідбек гравців, оскільки вони можуть надати цінну інформацію щодо того, як покращити гру перед фінальним випуском.

Фінальна збірка гри – це нарешті повна та завершена версія продукту, готова до випуску. Розробники ретельно перевіряють всі аспекти гри,

включаючи геймплей, графіку, звук та інші технічні аспекти. Вони впевнюються, що всі баги виправлені, і гра готова вразити своїх гравців.

На фінальному етапі розробники можуть також працювати над додатковими вмістом чи режимами гри, які можуть бути випущені в майбутніх оновленнях. Важливо, щоб гра була не просто готовою до випуску, але й вражала своєю унікальністю та якістю.

Після випуску гри на ринок настає Post Launch період, коли розробники зосереджуються на підтримці та розвитку продукту. Основна увага приділяється вивченню відгуків та реакції аудиторії на гру. Розробники вносять постійні покращення через оновлення, виправляють невеликі баги та додають новий контент для збереження інтересу гравців.

Post Launch період також може включати в себе стратегії маркетингу, акції та рекламні події, щоб привернути нових гравців та утримати існуючих. Взаємодія з гравцями через форуми, соціальні мережі та інші канали комунікації стає важливою для розуміння їхніх потреб та надання змістовного відгуку.

У Post Launch періоді розробники також можуть вивчати аналітику, досліджувати поведінку гравців та вносити стратегічні зміни для підтримки та розвитку гри в майбутньому. Це також відмінний момент для презентації вашого ігрового продукту потенційним інвесторам. Маючи на руках робочий прототип з відгуками користувачів та планом на майбутнє, ви можете презентувати свою програму як перспективний бізнес-проект, а не просто як ідею.

3 ПРОЕКТУВАННЯ ГРИ

3.1 Визначення вимог до програмного забезпечення та розробка технічного завдання

Технічне завдання (ТЗ) встановлює параметри, методи і структуру проекту. Воно формується на основі ретельного аналізу та досліджень, враховуючи відмінності клієнтських вимог і особливостей бізнес-процесів у всіх деталях [19]. Іншими словами, це детальний гайд, який визначає чіткі та послідовні кроки у створенні продукту, а також обладнання і технології, які будуть використовуватися, передбачаючи очікувані результати.

У зв'язку з тим, що цей проект є відеогрою, визначено лише одну категорію користувачів – гравці. Протягом гри учасник активно взаємодіє з ігровим процесом, впливаючи безпосередньо на його події.

Гра повинна дотримуватися таких правил:

- Головний персонаж може пересуватися вліво та вправо, стрибати та взаємодіяти з об'єктами та NPC.
- У грі є головне меню, меню паузи та меню налаштувань.
- На карті розташовані вороги, яких потрібно обходити чи обманювати, заводити у пастки.
- Персонаж може збирати об'єкти, такі як монети та ключі.
- Існують об'єкти, з якими можна взаємодіяти, наприклад, ящики, скрині та батуту.
- На полі розташовані пастки та прірви, які потрібно уникати.
- Інтерфейс гри відображає поточне здоров'я персонажа, кількість монет, ключів та врятованих звірів.
- Мета гри — врятувати певну кількість звірів із клітин та залишити рівень для його завершення.
- У випадку невдалого проходження рівня, надати можливість почати його спочатку, інформуючи гравця відповідним повідомленням.

Функціональна модель програмного застосунку представлена у вигляді UML-діаграми варіантів використання. Основним актором виступає користувач у ролі гравця. Діаграму наведено на рисунку 3.1.

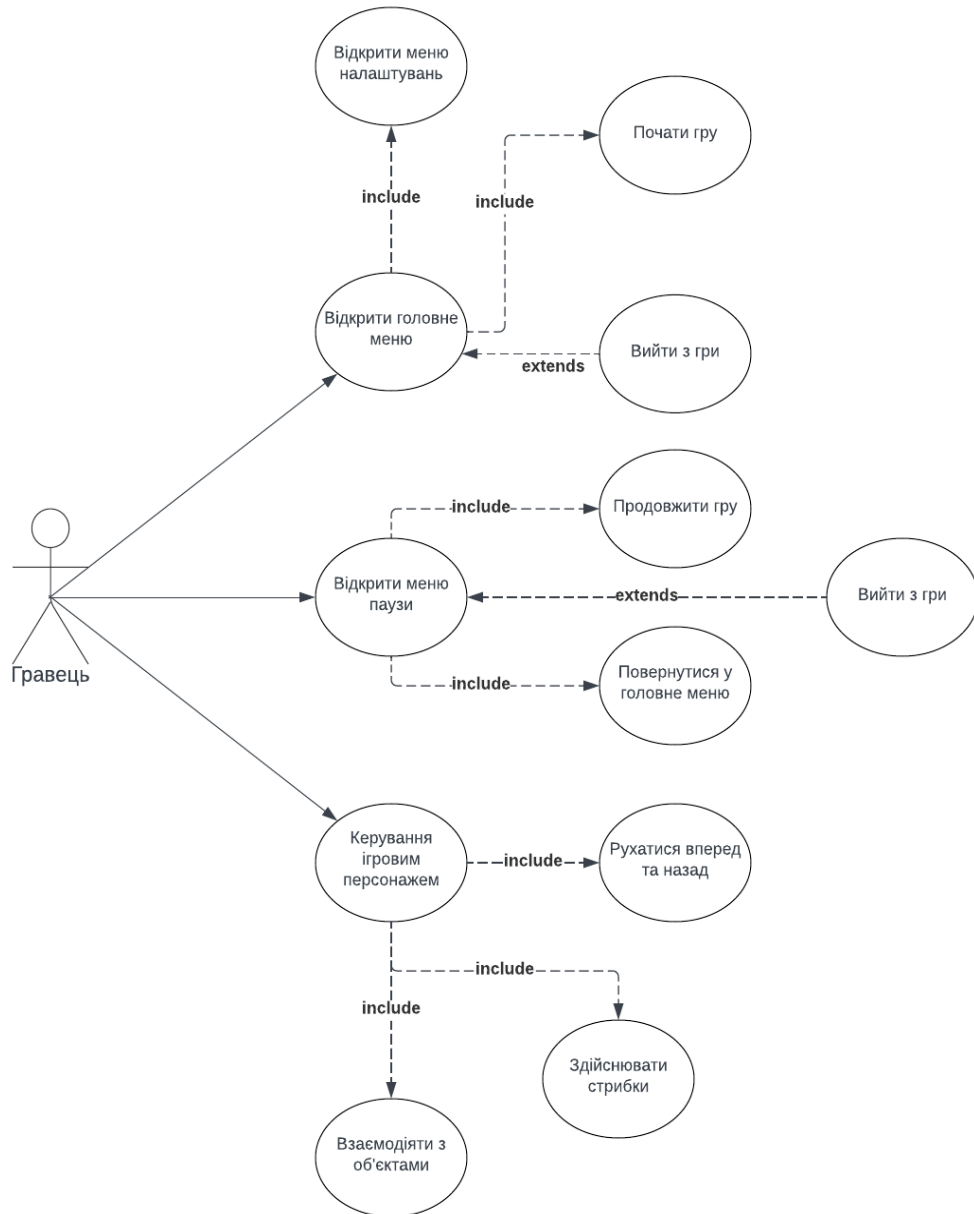


Рисунок 3.1 – Діаграма варіантів використання продукту

Розглянемо вимоги до апаратного забезпечення.

Для функціонування програми на ПК – повинна бути операційна система: Windows 7/Windows 8/Windows 10. Також потрібен бути встановлений DirectX 9.0 або пізніша версія.

Мінімальні характеристики:

- ОС: Windows 7 або вище
- Процесор: Intel Core 2 Duo 2.1 ghz or equivalent
- Оперативна пам'ять: 2 GB ОЗУ
- Відеокарта: 2nd Generation Intel Core HD Graphics (2000/3000), 512MB
- DirectX: версії 9.0 або вище
- Місце на диску: 100 MB

3.2 Проектування архітектури програмного забезпечення

Для подальшої розробки програмного забезпечення необхідно змодельовати його архітектуру та процеси [20]. З цією метою було розроблено схему процесу проходження ігрового застосунку дипломного проектування у нотації BPMN (Business Process Model and Notation) – модель та позначення бізнес-процесів (рис.3.2).

Опис процесу проходження ігрового застосунку:

- користувач запускає ігровий застосунок;
- після запуску програмного забезпечення на екрані відображається головне меню;
- користувач починає гру за допомогою відповідної кнопки;
- гра завантажує першу сцену з навчанням у початку;
- гравець управляє ігровим персонажем за допомогою відповідних кнопок на клавіатурі;
- гравець збирає усі потрібні об'єкти у рівні та рятує звірів з кліток, після цього йде до виходу;

- після завершення усіх рівнів гра завантажує фінальну сцену з подякою та можливістю виходу до головного меню або з гри;
- якщо не всі звіри були врятовані на рівні то користувач отримає повідомлення біля виходу що треба врятувати усіх;
- гравець збирає усі потрібні об'єкти у рівні та шукає і рятує звірів що лишилися, після цього йде до виходу.

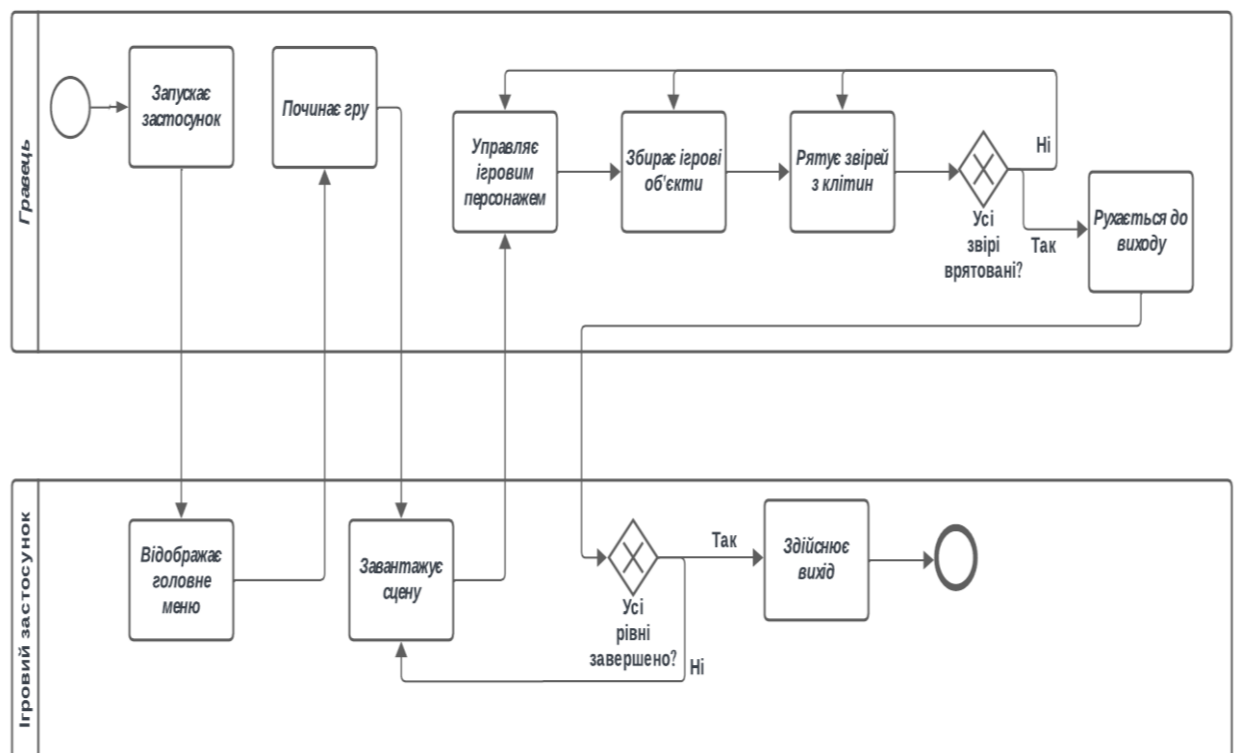


Рисунок 3.2 – BPMN-діаграма проходження ігрового застосунку

3.3 Підготовка графічних елементів

Піксельна графіка є популярним вибором серед інді-розробників ігор та осіб, які створюють 2D-ігри з обмеженими ресурсами [21]. Нижче наведено деякі з переваг та недоліків використання піксельної графіки у розробці ігор.

Переваги використання піксельної графіки:

- легкість у створенні: піксельна графіка вимагає дуже мало ресурсів і може бути створена за допомогою простого програмного забезпечення та інструментів;
- низька роздільна здатність: оскільки піксельна графіка використовує низьку роздільну здатність, загальний розмір гри може бути меншим, ніж у тих, що використовують графіку з високою роздільною здатністю.

Недоліки піксельної графіки:

- обмежена деталізація: через малий розмір кожного пікселя може бути складно створювати деталізовану графіку за допомогою піксельного мистецтва;
- обмежена масштабованість: піксельне мистецтво може погано масштабуватися при перегляді на великих екранах або пристроях.

Будь-який засіб, який дозволяє розміщувати квадрати на сітці, може служити потенційним піксельним редактором. Наприклад, Adobe Photoshop має всі необхідні функції для створення піксельних зображень, дозволяє вирівнювати роботу за піксельною сіткою та надає детальне керування, необхідне для створення ефектних зображень.

В будь-якому програмному забезпеченні вам доведеться часто використовувати вбудовані базові інструменти, такі як:

- Олівець. Для малювання пікселів.
- Лупа. Для збільшення та зменшення масштабу.
- Ластик. Для стирання зайвих пікселів.
- Палітра. Для вибору кольорів.

Варто уникати зберігання піксельної графіки у форматі JPG. Замість цього рекомендується використовувати формати PNG або GIF. Це загальноприйнятий тип файлу, і його часто встановлюють за замовчуванням, проте стиснення, яке використовується у форматі JPG,

може негативно вплинути на якість піксельної графіки та порушити крок за кроком процес роботи художника з пікселями.

Палітри для рисування в програмі Photoshop – це визначені набори кольорів, які використовуються художниками для створення своїх творінь. Вони можуть бути створені самим художником або використовувати готові палітри, які відображають специфічні гармонії кольорів, теми або стилі.

Фотошоп надає різноманітні інструменти для керування палітрою, включаючи можливість вибору, збереження та завантаження кольорів. Користувачі можуть вибрати кольори зі стандартної колірної моделі, такої як RGB (червоний, зелений, синій), або використовувати інші моделі, які відображають тон, насиченість та яскравість.

Палітри стають важливим інструментом для творчого процесу, допомагаючи художникам обирати та комбінувати кольори з метою досягнення певного ефекту, стилю або настрою у своїх роботах.

Для створення власної палітри кольорів можна скористатися сайтом coolors.co, де є можливість згенерувати випадкову палітру або обрати одну з тисячі вже створених. Обрану або створену палітру можна легко експортувати до будь-якого графічного редактора.

Іншим корисним ресурсом є Lospec, де доступно понад 2000 піксельних палітр для завантаження. Під більшістю палітр наведено зображення у відповідних кольорах, щоб художник міг на практиці оцінити, як ці кольори виглядатимуть у його роботі (рис.3.3).

Lospec пропонує різноманітні інструменти для створення та редагування графіки у стилі піксель-арту, включаючи генератори палітр, інструменти для конвертації та редагування зображень.

Спільнота Lospec надає доступ до широкого вибору палітр, що дозволяє художникам швидко вибрати та використовувати підходящі кольорові схеми для своїх робіт.

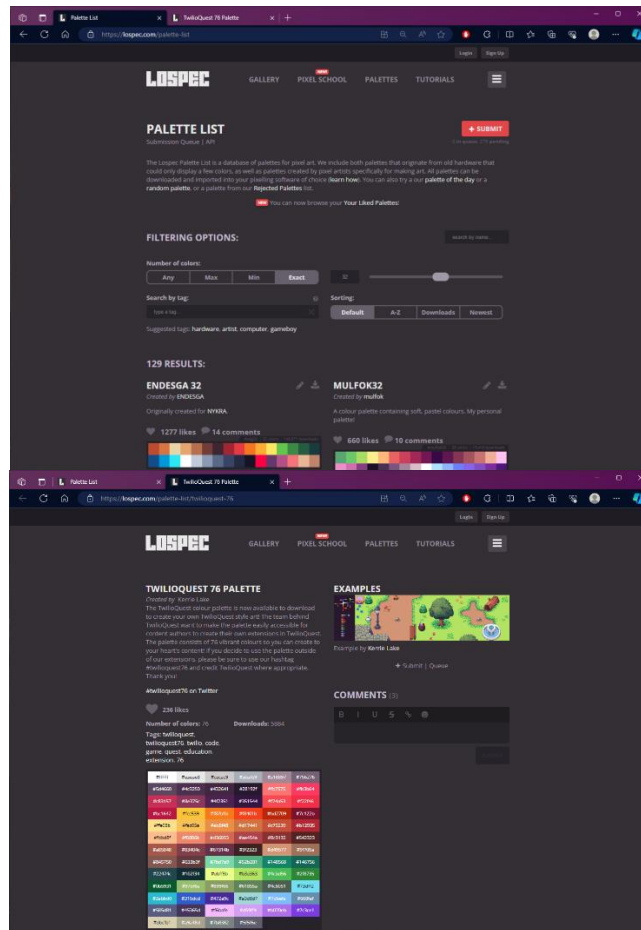


Рисунок 3.3. Пошук підходящої палітри у Lospec для створення графічних елементів

Всі графічні компоненти гри представлені у форматі зображень PNG. PNG (Portable Network Graphics) є растровим форматом для зберігання графічної інформації, використовуючи стиснення без втрат. Розміри зображень повинні бути одними з наступних значень: 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048 пікселів.

На даному етапі розробки формуються графічні елементи, такі як фон, платформи, зовнішні вигляди персонажів і перешкод, компоненти інтерфейсу (рис.3.4).

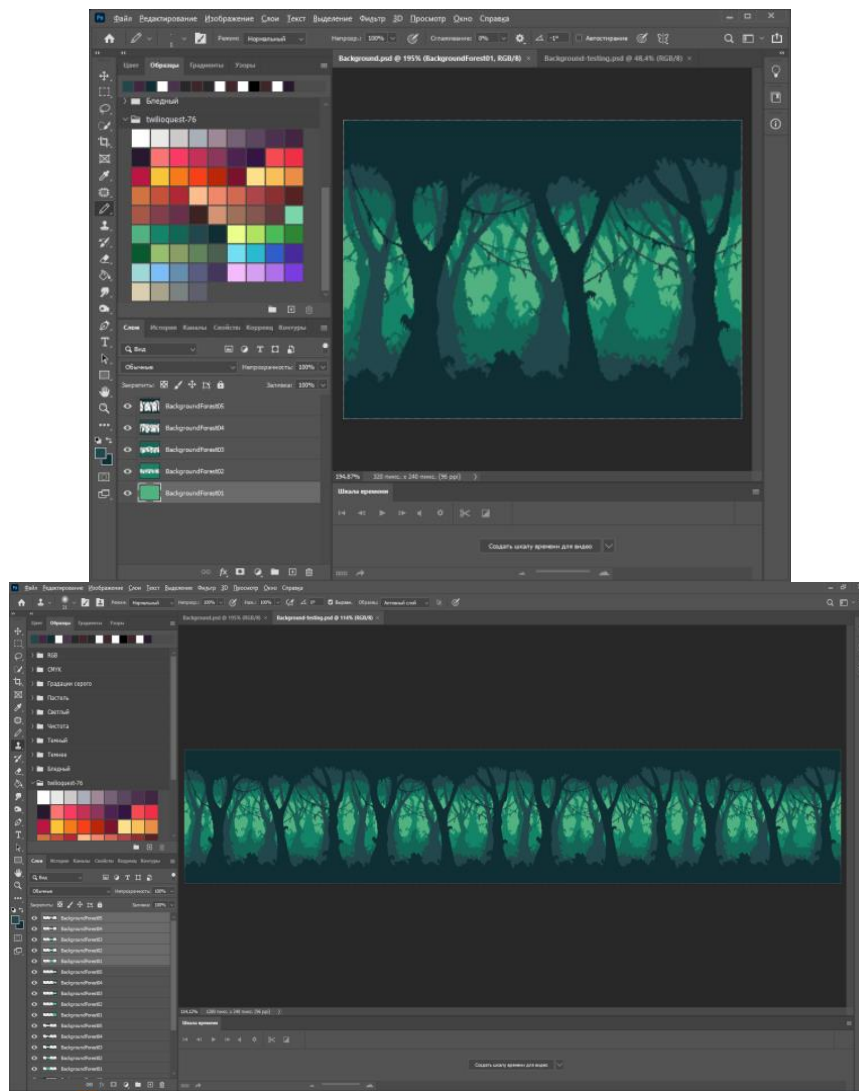


Рисунок 3.4 – Створення заднього фону та його тестування на замощування

Для подальшої анімації об'єктів необхідно відтворити всі можливі їхні стани, для кожного об'єкта та персонажа свої окремі (рис.3.5).

Також треба намалювати тайлсет для створення рівнів (рис.3.6). Тайлсет – це набір графічних елементів, відомих як тайли, які використовуються в іграх чи інших програмах для створення поверхонь, фонів і об'єктів. Тайли представляють собою невеликі графічні зображення, які можна організувати відповідно до визначеної сітки для створення сцени чи рівня.

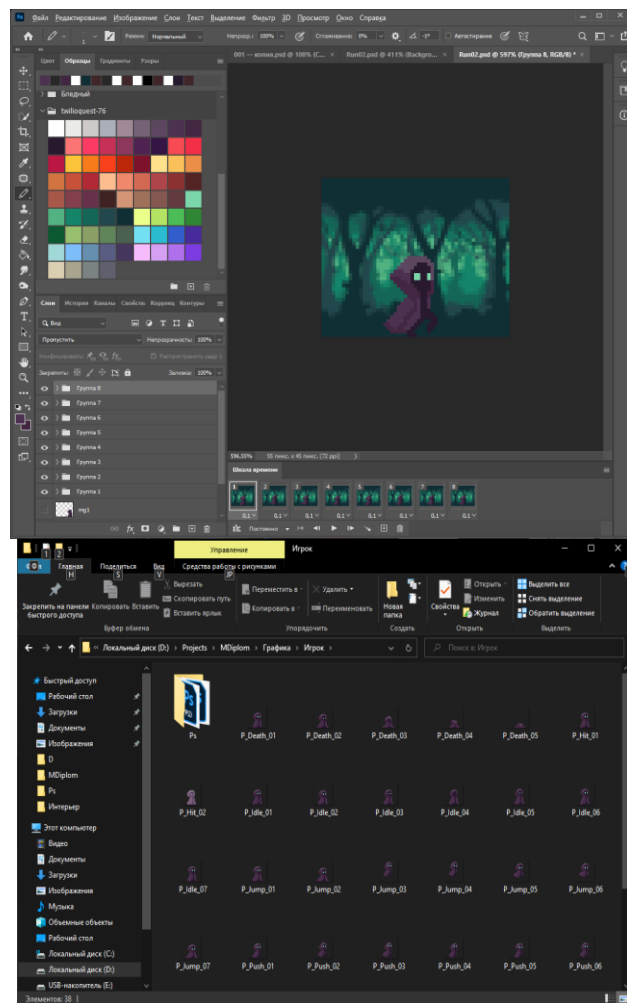


Рисунок 3.5 – Стани графічних елементів гри, створення анімацій у Фотошоп за допомогою шкали часу

Ключова особливість тайлсету полягає в тому, що він дозволяє використовувати однакові елементи (тайли) для повторного створення шаблонних частин гри. Наприклад, в 2D іграх тайлсет може містити різні типи ґрунту, стіни, об'єкти та інші елементи, які використовуються для будівництва гри.

Створюючи рівні чи сцени, розробник може використовувати ці тайли для компонування оточення чи об'єктів в грі. Використання тайлсету спрощує роботу з графікою, дозволяючи швидко створювати, редагувати та розширювати рівні чи сцени, зберігаючи при цьому консистентність стилю та ефективно використовуючи ресурси.

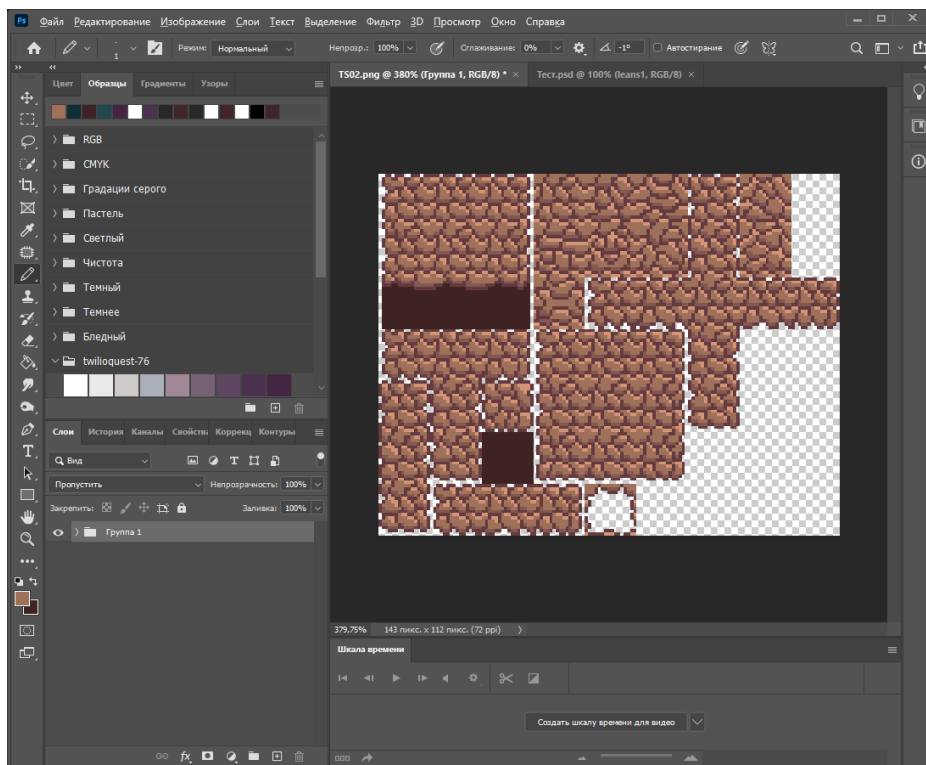


Рисунок 3.6 – Створення тайлсету для застосунку

Останнім кроком вносимо правки до графічних матеріалів та створюємо потрібні елементи для інтер'єру (рис.3.7.).

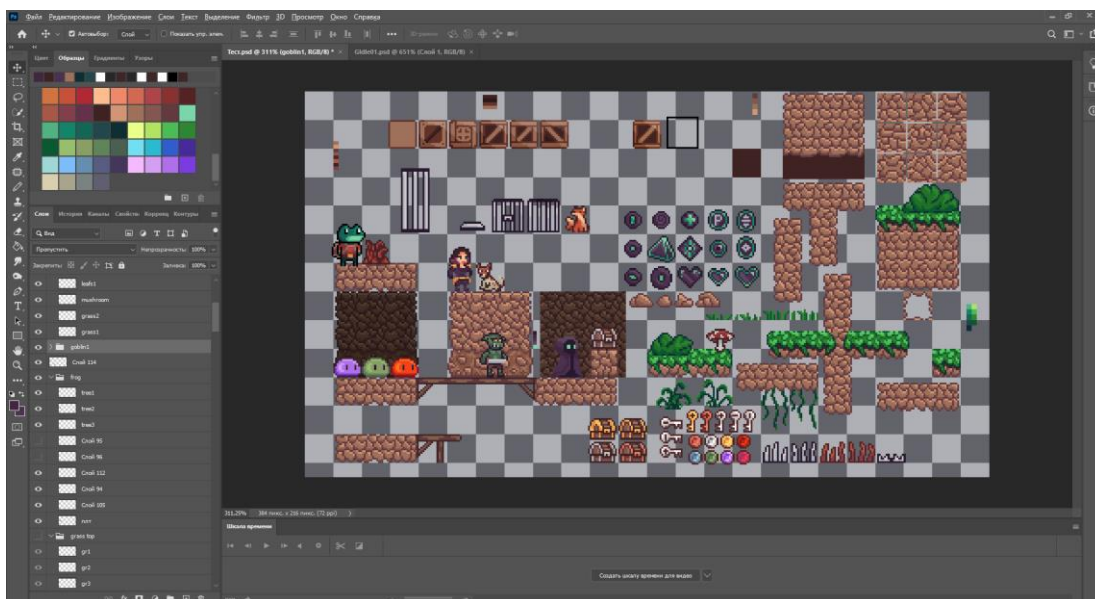


Рисунок 3.7 – Концепти оточення та інших об'єктів

3.4 Внутрішня ігрова логіка

Обраний жанр гри що розробляється – платформер. Платформери – це жанр відеоігор, де гравець управляє персонажем, що рухається по різних платформах, будь то статичні чи рухомі, використовуючи їх для подолання перешкод та досягнення мети гри. У традиційних двовимірних платформерах герой пересувається від лівого краю екрану до правого, однак існують ігри, де гравець може рухатися в будь-якому напрямку або розв'язувати загадки.

Один з можливих елементів таких ігор – отримання пауер-апів. Це спеціальні предмети, які надають гравцеві додаткові навички або можливості, такі як силові бар'єри, прискорення чи підвищення стрибків. Крім того, герой може збирати монети та ключі, які додають глибину грі та надають додаткові завдання для гравця.

Вороги, чи так звані монстри, варіюються за видами і інтелектуальними здібностями. Вони можуть активно переслідувати героя, випускати снаряди або взагалі рухатися за заданою траєкторією. Гравець, у свою чергу, може легко ліквідувати ворогів, стрибаючи на них або використовуючи доступну зброю.

Щоб підняти інтерес, рівні часто приховують секрети, які роблять гру більш захопливою для гравця. Графіка зазвичай відрізняється мальованим стилем, а персонажі можуть бути як міфічними істотами, так і антропоморфними тваринами (наприклад дракони, гобліни).

Платформери відносяться до категорії комп'ютерних ігор, що також включається до жанру аркад [22]. Основною характеристикою геймплею є стрибання по платформах, пересування вздовж різних проходів, збирання предметів і, як правило, завершення рівня та подолання різноманітних перешкод. Ігри цього жанру зазвичай не вимагають складних технічних рішень, але, незважаючи на це, можуть захопити гравця на тривалі години гри та привернути увагу різноманітної аудиторії.

Для гри вибрано ПК як ігрову платформу, оскільки це найзручніше середовище для одночасного розроблення та тестування гри. Крім того, комп'ютерні ігри залишаються найпопулярнішими серед усіх існуючих платформ.

Більшість ігор можна уявити як основну функцію, що містить усю ігрову логіку та активується або при взаємодії користувача, або після завершення певного часового інтервалу. Цей постійний цикл активації однієї і тієї ж основної функції називається ігровим циклом.

Давайте розглянемо детально структуру ігрової сцени. Початкова точка для головного персонажа – платформа, безпечна область, яку герой може використовувати для руху та стрибків. Платформи, по яких пересувається персонаж, представлені заздалегідь створеною картою. На платформах можуть знаходитися різноманітні статичні та динамічні елементи, такі як елементи, що завдають шкоди або надають гравцю монети та скарби. Також можуть бути присутні статичні об'єкти, такі як каміння, трава, кущі та ліани, які служать виключно для естетичної складової гри.

Однією з ключових механік гри є збір монет, ключів та визволення звірів з клітин. Ці об'єкти можуть бути розташовані на платформах, між ними та над перешкодами. Сундуки також можуть бути розміщені на платформах і містити монети або інші цінності. Гравець повинен уважно проходити рівень, щоб не пропустити сундуки зі скарбами та аптечки для поповнення здоров'я. Монети та сундуки представлені як динамічні об'єкти, монети можуть обертатися навколо своєї осі, а у сундуків відкривається кришка.

Динамічними графічними елементами є вороги, які взаємодіють з персонажем. Під час руху головного персонажа по платформах, рухається і фон гри, він також змінюється та є динамічним, підсилюючи враження руху та швидкості. Вороги можуть включати гоблінів, слаймів, шипи, прірви та пастки. Шипи та пастки утворюють перешкоду на платформах, яку герой повинен перестрибнути. Гобліни та слайми рухаються з різною швидкістю

та має можливість атакувати головного героя. Прірви знаходяться між платформами, і герой повинен їх перестрибати, щоб продовжити гру. Потрапляючи на будь-яку перешкоду у головного героя зменшується здоров'я, яке гравець може відновлювати за допомогою аптечок.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ

4.1 Вибір засобів розробки

Перед прийняттям остаточного вибору ігрового движку слід враховувати різноманітні фактори, які впливають на розробку гри:

- Кількість підтримуваних платформ та легкість компіляції проекту для них – важливо забезпечити широкий охоплення платформ та ефективність процесу компіляції проекту для них.
- Потенціал для майбутніх оновлень та розширення механік гри – переконайтеся, що обраний движок має потенціал для майбутніх оновлень та розширення функціоналу гри.
- Мова програмування – розглядайте можливість використання мов програмування, які вам зручні та знайомі.
- Витрати на навчання розробників – оцініть, наскільки легко розробники можуть оволодіти обраним движком та його інструментами.
- Легка масштабованість – розгляньте можливості легкої масштабованості проекту для відповіді на зростаючі вимоги.
- Вартість ліцензії або розмір відрахувань для розробників движка – розглядайте фінансові аспекти, такі як вартість ліцензії та можливі відрахування для розробників.
- Продуктивність на різних платформах – оцініть, як ефективно двигун працює як на сучасних, так і на застарілих платформах та пристроях.

Після ретельного аналізу усіх ключових важливих аспектів, стає очевидним, що Unity є найбільш відповідним вибором для проекту. Він об'єднує усі необхідні якості, такі як продуктивність у 2D, легкість переносу проекту на різні платформи, середовище, яке дозволить легко масштабувати проект та розширювати його механіку і можливості гри. Unity також

пропонує вбудований редактор спрайтів і анімації, що робить його ідеальним вибором для втілення ідей.

При розробці гри будемо використовувати як інтегроване середовище розробки (IDE) для роботи з Unity Microsoft Visual Studio 2022 (рис.4.1), ця середа забезпечить розширені можливості розробки, відлагодження та управління проектами. Ось кілька ключових аспектів використання Visual Studio 2022 з Unity:

- Інтеграція з Unity – Visual Studio має глибоку інтеграцію з Unity, що дозволяє вам легко редагувати та відлагоджувати код вашого проекту прямо в Visual Studio. При відкритті скрипта або іншого кодового файлу з Unity, Visual Studio автоматично відкриється.
- Підтримка мов програмування – Visual Studio підтримує різні мови програмування, зокрема C# - основну мову програмування для розробки в Unity. Ви можете зручно писати код, отримувати автодоповнення та використовувати розширені функції рефакторингу.
- Відлагодження – Visual Studio надає потужні інструменти для відлагодження коду в Unity. Ви можете встановлювати точки зупинки, відстежувати значення змінних та аналізувати виклики функцій для полегшення процесу виявлення та усунення помилок.
- Управління проектами – З Visual Studio ви можете зручно керувати вашим проектом Unity. Це включає створення нових класів, редагування файлів проекту, а також роботу зі сценами та ресурсами.
- Автодоповнення та підказки – Visual Studio надає автодоповнення коду та підказки, що полегшують написання коду та допомагають у використанні правильних методів та класів.
- Швидке переключення між файлами – Можливість швидкого переключення між файлами та проектами дозволяє розробникам легко орієнтуватися в коді та структурі проекту.

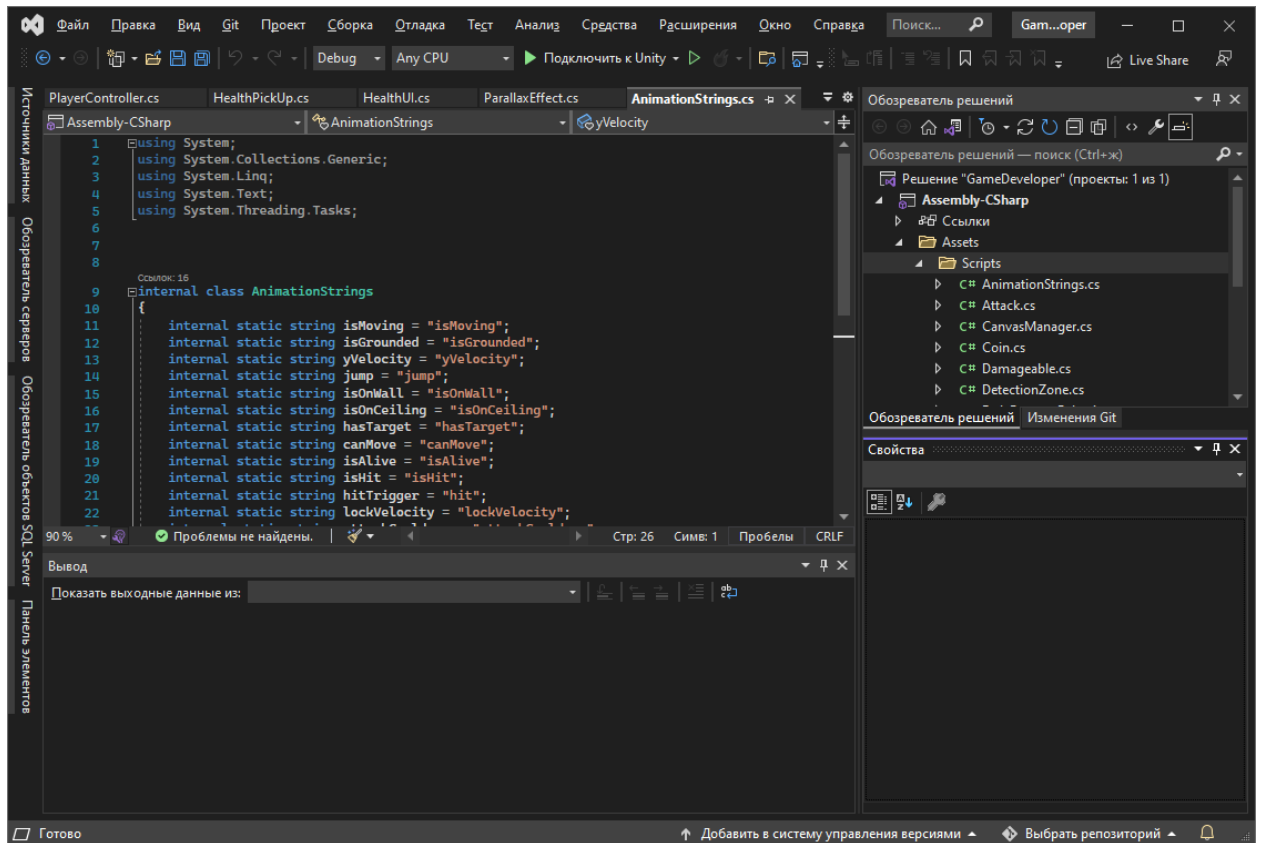


Рисунок 4.1 – Зовнішній вигляд Microsoft Visual Studio 2022

Для створення проекту буде використовуватися Unity Hub (рис.4.2). Це додаток, створений компанією Unity Technologies, який служить для керування та організації робочих проектів в середовищі Unity. Основна його мета – спростити роботу з різними версіями Unity та проектами, а також надати зручний інтерфейс для доступу до різних інструментів та ресурсів.

Unity Hub сприяє більш організованій та зручній роботі з проектами Unity, забезпечуючи централізоване місце для управління версіями двигуна, доступу до сервісів та ефективного переходу між різними проектами [23].

Основні функції Unity Hub:

- Управління версіями Unity: Unity Hub дозволяє легко встановлювати, оновлювати та видаляти різні версії Unity. Це корисно, оскільки різні проекти можуть вимагати різних версій двигуна Unity.

- Створення та управління проектами: Unity Hub дозволяє створювати нові проекти або відкривати існуючі. Ви можете легко перемикатися між різними проектами без необхідності виходу з програми.
- Інтеграція з Unity Services: Unity Hub надає зручний доступ до Unity Services, таких як Unity Collaborate, Cloud Build та Analytics. Ви можете легко включати ці сервіси для своїх проектів прямо через Unity Hub.
- Пошук та встановлення додатків (Add-ons): Unity Hub дозволяє вам легко знаходити та встановлювати додаткові компоненти, такі як модулі аналізу зображень, підтримка різних платформ тощо.
- Швидкий доступ до документації та ресурсів: Unity Hub має зручні посилання на офіційну документацію, форуми та ресурси, що може полегшити отримання інформації та допомогти вам вирішити проблеми.

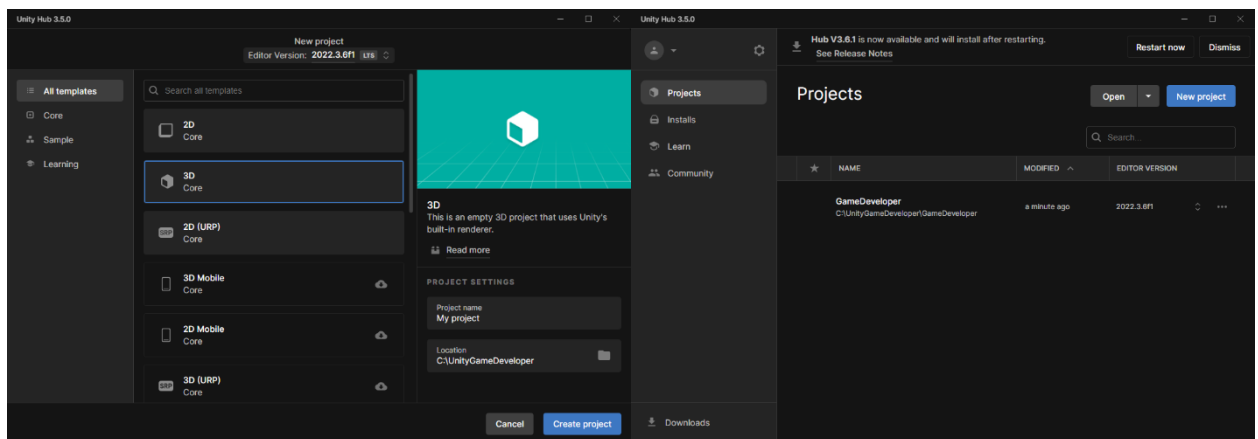


Рисунок 4.2 – Створення 2D проекту у Unity Hub

Так як для розробки був обраний ігровий движок Unity, а для написання та редагування програмного коду використовується Microsoft Visual Studio 2022, то знадобиться обладнання, що володіє технічними

характеристиками не нижче мінімально необхідних для коректно роботи даних засобів розробки.

Системні вимоги Unity Hub:

- ОС: Windows 7 SP1+, 8, 10, лише 64-розрядні версії; Mac OS X 10.13 +; Ubuntu 16.04, 18.04; CentOS 7.
- Графічний процесор: Відеокарта із підтримкою DX10 (версія шейдерів 4.0).

Системні вимоги Microsoft Visual Studio 2022:

- Процесор 64-біт, що працює на частоті 1,8 ГГц або швидше (рекомендується чотириядерний або краще). ARM-процесори не підтримуються.
- Мінімум 4 ГБ оперативної пам'яті (рекомендується 16 ГБ).
- Мінімум 2 віртуальних процесори та 8 ГБ оперативної пам'яті. Рекомендується 4 віртуальних процесори та 16 ГБ оперативної пам'яті.
- Мінімум 850 МБ та до 210 ГБ вільного місця на жорсткому диску (в залежності від встановлених функцій, зазвичай потрібно 20–50 ГБ вільного простору).
- Мінімальна роздільна здатність екрану WXGA (1366 на 768), але найкраще працює на роздільній здатності 1920 на 1080 або вище.

4.2 Структура та функціональне призначення модулів програми, їх взаємозв'язок

Unity – це потужний ігровий рушій, який має модульну структуру. Це означає, що він складається з різних модулів, кожен з яких виконує певну функцію, таку як обробка графіки, звуку, фізики, інтерфейсу користувача та інше. Ці модулі взаємодіють між собою, щоб створити цілісний досвід гри. Завдяки цій модульній структурі Unity є гнучким та налаштовуваним ігровим рушієм, який може бути адаптований до різних типів проектів.

Розглянемо детальніше модулі у Unity.

Unity GameObject – це основна одиниця віртуального світу гри. Він представляє об'єкт, який може бути відображений, взаємодіяти з іншими об'єктами та взагалі мати власну функціональність. GameObject сам по собі не виконує значної кількості функцій, але виступає як контейнер для компонентів, які визначають його характеристики та поведінку.

Кожен GameObject може мати різноманітні компоненти, такі як графічні (наприклад MeshRenderer для відображення 3D моделі), фізичні (Rigidbody для симуляції фізики), аудіо (AudioSource для відтворення звуків), та інші. Додавання та налаштування цих компонентів дозволяє створити повноцінний об'єкт у грі.

Одна з ключових характеристик GameObject – це його Transform. Компонент Transform визначає місцезнаходження, обертання та масштаб об'єкта у віртуальному просторі. Він визначає положення об'єкта відносно інших об'єктів та його розмір.

GameObject може бути активним чи неактивним, що дозволяє легко управляти його видимістю та взаємодією з іншими об'єктами. Крім того, вони можуть бути організовані у ієрархію, де деякі об'єкти є дочірніми до інших, що полегшує керування та структурування великих сцен.

Створення та взаємодія з GameObject є важливим елементом розробки в Unity, і вони використовуються для створення різноманітних об'єктів у грі, від персонажів та предметів до світла та частинок.

Unity Components є фундаментальними будівельними блоками для визначення поведінки та властивостей об'єктів в Unity. Кожен GameObject в системі Unity може містити один чи декілька компонентів, які надають об'єкту конкретну функціональність. Компоненти дозволяють програмістам та дизайнерам динамічно додавати та налаштовувати різноманітні аспекти гри без прямого втручання в код об'єкта. Найбільш поширеними компонентами є:

- Компонент Transform визначає просторове положення, обертання та масштаб об'єкта в грі. Всі об'єкти в Unity повинні мати цей компонент, оскільки він визначає їхнє місцеположення та розмір в ігровому просторі.
- Компонент MeshRenderer відповідає за відображення 3D-моделі об'єкта на екрані. Він використовується для надання вигляду об'єкта та його взаємодії з освітленням.
- Rigidbody надає фізичні властивості об'єкту, що дозволяє йому взаємодіяти з фізичним середовищем. Об'єкти з Rigidbody піддаються симуляції фізики, враховуючи сили, які діють на них.
- Collider визначає область, яка може взаємодіяти з іншими об'єктами в грі. Використовується для визначення колізій та областей взаємодії.
- Script (MonoBehaviour). Скрипти, які унаслідковані від класу MonoBehaviour, надають можливість програмно визначити поведінку об'єкта. Ці скрипти можуть містити методи, які викликаються в різних точках життєвого циклу об'єкта (Start, Update, FixedUpdate тощо).
- Компонент AudioSource відповідає за відтворення аудіофайлів, пов'язаних з об'єктом. Це може бути корисно для відтворення звукових ефектів чи музики.
- Компонент Light створює джерело світла в грі. Він визначає, як об'єкти відображають світло в ігровому просторі.

Кожен компонент може бути доданий та налаштований через інтерфейс редактора Unity або програмно з використанням мови програмування C#.

Unity Scene є концепцією, що відображає віртуальне середовище або рівень гри. Кожна сцена в Unity є унікальним файлом, який містить дані про розташування та конфігурацію об'єктів, а також параметри відображення та ігрового середовища. У сцені можна розташовувати та організовувати об'єкти, такі як персонажі, об'єкти оточення, світло та інші. Ієрархічна

структура дозволяє вам легко визначати взаємозв'язки між об'єктами та їх організацію. Сцена містить посилання на різноманітні ресурси, такі як 2D-моделі, текстури, звуки та інші активи. Це дозволяє легко використовувати та переносити елементи між різними сценами або проектами. Властивості сцени включають параметри камери, такі як поле зору та видимість, а також параметри освітлення, такі як напрямок світла та його інтенсивність. Це визначає вигляд та атмосферу гри. Сцени мають життєвий цикл, який включає етапи завантаження, виконання та вивантаження. Керування життєвим циклом дозволяє оптимізувати роботу гри та зберігати ресурси. Гра може містити кілька сцен, і перехід між ними відбувається, наприклад, при завантаженні нового рівня. Це дозволяє створювати великі та складні ігри, де кожна сцена представляє окремий рівень чи частину гри. Ви можете додавати скрипти (сценарії) до об'єктів у сцені, щоб визначити їхню логіку та взаємодію. Це робить сцену динамічною та інтерактивною.

Сцени у Unity служать основними будівельними блоками для конструювання гри. Розробники можуть легко керувати та маніпулювати різними аспектами гри, використовуючи сцену як основний інструмент для організації та взаємодії об'єктів.

Unity Editor – це інтегроване середовище розробки (IDE), що надає зручний та потужний інтерфейс для розробки ігор та додатків. Він дозволяє розробникам створювати, редагувати та відлагоджувати свої проекти, забезпечуючи широкий функціонал для роботи з ресурсами, об'єктами та налаштуваннями проекту, надає розробникам зручне середовище для всебічної роботи з грою.

Unity Editor має інтуїтивний та дружельюбний інтерфейс, що складається з різних панелей та вікон. Головні компоненти включають сцену, ієрархію об'єктів, проект, інспектор об'єкта та консоль для виведення повідомлень та помилок. Розробники можуть додавати, редагувати та видаляти об'єкти у сцені безпосередньо з Unity Editor. Це включає розташування та орієнтацію об'єктів, а також параметри компонентів, які

додають функціональність. Ресурси, такі як моделі, текстури, аудіофайли та інші активи, можна легко імпортувати, організовувати та використовувати через Unity Editor. Розташування ресурсів у проекті дозволяє легко управляти активами гри. Unity Editor дозволяє налаштовувати різні параметри проекту, включаючи налаштування платформи, налаштування камери, налаштування освітлення та інші глобальні параметри гри. У розділі ієрархії можна організовувати та визначати взаємозв'язки між об'єктами в сцені. Це полегшує керування великим числом об'єктів та їхньою взаємодією. Unity Editor надає зручні засоби для керування сценами, включаючи створення нових сцен, редагування існуючих та перехід між ними. Це важливо для створення великих та складних ігор з різними рівнями та частинами. Unity Editor надає зручні інструменти для відлагодження коду та перевірки стану об'єктів у реальному часі. Розробники можуть використовувати точки зупинки, спостерігачі та інші інструменти для ефективної роботи зі своїм кодом. Unity Editor підтримує імпорт та експорт різних форматів, що дозволяє легко обмінюватися активами та проектами між розробниками.

Unity Physics Engine є вбудованою бібліотекою фізики в середовищі Unity, яка дозволяє моделювати та симулювати реалістичну фізичну поведінку об'єктів у грі. Вона побудована на основі фреймворку Unity C# DOTS (Data-Oriented Technology Stack), що робить її потужною, ефективною та готовою до використання в ігрових проектах. Unity Physics Engine є безстатевою бібліотекою, що означає відсутність внутрішнього стану, який може впливати на результати симуляції. Це робить його стійким до різноманітних умов та дозволяє ефективно використовувати в мережевих іграх.

Використання DOTS Framework у Unity Physics дозволяє використовувати масштабовану, оптимізовану для різних апаратних засобів систему. Це полегшує створення ігор, що працюють ефективно на різних пристроях. Unity Physics Engine підтримує масштабовані для мережі симуляції фізики, що робить його ідеальним вибором для онлайн-ігор або ігор

з мережевими функціями. Система фізики має свій життєвий цикл, який включає етапи обробки введення, обчислення фізики, відновлення позицій об'єктів та інші. Це дозволяє точно контролювати та впливати на фізичну поведінку об'єктів. Unity Physics дозволяє визначати взаємодії між об'єктами, включаючи колізії, тригери та обчислення сил. Це створює реалістичну інтеракцію між об'єктами в грі. Система фізики легко інтегрується з іншими системами Unity, такими як анімація, аудіо та графіка, що дозволяє створювати повноцінні та реалістичні ігрові враження.

Unity Particle System – це потужний інструмент, який дозволяє розробникам створювати різноманітні візуальні ефекти, такі як вогняні кулі, дим, водяні бризки та багато інших. Система частинок в Unity забезпечує відтворення тисяч маленьких графічних об'єктів, названих частинками, для створення складних та реалістичних візуальних ефектів. Всі візуальні ефекти створюються з великої кількості частинок. Кожна частинка має різні властивості, такі як позиція, розмір, кольори та траєкторія руху.

Еміттер – це джерело частинок, яке визначає, де та як швидко частинки генеруються. Unity дозволяє додавати різні типи емітерів, такі як конус, коло або прямокутник, для створення різноманітних ефектів. Система частинок в Unity підтримує модульний підхід, де різні модулі можуть контролювати різні аспекти частинкового ефекту. Наприклад, модуль колізій визначає, як частинки реагують на об'єкти в грі. Unity Particle System надає можливість анімувати властивості частинок з часом. Інтерполяція між кадрами дозволяє створювати плавні та динамічні ефекти руху частинок. Для візуального вдосконалення частинок можна використовувати текстури та матеріали. Це дозволяє надати частинкам реалістичний вигляд та створити різноманітні візуальні ефекти.

Unity дозволяє налаштовувати режими змішування та прозорість для частинок, що робить ефекти більш гнучкими та інтегрованими з іншими об'єктами в грі. Система частинок може використовуватися для створення різноманітних графічних ефектів, таких як вогнені кулі, вибухи, снігопад,

дим, дощ та багато інших. Розробники можуть використовувати скрипти (сценарії) для динамічного керування системою частинок, що відкриває безмежні можливості для створення унікальних та складних ефектів.

Unity Event System – це ключовий компонент Unity, який надає можливість взаємодії об'єктів та реагування на події в грі чи додатку. Він служить механізмом передачі та обробки подій, таких як клікання мишею, натискання клавіші, дотик екрану та інші. Event System робить роботу з введенням та іншими подіями легкою та ефективною. Unity Event System базується на паттерні подій та делегатів в мові програмування C#. Він дозволяє об'єктам відправляти та слухати події, що відбуваються в грі. Один з основних аспектів Event System – це обробка введення. Наприклад, ви можете налаштовувати події на натискання кнопок миші, клавіш клавіатури, дотик екрану тощо. Event System дуже пов'язаний з інтерфейсом користувача (UI) в Unity. Він робить простим взаємодію та реагування на введення для UI-елементів, таких як кнопки, текстові поля, панелі тощо.

Основні компоненти Event System включають `EventSystem`, `InputModule` та `BaseInputModule`. Кожен з цих компонентів відповідає за різні аспекти обробки подій. Unity Event System дозволяє вам реагувати на події різними способами. Ви можете використовувати делегати, методи, об'єктні посилання, а також налаштовувати події безпосередньо в інспекторі Unity. Система подій Unity також підтримує перетягування та розміщення (drag and drop), що робить його корисним для реалізації функціоналу, пов'язаного з переміщенням об'єктів. Ви можете призначати методи та функції обробки подій, що дає можливість запускати конкретний код при виникненні певних подій.

Система анімації в Unity є потужним інструментом для створення рухливих та живих об'єктів у грі чи додатку. Вона дозволяє розробникам створювати анімації для персонажів, об'єктів та інших компонентів гри, використовуючи інтерактивні та інтуїтивні засоби.

Основним елементом системи анімації є компонент Animator. Він призначений для управління анімаціями об'єкта та взаємодії з різними станами об'єкта. Аніматор дозволяє створювати різні стани та визначати переходи між ними. Наприклад, персонаж може мати стани "ідти", "стояти", "стріляти", а переходи між ними відбуваються відповідно до дій гравця чи програмно. Unity підтримує використання анімаційних кліпів - коротких анімаційних сцен, що представляють конкретні рухи чи дії. Ці кліпи можуть бути створені в спеціалізованих програмах, імпортовані в Unity.

Система анімації має Curve Editor, який дозволяє розробникам редагувати і налаштовувати ключові кадри для анімацій, а також створювати згладжування та інтерполяції для плавності рухів. Розробники можуть параметризувати анімації, що дозволяє динамічно змінювати параметри, такі як швидкість, інтенсивність або напрямок руху, в залежності від умов та логіки гри. Unity Animation System підтримує обернену кінематику ІК (Inverse Kinematics), що робить можливим реалістичне позиціонування рук, ніг та інших частин персонажа в залежності від середовища чи об'єктів навколо. Blend Trees дозволяють змішувати різні анімаційні кліпи в залежності від введення або параметрів, створюючи більш складні та реалістичні анімації. Unity Animation System може бути використана для створення анімацій інтерфейсу користувача (UI), що робить гру чи додаток більш динамічним та привабливим.

Система звуку в Unity є важливим елементом для створення іммерсивних ігрових та додаткових досвідів. Unity Sound System надає розробникам інструменти для імпорту, управління та відтворення аудіофайлів в їхніх проектах. Unity підтримує імпорт різних форматів звукових файлів, таких як WAV, MP3, OGG та інші. Це дозволяє розробникам використовувати різноманітні аудіофайли в їхніх проектах. Unity дозволяє додавати різні звукові ефекти та фільтри, такі як ехо, реверберація, еквайзери та інші. Це дозволяє розробникам налаштовувати звуковий ландшафт для створення конкретного настрою чи атмосфери.

Звукова система легко інтегрується з анімаційною системою Unity, дозволяючи синхронізувати звукові ефекти з подіями в грі чи додатку. Unity дозволяє динамічно змінювати гучність звуку в реальному часі. Це дозволяє адаптувати рівень гучності відповідно до різних сценаріїв гри. Звукові події дозволяють легко визначати та відтворювати звуки відповідно до певних подій в грі, таких як стріляння, вибухи чи перемоги. Unity дозволяє використовувати музичні треки та звукові фони для створення настрою гри чи сцени. Розробники можуть контролювати та змінювати треки в реальному часі. Звукова система підтримує многоканальність, дозволяючи розробникам створювати складні звукові сценарії з використанням різних аудіоканалів. Unity дозволяє використовувати зовнішні ресурси та інтегрувати аудіофайли з різних джерел, що робить його потужним для проектів, що використовують різноманітні мультимедійні ресурси.

Інтерфейс користувача Unity (Unity UI), також відомий як uGUI, є системою, що надає засоби для створення та управління інтерфейсом користувача в іграх та додатках реального часу. Unity UI дозволяє розробникам легко створювати взаємодію з користувачем, реалізовувати меню, кнопки, текстові поля та інші елементи інтерфейсу.

Canvas є основною компонентою Unity UI, яка представляє область, на якій відображається інтерфейс. Він може бути налаштований для відображення в певному розмірі або розтягнутися на весь екран. Unity UI включає ряд готових елементів інтерфейсу, таких як Text (текст), Image (зображення), Button (кнопка), Slider (слайдер), Input Field (поле введення) та інші, які можна легко додавати до Canvas.

Кожен UI елемент має компонент Rect Transform, який визначає його розташування, розмір та обертання в просторі. Це дозволяє точно розташовувати елементи на екрані. Unity UI підтримує Layout Groups, такі як Vertical Layout Group та Horizontal Layout Group, які автоматично розташовують та організовують дочірні елементи в заданий спосіб.

Unity UI використовує Event System для обробки подій, таких як натискання кнопок, наведення миші та інші взаємодії з користувачем. Розробники можуть застосовувати анімації до UI елементів, щоб створити плавні переходи, реакції на введення або динамічні зміни в інтерфейсі.

Unity UI дозволяє використовувати власні зображення та графіку для створення унікального та візуально привабливого інтерфейсу. Для обробки великої кількості контенту Unity UI надає Scroll View, який дозволяє створювати прокрутні області з автоматичним управлінням. Unity UI дозволяє розробникам легко реалізовувати UI для досягнень, інвентаря, налаштувань гри та інших складних елементів інтерфейсу.

Всі ці модулі взаємодіють між собою, щоб створити цілісну гру. Це означає, що кожен аспект гри, від графіки та звуку до фізики та інтерфейсу користувача, є результатом взаємодії різних модулів. Ця взаємодія дозволяє Unity створювати багатопланові, динамічні ігрові досвіди. Крім того, завдяки модульній структурі Unity, розробники можуть легко налаштовувати та модифікувати окремі модулі, щоб відповідати конкретним потребам їхньої гри. Створення 2D проекту у Unity Hub

4.3 Розробка програмних модулів

Після створення 2D проекту у Unity Hub та запуску ми потрапляємо у головне вікно проекту, де основні компоненти включають вікно Сцени, вікно Анімації та вікно Гри (рис.4.3). Вікно Сцени служить для створення загальної композиції рівня та додавання нових об'єктів. Ми можемо легко редагувати та анімувати об'єкти у вікні Анімації. Вікно Гри показує актуальний вигляд гри з обраної камери.

Зліва ми бачимо вікно Ієрархії, яке містить усі об'єкти, що присутні в поточній сцені, включаючи камеру. Праворуч розташоване вікно Інспектора, яке відображає властивості обраного об'єкта. Ми можемо легко налаштовувати їх з цього вікна.

Знизу є три вікна: Вікно Проекту, вікно Аніматора та Консоль. Вікно Проекту показує всі об'єкти, що були додані до проекту, включаючи скрипти та анімації. Вікно Аніматора дозволяє створювати зв'язки та правила для переходу між різними анімаціями. Консоль служить для відображення помилок та винятків, які можуть виникнути під час роботи гри. Ці вікна забезпечують повний контроль та зручний моніторинг проекту в середовищі розробки Unity.

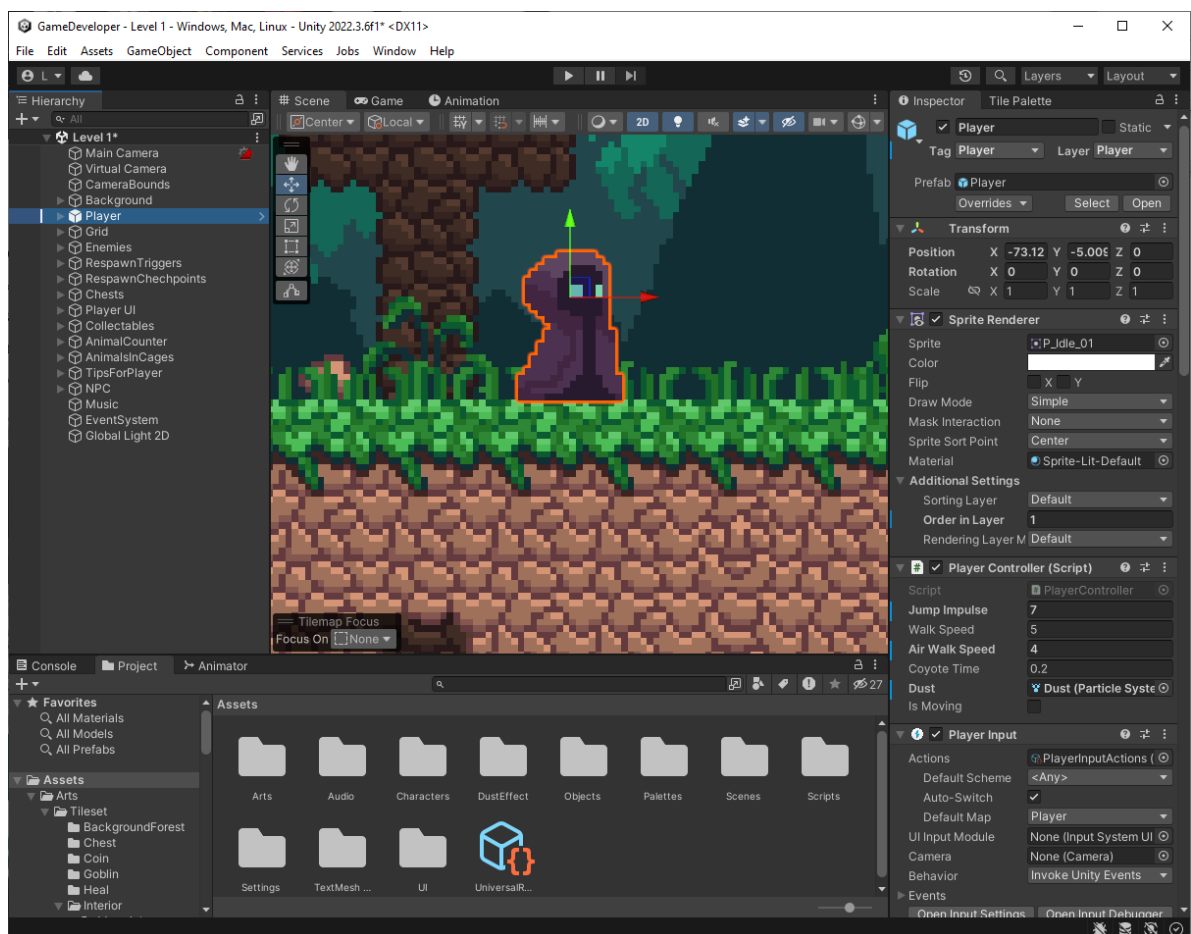


Рисунок 4.3 – Вікно 2D проекту у Unity

Перш ніж розробляти елементи гри треба завантажити набори з додатковими функціями, інструменти, розширення, так звані пакети, які призначені для поліпшення можливостей движка (рис.4.4).

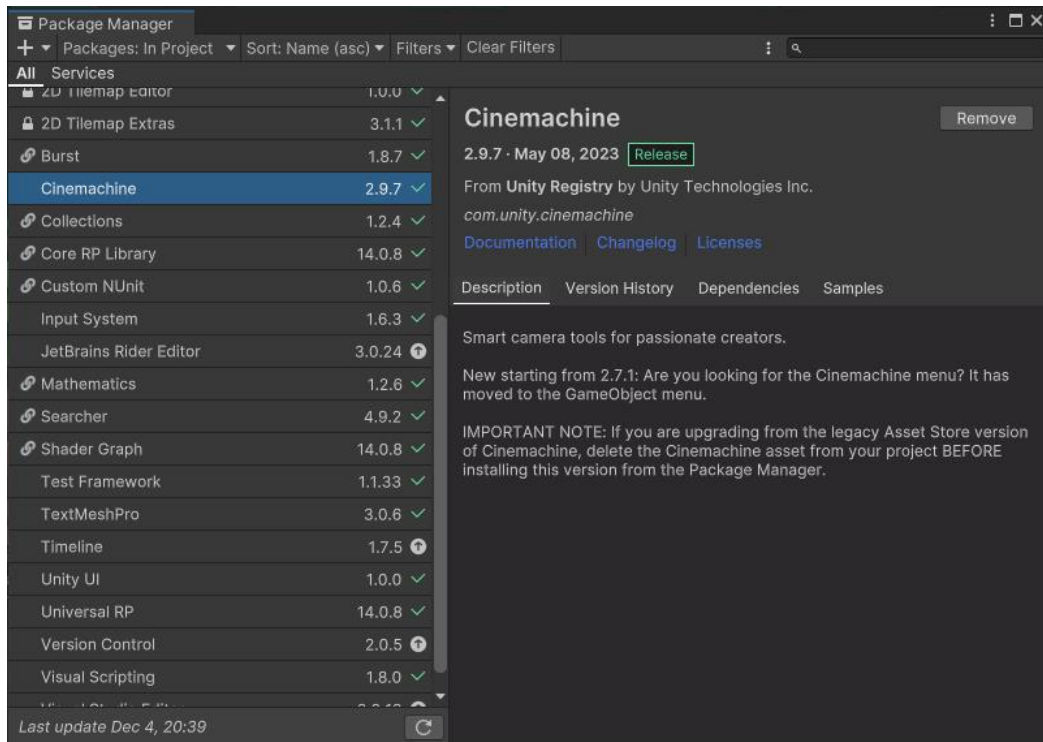


Рисунок 4.4 – Вікно Package Manager Unity

Cinemachine – це розширене рішення для камерного контролю в Unity. Воно надає потужні інструменти для створення динамічних камерних систем, що автоматизують слідування за об'єктами, розміщення камер, зум та інші аспекти створення кінематографічного досвіду гравця.

Input System – це нова система обробки введення в Unity, яка замінює старий Input Manager. Вона пропонує зручний і однорідний спосіб обробки введення, підтримує різноманітні пристрої (клавіатура, миша, геймпади) та полегшує розробку гри з різними вхідними методами.

TextMeshPro (TMP) – це розширення для відображення тексту в Unity, яке забезпечує велику кількість можливостей форматування та стилізації тексту. Воно підтримує розмітку, введення HTML-подібних тегів, і дає більше контролю над відображенням тексту у порівнянні зі стандартним текстовим компонентом Unity.

Unity UI – це система для створення і взаємодії з графічним інтерфейсом користувача (UI) в Unity. Забезпечує інструменти для створення

кнопок, текстових полів, вікон та інших елементів UI. Його використання дозволяє легко створювати та керувати інтерактивними елементами гри.

В першу чергу створимо палітру для нашого тайлсету (рис.4.5). Tile Palette дозволяє розміщувати та організовувати тайли (плитки) на сцені, щоб легко створювати різноманітні 2D-ландшафти, рівні або ігрові об'єкти. Заздалегідь було створено декілька тайлсетів, для переднього та заднього плану, для інтер'єру та ін.

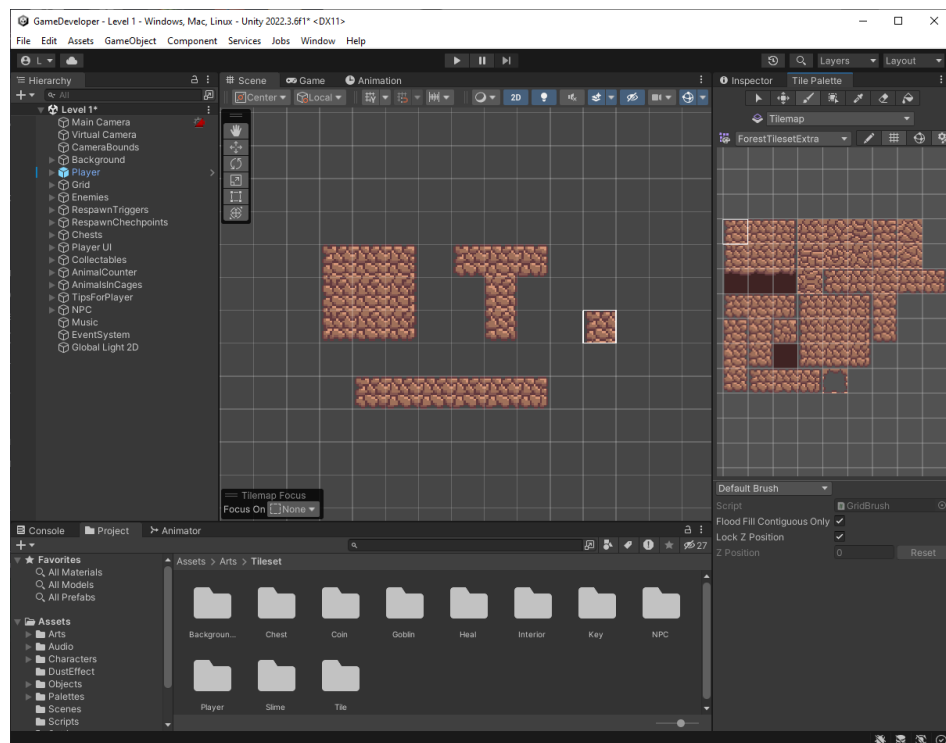


Рисунок 4.5 – Робота з Tile Palette

Окрім Tilemap Renderer були додані Tilemap Collider 2D, Rigidbody 2D, Composite Collider 2D для використання в створенні фізичної взаємодії та реалістичності середовища гри. Ці компоненти допомагають створювати віртуальний світ платформера, де гравець може взаємодіяти з оточенням, пересуватися за законами фізики, а оптимізація Composite Collider 2D полегшує управління фізичною взаємодією тайлованих структур.

Unity дозволяє зберігати об'єкти разом з усіма доданими властивостями і скриптами, щоб їх можна було використовувати повторно, просто перемістивши на сцену, не створюючи новий об'єкт. Такий об'єкт називається префабом, і для його створення достатньо просто перемістити створений об'єкт з вікна Ієрархії в вікно Проекту.

Наступним кроком створюємо головного персонажа. Додаємо усі потрібні компоненти, створюємо контролер анімації та потрібні скрипти для контролю персонажем, визначенням чи стоїть персонаж на поверхні, торкається стін, скільки здоров'я залишилось, та ін. (рис.4.6).

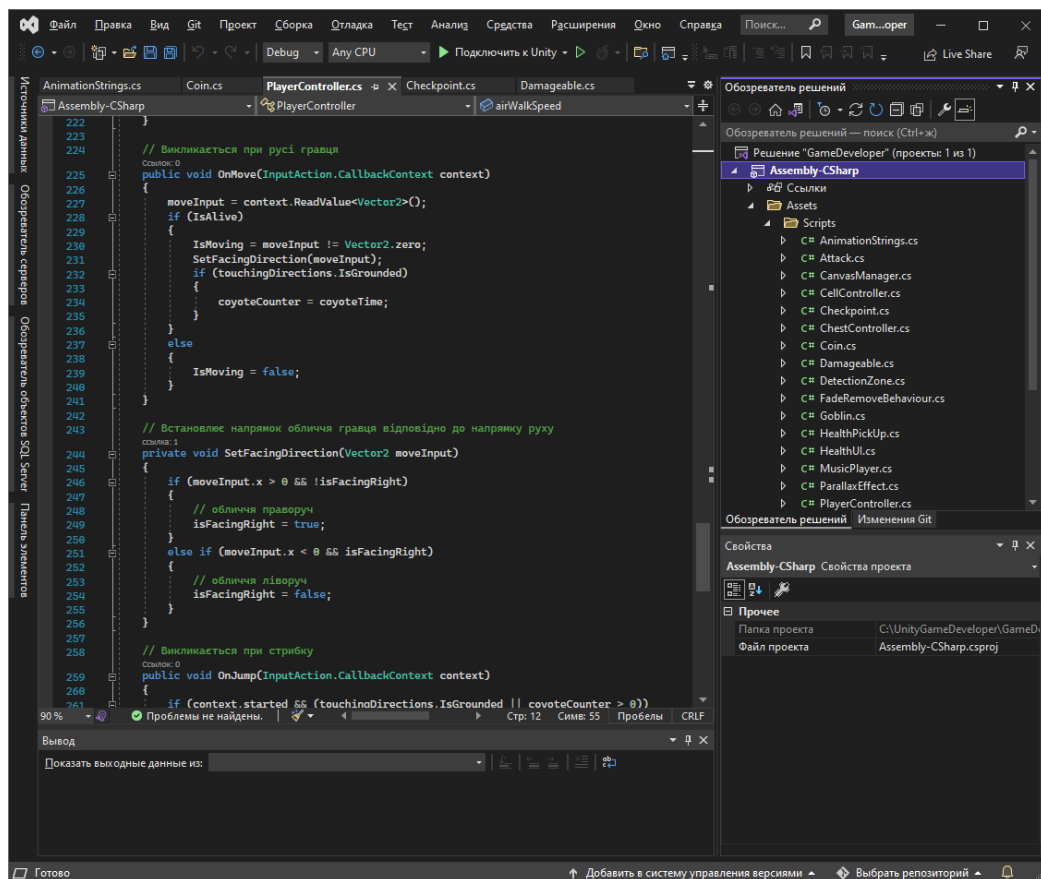


Рисунок 4.6 – Робота з кодом PlayerController у VS

За допомогою `Animation` створюємо кліпи для усіх анімацій, налаштовуємо швидкість для кожного. У `Animator` створюємо переходи між ними, визначаємо усі потрібні умови та сприяння взаємодії об'єкта з різними

станами. За допомогою числових, булевих та тригерних параметрів керуємо зміною анімацій між собою (рис.4.7).

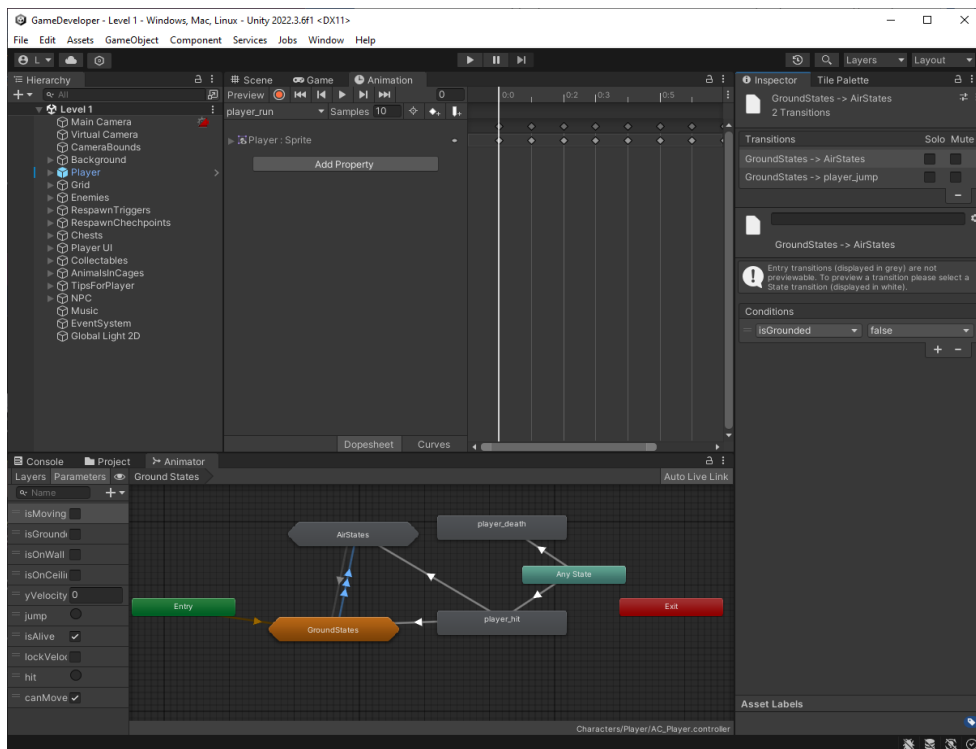


Рисунок 4.7 – Створення та налаштування анімацій

Таким же способом створюємо об'єкти та додаємо потрібні компоненти, скрипти для монет, ключів, скарбниць, клітин з тваринами, аптечок, додаємо пастки, прірви, шипи, різних ворогів, чекпоінти, нпс. Важливим елементом буде додавання інтер'єрних об'єктів на рівні. Після створення усіх об'єктів та компонентів створюємо потрібну кількість сцен для різних меню та сцени для кожного рівня гри. Заповнюємо кожний рівень об'єктами для збору гравцем, тваринами яких потрібно врятувати на кожному рівні та різними перешкодами, які гравець повинен буде подолати щоб пройти рівні та гру (рис.4.8). Важливим елементом також є інтерфейс гравця на якому зображена кількість монет, ключів, врятованих тварин та здоров'я гравця. Також на перший рівень потрібно додати підказки з навчанням управління.

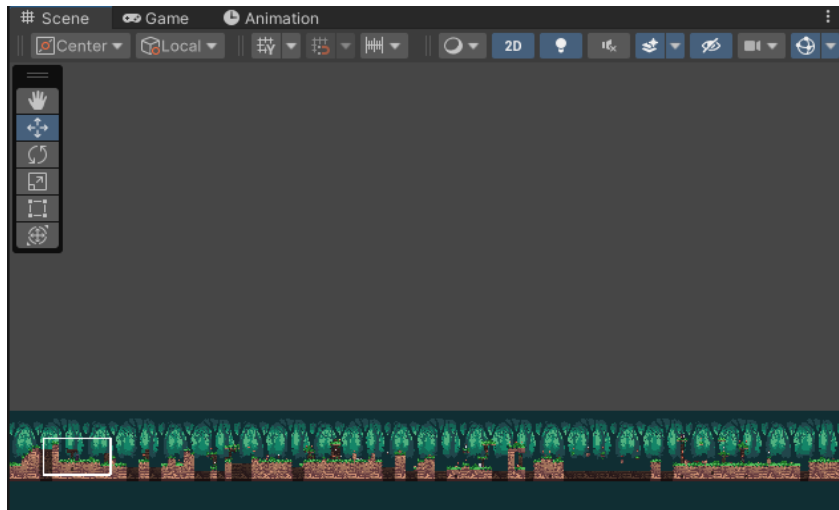


Рисунок 4.8 – Фрагмент першого рівня

Після завершення конфігурації ігрових сцен, анімацій, звуків та внутрішньої логіки гри, створюється збірка, яку можна надалі передавати та поширювати. Цей склад гри, який знаходиться на панелі налаштувань збірки (Build settings), отримав назву "Build". У налаштуваннях для цієї гри обрані 3 основні операційні системи: Windows, Mac та Linux. Дуже важливо додати у цьому вікні усі сцени (рис.4.9).

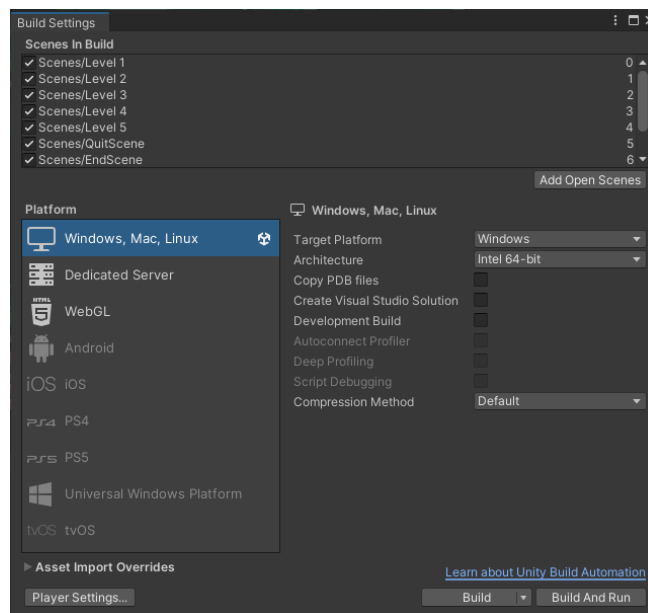


Рисунок 4.9 – Збірка проекту

4.4 Керівництво користувача

Після завантаження збірки користувачу потрібно лише запустити exe файл та почати гру (рис.4.10). Під час руху по рівням користувач буде отримувати спливаючі підказки як рухатися, стрибати, що треба збирати ключі для того, щоб відкривати скрині з монетами та клітини тварин (рис.4.11). На рівнях завжди є потрібна кількість ключів, щоб відкрити усі об'єкти.

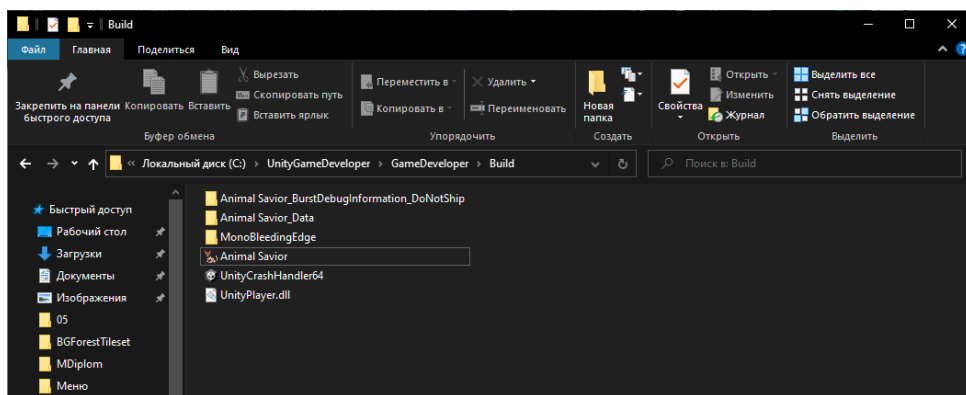


Рисунок 4.10 – Запуск застосунку

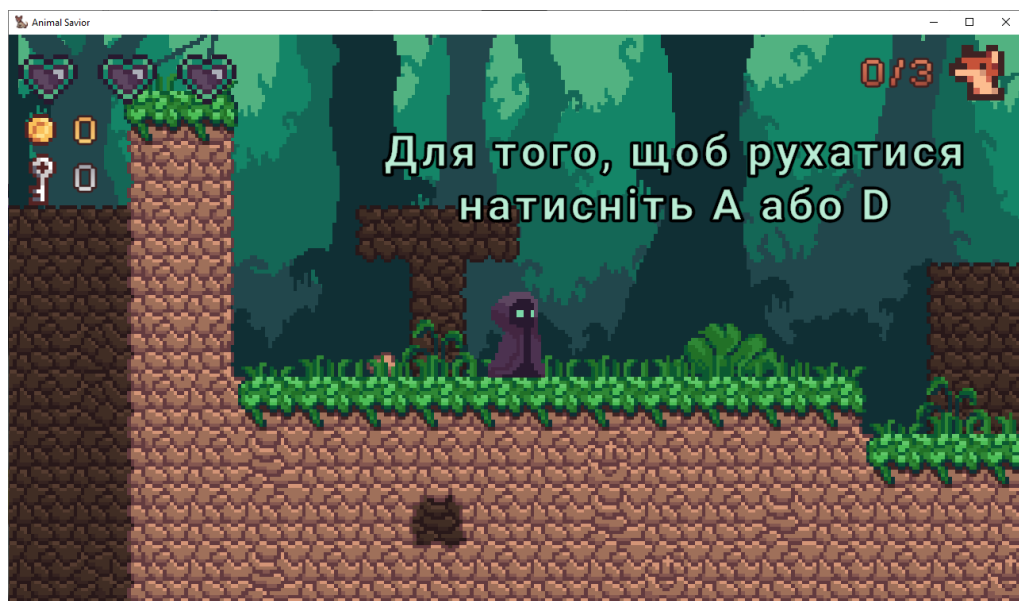


Рисунок 4.11 – Приклад підказки на першому рівні

Під час проходження рівнів гравець повинен буде подолати перешкоди, якщо наприклад персонаж впаде у прірву то він відправиться до найближчого чекпоінту та втратить 1 серце (рис.4.12).

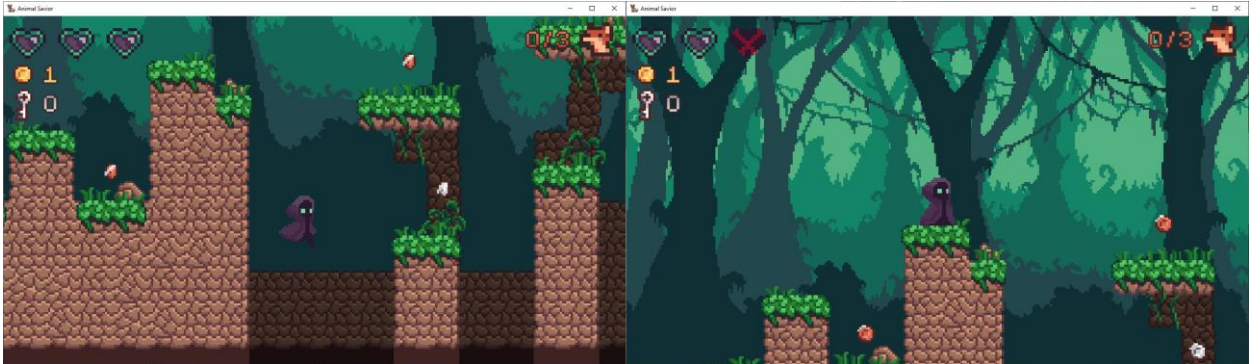


Рисунок 4.12 – Приклад втрати здоров'я з поверненням у чекпоінт

Для поповнення здоров'я користувач може знаходити аптечки в обмеженій кількості (рис.4.13). Після використання ключів на скрині з неї випадає випадкова кількість монет різної цінності, а при використанні ключів на клітинах з тваринами вони відкриваються та кількість врятованих тварин збільшується, ключ в обох випадках витрачається (рис.4.14).

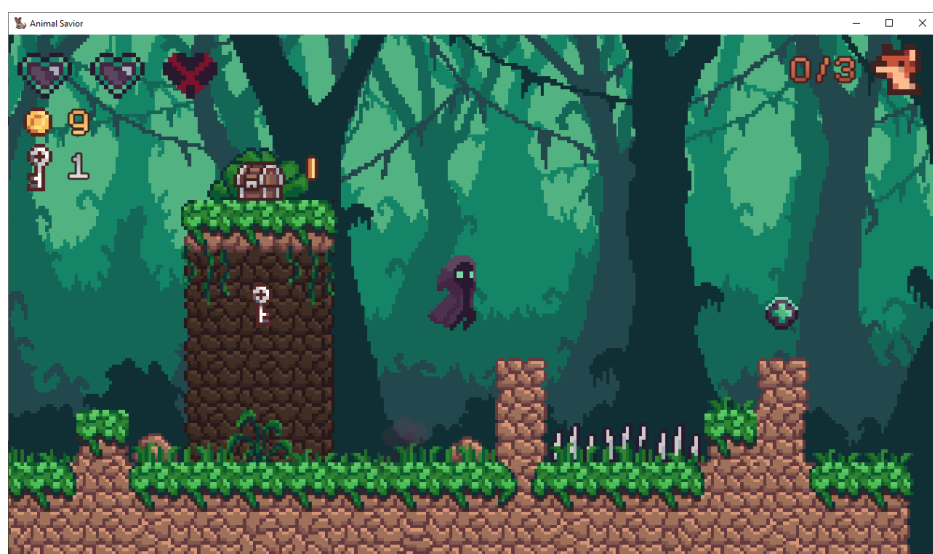


Рисунок 4.13 – Вигляд різних об'єктів на рівні

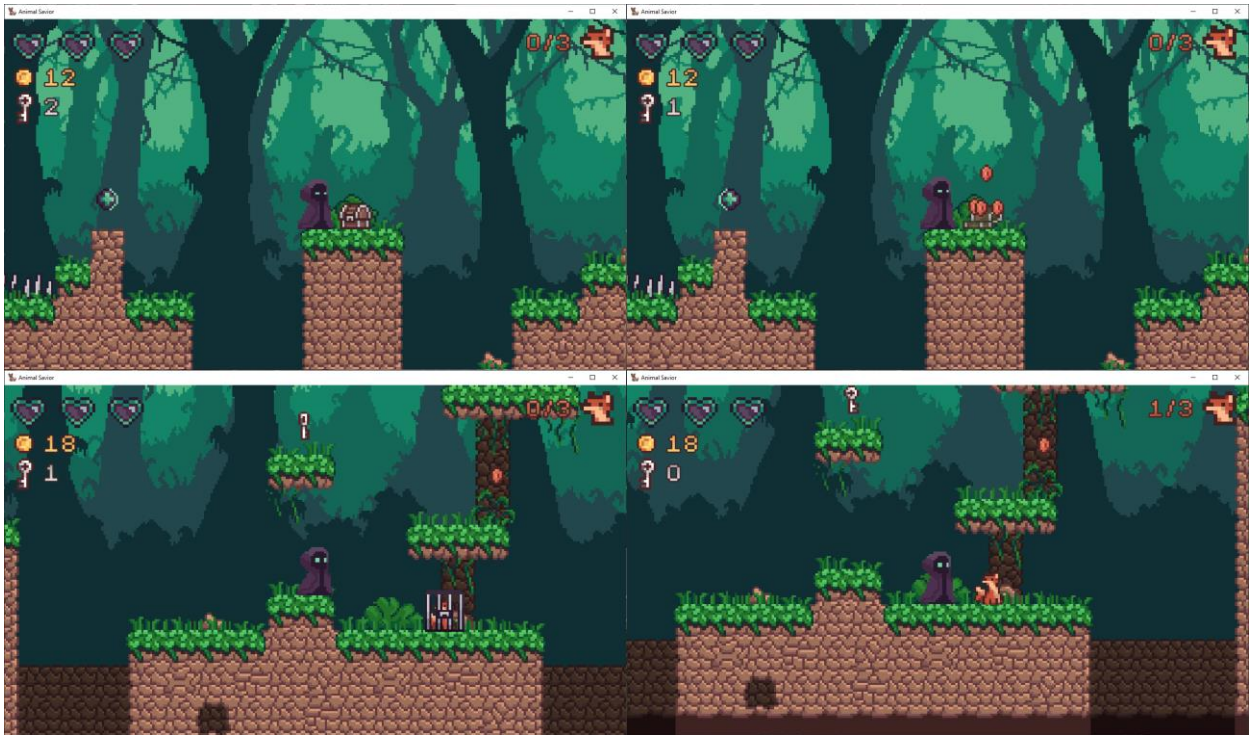


Рисунок 4.14 – Відкриття скринь та клітин

Після того як користувач врятує всіх тварин на рівні він повинен дістатися його кінця для завершення. Коли гравець пройде усі рівні він отримує похвалу за проходження та подяку за використання застосунку (рис.4.15).

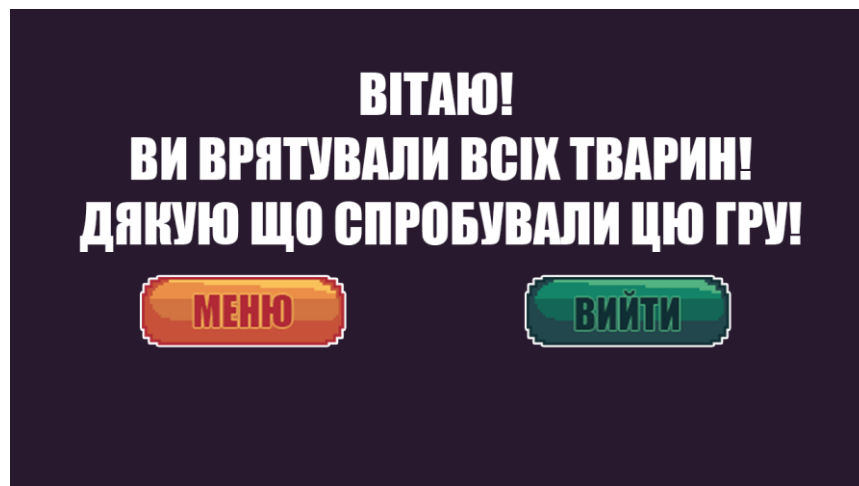


Рисунок 4.15 – Кінець гри

ВИСНОВКИ

Після глибокого аналізу світового та українського ігрового ринку можна визначити, що важливість ігор для різних платформ лише зростатиме в майбутньому. З покращенням технологій та зростанням доступу до Інтернету все більше осіб буде залучатися до різноманітних ігрових досвідів. Ігри стають не просто засобом розваги, але й способом викликати різні емоції та надавати радість користувачам. Відзначається, що цей сегмент індустрії стає все більш широким та приносить зростаючий прибуток, що свідчить про його значущість у економіці.

У даній дипломній роботі проведено детальний аналіз індустрії і всіх її аспектів. За отриманими даними можна з упевненістю стверджувати, що розвиток ігор буде проходити на тому ж стрімкому рівні, що і розвиток інформаційних технологій.

В рамках дипломної роботи було ретельно розглянуто процес створення гри, з особливою увагою до розробки графіки та анімації графічних елементів. Використання середовища розробки Unity 2D виявилось ефективним та доступним, дозволяючи реалізувати всі задумані функції завдяки внутрішнім можливостям движка.

У фінальному ігровому продукті було втілено значну кількість механік та інструментів Unity, що формують унікальний ігровий досвід при взаємодії користувача з грою.

Отже, розробка гри є досить актуальною темою, оскільки ігрова індустрія продовжує розростатися в Україні та світі взагалі і донині. На основі проведеного аналізу було обрано середовище розробки гри, розроблено вимоги до розробки 2D піксельного платформера в середовищі Unity та створено 2D піксельний платформер для розважальних та навчальних цілей. Ця розробка може бути перспективною та корисною з ігрової та освітньої точки зору. Дизайн, механіка та дизайн рівнів гри були оптимізовані для забезпечення безперебійного та приємного ігрового

процесу. Використання Unity забезпечило легку масштабованість і потенціал для майбутніх оновлень та розширення механік і можливостей гри.

Результати роботи доповідалися на Науковій конференції молодих вчених XXII ОДЕКУ 30 травня 2023р.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Індустрія відеоігор. URL: <https://uk.wikipedia.org/wiki/Відеогра> (дата звернення 1.11.2023)
2. Зростання ринку відеоігор. URL: <https://www.gamesindustry.biz/articles/2020-12-21-gamesindustry-biz-presents-the-year-in-numbers-2020> (дата звернення 01.11.2023)
3. Загальний дохід ігрової індустрії у світі URL: <https://www.statista.com/outlook/dmo/digital-media/video-games/worldwide> (дата звернення 01.11.2023)
4. 10 кращих країн/ринків за доходами від ігор URL: <https://newzoo.com/insights/rankings/top-10-countries-by-game-revenues> (дата звернення: 01.11.2023)
5. Монетизація у відеоіграх URL: https://en.wikipedia.org/wiki/Video_game_monetization (дата звернення: 03.11.2023)
6. Найприбутковіші ігрові компанії URL: <https://www.statista.com/chart/27523/public-companies-with-highest-game-revenue> (дата звернення: 03.11.2023)
7. Хто працює в індустрії геймінгу URL: <https://www.thebalancemoney.com/video-game-jobs-525965> (дата звернення: 03.11.2023)
8. Дохід ринку відеоігор у всьому світі за сегментами URL: <https://www.statista.com/statistics/292751/mobile-gaming-revenue-worldwide-device> (дата звернення: 07.11.2023)
9. Найбільш завантажувані мобільні ігри 2023 URL: <https://appfigures.com/resources/insights/20230908?f=4> (дата звернення: 07.11.2023)

10. Історія портативних ігрових пристроїв URL: https://en.wikipedia.org/wiki/Handheld_game_console (дата звернення: 07.11.2023)
11. Офіційний веб-сайт Unreal Engine URL: <https://www.unrealengine.com/en-US> (дата звернення: 09.11.2023)
12. Офіційна документація Unity URL: <https://docs.unity3d.com/Manual/index.html> (дата звернення: 09.11.2023)
13. Офіційна документація Godot Engine URL: <https://docs.godotengine.org/en/stable> (дата звернення: 09.11.2023)
14. Офіційна документація CryEngine URL: <https://docs.cryengine.com> (дата звернення: 09.11.2023)
15. Офіційний веб-сайт Source Engine URL: <https://developer.valvesoftware.com/wiki/Source> (дата звернення: 09.11.2023)
16. Офіційний веб-сайт Amazon Lumberyard URL: <https://aws.amazon.com/ru/lumberyard> (дата звернення: 09.11.2023)
17. Популярні ОС на ПК для геймінгу URL: <https://techjourney.net/windows-vs-mac-vs-linux-whichs-the-best-os-for-gaming> (дата звернення: 11.11.2023)
18. Етапи розробки відеоігор URL: <https://gamestudio.n-ix.com/video-game-development-stages-from-idea-to-release> (дата звернення: 12.11.2023)
19. Визначення вимог до програмного забезпечення та розробка технічного завдання URL: https://uk.wikipedia.org/wiki/Технічне_завдання (дата звернення: 15.11.2023)
20. Модель бізнес-процесу та нотація URL: https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation (дата звернення: 15.11.2023)
21. Що таке піксельна графіка? URL: <https://www.adobe.com/ua/creativecloud/design/discover/pixel-art.html> (дата звернення: 18.11.2023)

22. Типи платформених ігор URL: <https://retrostylegames.com/blog/types-of-platformer-games/> (дата звернення: 19.11.2023)

23. Що ви можете зробити з Unity Hub URL: <https://unity.com/unity-hub> (дата звернення: 22.11.2023)

ДОДАТОК А Вихідний код програми

```

using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.InputSystem;
[RequireComponent(typeof(Rigidbody2D), typeof(TouchingDirections),
typeof(Damageable))]
public class PlayerController : MonoBehaviour
{
    // Публічні змінні для налаштування гравця
    public float jumpImpulse = 7f; // Сила стрибка
    public float walkSpeed = 5f; // Швидкість руху на землі
    public float airWalkSpeed = 3f; // Швидкість руху у повітрі
    public float coyoteTime = .2f; // Час, протягом якого можна зробити стрибок
    після зіткнення з краєм платформи
    private float coyoteCounter; // Лічильник для відліку часу замикання
    // Вхідні дані для керування
    Vector2 moveInput;
    // Компоненти гравця
    Rigidbody2D rb;
    Animator animator;
    TouchingDirections touchingDirections;
    Damageable damageable;
    public ParticleSystem dust; // Система частинок для подачі ефекту пилу
    // Змінні для взаємодії з сундуками
    private bool isNearChest = false;
    private ChestController nearbyChest;
    private PlayerInventory playerInventory;
    // Змінні для взаємодії з клітинами
    private bool isNearCell = false;
    private CellController nearbyCell;
    // Змінні для визначення напрямку руху та чи гравець рухається
    [SerializeField]
    private bool _isMoving = false;
    private bool _isFacingRight = true;
    // Властивість для отримання поточної швидкості руху
    public float CurrentMoveSpeed
    {
        get
        {
            if (CanMove)
            {
                if (IsMoving && !touchingDirections.IsOnWall)
                {
                    if (touchingDirections.IsGrounded)
                    {
                        return walkSpeed;
                    }
                    else
                    {
                        // Швидкість руху у повітрі
                        return airWalkSpeed;
                    }
                }
            }
            else
            {
                // Безрухість
                return 0;
            }
        }
    }
}

```

```

        {
            // Рух заблоковано
            return 0;
        }
    }
}
// Властивість для визначення, чи гравець рухається
public bool IsMoving
{
    get
    {
        return _isMoving;
    }
    private set
    {
        _isMoving = value;
        animator.SetBool(AnimationStrings.isMoving, value);
    }
}
// Властивість для визначення напрямку вправо
public bool isFacingRight
{
    get
    {
        // Вертається true, якщо напрямок вправо
        return _isFacingRight;
    }
    private set
    {
        if (_isFacingRight != value)
        {
            // Перевертаємо локальний масштаб для обличчя гравця у протилежний
бік
            transform.localScale *= new Vector2(-1, 1);
        }
        _isFacingRight = value;
    }
}
// Властивість для визначення можливості рухатися
public bool CanMove
{
    get
    {
        return animator.GetBool(AnimationStrings.canMove);
    }
}
// Властивість для визначення чи гравець живий
public bool IsAlive
{
    get
    {
        return animator.GetBool(AnimationStrings.isAlive);
    }
}
// Викликається при створенні об'єкта
private void Awake()
{
    rb = GetComponent<Rigidbody2D>();
    animator = GetComponent<Animator>();
    touchingDirections = GetComponent<TouchingDirections>();
    damageable = GetComponent<Damageable>();
    playerInventory = GetComponent<PlayerInventory>();
}
// Викликається при натисканні кнопки взаємодії
public void OnInteraction(InputAction.CallbackContext context)

```

```

{
    // Взаємодія з сундуками
    if (context.started && isNearChest && playerInventory.HasKey() &&
!playerInventory.IsChestOpened)
    {
        if (nearbyChest != null)
        {
            nearbyChest.OpenChest(playerInventory);
        }
    }
    // Взаємодія з клітинами
    if (context.started && isNearCell && playerInventory.HasKey() &&
!playerInventory.IsCellOpened)
    {
        if (nearbyCell != null)
        {
            nearbyCell.OpenCell(playerInventory);
        }
    }
}
// Викликається при зіткненні з тригерами
private void OnTriggerEnter2D(Collider2D other)
{
    // Зона сундука
    if (other.CompareTag("Chest"))
    {
        isNearChest = true;
        nearbyChest = other.GetComponent<ChestController>();
    }
    // Зона клітини
    if (other.CompareTag("Cell"))
    {
        isNearCell = true;
        nearbyCell = other.GetComponent<CellController>();
    }
}
// Викликається при виході з тригерів
private void OnTriggerExit2D(Collider2D other)
{
    // Вихід із зони сундука
    if (other.CompareTag("Chest"))
    {
        isNearChest = false;
        nearbyChest = null; // Скидаємо посилання при виході з зони сундука
    }
    // Вихід із зони клітини
    if (other.CompareTag("Cell"))
    {
        isNearCell = false;
        nearbyCell = null; // Скидаємо посилання при виході з зони клітини
    }
}
// Викликається кожен фіксований кадр
private void FixedUpdate()
{
    // Зменшуємо лічильник часу замикання, якщо гравець у повітрі
    if (!touchingDirections.IsGrounded)
    {
        coyoteCounter -= Time.fixedDeltaTime;
    }
    // Встановлюємо швидкість руху гравця
    if (!damageable.LockVelocity)
        rb.velocity = new Vector2(moveInput.x * CurrentMoveSpeed,
rb.velocity.y);
    animator.SetFloat(AnimationStrings.yVelocity, rb.velocity.y);
}

```

```

    // Відтворюємо анімацію пилу при русі
    if (IsMoving && touchingDirections.IsGrounded && CanMove)
    {
        dust.Play();
    }
}
// Викликається при русі гравця
public void OnMove(InputAction.CallbackContext context)
{
    moveInput = context.ReadValue<Vector2>();
    if (IsAlive)
    {
        IsMoving = moveInput != Vector2.zero;
        SetFacingDirection(moveInput);
        if (touchingDirections.IsGrounded)
        {
            coyoteCounter = coyoteTime;
        }
    }
    else
    {
        IsMoving = false;
    }
}
// Встановлює напрямок обличчя гравця відповідно до напрямку руху
private void SetFacingDirection(Vector2 moveInput)
{
    if (moveInput.x > 0 && !isFacingRight)
    {
        // обличчя праворуч
        isFacingRight = true;
    }
    else if (moveInput.x < 0 && isFacingRight)
    {
        // обличчя ліворуч
        isFacingRight = false;
    }
}
// Викликається при стрибку
public void OnJump(InputAction.CallbackContext context)
{
    if (context.started && (touchingDirections.IsGrounded || coyoteCounter > 0))
    {
        animator.SetTrigger(AnimationStrings.jump);
        rb.velocity = new Vector2(rb.velocity.x, jumpImpulse);
        dust.Play();
        coyoteCounter = 0;
    }
}
// Викликається при отриманні пошкоджень
public void OnHit(int damage, Vector2 knockback)
{
    rb.velocity = new Vector2(knockback.x, rb.velocity.y + knockback.y);
}
}

```