

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 РОЛЬОВОЇ ГРИ ДЛЯ ANDROID НА ДВИГУНІ
UNITY»**

Виконав: студент 4 курсу, групи ПД-44

спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Роспутній Ю.О.

(прізвище та ініціали)

Керівник

Трінтіна Н.А.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Нормоконтроль

(прізвище та ініціали)

Київ – 2021

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

« ____ » _____ 2021 року

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ
РОСПУТНЬОМУ ЮРІЮ ОЛЕКСАНДРОВИЧУ

1. Тема роботи: «Розробка рольової гри для Android на двигуні Unity»

Керівник роботи Трінтіна Наталія Альбертівна, доцент, кандидат технічних наук затверджені наказом вищого навчального закладу від — «12» березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.1. Середовище розробки Unity 2019.3.14f1

3.2. Науково-технічна література, пов'язана з розробкою на Unity

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

4.1. Аналіз предметної області Android-додатків

4.2. Моделювання процесу створення Android-додатку

- 4.3. Розробка ігрових об'єктів
- 4.4. Розробка звукового супроводженн
- 5. Перелік графічного матеріалу
 - 5.1. Популярність ігор
 - 5.2. Переваги та недоліки подібних ігор
 - 5.3. Програмні засоби реалізації
 - 5.4. Огляд функціоналу гри
 - 5.5. Апробація результатів досліджень
- 6. Дата видачі завдання 19.04.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.21 – 22.04.21	
2	Аналіз існуючих прототипів	22.04.21 – 23.04.21	
3	Дослідження програмних засобів	23.04.21 – 01.05.21	
4	Моделювання об'єкту проектування	01.05.21 – 05.05.21	
5	Розробка функціоналу гри	05.05.21 – 07.05.21	
6	Вступ, висновки, реферат	07.05.21 – 08.05.21	
7	Розробка презентації застосунку	08.05.21 – 24.05.21	
8	Попередній захист роботи	25.05.21	

Студент

Керівник роботи

РЕФЕРАТ

Текстова частина бакалаврської роботи 32с., 19 рис., джерел

Ключеві слова: Unity, C#, SAI, Visual Studio Community , рольова гра, скрипт

Об'єкт дослідження – поліпшення якості та функціоналу ігор в Play Market.

Предмет дослідження – мобільна рольова гра, для Android.

Мета роботи – розробка рольової гри для Android на двигуні Unity та мові C#.

Методи дослідження – методи теорії інформації, методи структурного аналізу і проектування, методи розробки програмного забезпечення, методи тестування, валідації та верифікації програмного забезпечення.

Наукова новизна даної роботи полягає в наступному:

1. Розроблено гру, яка майже не має аналогів на ринку.
2. Встановлено, що двигун Unity є перспективним та одним з кращих двигунів для розробки ігор для різноманітних платформ.
3. На основі результатів аналізу ринку мобільних ігор, був розроблений додаток, який є достатньо унікальним для залучення широкої аудиторії.

В роботі виконано аналіз існуючих ігор аналогів. Встановлено переваги та недоліки існуючих ігор. В результаті аналізу було визначено основні потреби користувачів. Проаналізовано можливості середовища розробки Unity.

Галузь використання – гру може завантажити кожен користувач смартфона на операційній системі Android.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕНЬ	10
1.1 Ігрові платформи.....	10
1.2 Android ігри	11
1.2.1 Multiplayer Online Battle Arena(MOBA)	11
1.2.2 Massively Multiplayer Online Role Play Game(MMORPG):	12
1.2.3 Battle Royale	12
1.2.4 Study games	12
1.3 Огляд найближчих ігор аналогів.....	13
1.3.1 Ігри аналоги.....	13
1.3.2 Переваги та недоліки.....	15
2 ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ.....	17
2.1 Обрані технології розробки	17
2.1.1 Мова програмування C#	17
2.2 Середовище розробки.....	17
2.2.1 Unity	18
2.2.2 Visual Studio 2019 Community	19
3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	21
3.1 Використані програмні інструменти	21
3.1.1 Програма SAI	21
3.1.2 Microsoft OneNote	21
3.2 Основні відомості про гру	22
3.3 Діаграма нотації IDEF0	23
3.4 Діаграма використання додатку	26
3.5 Процес розробки	28
3.5.1 Розробка файлів локалізації.....	28
3.5.2 Розробка спрайтів та текстур.....	28
3.5.3 Розробка звукового супроводження	28
3.6 Розробка ігрових сцен	29
3.6.1 Перший запуск	29
3.6.2 Основна сцена гри	31
3.7 Розробка ігрових механік.....	33
3.7.1 Розробка вікна діалогу та квестів.....	33

4	РОЗРОБКА СКРИПТІВ.....	34
4.1	Скрипти локалізації.....	34
4.1.1	LocalizationControl.cs.....	34
4.1.2	LocalizedTextHolder.cs та LocalizedTextLoader.cs.....	34
4.2	Скрипти діалогу.....	34
4.2.1	DialogueDisplay та DialogueAdvanceHandler.....	34
4.2.2	Character.cs та Conversation.cs.....	34
4.2.3	ConversationLocale.cs.....	35
4.2.4	Скрипти гравця.....	35
4.2.5	Player.cs та PlayerObject.cs.....	35
4.2.6	SkillHolder.cs та Skill.cs.....	35
4.2.7	Скрипти бойових систем.....	35
4.3	Скрипти битви.....	35
4.3.1	Battlehandler.cs.....	35
4.3.2	Enemy.cs та EnemyObject.cs.....	36
4.4	Готовий продукт.....	36
	ВИСНОВОК.....	37
	СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	38

ВСТУП

Мобільні телефони давно вже давно увійшли в рутину кожної людини і зараз важко знайти когось у кого не буде при собі хоч якогось телефона. При цьому, більшість вже давно користується смартфонами, які можуть своїм функціоналом можуть суперничати зі стаціонарними комп'ютерами. На смартфоні можна робити все: слухати музику, робити знімки, спілкуватися з відео та ще багато речей, всього не перерахуєш.

Поступово з вдосконаленням технологій на смартфонах можливо навіть займатися професійною діяльністю, наприклад малювати чи, навіть, програмувати. І звісно, не обійшлося і без розваг. Кожен у кого є телефон хоч раз запускав якусь гру. Мобільні ігри не тільки допомагають витратити час, а і розслабитись після важкого дня чи зайнятись саморозвитком.

Актуальність даної роботи пояснюється високою популярністю додатків, а саме ігор серед користувачів смартфонів. Мобільні ігри дозволяють користувачу відпочити від реальності і поспілкуватися з друзями. Головною перевагою мого додатку є висока унікальність його відносно інших ігор в Play Market.

Метою випускної кваліфікаційної роботи є розробка рольової гри для Android на двигуні Unity.

Виходячи з встановленої мети, були поставлені наступні завдання:

- аналіз обраної предметної області;
- порівняння наявних аналогів додатків;
- вибір технологій і середовища розробки;
- розробка RPG гри на мові програмування C#.

1 АНАЛІЗ ОБЛАСТІ ДОСЛІДЖЕНЬ

1.1 Ігрові платформи

Ігри були популярні завжди і серед людей будь-якого віку, а все тому, що вони були ефективним способом відпочити від рутини. І першою ігровою платформою був сам навколишній світ, з розвитком цивілізації почали з'являтися настільні ігри.

Перші ж примітивні комп'ютерні ігри були розроблені в 1950-х і 1960-х роках. Вони працювали на таких платформах, як університетські мейнфрейми і комп'ютери EDSAC. Тобто з моменту виникнення мов програмування люди почали створювати різноманітні інтерфейси для гри з програмою.

З поступовим створенням і вдосконаленням операційних систем ігри також стали більш розвинуті в них з'явилась графіка та набагато більше можливостей. Так в наш час є декілька ігрових платформ, а саме: комп'ютери, консолі, мобільні пристрої. Хоча комп'ютери і залишаються найпопулярнішою платформою для ігор (по результатам опитування рисунок 1) але мобільні ігри займають 2 місце за популярністю, а це через те що в них можливо грати будь-де.

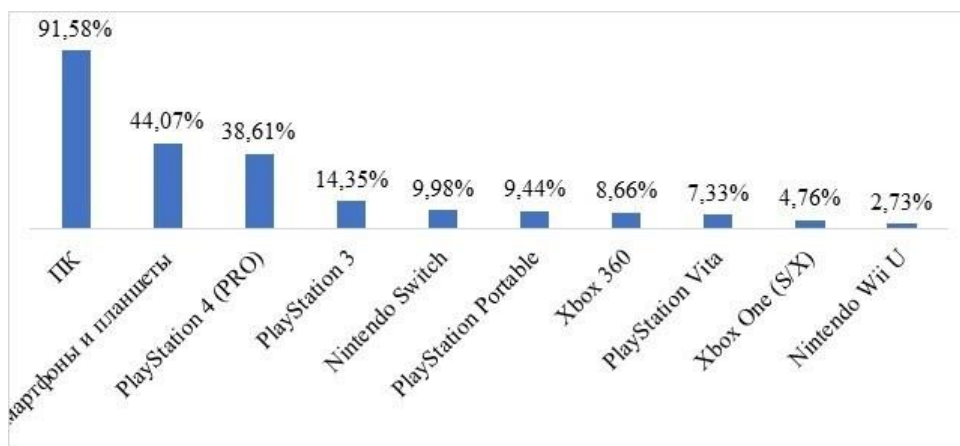


Рисунок 1 – Результати опитування

Особливої уваги заслуговує мобільна ОС Android.

Android — операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux. Підтримується альянсом Open Handset Alliance (ОНА). Хоча Android базується на ядрі Linux, він стоїть дещо осторонь Linux-спільноти та Linux-інфраструктури. Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення і застосування спираються на цю реалізацію Java. Деякі з переваг Android:

- 1) вільна інсталяція програм з сторонніх джерел;
- 2) вільний доступ до призначених для користувача файлів на зовнішніх носіях та у внутрішній пам'яті;
- 3) широкі можливості налаштування інтерфейсу;

Через те що Android є відкритою операційною системою існує багато безкоштовних середовищ розробки, найпопулярнішими є Android Studio, Unreal Engine та Unity Engine.

1.2 Android ігри

В індустрії мобільних ігор операційна система Android переважає над IOS. Хоча майже в кожному двигуні який підтримує розробку для Android підтримується розробка і для IOS але використання таких інструментів потребує додаткової оплати.

Додатки в Play Market в більшості є безкоштовними і кожен може завантажувати їх на свій смартфон якщо він відповідає мінімальним системним вимогам.

Проаналізувавши ринок мобільних додатків можна прийти до висновку, що найпопулярнішими є такі ігрові жанри:

1.2.1 Multiplayer Online Battle Arena(MOBA)

Суть ігор даного жанру полягає в знищенні бази ворога. Таку гру можна поділити на 2 фази: фаза «фарму», в якій гравець накопичує золото чи якісь ресурси перемагаючи неігрових персонажів та фаза битви, в якій гравець вже нападає зі своєю командою на команду ворога. Зазвичай в цьому жанрі битви проходять 5 на 5 і через те що існує велика кількість варіацій героїв а предметів які можна купити один такий матч в середньому триває 30 хвилин. Тому такий тип ігор підходить для тих у кого багато вільного часу.

1.2.2 Massively Multiplayer Online Role Play Game(MMORPG):

Даний жанр гри є рольовою грою з мультиплеєром. Зазвичай в іграх цього жанру є 3 або більше персонажів на вибір(в основному персонажа змінити неможливо), добре продумана сюжетна лінія та кооператив який дозволяє проходити гру в компанії друзів. Хоча спочатку може здатися, що для цього жанру не потрібно витратити так багато часу на матч, як у випадку з МОБА але насправді це далеко не так. Через переповненість контентом у ігор даних жанрів є багато щоденних квестів які потрібно виконати якщо ти не хочеш відстати від інших гравців.

1.2.3 Battle Royale

Ігровий жанр який набрав популярність не дуже давно і він ідеально підходить для тих хто хоче відчувати себе першим. Суть полягає в битві на виживання тобто гравець один або в команді з 1 і більше(до 3 інших гравців) висаджуються на віддалений острів де їм потрібно вести битву з гравцями або командами гравців(зазвичай максимум 100 гравців на матч) доки лише одна команда не залишиться живою і не стане переможцем при цьому зона битви кожні декілька хвилин зменшується. В цілому жанр чимось схожий на МОБА і підходить як для людей у яких багато вільного часу так і для тих хто просто хоче відпочити після роботи або важкого дня.

1.2.4 Study games

Також жанр який набрав популярність не так давно і є найкориснішим ігровим жанром серед усіх, так як його ціль не просто марнування часу, а

вивчення чогось нового. Таких ігор існує достатньо багато і включають вони в себе усі сфери, навчитися можна всьому. Зазвичай такі додатки дають користувачу можливість відстежувати свій прогрес і не потребують багато часу.

Останнім часом почало з'являтися багато схожих додатків але вони все рівно приносять щось нове або щось старе під новим кутом в ігрову індустрію. Тому не важливо який у гри жанр чи на яку аудиторію вона розрахована завжди знайдуться ті хто хоча б спробує її завантажити.

1.3 Огляд найближчих ігор аналогів

1.3.1 Ігри аналоги

– Гра Tap Titans 2

Основною задачею гри є битва з Титанами простими кліками по екрану. Хоча в грі є достатньо багато можливостей але через одноманітність геймплею(постійні кліки по екрану) вона дуже швидко набридає, але ще досі користується популярністю, як «вбивалка часу». Головну сцену гри зображено на рисунку 2.

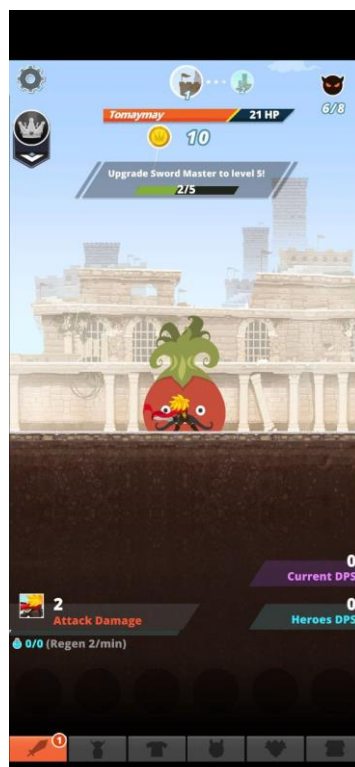


Рисунок 2 – Головна сцена гри Tap Titans 2

– Clicker Heroes

Гра дуже схожа на Tap Titans, де основною задачею є битва з ворогами кліками по екрану, в ній також можна зібрати свою команду яка буде допомагати перемагати ворогів. У гри немає якоїсь конкретної цілі все що повинен робити гравець це з раз у раз починати все спочатку при цьому збираючи реліквії які будуть полегшувати бої. Головну сцену гри зображено на рисунку 3.



Рисунок 3 – Головна сцена гри Clicker Heroes

На рисунку видно, що через погану адаптивність на великому екрані гра створює неприємні чорні зони. Для уникнення такого потрібно правильно масштабувати ігрові об'єкти та протестувати всі можливі варіанти.

– Dynasty Scrolls

Дана гра вже являє собою деяку подобу рольовій грі. В ній є певний сюжет, проходячи який гравець дізнається про історію головного героя та персонажів гри. Ціллю даної гри є збір та посилення своєї команди для проходження сюжету або битви з іншими гравцями. Хоча в грі достатньо багато різноманітного контенту, але все що потрібно робити в ній гравцю- це зібрати команду з певних персонажів кожен з яких має свої особливості і розпочинати битву на певному рівні або з іншим гравцем при цьому сам гравець не може ніяк взаємодіяти з командою під час бою. Головну сцену можна побачити на рисунку 4.



Рисунок 4 – Головна сцена гри Dynasty Scrolls

1.3.2 Переваги та недоліки

	Наявність сюжету	Наявність кінцівки	Різнманітність геймплею	Затрати часу
Defenders(можливо гра)	Продуманий сюжет, на	Є явна сюжетна	Є декілька персонажів	Не потребує високої

	який впливає вибір гравця	кінцівка але вона не заважає грати далі	на вибір, у яких різних спосіб атаки та різні навики	концентрації користувача
Tap Titans 2	Немає сюжету	Немає кінцівки	Одноманітний геймплей	Мінімальні затрати
Clicker Heroes	Немає сюжету	Немає кінцівки	Одноманітний геймплей	Мінімальні затрати
Dynasty scrolls	Є сюжетна лінія але ніяк не впливає на ціль гри	Є певний кінець сюжету але ніяк не впливає на гру	Процес гри завжди однаковий але в залежності від набору команди результат змінюється	Є велика кількість подій в різний час добі тому потрібно заходити по декілька разів на день

Таблиця 1 – Порівняння ігор аналогів

2 ВИБІР ТЕХНОЛОГІЙ І СЕРЕДОВИЩА РОЗРОБКИ

2.1 Обрані технології розробки

2.1.1 Мова програмування C#

C # - проста, сучасна об'єктно-орієнтована мова програмування, але підтримує також і компонентно-орієнтоване програмування. Розробка сучасних додатків все більше тяжіє до створення програмних компонентів у формі автономних і самописних пакетів, що реалізують окремі функціональні можливості. Важлива особливість таких компонентів – це модель програмування на основі властивостей, методів і подій. Кожен компонент має атрибути, які надають відомості про компоненти, а також вбудовані елементи документації. C # надає мовні конструкції, безпосередньо підтримуючих таку концепцію роботи. Завдяки цьому C # відмінно підходить для створення і застосування програмних компонентів.

Ось кілька функцій мови C #, що забезпечують надійність і стійкість додатків: Прибирання сміття автоматично звільняє пам'ять, зайняту знищеними і невикористовуваними об'єктами;

Обробка виключень дає структурований і розширюваний спосіб виявляти і обробляти помилки;

Строга типізація мови не дозволяє звертатися до неініціалізованих змінних, виходити за межі масиву або виконувати неконтрольоване приведення типів.

У C # існує єдина система типів. Всі типи C #, успадковують від одного кореневого типу об'єкт. Таким чином, всі типи використовують загальний набір 6 операцій, і значення будь-якого типу можна зберігати, передавати і обробляти схожим чином.

2.2 Середовище розробки

2.2.1 Unity

Unity – кроссплатформерне середовище розробки ігор. Unity дозволяє створювати додатки, що працюють під більш ніж 20 різними операційних системах, що включають персональні комп'ютери, ігрові консолі, мобільні пристрої, Інтернет-додатки та інші.

Випуск Unity відбувся в 2005 році і з того часу йде постійне розвиток. Спочатку Unity призначався виключно для комп'ютерів Mac, але потім поступово виходили поновлення, що дозволяє працювати під Windows і інші ОС. Основними перевагами Unity є наявність візуального середовища розробки, кроссплатформерності і модульної системи компонентів. До недоліків відносять появу складнощів при роботі з багатоконпонентними схемами і труднощі при підключенні зовнішніх бібліотек.

На Unity написані тисячі ігор, додатків та симуляцій, які охоплюють безліч платформ і жанрів. Разом з тим Unity використовується як великими розробниками, так і незалежними студіями. Редактор Unity має простий Drag & Drop інтерфейс, що складається з різних вікон, який легко налаштовувати, завдяки чому можна виробляти налагодження гри прямо в редакторі. Двигун підтримує тільки мову C #, хоча раніше можна було використовувати C++ та JavaScript.

Проект в Unity ділиться на сцени (рівні) - окремі файли, містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як, власне, об'єкти (моделі), так і порожні ігрові об'єкти - об'єкти, які не мають моделі. Об'єкти, в свою чергу містять набори компонентів, з якими і взаємодіють скрипти. До об'єктів можна застосовувати колізії. У редакторі є система успадкування об'єктів - дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

Вигляд редактора видно на рисунку 5.

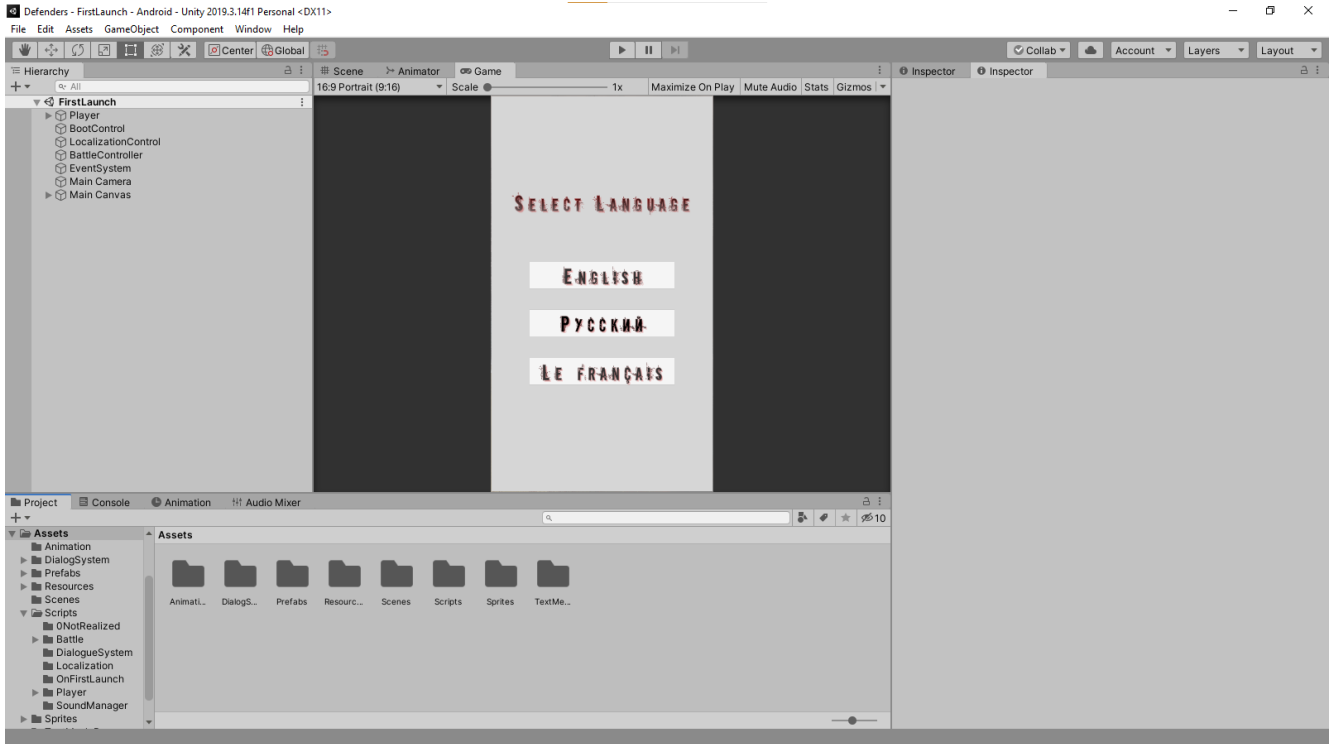


Рисунок 5 – Інтерфейс редактора Unity

2.2.2 Visual Studio 2019 Community

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Visual Studio включає один або декілька з наступних компонентів:

- Visual Basic .NET, а до його появи — Visual Basic;
- Visual C++;
- Visual C#;
- Visual F# (входить до складу Visual Studio 2010);

- Visual Studio Debugger.

Багато варіантів постачання також містять:

- Microsoft SQL Server;
- MSDE Visual Source Safe — файл-серверна система управління версіями.

Інтерфейс IDE показаний на рисунку 6.

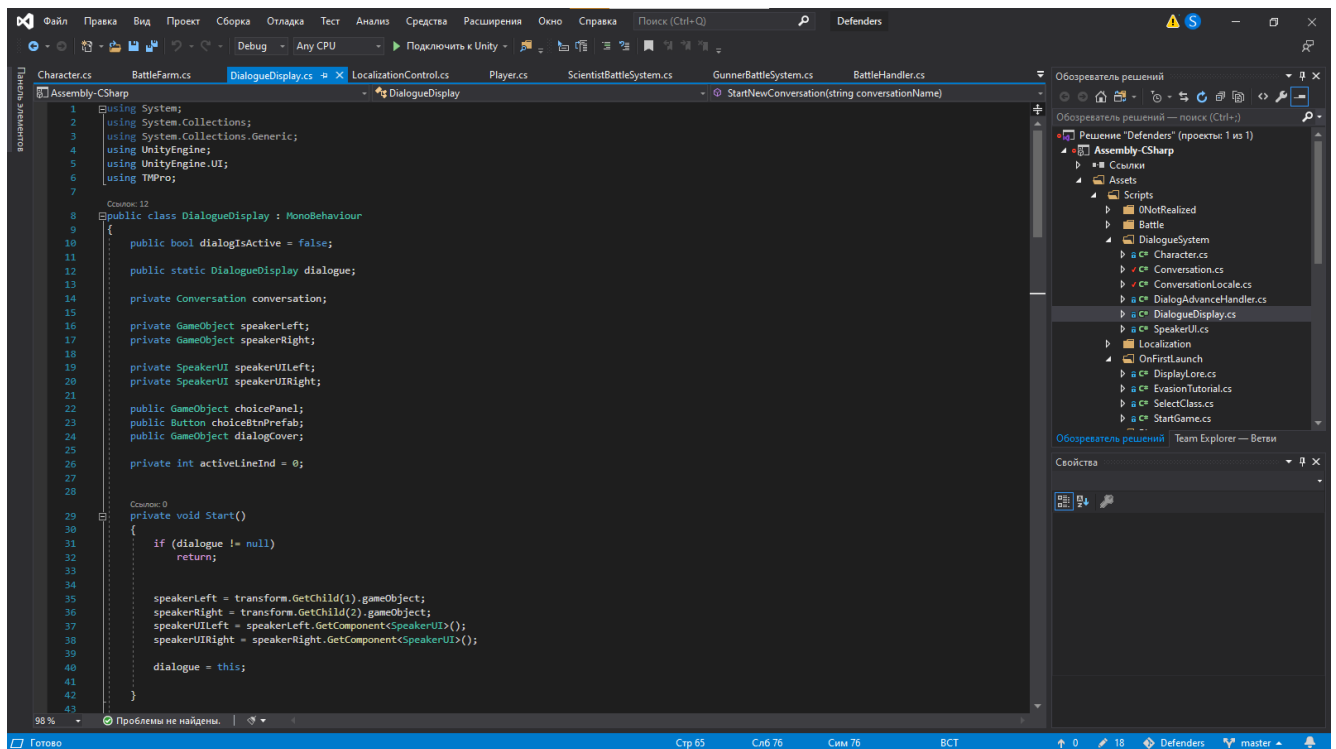


Рисунок 6 – Інтерфейс середовища VS Community 2019

3 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Використані програмні інструменти

3.1.1 Програма SAI

SAI або Easy Paint Tool SAI — легкий растровий графічний редактор та програмне забезпечення для малювання для Microsoft Windows, розроблене та опубліковане Systemax Softwar. Через те що програма займає дуже мало місця її легко можна помістити на флешку і працювати на будь якому комп'ютері. Використовувалася для малювання начерків персонажів та локацій гри для подальшого їх оброблення. Інтерфейс програми зображений на рисунку 7.

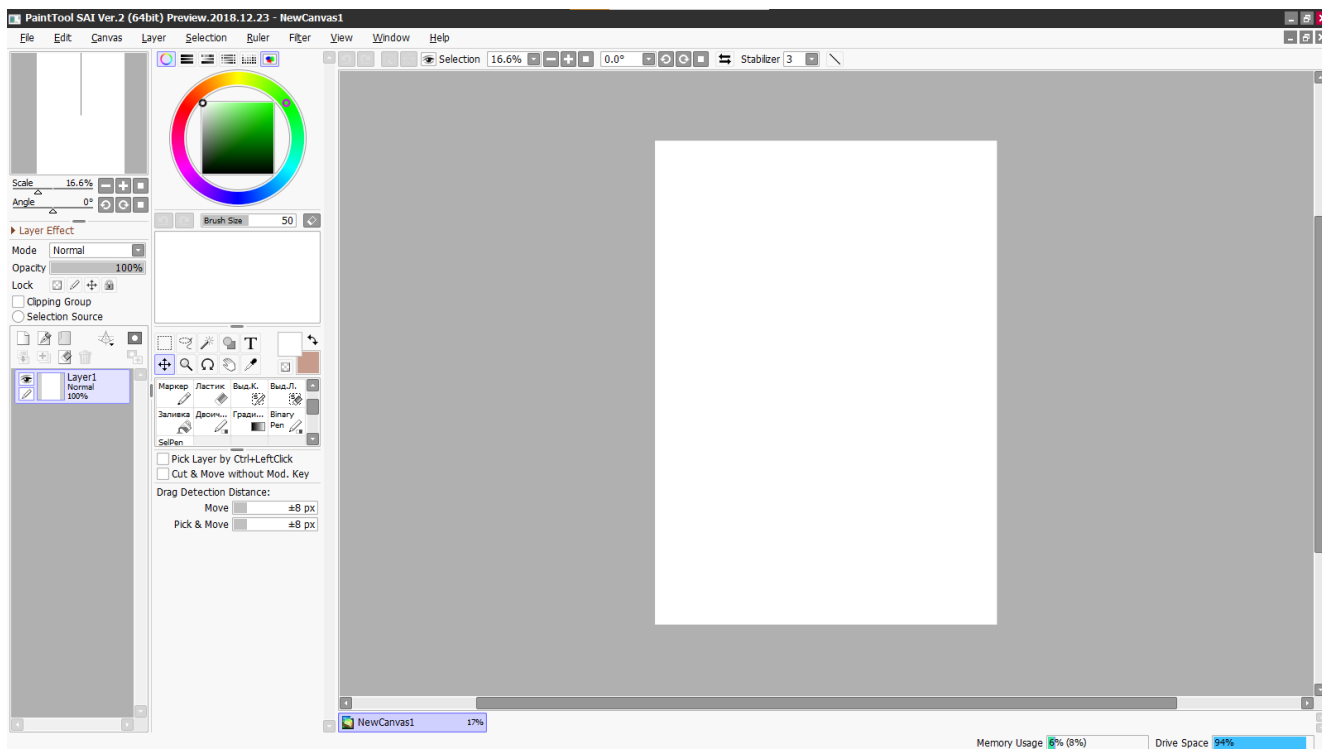


Рисунок 7 – Інтерфейс програми SAI

3.1.2 Microsoft OneNote

Microsoft Office OneNote — це застосунок для створення нотаток і організації особистої інформації від корпорації Microsoft, що є частиною пакету

Microsoft Office.

Найзручнішим є використання програми на планшетному комп'ютері, де присутня можливість рукописного введення тексту і додавання нотаток. Проте, часто Microsoft OneNote використовується і на звичайних комп'ютерах з операційною системою Microsoft Windows.

Також можна встановити на пристрої з: Android, Mac, Windows Phone тощо; доступна також веб-версія.

Інтерфейс програми зображено на рисунку 8.

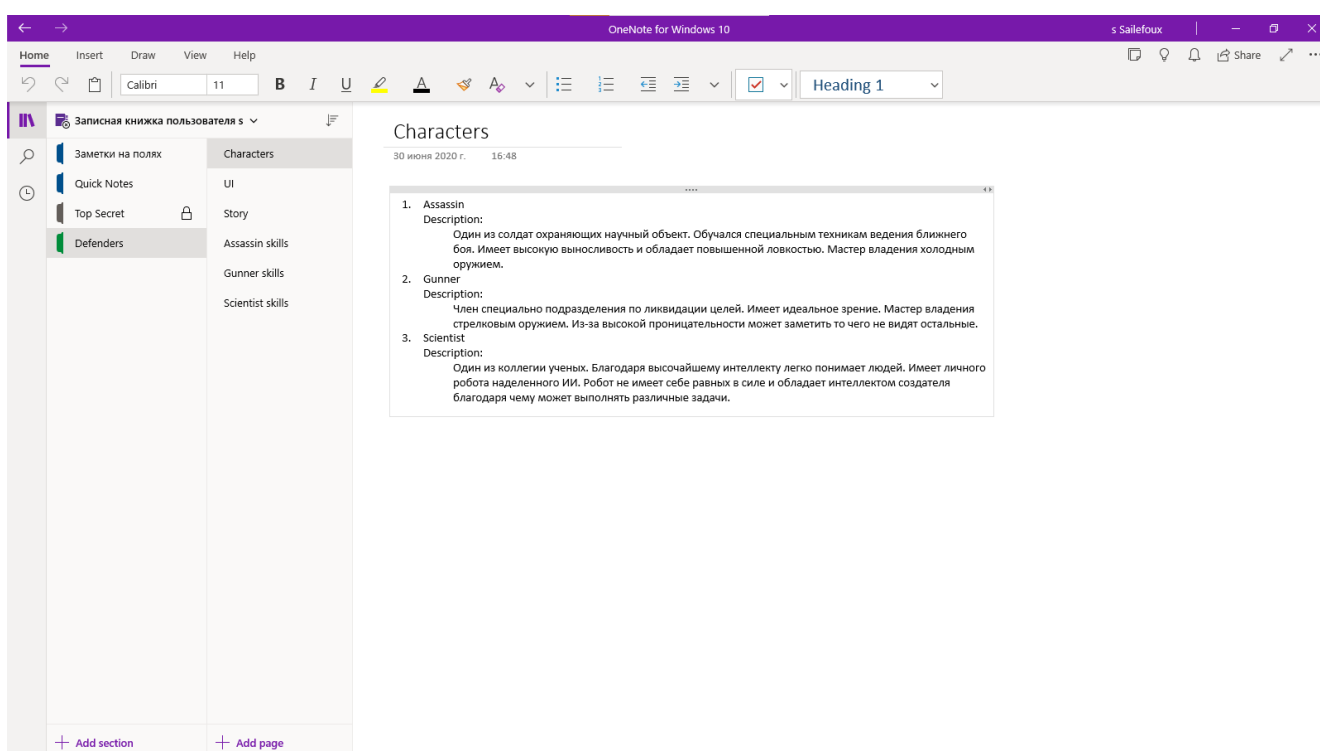


Рисунок 8 – Інтерфейс OneNote

3.2 Основні відомості про гру

Назва: Defenders

Жанр: RPG(рольова гра), Clicker.

Сетінг: постапокаліптика, кіберпанк.

Сюжет: сюжет відбувається в майбутньому де людство освоїло генну інженерію, кібернетику та було збудовано декілька автономних космічних

станцій на яких проживає велика кількість людей. Гравець є захисником якого відправили на планету, де він повинен буде виконувати поручення місцевих жителів.

Геймплей: є декілька персонажів у яких відрізняється стиль бою, набір команди, дослідження, крафт.

Вимоги до пристрою: Android 6 та вище.

Логотип гри зображено на рисунку 9.

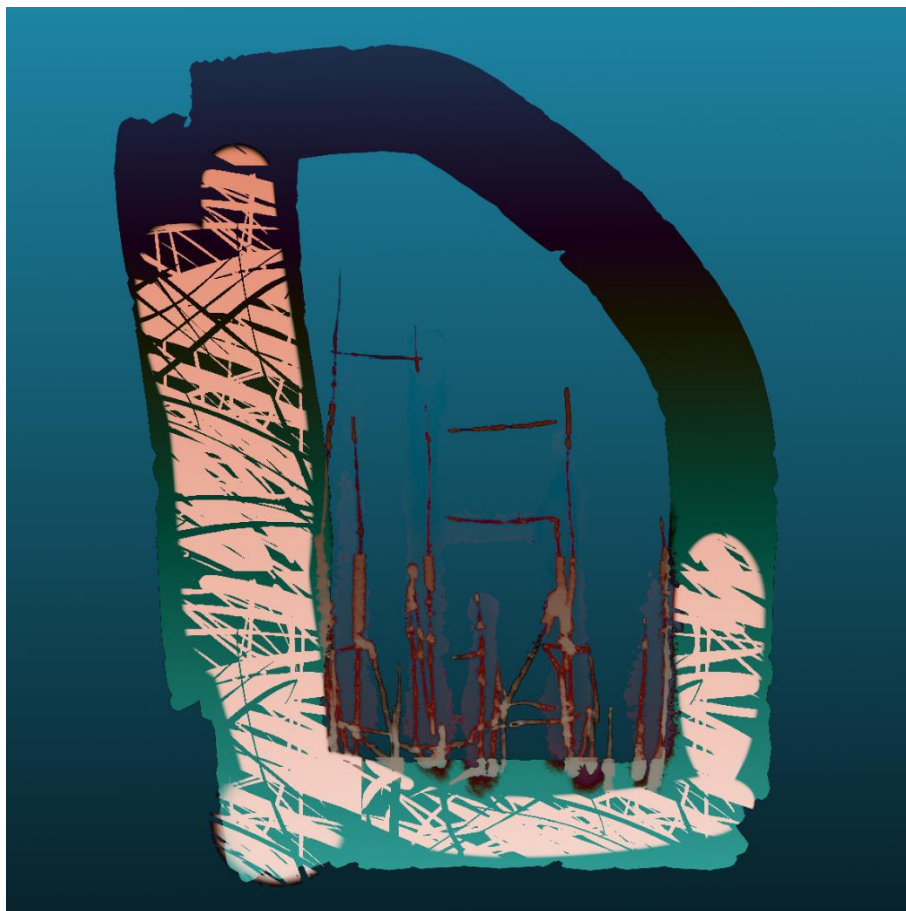


Рисунок 9 – логотип гри Defenders

3.3 Діаграма нотації IDEF0

Процес проектування додатку доцільно розпочати з побудови контекстної діаграми A-0. Така діаграма містить у собі стисле та доцільне пояснення кожного кроку реалізації, що дає змогу оцінити правильність обраного шляху виконання.

Головними елементами даної діаграми є: вхідні дані, вихідні дані, управління та механізми. Зазвичай вказують точку зору (виконавця робіт), предметну область та ціль виконання. Після проведеного аналізу відносно головних елементів контекстної діаграми «Створення рольової гри», було сформовано перелік даних:

- Вхідні дані: завдання на створення гри.
- Вихідні дані: реліз арк-версії готової гри.
- Управління: вимоги до гри, обмеження ПЗ, планування розробки.
- Механізми: команда проекту, програмні засоби, апаратне забезпечення.

Контекстна діаграма А-0, що була розроблена за допомогою програмного забезпечення Visual Paradigm на основі цих даних представлена на рисунку 10.

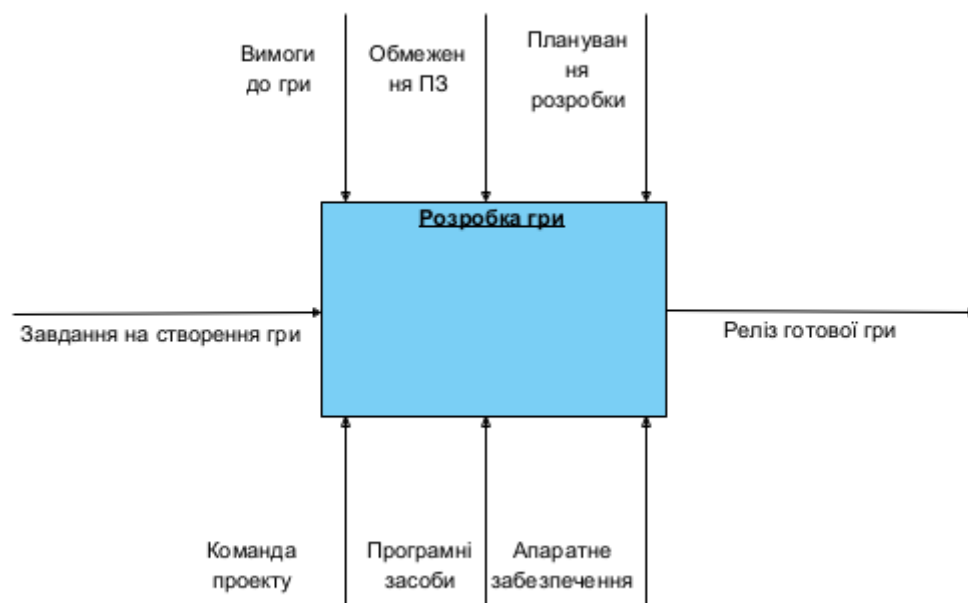


Рисунок 10 – контекстна діаграма А0

Оскільки така діаграма містить у собі лише загальну інформацію щодо системи, то виникає необхідність у виконанні декомпозиції.

Декомпозиція – це розподіл складного об’єкту, системи чи задачі на складові частини або елементи.

Даний процес дозволить детальніше розглянути логіку послідовності виконання робіт для реалізації запланованого продукту. Діаграма А-0 була розбита на три підрівні, а саме: розробка сюжету, реалізація основного функціоналу і тестування та відладка. Це обумовлено тим, що процес створення Android-додатку поділяється на три етапи:

- Перший етап: Написання сюжету гри, написання тексту для завдань гравця.
- Другий етап: реалізація основних механік гри.
- Третій етап: процес тестування гри та процес відладки у випадку знайдених помилок

При описі першого етапу були сформовані такі дані:

- Вхідні дані: завдання на створення гри.
- Вихідні дані: готовий текст сюжету.
- Управління: вимоги до гри, планування розробки.

При описі другого етапу були сформовані такі дані:

- Вхідні дані: файли завдання на створення гри.
- Вихідні дані: готові основні скрипти.
- Управління: вимоги до гри, планування розробки, обмеження ПЗ.
- Механізми: програмні засоби, апаратне забезпечення.

При описі третього етапу були сформовані такі дані:

- Вхідні дані: файли альфа-версії гри.
- Вихідні дані: реліз арк-версії готової гри.
- Управління: вимоги до гри, планування розробки.
- Механізми: програмні засоби, апаратне забезпечення.

Вигляд діаграми А-0 після декомпозиції на 3 етапи наведений на рисунку

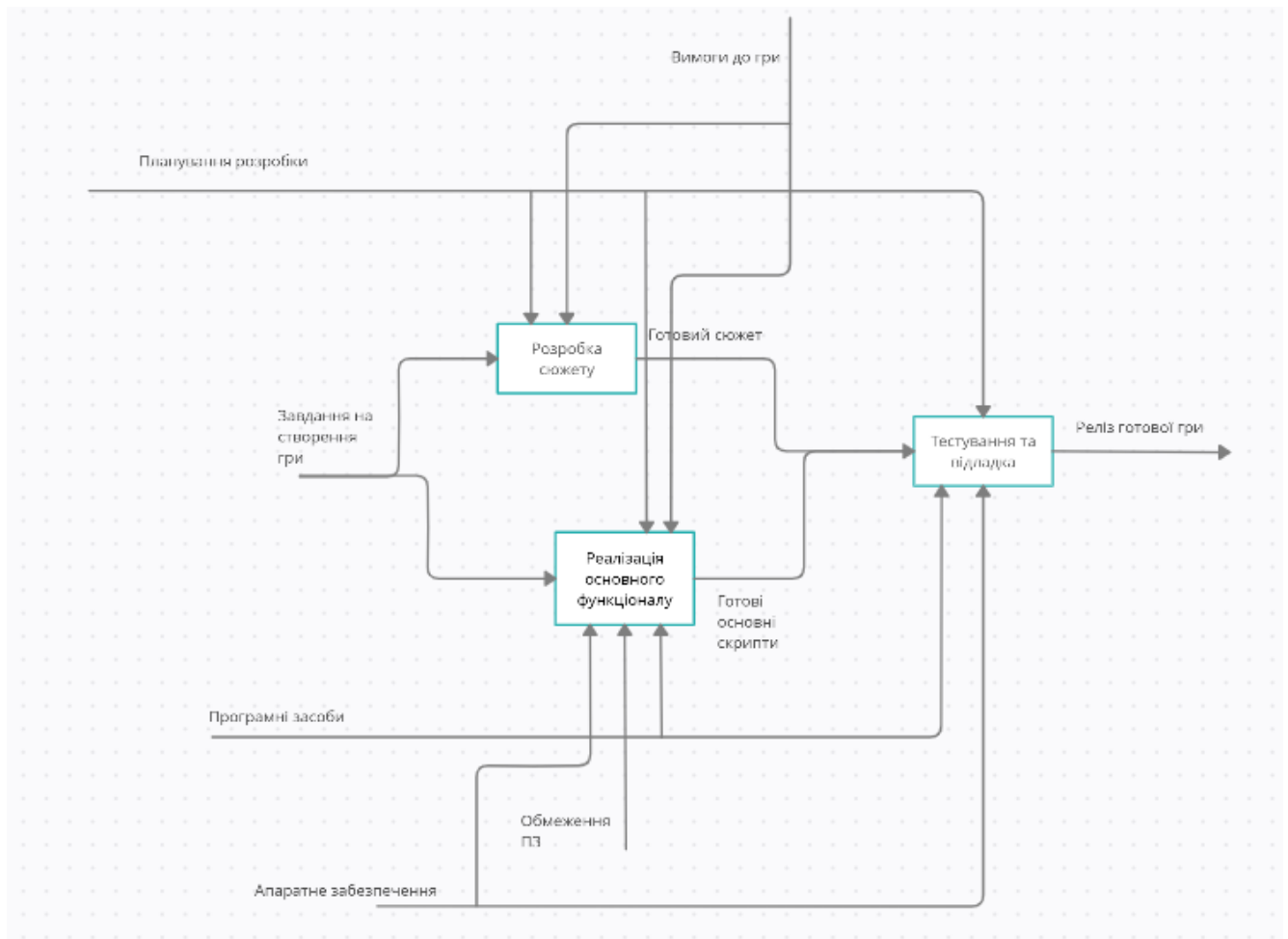


Рисунок 11 – Вигляд діаграми А0 після декомпозиції

3.4 Діаграма використання додатку

Діаграма використання (Use case) — діаграма, на якій зображено відношення між акторами та варіантами використання продукту.

Для розробки діаграми Use Case був визначений один актор – Гравець. Саме він обирає те, що буде відбуватись у грі. Інші актори відсутні, тому що гра не передбачає багатокористувальницького режиму гри через мережу або інтернет та не передбачає зв'язок запитів з базою даних для завантаження або прийняття результатів гри.

Після того, як був визначений актор, що буде взаємодіяти з системою, необхідно сформулювати перелік варіантів використання.

Варіанти використання ігрового Android-додатку:

- Змінити налаштувань додатку;
- Перейти в магазин;
- Перейти в розділ досліджень;
- Перейти на вкладку персонажа;
- Перейти на вкладку крафт;
- Перейти на вкладку квестів;
- Завершення роботи додатку;

Варіант використання «Зміна налаштувань додатку» має три варіанти використання:

- Обрати мову;
- Змінити налаштування звуку;
- Видалити збережені дані.

На основі сформованих даних про актора та всіх можливих варіантів використання ігрового Android-додатку була розроблена діаграма варіантів використання, яка зображена на рисунку 12.

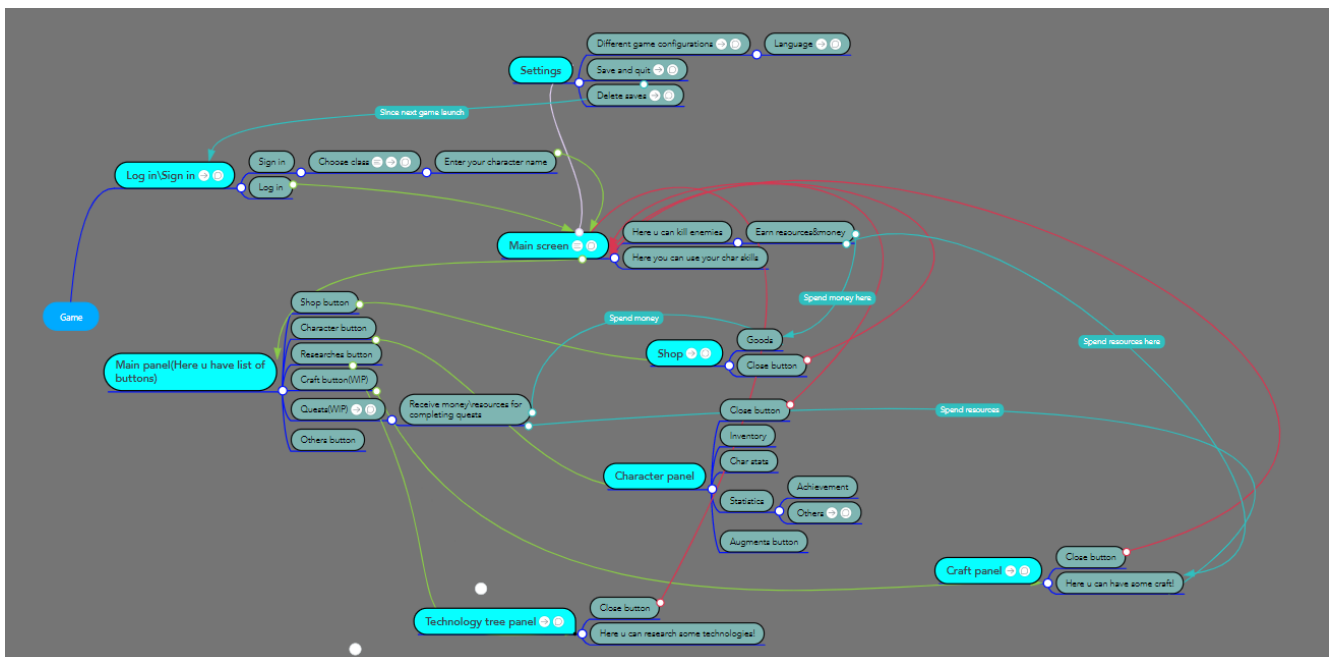


Рисунок 12 – Use case діаграма

3.5 Процес розробки

3.5.1 Розробка файлів локалізації

Перед початком розробки самого Android-додатку необхідно підготувати всі компоненти, що будуть використовуватись при розробці або при збірці проекту. Одним із таких компонентів є файли локалізації. Вони представляють собою масиви даних, що містять у собі слова, словосполучення або навіть речення, що використовуються у користувальницькому інтерфейсі. Зазвичай на кожен мову відводиться по одному окремому файлу в якому і будуть міститися абсолютно всі рядкові, що відповідають обраній мові.

У моєму Android-додатку будуть присутні файли локалізації для трьох мов: англійська, російська та французька. Ці файли є ідентичними за своєю структурою і відрізняються лише перекладом на ту чи іншу мову. Структура такого файлу представляє собою два масиви: масив правил та масив заголовків. У масиві правил містяться актуальні правила, у масиві заголовків містяться заголовки кожної з створених у майбутньому сцен.

3.5.2 Розробка спрайтів та текстур

Для того щоб гра була візуально привабливою необхідно намалювати бекграунди, спрайти персонажів, кнопок та елементів інтерфейсу. Створення графіки буде проводити дизайнер по коротким нарисам, які я представляю. Для того щоб користувач відчув усю атмосферу всі ілюстрації повинні підходити під сетінг гри і виконані в одному стилі.

3.5.3 Розробка звукового супроводження

Для кожної локації буде різна звукова доріжка яка відповідає темі гри. Також під час бою має бути звук ударів та звук використання умінь персонажу. Аудіо доріжки розроблюються самостійно

3.6 Розробка ігрових сцен

3.6.1 Перший запуск

При першому запуску користувач повинен буде обрати мову гри. Після вибору мови буде показана коротка передісторія світу гри і вибір персонажа для того щоб користувач мав змогу підібрати той стиль бою який йому найбільше підходить. Після вибору персонажа користувачу буде запропоновано ввести нікнейм. Вибір мови, передісторію та вибір персонажу зображено на рисунках 13, 14 та 15 відповідно.



Рисунок 14 – Вибір мови

Використано шрифти з відкритого джерела з подальшою кастомізацією.

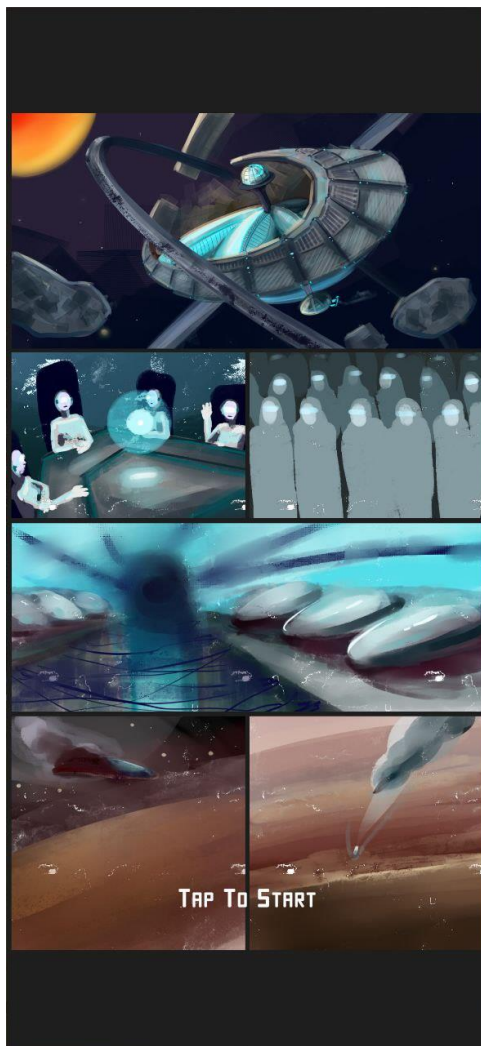


Рисунок 15 – Коротка анімована передісторія

Зображення з'являються поступово з відповідним музичним супроводом.



Рисунок 17 – Вікно вибору персонажа

3.6.2 Основна сцена гри

Основною сценою гри є місто з яким гравець буде взаємодіяти на протязі усієї гри. В цьому місті розташовані неігрові персонажі у кожного з яких є своя функція. Також для зручності всі неігрові персонажі будуть відображатися на верхній панелі у виді відповідно стилізованих кнопок.

У кожного неігрового персонажу буде своя сцена:

- При взаємодії з продавцем буде відображатися сцена магазину в якій користувач за певну кількість деякої ігрової валюти зможе придбати нове спорядження, певні витратні матеріали та інші корисні товари;
- Дослідник буде допомагати гравцеві відкривати нові вміння та технології, які будуть покращувати певну область ігрового процесу.

Дослідження будуть відбуватись поступово з витратою на них часу та деякої кількості ресурсів;

- Ремісник використовується для створення спорядження яке неможливо придбати в магазині. При цьому для його створення будуть використовуватися матеріали, які користувач зможе знайти у процесі проходження квестів та ігрових локацій.
- Командир – у нього в розпорядженні будуть добровольці які зожуть допомагати гравцю в отриманні деяких ресурсів. При цьому вони добувають їх навіть коли користувач не запустив гру, тобто в автономному режимі.

Основну сцену зображено на рисунку 18.



Рисунок 18 – Головна сцена гри

3.7 Розробка ігрових механік

3.7.1 Розробка вікна діалогу та квестів

Для того щоб у користувача була можливість взаємодіяти з неігровими персонажами потрібно в першу чергу розробити вікно діалогу з допомогою якого гравець зможе спілкуватись з персонажами, а далі на основі цього діалогового вікна розробити діалог з можливістю вибору та діалог, в якому гравцю запропонують виконати квест. Вікно діалогу зображено на рисунку 19.

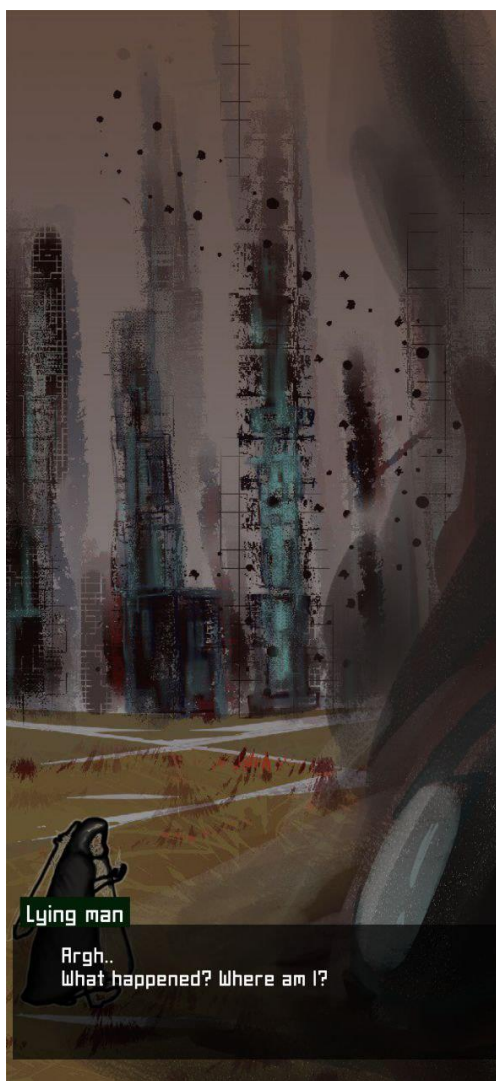


Рисунок 19 – Вікно діалогу

4 РОЗРОБКА СКРИПТІВ

4.1 Скрипти локалізації

4.1.1 LocalizationControl.cs

Основний скрипт локалізації який відповідає за переведення діалогу на потрібну мову. При кожній події зміни мови він змінює посилання на певний мовний пакет.

4.1.2 LocalizedTextHolder.cs та LocalizedTextLoader.cs

Дані скрипти відповідають за зберігання та завантаження відповідно локалізованого тексту. Використовуються виключно для коротких стандартних текстів(наприклад «Назад» , «Підтвердити», «Скасувати»).

4.2 Скрипти діалогу

4.2.1 DialogueDisplay та DialogueAdvanceHandler

Скрипти, які відповідають за відображення та ведення діалогу відповідно. Перший зберігає в собі всі потрібні посилання на персонажів, їх аватари та локалізований текст. Другий же відслідковує коли відбувається подія переходу до наступної фрази і обробляє її.

4.2.2 Character.cs та Conversation.cs

Дані скрипти не виконують прямої їх функції, а замість цього є спеціальним скриптовим об'єктом(Scriptable Object), який являє собою деяку подобу json формату і зберігає в собі всю інформацію про одного персонажа чи один діалог.

4.2.3 ConversationLocale.cs

Є допоміжним скриптом для спрощеного отримання локалізованої версії діалогу. Напряму спілкується зі скриптом контролю локалізації.

4.2.4 Скрипти гравця

4.2.5 Player.cs та PlayerObject.cs

Скрипти які відповідають за всі зміни станів гравця. Перший контролює всі зміни які користувач може виконати(зміна характеристик, початок бою, екіпірування спорядження). Другий є скриптовим об'єктом який всі ці дані зберігає і напряму взаємодіє з першим.

4.2.6 SkillHolder.cs та Skill.cs

Скрипти, які відповідають за відображення графіки на значках умінь гравця та приведення цих умінь в дію. Перший звертається до другого і користуючись повернутими даними підставляє відповідні значки і при натисканні на значок приводить уміння в дію, а другий зберігає в собі всі необхідні дані(графіку, ефект та перезарядку).

4.2.7 Скрипти бойових систем

Так як кожен клас у грі має свої унікальні особливості існує декілька різних скриптів, які містять у собі методи того чи іншого класу.

4.3 Скипти битви

4.3.1 Battlehandler.cs

Скрипт який відповідає за зміну стану гри і розпочинає битву. Є медіатором між гравцем та ворогами

4.3.2 Enemy.cs та EnemyObject.cs

Скрипти, які відповідають за поведінку ворога. Перший за певних умов буде використовувати уміння ворога і атакувати гравця. Другий є об'єктом який зберігає інформацію про ворога(кількість життя, уміння, силу атаки).

4.4 Готовий продукт

Готовий Android-додаток представляє собою скомпільовані створені сцени та скрипти у проекті з використанням різних версій JDK та SDK для підтримки більшості смартфонів з різними Android-версіями. Коду у відкритому доступі немає.

ВИСНОВОК

Під час виконання дипломного проекту був проведений аналіз предметної області. У результаті була виявлена висока унікальність майбутнього проекту. Ідея була придумана автором в якості натхнення були використані деякі механіки ігор, які використовувалися в порівнянні.

В аналітичній частині дипломного проекту були описані вимоги до розробки ігрового Android-додатка і обґрунтований вибір програмного забезпечення для реалізації завдання.

Завдяки аналізу засобів реалізації було обрано певний набір інструментарію для виконання проекту, а саме – мова програмування C#, ігровий двигун Unity3D та растровий редактор SAI.

Для передбачення помилок при створенні Android-додатку були побудовані контекстна діаграма IDEF0 та діаграма варіантів використання.

Процес реалізації Android-додатку був розділений на 3 етапи:

- процес підготовки під час якого було визначено сетінг кінцевого продукту, а також прописано початковий сюжет, та вміння персонажів;
- процес побудови і дизайну ігрових сцен, дизайн вікон функціональних можливостей(магазин, дослідження, крафт);
- процес реалізації програмного коду та пов'язання скриптів і сцен за допомогою ігрового двигуна.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1) Актуальность мобильных приложений, особенности их создания.

[Электронный ресурс] – Режим доступа до ресурсу: <http://blognat.ru/aktualnost-mobilnykhprilozhenijj-osobennosti-ikh-sozdaniya/>.



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
 НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
 ТЕХНОЛОГІЙ
 КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка рольової гри для Android на двигуні Unity

Виконав студент 4 курсу
 групи ПД-44
 Рослутній Ю.О.
 Керівник роботи
 Трінтіна Н.А.

Київ – 2021

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи:** розробити рольову гру для Android на двигуні Unity мовою C#.
- **Об'єкт дослідження:** поліпшення якості та функціоналу ігор в Play Market.
- **Предмет дослідження:** мобільна рольова гра, для Android.

2

АНАЛОГИ



Tap Titans 2



Dynasty Scrolls



Clicker Heroes

3

АНАЛОГИ

	Наявність сюжету	Наявність кінцівки	Різноманітність геймплею	Затрати часу
Defenders(моя гра)	Продуманий сюжет, на який впливає вибір гравця	Є явна сюжетна кінцівка але вона не заважає грати далі	Є декілька персонажів на вибір, у яких різний спосіб атаки та різні навички	Не потребує високої концентрації користувача
Tap Titans 2	Немає сюжету	Немає кінцівки	Одноманітний геймплей	Мінімальні затрати
Clicker Heroes	Немає сюжету	Немає кінцівки	Одноманітний геймплей	Мінімальні затрати
Dynasty scrolls	Є сюжетна лінія але ніяк не впливає на ціль гри	Є певний кінець сюжету але ніяк не впливає на гру	Процес гри завжди однаковий але в залежності від набору команди результат змінюється	Є велика кількість подій в ранній час доби тому потрібно заходити по декілька разів на день

4

ТЕХНІЧНІ ЗАВДАННЯ

- 1. *Прописати сюжет гри.*
- 2. *Створити дизайн ігрових елементів та персонажів.*
- 3. *Написати ігрові скрипти*
- 4. *Випустити гру в Play Market*

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Мова C#



Растровий графічний редактор SAI



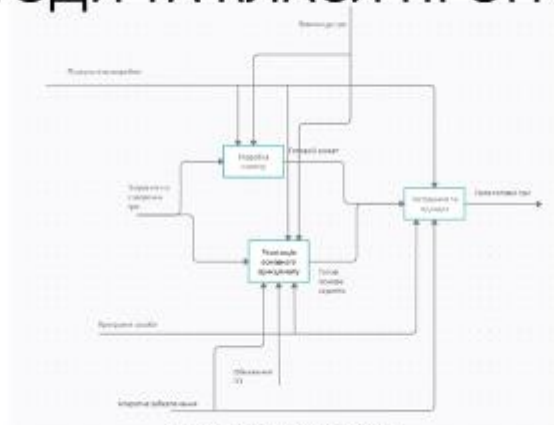
Ігровий двигун Unity



IDE VS Community 2019

6

МЕТОДИ ТА КЛАСИ ПРОГРАМИ



Діаграма нотації IDEF0

7



Use case діаграма

8

ВИСНОВКИ

У процесі виконання випускної роботи, були поставлені та виконані такі задачі:

- Проаналізовано ринок мобільних ігор
- Створена модель об'єкту програмування
- Розроблено необхідні ігрові механіки
- Перевірена функціональність гри

Для подальшого розвитку потрібно далі розвивати сюжет та оновлювати додаток

9

ДЯКУЮ ЗА УВАГУ!