

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ІНТЕРАКТИВНОЇ КАЗУАЛЬНОЇ
МОБІЛЬНОЇ ГРИ ПІД СИСТЕМУ ANDROID З ВИКОРИСТАННЯМ C#
.NET ТА UNITY»**

Виконав: студент 5 курсу, групи ППЗ– 51
спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Запорожченко А.М.

(прізвище та ініціали)

Керівник Аверічев І.М

(прізвище та ініціали)

Рецензент Зінченко О.В

(прізвище та ініціали)

Нормоконтроль Трінтіна Н.А.

(прізвище та ініціали)

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність - 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

О.В. Негоденко

« » 2023 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Запорожченко Антона Михайловича

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка інтерактивної казуальної мобільної гри під систему Android з використанням C# .NET та Unity»

Керівник роботи: Аверічев Ігор Миколайович, кандидат економічних наук

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “24” лютого 2023 року
№26.

2. Строк подання студентом роботи “1” червня 2023 року

3. Вхідні данні до роботи:

- 3.1 Науково-технічна література пов'язана з програмним забезпеченням щодо розробки ігор.

- 3.2 Офіційна документація ігрового рушія Unity.

- 3.3 Офіційна документація Microsoft Visual Studio.

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

- 4.1 Аналіз актуальності та проблематики розроблюваної гри.

- 4.2 Аналіз та вибір інструментів для реалізації обраної гри.

- 4.3 Проектування та реалізація гри.

- 4.4 Тестування та оптимізація гри.
- 5. Перелік демонстраційного матеріалу (назва основних слайдів)
 - 5.1 Титульний слайд.
 - 5.2 Мета, об'єкт, предмет дослідження.
 - 5.3 Задачі дипломної роботи.
 - 5.4 Аналіз аналогів.
 - 5.5 Концепт гри.
 - 5.6 Вимоги до ігрового контенту.
 - 5.7 Програмні засоби та інструменти реалізації.
 - 5.8 Блок-схема ігрового процесу.
 - 5.9 Блок-схема алгоритму основної механіки гри.
 - 5.10 Екранні форми.
 - 5.11 Апробація результатів дослідження.
 - 5.12 Висновки.
 - 5.13 Кінцевий слайд.
- 6. Дата видачі завдання “25” лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	25.02.23	Виконано
2	Дослідження аналогів та актуальності додатку	26.02.23-01.03.23	Виконано
3	Аналіз та вибір інструментів для розробки додатку	01.03.23-14.03.23	Виконано
4	Проектування додатку	15.03.23-20.03.23	Виконано
5	Реалізація додатку	20.03.23 - 04.04.23	Виконано
6	Оформлення роботи, висновки, реферат	04.04.23-15.04.23	Виконано
7	Розробка демонстраційних матеріалів	16.04.23-20.04.23	Виконано
8	Попередній захист	19.05.2023	Виконано
9	Подання роботи в деканат	01.06.2023	Виконано

Студент _____ Запорожченко А.М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Аверічев І.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 49с, 9 рис, 11 джерел

Ключові слова: Уповільнення часу, гра, Unity, C#, Android, геймплей, Asset, .NET, фреймворк, GIT.

Об'єкт дослідження – ігровий процес казуальної гри на платформу Android з використанням рушія Unity.

Предмет дослідження – інтерактивність та динаміка в казуальних мобільних іграх.

Мета роботи – зменшення порогу входу у гру за допомогою використання механіки уповільнення часу.

У дипломній роботі було проаналізовано існуючі аналоги та описані переваги та недоліки програмних інструментів, які використовувалися для створення додатку. Створені прототипи та вилучені кращі ідеї для створення власного концепту гри, описана роль уповільнення часу у грі.

Додаток реалізований за допомогою мови програмування C#, з використанням ігрового рушія Unity для розробки додатку на мобільний пристрій. Для створення візуальних елементів гри були використано Adobe Photoshop.

Мобільна гра може буди засобом що ненадовго відволіктись, або повністю зануритись у надзвичайно динамічний геймплей, тренуючи реакцію та отримувати задоволення.

Ключові слова: Казуальна гра, Unity, C#, монетизація, Android, геймплей, Asset, .NET, фреймворк, GIT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП.....	10
1 ТЕОРЕТИЧНА ЧАСТИНА	11
1.1 Казуальні ігри	11
1.2 Актуальність розробки мобільної казуальної гри	12
1.3 Процес створення цікавої гри	13
1.4 Роль зміни швидкості часу у грі	14
1.5 Створення дизайн-документу гри	15
Висновки до першого розділу	16
2 АНАЛІЗ НАЯВНИХ ІГОР ТА ЇХ ОСОБЛИВОСТЕЙ.....	17
2.1 Огляд казуальних ігор для мобільних платформ	17
2.2 Аналіз геймплею та механіки ігор.....	18
2.3 Огляд візуальної та звукової складових ігор.....	19
2.4 Визначення концепту та геймплею гри	20
Висновки до другого розділу	25
3 ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ	26
3.1 Огляд мов програмування та ігрових рушіїв	26
3.2 Обґрунтування вибору C# .NET та Unity.....	28
3.3 URP для Unity	31
3.4 Photoshop	32
Висновки до третього розділу.....	33
4 ПРОЕКТУВАННЯ ГРИ	35
4.1 Створення сценарію гри	35
4.2 Взаємодія з користувачем	36
4.3 Розробка архітектури гри	38
4.4 Проблеми створення гри з обмеженим бюджетом.....	42
4.5 Двовимірні ігри.....	44
Висновки до четвертого розділу.....	45
5 РЕАЛІЗАЦІЯ ГРИ.....	47
5.1 Реалізація уповільнення часу у Unity	47
5.2 Система часток	48
5.3 Фізика об'єктів	50
5.4 Типи колайдерів	51
5.5 Ігрові анімації	53
5.6 Оптимізація	55
Висновки до п'ятого розділу	59
ВИСНОВКИ	60
ПЕРЕЛІК ПОСИЛАНЬ.....	61
ДОДАТОК А	62

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Asset(ассет) – візуальна, звукова або програмна частина ігрового контенту, яка оброблюється грою для її взаємодії зі сценою.

Скрипт – це послідовність дій, описаних за допомогою мови програмування.

Колайдер – двовимірний або тривимірний об'єкт, що описує форму об'єкту на сцені.

Рендеринг – відображення елементів сцени на екрані.

ВСТУП

Сучасний розвиток технологій зробив мобільні пристрої незамінними частинами нашого життя, зокрема для розваг та відпочинку. Ринок мобільних ігор продовжує зростати, тому розробка ігор для мобільних платформ є важливою та актуальною задачею в галузі розробки програмного забезпечення.

Зростаюча популярність мобільних пристроїв зробили мобільні ігри однією з найбільш успішних галузей розваг. Щороку ринок мобільних ігор зростає на кілька мільярдів доларів, і за даними досліджень, в 2023 році він досягне обсягу більше 100 мільярдів доларів.

Казуальні ігри, в які можна грати відразу після завантаження та не вимагають великої кількості часу та зусиль, є одними з найбільш популярних жанрів мобільних ігор. Оскільки користувачі шукають прості та зрозумілі ігри, розробка інтерактивної казуальної мобільної гри є привабливою та перспективною задачею.

Це свідчить про те, що розробка ігор для мобільних платформ є важливою та актуальною задачею в галузі розробки програмного забезпечення. Окрім того, мобільні ігри є важливим інструментом маркетингу, який може допомогти привернути нових користувачів до бренду та збільшити продажі продуктів чи послуг.

Крім того, інтерактивні казуальні мобільні ігри зарекомендували себе як один з найбільш популярних жанрів серед користувачів мобільних пристроїв. Ці ігри відрізняються легкістю та доступністю, що робить їх привабливими для широкого кола гравців. Окрім того, з ростом популярності мобільних пристроїв та швидким розвитком технологій, запит на розробку мобільних ігор зростає.

Зазвичай казуальним іграм не вистачає динамічності, чи во навпаки заважкі і незрозумілі для звичайного користувача, у моїй роботі враховані та виправлені ці дві проблеми за допомогою використання динамічної зміни плинності часу у грі.

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Казуальні ігри

Казуальні ігри – це жанр комп'ютерних ігор, який призначений для масової аудиторії, включаючи тих, хто не має великого досвіду в геймінгу або не має багато часу на грати. Ці ігри зазвичай мають простий геймплей, легке керування і вимагають невеликого зусилля для того, щоб почати грати. Вони можуть бути доступні на різних платформах, таких як комп'ютери, мобільні пристрої та ігрові консолі.

Основна мета казуальних ігор – забезпечити короткий, приємний і веселий ігровий досвід. Ці ігри часто пропонують короткі рівні, прості головоломки або завдання, які можна легко виконати. Вони можуть мати яскраву і кольорову графіку, анімацію і звукові ефекти, що сприяють більшій привабливості для гравців включають в себе широкий спектр жанрів, таких як головоломки, аркади, логічні ігри, пошук предметів, симулятори, карточні ігри та багато інших. Вони можуть мати різні тематики, включаючи пригоди, спорт, природу, тварин, ферми, кулінарію, моду, музику тощо.

Такий тип ігор також став популярними на мобільних пристроях, завдяки своїй доступності та простоті. Вони дозволяють гравцям займатися геймінгом в будь-який час і в будь-якому місці.

Казуальні ігри мають на меті забезпечити короткий, веселий і безпечний геймплей, який можна насолоджуватися без надмірного напруження або вкладання великої кількості часу. Вони надають можливість розважитися, відпочити і насолодитися грою без складних правил або вимог до гравця, створюють невимушену та приємну геймінгову атмосферу, де гравці можуть насолоджуватися грою без надмірних зусиль та витрати часу. Вони підходять для широкої аудиторії і дозволяють кожному знайти задоволення від геймінгу.

1.2 Актуальність розробки мобільної казуальної гри

Казуальні ігри, в які можна грати відразу після завантаження та не вимагають великої кількості часу та зусиль, є одними з найбільш популярних жанрів мобільних ігор. Оскільки користувачі шукають прості та зрозумілі ігри, розробка інтерактивної казуальної мобільної гри є привабливою та перспективною задачею.

Також варто зазначити, що використання мови програмування C# .NET та ігрового движка Unity є досить популярним варіантом для розробки ігор для мобільних пристроїв, оскільки вони надають розробникам широкі можливості для створення якісного ігрового досвіду.

Тому, розробка інтерактивної казуальної мобільної гри під систему Android має високу актуальність та може стати важливим кроком для розробки та популяризації мобільних ігор. Результати дослідження можуть бути корисні для розробників мобільних ігор, що працюють з платформою Android, а також для студентів, які планують займатися розробкою ігор та хочуть поглибити свої знання в цій галузі.

Розробка мобільної гри є складним технічним завданням, яке включає в себе різноманітні аспекти, такі як програмування, графіка, дизайн та звук. Таким чином, розробка інтерактивної казуальної мобільної гри з використанням C# .NET та Unity є відмінним випробуванням технічних навичок та може стати цікавим викликом для будь-якого розробника програмного забезпечення.

Тому, дана тема дослідження є актуальною та перспективною в галузі розробки мобільних ігор та може мати великий потенціал для подальших досліджень і розвитку. Інтерактивні казуальні мобільні ігри зарекомендували себе як один з найбільш популярних жанрів серед користувачів мобільних пристроїв. Ці ігри відрізняються легкістю та доступністю, що робить їх привабливими для широкого кола гравців. Окрім того, з ростом популярності мобільних пристроїв та швидким розвитком технологій, запит на розробку мобільних ігор зростає.

Дослідження даної теми може допомогти в розвитку галузі мобільних ігор та підвищенні конкурентоздатності розробників на цьому ринку.

Актуальність теми дослідження зумовлена потребами ринку мобільних ігор, широкою популярністю жанру інтерактивних казуальних мобільних ігор, а також великою складністю процесу розробки мобільних ігор та потребою в кваліфікованих фахівцях у цій галузі. Гра втамувати запити ринку та задовольнити потреби користувачів.

1.3 Процес створення цікавої гри

Головною метою цікавої гри є надання користувачам можливості розважитись та провести час за допомогою цікавої та захоплюючої гри на їхній мобільній платформі.

Для досягнення цієї мети необхідно вирішити ряд завдань.

- **Перше завдання** полягає у проведенні аналізу наявних ігор для мобільних платформ, що мають популярність серед користувачів та визначити їхні переваги та недоліки. Цей аналіз дозволить визначити основні тенденції на ринку мобільних ігор та допоможе у створенні привабливої та цікавої гри для користувачів.

- **Друге завдання** є розробка концепції мобільної гри, яка би відповідала вимогам користувачів та відповідала основним тенденціям на ринку мобільних ігор. Для цього необхідно враховувати елементи геймдизайну та розробити геймплей, який би відповідав вимогам мобільної платформи та забезпечував захоплюючий та цікавий геймплей для користувачів.

- **Третє завдання** полягає в реалізації мобільної гри з використанням мови програмування C# .NET та ігрового движка Unity. Для цього необхідно створити необхідні компоненти гри, забезпечити взаємодію з користувачем та обробку даних. Це завдання дозволить створити функціональну та привабливу гру для користувачів.

- **Четверте завдання** полягає в тестуванні та виправленні помилок у розробленій мобільній грі. Для цього необхідно провести ретельне тестування гри на різних пристроях та в різних умовах, щоб виявити можливі помилки та недоліки.

Після виявлення проблем необхідно виправити їх та перевірити, чи не впливає це на роботу гри в цілому.

1.4 Роль зміни швидкості часу у грі

Уповільнення часу, зустрічається тільки в іграх, оскільки вони створюють віртуальний світ, де можна контролювати події та дії. Така можливість виникає завдяки програмній реалізації, де час може бути модифікований для створення сповільненого або прискореного ефекту.

У іграх реалізується спеціальна логіка, що дозволяє змінювати швидкість подій та руху об'єктів. Це може бути досягнуто за допомогою затримки фреймів, зміни часового масштабу або збільшення частоти оновлення гри.

В інших аспектах життя, поза ігровим середовищем, такі можливості неможливі, оскільки реальний світ підпорядкований природним законам та обмежений фізичними параметрами. Наприклад, уповільнення часу у реальному світі вимагало б зміни фундаментальних фізичних констант, що є практично неможливим.

Тому зміна часу, як явище, яке можна контролювати, здійснюється переважно у віртуальних ігрових середовищах, де можна маніпулювати факторами часу для створення бажаного ефекту.

Загальні можливості зміни плинності часу в іграх можуть включати ефекти битви в деяких іграх уповільнення часу використовується для показу ефектів битви більш детально. Це дає можливість гравцеві бачити і реагувати на швидкі рухи і атаки ворогів, а також планувати свої дії, дозволяє гравцеві аналізувати ситуацію, розглядати варіанти та приймати розсудливі рішення. Це особливо корисно в стратегічних або тактичних іграх, де гравець повинен розглянути багато факторів та планувати свої дії. Такий підхід особливо важливий в швидких іграх, де реакція гравця грає вирішальну роль, така механіка може використовуватись для створення ефектів настрою та атмосфери в грі. Наприклад, у моменти напруження або емоційних випробувань, уповільнення часу може підкреслити драматичність події

та зробити гравця більш зосередженим.

Зміна плинності часу додає глибину та варіативність до динаміки гри, дозволяючи гравцям насолоджуватися більш виваженим та збалансованим геймплейним досвідом. Він може бути використаний для створення ефектних сцен, підкреслення емоційних моментів та підвищення загального рівня захопленості гри.

Щоб знизити поріг входу гравця у гру в першу чергу було використано уповільнення часу. Гра є дуже динамічною, тому недосвідчені гравці можуть відчувати стрес та обурення, а зменшена швидкість плинності часу час надає змогу гравцеві можливість подумати над своїм наступним кроком.

1.5 Створення дизайн-документу гри

Створення дизайн-документу гри - це процес опису всіх аспектів гри перед початком її розробки. Дизайн-документ включає в себе концепцію гри, геймплейні механіки, вміст, характеристики персонажів, архітектуру рівнів, звуковий дизайн та інші важливі елементи.

У початковому етапі визначається загальна ідея гри, її жанр, цільова аудиторія та основні концептуальні характеристики. Затим проектується геймплей - описуються механіки гри, правила, системи прогресії, рівні та завдання, які гравець повинен виконати. Важливо врахувати взаємодію гравця з оточенням, можливості розвитку персонажів та розмаїття геймплейних ситуацій.

Окремий аспект дизайн-документу - вміст гри. Описуються персонажі, їхні характеристики, зовнішній вигляд та поведінка, світи, локації, об'єкти, з якими гравець буде взаємодіяти, а також діалоги, завдання та інші елементи, що впливають на вміст гри. Можна вказати стиль мистецтва, естетику та звуковий дизайн, що створюють відповідну атмосферу.

Дизайн-документ може включати архітектуру рівнів, описуючи структуру та послідовність рівнів, їх складові елементи та особливості. Важливо врахувати баланс складності, прогресію геймплею та наявність викликів для гравця.

Інформацію про монетизацію гри, маркетингову стратегію, плани тестування та важливі дедлайни також мають місце у дизайн документі. Він служить основним довідником для всіх членів команди розробки, допомагаючи зберегти узгодженість та спрямованість у процесі створення гри, вимагає багато роздумів, планування та дослідження, допомагає зрозуміти всі аспекти гри перед тим, як розпочати фактичну розробку, та забезпечує цілісність та якість проекту.

Висновки до першого розділу

У розділі було проаналізовано актуальність розробки мобільних казуальних. Зростання популярності смартфонів та планшетів, збільшення кількості користувачів мобільних додатків, створює великий попит на ігри цього типу.

Досліджено, що створення цікавої гри - складний і багатоетапний процес, який включає в себе різні етапи, такі як концепція гри, проектування, розробка, тестування та випуск. Ключовими факторами успіху є оригінальність ідеї, захоплюючий геймплей, привабливі графічні ефекти та висока якість виконання.

Описано чому зміна швидкості часу може бути цікавим механізмом у грі, що додає глибину інтерактивності. Це може включати уповільнення часу для допомоги гравцеві у складних ситуаціях або прискорення часу для прискорення прогресу гри. Зміна швидкості часу також може використовуватися для створення унікальних головоломок або геймплейних викликів.

Важливим кроком у процесі розробки гри - створення дизайн-документу. Він включає в себе опис геймплею, персонажів, світу гри, архітектури, механік гри та інші важливі деталі. Дизайн-документ допомагає команді розробників узгодити спільне бачення гри і забезпечити її правильну реалізацію.

Всі розглянуті теми є важливими для успішної розробки мобільної казуальної гри. Актуальність ринку мобільних ігор, правильне планування та процес розробки, використання інноваційних механік гри та детальний дизайн-документ - усе це впливає на якість гри та задоволення гравців.

2 АНАЛІЗ НАЯВНИХ ІГОР ТА ЇХ ОСОБЛИВОСТЕЙ

2.1 Огляд казуальних ігор для мобільних платформ

Для досягнення мети дослідження необхідно провести аналіз наявних ігор для мобільних платформ, що мають популярність серед користувачів та визначити їхні переваги та недоліки.

Казуальні ігри для мобільних платформ є дуже популярними серед користувачів, оскільки вони дозволяють провести час за грою в будь-якому місці та в будь-який час. Такі ігри характеризуються простотою геймплею, легкістю в освоєнні та різноманітністю жанрів. Для більшої зручності, казуальні ігри зазвичай мають простий інтерфейс та зрозумілі правила гри.

Аркадні ігри - основний жанр казуальних ігор для мобільних платформ. Цей жанр включає в себе ігри, які базуються на старих класичних іграх, таких як Pac-Man, Tetris або Space Invaders. Також до аркадних ігор відносяться ігри з простою механікою, такі як Flappy Bird або Fruit Ninja.

Інший популярний жанр - це пазли. Цей жанр включає в себе ігри, де користувач повинен збирати пазли, складаючи картинку з дрібних елементів. До пазлів відносяться такі ігри, як Candy Crush або Bejeweled, до казуальних ігор відносяться ігри, що базуються на фізиці. У таких іграх користувач повинен вирішувати головоломки та задачі, використовуючи знання фізики. Прикладом таких ігор є Angry Birds або Cut the Rope.

Було з'ясовано, в більшості казуальних ігор використовується спрощена графіка, яка не потребує великої обробки на стороні пристрою користувача та не впливає на продуктивність гри. Багато ігор мають прості сюжети, що не вимагають глибокого аналізу та розуміння, що є важливим чинником в успіху гри, особливо у сегменті мобільних ігор, де користувачі грають на коротких інтервалах часу.

Під час огляду було виявлено використання спрощеної графіки та звукового дизайну, що дозволяє оптимізувати продуктивність гри та забезпечити максимально комфортний ігровий процес на мобільних пристроях.

Популярність та особливість казуальних ігор для мобільних платформ наштовхує на розробку власної гри та стає успішним проектом, враховуючи особливості та тенденції ринку.

Основні ресурси казуальних ігор для дослідження:

- **Google Play** - це найбільш популярний магазин додатків для Android, де можна знайти безліч казуальних ігор. Тут можна знайти як безкоштовні, так і платні ігри.
- **App Store** - це магазин додатків для iOS, де також можна знайти безліч казуальних ігор. Тут також є як безкоштовні, так і платні ігри.
- **Steam** - це платформа для комп'ютерних ігор, на якій можна знайти казуальні ігри для досліджень. Тут є як безкоштовні, так і платні ігри.
- **itch.io** - це онлайн-магазин, де можна знайти безкоштовні та платні ігри для різних платформ, включаючи Android та iOS.
- **Kongregate** - це сайт, де можна знайти безкоштовні браузерні ігри для досліджень. Тут є багато казуальних ігор, які можна грати прямо в браузері.
- **Game Jolt** - це ще один онлайн-магазин, де можна знайти безкоштовні та платні ігри для різних платформ, включаючи Android та iOS.
- **GameSpot** - це сайт, який присвячений іграм, де можна знайти огляди та рецензії на різні казуальні ігри для досліджень.
- **IGN** - це ще один сайт, який присвячений іграм, де можна знайти огляди та рецензії на різні казуальні ігри для досліджень.

2.2 Аналіз геймплею та механіки ігор

Геймплей - це сукупність механік та елементів гри, які забезпечують гравцеві взаємодію з грою та розуміння її механіки. Механіка - це набір правил, за якими працює гра. Ці дві складові пов'язані між собою та визначають, як гра взаємодіє з гравцем та який досвід вона надає. Аналіз геймплею та механіки ігор є важливим етапом у розробці ігор, він дає змогу розробникам зрозуміти, що робить гру захоплюючою, цікавою та успішною. Цей аналіз дозволяє виявити проблеми та

недоліки у геймплеї, які можуть зменшити задоволення від гри та її популярність.

Основні елементи геймплею включають:

- **Цілі та завдання гри** - чітке визначення мети гри та завдань, які гравець повинен виконати, є ключовими елементами геймплею.
- **Механіки гри** - правила та інструменти, які гравець використовує для досягнення мети гри та виконання завдань.
- **Елементи гри** - об'єкти, які гравець взаємодіє з ними, такі як персонажі, предмети, середовище.
- **Звук та графіка** - якість звуку та графіки може суттєво впливати на сприйняття гравцем гри та насолоду від гри.
- **Інтерфейс користувача** - зручність та зрозумілість інтерфейсу користувача, таких як меню, налаштування, діалогові вікна, є важливими чинниками успіху гри.

У підвищенні привабливості гри для користувачів допомагає аналіз геймплею та механіки ігор. Крім того, аналіз геймплею та механіки ігор може допомогти у підвищенні грифів привабливості гри для користувачів, оскільки з'ясується, які елементи геймплею та механіки ігор найбільше цікавлять користувачів і можуть зробити гру більш захоплюючою та привабливою. Наприклад, якщо більшість популярних ігор мають велику кількість рівнів, то можливо варто врахувати це при розробці власної гри і включити багато різних рівнів з різними завданнями та складнощами.

2.3 Огляд візуальної та звукової складових ігор

Важливий елемент казуальних ігор - візуальна та звукова складові, які включають в собі графіку, дизайн персонажів та елементи гри, анімацію, музику, звукові ефекти та голосові команди.

При розробці необхідно визначити стиль графіки та дизайну, які відповідали б тематиці гри та були привабливими для користувачів. Наприклад, якщо це гра про космос, то можуть бути використані яскраві кольори та візуальні ефекти, щоб

передати відчуття космічної подорожі. Також використовують яскраву та привабливу графіку з великою кількістю деталей, що допомагає залучати користувачів та створює позитивний настрій.

Звукова складова важлива для створення атмосфери гри. Наприклад, у грі про зомбі, звукові ефекти можуть бути страшними та моторошними, що допомагає залучати увагу користувачів та створює відчуття напруги та неспокою. У той же час, у грі про ферму, звукові ефекти можуть бути спокійними та релаксуючими, що допомагає створити позитивну атмосферу та залучити користувачів до гри.

Під час розробки необхідно враховувати, що візуальна та звукова складові не повинні переважати над геймплеєм та механікою гри, але повинні допомагати їм. Графіка та звукові ефекти повинні бути відповідні тематиці та жанру гри, але не повинні перешкоджати її основному завданню - забезпечити гравцеві захоплюючий та веселий досвід гри.

Необхідно враховувати тематику гри та потреби цільової аудиторії при розробці казуальної мобільної гри з відповідною графікою та звуковими ефектами, та для створення привабливого та цікавого ігрового досвіду. Деталі графіки, дизайн персонажів та елементів гри, анімації, музики та звукових ефектів потрібно вирішувати в залежності від тематики та жанру гри. Наприклад, якщо це гра про пригоди, то може бути використана анімація, щоб передати рух персонажів та елементів гри, а музика може бути енергійною та мотивуючою, щоб допомогти гравцеві відчути динаміку гри.

2.4 Визначення концепту та геймплею гри

Казуальна гра - це тип відеоігор, який спрямований на привернення якомога більшої аудиторії, включаючи людей, які раніше не грали в відеоігри. Такі ігри мають простий та зрозумілий геймплей, легкий рівень складності та доступність для гравців будь-якого рівня.

Геймплей казуальної гри характеризується простими та зрозумілими механіками, які гравці можуть легко освоїти. Ці ігри зазвичай мають короткий час

гри, низький рівень складності та можуть бути використані для розваги або відпочинку.

Концепт може бути базований на різних жанрах, включаючи пазли, головоломки, аркади, стратегії та інші. Проте, незалежно від жанру, казуальні ігри зазвичай пропонують простий та зрозумілий інтерфейс, що дає можливість гравцеві швидко освоїти гру.

Для мобільних пристроїв та комп'ютерів казуальні ігри можуть бути безкоштовними, та містити внутрішні покупки, дозволяючи гравцям отримувати додаткові можливості та бонуси. При цьому вони мають просту та зрозумілу графіку та звукове оформлення.

Розробники наполегливо працюють, щоб привернути увагу гравців і надати їм якісні та цікаві ігрові враження. Мобільні ігри стають все популярнішими і поширюються серед користувачів смартфонів та планшетів.

При розробці мобільної гри, тематика гри має велике значення. Вибір теми може визначити успіх гри, оскільки вона повинна привертати увагу та інтересувати гравців. Для створення представленої мобільної гри обираємо японську тематику – гру про ніндзя.

Ніндзя - це фігури, які мають високу популярність в Японії та по всьому світу. Вони є символом сили, вправності та таємничості. З ніндзя пов'язано безліч легенд та оповідань, і саме ця тема надихнула мене на створення гри.

Концепт гри полягатиме в тому, що гравець буде нескінченно пересуватись по одній траєкторії та бити ворогів, уникаючи перешкоди, встановлюючи все нові рекорди. Узагальнюючи – це нескінченний раннер.

Нескінченні ігри-ранери стали дуже популярними серед геймерів останніх років. Ці ігри дозволяють гравцям насолоджуватися неперервним геймплеєм, де вони повинні пройти якомога більше відстані, уникаючи перешкод і збирати бонуси. Одна зі складових таких ігор - швидкість, що зробила їх більш динамічними та захопливими.

Нескінченна гра-ранер - це гра, в якій гравцю потрібно пройти якомога більшу відстань, збираючи монети та уникаючи перешкод, які з'являються на його

шляху. Ця гра заснована на безкінечному руху героя, який просто не зупиняється, тому гравці повинні бути дуже уважними та вмілими, щоб не потрапити в пастку.

Подолати якомога більшу відстань і отримати максимальну кількість балів – це головна мета ігор такого жанру. Чим далі гравець віддаляється від початку гри, тим складнішим стає його завдання, оскільки на його шляху з'являються все нові й нові перешкоди. Поступово гравець отримує нові можливості, які допоможуть йому подолати ще більшу відстань.

Такі ігри мають яскраву та кольорову графіку, яка сприяє настрою гравця і робить гру більш захопливою. Іноді ці ігри мають різні тематики, від наукової фантастики до пригодницьких історій, що надає їм різноманітності.

Однією з найбільших переваг нескінченної гри-ранера є те, що вона не має кінця. Гравець може грати безкоштовно стільки часу, скільки захоче, адже гра не має фіксованого закінчення. Це робить цю гру ідеальною для тих, хто хоче насолоджуватися геймплеєм без обмежень і додаткових витрат.

Багато з цих ігор можна грати на мобільних пристроях, що дозволяє гравцям грати в будь-який час і в будь-якому місці, вони не потребують великої кількості часу на виконання окремого завдання, що робить їх ідеальними для тих, хто має обмежені можливості вільного часу.

Нескінченні ігри-ранери будуть корисні для тих, хто хоче тренувати свою реакцію та координацію рухів. Граючи в такі ігри, гравці можуть покращити свої навички управління та реагування на непередбачувані ситуації, що може бути корисно в реальному житті.

Багато таких ігор мають лідерборди, на яких можна порівнювати свої результати з результатами інших гравців з усього світу що є викликом для тих, хто любить конкуренцію та бажає встановлювати рекорди. Це додає грі елементів змагання та стимулює гравців розвивати свої навички.



Рисунок 2.1 – Знімок екрану гри “Frogger”

Поява ігор та створення аркадної гри Frogger розпочалася в 1981 році, коли була створена аркадна гра Frogger. У цій грі гравець керує жабою, яку потрібно перевести через дорогу та річку, уникати автомобілів та крокодилів, щоб дістатися до своєї цілі.

З часом жанри ігор розвивалися та вдосконалювалися. У 1990-х роках були створені такі ігри-ранери, як Sonic the Hedgehog та Super Mario World, які стали настільки популярними, що до цього дня залишаються улюбленими серед гравців.

Протягом наступних років ігри-ранери стали досить популярними на мобільних пристроях та в Інтернеті. У 2009 році компанія Halfbrick Studios випустила гру Fruit Ninja, яка стала однією з перших ігор-ранерів на мобільних пристроях. У 2011 році була створена гра Temple Run, яка стала дуже популярною серед гравців, і з тих пір гра-ранер стала одним з найпопулярніших жанрів мобільних ігор.

Сьогодні ранери мають безліч варіацій та різноманітних тематик, від класичних швидких ігор до більш складних та вимогливих. Нескінченна гра-ранер - це чудовий спосіб для гравців провести свій вільний час та відпочити від повсякденних справ. Ігри-ранери продовжують розвиватися та еволюціонувати, і ми можемо очікувати ще більш захоплюючі та цікаві ігри у майбутньому.

Нескінченні ранери - це чудовий вибір для тих, хто хоче насолоджуватися безкінечним геймплеєм, поліпшити свої навички та відчутти азарт конкуренції. Ці

ігри доступні на багатьох платформах, вони прості в освоєнні та загалом динамічні та цікаві.

Запропонована гра заснована на швидкості та точності рухів. Гравець керує ніндзя, який повинен проходити гру, знищуючи ворогів та уникати їхніх атак. Головна особливість гри полягає в тому, що гра буде прискорюватись з часом, тому гравець повинен діяти дуже швидко та точно.

Аналізу маркетів консольних та комп'ютерних ігор виявив схожу по задуму гру Katana Zero, маркети мобільних ігор не виявив схожих за задумкою проєктів.



Рисунок 2.2 – Знімок екрану гри “Katana Zero ”

Katana Zero - це 2D екшн-платформер, розроблений студією Askiisoft і випущений у 2019 році для ПК та Nintendo Switch. Гравець виступає в ролі асасина з катаною, який отримує завдання від свого роботодавця та виконує їх, пробиваючись через численних ворогів.

Головними особливостями гри є її експресивна візуальна графіка, підкріплена відповідним саундтреком, і нелінійний сюжет, що залежить від виборів, які гравець робить під час гри. Гра також використовує механіку часу, що дозволяє гравцю повторно відтворювати свої дії, якщо він помилився.

Гра має декілька рівнів складності, що дозволяє гравцям з різним рівнем досвіду насолоджуватися нею. У грі є багато різноманітних ворогів та босів, кожен з яких має свої унікальні характеристики та тактику бою.

Висновки до другого розділу

У розділі було проаналізовано чому казуальні ігри для мобільних платформ є дуже популярними серед гравців. Вони зручні, доступні і прості у використанні. Ці ігри зазвичай мають простий геймплей, швидкі сесії гри і часто використовуються для розваги від часу до часу. Багато казуальних ігор мають простий дизайн, який приваблює широку аудиторію і не вимагає великих обчислювальних ресурсів мобільних пристроїв.

Виявлено, що геймплей є основною складовою кожної гри і визначає досвід гравця. Механіка ігор описує систему правил і взаємодію гравця з віртуальним середовищем. Гармонійне поєднання геймплею та механіки є важливим для створення захоплюючих ігрових досвідів.

Візуальна складова включає графіку, дизайн персонажів та об'єктів у грі. Звукова складова охоплює музику, звуки та діалоги, які доповнюють і покращують геймплей. Якість візуальних та звукових складових може суттєво впливати на іммерсивність та задоволення від гри.

Було розроблено концепт гри, що визначає тематику, історію та цілі гри. Геймплей описує спосіб гри, включаючи контроль гравця, взаємодію з оточенням та завданнями. Успішна гра повинна мати цікавий та захоплюючий геймплей, який відповідає концепту і задовольняє очікування гравців.

Аналіз цих тем допомагає краще розуміти складові казуальних ігор для мобільних платформ, процес їх створення та вплив на гравців. Оцінка геймплею, механіки, візуальної та звукової складових дозволяє створювати захоплюючі ігри, задовольняючи потреби широкої аудиторії гравців.

3 ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЙ

3.1 Огляд мов програмування та ігрових рушіїв

При розробці казуальних ігор, використання відповідних мов програмування та ігрових рушіїв дуже важливе. Для розробки мобільних ігор під Android-платформу, одним з найпопулярніших ігрових рушіїв є Unity. Unity є потужним ігровим движком, який дозволяє розробляти як 2D, так і 3D ігри. Окрім цього, Unity має велику кількість різноманітних інструментів для розробки графіки, аудіо та інших елементів гри, що робить його ідеальним вибором для розробки казуальних ігор.

Однією з основних мов програмування для розробки ігор на Unity є C#. C# - це мова програмування, яка є однією з найбільш популярних та потужних мов для розробки ігор. Вона має велику кількість функцій та бібліотек, що дозволяє швидко та ефективно розробляти як прості, так і складні ігри. Крім того, C# має велику кількість документації та прикладів, що дозволяє легко навчитися її використовувати.

Окрім Unity та C#, для розробки використовують інші ігрові рушії та мови програмування, такі як Unreal Engine, Java, Swift та інші. Кожен з них має свої переваги та недоліки, тому перед початком розробки варто ретельно вивчити всі можливі варіанти та обрати той, який найкраще підходить для даного проекту.

Використання відповідних ігрових рушіїв та мов програмування є ключовим етапом при створенні казуальних ігор, оскільки це впливає на швидкість та ефективність розробки, а також на якість готової гри та її можливості. Наприклад, якщо вам потрібно розробити гру з великою кількістю складних візуальних ефектів, то Unreal Engine може бути кращим вибором, оскільки він має потужну систему візуалізації. Якщо ж вам потрібно розробити мобільну гру, то Unity може бути кращим вибором, оскільки він добре підходить для розробки ігор на мобільних платформах.

Також важливо враховувати кваліфікацію та досвід розробника, оскільки

різні ігрові рушії та мови програмування можуть вимагати різних навичок та рівня складності. Наприклад, Unreal Engine може бути складнішим у використанні для новачків, тоді як Unity може бути більш простим та доступним.

Вибір ігрового рушія та мови програмування залежить від конкретної задачі, потреб розробника та його кваліфікації та досвіду. Ретельне вивчення можливих варіантів та порівняння їх переваг та недоліків дозволить обрати оптимальний варіант для успішної розробки казуальної гри. Якщо йдеться про розробку мобільної гри під Android з використанням C# .NET, то Unreal Engine не є найбільш оптимальним вибором для цього проекту. Unreal Engine підтримує мову програмування C++, яка не є мовою програмування за замовчуванням для платформи Android. Хоча Unreal Engine надає можливість використовувати C# шляхом встановлення плагінів, цей процес може бути складним та неефективним, особливо для початківців.

Unreal Engine є більш потужним рушієм, який може бути надмірним для створення казуальної мобільної гри. Він має велику кількість функціональних можливостей, що можуть бути корисними для створення складних та реалістичних ігрових середовищ. Однак для казуальної мобільної гри, яка часто має прості механіки та графіку, Unreal Engine може бути надмірним та ускладнити розробку.

Unity, з іншого боку, є легшим та більш простим у використанні рушієм, який підтримує мову програмування C#. Це дозволяє розробникам швидко створювати казуальні мобільні ігри з простими механіками та привабливою графікою. Крім того, Unity має велику спільноту розробників та надає доступ до великої кількості документації та плагінів, що полегшує розробку та підтримку гри.

Вибір між Unreal Engine та Unity залежить від конкретних потреб проекту та досвіду розробників. У випадку казуальної мобільної гри під Android з використанням C# .NET, Unity є оптимальним вибором.

Найпопулярнішими мовами програмування для розробки мобільних ігор є:

- **Java** - це мова програмування, яка використовується для розробки ігор на платформі Android. Java має велику кількість різноманітних фреймворків, бібліотек та інструментів, що дозволяють розробляти як 2D, так і 3D ігри. Крім

того, вона має велику кількість документації та прикладів, що дозволяє швидко навчитися її використовувати.

- **Swift** - це мова програмування, яка використовується для розробки ігор на платформі iOS. Swift дозволяє розробляти як 2D, так і 3D ігри, а також має велику кількість інструментів для розробки графіки та аудіо. Однією з переваг Swift є його швидкодія, що робить його ідеальним для розробки вимогливих до продуктивності ігор.

- **C#** - це мова програмування, яка використовується для розробки ігор на платформі Android та iOS з використанням Unity. C# є однією з найпопулярніших мов програмування для розробки ігор, оскільки має велику кількість функцій та бібліотек, що дозволяють швидко та ефективно розробляти як прості, так і складні ігри.

- **Kotlin** - це мова програмування, яка також використовується для розробки ігор на платформі Android. Kotlin має синтаксис, який дуже схожий на Java, але має велику кількість функцій та інструментів для розробки, що робить його більш продуктивним за Java. Крім того, Kotlin підтримує як Java, так і нативний код, що дозволяє розробляти більш гнучкі та ефективні ігри.

Мови програмування для розробки мобільної гри залежить від багатьох факторів, таких як мета гри, платформа, для якої розробляється гра, досвід розробника та бізнес-вимоги. Перед вибором мови програмування слід вивчити характеристики кожної мови та порівняти їх з вимогами проекту. Також варто враховувати наявність ресурсів для підтримки обраної мови програмування, таких як документація, форуми та готові бібліотеки.

3.2 Обґрунтування вибору C# .NET та Unity

C# є однією з найбільш популярних мов програмування для розробки ігор на різних платформах, включаючи мобільні, комп'ютерні та консольні ігри. Існує декілька причин, чому C# є відмінним вибором для розробки казуальних ігор:

- **Легкість вивчення:** C# є мовою програмування, яка досить легка

вивчити. Вона має простий синтаксис та логічну структуру, що дозволяє швидко навчитися її використовувати. Крім того, C# має велику кількість документації та прикладів, що допомагає новачкам швидко засвоїти мову.

- **Підтримка .NET Framework:** C# є мовою програмування, яка підтримує .NET Framework, що забезпечує швидку та ефективну роботу з багатьма різноманітними бібліотеками та інструментами.

- **Швидкість та ефективність:** C# є мовою програмування, яка дозволяє розробляти досить швидкі та ефективні ігри. Вона має підтримку багатьох технологій, які дозволяють оптимізувати швидкість та продуктивність ігор.

- **Підтримка Unity:** C# є однією з основних мов програмування для розробки ігор на Unity, одному з найпопулярніших ігрових рушіїв. Це означає, що розробники, які використовують C# для розробки ігор на Unity, мають доступ до великої кількості інструментів та бібліотек, які дозволяють розробляти як 2D, так і 3D ігри.

- **Спільнота:** C# має велику спільноту розробників, що дозволяє швидко знайти рішення для більшості проблем, з якими можуть стикнутися розробники. Велика кількість документації та прикладів також допомагає вивчити мову та почати розробку швидше.

C# є об'єктно-орієнтованою мовою програмування, що робить її ідеальною для розробки ігор. Об'єктно-орієнтоване програмування дозволяє створювати об'єкти, які містять дані та функції, що діють з цими даними. Це дозволяє розробникам створювати складні системи, де різні компоненти можуть взаємодіяти між собою та виконувати спільні завдання.

Об'єктно-орієнтоване програмування (ООП) - це парадигма програмування, що базується на концепції об'єктів, які взаємодіють один з одним для виконання різних операцій. Кожен об'єкт містить дані та функції, які забезпечують його функціональність. ООП дозволяє розбити програму на більш малий та зрозумілий код, що дозволяє збільшити ефективність та зробити її більш масштабованою.

У розробці ігор, ООП є основним принципом, що дозволяє створювати різні компоненти гри та інтегрувати їх для створення готової гри. Наприклад, в грі може

бути об'єкт гравця, який має свої дані та функції, інші об'єкти, які взаємодіють з гравцем, та багато інших компонентів, що забезпечують функціональність гри. За допомогою ООП можна створювати класи, які містять різні властивості та функції, що забезпечують роботу різних компонентів гри.

Робота в команді є також важливим аспектом розробки ігор. Вона дозволяє розподілити завдання між різними членами команди та підвищити ефективність розробки. Особливо важливим є співпраця між програмістами та дизайнерами гри, що дозволяє створити гру з більш якісним геймплеєм та візуальними ефектами.

ООП допомагає забезпечити більш ефективну роботу в команді. Кожен член команди може відповідати за свої класи та об'єкти, що забезпечує більшу структурованість коду та легшу його розуміння для інших учасників команди. Крім того, ця парадигма дозволяє створювати більш повторно використовуваний код, що зменшує зусилля, необхідні для розробки нових компонентів гри. Робота в команді також дозволяє проводити регулярні код-рев'ю та тестування, що забезпечує якість коду та готову гру. Загалом, ООП та робота в команді є важливими аспектами розробки ігор, що дозволяють створювати більш складні та якісні продукти.

Ще однією перевагою C# є його інтеграція з іншими технологіями Microsoft, такими як .NET Framework та Windows API. Це дозволяє легко створювати додатки для Windows, а також інтегрувати їх з іншими сервісами, які використовують технології Microsoft.

Парадигми та принципи програмування відіграють важливу роль у розробці ігор. Вони допомагають розробникам створювати більш ефективний та структурований код, який забезпечує відповідну функціональність гри.

Інші парадигми, такі як функціональне програмування, можуть бути корисними у розробці ігор, наприклад, у створенні алгоритмів штучного інтелекту (AI) або в обробці великих об'ємів даних.

Принципи програмування, такі як SOLID, DRY та KISS, також є важливими в розробці ігор. Вони допомагають створювати чистий та структурований код, що забезпечує покращення швидкості розробки, зменшення кількості помилок та полегшення підтримки програмного забезпечення.

Наприклад, принцип SOLID допомагає створювати класи, що мають одну відповідальність та легко змінюються, що дозволяє зменшити кількість помилок та полегшити підтримку коду. Також, принцип DRY (Don't Repeat Yourself) допомагає зменшити дублювання коду, що забезпечує зменшення кількості помилок та полегшення його підтримки.

Використання мови програмування C# є однією з кращих вибірок для розробки казуальних ігор на Unity, завдяки його простоті, ефективності, інтеграції з іншими технологіями та великій спільноті розробників.

3.3 URP для Unity

Стандартне середовище розробки візуальної складової проекту вимагає досить багато часу на налаштування і програмування. В якості альтернативи можна використовувати інше середовище яке має в собі Unity під кодовою назвою URP.

URP (Universal Render Pipeline) - це легка та оптимізована система візуалізації для Unity, яка дозволяє розробникам створювати більш ефективні та візуально привабливі ігри. URP включає в себе багато інструментів для настройки візуалізації, таких як налаштування освітлення, налаштування шейдерів та інші.

URP є частиною Unity, і вона може бути використана для створення різних типів ігор, включаючи 2D та 3D ігри. Вона також дозволяє розробникам створювати ігри для різних платформ, таких як мобільні пристрої, ПК та консолі.

URP має декілька особливостей, які роблять її привабливою для розробників ігор:

- **Ефективність** - URP розроблена з метою забезпечення максимальної продуктивності на різних платформах. Її оптимізована система візуалізації дозволяє розробникам створювати гладкі та привабливі ігри з високою кількістю полігонів та ефектів.
- **Налаштування** - URP має багато налаштувань, які дозволяють розробникам змінювати візуалізацію гри відповідно до їхніх потреб. Це дозволяє розробникам створювати унікальні візуальні ефекти та підлаштовувати гру

під свої потреби.

- **Підтримка на всіх платформах** - URP може бути використана для створення ігор на різних платформах, включаючи мобільні пристрої, ПК та консолі. Це дозволяє розробникам легко портувати свої ігри з однієї платформи на іншу, зберігаючи візуальну якість та продуктивність.
- **Нові можливості** - URP надає розробникам ігор нові можливості для створення візуальних ефектів та ігрових механік. Вона містить підтримку шейдерів, яка дозволяє створювати різні візуальні ефекти, такі як зміна кольору, світлові ефекти, зміна форми та інші.

Додавання різноманітних ефектів до кадрів гри після їх рендерингу досягається шляхом пост-процесингу. Це може бути корисно для створення різних ефектів, таких як розмивання, затінення, кольорні фільтри та інші.

3.4 Photoshop

Photoshop є потужним інструментом для створення графічних елементів і інтерфейсу користувача для ігор. Наведено кілька прикладів того, як можна використовувати Photoshop для створення елементів гри:

- **Створення концептів персонажів та ігрового середовища:** Ігрові концепти зазвичай розробляються у вигляді скетчів та ілюстрацій, які вказують, як буде виглядати персонаж або ігрове середовище. Photoshop дозволяє розширити ці ідеї, створивши детальніші зображення персонажів та ігрових об'єктів.
- **Редагування та оптимізація зображень:** Використання Photoshop дозволяє зробити зображення більш чіткими та яскравими, виправити помилки та змінити розмір зображень для використання їх у грі.
- **Інтерфейсні елементи:** Photoshop дозволяє створювати кнопки та різноманітні меню. Ці елементи можуть бути використані для навігації по меню гри та управління геймплеєм.
- **Створення анімацій та спрайтів:** За допомогою Photoshop можна створювати анімації для персонажів та ігрових об'єктів. Використовуючи

Photoshop, можна створювати спрайти, які відображають різні рухи та позиції персонажів.

- **Створення текстур:** Photoshop дозволяє створювати текстури та тло для ігрових об'єктів. Це може включати створення текстури для землі, стін або неба в ігровому середовищі.

Photoshop є потужним інструментом, який дозволяє розширити та вдосконалити ігровий дизайн. Він дозволяє експортувати створені графічні елементи у різні формати, що спрощує їх інтеграцію у гру. Існує безліч ресурсів та плагінів для Photoshop, які дозволяють розширити його можливості та полегшити процес створення графічних елементів для ігор.

Висновки до третього розділу

У розділі проаналізовано чому саме C# обрано мовою програмування. Це потужний та популярний інструмент, особливо в контексті розробки ігор. Має сучасний синтаксис, багатофункціональну стандартну бібліотеку і підтримку з боку розробників. Unity, з іншого боку, є одним з найпопулярніших ігрових рушіїв, яке підтримує C# як основну мову програмування. Unity надає потужні інструменти для створення ігор у 2D та 3D, а також дозволяє легко розгортати гру на різних платформах.

Зазначено, що URP є легким та оптимізованим рендерером для Unity. Він надає широкі можливості налаштування графічного движка, що дозволяє розробникам досягти високої якості візуальних ефектів у своїх іграх. URP також підтримує різні платформи, включаючи мобільні, що дозволяє створювати гри для широкого кола аудиторії.

Photoshop – це потужний графічний редактор, який знайшов широке застосування в розробці ігор. Він дозволяє створювати та редагувати графічні елементи, текстури, спрайти та інші важливі компоненти візуального дизайну гри.

Вибір мови програмування C# та ігрового рушія Unity є раціональним для розробки ігор. Додатково, використання URP дозволяє досягти високої якості

графіки, а Photoshop є корисним інструментом для створення візуальних ефектів та графічних компонентів гри. Разом ці технології допомагають розробникам створювати захоплюючі та професійні ігри.

4 ПРОЕКТУВАННЯ ГРИ

4.1 Створення сценарію гри

Створення сценарію казуальної гри - це важливий процес, який може забезпечити успішність.

Перед початком розробки сценарію потрібно визначити жанр гри. Чи буде це аркадна гра, головоломка або інша казуальна гра? Це допоможе вам сконцентруватися на тематиці, механіці та інших аспектах гри.

Спочатку потрібно визначити мету гри. Чи будуть гравці збирати бонуси або розв'язувати головоломки? Чи будуть вони досліджувати світ гри чи підкорювати різні рівні? Визначивши мету гри, з'являється змога більш чітко визначити сценарій.

Важливо створити цікавого головного героя. Це може бути персонаж зі своїми унікальними характеристиками, які відрізняють його від інших персонажів в грі.

Сюжет може бути простим або складним, але він повинен бути захопливим для гравців. Використовуйте діалоги, щоб розкрити характери персонажів та розкрити сюжет.

Механіка гри повинна відповідати жанру та меті гри. Визначення ігрових механік на початковому етапі значно спрощують подальшу розробку.

Рівні повинні бути цікавими та захоплюючими, а також відповідати меті гри та механіці гри. Треба враховувати, що рівні повинні бути прогресивними, що означає, що складність буде зростати з кожним рівнем.

Додавання елементів гри, так як різні бонуси, можливість виконувати завдання від інших персонажів, можливість змінювати зовнішність головного героя, зможуть зробити гру більш цікавою та захоплюючою.

Після створення сценарію гри потрібно протестувати гру та переконатись що вона захоплююча та викликає цікавість у гравців. виправлення помилок та додавання нових елементів покращить гру.

4.2 Взаємодія з користувачем

Взаємодія гри з користувачем відбувається за допомогою сенсорного екрану та звуку.

Гравець через сенсорний екран може включати різні методи інтерактивності, такі як дотик, свайп, тапання, мультитач жестів, та інші. Методи використовуються для контролю над персонажами гри, руху камери, стрільби та інших дій в грі.

Сенсорний екран є важливою складовою мобільних ігор, оскільки він дозволяє гравцеві взаємодіяти з грою за допомогою простих жестів та дотиків. Сенсорний екран забезпечує більш прямий та інтуїтивний спосіб керування грою, порівняно з клавіатурою та мишею, що зазвичай використовуються для ігор на комп'ютерах.

Залежно від жанру гри, сенсорний екран може мати різні функції. У іграх з жанру "Match-3" гравець може переміщати елементи на екрані шляхом перетягування їх пальцем, а у іграх з жанру "Runner" гравець може уникати перешкод та стрибати, натискаючи на екран.

Використання жестів виконуються завдяки сенсорному екрану що дозволяє виконувати дії для управління грою. Наприклад, жести "щипок" або "розведення" пальців можуть використовуватись для зміни масштабу екрана, або жести "застосування двох пальців" можуть використовуватись для переміщення камери у грі.

Для мобільних ігор складовою яка керує, відкриває та створює нові та цікаві ігрові елементи є сенсорний екран. Він дозволяє гравцям більш прямо та інтуїтивно керувати грою, і відкриває багато можливостей для творців ігор у створенні нових та цікавих ігрових елементів.

Сенсорний екран дозволяє гравцям взаємодіяти з різними ігровими об'єктами, такими як інтерфейсні елементи, текстові поля, зображення та інші, надаючи гравцям більше контролю над діями в грі. Взаємодія з гравцем через сенсорний екран може допомогти створити більш іммерсивний геймплей, зробити

ігровий процес більш інтуїтивним та зручним для гравців.

Взаємодія з гравцем через сенсорний екран додатково підвищує ігровий досвід для гравців, сприяючи більшій інтерактивності та залученню гравців до гри.

Функція дотику до екрану дозволяє розширити можливості геймплею, Це може включати різні жести, такі як торкання, свайп, кручення та стиснення, які дозволяють виконувати різноманітні дії, такі як переміщення, атака або стрільба. Крім того, сенсорний екран дозволяє використовувати ігри з більшою точністю, що особливо важливо для ігор, де потрібна велика точність керування, наприклад, у стратегічних або головоломкових іграх.

У деяких іграх дотик використовується для створення враження присутності, дозволяючи гравцям взаємодіяти з об'єктами в грі, використовуючи жести, які дуже нагадують дії в реальному житті. Наприклад, в грі-головоломці гравець може перетворювати кубики, розташовані на екрані, за допомогою жестів, що надає враження прямої взаємодії з об'єктами.

Однак наявність сенсорного екрана також може бути проблемою, оскільки деякі гравці можуть виявити його незручним для використання, особливо якщо гра потребує багато точності. Крім того, у деяких випадках сенсорний екран може втратити точність, якщо на нього нанесено багато відбитків пальців, що може стати проблемою для гравців, особливо для тих, хто грає у ігри з високими вимогами до точності керування.

Звук використовують для взаємодії з гравцем в іграх. Особливо це корисно в тих випадках, коли гравець не може або не повинен дивитися на екран. Наприклад, звук може використовуватися для повідомлення гравця про події в грі, таких як зміна рівня, отримання повідомлень або попередження про небезпеку.

Для допомоги гравцю в навігації та орієнтуванні в грі також використано звук. Може використовуватися. Наприклад, звукові ефекти можуть вказувати на те, де знаходяться предмети, які гравець повинен зібрати, або де знаходиться вихід з рівня.

Музикальні ефекти можуть використовуватися для створення настрою гри та забезпечення більш іммерсивного досвіду для гравця. Використання музики може

створювати певний настрій в грі, а звукові ефекти можуть допомогти гравцю більш наочно уявити ігровий світ. Деякі ігри можуть використовувати голосові команди або голосове керування для взаємодії з гравцем.

У деяких іграх звук використаний як ще один спосіб управління грою. Наприклад, гравець може використовувати звукові команди для керування персонажем або іншими ігровими об'єктами.

Взаємодія з гравцем через звук є особливо корисною в мобільних іграх, де звук може бути одним з найважливіших способів взаємодії з гравцем, оскільки сенсорний екран може бути досить малим, а також гравець може знаходитися в руху.

Музика та звукові ефекти використовуватися для створення настрою та атмосфери гри. В моторошних іграх звук може бути використаний для створення напруження та страху. В іграх екшену звук використовують для підсилення відчуття екшену та динаміки гри.

Створення незабутнього досвіду для гравців та покращення геймплею досягаються використанням звуку у грі. Це потужний інструмент для покращення геймплею та створення незабутнього досвіду для гравців.

4.3 Розробка архітектури гри

Розробка будь-якої гри починається з архітектури, яка визначає як організовані та взаємодіють між собою компоненти.

На початковому етапі розробки архітектури гри, потрібно визначити головні компоненти гри, такі як геймплей, графіка, звук, фізика, штучний інтелект, мережевий геймплей та інші. Далі, розробники повинні визначити, як ці компоненти взаємодіють між собою та які є залежності між ними.

Один із підходів до розробки архітектури гри - це розбити гру на окремі модулі або компоненти, які можна розробляти окремо і пізніше об'єднати в одну гру. Цей підхід дозволяє розробникам працювати над різними частинами гри

одночасно та зменшує ризик конфліктів між різними частинами гри.

Інший підхід - це використання шаблону проектування для розробки архітектури гри. Шаблони проектування - це загальні рішення для типових проблем, які можуть виникнути під час розробки програмного забезпечення. Використання шаблонів дозволяє розробникам швидко та ефективно вирішувати типові проблеми та зменшувати кількість помилок під час розробки гри.

Шаблони проектування - це загальні рішення для типових проблем, які можуть виникати під час проектування програмного забезпечення. Вони розробляються та застосовуються для вирішення повторюваних проблем під час проектування програмного забезпечення. Шаблони проектування можуть бути застосовані в різних мовах програмування та різних областях програмування, включаючи розробку ігор. Шаблон проектування у програмуванні іноді називають патерном.

Існує багато типів шаблонів проектування, і кожен з них спрямований на вирішення конкретної проблеми. Найпоширеніші шаблони проектування включають:

- **Singleton** - шаблон, що дозволяє забезпечити, що клас має тільки один екземпляр, і надає глобальну точку доступу до цього екземпляру.
- **Factory Method** - шаблон, який використовується для створення об'єктів без прямого визначення їх класу.
- **Observer** - дозволяє одним об'єктам слідкувати та реагувати на зміни в інших об'єктах.
- **Decorator** - дозволяє динамічно додавати нову функціональність до існуючих об'єктів.
- **Adapter** - це шаблон, який дозволяє використовувати існуючий клас з іншим інтерфейсом.

При проектуванні та реалізації програмного забезпечення зменшення помилок виникає за допомогою патернів проектування та дозволяє зробити код більш зрозумілим та ефективним.

Категорії шаблонного проектування:

- **Шаблон створення об'єктів (Creational Patterns)** - шаблони що допомагають вирішувати проблеми, пов'язані з створенням об'єктів. Наприклад, шаблон Фабрика допомагає створювати об'єкти без прив'язки до конкретних класів.
- **Шаблон структури (Structural Patterns)** - ці шаблони допомагають організувати об'єкти та класи в більш складні структури. Наприклад, паттерн Декоратор допомагає додавати нову функціональність до об'єкту, не змінюючи його основної структури.
- **Шаблон поведінки (Behavioral Patterns)** - допомагають вирішувати проблеми, пов'язані зі співпрацею між об'єктами. Наприклад, паттерн Стратегія допомагає вибирати різні алгоритми в залежності від контексту.

У галузі розробки ігор існує велика кількість шаблонів проектування, які можуть бути використані для розробки архітектури гри. Наприклад, шаблон "Модель-Вид-Контролер" (MVC) допомагає організувати код гри, розділяючи його на три складові: модель (дані), вид (представлення) та контролер (логіка).

Шаблон проектування MVC є дуже популярним у веб-розробці, але він також може бути використаний для розробки ігор. Дозволяє розділити логіку програми на три компоненти: модель (що відповідає за даних і бізнес-логіку), представлення (яке відображає дані користувачу) та контролер (який приймає вхідні дані та взаємодіє з моделлю для оновлення представлення).

Шаблон проектування MVC допомагає відокремити головні компоненти гри, такі як логіку вводу, гравець та інтерфейс користувача, та забезпечує легкість розробки та зберігання коду в хорошому порядку. Він також забезпечує зручний спосіб тестування окремих компонентів гри.

MVC є одним з найпопулярніших шаблонів проектування, який виник у світі розробки програмного забезпечення ще в 1970-х роках. Він був створений для розробки графічних оболонок для мови програмування Smalltalk, а згодом став популярним в інших галузях, таких як веб-розробка та розробка ігор.

Розробники гри можуть також використовувати стандартизовані фреймворки та бібліотеки, які вже містять реалізації різних паттернів проектування та інших компонентів, що зменшує час, необхідний для розробки гри та дозволяє швидше

переходити до розробки головної логіки гри. Вони повинні враховувати масштаб гри та забезпечувати масштабованість архітектури, щоб гра могла працювати як на слабших, так і на потужних комп'ютерах або пристроях.

Один з основних паттернів що був використаний створенні гри, це ін'єкції залежностей.

Zenject - це фреймворк для ін'єкції залежностей в Unity. Він дозволяє забезпечити кращу організацію коду та розбити його на більш малий і зрозумілий модулі, що полегшує розробку та підтримку гри.

Одним із головних переваг використання Zenject є можливість забезпечити ін'єкцію залежностей зі знижкою на зв'язування. Це означає, що залежності можна легко змінювати та замінювати, не змінюючи код класу, який використовує ці залежності. Це забезпечує більшу гнучкість та підтримку змін в коді.

Інша перевага полягає в тому, що Zenject є його можливість створювати різні контексти введення залежностей. Кожен контекст може мати свої власні налаштування та залежності, що дозволяє забезпечити більшу гнучкість у розробці та підтримці гри.

Даний фреймворк дозволяє забезпечити автоматичне введення залежностей, що значно полегшує процес розробки. Це забезпечує більшу продуктивність розробників та зменшує кількість помилок в коді.

Використання Zenject у розробці гри дозволяє забезпечити більшу гнучкість та підтримку змін в коді, що полегшує розробку та підтримку гри. Крім того, це забезпечує більшу продуктивність розробників та зменшує кількість помилок в коді.

Одним з ключових функціональних елементів фреймворку є рефлексія (reflection), яка дозволяє знаходити та автоматично вводити залежності в об'єкти відповідно до їх типу.

Рефлексія - це можливість програмно отримувати інформацію про типи об'єктів та їх властивості за допомогою рефлексії в мовах програмування. Це означає, що з допомогою рефлексії можна виконувати дії над об'єктами на основі інформації про їх тип та властивості.

За допомогою рефлексії можна отримувати доступ до імен класів, методів, полів, конструкторів та інших елементів програми під час її виконання. Це дає можливість створювати програми, які можуть адаптуватися до змінних умов та динамічної середовища виконання.

У випадку розробки гри з використанням Zenject, рефлексія може використовуватися для автоматичного вирішення залежностей між об'єктами. Наприклад, замість того, щоб явно вказувати, які об'єкти повинні бути внесені у контейнер залежностей, можна використовувати рефлексію для автоматичного сканування всіх класів та їх атрибутів, які мають залежності, та автоматично додавати їх у контейнер залежностей.

для динамічної зміни поведінки гри на основі змінних параметрів, які можуть змінюватися в ході гри також використовують рефлексію. Можна змінювати рівень складності гри в залежності від результатів гравця або дозволяти гравцю налаштовувати різні параметри гри, такі як гучність звуку чи налаштування керування.

Однак, використання рефлексії може знизити продуктивність програми, тому варто ретельно планувати її використання та використовувати її тільки тоді, коли це дійсно необхідно.

У Zenject рефлексія використовується для автоматичного введення залежностей у об'єкти. Розробники можуть помітити поля в об'єктах, які потребують введення залежностей, використовуючи атрибути. Далі, Zenject використовує рефлексію для знаходження об'єктів, які потрібно ввести, та автоматично встановлює їх у відповідні поля.

Використання рефлексії фреймворку дозволяє зменшити кількість коду, необхідного для введення залежностей, та забезпечує більш гнучку та складну архітектуру. Однак, важливо пам'ятати, що використання рефлексії може призвести до зниження продуктивності програми, тому потрібно використовувати її розумно та з обережністю.

4.4 Проблеми створення гри з обмеженим бюджетом

Однією з основних проблем при створенні гри з обмеженим бюджетом є висока вартість розробки графіки та анімації. Зокрема, створення 3D-моделей, текстур та анімацій може вимагати значних витрат на обладнання та програмне забезпечення. Крім того, найнятий персонал може вимагати великої оплати.

Створення гри з обмеженим бюджетом може мати свої проблеми. Один з головних недоліків - обмеженість фінансів, що може призвести до зниження якості гри. Крім того, низький бюджет може обмежити можливості залучення талановитих розробників та художників, що може вплинути на кінцевий продукт. Також, при обмеженому бюджеті можуть виникнути труднощі з рекламою та маркетингом, що може погіршити продажі гри та призвести до збитків.

Іншою проблемою є брак ресурсів для маркетингу та реклами гри. Без належного просування, гра може не знайти свою аудиторію, що призведе до невеликих продажів та збитків.

Важливо враховувати, що створення гри з обмеженим бюджетом може забрати більше часу та зусиль, ніж при розробці гри з більшим бюджетом. Це може призвести до затримок в запуску гри та втрати потенційних прибутків.

Необхідність бути творчим та знаходити інноваційні рішення для створення гри з обмеженими ресурсами – це ще один недолік обмеженого бюджету. Це може бути викликом для розробників, які можуть бути обмежені в своїй творчості та ідеях.

Незважаючи на ці проблеми, створення гри з обмеженим бюджетом можливе. Для цього необхідно зосередитися на основних аспектах гри та знайти творчі рішення для зменшення витрат. Наприклад, можна сконцентруватися на розробці геймплею та ігрової механіки, які можуть бути менш складними та вимагати меншої кількості ресурсів для розробки. Також можна розглянути можливість використання безкоштовного або відкритого програмного забезпечення для розробки графіки та анімації, а також спеціалізованих частин геймплею.

Наступним кроком є складання детального плану проекту з урахуванням бюджетних обмежень. Важливо ретельно розглянути всі етапи створення гри та

розписати витрати на кожен з них. Можна відстежувати витрати у документації проекту або за допомогою програмного забезпечення для управління проектами.

Інший засіб зниження витрат - використання безкоштовних або недорогих програмних засобів. Наприклад, у розробці графічних елементів можна використовувати безкоштовні програми, такі як GIMP або Krita, або придбати дешеві ресурси в інтернеті. Також можна скористатися безкоштовними бібліотеками звуків та музики, щоб створити аудіо ефекти та музику для гри.

Незважаючи на обмежений бюджет, потрібно забезпечити якість продукту, тому слід зосередитися на створенні простих, але якісних ігрових механік та графічних елементів, замість складних та витратних рішень. Можна розглянути можливість повернення фрілансерів або інших фахівців для роботи над окремими аспектами проекту.

Потрібно взяти до уваги ризики та несподіванки, що можуть виникнути під час розробки гри. Тому використання стратегії зниження ризиків – тестування та перевірка на різних етапах розробки, дозволить уникнути непередбачуваних обставин.

4.5 Двовимірні ігри

Створення гри залежить від вимог та цілей розробника. Якщо головна мета - створити гру зі спрощеною графікою, то двовимірна гра може бути кращим варіантом, оскільки вона вимагає менше ресурсів та знань у порівнянні з тривимірною графікою. Також, двовимірна гра може мати свій стиль та атмосферу, які важко досягти з тривимірною графікою.

Тривимірна графіка може дозволити більш реалістично зображати об'єкти та створювати більш іммерсивний геймплей. Крім того, тривимірна графіка дозволяє користувачам досліджувати об'єкти з різних кутів, що збільшує відчуття присутності та реалізму.

Для двовимірної гри не потрібно використовувати так багато ресурсів, що може знизити вимоги до комп'ютерної системи, на якій запускається гра. З іншого

боку, для тривимірних ігор потрібно значно більше ресурсів, що може стати проблемою для менш потужних комп'ютерів або для тих, у кого обмежений доступ до ресурсів.

Створення двовимірної гри може бути менш складним процесом, оскільки не потрібно працювати з об'ємними об'єктами та ефектами освітлення. Це означає, що розробка може бути більш швидкою та менш трудомісткою, що зменшує витрати на розробку гри.

Варто зазначити, що кінцевий вибір між двовимірною та тривимірною грою повинен залежати від типу гри та її мети, а також від вподобань та очікувань цільової аудиторії. Для деяких жанрів гри, таких як імітаційні ігри, тривимірна графіка може бути набагато більш ефективною.

Гра у двох вимірах може дозволити зосередитися на інших аспектах гри, таких як історія, геймплей, музика та звукові ефекти, що може зробити гру більш привабливою для гравців.

Тривимірна гра може мати свої переваги, створення двовимірної гри також має свої переваги, такі як менші вимоги до ресурсів, менша складність розробки та можливість зосередитися на інших аспектах гри.

Висновки до четвертого розділу

У даному розділі проаналізована взаємодія з користувачем, що включає в себе всі аспекти, пов'язані з комунікацією між гравцем і грою. Забезпечення зручного та інтуїтивного інтерфейсу, якісного керування та реакції на дії гравця є важливими елементами, які забезпечують позитивний геймплей та задоволення від гри.

Розглянута тема розробки архітектури, яка охоплює планування структури гри, включаючи рівні, управління ресурсами, алгоритми, мережеву взаємодію та інші аспекти. Добре продумана архітектура допомагає створити гру з ефективним використанням ресурсів, швидким завантаженням та стабільною роботою.

Тема створення гри з обмеженим бюджетом відображає реальні виклики, з

якими стикаються розробники ігор з недостатнім бюджетом. Вони повинні ефективно управляти ресурсами, знаходити рішення замість дорогих альтернатив, шукати можливості для креативного використання обмежених засобів.

Наведено переваги розробки двовимірної гри, що фокусується на розробці ігор у двовимірному просторі. Це можуть бути класичні аркадні ігри, платформери, головоломки та інші типи. Важливо враховувати особливості таких ігор, такі як візуальний стиль, фізика об'єктів, зручне керування тощо.

Розглянуті теми показують, що розробка гри - це складний і багатогранний процес, який вимагає уваги до деталей, креативності, ефективного управління ресурсами та взаємодії з користувачем. Розуміння цих аспектів допоможе розробникам створити якісні та захоплюючі ігрові продукти.

5 РЕАЛІЗАЦІЯ ГРИ

5.1 Реалізація уповільнення часу у Unity

TimeScale є вбудованим параметром в Unity, який контролює швидкість відтворення часу в грі. Цей параметр можна використовувати для створення ефекту зміни часу в грі.

Параметр приймає значення від 0 до 1, де 1 відповідає нормальній швидкості відтворення, а значення менше 1 зменшує швидкість відтворення часу, створюючи ефект уповільнення. Вам потрібно налаштувати TimeScale на значення, яке найкраще відповідає бажаній плинності часу.

Є можливість змінювати значення TimeScale під час геймплею, щоб активувати або деактивувати ефект зміни часу. Наприклад, ви можете уповільнити час, коли гравець активує певну навичку або знаходиться у важкому бою. Це можна зробити за допомогою скриптів, які змінюють значення TimeScale залежно від умов у грі.

Застосування візуальних ефектів, разом зі зміною TimeScale, таких як прискорені або уповільнені анімації, плавні переходи між кадрами, ефекти часток тощо. Ці ефекти допомагатимуть підкреслити уповільнення часу і зробити його більш візуально привабливим для гравців.

Додавання звукових ефектів, таких як звук зміни часу або ефекти ехо, може підсилити враження від ефекту часу. Це додає слуховий елемент до геймплею.

Використовуючи TimeScale можливо створити режим зміни часу для певних важливих подій у грі. Наприклад, при досягненні гравцем великого досягнення або важливого моменту, ви можете уповільнити час, щоб гравець мав більше часу для реакції або насолодитися цим моментом. Це додає драматичності і напруження до геймплею.

TimeScale можна комбінувати з іншими ефектами, такими як зміна камери, розмивання екрану, використання шейдерів і т.д., для створення ще більш складних і захоплюючих ефектів уповільнення часу. Це дозволяє вам створити унікальний

інтерактивний досвід для гравців.

Слід враховувати, що швидкість руху ігрових об'єктів також буде змінюватися. Вам може знадобитися збільшити часовий коефіцієнт для руху об'єктів, щоб забезпечити їх плавний рух у уповільненому часі.

5.2 Система часток

Система часток (particle system) грає важливу роль у акцентуванні уваги на уповільненні часу (slow motion) у грі. Завдяки системі часток можна створити вражаючі візуальні ефекти, які допомагають підкреслити зміну темпу геймплею і привернути увагу гравців до важливих моментів.

При зміні плинності часу можуть використовуватися частки, що показують розсіювання або розширення часу. Це можуть бути маленькі частки, що летять повільніше, розпливаються або міняють свою форму, створюючи ефект уповільнення.

При певних моментах у грі, коли відбувається уповільнення часу, можна використовувати систему часток для створення вражаючих ефектів вибуху. Це можуть бути яскраві іскри, частки пилу або енергії, які вибухають у повітрі, надаючи відчуття напруження та динамічності.

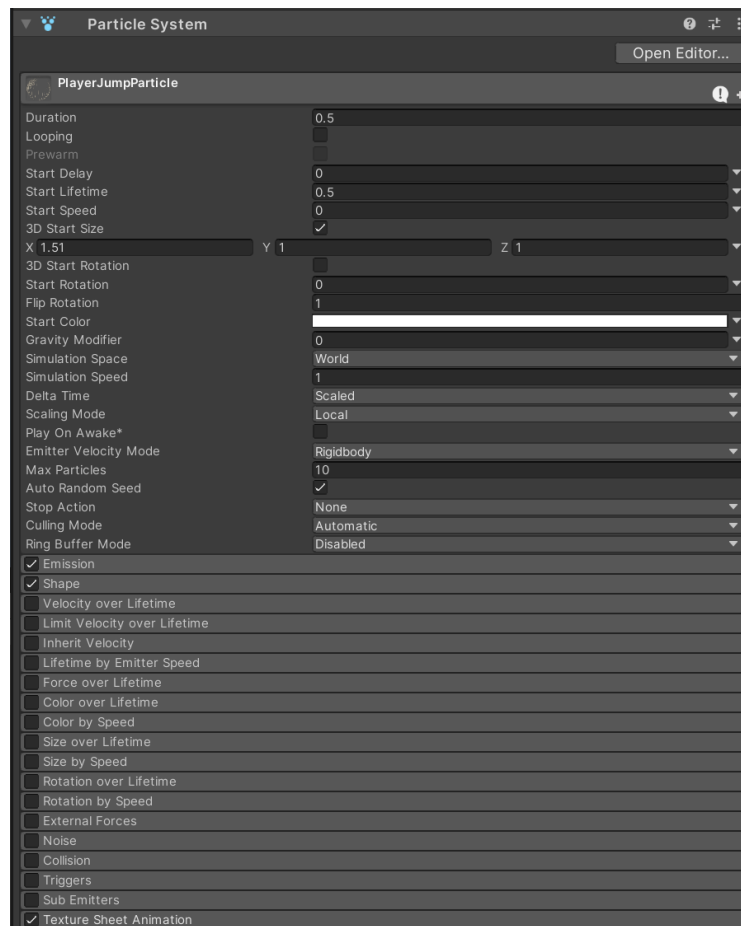


Рисунок 5.1 – Знімок панелі налаштування часток в Unity

У спотвореному часі створюються частки, що залишають сліди за рухом об'єктів або персонажів. Це може бути ефект тихих часток, що плавно пересуваються уздовж шляху руху, створюючи відчуття плинності і красивого малюнку.

Уповільнення часу підкреслено за допомогою світлових акцентів, таких як блискітки або флеші. Ці ефекти можуть нагадувати про вплив миготливого світла у реальному житті, що допомагає підкреслити відчуття зупинення часу і створює ефект вражаючої динаміки.

Система часток використовується для візуалізації одиниць часу у грі. створити частки, що летять у формі годинникових стрілок або пісочних годинників, що символізують зміну часу і нагадують гравцям про його наявність.

Система часток у поєднанні зі зміною TimeScale у Unity відкриває безліч можливостей для створення ефектів замедлення часу в грі. Це дозволяє

розробникам акцентувати увагу на важливих моментах, підсилити емоційну впливовість і створити захоплюючий досвід геймплею для гравців.

5.3 Фізика об'єктів

У Unity фізичні властивості об'єктів визначаються за допомогою компонента Rigidbody. Rigidbody дозволяє об'єктам у грі взаємодіяти з фізичною системою та відтворювати реалістичні фізичні ефекти. Основні фізичні властивості об'єктів у Unity включають:

- **Маса (Mass):** Маса об'єкта визначає його вагу та впливає на його поведінку під дією сили тяжіння та інших фізичних сил. Об'єкти з більшою масою будуть важче рухатися та вимагатимуть більше зусиль для їх переміщення.
- **Обмеження руху (Constraints):** Об'єкти можуть мати обмеження на свій рух, наприклад, фіксований рух по осі X або обмеження обертання. Це дозволяє контролювати, як об'єкт поводить себе під дією фізичних сил.
- **Тривимірна позиція та обертання (Position and Rotation):** Об'єкти з Rigidbody можуть займати конкретну позицію в просторі та мати певну орієнтацію. Вони можуть переміщатися, обертатися та взаємодіяти з іншими об'єктами у грі відповідно до фізичних правил.
- **Поведінка при зіткненні (Collision Behavior):** Об'єкти з Rigidbody можуть взаємодіяти між собою та з іншими колайдерами у грі. Вони можуть відштовхуватися, взаємодіяти з гравітацією, викликати події при зіткненні та інше.
- **Фізичні матеріали (Physics Materials):** Unity також надає можливість встановлювати фізичні матеріали для об'єктів, що впливають на їх поведінку при зіткненні. Матеріали можуть мати властивості, такі як коефіцієнт тертя тощо.

Rigidbody у Unity є важливим компонентом для відтворення реалістичної фізики об'єктів у грі. Він дозволяє об'єктам взаємодіяти з фізичною системою та реагувати на сили, такі як гравітація, стикування та зіткнення.

Один з основних аспектів Rigidbody - це маса об'єкта. Маса визначає його вагу та впливає на його поведінку під дією фізичних сил. Об'єкти з більшою масою

будуть важче рухатися та вимагатимуть більше зусиль для їх переміщення.

Rigidbody також надає можливість встановлювати обмеження руху об'єктів. Це означає, що можна контролювати, як об'єкт поводить себе під дією фізичних сил, наприклад, обмежувати його рух по певній осі або обертання.

Об'єкти з Rigidbody має визначену тривимірну позицію та обертання. Вони можуть переміщатися, обертатися та взаємодіяти з іншими об'єктами у грі відповідно до фізичних правил.

Взаємодія з іншими об'єктами є одним з важливих аспектів. Це дозволяє об'єктам зіткнутися, взаємодіяти з колайдерами. Це дозволяє створювати реалістичні сцени зі зіткненнями та фізичною взаємодією між об'єктами.

Окрім базових налаштувань, Rigidbody надає можливість застосовувати фізичні сили до об'єктів. Це дозволяє змінювати їх рух, надавати прискорення або зупиняти їх. Є можливість встановити коефіцієнти тертя, пружності та інші фізичні властивості, що впливають на поведінку об'єктів при зіткненні.

Компонент підтримує систему гравітації, яка автоматично додає силу тяжіння до об'єктів. Це забезпечує реалістичне падіння та сприяє натуральному руху об'єктів у грі.

Управління об'єктами з Rigidbody можна здійснювати за допомогою скриптів та фізичних методів, таких як AddForce для застосування сил, MovePosition для переміщення об'єкта або AddTorque для надання обертального руху. Це дозволяє програмно контролювати фізичну поведінку об'єктів у ваших ігрових сценах.

Використання компонента Rigidbody дозволяє створювати більш реалістичні та іммерсивні ігрові середовища, де об'єкти поведуться відповідно до фізичних законів та взаємодіють один з одним з урахуванням сил та зіткнень. Це важливий аспект при розробці багатьох ігрових жанрів, таких як симуляції, рейсинг або фізичні головоломки.

5.4 Типи колайдерів

У Unity існують різні типи колайдерів, які використовуються для визначення

форми та області взаємодії об'єктів у грі. Основні види колайдерів включають:

- **Колайдери прямокутника (Box Collider):** представляють прямокутну область взаємодії і є найпростішими у використанні. Вони часто використовуються для статичних об'єктів, таких як стіни або підлога.
- **Колайдери кулі (Sphere Collider):** моделюють область у формі кулі і використовуються для об'єктів, які мають сферичну форму, наприклад, м'ячі або голови персонажів.
- **Колайдери капсули (Capsule Collider):** подібні до колайдерів кулі, але мають циліндричну форму з плоскими кінцями. Їх використовують для моделювання об'єктів з формою трубки або тіла людини.
- **Колайдери сітки (Mesh Collider):** використовують точну форму мешу (трикутні полігональні сітки) об'єкта для визначення області взаємодії. Це дає більш точне відображення форми об'єкта, але може бути більш вимогливим до обчислювальних ресурсів.

Швидкість обробки колайдерів залежить від кількості колайдерів у сцені, складності їх форми та розміру об'єктів. Колайдери прямокутника та кулі мають найбільш ефективну обробку, оскільки вони прості у вирахуванні. Колайдери сітки, які використовують точний меш, можуть бути більш вимогливими до обчислювальних ресурсів та швидкості обробки.

Для оптимізації швидкості обробки колайдерів важливо використовувати оптимальну кількість та тип колайдерів для кожного об'єкта, уникати зайвого перекриття колайдерів та використовувати простіші колайдери, де це можливо. Варто розглядати використання методів оптимізації, таких як ієрархічне поділ колайдерів (Bounding Volume Hierarchies) або використання колайдерів на основі фізичних шейпів, які дозволяють більш швидку обробку колізій.

Враховуючи ці фактори та оптимізаційні підходи, можна досягти ефективної обробки колізій та забезпечити плавну та реалістичну фізичну взаємодію об'єктів у вашій грі.

Для гравця було обрано колайдер капсули, для ворогів – кулі. Це ті форми які найліпше підходять до візуальної форми цих компонентів.

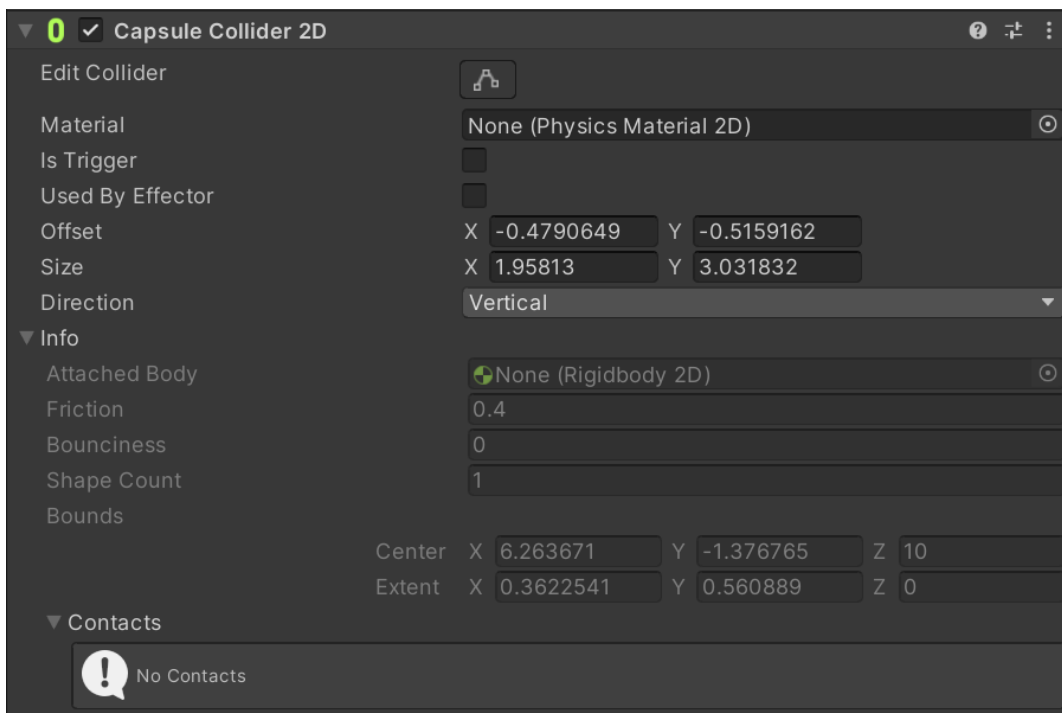


Рисунок 5.2 – Знімок панелі налаштування колайдери в Unity

5.5 Ігрові анімації

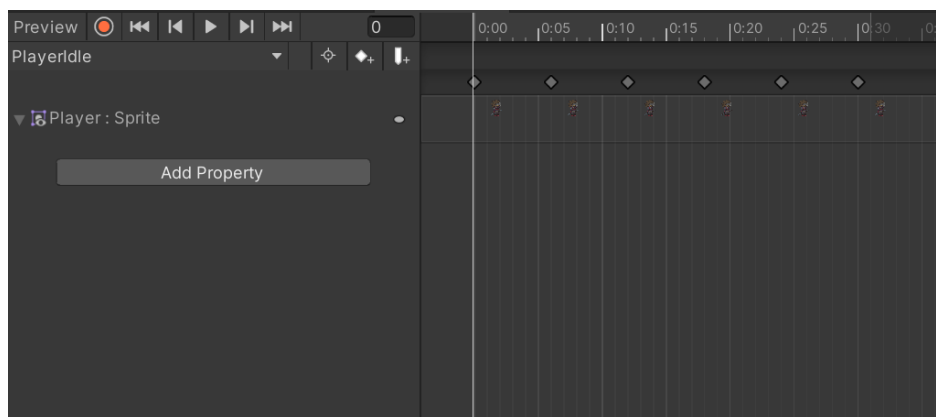


Рисунок 5.3 – Знімок панелі створення та налаштування анімації в Unity

Анімації відіграють важливу роль у геймплеї, допомагаючи створювати живі та реалістичні ігрові світи. Вони дозволяють оживити персонажів, об'єкти та оточення, надаючи їм рух та виразність. Ось деякі ключові аспекти та ролі анімацій у геймплеї:

Створення реалістичних рухів для персонажів у грі досягається шляхом

створення анімацій. Вони включають ходьбу, біг, стрибки, атаки, блокування та багато інших дій. Це дозволяє гравцеві взаємодіяти з персонажем та керувати його поведінкою. Анімації дозволяють персонажам взаємодіяти з оточуючим світом. Наприклад, вони можуть відкривати двері, піднімати предмети, спілкуватися з іншими персонажами, вхід у транспорт та багато іншого. Це покращує іммерсію гравця та реалізм гри.

У бойових сценах гри анімації включають атаки, блокування, ухилення та різноманітні рухи бойового мистецтва. Використання відповідних анімацій допомагає зробити бойову систему плавною, динамічною та виразною.

Для створення ефектів та спеціальних дій у грі використано анімації. Для ефекту стрибка, завдання шкоди ворогу. Вони роблять ігрові моменти захоплюючими та незабутніми.

Анімації допомагають передати емоції та виразність персонажів. Вони можуть включати рухи обличчя, жестів та постави, які відображають почуття персонажів, такі як радість, гнів, страх або сум. Це підсилює зв'язок гравця з персонажами та робить їх більш живими та цікавими.

У грі анімації здебільшого використані для візуального представлення стану ігрового персонажа, ворогів, оточення.



Рисунок 5.4 – Знімок спрайтової анімації

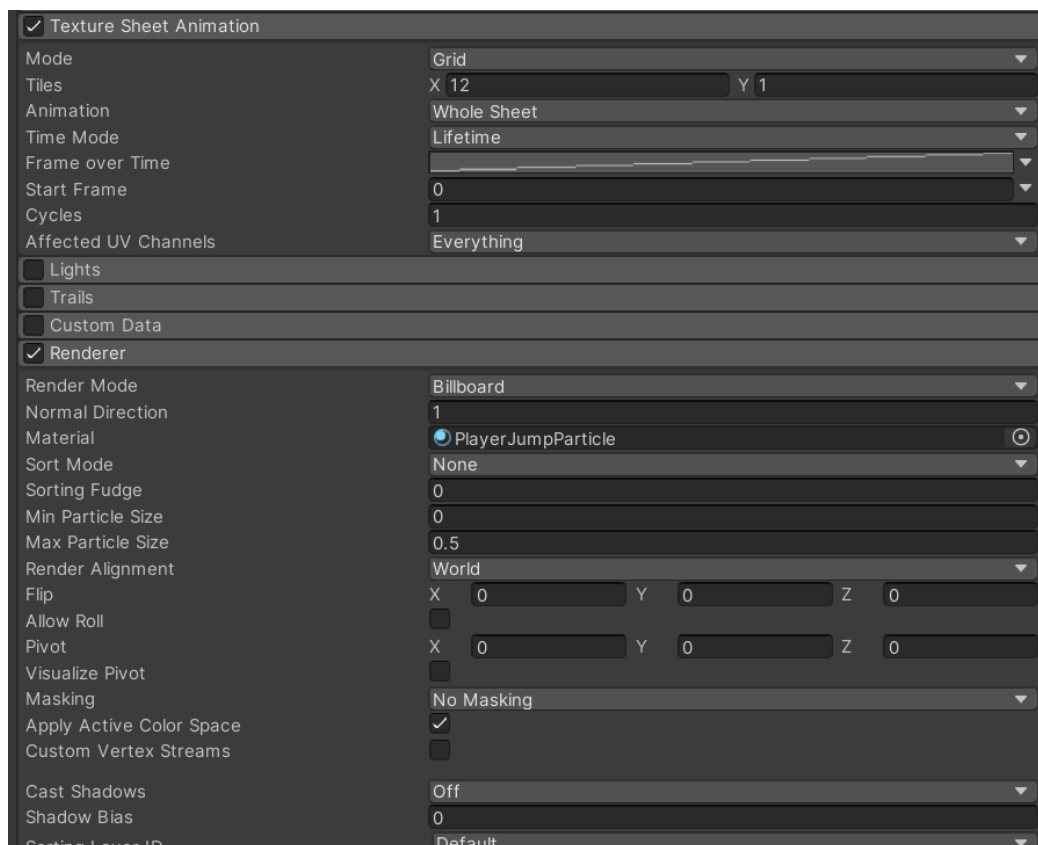


Рисунок 5.5 – Знімок панелі створення анімованих часток в Unity

За допомогою матеріалів та спрайтів з властивістю **Multiple**, можна створити системи часток які будуть програвати анімацію кожної частинки, кадри двовимірної анімації повинні знаходитись у єдиному файлі спрайту. Це було використано щоб створити такі ефекти як дим на місці стрибка гравця, ефект уповільнення часу, ефект вибуху ворогів.

Анімації додають геймплею глибину, реалізм та іммерсію. Вони допомагають розповісти історію, передати настрій та створити незабутні враження для гравців.

5.6 Оптимізація

Оптимізація проекту починається з середи ігрового рушія. Unity в собі має безліч влаштованих аналізаторів та методів аналізу продуктивності проекту. Основний компонент для аналізу – це Profiler.

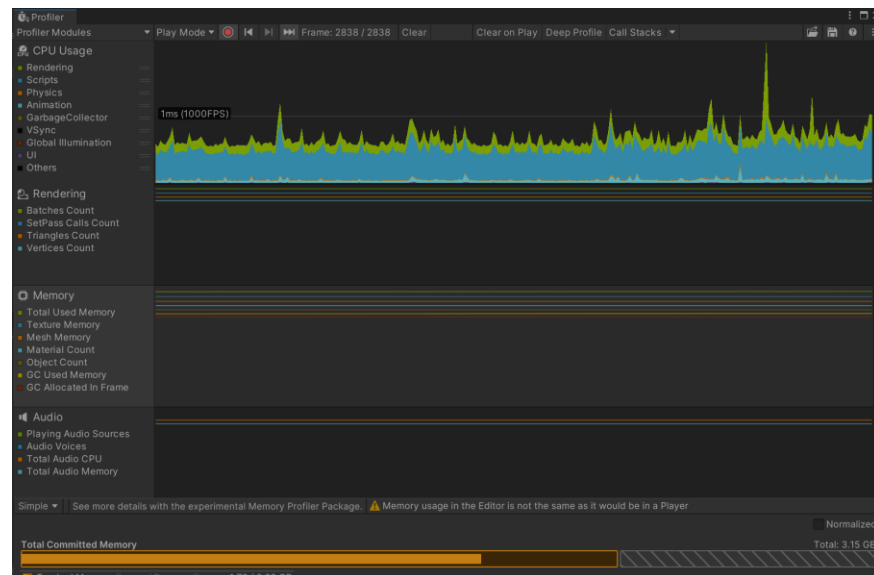


Рисунок 5.6 – Знімок панелі аналізу гри в Unity

Profiler є потужним інструментом, який надає детальну інформацію про використання ресурсів, час виконання коду, пам'ять та інші фактори, що впливають на продуктивність гри.

Профільювальник Unity надає наступні можливості:

- **Аналіз продуктивності:** Він дозволяє відстежувати час виконання окремих функцій, методів або частин коду, що допомагає ідентифікувати гарячі точки (hotspots) і зайві навантаження. Це дозволяє розуміти, які частини гри потребують оптимізації.
- **Графічний аналізатор:** Профільювальник Unity надає можливість відстежувати та аналізувати шейдери, рендеринг і використання GPU. Це дозволяє виявляти проблеми з відображенням, такі як зайвий розрахунок або надмірне використання графічних ресурсів.
- **Пам'ять та аллокатор:** Профільювальник дозволяє відстежувати використання пам'яті в грі, виявляти утечки пам'яті, аналізувати використання аллокатора. Це допомагає оптимізувати використання пам'яті та забезпечити її ефективне використання.
- **Статистика фреймів:** Профільювальник Unity надає статистику фреймів, таку як FPS (кадри в секунду), CPU використання та інші показники продуктивності. Це дозволяє відстежувати та аналізувати продуктивність гри під

час різних сцен та умов.

Профільювальник Unity є потужним інструментом для аналізу та оптимізації продуктивності гри. Використовуючи його, розробники можуть ідентифікувати проблеми ефективності, зайве навантаження та виконати оптимізацію для забезпечення плавного та швидкого досвіду гри.

Інший інструмент використаний для оптимізації - Фрейм-дебаггер (Frame Debugger). Це інструмент, який надає розробникам можливість аналізувати і візуалізувати процес рендерингу кадрів у грі. Він є частиною інтегрованого профільювальника в редакторі Unity і дозволяє детально розглядати кожен крок рендерингу, шейдери, проходи освітлення та інші елементи, що впливають на візуальний вигляд гри.

Фрейм-дебаггер дає змогу розробникам переглядати послідовність кроків рендерингу для кожного об'єкта на сцені. Це допомагає виявляти проблеми з порядком рендерингу, прозорістю, порядком прозорості, артефактами та іншими проблемами, що впливають на візуальний вигляд гри.

Надає інформацію про використання ресурсів, таких як текстур, буфери, матеріали та інші елементи, що впливають на продуктивність гри. Розробники можуть виявляти зайве використання ресурсів, такі як великі текстур або незрозумілі шейдери, які можуть призводити до зайвого навантаження на графічну карту або GPU.

Даний інструмент допомагає виявляти проблеми з продуктивністю, такі як зайво велика кількість рендерингу, затримки виконання шейдерів, проблеми з колізіями або фізикою. Це дозволяє розробникам виявити та усунути проблеми, що можуть призводити до низької швидкодії або несплавності гри.

Фрейм-дебаггер дозволяє експериментувати з різними налаштуваннями графічних параметрів, такими як роздільна здатність, якість тіней, рівень деталізації тощо, і відстежувати їх вплив на продуктивність гри. Це допомагає розробникам знайти баланс між візуальною якістю та продуктивністю, надає можливість аналізувати і дебажити шейдери, перевіряти їх правильність, виявляти помилки та відстежувати виконання шейдерів на різних об'єктах. Це дозволяє

розробникам вирішувати проблеми з візуалізацією та ефективністю шейдерів.

Це є потужний інструмент для візуального аналізу та оптимізації графіки в Unity, дозволяючи розробникам створювати оптимальні та візуально привабливі ігрові візуали.

Після реалізації всіх задумок, котрі прописані у дизайн-документі проекту, включаючи як програмну так і візуальну та звукову частини, за допомогою влаштованих у Unity інструментів потрібно створити збірку проекту на телефон. Перед цим необхідно перевірити чи для всіх плагінів та додаткових фреймворків, що були використані під час створення проекту, встановлена підходяща мінімальна та максимальна версія API Android

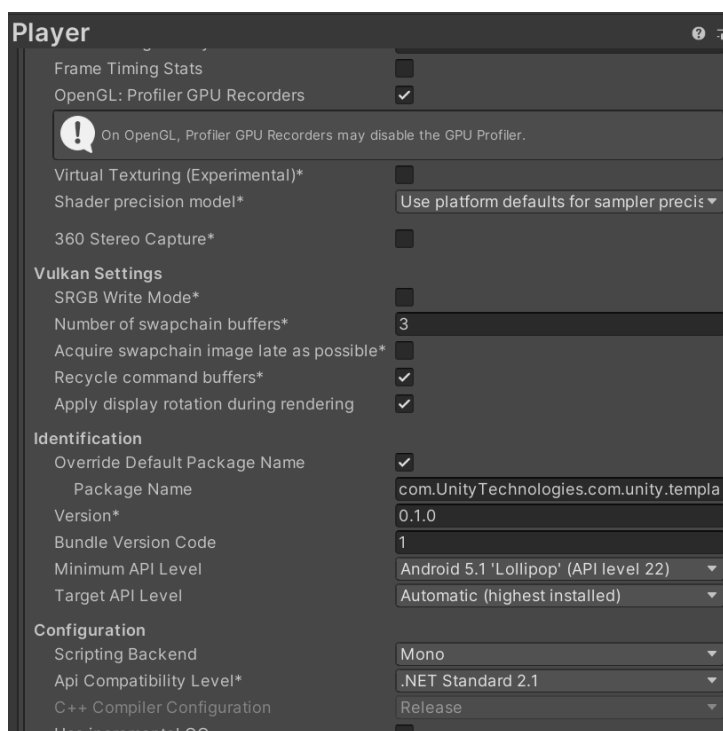


Рисунок 5.7 – Знімок панелі налаштування збірки в Unity

Є можливість обмежити кількість кадрів в секунду гри, що зробить розриви мінімальними, а роботу самої програми на вигляд більш постійною та надійною. Це обмеження діє на кшталт вертикальної синхронізації у комп'ютерних іграх.

Vsync (Vertical Synchronization) у іграх є технологією, яка синхронізує частоту оновлення екрану з кадровою частотою гри. Вона забезпечує гладкість відтворення графіки і запобігає неприємним явищам, таким як "розриви" (tearing)

та "статтекти" (stuttering). Коли Vsync увімкнений, гра буде відображатись на екрані з максимальною частотою оновлення, яку підтримує монітор. Це забезпечує гладкі переходи між кадрами та відсутність розривів у відображенні графіки. Коли гра працює з більшою швидкістю кадрів, ніж може відобразити монітор, VSYNC буде застосовуватись для обмеження кадрової частоти і підтримки синхронізації.

Після встановлення збірки, можна починати тестування гри в робочих умовах, використовуючи власноруч написані аналізатори влаштовані у гру, такі як показник середньої кількості кадрів за секунду та показники завантаженості ресурсів пристрою. Якщо кількості кадрів недостатньо, є потреба повернутись до проекту Unity та коду, провести повторний аналіз на рахунок оптимізаційних рішень. За допомогою перерахованих раніше інструментів виконати оптимізацію. У моєму випадку потрібно було значно зменшити кількість часток кожного окремого VFX ефекту, після цього середня кількість кадрів за секунду значно підросла і була достатньою для комфортного геймплею.

Висновки до п'ятого розділу

В цьому розділі було розглянуто основні інструменти рушія Unity що використовувалися під час створення гри, описані їх властивості.

Описано логіку відтворення основної механіки уповільнення часу за допомогою влаштованих у рушій методів. Проаналізовано як такий підхід до гри впливає на динаміку геймплею.

Проаналізовані інструменти для створення системи часток, та як за допомогою VFX ефектів можна більш ефективно та ефектно акцентувати увагу на основній механіці гри.

Створені анімації для оточення, для героя та ворогів. Описаний анімаційний підхід для створення VFX ефектів.

Проаналізовані та використані кращі практики для оптимізації гри, описані основні методи оптимізаційного аналізу проекту на Unity.

ВИСНОВКИ

У роботі проведено дослідження та розробка мобільної казуальної гри. В першому розділі була проаналізована актуальність такого виду ігор та визначено процес їх створення. Особлива увага приділена ролі зміни швидкості часу у грі та створенню дизайн-документу.

Другий розділ присвячений аналізу наявних ігор, їх геймплею, механіці, візуальним та звуковим складовим. Був визначений концепт та геймплей майбутньої гри.

У третьому розділі було обґрунтовано вибір технологій, зокрема мов програмування та ігрових рушіїв, які найкраще відповідають вимогам розробки мобільної казуальної гри.

Четвертий розділ присвячений проектуванню гри. Були розроблені сценарій, взаємодія з користувачем та архітектура гри. Досліджено проблеми, пов'язані з обмеженим бюджетом, а також зосереджено увагу на двовимірних іграх.

У п'ятому розділі описано реалізацію гри. Були розроблені методи уповільнення часу, система часток, фізика об'єктів, типи колайдерів, ігрові анімації та проведена оптимізація.

Результатом роботи є інтерактивна казуальна гра, з механікою зупинки часу що полегшує геймплей та зменшує поріг входу у гру, водночас з тим робить гру більш цікавою. Загальними висновками з роботи можна зазначити, що розробка мобільної казуальної гри є актуальним напрямом. Важливо враховувати роль зміни швидкості часу та створювати цікавий геймплей. Вибір технологій та правильне проектування гри є важливими етапами розробки. Реалізація гри вимагає уваги до деталей, таких як уповільнення часу, система часток, фізика об'єктів та оптимізація.

ПЕРЕЛІК ПОСИЛАНЬ

1. Top 10 game engines, 2022. [Електронний ресурс] – Ресурс доступу: <https://ulab.sumdu.edu.ua/uk/10-najkrashhih-igrovih-rushiiv>
2. Getting Started with Unity: Colliders and UnityScript, 2019. [Електронний ресурс] – Ресурс доступу: <https://coderlessons.com/articles/veb-razrobkaarticles/nachalo-raboty-s-unity-kollaidery-i-unityscript>
3. Mobile game development market in Ukraine, 2021. [Електронний ресурс] – Ресурс доступу: <https://it.comments.ua/ua/article/games/rinok-rozrobkimobilnih-igor-v-ukraini-vbivati-chas-uzhe-ne-tak-veselo-yak-ranishe690462.html>
4. Офіційний сайт гри “Katana Zero” [Електронний ресурс]: [Веб–сайт]. – електронні дані. – Режим доступу: <https://www.katanazero.com>
5. Endless runner [Електронний ресурс]: [Веб–сайт]. – електронні дані. – Режим доступу: <http://jerrymomoda.com/analysis-endless-runners/>
6. Game Engine [Електронний ресурс]: [Веб–сайт]. – електронні дані. – Режим доступу: https://en.wikipedia.org/wiki/Game_engine
7. Казуальні ігри [Електронний ресурс]: [Веб–сайт]. – електронні дані. – Режим доступу: <https://www.prostranstvo.media/uk/kazualni-igry-vdavana-prostota-na-mezhi-vysokogo-mystecztva/>
8. Офіційний сайт рушія Unity [Електронний ресурс]: [Веб–сайт]. – електронні дані. – Режим доступу: <https://docs.unity3d.com>
9. Офіційний сайт рушія Unreal Engine [Електронний ресурс]: [Веб–сайт]. – електронні дані. – Режим доступу: <https://www.unrealengine.com/>
10. Офіційний сайт фреймворку DOTWEEN [Електронний ресурс]: [Веб–сайт]. – електронні дані. – Режим доступу: <http://dotween.demigiant.com/>
11. Офіційний репозиторій фреймворку Zenject [Електронний ресурс]: [Веб–сайт]. – електронні дані. – Режим доступу: <https://github.com/modesttree/Zenject>

ДОДАТОК А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО -НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



«Розробка інтерактивної казуальної мобільної гри під систему Android з використанням C# .NET та Unity»

Виконав студент 5 курсу

групи ППЗ-51

Запорожченко Антон Михайлович

Керівник роботи

К.е.н, доцент кафедри ІПЗАверічев Ігор Миколайович

Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи : зменшення порогу входу у гру за допомогою використання механіки уповільнення часу

Об'єкт дослідження: ігровий процес казуальної гри на платформу Android з використанням рушія Unity

Предмет дослідження: інтерактивність та динаміка в казуальних мобільних іграх

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Проаналізувати ринок казуальних мобільних ігор.
2. Дослідити роль динамічного геймплею у залученні уваги гравця.
3. Створити дизайн -документ додатку, прописати основну механіку та концепт гри, визначити тематику .
4. Знайти та створити асети для гри згідно з тематикою та концептом
5. Створити програмну частину мобільної гри згідно з дизайн - документом
6. Провести аналіз можливої оптимізації частин додатку для підвищення середньої кількості кадрів за секунду, реалізувати оптимізацію.
7. Провести тестування додатку на пристроях різних поколінь з ОС Android.

3

АНАЛІЗ АНАЛОГІВ

Назва гри	NinJump	Katana Zero	The Messenger	Cyber Shadow	Infinite ninja
На мобільні платформи	+	-	-	+	+
Нескінченна гра	+	-	-	-	+
Механіка уповільнення часу	-	+	-	-	+
Переваги	Низький поріг входу, можна грати будь-де і будь-коли	Висока динаміка, різноманітні рівні, цікавий сюжет	Велика кількість ефектів, присуття механіка покращення ігрового спорядження	Цікавий сюжет, гра з логічним завершенням, низький поріг входу	Низький поріг входу, висока динаміка, нескінченний геймплей
Недоліки	Однотипна гра, швидко набридає, бракує різноманітності ігрових елементів	Гра орієнтована на досвідчених гравців, відсутня на мобільних пристроях	Відсутня на мобільних пристроях, занадто коротка	Мала кількість рівнів та ігрових елементів	Відсутній логічний кінець

4

КОНЦЕПТ ГРИ

1. Гравець дотиком по екрану обирає траєкторію руху свого персонажа, щоб переміститись до іншого місця, або завдати удару по ворогу
2. Під час руху та прицілюванні гра уповільнюється
3. Ігровий рівень постійно рухається, якщо гравець не встигає просунутись далі по рівню, він програє
4. Після знищення ворога гравець отримує рекордні бали
5. Швидкість генерації рівня і ворогів зростає з часом, що підвищує складність ігрового процесу
6. Якщо ворог завдав гравцю пошкодження, або гравець врізався у перешкоду - гравець програє, його прогрес втрачається і гра починається з початку
7. Чим далі гравець просунувся у грі і чим більше ворогів він знищив, тим більший рекорд він встановить
8. Гравець може змагатися з іншими гравцями у встановленні рекорду, або бити власні рекорди

5

ВИМОГИ ДО ІГРОВОГО КОНТЕНТУ

1. Ігрові механіки
 - Уповільнення часу під час підготовки атаки для концентрації уваги гравця на прорахунку найліпшої атаки
 - Уповільнення часу після знищення ворога, щоб отримати можливість вибрати наступну ціль або знайти шлях для уникнення можливих перешкод
 - Механіка підрахунку вдалих ігрових комбінацій та їх перелік до загального ігрового рахунку
2. Ігрове середовище
 - Процедурна генерація частин рівня для створення ефекту нескінченності
 - Генерація ворогів на згенерованому рівні
 - Генерація перешкод які гравець повинен уникати

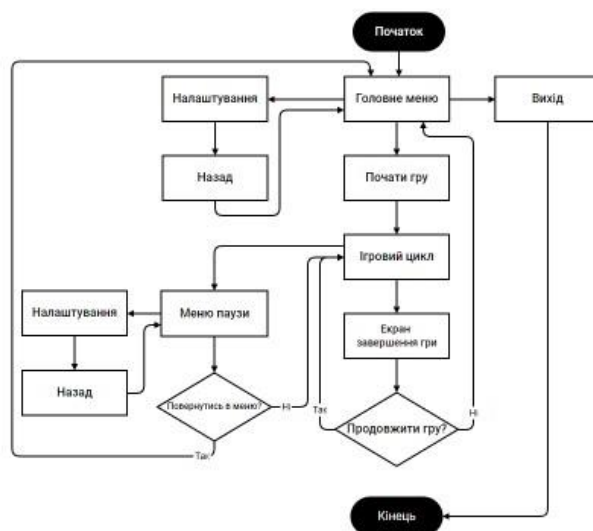
6

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



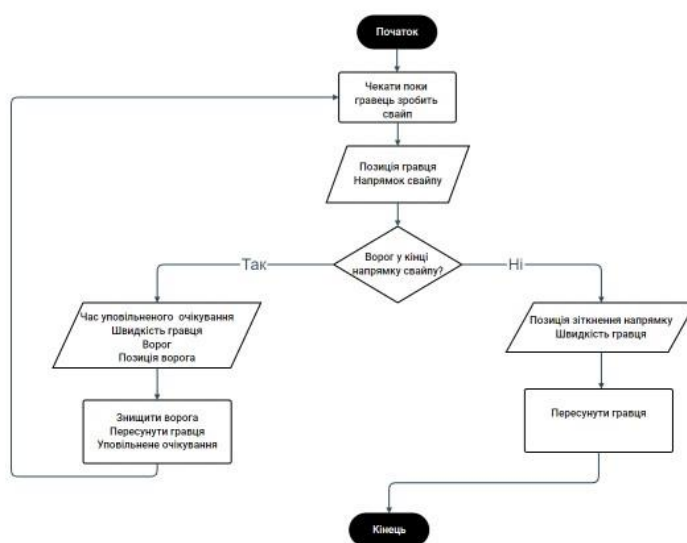
7

БЛОК-СХЕМА ІГРОВОГО ПРОЦЕСУ



8

БЛОК-СХЕМА АЛГОРИТМУ ОСНОВНОЇ МЕХАНІКИ ГРИ

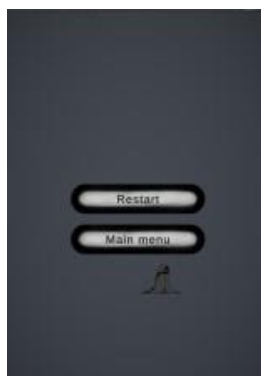


9

ЕКРАННІ ФОРМИ



Головне меню



Вікно кінця гри



Вікно налаштування



Вікно гри

10

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Аверічев І.М., Запорожченко А.М. Мобільна гра на Unity та інтернет речей для навчання. // Всеукраїнська науково-технічна конференція «Сучасний стан та перспективи розвитку IoT» – Київ: ДУТ, 2023.
2. Аверічев І.М., Запорожченко А.М. Використання Big Data і аналіз даних у грі на Unity // Всеукраїнська науково-технічна конференція «Сучасний стан та перспективи розвитку IoT» – Київ: ДУТ, 2023.

11

ВИСНОВКИ

1. Проаналізовано ринок мобільних казуальних ігрових додатків, виявлено кращі та найвдаліші ідеї, концепти.
2. Досліджено роль динамічного геймплею в інтерактивних іграх, вирішено використовувати уповільнення часу як основну концепцію.
3. Сформовано дизайн-документ гри, її тематика та стиль.
4. Створено програмну та візуальну частину гри згідно з дизайн-документом.
5. Виявлено та реалізовано оптимізацію додатку для підвищення середньої кількості кадрів за секунду.
6. Гру протестовано на пристроях різних поколінь.

12

ДЯКУЮ ЗА УВАГУ!