

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

# Дипломний проект

здобувача освіти денної форми навчання

РП.06.04.000.ДП

**ВОЛОХ МАКСИМ  
АНДРІЙОВИЧ**

м. Одеса  
2023 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

до дипломного проекту (роботи) на тему:

**Модифікація ігрових елементів оточення та персонажів для комп'ютерної гри «Террарія»**

Проектний матеріал складається з пояснювальної записки на 73 сторінках та графічного (презентаційного) матеріалу на 15 аркушах (слайдах).

Дипломник Волох (Волох М.А.)

Керівник Джабраїлов (Джабраїлов Д.В.)

**Консультанти:**

з економічної частини Копайгородська (Копайгородська Т.Г.)

з охорони праці Чорновол (Чорновол Н.І.)

з дотримання вимог ЄСКД Петрашова (Петрашова В.І.)

старший консультант Кунуп (Кунуп Т.В.)

**До захисту допущений**

Голова циклової комісії Кривченко (Кривченко Ю.В.)

Завідувач відділення Скорнякова (Скорнякова О.В.)

Захист «22» 06 2023 р.

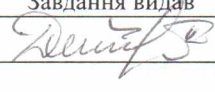
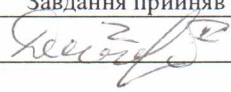
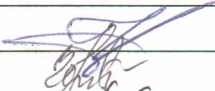



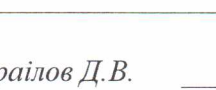
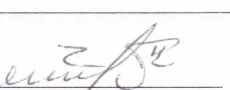
Протокол ДКК № 1

Оцінка ДКК 5 (відмінно)

Секретар ДКК Скорнякова



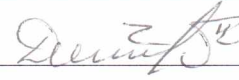
6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1. Технологічний розділ	Джабраїлов Д.В.		
2. Екон. частина	Копайгородська Т.Г.		
3. Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		


7. Дата видачі завдання 24.04.2023

Керівник

Джабраїлов Д.В.

  
(підпис)

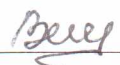
Завдання прийняв до виконання

  
(підпис)

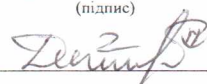
#### КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1.	Вступ. Постановка мети та задач проектування	19.05.2023	Виконав
2.	Опис предметної області	20.05.2023	Виконав
3.	Обґрунтування необхідності виконання модифікації гри «Террарія»	21.05.2023	Виконав
4.	Загальний технічний опис ігрових елементів оточення та персонажів	22.05.2023	Виконав
5.	Способи модифікації та пов'язані із цим технології та інструменти	23.05.2023	Виконав
6.	Проектування модифікації	24.05.2023	Виконав
7.	Розробка модифікації ігрових елементів оточення	25.05.2023	Виконав
8.	Розробка модифікації персонажів	28.05.2023	Виконав
9.	Тестування розробленої модифікації	31.06.2023	Виконав
10.	Виконання Економічної частини	02.06.2023	Виконав
11.	Виконання Охорони праці	04.06.2023	Виконав
12.	Підготовка матеріалів до захисту	06.06.2023	Виконав
13.	Попередній малий захист	12.06.2023	Виконав
14.	Проведення захисту дипломного проекту	19.06.2023	Виконав

Дипломник

  
(підпис)

Керівник

  
(підпис)



# ЗМІСТ

Вступ.....	5
1 Технологічний розділ.....	6
1.1 Опис предметної області.....	6
1.2 Обґрунтування необхідності виконання модифікації гри .....	9
1.3 Загальний технічний опис ігрових елементів оточення та персонажів .....	12
1.4 Способи модифікації, пов'язані із цим технології та інструментарій.....	18
1.5 Проектування модифікації.....	25
1.6 Модифікація ігрових елементів оточення .....	32
1.7 Модифікація нових персонажів.....	37
2 Економічна частина.. ..	47
2.1 Резюме.....	47
2.2 Визначення трудомісткості розробки програмного забезпечення .....	47
2.3 Розрахунок ціни програмного продукту .....	50
3. Охорона праці.....	52
1 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці даного програмного комплексу .....	52
2 Гігієнічні вимоги до виробничого середовища.....	52
2.1 Вимоги до приміщення .....	52
2.2 Освітлення .....	53
2.3 Шум.....	53
2.4 Вимоги до організації робочого місця працівника.....	53
2.5 Мікроклімат.....	54
2.6 Електробезпека.....	55
3 Пожежна безпека.....	56
Перелік використаних джерел.....	58
ДОДАТОК А. ЛІСТИНГ ПРОГРАМНОГО МОДУЛЯ.....	59
ДОДАТОК Б. ЛІСТИНГ ГРАФІЧНОГО МАТЕРІАЛУ.....	68

					<b>РП 06. 04 000. ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

## ВСТУП

Створення модифікацій для “Terraria” є дуже актуальним і відіграє значну роль у спільноті гри та загальному досвіді. Модифікації дозволяють гравцям розширювати та налаштовувати “Terraria” за межі оригінального вмісту, відкриваючи світ можливостей.

Моди можуть запроваджувати нові ігрові механізми, функції та завдання, покращуючи загальний ігровий процес. Вони можуть представляти нові локації, ворогів, головних ворогів, предмети, зброю, броню та аксесуари, розширюючи доступний вміст і надаючи гравцям нові завдання та нагороди. Модифікації можуть запроваджувати нові режими гри, покращувати багатокористувацьку гру або покращувати якість життя, роблячи гру приємнішою та доступнішою для більшого кола гравців.

Моди дозволяють гравцям налаштовувати “Terraria” відповідно до своїх уподобань, підлаштовуючи гру відповідно до свого стилю гри чи бажаного досвіду. Гравці можуть змінювати або створювати нові предмети, візуальні елементи, музику чи звукові ефекти, що дозволяє їм створювати унікальні теми, естетику чи атмосферу у своїх світах. Моди надають можливість створювати власних NPC, завдання, діалоги та сюжетні лінії, дозволяючи гравцям формувати розповідь і занурюватися у свої власні пригоди.

Модифікації подовжують довговічність “Terraria”, надаючи новий вміст і виклики, дозволяючи гравцям продовжувати досліджувати гру та насолоджуватися нею навіть після завершення вмісту базової гри. Моди пропонують безмежні можливості для відтворення, оскільки кожен мод може представити унікальний досвід і створити нову пригоду для гравців.

Загалом, створення модифікацій для “Terraria” додає величезну цінність грі та спільноті. Моди приносять новий вміст, параметри налаштування та ігровий процес, що робить “Terraria” більш динамічною та персоналізованою пригодою. Вони сприяють залученню спільноти, співпраці та інноваціям, гарантуючи, що гра продовжує розвиватися ще довго після її першого випуску.

					<i>РП 06. 04 000. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

# 1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

## 1.1 Опис предметної області

Предметна область включає в себе такі аспекти гри, з якими розробник буде працювати при створенні модифікації.

Предмети є одним з найважливіших аспектів гри “Terraria” (рисунок 1.1). Вони можуть бути зброєю, бронєю, інструментами, аксесуарами, матеріалами для створення предметів та багатьма іншими речами. При створенні модифікації можна додавати нові предмети, змінювати існуючі або надавати їм нові властивості, функціональність та унікальні ефекти. Можна створювати предмети з унікальними атаками, особливими навичками чи здібностями, різними рівнями магії або витривалості, а також з різними модифікаторами для покращення їхніх характеристик. Також можна додавати нові матеріали для створення предметів, що розширюють можливості гравців у створенні нових предметів.



Рисунок 1.1. Будівництво у грі “Terraria”

Неігровий персонаж - це персонажі, які зустрічаються в грі “Terraria”. Вони можуть бути торговцями, союзниками, ворогами або жителями селища (рисунок 1.2). Можна створювати нових неігрових персонажів, змінювати

					РП 06. 04 001. ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

їхню поведінку, розміщення та взаємодію з гравцями за допомогою модифікацій. Є можливим додавати нові типи неігрових персонажів з унікальними функціями та рольовими характеристиками. Наприклад, можливо створити NPC-тренера, який надає гравцям нові навички та бонуси, NPC-квестгівера, який дає завдання гравцям або NPC-торговця, який продає рідкісні предмети. Також можливо змінювати взаємодію неігрових персонажів з гравцями, додавати нові діалоги, взаємодіяти з іншими неігровими персонажами та створювати складні системи взаємодії між неігровими персонажами та гравцями.



Рисунок 1.2. Приклад союзних персонажів гри “Terraria”

Гра “Terraria” має різні механіки, такі як будівництво, бойова система, система створення предметів, розвиток персонажа, розсування блоків, економіка та багато інших. Можливо розширити та змінювати ці механіки, додавати нові функції, елементи ігрового процесу або поліпшувати існуючі механіки з допомогою модифікацій. Наприклад, можна додати нові типи блоків або розширити функціональність існуючих блоків. Можна вдосконалити бойову систему, додавши нові види зброї, заклинань або способів атаки. Також можливо впливати на розвиток персонажа, додавши

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

нові навички, класи або системи прокачки. Крім того, можна створювати нові види ворогів, босів або подій, що збагачують ігровий процес та надають гравцям нові виклики. Розробник має велику свободу в модифікації механік гри, що дозволяє створювати унікальні ігрові досвіди та надихати гравців на нові пригоди.

Можна працювати з графічними елементами гри, такими як моделі, тайли, ефекти анімації та звуковими ресурсами, щоб створювати нові вигляди, анімації та звукові ефекти для предметів, персонажів або подій. Можна створювати нові моделі та анімації для предметів, щоб надати їм унікальний вигляд або покращити їхню візуальну привабливість. Також можна змінювати моделі персонажів, дозволяючи гравцям створювати власний унікальний вигляд для своїх персонажів. Крім того, можна додавати нові звукові ефекти, звукові доріжки або змінювати існуючі звуки, щоб створити атмосферу та настрій, відповідні вашій модифікації. У розробника є можливість творчо використовувати графічні елементи та звукові ресурси для візуального та звукового поліпшення модифікації, що дозволяє створювати унікальні та захоплюючі враження для гравців.

“Terraria” підтримує мережеву гру, і можна створювати модифікації, які впливають на кооперативний режим, дозволяють гравцям співпрацювати, змагатися або взаємодіяти один з одним у спеціальних умовах (малюнок 1.3). Можна створювати модифікації, які додають нові мережеві функції, такі як нові режими гри, розширені можливості спільного будівництва, розбудови селищ та військових баз. Можна створювати командні режими, де гравці об'єднуються в команди та змагаються один з одним у спеціальних завданнях або битвах. Також можна додавати нові мережеві події та взаємодії, що дозволяють гравцям спілкуватися, торгувати, обмінюватися ресурсами та створювати спільні стратегії. Мережева взаємодія відкриває широкі можливості для творчості та соціальної взаємодії між гравцями в межах модифікації.

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

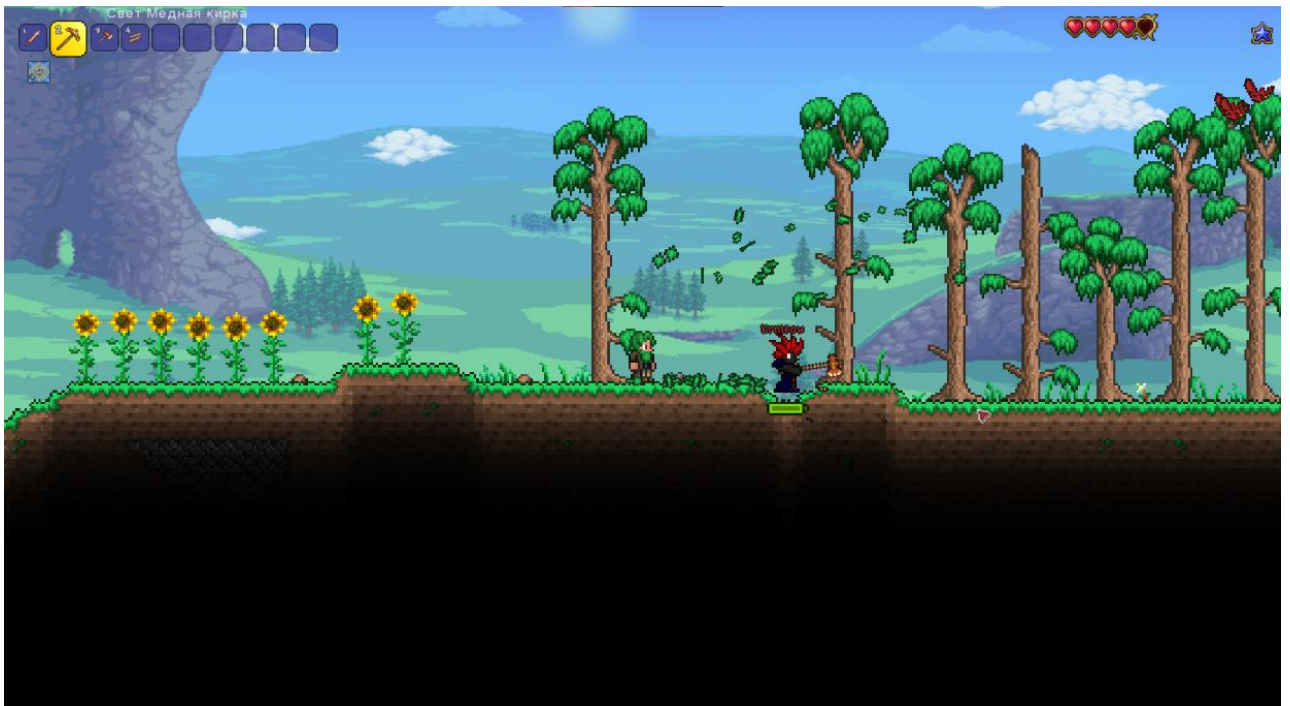


Рисунок 1.3. Приклад мережевої гри у грі “Terraria”

Це основні аспекти предметної області при створенні модифікацій для “Terraria”. Конкретні можливості та обмеження будуть залежати від інструментів розробки та API, які використовуються для створення модифікацій.

## 1.2 Обґрунтування необхідності виконання модифікації гри “Terraria”

Ігрові модифікації (або просто модифікації, скорочено "моди") - це зміни, які вносяться до існуючої комп'ютерної гри з метою модифікування її ігрового процесу, вигляду, функціональності та інших аспектів. Модифікації розширюють можливості гри та надають гравцям нові функції, предмети, персонажів, локації, механіки гри та інші елементи.

Зазвичай модифікації створюються громадськими розробниками (фанатами гри або програмістами), які використовують доступні інструменти розробки та API (інтерфейси програмування додатків), надані розробниками гри, для створення своїх власних змін. Ці модифікації потім можуть бути розповсюджені іншим гравцям для встановлення і використання у своїй копії гри.

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

Розширення ігрового процесу: Модифікації дозволяють розширити можливості гри та внести нові елементи, які роблять гру більш різноманітною і цікавою. Вони додають нові предмети з унікальними властивостями, що дають гравцям більше варіантів для експериментів та стратегій. Модифікації можуть також включати нових монстрів з різними атаками та поведінкою, що створює більш цікаві битви та непередбачувані ситуації.

Крім того, вони можуть додавати нові локації з унікальними виглядами та ресурсами, які стимулюють дослідження та пригоди гравців. Модифікації також можуть впроваджувати нові механіки гри, такі як розширена механіка створення предметів, економічна система, розвиток персонажа або нові режими гри. Ці елементи роблять ігровий процес більш захоплюючим, непередбачуваним і дають можливість гравцям зануритися в унікальний світ модифікації.

Творча свобода: Створення модифікацій дає можливість гравцям виразити свою творчість та створити щось унікальне. Вони можуть розробляти нові світи з власними землями, локаціями та географією, що дає непередбачувані пригоди та відкриття. Гравці також можуть створювати унікальні історії та завдання, які запрошують гравців на захоплюючі подорожі та випробування. Вони можуть творити нових персонажів з власними характерами, зовнішнім виглядом та унікальними навичками, що додає багато глибини інтеракції та розвитку.

Крім того, гравці можуть розробляти нові механіки гри, включаючи унікальні системи створення предметів, інтерактивні елементи гри та спеціальні функції, що надають унікальний характер їх модифікації. Таким чином, модифікації дають гравцям можливість створити гру, яка відповідає їх власним бажанням та фантазії, розкрити свою творчість і поділитися своїми ідеями з іншими гравцями.

Спільнота та обмін контентом: Створення модифікацій стимулює активне спілкування та обмін ідеями серед ігрової спільноти. Гравці можуть

					<b>РП 06. 04 001. ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

не лише створювати свої власні модифікації, але й ділитися ними з іншими гравцями. Це відкриває можливість для взаємодії з іншими розробниками модифікацій, обмінюватися досвідом та співпрацювати над проектами. Також, гравці можуть надавати відгуки один одному, що сприяє вдосконаленню модифікацій та поліпшенню ігрового процесу. Вони можуть обговорювати свої ідеї, розмірковувати над можливими вдосконаленнями та спільно працювати над новими проектами. Цей обмін інформацією та співпраця створюють динамічну та захоплюючу спільноту, яка підтримує гру “Terraria” та забезпечує постійний потік нового контенту для гравців. Кожна модифікація вносить свій унікальний вклад у розширення світу гри, роблячи її більш різноманітною та захоплюючою для всіх гравців.

Надання нових викликів та досвіду: Модифікації додають гравцям нові виклики та завдання, які розширюють можливості гри “Terraria” та збільшують її тривалість. Ці модифікації можуть включати складніші босів, нові режими гри, розширені системи створення предметів, великі події та інші захоплюючі елементи. Виконуючи ці нові завдання та зустрічаючи виклики, гравці отримують можливість розвиватися та продовжувати експериментувати у грі. Вони можуть вдосконалювати свої навички, виявляти нові стратегії та отримувати цінний досвід, що дозволяє їм стати кращими гравцями. Після завершення основного контенту гри, модифікації дозволяють гравцям продовжувати відчувати захоплення та емоції, надаючи нові враження та виклики. Вони створюють додатковий шар ігрового процесу, роблячи гру більш різноманітною та захоплюючою навіть для досвідчених гравців.

Підтримка та продовження гри: Модифікації відіграють важливу роль у підтримці та продовженні гри “Terraria”, допомагаючи зберігати інтерес гравців та збільшувати активну спільноту. Вони забезпечують постійне оновлення та розвиток гри, надаючи нові можливості та враження гравцям протягом тривалого часу.

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

Розробники модифікацій здатні вносити покращення, виправляти помилки та додавати новий контент до гри. Це дозволяє тримати гру актуальною та зацікавленість гравців, оскільки вони завжди матимуть щось нове для відкриття і дослідження. Розширення та оновлення, внесені модифікаціями, стимулюють гравців повертатися до гри знову і знову, забезпечуючи постійний потік нового контенту та викликів.

Завдяки розширенням, створеним спільнотою розробників, “Terraria” отримує додаткову підтримку та довготривалу життєву циклічність. Гравці можуть насолоджуватися відчуттям, що гра розвивається і продовжує відкривати нові горизонти навіть після випуску. Це робить “Terraria” грою, яка може зберігати свою популярність та активну спільноту протягом тривалого часу.

Загалом, створення модифікацій для гри “Terraria” важливе для збагачення ігрового процесу, залучення творчих талантів, співпраці у спільноті гравців та продовження життя гри. Вони створюють нові можливості та випробування для гравців, що робить їх важливою частиною екосистеми гри “Terraria”.

### **1.3 Загальний технічний опис ігрових елементів оточення та персонажів**

Блоки (Blocks): Гра базується на піксельній графіці, де світ складається з різних матеріалів, представлених у вигляді блоків. Ці блоки можуть бути взаємодійними з гравцем і середовищем. У грі присутні різноманітні типи блоків, такі як земля, каміння, дерево і інші, які можуть бути зруйновані, розміщені, переміщені та використовувані для створення будівель, структур і об'єктів у світі гри. Блоки є основною будівельною одиницею гри, яка дозволяє гравцеві інтерактивно взаємодіяти з оточуючим середовищем та втілювати свої творчі ідеї у віртуальному просторі. Вони відіграють ключову роль у створенні іммерсивного ігрового досвіду та відкривають безліч можливостей для експлорації, будівництва та взаємодії з ігровим світом. Приклад на рисунку 1.4:

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12



Рисунок 1.4. Приклад блоків у грі

Фонові шари (Background Layers): Поміж блоків у грі також присутні фонові шари, які використовуються для створення естетичного фону. Ці шари включають різноманітні декоративні елементи, такі як хмари, зірки, дерева та інші. Вони призначені для додання глибини та атмосферності грі, але зазвичай не взаємодіють з гравцем або іншими об'єктами.

Фонові шари створюються для створення візуально привабливого фонового середовища, яке доповнює основну ігровий процесну частину гри. Вони можуть використовуватися для створення різних настроїв і ефектів, допомагаючи підсилити іммерсію гравця в грі. Приклад на рисунку 1.5:



Рисунок 1.5. Приклад фонового шару у грі

Розрахунковий шар (Foreground Layer): Розрахунковий шар у грі містить рухомі об'єкти, такі як рослини, меблі, механізми, вороги та інші інтерактивні елементи. Цей шар відповідає за розміщення та обробку динамічних об'єктів, які можуть взаємодіяти з персонажем гравця та іншими об'єктами у грі.

Розрахунковий шар відтворює рух та поведінку цих об'єктів, включаючи їх взаємодію з фізичним середовищем та з іншими об'єктами у грі. Він грає важливу роль у створенні реалістичної та динамічної ігрової процесної обстановки, де гравець може взаємодіяти з різноманітними об'єктами та виконувати різноманітні дії. Приклад на рисунку 1.6:

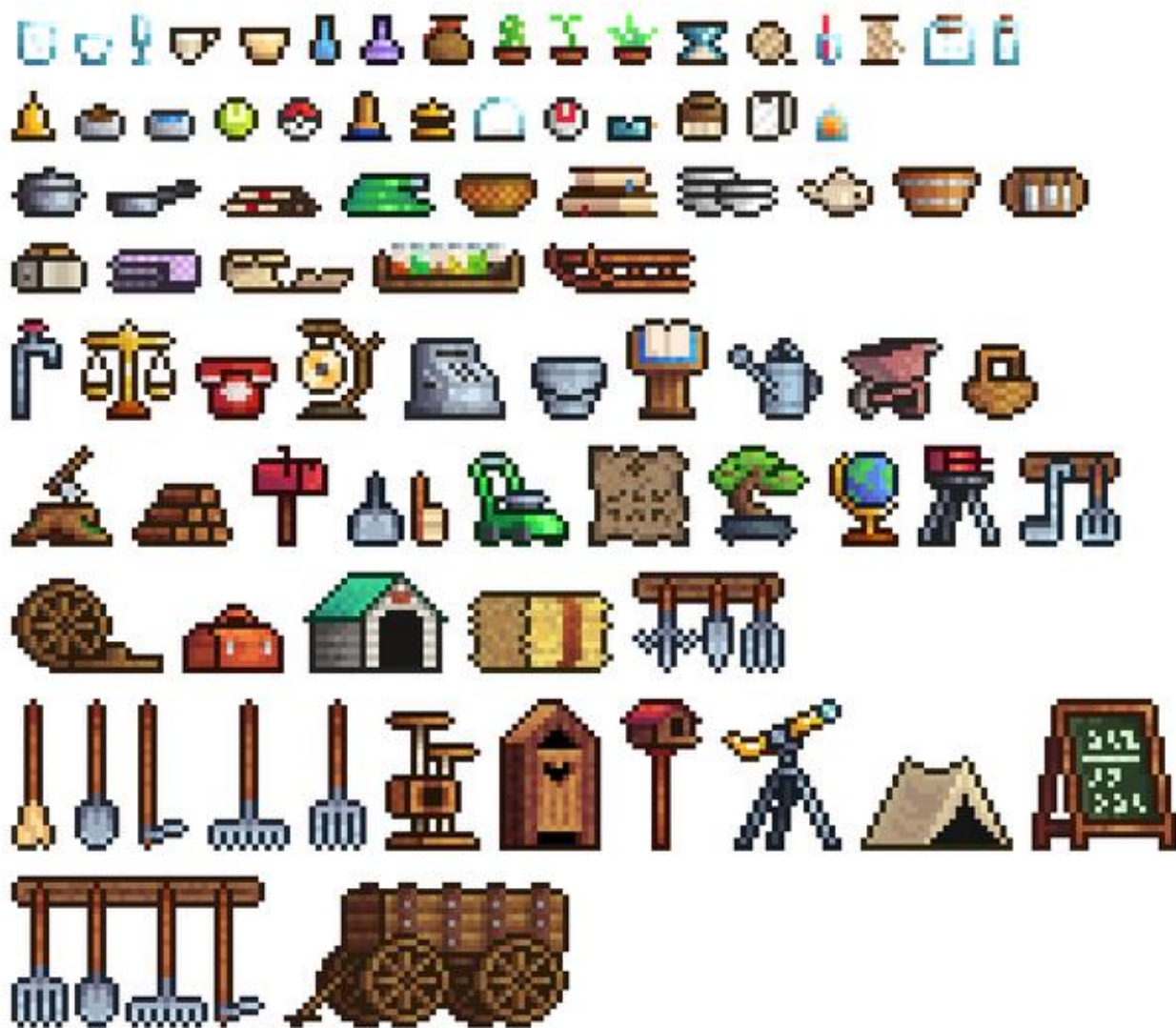


Рисунок 1.6. Приклад інтерактивних елементів

Гравець (Player): Гравець є контрольованим користувачем персонажем, яким керує сам гравець у грі. Він має можливість виконувати різноманітні дії, включаючи рух, стрибки, атаки, збирання ресурсів, взаємодію з об'єктами, створення предметів та виконання завдань. Гравець є основним джерелом взаємодії з грою, він відображається на екрані та слугує посередником між користувачем та ігровим світом. Його дії та рішення впливають на подальший розвиток гри та впровадження різних ігрових можливостей. Гравець забезпечує відчуття контролю та взаємодії з ігровим середовищем, роблячи гру більш цікавою та захоплюючою. Редактор створення гравця на рисунку 1.7:



Рисунок 1.7. Редактор створення гравця

Вороги (Enemies): У грі зустрічаються різноманітні вороги, такі як монстри, зомбі, скелети, механічні боси та інші. Вони володіють можливістю атакувати гравця, рухатися, взаємодіяти з оточенням та мати свої унікальні особливості та атаки. Вороги створюють виклик та небезпеку для гравця, і їх бойові характеристики можуть варіюватися, включаючи здоров'я, шкоду, швидкість, здатності до захисту та спеціальні навички.

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

Боротьба з ворогами може вимагати від гравця стратегічного планування, використання різноманітних зброї, вміння ухилятися та ефективного використання оточення. Успішне подолання ворожих перешкод дозволяє гравцеві рухатися вперед, отримувати нагороди та розкривати нові аспекти ігрового процесу. Приклад на рисунку 1.8:



Рисунок 1.8. Приклад ворогів у грі

NPC (Non-Playable Characters): Неігрові персонажі - це персонажі, якими керує штучний інтелект. Вони виступають у ролі некерованих гравцем персонажів і мають свої власні поведінкові та функціональні характеристики. Неігрові персонажі можуть надавати інформацію гравцю, торгувати предметами, запропонувати завдання, жити в будинках, надавати різні послуги гравцю та виконувати різноманітні функції в грі. Вони можуть мати унікальні діалоги, розмовляти з гравцем, виконувати скриптовані дії та інтеракції з іншими об'єктами та персонажами у грі. Неігрові персонажі додають глибину та життєвість до грального світу, створюючи можливості для інтерактивності та взаємодії гравця з оточуючим світом.

Спутники (Pets): Гравці мають можливість мати спутників, які можуть бути тваринами або фантастичними створіннями, що перебувають поруч з ними. Спутники можуть надавати різні переваги гравцю, такі як додаткову

силу в бою, допомогу у зборі ресурсів або покращення ігрових характеристик гравця. Вони можуть мати свою власну унікальну зовнішність та особистість. Спутники можуть бути використані як джерело розваги та декорації ігрового процесу, додавши естетичну складову та підкреслюючи стиль гравця.

Взаємодія зі спутниками може змінювати ігрові можливості та динаміку гри, забезпечуючи більш глибокий та особистий досвід для гравця.

Приклад на рисунку 1.9:



Рисунок 1.9. Приклад домашніх улюбленців у грі

Головні вороги у грі виступають як потужні та значущі противники з великими розмірами та високою силою. Вони пропонують складні механіки битви, які вимагають стратегічного підходу та розумного використання ресурсів для перемоги. Головні вороги створюють напружений та високовимірний досвід, що спонукає гравця до постійного вдосконалення своїх навичок та здібностей. Перемога над ними має велике значення в прогресі гри, оскільки нагороджує гравця та відкриває шлях до нового рівня складності. Головні вороги створюють незабутні моменти та надають винагороду, що підсилює інтерес гравця до подальшого розвитку гри.

Приклад на рисунку 1.10:

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

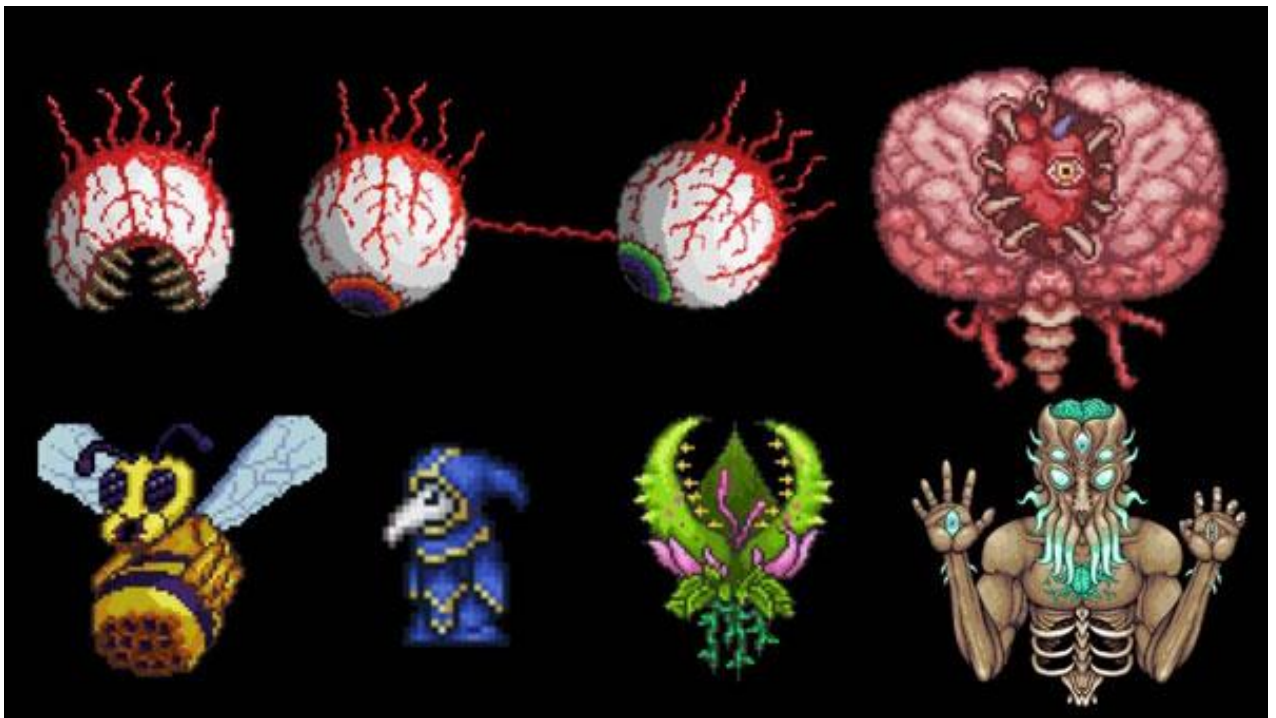


Рисунок 1.10. Приклад головних ворогів у грі

Це лише загальний опис ігрових елементів оточення та персонажів в грі “Terraria”. Кожен з цих елементів може мати багато різних варіацій, особливостей та механік, які додають різноманітність і глибину ігрового процесу.

#### **1.4 Способи модифікації, пов’язані із цим технології та інструментарій**

Модифікація гри Terraria також має свої особливі способи та пов'язана з певним технологічним стеком та інструментарієм.

tModLoader: tModLoader є офіційним інструментом для створення модифікацій у грі Terraria. Він надає зручний набір інструментів та API для розробки модів. Ви можете використовувати tModLoader для створення нових предметів, ворогів, блоків, локацій та видозміни існуючих. Приклад зміни графічних елементів на рисунку 1.11:



Рисунок 1.11. Зміни графічних елементів

tModLoader — популярний фреймворк для модифікації гри “Terraria”. Він служить платформою, яка дозволяє гравцям легко встановлювати, керувати та створювати модифікації для “Terraria”, розширюючи та налаштовуючи вміст і функції гри. tModLoader діє як міст між грою класична “Terraria” та модами. Він надає модифікаційний API (інтерфейс прикладного програмування), який дозволяє розробникам модифікацій створювати власний вміст і бездоганно інтегрувати його в гру. Крім того, tModLoader надає інструменти та утиліти для спрощення процесу модифікації.

Aseprite: Aseprite — це спеціалізований редактор піксельних зображень, призначений для створення спрайтів і анімацій. Він пропонує зручний інтерфейс, розроблений для піксельного мистецтва, з такими функціями, як редагування на рівні пікселів, кадри анімації та керування палітрою. Aseprite надає низку інструментів, спеціально оптимізованих для створення елементів піксельного мистецтва, що робить його популярним вибором серед моддерів “Terraria”.

Ruxel Edit: Ruxel Edit — ще один спеціальний редактор піксельних зображень, який підходить для роботи з графікою “Terraria”. Він забезпечує

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

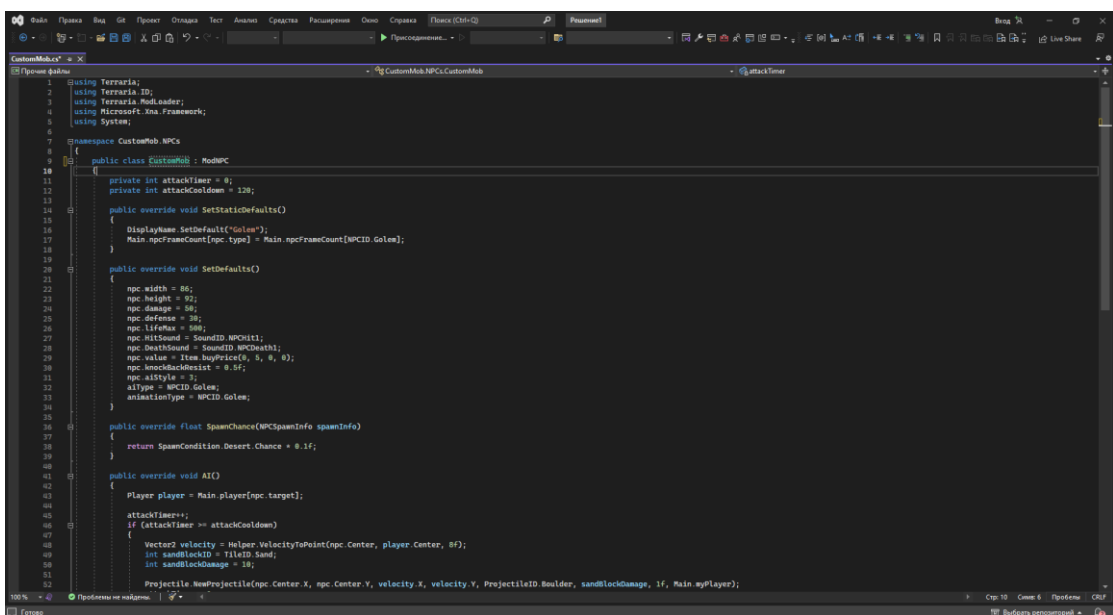
спрощений інтерфейс із такими функціями, як набори фрагментів, кадри анімації та керування шарами. Roxel Edit пропонує різні інструменти малювання та редагування, розроблені спеціально для піксельного мистецтва, що полегшує створення спрайтів і текстур у стилі “Terraria”.

Основною мовою програмування, яка використовується для модифікації “Terraria”, є C#. Сама “Terraria” написана на C#, а платформа модифікації tModLoader побудована на основі кодової бази гри, що дозволяє розробникам розширювати та змінювати її функціональні можливості.

C# — це об’єктно-орієнтована мова програмування загального призначення, розроблена Microsoft. Він широко використовується для різних програм, включаючи розробку ігор. C# відомий своєю читабельністю, простотою та надійністю, що робить його популярним вибором для розробки модифікацій для “Terraria”.

Створюючи модифікації для “Terraria” за допомогою tModLoader, ви пишете код на C#, щоб додавати нові функції, змінювати наявні ігрові механізми, створювати власні предмети, ворогів, неігрових персонажів або події тощо. Ви використовуєте наданий модифікаційний API та хуки для взаємодії з системами та активами гри.

Приклад написання коду на мові C# у Visual Studio на рисунку 1.12:



```
using Terraria;
using Terraria.ID;
using Terraria.ModLoader;
using Microsoft.Xna.Framework;
using System;

namespace CustomMod.NPCS
{
    public class CustomNPC : ModNPC
    {
        private int attackTimer = 0;
        private int attackCooldown = 120;

        public override void SetStaticDefaults()
        {
            DisplayName.SetDefault("Golem");
            Main.npcFrameCount[npc.type] = Main.npcFrameCount[NPCID.Golem];
        }

        public override void SetDefaults()
        {
            npc.width = 88;
            npc.height = 92;
            npc.damage = 88;
            npc.defense = 28;
            npc.lifeMax = 999;
            npc.HitSound = SoundID.NPCHit1;
            npc.DeathSound = SoundID.NPCDeath1;
            npc.value = Item.buyPrice(3, 0, 0);
            npc.knockbackResist = 0.5f;
            npc.aiStyle = 1;
            aiType = NPCID.Golem;
            animationType = NPCID.Golem;
        }

        public override float SpawnChance(MPSpawnInfo spawnInfo)
        {
            return SpawnCondition.Desert.Chance * 0.1f;
        }

        public override void AI()
        {
            Player player = Main.player[npc.target];
            attackTimer++;
            if (attackTimer == attackCooldown)
            {
                Vector2 velocity = Helper.VelocityToPoint(npc.Center, player.Center, 8f);
                int sandBlockID = TileID.Sand;
                int sandBlockDamage = 18;
                Projectile.NewProjectile(npc.Center.X, npc.Center.Y, velocity.X, velocity.Y, ProjectileID.Boulder, sandBlockDamage, 1f, Main.myPlayer);
            }
        }
    }
}
```

Рисунок 1.12. Написання коду на мові C# у Visual Studio

									Арк.
									20
Зм.	Арк.	№ докум.	Підпис	Дата					

РП 06. 04 001. ДП ПЗ

Piskel - це веб-графічний редактор, який спеціально призначений для створення піксельної графіки і анімації. Він працює у браузері, що дозволяє користувачам легко доступатися до нього без необхідності встановлення додаткового програмного забезпечення.

Режим піксельного малювання: Piskel надає зручний інтерфейс для створення та редагування зображень за допомогою піксельного рівня. Ви можете вибирати розмір холста та використовувати пензлі, відтінки кольорів та інші інструменти, щоб створити бажану графіку.

Анімація: Piskel підтримує створення анімацій, де ви можете створювати різні кадри зображень і відтворювати їх послідовно. Це корисно для створення анімованих персонажів або рухливих об'єктів у грі. Приклад створення анімації у Piskel на рисунку 1.13:

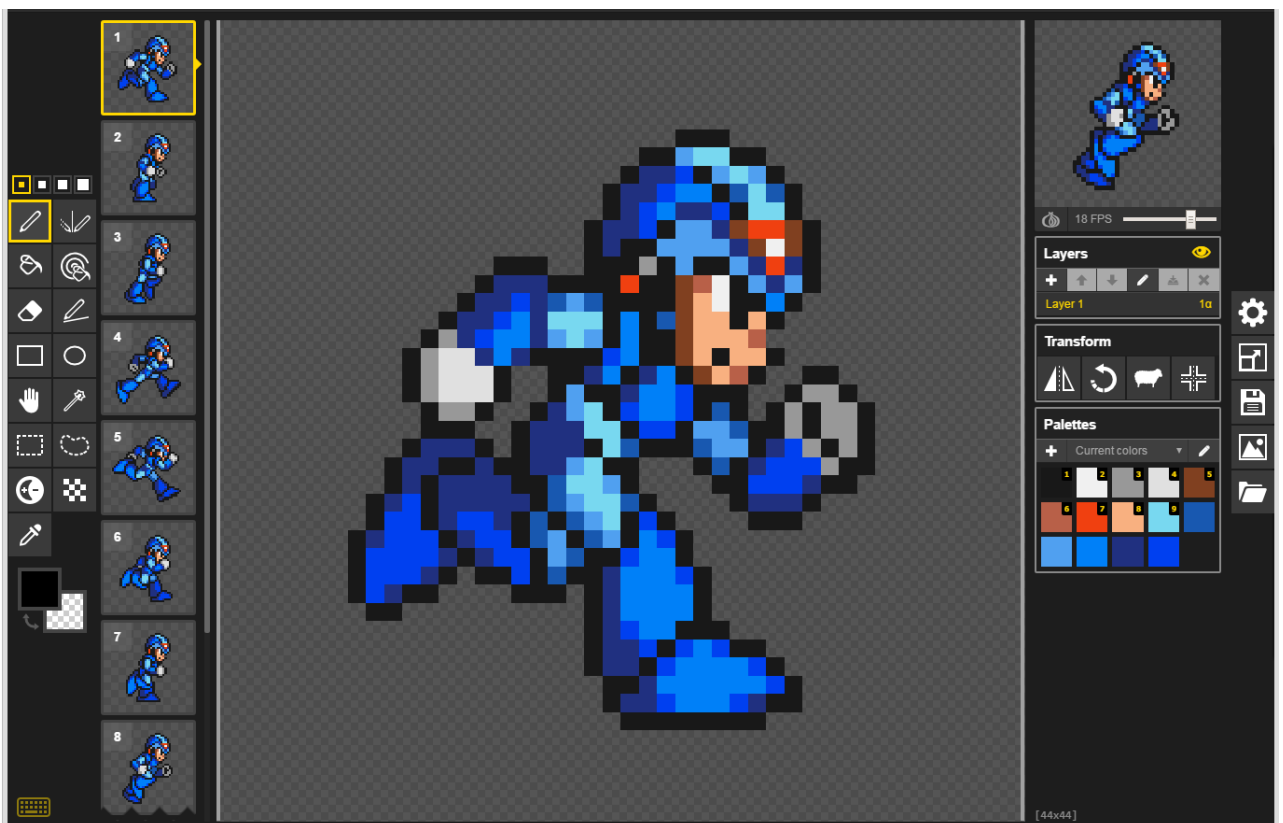


Рисунок 1.13. Створення анімації у Piskel

Збереження та завантаження: Ви можете зберігати свої роботи у форматі Piskel (.piskel) або експортувати їх у різні формати зображень, такі як PNG або GIF.

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

Шари: Piskel підтримує використання шарів, що дозволяє вам розділяти вашу графіку на окремі шари. Це полегшує редагування окремих елементів без впливу на інші.

Історія редагування: Редактор зберігає вашу історію редагування, що дозволяє вам переходити до попередніх станів зображення та скасовувати зміни.

Piskel є потужним інструментом для створення піксельної графіки та анімації безпосередньо у браузері. Він може бути використаний для створення моделей персонажів або ілюстрацій, які можуть бути використані у ваших іграх чи проектах.

TEdit: TEdit є одним із найпопулярніших редакторів у світі “Terraria”. Це дозволяє гравцям змінювати існуючі світи або створювати нові з нуля. За допомогою TEdit ви можете легко розміщувати та видаляти блоки, об’єкти та неігрові персонажі, а також змінювати рельєф і локації. Він забезпечує зручний інтерфейс із різноманітними інструментами редагування, такими як інструменти копіювання та вставки, заливки та пензля. TEdit також підтримує імпорт та експорт світів, дозволяючи гравцям ділитися своїми творіннями з іншими. Приклад з роботою у TEdit на рисунку 1.13:

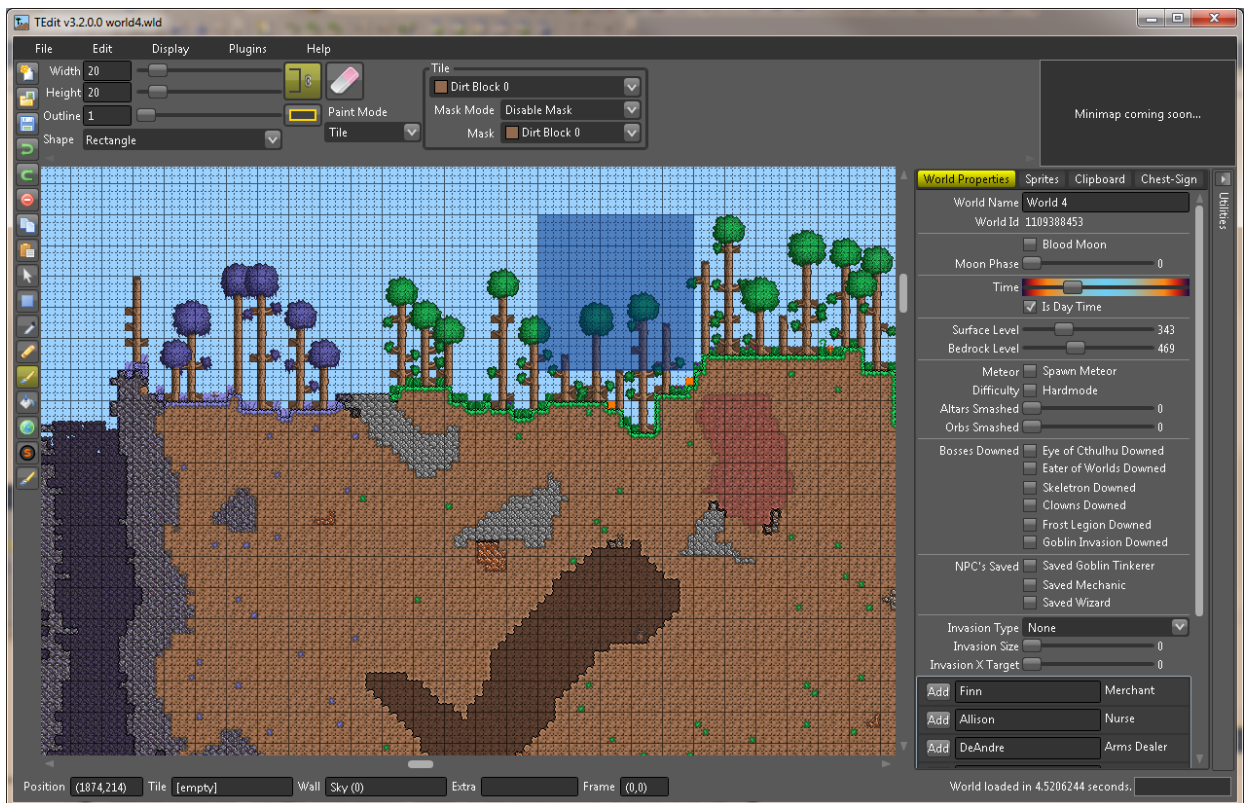


Рисунок 1.13. Робота у TEdit

Редагування світу: TEdit дозволяє вам редагувати різні аспекти світу Terraria, такі як блоки, фони, розміщення об'єктів та мобів, розмір світу тощо. Ви можете додавати, видаляти та переміщувати елементи у своєму світі зручним інтерфейсом редактора.

Інструменти малювання: TEdit надає різні інструменти для малювання, такі як пензлі, заливка, гумка тощо. Можна малювати блоки або фони у різних формах та кольорах, створюючи унікальні дизайни та ландшафти.

Режим перегляду: За допомогою режиму перегляду TEdit ви можете досліджувати свої світи безпосередньо в редакторі. Це дозволяє перевірити розміщення об'єктів, виявити помилки або оптимізувати розміщення елементів без необхідності запуску гри.

Комунарна підтримка: TEdit є популярним інструментом серед спільноти гравців Terraria. Часто спільнота створює свої власні схеми, функції та розширення для TEdit, що дозволяє використовувати сторонні ресурси та вдосконалювати досвід редагування світів.

TEdit є потужним інструментом для редагування світів у грі Terraria, який дозволяє вам створювати унікальні та цікаві світи для гри. Він дозволяє гравцям втілювати свої творчі ідеї та досліджувати безліч можливостей, які надає ця гра.

TerraCustom: TerraCustom — це генератор і редактор світу, який пропонує високий рівень налаштування. Це дозволяє гравцям регулювати параметри світу, такі як розмір, розподіл руди та макет локації. За допомогою TerraCustom ви можете створювати унікальні світи, пристосовані до конкретних тем або ігрового процесу. Він має графічний інтерфейс, який дозволяє переглядати та змінювати налаштування світу в реальному часі.

TEdit 3: TEdit 3 — це новіша версія оригінального TEdit з додатковими функціями та вдосконаленнями. Він пропонує покращену сумісність з останніми версіями “Terraria”. TEdit 3 надає інтуїтивно зрозумілий інтерфейс із рядом інструментів для редагування світу, включаючи пензлі, інструменти виділення та створення фігур. Це дозволяє точно контролювати створення світу, зміну місцевості та розміщення об’єктів і структур.

Інші інструменти: на додаток до TEdit і TerraCustom, існують інші сторонні інструменти, доступні для конкретних цілей модифікації “Terraria”. Такі інструменти, як WorldEdit, Super “Terraria” World Builder і T-SEdit, пропонують різні функції, як-от редагування рельєфу, власне створення світу та розміщення неігрових персонажів.

Використовуючи сторонні редактори рівня, важливо мати на увазі, що вони змінюють файли гри та можуть вимагати певних технічних знань або знайомства з модифікацією. Завжди створюйте резервні копії своїх світів “Terraria” перед використанням будь-яких зовнішніх інструментів і дотримуйтесь інструкцій, наданих відповідними розробниками.

Ці сторонні редактори рівнів надають гравцям “Terraria” розширені творчі можливості, дозволяючи їм створювати унікальні світи та досвід, що перевищує те, що пропонує базова гра. Вони роблять внесок у процвітаючу

					<b>РП 06. 04 001. ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

спільноту модифікаторів і дозволяють гравцям ділитися своїми творіннями з іншими, ще більше збагачуючи ігровий процес “Terraria”.

Цей технологічний стек та інструментарій надають розробникам можливість створювати різноманітні модифікації для гри Terraria, розширюючи її функціональність та створюючи унікальні геймплейні елементи для гравців.

## 1.5 Проектування модифікації

При створенні нового проекту необхідно здійснити налаштування VisualStudio, що без них модифікація не зможе працювати належним чином. Для досягнення цієї мети, потрібно увійти до середовища VisualStudio і звернутися до параметрів налаштувань, які включають "розробка класичних застосунків .NET" та "Розробка застосунків для універсальної платформи Windows" (рисунок 1.14). Ці конфігурації є необхідними для розробки модифікації, спрямованої на гру Terraria, тому що вони дають можливість користуватися XNA. Зображення налаштувань на рисунку. XNA (XNA's Not Acronymed) - це набір розробницьких інструментів, розроблений компанією Microsoft, призначений для створення ігрових програм для різних платформ, зокрема для Windows, Xbox 360 та Windows Phone. XNA базується на платформі .NET Framework і використовує мову програмування C#.

XNA Framework: Це основна складова XNA, яка надає інтерфейси програмування додатків (API) для розробки ігор. Вона включає класи та функції для роботи з графікою, звуком, мережею, введенням користувача та іншими аспектами ігрової розробки.

XNA Game Studio: Це інтегроване середовище розробки (IDE) для XNA, яке базується на Microsoft Visual Studio. Воно надає засоби для написання, відлагодження та збирання ігрових програм з використанням XNA Framework. XNA Game Studio також має різноманітні шаблони проектів, які полегшують початок роботи з розробкою ігор.

					<b>РП 06. 04 001. ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

Графічні ресурси: XNA надає можливості для роботи з 2D і 3D графікою. Ви можете створювати спрайти, текстури, моделі, анімацію та інші візуальні елементи для своїх ігор. XNA підтримує різні формати файлів для графічних ресурсів, такі як PNG, JPEG, BMP, FBX та інші.

XNA дозволяє вам використовувати звукові ефекти, музику та інші аудіо-ресурси в своїх іграх. Ви можете завантажувати і відтворювати звукові файли різних форматів, таких як WAV, MP3 та інші.

Управління введенням: XNA надає можливість взаємодіяти з введенням користувача, таким як клавіатура, миша, геймпади та інші пристрої вводу. Ви можете отримувати дані про натискання клавіш, рухи миші або джойстика, щоб керувати поведінкою вашої гри.

Мережеве програмування: XNA надає можливість розробляти мережеві ігри, де гравці можуть взаємодіяти один з одним через Інтернет або локальну мережу. Ви можете використовувати XNA для передачі даних між гравцями, синхронізації гри та інших мережевих функцій.

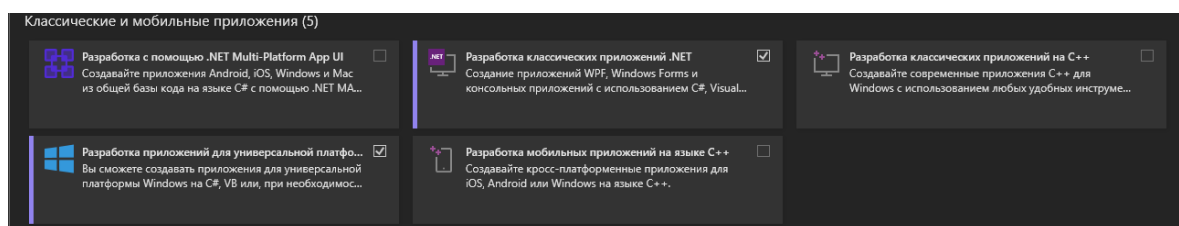


Рисунок 1.14. Налаштування Visual Studio

Перед тим як розпочати написання коду дуже зручно перш за все створити моделі для неігрових персонажів, які пізніше будуть використовуватися у грі. Це дозволить максимально упорядковано і систематично приступити до розробки. Для цього можна скористатися веб-графічним редактором під назвою Piskel, який працює безпосередньо у браузері. Piskel - це веб-графічний редактор, який спеціально призначений для створення піксельної графіки і анімації. Він працює у браузері, що дозволяє користувачам легко доступатися до нього без необхідності встановлення додаткового програмного забезпечення. Він надає достатній функціонал для створення бажаних моделей, не потребуючи великого обсягу

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

додаткових ресурсів або встановлення програмного забезпечення. Приклад інтерфейсу на рисунку 1.15:

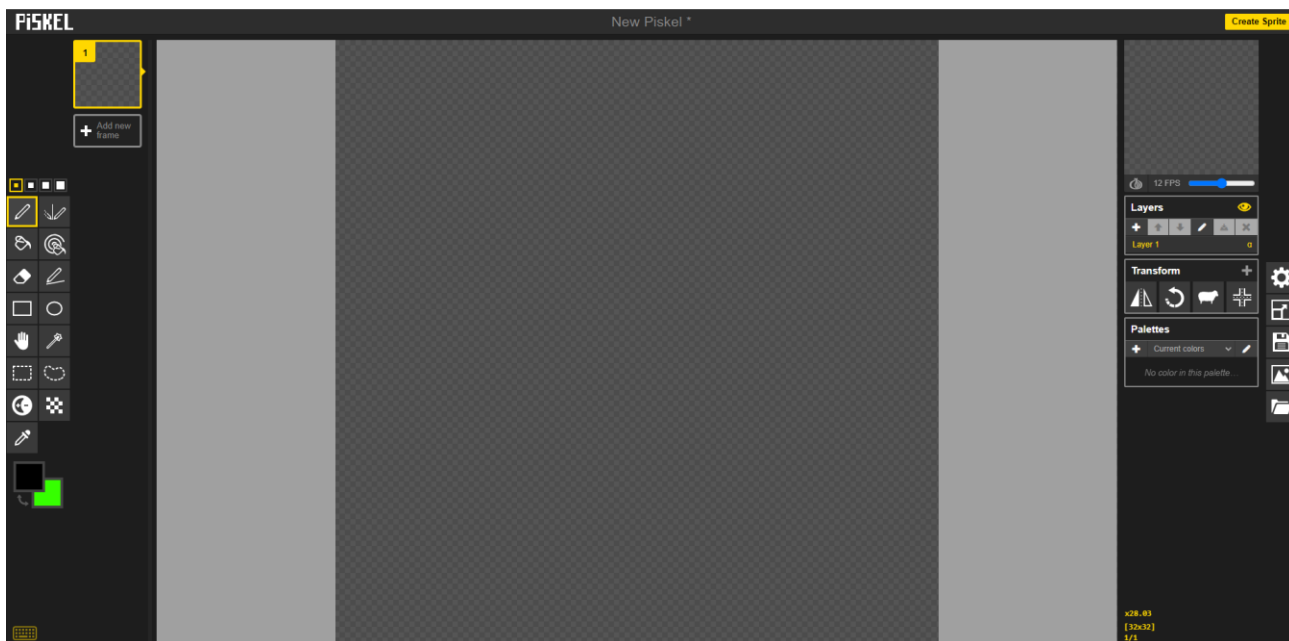


Рисунок 1.15. Інтерфейс графічного редактора

Після завершення створення моделей персонажів можна перейти до наступного кроку - написання коду. Для цього необхідно виконати послідовні дії. По-перше, потрібно створити новий проект. Для цього зайдіть в tModLoader, де головне меню повністю відповідає головному меню самої гри. В tModLoader ви зможете знайти необхідні інструменти та опції для створення нового проекту. Вид головного меню гри на рисунку 1.16:



Рисунок 1.16. Головне меню гри

					<b>РП 06. 04 001. ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Після активації tModLoader, наступним кроком є перехід до майстерні. Майстерня виступає робочим простором, де розробник або звичайний гравець можуть завантажувати, налаштовувати або створювати модифікації. В майстерні доступні різноманітні можливості, що дозволяють вам працювати з модифікаціями, вносити зміни до гри та налаштовувати їх відповідно до ваших потреб. Тут ви зможете завантажувати свої моди, налаштовувати їх параметри та створювати нові модифікації для покращення геймплею. Майстерня надає широкі можливості для творчості та розширення функціоналу гри, дозволяючи вам налаштовувати та персоналізувати свій власний ігровий досвід. Вигля майстерні на рисунку 1.18:



Рисунок 1.18. Майстерня

У майстерні існує окремий відділ для розробників, що надає можливість розпочати створення модифікацій. Для цього потрібно просто натиснути кнопку "Створити мод", що відкриє нове меню з різноманітними опціями. Тут розробникам пропонується створити новий мод, відкрити джерела для модифікацій або скопіювати вже наявні моди. Цей процес відкриває безліч можливостей для розробників, дозволяючи їм налаштовувати та створювати унікальні модифікації для гри. В майстерні розробники зможуть втілити свої творчі ідеї, вносити зміни до геймплею та додавати новий контент, розширюючи можливості гри для гравців. Приклад можливостей майстерні на рисунку 1.19:

					<b>РП 06. 04 001. ДП ПЗ</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28



Рисунок 1.19. Панель розробника

Після натискання кнопки "Створити мод" з'являється меню, де потрібно вказати деякі обов'язкові дані. Вам буде запропоновано ввести назву створюваної модифікації, яка буде відображатись у майстерні, а також вказати ініціали автора модифікації. Ці дані дозволяють ідентифікувати вашу модифікацію та з'ясувати, хто є її автором. Заповнення цих полів є важливою частиною процесу створення модифікації та дозволяє забезпечити належну інформаційну відмітку для вашого творчого внеску. Важливо залишити поле "Базовий меч" порожнім при введенні даних, оскільки tModLoader автоматично створить виконуваний файл під назвою "Базовий меч". У цьому файлі буде автоматично згенерований приклад коду для меча. Ви не повинні вводити жодних значень у це поле, оскільки tModLoader самостійно створить необхідний файл і заповнить його прикладовим кодом меча. Це спрощує процес створення модифікації, дозволяючи автоматично згенерувати основний код для базового меча, який можна подальше налаштувати та розширити за потребою. Приклад першочергових налаштувань на рисунку 1.20:

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		29



Рисунок 1.20. Створення моду

Важливо відкривати файли модифікації через tModLoader, оскільки це забезпечує зручність та ефективність у роботі. Без увімкненого фреймворка tModLoader неможливо буде успішно запустити та виконати код модифікації. Використання tModLoader дозволяє автоматично завантажувати та інтегрувати модифікації у гру, забезпечуючи правильне виконання коду та взаємодію з ігровим середовищем. Це забезпечує безпроблемну розробку та тестування модифікацій, що робить процес розробки набагато зручнішим та ефективнішим.

Після введення основних даних, tModLoader автоматично створить папку для вашої модифікації, де будуть розміщені виконувані файли моду. У цій папці вам необхідно завантажити моделі неігрових персонажів. Цей крок є важливим перед тим, як приступити до написання коду для вашої ігрової модифікації. Забезпечуючи наявність моделей персонажів у відповідних папках, це забезпечить коректну роботу модифікації і можливість взаємодії з цими персонажами під час гри. Тільки після успішного виконання попередніх кроків ви зможете приступити до написання коду для ігрової модифікації. Приклад вмісту папки модифікації на рисунку 1.21:

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		30

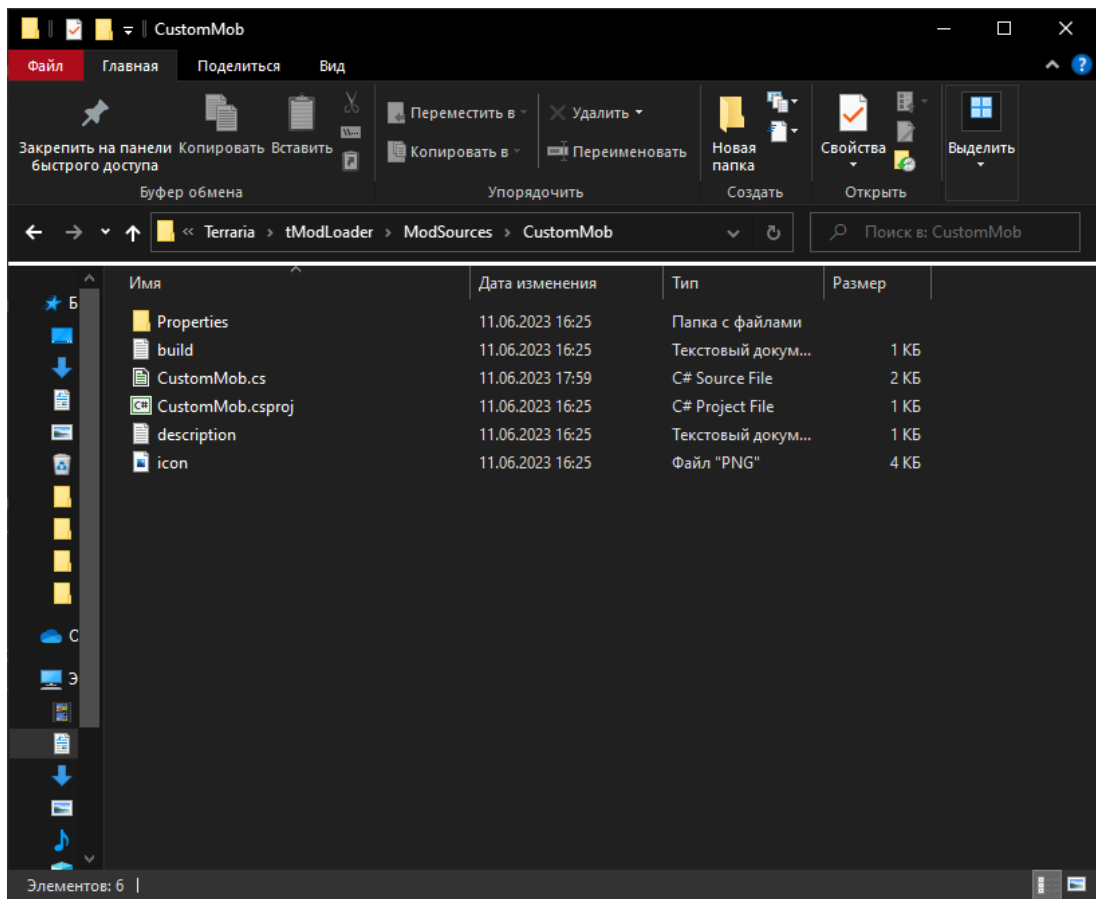


Рисунок 1.21. Вміст папки модифікації

## 1.6 Модифікація ігрових елементів оточення

Модифікація включає в себе нову локацію, яка зовнішнім виглядом нагадує пустелю пурпурного кольору (рисунок 1.22). Як і звичайна пустеля, при створенні світу малого розміру, ця пустеля здобуде скелясті формації, що надають їй унікальний вигляд. У разі створення світу великого розміру, пустеля пурпурного кольору буде представляти собою рівну та без великих перепадів висот місцевість. Це класична поведінка пустельних локацій при створенні світів різних розмірів. При генерації світу у пустелі можуть з'явитися черепа тварин, створюючи атмосферний образ. Черепа розташовані розсіпом по піску, тому що пустеля повинна нагадувати що вона має суровий клімат. Для того щоб створити потрібно у графічному редакторі намалювати модель основного блоку який буде всюди знаходитися у локації і тим самим створювати ефект локації. До таких ефектів належать: поява особливих типів ворогів які відносяться лише до цієї локації, підвищення

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

швидкості появи ворогів, зміна заднього фону локації, музикальний супровід у локації. На рисунку 1.23 приклад розгортки блоків пустельної локації.

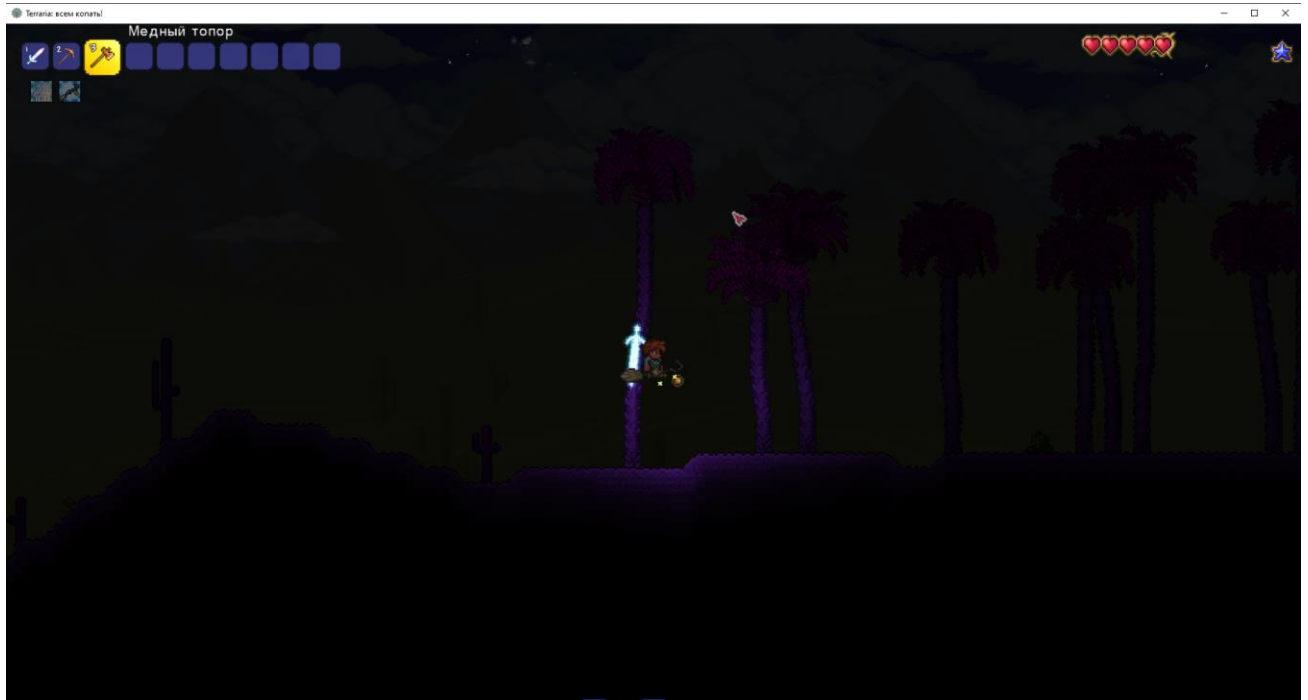


Рисунок 1.22. Видяд пурпурної пустелі

```
private void GenerateEternalNightDesert(GenerationProgress progress)
{
    for (int i = 0; i < Main.maxTilesX; i++)
    {
        for (int j = 0; j < Main.maxTilesY; j++)
        {
            if (Main.tile[i, j].active() && Main.tile[i, j].type == TileID.Sand)
            {
                Main.dayTime = false;
                Main.time = 0;
                Main.tile[i, j].wall = WallID.DesertBasiliskWall;
                Main.tile[i, j].wallColor(Color.Purple);
                Main.NewText("Пустелю охопила ніч!", Color.Purple);
                isInEternalNightDesert = true;
            }
            return;
        }
    }
}
```

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

}  
}  
}

У даному коді виконується генерація пустелі зі сталою ніччю (eternal night desert).

Починаючи з циклу for, перебираються всі тайли на карті за допомогою двох циклів. Для кожного тайлу, якщо він активний (має об'єкт) і його тип є TileID.Sand (пісок), то виконується наступне:

Змінна Main.dayTime встановлюється в false, що означає, що наступає ніч.

Змінна Main.time встановлюється в 0, що встановлює початок доби в 0 годин (початок ночі).

Тайлу присвоюється певний тип стіни WallID.DesertBasiliskWall, який означає, що пустеля отримує новий вид стіни.

Колір стіни встановлюється на фіолетовий шляхом виклику методу Main.tile[i, j].wallColor(Color.Purple).

Виводиться повідомлення за допомогою Main.NewText, що пустелю охопила ніч. Колір повідомлення також встановлюється на фіолетовий Color.Purple.

Змінна isInEternalNightDesert встановлюється в true, що позначає, що генерована пустеля має сталу ніч.

Оскільки було знайдено пісок і виконані всі необхідні дії, цикли припиняються за допомогою оператора return.

Отже, цей код перебирає всі тайли на карті і, якщо знаходиться активний тайл піску, створює пустелю зі сталою ніччю.

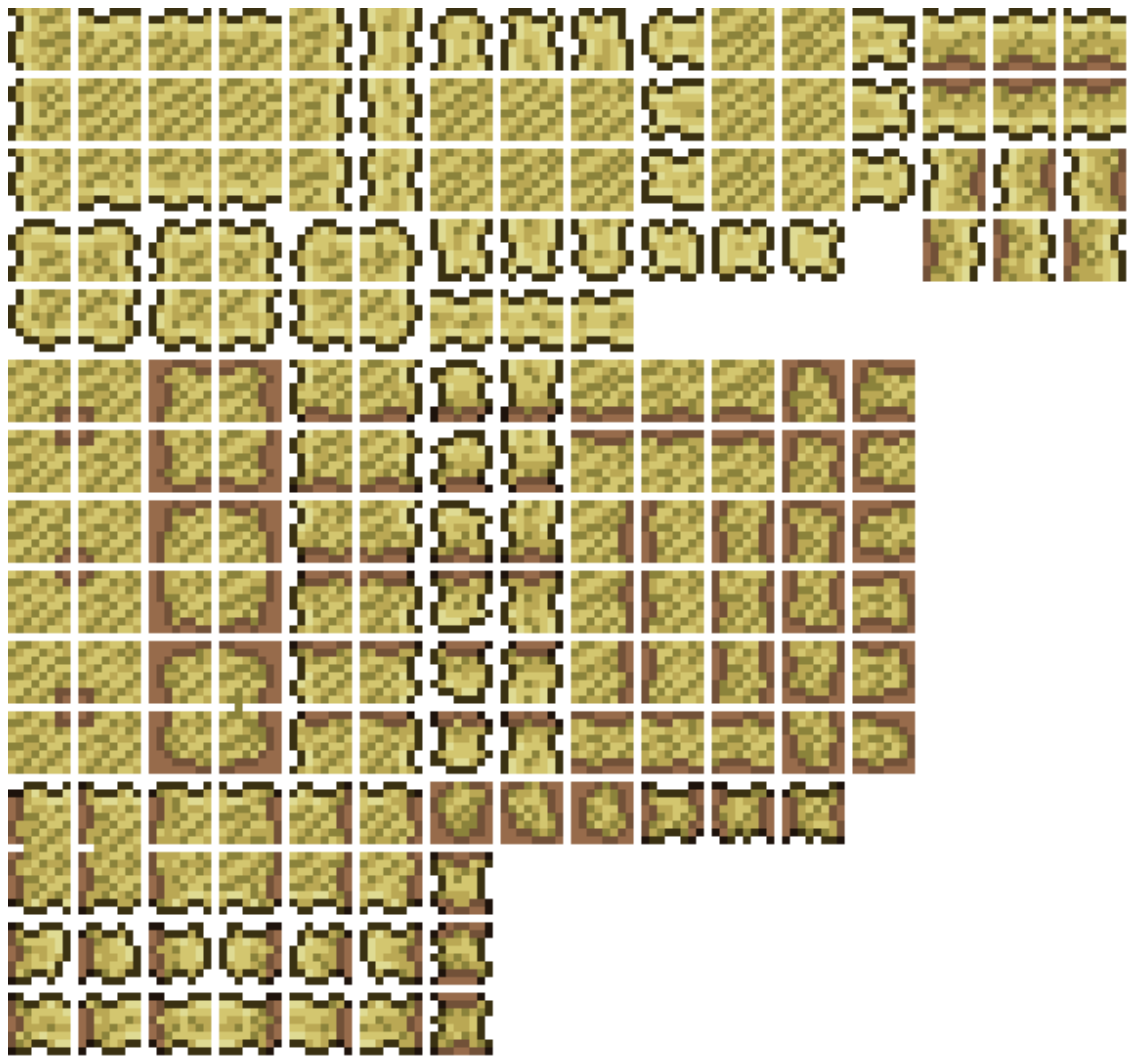


Рисунок 1.23. Розгортка моделей пустельних блоків

У цієї локації впродовж усього часу панує ніч, яка настає миттєво, як тільки гравець підходить достатньо близько до його меж. Ігровий час зупиняється і продовжується лише в той момент, коли гравець вирішує вийти за межі цієї локації. За це відповідаю ці строки коду:

```
public override void PostUpdate()
{
    bool isInDesert = false;

    Rectangle playerRect = new
Rectangle((int)Main.LocalPlayer.position.X / 16,
(int)Main.LocalPlayer.position.Y / 16, Main.LocalPlayer.width / 16,
Main.LocalPlayer.height / 16);

    for (int i = playerRect.X; i < playerRect.X + playerRect.Width; i++)
```

```

{
for (int j = playerRect.Y; j < playerRect.Y + playerRect.Height; j++)
{
if (Main.tile[i, j].active() && Main.tile[i, j].type == TileID.Sand)
{
isInDesert = true;
break;
}
}
}
if (isInDesert && !isInEternalNightDesert)
{
Main.dayTime = false;
Main.time = 0;
Main.NewText("Пустелю охопила ніч!", Color.Purple);
isInEternalNightDesert = true;
}
else if (!isInDesert && isInEternalNightDesert)
{
isInEternalNightDesert = false;
}
}

```

У даному кодї, який розміщено в методі PostUpdate, перевіряється, чи гравець знаходиться у пустелі і відповідно встановлюється стан пустелі зі сталою ніччю (isInEternalNightDesert).

Спочатку створюється змінна isInDesert, яка ініціалізується значенням false.

Створюється прямокутник playerRect, що представляє розмір та положення гравця на основі його позиції та розміру.

За допомогою вкладених циклів for перевіряється кожен тайл в межах прямокутника playerRect.

Якщо знайдений тайл активний (має об'єкт) і його тип є TileID.Sand (пісок), то змінна isInDesert встановлюється в true, і цикли припиняються за допомогою оператора break.

Після перевірки, якщо гравець знаходиться у пустелі і в попередньому кадрі вона не була вже зі сталою ніччю (isInEternalNightDesert), то виконується наступне:

Змінна Main.dayTime встановлюється в false, що означає, що наступає ніч.

Змінна Main.time встановлюється в 0, що встановлює початок доби в 0 годин (початок ночі).

Виводиться повідомлення за допомогою Main.NewText, що пустелю охопила ніч. Колір повідомлення встановлюється на фіолетовий (Color.Purple).

Змінна isInEternalNightDesert встановлюється в true, що позначає, що пустеля має сталу ніч.

У іншому випадку, якщо гравець не знаходиться у пустелі, але в попередньому кадрі пустеля мала сталу ніч (isInEternalNightDesert), то змінна isInEternalNightDesert встановлюється в false, означаючи, що стан пустелі зі сталою ніччю скасовується.

Отже, цей код перевіряє присутність гравця у пустелі і змінює стан пустелі на сталий нічний режим, якщо він знаходиться у пустелі, або скасовує сталий нічний режим, якщо гравець покинув пустелю.

Також головним правилом гри є те що кожна локація повинна мати свій музикальний супровід, який ще більше занурює гравця у атмосферу. У цьому аспекті пурпурна пустеля нічим не відрізняється від звичайної, бо має той самий музикальний супровід що й звичайна пустеля.

Ще однією особливістю цієї оації є те, що вона підвищує появу нічних ворогів. Ось що робить кожен рядок коду відповідаючий за це:

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		36

Main.monolithType = MonolithID.Desert; встановлює monolithType на біом пустелі, що впливає на частоту появи ворога.

Main.maxRaining = 0; гарантує, що в біомі Пустелі ніколи не йде дощ, що може вплинути на появу ворога.

Main.monolithSpawnBonus = 200; встановлює MonolithSpawnBonus до 200, фактично подвоюючи швидкість появи ворогів у пустелі. Додавши ці рядки, швидкість появи ворогів у спеціальній локації пустелі з вічною ніччю подвоїться порівняно зі швидкістю появи за замовчуванням.

## 1.7 Модифікація нових персонажів

У рамках модифікації впроваджено двох нових неігрових персонажів, які прикрашають гру та забезпечують більш різноманітні можливості для гравця. Перший з них – чорна кам'яна істота, яка виділяється від інших ворогів, з'являючись у пустелі. Вона має значно підвищені показники здоров'я, шкоди та захисту, зробивши її виразним та випробовувальним противником для гравця. Її висока стійкість дозволяє витримувати значну кількість пошкоджень, а потужність атаки може становити значну загрозу для гравця. Завдяки підвищеному рівню захисту, ця істота є менш вразливою до атак гравця. Спочатку була створена модель для самої кам'яної істоти у графічному редакторі Piskel. Процес створення кам'яної істоти на рисунку 1.24:

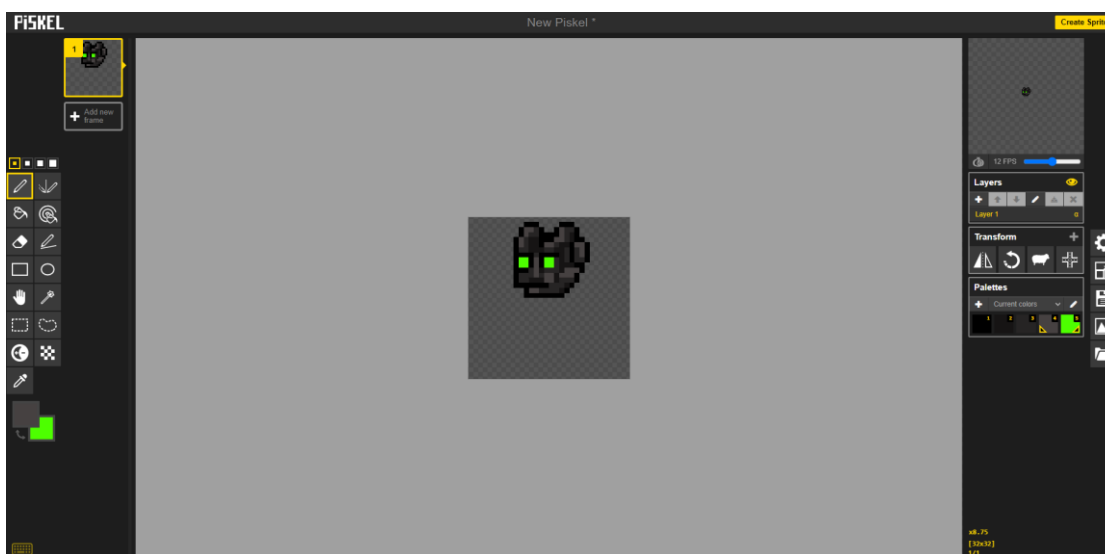


Рисунок 1.24. Створення кам'яної істоти

					РП 06. 04 001. ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

Спочатку був створений перший концепт кам'яної істоти, але він пізніше він був замінений на інший з меншими зеленими очима. Остаточна модель була завантажена у папку модифікації. Де вона завантажуватися від час виконання кода. На рисунку 1.25 схема кам'яної істоти:

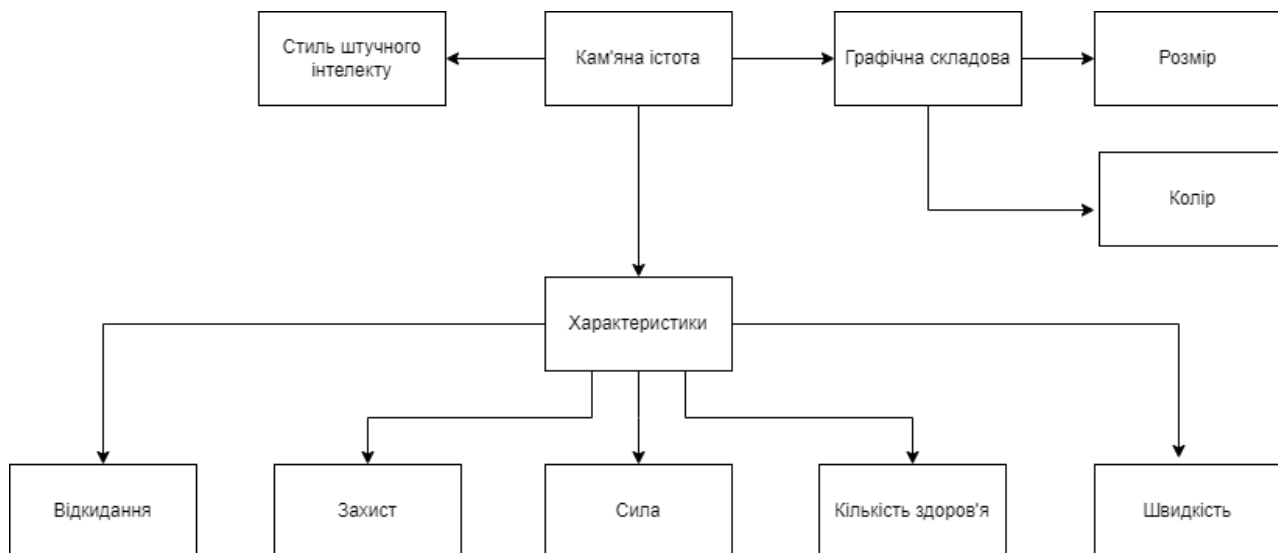


Рисунок 1.25. Схема кам'яної істоти

Другий новий персонаж – торговець, який додає елемент взаємодії та розвитку ігрового процесу. Гравець може здійснювати різні дії з цим персонажем, включаючи покупку та продаж різноманітних предметів. Взаємодія з торговцем може бути різною, дозволяючи гравцю використовувати його послуги для поліпшення ігрового процесу, отримання корисних предметів або отримання цінної інформації. Торговець може мати власний інвентар та пропонувати різноманітні товари.

Завдяки цим новим неігровим персонажам гравець отримує більше можливостей для виклику та розвитку, що додає глибини та різноманітності ігрового процесу. Чорна кам'яна істота створює складніші битви та випробовує навички гравця, тоді як торговець відкриває нові можливості для розвитку та прогресу в грі. Гравець може насолоджуватися взаємодією з цими персонажами, розширюючи світ гри та поглиблюючи свій іммерсивний досвід (рисунок 1.26). Характеристики кам'яної істоти такі:

`npc.width = 86;`

`npc.height = 92;`

```

npc.damage = 60;
npc.defense = 30;
npc.lifeMax = 500;
npc.HitSound = SoundID.NPCHit1;
npc.DeathSound = SoundID.NPCDeath1;
npc.value = Item.buyPrice(0, 5, 0, 0);
npc.knockBackResist = 0.5f;
npc.aiStyle = 3;
aiType = NPCID.Golem;

```



Рисунок 1.26. Вигляд кам'яної істоти

Він має досить невелику ймовірність появи порівняно з іншими ворогами. Ця кам'яна істота поводить себе дуже агресивно стосовно гравця. Він буде проводити атаку на середній дистанції, жбурляючи блоки піску у гравця, коли гравець попаде у його зону видимості. За таку поведунку відповідають такі строки коду.

```

public override void AI()
{
    Player player = Main.player[npc.target];
    attackTimer++;
    if (attackTimer >= attackCooldown)

```

```

{
    Vector2 velocity = Helper.VelocityToPoint(npc.Center, player.Center, 8f);
    int sandBlockID = TileID.Sand;
    int sandBlockDamage = 45;
    Projectile.NewProjectile(npc.Center.X,    npc.Center.Y,    velocity.X,
velocity.Y, ProjectileID.Boulder, sandBlockDamage, 1f, Main.myPlayer);
    attackTimer = 0;
}
}

```

У даному коді, який розміщений в методі AI для NPC, визначається поведінка NPC "rock golem" в бою з гравцем.

Спочатку отримується посилання на гравця за допомогою Player player = Main.player[npc.target];

Затім збільшується значення attackTimer на 1.

Умова if (attackTimer >= attackCooldown) перевіряє, чи час від останньої атаки NPC перевищує встановлений часовий інтервал attackCooldown.

Якщо умова виконується, виконується наступне:

Обчислюється вектор швидкості velocity за допомогою допоміжного методу Helper.VelocityToPoint, який обчислює швидкість напрямлення від центру NPC до центру гравця з швидкістю 8f.

Визначаються значення sandBlockID та sandBlockDamage для піску, який буде кинуте в гравця.

Створюється новий снаряд (Projectile) за допомогою Projectile.NewProjectile, що представляє снаряд типу ProjectileID.Boulder (буліджер), розташований у центрі NPC, зі швидкістю velocity, з пошкодженням sandBlockDamage і шаром зіпсованого піску зі значенням sandBlockID.

Значення attackTimer скидається до 0, щоб розпочати лічильник знову.

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

Отже, цей код визначає, що NPC "rock golem" атакує гравця киданням снаряду типу "буліджер" у напрямку гравця з певною швидкістю та пошкодженням.

Після своєї смерті ворог залишає після себе 10 блоків піску та 10 золотих монет у я кості винагороди за перемогу над ним. Його прототипом стала кам'яна істота із класичної версії гри (рисунок 1.27), яку можна знайти лише у підземеллі, яка має 100 одиниць здоров'я, 35 одиниць захисту та 45 одиниць дальнього бою. Ця істота теж має невелику ймовірність появи поблизу гравця.



Рисунок 1.27. Кам'яна істота класичної версії гри

Наступний неігровий персонаж це торговець. Він має здатність пропонувати гравцю придбати різноманітні предмети, спрямовані на виклик головних ворогів. Для того, щоб гравець міг придбати ці спеціальні предмети, він повинен наблизитися до неігрового персонажа та взаємодіяти з кнопкою "Магазин", яка призведе до відкриття інвентаря торговця. В інвентарі гравцеві будуть доступні приманки, спеціально створені для привернення уваги головних ворогів.

Крім того, для того, щоб здійснити покупку, гравець повинен мати достатню кількість монет, які можна отримати під час перемоги над звичайними ворогами. Це відображає механізм внутрішньої економіки гри, де гравець отримує винагороду у вигляді монет за успішні битви зі звичайними ворогами, що надає їм можливість отримати доступ до цінних предметів через торговця.

У цьому контексті, гравець має можливість використовувати стратегію, спираючись на предмети, придбані від торговця, для протистояння головним ворогам. Це створює додаткові можливості для гравця вплинути на хід гри, розширюючи їх можливості в битві з найсильнішими противниками. Така система торгівлі, заснована на доступі до приманок для головних ворогів, підкреслює значення стратегічного мислення та прийняття важливих рішень гравцем у контексті ігрового процесу.

Ось частина коду яка відповідає за наявність приманок у торговця:

```
shop.item[nextSlot].SetDefaults(ItemID.SlimeCrown);
nextSlot++;
shop.item[nextSlot].SetDefaults(ItemID.SuspiciousLookingEye);
nextSlot++;
shop.item[nextSlot].SetDefaults(ItemID.BloodyTear);
nextSlot++;
shop.item[nextSlot].SetDefaults(ItemID.GoblinBattleStandard);
```

У даному коді, використовуються об'єкти типу Item для налаштування предметів, які можна купити у торговця в магазині.

shop.item[nextSlot].SetDefaults(ItemID.SlimeCrown); - Налаштування першого предмета у магазині. Встановлюється тип предмета за допомогою ItemID.SlimeCrown, що представляє "Слизову корону". Значення типу предмета присвоюється об'єкту shop.item[nextSlot], де nextSlot вказує на наступний індекс у списку предметів магазину. Після цього значення nextSlot збільшується на 1 для переходу до наступного слоту магазину.

					<i>РП 06. 04 001. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

Аналогічно до попереднього кроку,  
shop.item[nextSlot].SetDefaults(ItemID.SuspiciousLookingEye); -  
Встановлюється тип предмета як "Підозріло виглядаюче око".

shop.item[nextSlot].SetDefaults(ItemID.BloodyTear); - Встановлюється  
тип предмета як "Кровава сльоза".

shop.item[nextSlot].SetDefaults(ItemID.GoblinBattleStandard); -  
Встановлюється тип предмета як "Гоблінський бойовий стяг".

Кожен з цих кроків додає новий предмет до списку доступних для продажу у магазині. Кожен наступний предмет розміщується в наступному слоті магазину, завдяки використанню змінної nextSlot, яка забезпечує правильне розташування предметів у списку магазину.

Зовнішній вигляд торговця відповідає класичному зображенню торговця у відеогрі, але з деякими відмінностями. Він повністю покритий білим кольором, що створює враження майже безтілесного існування. Початково концепція торговця передбачала його уявлення як привида, проте таке рішення не відповідало деякій логіці гри. Процес створення моделі торговця на рисунку 1.28.

Тому, замість цього, його зовнішній вигляд був змінений, і тепер торговець має білу оболонку, що надає йому унікальний вигляд і виділяє його серед інших персонажів у грі. Ця зміна вигляду торговця дозволяє зберегти логічну послідовність та гармонію в ігровому світі, одночасно надаючи йому певну естетичну привабливість. (рисунок 1.29).

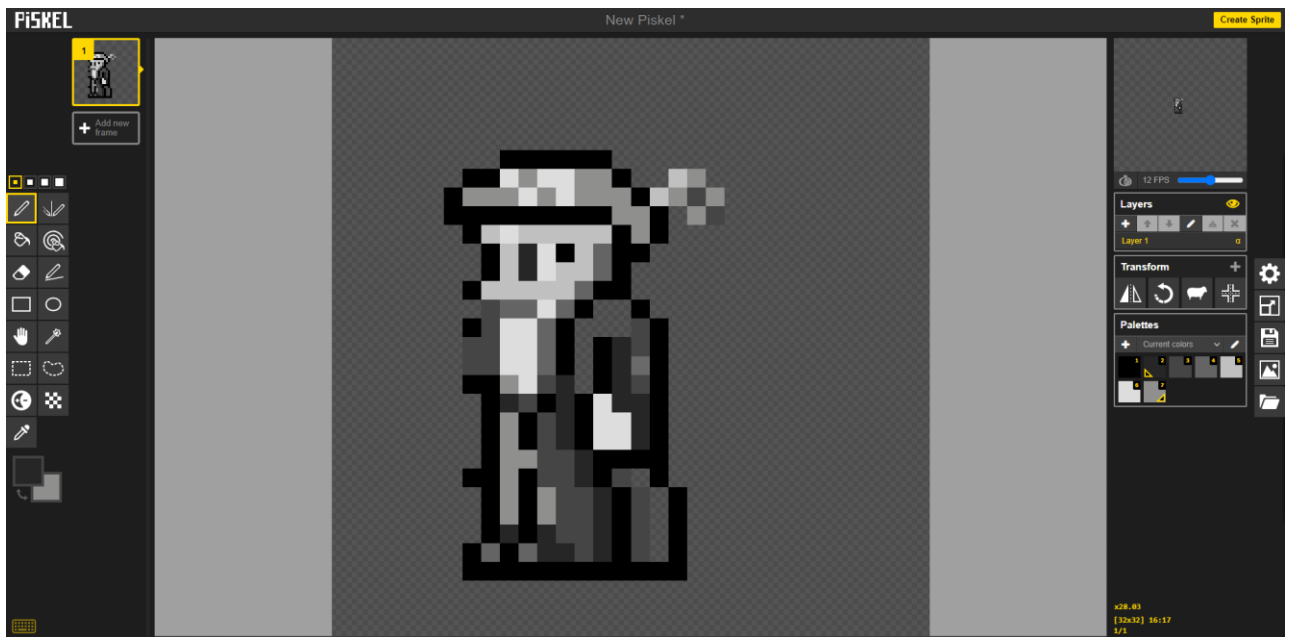


Рисунок 1.28. Процес створення моделі торговця



Рисунок 1.29. Вигляд торговця

Торговець також пропонує можливість спілкування з гравцем через набір прописаних реплік. Він послідовно висловлює ці репліки гравцеві під час інтеракції. Ці репліки можуть включати звичайні фрази, що мають на меті підтримати атмосферу гри та створити почуття взаємодії з персонажем. Крім того, він також може надавати корисні поради щодо ігрового процесу або механіки гри.

Це спілкування з торговцем додає глибини та іммерсії до ігрового процесу, роблячи його більш насиченим та цікавим для гравця. Репліки торговця можуть створювати почуття присутності живого персонажу у віртуальному світі, а також надавати корисну інформацію та підказки для подальшого прогресу гравця. Гравець може звернутися до торговця, щоб

дізнатися додаткові відомості або отримати поради, які допоможуть йому у вирішенні завдань та досягненні успіху в грі.

За це відповідають такі строки коду:

```
string[] chat = {  
    "Купи в мене приманку! Не пожалієш",  
    "Хочеш призвати Босса? В мене є те, що тобі потрібно!",  
    "Викликати їх можна тільки після заходу сонця!",  
    "Для перемоги над армією гоблінів треба лише колючі шари"  
};
```

Для появи торговця у вашому місті необхідно виконати кілька обов'язкових умов. Перш за все, в місті повинна бути принаймні одна вільна кімната, яка може бути використана для проживання торговця. Друга умова передбачає перемогу над головним ворогом під назвою "Око Ктулху". Це може бути складним завданням, оскільки цей ворог має значний рівень складності. Нарешті, гравець повинен мати при собі приманку для будь-якого головного ворога, яка є необхідною для залучення торговця до вашого міста.

Виконання цих умов забезпечує можливість появи торговця, який стане цінним додатком до вашої гри, надаючи нові можливості для торгівлі та прогресування гравця. За ці умови відповідають такі строки коду:

```
public override bool CanTownNPCSpawn(int numTownNPCs, int money)  
{  
    return numTownNPCs > 1 && NPC.downedBoss1 &&  
Main.player[Main.myPlayer].inventory.Any(item => item.type ==  
ItemID.SuspiciousLookingEye || item.type == ItemID.WormFood || item.type ==  
ItemID.SlimeCrown);  
}
```

Якщо всі відповідні умови будуть виконані, то неігровий персонаж, який виступає в ролі торговця, з'явиться у грі та займе вільну кімнату в будинку. Це означає, що гравець має можливість надати неігровому

персонажу власне проживання, де він буде здійснювати свою торговельну діяльність та спілкуватися з гравцем.

Важливо забезпечити відповідні умови для розташування неігрового персонажу, таким як наявність вільної кімнати у будинку. Цей простір повинен мати необхідні меблі та інші об'єкти, що забезпечать комфортне розташування торговця і дозволять йому здійснювати свою діяльність. З такими умовами, неігровий персонаж готовий прийняти гравця в грі і надавати послуги торгівлі, а також виконувати свої ролі та функції, визначені у грі.

Крім того, торговець має свою унікальну поведінку. У нічний час він постійно розшукуватиме місце для схованки, щоб забезпечити свою безпеку.

Алгоритм поведінки на рисунку 1.30:

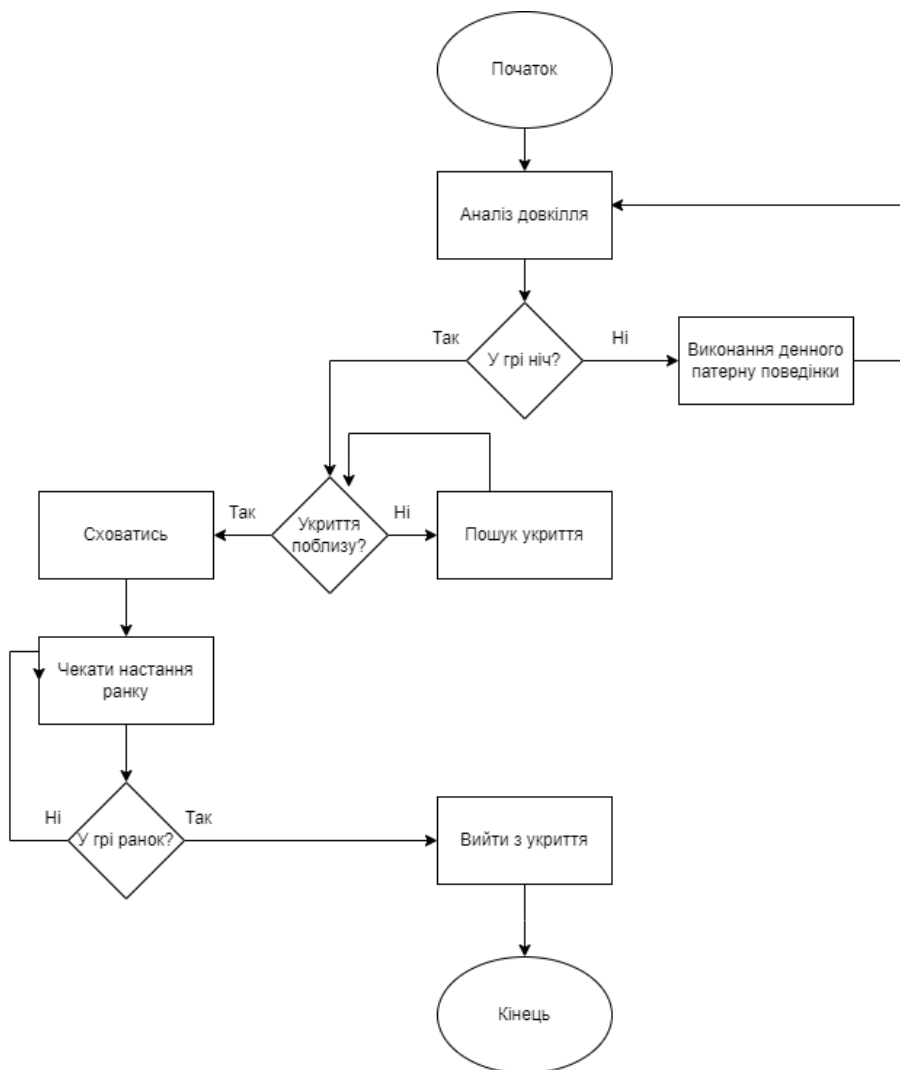


Рисунок 1.30. Алгоритм поведінки торговця

## 2 ЕКОНОМІЧНА ЧАСТИНА

### 2.1 Резюме

У моєму дипломному проєкті я розробив ігрову модифікацію, яка додає нову локацію та двох нових неігрових персонажів. Модифікація не є окремим програмним забезпеченням, а являє собою завантажуваним контентом до гри «Террарія» і має ті ж самі системні потреби. Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення.

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення.

Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнуою спроможністю каналів передачі даних. Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

### 2.2 Визначення трудомісткості розробки програмного забезпечення.

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

					<b>РП 06.04.002.00 ДП ПЗ</b>	<i>Лист</i>
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		47

Таблиця 2.1. -Каталог аналогів

Найменування ПП	Обсяг функції ПП – $V_0$ , усл. машинних командах.
1. ПП автоматизації засобів по каталогу	680 – 7000
2. ПП автоматизованих розрахунків	1300 – 8600
3. ПП загальної математики і ПП імітаційного моделювання	7800 – 8800

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить  $V_0$  в умовних машинних командах, трудомісткості визначати на основі табл.2.2

Таблиця.2.2

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера,  $K_k=0,7\div 0,8$ ):  $T_{ар} = 229 * 0,8 = 183,2$  (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{ТЗ} = T^a p \times L_1 \times K_H \quad (2.1)$$

$$T_{ПП} = T^a p \times L_2 \times K_H \quad (2.2)$$

$$T_{РП} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

$L_i$  – питома вага і-го етапу розробки (див. табл. 2.3.);

$K_H$  – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.4.);

$K_T$  – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.5.).

Таблиця 2.3.Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП.

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L <sub>1</sub> )	0,15	0,12	0,12
ТП (L <sub>2</sub> )	0,16	0,15	0,11
РП (L <sub>3</sub> )	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4. Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K <sub>н</sub>
А	Принципово нові ПП	1,75 – 1,2
Б	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПП маючий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.5. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПП типовими програмами, %	Значення K <sub>т</sub>
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ}=T_a \cdot L_1 \cdot K_n = 183,2 \cdot 0,12 \cdot 0,7 = 15,38 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість розробки технічного проекту

$$T_{ТП}=T_a \cdot L_2 \cdot K_n = 183,2 \cdot 0,11 \cdot 0,7 = 14,11 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки робочого проекту

$$T_{РП}=T_a \cdot L_3 \cdot K_n \cdot K_t = 183,2 \cdot 0,61 \cdot 0,7 \cdot 0,8 = 62,58 \text{ (люд/годин)} \quad (2.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання N<sub>ТЗ</sub>=3 (стр), розробка ТП N<sub>ТП</sub>=3(стр), розробка робочого проекту N<sub>РП</sub>=7 (стр), пояснювальна записка відповідно N<sub>ПЗ</sub>=36 (стр)

Розрахунок зведений у таблицю 2.6.

Таблиця 2.6. - Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин.		
1.ТЗ	$T_{P_{ТЗ}}=15,38$	$T_{кк}=0,7*N_{ТЗ}= 0,7*3=2,1$	$T_{нк}=0,15*N_{ТЗ}=0,15*3=0,45$
2.Розробка ТП	$T_{P_{ТП}}=14,11$	$T_{кк}=0,7*N_{ТП}=0,7*19=2,1$	$T_{нк}=0,15*N_{ТП}=0,15*19=0,45$
3.Розробка РП	$T_{P_{РП}}= 62,58$	$T_{кк}=0,7*N_{РП}=0,7*14=4,9$	$T_{нк}=0,15*N_{РП}=0,15*14=1,05$
4.Розробка ПЗ	$T_{P_{ПЗ}}=1,5*N_{ПЗ}=1,5*36 =54$	$T_{кк}=0,7*N_{ТЗ}=0,7*36=24,5$	$T_{нк}=0,15*N_{ПЗ}=0,15*36 =5,25$
Усього, в т.ч.:	186,87		
- на розробку	$\Sigma T_p=146,07$		
- контроль керівника		$\Sigma T_{кк}=33,6$	
- нормоконтроль			$\Sigma T_{нк}=7,2$

### 2.3 Розрахунок ціни програмного продукту.

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машино – години і витрати на розробку ПП. Розрахунок основної заробітної плати виконавців приведений у таблиці 2.7. Відповідно до статті 8 «Закону про Державний бюджет України на 2021» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2022 року - 6700 гривень; мінімальну погодинну тарифну ставку – 40,46 грн.

Таблиця 2.7. Розрахунок основної заробітної плати виконавців.

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	128,92	40,46	5216
2.Контроль керівника	33,6	100	3360
3.Нормоконтроль	7,2	100	7200
Усього	169,72	240,46	$\Sigma Z_o= 9596$

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.8

Таблиця 2.8.- Розрахунок матеріальних витрат на розробку ПП

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Продовження таблиці 2.8. Розрахунок матеріальних витрат на розробку				
Папір	Лист А4	65	4	260
Разом	-	-	-	$V_{mi} =$
Транспортно – заготівельні Витрати (10%)				$V_{тр\_з} = 0,1 \times V_{m1} = 0,1 * 260 = 26$
Усього				$V_m = V_{mi} + V_{тр\_з} = 286$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.9.

Таблиця 2.9. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	286	$V_m$ (див. табл. 2.8.)
2. Основна заробітна плата	9596	$Z_o$ (див. табл. 2.7.)
3. Додаткова заробітна плата	959,6	$Z_d = 0,1 \times Z_o = 9596 * 0,1$
4. Відрахування до єдиного фонду соціального внеску	2322,23	$V_{е.с.в.} = 0,22 \times (Z_o + Z_d) = 0,22 * (9596 + 959,6)$
5. Накладні витрати	3838,4	$V_{нак.} = 0,4 \times Z_o = 0,4 * 9596$
6. Повна собівартість	18635,63	$C_{пов} = V_m + Z_o + Z_d + V_{е.с.в.} + V_{нак.} =$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$P = (C_{п} * P) / 100 = (18635,63 * 10\%) / 100 = 18,64 \text{ грн}; \quad (2.4)$$

Де  $p$  – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$C_o = C_{п} + P = 18,64 + 18636,63 = 18655,27 \text{ грн}; \quad (2.5)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$C_p = C_o + ПДВ = 18655,27 + 18655,27 * 0,2 = 22386,32 \text{ грн}; \quad (2.6)$$

### 3 ОХОРОНА ПРАЦІ

У даному розділі ДП вирішується питання охорони праці програміста на стадії модифікації елементів оточення та персонажів у грі «Террарія».

#### 1 Аналіз небезпечних і шкідливих факторів, що впливають на програміста при розробці даного програмного комплексу

Робота з ПК супроводжується підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. На робочому місці користувача ПК виникають небезпечні та шкідливі фактори: підвищений рівень шуму, несприятливі мікрокліматичні умови, недостатній рівень освітленості, шкідливі речовини, підвищений рівень електромагнітних випромінювань радіочастот, висока напруга електричної мережі, статична електрика та інші.

#### 2 Гігієнічні вимоги до виробничого середовища

Вивчення умов праці на робочому місці користувача ПК є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів.

##### 2.1 Вимоги до приміщення

Слід відзначити, що площа одного робочого місця оператора ПК не повинна бути меншою за 6м<sup>2</sup>, а об'єм не менший за 20м<sup>3</sup>, Також приміщення повинно мати як штучне, так і природне освітлення. Воно має бути забезпечено системою опалення, кондиціонування повітря або припливно-витяжною вентиляцією.

Поверхня підлоги має бути рівною, неслизькою, з антистатичними властивостями.

					<b>РП 06.04.003.00 ДП ПЗ</b>	Арк
Вим.	Лист	№ документа	Підпис	Дата		52

## 2.2 Освітлення

Освітлення приміщення, у якому працює користувач персонального комп'ютера ( програміст), використовується змішане освітлення, тобто сполучення природного й штучного освітлення. Природне освітлення - здійснюється через вікна в зовнішніх стінах будинку. Штучне освітлення - використовується при недостатньому природному освітленні й здійснюється за допомогою двох систем: загального й місцевого освітлення. Для загального освітлення приміщення, де перебуває робоче місце програміста, використовуються газорозрядні лампи типу ЛД. Нормами для даних робіт встановлена необхідна освітленість робочого місця  $E_n=300$  лк (для робіт високої точності, коли найменший розмір об'єкта розрізнення дорівнює 0,3 – 0,5 мм).

## 2.3 Шум

У робочих приміщеннях основними джерелами акустичних шумів є шуми ПЕОМ. ЕОМ є також джерелами шумів електромагнітного походження (коливання елементів електромеханічних пристроїв під впливом змінних магнітних полів). Допустимий еквівалентний рівень шуму для робочого місця оператора складає 65 дБА. Під час виконання робіт з ПК у виробничих приміщеннях значення характеристик вібрації на робочих місцях мають не перевищувати допустимі відповідно до ДСанПіН 3.3.2.007-98, ДСН 3.3.6-039-99.

## 2.4 Вимоги до організації робочого місця працівника

Створення зручного робочого місця повинно бути одним з головних пріоритетів роботодавця. Створення зручного робочого місця повинно бути одним з головних пріоритетів роботодавця.

Для людини, що працює, потрібно створити санітарні умови, які б дали змогу їй плідно працювати, не перевтомлюючись та зберігати своє здоров'я. Для цього треба, щоб енергетичні витрати при праці компенсувалися відпочинком та умовами оточуючого середовища. Ці умови створюються забезпеченням для працюючого:

1. зручного робочого місця;

					<b>РП 06.04.003.00 ДП ПЗ</b>	Арк
Вим.	Лист	№ документа	Підпис	Дата		53

2. чистого повітря;
3. нормованої освітленості;
4. захисту від шуму та вібрацій;
5. захисту від дії шкідливих речовин та випромінювань;
6. робочим одягом та різними засобами індивідуального захисту;
7. побутовими приміщеннями та спеціальними службами, що призначені
8. створювати безпечні та нормальні умови праці.

Обладнання і організація робочого місця з ВДТ мають забезпечувати відповідність конструкцій всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру і особливостей трудової діяльності (ДСаПіН 3.3.2.-007-08). Конструкція робочого місця й взаємне розташування всіх його елементів (сидіння, органи керування, засобу відображення інформації) відповідають антропометричним, фізіологічним і психологічним вимогам, а також характеру роботи. Конструкція робочих меблів повинна забезпечувати можливість індивідуального регулювання відповідно росту працюючих для підтримки зручної пози.

Робочий стіл повинен бути пофарбований матовою фарбою. Дисплей розташований так, що його верхній край перебуває на рівні очей на відстані близько 70 см, що укладається в у припустимі рамки від 60 до 90 см

Робоче місце розташоване перпендикулярно віконним прорізам, це зроблено з тією метою, щоб виключити пряму й відбиту мерехтливність екрана від вікон і приладів штучного освітлення, якими є лампи накаливання. Частота мерехтіння екрана  $f_{\text{мер}}=100$  Гц, що відповідає умові  $f_{\text{мер}}>70$  Гц.

## 2.5 Мікроклімат

Найбільш значним фактором продуктивності й безпеки праці є виробничий мікроклімат, що характеризується температурою й вологістю повітря, швидкістю його руху, а також інтенсивністю радіації, і повинен відповідати ГОСТ 12.1.005-88 і СНиП 2.04.05-86, тому значення параметру мікроклімату повинно становити:

					<b>РП 06.04.003.00 ДП ПЗ</b>	Арк
Вим.	Лист	№ документа	Підпис	Дата		54

температуру повітря від 18-22 градусів Цельсію, вологість повітря від 40%-60%, та швидкість повітря від 0,1-0,2 м/с.

Для підтримки в приміщеннях нормального, що відповідає гігієнічним вимогам складу повітря, видалення з нього шкідливих газів, пару і пилу використають вентиляцію

Механічна вентиляція (кондиціонери вентилятори і т.ін.) залежно від напрямку руху повітряних потоків, може бути витяжною, нагнітаючої і нагнітаючовитяжною. При природній вентиляції (за допомогою вікон) повітря надходить у приміщення й віддаляється з нього внаслідок різниці температур, а також під дією вітру.

## 2.6 Електробезпека

Приміщення, де використовуються імпульсні джерела живлення відповідно до ОНТП24-86 і ПУЕ-87 відноситься до класу приміщень без підвищеної небезпеки поразки персоналу електричним струмом, оскільки відносна вологість повітря не перевищує 75%, температура не більш 35°C, відсутні хімічно агресивні середовища. Живлення електроприладів усередині приміщення здійснюється від двухфазної мережі з заземленою нейтралю напругою 220 В і частотою 50 Гц із використанням автоматів токового захисту. У приміщенні повинна бути застосована схема заземлення.

Заземлення повинно бути зроблено за допомогою гнучкого сплетеного мідного проводу діаметром порядку 1,5 мм<sup>2</sup>. Для зменшення значень напруг дотику і відповідних їм величин струмів, при нормальному й аварійному режимах роботи устаткування необхідно виконати повторне захисне заземлення нульового проводу. Відповідно до ГОСТ-12.2.007.0-75 все устаткування (крім ЕОМ - II клас) відноситься до I класу, воно має робочу ізоляцію відповідно до вимог ГОСТ 12.1.009-76. Підключення устаткування виконане відповідно до вимог ПБЕ та ПУЕ. Додаткових заходів по електробезпечності не потрібно.

					<b>РП 06.04.003.00 ДП ПЗ</b>	Арк
Вим.	Лист	№ документа	Підпис	Дата		55

### 3 Пожежна безпека

Коли від пожежі захищаються приміщення з персональними комп'ютерами, то слід враховувати специфіку вогнегасних речовин у вогнегасниках, які призводять під час гасіння до псування обладнання. Ці приміщення рекомендується оснащувати вуглекислотними вогнегасниками з урахуванням граничнодопустимої концентрації вогнегасної речовини.

Будинки, споруди, приміщення, технологічні установки повинні бути забезпечені первинними засобами пожежогасіння: вогнегасниками, ящиками з піском, покривалами з негорючого теплоізоляційного полотна, грубововняної тканини чи повсті, іншим пожежним інструментом, які використовуються для локалізації і ліквідації пожеж у початковій стадії їхнього розвитку.

Для зазначення місцезнаходження первинних засобів пожежогасіння слід установлювати відповідні знаки згідно з чинними державними стандартами. Знаки слід розміщувати на видних місцях на висоті 2-2,5 м від рівня підлоги як у середині, так і поза приміщеннями (у разі потреби).

Переносні вогнегасники повинні розміщуватися шляхом:

1. навішування на вертикальні конструкції на висоті не більше 1,5 м від рівня підлоги до нижнього торця вогнегасника і на відстані від дверей, достатній для її повного відчинення;
2. установлення в пожежні шафи пожежних кранів, або у спеціальні тумби;
3. навішування вогнегасників на кронштейни, розміщення їх у тумбах або пожежних шафах повинне забезпечувати можливість прочитання маркувальних написів на корпусі.

					<b>РП 06.04.003.00 ДП ПЗ</b>	Арк
Вим.	Лист	№ документа	Підпис	Дата		56

## ВИСНОВКИ

У моєму дипломному проекті я провів докладний опис предметної області гри "Terraria" і ретельно обґрунтував потребу в модифікації цієї гри. У рамках проекту я створив загальний технічний опис ігрових елементів оточення та персонажів і визначив методи модифікації, а також пов'язані з ними технології та інструментарій.

Я налаштував робоче середовище Visual Studio для зручного розроблення модифікацій, завантажив необхідний інструмент tModLoader та використовував графічний редактор для створення нових моделей.

Окрім цього, я спроектував різні модифікації, які впливають на ігрові елементи оточення. Зокрема, була створена нова локація, в якій завжди панує ніч, і було збільшено появу ворогів. Ніч починається у ній як тільки гравець входить у межі локації. Також, персонажі були модифіковані шляхом додавання нового торговця, з яким можна взаємодіяти по-різному, а точніше в нього можна купити корисні предмети і навіть просто поговорити. Також створення кам'яної істоти, яка безперервно атакуватиме гравця, якщо той наблизиться надто близько. Вона буде кидати у гравця блоки піску, тим самим наноситиме шкоду і усіляко заважати гравцю пересуватися по пустелі. Усі моделі були створені у графічному редакторі Piskel, що надає можливість створювати піксельні анімації.

Цей проект дозволив мені продемонструвати мої технічні навички та знання програмування, а також внести цікаві та захоплюючі зміни до ігрового процесу гри "Terraria".

					<i>РП 06. 04 000. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційний сайт TModLoader: [Website]. URL: <https://tmodloader.net/>
2. TModLoader Wiki:
  - Сайт: <https://github.com/tModLoader/tModLoader/wiki>
3. Terraria Community Forums: [Website]. URL: <https://forums.terraria.org/>
4. Beginning XNA 2.0 Game Programming From Novice to Professional
5. TModLoader GitHub: [Website]. URL: <https://github.com/tModLoader/tModLoader>
6. Terraria Fandom: [Website]. URL: <https://terraria.fandom.com/wiki/Desert>
7. C# docs: [Website]. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/>
8. Sean James - 2D Graphics with XNA Game Studio 4.0
9. Фленов М., Біблія C# 3-е видання., Київ, Фабула, 2016 р. 546 ст.
10. Aaron Reed - Learning XNA 4.0 (Game Development for the PC, Xbox 360, and Windows Phone 7)

					<i>РП 06. 04 000. ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

## ДОДАТОК А ЛІСТИНГ ПРОГРАМНОГО МОДУЛЯ

### Скрипт неігрового персонажа

```
using Terraria;
using Terraria.ID;
using Terraria.ModLoader;

namespace CustomMod.NPCs
{
    [AutoloadHead]
    public class CustomMerchant : ModNPC
    {
        public override bool Autoload(ref string name)
        {
            name = "Custom Merchant";
            return mod.Properties.Autoload;
        }

        public override void SetStaticDefaults()
        {
            DisplayName.SetDefault("Custom Merchant");
            Main.npcFrameCount[npc.type] = Main.npcFrameCount[NPCID.Merchant];
        }

        public override void SetDefaults()
        {
            npc.townNPC = true;
            npc.friendly = true;
            npc.width = 18;
            npc.height = 40;
        }
    }
}
```

```
npc.aiStyle = 7;
npc.defense = 15;
npc.lifeMax = 250;
npc.HitSound = SoundID.NPCHit1;
npc.DeathSound = SoundID.NPCDeath1;
npc.knockBackResist = 0.5f;
animationType = NPCID.Merchant;
}

public override string TownNPCName()
{
    return "Sam";
}

public override string GetChat()
{
    string[] chat = {
        "Купи в мене приманку! Не пожалієш",
        "Хочеш призвати Босса? В мене є те, що тобі потрібно!",
        "Викликати їх можна тільки після заходу сонця!",
        "Для перемоги над армією гоблінів треба лише колючі шари"
    };

    return Main.rand.Next(chat);
}

public override void SetupShop(Chest shop, ref int nextSlot)
{

```

```
    shop.item[nextSlot].SetDefaults(ItemID.SlimeCrown);  
  
    nextSlot++;  
  
    shop.item[nextSlot].SetDefaults(ItemID.SuspiciousLookingEye);  
  
    nextSlot++;  
  
    shop.item[nextSlot].SetDefaults(ItemID.BloodyTear);  
  
    nextSlot++;  
  
    shop.item[nextSlot].SetDefaults(ItemID.GoblinBattleStandard);  
}
```

```
public override bool CanTownNPCSpawn(int numTownNPCs, int money)  
{  
    return numTownNPCs > 1 && NPC.downedBoss1 &&  
Main.player[Main.myPlayer].inventory.Any(item => item.type ==  
ItemID.SuspiciousLookingEye || item.type == ItemID.WormFood || item.type ==  
ItemID.SlimeCrown);  
}
```

```
public override void TownNPCAttackStrength(ref int damage, ref float  
knockback)  
{  
    damage = 30;  
    knockback = 6f;  
}
```

```
public override void TownNPCAttackCooldown(ref int cooldown, ref int  
randExtraCooldown)  
{  
    cooldown = 30;
```

```

        randExtraCooldown = 10;
    }

    public override void TownNPCAttackProj(ref int projType, ref int attackDelay)
    {
        projType = ProjectileID.SpikyBall;
    }

    public override void TownNPCAttackProjSpeed(ref float multiplier, ref float
gravityCorrection, ref float randomOffset)
    {
        multiplier = 10f;
        randomOffset = 2f;
    }

    public override void NPCLoot()
    {
        Item.NewItem(npc.getRect(), ItemID.GoldCoin, 1);
    }
}
}
}

```

### **Скрипт нової генерації світу**

```

using Terraria;
using Terraria.ID;
using Terraria.ModLoader;
using Terraria.World.Generation;
using Microsoft.Xna.Framework;
using Terraria.GameContent.Generation;

```

```

namespace CustomDesert
{
    public class YourWorld : ModWorld
    {
        private bool isInEternalNightDesert = false;

        public override void ModifyWorldGenTasks(List<GenPass> tasks, ref float
totalWeight)
        {
            int desertIndex = tasks.FindIndex(genpass =>
genpass.Name.Equals("Desert"));

            if (desertIndex != -1)
            {
                tasks.RemoveAt(desertIndex);

                tasks.Insert(desertIndex, new PassLegacy("EternalNightDesert",
GenerateEternalNightDesert));
            }
        }

        public override void PostUpdate()
        {
            bool isInDesert = false;

            Rectangle playerRect = new Rectangle((int)Main.LocalPlayer.position.X / 16,
(int)Main.LocalPlayer.position.Y / 16, Main.LocalPlayer.width / 16,
Main.LocalPlayer.height / 16);

            for (int i = playerRect.X; i < playerRect.X + playerRect.Width; i++)

```

```

{
    for (int j = playerRect.Y; j < playerRect.Y + playerRect.Height; j++)
    {
        if (Main.tile[i, j].active() && Main.tile[i, j].type == TileID.Sand)
        {
            isInDesert = true;
            break;
        }
    }
}

if (isInDesert && !isInEternalNightDesert)
{
    Main.dayTime = false;
    Main.time = 0;
    Main.NewText("Пустеля охопила ніч!", Color.Purple);
    isInEternalNightDesert = true;
}
else if (!isInDesert && isInEternalNightDesert)
{
    isInEternalNightDesert = false;
}
}

private void GenerateEternalNightDesert(GenerationProgress progress)
{
    for (int i = 0; i < Main.maxTilesX; i++)
    {
        for (int j = 0; j < Main.maxTilesY; j++)

```



```
public class Golem : ModNPC
{
    private int attackTimer = 0;
    private int attackCooldown = 120;

    public override void SetStaticDefaults()
    {
        DisplayName.SetDefault("Golem");
        Main.npcFrameCount[npc.type] = Main.npcFrameCount[NPCID.Golem];
    }

    public override void SetDefaults()
    {
        npc.width = 86;
        npc.height = 92;
        npc.damage = 50;
        npc.defense = 30;
        npc.lifeMax = 500;
        npc.HitSound = SoundID.NPCHit1;
        npc.DeathSound = SoundID.NPCDeath1;
        npc.value = Item.buyPrice(0, 5, 0, 0);
        npc.knockBackResist = 0.5f;
        npc.aiStyle = 3;
        aiType = NPCID.Golem;
        animationType = NPCID.Golem;
    }

    public override float SpawnChance(NPCSpawnInfo spawnInfo)
    {
```

```

        return SpawnCondition.Desert.Chance * 0.1f;
    }

    public override void AI()
    {
        Player player = Main.player[npc.target];

        attackTimer++;
        if (attackTimer >= attackCooldown)
        {
            Vector2 velocity = Helper.VelocityToPoint(npc.Center, player.Center, 8f);
            int sandBlockID = TileID.Sand;
            int sandBlockDamage = 10;

            Projectile.NewProjectile(npc.Center.X,    npc.Center.Y,    velocity.X,
            velocity.Y, ProjectileID.Boulder, sandBlockDamage, 1f, Main.myPlayer);
            attackTimer = 0;
        }
    }

    public override void NPCLoot()
    {
        if (Main.rand.NextBool(3))
            Item.NewItem(npc.getRect(), ItemID.SandBlock, Main.rand.Next(10, 20));
    }
}

```

# ДОДАТОК Б. ЛІСТИНГ ГРАФІЧНОГО МАТЕРІАЛУ

## Модифікація ігрових елементів оточення та персонажів для комп'ютерної гри «Террарія»

ВОЛОХ МАКСИМ 4РП-06

## Опис гри «Террарія» та її ігрових елементів оточення та персонажів

Terraria — це популярна пригодницька пісочниця, розроблена Re-Logic. Він пропонує унікальне поєднання дослідження, створення предметів, будівництва та бою в двовимірному процедурно створеному світі. У Terraria гравцям доручено досліджувати навколишнє середовище та взаємодіяти з ним, щоб збирати ресурси, перемагати ворогів і відкривати приховані скарби.



## Мета виконання модифікації гри

Мета виконання модифікації гри, яку часто називають «модифікацією», може відрізнятись залежно від намірів розробника. Основними є:

- ▶ Виправлення помилок і проблем;
- ▶ Покращення ігрового процесу;
- ▶ Створення нових ігрових режимів.

## Огляд інструментів та програмного забезпечення для виконання модифікації

Коли мова заходить про модифікацію Terraria, одним із популярних інструментів, який надає широкі можливості модифікації, є TModLoader. TModLoader — це платформа модування з відкритим кодом, яка безпосередньо інтегрується з грою Terraria, дозволяючи розробникам створювати та поширювати власні моди за допомогою мови програмування C#



## Порівняння інших модифікацій

	MyCustomMod	CalamityMod	ThoriumMod
Новий вороги	Так	Так	Так
Новий торговець	Так	Так	Так
Новий головний ворог	Ні	Так	Так
Нова локація	Так	Так	Ні

## Що нового?

У цій модифікації були представлені три основні елементи гри, це:

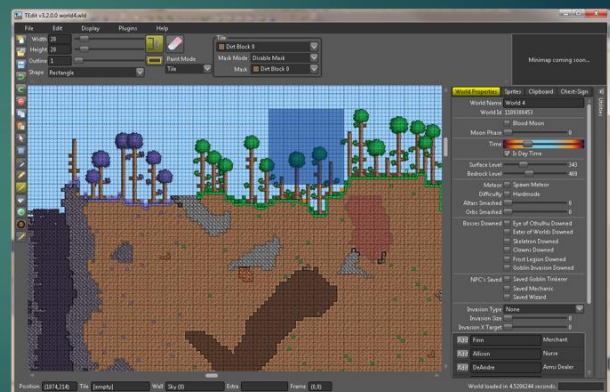
1. Новий неігровий персонаж;
2. Новий ворог;
3. Нова локація.

# Процес розробки модифікації ігрових елементів оточення

1. Концептуалізація та планування
2. Інструменти модифікації
3. Створення текстур
4. Інтеграція та балансування ігрового процесу

## Робота у TEdit

Потім надав блокам можливість генеруватися у світі та позначив місце генерація за допомогою TEdit.



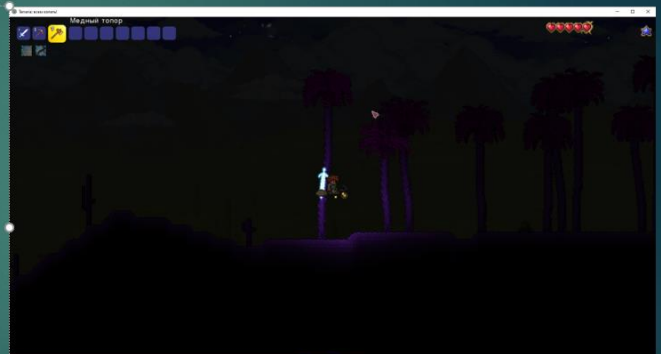
## Патерн блоків

Спочатку був завантажений весь патерн блоків піску, визначившись з виглядом майбутньої локації, перемалював патерн.



## Результат виконаної роботи

Результатом стала нова локація зі своїми особливими властивостями.



# Процес розробки модифікації ігрових персонажів

Концептуалізація та планування

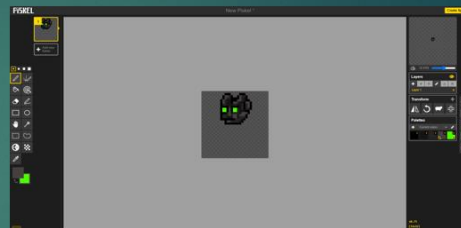
Інструменти модифікації

Здібності та поведінка

Створення текстур

Балансування та тестування

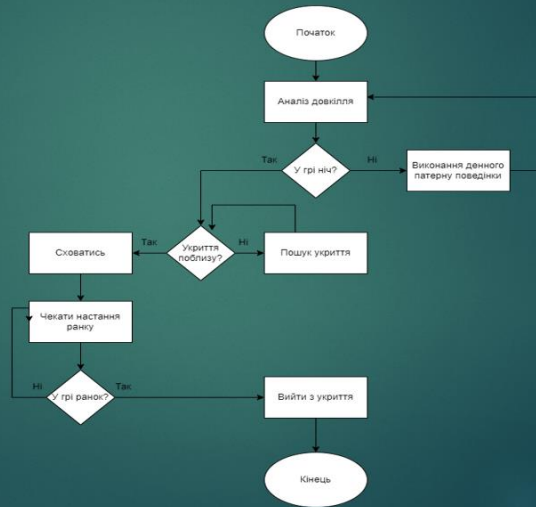
## Створення моделей персонажів



Спочатку було визначено вигляд майбутніх персонажів.  
Моделі були створені у графічному редакторі Piskel.

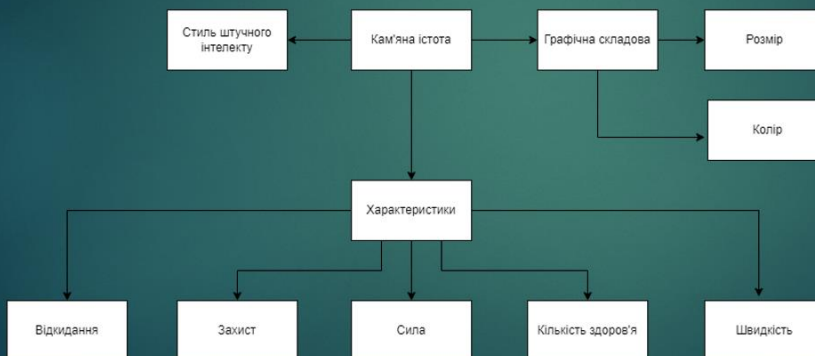
# Визначення поведінки персонажів

На даному алгоритмі зображена поведінка торговця у залежності від часу доби



# Схема кам'яної істоти

На даній схемі визначені усі складові кам'яної істоти



# Результат виконаної роботи

Результатом стали нова локація та нові персонажі



**ДОЗВІЛ  
НА РОЗМІЩЕННЯ  
ВИПУСКНОГО ДИПЛОМНОГО ПРОЕКТА  
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

*Волох Максим Андрійович,*  
здобувач освіти гр. 4РП-06, та

*Джабраїлов Дмитро Володимирович,*  
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускного дипломного проекту молодшого спеціаліста на тему:

*«Модифікація ігрових елементів оточення та персонажів для комп'ютерної гри «Террарія» (автор роботи – Волох М.А., керівник роботи – Джабраїлов Д.В.)*

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

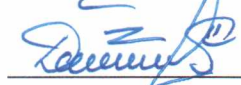
Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець



/ Волох М.А. /

Керівник



/ Джабраїлов Д.В./

« 09 » 06 20 23 р.

Ім'я користувача:  
Наталія Вікторівна Копусь

ID перевірки:  
1015556733

Дата перевірки:  
12.06.2023 10:49:25 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
12.06.2023 10:55:12 EEST

ID користувача:  
100011688

Назва документа: 4РП-06 Волох М.А.

Кількість сторінок: 53 Кількість слів: 9034 Кількість символів: 67900 Розмір файлу: 8.20 MB ID файлу: 1015208492

## 8.98% Схожість

Найбільша схожість: 2.28% з Інтернет-джерелом (<http://elcat.pnpu.edu.ua/docs/%d0%9c%d0%90%d0%a2%d0%95%d0%..>)

8.98% Джерела з Інтернету 538

Сторінка 55

Не знайдено джерел з Бібліотеки

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 31

Ім'я користувача:  
Наталія Вікторівна Копусь

ID перевірки:  
1015556733

Дата перевірки:  
12.06.2023 10:49:25 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
12.06.2023 10:55:12 EEST

ID користувача:  
100011688

Назва документа: 4РП-06 Волох М.А.

Кількість сторінок: 53 Кількість слів: 9034 Кількість символів: 67900 Розмір файлу: 8.20 MB ID файлу: 1015208492

## 8.98% Схожість

Найбільша схожість: 2.28% з Інтернет-джерелом (<http://elcat.pnpu.edu.ua/docs/%d0%9c%d0%90%d0%a2%d0%95%d0%95>..

8.98% Джерела з Інтернету 538 ..... Сторінка 55

Не знайдено джерел з Бібліотеки

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 31

## РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти  
відділення комп'ютерних систем

Волоха Максима Андрійовича

(прізвище, ім'я та по батькові)

Спеціальність 121 “ Інженерія програмного забезпечення ”

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Джабраїлов Дмитро Володимирович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Модифікація ігрових елементів оточення та персонажів для комп'ютерної гри «Террарія»

Обсяг розрахунково-пояснювальної записки 73 сторінок

Обсяг графічної (презентаційної) частини 15 аркушів (слайдів)

### ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню Представлений на рецензію дипломний проект повністю відповідає меті проектування та технічному завданню. Тематика дипломного проекту є актуальною для своєї галузі та присвячена питанням створення ігрових модифікацій в цілому та для комп'ютерної гри «Террарія» зокрема.

б) характеристика виконання кожного розділу дипломного проекту (роботи) Дипломний проект складається зі вступу, трьох розділів, висновків, переліку використаних джерел. У технологічному розділі розглянуті питання проблематики модифікації, сформовано вимоги до модифікації згідно до теми дипломного проекту та завданню, виконано проектування основних аспектів розробляємої модифікації. За допомогою відповідного програмного забезпечення реалізовані всі намічені зміни до ігрового процесу

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи) Графічна частина виконана на достатньо високому рівні у вигляді презентації із використанням офісного пакету Microsoft PowerPoint та Visio. Пояснювальна записка виконана акуратно та у відповідності до норм оформлення документів із використанням офісного пакету Microsoft Word. Загальна якість виконання документації – добра, академічного плагіату у роботі не виявлено

г) перелік позитивних якостей дипломного проекту (роботи) \_\_\_\_\_

1. Детально описано процес виконання модифікації;

2. Виконано проектування елементів модифікації із поясненнями на схемах;

3. Розроблено функціонуючу модифікацію для гри «Террарія» за допомогою відповідних іструментів розробки.

д) основні недоліки дипломного проекту (роботи) \_\_\_\_\_

1. Модифікація виконує часткову заміну ігрових елементів;

2. Не вистачає повного опису всіх модифікованих елементів гри

Оцінка розрахункової частини \_\_\_\_\_ Добре

Оцінка графічної частини \_\_\_\_\_ Добре

Загальна оцінка \_\_\_\_\_ Добре

Прізвище, ім'я, по батькові рецензента \_\_\_\_\_ Царьов Роман Юрійович

Місце роботи і посада рецензента \_\_\_\_\_ Державний університет інтелектуальних технологій і зв'язку, старший викладач кафедри комп'ютерної інженерії та інформаційних систем

Підпис: \_\_\_\_\_

« 16 » серпня 2023 р.

ПІДПИС ПОСВІДОЧУЄ  
НАЧАЛЬНИК ВІДДІЛУ  
КАДРІВ АУІТЗ



**ВІДГУК**

керівника на дипломний проект здобувача (здобувачки) освіти  
відділення комп'ютерних систем

Волоха Максима Андрійовича

(прізвище, ім'я та по батькові)

Спеціальність: 121 "Інженерія програмного забезпечення"

Освітня програма: «Розробка програмного забезпечення»

Тема дипломного проекту: Модифікація ігрових елементів оточення та персонажів для комп'ютерної гри «Террарія»

**ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ**

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка містить 73 сторінок. У пояснювальній записці виконано опис предметної області, способи модифікації гри «Террарія». Проведено проектування та модифікація елементів оточення та персонажів гри. Графічна частина складається з 15 слайдів мультимедійної презентації, які передбачені технічним завданням. Якість виконання пояснювальної записки та графічної частини добра, розробку виконано в повному обсязі.

б) самостійність роботи над проектом: Протягом всього строку дипломного проектування та переддипломної практики здобувач освіти Волох М.А. поступово та послідовно виконував всі етапи розробки. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника.

в) теоретична підготовка випускника (випускниці): Здобувач освіти Волох М.А. під час роботи над дипломним проектом вивчив достатню кількість літературних джерел та матеріалів за даною тематикою.

Вважаю, що теоретична підготовка дипломника добра і він готовий до захисту дипломного проекту

г) вміння розв'язувати виробничі та конструкторські питання \_\_\_\_\_  
Під час дипломного проектування здобувач освіти Волох М.А. мав змогу  
самостійно приймати рішення з реалізації модифікацій ігрових елементів та  
персонажів гри, та показав вміння організовано працювати над поставленим  
завданням, складати схеми та проводити розробку коду за допомогою  
актуальних для теми комп'ютерних програмних засобів.

Оцінка розрахункової частини \_\_\_\_\_ (5) Відмінно

Оцінка графічної частини \_\_\_\_\_ (4) Добре

Загальна оцінка \_\_\_\_\_ (5) Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту \_\_\_\_\_  
Джабраїлов Дмитро Володимирович

Місце роботи і посада керівника дипломного проекту \_\_\_\_\_  
ВСП "Одеський технічний фаховий коледж ОНТУ", викладач  
комісії комп'ютерних технологій та програмної інженерії

Підпис \_\_\_\_\_

«09» 06 2023 р.