

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

Кафедра інженерії програмного забезпечення

**Пояснювальна записка**

до бакалаврської роботи  
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА ГРИ «SURVIVAL ISLAND» З  
ВИКОРИСТАННЯМ ІГРОВОГО РУШІЯ UNITY МООВОЮ C#»**

Виконав: студент 4 курсу, групи ПД–41  
спеціальності  
121 Інженерія програмного забезпечення  
(шифр і назва спеціальності/спеціалізації)

Павліченко В.Ю.

(прізвище та ініціали)

Керівник Гребенюк В. В.  
(прізвище та ініціали)

Рецензент \_\_\_\_\_  
(прізвище та ініціали)

Київ – 2023

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

Кафедра Інженерії програмного забезпечення  
Ступінь вищої освіти - «Бакалавр»  
Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри  
Інженерії програмного забезпечення  
Негоденко О.В.  
«\_\_\_\_\_» \_\_\_\_\_ 2023 року

**З А В Д А Н Н Я**  
**НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА**

**ПАВЛІЧЕНКА ВАДИМА ЮРІЙОВИЧА**

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка гри «Survival Island» з використанням ігрового рушія unity мовою C#»

Керівник роботи: Гребенюк В. В., доктор філософії, доцент кафедри ІПЗ  
(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 дані до роботи:

3.1 Науково-технічна література з питань, пов'язаних з програмним забезпеченням щодо розробки ігор.

3.2 Офіційна документація Unity.

3.3 Офіційна документація JetBrains Rider.

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 гри.

#### 4.4 Висновки.

#### 5. Перелік демонстраційного матеріалу (назва основних слайдів)

5.1 Титульний слайд.

5.2 Мета, об'єкт, предмет, наукова новизна дослідження.

5.3 Актуальність.

5.4 Аналіз аналогів.

5.5 Технічні завдання.

5.6 Програмні засоби та інструменти реалізації.

5.7 Реалізація програми.

5.8 Апробація результатів дослідження.

5.9 Висновки.

5.10 Кінцевий слайд.

6. Дата видачі завдання « 25 » лютого 2023р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	11.04.2023	Виконано
2	Дослідження аналогів та актуальності додатку	13.04.2023	Виконано
3	Аналіз та вибір інструментів для розробки додатку	14.04.2023	Виконано
4	Проектування та реалізація	02.05.2023	Виконано
5	Вступ, висновки, реферат	17.05.2023	Виконано
6	Розробка обов'язкових демонстраційних матеріалів	18.05.2023	Виконано
7	Попередній захист роботи		
8	Здача роботи	01.06.2023	

Студент \_\_\_\_\_ Павліченко В.Ю.  
( підпис ) (прізвище та ініціали)

Керівник роботи \_\_\_\_\_ Гребенюк В.В .  
( підпис ) (прізвище та ініціали)





## РЕФЕРАТ

Текстова частина бакалаврської роботи 39 с., 2 табл., 30 рис., 8 джерел.  
Ключові слова: ігровий рушій, Unity, сюжет, комп'ютерна гра.

Об'єкт дослідження – геймплей, що базується на реальному виживанні в світі.

Предмет дослідження – технології розробки комп'ютерної гри за допомогою ігрового рушія Unity.

Мета роботи – збільшення зацікавленості потенційних гравців до ігор в жанрі виживання що реалізується за рахунок геймплею, що базується на реальному виживанні в світі.

Наукова новизна дипломної роботи полягає у впровадженні і дослідженні інноваційних геймплейних механік, створенні унікального сценарію та використанні потужних функціональних можливостей Unity для реалізації цього проекту.

У рамках дипломного проекту було проведено аналіз ринку комп'ютерних ігор у жанрі виживання. Було розглянуто переваги та недоліки різних програмних інструментів для розробки ігор. Також детально вивчено особливості розробки комп'ютерних ігор жанру виживання.

Комп'ютерна гра була розроблена за допомогою ігрового рушія Unity, а програмування ігрової логіки проводилося з використанням мови програмування C#.

Ця гра може бути використана як швидкий спосіб ненадовго відволіктися, так і повноцінна гра, яка забезпечує поглиблене занурення у світ гри та відволікає від повсякденних проблем.

## ЗМІСТ

<b>ВСТУП.....</b>	<b>8</b>
<b>1. ОГЛЯД АРХІТЕКТУРИ ІСНУЮЧИХ ІГОР В ЖАНРІ ВИЖИВАННЯ ТА АНАЛІЗ ПРОЕКТУ .....</b>	<b>9</b>
1.1. Аналіз та загальна характеристика комп'ютерних ігор та жанру виживання	9
1.2. Аналіз вже існуючих ігор в жанрі виживання.....	11
1.3. Актуальність проекту.....	14
Висновок до першого розділу.....	17
<b>2. ПРОЕКТУВАННЯ ГРИ В ЖАНРІ ВИЖИВАННЯ.....</b>	<b>18</b>
2.1. Аналіз та обґрунтування вибору середовища розробки проекту.....	18
2.1.1. Unity Game Engine.....	18
2.1.2. Unreal Engine .....	20
2.1.3. GameMaker Studio .....	21
2.2. Загальний алгоритм реалізації проекту.....	23
2.3. Функціонал продукту .....	25
2.4. Підготовка матеріалів.....	25
Висновок до другого розділу.....	29
<b>3. ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ В ЖАНРІ ВИЖИВАННЯ .....</b>	<b>30</b>
3.1. Прототипування гри. UI та UX дизайн.....	30
3.2. Підготовка робочого простору.....	31
3.3. Аналіз використаних для розробки проекту інструментів Unity .....	35
3.4. Написання коду.....	35
3.5. Створення графічних елементів.....	39
Висновок до третього розділу.....	43
<b>ВИСНОВКИ.....</b>	<b>44</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>45</b>
<b>ДОДАТОК А .....</b>	<b>46</b>

## ВСТУП

Об'єкт дослідження – геймплей, що базується на реальному виживанні в світі.

Предмет дослідження – технології розробки комп'ютерної гри за допомогою ігрового рушія Unity.

Мета роботи – збільшення зацікавленості потенційних гравців до ігор в жанрі виживання що реалізується за рахунок геймплею, що базується на реальному виживанні в світі.

Наукова новизна дипломної роботи полягає у впровадженні і дослідженні інноваційних геймплейних механік, створенні унікального сценарію та використанні потужних функціональних можливостей Unity для реалізації цього проекту.

У рамках дипломного проекту було проведено аналіз ринку комп'ютерних ігор у жанрі виживання. Було розглянуто переваги та недоліки різних програмних інструментів для розробки ігор. Також детально вивчено особливості розробки комп'ютерних ігор жанру виживання.

Комп'ютерна гра була розроблена за допомогою ігрового рушія Unity, а програмування ігрової логіки проводилося з використанням мови програмування C#.

Ця гра може бути використана як швидкий спосіб ненадовго відволіктися, так і повноцінна гра, яка забезпечує поглиблене занурення у світ гри та відволікає від повсякденних проблем.



# 1. ОГЛЯД АРХІТЕКТУРИ ІСНУЮЧИХ ІГОР В ЖАНРІ ВИЖИВАННЯ ТА АНАЛІЗ ПРОЕКТУ

## 1.1. Аналіз та загальна характеристика комп'ютерних ігор та жанру виживання

Комп'ютерна гра – це вид розважального або навчального програмного забезпечення, яке виконується на комп'ютері і надає можливість гравцю взаємодіяти з віртуальним світом або сценарієм. Гравець керує персонажами, вирішує завдання, виконує різні дії або просто насолоджується віртуальним середовищем, отримуючи задоволення або навчальну користь від гри. Комп'ютерні ігри можуть бути розроблені для різних платформ, включаючи персональні комп'ютери, ігрові консолі, мобільні пристрої та інші.

Жанр виживання у комп'ютерних іграх характеризується фокусом на виживанні гравця в умовах ворожого або ворожого середовища, де необхідно збирати ресурси, будувати притулок, шукати їжу та воду, виробляти зброя, виконувати завдання та протистояти небезпекам. Цей жанр вимагає від гравця стратегічного мислення, планування ресурсів, вміння приймати рішення в екстремальних умовах та адаптуватися до непередбачуваних ситуацій, що робить його захоплюючим та викликаючим емоційну напругу у гравця.

Дослідження популярних і успішних грів жанру survival є важливим кроком для розуміння основних характеристик, механік та елементів, які визначають цей жанр. Розглянемо деякі з них:

1. Елемент виживання: Одним з основних аспектів жанру survival є необхідність забезпечити виживання персонажа в несприятливих умовах. Це може включати пошук їжі та води, збір ресурсів для виготовлення предметів, будівництво притулку та захист від ворогів або природних небезпек.

2. Розвиток персонажа: Багато ігор жанру survival пропонують систему прогресу та розвитку персонажа. Гравці можуть отримувати досвід, покращувати навички та отримувати нові можливості, що дозволяє їм ставати

сильнішими та більш вправними у виживанні.

3. Випадковість: Багато ігор жанру survival використовують випадковість для створення непередбачуваних ситуацій. Наприклад, генерація випадкової карти, випадкове розташування ресурсів та ворогів, зміна погодних умов тощо. Це робить кожну гру унікальною та вимагає від гравця адаптації до змінних умов.

4. Механіка крафту: Багато ігор жанру survival мають систему крафту, що дозволяє гравцям створювати нові предмети зі зібраних ресурсів. Це може включати виготовлення знарядь для полювання, інструментів для будівництва або захисту, аптечок тощо. Механіка крафту додає гнучкість та можливості для творчого використання ресурсів.

5. Атмосфера та настрої: Однією з ключових складових успіху ігор жанру survival є створення особливої атмосфери та настрою. Це може бути постапокаліптичний світ, дикий острів, пустеля або інша ворожа територія. Використання візуальних ефектів, звукового супроводу та музики допомагає створити поглиблення гравця у гру та відчуття загрози та необхідності виживання.

Продовжуючи дослідження стандартних етапів виживання, гра повинна забезпечувати відповідність цих етапів концепції гри і її особливостям. Основні характеристики та механіки, які визначають жанр survival, повинні бути належним чином втілені в грі. Деякі з них включають:

1. Управління ресурсами: Гравець повинен маніпулювати різними видами ресурсів, такими як їжа, вода, матеріали та інструменти, для задоволення потреб персонажа. Це може включати збирання ресурсів, обробку їжі, ремонт знарядь та інше.

2. Дослідження: Гравець повинен мати можливість досліджувати великий відкритий світ гри, включаючи різноманітні місцевості, локації та підземелля. Це сприяє відкриттю нових ресурсів, секретів та викликів для гравця.

3. Прогресія та розвиток персонажа: Гравець повинен мати

можливість розвивати свого персонажа, покращувати його навички, отримувати нові можливості та виконувати завдання, які ведуть до зростання його сил, навичок і можливостей.

4. Управління ресурсами та здоров'ям: Гравець повинен бути свідомим стану своїх ресурсів, таких як здоров'я, голод і спрага.

Дослідження популярних і успішних ігор жанру survival дозволить з'ясувати, які саме характеристики та елементи привертають увагу гравців, які механіки є ефективними та як можна поєднати їх у своїй власній грі. Таке дослідження дасть змогу зрозуміти потреби та очікування цільової аудиторії і допоможе розробити ігру, яка буде привабливою для гравців та успішно конкурувати на ринку ігрової індустрії.

## **1.2. Аналіз вже існуючих ігор в жанрі виживання**

Жанр виживання (Survival genre) в ігровій індустрії є одним з найпопулярніших та захоплюючих. Він пропонує гравцям унікальний досвід виживання у вражаючих віртуальних світах, де вони змушені боротися за своє виживання у небезпечних умовах, здобувати ресурси, будувати притулок і виробляти знаряддя для боротьби з ворогами або природними загрозами.

Історія жанру виживання у ігровій індустрії сягає далеко назад. Перші витoki жанру можна відстежити до текстових пригодницьких ігор 1970-х та 1980-х років, де гравцям пропонувалося приймати рішення в екстремальних ситуаціях, враховуючи обмежені ресурси та непередбачувані події. Проте, справжній прорив жанру стався з появою комп'ютерних ігор.

Один з відомих прикладів ранніх ігор жанру виживання є «Rogue» (1980) – текстова рогалик-гра, де гравці змагалися за виживання у випадково генерованих підземельних лабіринтах. Ця гра встановила основні елементи жанру, такі як перманентна смерть персонажа і випадкова генерація рівнів (Рисунок 1.1).

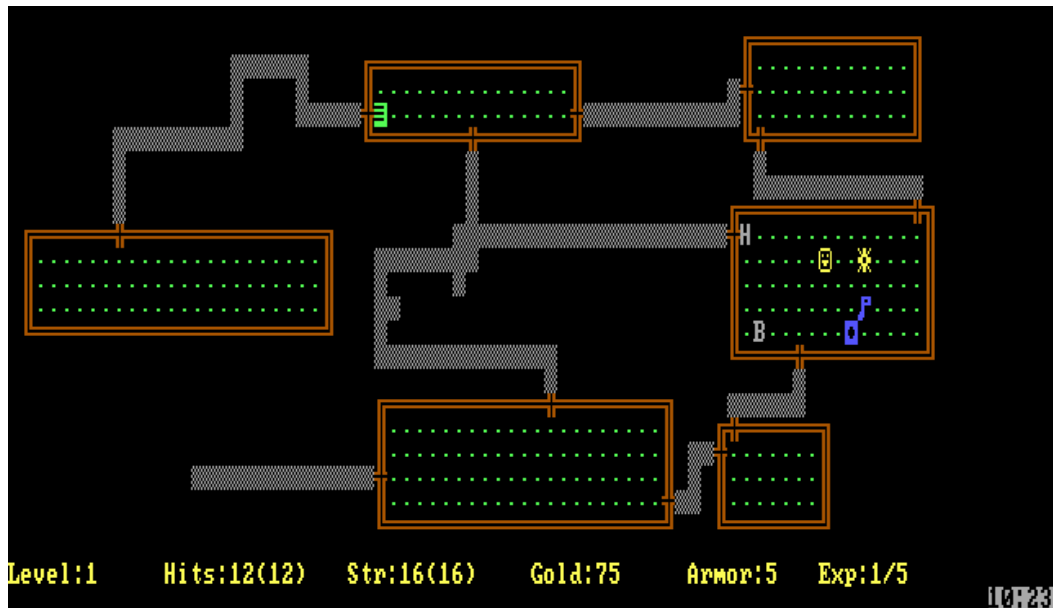


Рисунок 1.1 – Знімок екрану гри «Rogue»

У подальшому, з розвитком технологій ігрової індустрії, жанр виживання отримав нові можливості. З'явилися ігри з тривимірною графікою та реалістичною фізикою, що дозволяли створювати більш іммерсивні та захоплюючі ігрові світи.

Journey - це незвичайна комп'ютерна гра, випущена в 2012 році. Гравець виступає у ролі безіменної фігури, яка подорожує прекрасним пустельним світом. Гра пропонує мальовничі локації, захоплюючу музику та емоційний досвід (Рисунок 1.2). Одним із центральних аспектів гри є спілкування з іншими гравцями, але без можливості вербального або текстового спілкування. Це створює унікальний і майже медитативний досвід, де гравці можуть спільно досліджувати світ, допомагати одне одному та відчутти спільну подорож. Гра Journey отримала велику славу за свою красу, артистичність та здатність передати глибокі емоції, що робить її незабутнім досвідом для гравців.



Рисунок 1.2 – Знімок екрану гри «Journey»

Graveyard Keeper - це комп'ютерна гра, випущена у 2018 році. У грі користувач грає роль старости цвинтаря, який керує власним цвинтарем у середньовічному світі. Основне завдання - доглядати за цвинтарем, поховати тіла, розвивати споруди та торгувати. Гра поєднує управління ресурсами, крафтинг та взаємодію з персонажами. Вибір гравця впливатиме на хід гри, а графіка та гумористичний підхід роблять її унікальною та захоплюючою. Graveyard Keeper - це незвичайний ігровий досвід для тих, хто цікавиться симуляторами та пригодами. (Рисунок 1.3).



Рисунок 1.3 – Знімок екрану гри «Graveyard Keeper»

Усе більше ігор жанру виживання почали поєднувати елементи

виживання з іншими жанрами, такими як головоломки, рольові ігри, екшн та інші. Це дозволило створювати більш різноманітні та цікаві ігрові досвіди для гравців.

Проаналізувавши результати огляду представників цього жанру можна звести отриманні дані в окрему таблицю.

Таблиця 1.1 – Зведені результати аналізу характеристик ігор в жанрі виживання.

Властивості	Rogue	Journey	Graveyard Keeper	<b>Survival Island</b>
Платформи	PC	PS4, PS3	PC	<b>PC</b>
Візуальний стиль, що вирізняє серед інших	-	+	+	<b>+</b>
Управління однією рукою	+	+	-	<b>+</b>
Реіграбельність	-	+	-	<b>+</b>
Система крафту	-	-	+	<b>+</b>
Наявність ворогів	+	-	+	<b>-</b>
Наявність місій	+	+	+	<b>+</b>

### 1.3.Актуальність проекту

Актуальність проекту виявляється у декількох аспектах.

1. Популярність комп'ютерних ігор: Галузь комп'ютерних ігор є однією з найшвидше розвиваючихся і привабливих для користувачів. Ігри є не тільки формою розваги, але й способом відпочинку, навчання та соціальної

взаємодії. Високий попит на нові та захопливі ігрові проекти створює потребу в постійному розробленні та вдосконаленні нових ігор.

2. Технологічний розвиток: Розвиток технологій у галузі комп'ютерної графіки, штучного інтелекту, фізичної симуляції та інших сфер інформаційної технології стимулює створення більш складних і реалістичних ігрових досвідів. Розробка проекту з використанням популярного ігрового рушія Unity та мови програмування C# дозволяє використати сучасні технології та створити гру з високоякісною графікою та реалістичною фізикою.

3. Навчальні можливості: Розробка ігри з використанням Unity та мови C# надає студентам можливість практично застосувати свої знання та навички в галузі розробки програмного забезпечення. Вивчення процесу розробки ігор дозволяє краще розуміти алгоритми, архітектуру програмного забезпечення та використання візуальних інструментів для створення графічного інтерфейсу та геймплею.

4. Дослідницький аспект: Розробка гри відкриває можливості для дослідження ігрових механік, взаємодії з гравцем, наративних структур, ефектів та інших аспектів геймдизайну. Це дозволяє вдосконалювати та вивчати нові підходи до створення ігрового досвіду, впливати на емоційну взаємодію гравця з грою та розробляти нові інноваційні ідеї.

5. Розвиток української ігрової індустрії: Розробка ігор національними розробниками є важливим чинником розвитку української ігрової індустрії. Проект, що має локалізацію на українську мову, сприяє поширенню ігор на внутрішньому ринку та впливає на популяризацію української геймдев-спільноти як на внутрішньому, так і на зовнішньому рівнях.

6. Креативний потенціал: Розробка власної гри дає можливість реалізувати креативні ідеї, втілити унікальний наратив, створити неповторну атмосферу та дослідити нові жанрові комбінації. В процесі розробки студент може розвинути свої навички у геймдизайні, візуальному мистецтві, звуковому дизайні та інших аспектах гри, що сприятиме його особистому і професійному

зростанню.

Отже, актуальність проекту виявляється у його відповідності сучасним тенденціям і потребам галузі ігрової розробки, можливостям дослідження та творчого розвитку.

#### **1.4. Об'єкт, предмет та мета проекту**

Визначивши актуальність проекту перейдемо до формулювання об'єкту та предмету дослідження. А також необхідним є визначення мети роботи.

Об'єкт дослідження – геймплей, що базується на реальному виживанні в світі.

Предмет дослідження – технології розробки комп'ютерної гри за допомогою ігрового рушія Unity.

Мета роботи – збільшення зацікавленості потенційних гравців до ігор в жанрі виживання що реалізується за рахунок геймплею, що базується на реальному виживанні в світі.



## Висновок до першого розділу

В цьому розділі було підкреслено, що комп'ютерна гра – це вид розважального або навчального програмного забезпечення, яке виконується на комп'ютері і надає можливість гравцю взаємодіяти з віртуальним світом або сценарієм. Від різновидів взаємодії гравця з грою і основної її мети визначається жанр. Одним з таких жанрів є виживання. Її основні характеристики:

1. Елемент виживання;
2. Розвиток персонажа;
3. Випадковість;
4. Механіка крафту;
5. Атмосфера та настрій.

Наступним в цьому розділі було розглянуто вже існуючі ігри розроблені в жанрі виживання на прикладі таких продуктів як:

1. Rogue;
2. Journey;
3. Graveyard Keeper.

Було проаналізовано їх особливості, загальні риси, переваги та недоліки, технології, що були використані при розробці цих ігор та їх загальна популярність і актуальність у суспільстві.

Також була проаналізована актуальність розробки нової гри в цьому жанрі. Оскільки на ринку комп'ютерних ігор кількість користувачів з кожним роком збільшується, а аудиторія, що любить жанр візуальних ігор тільки росте – гра матиме свою актуальність.

Сформована мета проекту – збільшення зацікавленості потенційних гравців до ігор в жанрі виживання що реалізується за рахунок геймплею, що базується на реальному виживанні в світі.

## 2. ПРОЕКТУВАННЯ ГРИ В ЖАНРІ ВИЖИВАННЯ

### 2.1. Аналіз та обґрунтування вибору середовища розробки проекту

У світі комп'ютерних ігор розробка власної гри є захоплюючою та стимулюючою справою. Для успішної реалізації проекту необхідно обрати відповідне середовище розробки, яке забезпечить необхідні інструменти та можливості для створення професійної гри. Одними з найпопулярніших та потужних середовищ є Unity, Unreal Engine та GameMaker Studio.

Наведемо огляд трьох програмних засобів, їх основні характеристики, переваги та недоліки.

#### 2.1.1. Unity Game Engine

Unity Game Engine є одним з найпопулярніших і потужних програмних засобів для розробки ігор, включаючи гри жанру «Survival». Основні характеристики Unity включають:

1. Потужність: Unity надає широкі можливості для створення графічно багатих ігор з різноманітними ефектами і фізикою.
2. Кросплатформенність: Гри, розроблені в Unity, можуть бути легко портовані на різні платформи, такі як Windows, macOS, iOS, Android, Xbox, PlayStation та інші.
3. Широкі можливості розробки: Unity забезпечує доступ до багатьох інструментів та ресурсів, які допомагають розробникам створювати гри, включаючи готові компоненти, скриптовання на мові C# та візуальний редактор.

Переваги Unity для розробки гри «Survival Island» включають:

1. Широкі можливості розробки: Unity надає розробникам багато інструментів і ресурсів, які допомагають створювати комплексну гру, включаючи графіку, фізику, звук, штучний інтелект та інші аспекти гри.
2. Велика спільнота розробників: Unity має велику спільноту

розробників, що означає наявність великої кількості документації, підручників, форумів та інших ресурсів, які можуть допомогти розробникам у вирішенні проблем та отриманні підтримки.

Недоліки Unity для розробки гри «Survival Island» можуть включати:

1. Висока складність для новачків: Unity може бути дещо складним для вивчення і розуміння для новачків, особливо якщо вони не мають попереднього досвіду в розробці ігор.

2. Вартість: Unity має безкоштовну версію для особистого використання, але для комерційної розробки можуть бути необхідні платні пакети або додаткові розширення.

3. Залежність від Unity-екосистеми: Розробка гри в Unity може обмежуватись функціями та можливостями, які надаються самим Unity та його екосистемою. Розробники можуть бути залежні від оновлень та підтримки Unity.

Програмне забезпечення Unity включає в себе інтегровану розробку середовища, візуальний редактор, засоби для моделювання 3D-об'єктів, налаштування фізики, розробки штучного інтелекту, створення анімацій, компіляції та розгортання гри на різних платформах (Рисунок 2.1).

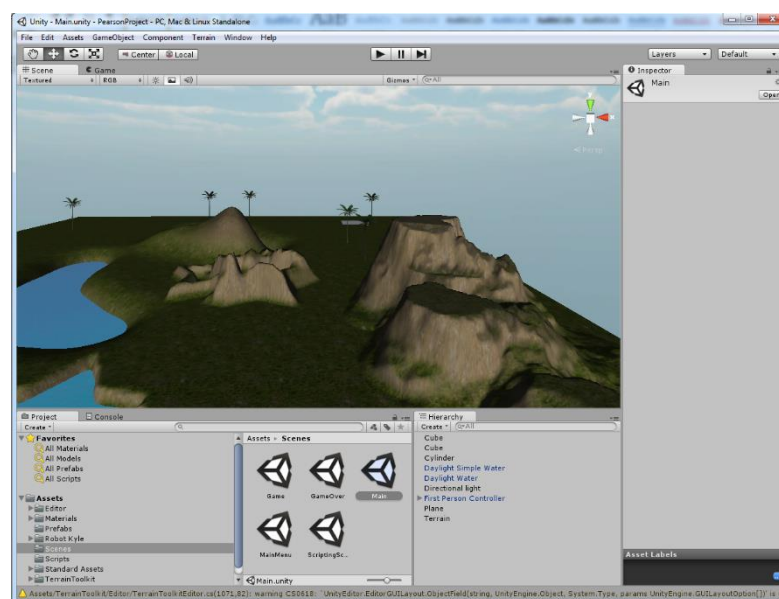


Рисунок 2.1 – Інтерфейс ПЗ Unity

### 2.1.2. Unreal Engine

Unreal Engine є ще одним потужним програмним засобом для розробки ігор, який також може бути використаний для створення гри жанру «Survival». Основні характеристики Unreal Engine включають:

1. Графічна потужність: Unreal Engine забезпечує високоякісну графіку, фотореалістичні ефекти, фізику і освітлення, що дозволяє створювати деталізовані і вражаючі візуальні ефекти.

2. Віртуальна реальність: Unreal Engine має вбудовану підтримку віртуальної реальності, що дозволяє створювати ігри, які підтримують VR-пристрої і забезпечують іммерсивний геймплей.

3. Широкі можливості розробки: Unreal Engine надає розробникам доступ до багатьох інструментів, бібліотек та ресурсів, які допомагають створювати складні механіки гри, включаючи штучний інтелект, фізику, звукові ефекти та багато іншого.

Переваги Unreal Engine для розробки гри «Survival Island» включають:

1. Потужний графічний двигун: Unreal Engine забезпечує високоякісну графіку і візуальні ефекти, що дозволяє створювати реалістичне середовище для гри.

2. Велика спільнота розробників: Unreal Engine має активну спільноту розробників, що означає доступ до великої кількості ресурсів, документації, форумів та підтримки.

Недоліки Unreal Engine для розробки гри «Survival Island» можуть включати:

1. Вимогливість до апаратного забезпечення: Unreal Engine може вимагати потужного комп'ютера для розробки і тестування гри, особливо якщо використовуються високоякісні графічні ефекти.

2. Висока складність для новачків: Unreal Engine може бути складним для вивчення та освоєння для розробників без попереднього досвіду в галузі розробки ігор або програмування.

Програмне забезпечення Unreal Engine включає в себе інтегроване

середовище розробки, візуальний редактор, редактор матеріалів, інструменти для створення 3D-об'єктів, анімації, програмування на мові Blueprint або C++, а також компіляцію та розгортання гри на різних платформах (Рисунок 2.2).

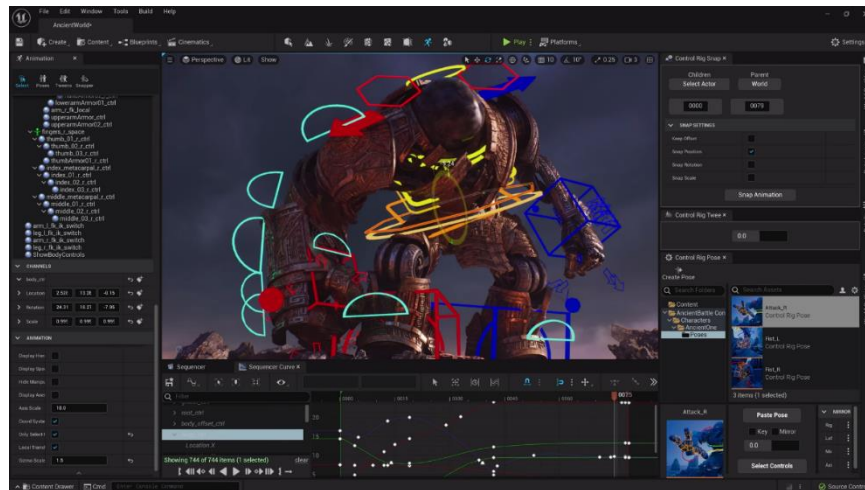


Рисунок 2.2 – Інтерфейс ПЗ Unreal Engine

### 2.1.3. GameMaker Studio

GameMaker Studio є програмним засобом для розробки 2D-ігор, який також може бути використаний для створення гри жанру «Survival». Основні характеристики GameMaker Studio включають:

1. Простота використання: GameMaker Studio має інтуїтивний візуальний редактор, що дозволяє створювати гру без необхідності програмування. Він також надає мову програмування GML (GameMaker Language) для розширених можливостей.

2. Розширюваність: GameMaker Studio дозволяє розробникам використовувати власні скрипти та розширення, що дає більшу гнучкість у створенні унікальних механік та функцій гри.

3. Кросплатформенність: Гри, розроблені в GameMaker Studio, можуть бути експортовані на різні платформи, включаючи Windows, macOS, iOS, Android та інші.

Переваги GameMaker Studio для розробки гри «Survival Island» включають:

1. Легкість використання: GameMaker Studio підходить для початківців і розробників без попереднього досвіду в розробці ігор.
2. Вбудовані функції: GameMaker Studio має велику кількість вбудованих функцій і засобів, що допомагають створювати базові механіки виживання, такі як управління персонажем, ресурси, штучний інтелект і т.д.
3. Швидке прототипування: GameMaker Studio дозволяє швидко створювати прототипи гри, що дозволяє швидко перевірити концепцію та геймплей перед детальною розробкою.

Недоліки GameMaker Studio для розробки гри «Survival Island» можуть включати:

1. Обмеженість у 3D-графіці: GameMaker Studio більш придатний для 2D-ігор, тому для створення складних 3D-моделей та ефектів можуть знадобитись додаткові інструменти або інтеграція з іншими програмами.
2. Обмеженість в розширеному програмуванні: Хоча GameMaker Studio має мову програмування GML для розширеного програмування, вона може бути обмеженою порівняно з мовами програмування, такими як C# або C++.

Програмне забезпечення GameMaker Studio має вбудоване середовище розробки, візуальний редактор для створення рівнів, об'єктів, анімацій, інструменти для налаштування фізики, звуку, колізій та інші ресурси (Рисунок 2.3).



Рисунок 2.3 – Інтерфейс ПЗ GameMaker Studio

Порівняльна характеристика розглянутих додатків наведено у таблиці 2.1.

Таблиця 2.1 – Порівняльна таблиця програмних засобів для розробки гри у жанрі виживання

<b>Програмний засіб</b>	<b>Основні характеристики</b>	<b>Переваги</b>	<b>Недоліки</b>
Unity Game Engine	Потужність, кросплатформенність, широкі можливості	Широкі можливості розробки, велика спільнота розробників	Висока складність для новачків, можливі проблеми з оптимізацією
Unreal Engine	Висока якість графіки, реалістичні фізичні ефекти	Висока якість графіки, розширені можливості розробки	Вимогливість до апаратного забезпечення, складність для новачків
GameMaker Studio	Простота використання, швидкість розробки	Легкість вивчення, візуальне програмування	Обмежені можливості порівняно з іншими інструментами розробки

## 2.2. Загальний алгоритм реалізації проекту

Загальний алгоритм реалізації проекту може включати наступні кроки:

1. Визначення концепції гри: Спочатку необхідно визначити концепцію та основні ідеї гри. Це включає вибір жанру, створення унікального сюжету, геймплею та інших ключових елементів;

2. Розробка дизайну гри: Наступним кроком є розробка дизайну гри,

включаючи візуальні ефекти, графіку, звукове супроводження та інтерфейс користувача. Це допомагає створити привабливе та зручне для гравців оточення;

3. Створення ігрових механік: Для реалізації гри потрібно визначити ігрові механіки, такі як керування гравцем, фізика об'єктів, штучний інтелект та інші геймплейні елементи. Це вимагає програмування і використання мови C# для створення логіки гри;

4. Розробка завдань: Для забезпечення гри необхідно розробити рівні та завдання, які будуть виконувати гравці. Це включає створення локацій, завдань та інших елементів, які роблять гру цікавою та викликають захоплення;

5. Тестування та налагодження: Після реалізації гри важливо провести тестування для виявлення помилок, недоліків та вдосконалення геймплею. Налagodження гри допоможе покращити її стабільність та оптимізувати роботу;

6. Документація: Документацією програмного забезпечення вважається супроводжуючі документи з описом деталей необхідних для використання продукту. До документації відносять :відомість власників оригіналів, текст програми, опис програми, програма і методика випробувань, технічне завдання, пояснювальна записка, тощо

7. Впровадження та підтримка проекту у майбутньому: Останній крок полягає в оптимізації гри, щоб забезпечити її оптимальну продуктивність та сумісність з різними платформами. Важливо виявити та усунути можливі проблеми з продуктивністю, завантаженням ресурсів та іншими технічними аспектами гри. Після завершення оптимізації гра готова до релізу, тобто до випуску на ринок.

Загальний алгоритм реалізації проекту включає кроки від концептуалізації та дизайну до програмування, тестування та оптимізації. Цей підхід допомагає забезпечити ефективний процес розробки та створити якісну та захоплюючу гру для гравців.



### 2.3. Функціонал продукту

Розроблений продукт являє собою комп'ютерну гру в жанрі виживання. Основна мета самої гри – забезпечити гравцю поглиблене занурення у світ гри та відволікти від повсякденних проблем.

Отже основні вимоги до продукту є такими:

1. Різноманітність ігрових механік задля покращення реіграбельності.
2. Система крафту задля зацікавлення гравця залишитись в світі та покращуватись.
3. Рандомна генерація ресурсів в світі, щоб при новій спробі змінити досвід гравця.
4. Керування, що дозволяє грати в гру однією рукою.
5. Відображення всіх візуальних та текстових частин гри

В першу чергу реалізуються саме ці функції продукту, так як є основними реалізаціями критеріїв створення гри в жанрі виживання.

### 2.4. Підготовка матеріалів

Отже розробка гри в жанрі виживання починається з ідеї сюжету та його написання. Важливу роль відіграє його складність та варіативність. Було обрано реалізовувати лінійний сюжет. Сюжет гри зручний та зрозумілий користувачу: необхідно шукати провізію на безлюдному острові для виживання. За допомогою різноманітних інструментів (кирка, сокира, молоток, тощо), які персонаж може зберігати в інвентарі (Рисунок 2.4) та природних ресурсів (каміння, дерево) персонаж має змогу відкривати нові локації та продовжувати шукати там різноманітні ресурси задля виживання (Рисунок 2.5). Процес створення локації можна побачити на рисунку 2.6.



Рисунок 2.4 – Процес створення інвентарю



Рисунок 2.5 – Проектування завдання для відкриття нової локації

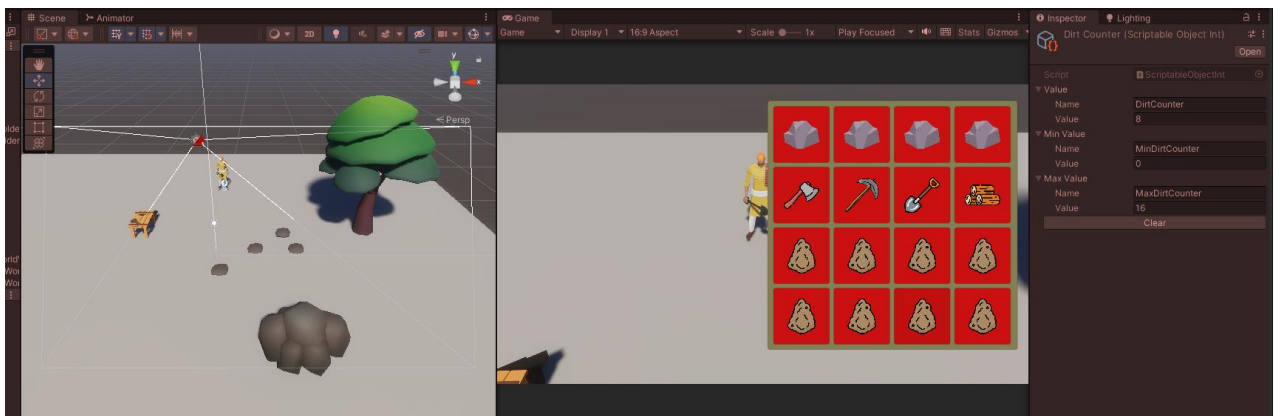


Рисунок 2.6 – Процес створення локації

Також було дуже важливо реалізувати анімацію того, як персонаж добуває різноманітні ресурси (рубав дерево, каміння, тощо). Процес реалізації цієї задачі представлений на рисунку 2.7.



Рисунок 2.7 – Процес створення анімації під час добування ресурсів персонажем

У грі також передбачена можливість крафту, тобто виготовлення зброї зі знайдених природних ресурсів на острові (Рисунок 2.8), а також реалізовано анімацію руху персонажу (Рисунок 2.9).



Рисунок 2.8 – Система крафту



Рисунок 2.9 – Анімація руху персонажу

Для того, щоб гравцю було зрозуміло, що він обрав потрібний інструмент, реалізовано анімацію тримання в руці того чи іншого предмету (Рисунок 2.10).



Рисунок 2.10 – Анімація тримання в руці предмету

## Висновок до другого розділу

Цей розділ був присвячений підготовці для створення майбутнього програмного забезпечення, а саме комп'ютерної гри в жанрі виживання.

В першу чергу було проаналізовано інструменти реалізації комп'ютерних в цілому. Було детально проаналізовано рушії Unity Game Engine, Unreal Engine, GameMaker Studio. Розглянуто головні недоліки та переваги вищезазначених розширень для цих рушіїв.

Обрано рушієм для реалізації та розробки власного проекту Unity, а мовою програмування було обрано C#.

Наступним кроком був написаний детальний алгоритм створення програмного продукту в цілому, виділеними етапами розробки стали такі пункти як:

1. Визначення концепції гри;
2. Розробка дизайну гри;
3. Створення ігрових механік;
4. Розробка завдань;
5. Тестування та налагодження;
6. Документація;
7. Впровадження та підтримка проекту у майбутньому.

За проведеним аналізом було визначено головні функції гри в жанрі виживання:

1. Забезпечення виживання гравця;
2. Пошук ресурсів;
3. Виготовлення знарядь та зброї;
4. Взаємодія з оточуючим середовищем.

В першу чергу будуть реалізовані саме ці задачі для майбутнього додатка. Додатково було розроблено дизайн персонажу гри і спроектовані основні локації, а також основне меню, меню інвентарю, тощо.

### 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ В ЖАНРІ ВИЖИВАННЯ

#### 3.1. Прототипування гри. UI та UX дизайн

Прототипування гри і розробка UI/UX дизайну є важливими етапами процесу розробки комп'ютерної гри. На цьому етапі створюються прототипи інтерфейсу гри, які допомагають визначити структуру гри, навігацію, взаємодію з користувачем та візуальне представлення.

Прототипування гри полягає у створенні простого та інтерактивного варіанту гри, який дозволяє перевірити функціональність та геймплейні механіки. Це може бути прототип рівня, де гравець може пересуватися та взаємодіяти з об'єктами, або прототип інтерфейсу, де можна спробувати основні операції та функції гри.

UI (інтерфейс користувача) та UX (взаємодія користувача) дизайн включає розробку графічного вигляду, анімації, кольорової гами, шрифтів, а також розташування та взаємодію елементів інтерфейсу. Мета дизайну - забезпечити зручність, логічність та естетичність інтерфейсу, щоб користувачі могли легко орієнтуватися та взаємодіяти з грою.

Під час прототипування гри і розробки UI/UX дизайну важливо залучати тестування та збирати фідбек від потенційних користувачів. Це дозволяє виявити слабкі місця і внести відповідні зміни ще на ранніх стадіях розробки. Якісний UI/UX дизайн сприяє поліпшенню користувацького досвіду та популярності гри серед гравців.

За допомогою UI/UX дизайну в даній грі були розроблені основне меню, меню інвентарю та крафту, а також стрічка, яка показує гравцю рівень ситості персонажу, що прямо впливає на його виживання (Рисунок 3.1).



Рисунок 3.1 – Рівень ситості персонажу

### 3.2. Підготовка робочого простору

Підготовка робочого простору є важливим кроком перед початком розробки проекту. Цей етап включає наступні дії:

1. Встановлення необхідного програмного забезпечення: Для розробки гри з використанням ігрового рушія Unity мовою C# необхідно було встановити програмне забезпечення Unity та інтегроване середовище розробки (IDE) JetBrains Rider.
2. Налаштування робочого середовища;
3. Налаштування версійного контролю: Для ефективної роботи з розробкою проекту використовувалась система версійного контролю Git (Рисунок 3.2). Це дозволило зберігати та керувати версіями коду і ресурсів.



Рисунок 3.2 – Керування версіями коду за допомогою GitHub

Вікно проекту (Рисунок 3.3) є одним з найважливіших інструментів Unity, адже саме тут розробник має можливість редагувати файли програми.

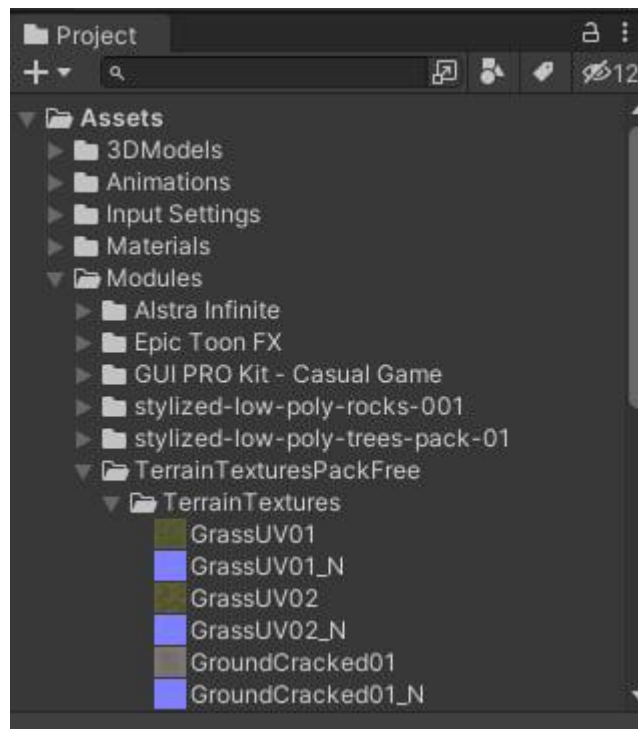


Рисунок 3.3 – Вікно проекту

Інспектор – це один з основних інструментів, які надає ігровий рушій Unity для огляду та редагування компонентів об'єктів в грі. Це вікно знаходиться в інтегрованій середовищі розробки Unity та відображає список компонентів, прикріплених до вибраного об'єкта.

Інспектор надає розробникам можливість переглядати та змінювати властивості компонентів, такі як позиція, розмір, кольори, параметри фізичної взаємодії, анімації та інші. Він також відображає інформацію про скрипти, прикріплені до об'єктів, і дозволяє редагувати їх код безпосередньо в середовищі розробки.

Інспектор є потужним інструментом для швидкого налагодження і тестування гри, оскільки дозволяє розробникам змінювати параметри компонентів в реальному часі та спостерігати за їхніми змінами під час виконання гри. Він полегшує взаємодію з компонентами та дозволяє швидко здійснювати зміни для досягнення бажаних результатів у розробці гри (Рисунок 3.4).



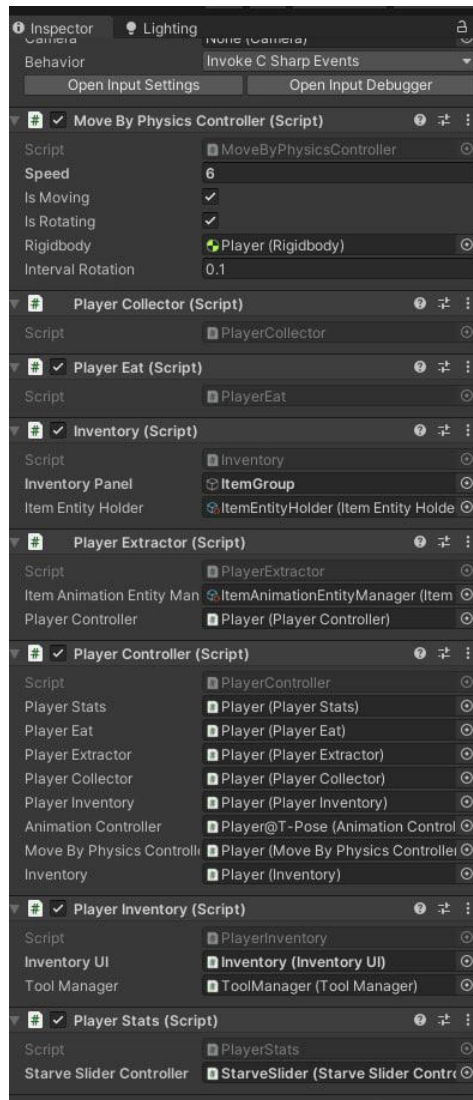


Рисунок 3.4 – Інспектор для огляду компонентів

Ієрархія об'єктів на сцені – це структуроване відображення всіх об'єктів, які присутні на поточній сцені в інтегрованій середовищі розробки Unity.

В ієрархії об'єктів можна побачити список всіх об'єктів на сцені, включаючи головний об'єкт, камери, освітлення, меші, аудіо джерела та інші. Кожен об'єкт може мати дочірні об'єкти, які знаходяться в його підпорядкуванні.

Ієрархія об'єктів надає зручний спосіб керування об'єктами на сцені. Розробники можуть змінювати позицію, поворот, масштаб та інші властивості об'єктів, перетягуючи їх у ієрархії. Також можна змінювати порядок розташування об'єктів, впливаючи на їхнє відображення та взаємодію на сцені.

В даній роботі ієрархії об'єктів реалізована наступним чином (Рисунок 3.5):

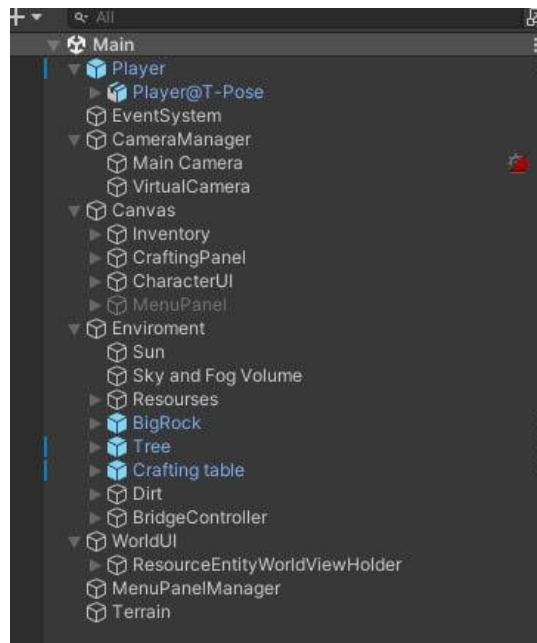


Рисунок 3.5 – Ієрархія об'єктів на сцені

При проектуванні різноманітних додатків, а особливо комп'ютерних ігор, розробнику важливо контролювати процес виявлення будь яких помилок. Для цього слугує вікно консолі.

У вікні консолі можна побачити повідомлення про роботу скриптів, помилки в коді, інформацію про завантаження ресурсів, статуси підключених модулів та іншу корисну інформацію. Консоль також може відображати результати виконання спеціальних функцій або команд, які виконуються в процесі розробки (Рисунок 3.6).

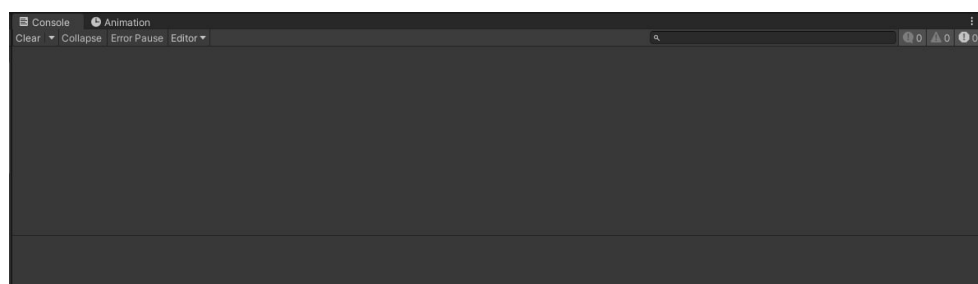


Рисунок 3.6 – Вікно консолі

### **3.3. Аналіз використаних для розробки проекту інструментів Unity**

Unity – це потужний ігровий рушій, який використовується для розробки комп'ютерних ігор та інтерактивних додатків. Його широко використовують у галузі геймдеву, оскільки він надає розробникам багато можливостей і інструментів для створення гри з нуля.

Основні інструменти, що використовувались для розробки проекту:

1. Редактор сцен;
2. Компонентна система;
3. Скриптування;
4. Графічний редактор;
5. Фізична симуляція;
6. Анімація;
7. Тестування та налагодження.

Ці інструменти Unity допомогли ефективно створювати, тестувати та розгортати комп'ютерні ігри з використанням рушія Unity в рамках даного дипломного проекту. Вони забезпечують широкий спектр функціональності і дозволяють реалізувати різноманітні ідеї та концепції гри.

### **3.4. Написання коду**

Під час розробки гри написання коду відіграє важливу роль у реалізації функціональності та поведінки гри.

Код відповідає за обробку подій, управління персонажем, взаємодію з об'єктами, реалізацію логіки гри та багато іншого. Використання мови програмування C# у поєднанні з Unity API дозволило ефективно створити та керувати різноманітними аспектами гри.

Написання коду є ключовим етапом у розробці гри, оскільки він дозволяє перетворити концепцію гри на функціонуючий продукт з багатою функціональністю та захоплюючим геймплеєм.

Розглянемо більш детально основні елементи коду, які реалізують:

1. За відкриття нової локації у разі виконання завдань – в нашому випадку мосту, відповідає клас BridgeController (Рисунок 3.7);

```
using System;
using ReceiverObjects;
using UnityEngine;

namespace Environment.ObjectControllers
{
    public class BridgeController : MonoBehaviour
    {
        [SerializeField] private ReceiverBase _receiver;
        [SerializeField] private GameObject _destroyedBridge;
        [SerializeField] private GameObject _fixedBridge;

        private void OnEnable()
        {
            _receiver.OnReceived += FixBridge;
        }

        private void OnDisable()
        {
            _receiver.OnReceived -= FixBridge;
        }

        private void FixBridge(bool isFixed)
        {
            if (!isFixed)
            {
                return;
            }

            _destroyedBridge.SetActive(false);
            _fixedBridge.SetActive(true);
        }
    }
}
```

Рисунок 3.7 – Реалізація класу BridgeController

2. За реалізацію збільшення рівня ситості у разі прийому їжі персонажем відповідає клас PlayerEat (Рисунок 3.8);

```

using ...

namespace Player
{
    public class PlayerEat : MonoBehaviour
    {
        public event Action<float> OnRecoveryStarve;

        private Inventory _inventory;

        public void Init(Inventory inventory)
        {
            _inventory = inventory;
        }

        private void OnEnable()
        {
            _inventory.OnUseItem += TryToEat;
        }

        private void OnDisable()
        {
            _inventory.OnUseItem -= TryToEat;
        }

        private void TryToEat(Item item)
        {
            EatableItem eatableItem = item as EatableItem;

            if (eatableItem == null)
            {
                return;
            }

            OnRecoveryStarve?.Invoke(eatableItem.StarveRecoveryPoints);
            _inventory.RemoveItem(item);
        }
    }
}

```

Рисунок 3.8 – Реалізація класу PlayerEat

3. Основний клас PlayerController, що об'єднує в собі інші класи, основні методи, що забезпечують рух, наповнення інвентарю, добування ресурсів, а також ініціалізацію самої гри (Рисунок 3.9-3.11).

```

using System.Collections.Generic;
using Extractable;
using PlayerInventory;
using PlayerInventory.ItemFolder;
using UnityEngine;
using UnityEngine.SceneManagement;

namespace Player
{
    1 asset usage 2 usages
    public class PlayerController : MonoBehaviour
    {
        [SerializeField] private PlayerStats _playerStats; 1 Player (PlayerStats)
        [SerializeField] private PlayerEat _playerEat; 1 Player (PlayerEat)
        [SerializeField] private PlayerExtractor _playerExtractor; 1 Player (PlayerExtractor)
        [SerializeField] private PlayerCollector _playerCollector; 1 Player (PlayerCollector)
        [SerializeField] private PlayerInventory _playerInventory; 1 Player (PlayerInventory) More...
        [SerializeField] private AnimationController _animationController; 1 Changed in 1 asset
        [SerializeField] private MoveByPhysicsController _moveByPhysicsController; 1 Player (MoveByPhysicsController)
        [SerializeField] private Inventory _inventory; 1 Player (Inventory)

        1 usage
        public AnimationController AnimationController => _animationController;
        1 usage
        public Inventory Inventory => _inventory;

        Event function
        private void Awake()
        {
            Init();
        }

        Event function
        private void OnEnable()
        {
            _playerEat.OnRecoveryStarve += RecoveryStarve;
            _playerStats.OnDied += OnDied;
            _playerExtractor.OnExtracted += FillInventory;
            _moveByPhysicsController.OnPlayerMove += SetActiveMove;
            _animationController.OnMiningHit += _playerExtractor.HitExtract;
        }
    }
}

```

Рисунок 3.9– Реалізація класу PlayerController

```

private void OnDisable()
{
    _playerEat.OnRecoveryStarve -= RecoveryStarve;
    _playerStats.OnDied -= OnDied;
    _playerExtractor.OnExtracted -= FillInventory;
    _moveByPhysicsController.OnPlayerMove -= SetActiveMove;
    _animationController.OnMiningHit -= _playerExtractor.HitExtract;
}

2 usages
public bool CheckEquippedInstrument(IExtractable extractable)
{
    return _playerInventory.CurrentInstrument is not null &&
        extractable.Tool.Name == _playerInventory.CurrentInstrument.Name;
}

2 usages
private void FillInventory(List<Item> items)
{
    foreach (Item item in items)
    {
        _playerCollector.PickUp(item);
    }
}

Frequently called 2 usages
private void SetActiveMove(bool isActive)
{
    _animationController.SetMove(isActive);
}

Frequently called 2 usages
private void OnDied()
{
    SceneManager.LoadScene(0);
}
}

```

Рисунок 3.10 – Продовження коду класу PlayerController

```

Frequently called 2 usages
private void OnDied()
{
    SceneManager.LoadScene(0);
}

2 usages
private void RecoveryStarve(float recoveryValue)
{
    _playerStats.RecoveryStarve(recoveryValue);
}

1 usage
private void Init()
{
    _playerCollector.Init(_inventory);
    _playerEat.Init(_inventory);
    _playerInventory.Init(_inventory);
}
}

```

Рисунок 3.11 – Продовження коду класу PlayerController

### 3.5. Створення графічних елементів

Також для реалізації повноцінного додатку створеної гри необхідно реалізувати головне меню, та UI елементи керування грою під час її проходження.

Ця сцена буде зустрічати гравця при запуску гри. Отже воно має задачу зрозуміло піднести інформацію того, як почати гру або вийти з додатка (Рисунок 3.12).



Рисунок 3.12 – Головне меню гри

Отже це і буде реалізовано в сцені головно меню. Для покращення візуальної складової було використано безкоштовні UI елементи з AssetStore.

Реалізація головного меню являє собою створення кнопок та написання коду, за що ці кнопки відповідають. Отже необхідно створити окремий файл з кодом, де буде прописані дві задачі : Відкриття сцени з самою грою для кнопки «Start», закриття всього додатку для кнопки «Quit».

Також було реалізовано меню виходу з гри, інвентарю та меню крафту, яке було представлено на процесі створення в попередніх розділах (Рисунки 3.13, 3,14)



Рисунок 3.13 – Меню виходу з гри



Рисунок 3.14 – Меню інвентарю



Так як основна ідея гри полягає в пошуку провізії та відкритті нових територій, то було важливим створення завдань для персонажу, при виконанні яких він відкриватиме нову локацію. Приклад такого завдання був представлений у попередніх розділах на рисунку 2.5, а в результаті це виглядає наступним чином (Рисунок 3.15):

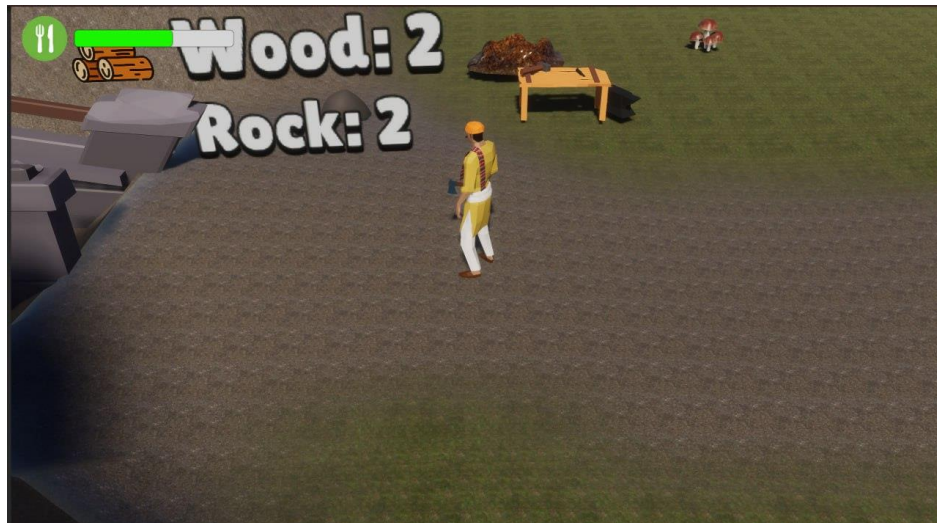


Рисунок 3.15 – Завдання для відкриття нової локації

Відкриття нової локації після виконання завдань представлено на рисунку 3.16:



Рисунок 3.16 – Відкриття нової локації

Як було сказано раніше, у грі «Survival Island» передбачено анімацію добування персонажем ресурсів. Наприклад, на рисунку 3.17 можна побачити процес рубання дерева.



Рисунок 3.17 – Анімація рубання дерева

Варто зазначити, що на всіх рисунках в роботі видно, як змінюється рівень ситості персонажу. З кожним разом цей рівень зменшується все швидше задля мотивації гравця шукати провізію та відкривати нові горизонти.

За бажанням, гравець може зберегти прогрес і повернутись до гри в будь який зручний для нього час.

## Висновок до третього розділу

Отже в цьому розділі ми розглянули процес розробки програмного продукту та реалізацію основних задач проекту і особливостей гри у жанрі виживання, використовуючи для цього ігровий рушій Unity, а також мову програмування C#.

Реалізація проекту була розпочата з прототипування візуальної частини гри.

Описано алгоритм роботи додатку та візуальних компонентів гри.

Створено зручний інтерфейс для відображення ігрової частини проекту.

Також було розроблено головне меню гри та внутрішньо-ігрове меню

За результатами цього розділу основним функціоналом розробленого прототипу гри в жанрі виживання можна вважати:

1. Головне меню гри з рядом функцій;
2. Зрозумілий та зручний інтерфейс;
3. Чіткі умови для проходження гри;
4. Внутрішньо-ігрове меню;
5. Відображення необхідних сцен;
6. Реалізація лінійного сюжету.

## ВИСНОВКИ

Результатом виконаної дипломної роботи є комп'ютерна гра в жанрі виживання. У роботі описується покрокове створення програмного продукту цього типу починаючи з аналізу поставленої задачі та завершуючи створенням гри.

1. Проаналізовано ринок комп'ютерних ігор і встановлено основні характеристики майбутнього продукту за вимогами жанру. Для проектування власної гри було проведено аналіз існуючих ігор в даному жанрі та виділено необхідні риси майбутнього продукту. Розглянуто його актуальність.

2. Розглянуто основні ресурси для розробки комп'ютерної гри в обраному жанрі. Проаналізовано та обрано для реалізації такі інструменти як ігровий рушій Unity та середовище розробки JetBrains Rider.

3. Розглянутий алгоритм створення комп'ютерних ігор в жанрі виживання.

4. За результатами спроектовано та розроблено мобільну гру "Survival Island" на основі аналізу потреб користувачів.

5. Додано суміжні механіки для покращення user experience, такі як система збереження, налаштування UI, покращення управління.

6. Виконано внутрішнє тестування гри та підготовка до публікації.

Отже результатом проведеної роботи є прототип гри в жанрі виживання, що має всі ознаки свого жанру, головну мету, умови проходження і зручний інтерфейс для користування та відповідає поставленим вимогам. Прототип може бути довершений іншими деталями та функціями і офіційно виданий.

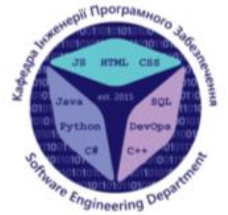
## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ЕНЦИКЛОПЕДІЯ СУЧАСНОЇ УКРАЇНИ. Комп'ютерна гра [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: [https://esu.com.ua/search\\_articles.php?id=4393](https://esu.com.ua/search_articles.php?id=4393)
2. Класифікація ігор. Класифікація за жанрами [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://sites.google.com/site/wwwbossfighttvua/klasifikacia-igor/klasifikacia-za-zanrami>
3. Офіційний сайт гри «Rogue» [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://www.roguecompany.com/>
4. Офіційний сайт гри «Graveyard Keeper» [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://www.graveyardkeeper.com/>
5. Ігровий рушій [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: [https://uk.wikipedia.org/wiki/%D0%86%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9\\_%D1%80%D1%83%D1%88%D1%96%D0%B9](https://uk.wikipedia.org/wiki/%D0%86%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D0%B9_%D1%80%D1%83%D1%88%D1%96%D0%B9)
6. Офіційний сайт рушія Unity [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://docs.unity3d.com>
7. Офіційний сайт розширення для Unity Fungus [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://fungusgames.com>
8. Документація до ігрового рушія Unity [Електронний ресурс]: [Веб-сайт]. – електронні дані. – Режим доступу: <https://docs.unity3d.com/2019.4/Documentation/Manual/index.html>

## ДОДАТОК А



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



### Розробка гри «Survival Island» з використанням ігрового рушія Unity мовою C#

Виконав студент 4 курсу  
Групи ПД-41  
Павліченко Вадим Юрійович  
Керівник роботи

Доктор філософії, Доцент кафедри ІПЗ, Гребенюк В.В

Київ – 2023

### МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – збільшення зацікавленості потенційних гравців до ігор в жанрі виживання що реалізується за рахунок геймплею, що базується на реальному виживанні в світі.
- **Об'єкт дослідження** – геймплей, що базується на реальному виживанні в світі.
- **Предмет дослідження** – технології розробки комп'ютерної гри за допомогою ігрового рушія Unity.

## ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Проаналізувати ринок комп'ютерних ігор і визначити основні характеристики майбутнього продукту за вимогами жанру
2. Розглянути основні ресурси для розробки комп'ютерної гри в обраному жанрі
3. Ознайомитись з аналогами комп'ютерних ігор в жанрі виживання
4. Розробити вимоги до комп'ютерної гри на основі аналізу переваг та недоліків аналогів
5. Розробити і створити додаток комп'ютерної гри в жанрі виживання

3

## АНАЛІЗ АНАЛОГІВ

Властивості	Rogue	Journey	Graveyard Keeper	Survival Island
Платформи	PC	PS4, PS3	PC	PC
Візуальний стиль, що вирізняє серед інших	-	+	+	+
Управління однією рукою	+	+	-	+
Реіграбельність	-	+	-	+
Система крафту	-	-	+	+
Наявність ворогів	+	-	+	-
Наявність місій	+	+	+	+

4

## ВИМОГИ ДО ІГРОВОГО КОНТЕНТУ

1. Різноманітність ігрових механік задля покращення реіграбельності.
2. Зміна сцен відповідно до сюжету.
3. Система крафту задля зацікавлення гравця залишитись в світі та покращуватись.
4. Відображення всіх візуальних та текстових частин гри.
5. Рандомна генерація ресурсів в світі, щоб при новій спробі змінити досвід гравця.
6. Керування, що дозволяє грати в гру однією рукою.

5

## КОНЦЕПТ ГРИ

1. Гравець опиняється на безлюдному острові.
2. Головна мета гравця - шукати їжу для виживання.
3. Гравець може покращувати своє середовище існування шляхом пошуку ресурсів для відкриття нових локацій.
4. Гравець має змогу займатись крафтом для того, щоб відкрити нові знаряддя щоб добувати нові види ресурсів.
5. Рівень ситості з часом зменшується швидше, що стимулює гравця на розвиток.
6. Гра продовжується, поки рівень ситості гравця не досягне нуля.

6

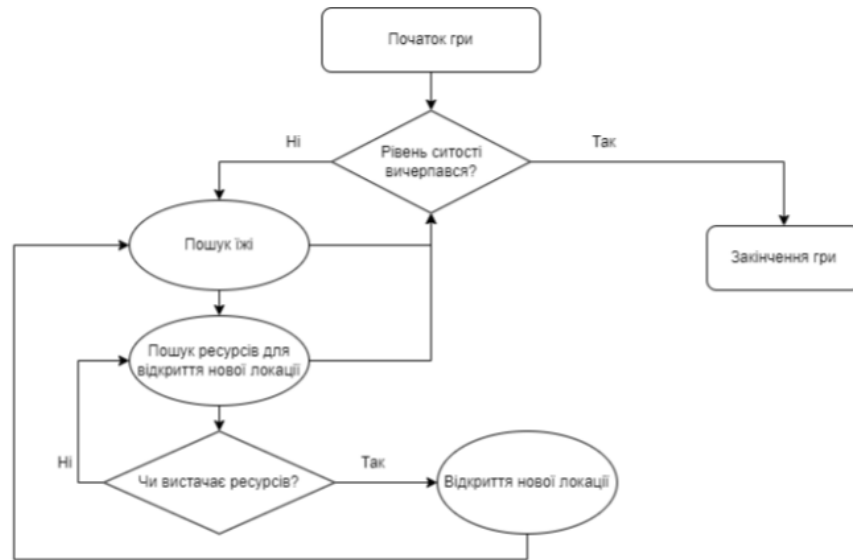


### ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



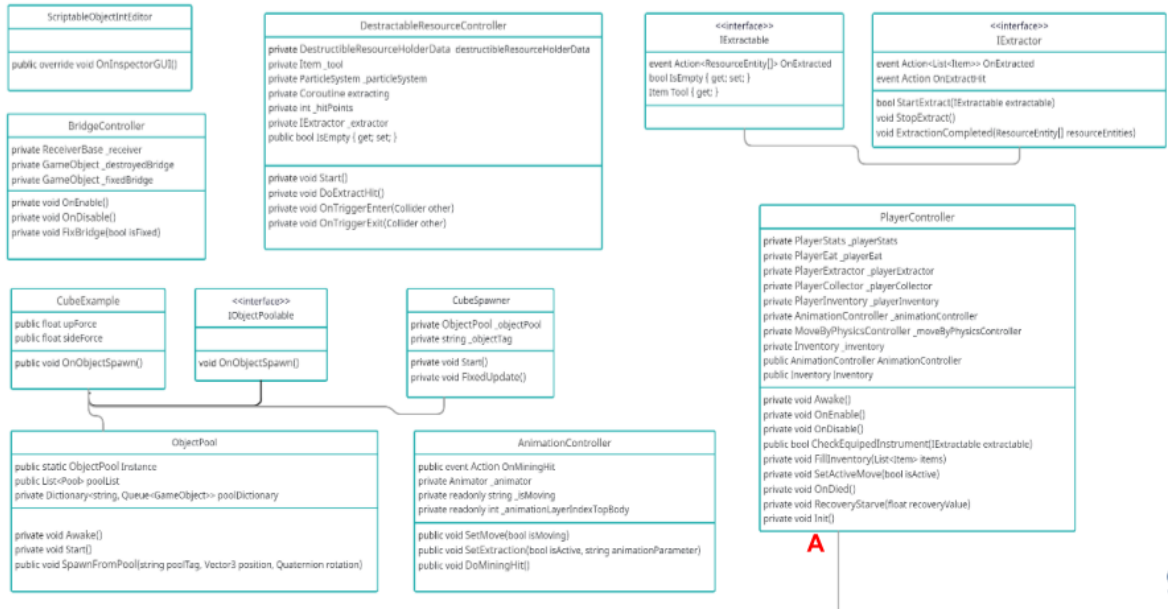
7

### БЛОК-СХЕМА ГРИ

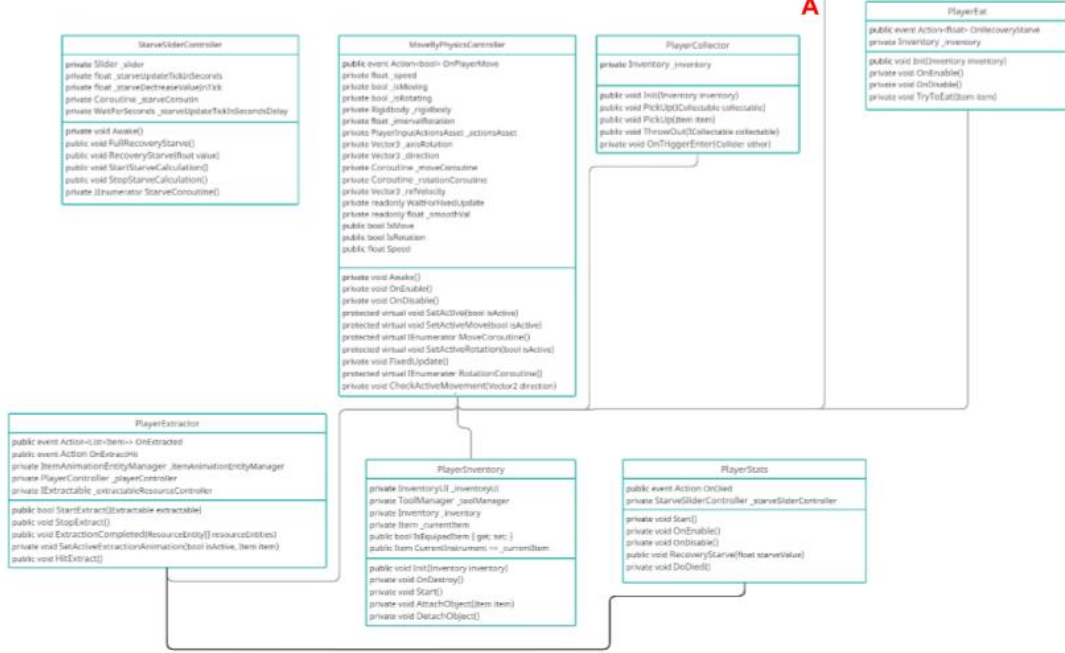


8

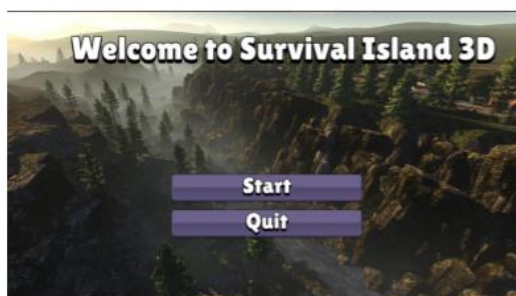
# ДІАГРАМА КЛАСІВ



# ДІАГРАМА КЛАСІВ



## ІНТЕРФЕЙС ГРИ



Головне меню гри



Меню виходу з гри

11

## ІНТЕРФЕЙС ГРИ



Меню інвентарю



Завдання для відкриття нової локації

12

## ВИСНОВКИ

1. Проаналізовано ринок комп'ютерних ігор і визначені основні характеристики майбутнього продукту за вимогами жанру.
2. Проаналізовані технічні засоби, що були використані для розробки даної комп'ютерної гри.
3. Проаналізовані існуючі ігри, що є аналогами в жанрі виживання та знайдено їх переваги та недоліки.
4. Розроблено вимоги до гри на основі аналізу переваг та недоліків існуючих аналогів жанру виживання.
5. Спроектовано та розроблено мобільну гру "Survival Island" на основі аналізу потреб користувачів.
6. Додано суміжні механіки для покращення user experience.
7. Виконано внутрішнє тестування гри та підготовка до публікації.

13

ДЯКУЮ ЗА УВАГУ!