

Міністерство освіти і науки України  
Харківський фаховий радіотехнічний коледж  
Циклова комісія “Комп’ютерних наук”

## КУРСОВА РОБОТА

з дисципліни

«Конструювання програмного забезпечення»

на тему:

«Розробка 3D гри "Modern Soldier" у жанрі аркади за допомогою Unity»

Виконав:

студент групи ПІ – 410

Сербін Ян Євгенович

Керівник роботи

к.т.н. Федосєєва А.О.

Харків 2024

ЗАЛІКОВИЙ ЛИСТ  
ХАРКІВСЬКИЙ ФАХОВИЙ РАДІОТЕХНІЧНИЙ КОЛЕДЖ

Циклова комісія КОМП'ЮТЕРНИХ НАУК  
Спеціальність 121 «Інженерія програмного забезпечення»

Курс – 4 Група - ПП-410 Семестр – 7

КУРСОВА РОБОТА

студента Сербіна Яна Євгеновича

на тему: «Розробка 3D гри "Modern Soldier" у жанрі аркади за допомогою Unity»

ДОПУЩЕНИЙ ДО ЗАХИСТУ

Керівник роботи \_\_\_\_\_/к.т.н. Федосєєва Аліна Олександрівна/

ОЦІНКА \_\_\_\_\_

Члени комісії:

\_\_\_\_\_/Федосєєва А.О./

\_\_\_\_\_/Лісаєв Д.В./

\_\_\_\_\_/Пироженко В.В./

Дата захисту: «17» січня 2024 р.

## ЗМІСТ

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	2
2. ПОБУДОВА ПРОФІЛЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .	9
3. ПОБУДОВА ТЕХНІЧНОГО ЗАВДАННЯ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ «MODERN SOLDIER» .....	12
4. ПЛАНУВАННЯ ТЕСТУВАННЯ.....	16
5. ВИКОНАННЯ ТЕСТУВАННЯ .....	21
6. ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	23
7. ВИСНОВКИ .....	34
8. СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	34

## ВСТУП

Комп'ютерні ігри актуальні на даний момент часу тому що вони приносять гроші, дуже багато ігор створюються для того, щоб заробити на людях які в них грають, взагалом це роблять компанії, які випускають ігри безкоштовними і вносять в них рекламу, систему апгрейду за гроші, преміум акаунт, який дозволяє швидко проходити гру, тощо. Ігри мають настільки істотний вплив на сучасне суспільство, що останнім часом з'являється стійка тенденція до гейміфікації для неігрового прикладного програмного забезпечення. Існують дуже багато різних аркад ігор, але вони схожі одна на одну, взагалом їх випускають конвеєром на мобільні пристрої, для того щоб була конкуренція, тому в них дуже нудно грати.

З усього цього можна зробити висновок, що комп'ютерні ігри щільно вплили до нашого нинішнього життя. Сфера їх використання за останні роки продовжує зростати, ігри використовують не тільки для розваг, але й для: проведення наукових досліджень, навчання та прибутку.

На цій курсовій роботі ми повинні розробити повне функціонуюче програмне забезпечення, а саме, гру аркаду-шутер, де головний герой нищить ворогів, за допомогою зброї, яку купив у торговця зброєю.

Гра повинна бути дуже цікавою та захоплюючою. Усього в ній буде 2 зони: безпечна зона та зона війни, зброя, торговець. На кожній зоні різний ландшафт та механіка, вороги з'являються через деякий час.

## 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

На сьогоднішній день ігри в нашому світі всюди. Розвага – одна з функцій комп'ютера. Зараз дуже багато ігор з реалістичною графікою. І дуже багато людей грає в них. Я покажу статистику, яка показує скільки людей грають у відеоігри [1<https://marketsplash.com/ru/statistika-vidieoighr>]. Результати розрахунків наведені на рисунку 1.1.

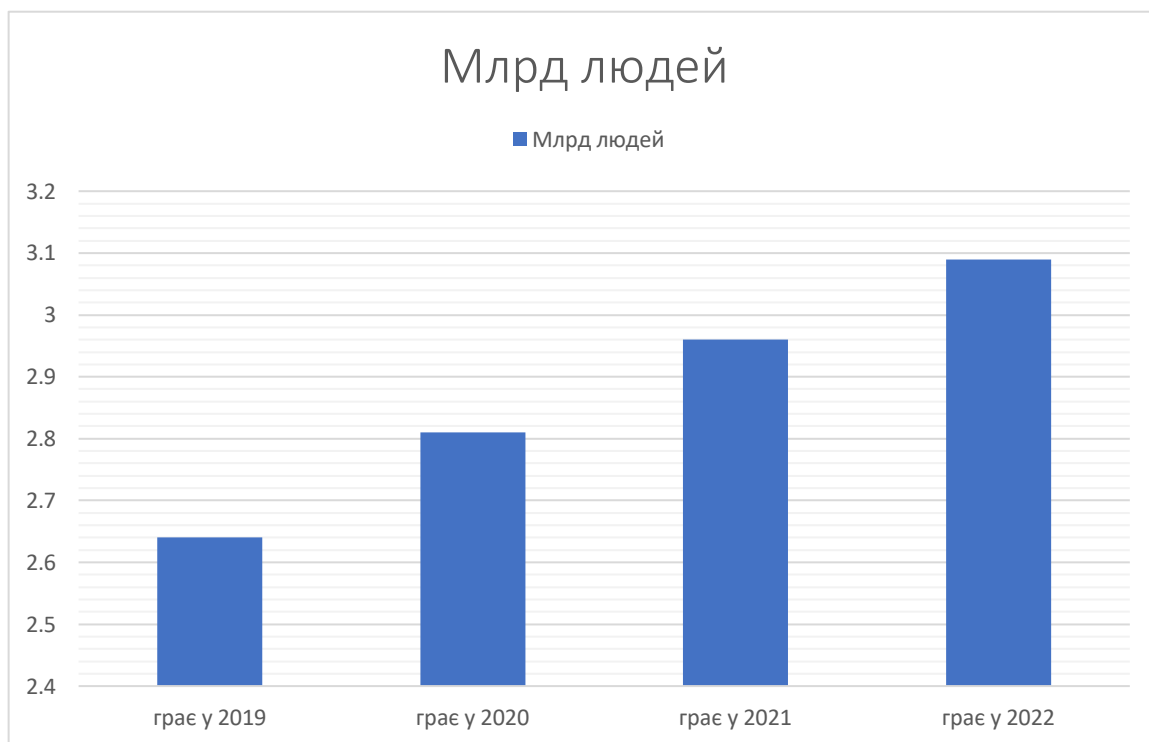


Рисунок 1.1 – Результати розрахунків, джерело:

У світі багато жанрів ком'ютерних ігор але виділяють основні такі:

- Шутер;
- Стратегія;
- Рольова гра;
- Симулятор;
- MMORPG;
- MOBA;
- Souls-like;
- Інші жанри.

Аркада (arcade) - це жанр комп'ютерних ігор, що має на увазі короткий за часом інтенсивний ігровий процес. Наприклад, літаючі літачки, тетріс, ігри в стилі "зібрати кульки в ряд" тощо.

При цьому важливо розуміти, що аркада часто поєднана з іншими жанрами. Наприклад, бувають гонки, симулятори, шутери, платформери, файтинг і так далі. У цьому сенсі цей жанр лише говорить геймерам щось у стилі "це буде швидко та весело".

Також варто відзначити, що у аркадних ігор бувають свої особливості. Наприклад, у іграшок у стилі "збери кульки в ряд" може взагалі не бути кінцівки як такої, хіба що користувач закриває гру. Тому з одного боку таку аркаду можна назвати "нескінченною". Але з іншого боку навряд чи знайдеться людина, яка, скажімо, 10 годин до ряду збиратиме кульки. Таких ігор дуже багато, загалом вони зустрічаються у мобільних додатках і вони одна від одної нічим не відрізняються, тому в них мало хто дуже багато грає, але ці ігри згодяться для того щоб добре провести час у метро або у будь-якому іншому транспорті, пограв 5-10 хвилин і стало нудно.

Є аркади де головний герой повинен знищити ворогів за допомогою кулаків, спец прийомів, скилів, зброї, техніки, тощо. Такі ігри називаються файтингами-аркадою, шутерами-аркадою, стратегією-аркадою, а також ММО-аркада. Головна мета таких ігор якомога більше часу ігрок повинен знаходитись у грі, а якщо в цій грі є мультиплеєр, то в цю гру буде дуже багато людей грати, тому що ботів-ворогів нудно дуже багато часу знищувати, а система онлайн-змагань в таких іграх ключ к онлайну. Також, якщо ця гра набула великої популярності можна додавати наприклад гарну розмальовку на зброю, за яку ігроку буде не соромно віддати гроші, таким чином робиться прибуток за гру.

У нашому випадку я зроблю аркаду-шутер, де головний герой буде у безпечній зоні закуплятися а у зоні війни воювати проти ворогів, на даному етапі розробки я планую зробити війну проти монстрів нескінченною, 4 типів зброї: кулемет, автоматична гвинтівка, граната, пістолет-кулемет.

Проектування нашої аркади-шутеру треба розпочати з діаграми, вона допоможе детальніше зрозуміти яким чином додаток буде функціонувати. Для побудови діаграми використовують мову UML.

Аналог нашої гри, наприклад Fallout 4.

UML – це уніфікована графічна мова моделювання для опису, візуалізації, проектування та документування об'єктно-орієнтованих систем. Моделі на UML використовуються на всіх етапах життєвого циклу ПЗ, починаючи з аналізу ринку і закінчуючи супроводом системи.

Щоб описати взаємодію між користувачем та нашим додатком слід використати діаграму «Використання» Діаграма використання (прецедентів) – це найбільш загальне представлення функціональності системи. Ця діаграма дозволяє переглянути всі можливі дії користувача в грі.

Діаграма використання представлена на рисунку 1.2.

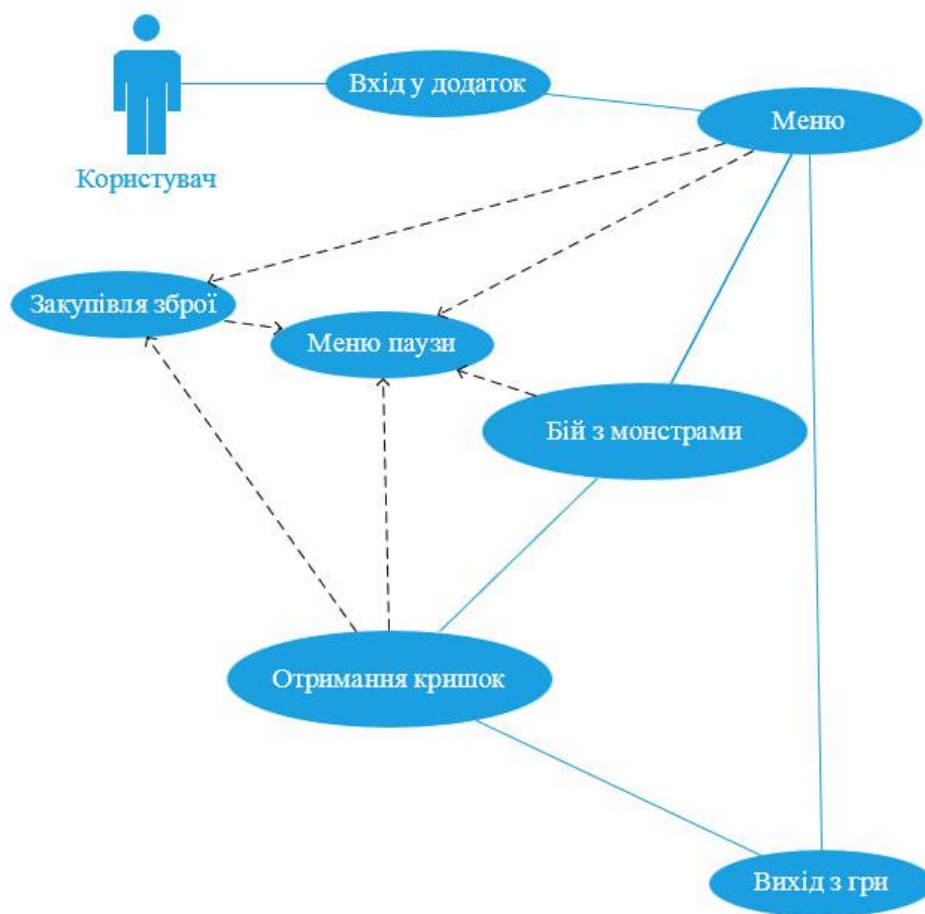


Рисунок 1.2 – Діаграма використання.

Діаграма послідовності вказує на спосіб опису поведінки системи на основі вказівки послідовності переданих повідомлень (впорядковані у часі), рисунок 1.3.



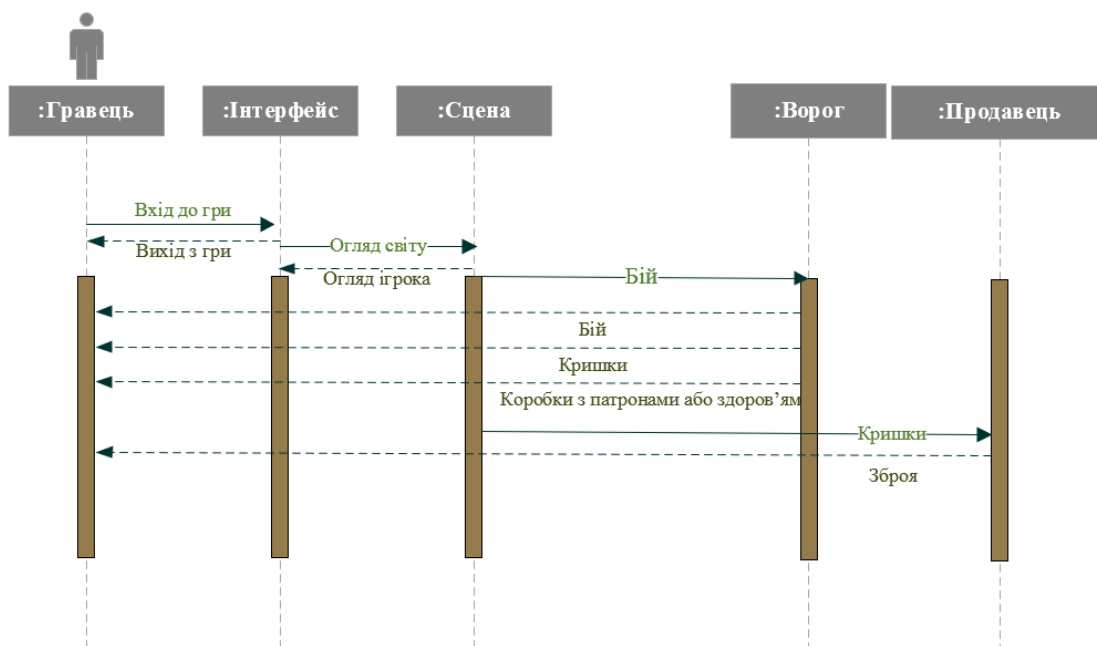


Рисунок 1.3 – Діаграма послідовності.

Для визначення основних рис аркади треба провести проектування. Щоб відобразити повну функціональність ПЗ використовують метод SADT та нотація IDEF0.

IDEF0 – методологія функціонального моделювання і графічного опису процесів, призначена для формалізації і опису бізнес-процесів.

Особливістю IDEF0 є її акцент на ієрархічне представлення об'єктів, що значно полегшує розуміння предметної області. В IDEF0 розглядаються логічні зв'язки між роботами, а не послідовність їх виконання в часі. IDEF0-діаграми (Три рівні деталізації) представлено на рисунках 1.4, 1.5, 1.6.

Рисунок 1.4 – IDEF0-діаграма A0 рівень

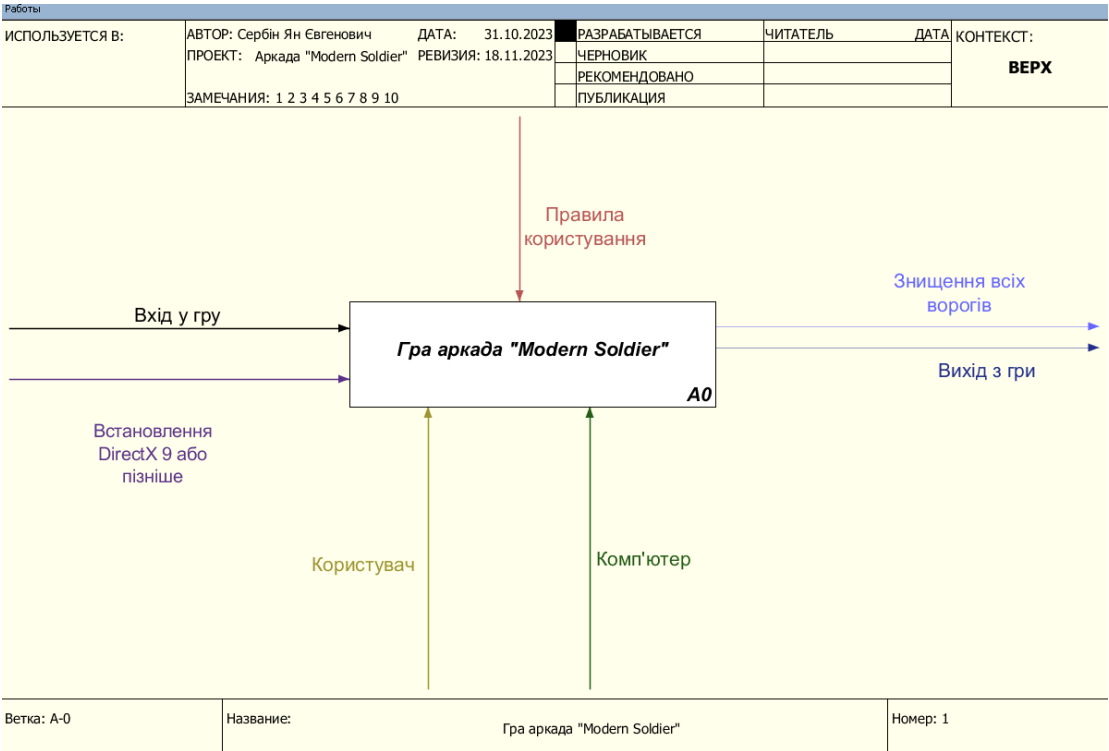


Рисунок 1.5 – IDEF0-діаграма A1 рівень

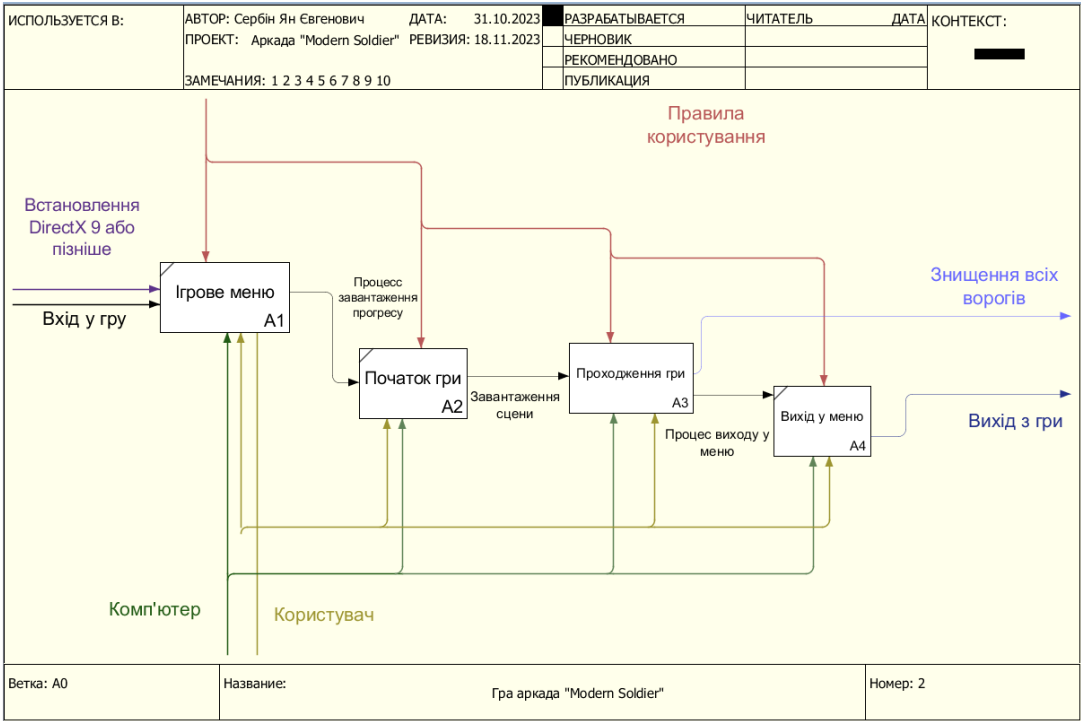
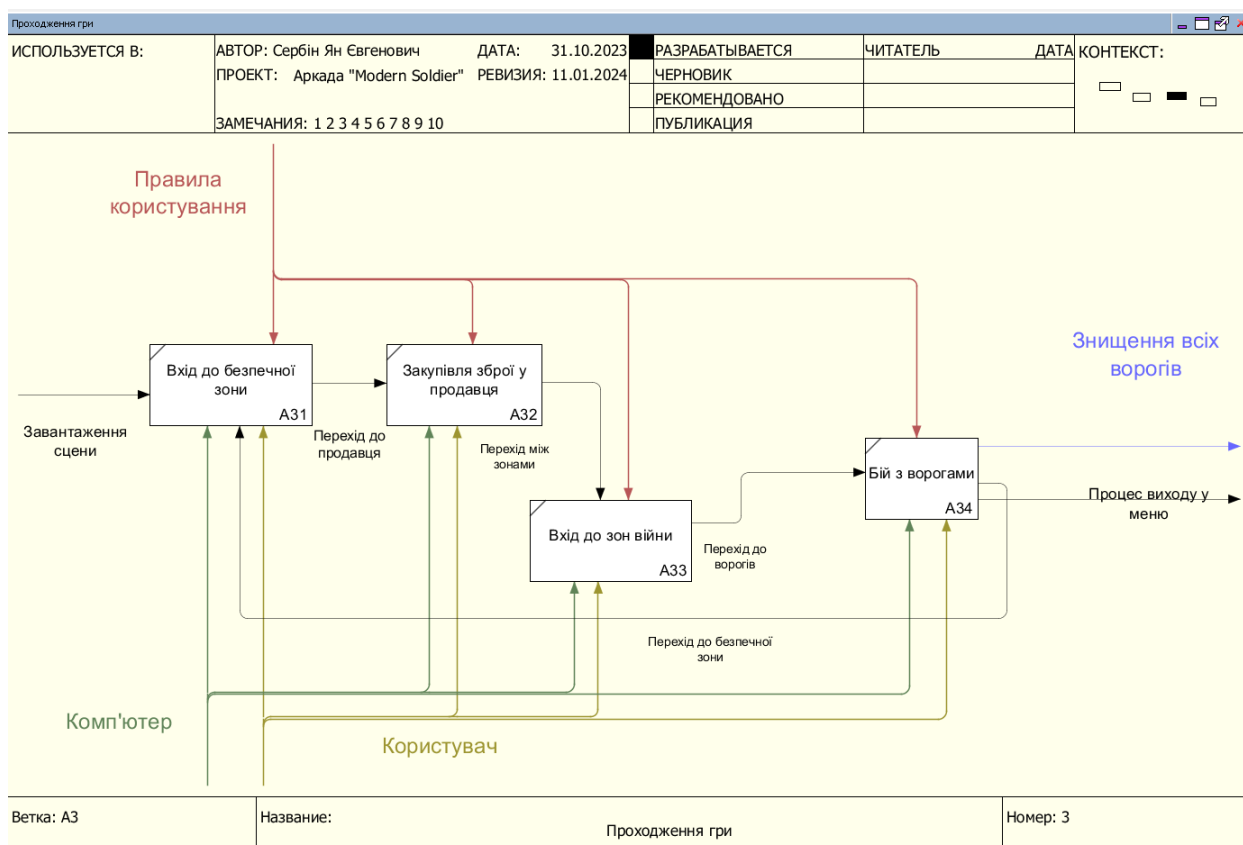


Рисунок 1.6 – IDEF0-діаграма A2 рівень



Отже, тепер після побудови IDEF0 діаграми аркадної гри, можемо перейти до побудови профілю вимог до програмного забезпечення.

Висновки за розділом 1:

- Проведено аналіз предметної області, перегляд існуючих 3D аркад, історія їх виникнення, тощо.
- Побудовано діаграму використання та послідовності.
- Проведено проектування функціоналу за допомогою ПЗ «Ramus» та побудовано IDEF0 діаграма.

## 2. ПОБУДОВА ПРОФІЛЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У рамках проектування ПЗ відбувається створення вимог до програмного забезпечення і будується їх нормативний профіль, який являє собою перелік функціональних і нефункціональних вимог.

У таблиці 2.1 представлено нормативний профіль вимог до ігрового додатку.

Таблиця 2.1 – Нормативний профіль вимог до ПЗ

Вимоги	Повнота	Необхідність	Однозначність	Перевіряємість	Здійсненність	Коректність
1	2	3	4	5	6	7
Бізнес вимоги						
Відсутність реклами	+	+	+	+	+	+
Вимоги користувача						
Вхід у додаток	+	+	+	+	+	+
Вихід з додатку	+	+	+	+	+	+
Кидання гранати	+	+	+	+	+	+
Рух у просторі	+	+	+	+	+	+
Стрибок	+	+	+	+	+	+
Подвійний стрибок	-	-	-	-	+	-
Поповнення здоров'я	+	+	+	+	+	+
Збір кришок, боеприпасів	+	+	+	+	+	+
Зони безпеки, війни	+	+	+	+	+	+
Атака на ворога	+	+	+	+	+	+
Смерть гравця	+	+	+	+	+	+
Смерть ворога	+	+	+	+	+	+
Респавн	+	+	+	+	+	+
Можливість присідати	+	+	+	+	+	+
Торгівля	+	+	+	+	+	+
Розмови	-	-	-	-	-	-

Вид від 1 та 3 обличчя	+	+	+	+	+	+
------------------------	---	---	---	---	---	---

Продовження таблиці 2.1

Функціональні вимоги						
1	2	3	4	5	6	7
Створення інтерфейсу	+	+	+	+	+	+
Створення меню та меню паузи	+	+	+	+	+	+
Створення кнопки входу у додаток	+	+	+	+	+	+
Створення кнопки виходу з додатку	+	+	+	+	+	+
Створення скидання ігрового процесу	+	+	+	+	+	+
Створення кнопки руху	+	+	+	+	+	+
Створення кнопки присідання	+	+	+	+	+	+
Створення кнопки переключення поглядів 1 та 3 обличчя	+	+	+	+	+	+
Створення кнопки стрибка	+	+	+	+	+	+
Створення кнопки подвійного стрибка	-	-	-	-	+	-
Розробка функції поповнення здоров'я	+	+	+	+	+	+
Розробка функції збору кришок, боєприпасів, аптечок	+	+	+	+	+	+
Розробка фізики пострілів	+	+	+	+	+	+
Розробка атаки на ворога	+	+	+	+	+	+
Розробка функції смерті гравця	+	+	+	+	+	+
Розробка функції смерті ворога	+	+	+	+	+	+
Створення респауну після смерті	+	+	+	+	+	+
Розробка функції вибору зброї у продавця	+	+	+	+	+	+
Розробка покупки зброї у продавця	+	-	+	+	+	+
Створення можливості перегляду товарів	+	+	+	+	+	+
Бізнес правила						
Завантаження з платформ платне	+	+	+	+	+	+
Атрибути якості						
Юзабіліті	+	+	+	+	+	+
Стабільність	+	+	+	+	+	+
Надійність	+	+	+	+	+	+
Ремонтопридатність	+	+	+	+	+	+
Обмеження						
Одна мова додатку	+	+	+	+	+	+
Заборона самопідриву персонажа зброєю	+	+	+	+	+	+
Заборона стрільби у безпечній зоні	+	+	+	+	+	+
Заборона стрільби без наявності боєприпасів	+	+	+	+	+	+
Заборона покупки товарів, коли кришок немає	+	+	+	+	+	+
Заборона підняття аптечки, коли здоров'я повне	+	+	+	+	+	+

**Висновки за розділом 2:**

– Виходячи з створених вимог ми сформуємо технічне завдання для нашого програмного забезпечення; – Сформульовані функціональні та нефункціональні вимоги.

### 3. ПОБУДОВА ТЕХНІЧНОГО ЗАВДАННЯ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ «MODERN SOLDIER»

У рамках проектування програмного забезпечення відбувається оголошення технічного завдання, у рамках якого потрібно побудувати технічне завдання до нашого ПЗ, сформулювати вимоги до складу та параметри технічних засобів, обрати стадії та етапи розробки та сформулювати умови експлуатації. У таблиці 3.1 представлено технічне завдання.

Таблиця 3.1 – Технічне завдання

Розділ	Підрозділ	Опис
1	2	3
Вступ	Назва програми	«MODERN SOLDIER»
	Предметна область	Гра для розваги дітей
Вимоги до ПЗ	Вимоги до функціональних характеристик	Створення інтерфейсу Створення меню Створення кнопки входу у додаток Створення кнопки виходу з додатку Створення кнопки кидання гранати Створення кнопки руху, присідання Створення кнопки стрибка Розробка функції поповнення здоров'я Розробка функції збору кришок, боєприпасів, аптечок Розробка фізики пострілів Розробка атаки на ворога Розробка функції смерті гравця Розробка функції смерті ворога Створення респауну після смерті Розробка функції вибору зброї у продавця Розробка покупки зброї у продавця Створення можливості перегляду товарів

## Продовження таблиці 3.1

1	2	3
	Вимоги до нефункціональних характеристик	Завантаження з платформ платне Юзабіліті Стабільність, Надійність, Ремонтопридатність Одна мова додатку Заборона самопідриву персонажа зброєю Заборона стрільби у безпечній зоні Заборона стрільби без наявності боєприпасів Заборона покупки товарів, коли кришок немає Заборона підняття аптечки, коли здоров'я повне
	Вимоги до складу та параметрам технічних засобів	Процесор з тактовою частотою не нижче 2.2 ГГц; Оперативний запам'ятовуючий пристрій – 4048 Мб; 500 Мб вільного місця на жорсткому диску
Стадії та етапи розробки		01.10.2023 – 01.01.2024 1) Аналіз предметної галузі 01.10.2023 – 10.5.2023 2) Створенні UML та IDEF0 – діаграм 5.10.2023 – 10.10.2023 3) Побудова профілю вимог до додатку 10.10.2023 – 20.10.2023 4) Розробка технічного завдання 20.10.2023 – 01.11.2023 5) Створення програми 01.11.2023 – 01.12.2023 6) Тестування 01.12.2023 – 29.12.2023



Продовження таблиці 3.1

1	2	3
		7) Завантаження програми у загальний доступ 29.12.2023 – 01.01.2024
Умови експлуатації	Види обслуговування	Додавання нових ворогів, персонажів, зброї. Виправлення багів. Зміна локації, додання нових взаємодій зі світом
	Необхідна кількість персоналу	Розробник
	Вимоги до вихідного коду та мов програмування	1)Мова програмування C# 2)Рушій Unity
	Вимоги до захисту інформації та додатків	1)ПЗ не використовує конфіденційну інформацію

Продовження таблиці 3.1

1	2	3
Техніко-економічні показники	Економічна Ефективність	Середня. Розробник отримуватиме прибуток за рахунок завантаження додатку клієнтами.
	Стратегічна цінність	Середня. Так як це перший проект розробника від успіху цього проекту залежить довіра до розробника.
Порядок контролю та приймання	Види тестування програмного забезпечення	UX/UI тестування Інсталяційне тестування Димне тестування Тестування продуктивності Функціональне тестування
	Загальні вимоги до прийняття	По завершенню розробки ПЗ ми маємо: Приємний usability інтерфейс для користувача; Вісім цікавих та захоплюючих рівнів, з різним рівнем складності; функціональна повнота не менше ніж 95%; відсутність багів серйозності S1, S2, S3.

**Висновки за розділом 3:**

- Побудовано технічне завдання до нашого ПЗ;
- Сформовано Вимоги до складу та параметри технічних засобів;
- Обрано стадії та етапи розробки;
- Сформовано умови експлуатації.

#### 4. ПЛАНУВАННЯ ТЕСТУВАННЯ

Обов'язковим елементом в реалізації технічного завдання має бути передбачено процес планування тестування, яке призначене для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись.

Під час планування тестування необхідно передбачити сценарії користувача. Для цього необхідно створити набір тест-кейсів, тобто кінцевого набору тестових випадків, за допомогою яких може бути протестована гра. Планування тестування виконується в вигляді тест-кейсів, які наведено у таблицях 4.1–4.5.

Назва:	Початок гри	
Функція:	NewGame	
Дія	Очікуваний результат	Результат тесту -пройдено -провалено -заблоковано
Передумова:		
Відкрити додаток	додаток відкрито	пройдено
Шаги тесту:		
Натиснути кнопку New Game	Ігровий рівень запущено	пройдено
Постумова		
Вийти до головного меню	Натиснуто esc і обраний пункт Save and exit	пройдено

Таблиця 4.1 – Тест-кейс 1

Назва:	Закупівля зброї у торговця	
Функція:	BuyGuns	
Дія	Очікуваний результат	Результат тесту -пройдено -провалено -заблоковано
Передумова:		
Відкрити додаток	додаток відкрито	пройдено
Натиснути кнопку New Game	кнопка натиснута, ігровий рівень запущено	пройдено
Шаги тесту:		
Підійти до торговця зброєю	До торговця підійшли	пройдено
Натиснути кнопку F та купити доступну зброю	зброю куплено	пройдено
Постумова		
Вийти до головного меню	Натиснуто esc і обраний пункт Save and exit	пройдено

Таблиця 4.2 – Тест-кейс 2

Назва:	Похід до зони війни	
Функція:	ZoneOfWar	
Дія	Очікуваний результат	Результат тесту -пройдено -провалено -заблоковано
Передумова:		
Відкрити додаток	додаток відкрито	пройдено
Натиснути кнопку Play	кнопка натиснута, ігровий рівень запущено	пройдено
Шаги тесту:		
Підійти до порталу, який веде до зони війни	До порталу підійдено	пройдено
Зайти до зони війни	У зону зайдено	пройдено
Постумова		
Вийти до головного меню	Натиснуто esc і обраний пункт Save and exit	пройдено

Таблиця 4.3 – Тест-кейс 3

Назва:	Постріли зі зброї	
Функція:	WeaponShoot	
Дія	Очікуваний результат	Результат тесту -пройдено -провалено -заблоковано
Передумова:		
Відкрити додаток	додаток відкрито	пройдено
Натиснути кнопку New Game	кнопка натиснута, ігровий рівень запущено	пройдено
Шаги тесту:		
Підійти до порталу, який веде до зони війни	До порталу підійдено	пройдено
Зайти до зони війни	У зону зайдено	пройдено
Натиснути ліву кнопку миші або зажати її	Кнопку натиснуто або зажато	пройдено
Побачити сліди від куль та ефекти пострілів	Ефекти та сліди пострілів побачено	пройдено
Постумова		
Вийти до головного меню	Натиснуто esc і обраний пункт Save and exit	пройдено

Таблиця 4.4 – Тест-кейс 4

Назва:	Кидання гранати	
Функція:	GrenadeOut	
Дія	Очікуваний результат	Результат тесту -пройдено -провалено -заблоковано
Передумова:		
Відкрити додаток	додаток відкрито	пройдено
Натиснути кнопку New Game	кнопка натиснута, ігровий рівень запущено	пройдено
Шаги тесту:		
Підійти до порталу, який веде до зони війни	До порталу підійдено	пройдено
Зайти до зони війни	У зону зайдено	пройдено
Натиснути кнопку G на клавіатурі	Кнопку натиснуто	пройдено
Побачити гранату та ефект від вибуху гранати	Гранату та ефекти побачено	провалено
Постумова		
Вийти до головного меню	Натиснуто esc і обраний пункт Save and exit	пройдено

Таблиця 4.5 – Тест кейс 5

## Висновки за розділом 4:

- спроектовано 4 тест-кейси для тестування ігрового паспортного додатку;
- усі тест-кейси у процесі тестування пройдені, окрім одного, тому необхідно сформулювати баг-репорт.

## 5. ВИКОНАННЯ ТЕСТУВАННЯ

Тестування програмного забезпечення – процес перевірки відповідності заявлених до продукта вимог і реально реалізованої функціональності, здійснюваний шляхом спостереження за його роботою в штучно створених ситуаціях і на обмеженому наборі тестів, обраних певним чином.



Баг-репорт – це документ, що описує ситуацію або послідовність дій, яка привела до некоректної роботи об'єкта тестування із зазначенням причин і очікуваного результату. Баг-репорт для провального тест-кейсу наведено у таблиці 5.1.

Таблиця 5.1 – Баг репорт 1

Стислий опис	При натисканні на кнопку G, граната не кидається
Проект	"Modern Soldier"
Компонент додатка	Кидання гранати
Номер версії	0.1
Важливість	Критична
Пріоритет	Високий
Статус	Виправлено
Автор	Сербін Ян
Призначено на	Сербін Ян
Шаги відтворення	1.Відкрити гру 2.Натиснути кнопку New Game 3.Знайти портал та зайти до нього 4.Натиснути на кнопку G 5.Перевірити кидання гранати 6.Вийти з гри



*Продовження таблиці 5.1*

Фактичний результат	 A screenshot from a first-person shooter game showing a soldier in a camouflage uniform and helmet standing in a flat, dark, featureless landscape under a clear blue sky. The soldier is facing away from the camera, looking towards the horizon.
Очікуваний результат	 A screenshot from the same game, showing the same soldier in the same landscape. However, a white grid-like object is floating in the air to the right of the soldier. Above the grid, the text 'DAMAGE: 0' is visible. The soldier is still facing away from the camera.

Пріоритет багу говорить про те, що виправлення обов'язкове, тому що гравець не може коректно грати у цю гру без її важливої механіки.

Баг було виправлено.

Висновки за розділом 5:

- сформовано баг-репорт на провальний тест-кейс;
- баг було виправлено.

## 6. ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

6.1. Засоби розробки програмного забезпечення У якості засобів розробки було обрано мову програмування C# та рушій Unity. C# (вимовляється Сі-шарп) — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research (належить Microsoft). Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функціїчлени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато від своїх попередників – мов C++, Object Pascal, Модуля і Smalltalk – C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем [4]. Unity – кроссплатформенна середовище розробки комп'ютерних ігор, розроблена американською компанією Unity Technologies. Unity дозволяє створювати програми, що працюють на більш ніж 25 різних платформах, що включають персональні комп'ютери, ігрові консолі, мобільні пристрої, інтернетпрограми та інші. Випуск Unity відбувся у 2005 році і з того часу триває постійний розвиток [5]. Основними перевагами Unity є наявність візуального середовища розробки, міжплатформної підтримки та модульної системи компонентів. До недоліків відносять появу складнощів при роботі з багатокомпонентними схемами та утруднення при підключенні зовнішніх бібліотек [6]. На Unity написані тисячі ігор, програм, візуалізації математичних моделей, які охоплюють безліч платформ і жанрів.

## 6.2. Інструкція користувача

На рисунку 6.1 представлено головне меню гри “Modern Soldier”.



Рисунок 6.1 – Головне меню

По натисканню кнопки Play буде надана можливість грати. По натисканню кнопки Exit – додаток закриється. По натисканню кнопки Settings – відкриється вікно регулювання звуку(Рисунок 6.2).

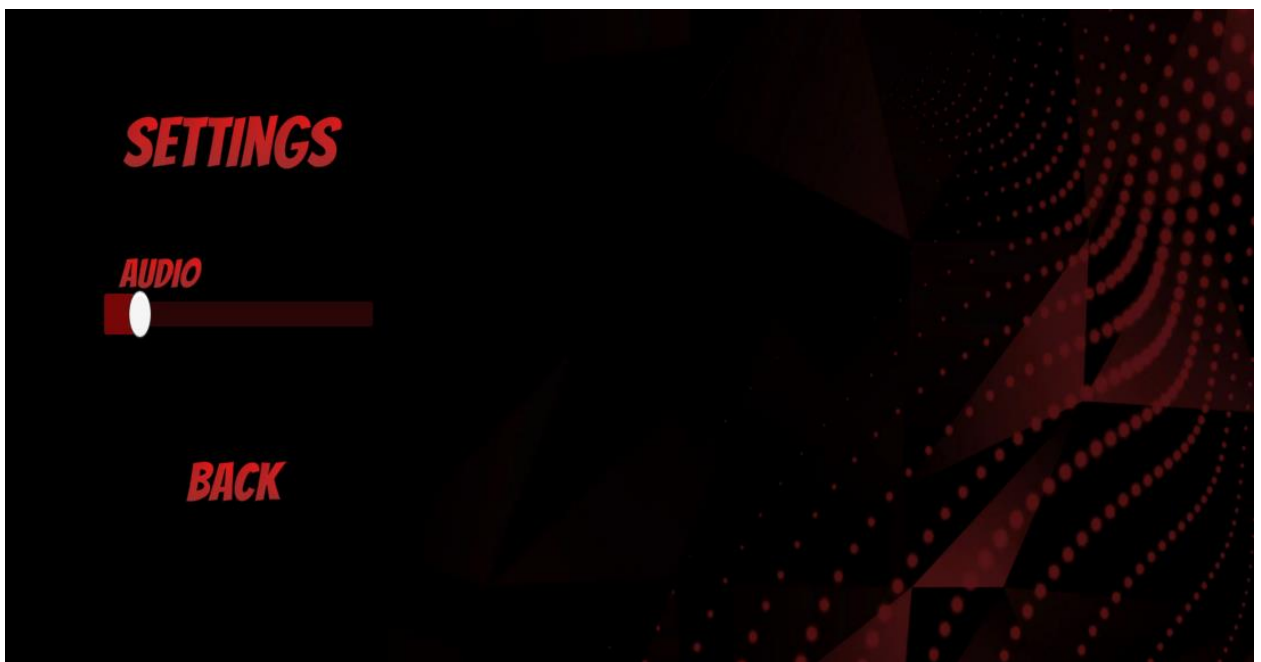


Рисунок 6.2 – Вікно регулювання звуку

Натиснувши кнопку BACK, ви повернетесь на попереднє меню.

В цій грі є 4 локації, де у першій локації вороги слабкіші у 4 рази ніж у 4 локації.

Натиснувши кнопку PLAY почнеться гра(Рисунок 6.3)



Рисунок 6.3 – Початок гри

Щоб поставити гру на паузу потрібно натиснути клавішу ESC(Рисунок 6.4)



Рисунок 6.4 – Вікно паузи

У цієї гри немає цілей, цю гру можна назвати пісочницею або РПГ, де тобі потрібно прокачати зброю для того, щоб швидше долати ворогів.

Торгівець, який продає зброю (Рисунок 6.5)



Рисунок 6.6 – Торгівець зброєю

Зброя, яку можна придбати помічається білим, а ту що не можна – сірим (Рисунок 6.7)



Рисунок 6.7 – Зброя, яку продає торгівець



При натисканні на зброю, яка підсвічується білим, вона покупається (Рисунок 6.8) та можна їй користуватись (Рисунок 6.9), для того, щоб переключитися на цю зброю потрібно натиснути клавішу 2, або прокрутити колесо миші угору, у кожній зброї різна кількість шкоди, різна скорострільність, різна кількість боєприпасів, звук, тощо.



Рисунок 6.8 – Придбання зброї



Рисунок 6.9 – Придбана зброя

Також якщо натиснути клавішу Т можна включити вигляд від 1 обличчя (Рисунок 6.10)

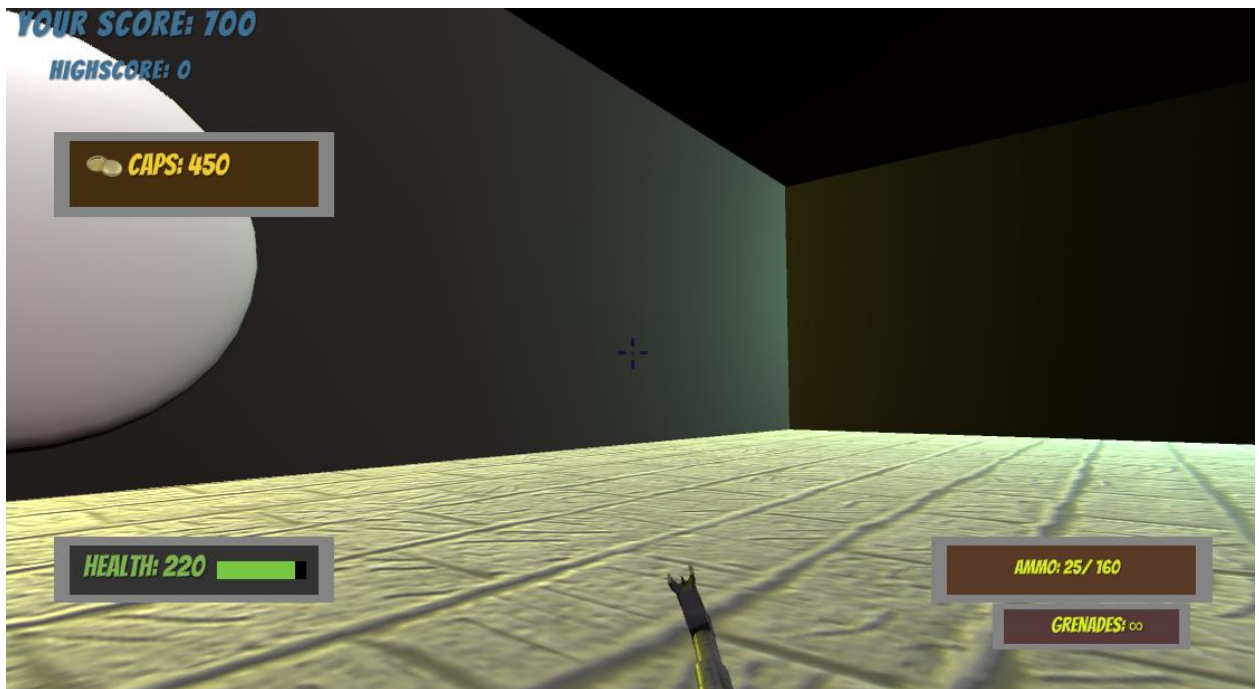


Рисунок 6.10 – Вид від 1 обличчя

Якщо натиснути праву кнопку миші, то можна прицілитись (Рисунок 6.11)

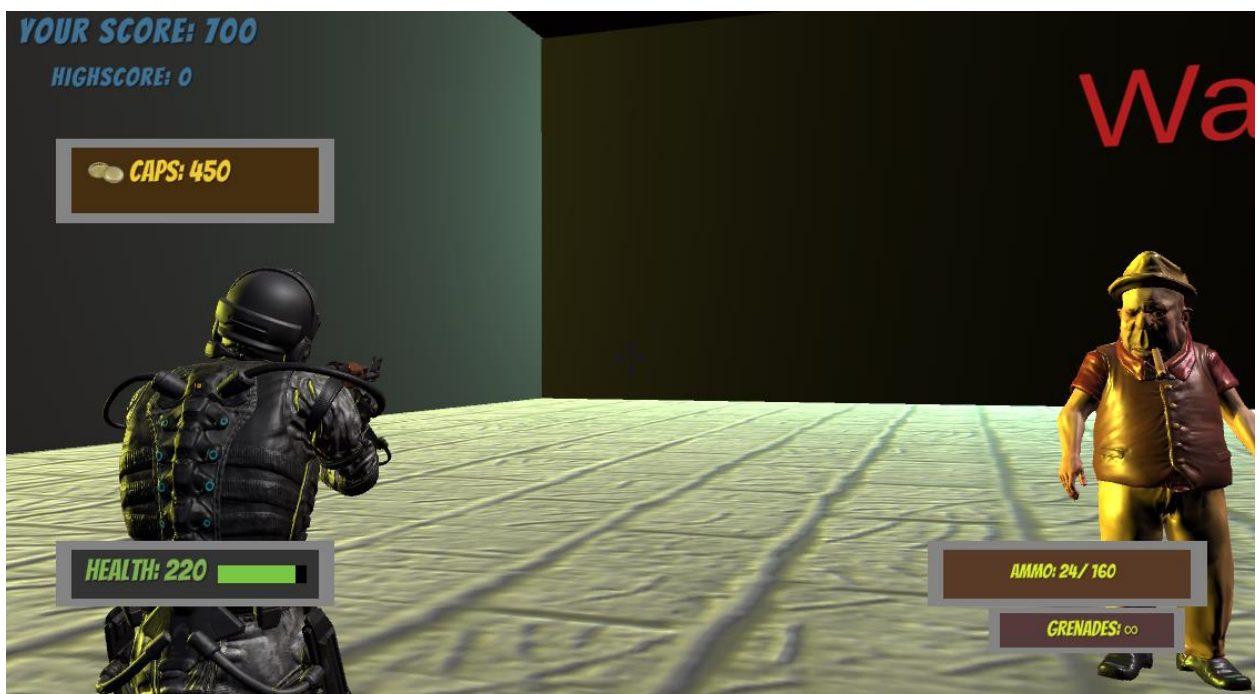


Рисунок 6.11 – Прицілювання



Якщо зажати клавішу G, то персонаж буде користуватись гранатою, та буде відображений інтерфейс польоту гранати (Рисунок 6.12), якщо відпусти цю клавішу, персонаж кине гранату (Рисунок 6.13) та вона підірветься (Рисунок 6.14)



Рисунок 6.12 – Інтерфейс польоту гранати



Рисунок 6.13 – Кидання гранати





Рисунок 6.14 – Ефект вибуху гранати

Також є зони де знаходяться вороги різної ступені важкості(Рисунок 6.15)



Рисунок 6.15 – Зображення порталу для входу у зону війни

Коли персонаж зайде у цей портал, то на нього відразу нападуть вороги (Рисунок 6.16), коли персонаж їх здолає, то з них випадуть патрони або аптечка з деяким шансом, а також дадуть кришки та кількість очок збільшиться (Рисунок 6.17)



Рисунок 6.16 – Вороги



Рисунок 6.17 – Дроп з ворогів

## Код дропу з ворогів (Рисунок 6.18)

```

class Z
{
    void Die()
    {
        _sombieDeathStatus = true;

        if (OnZombieDeath != null)
        {
            OnZombieDeath();
        }

        SpawnScript.currentZombies--;
        _ragdollManager.TriggerRagdoll();
        Debug.Log("Zombie Death");
        _animator.enabled = false;
        enabled = false;
        Destroy(gameObject, 6f);
    }
}

class Z
{
    void Drop()
    {
        MoneyScript._playerMoney += _sombieMoneyDrop;
        ScoreScript.Score += _sombieScoreGive;
        System.Random rnd = new();
        int number = rnd.Next(0, 3);
        if (number <= _dropList.Length - 1)
        {
            switch (number)
            {
                case 0:
                    GameObject copy1 = Instantiate(_dropList[number], transform);
                    _destination.transform.position = transform.position;
                    copy1.transform.SetParent(_destination.transform);
                    break;
                case 1:
                    GameObject copy2 = Instantiate(_dropList[number], transform);
                    _destination.transform.position = transform.position;
                    copy2.transform.SetParent(_destination.transform);
                    break;
            }
        }
        else
        {
            Debug.Log("Zombie Dropped nothing");
        }
    }
}

```

Рисунок 6.18 – Код дропу ворогів та смерті зомбі

Вороги респавняться через деякий час, тому на них можна “Нафармити” наступну зброю, з якої можна долати наступних ворогів (Рисунок 6.19).

```

using System.Collections;
using UnityEngine;

// Скрипт Unity (ссылка на ресурсы: 4) | Ссылка: 1
public class SpawnScript : MonoBehaviour
{
    [SerializeField] GameObject zombiePrefab;
    [SerializeField] Transform spawnPoint;
    [SerializeField] float spawnRadius = 5f;
    [SerializeField] int maxZombies = 20;
    [SerializeField] int spawnDelay = 10;

    public static int currentZombies = 0;

    // Сообщение Unity | Ссылка: 0
    void Start()
    {
        StartCoroutine(SpawnZombies(maxZombies));
    }

    // Ссылка: 1
    IEnumerator SpawnZombies(int count)
    {
        while (currentZombies < count)
        {
            SpawnZombie();
            yield return new WaitForSeconds(spawnDelay);
        }
    }

    // Ссылка: 2
    void SpawnZombie()
    {
        Vector3 randomSpawnPosition = spawnPoint.position + Random.insideUnitSphere * spawnRadius;
        Quaternion spawnRotation = Quaternion.Euler(0, Random.Range(0, 360), 0);
        GameObject newZombie = Instantiate(zombiePrefab, randomSpawnPosition, spawnRotation);
        Ch1ZombieMeleeController zombieController = newZombie.GetComponent<Ch1ZombieMeleeController>();
        zombieController.OnZombieDeath += OnZombieDeath;

        newZombie.SetActive(true);
        newZombie.transform.localScale = zombiePrefab.transform.localScale;
        currentZombies++;
    }

    // Ссылка: 1
    void OnZombieDeath()
    {
        currentZombies--;
        SpawnZombie();
    }
}

```

Рисунок 6.19 – Код спавну зомбі

## Висновки за розділом 6:

- розглянуто засоби розробки десктопного програмного забезпечення;
- представлено інструкцію користувача з вказуванням основних екранів та залежності коду і отриманого візуального результату

## 7. ВИСНОВКИ

Десктопні додатки є одним з найрозповсюджених додатків на сьогоднішній день. Це, насамперед, пов'язано із тим, що ПК для кожної людини є як машина для навчання так і просто машина для відпочинку або прогання часу. У ході виконання курсової роботи було розроблено 3D шутер -аркаду «Modern Soldier» для того щоб користувач міг чимось себе зайняти на деякий час. Додаток має такий функціонал: стрільба; ходьба, стрибок, присідання, кидання гранати, торгівля; Було проведено тестування додатку, виявлений баг, який був виправлений. Надано інструкцію користувача із описом функціональності додатку. Подальша розробка може бути спрямована на розширення різноманітності ворогів, додавання нової зброї, впровадження системи скилів, змінення територій, зміни інтерфейсу та урізноманітнення функціоналу.

## 8. СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] Документація з Unity

<https://docs.unity3d.com/ru/530/Manual/UnityManual.html>

[2] Герберт Шилд C#4.0. <https://readli.net/c-4-0-polnoe-rukovodstvo/>

[3] Grenade throwing in Unity Full Guide

<https://www.youtube.com/watch?v=e2c00JsIe44&pp=ygUTZ3JhbmdFkZSB0aHJvdYB1bml0eQ%3D%3D>

[4] Projectile Trajectory <https://www.youtube.com/watch?v=8mGZBYsSXCQ>

[5] Playlist in youtube about how to create spawnSystem in Unity (Not working today) <https://www.youtube.com/watch?v=6Xs1mkutG2o&list=PLA-xaldQ72ryGL-DyIGasa0qa6mIMcic6&index=42>